สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาเกมที่เรียนรู้ได้

DEVELOPING GAME THAT CAN LEARN

T119268

เจษฎา เจริญกิจวรวุฒิ
ชนินทร์ ภักดีธรรมะสกุล
นวกานต์ ประเสริฐเจริญกิจ

เลขหมู่.....
ลงทะเบียน 119268
วัน,เดือน,ปี - 6 S.A. 2554

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาเกมที่เรียนรู้ได้

DEVELOPING GAME THAT CAN LEARN

ผู้จัดทำ

1. นายเจษฎา เจริญกิจวรวุฒิ รหัสนักศึกษา 50010268
2. นายชนินทร์ ภักดีธรรมะสกุล รหัสนักศึกษา 50010302
3. นายนวกานต์ ประเสริฐเจริญกิจ รหัสนักศึกษา 50010778

_____ อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.เกียรติกูล เจียรนัยธนะกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเกมที่เรียนรู้ได้

นายเจษฎา	เจริญกิจวรวุฒิ	50010268
นายชนินทร์	ภักดีธรรมะสกุล	50010302
นายนวกานต์	ประเสริฐเจริญกิจ	50010778
รศ. ดร.เกียรติกูล	เจียรนัยธนะกิจ	อาจารย์ที่ปรึกษา

ปีการศึกษา 2553

บทคัดย่อ

อุปนิสัยอันชาญฉลาดของ NPC (Non-Player Character) เป็นสิ่งสำคัญของเกมประเภท RTS (Real-Time Strategy) ทำให้มีการพัฒนาวิธีการด้าน AI (Artificail Intelligence) มากมายเพื่อสร้างอุปนิสัยอันชาญฉลาดนี้แต่ส่วนใหญ่มักไม่มีประสิทธิภาพเพียงพอเพราะเกมประเภท RTS นั้นยากต่อการเก็บชุดข้อมูลอินพุทเอาท์พุทที่ถูกต้องเพื่อใช้ในการเรียนรู้แต่กลับง่ายในการวัดประสิทธิภาพของ NPC หลังจบเกม วิธีการที่กล่าวได้ว่าประสบความสำเร็จมากที่สุดในด้านนี้คือ NEAT (NeoruEvolution of Augmenting Topologies) ซึ่งเป็นวิธีที่ใช้ GA (Genetic Algorithm) สำหรับวิวัฒน์น้ำหนักและโครงสร้างของ ANN (Artificial Neural Network) เราได้ศึกษาเกี่ยวกับขั้นตอนวิธีการค้นหาแบบเอสตาร์, GA, ANN, และ NEAT จากนั้นเราได้พัฒนาเกมประเภท RTS ที่มีชื่อว่า "เทศกาลสงกรานต์: สแปลชอาวร์" ซึ่งเป็นเกมที่มีการใช้ ANN ที่ผ่านการเรียนรู้แบบออฟไลน์มาโดย NEAT ร่วมกับรูปแบบอินพุทเอาท์พุทและขั้นตอนการเรียนรู้ที่เราได้ออกแบบไว้ นอกจากนี้เรายังใช้ขั้นตอนวิธีการค้นหาแบบเอสตาร์ในการแก้ปัญหาการค้นหาเส้นทางของตัวละครภายในเกมทั้งแบบเดี่ยวและแบบกลุ่มซึ่งถือเป็นปัญหาใหญ่อีกเรื่องหนึ่งของเกมประเภท RTS เราได้ทำการทดลองสร้าง ANN ขึ้นด้วยวิธีการเรียนรู้ที่เราออกแบบไว้เพื่อแสดงให้เห็นถึงประสิทธิภาพและประโยชน์ของ NEAT ที่มีต่อเกมประเภท RTS ที่มีความซับซ้อนคล้ายๆ กับเกมที่เราได้พัฒนาขึ้น เราได้ใส่ผลลัพธ์ที่ได้จากการเรียนรู้ลงไปในตัวเกมและเผยแพร่เกมออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVELOPING GAME THAT CAN LEARN

Mr. Jessada	Charoenkitvorawut	50010268
Mr. Chanin	Pakdeethammasakul	50010302
Mr. Nawakan	Prasertcharoenkit	50010778
Assoc.Prof.Dr.Kietikul	Jearanaitanakij	Advisor

Academic Year 2010

ABSTRACT

Intelligence behavior of NPC (Non-Player Character) is the impact in RTS (Real-Time Strategy) game. Many AI (Artificial Intelligence) techniques have been developed for creating it, but most of them are not efficient because RTS game is difficult or impossible to create a syllabus of correct input-output pairs but easy to measure NPC performance at the end. One of the most successful is NEAT (NeoruEvolution of Augmenting Topologies) which is a GA (Genetic Algorithm) for the generation of evolving both weighting parameters and structures of ANN (Artificial Neural Network). We learned about A* Search, GA, ANN, and NEAT. We developed RTS game "Songkran Festival: Splash Hour". It uses ANN which was offline learned by using NEAT along with our designed input and output nodes for ANN and learning flow. We also used A* Search algorithm to get us through single and group pathfinding problem which is the other impact in RTS game. We created ANN with our learning flow to show NEAT's performance for complex RTS game likes ours. We integrated ANN into our game and finally released it.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะเสร็จสมบูรณ์ด้วยดีไม่ได้ถ้าขาดคำปรึกษาและการชี้แนะสิ่งต่างๆ ที่เป็นประโยชน์จาก รศ. ดร.เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา

ขอขอบคุณคณาจารย์ทุกท่านในสาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ประสิทธิ์ประสาทวิชาความรู้แก่ข้าพเจ้า รวมไปถึงคำปรึกษาต่างๆ ที่ให้แก่ข้าพเจ้า

ขอขอบคุณสาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้สนับสนุนไฟฟ้าและอินเตอร์เน็ต รวมถึงปริญญานิพนธ์ของรุ่นพี่สำหรับเป็นแหล่งศึกษาเพิ่มเติม

ขอกราบขอบพระคุณครอบครัวของผู้จัดทำทุกคนที่เข้าใจและคอยให้กำลังใจเสมอมา

ขอขอบคุณนางสาวทักษณา พิริยะกุลที่ให้ความช่วยเหลือทางด้านองค์ประกอบศิลป์

ขอขอบคุณเพื่อนๆ ทุกคนที่ให้ความร่วมมือทดสอบเกมของข้าพเจ้า พร้อมทั้งให้ความเห็นต่างๆ ที่เป็นประโยชน์ต่อโครงงานนี้

ขอบคุณพี่เจษฎ์ สิทธิดำรงที่ให้คำปรึกษาในการใช้เกมเอนจิ้น และโปรแกรมปั้นโมเดลต่างๆ

ขอบคุณพี่ๆ บริษัท ดีบัสซ์ จำกัด ที่ให้คำปรึกษาในการทำโครงงานตลอดมา

เจษฎา เจริญกิจวรวุฒิ
ชนินทร์ ภักดีธรรมะสกุล
นวกานต์ ประเสริฐเจริญกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญภาพ VII

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินการ 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

1.6 ส่วนประกอบของปริญญานิพนธ์ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 5

2.1 ขั้นตอนวิธีการค้นหาแบบเอสตาร์ 5

2.2 ขั้นตอนวิธีเชิงพันธุกรรม 8

2.3 โครงข่ายประสาทเทียม 28

2.4 NeuroEvolution of Augmenting Topologies (NEAT) 37

2.5 การใช้งาน Blender 46

2.6 การใช้งาน Unity3D 53

บทที่ 3 การออกแบบ 67

3.1 แนวคิดของเกม 67

3.2 การออกแบบเกม 68

3.3 การออกแบบโครงสร้างและขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียม 88

3.4 การประยุกต์ใช้ปัญญาประดิษฐ์กับเกม 98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง 106

4.1 โมเดล 106

4.2 การหาเส้นทางและเดินหลบสิ่งกีดขวาง 108

4.3 การใช้ NEAT แก้ปัญหาตรรกศาสตร์ XOR 111

4.4 การใช้ NEAT แก้ปัญหา DPV (Double Pole Balancing with Velocities) 113

4.5 การใช้ NEAT สร้าง ANN 115

4.6 การสร้างเกมด้วย Unity3D 117

บทที่ 5 บทสรุป 119

5.1 บทสรุป 119

5.2 ปัญหาอุปสรรคและแนวทางในการแก้ไข 119

5.3 ผลการดำเนินงาน 120

5.4 แนวทางในการพัฒนาต่อ 121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

2.1 ระยะห่างระหว่างเมืองต่างๆ กับเมืองบูคาเรสต์เมื่อวัดเป็นเส้นตรง 7

2.2 ตัวอย่างการเข้ารหัสที่ใช้กับปัญหา $f(15x - x^2)$ 18

2.3 ตัวอย่างตารางค่า fitness 19

2.4 ตำแหน่งที่ตั้งของเมืองต่างๆ 22

2.5 เปรียบเทียบความสามารถของเกมเอนจิ้น 53

3.1 พลังและความสามารถของตัวละครแต่ละประเภท 70

3.2 พลังและความสามารถของอาวุธแต่ละประเภท 70

3.3 อินพุทที่เป็นจำนวนไม่ต่อเนื่องของโครงข่ายประสาทเทียม 89

3.4 อินพุทที่มาจากตัวละครในทีมของโครงข่ายประสาทเทียม 89

3.5 อินพุทที่มาจากตัวละครทีมอื่นของโครงข่ายประสาทเทียม 90

3.6 เอาท์พุทของโครงข่ายประสาทเทียมของตัวละครแต่ละประเภท 91

4.1 เวลาที่ใช้ในการวิ่งแต่ละรอบ 109

4.2 เวลาที่ใช้ในการวิ่งหลบหลีกสิ่งกีดขวางแต่ละรอบ 111

4.3 คำตอบของตรรกศาสตร์ XOR 112

4.4 ตัวแปรที่ใช้ในการทดลองแก้ปัญหาตรรกศาสตร์ XOR 112

4.5 ผลการทดลองการแก้ปัญหาตรรกศาสตร์ XOR 113

4.6 ตัวแปรที่ใช้ในการทดลองแก้ปัญหา DPV 114

4.7 โหนดของโครงข่ายประสาทเทียมที่ใช้ในการทดลองแก้ปัญหา DPV 114

4.8 ผลการทดลองแก้ปัญหา DPV 115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป หน้า

2.1 ที่ตั้งเมืองต่างๆ และระยะทางระหว่างเมืองในประเทศโรมาเนีย 6

2.2 การท่องโหนดในต้นไม้ของกระบวนการเอสตาร์ 8

2.3 ตัวอย่างการเข้ารหัสแบบเลขฐานสอง 9

2.4 ตัวอย่างการเข้ารหัสแบบเรียงสับเปลี่ยน 10

2.5 ตัวอย่างการเข้ารหัสแบบแทนค่าข้อมูล 10

2.6 ตัวอย่างการเข้ารหัสแบบต้นไม้ 11

2.7 ตัวอย่างการคัดเลือกแบบวงล้อรูเรทท์ 12

2.8 ตัวอย่างก่อนผ่านขั้นตอนและหลังผ่านขั้นตอนการคัดเลือกโดยการจัดอันดับ 13

2.9 ตัวอย่างการครอสโอเวอร์ 14

2.10 ตัวอย่างมิวเทชัน 15

2.11 ขั้นตอนพื้นฐานของขั้นตอนวิธีเชิงพันธุกรรม 17

2.12 ตัวอย่างการคัดเลือกแบบวงล้อรูเลทท์ที่ใช้กับปัญหา $f(15x - x^2)$ 20

2.13 ตัวอย่างขั้นตอนของขั้นตอนวิธีเชิงพันธุกรรมที่ใช้กับปัญหา $f(15x - x^2)$ 21

2.14 เปรียบเทียบผลลัพธ์จากตัวอย่างระหว่างประชากรรุ่นแรกและรุ่นสุดท้าย 22

2.15 จำลองตำแหน่งที่ตั้งของเมืองต่างๆ 23

2.16 ตัวอย่างการเข้ารหัสที่ใช้กับ TSP 24

2.17 โครโมโซมที่ถูกสุ่มขึ้นมาเป็นประชากรรุ่นแรก 24

2.18 ค่าความเหมาะสมของโครโมโซมแต่ละตัว 25

2.19 ผลลัพธ์หลังจากนำโครโมโซมใส่ลงในวงล้อรูเลทท์ 25

2.20 การครอสโอเวอร์ที่ใช้กับ TSP 26

2.21 มิวเทชันที่ใช้กับ TSP 27

2.22 population ในรุ่นต่อไป 27

2.23 โครงข่ายประสาทในสิ่งมีชีวิต 28

2.24 โครงข่ายประสาทเทียม 30

2.25 เอาท์พุทที่ได้จากนิวรอน 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
2.26 โครงสร้างทั่วไปของโครงข่ายประสาทเทียม	31
2.27 กราฟฟังก์ชันโลจิสติก	31
2.28 การเรียนรู้แบบมีผู้สอน	32
2.29 การเรียนรู้แบบไม่มีผู้สอน	33
2.30 โครงข่ายประสาทเทียมหลายชั้น	34
2.31 โครงข่ายประสาทเทียมสามชั้นที่ใช้แก้ตรรกศาสตร์ XOR	35
2.32 โครงข่ายประสาทเทียมสามชั้นที่ใช้แก้ตรรกศาสตร์ XOR หลังการเรียนรู้รอบสุดท้าย	36
2.33 กระบวนการเบื้องต้นของของ NEAT	38
2.34 การเข้ารหัสใน NEAT	39
2.35 ครอสโอเวอร์ของ NEAT	40
2.36 ปัญหาคอมเพตติงคอนเวนชัน	41
2.37 มิวเทชันเพิ่มการเชื่อมต่อ	42
2.38 มิวเทชันเพิ่มโหนด	42
2.39 กระบวนการทั้งหมดของ NEAT	45
2.40 หน้าต่างการทำงานทั่วไปของโปรแกรม Blender	47
2.41 เมนูสร้างโมเดลต้นแบบ	48
2.42 วัตถุที่ถูกสร้าง	48
2.43 โมเดลที่ถูกเลือก	49
2.44 คำสั่งด้านล่าง View Port	49
2.45 วัตถุที่เข้าสู่ Edit Mode	49
2.46 เลือกจุด	50
2.47 เลือกเส้น	50
2.48 เลือกหน้าตัด	50
2.49 การ Extrude	51
2.50 การ Subsurf	51
2.51 การ Subdivide กับหน้าตัด	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
2.52 การ Subdivide กับเส้น	52
2.53 การใช้ Knife	53
2.54 หน้าตาของตัวโปรแกรม Unity3D	54
2.55 หน้าต่าง Project	55
2.56 การ Create Assets	55
2.57 หน้าต่าง Hierarchy	56
2.58 เครื่องมือต่างๆ ใน Tool Bar	56
2.59 หน้าต่าง Scene View	57
2.60 หน้าต่าง Game View	57
2.61 หน้าต่าง Inspector	58
2.62 การสร้าง Empty GameObject	58
2.63 GameObject ที่ได้และที่เปลี่ยนชื่อ	59
2.64 การสร้าง C# Script ขึ้นมาเพื่อใช้งาน	59
2.65 การแก้ชื่อไฟล์	60
2.66 หน้าต่างที่ใช้สำหรับการแก้ Script	60
2.67 ตัวอย่าง Script	61
2.68 การใส่ Script ลงไปใน GameObject	62
2.69 Log ที่ขึ้นข้างใต้ของ Game View หรือ Scene View	62
2.70 หน้าต่างที่แสดงขึ้นมาหลังดับเบิ้ลคลิก ผลลัพธ์ของ Script	63
2.71 หน้าต่าง Inspector ที่ใช้ปรับแต่งโมเดล	64
2.72 ปรับแต่งรายละเอียดของ Character Controller	66
3.1 ส่วนประกอบและสัดส่วนของแผนที่	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
3.2 เมนูทรีของเกม	74
3.3 ตัวละคร Raincoater	75
3.4 ตัวละคร Powder Smearer	75
3.5 ตัวละคร Watergunslinger	76
3.6 ตัวละคร Water Bomber	76
3.7 ตัวละคร Caretaker	77
3.8 ตัวละคร Summoner	77
3.9 หน้าจอ Main Menu	78
3.10 หน้าจอ Character Selection	79
3.11 หน้าจอ Game	79
3.12 ยูสเคสไดอะแกรมของเกม	80
3.13 ยูสเคสเดสคริปชันสำหรับยูสเคส Move Camera	81
3.14 ยูสเคสเดสคริปชันสำหรับยูสเคส Rotate Camera	81
3.15 ยูสเคสเดสคริปชันสำหรับยูสเคส Select Character	82
3.16 ยูสเคสเดสคริปชันสำหรับยูสเคส Select Group of Character	82
3.17 ยูสเคสเดสคริปชันสำหรับยูสเคส Move	83
3.18 ยูสเคสเดสคริปชันสำหรับยูสเคส Attack	83
3.19 ยูสเคสเดสคริปชันสำหรับยูสเคส Use Skill	84
3.20 ยูสเคสเดสคริปชันสำหรับยูสเคส Get Item	84
3.21 คลาสไดอะแกรมใน Problem domain	85
3.22 Statechart ของ Character	87
3.23 สั่งให้ตัวละครเดินหาเป้าหมาย	91
3.24 สั่งให้ตัวละครหลบหนีเอาตัวรอด	92
3.25 สั่งให้ตัวละครเดินไปเก็บกล่อง	92
3.26 สั่งให้ตัวละครโจมตี	92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป หน้า

3.27 สั่งให้ตัวละคร Raincoater ใช้ความสามารถ Bunker 93
3.28 สั่งให้ตัวละคร Powder Smearer ใช้ความสามารถ Dash 93
3.29 สั่งให้ตัวละคร Powder Smearer ใช้ความสามารถ Lake of powder 94
3.30 สั่งให้ตัวละคร Water Bomber ใช้ความสามารถ Hareruya 94
3.31 สั่งให้ตัวละคร Caretaker ใช้ความสามารถ Towel 94
3.32 สั่งให้ตัวละคร Caretaker ใช้ความสามารถ Hairdryer 95
3.33 สั่งให้ตัวละคร Caretaker ใช้ความสามารถ Giant electric fan 95
3.34 สั่งให้ตัวละคร Charmer ใช้ความสามารถ Motor-tricycle 95
3.35 สั่งให้ตัวละคร Charmer ใช้ความสามารถ Vanz boy 'N' Skoyz girl 96
3.36 สั่งให้ตัวละคร Charmer ใช้ความสามารถ Pick-up truck 96
3.37 สั่งให้ตัวละคร Charmer ใช้ความสามารถ Fire apparatus 96
3.38 ขั้นตอนการเรียนรู้ 97
3.39 สมการค่าความเหมาะสมของตัวละครชนิดต่างๆ 98
3.40 โหนดที่มุมสิ่งกีดขวาง 99
3.41 Flowchart แสดงการทำงานของวิธีการหลบหลีก 101
3.42 แสดงทิศทางที่ใช้ในการตรวจสอบการชนและเส้นทางการหลบหลีกของตัวละคร 102
3.43 ปัญหาที่เกิดขึ้นในกรณีที่ตัวละครวิ่งเข้าไปติดในช่อง 104
4.1 โมเดลต้นแบบ Watergunslinger 106
4.2 โมเดลต้นแบบของ Power Smearer 106
4.3 โมเดลต้นแบบของ Raincoater 107
4.4 โมเดลต้นแบบของ Water Bomber 107
4.5 โมเดลต้นแบบของ Healer 107
4.6 โมเดลต้นแบบของ Summoner 108
4.7 เขาวงกตจำลองที่ใช้สำหรับการทดลองขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์ 108
4.8 ขณะทดลองขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์ 109
4.9 เส้นทางที่ตัวละครเลือกในการทดลองขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์ 110
4.10 สิ่งกีดขวางสำหรับการทดลองการวิ่งหลบหลีก 110
4.11 ขณะทดลองการเดินหลบหลีก 111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป หน้า

4.12 ปัญหา DPV 113
4.13 จำนวนครั้งที่ชนะ 115
4.14 คะแนนที่ได้ (คิดจากอันดับที่ได้ในแต่ละรอบ) 116
4.15 ความเปียกที่ทำได้ในแต่ละรอบ 116
4.16 โครงสร้าง ANN ของ Tank 116
4.17 โครงสร้าง ANN ของ Summoner 117
4.18 หน้าจอเลือกตัวละคร 117
4.19 กล่องบรรจุทรัพยากร 118
4.20 สภาพภายในเกม 118
4.21 ค่าความเปียกที่แต่ละทีมทำได้ 118
5.1 แผนการดำเนินงานด้านปัญญาประดิษฐ์ 120
5.2 แผนการดำเนินงานด้านตัวเกม 120
5.3 แผนการดำเนินงานด้านองค์ประกอบศิลป์ 121
5.4 แผนการดำเนินงานด้านเสียง 121
5.5 แผนการดำเนินงานภายหลังการพัฒนาองค์ประกอบต่างๆ ของเกม 121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

การพัฒนาเกมประเภท RTS (Real-Time Strategy) แบบวางแผนการรบ (War game) เป็นเกมที่มักจะจำลองสงครามที่แฝงความรุนแรง ผู้เล่นจะต้องบริหารทรัพยากรต่างๆ รวมทั้งวางแผนสั่งการตัวละครต่างๆ ของตนเอง จึงเป็นเกมที่จำเป็นต้องนำเอาปัญญาประดิษฐ์เข้ามาประยุกต์ใช้เพื่อช่วยในเรื่องของการค้นหาเส้นทางในการเดินทางของตัวละครไปยังตำแหน่งที่ต้องการ และการตัดสินใจใช้แผนการต่างๆ ของ NPC (Non Player Character) เพื่อเข้าต่อสู้กับศัตรูได้อย่างมีประสิทธิภาพ

การแก้ปัญหาเกี่ยวกับการค้นหาเส้นทางด้วยขั้นตอนวิธีแบบเอสตาร์เป็นขั้นตอนวิธีที่ถูกใช้อย่างแพร่หลายในเกมประเภท RTS แต่รายละเอียดการนำไปประยุกต์ใช้จริงๆ นั้นจะแตกต่างกันออกไปตามลักษณะของเกมทั้งในเรื่องของสภาพแวดล้อมของเกม, เวลาที่จำกัดสำหรับใช้ในการประมวลผล, การสั่งการตัวละครแบบกลุ่ม, และการหลบหลีกสิ่งกีดขวางรวมถึงการหลบหลีกกันเองระหว่างตัวละคร

การตัดสินใจใช้แผนการต่างๆ ของ NPC (Non Player Character) เพื่อเข้าต่อสู้กับศัตรูเป็นการตัดสินใจที่มีความซับซ้อนเนื่องจากมีปัจจัยหลายๆ อย่างที่ต้องพิจารณา เช่น สถานะของตัวละคร, ตำแหน่งของตัวละคร, ความสามารถของตัวละคร, สภาพแวดล้อมของตัวละคร, และจำนวนทรัพยากรที่มีอยู่ เป็นต้น การตัดสินใจเหล่านี้มักไม่รู้ผลในทันทีว่าเป็นการตัดสินใจที่ดีหรือไม่ เพราะบางครั้งการตัดสินใจที่ทำให้เสียเปรียบ อาจทำให้ได้เปรียบในภายหลังได้

โครงงานนี้เป็นการพัฒนาเกมมหาสงกรานต์: สแปลชอาวร์ ซึ่งเป็นเกมประเภท RTS แบบวางแผนการรบที่นำเสนอการเล่นน้ำสงกรานต์แทนการทำสงครามที่มักพบในเกมประเภทนี้เพื่อลดความรุนแรงและเปลี่ยนรูปแบบของเกมไปในทางที่สร้างสรรค์ขึ้น โดยในส่วนของปัญญาประดิษฐ์ได้มีการนำขั้นตอนวิธีแบบเอสตาร์มาใช้แก้ปัญหาเกี่ยวกับการค้นหาเส้นทาง มีการแก้ปัญหาการหลบหลีกสิ่งกีดขวางรวมถึงการหลบหลีกกันเองระหว่างตัวละครรวมทั้งการแปลงสภาพแวดล้อมของเกมเพื่อใช้ในขั้นตอนวิธีแบบเอสตาร์ได้อย่างมีประสิทธิภาพ นอกจากนี้ยังมีการนำเอาโครงข่ายประสาทเทียมซึ่งใช้การเรียนรู้แบบออฟไลน์ด้วย NEAT (NeuroEvolution of Augmenting Topologies) มาช่วยในการตัดสินใจใช้แผนการต่างๆ ของ NPC เนื่องจากเป็นวิธีการเรียนรู้ที่มีการวัดประสิทธิภาพในตอนจบเกมจึงเหมาะกับการตัดสินใจที่ไม่รู้ผลในทันทีของเกมประเภทนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อปรับเปลี่ยนรูปแบบของเกมประเภท RTS (Real-Time Strategy) แบบวางแผนการรบ (War game) ไปในทางที่สร้างสรรค์ขึ้น
2) เพื่อนำความรู้ความเข้าใจที่เกี่ยวกับความหมาย, หลักการ, และขั้นตอนการทำงานของขั้นตอนวิธีการค้นหาแบบเอสตาร์มาประยุกต์ใช้ในการทำเกมประเภท RTS แบบวางแผนการรบ
3) เพื่อนำความรู้ความเข้าใจที่เกี่ยวกับความหมาย, หลักการ, และขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรม ไปประยุกต์ใช้กับ NEAT (NeuroEvolution of Augmenting Topologies)
4) เพื่อนำความรู้ความเข้าใจที่เกี่ยวกับความหมาย, หลักการ, และขั้นตอนการทำงานของโครงข่ายประสาทเทียม ไปประยุกต์ใช้กับ NEAT
5) เพื่อนำความรู้ความเข้าใจที่เกี่ยวกับความหมาย, หลักการ, และขั้นตอนการทำงานของ NEAT มาใช้ในการสอนโครงข่ายประสาทเทียม เพื่อใช้ในการตัดสินใจของ NPC (Non Player Character) ในเกมประเภท RTS แบบวางแผนการรบ

1.3 ขอบเขตของโครงงาน

โครงงานการพัฒนาเกมที่เรียนรู้ได้เป็นการพัฒนาเกมประเภท RTS (Real-Time Strategy) แบบวางแผนการรบ (War game) ที่มีชื่อว่า เทศกาลสงกรานต์: สแปลชอาวร์ โดยที่ NPC (Non Player Character) ภายในเกมนี้สามารถหาเส้นทางเพื่อเดินไปยังจุดหมายที่ต้องการได้โดยมีการหลบสิ่งกีดขวางและหลบหลีกกันเองได้ด้วย นอกจากนี้ NPC ยังสามารถตัดสินใจใช้แผนการต่อสู้ต่างๆ ตามสภาพแวดล้อมภายในเกมขณะนั้น เพื่อต่อสู้กับผู้เล่นได้อย่างชาญฉลาด โดยทำการสร้างโมเดลที่ใช้ในเกมด้วย Blender จากนั้นใช้ Unity3D เป็นเกมเอนจินในการเขียนตัวเกมหลักขึ้นมา จากนั้นเขียนโปรแกรมส่วนของการหาเส้นทางด้วยภาษา C# เพื่อใช้เป็นสคริปต์สำหรับ Unity3D จากนั้นเขียนอีกโปรแกรมเป็นส่วนของขั้นตอนการเรียนรู้แบบออฟไลน์ขึ้นมา ทำการเรียนรู้จนได้โครงข่ายประสาทเทียมที่เป็นผลลัพธ์ออกมาแล้วนำไปใช้กับ NPC ในตัวเกมหลัก

1.4 วิธีการดำเนินการ

1) ศึกษาทฤษฎีพื้นฐานเกี่ยวกับการแก้ปัญหาทั่วๆ ไปด้วยขั้นตอนวิธีการค้นหาแบบเอสตาร์
2) ศึกษาทฤษฎีพื้นฐานเกี่ยวกับการแก้ปัญหาทั่วๆ ไปด้วยขั้นตอนวิธีเชิงพันธุกรรม
3) ศึกษาทฤษฎีพื้นฐานเกี่ยวกับการแก้ปัญหาทั่วๆ ไปด้วยโครงข่ายประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ศึกษาทฤษฎีพื้นฐานเกี่ยวกับการแก้ปัญหาทั่วๆ ไปด้วย NEAT (NeuroEvolution of Augmenting Topologies)
5) ศึกษาการใช้งานโปรแกรม Blender
6) ศึกษาการใช้งาน Unity3D
7) ออกแบบเกม
8) ออกแบบปัญญาประดิษฐ์
9) พัฒนาตัวเกมหลัก
10) พัฒนาส่วนของการค้นหาเส้นทาง
11) พัฒนาส่วนของการเรียนรู้
12) ทดลองและบันทึกผล
13) ทดสอบความถูกต้องและปรับปรุงแก้ไขส่วนที่ผิดพลาดของเกม
14) สร้างแพคเกจตัวเกม
15) สรุปผลโครงงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ความรู้ความเข้าใจอย่างลึกซึ้งเกี่ยวกับกระบวนการทำงานเบื้องต้นของขั้นตอนวิธีการค้นหาแบบเอสตาร์, ขั้นตอนวิธีเชิงพันธุกรรม, โครงข่ายประสาทเทียม, และ NEAT (NeuroEvolution of Augmenting Topologies)
2) สามารถแก้ปัญหาการตัดสินใจใช้แผนการต่างๆ ของ NPC (Non Player Character) ของเกมประเภท RTS (Real-Time Strategy) แบบวางแผนการรบ (War game) ได้ด้วยการประยุกต์ใช้ความรู้เกี่ยวกับขั้นตอนวิธีเชิงพันธุกรรม, โครงข่ายประสาทเทียม, และ NEAT (NeuroEvolution of Augmenting Topologies)
3) สามารถแก้ปัญหาการค้นหาเส้นทางของตัวละครของเกมประเภท RTS แบบวางแผนการรบได้ด้วยการประยุกต์ใช้ความรู้เกี่ยวกับขั้นตอนวิธีการค้นหาแบบเอสตาร์
4) สามารถพัฒนาเกมได้อย่างเป็นระบบ

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บท ได้แก่

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องที่ใช้ในโครงงาน ซึ่งประกอบด้วยขั้นตอนวิธีการค้นหาแบบเอสตาร์ ขั้นตอนวิธีเชิงพันธุกรรม โครงข่ายประสาทเทียม NeuroEvolution of Augmenting Topologies (NEAT) การใช้งาน Blender และการใช้งาน Unity3D

บทที่ 3 กล่าวถึงการออกแบบโครงงาน โดยการนำความรู้ที่ได้จากทฤษฎีบทที่ 2 มาประยุกต์ใช้ในการพัฒนาโครงงานนี้ ซึ่งประกอบด้วยแนวคิดของเกม การออกแบบเกม การออกแบบโครงสร้างและขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียม และการประยุกต์ใช้ปัญญาประดิษฐ์กับเกม

บทที่ 4 กล่าวถึงผลการทดลอง ซึ่งประกอบด้วยโมเดล การหาเส้นทางและเดินหลบสิ่งกีดขวาง การใช้ NEAT แก้ปัญหาตรรกศาสตร์ XOR การใช้ NEAT แก้ปัญหา DPV (Double Pole Balancing with Velocities) การใช้ NEAT สร้าง ANN และการสร้างเกมด้วย Unity3D

บทที่ 5 เป็นบทสรุป ซึ่งกล่าวถึงบทสรุปของโครงงาน ปัญหาอุปสรรคและแนวทางในการแก้ไข ผลการดำเนินงาน และแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ขั้นตอนวิธีการค้นหาแบบเอสตาร์

ขั้นตอนวิธีการค้นหาแบบเอสตาร์ (A* search algorithm) เป็นขั้นตอนวิธีการที่ได้รับความนิยมเป็นอย่างมากเนื่องจากเป็นขั้นตอนวิธีที่มีความแม่นยำและมีสมรรถภาพมากในการค้นหาเส้นทาง ซึ่งสามารถเลือกเส้นทางในการเดินระหว่างโหนดใดๆ ได้อย่างมีประสิทธิภาพ จึงเป็นขั้นตอนวิธีสำหรับการค้นหาเส้นทางที่ถูกใช้อย่างแพร่หลาย ถูกคิดค้นโดย Peter Hart, Nils Nilsson, และ Bertram Raphael ในปี 1968 ถือเป็นขั้นตอนวิธีที่ถูกพัฒนาเพิ่มขึ้นต่อจากขั้นตอนวิธีของ Edsger Dijkstra's 1959 algorithm โดยมีการใช้ฮิวริสติก (Heuristic) เข้ามาช่วยในการทำงานด้วย

ขั้นตอนวิธีการค้นหาแบบเอสตาร์เป็นการค้นหาเส้นทางที่ดีที่สุดและมีค่าใช้จ่าย (Cost) น้อยที่สุดจากโหนดใดๆ ไปยังโหนดเป้าหมาย โดยใช้ฮิวริสติก (Heuristic function) ซึ่งมีสัญลักษณ์เป็น $f(x)$ ในการคำนวณค่าลำดับของโหนดที่ถูกเดินผ่าน โดยใน $f(x)$ ประกอบด้วย 2 ฟังก์ชั่นย่อย ได้แก่

1) การคำนวณค่าใช้จ่ายสำหรับโหนดใดๆ คือค่าใช้จ่ายจากโหนดเริ่มต้นมายังโหนดปัจจุบัน ซึ่งมีสัญลักษณ์เป็น $g(x)$
2) การคำนวณค่าประมาณฮิวริสติก คือค่าประมาณคุณภาพของโหนดหนึ่งๆ มักจะเป็นระยะห่างจากโหนดปัจจุบันไปยังโหนดเป้าหมาย ซึ่งมีสัญลักษณ์เป็น $h(x)$

$h(x)$ เป็นส่วนหนึ่งของ $f(x)$ ที่ต้องเป็นค่าฮิวริสติกที่ยอมรับได้ ซึ่งต้องไม่มากเกินกว่าระยะห่างจากโหนดปัจจุบันไปยังโหนดเป้าหมาย ดังนั้นในการทำงานดังเช่นการหาเส้นทางนั้น $h(x)$ อาจเป็นระยะทางที่เป็นเส้นตรงจากโหนดปัจจุบันไปยังโหนดเป้าหมาย ซึ่งก็คือระยะห่างที่สั้นที่สุดระหว่างโหนดปัจจุบันกับโหนดเป้าหมาย

2.1.1 การทำงานของขั้นตอนวิธีการค้นหาแบบเอสตาร์

การทำงานของขั้นตอนวิธีการค้นหาแบบเอสตาร์ทำงานคล้ายๆกับขั้นตอนวิธีการค้นหาแบบกรีดี (Greedy best first search) คือมีการหาเส้นทางที่ปรากฏความเป็นไปได้ว่าจะนำไปสู่โหนดเป้าหมาย แต่สิ่งที่แตกต่างคือขั้นตอนวิธีการค้นหาแบบกรีดีนั้นมีการประเมินเส้นทางจากค่าเสียหายที่เดินไปถึงโหนดเป้าหมายแล้ว ในขณะที่การทำงานของขั้นตอนวิธีการค้นหาแบบเอสตาร์จะประเมินค่าเสียหายจาก 2 ส่วนรวมกันดังสมการ 2.1

$$f(x) = g(x) + h(x) \tag{2.1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานเริ่มจากการคำนวณค่าฮิวริสติกของโหนดที่เดินผ่านใส่ลงไปในคิวลำดับความสำคัญ (Priority queue) โดย $f(x)$ ที่ต่ำกว่าจะถูกให้แก่โหนด x (ค่า $f(x)$ ที่ต่ำกว่าจะมีค่าความสำคัญที่สูงกว่า) ซึ่งในแต่ละขั้นตอนของการทำงาน โหนดที่มีค่า $f(x)$ ต่ำสุดจะถูกเอาออกจากคิวลำดับความสำคัญ ในขณะที่โหนดข้างเคียงก็ทำเช่นเดียวกันและถูกเพิ่มลงในคิวลำดับความสำคัญ จนกระทั่งโหนดเป้าหมายมีค่า $f(x)$ ต่ำกว่าโหนดใดๆ หรือคิวลำดับความสำคัญว่าง (โหนดเป้าหมายอาจถูกเดินผ่านหลายครั้ง ถ้ายังมีระยะทางที่สั้นกว่าที่จะไปถึงโหนดเป้าหมายนั้นอยู่) ซึ่งหมายความว่าถ้าค่า $f(x)$ สุดท้ายของโหนดเป้าหมายที่ได้เป็นค่าต่ำที่สุดแล้ว $h(x)$ จะมีค่าเป็น 0

2.1.2 ตัวอย่าง

ตัวอย่างนี้เป็นการค้นหาเส้นทางที่สั้นที่สุดจากเมืองอารัด (Arad) ไปยังเมืองบูคาเรสต์ (Bucharest) ในประเทศโรมาเนีย (Romania) จากแผนที่แสดงที่ตั้งเมืองต่างๆ ดังรูป 2.1

รูป 2.1 ที่ตั้งเมืองต่างๆ และระยะทางระหว่างเมืองในประเทศโรมาเนีย

ซึ่งระยะห่างจากแต่ละเมืองถึงเมืองเป้าหมาย (บูคาเรสต์) จะวัดเป็นเส้นตรง เพื่อให้ได้ระยะห่างที่สั้นที่สุดดังตาราง 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 ระยะห่างระหว่างเมืองต่างๆ กับเมืองบูคาเรสต์เมื่อวัดเป็นเส้นตรง

Straight Line Distances to Bucharest

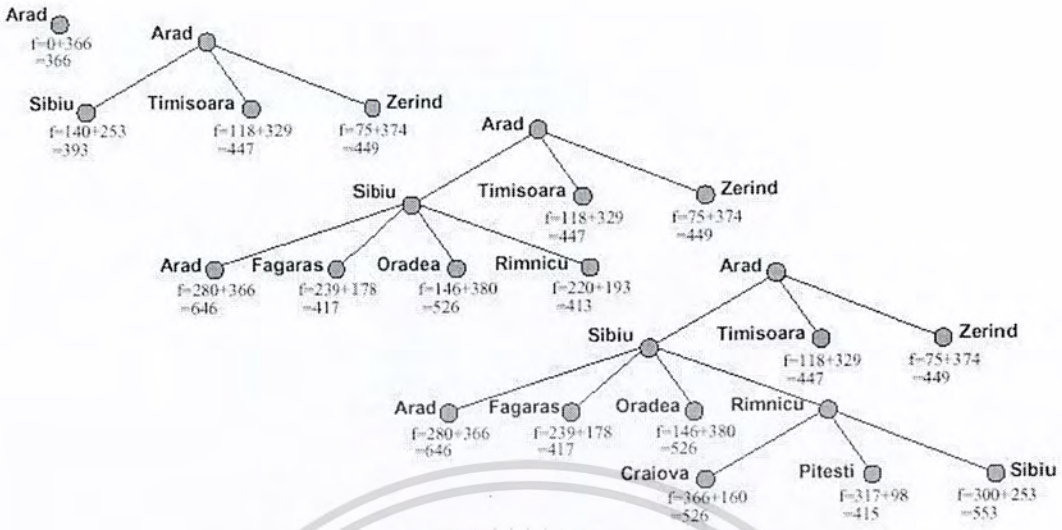
Town	SLD	Town	SLD
Arad	366	Mehadai	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Dobreta	242	Pitesti	98
Eforie	161	Rimnicu	193
Fagaras	178	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

การใช้ขั้นตอนวิธีการค้นหาแบบเอสตาร์ในการแก้ปัญหานี้เริ่มที่โหนดเริ่มต้นคือเมืองอารัด โดยสามารถคำนวณหาค่า $f(x)$ ได้ดังสมการ 2.2

$$f(x) = g(x) + h(x) = 0 + 366 = 366 \tag{2.2}$$

เนื่องจากในโหนดเริ่มต้นยังไม่มีการเดินไปไหน ดังนั้น $g(x)$ จึงยังคงเป็น 0 ส่วนค่า $h(x)$ สามารถดูได้จากแผนที่ด้านบน หลังจากนั้นจึงนำโหนดเมืองอารัดใส่ลงไปในคิวลำดับความสำคัญ และทำตามขั้นตอนวิธีคือทำการเลือกโหนดที่มีค่า $f(x)$ น้อยที่สุด ซึ่งในที่นี้มีโหนดเดียวคือโหนดเมืองอารัดเพราะเป็นโหนดเริ่มต้น จึงทำการเลือกออกมาจากคิวลำดับความสำคัญและแตกโหนดไปยังเมืองต่างๆที่สามารถเดินไปได้ แล้วก็ทำการคำนวณหาค่า $f(x)$ ของโหนดต่างๆที่ถูกแตกออกมาแล้วใส่ลงไปในคิวลำดับความสำคัญ และทำการเลือกโหนดที่มีค่า $f(x)$ น้อยสุดออกมาแตกต่อไป ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงโหนดเป้าหมายคือเมืองบูคาเรสต์และ $f(x)$ บนโหนดเป้าหมายนั้นมีค่าน้อยที่สุดในคิวลำดับความสำคัญ หรือในกรณีที่โหนดในคิวลำดับความสำคัญนั้นถูกเอาออกไปหมดแล้ว ก็จะได้เส้นทางที่สั้นที่สุด ซึ่งกระบวนการบางส่วนสามารถแสดงเป็นแผนภาพของการท่องโหนดในต้นไม้ได้ดังรูป 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.2 การท่องโหนดในต้นไม้ของกระบวนการเอสตาร์

2.1.3 ประโยชน์ต่อการพัฒนาเกม

เนื่องจากเกมยิงมุมมองบุคคลที่หนึ่ง (First Person Shooter) เป็นเกมที่ต้องมีการหาเส้นทางการเดินทางของ NPC (Non Player Character) อยู่ตลอดเวลา ทำให้มีการนำขั้นตอนวิธีการค้นหาแบบเอสตาร์ไปใช้อย่างแพร่หลาย โดยการสร้าง Waypoint ขึ้นมาเป็นโหนดสำหรับวิธีการค้นหาแบบเอสตาร์ นอกจากเกมยิงมุมมองบุคคลที่หนึ่งแล้ว ยังมีเกมวางแผน (Strategy game) ซึ่งต้องมีการหาเส้นทางทั้งของ NPC และ PC (Player Character) ก็มีการนำเอาวิธีการค้นหาแบบเอสตาร์ไปใช้ด้วยเช่นกัน

2.2 ขั้นตอนวิธีเชิงพันธุกรรม

ในช่วงก่อนคริสตศักราช 1970 John Holland หนึ่งในกลุ่มคนที่คิดค้นการคำนวณเชิงวิวัฒนาการ (Evolutionary Computation: EC) ได้นำเสนอแนวคิดของขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithms: GAs) ที่มีลักษณะเป็นขั้นตอนวิธีเชิงวิวัฒนาการรูปแบบหนึ่ง โดยมีจุดมุ่งหมายที่จะทำให้เครื่องคอมพิวเตอร์สามารถกระทำในสิ่งที่ธรรมชาติกระทำได้ เขาได้สร้างขั้นตอนวิธี (Algorithms) ซึ่งรับชุดข้อมูลของเลขฐานสองเข้าไป โดยพิจารณาให้ ขั้นตอนวิธีนี้เปรียบเสมือนรูปแบบทางนามธรรมของการพัฒนาแบบที่ธรรมชาติใช้ โดยจะแทนการทำงานขั้นตอนต่างๆ จากประชากร (Population) หนึ่งซึ่งก็คือชุดของโครโมโซม (Chromosome) ไปเป็นประชากรรุ่น (Generation) ใหม่ โดยอาศัยหลักการคัดเลือกตามธรรมชาติ (Natural Selection) โดยใช้ตัวปฏิบัติการทางพันธุกรรม (Genetic Operator) ได้แก่ การสืบพันธุ์ (Reproduction) การครอสโอเวอร์ (Crossover) และมิวเทชัน (Mutation) ซึ่งในขั้นตอนวิธีนั้นแต่ละโครโมโซมจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วยยีน (Gene) หลายๆ ตัวและซึ่งกลุ่มของยีนเหล่านี้ก็จะถูกแทนด้วยชุดข้อมูลเลขฐานสอง

ธรรมชาติมีความสามารถในการเรียนรู้และปรับเปลี่ยนให้เหมาะสมได้ด้วยตัวเอง ซึ่งในที่นี้ก็คือ ธรรมชาติสามารถหาโครโมโซมที่ดีได้โดยที่ไม่ต้องมีองค์ความรู้ใดๆ เลย ซึ่งขั้นตอนวิธีเชิงพันธุกรรมก็จะกระทำแบบนี้เช่นเดียวกัน สองกระบวนการที่จะเชื่อมขั้นตอนวิธีเชิงพันธุกรรมกับปัญหาต่างๆ เพื่อแก้ปัญหาเหล่านั้น ก็คือ การเข้ารหัส (Encoding) และการประเมินค่า (Evaluation)

2.2.1 การเข้ารหัส

John Holland ใช้การเข้ารหัสโดยการแทนแต่ละโครโมโซมเป็นข้อมูลชนิดเลขฐานสอง ถึงแม้ว่าในช่วงเวลาต่อมาจะการเข้ารหัสโดยการแทนค่าลงในโครโมโซมด้วยรูปแบบอื่นๆเกิดขึ้น แต่ก็ไม่มีกระบวนการใดที่ดีที่สุดที่จะเหมาะสมกับทุกๆ ปัญหา โดยในบทนี้เราจะใช้การแทนด้วยเลขฐานสองเป็นหลัก ซึ่งเป็นเทคนิคที่ได้รับความนิยมมากที่สุด

2.2.1.1 การเข้ารหัสแบบเลขฐานสอง (Binary Encoding)

การเข้ารหัสแบบเลขฐานสองเป็นวิธีที่นิยมใช้กันมากเพราะเป็นการเข้ารหัสวิธีแรกที่ใช้ในขั้นตอนวิธีเชิงพันธุกรรม โดยในการเข้ารหัสแบบเลขฐานสองนั้น ทุกๆโครโมโซมจะถูกแทนด้วยข้อมูลที่เป็นเลขฐานสองทั้งหมดดังเช่นรูป 2.3

1	0	0	1	1	1	0	1	1	1	0	1	0	0

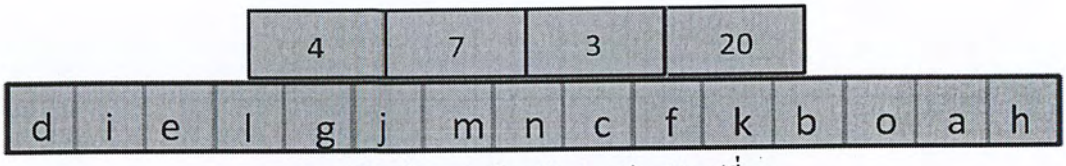
รูป 2.3 ตัวอย่างการเข้ารหัสแบบเลขฐานสอง

ข้อดีของการเข้ารหัสแบบเลขฐานสองคือการทำให้โครโมโซมมีความหลากหลายมากขึ้น ถึงแม้จะมีจำนวนยีนที่อยู่ในโครโมโซมน้อยก็ตาม แต่มีข้อเสียคือ การเข้ารหัสแบบนี้ยังไม่เป็นไปตามธรรมชาติ เนื่องจากธรรมชาติมีความสามารถในการปรับเปลี่ยนโครงสร้างของโครโมโซมซึ่งมีความยืดหยุ่นมากกว่าการเข้ารหัสแบบเลขฐานสอง ซึ่งเป็นกระบวนการที่มีการบังคับขอบเขตของผลลัพธ์ตามจำนวนบิตที่ถูกกำหนดไว้ในตอนเริ่มต้น จึงทำให้มีปัญหาตามมาในกรณีที่ผลลัพธ์ที่ดีอยู่นอกขอบเขตของตัวเลข จึงต้องแก้ไขเสมอหลังจากผ่านการใช้ตัวปฏิบัติการทางพันธุกรรม

2.2.1.2 การเข้ารหัสแบบเรียงสับเปลี่ยน (Permutation Encoding)

สามารถใช้ในการแก้ปัญหาการจัดเรียงลำดับ เช่น TSP (Travelling Salesman Problem) หรือ TOP (Task Ordering Problem) ในการเข้ารหัสแบบเรียงสับเปลี่ยน ทุกๆโครโมโซมจะถูกแทนด้วยข้อมูลที่เป็นตัวเลขหรือตัวอักษร ซึ่งจะมีการเรียงลำดับอยู่ภายในโครโมโซม ดังเช่นรูป 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



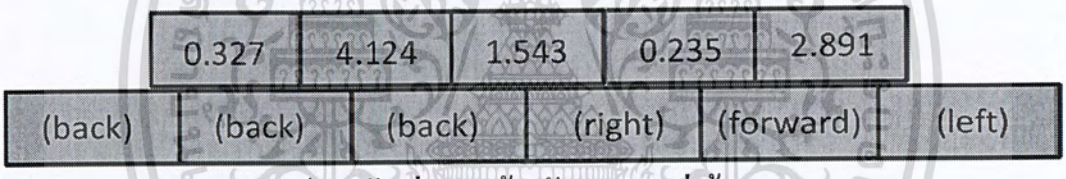
รูป 2.4 ตัวอย่างการเข้ารหัสแบบเรียงสับเปลี่ยน

การเข้ารหัสแบบเรียงสับเปลี่ยนสามารถใช้ได้ดีสำหรับการแก้ปัญหาประเภทการจัดเรียงลำดับเท่านั้น สำหรับการแก้ปัญหาเหล่านี้ ในการการครอสโอเวอร์และมิวเทชัน จะต้องคำนึงถึงข้อมูลในโครโมโซมซึ่งจะต้องสอดคล้องกับข้อมูลจริงที่นำมาเรียงจากปัญหาที่แก้ด้วย

2.2.1.3 การเข้ารหัสแบบแทนค่าข้อมูล (Value Encoding)

การเข้ารหัสแบบแทนค่าข้อมูลเป็นการแทนค่าหรือข้อมูลลงไปตรงๆ สามารถใช้ได้ในกรณีที่ปัญหาที่จะแก้ ใช้ค่าหรือข้อมูลที่มีความยุ่งยากซับซ้อนต่อการเข้ารหัส เช่น ปัญหาที่ใช้ค่าที่มีจุดทศนิยม ซึ่งการที่จะใช้การเข้ารหัสแบบเลขฐานสอง แก้ปัญหานี้จะทำได้ยากมาก

ในการเข้ารหัสแบบแทนค่าข้อมูล ทุกๆโครโมโซมจะถูกแทนด้วยค่าอะไรก็ได้ซึ่งขึ้นอยู่กับปัญหาที่จะแก้



รูป 2.5 ตัวอย่างการเข้ารหัสแบบแทนค่าข้อมูล

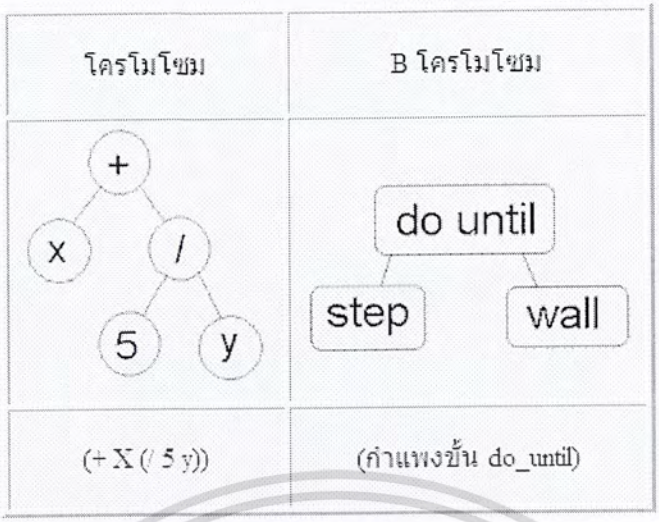
การเข้ารหัสแบบแทนค่าข้อมูลเป็นวิธีที่ดีมากสำหรับการแก้ปัญหาบางปัญหาโดยเฉพาะ แต่มีข้อเสียคือ การเข้ารหัสแบบแทนค่าข้อมูลส่วนใหญ่จะต้องมีการพัฒนาตัวปฏิบัติการทางพันธุกรรมให้เหมาะสมด้วย

2.2.1.4 การเข้ารหัสแบบต้นไม้ (Tree Encoding)

ส่วนใหญ่จะใช้ในการพัฒนาโปรแกรม (Program) หรือนิพจน์ (Expression) สำหรับ GP (Genetic Programming)

ในการเข้ารหัสแบบต้นไม้ ทุกๆโครโมโซมจะถูกแทนด้วยข้อมูลที่มีลักษณะโครงสร้างเป็นต้นไม้ (Tree)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.6 ตัวอย่างการเข้ารหัสแบบต้นไม้

การเข้ารหัสแบบต้นไม้เป็นวิธีที่ดีสำหรับการพัฒนาโปรแกรม ในภาษาลิสป์ (LISP) ส่วนใหญ่จะใช้วิธีนี้ เพราะในโปรแกรมของมันถูกแทนด้วยรูปแบบนี้และสามารถทำให้ง่ายต่อการวิเคราะห์ซึ่งอยู่ในรูปแบบต้นไม้ ดังนั้นจึงทำให้การใช้ตัวปฏิบัติการทางพันธุกรรมมีความง่ายในการสร้างความสัมพันธ์ของโครโมโซม

2.2.2 ฟังก์ชันประเมินค่า (Evaluation Function)

ฟังก์ชันประเมินค่าใช้วัดค่าความเหมาะสม (Fitness) ของโครโมโซมต่อปัญหาที่ต้องการจะแก้ โดยขั้นตอนวิธีเชิงพันธุกรรมจะนำค่าความเหมาะสมของโครโมโซมแต่ละตัวที่ได้ไปใช้ในการคัดเลือก (Selection) โครโมโซมเพื่อนำมาสร้างโครโมโซมรุ่นใหม่โดยใช้ตัวปฏิบัติการทางพันธุกรรมในการสร้างโครโมโซม

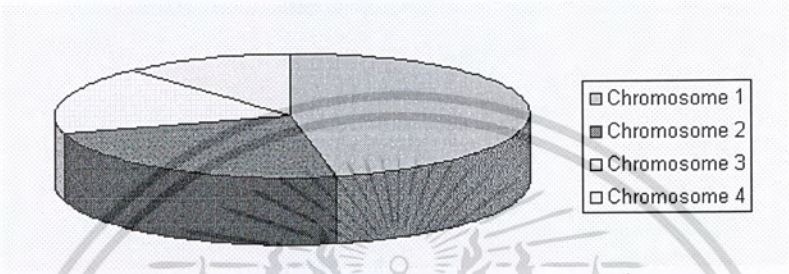
2.2.3 การคัดเลือก (Selection)

โครโมโซมถูกเลือกจากประชากรทั้งรุ่นเพื่อเข้าสู่กระบวนการสร้างโครโมโซมรุ่นถัดไป แต่ปัญหาคือการเลือกนั้นจะใช้หลักเกณฑ์อะไรในการเลือก จากหลักที่ว่าโครโมโซมที่มีค่าความเหมาะสมมากที่สุดในแต่ละรุ่นควรเป็นตัวที่อยู่รอด และนำไปใช้ในสร้างโครโมโซมในรุ่นต่อไป จึงทำให้เกิดหลายวิธีการที่จะใช้ในการเลือกโครโมโซม เช่น การคัดเลือกแบบวงล้อรูเรทท์ (Roulette Wheel Selection), การคัดเลือกแบบโบลต์ซมันน์ (Boltzmann Selection), การคัดเลือกโดยการประลอง (Tournament Selection), การคัดเลือกโดยการจัดอันดับ (Rank Selection), การคัดเลือกแบบสภาวะคงตัว (steady state selection) และวิธีอื่นๆ ซึ่งในที่นี้จะขอกล่าวถึงเฉพาะบางวิธีที่ค่อนข้างจะได้รับความนิยมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.1 การคัดเลือกแบบวงล้อรูเรทท์ (Roulette Wheel Selection)

เป็นวิธีการคัดเลือกที่ใช้หลักว่าโครโมโซมที่ดีกว่าจะมีโอกาสถูกเลือกมากกว่าโครโมโซมที่ดีน้อยกว่า และใช้หลักการของวงล้อรูเรทท์ (Roulette Wheel) ซึ่งจะวางโครโมโซมทุกๆ ตัว ในประชากรรุ่นนั้นลงบนวงล้อรูเลทท์ โดยโครโมโซมที่มีค่าความเหมาะสมมากจะมีพื้นที่มากกว่าตัวที่มีค่าความเหมาะสมน้อยดังรูป 2.7 หลังจากนั้นจะใช้วิธีการสุ่มเลือกพื้นที่บนวงล้อรูเลทท์เพื่อเลือกโครโมโซมออกมาซึ่งเปรียบเสมือนการหมุนวงล้อรูเลทท์



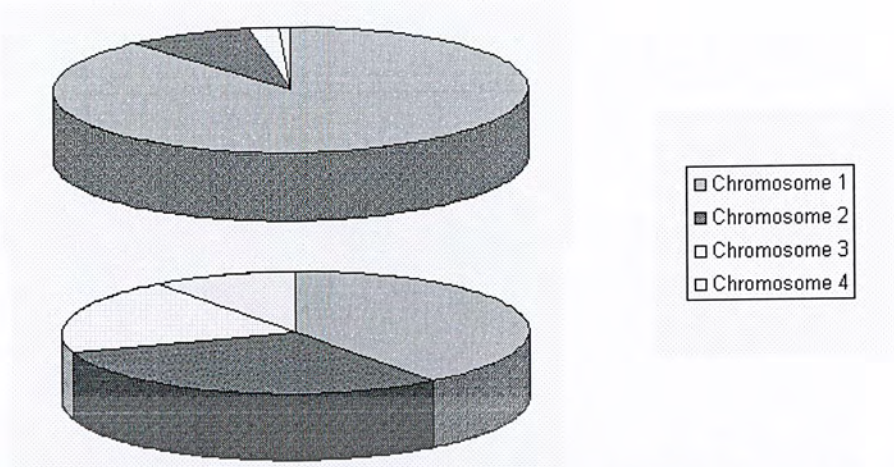
รูป 2.7 ตัวอย่างการคัดเลือกแบบวงล้อรูเรทท์

การคัดเลือกแบบวงล้อรูเรทท์จะมีปัญหาเมื่อค่าความเหมาะสมของแต่ละโครโมโซมที่อยู่ในประชากรรุ่นเดียวกันมีความแตกต่างกันมาก ตัวอย่างเช่น ถ้าโครโมโซมตัวที่ดีที่สุดยึดครองพื้นที่บนวงล้อรูเลทท์มากกว่า 90% แล้ว จะทำให้โครโมโซมตัวอื่นๆ ที่อยู่ในประชากรรุ่นเดียวกันอาจไม่มีโอกาสที่จะถูกเลือกเลย

2.2.3.2 การคัดเลือกโดยการจัดอันดับ (Rank Selection)

การคัดเลือกโดยการจัดอันดับ คือ การใช้วิธีเรียงลำดับตามค่าความเหมาะสม โดยให้โครโมโซมที่มีค่าความเหมาะสมน้อยที่สุดมีค่าความเหมาะสมใหม่เท่ากับ 1 จากนั้นโครโมโซมที่มีค่าความเหมาะสมน้อยรองลงมามีค่าเท่ากับ 2 ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงโครโมโซมที่มีค่าความเหมาะสมมากที่สุด ให้มีค่าความเหมาะใหม่เท่ากับ N โดยที่ N คือ จำนวนของโครโมโซมของประชากรในรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.8 ตัวอย่างก่อนผ่านขั้นตอนและหลังผ่านขั้นตอนการคัดเลือกโดยการจัดอันดับ

หลังจากผ่านการคัดเลือกโดยการจัดอันดับแล้ว โครโมโซมทุกตัวจะมีโอกาสที่จะถูกเลือก แต่วิธีนี้จะทำให้ได้คำตอบช้าเพราะว่าโครโมโซมตัวที่ดีที่สุดจะมีโอกาสถูกเลือกไม่แตกต่างจากโครโมโซมตัวอื่นๆ มากนัก

2.2.4 ตัวปฏิบัติการทางพันธุกรรม (Genetic Operation)

กระบวนการที่ใช้ในการกระทำกับโครโมโซมที่ถูกเลือก เพื่อให้ได้โครโมโซมของประชากรในรุ่นใหม่ มีอยู่ 3 วิธีคือ การสืบพันธุ์ การครอสโอเวอร์และมิวเทชัน นอกจากนี้อาจจะมีกระบวนการอื่นๆ ที่สร้างขึ้นมาเพื่อแก้ปัญหาใดๆ โดยเฉพาะ แต่ในที่นี้กล่าวถึงเฉพาะ 3 วิธีการหลักๆ

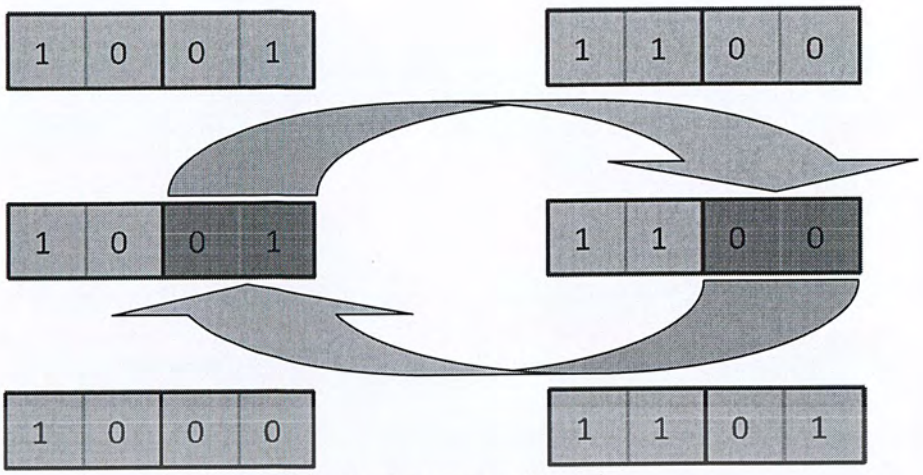
2.2.4.1 การสืบพันธุ์

การสืบพันธุ์เป็นการสร้างโครโมโซมตัวใหม่โดยนำโครโมโซมจากรุ่นก่อนไปสร้างเป็นโครโมโซมตัวใหม่โดยไม่มีการเปลี่ยนแปลงใดๆ

2.2.4.2 การครอสโอเวอร์

การครอสโอเวอร์จะใช้โครโมโซม 2 ตัว ในขั้นแรกคือการสุ่มเลือกจุดตัดบนโครโมโซมซึ่งโครโมโซมจะใช้จุดๆ นี้ในการหยุดการแลกเปลี่ยนยีนระหว่างกัน หลังจากนั้นก็เริ่มแลกเปลี่ยนยีนซึ่งกันและกัน สุดท้ายก็จะได้ผลลัพธ์เป็นโครโมโซมตัวใหม่ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.9 ตัวอย่างการครอสโอเวอร์

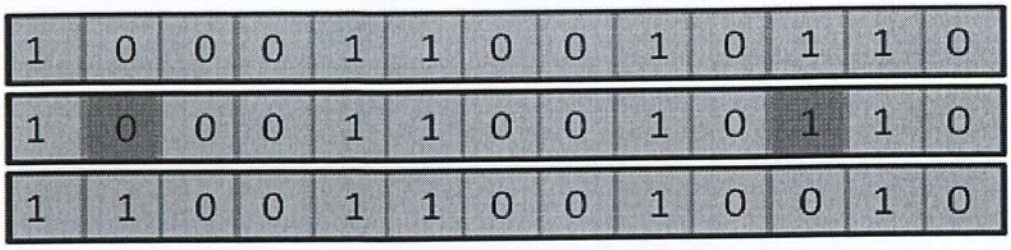
นอกจากนี้ยังมีวิธีการการครอสโอเวอร์แบบอื่นๆ นอกเหนือจากในรูป แต่ก็ยังมีความคล้ายคลึงกับแบบนี้อยู่ เช่น การการครอสโอเวอร์แบบที่มีการสุ่มเลือกจุดตัดมากกว่า 1 จุด เป็นต้น การครอสโอเวอร์มีได้หลากหลายรูปแบบ ซึ่งจะขึ้นอยู่กับการเข้ารหัสในขั้นต้น ดังนั้นจึงเป็นไปได้ที่จะมีการครอสโอเวอร์แบบที่พัฒนาขึ้นมาเพื่อแก้ปัญหาใดปัญหาหนึ่งโดยเฉพาะ ซึ่งสามารถเพิ่มประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมได้

2.2.4.3 มิวเทชัน

มิวเทชันจะใช้โครโมโซม 1 ตัว ขั้นตอนก็คือสุ่มเลือกยีนภายในโครโมโซมมาหนึ่งคู่ จากนั้นก็สลับตำแหน่งซึ่งกันและกัน ก็จะทำให้ได้โครโมโซมสำหรับประชากรรุ่นใหม่ขึ้นมา

Holland ได้นำเสนอว่ามิวเทชันเปรียบเสมือนตัวดำเนินงานเบื้องหลัง หน้าที่ของมันคือการทำให้แน่ใจว่าผลลัพธ์ที่ได้จะไม่ติดอยู่ในค่าดีที่สุดเฉพาะบริเวณ (Local Optimum) เนื่องจากถ้ามีเพียงขั้นตอนการทำงานตั้งแต่การคัดเลือกจนถึงการสืบพันธุ์และการครอสโอเวอร์จะทำให้ได้กลุ่มของโครโมโซมที่มีคุณสมบัติคล้ายๆ กัน และจึงส่งผลให้ค่าเฉลี่ยค่าความเหมาะสมของประชากรในแต่ละรุ่นไม่สามารถที่จะปรับปรุงให้ดีขึ้นได้มากนัก ถึงอย่างไรก็ตามผลลัพธ์ที่ได้ควรจะมีประสิทธิภาพมากกว่าที่จะเป็นผลลัพธ์ที่อยู่ในค่าดีที่สุดเฉพาะบริเวณ ดังนั้นจึงต้องมีมิวเทชัน ซึ่งเปรียบเสมือนการสุ่มหาผลลัพธ์ไม่เจาะจงที่จุดใดจุดหนึ่ง จึงทำให้ได้ผลลัพธ์ที่มีความหลากหลายมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.10 ตัวอย่างมิวเทชัน

2.2.5 ขั้นตอนพื้นฐาน

ขั้นตอนวิธีเชิงพันธุกรรมมีลักษณะเป็นขั้นตอนวิธีค้นหาแบบสุ่ม (Stochastic Search Algorithms) ชนิดหนึ่ง โดยใช้หลักการพัฒนาทางชีวภาพ เพื่อเป็นการให้มองเห็นขั้นตอนการทำงานได้อย่างชัดเจน ต่อไปจะเป็นการอธิบายขั้นตอนการทำงานพื้นฐานของขั้นตอนวิธีเชิงพันธุกรรมที่ใช้ในการแก้ปัญหาที่สามารถแทนคำตอบของมันด้วยข้อมูลชนิดเลขฐานสองได้ โดยที่ขั้นตอนวิธีเชิงพันธุกรรมนี้เป็นเพียงพื้นฐานสำหรับแก้ปัญหาในรูปแบบอื่นๆ ซึ่งเวลานำไปใช้งานจริงอาจจะต้องมีการปรับเปลี่ยนเพื่อให้เหมาะสมกับปัญหานั้นๆ ขั้นตอนพื้นฐานมีขั้นตอนดังนี้

1) แทนชุดของตัวแปรที่อยู่ในขอบเขตของปัญหาให้อยู่ในรูปของโครโมโซมโดยที่มีการกำหนดความยาวที่แน่นอน จากนั้นให้กำหนดจำนวนโครโมโซมที่จะเป็นประชากรในแต่ละรุ่นเป็นจำนวน N ตัว และกำหนดโอกาสการเกิดการครอสโอเวอร์ (p_c) และอัตราการเกิดมิวเทชัน (p_m)
2) สร้างฟังก์ชันค่าความเหมาะสม (Fitness Function) ซึ่งเป็นสิ่งที่ใช้วัดค่าความเหมาะสมของโครโมโซมที่อยู่ในขอบเขตของปัญหา โดยที่ฟังก์ชันค่าความเหมาะสมจะประมาณค่าความเหมาะสม เพื่อนำไปใช้ในการเลือกโครโมโซมที่จะนำไปใช้ในการสร้างโครโมโซมที่จะเป็นประชากรรุ่นใหม่
3) สุ่มค่าขึ้นมาเพื่อใส่เป็นค่าตั้งต้นให้กับโครโมโซมแต่ละตัว ซึ่งมีจำนวน N ตัว ได้ค่าคือ $x_1, x_2, \dots, x_n$
4) คำนวณค่าความเหมาะสมของโครโมโซมแต่ละโครโมโซมโดยใช้ฟังก์ชันค่าความเหมาะสม ได้ค่าคือ $f(x_1), f(x_2), \dots, f(x_n)$
5) เลือกโครโมโซมขึ้นมาหนึ่งคู่จากชุดของโครโมโซมที่อยู่ในประชากรรุ่นปัจจุบัน จากนั้นจึงเลือกโดยการใช้หลักการความน่าจะเป็นในการเลือก โดยที่โครโมโซมที่มีค่าความเหมาะสมสูง ก็จะมีโอกาสที่จะถูกเลือกสูงกว่าโครโมโซมที่มีค่าความเหมาะสมต่ำ
6) สร้างคู่ของ โครโมโซมในประชากรรุ่นใหม่ โดยนำโครโมโซมที่ถูกเลือกมาจากขั้นตอนที่ 5 มาเข้าสู่ใช้ตัวปฏิบัติการทางพันธุกรรม

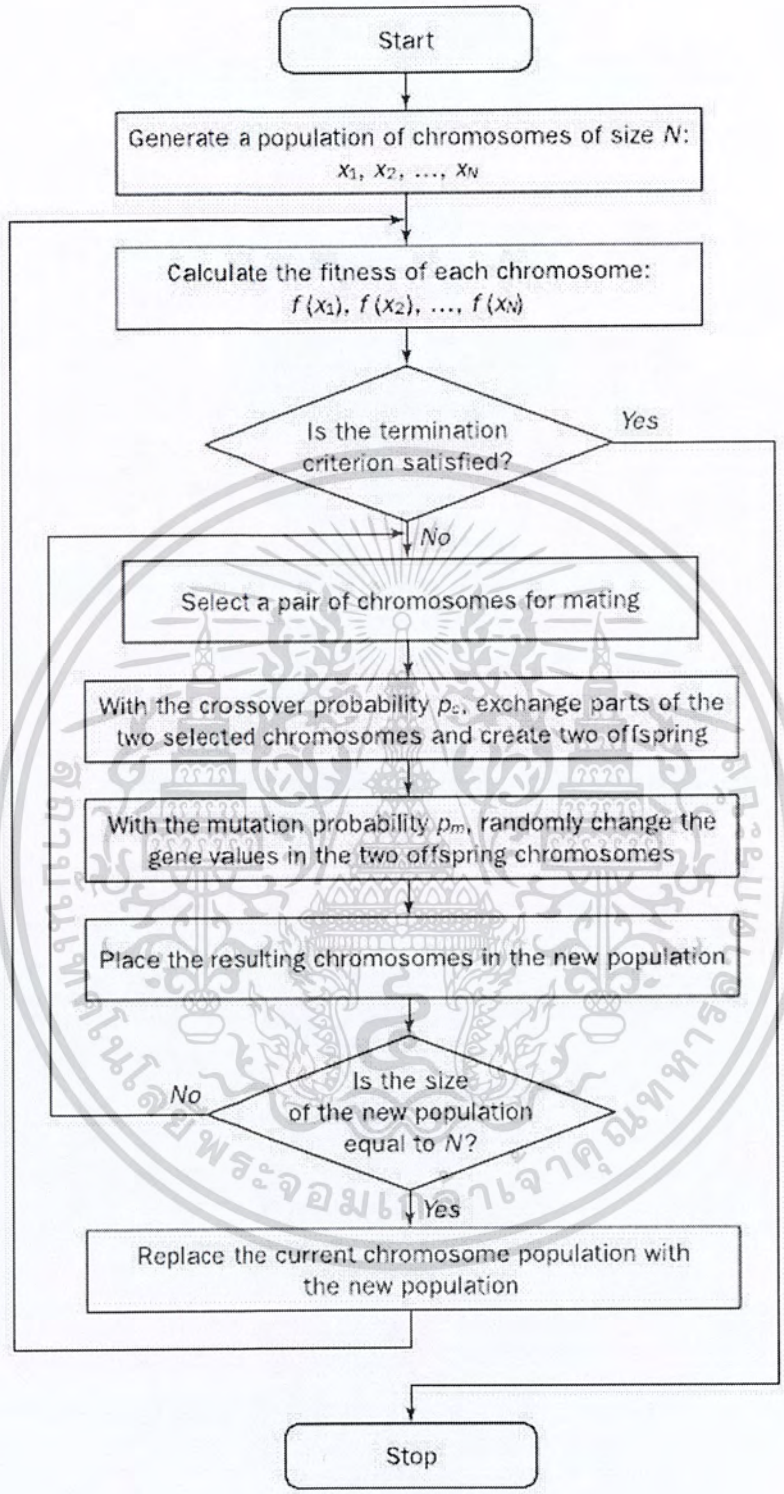
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) นำโครโมโซมที่ถูกสร้างขึ้นมาลงในโครโมโซมในประชากรรุ่นใหม่
8) กลับไปทำในขั้นตอนที่ 5 จนกว่าจำนวนโครโมโซมในประชากรรุ่นใหม่จะเท่ากับจำนวนโครโมโซมตั้งต้น จำนวน N ตัว
9) นำโครโมโซมในประชากรรุ่นใหม่แทนที่ลงไปในโครโมโซมในประชากรรุ่นปัจจุบัน
10) นำโครโมโซมที่ถูกสร้างขึ้นมาลงในโครโมโซมในประชากรรุ่นใหม่

จะสังเกตได้ว่าขั้นตอนพื้นฐานของขั้นตอนวิธีเชิงพันธุกรรมมีลักษณะการทำงานแบบกระบวนการวนรอบ (Iterative Process) คือ การทำงานแบบวนกลับไปทำใหม่เรื่อยๆ โดยที่แต่ละรอบเรียกว่ารุ่น (Generation) โดยทั่วไปจำนวนรุ่นจะอยู่ในช่วง 50 ถึงมากกว่า 500 รุ่น ชุดของรุ่นทั้งหมดเรียกว่า การทำงาน (Run) โดยผลลัพธ์ที่ได้หลังการทำงานจะเป็นผลลัพธ์ที่มีค่าความเหมาะสมสูงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูป 2.11 ขั้นตอนพื้นฐานของขั้นตอนวิธีเชิงพันธุกรรม

119268
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากขั้นตอนวิธีเชิงพันธุกรรมใช้ขั้นตอนวิธีค้นหาแบบสุ่ม ซึ่งค่าความเหมาะสมของโครโมโซมในแต่ละรุ่น จะเสถียรขึ้นในรุ่นหลังๆ (เมื่อโครโมโซมมีค่าความเหมาะสมสูงมาก) ซึ่งเมื่อมาถึงจุดนี้ ก็ให้ออกจากขั้นตอนวิธีเชิงพันธุกรรม แต่ถ้าค่าความเหมาะสมยังน้อยอยู่หรือยังไม่เสถียรหรือยังไม่พึงพอใจต่อผลลัพธ์ที่ได้ ก็ให้กลับไปทำขั้นตอนวิธีเชิงพันธุกรรมต่อ

2.2.6 ตัวอย่าง

ตัวอย่างต่อไปนี้จะช่วยให้เข้าใจการทำงานของขั้นตอนวิธีเชิงพันธุกรรมยิ่งขึ้น ปัญหาคือการหาค่าที่มากที่สุดของ $f(15x - x^2)$ ซึ่ง x มีค่าอยู่ระหว่าง 0 ถึง 15 โดยจะสมมติว่า x เป็นข้อมูลชนิดจำนวนเต็มเท่านั้น ดังนั้นวิธีการเข้ารหัสที่เหมาะสมก็คือการเข้ารหัสแบบเลขฐานสอง โดยที่หนึ่งโครโมโซมสามารถมีได้ 4 ยีนดังตาราง 2.2

ตาราง 2.2 ตัวอย่างการเข้ารหัสที่ใช้กับปัญหา $f(15x - x^2)$

Integer	Binary code	Integer	Binary code	Integer	Binary code
1	0 0 0 1	6	0 1 1 0	11	1 0 1 1
2	0 0 1 0	7	0 1 1 1	12	1 1 0 0
3	0 0 1 1	8	1 0 0 0	13	1 1 0 1
4	0 1 0 0	9	1 0 0 1	14	1 1 1 0
5	0 1 0 1	10	1 0 1 0	15	1 1 1 1

สมมติให้ในแต่ละรุ่นมีจำนวนโครโมโซม N เท่ากับ 6 ตัว ความน่าจะเป็นในการเกิดการครอสโอเวอร์ (p_c) เท่ากับ 0.7 และความน่าจะเป็นที่จะเกิดมิวเทชัน (p_m) เท่ากับ 0.001 (ค่า (p_c) และ (p_m) ที่เลือกเป็นค่าที่ใช้โดยส่วนใหญ่ในขั้นตอนวิธีเชิงพันธุกรรม) จากนั้นกำหนด fitness function ดังสมการ 2.3

$$f(x) = 15x - x^2 \tag{2.3}$$

ขั้นตอนวิธีเชิงพันธุกรรมกำหนดค่าเริ่มต้นให้แก่โครโมโซมทุกๆ ตัวในรุ่นแรก โดยการใส่ค่าลงไปในโครโมโซม 6 ตัว ตัวละ 4 บิทด้วยการสุ่มตัวเลข 0 และ 1

ในความเป็นจริง การแก้ปัญหาโดยทั่วไปควรมีจำนวนโครโมโซมในแต่ละรุ่นประมาณ 1,000 ตัวขึ้นไป

ขั้นตอนต่อไปคือการคำนวณค่าความเหมาะสมสำหรับแต่ละโครโมโซมซึ่งผลลัพธ์ที่ได้จะแสดงในตารางต่อไปจากนี้ ค่าเฉลี่ยค่าความเหมาะสมในรุ่นตั้งต้นนี้มีค่าเท่ากับ 36 ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการต่อจากนี้จะเป็นการปรับปรุงค่านี้ให้สูงขึ้นโดยการใช้หลักการคัดเลือกตามธรรมชาติต่อไป

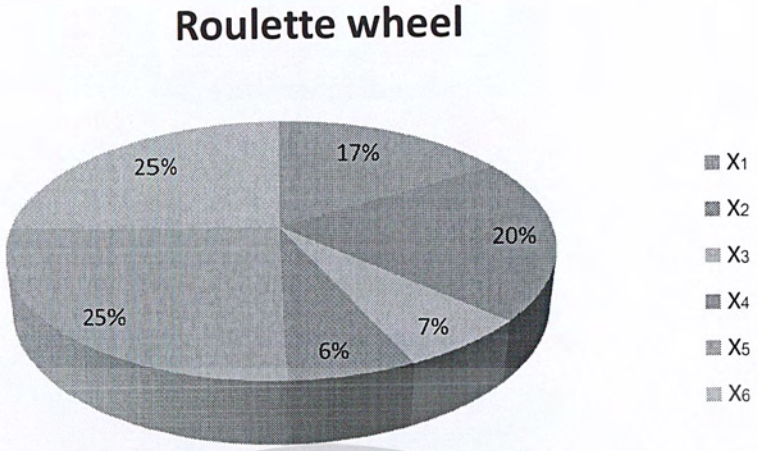
ตาราง 2.3 ตัวอย่างตารางค่า fitness

โครโมโซมlabel	โครโมโซมstring	Decoded integer	โครโมโซม fitness	Fitness ratio, %
X1	1 1 0 0	12	36	16.5
X2	0 1 0 0	4	44	20.2
X3	0 0 0 1	1	14	6.4
X4	1 1 1 0	14	14	6.4
X5	0 1 1 1	7	56	25.7
X6	1 0 0 1	9	54	24.8

จากตาราง 2.3 ในคอลัมน์สุดท้าย ได้แสดงให้เห็นถึงอัตราส่วนของค่าความเหมาะสมของโครโมโซมแต่ละตัวต่อผลรวมของค่าความเหมาะสมทั้งหมดในรุ่นนั้น ซึ่งจะใช้อัตราส่วนนี้ในการคำนวณหาโอกาสในการถูกเลือกของโครโมโซมแต่ละตัว ดังนั้นโครโมโซม X5 และ X6 จะมีโอกาสถูกเลือกมากกว่าตัวอื่นๆ ขณะที่โครโมโซม X3 และ X4 จะมีโอกาสถูกเลือกน้อยกว่าตัวอื่นๆ ซึ่งผลลัพธ์ที่ได้จากการคัดเลือก คือการทำให้ค่าเฉลี่ยของค่าความเหมาะสมมีค่าสูงขึ้นในประชากรรุ่นต่อไป

วิธีที่นิยมใช้ในการคัดเลือกวิธีหนึ่งก็คือการคัดเลือกแบบวงล้อรูเลทท์ โดยการแบ่งพื้นที่บนวงล้อรูเลทท์ให้แก่โครโมโซมแต่ละตัว ซึ่งโครโมโซมตัวมีค่าความเหมาะสมสูงก็จะมีพื้นที่บนวงล้อรูเลทท์มาก จากนั้นก็เลือกโครโมโซมขึ้นมาโดยการสุ่มค่าเลือกค่าบนเส้นรอบของวงล้อรูเลทท์[1,100] (เปรียบเสมือนการหมุนวงล้อรูเลทท์) ถ้าตกบนพื้นที่ของโครโมโซมตัวไหน โครโมโซมตัวนั้นก็คือตัวที่ถูกเลือกขึ้นมา

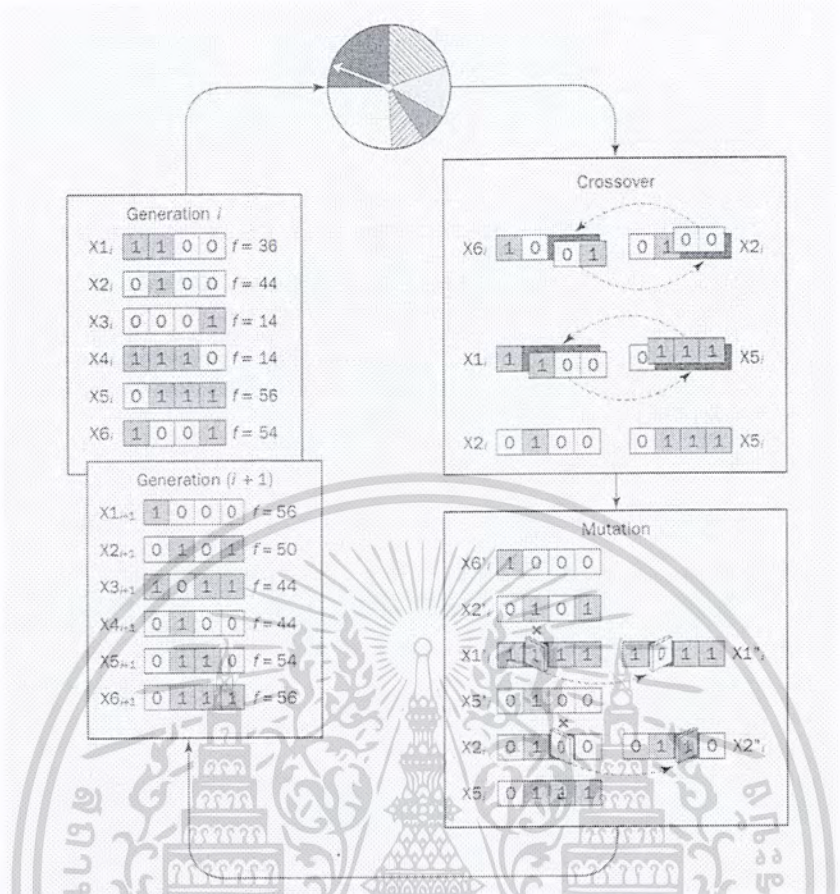
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.12 ตัวอย่างการคัดเลือกแบบวงล้อรูเลทท์ที่ใช้กับปัญหา $f(15x - x^2)$

จากตัวอย่างนี้ เราได้กำหนดค่าเริ่มต้นให้แก่โครโมโซมทั้ง 6 ตัว และเพื่อให้ได้โครโมโซมในรุ่นถัดไปที่มีจำนวนโครโมโซมเท่าเดิม จึงควรจะมีการหมุนวงล้อรูเลทท์ทั้งหมด 6 ครั้ง โดยสมมุติให้ใน 2 ครั้งแรกโครโมโซมที่ถูกเลือกคือ X2 และ X6 มาเป็นโครโมโซมตั้งต้น (Parents) และ 2 ครั้งถัดมาได้ X1 และ X5 2 ครั้งสุดท้ายได้ X2 และ X5 จากนั้นนำโครโมโซมแต่ละคู่มาเข้าสู่การครอสโอเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.13 ตัวอย่างขั้นตอนของขั้นตอนวิธีเชิงพันธุกรรมที่ใช้กับปัญหา $f(15x - x^2)$

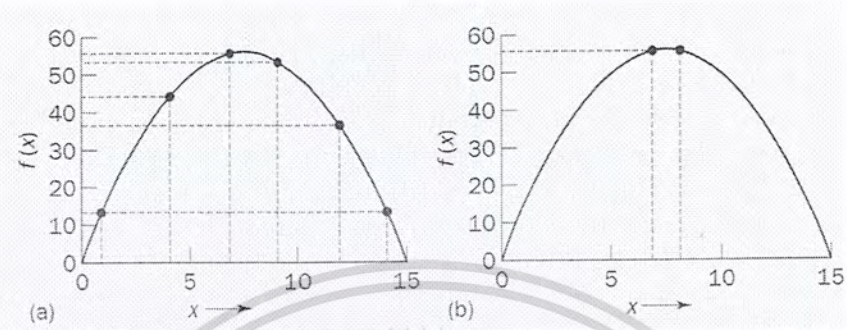
การครอสโอเวอร์ในตัวอย่างนี้ ได้กำหนดค่าความน่าจะเป็นในการเกิดการครอสโอเวอร์เป็น 0.7 โดยกระบวนการในขั้นแรกคือการสุ่มหาจุดตัดที่จะเกิดการครอสโอเวอร์ของโครโมโซมทั้งคู่ จากนั้นก็แลกเปลี่ยน กลุ่มของยีนหลังจุดตัดนั้นซึ่งกันและกัน ซึ่งในตัวอย่างนี้สมมุติให้โครโมโซม X6 และ X2 มีจุดตัดอยู่ที่หลังยีนตัวที่ 2 ซึ่งหลังจากผ่านการครอสโอเวอร์แล้ว ก็จะได้โครโมโซมตัวใหม่ขึ้นมา 2 ตัว

หากคู่โครโมโซมที่ถูกเลือกมาไม่เกิดการครอสโอเวอร์ คู่โครโมโซมคู่นั้นจะถูกคัดลอกออกมาเป็นโครโมโซมตัวใหม่ 2 ตัวที่ไม่มีการเปลี่ยนแปลงใดๆ หรือที่เรียกว่า การสืบพันธุ์ ในที่นี้สมมติให้โครโมโซม X2 และ X5 ซึ่งถูกเลือกมา แต่ไม่เกิดการครอสโอเวอร์ จึงเกิดการสืบพันธุ์ขึ้น

เมื่อผ่านการครอสโอเวอร์มาแล้ว ต่อมาจะเข้าสู่มิวเทชัน โดยกำหนดให้ค่าความน่าจะเป็นในการเกิดมิวเทชันมีค่าประมาณ 0.01 ถึง 0.001 ซึ่งในขั้นตอนแรกคือการสุ่มเลือกยีนภายในโครโมโซมที่เกิดมิวเทชันและทำการเปลี่ยนค่าในยีนนั้น ในตัวอย่างนี้สมมติให้โครโมโซมX1 และ X2 เกิดมิวเทชันที่ยีนตัวที่ 2 และ 3 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผ่านทั้งกระบวนการเหล่านี้มาทั้งหมดก็ถือว่าเป็นการจบกระบวนการของขั้นตอนวิธีเชิงพันธุกรรมใน 1 รุ่น ซึ่งจะวนกลับไปทำในขั้นตอนการคัดเลือกไปเรื่อยๆ จนกว่าจะได้ผลลัพธ์เป็นที่พึงพอใจ



รูป 2.14 เปรียบเทียบผลลัพธ์จากตัวอย่างระหว่างประชากรรุ่นแรกและรุ่นสุดท้าย

จากรูป 2.14 จะเห็นได้ว่าผลลัพธ์ที่ได้เมื่อนำไปเทียบกับโครโมโซมในรุ่นแรก จะเห็นได้ว่าในโครโมโซมรุ่นแรก ค่าความเหมาะสมมีการกระจัดกระจาย ซึ่งมีทั้งโครโมโซมตัวที่มีค่าความเหมาะสม น้อยและตัวที่มีค่ามาก แต่ผลลัพธ์สุดท้ายโครโมโซมจะมีเฉพาะโครโมโซมที่มีค่า fitness สูงเท่านั้น

2.2.7 TSP (Traveling Salesman Problem)

TSP เป็นปัญหาการจัดลำดับประเภทหนึ่งจากจุดประสงค์คือการหาเส้นทางที่สั้นที่สุดบนกราฟแบบที่ไม่มีทิศทาง (Undirected Graph) ซึ่งแทนเมืองหรือโหนดต่างๆ ที่จะต้องผ่าน TSP จะเริ่มที่โหนดๆ หนึ่ง แล้วจึงเดินผ่านไปยังทุกๆ โหนด โดยการผ่านแต่ละโหนด จะผ่านได้แค่โหนดละหนึ่งครั้งเท่านั้น และสุดท้ายก็กลับมาที่โหนดเริ่มต้น

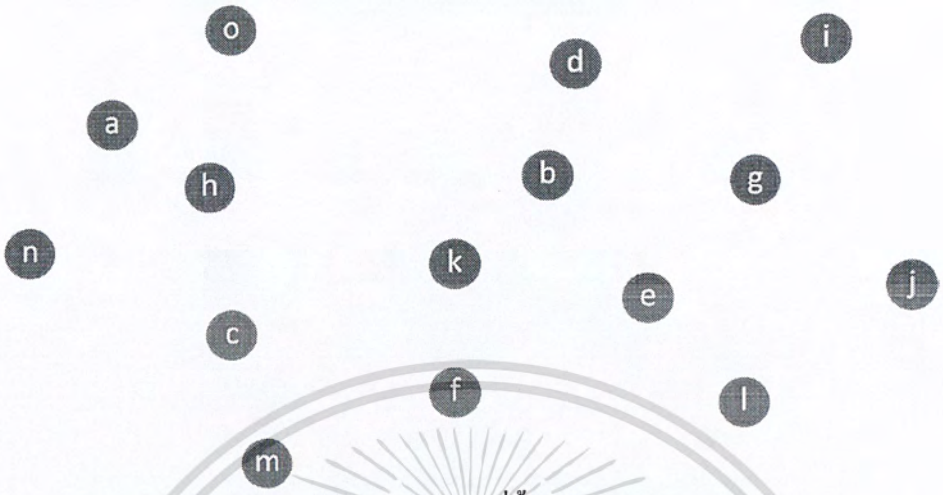
ต่อไปนี้จะเป็นการจำลองปัญหา TSP ขึ้นมาเพื่อแสดงให้เห็นถึงขั้นตอนของขั้นตอนวิธีเชิงพันธุกรรมโดยกำหนดให้ตำแหน่งของเมืองต่างๆ เป็นดังตาราง 2.4

ตาราง 2.4 ตำแหน่งที่ตั้งของเมืองต่างๆ

Cities names	Coordinates
a, b, c, d,	(110,54), (236,110), (153,151), (227,49),
e, f, g, h,	(307,176), (220,211), (341,90), (149,91),
I, j, k, l,	(335,40), (371,150), (218,161), (334,239),
m, n, o	(148,227), (49,128), (183,39)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งถ้านำพิกัดเหล่านี้มาสร้างให้อยู่ในรูปของกราฟ ก็จะได้ตำแหน่งดังรูป 2.15

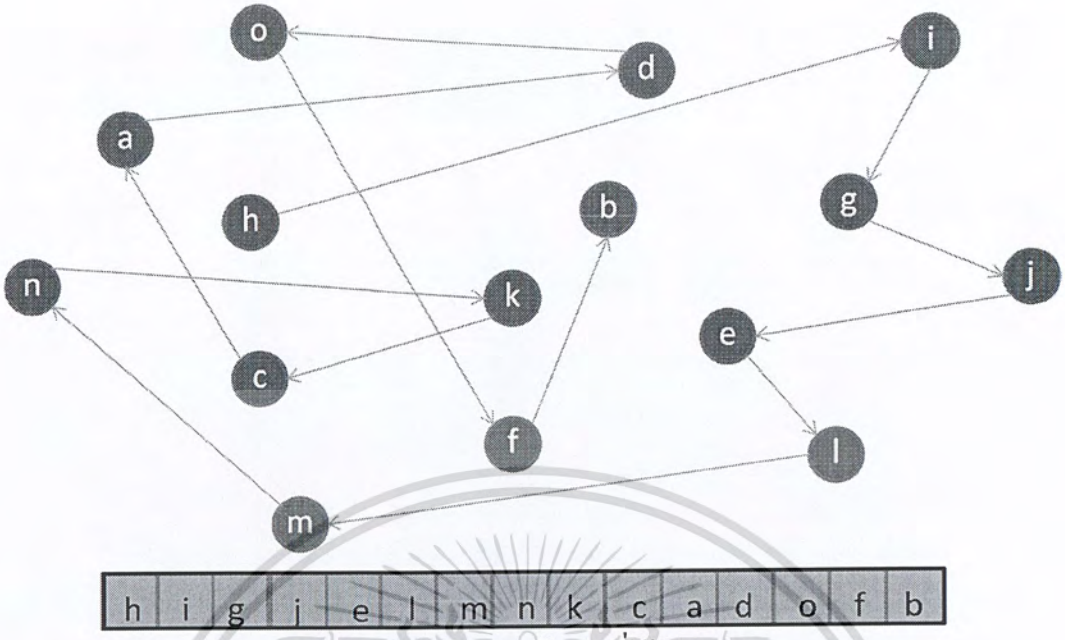


รูป 2.15 จำลองตำแหน่งที่ตั้งของเมืองต่างๆ

การเข้ารหัสของปัญหานี้ลงโครโมโซมจะเป็นการเข้ารหัสแบบเรียงสับเปลี่ยน เนื่องจากเป็นปัญหาของการจัดลำดับ โดยการเรียงชื่อเมืองต่างๆ ที่เดินผ่านลงไปในแต่ละยีนของโครโมโซม ซึ่งเมืองที่ถูกผ่านก่อนก็จะถูกใส่ลงไปก่อน

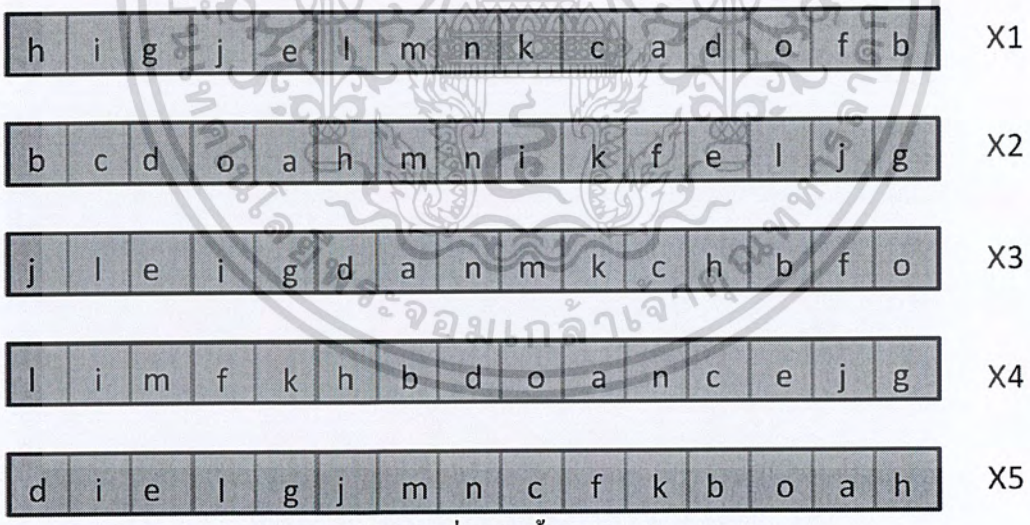
ในตัวอย่างนี้จะกำหนดให้มีโครโมโซมตั้งต้น 5 ตัวในประชากร 1 รุ่นซึ่งในการหาคำตอบจริงๆ เพื่อให้ได้คำตอบที่ถูกต้องแม่นยำ จะต้องมากกว่าจำนวนนี้มากหลายร้อยเท่า แต่เนื่องด้วยเราต้องการเพียงแค่ศึกษาขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมเท่านั้น การกำหนดเพียงเท่านี้ก็เพียงพอต่อการแสดงขั้นตอนต่างๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.16 ตัวอย่างการเข้ารหัสที่ใช้กับ TSP

เมื่อกำหนดรูปแบบการเข้ารหัสเรียบร้อยแล้ว จึงสุ่มการเรียงลำดับเส้นทางการผ่านของแต่ละโหนดลงไปในโครโมโซมทั้ง 5 ตัว



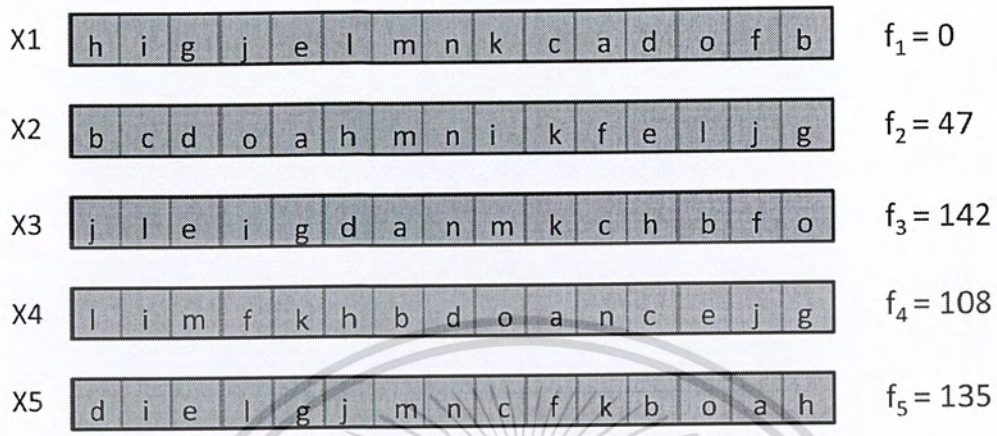
รูป 2.17 โครโมโซมที่ถูกสุ่มขึ้นมาเป็นประชากรรุ่นแรก

เมื่อได้โครโมโซมมาครบจำนวนประชากรในหนึ่งรุ่นแล้ว ต่อไปคือการหาค่าความเหมาะสมโดยใช้ฟังก์ชันค่าความเหมาะสมดังสมการ 2.4

$$f_i = D_{max} - D_i \tag{2.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย D_{max} คือ ระยะทางที่มีค่าสูงที่สุดของโครโมโซมในประชากรรุ่นนั้น และ D_i คือ ระยะทางที่โครโมโซมแต่ละตัวใช้ในประชากรรุ่นนั้น

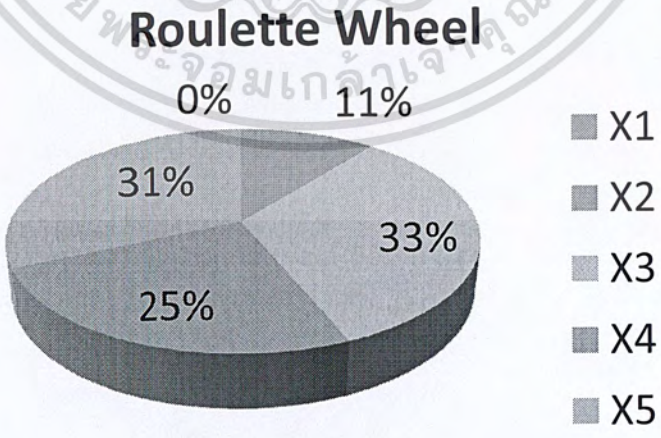


รูป 2.18 ค่าความเหมาะสมของโครโมโซมแต่ละตัว

ขั้นตอนต่อไปคือการคัดเลือกซึ่งใช้การคัดเลือกแบบวงล้อรูเลทท์โดยใช้สมการ 2.5

$$P_i = f_i \div \sum f_i \tag{2.5}$$

ซึ่งจะพบว่าสมการดังกล่าวก็คือสมการที่ใช้ในการหาอัตราส่วนของค่าความเหมาะสมนั่นเอง หลังจากนั้นนำโครโมโซมทั้งหมดในประชากรรุ่นมาใส่ลงในวงล้อรูเลทท์ตามอัตราส่วนค่าความเหมาะสมที่คำนวณได้ดังรูป 2.19



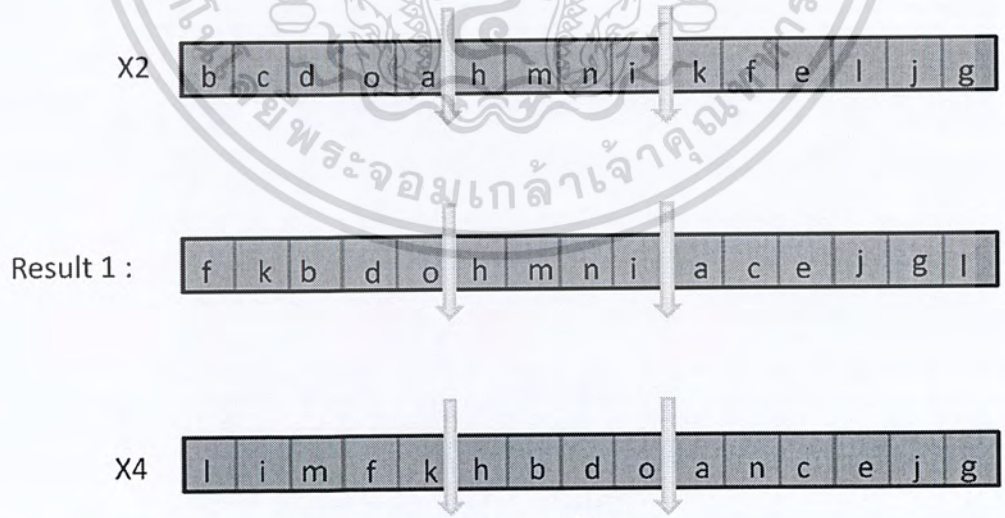
รูป 2.19 ผลลัพธ์หลังจากนำโครโมโซมใส่ลงในวงล้อรูเลทท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการสุ่มเลือกโครโมโซมจากวงล้อรูเลทท์ซึ่งเปรียบเสมือนการหมุนวงล้อรูเลทท์จนได้โครโมโซมขึ้นมา 3 ตัว ซึ่งในที่นี้ กำหนดให้ 2 ตัวแรก จะถูกนำมาผ่านการครอสโอเวอร์และตัวสุดท้ายจะนำผ่านมิวเทชัน ทั้งนี้จะขึ้นอยู่กับความน่าจะเป็นของการเกิดการครอสโอเวอร์และมิวเทชันซึ่งกำหนดให้เป็น 0.9 และ 0.04 ตามลำดับ

สมมติให้โครโมโซม 2 ตัวแรกที่ถูกเลือกมาทำการครอสโอเวอร์คือโครโมโซม X2 กับ X4 และตัวสุดท้ายที่จะถูกนำไปทำมิวเทชันคือ X3 จากนั้นทำการสุ่มค่าขึ้นมาเพื่อเทียบกับโอกาสการครอสโอเวอร์และมิวเทชัน ซึ่งในที่นี้สมมุติให้ค่าที่สุ่มได้เป็น 0.829 สำหรับการครอสโอเวอร์ และ 0.035 สำหรับมิวเทชัน ซึ่งค่าทั้งสองอยู่ในขอบเขตของการเกิดการครอสโอเวอร์และการเกิดมิวเทชัน

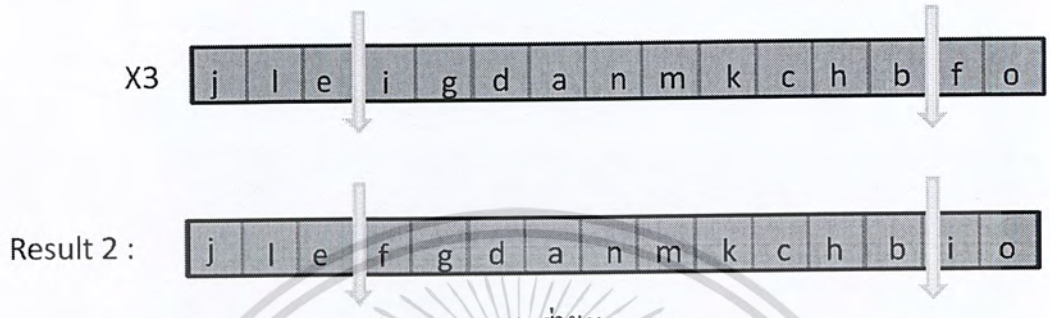
การครอสโอเวอร์สำหรับตัวอย่างนี้จะต้องคำนึงถึงชื่อเมืองต่างๆ ภายในโครโมโซมด้วย เนื่องจากเป็นการเรียงลำดับที่ไม่มีการวนกลับมาที่เดิม ทำให้ชื่อเมืองทั้งหมดภายในโครโมโซมจะต้องไม่ซ้ำ จึงมีการครอสโอเวอร์ที่มีความพิเศษมากขึ้นเพื่อใช้แก้ปัญหานี้ ซึ่งในขั้นแรกก็คือการสุ่มหาจุดตัด 2 จุด สำหรับโครโมโซมทั้งคู่ และทำการนำส่วนของโครโมโซมตัวแรกที่อยู่ระหว่างจุดตัดมาใส่ลงในโครโมโซมตัวใหม่ จากนั้นจึงไล่ใส่ส่วนของโครโมโซมตัวที่ 2 จากทางขวาจุดตัดด้านขวาสุด ไล่ไปเรื่อยๆ จนสุดแล้วกลับไปเริ่มใส่ตรงด้านซ้ายสุด โดยการที่จะใส่ส่วนของโครโมโซมตัวที่ 2 ลงไปนั้น จะต้องตรวจสอบกับส่วนของโครโมโซมของตัวแรกที่ถูกใส่ลงมาก่อนหน้าด้วยว่าซ้ำหรือไม่ ถ้าซ้ำก็ให้ข้ามไปใส่ตัวถัดไป ผลลัพธ์สุดท้ายก็จะได้โครโมโซมตัวใหม่ขึ้นมาหนึ่งตัว



รูป 2.20 การครอสโอเวอร์ที่ใช้กับ TSP

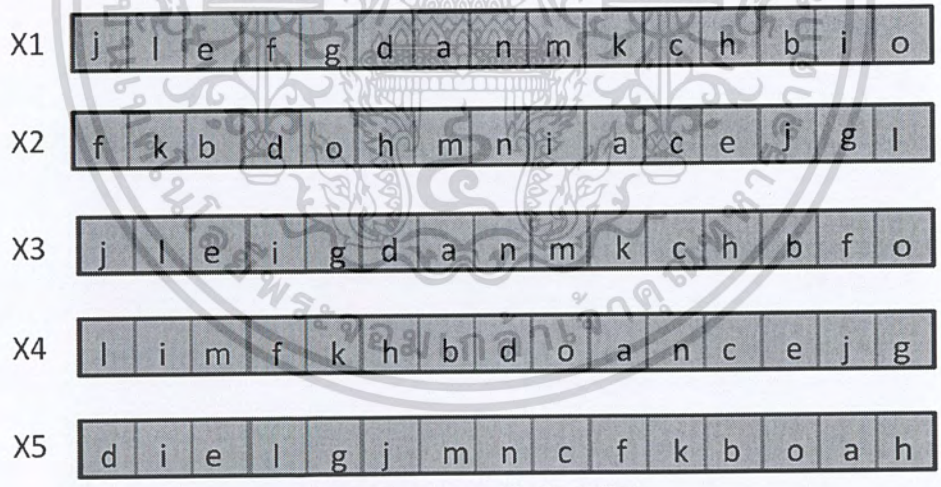
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการต่อไปคือมิวเทชันซึ่งเช่นเดียวกับการครอสโอเวอร์ที่ต้องรักษาจุดประสงค์ของปัญหานี้ไว้ โดยในขั้นแรกคือการสุ่มเลือกจุดตัดภายในโครโมโซมที่ถูกเลือก ขึ้นมา 2 ตำแหน่ง จากนั้นทำการแลกเปลี่ยนยีนที่อยู่หลังจะจุดตัดนั้น ผลลัพธ์สุดท้ายก็จะได้โครโมโซมตัวใหม่ขึ้นมาอีกหนึ่งตัว



รูป 2.21 มิวเทชันที่ใช้กับ TSP

เมื่อได้โครโมโซมตัวใหม่ 2 ตัวมาแล้ว ต่อไปคือการนำไปแทนที่โครโมโซมที่มีค่าความเหมาะสมน้อยที่สุด 2 ตัว ในประชากรรุ่นเดิม ซึ่งเมื่อแทนแล้วก็จะได้ประชากรรุ่นใหม่ ซึ่งจะนำวนกลับไปเข้าสู่การคัดเลือก ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งได้ผลลัพธ์เป็นที่พึงพอใจ



รูป 2.22 population ในรุ่นต่อไป

2.2.8 ข้อดีและข้อเสียของขั้นตอนวิธีเชิงพันธุกรรม

2.2.8.1 ข้อดี

จุดเด่นของขั้นตอนวิธีเชิงพันธุกรรม คือการแก้ปัญหาในเรื่องของค่าดีที่สุดเฉพาะบริเวณ เนื่องจากมีมิวเทชันทำให้ขอบเขตของผลลัพธ์ทั้งหมด มีโอกาสที่จะถูกทดลองว่าเป็นผลลัพธ์ที่ดีที่สุดหรือไม่ และมีการใช้กฎของความน่าจะเป็นในการเลือกผลลัพธ์ ซึ่งจะทำให้การหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์เป็นไปอย่างมีประสิทธิภาพและรวดเร็วมากขึ้น นอกจากนี้กระบวนการทางขั้นตอนวิธีเชิงพันธุกรรมเป็นกระบวนการที่ค่อนข้างง่าย ไม่ซับซ้อน แต่สามารถให้ผลลัพธ์ที่ดีได้

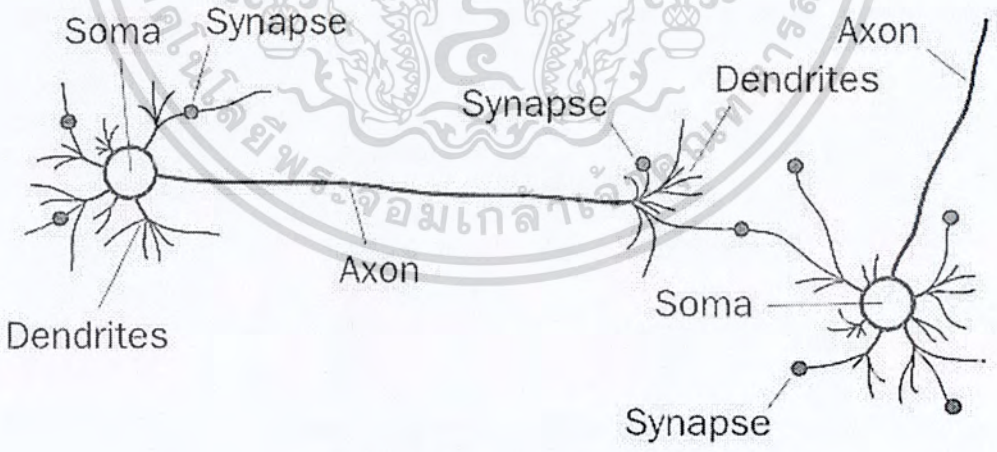
2.2.8.2 ข้อเสีย

เนื่องจากขั้นตอนวิธีเชิงพันธุกรรมเป็นเพียงกระบวนการหรือหลักการพื้นฐานเท่านั้น ซึ่งถ้าจะนำไปใช้งานในการแก้ปัญหาจริงๆ แล้ว ยังต้องมีการนำไปพัฒนาต่ออีกค่อนข้างเยอะเพื่อให้รองรับกับปัญหานั้น และยังมีข้อจำกัดในเรื่องของความยาวของโครโมโซม ซึ่งถ้ายาวมากเกินไปหรือโครโมโซมมีการเก็บข้อมูลมากเกินไป จะทำให้ใช้เวลาในการหาผลลัพธ์นานมากขึ้น นอกจากนี้ ถ้าประมวลผลเพื่อหาผลลัพธ์โดยการใช้จำนวนรอบที่น้อยเกินไป ก็อาจจะทำให้ผลลัพธ์ที่ได้นั้น ไม่ใช่ผลลัพธ์ที่ดีที่สุด

2.3 โครงข่ายประสาทเทียม

2.3.1 ความหมายของโครงข่ายประสาทในสิ่งมีชีวิต (Biological Neural Network)

โครงข่ายประสาท คือ รูปแบบโครงข่ายที่สมองใช้ในการทำงาน ซึ่งในสมองมีเซลล์ประสาท (Neuron) จำนวนมากเชื่อมต่อกันด้วยจุดประสานประสาท (Synapse) จนกลายเป็นเครือข่ายขนาดใหญ่ สมองมนุษย์โดยทั่วไปมีเซลล์ประสาทประมาณ 10 พันล้านเซลล์และมีจุดประสานประสาทประมาณ 60 พันล้านจุดซึ่งเมื่อทำงานร่วมกันจะมีความเร็วมากกว่าคอมพิวเตอร์ในยุคปัจจุบัน



รูป 2.23 โครงข่ายประสาทในสิ่งมีชีวิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.23 เซลล์ประสาทซึ่งถือเป็นหน่วยปฏิบัติการหลักของระบบประสาท ประกอบด้วยโครงสร้างแบบง่ายๆ 2 ส่วน ดังนี้

1) ตัวเซลล์ (Soma)
2) แขนงเซลล์ประสาท (Neurites) ซึ่งเป็นเส้นใยที่ยื่นออกมาจากตัวเซลล์ แบ่งออกเป็น 2 ประเภท ดังนี้
   2.1) แอกซอน (Axon) เป็นส่วนถ่ายทอด นำส่ง และปลดปล่อยสัญญาณประสาทไปยังเซลล์ประสาทตัวอื่น ซึ่งเป็นเหมือน output ของเซลล์
   2.2) เดนไดรต์ (Dendrite) เป็นส่วนที่รับสัญญาณประสาทจากเซลล์ประสาทอื่นๆ เข้าไปสู่ตัวเซลล์ประสาท ซึ่งเป็นเหมือนอินพุท (input) ของเซลล์

บริเวณที่แอกซอนของเซลล์ประสาทตัวหนึ่งไปบรรจบกับเดนไดรต์ของเซลล์ประสาทตัวอื่นเพื่อส่งสัญญาณสื่อสารกันนี้เองที่เรียกว่าจุดประสานประสาท

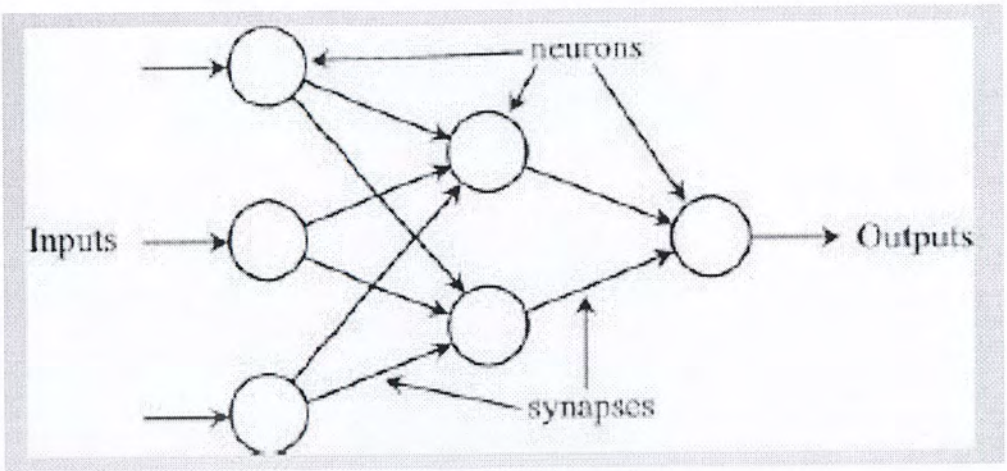
2.3.2 ความหมายของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial Neural Network (ANN) หรือ Neural Network หรือ Neural Net (NN) เป็นการจำลองโครงข่ายประสาทของสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้ในรูปแบบของการรู้จำแบบ (Pattern Recognition) ความสามารถการคาดเดาอย่างมีเหตุมีผล (Generalize) และความสามารถในการเรียนรู้ (Learning) ซึ่งหากให้ข้อมูลตัวอย่างกับโครงข่ายประสาทเทียมเพื่อใช้ในการเรียนรู้มากพอในระดับหนึ่ง โครงข่ายประสาทเทียมจะสามารถรู้จำแบบของข้อมูลตัวอย่าง ทำให้สามารถคาดเดาเอาท์พุท (Output) ของอินพุท (Input) ที่ไม่เคยพบแต่มีลักษณะคล้ายกับข้อมูลตัวอย่างได้ เช่น คาดเดาตัวอักษร ตรวจสอบเสียงพูดของมนุษย์ ตรวจหาวัตถุระเบิด เป็นต้น ยิ่งไปกว่านั้นโครงข่ายประสาทเทียมยังสามารถตรวจสอบสิ่งที่ผู้เชี่ยวชาญไม่สามารถตรวจสอบพบได้ เช่น การตรวจสอบรูปแบบของข้อมูลการใช้บัตรเครดิตเพื่อตรวจจับการทุจริต เป็นต้น

2.3.3 ส่วนประกอบของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมประกอบด้วยโหนดหรือเรียกอีกอย่างว่า นิวรอน (Neuron) ซึ่งจำลองมาจากเซลล์ประสาท แต่ละโหนดมีอินพุทซึ่งจำลองมาจากเดนไดรต์และเอาท์พุทซึ่งจำลองมาจากแอกซอน เชื่อมต่อกันด้วยเส้นเชื่อมที่มีค่าน้ำหนัก (Weighted Link) ซึ่งจำลองมาจากจุดประสานประสาทดังรูป 2.24

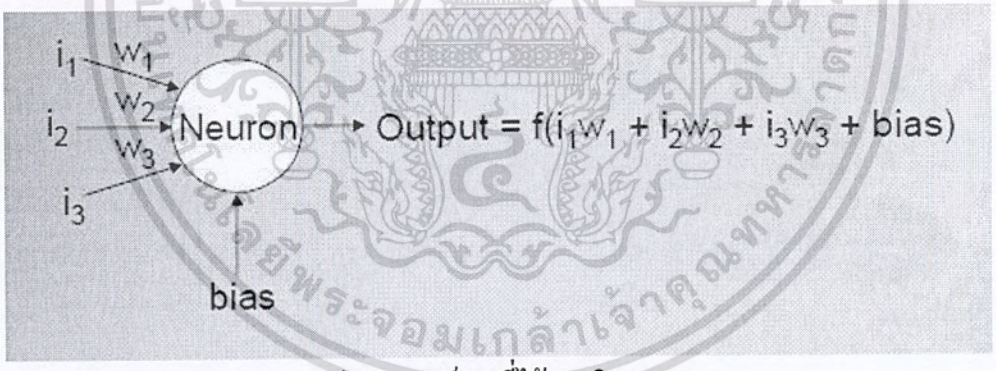
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.24 โครงข่ายประสาทเทียม

2.3.4 หลักการทำงานเบื้องต้นของโครงข่ายประสาทเทียม

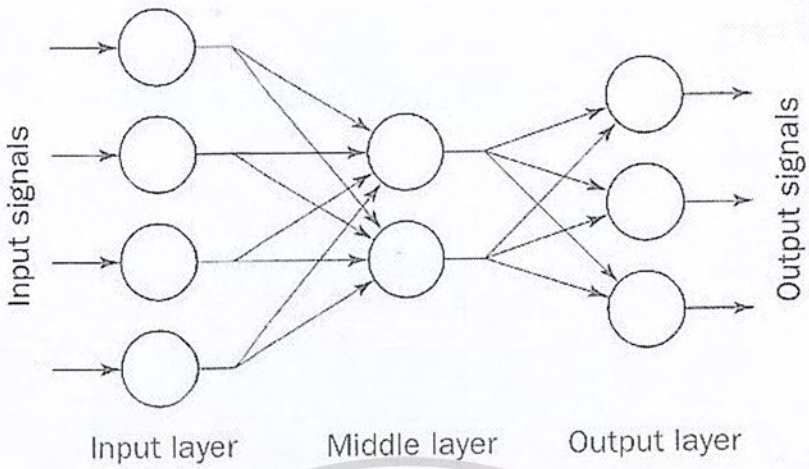
นิวรอนแต่ละตัวจะรับอินพุทเข้ามาเพื่อหาเอาท์พุทโดยเอาท์พุทของนิวรอนเป็นผลลัพธ์ของฟังก์ชันจากผลคูณระหว่างอินพุทกับค่าน้ำหนัก (Weight) ของแต่ละเส้นทาง (Link) ที่เข้าสู่นิวรอนและค่าไบอัส (Bias) หรือที่เรียกว่าค่ากระตุ้น (Threshold) ดังรูป 2.25 ซึ่งฟังก์ชันนี้เรียกว่าฟังก์ชันกระตุ้น (Activation Function)



รูป 2.25 เอาท์พุทที่ได้จากนิวรอน

โครงข่ายประสาทเทียมแบ่งตำแหน่งของนิวรอนออกเป็นชั้น (Layer) ทำให้เอาท์พุทของนิวรอนในชั้นแรกจะเป็นอินพุทของนิวรอนในชั้นถัดไป ดังรูป 2.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

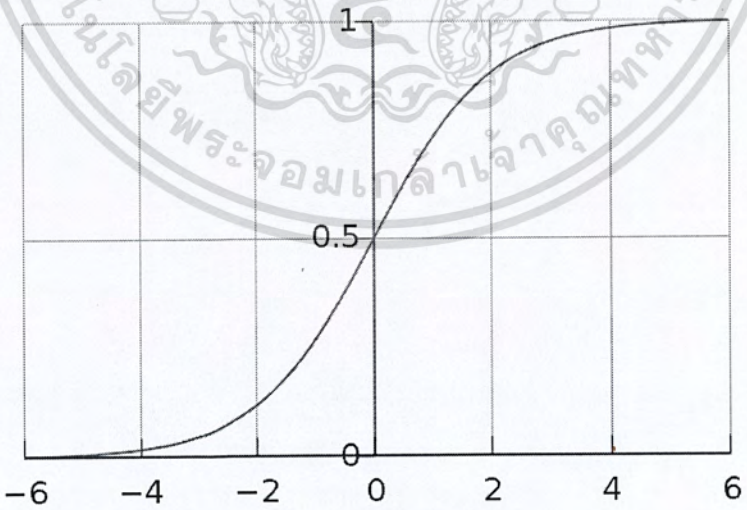


รูป 2.26 โครงสร้างทั่วไปของโครงข่ายประสาทเทียม

2.3.5 ฟังก์ชันกระตุ้น (Activation Function)

ฟังก์ชันกระตุ้นเป็นฟังก์ชันที่ใช้หาเอาท์พุทของนิวรอนแต่ละตัวโดยใช้ผลคูณระหว่างอินพุทกับค่าน้ำหนักของแต่ละเส้นทางที่เข้าสู่นิวรอนและค่ากระตุ้น

โครงข่ายประสาทเทียมส่วนมากใช้ฟังก์ชันซิกมอยด์ (Sigmoid Function) เป็นฟังก์ชันกระตุ้นที่เป็นฟังก์ชันต่อเนื่องที่มีอนุพันธ์ (Derivative) ซึ่งมีค่าแบ่งออกเป็น 2 ช่วง คือ ช่วงที่มีเป็นค่าบวกและช่วงที่มีค่าเป็นลบ โดยมีการแบ่งระหว่างทั้งสองช่วงเพียงจุดเดียว นอกจากนี้ยังเป็นฟังก์ชันที่มีขอบเขตซึ่งไม่มีผลลัพธ์ใดๆ จะมีค่าเท่ากับหรือเกินออกไปจากขอบเขตได้ ดังรูป 2.27



รูป 2.27 กราฟฟังก์ชันโลจิสติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันซิกมอยด์ที่นิยมใช้กันมาก คือ ฟังก์ชันโลจิสติก (Logistic Function) ดังสมการ 2.6

$$f(x) = 1/(1 + e^{-x}) \tag{2.6}$$

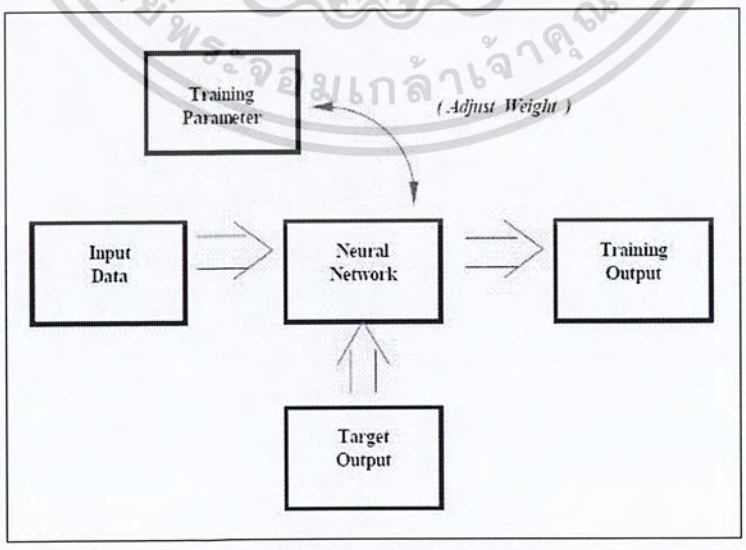
ฟังก์ชันซิกมอยด์อื่นๆ ที่ใช้กัน ได้แก่ ฟังก์ชันไฮเพอร์โบลิกแทนเจนท์ (Hyperbolic Tangent Function) และฟังก์ชันอาร์คแทนเจนท์ (Arctangent Function)

นอกจากฟังก์ชันซิกมอยด์แล้ว ยังมีฟังก์ชันอื่นๆ ที่สามารถใช้เป็นฟังก์ชันกระตุ้นได้อีก ได้แก่ ฟังก์ชันขั้นบันได (Step Function), ฟังก์ชันเครื่องหมาย (Sign Function) และฟังก์ชันเชิงเส้น (Linear Function)

2.3.6 การเรียนรู้ (Learning)

การเรียนรู้ คือ กระบวนการที่โครงข่ายประสาทเทียมถูกฝึกเพื่อปรับเปลี่ยนค่าน้ำหนัก เนื่องจากค่าน้ำหนักเป็นตัวแปรสำคัญสำหรับโครงข่ายประสาทเทียม เพราะค่าน้ำหนักที่เหมาะสมจะทำให้โครงข่ายประสาทเทียมสามารถหาค่าเอาท์พุทได้อย่างมีเหตุผล ซึ่งการจะได้ค่าน้ำหนักที่เหมาะสมมานั้น จำเป็นต้องให้โครงข่ายประสาทเทียมเรียนรู้ด้วยชุดข้อมูลตัวอย่างซึ่งโครงข่ายประสาทเทียมจะปรับเปลี่ยนค่าน้ำหนักไปเรื่อยๆ จนกว่าจะได้ค่าเอาท์พุทที่น่าพอใจ การเรียนรู้แบ่งออกเป็น 3 ประเภท ดังนี้

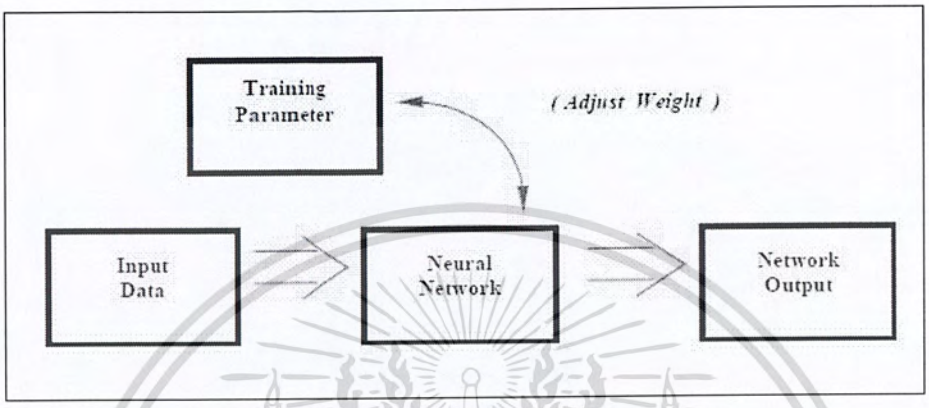
1) การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นการเรียนรู้ที่มีการตรวจเอาท์พุทที่ได้มาจากโครงข่ายประสาทเทียมด้วยชุดข้อมูลที่มีเอาท์พุทที่ถูกต้อง เพื่อให้โครงข่ายประสาทเทียมปรับปรุงคุณภาพโดยปรับเปลี่ยนค่าน้ำหนัก เพื่อลดความต่างระหว่างเอาท์พุทที่จะได้ในครั้งต่อไปกับเอาท์พุทที่ถูกต้อง



รูป 2.28 การเรียนรู้แบบมีผู้สอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เป็นการเรียนรู้ที่มีเฉพาะอินพุท ไม่มีการให้เอาท์พุทที่ถูกต้องแก่โครงข่ายประสาทเทียม ดังนั้นโครงข่ายประสาทเทียมต้องปรับเปลี่ยนค่าน้ำหนักเองตามแต่ละอินพุทที่ให้ไป ทำให้อินพุทเหมือนกันจะได้เอาท์พุทที่เหมือนกัน



รูป 2.29 การเรียนรู้แบบไม่มีผู้สอน

3) การเรียนรู้แบบเสริมกำลัง (Reinforcement learning) เป็นการเรียนรู้ที่ไม่มีให้ทั้งอินพุทและเอาท์พุท มักใช้การให้รางวัล (Reward) เพื่อสอนให้รู้ว่าสิ่งที่ทำไปนั้นดีมากน้อยเพียงใด

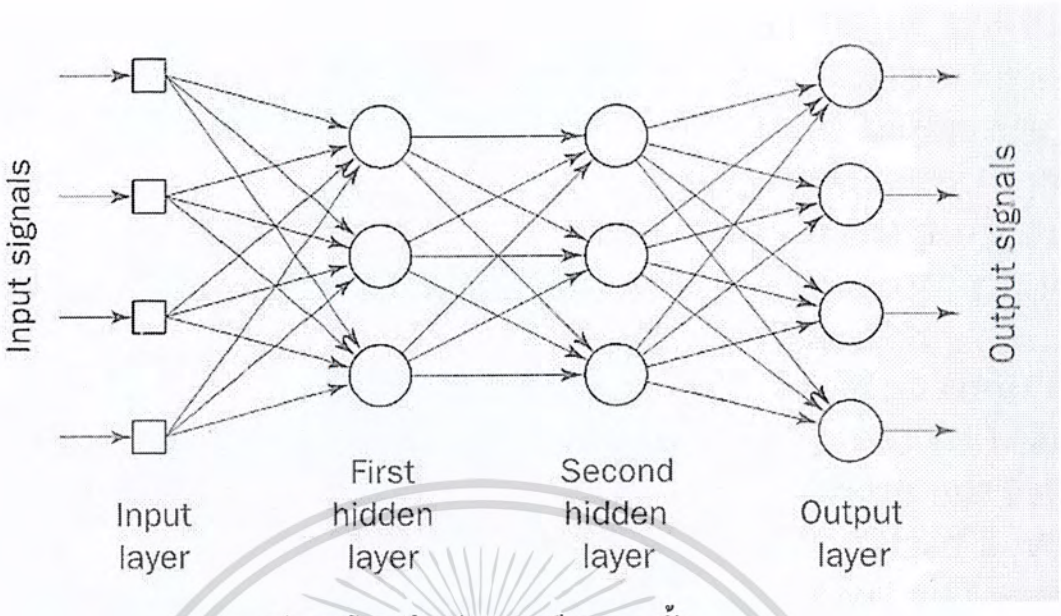
การให้ชุดข้อมูลชุดหนึ่งแก่โครงข่ายประสาทเทียม เพื่อปรับเปลี่ยนค่าน้ำหนัก ตามแต่ละข้อมูลภายในชุดข้อมูลจนครบทั้งชุดเรียกว่า 1 รอบของการคำนวณ (1 Epoch) ซึ่งโดยทั่วไปแล้วจะต้องใช้รอบของการคำนวณอย่างมาก ในการฝึกโครงข่ายประสาทเทียม

2.3.7 โครงข่ายประสาทเทียมหลายชั้น

2.3.7.1 ลักษณะทั่วไปของโครงข่ายประสาทเทียมหลายชั้น

โครงข่ายประสาทเทียมหลายชั้น (Multilayer Neural Network) คือ โครงข่ายประสาทเทียมที่มีการจัดเรียงนิวรอนเป็นชั้นตั้งแต่ 3 ชั้นขึ้นไป โดยมีชั้นอินพุท (Input Layer) เป็นชั้นแรก ชั้นซ่อนตัว (Hidden Layer) เป็นชั้นที่อยู่ตรงกลางและ ชั้นเอาท์พุท (Output Layer) เป็นชั้นสุดท้ายดังรูป 2.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.30 โครงข่ายประสาทเทียมหลายชั้น

ชั้นอินพุทมีหน้าที่รับอินพุทจากภายนอกโครงข่ายประสาทเทียมเพื่อส่งต่อให้กับชั้นซ่อนตัว ส่วนชั้นเอาท์พุทมีหน้าที่แสดงค่าเอาท์พุทของโครงข่ายประสาทเทียม

ชั้นซ่อนตัวมีหน้าที่ตรวจจับรูปแบบของชุดข้อมูล ซึ่งหากมีชั้นซ่อนตัว 1 ชั้นจะสามารถตรวจจับชุดข้อมูลที่เป็นฟังก์ชันต่อเนื่องได้ หากเพิ่มอีก 1 ชั้นจะสามารถตรวจจับได้แม้กระทั่งชุดข้อมูลที่เป็นฟังก์ชันไม่ต่อเนื่อง

โครงข่ายประสาทเทียมแต่ละชั้นประกอบด้วย 10 ถึง 1000 นิวรอนทำให้การเพิ่มชั้นขึ้น 1 ชั้น จะเพิ่มเวลาที่ใช้ในการเรียนรู้ขึ้นอย่างมาก เพราะฉะนั้นโดยทั่วไปแล้ว โครงข่ายประสาทเทียมหลายชั้นจึงมีเพียง 3 ถึง 4 ชั้นเท่านั้น

2.3.7.2 การคำนวณเอาท์พุทจากแต่ละโหนด

เริ่มจากการกำหนดค่าเริ่มต้น (Initialization) สำหรับชั้นอินพุท โดยการสุ่มค่าน้ำหนักและค่ากระตุ้นทั้งหมดในขอบเขตที่ได้จากผลการทดลองของ Haykin ในปีคริสต์ศักราช 1999 ดังสมการ 2.7 เมื่อ F_i เป็นผลรวมของอินพุทของทุกนิวรอนในชั้นซ่อนตัว

$$\left(\frac{-2.4}{F_i}, \frac{+2.4}{F_i}\right) \tag{2.7}$$

สำหรับในชั้นอื่นใช้ชุดข้อมูลอินพุท $x_1(p), x_2(p), \ldots, x_n(p)$ และเอาท์พุทที่ต้องการ $y_{d,1}(p), y_{d,2}(p), \ldots, y_{d,n}(p)$ ที่เตรียมไว้ใช้ในการเรียนรู้ครั้งที่ p โดยมีการคำนวณออกเป็น 2 แบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) คำนวณเอาท์พุทของนิวรอนแต่ละตัวในชั้นซ่อนตัวด้วยสมการ 2.8 เมื่อ n คือ จำนวนอินพุทของนิวรอน j ในชั้นซ่อนตัว, θj คือ ค่ากระตุ้นของนิวรอน j และ sigmoid คือ ฟังก์ชันซิกมอยด์

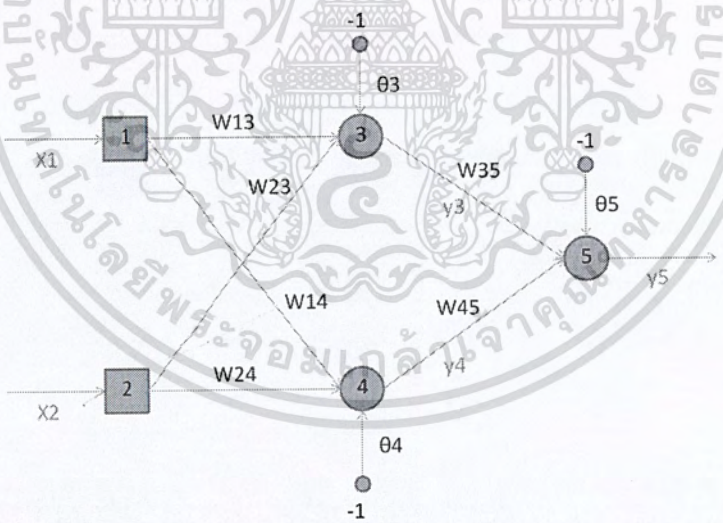
$$y_i(p) = sigmoid\left[\sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta j\right] \quad (2.8)$$

2) คำนวณเอาท์พุทของนิวรอนแต่ละตัวในชั้นเอาท์พุทด้วยสมการ 2.9 เมื่อ m คือ จำนวนอินพุทของนิวรอน k ในชั้นเอาท์พุท

$$y_k(p) = sigmoid\left[\sum_{j=1}^{m} x_{jk}(p) \times w_{jk}(p) - \theta j\right] \quad (2.9)$$

2.3.7.3 โครงข่ายประสาทเทียมสามชั้นที่ใช้แก้ตรรกศาสตร์ XOR

ต่อไปนี้เป็นตัวอย่างการคำนวณหาเอาท์พุทจากแต่ละโหนดของโครงข่ายประสาทเทียมสามชั้นที่ใช้แก้ตรรกศาสตร์ XOR (Three-layer Neural Network for Solving the Exclusive-OR Operation) ซึ่งมีรูปแบบดังรูป 2.31

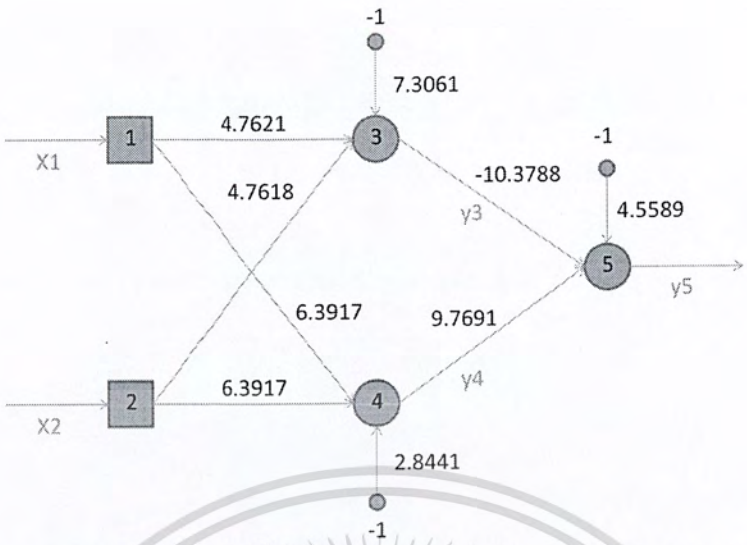


รูป 2.31 โครงข่ายประสาทเทียมสามชั้นที่ใช้แก้ตรรกศาสตร์ XOR

นิวรอน 1 และ 2 ในชั้นอินพุทมีหน้าที่รับ X1 และ X2 เพื่อส่งต่อไปยัง นิวรอน 3 และ 4 ดังนั้น $x_{13} = x_{14} = x_1$ และ $x_{23} = x_{24} = x_2$

ผลกระทบของค่ากระตุ้นต่อนิวรอนในชั้นซ่อนตัวและชั้นเอาท์พุทหาได้จากผลคูณระหว่างค่ากระตุ้น θ กับ -1 ซึ่งเป็นอินพุทที่มีค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.32 โครงข่ายประสาทเทียมสามชั้นที่ใช้แก้ตรรกศาสตร์ XOR หลังการเรียนรู้รอบสุดท้าย

จากรูป 2.32 คือ ค่าน้ำหนักและค่ากระตุ้นจะถูกปรับเปลี่ยนจากการเรียนรู้ ซึ่งในตัวอย่างนี้ได้ผลดังนี้ $w_{13} = 4.7621,\ w_{14} = 6.3917, w_{23} = 4.7618,$ $w_{24} = 6.3917, w_{35} = -10.3788,\ w_{45} = 9.7691,$ $\theta_3 = 7.3061,\ \theta_4 = 2.8441,$ และ $\theta_5 = 4.5589$

จากนั้นทดลองนำโครงข่ายประสาทเทียมนี้เข้าสู่การกระตุ้น โดยในตัวอย่างนี้กำหนดให้ x_1=1 และ x_2=1 จากนั้นจึงคำนวณค่าเอาท์พุทของนิวรอนในชั้นซ่อนตัวซึ่งก็คือนิวรอน 3 และ 4 ดังสมการ 2.10 และ 2.11 ซึ่งจะสามารถหาเอาท์พุทของนิวรอน 5 ได้ดังสมการ 2.12

$$y_3 = \text{sigmoid}(x_1 w_{13} + x_2 w_{23} - \theta_3) = \frac{1}{[1+e^{-(1\times4.7621+1\times4.7618-1\times7.3061)}]} = 0.9018 \quad (2.10)$$

$$y_4 = \text{sigmoid}(x_1 w_{14} + x_2 w_{24} - \theta_4) = \frac{1}{[1+e^{-(1\times6.3917+1\times6.3917-1\times2.8441)}]} = 0.9999 \quad (2.11)$$

$$y_5 = \text{sigmoid}(y_3 w_{35} + y_4 w_{45} - \theta_5) = \frac{1}{[1+e^{-(0.9018\times-10.3788+0.9999\times9.7691-1\times4.5589)}]} = 0.01551 \quad (2.12)$$

จากสมการ 2.12 พบว่าเอาท์พุทสุดท้ายที่ได้ใกล้เคียงกับคำตอบจริงๆ ของการนำ 1 ไป XOR กับ 1 ซึ่งจะได้คำตอบเป็น 0 แสดงให้เห็นว่าโครงข่ายประสาทเทียมสามารถแก้ปัญหานี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.8 ข้อดีข้อเสียของโครงข่ายประสาทเทียม

2.3.8.1 ข้อดี

โครงข่ายประสาทเทียมสามารถแก้ไขปัญหาได้มากมาย มีความแม่นยำสูง รวมทั้งยังสามารถรู้จำแบบ จดจำข้อมูล แยกแยะข้อมูล และคาดคะเนสิ่งที่ไม่เคยพบมาก่อนได้ ทำให้มีการนำโครงข่ายประสาทเทียมไปใช้ประโยชน์กันอย่างแพร่หลาย เช่น นำไปใช้ตัดสินใจในเกม เช่น เกมแบ็คแกมมอน (Backgammon) และเกมหมากรุก (Chess), นำไปใช้จดจำแบบรูป (Pattern Recognition) เช่น จดจำใบหน้า, นำไปใช้ด้านวิเคราะห์ข้อมูลทางธุรกิจ (Data Mining), และนำไปใช้ตรวจจับขยะไปรษณีย์อิเล็กทรอนิกส์ (Spam) เป็นต้น

2.3.8.2 ข้อเสีย

ถึงแม้ว่าโครงข่ายประสาทเทียมจะสามารถแก้ไขปัญหาได้มากมาย แต่ก็ยังคงมีข้อจำกัดอยู่ เนื่องจากขั้นตอนวิธีเรียนรู้แบบส่งค่ากลับที่ใช้กันอย่างแพร่หลายในการทำโครงข่ายประสาทเทียมนั้นไม่สามารถค้นหาคำตอบที่ดีที่สุดจริงๆ ได้เสมอไป เนื่องจาก ขั้นตอนวิธีเรียนรู้แบบส่งค่ากลับไม่ใช่ขั้นตอนวิธีค้นหาแบบสุ่ม ดังนั้นค่าน้ำหนักจึงเปลี่ยนแปลงไม่มาก เป็นเหตุให้ค่าผิดพลาดที่ได้มีปัญหาเรื่องค่าน้อยที่สุดเฉพาะบริเวณ (Local Minima) ซึ่งมักใช้ SA (Simulated Annealing) หรือขั้นตอนวิธีเชิงพันธุกรรมมาช่วยแก้ปัญหานี้

นอกจากนี้ การออกแบบรูปแบบของโครงข่ายประสาทเทียมนั้นทำได้ยาก มักขึ้นอยู่กับการคาดคะเนและประสบการณ์ของผู้ออกแบบ

2.4 NeuroEvolution of Augmenting Topologies (NEAT)

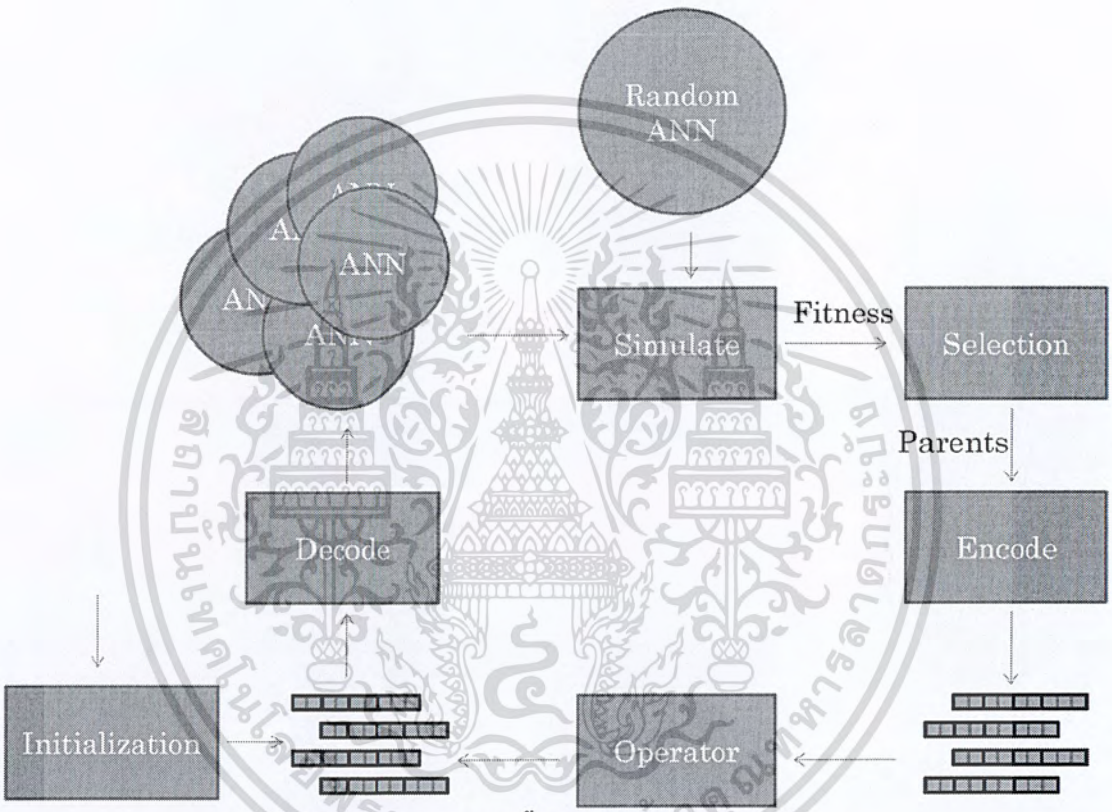
NE (Neuroevolution) เป็นการเรียนรู้ด้วยคอมพิวเตอร์ (Machine Learning) ที่ใช้ขั้นตอนวิธีเชิงพันธุกรรมสำหรับวิวัฒน์โครงข่ายประสาทเทียม มีประโยชน์อย่างมากสำหรับโครงข่ายประสาทเทียมที่ใช้ในเกมและโปรแกรมควบคุมหุ่นยนต์ เพราะเป็นปัญหาที่ง่ายต่อการวัดประสิทธิภาพในตอนท้ายแต่เป็นการยากมากหรือเป็นไปไม่ได้เลยที่จะจัดทำชุดข้อมูลตัวอย่างอินพุทและเอาท์พุทที่ถูกต้องเพื่อใช้สำหรับการเรียนรู้แบบมีผู้สอน ดังนั้น NE จึงถูกจัดเป็นการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

NEAT (NeuroEvolution of Augmenting Topologies) เป็น NE ประเภทหนึ่ง ถูกคิดค้นโดย Ken Stanley ในปี 2002 ที่ The University of Texas at Austin เป็นวิธีที่ใช้ขั้นตอนวิธีเชิงพันธุกรรมวิวัฒน์ทั้งน้ำหนักของเส้นทางระหว่างโหนดและโครงสร้างของโครงข่ายประสาทเทียมซึ่งจากมีทั้งการเพิ่มลดเส้นทางระหว่างโหนดและการเพิ่มลดโหนด ช่วยประหยัดเวลาที่ใช้ในการออกแบบโครงสร้างของโครงข่ายประสาทเทียมโดยมนุษย์เป็นอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NEAT เป็นวิธีที่วิวัฒน์โครงข่ายประสาทเทียมได้อย่างมีประสิทธิภาพและรวดเร็วที่สุดในบรรดา NE แบบอื่นๆ ในปัญหาด้านการควบคุมทั่วๆ ไป (Simple Control Task)

NEAT มีขั้นตอนพื้นฐานคล้ายๆ กับ NE แบบอื่นๆ ดังรูป 2.33 ซึ่งจีโนมจะถูกแปลงเป็นฟีโนมในรูปของโครงข่ายประสาทเทียมเพื่อใช้ในการหาค่าความเหมาะสมโดยอาศัยหลักการของโครงข่ายประสาทเทียม นอกจากขั้นตอนพื้นฐานแล้ว NEAT มีเอกลักษณ์ในเรื่องของการเริ่มต้นด้วยโครงสร้างที่เล็กที่สุด, การเข้ารหัส, การครอสโอเวอร์, มิวเทชัน, และการจัดเผ่าพันธุ์



รูป 2.33 กระบวนการเบื้องต้นของของ NEAT

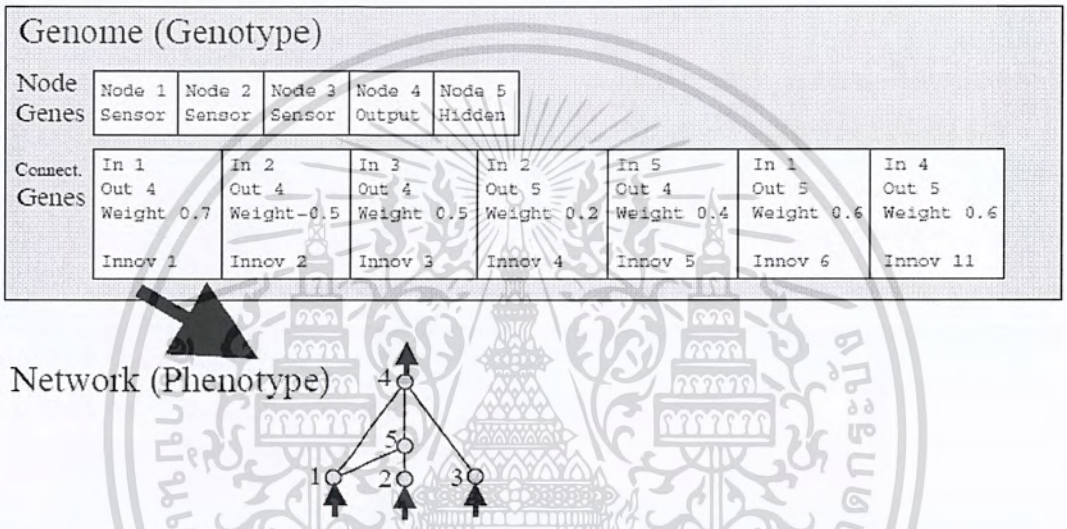
2.4.1 การเริ่มด้วยโครงสร้างที่เล็กที่สุด

โดยปกติแล้ว NE ที่สามารถวิวัฒน์ทั้งน้ำหนักของเส้นทางระหว่างโหนดและโครงสร้างของโครงข่ายประสาทเทียมจะเริ่มจากการสร้างโครงข่ายประสาทเทียมที่ถูกสุ่มโครงสร้างขึ้นมา แต่สำหรับ NEAT นั้นจะเริ่มจากการสร้างโครงข่ายประสาทเทียมที่เล็กที่สุดขึ้นมาก่อน (มีแต่โหนดที่อยู่ในชั้นอินพุทและโหนดที่อยู่ในชั้นเอาท์พุทโดยที่โหนดในชั้นอินพุทแต่ละโหนดเชื่อมต่อไปยังโหนดที่อยู่ในเอาท์พุททุกโหนด) ทำให้เมื่อได้โครงข่ายประสาทเทียมที่มีประสิทธิภาพแล้ว โครงข่ายประสาทเทียมที่เป็นผลลัพธ์จาก NEAT จะมีขนาดของโครงสร้างที่เล็กกว่าโครงข่ายประสาทเทียมที่เป็นผลลัพธ์จาก NE แบบอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การเข้ารหัส (Encoding)

ในการเข้ารหัสของ NEAT นั้นถูกออกแบบมาให้สามารถทำการครอสโอเวอร์ (Crossover) ได้อย่างมีประสิทธิภาพโดยในแต่ละจีโนม (Genome) จะเก็บเซ็ทของข้อมูล 2 ประเภท ได้แก่ ยีนโหนดและยีนการเชื่อมต่อ ยีนโหนดประกอบไปด้วยเลขประจำโหนดและประเภทของโหนด ส่วนยีนการเชื่อมต่อประกอบไปด้วย หมายเลขประจำโหนดของโหนดต้นทาง, หมายเลขประจำโหนดของโหนดปลายทาง, ค่าน้ำหนักของการเชื่อมต่อ, และเลขประจำตัวของยีนการเชื่อมต่อนั้นๆ ดังรูป 2.34



รูป 2.34 การเข้ารหัสใน NEAT

เมื่อใดก็ตามที่มียีนใหม่เพิ่มขึ้นมาจากมิวเทชัน ยีนใหม่นี้จะถูกนำไปเปรียบเทียบกับยีนทั้งหมดที่เคยเกิดขึ้นในระบบ หากพบว่าไม่มียีนใดมีลักษณะเหมือนกับยีนใหม่นี้ เลขประจำตัวยีนล่าสุดก็จะถูกเพิ่มขึ้นและมอบหมายเลขนั้นให้เป็นเลขประจำตัวของยีนที่เกิดขึ้นมาใหม่ ดังนั้นเลขดังกล่าวจึงบอกถึงลำดับของการเกิดขึ้นของยีนต่างๆ อีกด้วย แต่หากยีนใหม่นี้ถูกพบว่ามีลักษณะเหมือนกับยีนที่เคยเกิดขึ้นมาแล้วก่อนหน้า ก็จะใช้เลขประจำตัวยีนของยีนที่ถูกพบว่าเหมือนกัน

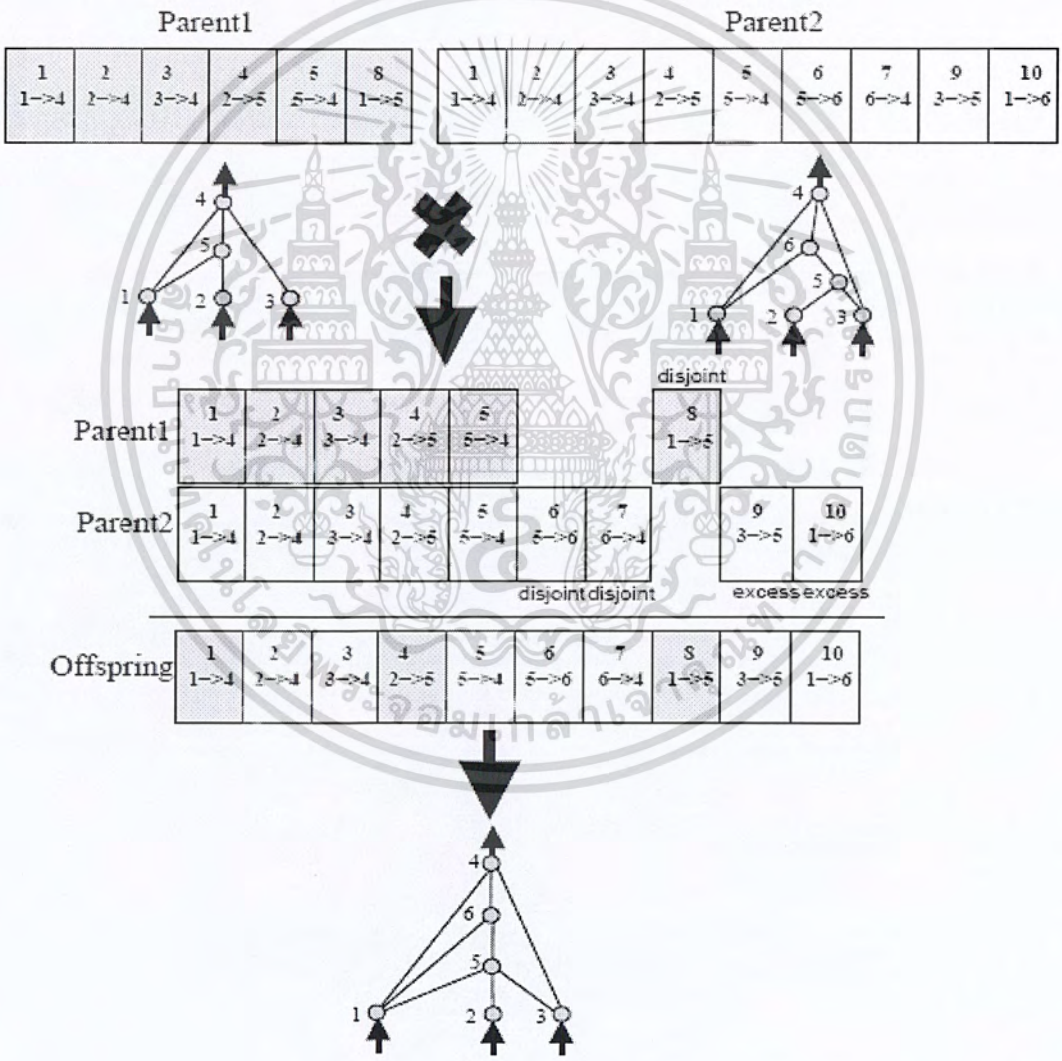
เลขประจำตัวของยีนทั้งสองแบบมีจุดประสงค์เพื่อใช้ในการทำบันทึกประวัติ (Historical Marking) เพื่อบอกถึงความเป็นมาของแต่ละจีโนม เนื่องจากจีโนมที่มีต้นกำเนิดเดียวกัน มักจะมีโหนดยีนที่เกิดขึ้นในช่วงแรกๆ คล้ายกัน การทำบันทึกประวัตินี้มีประโยชน์อย่างมากต่อการครอสโอเวอร์และการจัดการเผ่าพันธุ์

2.4.3 ครอสโอเวอร์ (Crossover)

การทำบันทึกประวัติทำให้ NEAT มีประสิทธิภาพสูง ระบบสามารถรู้ได้ว่ายีนตัวไหนสอดคล้องกับยีนตัวไหน เมื่อเกิดการครอสโอเวอร์ ยีนของแต่ละจีโนมทั้งฝ่ายพ่อและแม่จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

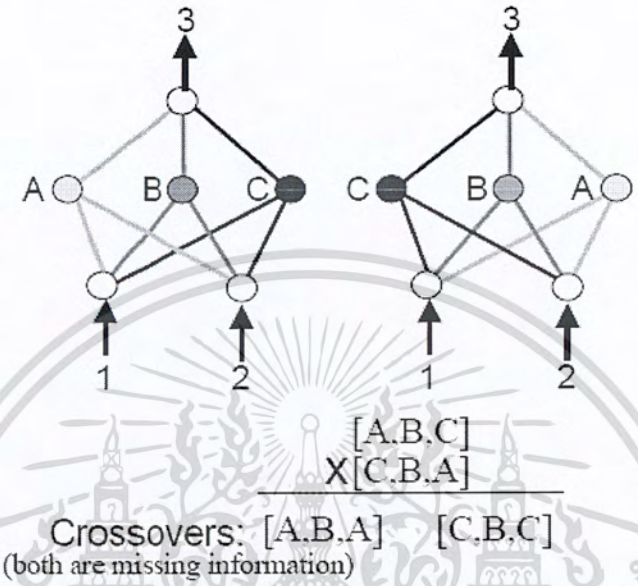
เรียงลำดับด้วยเลขประจำตัวแล้วทำการจับคู่ยีนที่มีเลขประจำตัวเดียวกันของจีโนมทั้งสองฝ่าย ยีนที่ถูกจับคู่เหล่านี้จะถูกเรียกว่ายีนที่สอดคล้องกัน (Matching Gene) ส่วนยีนที่ไม่มีคู่จะเรียกว่า Disjoint หรือ Excess ขึ้นกับว่ายีนดังกล่าวมีเลขประจำตัวมีค่ามากกว่าหรือน้อยกว่าเลขประจำตัวของยีนที่เกิดทีหลังสุดของจีโนมอีกฝ่าย หากน้อยกว่าจะถูกเรียกว่าดิสจอยน์ (Disjoint) แต่หากมากกว่าจะถูกเรียกว่า เอ็กเซสส์ (Excess) ในการครอสโอเวอร์นั้น จีโนมที่เกิดขึ้นใหม่จะเลือกยีนที่สอดคล้องกันแบบสุ่มมาจากพ่อหรือแม่ก็ได้ แต่จะเลือกยีนที่เป็นดิสจอยน์และเอ็กเซสส์มาจากจีโนมที่มีค่าความเหมาะสมดีกว่าเท่านั้น แต่หากจีโนมทั้งสองมีประสิทภาพดีเท่ากันจะสุ่มเลือกหรือไม่เลือกจากทั้งสองจีโนม ดังรูป 2.35



รูป 2.35 ครอสโอเวอร์ของ NEAT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การครอสโอเวอร์ของ NEAT แบบนี้มีส่วนสำคัญในการทำให้ NEAT มีประสิทธิภาพกว่า NE แบบอื่นๆ เพราะเป็นการแก้ปัญหาคอมเพตติงคอนเวนชัน (Competing Conventions Problem) ดังรูป 2.36 ที่เป็นปัญหาหลักของ NE แบบอื่นๆ



รูป 2.36 ปัญหาคอมเพตติงคอนเวนชัน

ปัญหาคอมเพตติงคอนเวนชันคือ ปัญหาของการเข้ารหัสที่หลากหลายของโครงข่ายประสาทเทียมที่มีโครงสร้างเหมือนกัน ซึ่งจะทำให้การครอสโอเวอร์ด้อยประสิทธิภาพ

2.4.4 มิวเทชัน (Mutation)

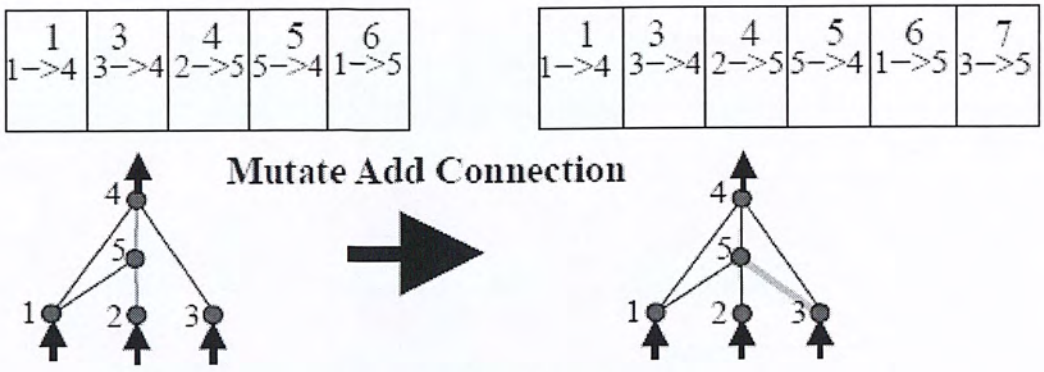
มิวเทชันของ NEAT นั้น จะมีการเปลี่ยนแปลงทั้งค่าน้ำหนักของการเชื่อมต่อ และโครงสร้างของการเชื่อมต่อ สำหรับมิวเทชันเปลี่ยนแปลงค่าน้ำหนักนั้นแบ่งออกเป็น 2 แบบ ดังนี้

1) มิวเทชันน้ำหนักแบบเพอเทอเบชัน (Perturbation) เป็นการเพิ่มลดค่าน้ำหนักจากเดิมในขอบเขตการเพิ่มลดที่กำหนดไว้
2) มิวเทชันน้ำหนักแบบปรับเปลี่ยนกระทันหัน เป็นการสุ่มค่าน้ำหนักใหม่ภายในขอบเขตที่กำหนดโดยไม่สนค่าน้ำหนักเดิม

สำหรับมิวเทชันซึ่งกระทบกับโครงสร้าง (Structural Mutation) สามารถเกิดขึ้นได้ 4 แบบ ดังนี้

1) มิวเทชันเพิ่มการเชื่อมต่อ จะสร้างการเชื่อมต่อระหว่าง 2 โหนดที่ยังไม่ได้เชื่อมต่อกันขึ้นมาโดยสุ่มค่าของน้ำหนักในการเชื่อมต่อ ดังรูป 2.37

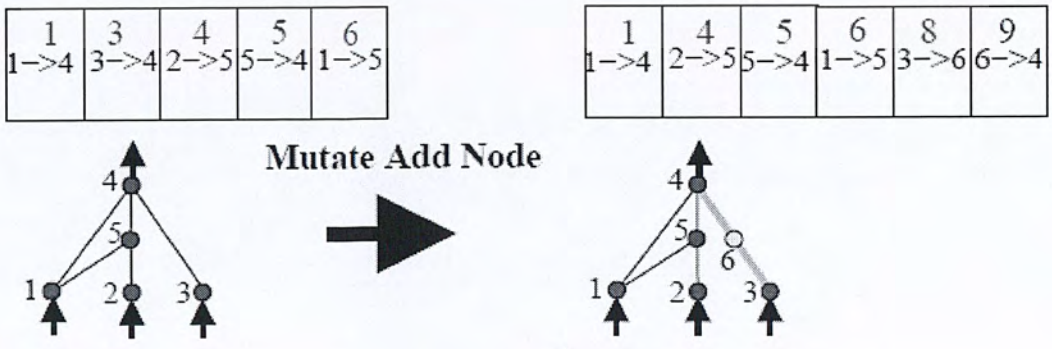
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.37 มิวเทชันเพิ่มการเชื่อมต่อ

2) มิวเทชันลดการเชื่อมต่อ จะสุ่มเลือกการเชื่อมต่อขึ้นมาแล้วลบการเชื่อมต่อนั้นออกจากจีโนม หากมีโหนดใดในชั้นซ่อนตัวที่ไม่มีการเชื่อมเข้าหรือออกจากโหนดก็จะโดนลบออกจากจีโนมเช่นกัน และหากมีโหนดถูกลบออกจากจีโนม การเชื่อมต่อที่เป็นลูปเข้าออกโหนดที่ถูกลบก็จะถูกลบออกจากจีโนมด้วยเช่นกัน

3) มิวเทชันเพิ่มโหนด จะสุ่มเลือกการเชื่อมต่อที่มีอยู่ขึ้นมา โดยที่จะมีโหนดใหม่ขึ้นมาแทนที่การเชื่อมต่อนั้นแล้วการเชื่อมต่อนั้นจะถูกลบออกจากจีโนม แล้วสร้างการเชื่อมต่อใหม่ 2 การเชื่อมต่อเข้ามาแทนที่ ดังรูป 2.38 การเชื่อมต่อใหม่ที่เข้ามายังโหนดที่ถูกสร้างใหม่นี้จะมีค่าน้ำหนักเท่ากับ 1 เสมอ และการเชื่อมต่อที่ออกจากโหนดดังกล่าวจะมีค่าน้ำหนักเท่ากับโหนดเก่าที่ถูกปิดไป เหตุที่ต้องกำหนดค่าน้ำหนักของการเชื่อมต่อในลักษณะนี้เพราะต้องการลดผลกระทบของมิวเทชันโดยพยายามรักษาลักษณะเดิมของโครงข่ายประสาทเทียมไว้ เพราะโดยส่วนมากหากสุ่มค่าน้ำหนักขึ้น มักจะทำให้โครงข่ายประสาทเทียมที่เกิดมิวเทชันขึ้นมีค่าความเหมาะสมแย่ลงกว่าเดิมและยังไม่ทันที่จะได้รับการพัฒนาให้ดีขึ้นก็จะโดนคัดออกจากประชากรไปเสียก่อน



รูป 2.38 มิวเทชันเพิ่มโหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) มิวเทชันลดโหนด จะสุ่มลบโหนดในชั้นซ่อนตัวออกจากจีโนม โดยจะมีผลให้การเชื่อมต่อที่ผ่านโหนดนั้นทั้งหมดต้องถูกลบออกจากจีโนมด้วยเช่นกัน

2.4.5 การจัดเผ่าพันธุ์ (Speciation)

เนื่องจากโครงสร้างของโครงข่ายประสาทเทียมบางแบบที่ถูกมิวเทชันขึ้นมาใหม่มักจะมีค่าความเหมาะสมน้อยกว่าโครงสร้างแบบเดิมๆ ในช่วงแรก แต่หากถูกพัฒนามากพออาจจะมีค่าความเหมาะสมมากกว่าโครงสร้างแบบเดิมๆ ได้ NEAT จึงพยายามปกป้องโครงสร้างแบบใหม่ที่ถูกมิวเทชันขึ้นไม่ให้ถูกคัดออกในขั้นตอนการคัดเลือกโดยการใช้การจัดการเผ่าพันธุ์

การจัดเผ่าพันธุ์อาศัยการทำบันทึกประวัติเพื่อจัดกลุ่มจีโนมที่มียีนคล้ายคลึงกันให้อยู่ในกลุ่มเดียวกัน โดยการจะวัดความคล้ายคลึงนี้ใช้ค่าความห่างของต้นตระกูล (δ) ดังสมการ 2.13

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \overline{W} \tag{2.13}$$

ค่า E คือจำนวนเอ็กเซสส์, ค่า D คือจำนวนดิสจอยน์, ค่า $\overline{W}$ คือค่าเฉลี่ยของส่วนต่างของน้ำหนักระหว่างยีนที่เข้าคู่กัน, ค่าคงที่ c_1, c_2, c_3 นั้นมีไว้สำหรับปรับความสำคัญของทั้ง 3 ค่า และ N คือความต่างของจำนวนยีนของจีโนมทั้ง 2 ตัว normalize โดยขนาดของจีโนม (มีค่าเป็น 1 ในกรณีที่โครงสร้างของจีโนมทั้งสองมีขนาดเล็ก เช่นมีแค่ 20 ยีน)

เมื่อมีจีโนมใหม่ๆ เกิดขึ้น จีโนมใหม่นี้จะถูกนำไปเปรียบเทียบกับจีโนมซึ่งเป็นตัวแทนของแต่ละเผ่าพันธุ์ที่มีอยู่แล้วเพื่อหาค่า δ ไปเปรียบเทียบกับขอบเขตที่กำหนดไว้ (δ_t) หากค่า δ ไม่เลยขอบเขต จีโนมใหม่นี้จะถูกบรรจุเป็นเผ่าพันธุ์เดียวกับตัวแทนที่นำมาเปรียบเทียบ แต่หากค่า δ เลยขอบเขต จะมีการสร้างเผ่าพันธุ์ใหม่ขึ้นโดยมีจีโนมใหม่นี้เป็นตัวแทนของเผ่าพันธุ์

การมีอยู่ของเผ่าพันธุ์สามารถปกป้องโครงสร้างใหม่ๆ ของโครงข่ายประสาทเทียมได้โดยใช้กระบวนการแบ่งปันค่าความเหมาะสมระหว่างสมาชิกในเผ่าพันธุ์เดียวกัน ดังนั้นค่าความเหมาะสมของจีโนมที่ i ที่ถูกใช้ในการคัดเลือกนั้น จะใช้ค่าความเหมาะสมปรับแต่ง (f_i') ซึ่งถูกลดหย่อนลงจากค่าความเหมาะสมที่ได้มาจริงๆ (f_i) ตามสมการ 2.14 โดยที่ n คือจำนวนประชากรทั้งหมด

$$f_i' = \frac{f_i}{\sum_{j=1}^{n} sh(\delta(\mathrm{i,j}))} \tag{2.14}$$

ฟังก์ชัน sh จะมีค่าเป็น 0 เมื่อค่า $\delta(\mathrm{i,j})$ มีค่ามากกว่า δ_t มิฉะนั้น $\delta(\mathrm{i,j})$ จะมีค่าเป็น 1 ดังนั้น $\sum_{j=1}^{n} sh(\delta(\mathrm{i,j}))$ คือจำนวนของจีโนมที่อยู่ในเผ่าพันธุ์เดียวกันกับจีโนม i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนจีโนมในแต่ละเผ่าพันธุ์จะถูกควบคุมโดยการกำหนดให้แต่ละเผ่าพันธุ์สามารถสร้างจีโนมในรุ่นใหม่ได้แตกต่างกัน โดยจำนวนจีโนม (N_j') ที่เผ่าพันธุ์ j จะสามารถสร้างได้นั้นเป็นไปตามสมการ 2.15 โดยที่ N_j คือจำนวนจีโนมปัจจุบันในเผ่าพันธุ์ j, และ $\overline{f'}$ คือค่าเฉลี่ยของค่าความเหมาะสมปรับแต่งของประชากรทั้งหมดจากทุกเผ่าพันธุ์

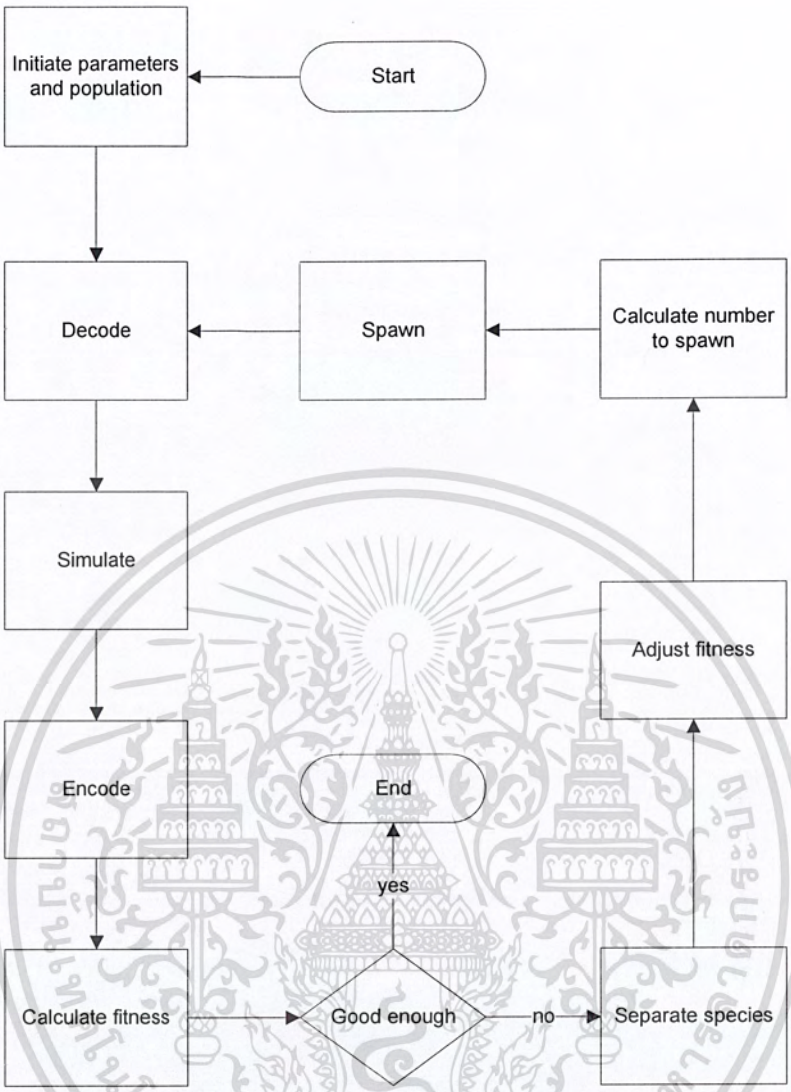
$$N_j' = \frac{\sum_{i=1}^{N_j} f_{ij}'}{\overline{f'}} \tag{2.15}$$

เผ่าพันธุ์แต่ละเผ่าพันธุ์จะมีอายุของตัวเอง ซึ่งอายุจะเพิ่มขึ้น 1 ทุกครั้งที่ผ่านการวิวัฒนาการไป 1 รุ่น ส่งผลให้เมื่ออายุของเผ่าพันธุ์เกินกว่าขอบเขตที่กำหนดไว้ เผ่าพันธุ์นั้นจะถูกกำจัดออกไปหากไม่ใช่เผ่าพันธุ์ที่มีจีโนมที่มีค่าความเหมาะสมสูงสุดอยู่ในเผ่าพันธุ์นั้น

2.4.6 กระบวนการของ NEAT

ในหัวข้อนี้จะกล่าวถึงกระบวนการทั้งหมดของ NEAT ดังรูป 2.39 ซึ่งเป็นการรวมเอาเอกลักษณ์ของ NEAT ทั้งหมดใส่เพิ่มเติมลงไปในกระบวนการของ NE แบบทั่วๆ ไป โดยมีรายละเอียดของแต่ละขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.39 กระบวนการทั้งหมดของ NEAT

1) Initiate parameters and population: ตั้งค่าเริ่มต้นของตัวแปรที่ใช้ใน NEAT เช่น จำนวนประชากร, อัตราการครอสโอเวอร์, อัตรามิวเทชันเพิ่มโหนด เป็นต้น และมีการสร้างจีโนมที่มีโครงสร้างแบบเล็กที่สุดดังหัวข้อ 2.4.1 ออกมาเป็นจำนวนเท่ากับจำนวนประชากร โดยค่าน้ำหนักในเส้นเชื่อมต่อแต่ละเส้นจะถูกสุ่มค่าขึ้นมาทั้งหมด
2) Decode: แปลงจากจีโนมแต่ละตัวไปเป็นโครงข่ายประสาทเทียม โดยอาศัยรูปแบบดังรูป 2.39 ในการแปลง
3) Simulate: นำโครงข่ายประสาทเทียมแต่ละตัวเข้าไปแก้ปัญหาที่เป็นโจทย์ โดยใช้ขั้นตอนในการเอาท์พุทจากหัวข้อ 2.3.3.2
4) Encode: แปลงจากโครงข่ายประสาทเทียมแต่ละตัวกลับไปเป็นจีโนม โดยอาศัยรูปแบบดังรูป 2.39 ในการแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) Calculate fitness: วัดค่าความเหมาะสมจากประสิทธิภาพของการแก้ปัญหาของโครงข่ายประสาทเทียม
6) Good enough: หากพบว่ามีโครงข่ายประสาทเทียมใดมีค่าความเหมาะสมเป็นที่น่าพึงพอใจแล้ว จะถือว่าจบการเรียนรู้โดยมีผลลัพธ์เป็นโครงข่ายประสาทเทียมที่มีค่าความเหมาะสมที่ดีที่สุดในรุ่นปัจจุบัน
7) Sperate species: แบ่งจีโนมที่มีลักษณะคล้ายกันให้อยู่ในเผ่าพันธุ์เดียวกันตามวิธีในหัวข้อ 2.4.5 โดยใช้สมการ 2.13 ในการหาความคล้ายกันของจีโนม
8) Adjust fitness: คำนวณค่าความเหมาะสมปรับแต่งดังสมการ 2.14 ให้กับแต่ละจีโนม
9) Calculate number to spawn: คำนวณจำนวนจีโนมรุ่นถัดไปที่แต่ละเผ่าพันธุ์ต้องสร้าง
10) Spawn: แต่ละเผ่าพันธุ์ต้องสร้างจีโนมรุ่นใหม่ขึ้นมา โดยใช้การครอสโอเวอร์และมิวเทชันดังหัวข้อ 2.4.3 และ 2.4.4
11) Loop: นำจีโนมรุ่นใหม่ที่ได้ไปเข้าสู่ขั้นตอนที่ 2 ใหม่

2.4.7 ข้อดีข้อเสียของ NEAT

NEAT เป็น NE ซึ่งได้รับการพิสูจน์แล้วว่ามีประสิทธิภาพในการแก้ปัญหาที่ต้องใช้การเรียนรู้แบบเสริมกำลังได้เป็นอย่างดี ไม่ว่าจะเป็นการแก้ปัญหา DPV (Double Pole Balancing with Velocities), การแก้ปัญหา DPNV (Double Pole Balancing Without Velocities), การบังคับรถแข่งในเกมรถแข่ง, การเลือกแผนการในเกมประเภท TBS (Turn-Based Strategy) และเกมประเภท RTS (Real-Time Strategy), หรือแม้แต่การสร้างสรรค์กระสุนปืนรูปแบบใหม่ๆ ของเกมแนวเครื่องบินรบ นอกจากนี้ rtNEAT (real-time NeuroEvolution of Augmenting Topologies) ซึ่งเป็นแนวคิดที่พัฒนาขึ้นมาจาก NEAT ยังสามารถสร้างการเรียนรู้แบบออนไลน์ได้ดีอีกด้วย

แต่สำหรับเกมที่มีปัญหาที่มีความซับซ้อนสูงนั้น NEAT ยังทำได้ไม่ดีนัก แต่ก็กำลังมีการพัฒนา Cascade-NEAT ขึ้นมาเพื่อรองรับปัญหานี้อยู่เช่นกัน

2.4 การใช้งาน Blender

การทำโมเดล 3 มิติจำเป็นต้องมีการใช้เครื่องมือในการสร้าง จึงมีความจำเป็นต้องเลือกซอฟท์แวร์ (Software) ตัวใดตัวหนึ่งขึ้นมาใช้งาน เนื่องจากการใช้ซอฟท์แวร์ที่มีการละเมิดลิขสิทธิ์ไม่ใช่สิ่งที่ดี และการสร้างโมเดล 3 มิตินั้น ไม่ว่าเครื่องมือไหนก็สามารถทำงานได้คล้ายคลึงกัน ซอฟท์แวร์นี้อาจจะเทียบกับซอฟท์แวร์ที่มีราคาสูงไม่ได้ แต่ก็สามารถสร้างโมเดลที่เทียบเคียงได้เช่นกัน

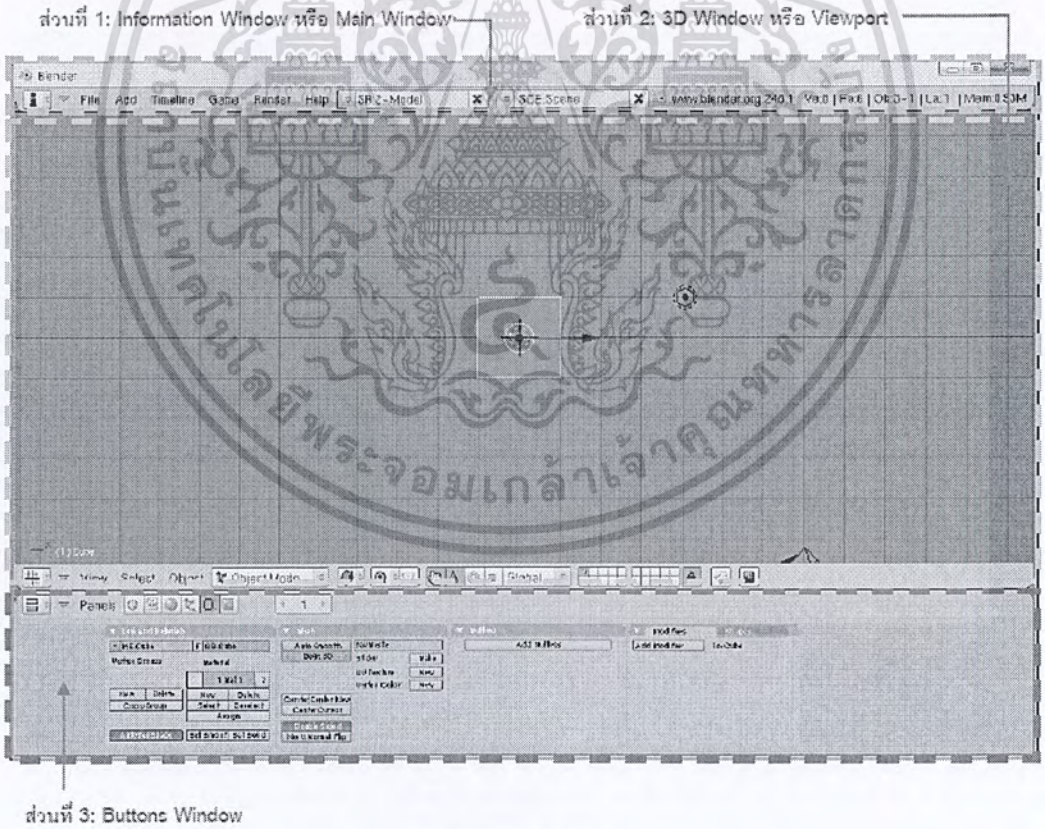
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Blender มีการใช้สคริปต์ (Script) ภาษาไพธอน (Python) อยู่บ้าง ดังนั้นก่อนเรียกใช้โปรแกรม Blender นั้นก็ควรจะลงไพธอน 2.6 ไว้ก่อนเช่นกัน (Blender ต้องการไพธอน 2.6) หลังจากนั้นทำการติดตั้งโปรแกรม Blender และเปิดใช้งาน

2.4.1 การสร้างโมเดลสามมิติ

หน้าต่างการทำงานทั่วไปจะถูกแบ่งออกเป็น 3 ส่วนดังรูป 2.40 ดังนี้

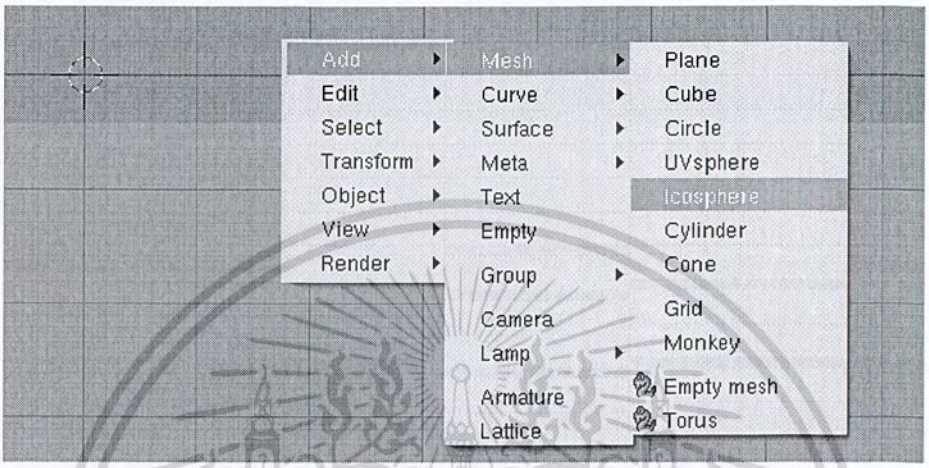
1) Information Window หรือ Main Window จะประกอบด้วยเมนูหลักที่เก็บคำสั่งต่างๆ สำหรับใช้ในการทำงาน
2) 3D Window หรือ View Port เป็นส่วนที่แสดงผลวัตถุที่เราทำงานอยู่ โดยอ้างอิงตามแกน x y และ z ผู้ใช้สามารถจัดการกับมุมมองต่างๆ ได้ เช่น เลื่อนมุมมอง ขยายมุม กล้องเข้าออก เพื่อช่วยปรับแต่งมุมมองให้ทำงานได้ง่ายยิ่งขึ้น
3) Buttons Window เป็นส่วนที่เก็บรวบรวมเครื่องมือในการสร้างงานไว้ เช่น การปรับแต่งแก้ไขรูปทรงของวัตถุ การกำหนดแสงไฟ เป็นต้น



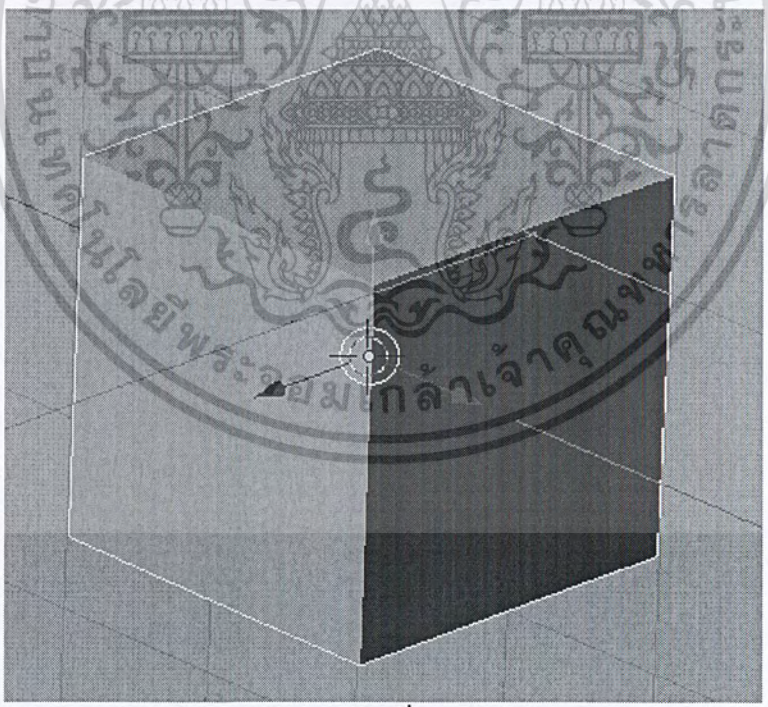
รูป 2.40 หน้าต่างการทำงานทั่วไปของโปรแกรม Blender

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้สามารถสร้างโมเดลต้นแบบเช่น กล่องลูกบาศก์ ระนาบสี่เหลี่ยม ทรงกระบอก ทรงกลม หรืออื่นๆ ได้จากการกด Space Bar ในขณะที่เมาส์อยู่ในส่วนของ View Port โดยจะมีเมนูดังรูป 2.40 ขึ้นมาให้เลือกว่าจะสร้างโมเดลแบบใดขึ้นมา โดยที่โมเดลที่ถูกสร้างนั้น จะขึ้นมาอยู่บนบริเวณของ 3D Cursor ดังรูป 2.42



รูป 2.41 เมนูสร้างโมเดลต้นแบบ

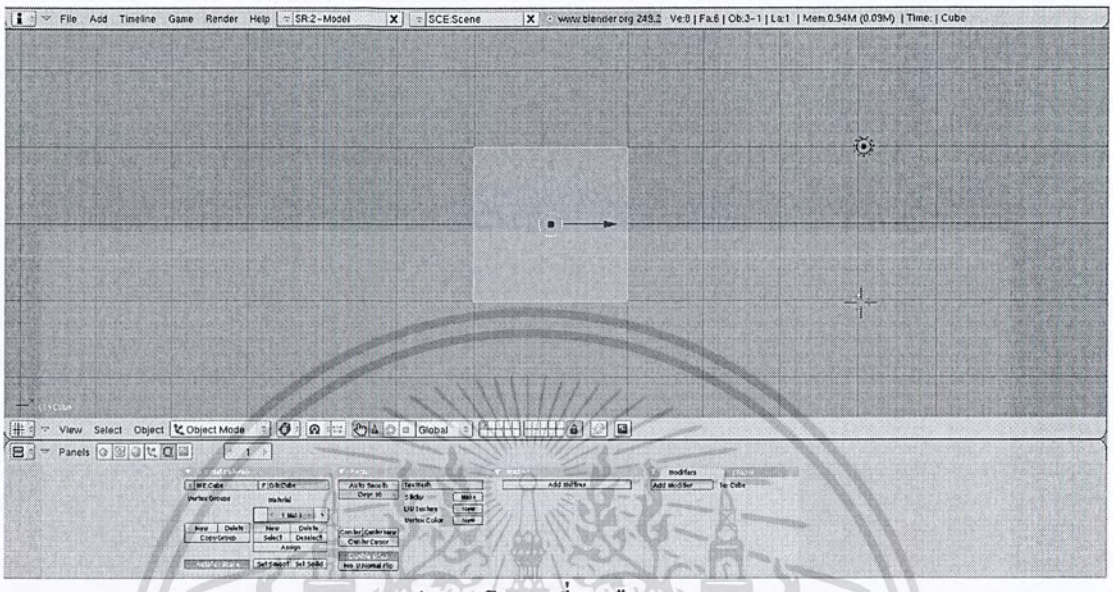


รูป 2.42 วัตถุที่ถูกสร้าง

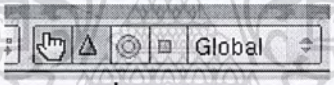
ในหน้าจอ View Port หากมีวัตถุหลายชนิด ผู้ใช้สามารถใช้คลิกขวาในการเลือกวัตถุใดๆ ที่ต้องการดังรูป 2.43 ซึ่งวัตถุที่ถูกเลือกจะมีขอบสีชมพูขึ้นและหากผู้ใช้ต้องการย้ายที่ หมุน หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขยายขนาด ก็สามารถทำได้โดยการใช้คำสั่งที่อยู่ด้านล่าง View Port ดังรูป 2.44 เรียงตามลำดับคือ เปิดการแก้ไขวัตถุ, ย้ายที่วัตถุ, หมุนวัตถุและขยายวัตถุ และใช้คลิกกลางแล้วเลื่อนเพื่อหมุนมุมกล้อง

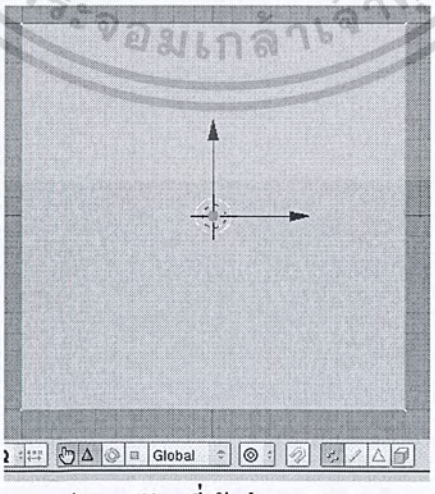


รูป 2.43 โมเดลที่ถูกเลือก



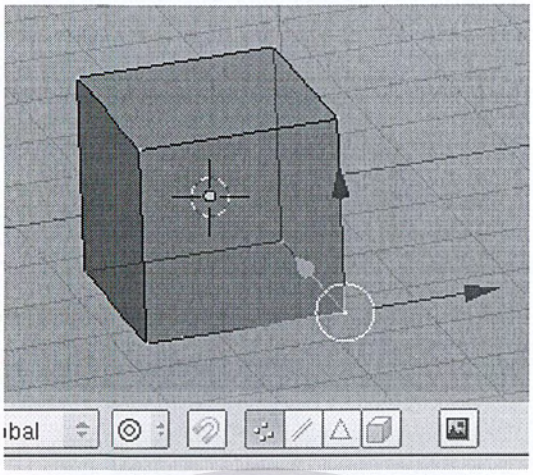
รูป 2.44 คำสั่งด้านล่าง View Port

กด Tab เพื่อนำวัตถุดังกล่าวจาก Object Mode เข้าสู่ Edit Mode ดังรูป 2.45 เพื่อที่จะทำการแก้ไขวัตถุในรูปแบบของจุดดังรูป 2.46, เส้นดังรูป 2.47, หรือหน้าตัดโดยตรงดังรูป 2.48

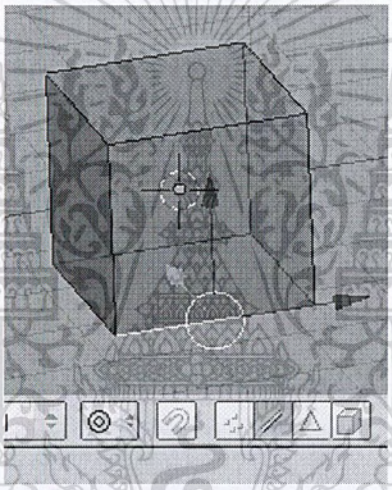


รูป 2.45 วัตถุที่เข้าสู่ Edit Mode

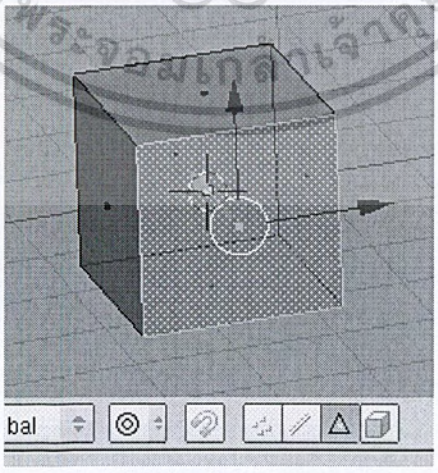
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.46 เลือกจุด



รูป 2.47 เลือกเส้น

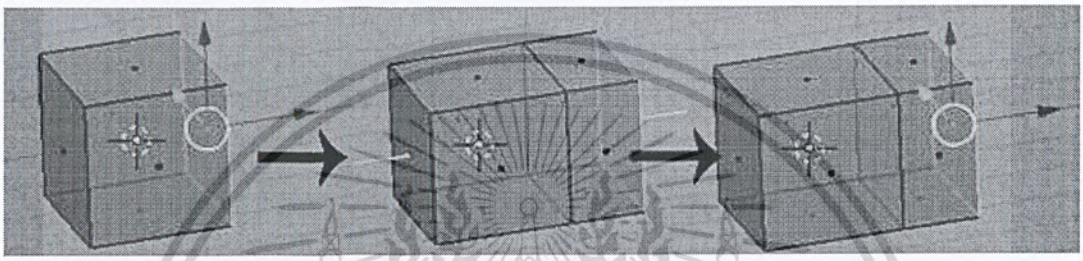


รูป 2.48 เลือกหน้าตัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

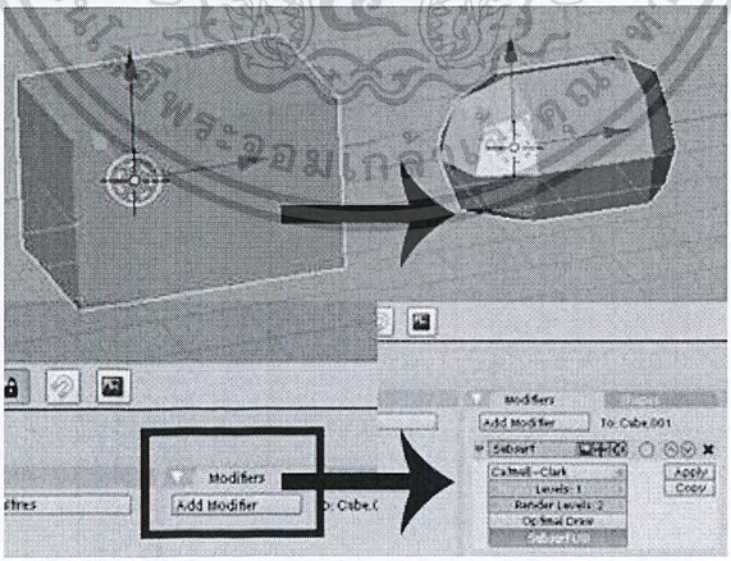
ทั้งใน Edit Mode และ Object Mode ผู้ใช้สามารถกด z เพื่อเปลี่ยนวัตถุที่ถูกเลือกเป็นแบบใสได้ ซึ่งจะมีประโยชน์ในการเลือกหน้าตัดหรือเลือกจุดที่ซ้อนกันได้

การสร้างโมเดลที่มีรายละเอียดสูงๆ นั้นมักจะต้องทำการ Extrude ดังรูป 2.49 เพื่อทำการดึงจุด, เส้น, หน้าตัดออกมาเพื่อสร้างรายละเอียด โดยผู้ใช้เลือกจุด, เส้น, หน้าตัดที่ต้องการดึงออกมา และกด e ผู้ใช้ก็จะสามารถดึงหน้าจุด, เส้น, หน้าตัดนั้นๆ ออกมาได้ ซึ่งในขณะ Extrude จะมีเส้นตรงกลางที่แสดงว่ากำลัง Extrude ในแนวใดปรากฏขึ้นมา สามารถเปลี่ยนแนวได้โดยการใช้ปุ่มเมาส์กลาง



รูป 2.49 การ Extrude

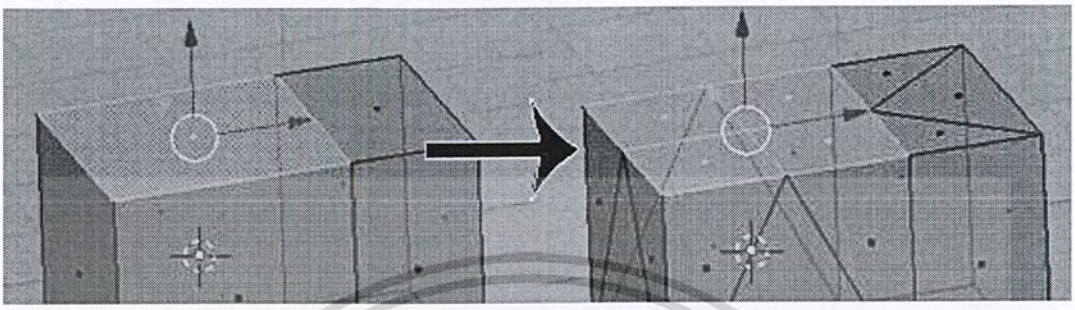
ถ้าผู้ใช้ต้องการเพิ่มความละเอียดให้โมเดล ผู้ใช้สามารถเพิ่มได้ที่ Buttons Window ที่ Modifier เลือก Add Modifier แล้วเลือก Subsurf เพื่อเพิ่มความละเอียดดังรูป 2.50 แต่การใช้ Subsurf นั้นจะทำให้โมเดลเสียรูปร่างที่เป็นอยู่ โดยก่อนการแก้ไข ต้องทำการเลือกวัตถุก่อน โดยการทำ Subsurf นั้นสามารถทำได้ทั้งใน Object Mode และ Edit Mode



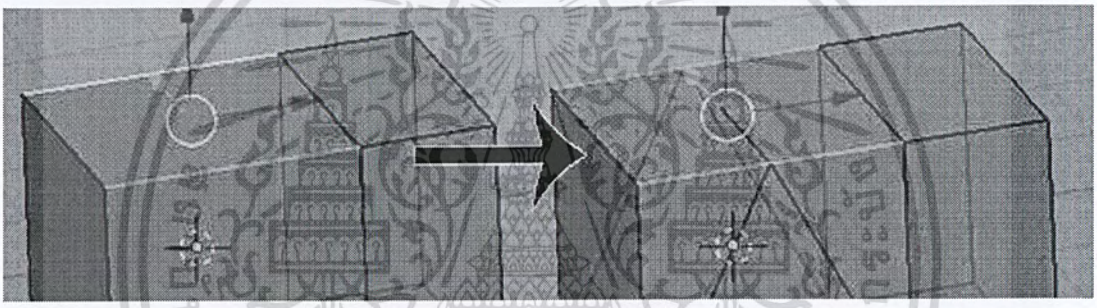
รูป 2.50 การ Subsurf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการตัดหน้าตัดเป็นส่วนๆ จะใช้ปุ่ม Subdivide ในขณะที่อยู่ใน Edit Mode โดยเลือกเส้น เป็นคู่กดโดย Shift ค้างแล้วคลิกขวาอีกเส้นหนึ่ง หรือเลือกหน้าตัดก็ได้ แล้วกดที่ปุ่ม Sub Divide หน้าตัดหรือเส้นคู่นั้นๆ ก็จะถูกแบ่งดังรูป 2.51 และ 2.52



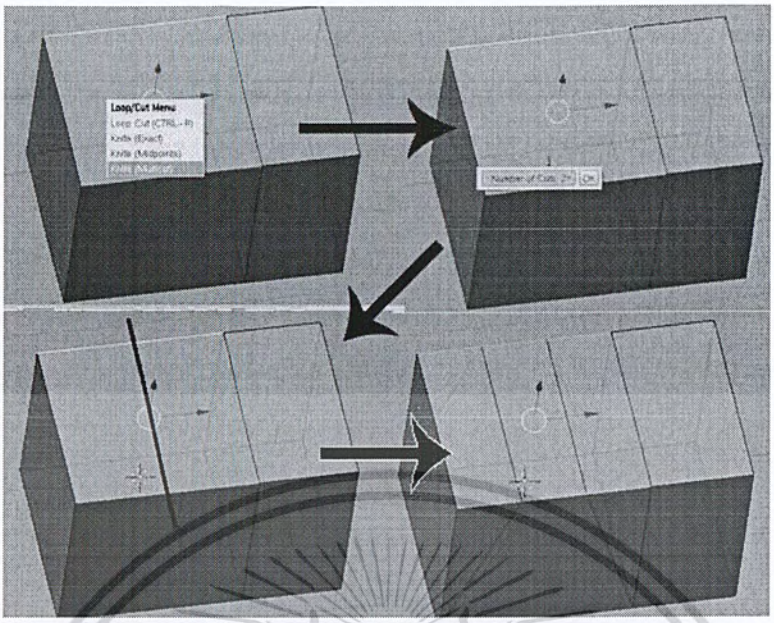
รูป 2.51 การ Subdivide กับหน้าตัด



รูป 2.52 การ Subdivide กับเส้น

หากต้องการตัดมากกว่า 2 ส่วนจะใช้เครื่องมือที่เรียกว่า Knife แทนโดยการเลือกที่เส้นใดๆ หรือหน้าตัดใดๆ แล้วกด k เลือก Knife (Multicut) แล้วเลือกจำนวนที่เส้นที่จะตัด (ปริมาณหน้าตัดที่ได้จะเท่ากับจำนวนเส้นบวกด้วย 1) แล้วคลิกซ้ายค้างลากเมาส์ไปในทิศทางที่จะตัดปล่อยเมาส์และกดปุ่ม Enter ดังรูป 2.53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.53 การใช้ Knife

2.5 การใช้งาน Unity3D

การจะสร้างเกมขึ้นมาหนึ่งเกมนั้น หากต้องสร้างตั้งแต่ศูนย์หรือเริ่มทำขึ้นมาเองจะต้องใช้เวลาปริมาณมหาศาลกับการออกแบบที่ดีมาก ถึงจะสามารถสร้างเกมเอนจิ้น (Game Engine) ที่มีประสิทธิภาพทั้งการบริหารทรัพยากร และการใช้งาน ดังนั้นการเริ่มต้นสร้างตั้งแต่แรกนั้นย่อมมีปัญหากับการสร้างเกมเพื่อใช้ในการศึกษาทางด้านของปัญญาประดิษฐ์ จึงได้หาเกมเอนจิ้นเพื่อนำมาใช้ให้เกิดประโยชน์สูงสุด

เกมเอนจิ้นที่น่าสนใจมี 2 ตัวคือ Unreal Engine และ Unity3D ซึ่งแต่ละตัวก็มีจุดเด่นต่างกันซึ่งจะนำเสนอข้อแตกต่างตามตาราง 2.5

ตาราง 2.5 เปรียบเทียบความสามารถของเกมเอนจิ้น

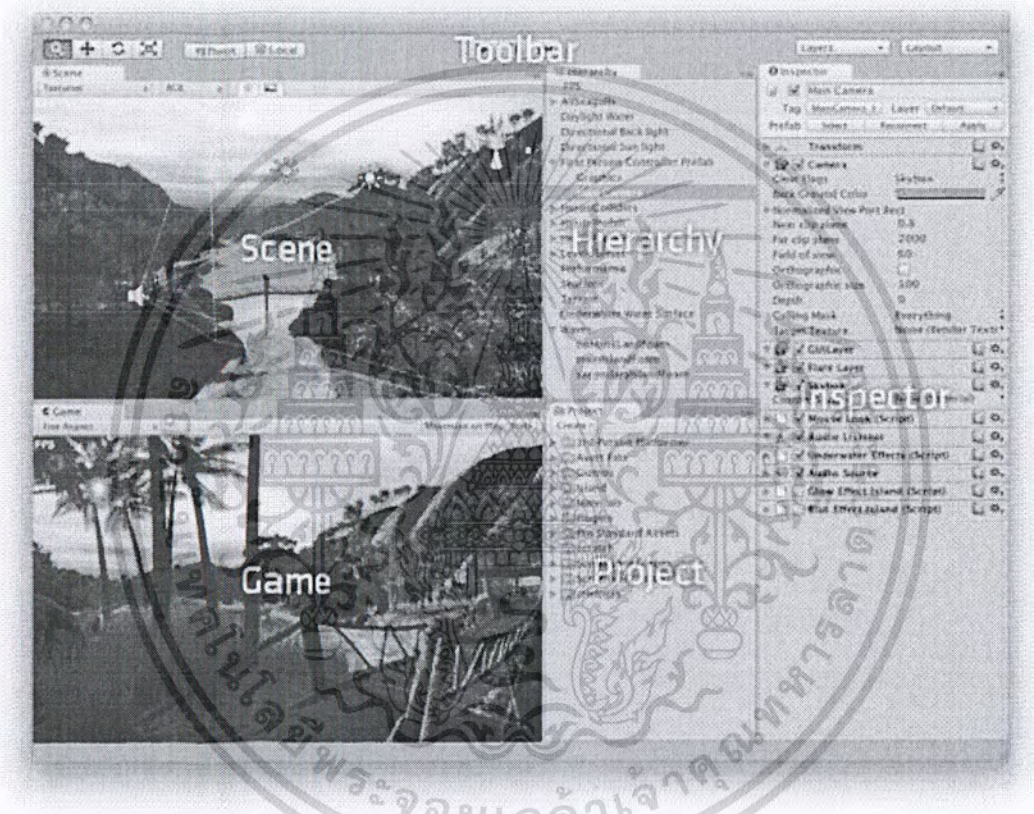
หัวข้อ	Unreal Engine	Unity3D
กราฟฟิค	ดีมาก	พอใช้ได้
ความเร็วในการประมวลผล	สูงมาก	สูง
ข้อมูลอ้างอิง	น้อย	มาก
User Interface ที่ใช้งานง่าย	ยากปานกลาง	ง่าย
ภาษาที่ใช้ในการพัฒนา	C++	JavaScript, Boo script, C#
Os ที่สามารถรันเกมได้	Window	Window, Mac OS
ราคา	ฟรีกรณีที่ใช้เพื่อการศึกษา	ฟรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลสรุปคือเลือกใช้ Unity3D เนื่องจากข้อมูลอ้างอิงที่มีปริมาณมาก และใช้ง่าย เนื่องจากการใช้งานเพียงการสร้างเกมขึ้นมาทดสอบการใช้ปัญญาประดิษฐ์ ไม่ได้ต้องการกราฟฟิคที่สวยงามมากแต่อย่างใด โดยใช้ภาษา C# ในการพัฒนา

2.5.1 วิธีการใช้เบื้องต้น

เมื่อผู้ใช้ได้ลงโปรแกรมเสร็จเรียบร้อย และลงทะเบียนกับทางบริษัทผู้ผลิตแล้ว จะพบกับหน้าจอหลักของ Unity3D ที่จะใช้ในการสร้างเกม

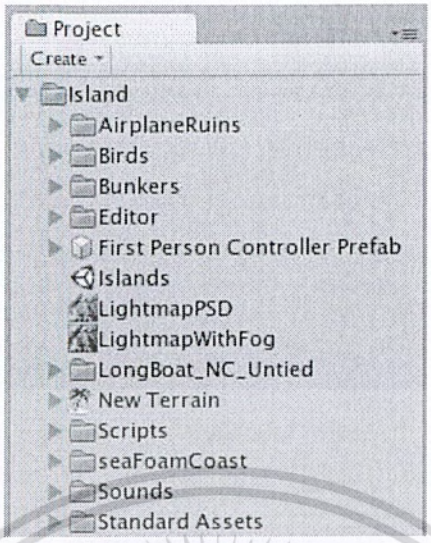


รูป 2.54 หน้าตาของตัวโปรแกรม Unity3D

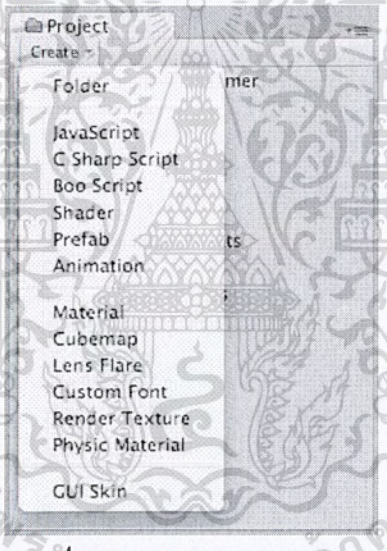
ในส่วนต่างๆ ของโปรแกรมก็จะมีส่วนต่างๆ ดังนี้

1) Project View ในทุกๆ โปรเจค (Project) จะมีโฟลเดอร์ (Folder) ชื่อว่าแอสเซ็ท (Assets) อยู่เสมอ ในโฟลเดอร์นี้จะเก็บแอสเซ็ทที่จะใช้ในการสร้างเกมเช่น ฉาก (Scenes), Scripts, โมเดล 3 มิติ (3D Models), เท็กซ์เจอร์ (Textures), ไฟล์เสียง (Audio Files) และ Prefabs ซึ่งผู้ใช่สามารถสร้างแอสเซ็ทใหม่ๆ ได้โดยการใช้ปุ่ม Create ทางด้านบนของ Project View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



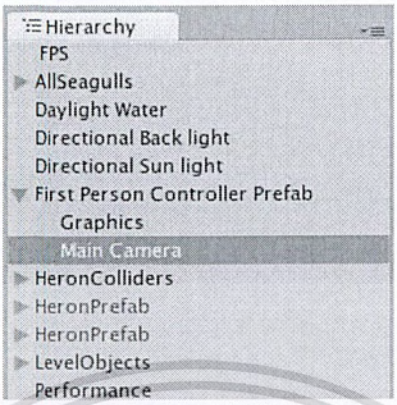
รูป 2.55 หน้าต่าง Project



รูป 2.56 การ Create Assets

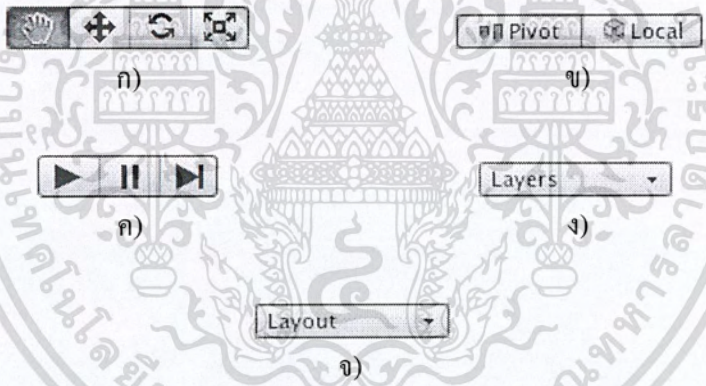
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Hierarchy จะเก็บ GameObject ทั้งหมดในเกมไว้



รูป 2.57 หน้าต่าง Hierarchy

3) Tool Bar จะประกอบด้วย 5 ส่วนดังนี้ Transform Tools, Transform Gizmo Toggles, Play/Pause/Step Buttons , Layers Drop-down, Layout Drop-down



ก) ข) ค) ง) จ)

รูป 2.58 เครื่องมือต่างๆ ใน Tool Bar

ก) เครื่องมือ Transform Tools

ข) เครื่องมือ Transform Gizmo Toggles

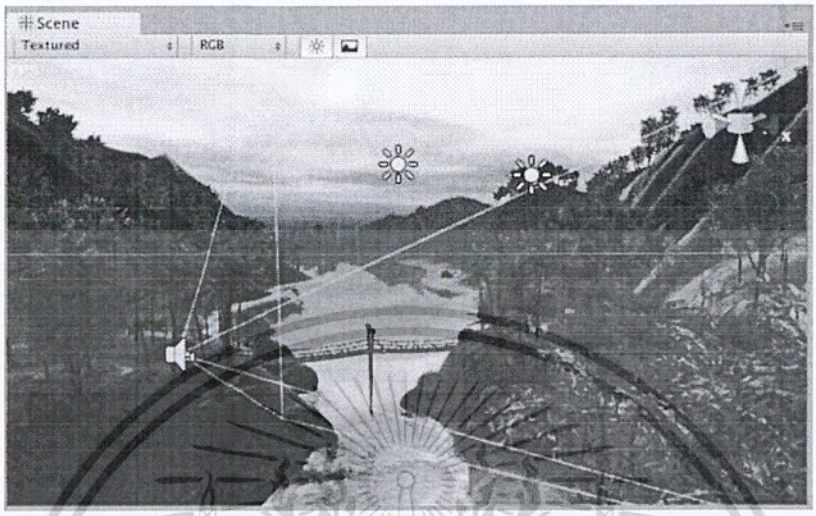
ค) เครื่องมือ Play/Pause/Step Buttons

ง) เครื่องมือ Layers Drop-down

จ) เครื่องมือ Layout Drop-down

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Scene View เป็นหน้าจอสำหรับการจัดวางองค์ประกอบฉากทั้งสภาพแวดล้อม, ตัวละคร, กล้อง และ Game Objects ทั้งหมด



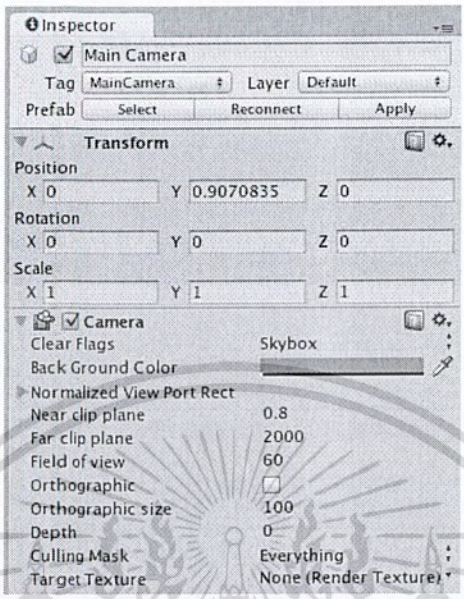
รูป 2.59 หน้าต่าง Scene View

5) Game View คือหน้าจอที่แสดงภาพจากมุมมองของกล้อง

รูป 2.60 หน้าต่าง Game View

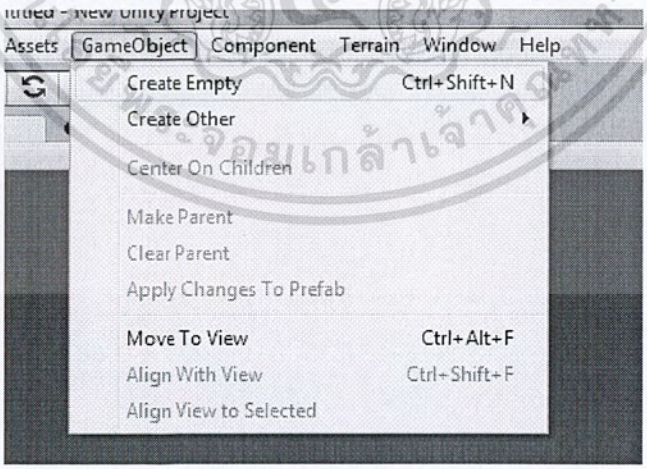
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) Inspector มีไว้สำหรับการกำหนดค่าต่างๆ ให้กับ GameObject



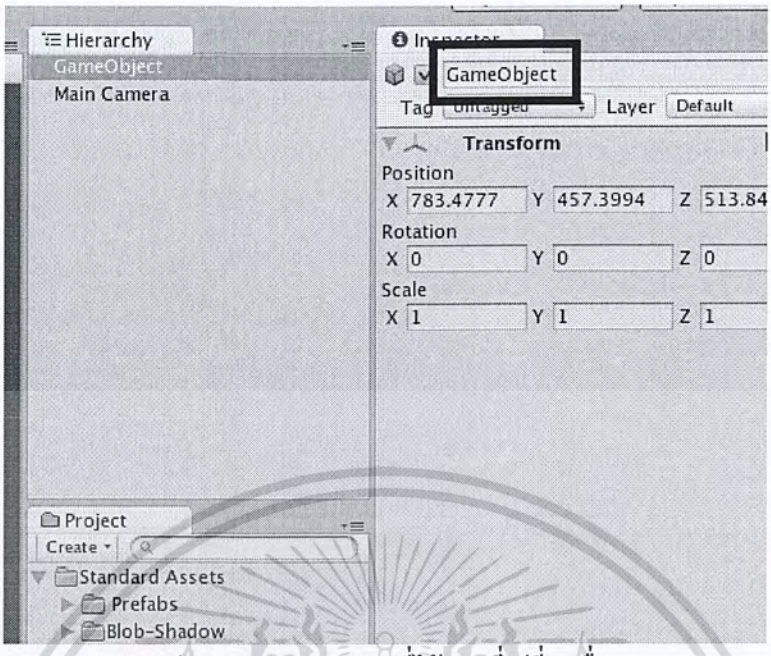
รูป 2.61 หน้าต่าง Inspector

หลักการทำงานเบื้องต้นของ Unity3D คือการสร้าง GameObject ขึ้นมา และนำ Script ต่างๆ ไปใส่ไว้ใน GameObject นั้นๆ โดยการคลิกที่คำสั่ง GameObject ด้านบนของโปรแกรมและเลือก Create Empty เพื่อสร้าง Empty GameObject ขึ้นมา 1 ตัว ซึ่งผู้ใช้สามารถเปลี่ยนชื่อ GameObject ได้โดยการเปลี่ยนชื่อที่ Inspector และกด Enter



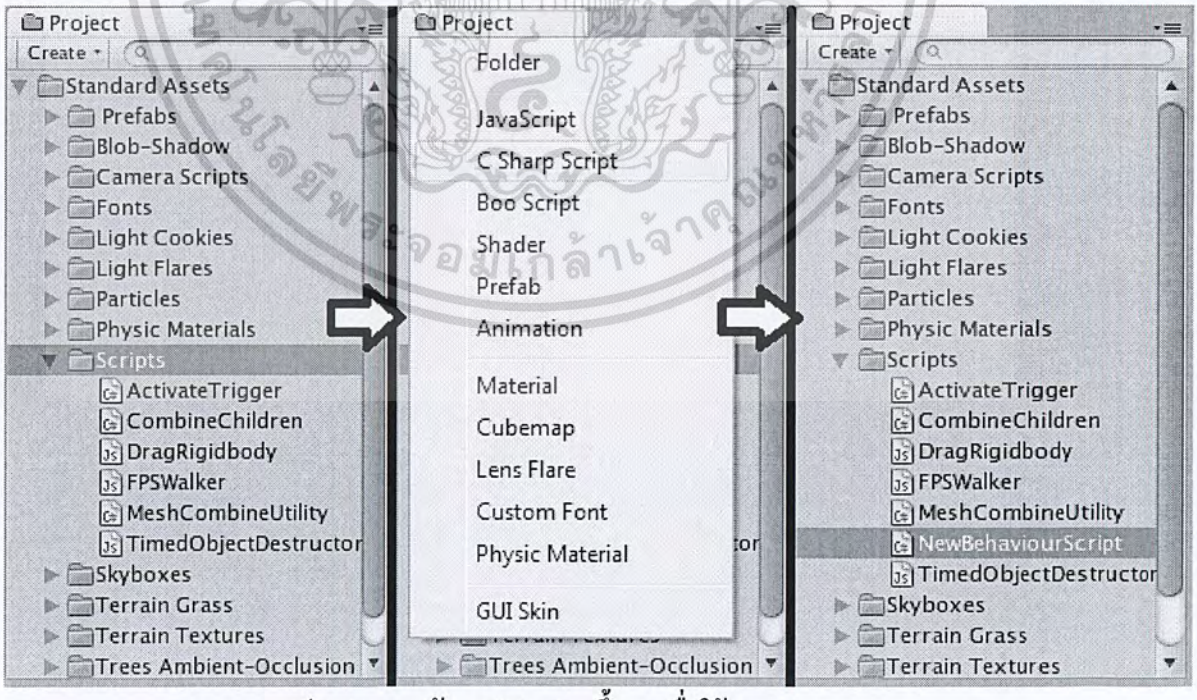
รูป 2.62 การสร้าง Empty GameObject

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.63 GameObject ที่ได้และที่เปลี่ยนชื่อ

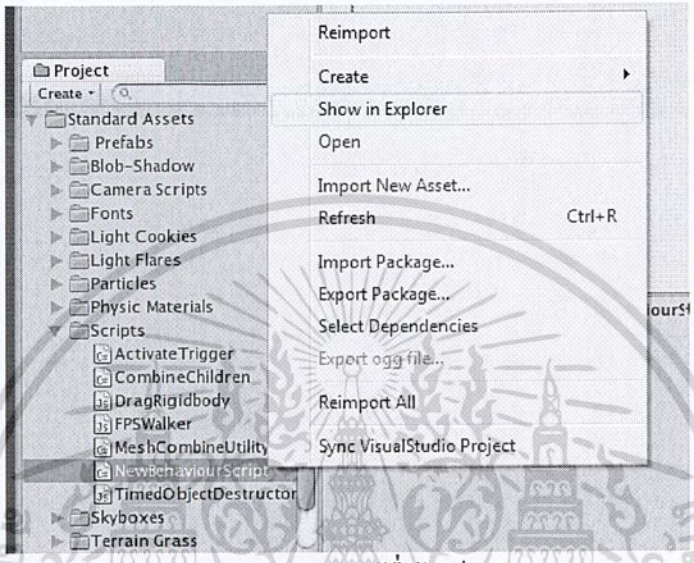
จากนั้นจะทำการสร้างScriptเพื่อนำไปใส่ใน GameObject ซึ่งการสร้างScriptนั้นทำได้โดยการเลือกโฟลเดอร์ Scripts จากนั้นคลิกซ้ายที่ปุ่ม Create ที่ด้านบนของ Project View และเลือก Create C# Script



รูป 2.64 การสร้าง C# Script ขึ้นมาเพื่อใช้งาน

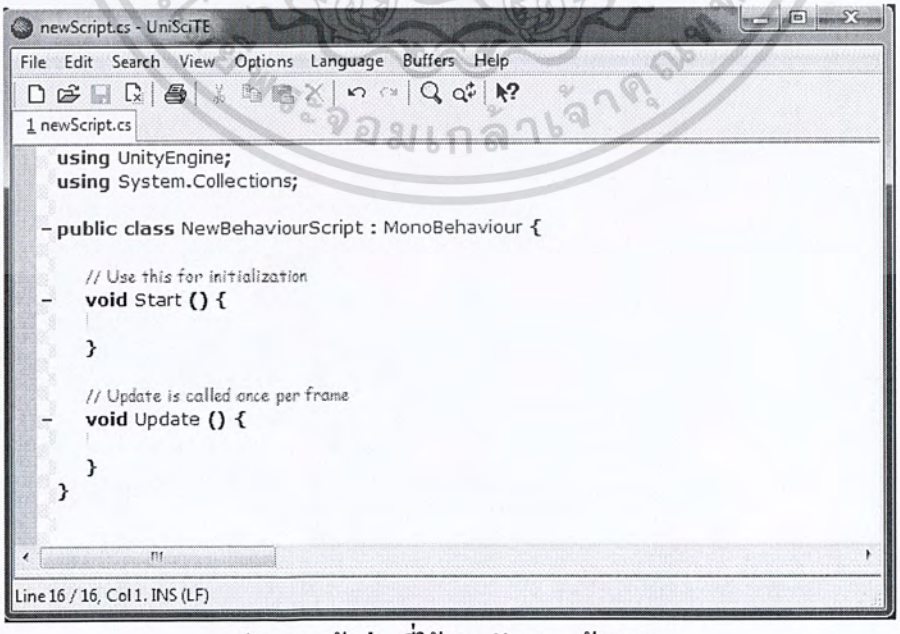
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากต้องการจะเปลี่ยนชื่อไฟล์ของ Script ทำได้โดยการคลิกขวาที่ Script นั้นๆ แล้วเลือก Show in Explorer เพื่อแก้ในหน้าต่างของโฟลเดอร์ แต่หากต้องการย้ายไฟล์ควรจะย้ายใน Project View เพราะตัว Project View จะทำการแก้ Path สำหรับการทำงานของ Assets ที่ถูกใช้ไปแล้วให้ด้วย



รูป 2.65 การแก้ชื่อไฟล์

จากนั้นคลิกซ้าย 2 ครั้งที่ Script ที่สร้างขึ้นมาใหม่ จะพบหน้าต่างที่ใช้สำหรับแก้ Script (สำหรับตัวอย่าง Script นั้นถูกเปลี่ยนชื่อเป็น newScript.cs)



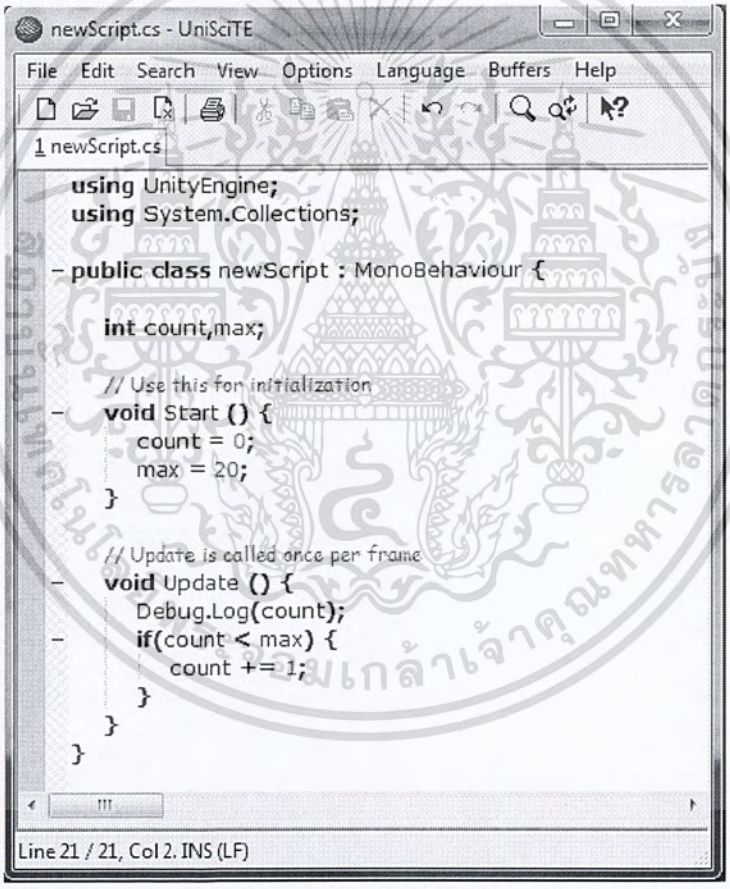
รูป 2.66 หน้าต่างที่ใช้สำหรับการแก้ Script

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งแรกที่ต้องทำหลังการเปลี่ยนชื่อคือการเปลี่ยนชื่อ Class ให้ตรงกับชื่อไฟล์ (หรือจะใช้ชื่อเดิมก็ได้ เพียงแต่ผู้ใช้ต้องเปลี่ยนชื่อ GameObject ที่จะใส่ Script นี้ให้มีชื่อตรงกับ Class ที่กำลังแก้ไขอยู่นี้)

จากหน้าต่างดังกล่าวผู้ใช้จะพบว่าตัวเกมเอนจิ้นนั้นได้สร้างโครงสร้างของ Script ให้เองแล้ว โดยมีเมธอด (Method) เริ่มต้น 2 เมธอด คือ Start() และ Update() ซึ่ง เมธอด Start() นั้นจะถูกสั่งให้ทำงานทันทีที่ GameObject นั้นถูกสร้างขึ้นมาหรือที่เรียกกันว่า Initialize นั้นเอง ส่วน Update() จะถูกเรียกให้ทำงานทุกๆ ครั้งที่เปลี่ยนเฟรม

สมมติว่า Script นั้นเป็นดังนี้ โดยฟังก์ชั่น Debug.Log(); จะทำการแสดงผลสิ่งที่อยู่ในวงเล็บขึ้นไปบน log ของโปรแกรมนั่นเอง

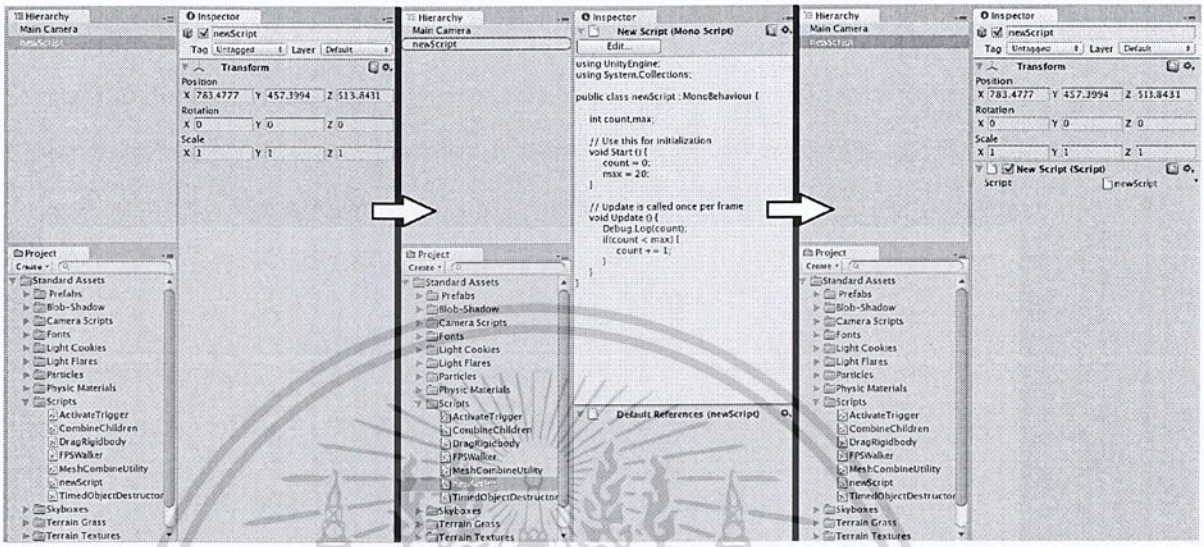


รูป 2.67 ตัวอย่าง Script

Script นี้จะทำการแสดงผลเลข 0 ถึง 20 ลงบน log ของ Unity3D เนื่องจากเมธอด Start() นั้นจะถูกเรียกเพียงตอนสร้าง GameObject เพียงครั้งเดียว ดังนั้นค่า count จะไม่มีวันกลับมาเป็น 0 อีก และเมื่อหน้าจอเปลี่ยนเฟรมจะทำการเรียก Update ขึ้นมาใช้ ซึ่งจะเพิ่มค่า count ขึ้นเรื่อยๆ จนถึง

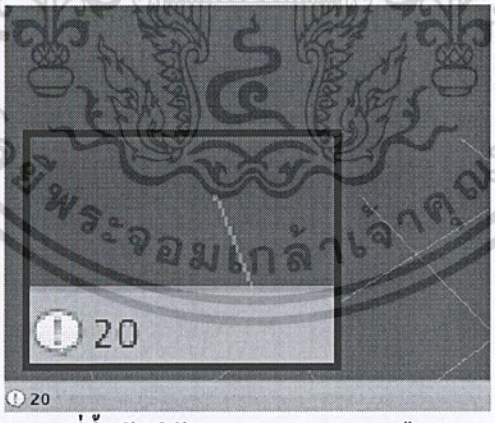
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการนำ Script ลงไปใน GameObject นั้นสามารถทำได้โดยการลากเอา Script ที่อยู่ใน Project View ขึ้นไปวางใส่ GameObject ที่อยู่บน Hierarchy



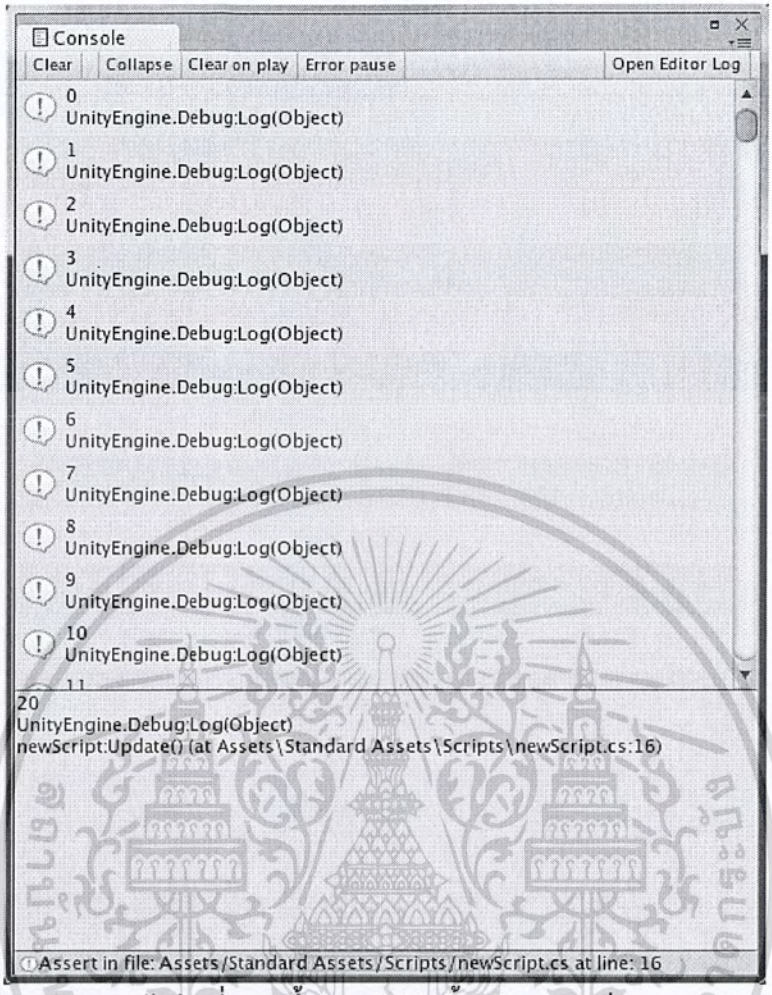
รูป 2.68 การใส่ Script ลงไปใน GameObject

จากนั้นก็กดปุ่ม Play ในส่วนของ Toolbar จะพบกับ Message ที่เกิดขึ้นด้านล่างของ Game View หรือ Scene View และเมื่อทำการดับเบิ้ลคลิกขึ้นมาจะพบส่วนของ Log



รูป 2.69 Log ที่ขึ้นข้างใต้ของ Game View หรือ Scene View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.70 หน้าต่างที่แสดงขึ้นมาหลังดับเบิ้ลคลิก ผลลัพธ์ของ Script

2.5.2 การทำงานด้านโมเดลและอนิเมชั่น

Unity3D เป็นเกมเอนจินที่รองรับการใช้งานโมเดลและอนิเมชั่น 3 มิติ โดยสามารถนำไฟล์โมเดลแบบต่างๆ เข้ามาใช้ภายในเอนจินเพื่อเรียกใช้งานตามที่ผู้พัฒนาเกมต้องการได้

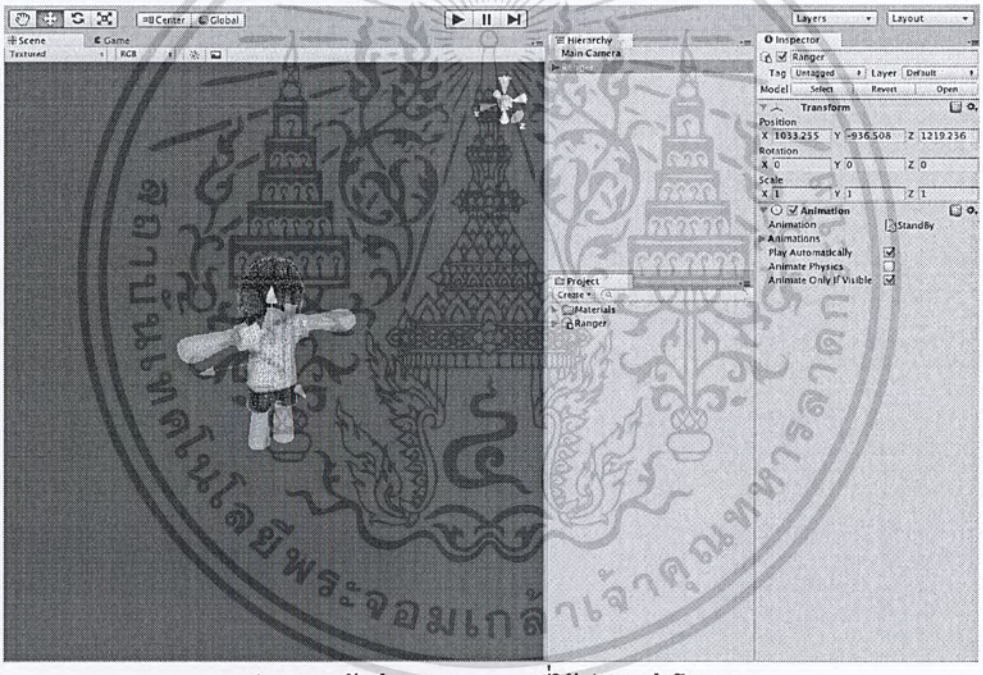
2.5.2.1 การนำเข้าโมเดล 3 มิติ

ในการนำโมเดลเข้าไปใช้ใน Unity3D โมเดลนั้นต้องเป็นโมเดลที่ถูกสร้างจากโปรแกรมสร้างโมเดลสามมิติที่ Unity3D รองรับด้วย ซึ่งโปรแกรมสร้างโมเดลสามมิติที่ Unity3D รองรับนั้นมีหลากหลายโปรแกรม ได้แก่ Maya, Cinema 4D, 3ds Max, Cheetah3d, Modo, Lightwave และ Blender ซึ่งภายในโครงงานนี้ได้นำเสนอโดยการใช้โปรแกรม Blender ตลอดทั้งโครงงาน เนื่องจากเป็นซอฟต์แวร์ที่สามารถนำมาใช้งานได้ฟรี นอกจากการพิจารณาโปรแกรมสร้างโมเดลสามมิติที่รองรับแล้วนั้น ยังต้องดูที่สกุลไฟล์โมเดลสามมิติด้วย ซึ่งสกุลไฟล์โมเดลสามมิติที่ Unity3D รองรับ ได้แก่ .fbx, .dae, .3ds, .dxf และ .obj แต่สกุลไฟล์ที่นิยมใช้กันคือ .fbx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปนี้เป็นตัวอย่างของการนำเข้าโมเดลจากโปรแกรม Blender ซึ่งในตัวอย่างนี้ได้มีการจัดเตรียมโมเดลที่มีสกุลไฟล์ .fbx ไว้เรียบร้อยแล้ว ซึ่งเป็นโมเดลที่สามารถนำเข้า Unity3D ได้โดยตรงเนื่องจากมีการรองรับสกุลไฟล์ดังกล่าว แต่ในกรณีที่สกุลไฟล์ของโมเดลไม่ได้ถูกรองรับโดย Unity3D นั้น ก็สามารถทำการแปลงสกุลไฟล์ให้อยู่ในรูปแบบที่ Unity3D รองรับได้โดยใช้โปรแกรมสร้างโมเดลสามมิติทั่วไปทำการแปลงได้

หลังจากที่ได้เตรียมโมเดลพร้อมใช้งานเรียบร้อยแล้ว ก็สามารถนำเข้า Unity3D ได้โดยคลิกขวาที่หน้าต่าง Project แล้วเลือก Import New Asset... แล้วทำการเลือกเลือกไฟล์โมเดลสามมิติที่ได้ทำการเตรียมไว้ (เราสามารถนำเข้าสิ่งอื่นๆที่ Unity3D รองรับนอกเหนือจากโมเดล ได้โดยใช้วิธีเดียวกันนี้) ซึ่งหลังจากนำเข้ามาแล้ว เราสามารถปรับแต่งขนาดของโมเดลให้เล็กลงหรือใหญ่ขึ้นได้ในหน้าต่าง Inspector ดังรูป 2.73



รูป 2.71 หน้าต่าง Inspector ที่ใช้ปรับแต่งโมเดล

ขนาดของโมเดลที่ถูกนำเข้ามาจะถูกกำหนดขนาดมาจากโปรแกรมโมเดลสามมิติรวมถึงจุดศูนย์กลางของโมเดลด้วย ซึ่งหลังจากทำการนำเข้ามาใน Unity3D ให้ทำการคลิกลากโมเดลไปวางในหน้าต่าง Scene (ซึ่งเป็นการใช้งานแบบสแตติก) แล้วคลิกที่ชื่อโมเดลในหน้าต่าง Hierarchy จะสังเกตว่าเราสามารถแก้ไขขนาดได้ด้วยการเพิ่มจำนวนเท่าในทิศทางแกนต่างๆ ในช่อง Scale ของหน้าต่าง Inspector แต่ไม่สามารถแก้ไขตำแหน่งศูนย์กลางของตัวโมเดลได้ ดังนั้นจึงควรวางตำแหน่งให้ดีมาจากโปรแกรมสร้างโมเดลสามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

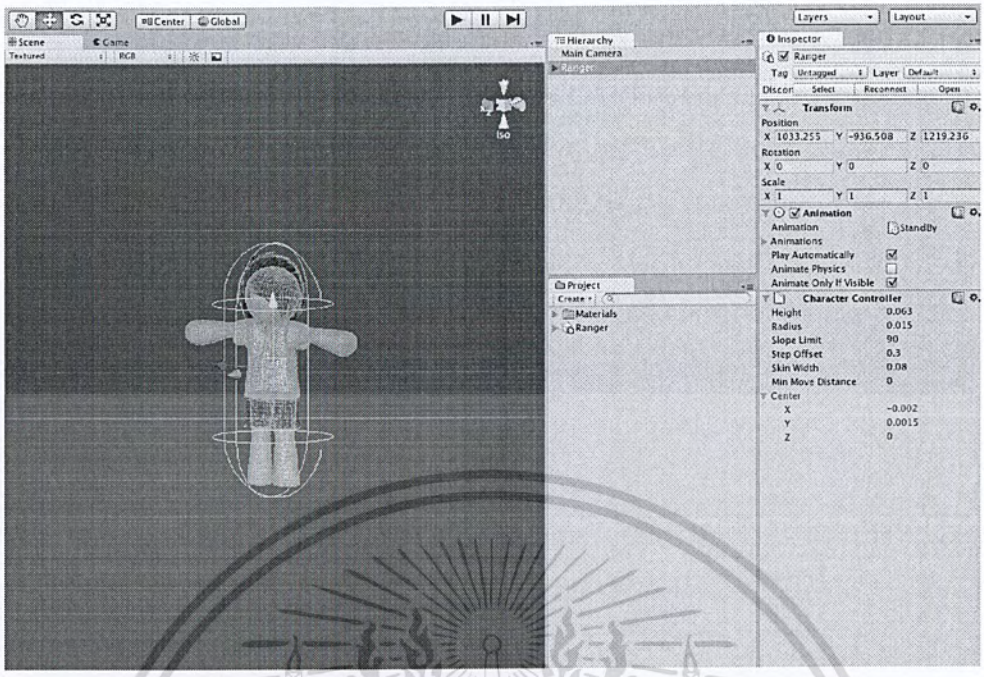
2.5.2.2 การเรียกใช้งานโมเดลและอนิเมชั่น

การนำโมเดลจากหน้าต่าง Project ออกมาใช้งานมี 2 วิธีคือ การใช้งานแบบสแตติก ทำได้โดยการลากลงไปวางในหน้าต่าง Scene ซึ่งโมเดลจะปรากฏให้เห็นภายในเกมตั้งแต่ตอนเริ่มเกม ส่วนอีกวิธีคือการใช้งานแบบไดนามิก ทำได้โดยการเลือกใช้งานจากโค้ด เพื่อให้โมเดลถูกสร้างขึ้นมาในระหว่างที่เกมกำลังดำเนินอยู่ ซึ่งสามารถเลือกใช้ได้ทั้ง 2 วิธีตามความเหมาะสม

ในการใช้งานโมเดลที่มีการสร้างเกินหนึ่งครั้งขึ้นไป หรือมีการใช้งานซ้ำ เราควรใช้สิ่งที่เรียกว่า Prefab เพราะ Prefab จะทำหน้าที่เป็นเสมือนแม่แบบของโมเดลใดๆ เช่น ถ้ามีการนำเข้าโมเดลเข้ามา แล้วทำ Prefab ให้โมเดลนั้น หลังจากนั้นทำการเพิ่มคอมโพเนนต์ต่างๆ หรือปรับแต่งโมเดลซึ่งอยู่ภายใต้ Prefab แล้ว เราสามารถนำโมเดลซึ่งถูกปรับแต่งแล้ว มาใช้งานได้หลายครั้งโดยที่การปรับแต่งต่างๆยังคงอยู่ในทุกๆโมเดลที่ถูกสร้างมาจาก Prefab เดียวกัน เป็นต้น ซึ่งถ้าไม่มีการทำ Prefab นั้น ก็ต้องปรับแต่งโมเดลทุกๆครั้งที่มีการเรียกใช้ แม้ว่าการปรับแต่งนั้นจะเหมือนๆกันก็ตาม ส่วนการสร้าง Prefab นั้น ทำได้โดยการสร้างโมเดลแบบสแตติกก่อน แล้วปรับแต่งให้เรียบร้อย หลังจากนั้นคลิกขวาในหน้าต่าง Project แล้วเลือก Create แล้วเลือก Prefab ก็จะได้ Prefab ขึ้นมาหนึ่งอัน แล้วทำการลากโมเดลที่อยู่ในหน้าต่าง Hierarchy ซึ่งถูกปรับแต่งแล้วไปใส่ใน Prefab ก็จะได้แม่แบบของโมเดลในการใช้งานครั้งต่อๆไป

ในการควบคุมโมเดลนั้นสามารถทำได้หลายวิธี แต่ในโครงงานนี้ได้ใช้งานจากคอมโพเนนต์ที่เรียกว่า Character Controller ซึ่งเป็นคอมโพเนนต์ที่ช่วยให้ผู้ใช้งานสามารถควบคุมโมเดลให้เคลื่อนที่ไปยังที่ต่างๆได้ รวมถึงมีตัวตรวจสอบการชนแบบแคปซูล (Capsule Collider) ซึ่งคอยตรวจว่ามีการชนเกิดขึ้น, ชนอะไร และทำการป้องกันไม่ให้โมเดลอื่นเข้ามาทับโมเดลได้ โดยการติดตั้งคอมโพเนนต์ Character Controller ให้แก่โมเดล ทำได้โดยการเลือกที่โมเดลที่ต้องการแล้วคลิกที่เมนู Component และเลือก Physics จากนั้นเลือก Character Controller ซึ่งหลังจากติดตั้งคอมโพเนนต์ดังกล่าวเรียบร้อยแล้ว ต้องทำการปรับขนาดของตัวตรวจสอบการชนให้มีขนาดเท่ากับโมเดล ซึ่งสามารถปรับแต่งได้ในส่วนของคอมโพเนนต์ Character Controller ที่เพิ่มเข้าไปดังรูป 2.41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.72 ปรับแต่งรายละเอียดของ Character Controller

หลังจากที่ติดตั้งคอมโพเนนต์ให้แก่โมเดลแล้ว ทำให้สามารถควบคุมโมเดลจากการเขียนโปรแกรมได้ง่ายขึ้น เนื่องจาก Unity3D ได้ทำการเตรียมไลบรารี่สำหรับช่วยการควบคุมโมเดลผ่านคอมโพเนนต์ Character Controller ไว้ให้แล้ว

ในการเรียกอนิเมชั่นของโมเดลภายใน Unity3D นั้น โมเดลที่มีอนิเมชั่นต้องมีสกุลไฟล์เป็น .fbx เนื่องจากไฟล์สกุล .fbx มีการเก็บข้อมูลรายละเอียดทางด้านกระดูกของโมเดล รวมถึงอนิเมชั่นต่างๆ โดยการเรียกให้โมเดลแสดงอนิเมชั่นนั้น ทำได้โดยการเขียนโปรแกรมเรียกใช้งานผ่านไลบรารี่ที่ Unity3D ได้เตรียมไว้ให้ ซึ่งสามารถปรับแต่งอนิเมชั่นได้ เช่น การแบ่งอนิเมชั่นออกเป็นหลายๆเลเยอร์, การจัดลำดับของอนิเมชั่นที่จะให้แสดง, การปรับเฟรมเรทของอนิเมชั่น เป็นต้น โดยค่าเฟรมเรทของอนิเมชั่นที่มาจากโปรแกรมสามมิตินั้น Unity3D ได้กำหนดให้มีค่าเท่ากับหนึ่ง ซึ่งเราสามารถเข้าไปปรับค่าเฟรมเรทนี้ให้เร็วขึ้นหรือช้าลงในโค้ดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

3.1 แนวคิดของเกม

การพัฒนาเกมจะต้องมีการกำหนดแนวคิดเบื้องต้นของเกม เพื่อให้ทราบถึงจุดเด่นของเกม และนำไปสู่การออกแบบเกมตามแนวคิดเหล่านี้

3.1.1 คอนเซปเบื้องต้น

จำลองการเล่นสงกรานต์ในรูปแบบเกมประเภท RTS (Real-Time Strategy)

3.1.2 เนื้อเรื่องย่อ

ณ ประเทศไทย ประเทศแห่งรอยยิ้ม ทุกๆ ปี เหล่าผู้กล้าจะออกมารวมตัวกันในช่วงเทศกาลสงกรานต์ เพื่อประลองกันแย่งชิงความเป็นหนึ่งในการเล่นสงกรานต์ คุณจะต้องรวบรวมเหล่าผู้กล้าและนำพวกเขาเข้าประลองกับเหล่าผู้กล้ากลุ่มอื่นๆ ซึ่งการจะเอาชนะผู้กล้ากลุ่มอื่นๆ ได้นั้น จะต้องอาศัยไหวพริบและแผนการเล่นสงกรานต์อันแยบยลของคุณเอง

3.1.3 จุดเด่น

1) ใช้มุมกล้องแบบมุมมองบุคคลที่สาม
2) มีตัวละคร 6 ประเภท ได้แก่ Tank, Melee, Ranger, Siege, Healer, และ Summoner
3) สามารถสร้างทีมด้วยตัวละคร 4 ตัว โดยเลือกจากตัวละคร 6 ประเภท
4) สามารถบังคับตัวละครให้เคลื่อนที่, ยิงปืนฉีดน้ำ, และใช้ความสามารถพิเศษได้
5) สามารถบังคับตัวละครได้ 4 ตัวพร้อมๆ กัน
6) NPC (Non Player Character) ถูกควบคุมโดยโครงข่ายประสาทเทียม
7) โครงข่ายประสาทเทียมผ่านการเรียนรู้โดยใช้ NEAT (NeuroEvolution of Augmenting Topologies)
8) สามารถเรียกดูประวัติความเป็นมาของเทศกาลสงกรานต์ได้
9) สามารถเล่นได้ทั้งบนระบบปฏิบัติการ Windows และ MAC OS
10) สามารถปรับเปลี่ยน Resolution ได้

3.1.4 ชื่อ, ประเภท, แพลทฟอร์ม, และกลุ่มเป้าหมายของเกม

เกมนี้มีชื่อว่า เทศกาลสงกรานต์: สแปลชอาวร์ โดยถูกพัฒนาให้เป็นเกมประเภท RTS (Real-Time Strategy) ลงบนแพลทฟอร์ม PC (Windows and MAC OS) เหมาะสำหรับผู้เล่นทุกเพศที่มีอายุระหว่าง 15 – 28 ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบเกม

การจะพัฒนาเกมขึ้นมานั้น จำเป็นจะต้องออกแบบเกมในหลายๆ ด้าน ทั้งในด้านรูปแบบของเกม, องค์ประกอบศิลป์ของเกม, และการออกแบบด้านเทคนิคต่างๆ

3.2.1 การออกแบบรูปแบบของเกม

การออกแบบรูปแบบของเกมทำขึ้นเพื่อสร้างความเข้าใจต่อรูปแบบการเล่น, การควบคุม, และภาพรวมในหลายๆ ง่ายที่เกี่ยวข้องกับเกมดีไซน์ (Game Design)

3.2.1.1 รูปแบบและกติกาการเล่น

ผู้เล่น 1 คนสามารถตั้งทีม ที่มีสมาชิกในทีมได้ 4 ตัวละคร โดยเลือกเอาจากตัวละครที่มีทั้งหมด 6 แบบ สามารถเลือกซ้ำได้ไม่เกิน 2 ตัว เมื่อเลือกแล้ว ผู้เล่นจะต้องนำกทีมของตนเข้าเล่นน้ำกับทีมที่ควบคุมโดยปัญญาประดิษฐ์ทั้งหมดจำนวน 1-3 ทีม โดยมีเวลาจำกัด 10 นาที

ในการเริ่มเกมจะทำการสุ่มที่อยู่ของกลุ่มต่างๆ ในแผนที่ และมีระดับความสะอาดเป็นค่าพลังชีวิตของตัวละครแต่ละตัวซึ่งจะมีค่าเต็มเมื่อเริ่มเกม ในการจู่โจมนั้นตัวละครแต่ละตัวจะมีทรัพยากรที่ใช้ในการจู่โจมจำกัด ยกตัวอย่างเช่น การใช้ปืนฉีดน้ำจะต้องใช้ทรัพยากรน้ำ การใช้แป้งเข้าปะจำเป็นต้องใช้ทรัพยากรแป้ง และเมื่อทรัพยากรใกล้จะหมดแล้ว ผู้เล่นสามารถหาทรัพยากรได้จากกล่องที่สุ่มตกทั่วไปในแผนที่ เมื่อตัวละครถูกจู่โจมจะทำให้ระดับความสะอาดลดลง และเมื่อระดับความสะอาดลดลงจนหมด ตัวละครจะหนีออกจากฉากซึ่งหากทีมใดที่ตัวละครหนีออกจากฉากจนหมดจะแพ้ทันที ระดับความสะอาดนั้นสามารถถูกเพิ่มได้โดยความสามารถของตัวละครประเภท Healer

ในการตัดสินแพ้ชนะ จะตัดสินจากทีมที่มีตัวละครเหลืออยู่ในฉากเป็นทีมสุดท้าย หรือในกรณีที่หมดเวลาการแข่งขัน ก็จะนับระดับความสะอาดของตัวละครที่เหลืออยู่ทั้งหมด ทีมที่มีระดับความสะอาดเหลือมากที่สุดจะเป็นผู้ชนะ

3.2.1.2 วิธีการควบคุม

ผู้เล่นสามารถใช้เมาส์และคีย์บอร์ดควบคุมการทำงานของเกมได้ดังนี้

1) เลื่อนมุมกล้องได้โดยการเลื่อนเคอเซอร์เมาส์ไปที่ขอบของหน้าจอในทิศทางที่ต้องการจะเลื่อนมุมกล้องไป
2) หมุนมุมกล้องได้โดยการคลิกเมาส์ปุ่มกลางแล้วเลื่อนเมาส์ไปตามทิศทางที่ต้องการ
3) เลือกตัวละครโดยการคลิกซ้ายบนตัวละครที่ต้องการ
4) เลือกกลุ่มของตัวละครโดยการคลิกซ้ายแล้วลากกรอบสี่เหลี่ยมครอบกลุ่มของตัวละครที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เลือกตัวละครโดยการกดแป้นเลข 1, 2, 3, และ 4 บนคีย์บอร์ดตามหมายเลขตัวละครในทีมที่ต้องการ
6) เลือกตัวละครในทีมทั้งหมดโดยการกดแป้นเลข 5 บนคีย์บอร์ด
7) เมื่อเลือกตัวละครอยู่สามารถคลิกขวาไปยังพื้นที่ว่างเพื่อสั่งให้ตัวละครเดินไปยังพื้นที่นั้นได้
8) เมื่อเลือกตัวละครอยู่สามารถคลิกขวาบนตัวละครของศัตรูเพื่อสั่งโจมตีได้
9) เมื่อเลือกตัวละครอยู่สามารถเลือกความสามารถพิเศษของตัวละครได้โดยการคลิกเมาส์ซ้ายบนไอคอนรูปความสามารถของตัวละครที่เลือกอยู่
10) เมื่อเลือกตัวละครอยู่สามารถเลือกความสามารถพิเศษของตัวละครได้โดยการกดแป้นตัวอักษรตามตัวอักษรบนไอคอนรูปความสามารถของตัวละครที่เลือกอยู่
11) เมื่อเลือกความสามารถพิเศษของตัวละครแล้วสามารถเลือกเป้าหมายได้โดยการคลิกซ้ายบนตัวละครที่ต้องการ
12) เมื่อเลือกตัวละครอยู่สามารถคลิกขวาบนกล่องเพื่อเติมทรัพยากรของตัวละครได้
13) กดแป้น h บนคีย์บอร์ดขณะเล่นสำหรับเรียกดูข้อมูลช่วยเหลือเบื้องต้น
14) กดแป้น ESC บนคีย์บอร์ดขณะเล่นเพื่อหยุดเกมและเรียกดูเมนูย่อย
15) คลิกซ้ายบนปุ่มต่างๆ ภายในเกมเพื่อเลือกปุ่มเหล่านั้น

3.2.1.3 ความสามารถของตัวละคร

เนื่องจากตัวละครแต่ละประเภทมีความเก่งแตกต่างกันออกไป ตัวละครแต่ละตัวจึงถูกออกแบบให้มีพลังและความสามารถแตกต่างกัน ดังตาราง 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.1 พลังและความสามารถของตัวละครแต่ละประเภท

ข้อมูลตัวละคร						
Name	Raincoater	Powder Smearer	Watergunslinger	Water Bomber	Caretaker	Charmer
ประเภท	Tank	Melee	Ranger	Siege	Healer	Summoner
หน้าที่	ปกป้องสมาชิกในทีม	บุกประแป้งระยะใกล้	ยิงปืนฉีดน้ำจากระยะไกล	จู่โจมระยะไกลข้ามสิ่งกีดขวาง	ทำความสะอาดเพื่อร่วมทีม	เรียกผู้ช่วยมาปกป้อง
ความสะอาด	500	420	380	360	320	320
ทรัพยากร	100	100	100	100	50	20
พลังงาน	0	0	0	0	50	80
ทรัพยากรที่ได้ต่อกล่อง	50	50	50	50	25	10
พลังงานที่ได้ต่อกล่อง	0	0	0	0	25	40
ความเร็วในการเคลื่อนที่ (m/s)	0.8	1.6	1.4	1	1.2	1.2
อาวุธ	Umbrella water gun	Bowl of powder	Dual water gun, Water sniper	Water bomb	Water gun Jr.	Bubble gun
ความสามารถพิเศษ	Bunker	Dash, Lake of powder	Change stance	Hareruya, Kamikaze	Towel, Hair dryer, Giant electric fan	Motor-tricycle, Vanz boy 'N' Skoyz girl, Pick-up truck, Fire apparatus

เช่นเดียวกับอาวุธที่ตัวละครแต่ละตัวใช้ จำเป็นต้องมีพลังความสามารถแตกต่างกันตามหน้าที่และลักษณะเด่นของตัวละครแต่ละตัว อาวุธแต่ละอย่างจึงถูกออกแบบตามตาราง 3.2

ตาราง 3.2 พลังและความสามารถของอาวุธแต่ละประเภท

ข้อมูลอาวุธ					
ชื่อ	ระยะการโจมตี (m)	เป้าหมาย	พลังโจมตี (ความเลอะ / s)	ทรัพยากรที่ใช้ (ทรัพยากร / s)	หมายเหตุ
Umbrella water gun	5	1	5	1	
Bowl of powder	1	1	30	10	
Dual water gun	10	1	10	2	ความเร็วในการเคลื่อนที่ = 1.4 m/s
Water sniper	20	1	35	5	ความเร็วในการเคลื่อนที่ = 0.6 m/s
Water bomb	10	วงกลมรัศมี 3 m	30	10	
Water gun Jr.	3	1	5	1	
Bubble gun	3	1	2	1	

ความสามารถพิเศษของตัวละครแต่ละตัวมีผลแตกต่างกันออกไปเมื่อถูกใช้ เนื่องจากมีจุดประสงค์ในการใช้แตกต่างกัน ความสามารถพิเศษในเกมมีทั้งหมดดังนี้

1) Bunker เป็นความสามารถแบบเปิดปิดของ Tank ใช้เวลาในการเรียกใช้ 3 วินาที สามารถเรียกใช้ได้ใหม่ทันที ไม่ต้องการเป้าหมาย ไม่ใช้พลังงาน มีผลทำให้มีร่มยักษ์ตั้งอยู่ด้านหน้าตัวละคร เพื่อป้องกันความเสียหายและการโจมตีระยะประชิดจากด้านหน้า
2) Dash เป็นความสามารถแบบเรียกใช้ของ Melee เรียกใช้ได้ทันทีแต่หากจะเรียกใช้ซ้ำต้องรอ 15 วินาที ไม่ต้องการเป้าหมาย ไม่ใช้พลังงาน มีผลทำให้ตัวละครพุ่งไปข้างหน้าทันที 5 เมตร ซึ่งถ้าเกิดไปชนกับตัวละครตัวอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรัพยากรของตัวเองจะลดลง 10 และส่งผลให้ตัวละครที่ถูกชนถูกลดระดับความสะอาดลง 60

3) Lake of powder เป็นความสามารถแบบเรียกใช้ของ Melee ใช้เวลาในการเรียกใช้ 1 วินาทีแต่หากจะเรียกใช้ซ้ำต้องรอ 5 วินาที ใช้ทรัพยากร 20 มีผลทำให้มีพื้นที่ที่เต็มไปด้วยแป้งกว้าง 2 เมตร ยาว 6 เมตรเกิดขึ้นด้านหน้าของตนเป็นเวลา 10 วินาที ซึ่งทำให้ตัวละครที่อยู่ภายในพื้นที่นี้จะเคลื่อนที่ช้าลง 2 เท่า

4) Change Stance เป็นความสามารถแบบเรียกใช้ของ Ranger ใช้เวลาในการเรียกใช้ 2 วินาที สามารถเรียกใช้ซ้ำได้ทันที ไม่ต้องการเป้าหมาย ไม่ใช้พลังงาน มีผลทำให้เกิดการสลับอาวุธระหว่าง Dual water gun กับ Water sniper

5) Hareruya เป็นความสามารถแบบเรียกใช้ของ Area ใช้เวลาในการเรียกใช้ 1 วินาทีแต่หากจะเรียกใช้ซ้ำต้องรอ 60 วินาที มีเป้าหมายเป็นพื้นที่วงกลมรัศมี 5 เมตรที่มีจุดศูนย์กลางห่างจากตัวเองได้มากสุด 10 เมตร ใช้ทรัพยากรที่เหลืออยู่ทั้งหมด มีผลทำให้ศัตรูที่อยู่ในขอบเขตการโจมตีถูกลดระดับความสะอาดลง 2.5 เท่าของทรัพยากรที่ใช้ไปกับความสามารถนี้

6) Kamikaze เป็นความสามารถแบบเรียกใช้ของ Area ใช้เวลาในการเรียกใช้ 1 วินาที เมื่อใช้แล้วระดับความสะอาดของตัวเองจะมีค่าเป็น 0 ทันที มีเป้าหมายเป็นพื้นที่วงกลมรัศมี 5 เมตรที่มีจุดคือตำแหน่งของตัวเอง ใช้ทรัพยากรที่เหลืออยู่ทั้งหมด มีผลทำให้ศัตรูที่อยู่ในขอบเขตการโจมตีถูกลดระดับความสะอาดลง 1/6 เท่าของระดับความสะอาดที่เหลืออยู่ของตัวเอง และ 3 เท่าของทรัพยากรที่ใช้ไปกับความสามารถนี้

7) Towel เป็นความสามารถแบบใช้อย่างต่อเนื่องของ Healer หากจะเรียกใช้ซ้ำต้องรอ 5 วินาที มีเป้าหมายคือเพื่อนร่วมทีม 1 คนที่อยู่ห่างจากตนเอง 1 เมตร ใช้พลังงาน 2 หน่วย มีผลทำให้ระดับความสะอาดของเป้าหมายเพิ่มขึ้น 10 หน่วยต่อวินาที เป็นเวลา 3 วินาที

8) Hairdryer เป็นความสามารถแบบเรียกใช้ของ Healer ใช้เวลาในการเรียกใช้ 1 วินาที หากจะเรียกใช้ซ้ำต้องรอ 12 วินาที มีเป้าหมายคือเพื่อนร่วมทีม 1 คนที่อยู่ห่างจากตนเอง 1 เมตร ใช้พลังงาน 14 หน่วย มีผลทำให้ระดับความสะอาดของเป้าหมายเพิ่มขึ้นทันที 70 หน่วย

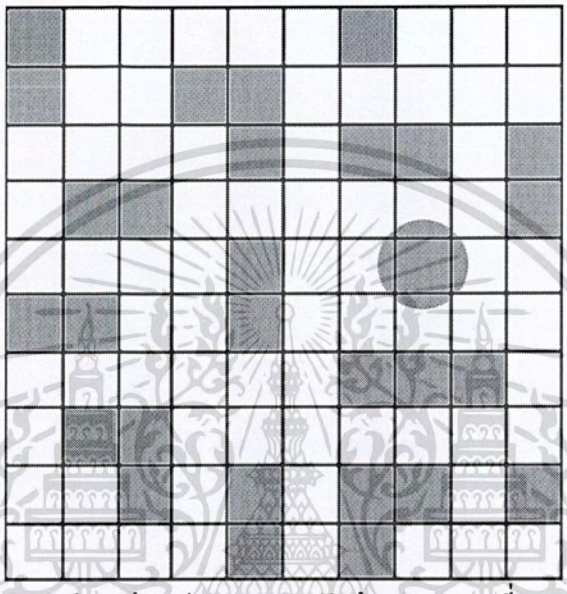
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) Giant electric fan เป็นความสามารถแบบเรียกใช้ของ Healer ใช้เวลาในการเรียกใช้ 5 วินาทีแต่หากจะเรียกใช้ซ้ำต้องรอ 120 วินาที มีเป้าหมายเป็นพื้นที่วงกลมรัศมี 5 เมตรที่มีจุดศูนย์กลางห่างจากตัวเองได้มากสุด 5 เมตร ใช้พลังงาน 28 หน่วย มีผลทำให้เพื่อนร่วมทีมที่อยู่ในขอบเขตของความสามารถนี้ระดับความสะอาดเพิ่มขึ้น 20 หน่วยต่อวินาที โดยพื้นที่เป้าหมายจะอยู่ได้นาน 20 วินาที
10) Motor-tricycle เป็นความสามารถแบบเรียกใช้ของ Summoner ใช้เวลาในการเรียกใช้ 3 วินาทีแต่หากจะเรียกใช้ซ้ำต้องรอ 60 วินาที มีเป้าหมายเป็นพื้นที่วงกลมรัศมี 4 เมตรที่มีจุดศูนย์กลางห่างจากตัวเองได้มากสุด 15 เมตร ใช้พลังงาน 10 หน่วย มีผลทำให้ศัตรูที่อยู่ในขอบเขตการโจมตีถูกลดระดับความสะอาดลง 80 หน่วย มีผลน้อยลงตามระยะห่างของศัตรูกับจุดศูนย์กลางของพื้นที่ของความสามารถนี้
11) Vanz boy 'N' Skoyz girl เป็นความสามารถแบบเรียกใช้ของ Summoner ใช้เวลาในการเรียกใช้ 3.5 วินาทีแต่หากจะเรียกใช้ซ้ำต้องรอ 60 วินาที มีเป้าหมายเป็นพื้นที่กว้าง 5 เมตร ยาว 7 เมตร ที่มีจุดศูนย์กลางห่างจากตัวเองได้มากสุด 15 เมตร ใช้พลังงาน 20 หน่วย มีผลทำให้ศัตรูที่อยู่ในขอบเขตการโจมตีถูกลดระดับความสะอาดลง 40 หน่วยต่อวินาที โดยพื้นที่เป้าหมายจะอยู่ได้นาน 3 วินาที
12) Pick-up truck เป็นความสามารถแบบเรียกใช้ของ Summoner ใช้เวลาในการเรียกใช้ 4 วินาทีแต่หากจะเรียกใช้ซ้ำต้องรอ 90 วินาที มีเป้าหมายเป็นพื้นที่วงกลมรัศมี 7 เมตร ที่มีจุดศูนย์กลางห่างจากตัวเองได้มากสุด 15 เมตร ใช้พลังงาน 40 หน่วย มีผลทำให้ศัตรูที่อยู่ในขอบเขตการโจมตีถูกลดระดับความสะอาดลง 40 หน่วยต่อวินาที โดยพื้นที่เป้าหมายจะอยู่ได้นาน 3 วินาที และถูกลดระดับความสะอาดเพิ่มเติมอีก 70 หน่วยสองครั้ง แต่การลดเพิ่มเติมนี้จะมีผลลดลงตามระยะห่างระหว่างศัตรูกับจุดศูนย์กลางของพื้นที่ของความสามารถนี้
13) Fire apparatus เป็นความสามารถแบบเรียกใช้ของ Summoner ใช้เวลาในการเรียกใช้ 6 วินาทีแต่หากจะเรียกใช้ซ้ำต้องรอ 120 วินาที มีเป้าหมายเป็นพื้นที่กว้าง 7 เมตร ยาว 10 เมตร ที่มีจุดศูนย์กลางห่างจากตัวเองได้มากสุด 15 เมตร ใช้พลังงาน 60 หน่วย มีผลทำให้ศัตรูที่อยู่ในขอบเขตการโจมตีถูกลดระดับความสะอาดลง 100 หน่วยต่อวินาที โดยพื้นที่เป้าหมายจะอยู่ได้นาน 5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.4 คุณสมบัติของแผนที่ที่ใช้ในการเล่น

แผนที่ที่ใช้ในการเล่นเป็นแผนที่ที่จำลองสภาพใจกลางเมืองกรุงเทพมหานคร ขนาด 100 ตารางเมตร ดังรูป 3.1 ประกอบไปด้วยอาคารมากมาย, ถนน, และน้ำพุ โดยมีตำแหน่งที่ตัวละครแต่ละทีมจะถูกจัดวางไว้ตอนเริ่มเกมทั้งหมด 6 ตำแหน่ง มีตำแหน่งของกล่อง 16 ตำแหน่ง ใช้เพลงประกอบฉากเป็นเพลงรำวงวันสงกรานต์

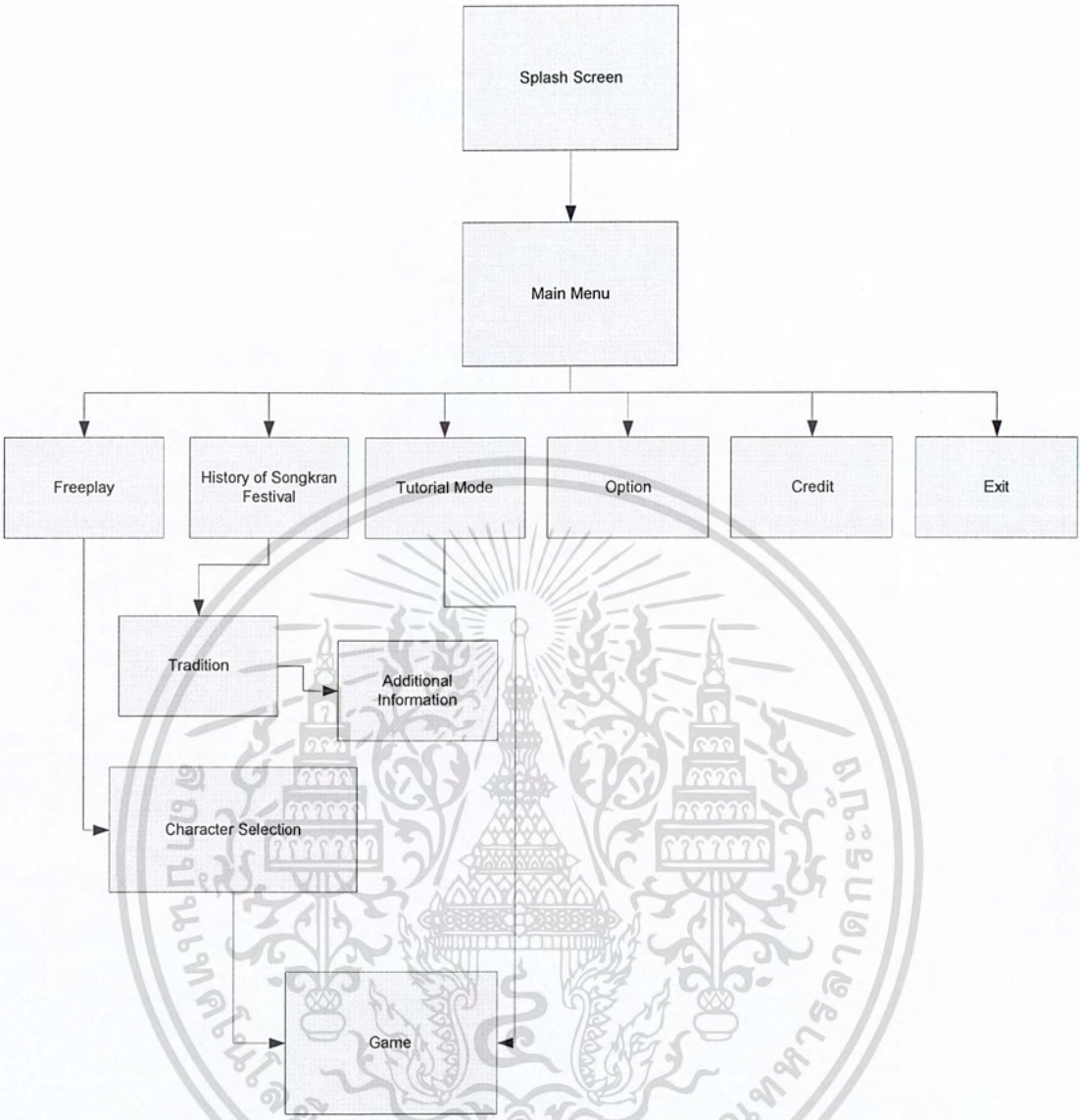


รูป 3.1 ส่วนประกอบและสัดส่วนของแผนที่

3.2.1.5 เมนูทรี (Menu Tree)

หน้าจอแต่ละหน้าจอของเกมถูกออกดังรูป 3.2 โดยมีเมนูหลักที่เก็บรวบรวมปุ่มไปยังหน้าจอหลักอื่นๆ และออกแบบให้มีจำนวนหน้าจอที่จะเข้าถึงการเล่นให้น้อยที่สุด เพื่อให้ผู้เล่นได้เข้าเล่นได้อย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.2 เมนูทรีของเกม

3.2.2 การออกแบบองค์ประกอบศิลป์ของเกม

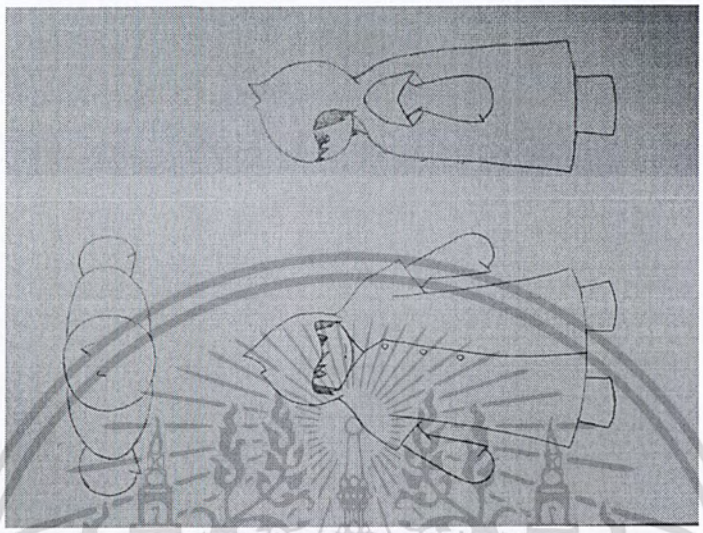
เกม เทศกาลสงกรานต์: สแปลชอาวร์ เป็นเกมที่เน้นกลุ่มเป้าหมายอายุ 15 – 28 ปี ดังนั้น จึงออกแบบสีสันให้สดใส แฝงด้วยความเท่เล็กน้อย เน้นความน่ารักแบบพอเหมาะกับอายุผู้เล่น

3.2.2.1 การออกแบบตัวละคร

ตัวละครถูกออกแบบให้เป็นแนว SD (Super Deformed) คือ ไม่สมจริง แต่เน้นเป็นแบบการ์ตูน เพื่อทำให้สามารถออกแบบการการเล่นน้ำแบบเท่ๆ ได้ ตัวละครแต่ละตัวถูกออกแบบดังนี้

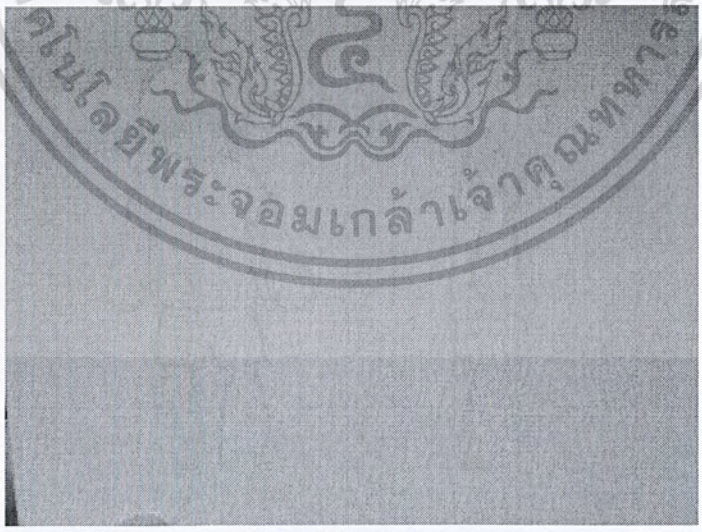
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) Tank ถูกออกแบบดังรูป 3.3 เป็นตัวละครที่ต้องปกป้องเพื่อนๆ จากน้ำ โดยจึงต้องใส่เสื้อกันฝน และมาด้วยความสามารถในการกางร่มยักษ์เพื่อปกป้องเพื่อนๆ



รูป 3.3 ตัวละคร Tank

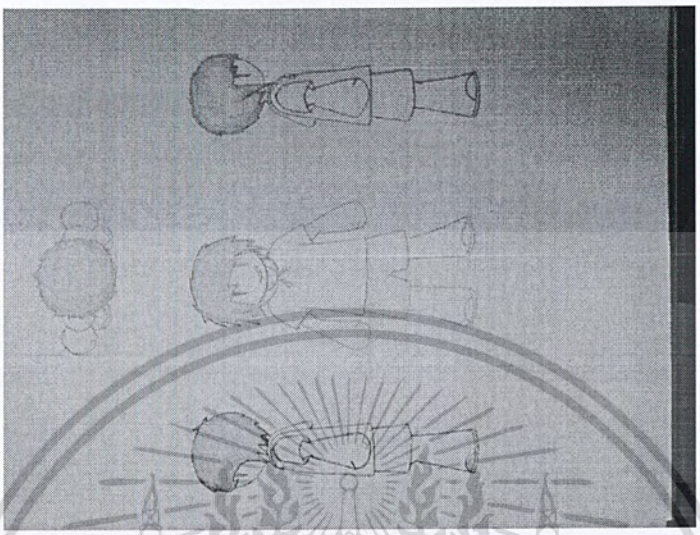
2) Melee ถูกออกแบบดังรูป 3.4 เป็นตัวละครที่ใส่เสื้อยืดกางเกงขาสั้นแบบสบายๆ เพราะต้องเน้นความรวดเร็วในการเข้าปะแป้งในระยะใกล้



รูป 3.4 ตัวละคร Melee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Ranger ถูกออกแบบดังรูป 3.5 เป็นตัวละครที่มาพร้อมกับปืนฉีดน้ำทั้งแบบระยะไกลและระยะใกล้ โดยมีจุดเด่นที่เสื้อฮูด



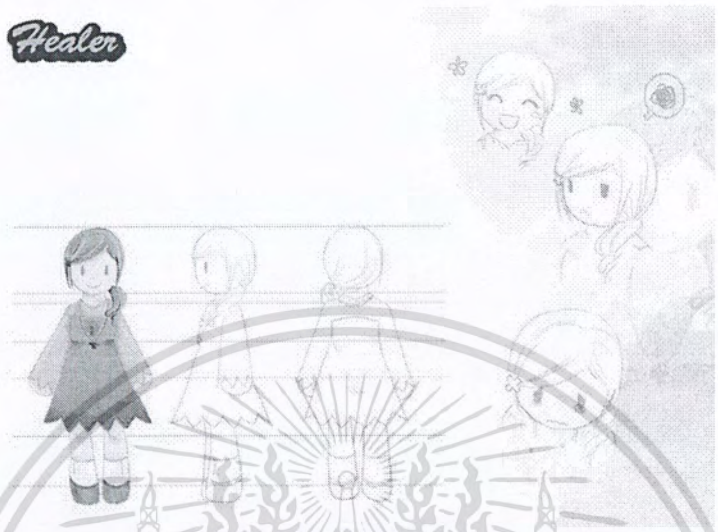
รูป 3.5 ตัวละคร Ranger

4) Area ถูกออกแบบดังรูป 3.6 เป็นตัวละครที่มีลูกโป่งน้ำเป็นอาวุธที่สามารถระเบิดน้ำออกเป็นวงกว้าง จึงมีจุดเด่นที่ทรงผมคล้ายหยดน้ำ

รูป 3.6 ตัวละคร Area

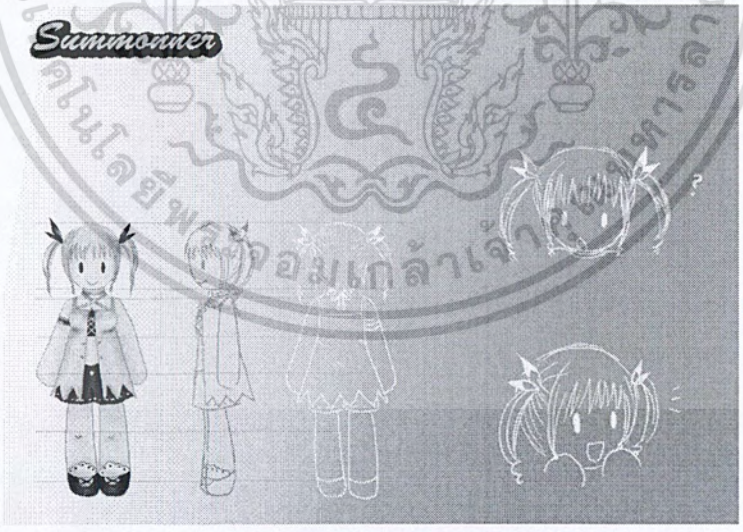
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) Healer ถูกออกแบบดังรูป 3.7 เป็นตัวละครที่มีความโอบอ้อมอารี แต่งตัวด้วยสีสันสดใส เน้นสีชมพูเป็นสีประจำตัว



รูป 3.7 ตัวละคร Healer

6) Summoner ถูกออกแบบดังรูป 3.8 เป็นตัวละครที่สามารถเรียกเหล่าเพื่อนผู้หลงไหลในความน่ารักของตนเองออกมาช่วยเหลือ



รูป 3.8 ตัวละคร Summoner

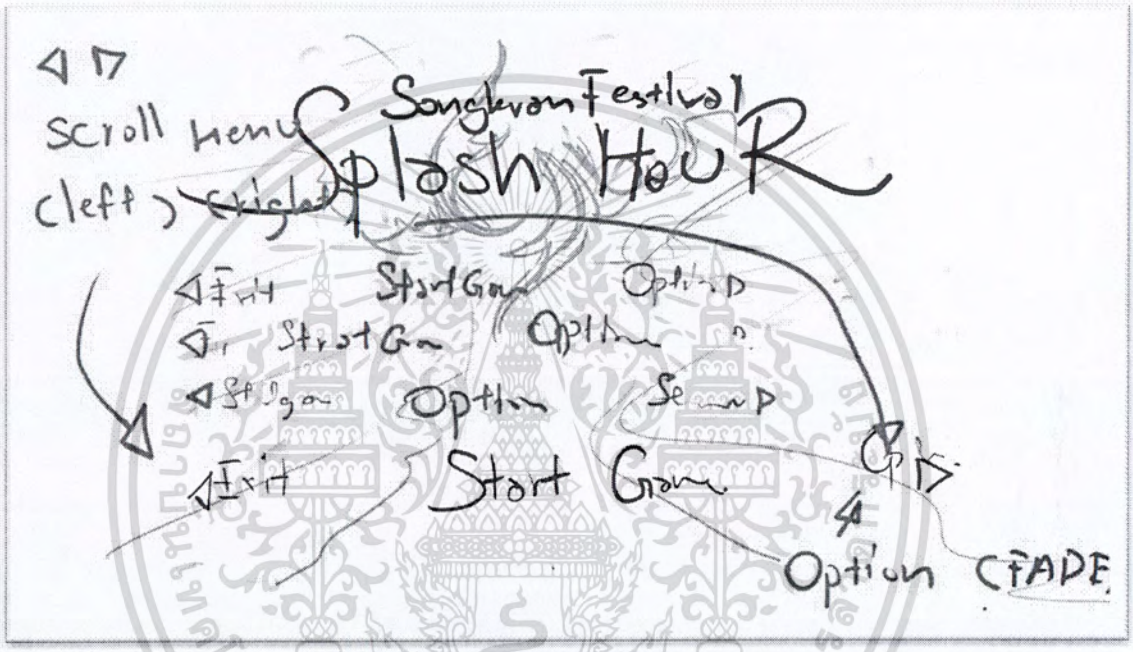
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 การออกแบบหน้าจอ

ในหัวข้อนี้จะนำเสนอการออกแบบหน้าจอบางหน้าจอภายในเกม โดยยกมาเฉพาะหน้าจอที่ใช้หลักๆ เท่านั้น

3.2.2.2.1 หน้าจอ Main Menu

หน้าจอ Main Menu ถูกออกแบบให้มีโลโก้และชื่อเกมอยู่ด้านบนของจอ และมีเมนูที่สามารถเลื่อนได้อยู่ด้านล่างของจอ ดังรูป 3.9

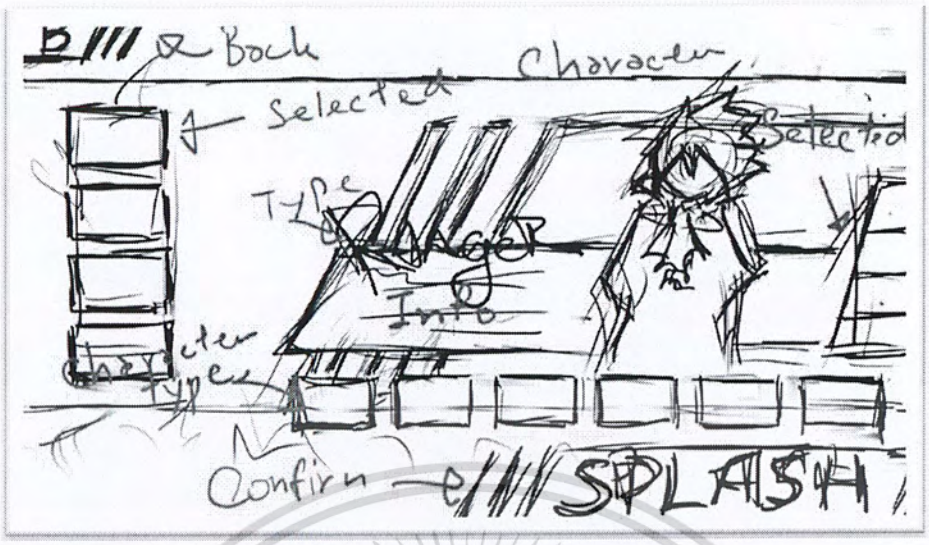


รูป 3.9 หน้าจอ Main Menu

3.2.2.2.2 หน้าจอ Character Selection

หน้าจอ Character Selection ถูกออกแบบดังรูป 3.10 ให้มีแถบตัวละครที่ให้เลือกอยู่ทางด้านล่างของจอ มีปุ่มเลือกตำแหน่งในการจัดวางตัวละครอยู่ทางขวาของหน้าจอ และมีกรอบแสดงตัวละครที่เลือกอยู่ปัจจุบันอยู่ทางซ้ายของหน้าจอ เมื่อผู้เล่นเลือกตัวละครจนครบแล้วจึงกดปุ่ม SPLASH ที่อยู่ทางด้านขวาล่างของจอเพื่อเข้าเล่น

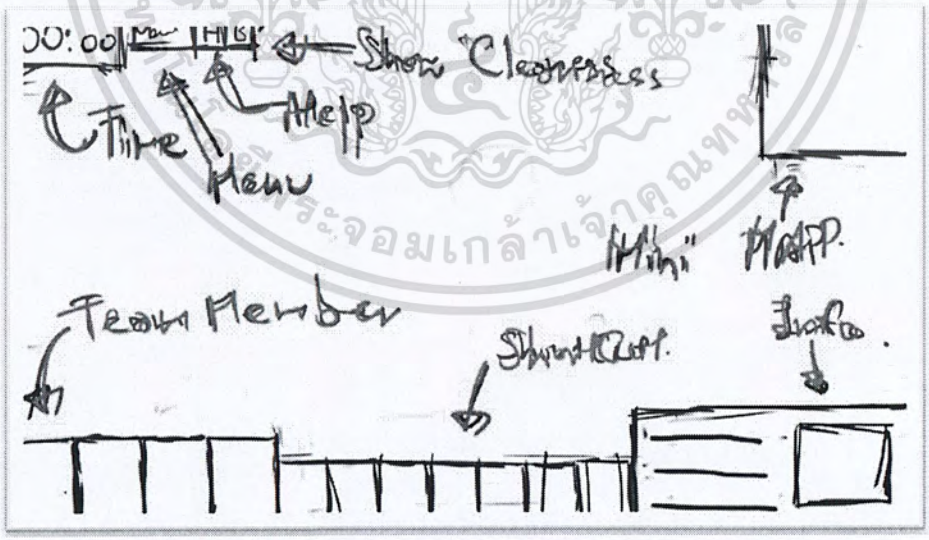
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.10 หน้าจอ Character Selection

3.2.2.2.3 หน้าจอ Game

หน้าจอ Game หรือหน้าจอขณะเข้าเล่นเกม ถูกออกแบบดังรูป 3.11 ให้มีแผนที่ขนาดเล็กอยู่ทางด้านขวาบนของหน้าจอ ส่วนด้านซ้ายบนของจอเป็นเวลาที่เหลืออยู่สำหรับการเล่น ส่วนด้านซ้ายล่างของจอเป็นสถานะต่างๆ ของตัวละครของผู้เล่น ถัดมาเป็นปุ่มสำหรับเรียกใช้ความสามารถพิเศษของตัวละครที่เลือก



รูป 3.11 หน้าจอ Game

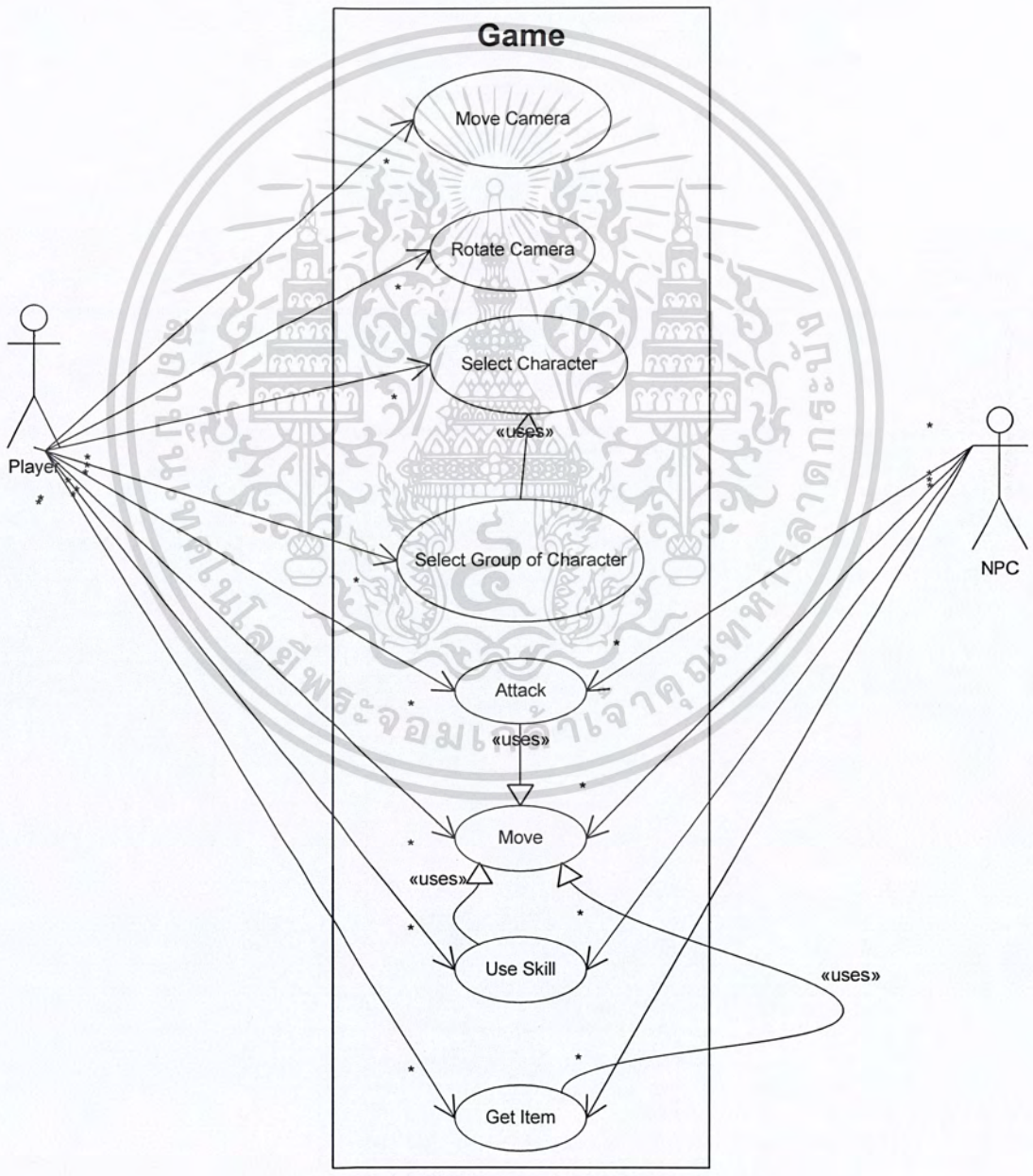
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบบโครงสร้างการทำงานของเกม

การออกแบบโครงสร้างเป็นการนำเอาการออกแบบเบื้องต้นทั้งในภาพรวมและด้านองค์ประกอบศิลป์มาออกแบบโครงสร้างที่สามารถนำไปพัฒนาได้จริง

3.2.3.1 ยูสเคสไดอะแกรม (Use Case Diagram)

จากการออกแบบเกมในหัวข้อ 3.1 สามารถนำมาเขียนสรุปสิ่งที่ผู้เล่นสามารถทำได้ในเกมนี้ด้วยยูสเคสไดอะแกรม (Use Case Diagram) และยูสเคสเดสคริปชัน (Use Case Description) ดังรูป 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19 และ 3.20



รูป 3.12 ยูสเคสไดอะแกรมของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title: Move Camera

Pre-Condition:
- None

Normal Flow of Events:
1. Move the cursor to the edge of the screen.
2. Camera view move corresponding to mouse position.

Termination:
- Move the cursor out of the edge of the screen.

Post-Condition:
- Camera view is changed.

รูป 3.13 ยูสเคสเดสคริปชันสำหรับยูสเคส Move Camera

Title: Rotate Camera

Pre-Condition:
- None

Normal Flow of Events:
1. Hold middle mouse button and drag to the direction.
2. Camera rotated corresponding to the direction.

Termination:
- Release middle mouse button.

Post-Condition:
- Camera angle is changed.

รูป 3.14 ยูสเคสเดสคริปชันสำหรับยูสเคส Rotate Camera

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title: Select Character

Pre-Condition:
-The Character must be alive.

Normal Flow of Events:
1. Choose the Character.
    1a. Player left click on the Character.
    1b. Player presses a hotkey.
2. Draw cursor around the Character.
3. Show info about the Character.
4. Show skill and action hotkey of the Character.
5. Show Character's health and energy.

Termination:
- The Character is eliminated.
- Select other Character.

Post-Condition:
- The Character is active.

รูป 3.15 ยูสเคสเดสคริปชันสำหรับยูสเคส Select Character

Title: Select Group of Characters

Pre-Condition:
- All Characters in the group must be alive.

Normal Flow of Events:
1. Choose the Characters.
    1a. Player left drag mouse cover the group of Characters.
    1b. Player presses a hotkey.
2. Draw cursor around the Character.
3. Show info about the group of Characters.
4. Show skill hotkey.
5. Show Character's health and energy.

Termination:
- The Character is eliminated.
- Select other Character.

Post-Condition:
- The Character is active.

รูป 3.16 ยูสเคสเดสคริปชันสำหรับยูสเคส Select Group of Character

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title: Move

Pre-Condition:
- The Character must be selected.

Normal Flow of Events:
1. Assign destination.
    1a1. Right click on the destination.
    1b1. Left click on the move icon.
    1b2. Left click on the destination.
    1c1. Press a hotkey.
    1c2. Left click on the destination.
2. Path finding.
3. The Character animation changes to Moving.
4. The Character moves to the destination.

Termination:
- The Character is at the destination.
- The Character is eliminated.

Post-Condition:
- The Character position has changed.
- The Character animation changes to Idle.

รูป 3.17 ยูสเคสเดสคริปชันสำหรับยูสเคส Move

Title: Attack

Pre-Condition:
- The Character must be selected.
- The target must be a Character which is alive for single target attack.
- The target must be an area for area attack.

Normal Flow of Events:
1. Assign target.
1a1. Right click on the target.
1b1. Left click on the attack icon.
1b2. Left click on the target.
2. Use Move use-case to move Character into attack range.
3. The Character animation changes to Attacking.
4. The Character resource is decreased.
5. The target clean is decreased.
6. Loop step 4 and 5.

Termination:
- The Character performs other action.
- The Character is eliminated.
- The Character runs out of resource.
- The target is eliminated.

Post-Condition:
- The Character animation changes to Idle.

รูป 3.18 ยูสเคสเดสคริปชันสำหรับยูสเคส Attack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title: Use Skill

Pre-Condition:
- The Character must be selected.
- The target must be a Character which is alive for single target skill.
- The target must be an area for area skill.

Normal Flow of Events:
1. Assign target.
1a1. Right click on the target.
1b1. Left click on the attack icon.
1b2. Left click on the target.
2. Use Move use-case to move Character into skill range.
3. The Character animation changes to Casting.
4. The Character resource is decreased.
5. Set cooldown for the skill.
6. The Character waits for casting time.
8. Perform skill effect.

Termination:
- The Character performs other action.
- The Character is eliminated.
- The Character is interrupted.
- The target is eliminated.
- End of skill effect.

Post-Condition:
- The Character animation changes to Idle.

รูป 3.19 ยูสเคสเดสคริปชันสำหรับยูสเคส Use Skill

Title: Get Item

Pre-Condition:
- The Character must be selected.

Normal Flow of Events:
1. Choose item.
    1a1. Right click on the item.
    1b1. Left click on the move icon.
    1b2. Left click on the item.
    1c1. Press a hotkey.
    1c2. Left click on the item.
2. Use Move use-case to move the Character to the item.
3. The item disappeared.
4. Show effect.

Termination:
- The effect disappeared.

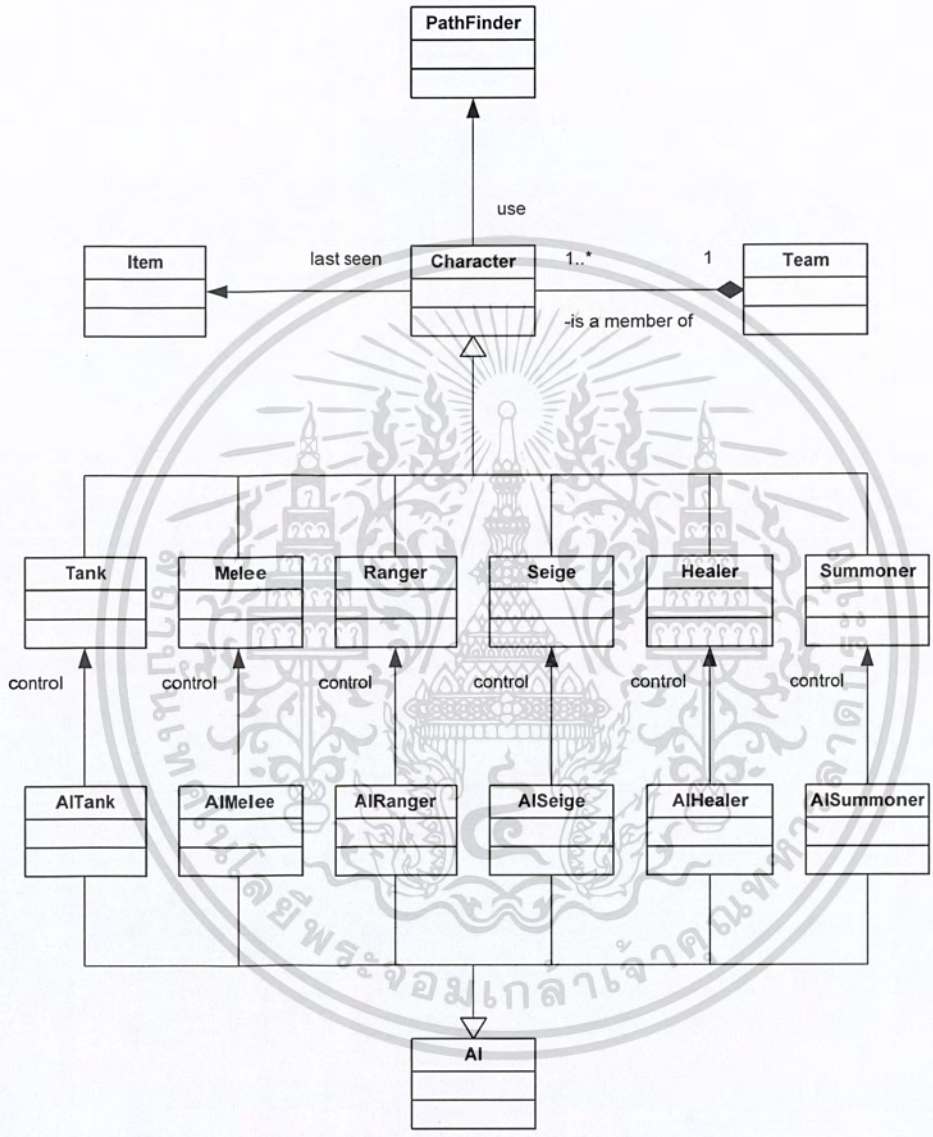
Post-Condition:
- The Character resource is increased.

รูป 3.20 ยูสเคสเดสคริปชันสำหรับยูสเคส Get Item

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.2 คลาสไดอะแกรม (Class Diagram)

คลาสไดอะแกรม (Class Diagram) ต่อไปนี้เป็นการแสดงความสัมพันธ์ระหว่างคลาสที่อยู่ในขอบเขตปัญหาหลักของการพัฒนาเกม (Problem domain)



รูป 3.21 คลาสไดอะแกรมใน Problem domain

จากรูป 3.21 จะสังเกตได้ว่าคลาสของตัวละครแต่ละชนิด ได้แก่ Tank, Melee, Ranger, Seige, Healer และ Summoner ได้ถูกสืบทอดมาจากคลาสเดียวกันคือคลาส Character เนื่องจากคลาสของตัวละครแต่ละชนิดมีสกิลที่แตกต่างกัน รวมถึงมีการกำหนดค่าพารามิเตอร์ที่แตกต่างกัน เช่น ปริมาณความอึดต่อถูกโจมตี, ความแรงของการโจมตี, ปริมาณน้ำที่ใช้ในการโจมตีแต่ละครั้ง เป็นต้น โดยที่ตัวละครแต่ละชนิดยังคงมีการทำงานบางอย่างที่เหมือนกัน เช่น การวิ่ง,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโจมตีโดยพื้นฐาน, การหลบหลีกสิ่งกีดขวาง เป็นต้น ดังนั้นจึงได้มีการสืบทอดตัวละครแต่ละชนิดมาจากคลาสเดียวกัน

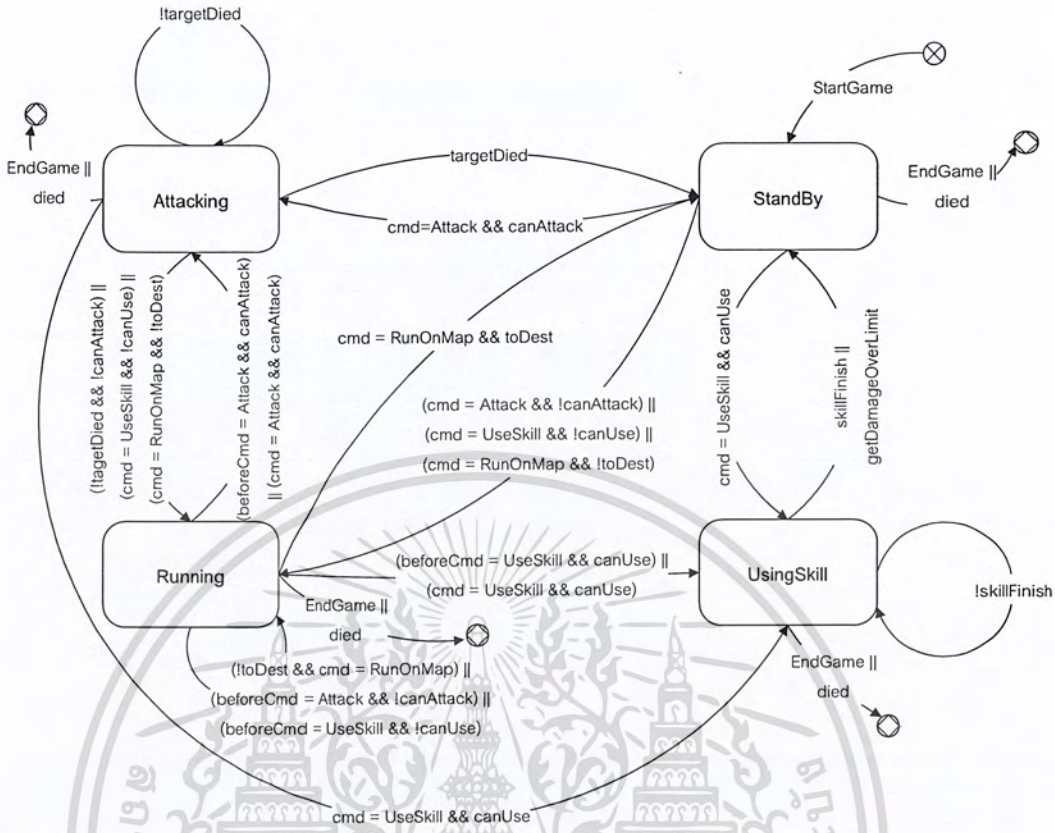
ในขณะที่คลาส Character ซึ่งเป็นคลาสแม่ของคลาสตัวละครชนิดต่างๆนั้น มีความสัมพันธ์แบบการเรียกใช้กับ 2 คลาส ได้แก่ การเรียกใช้งานคลาส PathFinder ซึ่งเป็นคลาสที่ทำงานเกี่ยวกับการค้นหาเส้นทางทั้งหมด ตั้งแต่การกำหนดโหนดจนถึงการหาเส้นทาง โดยคลาส PathFinder จะทำการวางโหนดในตอนเริ่มเกมเพื่อให้ทุกๆตัวละครที่ถูกสร้างขึ้นในเกมมาใช้งาน โดยตัวละครจะทำการร้องขอการค้นหาเส้นทางเมื่อถึงเวลาที่ต้องการจะวิ่งไปที่ใดๆ ซึ่งภายในคลาส PathFinder มีเมธอดหนึ่งเพื่อรองรับการร้องขอดังกล่าวจากตัวละคร และความสัมพันธ์กับอีกคลาสหนึ่งคือคลาส Item ซึ่งเป็นคลาสที่ระบุถึงไอเทมต่างๆซึ่งเป็นประโยชน์แก่ตัวละคร เช่น น้ำ, แป้ง, แบตเตอรี่ เป็นต้น โดยความสัมพันธ์ระหว่าง 2 คลาสดังกล่าวจะเกิดขึ้นเมื่อตัวละครเดินไปพบไอเทมใดๆก็ตาม แล้วทำการเก็บไอเทมนั้น หลังจากนั้นตัวละครจะทำการเพิ่มค่าแอททริบิวต์ต่างๆตามที่กำหนดไว้ในไอเทม นอกจากนั้นแล้ว คลาส Character ยังมีความสัมพันธ์แบบเป็นส่วนหนึ่งของ (Aggregation) กับคลาส Team เพราะว่าตัวละครทุกๆตัวที่ปรากฏอยู่ในเกมต้องเป็นสมาชิกทีมใดทีมหนึ่งเสมอ โดยภายในคลาสทีมมีแอตทริบิวต์สำหรับการระบุค่าสถานะต่างๆที่สำคัญของทีม เช่น คะแนนรวม, ปริมาณเลือดรวม, สมาชิกในทีม เป็นต้น

ในการควบคุมตัวละครแต่ละชนิดภายในเกม แบ่งออกเป็น 2 แบบคือ แบบที่หนึ่งเป็นการควบคุมจากผู้เล่น และแบบที่สองเป็นการควบคุมจากปัญญาประดิษฐ์ โดยใน Class diagram นี้ได้แสดงให้เห็นว่าการใช้ปัญญาประดิษฐ์ควบคุมตัวละคร มีการแบ่งคลาสของปัญญาประดิษฐ์สำหรับรองรับตัวละครทุกๆชนิด ได้แก่ AI Tank, AIMelee, AIRanger, AISeige, AIHealer และ AISummoner เนื่องจากตัวละครแต่ละชนิดมีค่าแอตทริบิวต์ตั้งต้นและความสามารถหรือสกิลที่แตกต่างกัน ทำให้มีการตัดสินใจที่แตกต่างกันด้วย แต่ตัวละครแต่ละชนิดยังคงมีผลจากการตัดสินใจบางอย่างที่ทำงานเหมือนกันอยู่ เช่น การวิ่ง, การหนี, การเข้าโจมตี เป็นต้น ดังนั้นปัญญาประดิษฐ์ที่ควบคุมการทำงานของตัวละครแต่ละชนิดจึงมีการสืบทอดมาจากคลาสเดียวกัน

3.2.3.3 สเตทชาร์ทไดอะแกรม (Statechart Diagram)

เนื่องจากคลาส Character มีการทำงานที่ซับซ้อน จึงมีความจำเป็นที่ต้องมีสเตทชาร์ทไดอะแกรม (Statechart Diagram) ขึ้นมาเพื่ออธิบายการทำงานดังรูป 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.22 Statechart ของ Character

สถานะของ Class Character ประกอบด้วย 4 สถานะหลักคือ

1) StandBy เป็นสถานะเริ่มต้นของคลาส Character หลังจากที่มีการสร้างวัตถุของคลาส Character หรือตัวละครขึ้นมา โดยที่ตัวละครจะกลับเข้ามาในสถานะนี้ทุกครั้งหลังจากทำงานใดๆเสร็จสิ้น และถ้าตัวละครอยู่ในสถานะนี้แล้วได้รับคำสั่งเพื่อให้ไปทำสิ่งต่างๆ โดยที่เป้าหมายห่างออกไป ทำให้ไม่สามารถทำได้ทันที ตัวละครจะเปลี่ยนสถานะเป็น Running เพื่อไปทำตามเป้าหมายดังกล่าว แต่ถ้าเป้าหมายอยู่ในรัศมีที่สามารถทำได้ ก็จะเปลี่ยนสถานะไปทำตามเป้าหมายดังกล่าวทันที
2) Attacking เป็นสถานะในขณะที่ตัวละครกำลังโจมตีฝ่ายตรงข้ามอยู่ โดยจะมีการเปลี่ยนสถานะเมื่อได้รับคำสั่งให้ไปทำสิ่งอื่น แต่ถ้าไม่ได้รับคำสั่งใดๆเข้ามา ตัวละครก็จะทำการโจมตีไปเรื่อยๆจนกระทั่งฝ่ายตรงข้ามพ่ายแพ้ แล้วกลับเข้าสู่สถานะ StandBy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Running เป็นสถานะในขณะที่ตัวละครกำลังวิ่งอยู่ โดยในการวิ่งนั้น อาจมาจากคำสั่งให้ตัวละครวิ่งไปยังตำแหน่งใดๆบนแผนที่ หรืออาจมาจากคำสั่งที่ถูกสั่งให้ไปทำสิ่งใดๆก็ตามที่อยู่เกินขอบเขตรัศมีที่สามารถจะทำได้ ซึ่งต้องวิ่งไปให้ถึงก่อนถึงจะทำได้ ดังนั้นการเปลี่ยนสถานะขณะที่อยู่ในสถานะ Running มีได้ 2 กรณีคือ กลับเข้าสู่สถานะ StandBy ในกรณีที่ถูกสั่งให้วิ่งไปยังตำแหน่งใดๆบนแผนที่ แล้ววิ่งไปถึงแล้ว และอีกกรณีคือการวิ่งไปถึงเป้าหมายที่ต้องการแล้วเปลี่ยนเป็นสถานะที่ต้องการกระทำ ได้แก่ สถานะ Attacking และ UsingSkill

4) UsingSkill ตัวละครจะอยู่ในสถานะนี้เมื่อกำลังใช้สกิลใดๆอยู่ โดยสามารถออกจากสถานะได้มีอยู่ 2 กรณีคือ การใช้สกิลเสร็จสิ้นลง และกรณีที่โดนโจมตีจนไม่สามารถต้านทานได้จากฝ่ายตรงข้าม นอกจากนี้สถานะ UsingSkill ของแต่ละชนิดตัวละครอาจมีแตกต่างนอกเหนือจากนี้บ้างเล็กน้อย แต่ส่วนใหญ่จะเป็นเช่น สเตทชาร์ทไดอะแกรมนี้

3.3 การออกแบบโครงสร้างและขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมที่ใช้ในควบคุมตัวละคร NPC (Non Player Character) ใช้ฟังก์ชันโลจิสติกดังสมการ 2.6 เป็นฟังก์ชันกระตุ้น โครงข่ายประสาทเทียมที่จะถูกนำมาใช้งานได้นั้นต้องเป็นโครงข่ายประสาทเทียมที่ผ่านขั้นตอนการเรียนรู้แบบออฟไลน์ซึ่งใช้วิธี NEAT (NeuroEvolution of Augmenting Topologies) มาก่อน ซึ่ง NEAT จะสร้างโครงสร้างและน้ำหนักที่เชื่อมระหว่างโหนดมาให้อย่างเหมาะสมแล้ว ดังนั้นประสิทธิภาพในการตัดสินใจของโครงข่ายประสาทเทียมจึงขึ้นอยู่กับการออกแบบอินพุทและเอาท์พุทของโครงข่ายประสาทเทียมที่ดี และการออกแบบกระบวนการการเรียนรู้ที่เหมาะสม

3.3.1 การออกแบบอินพุทของโครงข่ายประสาทเทียม

อินพุทของโครงข่ายประสาทเทียมมีหน้าที่สำคัญในการรับสภาพแวดล้อมภายในเกมรอบตัวละครที่ถูกโครงข่ายประสาทเทียมควบคุมอยู่ เพื่อนำไปใช้ในการตัดสินใจเลือกแผนการต่างๆ โดยที่ทุกอินพุทมีขอบเขตอยู่ที่ [0, 1] เท่านั้น บางอินพุทของโครงข่ายประสาทเทียมมีค่าเป็นจำนวนไม่ต่อเนื่อง และมีค่าเปลี่ยนไปตามลักษณะและสถานะของตัวละคร โดยมีค่าตามตาราง 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.3 อินพุทที่เป็นจำนวนไม่ต่อเนื่องของโครงข่ายประสาทเทียม

Value specification			
Name	Character	Value	Unknown value
Type	All	0.1: Tank, 0.2: Melee, 0.3: Ranger, 0.4: Seige, 0.5: Healer, 0.6: Summoner	0
Status	Tank	0: Eliminated, 0.1: Idle, 0.2: Running, 0.3: Attacking, 0.4: Casting, 0.5: BunkerMode	0.1
Status	Melee	0: Eliminated, 0.1: Idle, 0.2: Running, 0.3: Attacking, 0.4: Casting, 0.5: Invulnerable	0.1
Status	Ranger	0: Eliminated, 0.1: Idle in DualMode, 0.2: Running in DualMode, 0.3: Attacking in DualMode, 0.4: Casting in DualMode, 0.5: Idle in SniperMode, 0.6: Running in SniperMode, 0.7: Attacking in SniperMode, 0.8 Casting in SniperMode	0.1
Status	Seige	0: Eliminated, 0.1: Idle, 0.2: Running, 0.3: Attacking, 0.4: Casting	0.1
Status	Healer	0: Eliminated, 0.1: Idle, 0.2: Running, 0.3: Attacking, 0.4: Casting	0.1
Status	Summoner	0: Eliminated, 0.1: Idle, 0.2: Running, 0.3: Attacking, 0.4: Casting	0.1

อินพุทของโครงข่ายประสาทเทียมถูกออกแบ่งออกเป็นสองส่วนใหญ่ๆ คือส่วนที่รับค่ามาจากลักษณะและสถานะของตัวละครที่โครงข่ายประสาทเทียมควบคุมอยู่รวมทั้งตัวละครอื่นๆ ที่อยู่ในทีมเดียวกันมีรายละเอียดตามตาราง 3.4 และอีกส่วนเป็นส่วนที่รับค่ามาจากลักษณะและสถานะของตัวละครฝ่ายศัตรู โดยมีรายละเอียดตามตาราง 3.5

ตาราง 3.4 อินพุทที่มาจากตัวละครในทีมของโครงข่ายประสาทเทียม

		Input node detail		
50	1st enemy: 1st team	Type	Up to this character type	0
51		Clean	current clean / maximum clean	1
52		DistanceX	|this character position x - my position x| / map size along x axis	1
53		DistanceZ	|this character position z - my position z| / map size along z axis	1
54		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
55		Status	Up to this character type	0.1
56		BeingAttacked	0: No, 1: Yes	0
57		BeingCleaned	0: No, 1: Yes	0
58		BeingSlow	0: No, 1: Yes	0
59	2nd enemy: 1st team	Type	Up to this character type	0
60		Clean	current clean / maximum clean	1
61		DistanceX	|this character position x - my position x| / map size along x axis	1
62		DistanceZ	|this character position z - my position z| / map size along z axis	1
63		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
64		Status	Up to this character type	0.1
65		BeingAttacked	0: No, 1: Yes	0
66		BeingCleaned	0: No, 1: Yes	0
67		BeingSlow	0: No, 1: Yes	0
68	3rd enemy: 1st team	Type	Up to this character type	0
69		Clean	current clean / maximum clean	1
70		DistanceX	|this character position x - my position x| / map size along x axis	1
71		DistanceZ	|this character position z - my position z| / map size along z axis	1
72		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
73		Status	Up to this character type	0.1
74		BeingAttacked	0: No, 1: Yes	0
75		BeingCleaned	0: No, 1: Yes	0
76		BeingSlow	0: No, 1: Yes	0
77	4th enemy: 1st team	Type	Up to this character type	0
78		Clean	current clean / maximum clean	1
79		DistanceX	|this character position x - my position x| / map size along x axis	1
80		DistanceZ	|this character position z - my position z| / map size along z axis	1
81		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
82		Status	Up to this character type	0.1
83		BeingAttacked	0: No, 1: Yes	0
84		BeingCleaned	0: No, 1: Yes	0
85		BeingSlow	0: No, 1: Yes	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.5 อินพุทที่มาจากตัวละครทีมอื่นของโครงข่ายประสาทเทียม

Input node detail				
46	1st enemy: 1st team	Type	Up to this character type	0
47		Clean	current clean / maximum clean	1
48		DistanceX	|this character position x - my position x| / map size along x axis	1
49		DistanceZ	|this character position z - my position z| / map size along z axis	1
50		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
51		Status	Up to this character type	0.1
52		BeingAttacked	0: No, 1: Yes	0
53		BeingCleaned	0: No, 1: Yes	0
54		BeingSlow	0: No, 1: Yes	0
55	2nd enemy: 1st team	Type	Up to this character type	0
56		Clean	current clean / maximum clean	1
57		DistanceX	|this character position x - my position x| / map size along x axis	1
58		DistanceZ	|this character position z - my position z| / map size along z axis	1
59		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
60		Status	Up to this character type	0.1
61		BeingAttacked	0: No, 1: Yes	0
62		BeingCleaned	0: No, 1: Yes	0
63		BeingSlow	0: No, 1: Yes	0
64	3rd enemy: 1st team	Type	Up to this character type	0
65		Clean	current clean / maximum clean	1
66		DistanceX	|this character position x - my position x| / map size along x axis	1
67		DistanceZ	|this character position z - my position z| / map size along z axis	1
68		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
69		Status	Up to this character type	0.1
70		BeingAttacked	0: No, 1: Yes	0
71		BeingCleaned	0: No, 1: Yes	0
72		BeingSlow	0: No, 1: Yes	0
73	4th enemy: 1st team	Type	Up to this character type	0
74		Clean	current clean / maximum clean	1
75		DistanceX	|this character position x - my position x| / map size along x axis	1
76		DistanceZ	|this character position z - my position z| / map size along z axis	1
77		Direction	((unit vector(this character position z to my position) dot (this character direction unit vector))+1)/2	0
78		Status	Up to this character type	0.1
79		BeingAttacked	0: No, 1: Yes	0
80		BeingCleaned	0: No, 1: Yes	0
81		BeingSlow	0: No, 1: Yes	0

เมื่อรวมการรับอินพุทจากลักษณะและสถานะของตัวละครทุกตัวแล้วจะมีอินพุทโหนดรวมทั้งสิ้น 157 โหนด

3.3.2 การออกแบบเอาท์พุทของโครงข่ายประสาทเทียม

เอาท์พุทของโครงข่ายประสาทเทียมถูกออกแบบให้เป็นลักษณะนิสัย (Behavior) ของตัวละคร โดยที่เอาท์พุทแต่ละโหนดแทนการเรียกใช้สคริปต์ที่เขียนเอาไว้สำหรับตัวละครประเภทนั้นๆ โดยที่หากเอาท์พุทใดมีค่ามากที่สุด สคริปต์ประจำโหนดนั้นจะถูกเรียกใช้ รายละเอียดของเอาท์พุทของแต่ละตัวละครเป็นไปตามตาราง 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.6 เอาท์พุทของโครงข่ายประสาทเทียมของตัวละครแต่ละประเภท

Output node detail		
All character		
No.	Name	Value
0	RunTo	0 to 1
1	RunAway	0 to 1
2	FindItem	0 to 1
3	Attack	0 to 1
Tank		
4	UseBunker	0 to 1
Melee		
4	UseDash	0 to 1
5	UseLakeofPowder	0 to 1
Ranger		
4	ChangeStance	0 to 1
Seige		
4	UseHareruya	0 to 1
5	UseKamikaze	0 to 1
Healer		
4	UseTowel	0 to 1
5	UseHairDryer	0 to 1
6	UseGiantElectricFan	0 to 1
Summoner		
4	UseMotor-tricycle	0 to 1
5	UseVanzBoyandSkoyzGirl	0 to 1
6	UsePick-upTruck	0 to 1
7	UseFireApparatus	0 to 1

สคริปต์ที่ระบุลักษณะนิสัยของตัวละครนั้นถูกเขียนขึ้นโดยการเก็บข้อมูลจากการเล่นเกมประเภท RTS (Real-Time Strategy) ของผู้เล่นซึ่งเป็นคนจริงๆ เมื่อต้องตัดสินใจทำการกระทำตามชื่อของสคริปต์ที่จะเขียน ในที่นี้จะแสดงรายละเอียดของสคริปต์โดยใช้ซูโดโค้ด โดยสคริปต์ที่เขียนไว้มีรายละเอียดดังนี้

1) RunTo เป็นสคริปต์ที่สั่งให้ตัวละครเดินหาเป้าหมาย มีซูโดโค้ดดังรูป 3.23

```
RunTo
{
LOOP Found any character in vision
        IF That character is in the same team
                IF Distance between that character and itself > 10 m
                        RunTo that character
                        Return
                ELSE
                        Next
        ELSE
                Attack that character
                Return
RunTo a random position
}
```

รูป 3.23 สั่งให้ตัวละครเดินหาเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) RunAway เป็นสคริปต์ที่สั่งให้ตัวละครหลบหนีเอาตัวรอด มีซูโดโค้ดดังรูป 3.24

```
RunAway
{
Keep Enemies into ListOfEnemies
IF ListOfEnemies.Count = 1
        RunTo the opposite direction of  ListOfEnemies[0] 10 m
ELSE IF ListOfEnemies.Count > 1
        DECLARE BestHole = 0
        DECLARE NowHole
        DECLARE SelectedLine1
        DECLARE SelectedLine2
        FOR i = 0 TO ListOfEnemies.Count - 1
                IF i = 0
                        NowAngle = Angle between ListOfEnemies[i]
                                   and ListOfEnemies[ListOfEnemies.Count - 1]
                ELSE IF i != ListOfEnemies.Count - 1
                        NowHole = (Angle between ListOfEnemies[i] and ListOfEnemies[i - 1] ) *
                                  (Average of distance ListOfEnemies[i] and ListOfEnemies[i - 1])
                                   / (Difference of distance from itself to ListOfEnemies[i]
                                   and ListOfEnemies[i - 1])
                IF NowHole > BestHole
                        BestHole = NowHole
                        SelectedLine1 = Line from itself to ListOfEnemies[i]
                        SelectedLine2 = Line from itself to ListOfEnemies[i - 1]
        RunTo middle line between SelectedLine1 and SelectedLine2 10 m
ELSE
        RunTo a random position
}
```

รูป 3.24 สั่งให้ตัวละครหลบหนีเอาตัวรอด

3) FindItem เป็นสคริปต์ที่สั่งให้ตัวละครเดินไปเก็บกล่อง มีซูโดโค้ดดังรูป 3.25

```
vector3 item[10];
int count; // for counting item found

findItem() {
        int i; // for loop
        int nearest = -1;
        if(count != 0) {
                nearest = 0;
                for(i = 0;i < count - 1;i++) {
                        if( checkIfThisItemIsNearerThanBefore(item[nearest] ,item[i]) ) {
                                itemNearest = i;
                        }
                }
        }
        getDestination(item[itemNearest]);
        walkToDestination();
}
```

รูป 3.25 สั่งให้ตัวละครเดินไปเก็บกล่อง

4) Attack เป็นสคริปต์ที่สั่งให้ตัวละครเข้าจู่โจมศัตรู มีซูโดโค้ดดังรูป 3.26

```
Attack
{
FIND the nearest summoner who is casting with in the screen
IF Found
        Attack it
ELSE
        FIND the lowest clean enemy with in attack range + distance(I can move in 1s)
        IF Found
                Attack it
        ELSE
                FIND the lowest clean enemy in the screen
                IF Found
                        Attack it
                ELSE
                        Just stand there
}
```

รูป 3.26 สั่งให้ตัวละครโจมตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) UseBunker เป็นสคริปต์ที่สั่งให้ตัวละคร Tank ใช้ความสามารถ Bunker มีซูโดโค้ดดังรูป 3.27

```
useBunker() {
        if(checkIfDestinationIsEmpty()) {
                placeBunker();
                status = usingBunker;
        }
}

bunker() {
        if our healer or summoner or area exist:
                find the area most comprehensive of all healer and summoner and area
                walk to that area
                useBunker
}
```

รูป 3.27 สั่งให้ตัวละคร Tank ใช้ความสามารถ Bunker

6) UseDash เป็นสคริปต์ที่สั่งให้ตัวละคร Melee ใช้ความสามารถ Dash มีซูโดโค้ดดังรูป 3.28

```
useDash() {
        if(checkIfCollisionInFiveMetre()) {
                status = invulnerable(); // อมตะชั่วขณะ
                if(collisionBuilding()) {
                        gettingDistantToColision() - 0.5 ; //if return point that collision
                        status = nomal;
                } else if (collisionCharacter) {
                        gettingDistantToColision() - 0.5 ; //if return point that collision
                        sendDamageToCharacter();
                        status = attacking;
                }
        } else {getToThePoint();}
}

dash() {
        set temp = -1
        if no enemy in area:
                useDash
                return
        loop all enemy in area:
                if any enemy is close than 5 m and have cn <= 60:
                        useDash
                        return
                else if cn <= 60:
                        set temp = enemy id
        if temp = -1:
                if enemy healer is exist in area:
                        get healer id
        if distant between this character and enemy in enemy id => 15 OR <= 5:
                useDash
        else
                walk to enemy target until distant is <= 5
}
```

รูป 3.28 สั่งให้ตัวละคร Melee ใช้ความสามารถ Dash

7) UseLakeOfPowder เป็นสคริปต์ที่สั่งให้ตัวละคร Melee ใช้ความสามารถ Lake of powder มีซูโดโค้ดดังรูป 3.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UseLakeofPowder
{
IF no enemy in the screen
        Just use it
ELSE
        Caculate distance from me to each enemy in the screen
        Store in a parallel queue (distance and enemy position)
        Sort the queue from min to max distance
        Store the nearest enemy position
        FOR queue's size
                pop queue
                Change direction to this enemy position
                Detect all character in skill's effect area
                IF number of enemy in the area > number of team member in the area
                        Use it
        IF hasn't used it yet
                Use it with stored enemy position
}
```

รูป 3.29 สั่งให้ตัวละคร Melee ใช้ความสามารถ Lake of powder

8) ChangeStance เป็นเอาท์พุทที่ไม่เป็นสคริปต์ เพราะเป็นการใช้ความสามารถตรงๆ ไม่มีการเลือกเป้าหมายหรือทิศทาง

9) UseHareruya เป็นสคริปต์ที่สั่งให้ตัวละคร Area ใช้ความสามารถ Hareruya มีซูโดโค้ดดังรูป 3.30

```
UseHareruya
{
        Create area of raycasting, assign its radius is radius of UseHareruya's area
        DECLARE BestArea = null
        DECLARE NowArea
        DECLARE BestAmount
        LOOP All over vision
                NowArea = Use area of raycasting
                IF Amount of enemies in NowArea > BestAmount
                        BestArea = NowArea
                        BestAmount = Amount of enemies in NowArea
        IF BestArea != null
                UseHareruya at BestArea
        ELSE
                UseHareruya at 10 m from itself
}
```

รูป 3.30 สั่งให้ตัวละคร Area ใช้ความสามารถ Hareruya

10) UseKamikaze เป็นเอาท์พุทที่ไม่เป็นสคริปต์ เพราะเป็นการใช้ความสามารถตรงๆ ไม่มีการเลือกเป้าหมายหรือทิศทาง

11) UseTowel เป็นสคริปต์ที่สั่งให้ตัวละคร Healer ใช้ความสามารถ Towel มีซูโดโค้ดดังรูป 3.31

```
UseTowel
{
FIND the lowest clean team member (include self)
     with in skill range + distance(I can move in 1s)
     and clean + 0.8*Towel healing power < maxclean
If Found
        Use Towel to that character
ELSE
        RANDOM Pick Team member in vision (include self)
        Use Towel to picked character
}
```

รูป 3.31 สั่งให้ตัวละคร Healer ใช้ความสามารถ Towel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12) UseHairDryer เป็นสคริปต์ที่สั่งให้ตัวละคร Healer ใช้ความสามารถ Hairdryer มีซูโดโค้ดดังรูป 3.32

```
UseHairDryer
{
FIND the lowest clean team member (include self)
     with in skill range + distance(I can move in 1s)
     and clean + 0.5*HairDryer healing power < maxclean
If Found
        Use Towel to that character
ELSE
        RANDOM Pick Team member in vision (include self)
        Use Towel to picked character
}
```

รูป 3.32 สั่งให้ตัวละคร Healer ใช้ความสามารถ Hairdryer

13) UseGiantElectricFan เป็นสคริปต์ที่สั่งให้ตัวละคร Healer ใช้ความสามารถ Giant electric fan มีซูโดโค้ดดังรูป 3.33

```
UseGiantElectricFan() {
        findTheWayBehindPointOfDeploying();
        if(!collision()) {
                walkToTheTarget();
                placeAnElectricFan();
        }
}

electricFan() {
        find area that most conprehensive of all team member exist
        runTothePoint
        useElectricFan
}
```

รูป 3.33 สั่งให้ตัวละคร Healer ใช้ความสามารถ Giant electric fan

14) UseMotor-tricycle เป็นสคริปต์ที่สั่งให้ตัวละคร Summoner ใช้ความสามารถ Motor-tricycle มีซูโดโค้ดดังรูป 3.34

```
UseMotor-tricycle
{
        Create area of raycasting, assign its radius is radius of UseMotor-tricycle's area
        DECLARE BestArea = null
        DECLARE NowArea
        DECLARE BestAmount
        LOOP All over vision
                NowArea = Use area of raycasting
                IF Amount of enemies in NowArea > BestAmount
                        BestArea = NowArea
                        BestAmount = Amount of enemies in NowArea
        IF BestArea != null
                UseMotor-tricycle at BestArea
        ELSE
                UseMotor-tricycle at 10 m from itself
}
```

รูป 3.34 สั่งให้ตัวละคร Summoner ใช้ความสามารถ Motor-tricycle

15) UseVanzBoyandSkoyzGirl เป็นสคริปต์ที่สั่งให้ตัวละคร Summoner ใช้ความสามารถ Vanz boy 'N' Skoyz girl มีซูโดโค้ดดังรูป 3.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UseVanzBoyandSkoyzGirl
{
        Create area of raycasting, assign its radius is radius of UseVanzBoyandSkoyzGirl's area
        DECLARE BestArea = null
        DECLARE NowArea
        DECLARE BestAmount
        LOOP All over vision
                NowArea = Use area of raycasting
                IF Amount of enemies in NowArea > BestAmount
                        BestArea = NowArea
                        BestAmount = Amount of enemies in NowArea
        IF BestArea != null
                UseVanzBoyandSkoyzGirl at BestArea
        ELSE
                UseVanzBoyandSkoyzGirl at 10 m from itself
}
```

รูป 3.35 สั่งให้ตัวละคร Summoner ใช้ความสามารถ Vanz boy 'N' Skoyz girl

16) UsePick-upTruck เป็นสคริปต์ที่สั่งให้ตัวละคร Summoner ใช้ความสามารถ Pick-up truck มีซูโดโค้ดดังรูป 3.36

```
UsePick-upTruck
{
        Create area of raycasting, assign its radius is radius of UsePick-upTruck's area
        DECLARE BestArea = null
        DECLARE NowArea
        DECLARE BestAmount
        LOOP All over vision
                NowArea = Use area of raycasting
                IF Amount of enemies in NowArea > BestAmount
                        BestArea = NowArea
                        BestAmount = Amount of enemies in NowArea
        IF BestArea != null
                UsePick-upTruck at BestArea
        ELSE
                UsePick-upTruck at 10 m from itself
}
```

รูป 3.36 สั่งให้ตัวละคร Summoner ใช้ความสามารถ Pick-up truck

17) UseFireApparatus เป็นสคริปต์ที่สั่งให้ตัวละคร Summoner ใช้ความสามารถ Fire apparatus มีซูโดโค้ดดังรูป 3.37

```
UseFireApparatus
{
        Create area of raycasting, assign its radius is radius of UseFireApparatus's area
        DECLARE BestArea = null
        DECLARE NowArea
        DECLARE BestAmount
        LOOP All over vision
                NowArea = Use area of raycasting
                IF Amount of enemies in NowArea > BestAmount
                        BestArea = NowArea
                        BestAmount = Amount of enemies in NowArea
        IF BestArea != null
                UseFireApparatus at BestArea
        ELSE
                UseFireApparatus at 10 m from itself
}
```

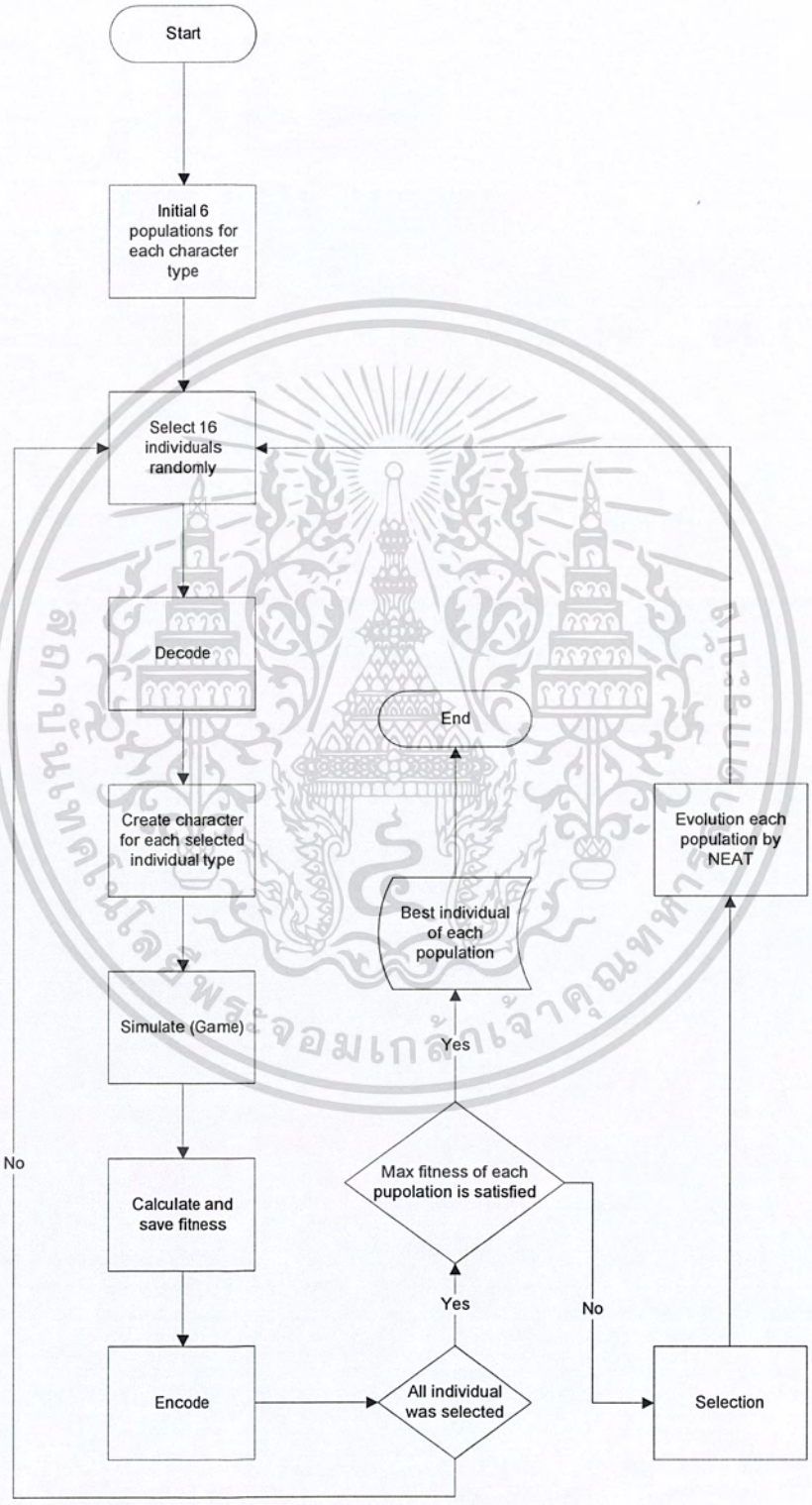
รูป 3.37 สั่งให้ตัวละคร Summoner ใช้ความสามารถ Fire apparatus

3.3.3 การออกแบบขั้นตอนการเรียนรู้

เนื่องจากโครงข่ายประสาทเทียมที่ใช้ควบคุม NPC (Non Player Character) มีโครงสร้างแตกต่างกันออกไปตามประเภทของตัวละคร จึงต้องมีการออกแบบขั้นตอนการเรียนรู้ขึ้น เพื่อให้โครงข่ายประสาทเทียมได้ต่อสู้กันเองจนครบ ขั้นตอนการเรียนรู้ที่ใช้เป็นไปตามรูป 3.38 โดยมีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างประชากรของโครงข่ายประสาทเทียมของตัวละครแต่ละประเภทขึ้นมา แล้วสุ่มเลือกออกมาใช้ในเกมเพื่อวัดประสิทธิภาพ



รูป 3.38 ขั้นตอนการเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสุ่มเลือกจะเกิดขึ้นไปเรื่อยๆ จนกว่าโครงข่ายประสาทเทียมทุกตัวจะถูกใช้ในเกมจนครบ แล้วจึงเข้าสู่ขั้นตอนการประเมินประสิทธิภาพแล้วกลับสู่ NEAT (NeuroEvolution of Augmenting Topologies) เพื่อหาประชากรรุ่นใหม่ของโครงข่ายประสาทเทียมของตัวละครแต่ละประเภทออกมาแล้วเข้าสู่กระบวนการสุ่มเลือกต่อไป กระบวนการนี้จะดำเนินซ้ำไปเรื่อยๆ จนกว่าจะได้ประสิทธิภาพที่น่าพอใจ

ในการหาค่าความเหมาะสมสำหรับตัวละครแต่ละประเภทนั้นจะแตกต่างกันออกไปตามหน้าที่ของตัวละครแต่ละประเภท ค่าความเหมาะสมของตัวละครต่างๆ ดังนี้

```
Tank: damageDealt-damageGot+ lifeTime/6 + damageUmbrellaGot*8 + collideUmbrella*50
Healer: damageDealt - damageGot + lifeTime/6 + healingDamage*4
Melee: damageDealt - damageGot + damageOfDash + lifeTime/6 + enemyInStickyFloor - memberInStickyFloor
Ranger: damageDealt - damageGot + lifeTime/6
Area: damageDealt + hareruyaDamage*1.5 + kamikazeDamage - damageGot + lifeTime/6 - friendlyDamage*2
Summoner: damageDealt + skillDamage*2 - damageGot + lifeTime/6 - friendlyDamage
```

รูป 3.39 สมการค่าความเหมาะสมของตัวละครชนิดต่างๆ

3.4 การประยุกต์ใช้ปัญญาประดิษฐ์กับเกม

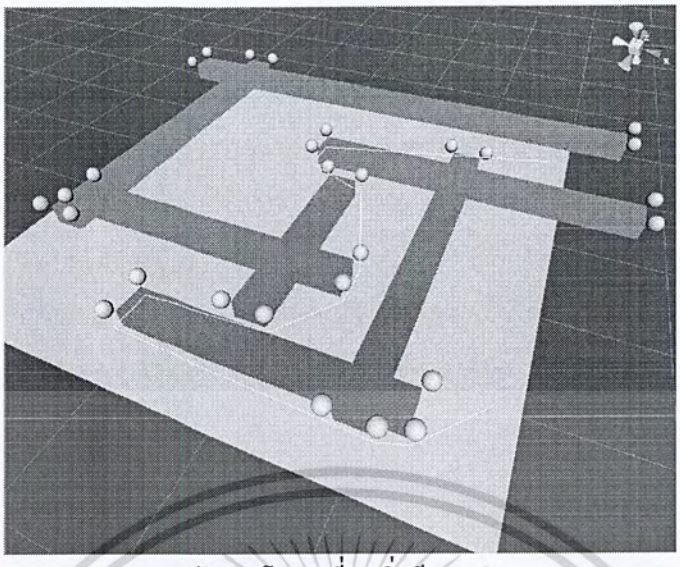
3.4.1 การค้นหาเส้นทางและการหลบหลีกสิ่งกีดขวาง

ภายในโครงงานนี้ได้จัดการเรื่องการวิ่งไปยังจุดต่างๆ โดยแบ่งออกเป็น 2 ส่วน ดังนี้

3.4.1.1 การประยุกต์ใช้ A* Search algorithm

การจะใช้วิธีการค้นหาแบบเอสตาร์ได้นั้น ก่อนอื่นต้องสร้างโหนดที่จะใช้ในการค้นหาขึ้นมาก่อน ด้วยสมมุติฐานที่ว่าเส้นทางที่สั้นที่สุดคือเส้นทางที่วิ่งระหว่างมุมของสิ่งกีดขวาง ต้นทางไปยังมุมของสิ่งกีดขวางปลายทาง จึงสร้างแต่ละโหนดไว้ที่มุมของสิ่งกีดขวางทุกอย่างดังรูป 3.39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.40 โหนดที่มุมสิ่งกีดขวาง

เมื่อได้โหนดมาแล้วก็ทำการหาว่าโหนดข้างเคียง โดยโหนดข้างเคียงคือโหนดที่สามารถลากเส้นตรงเชื่อมกันได้โดยไม่ผ่านสิ่งกีดขวางใดๆ และเก็บความยาวของเส้นตรงไว้เป็นค่าใช้จ่ายในการเดินทางระหว่างโหนดทั้งสอง

เมื่อได้โหนดและค่าใช้จ่ายในการเดินทางระหว่างแต่ละโหนดเรียบร้อยแล้ว จึงเริ่มวิธีการค้นหาแบบเอสตาร์โดยใช้มินฮีป (Min Heap) เป็นโครงสร้างข้อมูลที่ทำหน้าที่เป็นคิวลำดับ

ฟังก์ชัน $g(x)$ เป็นค่าใช้จ่ายที่ใช้ในการเดินระหว่างโหนด ซึ่งก็คือความยาวของเส้นตรงที่เก็บเอาไว้ ส่วนฟังก์ชัน $h(x)$ เป็นระยะห่างระหว่างโหนดในแนวแกน X รวมกับระยะห่างระหว่างโหนดในแนวแกน Z ซึ่งเป็นเหมือนค่าประมาณคุณภาพของโหนดนั้นๆ

เพื่อให้การค้นหามีความเร็วมากขึ้น จึงมีลงทะเบียนแต่ละโหนดเมื่อไปถึงโหนดนั้นเป็นครั้งแรก เพื่อช่วยตัดการค้นหาที่ไปยังโหนดเดิมออกไป

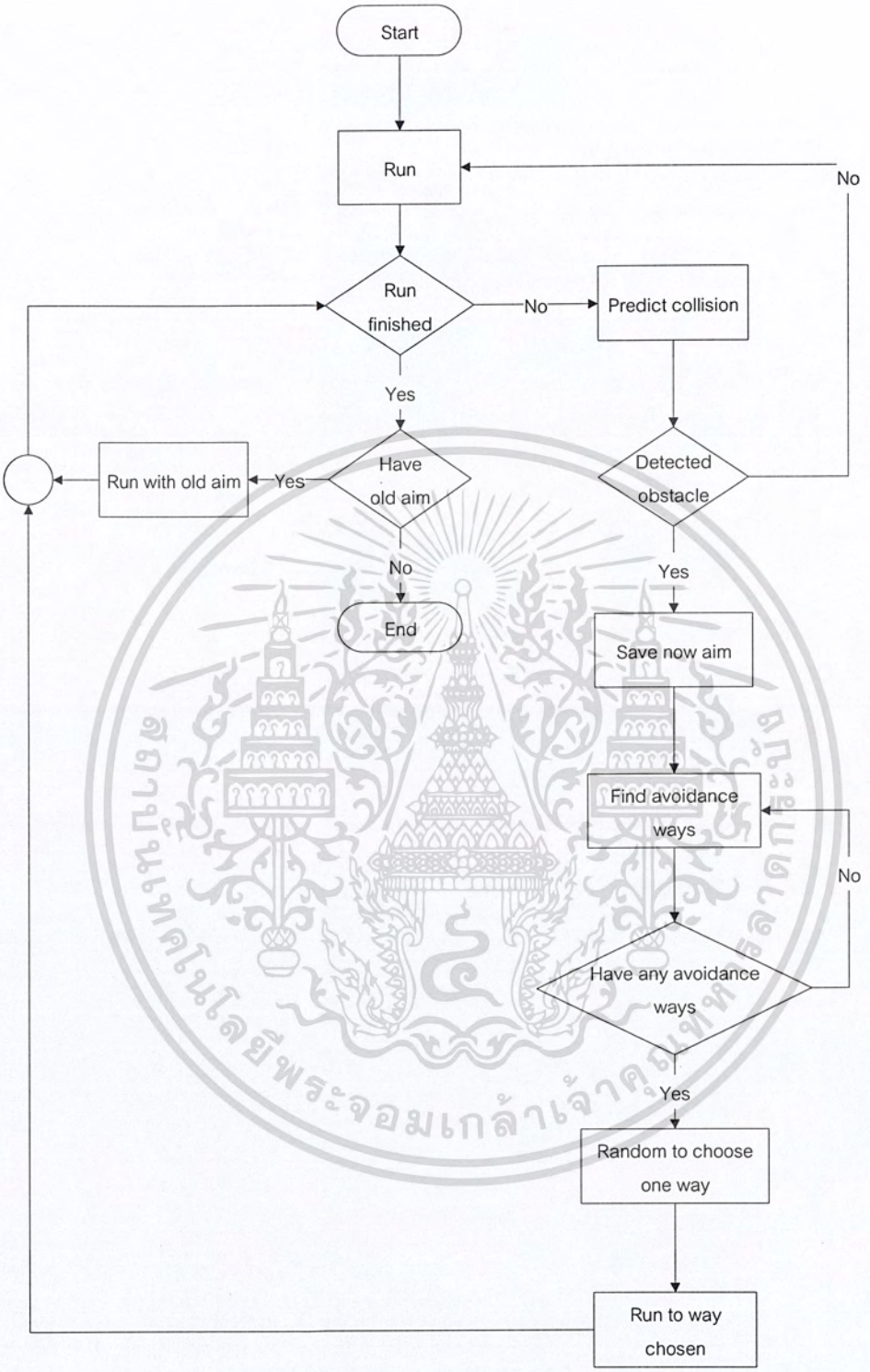
3.4.1.2 วิธีการหลบหลีกสิ่งกีดขวาง

ปัญหาการเดินชนสิ่งกีดขวางจะเกิดขึ้นได้ ก็ต่อเมื่อสิ่งกีดขวางดังกล่าวมีตัวตรวจสอบการชน (Collider) ซึ่งเป็นสิ่งที่ป้องกันการซ้อนทับจากวัตถุอื่น โดยภายในโครงงานนี้มีการจำแนกสิ่งกีดขวางออกเป็น 2 ประเภท ได้แก่ สิ่งกีดขวางแบบสแตติก (Static obstacle) คือวัตถุที่ไม่มีการเคลื่อนที่และถูกสร้างขึ้นตั้งแต่ตอนเริ่มเกม เช่น ตึก, ต้นไม้, ม้านั่ง เป็นต้น และสิ่งกีดขวางแบบไดนามิก (Dynamic obstacle) คือสิ่งกีดขวางที่สามารถเคลื่อนที่ได้หรือสิ่งกีดขวางที่ถูกสร้างในระหว่างที่เกมกำลังดำเนินอยู่ ซึ่งสิ่งกีดขวางแบบไดนามิกที่เคลื่อนที่ได้ภายในโครงงานนี้มีเพียงตัวละครเท่านั้น ซึ่งถ้าไม่มีการป้องกันการเดินชนแล้ว จะทำให้ตัวละครไม่สามารถเดินผ่านสิ่งกีดขวางไปได้ ดังนั้นวิธีการหลบหลีกสิ่งกีดขวางจึงเป็นสิ่งที่สำคัญอีกสิ่งหนึ่งในการพัฒนาเกม โดยเฉพาะอย่างยิ่งในเกมแนววางแผนดังเช่นในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การป้องกันการชนที่เกิดจากสิ่งกีดขวางแบบสแตติกนั้น ได้เกิดขึ้นภายในขั้นตอนการหาเส้นทางเรียบร้อยแล้ว เนื่องจากในการสร้างโหนดแต่ละโหนดในตอนเริ่มเกมเพื่อรองรับเส้นทางการเดินของตัวละครนั้น เป็นการบังคับให้ตัวละครเดินในทิศทางตามโหนดที่กำหนดไว้แล้ว ดังนั้นในหัวข้อนี้จะไม่ขอพูดถึงวิธีการหลบหลีกสิ่งกีดขวางสำหรับสิ่งกีดขวางแบบสแตติก ส่วนวิธีการหลบหลีกสิ่งกีดขวางสำหรับสิ่งกีดขวางแบบไดนามิกนั้นสามารถแก้ได้โดยการใช้วิธีทำนายการชนล่วงหน้า แล้วหาเส้นเบนออกไปในเส้นทางอื่น โดยขั้นตอนวิธีดังกล่าวได้ถูกนำมาใส่ให้แก่ตัวละครทุกๆตัวซึ่งมีการเคลื่อนที่ โดยมีการเรียกใช้ทุกครั้งที่มีการเคลื่อนที่ ซึ่งภายในโครงงานฉบับนี้ได้มีวิธีการหลบหลีกสิ่งกีดขวางตาม Flow chart ดังรูป 3.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

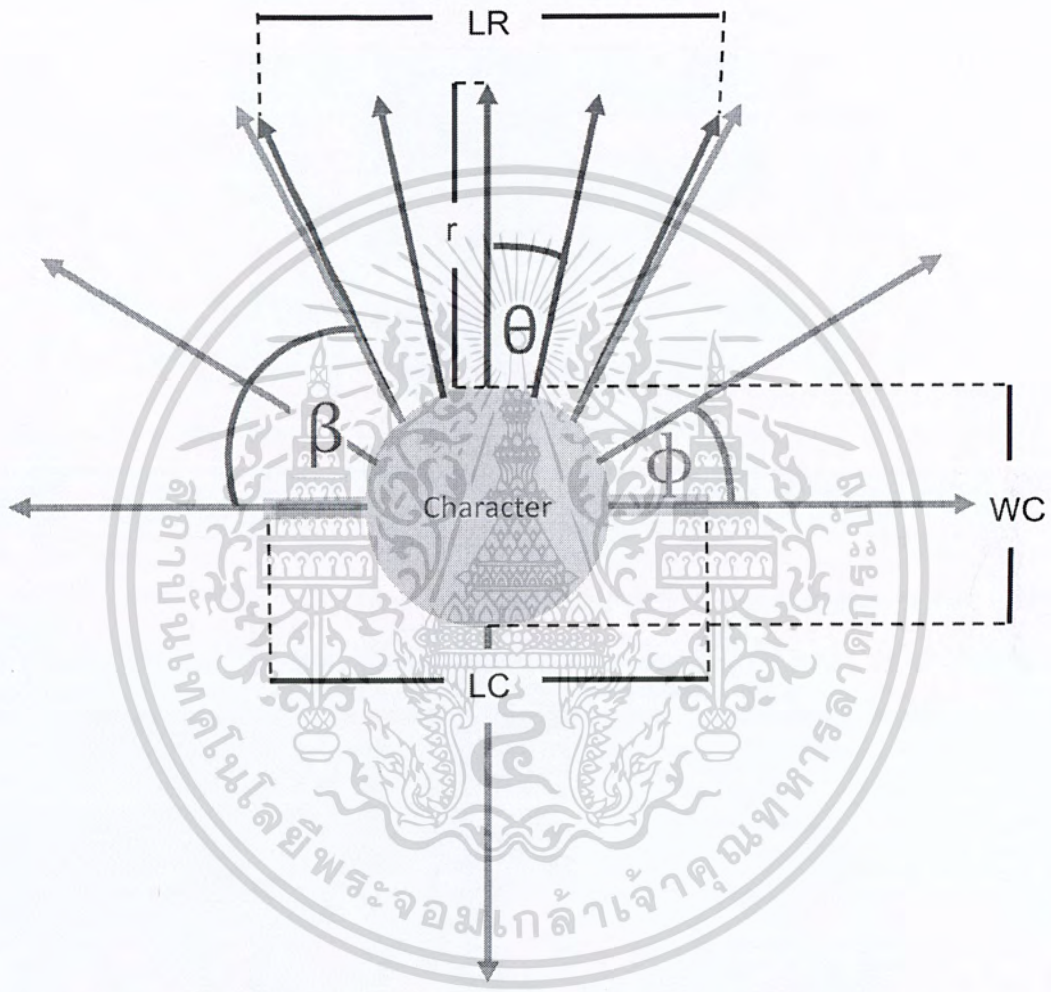


รูป 3.41 Flowchart แสดงการทำงานของวิธีการหลบหลีก

การทำงานของ Flow chart นี้จะเกิดขึ้นทุกครั้งที่มีการวิ่งหรือการเคลื่อนที่ของตัวละคร ดังนั้นถ้าไม่มีการเคลื่อนที่หรือในกรณีที่ตัวละครบรรลุเป้าหมายแล้ว จะไม่มีการใช้งาน Flow chart นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานภายใน Flow chart นี้ส่วนหลักๆ คือการทำนายการชน (Predict collision) ซึ่งสามารถทำได้โดยทำการยิงรังสี (Raycast) ไปข้างหน้าเพื่อตรวจสอบว่ามีสิ่งกีดขวางอยู่ข้างหน้าหรือไม่ โดยในโครงงานฉบับนี้ได้ทำการยิงรังสีออกไปจากตัวละครตามเส้นสีแดงในรูป 3.41 ซึ่งเป็นมุมมองจากด้านบน



รูป3.42 ทิศทางที่ใช้ในการตรวจสอบการชนและเส้นทางการหลบหลีกของตัวละคร

จากรูปด้านบน ในการยิงรังสีเพื่อทำนายการชนไปยังสิ่งกีดขวางนั้น ภายในโครงงานนี้ได้ทำการยิงออกเป็นส่วนโค้งของวงกลม โดยความยาวของส่วนโค้ง (LR) นั้น ควรมีค่าใกล้เคียงกับความยาวของตัวละคร (LC) เพราะถ้ามากเกินไป จะทำให้ถูกทำนายว่ามีการชนบ่อยมากขึ้น ซึ่งในบางครั้งอาจจะเดินผ่านไปได้โดยไม่มีการชนเกิดขึ้น แต่ถ้าน้อยเกินไป ก็จะทำให้ถูกทำนายว่ามีการชนน้อยเกินไป ซึ่งในบางครั้งอาจเดินไปแล้วติดสิ่งกีดขวางได้ นอกจากนี้มุม θ ที่อยู่ระหว่างรังสีที่ยิงออกไปในการตรวจจับการชนและความยาวรัศมี r ของเส้นรังสี ต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์กับความกว้างของตัวละคร (WC) โดยไม่สนใจในเรื่องจำนวนของเส้นรังสีที่ถูกยิงออกไป แต่ต้องเป็นไปตามสมการด้านล่างนี้

$$2r * tan(\theta/2) \leq WC \tag{3.1}$$

ถ้ามีการตรวจพบสิ่งกีดขวางแล้วขั้นตอนต่อไปคือการเก็บเป้าหมายของการวิ่ง ที่ต้องเก็บเพราะภายในโครงงานนี้ เมื่อตัวละครพบสิ่งกีดขวางแล้วตัวละครจะเปลี่ยนเป้าหมายเป็นการวิ่งเพื่อหลบหลีกสิ่งกีดขวางแทน ดังนั้นจึงต้องมีการเก็บเป้าหมายเดิมเอาไว้ เพื่อจะได้กลับมาทำหลังจากการหลบหลีกสิ่งกีดขวางผ่านไปได้แล้ว

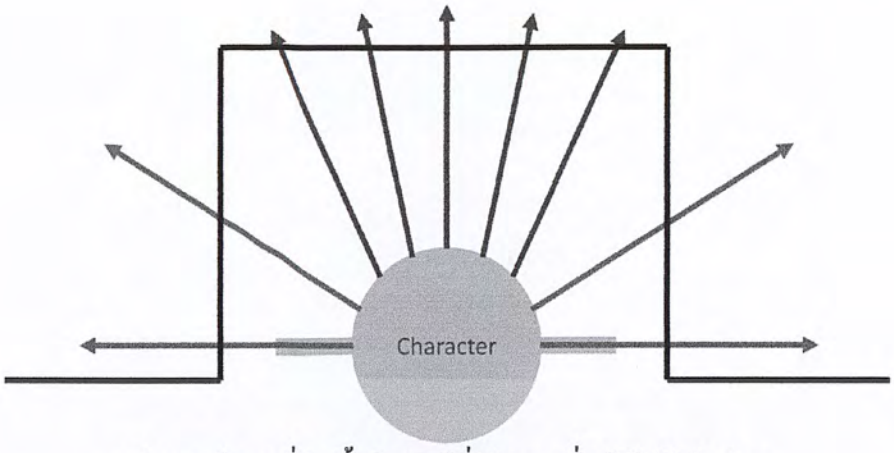
ขั้นตอนต่อไปคือการหาเส้นทางการหลบหลีก ภายในโครงงานนี้ได้กำหนดเส้นทางการหลบหลีกไว้ตามเส้นสีเขียว ซึ่งแบ่งเป็น 3 ส่วนคือซ้าย, ขวา และหลัง โดยที่ไม่มีการกำหนดเส้นที่ตรงไปข้างหน้าให้เป็นเส้นทางการหลบหลีก เนื่องจากการชนส่วนใหญ่ที่เกิดขึ้นจะการชนที่เกิดขึ้นที่ด้านหน้าของตัวละครอยู่แล้ว ทำให้ไม่มีความจะเป็นที่จะต้องกำหนดเส้นทางด้านหน้าเป็นหนึ่งเส้นทางการหลบหลีก

มุม β ที่อยู่ระหว่างเส้นรังสีที่ใช้ในการหลบหลีกการชนเส้นแรกซึ่งอยู่ในแนวระนาบกับเส้นสุดท้ายซึ่งอยู่ติดกับเส้นตรวจจับการชนนั้น ต้องเป็นไปตามสมการด้านล่างนี้

$$\beta \leq \pi/2 - (n-1) * tan(\theta/2) \tag{3.2}$$

จากสมการค่าต้น ค่า n คือจำนวนเส้นรังสีที่ใช้ในการตรวจจับการชน ซึ่งความหมายในสมการนี้หมายถึงขอบเขตของเส้นทางการหลบหลีกจะไม่เข้าไปซ้อนทับกับขอบเขตของรังสีการตรวจจับการชน โดยที่มุม ϕ ซึ่งเป็นมุมระหว่างเส้นทางการหลบหลีกที่อยู่ฝั่งซ้ายและขวา อาจมีค่าเท่าใดก็ได้ และไม่สนใจในเรื่องของจำนวนของเส้นทางการหลบหลีกทีอยู่ระหว่างมุม β เนื่องจากทุกๆเส้นทางที่อยู่ระหว่างมุมนี้ สามารถใช้ในการหลบหลีกได้หมด ดังนั้นความกว้างของมุม ϕ จึงมีค่าใดๆก็ได้ โดยภายในโครงงานนี้กำหนดให้มุม ϕ มีค่าเป็น 30 องศา และในเส้นทางหลบหลีกที่อยู่ด้านหลังของตัวละครจะถูกใช้ในกรณีที่ตัวละครวิ่งเข้าไปติดในช่องซึ่งมีขนาดใกล้เคียงกับความยาวของตัวละครดังรูป 3.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.43 ปัญหาที่เกิดขึ้นในกรณีที่ตัวละครวิ่งเข้าไปติดในช่อง

ซึ่งถ้าไม่มีเส้นทางการหลบหลีกที่เตรียมไว้สำหรับการกลับหลังออกมา ซึ่งถ้าไม่มีเส้นทางการหลบหลีกที่เป็นการกลับหลังหันออกมา ตัวละครก็จะไม่สามารถออกจากช่องเช่นนี้ได้ โดยการใช้งานเส้นทางที่กลับหลังหันนั้น จะใช้ในกรณีเดียวเมื่อรังสีตรวจจับการชนมีการตรวจจับการชนได้ทุกเส้น

และก่อนที่ตัวละครจะเลือกเส้นทางการหลบหลีกนั้น จะต้องมีการตรวจสอบเพื่อให้แน่ใจอีกทีว่าเส้นทางการหลบหลีกทุกเส้นทางนั้นสามารถใช้ได้จริง ซึ่งทำได้โดยการยิงรังสีตรวจจับการชนออกไปเป็นระยะทางเท่ากับรังสีเส้นสีแดงที่ใช้ในการตรวจจับการชน โดยยิงออกไปยังทุกเส้นทางที่ได้กำหนดให้เป็นเส้นทางที่ใช้ในการหลบหลีก ซึ่งถ้ามีเส้นทางหลบหลีกใดๆที่มีการชนเกิดขึ้น เส้นทางนั้นก็จะถูกตัดออกจากการเลือกใช้ และถ้าไม่มีเส้นทางใดสามารถใช้ได้เลย ตัวละครก็จะทำการรอ โดยในขณะที่รออยู่ก็จะทำการยิงรังสีตรวจจับการชนออกไปในเส้นทางที่ใช้ในการหลบหลีก เพื่อดูว่าสามารถไปได้หรือยัง ซึ่งการทำงานในลักษณะนี้อาจเกิดขึ้นได้ในกรณี เช่น มีตัวละครตัวอื่นมายืนบังอยู่ จึงต้องรอให้ตัวละครตัวนั้นวิ่งออกไปจากเส้นทาง เป็นต้น

ในกรณีที่มีเส้นทางการหลบหลีกให้เลือกหลายเส้นทางนั้น ตัวละครจะทำการสุ่มเลือกเส้นทางดังกล่าว และหลังจากทำการเลือกเส้นทางแล้ว ต่อไปคือการทำการยิงรังสีตรวจจับการชนอีกครั้ง โดยยิงออกไปเป็นระยะทางเท่ากับระยะทางสูงสุดที่ใช้ในการวิ่งหลบหลีก เพื่อดูว่าถ้าวิ่งไปแล้วจะชนอีกหรือไม่ เพื่อเป็นการป้องกันปัญหาการชนแบบลูกโซ่ในระดับหนึ่ง ซึ่งถ้าตรวจไม่พบการชน ก็จะทำการวิ่งออกไปเป็นระยะทางเท่ากับระยะทางสูงสุดที่กำหนดไว้ แต่ถ้ามีการชนเกิดขึ้นก็จะทำการวิ่งเป็นระยะทางตามสมการด้านล่างนี้

$$l = DH - r \tag{3.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ l คือระยะทางที่ใช้ในการวิ่ง และ DH คือระยะห่างจากตัวละครถึงจุดที่เกิดการชน ในกรณีที่สามารถผ่านสิ่งกีดขวางไปได้แล้ว ตัวละครก็จะกลับไปวิ่งเพื่อให้บรรลุเป้าหมายเดิม และเนื่องจากภายในเกมนี้มีเคลื่อนไหวแบบเวลาจริง (Real time) ซึ่งอาจทำให้การทำนายการชนอาจคลาดเคลื่อนได้ จึงต้องมีการป้องกันการชนจะเกิดขึ้นอยู่ตลอดเวลาที่ทำการวิ่ง โดยการทำตามขั้นตอนทั้งหมดนี้จนกว่าจะบรรลุเป้าหมายที่แท้จริงหรือเป้าหมายหลักของการวิ่งตามที่ตัวละครได้รับมอบหมาย ณ ตอนนั้น

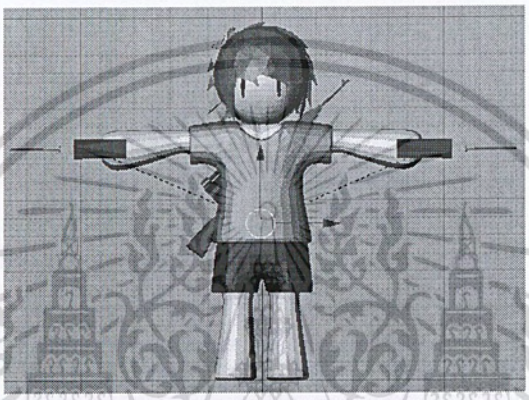
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 โมเดล

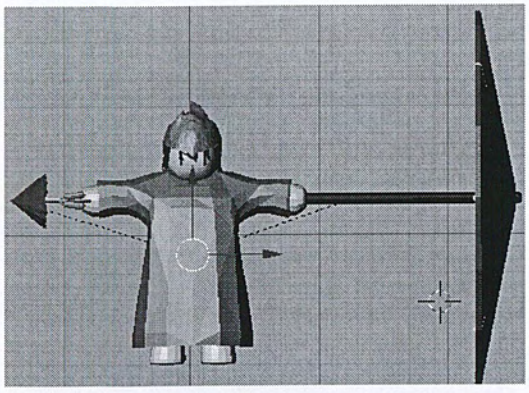
ในการสร้างโมเดลนั้น ได้ใช้ Blender ในการสร้างโมเดลทั้งหมด 6 ตัว โดยมีรูปร่างดังรูป 4.1, 4.2, 4.3, 4.4, 4.5 และ 4.6



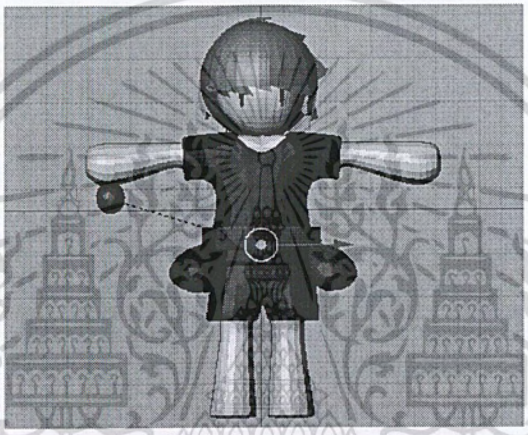
รูป 4.1 โมเดลต้นแบบ Ranger

รูป 4.2 โมเดลต้นแบบของ Power Smearer

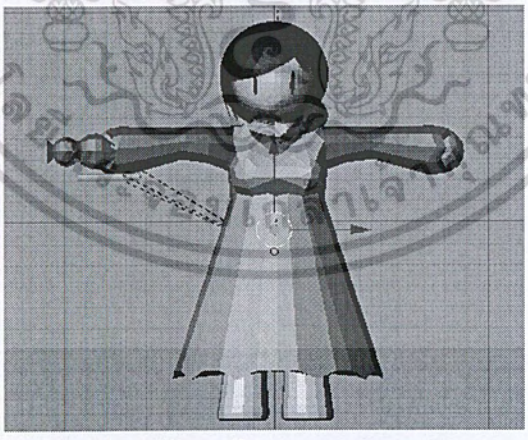
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.3 โมเดลต้นแบบของ Tank

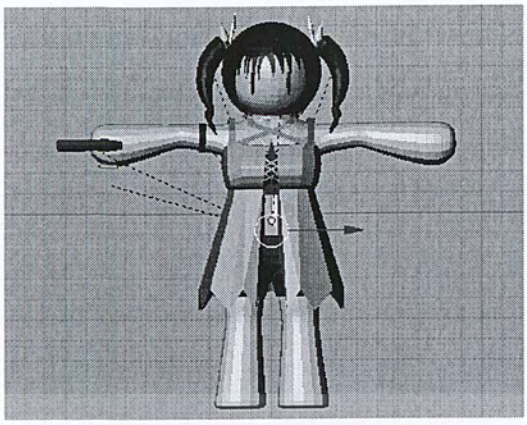


รูป 4.4 โมเดลต้นแบบของ Area



รูป 4.5 โมเดลต้นแบบของ Healer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

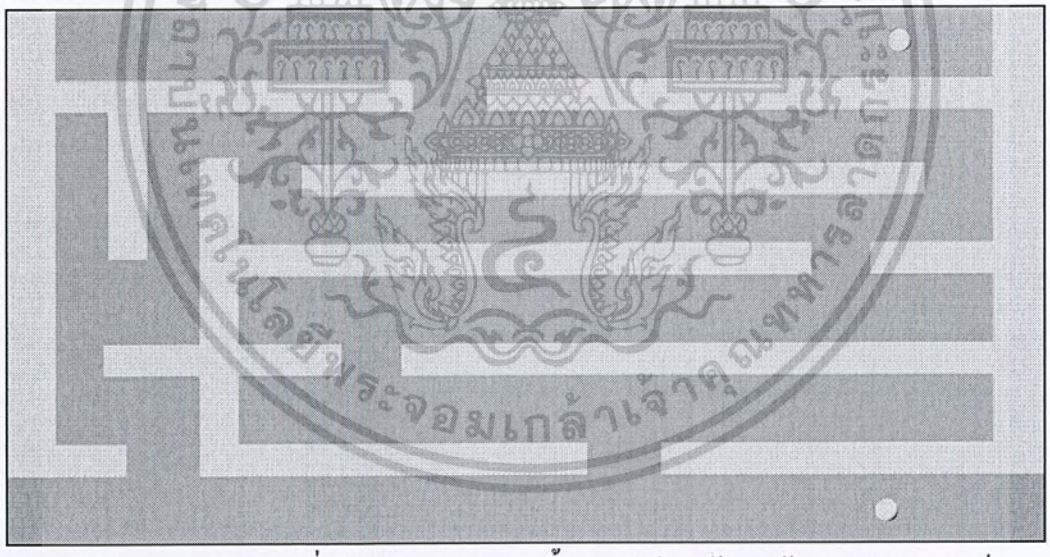


รูป 4.6 โมเดลต้นแบบของ Summoner

4.2 การหาเส้นทางและเดินหลบหลีกสิ่งกีดขวาง

4.2.1 การใช้ขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์

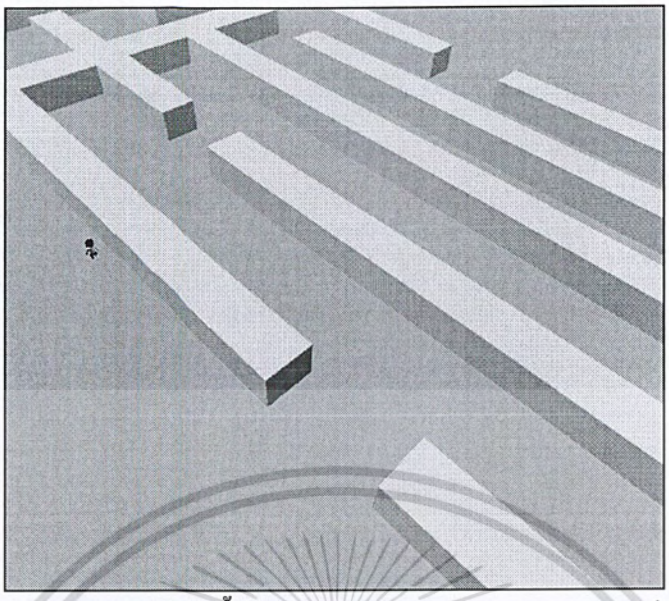
ในการทดลองซึ่งนำเอาการประยุกต์ใช้ขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์มาใช้นั้น ได้มีการสร้างเขาวงกตจำลองขึ้นมาดังรูป 4.7



รูป 4.7 เขาวงกตจำลองที่ใช้สำหรับการทดลองขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์

จากรูปตำแหน่งวงกลมด้านล่างกำหนดให้เป็นจุดเริ่มต้นและตำแหน่งวงกลมด้านบนเป็นจุดปลายทางสำหรับการวิ่งของตัวละคร รูป 4.8 เป็นรูปขณะที่กำลังทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.8 ขณะทดลองขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์

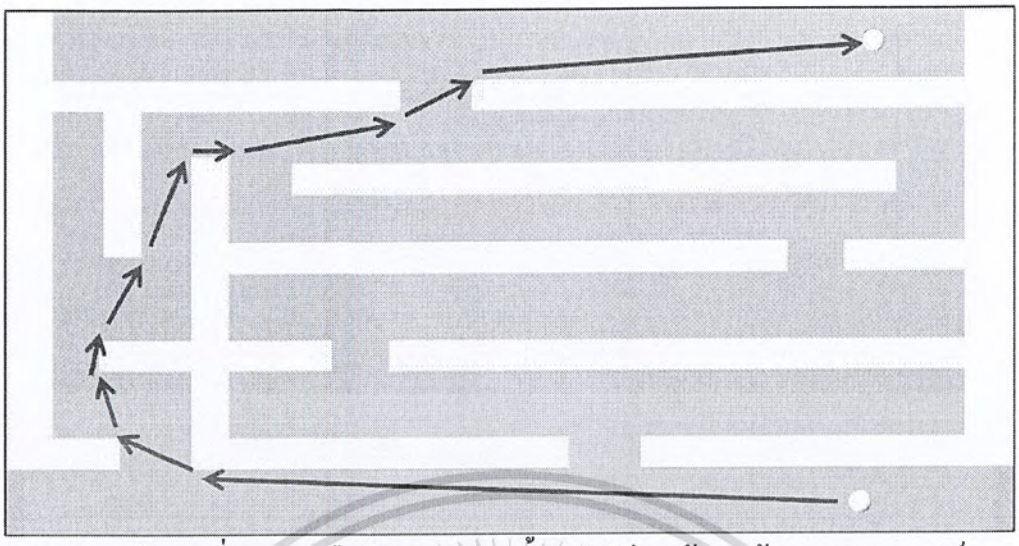
หลังจากได้ทำการทดลองเป็นจำนวน 10 รอบ พบว่าเวลาที่ใช้ในการวิ่งไปถึงจุดหมายมีค่าใกล้เคียงกันทุกครั้ง ซึ่งผลการทดลองเป็นดังตาราง 4.1

ตาราง 4.1 เวลาที่ใช้ในการวิ่งแต่ละรอบ

ครั้งที่	เวลาที่ใช้
1	15.87154
2	15.86288
3	15.80703
4	15.76604
5	15.80917
6	15.7887
7	15.87617
8	15.87377
9	15.91688
10	15.83574
ค่าเฉลี่ย	15.840792

โดยที่ตัวละครได้ทำการเลือกเส้นทางเดิมตลอด โดยเส้นทางที่ตัวละครเลือกใช้คือเส้นทางที่สั้นที่สุดและเป็นดังรูป 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.9 เส้นทางที่ตัวละครเลือกในการทดลองขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์

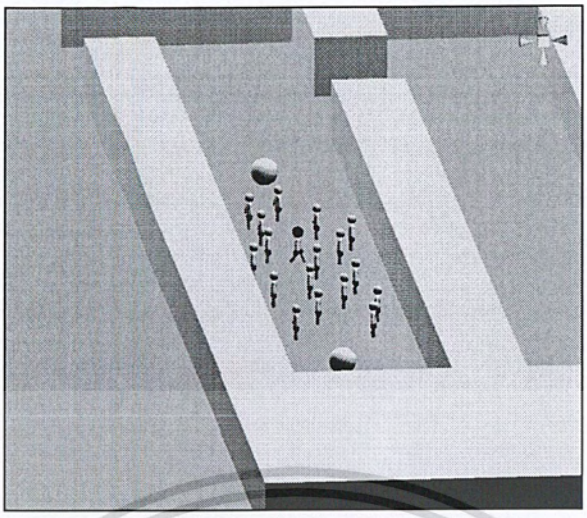
4.2.2 การทดลองการเดินหลบหลีก

เนื่องจากการเดินหลบหลีกเป็นวิธีการที่ใช้ในการหลบหลีกตัวละครด้วยกันโดยเฉพาะ ดังนั้นการทดลองจึงมาการจำลองสถานการณ์ที่มีตัวละครวางขวางทางอยู่ดังรูป 4.8

รูป 4.10 สิ่งกีดขวางสำหรับการทดลองการวิ่งหลบหลีก

โดยกำหนดให้ตำแหน่งวงกลมด้านซ้ายมือเป็นจุดเริ่มต้น และตำแหน่งวงกลมด้านขวามือเป็นจุดปลายทาง โดยรูป 4.11 เป็นรูปขณะทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.11 ขณะทดลองการเดินหลบหลีก

หลังจากทำการทดลองทั้งหมด 10 ครั้ง พบว่าตัวละครสามารถผ่านไปได้ทุกครั้ง โดยมีผลการทดลองเป็นดังตาราง 4.2

ตาราง 4.2 เวลาที่ใช้ในการวิ่งหลบหลีกสิ่งกีดขวางแต่ละรอบ

ครั้งที่	เวลาที่ใช้
1	4.940535
2	6.712934
3	8.36388
4	9.448165
5	10.22434
6	5.328392
7	6.763091
8	7.025027
9	7.545333
10	5.530702
ค่าเฉลี่ย	7.1882399

และยังพบอีกว่าในการทดลองแต่ละครั้งมีการใช้เวลาที่แตกต่างกัน ซึ่งเกิดจากการสุ่มเลือกเส้นทาง

4.3 การใช้ NEAT แก้ปัญหาตรรกศาสตร์ XOR

ปัญหาตรรกศาสตร์เป็นปัญหาพื้นฐานของการเรียนรู้แบบมีผู้สอน (Supervised Learning) ซึ่งโครงข่ายประสาทเทียมจะต้องตอบคำถามตรรกศาสตร์ XOR ได้ถูกต้องตามตาราง 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.3 คำตอบของตรรกศาสตร์ XOR

Input1	Input2	Output
0	0	0
0	1	1
1	0	1
1	1	0

ถ้าเราใช้การเรียนรู้แบบมีผู้สอน เราต้องเปรียบเทียบระหว่างคำตอบที่โครงข่ายประสาทเทียมตอบกับคำตอบที่ถูกต้องเพื่อให้โครงข่ายประสาทเทียมปรับแต่งค่าน้ำหนักของเส้นเชื่อมต่อระหว่างโหนดตามความผิดพลาดของคำตอบที่ตอบออกไป

แต่ในการทดลองนี้เราใช้ NEAT (NeuroEvolution of Augmenting Topologies) ซึ่งเป็นการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) โดยใช้ค่าความเหมาะสมเป็น 4 ลบด้วยผลรวมความผิดพลาดของการตอบคำถามทั้งชุด

จำนวนอินพุทโหนดสำหรับการทดลองนี้คือ 2 โหนด เอาท์พุท 1 โหนด และไบอัสที่เชื่อกับเอาท์พุทอีก 1 โหนดส่วนฟังก์ชันกระตุ้นใช้สมการ 4.1

$$f(x) = 1/(1 + e^{-4.9x}) \tag{4.1}$$

ตัวแปรต่างๆ ที่ใช้ในการทดลองนี้ส่วนใหญ่เป็นตัวแปรที่คล้ายกับตัวแปรของขั้นตอนวิธีเชิงพันธุกรรม โดยมีรายละเอียดตามตาราง 4.4

ตาราง 4.4 ตัวแปรที่ใช้ในการทดลองแก้ปัญหาตรรกศาสตร์ XOR

Parameter	Value
Mutate add connection probality	0.5
Mutate add node probality	0.03
Mutate connection weights probality	0.5
Number of input	2
Number of output	1
Number of population	150
Offspring asexual probality	0.8
Offspring sexual probality	0.2

เมื่อทำการทดลอง NEAT จะวิวัฒน์น้ำหนักของการเชื่อมต่อระหว่างโหนดและโครงสร้าง ทำให้เมื่อการเรียนรู้จบลง โครงข่ายประสาทเทียมที่เป็นผลลัพธ์จะมีจำนวนโหนดในชั้นซ่อนตัวเพิ่มขึ้น โดยทำการทดลองจนกว่าจะได้โครงข่ายประสาทเทียมที่มีค่าความเหมาะสมมากกว่า 3.9 จึงจะหยุดการทดลอง แล้วทำซ้ำจนครบ 10 ครั้งจึงได้ผลการทดลองดังตาราง 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

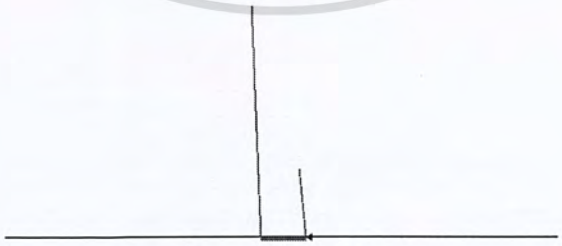
ตาราง 4.5 ผลการทดลองการแก้ปัญหาตรรกศาสตร์ XOR

	Generation	Hidden	Connection	time
Experiment 1	44	2	9	0.7233953
Experiment 2	203	3	17	3.200596
Experiment 3	61	4	27	0.9678112
Experiment 4	74	2	8	1.189951
Experiment 5	46	2	7	0.7539536
Experiment 6	118	4	33	1.863556
Experiment 7	92	4	29	1.474046
Experiment 8	54	2	8	0.8681533
Experiment 9	64	4	30	1.023344
Experiment 10	66	3	19	1.067759
Average	82.2	3	18.7	1.31325654

จากผลการทดลองแสดงให้เห็นว่า NEAT สามารถแก้ปัญหาตรรกศาสตร์ XOR ได้จริง โดยใช้จำนวนรุ่นเฉลี่ยที่ 82.2 รุ่น ใช้เวลาเฉลี่ย 1.31325654 วินาที แต่ยังประสบปัญหาเรื่องการเกินของจำนวนโหนดในชั้นซ่อนตัวและจำนวนการเชื่อมต่อ เนื่องจากโดยปกติแล้วหากใช้การเรียนรู้แบบมีผู้สอน จะใช้โหนดในชั้นซ่อนตัวเพียง 2 โหนดก็เพียงพอแล้ว อย่างไรก็ตาม จำนวนโหนดในตารางนั้นเป็นของโครงข่ายประสาทเทียมที่มีค่าความเหมาะสมสูงสุดซึ่งโดยส่วนใหญ่เรามักพบโครงข่ายประสาทเทียมที่ตอบคำถามได้ดีรองลงมาในขอบเขตที่ยอมรับได้และมีโหนดในชั้นซ่อนตัวเพียง 2 โหนดเสมอและมีจำนวนการเชื่อมต่อที่ยอมรับได้ ปัญหานี้จึงไม่ใช่ปัญหาใหญ่

4.4 การใช้ NEAT แก้ปัญหา DPV (Double Pole Balancing with Velocities)

ปัญหา DPV เป็นปัญหาที่ใช้วัดประสิทธิภาพของการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) โดยเป็นปัญหาซึ่งปัญญาประดิษฐ์ต้องออกแรงกระทำต่อรถเข็นที่มีไม้วางตั้งอยู่สองแท่งดังรูป 4.12 เพื่อพยายามไม่ให้ไม้ทั้งสองนั้นตกลงมา



รูป 4.12 ปัญหา DPV

ปัญหานี้มีตัวแปรที่ต้องตั้งค่ามากมาย ทั้งตัวแปรที่เกี่ยวข้องกับสภาพแวดล้อมในการทดลองและตัวแปรที่เกี่ยวข้องกับกระบวนการ NEAT ซึ่งตัวแปรต่างๆ มีรายละเอียดตามตาราง 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.6 ตัวแปรที่ใช้ในการทดลองแก้ปัญหา DPV

Parameter	Value
1st pole angle start	1°
1st pole length	1 m
1st pole mass	0.1 kg
2nd pole length	0.1 m
2nd pole mass	0.01 kg
Cart mass	1 kg
Compatibility threshold	1.6
Gravity	-9.8
Max force magnitude	10 N
Max time steps	100000
Mutate add connection probality	0.5
Mutate add node probality	0
Mutate connection weights probality	0.5
Number of input	6
Number of output	1
Number of population	1000
Offspring asexual probality	0.8
Offspring sexual probality	0.2
Pole angle threshold	36°
Second per step	0.01 s
Species count max	50
Track length	4.8 meters

โครงสร้างของโครงข่ายประสาทเทียมมีอินพุท 6 โหนด เอาท์พุท 1 โหนด และไบอัส 1 โหนด โดยแต่ละโหนดมีรายละเอียดตามตาราง 4.7

ตาราง 4.7 โหนดของโครงข่ายประสาทเทียมที่ใช้ในการทดลองแก้ปัญหา DPV

Node	Name	Value (+-1.0)
Input1	Cart position (meters)	(x/half of track length)-1
Input2	Cart velocity (m/s)	(v/0.75)-1
Input3	1st Pole angle (radians)	(r/Pole angle threshold)
Input4	1st Pole angular velocity (radians/sec)	rs
Input5	2nd Pole angle (radians)	(r/Pole angle threshold)
Input6	2nd Pole angular velocity (radians/sec)	rs
Output1	Force on cart	f

ในส่วนของฟังก์ชันกระตุ้นที่ใช้ในโครงข่ายประสาทเทียมในการทดลองนี้จะใช้สมการ Steepened Sigmoid ดังสมการ 4.1

หลังจากใช้ NEAT ในการสอนโครงข่ายประสาทเทียมจนกว่าจะสามารถออกแรงกระทำต่อรถเข็นทั้งหมด 100000 ครั้ง (Max time steps) โดยที่ไม้ทั้งสองยังทำมุมไม่เลยขอบเขตที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมด 10 ครั้ง ได้ผลการทดลองดังตาราง 4.8 โดยค่าความเหมาะสมที่ใช้ในการทดลองนี้คือ ตัวแปร Max time steps ลบออกด้วยจำนวนครั้งที่ออกแรงกระทำต่อรถเข็น

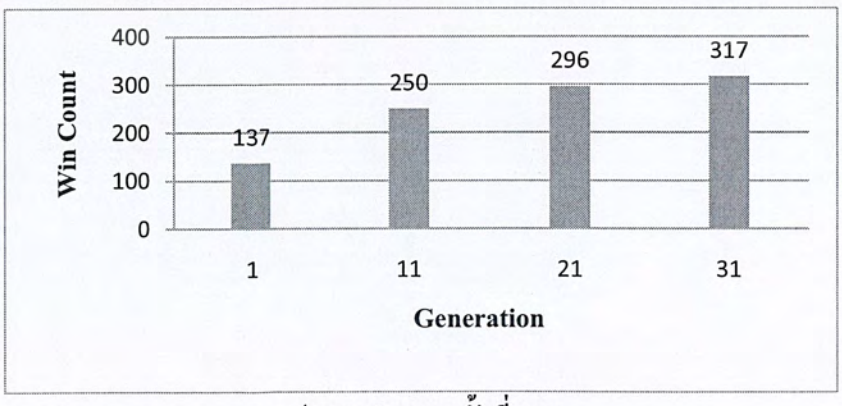
ตาราง 4.8 ผลการทดลองแก้ปัญหา DPV

	Generation	Time
Experiment 1	6	0.7240917
Experiment 2	11	1.767776
Experiment 3	11	1.85943
Experiment 4	16	2.861363
Experiment 5	8	1.31432
Experiment 6	11	1.789641
Experiment 7	10	1.891891
Experiment 8	8	1.112762
Experiment 9	10	1.40277
Experiment 10	5	0.5152565
Average	9.6	1.52393012

จากผลการทดลองแสดงให้เห็นว่า NEAT สามารถใช้ในการสอนโครงข่ายประสาทเทียมให้แก้ปัญหา DPV ได้จริง โดยใช้จำนวนรุ่นของประชากรเฉลี่ย 9.6 รุ่นและใช้เวลาในการสอนโดยเฉลี่ย 1.52393012 วินาที

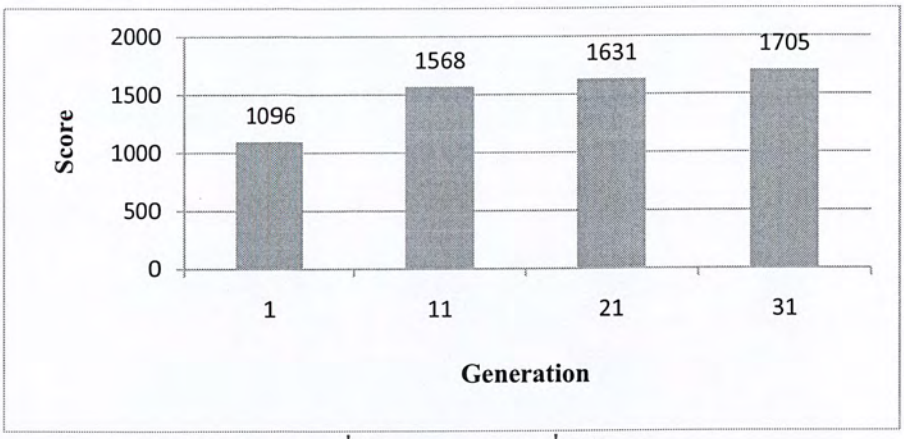
4.5 การใช้ NEAT สร้าง ANN

หลังจากทำการประยุกต์ใช้ NEAT ตามที่ได้ออกแบบไว้และได้ทำการทดลองฝึกฝน AI โดยการปล่อยให้เล่นกันเองไปเรื่อยๆ จนได้ทั้งหมด 31 รุ่น โดยมีประชากร 1000 ตัวในแต่ละรุ่น มีอัตราการครอสโอเวอร์ 0.2 อัตรามิวเทชันค่าน้ำหนัก 0.5 อัตรามิวเทชันเพิ่มการเชื่อมต่อ 0.5 อัตรามิวเทชันเพิ่มโหนด 0.03 และค่าความห่างของต้นตระกูล 1000 จากนั้นสุ่มประชากรจากรุ่นที่ 1, 11, 21, และ 31 มาสู้กัน 1000 ครั้ง ได้ผลดังกราฟต่างๆ ต่อไปนี้

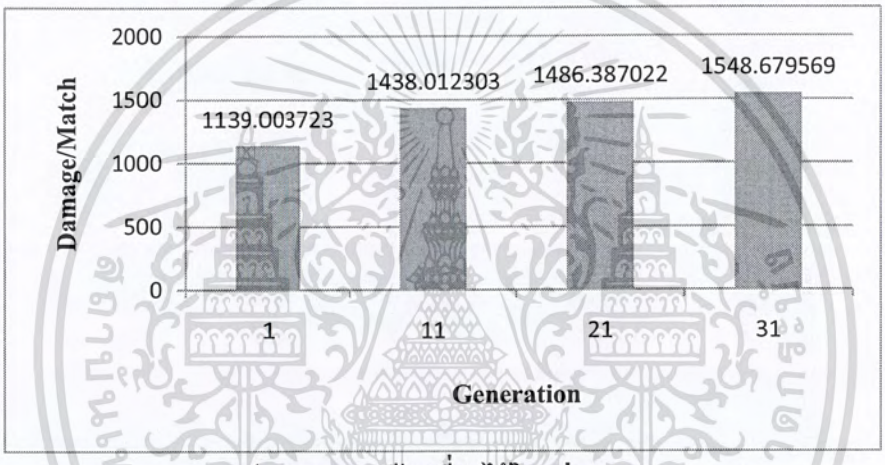


รูป 4.13 จำนวนครั้งที่ชนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

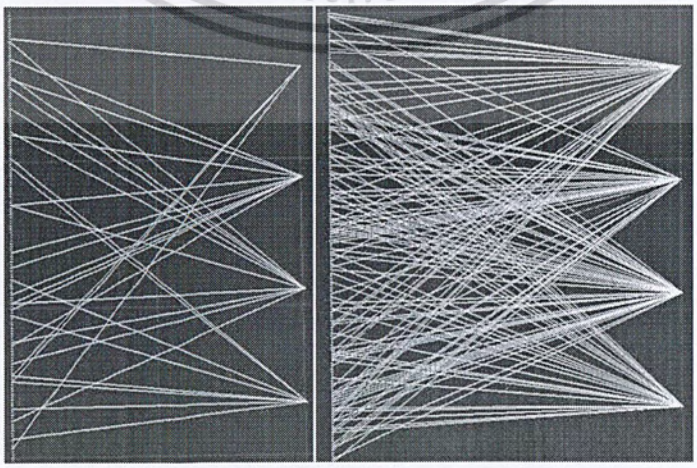


รูป 4.14 คะแนนที่ได้ (คิดจากอันดับที่ได้ในแต่ละรอบ)



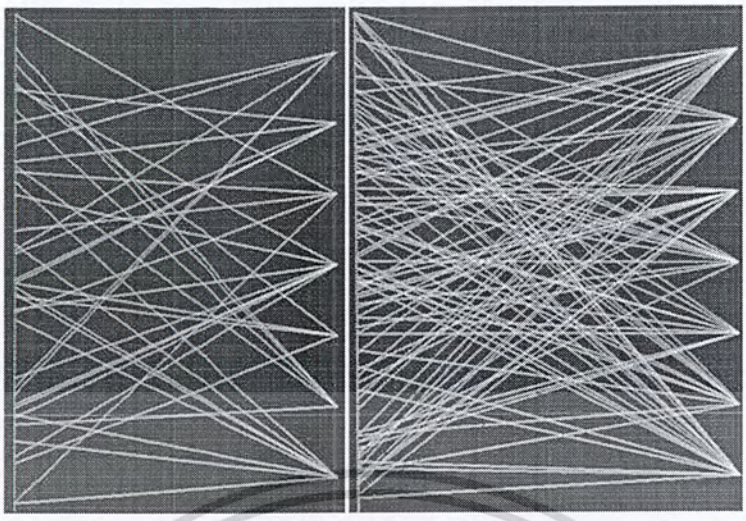
รูป 4.15 ความเปียกที่ทำได้ในแต่ละรอบ

รูป 4.16 และ 4.17 แสดงการเปรียบเทียบโครงสร้าง Neural Network ของ Tank และ Summonerตามลำดับ ระหว่าง Generation ที่ 1 และ Generation ที่ 31



รูป 4.16 โครงสร้าง ANN ของ Tank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.17 โครงสร้าง ANN ของ Summoner

4.6 การสร้างเกมด้วย Unity3D

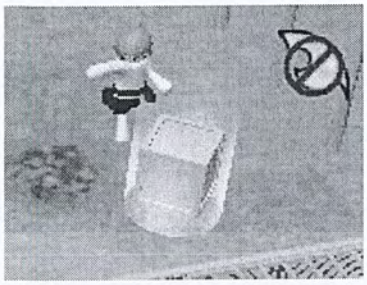
ผู้เล่น 1 คนสามารถตั้งทีม ที่มีสมาชิกในทีมได้ 4 ตัวละคร โดยเลือกเอาจากตัวละครที่มีทั้งหมด 6 แบบดังรูป 7 โดยสามารถเลือกซ้ำได้ไม่เกิน 2 ตัว เมื่อเลือกแล้ว ผู้เล่นจะต้องนำกทีมของตนเข้าเล่นน้ำกับทีมที่ควบคุมโดยปัญญาประดิษฐ์ทั้งหมดจำนวน 1-3 ทีม

รูป 4.18 หน้าจอเลือกตัวละคร

ในการเริ่มเกมจะทำการสุ่มที่อยู่ของกลุ่มต่างๆ ในแผนที่ และมีระดับความสะอาดเป็นค่าพลังชีวิตของตัวละครแต่ละตัวซึ่งจะมีค่าเต็มเมื่อเริ่มเกม ในการจู่โจมนั้นตัวละครแต่ละตัวจะมีทรัพยากรที่ใช้ในการจู่โจมจำกัด ยกตัวอย่างเช่น การใช้ปืนฉีดน้ำจะต้องใช้ทรัพยากรน้ำ การใช้แป้ง

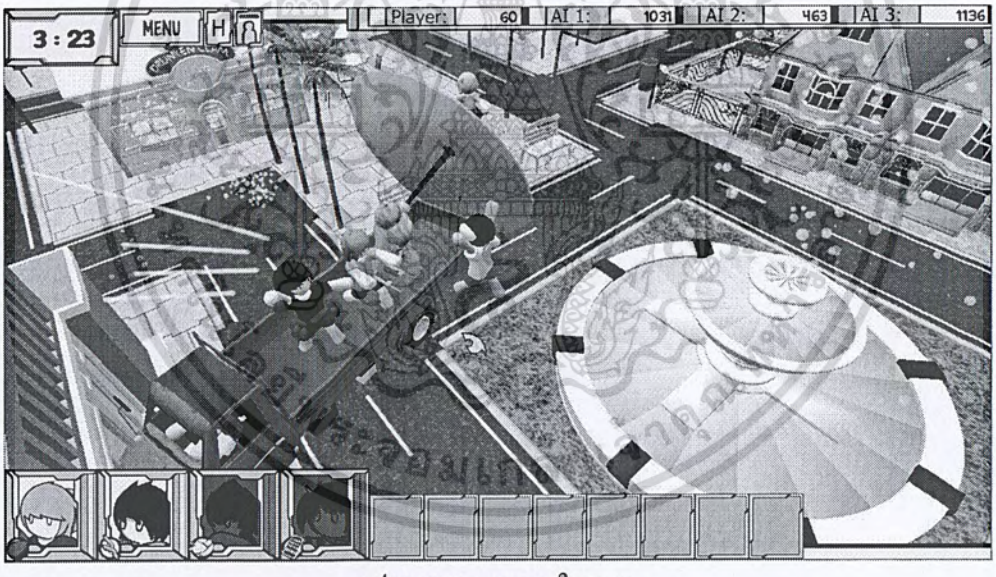
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าปะนั้น จำเป็นต้องใช้ทรัพยากรแป้ง และเมื่อทรัพยากรใกล้จะหมดแล้ว ผู้เล่นสามารถหาทรัพยากรได้จากกล่องที่ตกตามจุดต่างๆ ในแผนที่



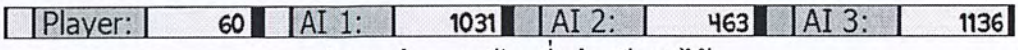
รูป 4.19 กล่องบรรจุทรัพยากร

เมื่อตัวละครถูกจู่โจมจะทำให้ระดับความสะอาดลดลง และเมื่อระดับความสะอาดลดลงจนหมด ตัวละครจะหนีออกจากฉากซึ่งหากทีมใดที่ตัวละครหนีออกจากฉากจนหมดจะแพ้ทันที ระดับความสะอาดนั้นสามารถถูกเพิ่มได้โดยความสามารถของตัวละครประเภท Healer



รูป 4.20 สภาพภายในเกม

ในการตัดสินแพ้ชนะ จะตัดสินจากค่าความสะอาดรวมกับค่าความเปียกที่ทำได้ของทั้งทีม โดยจะทำการคำนวณรวมหลังจากที่เกมจบไปแล้ว โดยจะมีการแสดงค่าความเปียกที่แต่ละทีมทำได้อยู่บน User interface ดังรูป 4.21



รูป 4.21 ค่าความเปียกที่แต่ละทีมทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 บทสรุป

โครงงานนี้ได้ทำการศึกษาหลักการของขั้นตอนวิธีเชิงพันธุกรรมและโครงข่ายประสาทเทียม เพื่อนำไปสู่การศึกษาหลักการของ NEAT (NeuroEvolution of Augmented Topology) ซึ่งเป็น NE (Neuroevolution) ประเภทหนึ่งที่ใช้ขั้นตอนวิธีเชิงพันธุกรรมวิวัฒน์ทั้งน้ำหนักของเส้นทางระหว่างโหนดและโครงสร้างของโครงข่ายประสาทเทียมซึ่งมีทั้งการเพิ่มลดเส้นทางระหว่างโหนดและการเพิ่มลดโหนด นอกจากนั้นยังได้ศึกษาการค้นหาเส้นทางโดยใช้ขั้นตอนวิธีการค้นหาแบบเอสตาร์, การใช้เกมเอนจิน Unity3D ในการสร้างเกม, และการใช้โปรแกรม Blender สำหรับการสร้างโมเดล ซึ่งเป็นความรู้พื้นฐานที่จำเป็นต่อการพัฒนาเกม

จากนั้นเราได้ใช้ Unity3D เป็นโปรแกรมหลักและเลือกใช้ภาษา C# ในการพัฒนาเกมประเภท RTS (Real-Time Strategy) ที่มีชื่อว่า "เทศกาลสงกรานต์: สแปลชอาวร์" ซึ่งมีการออกแบบให้สามารถประยุกต์ใช้ปัญญาประดิษฐ์ทั้งหลักการของ NEAT เพื่อใช้ในการสร้างอุปนิสัยที่ชาญฉลาดของ NPC (Non-Player Character) ด้วยการเรียนรู้แบบออฟไลน์และขั้นตอนวิธีการค้นหาเส้นทางแบบเอสตาร์เพื่อใช้ค้นหาเส้นทางให้กับตัวละครในเกม

เราได้ทำการทดลองต่างๆ ได้แก่ การนำ NEAT ไปใช้ทดลองกับการแก้ปัญหาตรรกศาสตร์ XOR เพื่อพิสูจน์ความสามารถในการแก้ปัญหาของการเรียนรู้แบบมีผู้สอนและแก้ปัญหา DPV (Double Pole Balancing with Velocities) เพื่อพิสูจน์ความสามารถในการแก้ปัญหาของการเรียนรู้แบบเสริมกำลัง, การทดลองการประยุกต์ใช้ขั้นตอนวิธีการค้นหาแบบเอสตาร์รวมถึงวิธีการหลบหลีกสิ่งกีดขวาง และทดลองนำโมเดลกับอนิเมชั่นที่ได้สร้างขึ้นจากโปรแกรม Blender ไปใช้ในโปรแกรม Unity3D ผลปรากฏว่าทั้ง NEAT, ขั้นตอนวิธีการค้นหาแบบเอสตาร์รวมถึงวิธีการหลบหลีกสิ่งกีดขวาง และการทำโมเดลกับอนิเมชั่นสามารถถูกนำไปใช้งานได้อย่างดี

หลังจากพัฒนาตัวเกมเสร็จแล้ว เราได้นำ NEAT ไปใช้ตามกระบวนการที่ได้วางแผนไว้ จนได้ ANN ที่มีประสิทธิภาพออกมา แล้วนำข้อมูลของ ANN ที่ได้มา ไปใช้ในตัวเกมขณะเล่นจริง

5.2 ปัญหาอุปสรรคและแนวทางในการแก้ไข

1) Unity3D เวอร์ชันฟรีไม่อนุญาตให้ใช้ฟังก์ชันการหาเส้นทาง แนวทางแก้ไขคือการเขียนฟังก์ชันการหาเส้นทางขึ้นมาเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

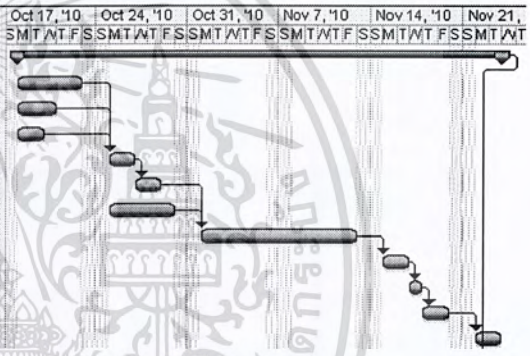
2) การเรียนรู้ของโครงข่ายประสาทเทียมจะใช้เวลานานในการจำลองการเล่นเกม แก้ไขโดยการเร่งเวลาในเกมให้เร็วขึ้น 30 เท่า

5.3 ผลการดำเนินงาน

ในการพัฒนานั้นมีการสร้างเกมตามที่ได้ออกแบบไว้ขึ้นมา โดยมีการนำ NEAT และขั้นตอนวิธีการค้นหาแบบเอสตาร์รวมถึงวิธีการหลบหลีกสิ่งกีดขวางที่ได้ทำการศึกษาและทดลอง ไปประยุกต์ใช้บนเกมดังกล่าว เพื่อให้เกมมีความสนุกสนานตื่นเต้นและมีความสมบูรณ์แบบมากยิ่งขึ้น

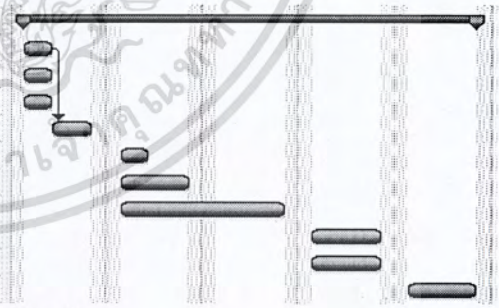
แผนการดำเนินงานถูกแยกออกเป็นส่วนต่างๆ อย่างชัดเจน โดยในการพัฒนานั้นจะแยกออกเป็นด้านปัญญาประดิษฐ์, ด้านตัวเกม, ด้านองค์ประกอบศิลป์, และด้านเสียง รายละเอียดของแต่ละช่วงของแผนการดำเนินงานเป็นไปดังรูป 5.1, 5.2, 5.3, 5.4, และ 5.5

	Task Name	Duration	Start	Finish	Prede
1	⊟ AI	27 days	Mon 10/18/10	Tue 11/23/10	
2	Implement script	5 days	Mon 10/18/10	Fri 10/22/10	
3	Implement ANN input	3 days	Mon 10/18/10	Wed 10/20/10	
4	Create NPC	2 days	Mon 10/18/10	Tue 10/19/10	
5	Integrate ANN into NPC	2 days	Mon 10/25/10	Tue 10/26/10	2,3,4
6	Verify NPC	2 days	Wed 10/27/10	Thu 10/28/10	5
7	Create learning flow	5 days	Mon 10/25/10	Fri 10/29/10	
8	Learning	10 days	Mon 11/1/10	Fri 11/12/10	6,7
9	Verify result	2 days	Mon 11/15/10	Tue 11/16/10	8
10	Integrate result into game	1 day	Wed 11/17/10	Wed 11/17/10	9
11	Verify gameplay	2 days	Thu 11/18/10	Fri 11/19/10	10
12	Summerize	2 days	Mon 11/22/10	Tue 11/23/10	11



รูป 5.1 แผนการดำเนินงานด้านปัญญาประดิษฐ์

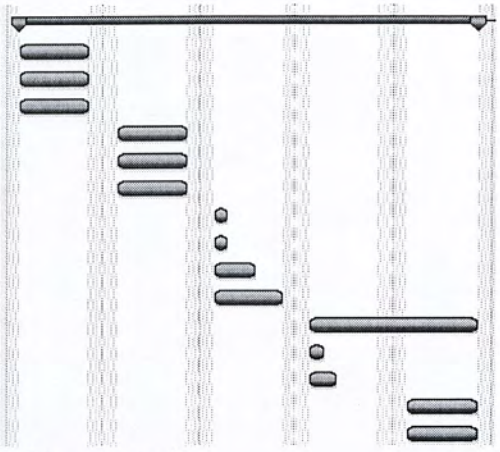
13	⊟ Game flow	25 days	Mon 10/18/10	Fri 11/19/10	
14	Main Menu	2 days	Mon 10/18/10	Tue 10/19/10	
15	Splash Screen	2 days	Mon 10/18/10	Tue 10/19/10	
16	Credit Scene	2 days	Mon 10/18/10	Tue 10/19/10	
17	Option Menu	3 days	Wed 10/20/10	Fri 10/22/10	14
18	Sonkran History Slides	2 days	Mon 10/25/10	Tue 10/26/10	
19	Character Selection Menu	5 days	Mon 10/25/10	Fri 10/29/10	
20	Ingame UI	10 days	Mon 10/25/10	Fri 11/5/10	
21	Fog of war	5 days	Mon 11/8/10	Fri 11/12/10	
22	Result Scene	5 days	Mon 11/8/10	Fri 11/12/10	
23	Map	5 days	Mon 11/15/10	Fri 11/19/10	



รูป 5.2 แผนการดำเนินงานด้านตัวเกม

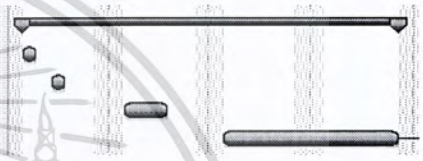
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

24	**Art**	**25 days**	**Mon 10/18/10**	**Fri 11/19/10**
25	Raincoater	5 days	Mon 10/18/10	Fri 10/22/10
26	Powder Smearer	5 days	Mon 10/18/10	Fri 10/22/10
27	Watergunslinger	5 days	Mon 10/18/10	Fri 10/22/10
28	Water Bomber	5 days	Mon 10/25/10	Fri 10/29/10
29	Caretaker	5 days	Mon 10/25/10	Fri 10/29/10
30	Charmer	5 days	Mon 10/25/10	Fri 10/29/10
31	KMITL Splash Screen	1 day	Mon 11/1/10	Mon 11/1/10
32	Unity Splash Screen	1 day	Mon 11/1/10	Mon 11/1/10
33	Main Menu	3 days	Mon 11/1/10	Wed 11/3/10
34	Character Selection Menu	5 days	Mon 11/1/10	Fri 11/5/10
35	Ingame UI	10 days	Mon 11/8/10	Fri 11/19/10
36	Option Menu	1 day	Mon 11/8/10	Mon 11/8/10
37	Credit Scene	2 days	Mon 11/8/10	Tue 11/9/10
38	Songkran History Slides	5 days	Mon 11/15/10	Fri 11/19/10
39	Result Scene	5 days	Mon 11/15/10	Fri 11/19/10



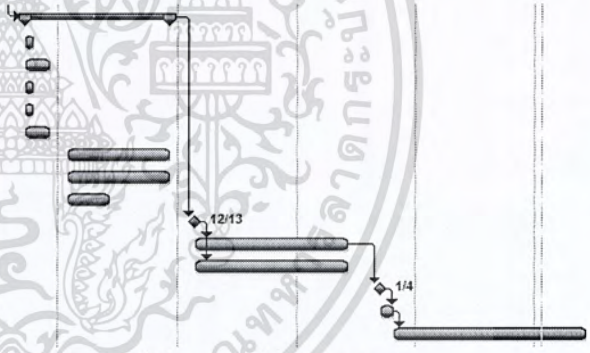
รูป 5.3 แผนการดำเนินงานด้านองค์ประกอบศิลป์

40	**Sound**	**20 days**	**Mon 10/25/10**	**Fri 11/19/10**
41	Main Menu	1 day	Mon 10/25/10	Mon 10/25/10
42	Credit Scene	1 day	Wed 10/27/10	Wed 10/27/10
43	Ingame	3 days	Mon 11/1/10	Wed 11/3/10
44	Character	10 days	Mon 11/8/10	Fri 11/19/10



รูป 5.4 แผนการดำเนินงานด้านเสียง

45	**Integrate art and sound into game**	**Wed 11/24/10**	**Fri 12/10/10**	
46	Splash Screen	Wed 11/24/10	Wed 11/24/10	
47	Main Menu	Wed 11/24/10	Fri 11/26/10	
48	Option Menu	Wed 11/24/10	Wed 11/24/10	
49	Credit Scene	Wed 11/24/10	Wed 11/24/10	
50	Songkran history slides	Wed 11/24/10	Fri 11/26/10	
51	Character Selection Menu	Mon 11/29/10	Fri 12/10/10	
52	Ingame UI	Mon 11/29/10	Fri 12/10/10	
53	Tutorial mode	Mon 11/29/10	Fri 12/3/10	
54	Code freeze	Mon 12/13/10	Mon 12/13/10	
55	Verify	Tue 12/14/10	Fri 12/31/10	
56	Validate	Tue 12/14/10	Fri 12/31/10	
57	Release	Mon 1/3/11	Tue 1/4/11	
58	Postmotem	Wed 1/5/11	Thu 1/6/11	
59	Documentation	Thu 1/6/11	Fri 1/28/11	



รูป 5.5 แผนการดำเนินงานภายหลังการพัฒนาองค์ประกอบต่างๆ ของเกม

5.4 แนวทางในการพัฒนาต่อ

ในการพัฒนาเกมต่อนั้นสามารถเพิ่มโหมดการเล่นต่างๆ อาทิเช่นโหมดเนื้อเรื่องหรือโหมดเล่นหลายคนผ่านระบบเครือข่าย รวมถึงการพัฒนาอุปนิสัยของปัญญาประดิษฐ์เช่นการนำโครงข่ายประสาทเทียมมาใช้ในการตัดสินใจเป็นกลุ่มและตัดสินใจว่าการกระทำดังกล่าวที่ผ่านการตัดสินใจจากโครงข่ายประสาทเทียมที่มีอยู่แล้วไปคิดต่อว่าจะทำงานนั้นๆ อย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Negnevitsky, Michael. 2005. **Artificial Intelligence: A Guide to Intelligent Ststem.** 2nd ed. Great Britain : Biddles Ltd, King's Lynn.

Russell, S. and Norvig, P. 2002. **Artificial Intelligence: A Modern Approach.** 2nd ed. New Jersey : Prentice Hall.

Goldstone, W. 2009. **Unity Game Development Essentials.** Birmingham : Packt Publishing.

Stanley, K. O. and Miikkulainen, R. 2002. "Evolving Neural Networks through Augmenting Topologies." **Evolutionary Computation.** 10(2).

Marek, O. 1998. **Introduction to Genetic Algorithms.** [Online]. Available : http://www.obitko.com/tutorials/genetic-algorithms/index.php.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้