ปัญญาประดิษฐ์แบบปรับตัวได้อัตโนมัติสำหรับเกมต่อสู้

นาย สรายุทธ เหลืองเรืองโรจน์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

AUTOMATIC ADAPTIVE AI FOR FIGHTING GAME

Mr. Sarayut Lueangrueangroj

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 2009

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ปัญญาประดิษฐ์แบบปรับตัวได้อัตโนมัติสำหรับเกมต่อสู้
โดย	นาย สรายุทธ เหลืองเรืองโรจน์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.วิษณุ โคตรจรัส

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อรรถวิทย์ สุดแสง)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.วิษณุ โคตรจรัส)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.จรัสศรี รุ่งรัตนาอุบล)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สรายุทธ เหลืองเรืองโรจน์ : ปัญญาประดิษฐ์แบบปรับตัวได้อัตโนมัติสำหรับเกมต่อสู้. (AUTOMATIC ADAPTIVE AI FOR FIGHTING GAME) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร. วิษณุ โคตรจรัส, 80 หน้า.

เกมต่อสู้เป็นเกมประเภทหนึ่งที่ได้รับความนิยมเป็นอย่างสูง เกมต่อสู้ที่ดีจะต้องทำให้ผู้เล่นได้ต่อสู้กับคู่ต่อสู้ที่ให้ความท้าทาย แต่ว่าส่วนใหญ่แล้วปัญญาประดิษฐ์ที่มากับเกมต่อสู้นั้นสามารถคาดเดาพฤติกรรมได้ง่ายและมีพฤติกรรมซ้ำซ้อนเพราะว่าใช้สคริปต์ที่ตั้งไว้ก่อนในการตัดสินใจ ผู้เล่นส่วนใหญ่จึงเลือกที่จะเล่นกับคู่แข่งที่เป็นมนุษย์เนื่องจากสามารถให้ประสบการณ์ในการเล่นที่ท้าทายได้หลากหลายรูปแบบกว่า เพื่อให้ตัวละครที่ควบคุมโดยคอมพิวเตอร์สามารถให้ความท้าทายได้หลายระดับ วิทยานิพนธ์ฉบับนี้จึงนำเสนอเทคนิคปัญญาประดิษฐ์สำหรับเกมต่อสู้แบบใหม่ ด้วยการใช้งานเทคนิคซึ่งสามารถเขียนโปรแกรมได้ง่ายและมีพื้นฐานจากการเรียนรู้แบบรีนฟอร์สเมนท์ ตัวปัญญาประดิษฐ์ที่สร้างขึ้นนั้นมีความสามารถในการเรียนรู้และปรับตัวตามเวลาจริงในระหว่างการเล่น โดยปรับตัวตามการเล่นของผู้เล่นในเวลานั้นให้เล่นได้ดีขึ้นเมื่อต่อสู้กับผู้เล่นคนนั้น ซึ่งจะทำให้เกิดความท้าทายระดับต่างๆสำหรับผู้เล่นนั้น ตัวปัญญาประดิษฐ์นั้นพัฒนาและทดสอบบนสภาพแวดล้อมในการทดสอบสำหรับเกมที่ชื่อว่า ไอเทม ซึ่งสร้างจากโปรแกรมอีมูเลเตอร์ของเครื่องเล่นเกมแบบพกพา โดยมีความสามารถในการสร้างและทดสอบปัญญาประดิษฐ์กับเกมที่วางจำหน่ายจริงในท้องตลาดได้ การใช้เกมที่มีขายจริงในการทดสอบปัญญาประดิษฐ์นั้นช่วยให้แน่ใจได้ว่าเอไอที่สร้างขึ้นสามารถนำไปใช้ได้จริง เกมที่ใช้ในการทดสอบคือสตรีทไฟเตอร์ซีโร่สามอัพเปอร์ ผลการทดลองแสดงให้เห็นว่าปัญญาประดิษฐ์ที่ได้พัฒนาขึ้นใหม่นี้สามารถพัฒนาตนเองได้ในระหว่างการเล่นและสามารถให้ประสบการณ์ในการเล่นที่ผู้เล่นพึงพอใจได้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต สรายุทธ เหลืองเรืองโรจน์

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2552

4970620521 : MAJOR Computer Engineering

KEYWORDS : FIGHTING GAME / ARTIFICIAL INTELLIGENCE / ONLINE LEARNING / DYNAMIC SCRIPTING

SARAYUT LUEANGRUEANGROJ : AUTOMATIC ADAPTIVE AI FOR FIGHTING GAME. THESIS ADVISOR : ASST PROF. VISHNU KOTRAJARAS, Ph.D., 80 pp.

The fighting game is one of the most successful genres in the gaming industry. Playing against a challenging opponent is the heart of a great fighting game. However, most of the time, an artificially intelligent opponent that comes with a fighting game is too predictable and repetitive because it is pre-scripted. Most players resort to playing with live opponents because real people provide a greater variety of challenges. To allow a computer-controlled character to provide various levels of challenge, this thesis introduce a novel artificial intelligence (AI) approach for fighting games. Using an easily implemented technique loosely based on reinforcement learning, the AI has an ability to learn and adapt in real-time during a playing session according to how the player of that session plays the game. The AI improves its performance against the player, which ultimately provides various challenge levels for the player. The AI was developed and tested on testbed AI-TEM, created from a handheld machine emulator. Real commercial games can be played on AI-TEM and used for AI testing. Using a commercial game to test our AI helps to ensure that the AI is applicable in real games. With Street Fighter Zero 3 Upper as a case study, the experimental result shows that the enhanced AI can improve itself during game play and provide a more satisfying experience for players.

Department : Computer Engineering Student's Signature .....

Field of Study : Computer Science Advisor's Signature .....

Academic Year : 2009

กิตติกรรมประกาศ

ขอขอบคุณ บิดา มารดา น้องชาย ครอบครัวอันเป็นที่รัก ที่เป็นกำลังสำคัญ ให้ความช่วยเหลือในทุกๆด้านแก่ผู้เขียน จนสามารถทำวิทยานิพนธ์ฉบับนี้เสร็จสิ้นได้

ขอขอบคุณอาจารย์ที่ปรึกษา ผศ.ดร.วิษณุ โคตรจรัส ผู้ให้แนวทาง ข้อคิด คำปรึกษา ตลอดจนรับฟังความคิดเห็นด้านต่างๆ และเป็นผู้ตรวจทานแก้ไขให้วิทยานิพนธ์ฉบับนี้เสร็จสิ้นลงได้

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ ผศ.ดร. อรรถวิทย์ สุดแสง ผศ.ดร. เศรษฐา ปานงาม และ ผศ.ดร.จรัสศรี รุ่งรัตนาอุบล ที่ได้กรุณาช่วยให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น

ขอขอบคุณสมาชิกทุกคนในห้องปฏิบัติการเกมที่ได้ให้คำแนะนำเพิ่มเติมและช่วยเหลือในการทดลองเสมอมา

สุดท้ายนี้ขอขอบคุณผู้พัฒนาเกมทั้งหลายที่สร้างสรรค์ผลงานอันสนุกสนานประทับใจ มอบความฝัน และเป็นแรงบันดาลใจให้กับผู้เขียนในการทำวิทยานิพนธ์ฉบับนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ญ

สารบัญภาพ..... ฏ

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของการวิจัย..... 3

1.3 ขอบเขตของการวิจัย..... 3

1.4 ข้อจำกัดของการวิจัย..... 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 4

1.6 วิธีดำเนินการวิจัย..... 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 5

2.1 เกมต่อสู้..... 5

2.2 เอไอ ปัญญาประดิษฐ์..... 7

2.3 ไอ-เทม..... 7

2.4 โกสเอไอ..... 8

2.5 ไดนามิคสคริปติง..... 9

2.6 Temporal Difference Learning..... 12

2.7 Learning to fight..... 13

2.8 Automatic computer game balancing:a reinforcement learning approach 13

2.9 A method for online adaption of computer-game AI rulebase..... 14

บทที่ 3 ระบบต้นแบบ โกสเอไอที่ปรับอัตราการออกท่าที่เลียนแบบได้ตามเวลาจริง..... 15

3.1 เครื่องมือและสภาพแวดล้อมที่ใช้ในการพัฒนา..... 15

3.2 ภาพรวมของระบบเอไอ..... 15

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

3.3 รายละเอียดของสถานการณ์ในเกม..... 16
3.4 โครงสร้างของเคส..... 16
3.5 รายละเอียดการทำงานของเอไอ..... 17
3.5.1 ส่วนการลอกเลียนแบบผู้เล่น..... 17
3.5.2 ส่วนการเรียนรู้ของเอไอ..... 18
3.5.3 การคิดคะแนนท่าประสิทธิภาพเมื่อท่าใช้สร้างความเสียหายแก่ฝ่ายตรงข้ามได้..... 19
3.5.4 การใช้ข้อมูลเคสในการเล่นเกม..... 20
3.5.5 การเลือกท่าจากการพิจารณาค่าความน่าจะเป็น..... 22
3.6 การทดลอง..... 23
3.7 สรุปผลการทดลอง..... 25
บทที่ 4 ปัญญาประดิษฐ์แบบปรับตัวได้อัตโนมัติ..... 27
4.1 เครื่องมือและสภาพแวดล้อมที่ใช้ในการพัฒนา..... 27
4.2 ภาพรวมของระบบเอไอ..... 27
4.3 รายละเอียดของสถานการณ์ในเกม..... 28
4.4 โครงสร้างของเคส..... 29
4.5 รายละเอียดการทำงานของเอไอ..... 29
4.5.1 สังเกตและบันทึกข้อมูลเบื้องต้นระหว่างความสัมพันธ์ของสถานการณ์ในเกมและการกระทำที่ตัวละครแสดงออก..... 30
4.5.2 สร้างข้อมูลเคส..... 30
4.5.3 ปรับปรุงข้อมูลเคส..... 32
4.5.4 การเลือกท่ามาใช้ในสถานการณ์ต่างๆ..... 34
4.5.5 การเรียนรู้การกระทำเพื่อหวังผลในระยะยาว..... 37
4.5.6 การคิดค่าคะแนนประสิทธิภาพเมื่อท่าใช้สร้างความเสียหายแก่ฝ่ายตรงข้ามได้..... 40
บทที่ 5 การทดลองเอไอและวิเคราะห์ผล..... 41
5.1 การทดลอง..... 41
5.2 ผลการทดลอง..... 41
5.3 วิเคราะห์ผลการทดลอง..... 43

หน้า

5.3.1 ความรู้เกี่ยวกับท่าที่จะนำมาใช้งาน..... 45
5.3.2 จังหวะการแสดงออกของท่า..... 46
5.3.3 Short animation..... 47
5.3.4 อนิเมชันที่ซ้ำซ้อนกัน..... 47
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ..... 48
6.1 สรุปผลการวิจัย..... 48
6.2 อภิปรายการวิจัยและข้อเสนอแนะ..... 48
รายการอ้างอิง..... 50
บรรณานุกรม..... 52
ภาคผนวก..... 53
ภาคผนวก ก ผลงานการตีพิมพ์..... 54
ภาคผนวก ข รายละเอียดปลีกย่อยเกี่ยวกับเอไอ..... 64
ประวัติผู้เขียนวิทยานิพนธ์..... 80

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ หน้า
ตารางที่ 1 เปรียบเทียบค่าประสิทธิภาพระหว่าง GAI และ AGAI..... 24
ตารางที่ 2 จำนวนครั้งที่มีการนำเคสที่เรียนรู้ได้มาใช้งานเทียบกับจำนวนครั้งที่ออกท่าแบบสุ่มและจำนวนเคสที่เรียนรู้ได้ของ GAI และ AGAI..... 25
ตารางที่ 3 ตัวอย่างการปรับปรุงค่าประสิทธิภาพของท่าต่อเนื่องในเคส..... 39
ตารางที่ 4 ตัวอย่างการคิดคะแนนท่าประสิทธิภาพของท่า..... 40
ตารางที่ 5 เปรียบเทียบค่าประสิทธิภาพระหว่าง AGAI และ AAI..... 43
ตารางที่ 6 จำนวนครั้งที่มีการนำเคสที่เรียนรู้ได้มาใช้งานเทียบกับจำนวนครั้งที่ออกท่าแบบสุ่มและจำนวนเคสที่เรียนรู้ได้ของ AGAI และ AAI..... 43
ตารางที่ 7 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและผู้เล่นตามแนวแกน x..... 65
ตารางที่ 8 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและผู้เล่นตามแนวแกน y..... 65
ตารางที่ 9 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและกระสุนของฝ่ายใดก็ตามตามแนวแกน x..... 65
ตารางที่ 10 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและกระสุนของฝ่ายใดก็ตามตามแนวแกน y..... 66
ตารางที่ 11 รายละเอียดสถานะที่เป็นไปได้ของตัวละครฝ่ายเอไอ..... 66
ตารางที่ 12 รายละเอียดแสดงสถานะติดมุมของตัวละคร..... 66
ตารางที่ 13 รูปแบบการกระทำทั้งหมดของตัวละคร Ryu..... 67
ตารางที่ 14 รายละเอียดการเข้ารหัสสถานการณ์ในเกม..... 68
ตารางที่ 15 รายละเอียดการเข้ารหัสที่ไม่มีการนำมาใช้..... 69

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่ หน้า
รูปที่ 1 ตัวอย่างเกมต่อสู้แบบสองมิติ Street Fighter II Turbo HD Remix..... 6
รูปที่ 2 ตัวอย่างเกมต่อสู้แบบสามมิติ Tekken 6..... 6
รูปที่ 3 ตัวอย่างเกมต่อสู้แบบกึ่งสองมิติและสามมิติ Street fighter IV..... 7
รูปที่ 4 ไอ-เทม สภาพแวดล้อมในการทดสอบเอไอ..... 8
รูปที่ 5 การทำงานของไดนามิคสคริปติง..... 11
รูปที่ 6 ตัวอย่างโครงสร้างของเคสของ AGAI..... 17
รูปที่ 7 ผังงานแสดงขั้นตอนการทำงานส่วนลอกเลียนแบบผู้เล่นของ AGAI..... 18
รูปที่ 8 ผังงานแสดงขั้นตอนการทำงานของส่วนการเรียนรู้ของ AGAI..... 19
รูปที่ 9 ผังงานแสดงขั้นตอนการใช้ข้อมูลเคสในการเล่นเกมของ AGAI..... 22
รูปที่ 10 กราฟแสดงผลการทดลองของระบบเอไอต้นแบบ AGAI..... 23
รูปที่ 11 Street Fighter Zero 3 Upper..... 27
รูปที่ 12 องค์ประกอบหลักของเอไอ..... 28
รูปที่ 13 โครงสร้างของเคส..... 29
รูปที่ 14 ข้อมูลคู่อันดับ (Casekey , การกระทำ) ของเอไอและผู้เล่น..... 30
รูปที่ 15 กระบวนการสร้างข้อมูลเคส..... 32
รูปที่ 16 กระบวนการปรับปรุงข้อมูลภายในเคส..... 34
รูปที่ 17 กระบวนการเลือกท่า..... 37
รูปที่ 18 กราฟแสดงผลการทดลอง..... 40
รูปที่ 19 รายละเอียดการกำหนดค่าของบิตต่างๆในตัวแปรอินทีเจอร์ที่นำเอามาใช้เป็นค่า Casekey..... 68
รูปที่ 20 สถานการณ์ขณะตัวละครฝ่ายผู้เล่นทำการปล่อยกระสุนโจมตี..... 74
รูปที่ 21 สถานการณ์ขณะที่ตัวละครฝ่ายเอไอโดนโจมตีด้วยกระสุน..... 74
รูปที่ 22 สถานการณ์ขณะตัวละครฝ่ายเอไอกำลังจะถูกโจมตีด้วยกระสุน..... 75

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อุตสาหกรรมวีดีโอเกมนั้นมีการพัฒนาไปเป็นอย่างมาก จากในช่วงเริ่มแรกที่เป็นเพียงงานอดิเรกที่ได้รับความสนใจอยู่ในกลุ่มเฉพาะ จนก้าวขึ้นมาเป็นที่รู้จักแพร่หลายและได้รับความนิยมอย่างสูงไปทั่วโลก กลายเป็นหนึ่งในอุตสาหกรรมที่มีมูลค่าสูงที่สุดในโลกอุตสาหกรรมหนึ่งในกลุ่มอุตสาหกรรมประเภทเพื่อความบันเทิงในปัจจุบัน เกมที่ได้รับความนิยมสูงนั้นสามารถทำยอดจำหน่ายได้หลายล้านชุดทั่วโลก และถึงหลายสิบล้านหรือร้อยล้านชุดในบางเกมที่มีการวางจำหน่ายมาหลายภาค สร้างมูลค่าจากการขายมากกว่าภาพยนตร์ที่ได้รับความนิยมไปทั่วโลกเสียอีก เกมเหล่านี้นั้นสามารถทำรายได้เข้าสู่บริษัทผู้ผลิตอย่างมหาศาล และเป็นอุตสาหกรรมที่มีความน่าสนใจเป็นอย่างมาก ผู้ผลิตมีการพัฒนาเกมใหม่ๆ และรูปแบบการเล่นใหม่ๆออกมาสู่ตลาดอย่างต่อเนื่อง มีการวิจัยสร้างสรรค์และพัฒนานวัตกรรมและเทคโนโลยีหลากหลายรูปแบบเพื่อนำมาใช้กับเกมโดยเฉพาะ กล่าวได้ว่าเกมนั้นเป็นหนึ่งในอุตสาหกรรมที่มีส่วนในการขับเคลื่อนเทคโนโลยีให้มีการพัฒนาไปได้

เกมต่อสู้นั้นเป็นเกมประเภทหนึ่งที่อยู่คู่กับอุตสาหกรรมเกมมาอย่างยาวนานตั้งแต่ช่วงเริ่มต้นของอุตสาหกรรม มีการพัฒนาและวางจำหน่ายเกมต่อสู้จำนวนมากและได้รับความนิยมมายาวนานอย่างต่อเนื่อง เช่นเกมในชุด Street Fighter, The King of Fighters, Tekken เป็นต้น ในเรื่องของการพัฒนานั้น เกมต่อสู้มีการพัฒนาไปเป็นอย่างมากในหลายด้าน ทั้งในด้านภาพกราฟิกของตัวเกมที่เริ่มแรกมีลักษณะเป็นสองมิติโดยมีตัวละครเป็นรูปสไปร์ท (sprite) สองมิติ มาเป็นสามมิติและมีตัวละครเป็นโมเดลสามมิติ และในด้านรูปแบบการเล่น ในด้านระบบต่างๆของเกมต่อสู้ที่มีการพัฒนาไปอย่างมากนั้น มีเพียงส่วนหนึ่งที่แทบจะไม่ได้มีการพัฒนาไปเลยตั้งแต่ช่วงเริ่มต้น นั่นคือส่วนระบบปัญญาประดิษฐ์ (เอไอ จากนี้ไปจะขอใช้คำนี้เรียกทับศัพท์จากคำศัพท์ภาษาอังกฤษเมื่อกล่าวถึงปัญญาประดิษฐ์) ของเกมต่อสู้ ในเกมต่อสู้ทั่วไปโดยส่วนใหญ่นั้น เอไอจะทำงานโดยใช้ระบบสคริปต์ (script) กล่าวคือ จะมีลักษณะเป็นเหมือนโปรแกรมที่คอยควบคุมการโต้ตอบแสดงออกของตัวละครฝ่ายเอไอในเกมให้มีลักษณะเป็นไปตามที่โปรแกรมกำหนดมา โดยจะมีลักษณะการแสดงออกอ้างอิงกับสถานการณ์ในเกม เช่นให้ตัวละครที่ฝ่ายเอไอเป็นผู้ควบคุมแสดงออกแบบหนึ่งเมื่อเจอสถานการณ์แบบหนึ่ง และแสดงออกอีกแบบหนึ่งเมื่อเจอสถานการณ์อีกแบบหนึ่ง สิ่งที่ได้จากระบบเอไอแบบนี้คือ ตัวละครที่เอไอเป็นผู้ควบคุม ที่สามารถ

ควบคุมสั่งการให้แสดงออกในแบบต่างๆได้ง่ายไม่ซับซ้อน เป็นไปตามที่ผู้พัฒนาเกมกำหนดและต้องการ และสามารถตอบสนองต่อการเล่นกับผู้เล่นที่เป็นมนุษย์จริงๆได้ โดยให้ความท้าทายในการเล่นได้ในระดับหนึ่ง

อย่างไรก็ตามเอไอที่ทำงานโดยใช้ระบบสคริปต์นั้นก็ยังมีข้อด้อยอยู่ เนื่องจากการโต้ตอบต่อสถานการณ์ต่างๆในเกมนั้นเป็นไปอย่างมีแบบแผนแน่นอนตามที่เอไอถูกโปรแกรมกำหนดมา ดังนั้นการแสดงออกที่ตัวละครฝ่ายเอไอสามารถทำได้จึงมีรูปแบบตายตัว จำกัด และคาดเดาได้ง่าย ถึงแม้ว่าเอไอที่ใช้ระบบสคริปต์นี้จะสามารถตอบสนองต่อความต้องการของผู้เล่นที่ต้องการความท้าทายในการเล่นได้ในระดับหนึ่ง แต่ผู้เล่นที่มีประสบการณ์ในการเล่นสูงในระดับหนึ่งแล้วจะเลือกที่จะเล่นกับฝ่ายตรงข้ามที่เป็นมนุษย์จริงๆมากกว่า เนื่องจากสามารถแสดงออกถึงรูปแบบการเล่นที่หลากหลายกว่าเอไอปกติมาก เนื่องจากผู้เล่นที่เป็นมนุษย์นั้นสามารถได้รับประสบการณ์จากการเล่น และสามารถพัฒนารูปแบบการเล่นให้ดียิ่งขึ้นไปได้ ซึ่งต่างกับเอไอที่มีรูปแบบการเล่นคงที่ในลักษณะเดียวตลอดตามที่ถูกโปรแกรมกำหนดมา ทำให้การเล่นกับผู้เล่นที่เป็นมนุษย์นั้นสามารถเล่นด้วยได้อย่างสนุกและท้าทายมากกว่าเล่นกับเอไอของเกมปกติ เช่นดังที่ปรากฏว่า แม้จะเป็นเอไอที่มีระดับความเก่งสูงสุดของเกมแล้ว ถึงแม้ว่าในตอนแรกผู้เล่นจะไม่สามารถเล่นด้วยแล้วเอาชนะได้โดยง่าย แต่ด้วยการโต้ตอบที่มีแบบแผนของเอไอ เมื่อเล่นผ่านไปชั่วเวลาหนึ่ง ผู้เล่นสามารถที่จะเรียนรู้และคาดเดาการกระทำของเอไอได้อย่างแม่นยำว่า เอไอจะแสดงการกระทำอะไรออกมา จึงสามารถที่จะโต้ตอบและเอาชนะเอไอที่มีระดับความเก่งสูงสุดของเกมได้โดยง่ายดาย ด้วยสาเหตุนี้ผู้เล่นที่มีประสบการณ์จึงมักที่จะเลือกเล่นกับผู้เล่นด้วยกันเองมากกว่าเอไอของตัวเกม ซึ่งตรงจุดนี้ทำให้เอไอของเกมต่อสู้ถูกลดความสำคัญลงไป กลายเป็นเพียงสิ่งที่ติดมากับตัวเกมเท่านั้น

ในการเล่นกับผู้เล่นคนอื่นนั้น ในอดีตมีข้อจำกัดเพราะผู้เล่นจำเป็นต้องเดินทางไปพบกับผู้เล่นคนอื่น นอกจากนี้ความจำเป็นที่ต้องมีเครื่องเกมสำหรับเล่นอาจทำให้ผู้เล่นต้องนำเครื่องเกมเดินทางไปด้วย ซึ่งทำให้เกิดความไม่สะดวกขึ้น แต่ในปัจจุบัน ด้วยระบบอินเทอร์เน็ตที่มีการพัฒนาไปอย่างมาก รวมถึงเกมต่อสู้ที่วางจำหน่ายในท้องตลาดส่วนใหญ่จะมีระบบการเล่นแบบออนไลน์ใส่มาในเกมด้วย ทำให้ผู้เล่นสามารถที่จะเล่นกับผู้เล่นคนอื่นแบบออนไลน์ผ่านระบบอินเทอร์เน็ตได้ โดยที่ไม่จำเป็นต้องเดินทางไปหาอีกต่อไป อย่างไรก็ตามแม้ระบบอินเทอร์เน็ตจะช่วยแก้ปัญหาเรื่องการค้นหาผู้เล่นได้ แต่กลับมีปัญหาอื่นเกิดขึ้นมาแทน ความเร็วอินเทอร์เน็ตนั้นมีผลอย่างมากในการสร้างสภาพแวดล้อมในการเล่นที่ผู้เล่นสามารถเล่นได้อย่างพึงพอใจ โดยเฉพาะกับเกมต่อสู้ที่ต้องการความเร็วในการรับส่งข้อมูลประมวลผลและความแม่นยำอย่างสูง

ด้วยแล้ว ความเร็วอินเทอร์เน็ตในประเทศกำลังพัฒนาที่ไม่เร็วได้มาตรฐานเพียงพอนั้น สามารถเปลี่ยนประสบการณ์ในการเล่นที่น่าพอใจ ไปเป็นประสบการณ์ที่ทำให้ผู้เล่นอยากจะเลิกเล่นไปเลยได้ เช่นทำให้ตัวละครไม่สามารถตอบสนองได้ทันเวลาตามที่ผู้เล่นควบคุม นอกจากนั้นแล้วการเล่นแบบออนไลน์นั้นต้องให้ผู้เล่นทั้งสองฝ่ายออนไลน์ในเวลาเดียวกันด้วย หากผู้เล่นต้องการเล่นแต่ระบบออนไลน์ไม่เอื้ออำนวยหรือไม่สามารถค้นหาผู้เล่นคนอื่นได้ เอไอของเกมจึงเป็นสิ่งจำเป็นที่ผู้เล่นต้องนำมาใช้ แต่เมื่อคำนึงถึงระดับประสบการณ์ในการเล่นที่เอไอสามารถให้แก่ผู้เล่นได้แล้ว อาจทำให้ผู้เล่นหมดความสนใจที่จะเล่นไปก็ได้ ซึ่งทำให้คุณค่าของตัวเกมนั้นลดลง

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการสร้างระบบเอไอสำหรับเกมต่อสู้ที่มีความสามารถในการพัฒนาตัวเองขึ้นได้แบบอัตโนมัติในเวลาจริง โดยอาศัยประสบการณ์ที่ได้จากการเล่น ลบข้อจำกัดของเอไอเกมต่อสู้ปกติ ที่มีการตอบสนองในรูปแบบเดิมซ้ำๆ และให้ประสบการณ์ในการเล่นที่ท้าทายมากกว่าการเล่นกับเอไอของเกมปกติได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อสร้างเอไอสำหรับเกมต่อสู้ที่สามารถปรับตัวตามผู้เล่นได้และพัฒนาตัวเองได้แบบอัตโนมัติในเวลาจริง โดยอาศัยประสบการณ์ที่ได้จากการเล่น ลบข้อจำกัดของเอไอแบบปกติ สร้างความท้าทายให้กับผู้เล่น ซึ่งสามารถเพิ่มมูลค่าให้กับตัวเกมได้

1.3 ขอบเขตของการวิจัย

- กรณีศึกษาคือเกม Street Fighter Zero 3 Upper
- ทำการพัฒนาเอไอและทดลองบนสภาพแวดล้อมสำหรับทดสอบเอไอ (Testbed) ที่ชื่อว่า AI-TEM
- ทำการสร้างเอไอและทดลองเป็นกรณีศึกษาโดยใช้ข้อมูลของตัวละครในเกมเพียงตัวเดียวคือ ตัวละคร Ryu

1.4 ข้อจำกัดของการวิจัย

งานวิจัยนี้วิจัยและพัฒนาบนเกม Street Fighter Zero 3 Upper ของเครื่องเกม Nintendo Gameboy Advance (GBA) โดยใช้เพียงตัวละคร Ryu เท่านั้น โปรแกรมและสภาพแวดล้อมในการทดลองถูกปรับแต่งมาเพื่อใช้กับเกมและตัวละครที่กำหนด เพื่อเป็นกรณีศึกษาเท่านั้น อย่างไรก็ตาม หากต้องการนำไปใช้กับตัวละครอื่นหรือเกมอื่นๆ สามารถนำแนวคิดในงานวิจัยนี้ไปปรับใช้ได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- ได้เอไอที่สามารถเรียนรู้ ปรับตัว และพัฒนาตามผู้เล่นได้แบบอัตโนมัติในเวลาจริง และมีคุณภาพใช้ได้ในระดับเดียวกับเกมที่มีวางขายในท้องตลาด
- เป็นระบบเอไอใหม่ที่ยังไม่เคยมีการนำมาใช้ในเกมต่อสู้ ซึ่งสามารถเพิ่มมูลค่าให้กับตัวเกมได้

1.6 วิธีดำเนินการวิจัย

1. ศึกษาวิธีควบคุมและใช้งาน Testbed AI-TEM
2. พัฒนาส่วนติดต่อใช้งานกับ Testbed
3. สร้างส่วนข้อมูลอนิเมชันและข้อมูลท่ากดของตัวละครที่ใช้ในการทดสอบใส่ลงไปใน Testbed
4. พัฒนาส่วนฟังก์ชันการลอกแบบการเล่นของผู้เล่นใส่ลงไปใน Testbed
5. พัฒนาส่วนฟังก์ชันการทดลองใช้ท่าใหม่สำหรับสถานการณ์ต่างๆใส่ลงไปใน Testbed
6. พัฒนาส่วนฟังก์ชันเรียนรู้ที่รวมการลอกแบบผู้เล่นและทดลองใช้ท่าใหม่เข้าไว้ด้วยกัน
7. ทดลองใช้งานและปรับแต่งโมดูลการเรียนรู้ให้มีความเหมาะสมมากขึ้น
8. ออกแบบวิธีการวัดผลและเงื่อนไขการวัดผล
9. ทำการทดลองและเก็บผลจากกลุ่มผู้ทดลอง
10. ประเมินผลการทดลองและปรับปรุงฟังก์ชันในข้อ 4 ถึง 7 และทดลองเพิ่มเติมหากผลไม่เป็นที่น่าพอใจ
11. สรุปผลการวิจัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 เกมต่อสู้ (Fighting Game) เป็นเกมที่ผู้เล่นจะเลือกตัวละครของตนเองมา และใช้ทำการต่อสู้กับตัวละครของฝ่ายตรงข้าม โดยมีวิธีการเล่นคือทำการโจมตีโดยใช้ท่าต่างๆของตัวละครโจมตีจนพลังชีวิตของตัวละครฝ่ายตรงข้ามหมดถือเป็นการชนะหนึ่งยก และต้องชนะจำนวนยกมากกว่าครึ่งหนึ่งของจำนวนยกทั้งหมดจึงจะถือว่าเป็นการชนะหนึ่งนัด เกมต่อสู้ตั้งแต่อดีตจนถึงปัจจุบันนั้นมีหลายรูปแบบ ซึ่งแบ่งได้โดยใช้ลักษณะกราฟิกของตัวเกมเป็นหลัก ตั้งแต่เกมที่มีภาพกราฟิกเป็นสองมิติซึ่งจะเรียกว่าเกมต่อสู้แบบสองมิติ (2D) โดยตัวละครจะมีลักษณะเป็นภาพสไปรท์ (sprite) สองมิติและสามารถเคลื่อนที่ได้ในระนาบซ้ายขวาเพียงระนาบเดียว เกมที่มีกราฟิกสามมิติซึ่งเรียกว่าเกมต่อสู้แบบสามมิติ (3D) โดยตัวละครจะเป็นโมเดลสามมิติและอยู่ในฉากสามมิติ ตัวละครสามารถเคลื่อนที่ในระนาบซ้ายขวาและในระนาบที่มีความลึกได้ จนถึงเกมต่อสู้ที่รวมลักษณะทั้งสองแบบเข้าไว้ด้วยกันซึ่งเรียกอย่างไม่เป็นทางการว่าเกมแบบสองจุดห้ามิติ (2.5D) ซึ่งตัวละครเป็นโมเดลสามมิติในฉากสามมิติแต่มีลักษณะการเล่นที่เคลื่อนที่ได้เพียงในระนาบซ้ายขวาสองมิติเท่านั้น ซึ่งไม่ว่าจะเป็นแบบใดก็ล้วนแล้วแต่มีเกมที่ได้รับความนิยมเป็นอย่างสูงจากผู้เล่นวางจำหน่ายมาอย่างต่อเนื่องจนถึงปัจจุบัน ตัวอย่างเกมต่อสู้ที่ได้รับความนิยมมากแบ่งตามประเภทดังนี้

เกมต่อสู้สองมิติ (2D) เช่น เกมในชุด Street Fighter [1] เช่น เกม Street Fighter II วางจำหน่ายในปี ค.ศ. 1991 ซึ่งได้รับความนิยมสูงมากไปทั่วโลก มีการพอร์ทไปยังเครื่องเกมรุ่นต่างๆในภายหลังมากมาย และได้รับการสร้างใหม่เป็นแบบความละเอียดสูงวางจำหน่ายในเครือข่าย Xbox Live และ PlayStation Network ในปี ค.ศ. 2009 เกม Street Fighter III Third Strike (ปี ค.ศ. 1999) ซึ่งเป็นหนึ่งในเกมต่อสู้สองมิติที่ได้รับการยอมรับว่ามีระบบการเล่นที่ดีและสมดุลที่สุดเกมหนึ่ง เกม Street Fighter Zero 3 เกมในชุด The King of Fighters [2] ซึ่งภาคแรกสุดออกสู่ตลาดในปี ค.ศ. 1994 และมีการวางจำหน่ายภาคล่าสุดในปี ค.ศ. 2009 รูปที่ 1 แสดงภาพเกมต่อสู้แบบสองมิติ

รูปที่ 1 ตัวอย่างเกมต่อสู้แบบสองมิติ Street Fighter II Turbo HD Remix [www.ign.com]

เกมต่อสู้สามมิติ (3D) เช่น เกมในชุด Virtua Fighter [3] เกมในชุด Soul Calibur [4] เกมในชุด Tekken [5] เป็นต้น รูปที่ 2 แสดงภาพเกมต่อสู้แบบสามมิติ

รูปที่ 2 ตัวอย่างเกมต่อสู้แบบสามมิติ Tekken 6 [5]

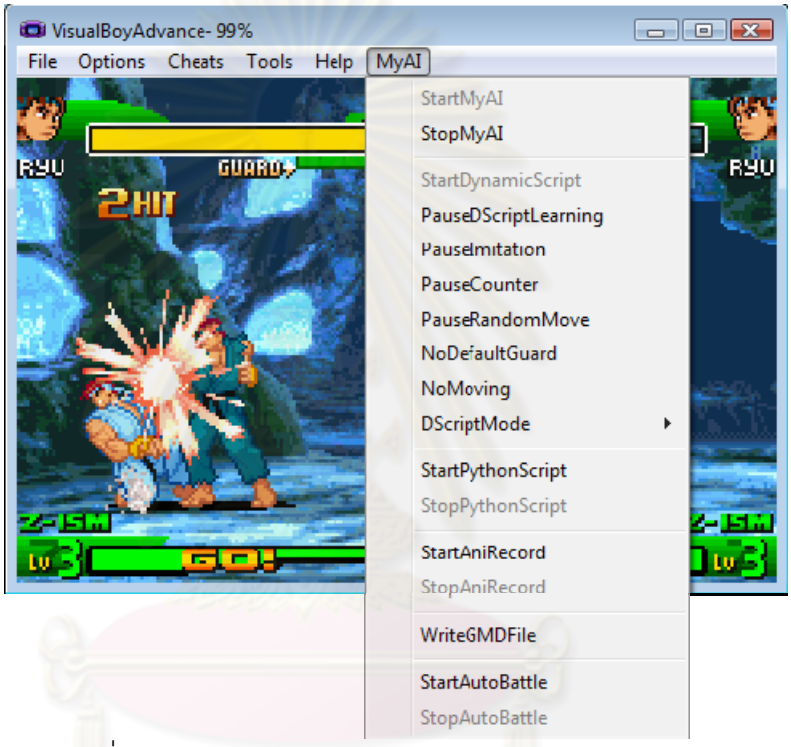
ส่วนเกมที่รวมลักษณะรูปแบบทั้งสองและสามมิติเข้าด้วยกันหรือที่เรียกอย่างไม่เป็นทางการว่าเกมแบบสองจุดห้ามิติ (2.5D) เช่น เกม Street Fighter IV เปิดตัวออกสู่ตลาดในปี ค.ศ. 2008 และได้รับความนิยมสูงมากไปทั่วโลก รูปที่ 3 แสดงภาพเกมต่อสู้ที่เป็นกึ่งสองมิติและสามมิติ

รูปที่ 3 ตัวอย่างเกมต่อสู้แบบกึ่งสองมิติและสามมิติ Street fighter IV [www.ign.com]

2.2 **เอไอ ปัญญาประดิษฐ์** (AI, Artificial Intelligence) สำหรับความหมายของเอไอนั้น สำหรับในวงการเกมแล้วหมายถึงฟังก์ชันที่ทำหน้าที่ควบคุมการทำงานของตัวละครหรือสภาพแวดล้อมภายในเกม โดยไม่คำนึงถึงวิธีการทำงานหรือวิธีการสร้าง ซึ่งจะเห็นว่าความหมายแตกต่างไปจากเอไอในวงการอื่น [6] เอไอในเกมไม่จำเป็นต้องเป็นระบบที่มีความสามารถในการเรียนรู้อย่างข่ายงานประสาทเทียม หรือขั้นตอนวิธีเชิงพันธุกรรม แต่อาจเป็นระบบที่โต้ตอบตามที่ถูกกำหนดมาโดยไม่จำเป็นต้องมีการเรียนรู้แต่อย่างใด เพียงทำให้ผู้เล่นเกิดความพึงพอใจจากการเล่นหรือเกิดความรู้สึกท้าทายเท่านั้น

2.3 **ไอ-เทม** (AI-TEM :AI Testbed in EMulator) [7] [10] เป็นสภาพแวดล้อมในการทดสอบเอไอ (Testbed) สำหรับเกม ที่มีการพัฒนาขึ้นมาจากโปรแกรมอีมูเลเตอร์ของเครื่องเกม Nintendo Gameboy Advance (GBA) ที่ชื่อว่า VisualBoyAdvance (VBA) [8] ซึ่งเป็นโปรแกรมบนคอมพิวเตอร์ที่จะทำการจำลองการทำงานของเครื่องเกม GBA โดยจะทำงานร่วมกับรอม (ROM) ของเกมที่เป็นไฟล์ข้อมูลเกมของเครื่อง GBA ทำให้สามารถเล่นเกมของเครื่อง GBA ได้บนคอมพิวเตอร์ โดยที่ไม่จำเป็นต้องมีเครื่องเกมหรือตัวเกมจริง เกมที่นำมาเล่นนั้นจะทำงานบนไอ-เทม โดยที่ตัวไอ-เทมนั้น มีความสามารถในการที่จะตรวจสอบและควบคุมค่าในหน่วยความจำตำแหน่งต่างๆของตัวอีมูเลเตอร์ได้ ทำให้สามารถควบคุมสภาพแวดล้อมต่างๆภายในเกมให้เป็นไปตามที่ผู้พัฒนาเอไอต้องการ ทำให้ผู้พัฒนาสามารถจัดการกับส่วนต่างๆของเกมได้ โดยที่ไม่จำเป็นต้องเข้าไปยุ่งเกี่ยวกับโค้ดของตัวเกม ซึ่งโดยปกติแล้วก็เป็นเรื่องที่ไม่สามารถกระทำได้เนื่องจากโค้ดของตัวเกมนั้นไม่ได้มีการเปิดเผยออกสู่สาธารณะ ตัวไอ-เทมนั้น สามารถที่จะใส่เอไอโมดูลเข้าไปเพื่อทดสอบทำงานได้ทั้งทางโค้ดของตัวไอ-เทมโดยตรง หรือใส่เป็นไพธอนสคริปต์ (Python script) ผ่านทางอินเตอร์เฟสที่จัดเตรียมไว้ก็ได้ นอกจากนั้นแล้ว การที่ผู้พัฒนาสามารถ

ทดสอบเอไอกับเกมที่มีวางจำหน่ายในท้องตลาดจริงๆได้ ทำให้มั่นใจได้ว่า เอไอที่ผ่านการทดสอบด้วยไอ-เทมมีคุณภาพเพียงพอสำหรับเกมที่มีวางจำหน่ายจริงในท้องตลาด ซึ่งเป็นจุดเด่นที่สำคัญของไอ-เทม ถึงแม้ว่าในปัจจุบัน GBA จะถือว่าเป็นเครื่องเกมที่มีอายุมากและไม่มีการพัฒนาเกมลงให้อีกแล้วนั้น อย่างไรก็ตามเกมหลายเกมบน GBA ยังคงได้รับการพิจารณาว่าเป็นเกมที่มีคุณภาพดีเทียบกับมาตรฐานเกมในปัจจุบัน สามารถใช้ทดสอบเอไอได้ รูปที่ 4 แสดงภาพหน้าจอของไอ-เทมในขณะที่รันเกมอยู่



รูปที่ 4 ไอ-เทม สภาพแวดล้อมในการทดสอบเอไอ

2.4 **ปัญญาประดิษฐ์เลียนแบบผู้เล่น หรือโกสเอไอ** (Ghost AI) เป็นระบบเอไอที่มีความสามารถในการลอกเลียนรูปแบบการเล่นของผู้เล่นและสร้างเป็นไฟล์ข้อมูลไว้ โดยที่ผู้เล่นสามารถที่จะนำไฟล์เอไอนี้ไปเล่นด้วยได้ เสมือนกับได้เล่นกับผู้เล่นที่เป็นเจ้าของเอไอตัวนั้น โดยไม่จำเป็นต้องเล่นกับผู้เล่นคนนั้นจริงๆ เป็นการเพิ่มมูลค่าและความน่าสนใจในตัวเกมให้กับผู้เล่นที่เล่นเกมนั้น โดยมีการนำไปใช้ในเกมที่มีวางจำหน่ายจริงในท้องตลาดเช่น ในเกม Virtua Fighter 4 (ค.ศ. 2002) ที่มีระบบเอไอที่สามารถให้ผู้เล่นทำการสอนวิธีการเล่นให้กับเอไอ และบันทึกข้อมูลนั้นนำไปให้ผู้เล่นคนอื่นเล่นด้วยได้ และในเกม Tekken 5 Dark Resurrection (ค.ศ. 2005) และ Tekken 6 Bloodline Rebellion (ค.ศ. 2008) ที่ตัวเกมจะทำการสร้างไฟล์เอไอที่เลียนแบบรูปแบบการเล่นของผู้เล่นขึ้นมาที่ตู้เกมที่ผู้เล่นคนนั้นเล่น โดยที่ผู้เล่นคนอื่นสามารถมาเล่นกับเอไอของผู้เล่นที่ถูกสร้างขึ้นมานี้ได้ที่ตู้เกมตู้นั้น และในเกมชุดที่ลงให้กับเครื่องเกม Sony PlayStation

Portable (PSP) ตัวเกมมีระบบที่สามารถให้ผู้เล่นอัพโหลดตัวเอไอที่สร้างขึ้นนี้ไปไว้ที่เซิร์ฟเวอร์ที่ทางผู้พัฒนาจัดเตรียมไว้ เพื่อให้ผู้เล่นคนอื่นสามารถมาดาว์นโหลดไปเล่นได้ รวมทั้งยังมีกลุ่มผู้เล่นที่รวมตัวกันแจกจ่ายไฟล์เอไอของเกมนี้ผ่านทางเว็บไซต์ธรรมดาอีกด้วย

มีงานวิจัยที่ได้นำเสนอถึงแนวทางในการพัฒนาและสร้างโกสเอไอนี้ขึ้นมา งานวิจัยของวรพจน์ [9] [10] ได้นำเสนอถึงวิธีการพัฒนาและสร้างโกสเอไอขึ้น โดยทำการวิจัยและพัฒนาบนระบบทดสอบไอ-เทม และใช้เกม Street Fighter Zero 3 Upper ของเครื่องเกม GBA เป็นกรณีศึกษา ผลการวิจัยที่ได้แสดงให้เห็นว่าโกสเอไอที่สร้างได้ สามารถเรียนรู้และลอกเลียนรูปแบบการเล่นของผู้เล่นได้ในระดับที่น่าพอใจในระดับหนึ่ง และได้วิธีการในการสร้างโกสเอไอขึ้นมาเพื่อใช้งานกับเกมต่อสู้ได้

อย่างไรก็ตามแม้เทคนิคโกสเอไอจะสามารถให้ประสบการณ์ในการเล่นที่น่าสนใจให้กับผู้เล่นได้ แต่ก็ยังมีข้อจำกัดตรงที่เอไอที่ได้นั้นไม่มีความสามารถในการพัฒนา มีรูปแบบการเล่นเพียงรูปแบบเดียวคือที่ลอกเลียนมาจากผู้เล่นที่เป็นเจ้าของเอไอเท่านั้น ซึ่งทำให้เกิดข้อจำกัดเช่นเดียวกับเอไอของเกมปกติ ที่เมื่อผู้เล่นสามารถจดจำรูปแบบการเล่นของเอไอได้แล้ว ก็จะสามารถเอาชนะเอไอได้โดยง่าย

2.5 **ไดนามิคสคริปติง** (Dynamic Scripting) [11] [12] เป็นเทคนิคเอไอสำหรับเกมที่ถูกนำเสนอโดย Spronck, P. ซึ่งได้แรงบันดาลใจมาจากวิธีการเรียนรู้แบบรีนฟอร์สเมนท์ (Reinforcement Learning) โดย Spronck นำเสนอว่าคุณภาพของเกมนั้นขึ้นอยู่กับคุณค่าทางด้านความบันเทิงของมัน เกมโดยปกตินั้นจะมีลักษณะของเอไอที่อาจจะไม่เป็นที่พอใจของผู้เล่น ซึ่งเอไอในเกมมักจะไม่ฉลาดเพียงพอที่จะสู้กับผู้เล่นได้ ผู้เล่นจึงต้องการเล่นกับผู้เล่นที่เป็นมนุษย์จริงๆ มากกว่า การปรับปรุงเอไอให้มีความฉลาดและสามารถสู้กับผู้เล่นได้เก่งขึ้นจึงเป็นสิ่งสำคัญเพื่อใช้ในกรณีที่ไม่สามารถหาผู้เล่นที่เป็นมนุษย์จริงๆได้ ซึ่งจะทำให้เกมมีความท้าทายมากขึ้นตามไปด้วย โดย Spronck ศึกษาเกมสวมบทบาท (Computer Role Playing Games (CRPGs)) ซึ่งเกมประเภทนี้ในแต่ละตาของการเล่นนั้นมีความเป็นไปได้ในการทำสิ่งต่างๆได้หลากหลาย ทำให้โดยปกติแล้วผู้พัฒนาเกมส่วนใหญ่เลือกที่จะใช้ สคริปต์ (script) เป็นตัวกำหนดคำสั่งเอไอของเกมแทน เนื่องจากสามารถที่จะควบคุมได้ง่ายกว่า และได้ผลที่มีความแน่นอนมากกว่าไม่เกิดปัญหาการแสดงออกที่มีลักษณะแปลกๆที่ไม่เหมาะสม โดยลักษณะของสคริปต์คำสั่งนี้มักจะเป็นรูปแบบตายตัวและมีความยาวและความซับซ้อนสูงซึ่งทำให้เกิดปัญหาก็คือ

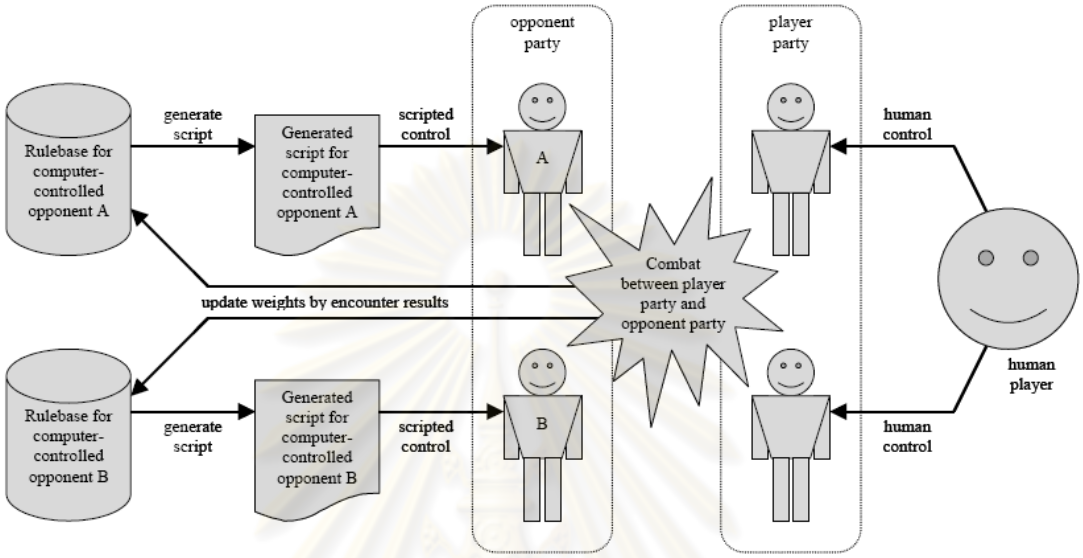
ปัญหาของความซับซ้อน (problem of complexity) และ ปัญหาของการปรับตัว (problem of adaptability) ซึ่งมีรายละเอียดดังนี้

- ปัญหาของความซับซ้อน เนื่องจากการกระทำมีความซับซ้อนมากจึงทำให้มักจะมีจุดอ่อนอยู่ทำให้ถูกผู้เล่นจับจุดได้และเอาชนะได้โดยง่าย
- ปัญหาของการปรับตัว เนื่องจากการกระทำที่ถูกกำหนดมีรูปแบบตายตัวทำให้อาจจะไม่เหมาะกับผู้เล่นที่มีระดับความสามารถต่างๆ กันไป ผู้เล่นอาจจะพบว่าเกมยากเกินไปหรือง่ายเกินไป

ปัญหาเหล่านี้ทำให้คุณค่าทางด้านความบันเทิงของตัวเกมลดลง การแก้ปัญหาเหล่านี้ทำได้โดยการนำเอาเทคนิคการเรียนรู้ของเครื่อง (Machine learning) เข้ามาช่วยซึ่งทำได้สองแบบคือ การเรียนรู้แบบออฟไลน์ (offline learning) ซึ่งเป็นการแก้ไขปัญหาก่อนที่เกมจะออกวางจำหน่าย โดยทำการทดลองปรับสมดุลของเกมไป และการเรียนรู้แบบออนไลน์ (online learning) ซึ่งจะเป็นการแก้ปัญหาหลังจากเกมได้วางจำหน่ายไปแล้ว โดยระบบจะทำการเรียนรู้ระหว่างที่เกมกำลังถูกเล่นอยู่ ซึ่งงานของ Spronck จะนำเสนอไปทางด้านการเรียนรู้แบบออนไลน์เพื่อให้เอไอมีความสามารถปรับตัวให้เก่งขึ้นเพื่อเอาชนะผู้เล่นได้ในเวลาสั้น โดยการเรียนรู้แบบออนไลน์สามารถที่จะช่วยให้เอไอจัดการซ่อมแซมจุดอ่อนในสคริปต์ของตนเอง และสามารถปรับตัวตามลักษณะการเล่นของผู้เล่นได้ ซึ่งเป็นการแก้ปัญหาของความซับซ้อนและปัญหาของการปรับตัวที่เกิดจากการใช้สคริปต์ในแง่ของการทำให้เอไอฉลาดขึ้น โดย Spronck เสนอว่าระบบเรียนรู้ที่จะนำมาใช้นั้นจะต้องมีคุณสมบัติคือ

- ความรวดเร็ว (Fast) เนื่องจากการเรียนรู้แบบออนไลน์เกิดขึ้นระหว่างการเล่นเกม ระบบการเรียนรู้จึงจำเป็นต้องมีความรวดเร็วในการทำการเรียนรู้เพื่อไม่ให้ส่งผลกระทบต่อการเล่น และการทำงานของเกม
- ประสิทธิผล (Effective) เพื่อให้ผู้เล่นได้รับความบันเทิง สคริปต์ที่มีการปรับตัวจะต้องให้ความท้าทายได้อย่างน้อยเท่ากับสคริปต์ที่ถูกออกแบบมาตามปกติ
- ทนทาน (Robust) ระบบการเรียนรู้ต้องสามารถครอบคลุมถึงความไม่แน่นอนที่มีอยู่ในระบบของตัวเกมปกติได้
- ประสิทธิภาพ (Efficient) ในเกมนั้นผู้เล่นจะได้พบกับคู่ต่อสู้เป็นจำนวนจำกัดดังนั้นระบบเรียนรู้ต้องสามารถเรียนรู้ได้จากการทดสอบเป็นจำนวนครั้งที่ไม่มาก

Spronck ได้นำเสนอระบบการเรียนรู้แบบออนไลน์ที่เรียกว่าไดนามิคสคริปติง (Dynamic Scripting) ซึ่งมีคุณสมบัติสี่ข้อตามที่กล่าวข้างต้น โดยมุ่งเน้นให้ใช้กับเกมที่มีอยู่ในท้องตลาดจริงๆ ได้ ไดนามิคสคริปติงนั้นมีขั้นตอนการทำงานตามรูปที่ 5



รูปที่ 5 การทำงานของไดนามิคสคริปติง

จากรูปที่ 5 ในการทำงานของไดนามิคสคริปติงนั้น เริ่มแรกผู้สร้างเกมจะต้องออกแบบฐานข้อมูลกฎ (Rulebase) ขึ้นมาเสียก่อน โดยกฎนั้นอยู่ในรูปแบบของสคริปต์ที่ผู้สร้างเกมใช้ความรู้เฉพาะของเกม (Domain Knowledge) กำหนดขึ้นมา จากนั้นในสถานการณ์ต่างๆระบบจะทำการเลือกกฎต่างๆจากฐานข้อมูลกฎขึ้นมาตามค่าน้ำหนัก (weight) โดยกฎที่มีค่าน้ำหนักมากจะมีโอกาสถูกเลือกไปใช้งานมาก เมื่อเล่นสู้กับผู้เล่นไปจนรู้แพ้ชนะในหนึ่งรอบการเล่นแล้ว ระบบจะทำการประเมินผลการเล่นและปรับค่าน้ำหนักของกฎแต่ละข้อตามผลการเล่นที่ประเมินได้ โดยกฎที่ทำให้ได้ผลไม่ดีก็จะถูกลงโทษโดยการลดค่าน้ำหนักลง ส่วนกฎที่ทำได้ดีก็จะได้รางวัลโดยการเพิ่มค่าน้ำหนักขึ้น ในการเล่นครั้งต่อไปศัตรูที่สร้างมาจากกฎเหล่านี้ก็จะมีความสามารถมากขึ้นและปรับตัวตามการเล่นของผู้เล่น โดยในงานของ Spronck นี้ได้นำเทคนิคนี้ไปทดสอบกับเกมสองเกมคือเกมที่ทดลองสร้างเลียนแบบเกม BALDUR'S GATE กับเกมที่อยู่ในท้องตลาดจริงๆคือเกม Neverwinter Nights ผลการทดลองที่ได้แสดงให้เห็นว่าเทคนิคไดนามิคสคริปติงสามารถนำไปใช้งานได้จริง อีกทั้งยังเป็นการปรับปรุงการแสดงออกทำให้ตัวละครหรือสภาพแวดล้อมในเกมมีการแสดงออกที่หลากหลาย เหมาะสมกับสถานการณ์ ในขณะที่ยังคงความสามารถในการแสดงออกเฉพาะสิ่งที่ถูกกำหนดไว้ก่อนแล้วได้ ทำให้ไม่เกิดปัญหาเกมมีการแสดงออกแปลกๆหรือไม่เหมาะสม

2.6 Temporal Difference Learning [13] งานวิจัยในวิทยานิพนธ์ฉบับนี้ นำเอาเทคนิคไดนามิคสคริปติงมาใช้กับการเรียนรู้ของเอไอสำหรับเกมต่อสู้ เทคนิคไดนามิคสคริปติงนั้นเป็นการนำเอาวิธีการเรียนรู้แบบรีนฟอร์สเมนท์ ซึ่งเป็นประเภทหนึ่งของการเรียนรู้แบบ Temporal difference learning มาปรับใช้กับเกม โดยแทนที่การกระทำที่ส่งผลต่อสภาพแวดล้อมของระบบ ซึ่งสำหรับเกมแล้วก็คือสภาพสถานการณ์ในเกมในช่วงเวลาหนึ่ง ด้วยสคริปต์คำสั่งสำหรับสั่งให้เอเจนท์ (agent) ในเกมแสดงออกซึ่งการกระทำต่างๆ ในการเรียนรู้แบบรีนฟอร์สเมนท์จะวัดผลการกระทำที่เอเจนท์แสดงออกด้วยฟังก์ชันการให้คะแนน การกระทำต่างๆซึ่งทำแล้วส่งผลให้สภาพแวดล้อมของระบบเปลี่ยนไปจะถูกประเมินผลและให้คะแนนรางวัล จุดมุ่งหมายของการเรียนรู้แบบรีนฟอร์สเมนท์นั้น มุ่งไปที่การหาชุดลำดับของการกระทำที่เหมาะสมที่สุดในสถานการณ์หนึ่งซึ่งทำให้เกิดการเปลี่ยนแปลงกับสถานการณ์ปัจจุบันและนำไปสู่สถานการณ์อันเป็นเป้าหมายได้ ซึ่งชุดลำดับของการกระทำนี้ จะเป็นชุดลำดับของการกระทำที่ได้คะแนนรวมจากฟังก์ชันการให้รางวัลมากที่สุดที่นำไปสู่สถานการณ์อันเป็นเป้าหมาย โดยมีการให้คะแนนกับการกระทำที่นำไปสู่สถานะเป้าหมาย และการกระทำที่เมื่อทำแล้วนำไปสู่สถานะที่สามารถไปสู่สถานะที่เป็นเป้าหมายได้ โดยจะได้คะแนนลดหลั่นลงไปตามลำดับขั้น ซึ่งได้มีการนำเอาหลักการนี้มาใช้ในขั้นตอนการเรียนรู้การกระทำเพื่อหวังผลระยะยาวของเอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้

ในการเรียนรู้แบบ Temporal difference learning ยกตัวอย่างเช่น Q learning ซึ่งเป็นการเรียนรู้แบบรีนฟอร์สเมนท์ชนิดหนึ่งนั้น เอเจนท์ไม่จำเป็นที่จะต้องมีความรู้ใดๆเกี่ยวกับผลของการกระทำที่มีต่อสภาพแวดล้อมของระบบเลย อาศัยเพียงการเรียนรู้จากการทดลองแสดงออกและถูกประเมินผลโดยการให้ค่าคะแนนรางวัลที่ได้จากการแสดงออกแต่ละการกระทำเท่านั้น ในการเรียนรู้ตามวิธี Q learning การกระทำแสดงออกต่างๆจะมีค่าคะแนนรางวัลของตัวเองอยู่แล้วโดยที่ยังไม่ได้ถูกนำมาทดลองแสดงออกและประเมินค่า ซึ่งค่าคะแนนที่มีอยู่แล้วนี้ ได้มาจากฟังก์ชันการให้รางวัลแบบประมาณค่า ซึ่งประมาณว่าการแสดงออกต่างๆน่าจะได้ค่าคะแนนเป็นเท่าไหร่ และเมื่อมีการแสดงออกการกระทำนั้นเกิดขึ้นจริง จึงนำเอาค่าคะแนนที่ได้จากฟังก์ชันการให้คะแนนจริงมาปรับปรุงค่าคะแนนที่ได้จากการประมาณอีกทีหนึ่ง โดยนำเอาค่าคะแนนประมาณของทางเลือกที่ดีที่สุดในการแสดงออกซึ่งการกระทำต่อไปมาใช้ในการปรับด้วย ซึ่งหากมีการทดลองการกระทำได้ครบทุกรูปแบบ เมื่อทำการเรียนรู้ด้วยวิธีการนี้ซ้ำไปเรื่อยๆ รูปแบบการกระทำจะปรับเข้าไปสู่รูปแบบการกระทำที่ดีที่สุดได้ด้วยตนเอง ให้ผลเป็นชุดรูปแบบการกระทำที่ดีที่สุดอันนำไปสู่สถานการณ์เป้าหมายได้ ซึ่งเป็นจุดเด่นของวิธีการเรียนรู้แบบ Q learning ในขั้นตอนการเรียนรู้นั้น การเปรียบเทียบปรับค่าคะแนนรางวัลที่ได้จากฟังก์ชันประมาณและฟังก์ชันจริง จะ

กระทำเมื่อสถานการณ์ในระบบเปลี่ยนไปจากผลของการแสดงออกการกระทำครั้งหนึ่ง โดยพิจารณาถึงทางเลือกโดยอาศัยการประมาณค่าคะแนนรางวัลที่ดีที่สุดสำหรับการกระทำที่จะแสดงออกในสถานการณ์ต่อไปด้วย เปรียบเสมือนการมองล่วงหน้าไปอีกหนึ่งสถานการณ์ การเรียนรู้สามารถที่จะเพิ่มประสิทธิภาพขึ้นได้ด้วยการมองล่วงหน้าไปมากกว่าหนึ่งสถานการณ์ โดยพิจารณาถึงทางเลือกการกระทำที่ดีที่สุดในอีกสองหรือสาม จนไปถึง n สถานการณ์ถัดไป ซึ่งจะช่วยให้การเรียนรู้มีประสิทธิภาพขึ้นได้

2.7 Learning to fight. (Grapel, T. 2004) [14] เสนอการนำเอาการเรียนรู้แบบรีนฟอร์สเมนท์ (Reinforcement Learning) ไปใช้ในการเรียนรู้ของเกม โดยมีจุดมุ่งหมายเพื่อสร้างเอเจนท์ (agent) ที่สามารถพัฒนาเทคนิคการเล่นแบบสอดคล้องกับรางวัลที่ได้จากกการเรียนรู้ได้ โดยนำไปทดสอบกับเกมต่อสู้สำหรับเครื่องเกม Microsoft Xbox ที่ชื่อว่า Tao Feng ผลการทดสอบแสดงให้เห็นว่าฟังก์ชันการให้รางวัล (Reward function) มีผลสำคัญต่อการเรียนรู้ของเอเจนท์ การเรียนรู้ที่ดีสามารถได้จากการปรับใช้ฟังก์ชันการให้รางวัลอย่างเหมาะสม รวมทั้งความเร็วในการเรียนรู้ก็ได้รับผลจากฟังก์ชันการให้รางวัลเช่นเดียวกัน การทดลองแสดงให้เห็นว่าเบื้องต้นผลการเรียนรู้ยังไม่ดีและใช้เวลาในการเรียนรู้นาน แต่เมื่อทำการปรับฟังก์ชันการให้รางวัลแล้วจะได้ผลที่ดีขึ้นและใช้เวลาในการเรียนรู้ลดลง และแสดงให้เห็นว่าเทคนิคด้านการเรียนรู้ของเครื่องสามารถนำมาใช้เพื่อการเรียนรู้สำหรับเอเจนท์ในเกมต่อสู้ได้

2.8 Automatic computer game balancing: a reinforcement learning approach. (Andrade, G. 2005) [15] เสนอระบบเกมที่เน้นไปทางด้านการทำให้ระดับความยากง่ายของเกมมีความสมดุลกับความสามารถของผู้เล่นเกม เพื่อให้เกมมีความท้าทายและสามารถเล่นได้สนุกโดยไม่ยากหรือง่ายเกินไป โดยเกมจะมีการเปลี่ยนแปลงสมดุลความยากง่ายของเกมแบบพลวัตตามความสามารถที่เพิ่มขึ้นหรือลดลงของผู้เล่น ตัวเกมจะใช้วิธีการเรียนรู้แบบรีนฟอร์สเมนท์เรียนรู้การเล่นของผู้เล่นและปรับตัวตาม ในการทดสอบจะใช้เกมต่อสู้ที่ชื่อว่า Knock'Em เป็นเกมทดสอบและให้เอเจนท์ในเกมที่ทำหน้าที่เป็นตัวทดสอบเล่นกับโปรแกรมที่ทำหน้าที่แทนผู้เล่น โดยจะมีรูปแบบการเล่นต่างกันไปตามลักษณะผู้เล่นจริง โดยทดสอบเทียบระหว่างเอเจนท์ที่มีการเรียนรู้และปรับตัวแบบพลวัตตามความสามารถของผู้เล่นเทียบกับเอเจนท์ที่เรียนรู้และเล่นแบบดีที่สุดโดยไม่ได้มีการปรับตัว โดยทั้งสองแบบใช้การเรียนรู้แบบเดียวกัน ผลการทดสอบแสดงให้เห็นว่าเอเจนท์แบบที่ไม่ได้มีการปรับตัวนั้นสามารถเอาชนะผู้เล่นได้มากกว่า ในขณะที่เอเจนท์แบบที่มีการปรับตัวนั้นจะมีอัตราการแพ้ชนะกับผู้เล่นพอๆ กัน ซึ่งตรงตามวัตถุประสงค์ของงานวิจัยที่ต้องการสร้างความสมดุลให้กับเกม งานวิจัยนี้แสดงให้เห็นถึงแนวทางหนึ่งในการสร้างเอไอสำหรับ

เกมต่อสู้ที่มีความสามารถมากกว่าเอไอปกติของเกม เพื่อทำให้ผู้เล่นสามารถเล่นได้สนุกมากขึ้น เป็นการเพิ่มมูลค่าให้กับตัวเกมทางหนึ่ง

2.9 A method for online adaptation of computer-game AI rulebase. (Thawonmas, R. 2006) [16] นำเอาเทคนิคไดนามิคสคริปติงมาใช้กับการเรียนรู้ของเกมต่อสู้ โดยนำเสนอว่าในกรณีที่ฐานข้อมูลกฎมีการออกแบบมาไม่ดี การเรียนรู้ที่ได้จากการทำไดนามิคสคริปติงจะไม่ดี และเสนอวิธีแก้ไขโดยการนำกฎที่มีค่าน้ำหนักไม่ดีต่ำกว่าค่ากำหนดค่าหนึ่งออกจากฐานข้อมูลกฎและใส่กฎใหม่ที่ทำการสุ่มขึ้นมาซึ่งยังไม่เคยมีในฐานข้อมูลกฎเข้าไปแทนเพื่อนำไปใช้เรียนรู้เพิ่ม ผลที่ได้จากการทดลองพบว่าวิธีนี้ให้ผลการเรียนรู้ที่ดีขึ้นเพราะกฎที่ทำได้ไม่ดีจะถูกลบหายไปและแทนด้วยกฎใหม่ ซึ่งแสดงให้เห็นว่าในการใช้งานไดนามิคสคริปติงนั้น การออกแบบและจัดการฐานข้อมูลกฎที่ดี สามารถช่วยให้ได้การเรียนรู้ที่ดีขึ้นได้

งานวิจัยที่ใช้ไดนามิคสคริปติง หรือการเรียนรู้แบบรีนฟอร์สเมนท์นั้น สามารถทำให้ตัวละครในเกมมีการพัฒนาขึ้นได้ในระหว่างการเล่นจริง แต่การเรียนรู้ของตัวละครในเกมนั้นเป็นการเรียนรู้จากการลองผิดลองถูกเท่านั้น ขาดการสังเกตพฤติกรรมของฝ่ายตรงข้ามเพื่อเอามาพัฒนาตนเองด้วยการเลียนแบบ ตรงกันข้ามกับปัญญาประดิษฐ์เลียนแบบผู้เล่น ซึ่งเลียนแบบได้แต่พัฒนาพฤติกรรมด้วยตนเองไม่ได้ ดังนั้นในงานวิทยานิพนธ์นี้ ทางผู้เขียนจึงมีความตั้งใจสร้างปัญญาประดิษฐ์แบบใหม่ที่รวมข้อดีของทั้งการเลียนแบบและการพัฒนาตนเองได้เอาไว้ เป็นการขจัดจุดอ่อนของปัญญาประดิษฐ์ทั้งสองวิธีข้างต้นในเวลาเดียวกัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
ระบบต้นแบบ
โกสเอไอที่ปรับอัตราการออกท่าที่เลียนแบบได้ตามเวลาจริง

จากงานวิจัยเรื่องโกสเอไอ พบว่าเทคนิคโกสเอไอให้ผลเป็นเอไอที่มีความสามารถในการเลียนแบบรูปแบบการเล่นของผู้เล่นได้ดีในระดับหนึ่ง อย่างไรตาม โกสเอไอไม่สามารถที่จะพัฒนารูปแบบการเล่นให้เกินไปกว่ารูปแบบที่ลอกเลียนมาจากผู้เล่นได้ ซึ่งเป็นข้อจำกัดเช่นเดียวกันกับเอไอของเกมปกติ

ในการพัฒนาสร้างระบบเอไอที่สามารถเรียนรู้รูปแบบการเล่นของผู้เล่นและสามารถพัฒนารูปแบบการเล่นของตัวเองได้อย่างอัตโนมัติตามเวลาจริงนั้น ทางผู้เขียนได้สร้างระบบเอไอต้นแบบที่มีความสามารถในการปรับตัว (ในที่นี้คือปรับอัตราการออกท่าที่ได้ลอกแบบผู้เล่นมา) ตามรูปแบบการเล่นของผู้เล่นได้ตามเวลาจริงขึ้นมาเพื่อเป็นระบบทดสอบเบื้องต้น ก่อนลงมือสร้างระบบซึ่งสามารถคิดค้นท่าเองได้อีกทีหนึ่ง ระบบปัญญาประดิษฐ์เบื้องต้นนี้มีชื่อว่าระบบ AGAI (Adaptive Ghost AI) [17] ในบทนี้จะเป็นการอธิบายรายละเอียดต่างๆรวมทั้งผลการทดลองของระบบ AGAI นี้

3.1 เครื่องมือและสภาพแวดล้อมที่ใช้ในการพัฒนา

ในการสร้างและทดสอบระบบ AGAI จะทำการทดสอบบนระบบทดสอบไอ-เทม และใช้เกม Street Fighter Zero 3 Upper เป็นเกมทดสอบ

3.2 ภาพรวมของระบบเอไอ

ระบบต้นแบบ AGAI ที่พัฒนาขึ้นในส่วนนี้พัฒนาขึ้นโดยอาศัยแนวคิดจากระบบโกสเอไอ [9] [10] มาเป็นพื้นฐาน โดยมีความสามารถพื้นฐานเช่นเดียวกันกับระบบโกสเอไอ คือสามารถเลียนแบบรูปแบบการเล่นของผู้เล่นที่เอไอทำการเล่นด้วยได้ แต่จะแตกต่างกับระบบโกสเอไอปกติ ระบบ AGAI ที่พัฒนาขึ้นมานี้ อัตราการออกท่าทางต่างๆในสถานการณ์หนึ่ง จะเปลี่ยนแปลงไปจากรูปแบบเดิมมาเป็นการพิจารณาจากผลลัพธ์ที่ท่าทางที่แสดงออกไปนั้นสามารถทำได้แทน ซึ่งหมายความว่าท่าการกระทำที่สามารถให้ผลลัพธ์ในการเล่นที่ดี (สำหรับเกมต่อสู้คือการสร้างความเสียหายแก่ตัวละครฝ่ายตรงข้ามได้) จะมีโอกาสในการแสดงออกมากกว่าท่าการกระทำที่ให้ผลลัพธ์ในการเล่นไม่ดี (แสดงออกแล้วถูกฝ่ายตรงข้ามสร้างความเสียหายให้) ซึ่งต่างจากโกสเอไอ

ปกติที่แสดงออกท่าตามความถี่ที่ผู้เล่นเคยใช้ท่านั้นโดยไม่ได้สนใจถึงผลที่เกิดจากการใช้ท่านั้นๆ เลย AGAI สามารถใช้รูปแบบการเล่นที่ลอกเลียนมาจากผู้เล่นได้อย่างมีประสิทธิภาพมากขึ้นจากการเรียนรู้ผลของการกระทำรูปแบบต่างๆ และปรับตัวให้ได้การกระทำที่เหมาะสมกับสถานการณ์ในการเล่นเกมได้

3.3 **รายละเอียดของสถานการณ์ในเกม**

ในการเล่นเกมนั้น เอไอจะทำงานโดยใช้การอ้างอิงกับสถานการณ์ที่เกิดขึ้นในเกม ซึ่งรายละเอียดที่นำมาใช้เป็นข้อกำหนดของสถานการณ์ในเกมนั้นมีดังนี้

- ข้อมูลระยะห่างระหว่างตัวละครฝ่ายเอไอและตัวละครฝ่ายผู้เล่นในแนวแกน x และแนวแกน y
- ข้อมูลระยะห่างระหว่างตัวละครของฝ่ายเอไอกับกระสุนของฝ่ายใดก็ตามทั้งในแนวแกน x และแนวแกน y
- สถานะของตัวละครฝ่ายเอไอ

ข้อมูลข้างต้นนี้จะถูกเข้ารหัสเป็นค่าชื่อว่า Casekey ซึ่งเป็นค่าตัวแทนของสถานการณ์ในเกม สำหรับรายละเอียดตัวอย่างการเข้ารหัสสถานการณ์ในเกมเพื่อให้ได้ค่า Casekey นั้น มีการอธิบายในภาคผนวก ข ซึ่งในหัวข้อ 1 อธิบายถึงรายละเอียดของสถานการณ์ในเคสและในหัวข้อ 2 อธิบายถึงการเข้ารหัสสถานการณ์ในเกม

3.4 **โครงสร้างของเคส** (Case)

ข้อมูลของสถานการณ์ในเกมจะถูกนำมาสร้างเป็นเคส (case) ซึ่งเป็นข้อมูลของสถานการณ์ในเกมที่เอไอจะนำไปใช้ในการเล่นเกมต่อไป โครงสร้างของเคสนั้นจะประกอบไปด้วย

- Casekey เป็นค่าที่เป็นตัวแทนแสดงถึงสถานการณ์ในเกมของเคสนั้น
- ข้อมูลของท่าที่ทำการลอกแบบจากผู้เล่นมา โดยแต่ละท่าจะมีค่าความถี่ (Frequency) และค่าประสิทธิภาพ (Performance score) เป็นของตัวเอง โดยที่ค่าความถี่คือ จำนวนครั้งที่ผู้เล่นที่เอไอทำการลอกแบบใช้ท่านั้นในสถานการณ์การเล่นเกมที่แทนด้วย Casekey ส่วนค่าประสิทธิภาพคือค่าคะแนนที่ท่านั้นสามารถทำได้ในการใช้ท่านั้นในสถานการณ์ที่แทนด้วย Casekey รูปที่ 6 แสดงตัวอย่างโครงสร้างเคสของ AGAI ท่าจะ

แสดงอยู่ในรูปแบบของหมายเลขของท่าและตามด้วยค่าความถี่และค่าประสิทธิภาพของท่านั้นในวงเล็บ

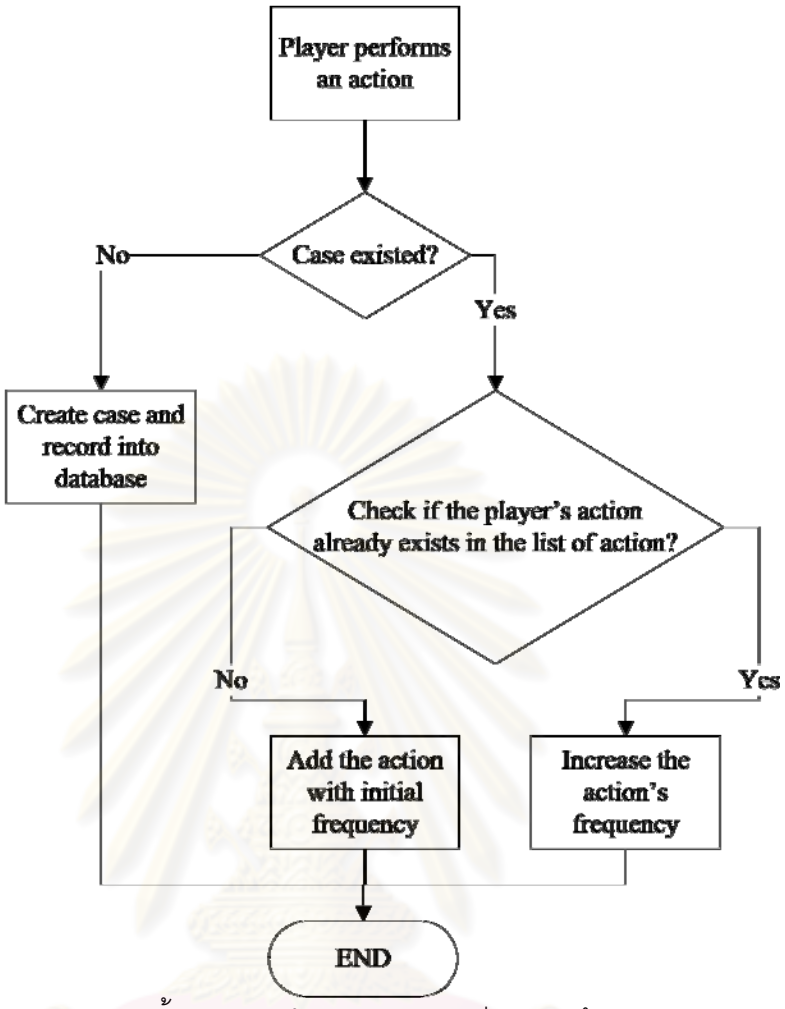
CASEKEY = 7414277
Imitated actions : 1[3,10], 3[15,2], 22[5,30], 63[7,64]

รูปที่ 6 ตัวอย่างโครงสร้างของเคสของ AGAI

3.5 รายละเอียดการทำงานของเอไอ

ในส่วนของการทำงานของ AGAI นั้น สามารถแบ่งได้เป็นสองส่วนซึ่งทำงานเกี่ยวเนื่องกัน ส่วนแรกคือส่วนการลอกเลียนแบบผู้เล่นที่เอไอจะทำการสังเกตพฤติกรรมของผู้เล่นและสร้างเคสขึ้นมา เพื่อลอกเลียนรูปแบบการเล่นของผู้เล่น และอีกส่วนหนึ่งคือส่วนการเรียนรู้ของเอไอที่จะประเมินผลลัพธ์จากการใช้ท่าของผู้เล่นและของเอไอเอง และให้คะแนนท่านั้นเพื่อเอาไปใช้ในการเลือกท่ามาใช้ทำการเล่นเกมต่อไป

3.5.1 ส่วนการลอกเลียนแบบผู้เล่น ในระหว่างการเล่นเกมนั้น เอไอจะทำการสังเกตพฤติกรรมของผู้เล่นว่ามีการแสดงออกตอบสนองต่อสถานการณ์ในเกมอย่างไร เมื่อผู้เล่นทำการแสดงออกการกระทำใดๆก็ตาม เอไอจะทำการตรวจสอบการกระทำและสถานการณ์ของเกมในเวลานั้น และนำเอาข้อมูลที่ได้ไปสร้างเป็นเคสขึ้น โดยจะมีค่า Casekey ตามที่ได้จากการเข้ารหัสสถานการณ์ของเกมในเวลานั้น และเพิ่มท่าที่ผู้เล่นใช้เข้าไปในส่วนเก็บท่าที่ลอกเลียนผู้เล่นมา โดยจะมีค่าความถี่เริ่มต้นของท่านั้นเท่ากับหนึ่ง ซึ่งหมายความว่าในสถานการณ์นั้น เอไอพบว่าผู้เล่นเคยใช้ท่าที่บันทึกเอาไว้นี้มาแล้วทั้งหมดหนึ่งครั้ง ซึ่งในครั้งต่อไปเมื่อเอไอพบกับสถานการณ์ที่ได้บันทึกเอาไว้นี้อีกครั้ง เอไอจะตรวจสอบว่าท่าที่ผู้เล่นใช้นั้นได้เคยมีการบันทึกเอาไว้หรือยัง หากยังท่านั้นจะถูกบันทึกเพิ่มเข้าไปในส่วนเก็บท่าที่ลอกเลียนผู้เล่นมาพร้อมกับค่าความถี่เริ่มต้น แต่หากท่านั้นได้ถูกบันทึกเอาไว้ในส่วนเก็บท่าของสถานการณ์นี้แล้ว ความถี่ของท่านั้นจะเพิ่มขึ้นด้วยค่าเท่ากับหนึ่ง รูปที่ 7 แสดงผังงานการทำงานของส่วนลอกเลียนแบบผู้เล่นของ AGAI



รูปที่ 7 ผังงานแสดงขั้นตอนการทำงานส่วนลอกเลียนแบบผู้เล่นของ AGAI

3.5.2 ส่วนการเรียนรู้ของเอไอ ในระหว่างการเล่นเกม เมื่อมีความเสียหายเกิดขึ้นกับตัวละครของฝ่ายใดตาม เอไอจะทำการตรวจสอบสถานการณ์ที่เกิดขึ้น เนื่องจากทุกสถานการณ์การกระทำที่ผู้เล่นแสดงออกมานั้นจะถูกตรวจสอบและสร้างเป็นเคสขึ้นตั้งแต่ที่การกระทำนั้นเริ่มแสดงออก และการกระทำใดๆที่ตัวละครฝ่ายเอไอแสดงออกนั้น ได้มาจากการเล่นตามเคสที่ลอกเลียนแบบมาจากผู้เล่นทั้งสิ้น ดังนั้นสถานการณ์ที่เกิดความเสียหายขึ้นนี้จะถูกบันทึกเอาไว้เป็นเคสแล้วอย่างแน่นอน เอไอจึงไม่จำเป็นต้องตรวจสอบว่าสถานการณ์ที่เกิดขึ้นนี้เคยมีการบันทึกเอาไว้หรือไม่ ส่วนความเสียหายที่เกิดขึ้นนั้น เอไอจะพิจารณาแยกเป็นสองกรณีคือ

- กรณีที่เอไอเป็นฝ่ายได้รับความเสียหาย ท่าที่ผู้เล่นใช้สร้างความเสียหายให้กับเอไอจะได้รับคะแนนเพิ่มขึ้น และท่าที่เอไอใช้จะถูกลดคะแนนลง
- กรณีที่ผู้เล่นเป็นฝ่ายได้รับความเสียหาย ท่าที่ผู้เล่นใช้จะถูกลดคะแนนลงและท่าที่เอไอใช้จะได้รับคะแนนเพิ่มขึ้น

รูปที่ 8 แสดงผังงานการทำงานของส่วนเรียนรู้ของ AGAI

รูปที่ 8 ผังงานแสดงขั้นตอนการทำงานของส่วนการเรียนรู้ของ AGAI

3.5.3 การคิดค่าคะแนนประสิทธิภาพเมื่อท่าใช้สร้างความเสียหายแก่ฝ่ายตรงข้ามได้ ค่าคะแนนประสิทธิภาพของท่านั้น คือค่าคะแนนที่ให้กับท่าต่างๆเมื่อท่านั้นสามารถสร้างความเสียหายให้กับฝ่ายตรงข้ามได้ โดยค่าคะแนนประสิทธิภาพนี้จะคิดจากค่าความเสียหายที่ท่านั้นสร้างให้กับฝ่ายตรงข้าม โดยที่ระบบสามารถรู้ค่าความเสียหายนี้ได้ โดยอาศัยการตรวจสอบจากค่าในหน่วยความจำของ testbed AI-TEM ในตำแหน่งที่เก็บข้อมูลค่าความเสียหายในแต่ละรอบของการทำงาน ซึ่งข้อมูลค่าความเสียหายหลังจากโดนการโจมตีแต่ละครั้งที่ได้จากตัวเกมโดยตรงนั้น จะแสดงค่าเป็นค่าความเสียหายของท่านั้นเองบวกรวมกับค่าความเสียหายของท่าก่อนหน้าที่เกิดขึ้นเป็นการโจมตีต่อเนื่องกันกับท่าปัจจุบัน (ถ้ามีท่าก่อนหน้า) และค่าที่ตรวจสอบได้จะกลับเป็นศูนย์เมื่อไม่มีการทำความเสียหายเกิดขึ้นในเกมแล้ว ซึ่งในการนำเอาค่าความเสียหายมาคิดเป็นค่าคะแนนของท่าในเคสนั้น จะคิดโดยใช้หลักการเดียวกับข้อมูลที่ได้จากเกมเลย โดยมีรายละเอียดดังนี้

ถ้ามีการสร้างความเสียหายด้วยท่า A B และ C ตามลำดับ ท่า A ได้ค่าคะแนนจากค่าความเสียหายเป็น 1 ท่า B สร้างความเสียหายได้เท่ากับ 2 จะได้ค่าคะแนนจากค่าความเสียหายที่ท่า B สร้างได้เอง รวมกับค่าคะแนนจากความเสียหายที่ท่า A สร้างได้เป็น 2+1=3 ท่า C สร้างความเสียหายได้ 4 จะได้ค่าคะแนนโดยคิดบวกจากค่าคะแนนของท่าก่อนหน้าเช่นกันเป็น 3+4=7 จะเห็นว่าค่าคะแนนที่นำมาจากความเสียหายที่แต่ละท่าสร้าง จะได้จากค่าคะแนนของท่าเองบวกทบกับค่าคะแนนของท่าก่อนหน้าเข้าไปด้วย ยิ่งมีการโจมตีแบบต่อเนื่องยาว ท่าที่อยู่ด้านหลังก็จะยิ่งได้ค่าคะแนนมากตามไปด้วย

การคิดค่าคะแนนแบบบวกทบจากค่าคะแนนของท่าก่อนหน้าแบบนี้มีข้อดีคือ จะช่วยให้เห็นได้ชัดว่าตัวละครสามารถเล่นได้ดีหรือไม่โดยดูจากค่าคะแนนรวมที่ตัวละครสามารถทำได้ ยิ่งตัวละครมีการโจมตีต่อเนื่องได้มากหรือใช้ท่าโจมตีที่ก่อให้เกิดความเสียหายมากแล้ว ก็จะมีค่าคะแนนโดยรวมที่ทำได้มาก โดยต่างจากการคิดค่าคะแนนแบบปกติตามจริงที่แต่ละท่าทำได้ ซึ่งค่าคะแนนรวมจะเท่ากันหมดทุกรอบการเล่น โดยจะมีค่าเท่ากับค่าพลังชีวิตของตัวละครนั่นเอง

3.5.4 การใช้ข้อมูลเคสในการเล่นเกม ในระหว่างการเล่นเกม เอไอจะทำการตรวจสอบสถานการณ์ของตัวละครฝ่ายผู้เล่นว่าตรงกับสถานการณ์ที่ได้บันทึกเอาไว้หรือไม่ โดยอาศัยการเปรียบเทียบค่า Casekey ของสถานการณ์ในเกมกับค่า Casekey ของเคสที่ได้บันทึกเอาไว้

- กรณีที่สถานการณ์ในเกมไม่ตรงกับสถานการณ์ที่ได้บันทึกเอาไว้เลย เอไอจะทำการแสดงออกท่าการกระทำแบบสุ่มออกไป ทั้งนี้ท่าการกระทำที่สุ่มแสดงออกไปนั้น ไม่ว่าจะก่อให้เกิดผลเช่นใด ก็จะไม่นำมาคิดผลต่อการเรียนรู้ของเอไอ ท่าการกระทำที่สุ่มแสดงออกไป มีเป้าหมายเพียงเพื่อดูผลการตอบสนองของผู้เล่นที่มีต่อการกระทำนั้น แล้วนำเอามาสร้างเป็นข้อมูลเคสเพื่อใช้ในการลอกเลียนแบบผู้เล่นเท่านั้น

- กรณีที่สถานการณ์ในเกมตรงกับสถานการณ์ที่ได้มีการบันทึกเอาไว้ หมายความว่าตัวละครฝ่ายผู้เล่นอยู่ในสถานการณ์เดียวกันกับตัวละครของฝ่ายเอไอในขณะที่ได้มีการบันทึกข้อมูลสถานการณ์เอาไว้ เอไอจะทำการเลือกท่าจากส่วนเก็บข้อมูลท่าเลียนแบบผู้เล่นที่บันทึกเอาไว้มาใช้ เพื่อเป็นการเลียนแบบรูปแบบการเล่นของผู้เล่น ในการเลือกท่าเพื่อมาแสดงออกนั้นนั้น จากเดิมในระบบโกสเอไอจะเลือกท่าตามความน่าจะเป็นจากจำนวนความถี่ที่ผู้เล่นทำการใช้ท่าแต่ละท่าออกมาในสถานการณ์นั้น ท่าที่ผู้เล่นใช้บ่อยในสถานการณ์นั้น จะมีโอกาสถูกเลือกออกมาใช้มากกว่า เพื่อเป็นการเลียนแบบลักษณะ

การออกท่าของผู้เล่น สำหรับระบบ AGAI การเลือกท่ามาใช้นั้นจะมีรายละเอียดต่างออกไป โดยจะมีรายละเอียดดังนี้

1. เอไอจะทำการคำนวณความน่าจะเป็นของท่าที่จะถูกเลือกจากค่าความถี่ที่ได้บันทึกเอาไว้ของแต่ล่ะท่า ค่าความน่าจะเป็นของท่าที่ได้จากการใช้ค่าความถี่ของท่านั้นมาคำนวณนี้เรียกว่า P_f เป็นค่าความน่าจะเป็นที่ท่าจะถูกเลือกเมื่อพิจารณาถึงการเลียนแบบผู้เล่นเพียงอย่างเดียว

2. เอไอจะทำการคำนวณความน่าจะเป็นของท่าที่จะถูกเลือกจากค่าคะแนนของท่านั้นที่ได้บันทึกเอาไว้แต่ล่ะท่า ค่าความน่าจะเป็นของท่าที่ได้จากการใช้ค่าคะแนนของท่านั้นมาคำนวณนี้เรียกว่า P_w เป็นค่าความน่าจะเป็นที่ท่าจะถูกเลือกเมื่อพิจารณาถึงประสิทธิภาพที่ท่านั้นสามารถทำได้เพียงอย่างเดียว

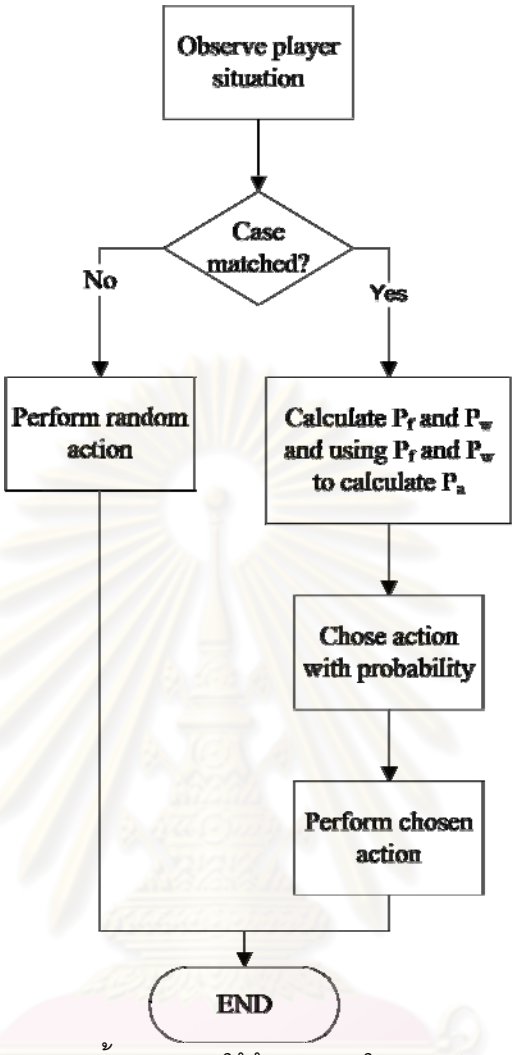
3. เอไอจะทำการคำนวณค่าความน่าจะเป็นของแต่ละท่าที่จะถูกเลือก โดยอาศัยค่า P_f และ P_w ของแต่ล่ะท่า โดยจะใช้ค่า P_f มาปรับค่า P_w ตามสมการ (1) เรียกค่าความน่าจะเป็นใหม่นี้ว่า P_a ค่า P_a นี้เป็นค่าที่เอไอนำเอามาใช้ในการตัดสินใจเลือกท่ามาแสดงออก

$$P_a = (1-C)P_w + (C)P_f \quad (1)$$

ค่า C ที่ใช้ในการทดลองมีค่าเท่ากับ 0.1

รูปที่ 9 แสดงผังงานการใช้ข้อมูลเคสในการเล่นเกมของ AGAI

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

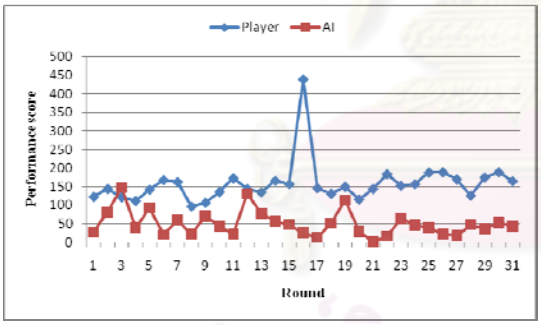


รูปที่ 9 ผังงานแสดงขั้นตอนการใช้ข้อมูลเคสในการเล่นเกมของ AGAI

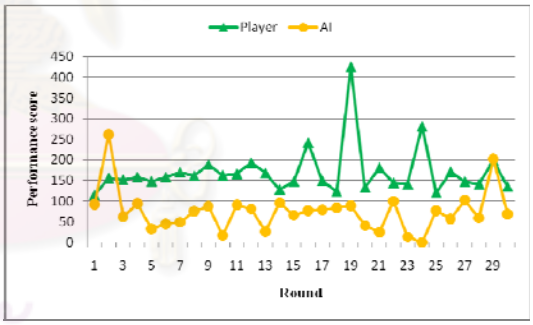
3.5.5 การเลือกท่าจากการพิจารณาค่าความน่าจะเป็น เมื่อได้ค่า P_a ของแต่ละท่า ซึ่งเป็นค่าความน่าจะเป็นที่ท่านั้นจะถูกเลือกแล้ว เอไอจะเลือกท่าออกมาใช้งานโดยค่า P_a จะถูกแปลงให้อยู่ในอัตราส่วนร้อยละ ซึ่งจะได้เป็นค่าร้อยละที่แต่ละท่าจะถูกเลือก แล้วทำการแบ่งช่วงตัวเลขจากค่าร้อยละเต็ม 100 ให้กับท่าต่างๆ ตามค่าร้อยละของท่านั้น จากนั้นทำการสุ่มตัวเลขที่มีค่าระหว่าง 1 ถึง 100 ขึ้นมาค่าหนึ่ง และดูว่าตัวเลขนั้นตกอยู่ในช่วงของท่าไหน ท่านั้นก็จะถูกเลือกออกมาใช้งาน ยกตัวอย่างเช่น ในเคสมีท่า A B และ C เมื่อคำนวณค่า P_a และแปลงเป็นค่าร้อยละแล้วมีค่า 45 20 และ 35 ตามลำดับ เมื่อทำการกำหนดช่วงให้กับท่า จะได้ว่าช่วงตัวเลข 1-45 เป็นท่า A ช่วง 46-65 เป็นท่า B และช่วง 66-100 เป็นท่า C หากสุ่มตัวเลขได้ค่า 78 ท่า C ก็จะถูกเลือกมาใช้เป็นต้น

3.6 การทดลอง

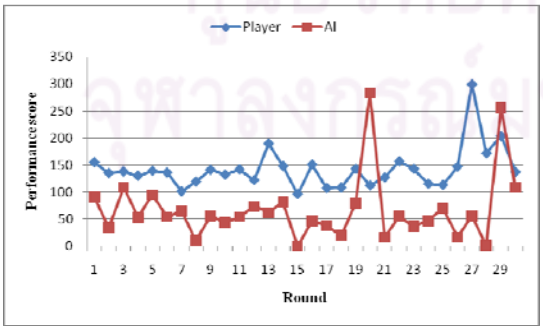
ในส่วนของการทดลองนั้น ได้ทำการทดลองโดยให้ผู้ทดลองทำการเล่นกับเอไอสองแบบ คือเอไอเลียนแบบผู้เล่นแบบออนไลน์ปกติที่ไม่มีการปรับอัตราการออกท่า (GAI) ซึ่งจะเหมือนกับโกสเอไอปกติ เพียงแต่มีการเลียนแบบท่าเป็นแบบออนไลน์ และเอไอเลียนแบบผู้เล่นแบบออนไลน์ที่มีการปรับอัตราการออกท่า (AGAI) โดยมีผู้ทดลองจำนวน 5 คน ทำการเล่นกับเอไอทั้งสองแบบเป็นจำนวนแบบละ 10 นัด โดยผู้ทดลองมีประสบการณ์และความสามารถในการเล่นเกมต่อสู้แตกต่างกันไปทั้งผู้ทดลองที่เพิ่งเริ่มเล่นและผู้ทดลองที่เชี่ยวชาญในการเล่น ทำการทดลองโดยไม่ให้ผู้ทดลองทราบว่าตนกำลังแข่งกับเอไอแบบใดอยู่ ในแต่ละนัดจะต้องเอาชนะฝ่ายตรงข้ามให้ได้ 3 ยก จึงจะถือเป็นการชนะ 1 นัด ผลการทดลองแสดงดังรูปที่ 10 ตารางที่ 1 และ ตารางที่ 2 โดยกราฟจะแสดงค่าประสิทธิภาพที่ผู้เล่นและเอไอทำได้จากการสร้างความเสียหายแก่ฝ่ายตรงข้ามในแต่ละยกของการเล่นเกม ตารางที่ 1 แสดงการเปรียบเทียบค่าประสิทธิภาพระหว่าง GAI และ AGAI ตารางที่ 2 แสดงค่าร้อยละของจำนวนครั้งที่เอไอมีการนำเอาข้อมูลเคสที่เรียนรู้ได้มาใช้งานในแต่ละยกของการเล่นเทียบกับร้อยละของจำนวนครั้งที่เอไอทำการสุ่มออกท่าในแต่ละยกของการเล่น และจำนวนเคสทั้งหมดที่เอไอเรียนรู้ได้



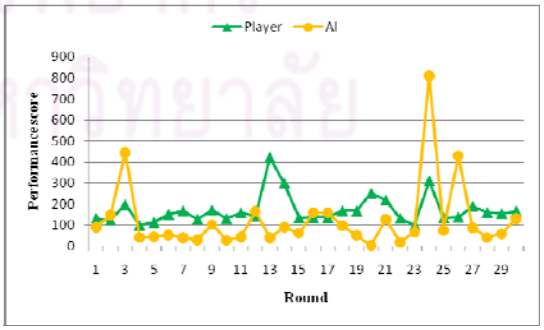
Player 1 – GAI



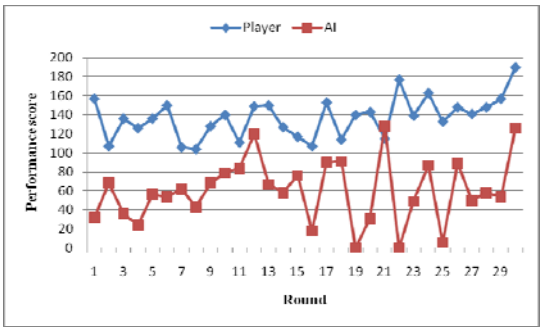
Player 2 – AGAI



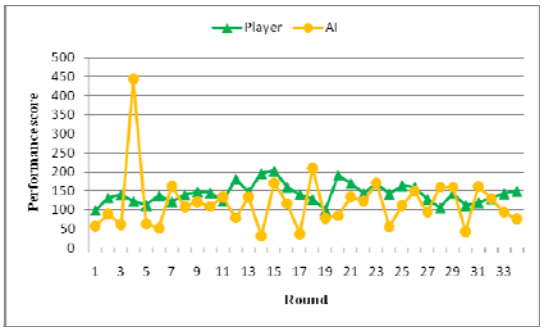
Player 2 – GAI



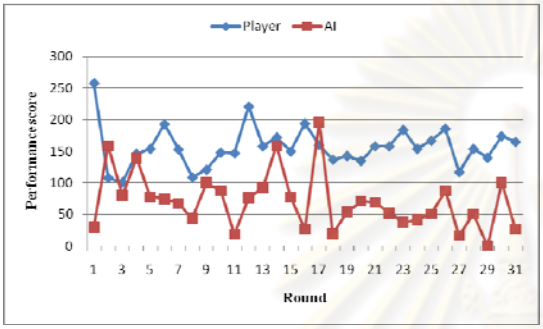
Player 2 – AGAI



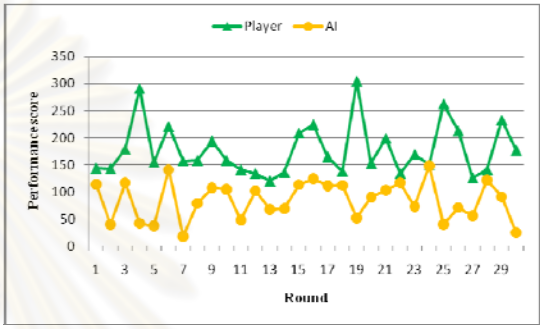
Player 3 – GAI



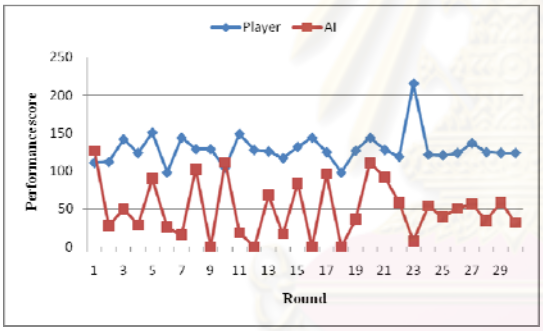
Player 3 – AGAI



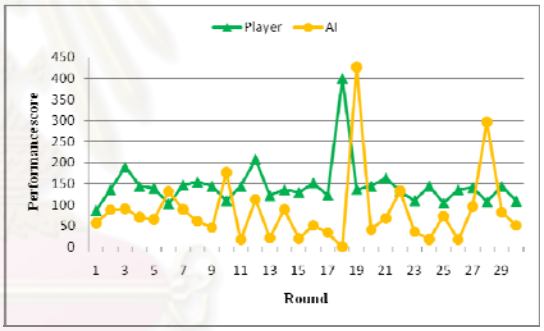
Player 4 – GAI



Player 4 – AGAI



Player 5 – GAI



Player 5 - AGAI

รูปที่ 10 กราฟแสดงผลการทดลองของระบบเอไอต้นแบบ AGAI

ตารางที่ 1 เปรียบเทียบค่าประสิทธิภาพระหว่าง GAI และ AGAI

Player	AVERAGE PERFORMANCE SCORE	
	Player's score – GAI's score	Player's score – AGAI's score
1	107.93	95.03
2	75.00	47.10
3	76.93	24.85
4	86.41	93.16
5	79.23	59.46
AVERAGE	85.10	63.92

ตารางที่ 2 จำนวนครั้งที่มีการนำเคสที่เรียนรู้ได้มาใช้งานเทียบกับจำนวนครั้งที่ออกท่าแบบสุ่มและจำนวนเคสที่เรียนรู้ได้ของ GAI และ AGAI

Player	AVERAGE % OF RANDOM ACTION PER ROUND		AVERAGE % OF CASE UTILIZATION PER ROUND		TOTAL NUMBER OF LEARNED CASE	
	GAI	AGAI	GAI	AGAI	GAI	AGAI
1	8.15	9.51	91.85	90.49	205.00	285.00
2	6.61	9.07	93.39	90.93	207.00	293.00
3	5.94	6.10	94.06	93.90	146.00	250.00
4	7.56	6.64	92.44	93.36	160.00	226.00
5	7.00	9.67	93.00	90.33	108.00	156.00
AVERAGE	7.05	8.20	92.93	91.80	165.20	242.00

3.7 สรุปผลการทดลอง

จากผลการทดลอง ในส่วนการเล่นระหว่างผู้เล่นกับเอไอที่ทำการเลียนแบบผู้เล่นอย่างเดียวโดยไม่มีการปรับอัตราการออกท่า (GAI) นั้น ค่าประสิทธิภาพที่ผู้เล่นทำได้จากการสร้างความเสียหายให้แก่เอไอ แสดงถึงแนวโน้มที่จะเพิ่มมากขึ้นตามระยะเวลาการเล่นที่ผ่านไป ซึ่งแสดงให้เห็นว่าผู้เล่นที่เป็นมนุษย์นั้นสามารถเรียนรู้พัฒนารูปแบบการเล่นของตัวเองได้เร็วกว่าเอไอ ในขณะที่ค่าประสิทธิภาพที่เอไอทำได้จากการสร้างความเสียหายให้แก่ผู้เล่น พบว่ามีค่าอยู่ในลักษณะเปลี่ยนแปลงขึ้นลงอยู่โดยตลอด ซึ่งอธิบายได้ว่า น่าจะมาจากการที่เอไอทำการเลียนแบบรูปแบบการเล่นของผู้เล่น ซึ่งเมื่อเลียนแบบแล้วทำให้เอไอสามารถเล่นได้ดีขึ้น กราฟจึงมีช่วงขึ้น ผู้เล่นเองก็มีการปรับปรุงรูปแบบการเล่นของตัวเอง ซึ่งทำให้เอไอต้องทำการเลียนแบบรูปแบบการเล่นของผู้เล่นเพิ่มเติม กราฟจึงแสดงช่วงตกลง เป็นเช่นนี้สลับกันไป

สำหรับในส่วนของการเล่นระหว่างผู้เล่นกับเอไอที่ทำการเลียนแบบผู้เล่นและมีการปรับอัตราการออกท่า (AGAI) เมื่อเปรียบเทียบระหว่างค่าประสิทธิภาพที่ผู้เล่นทำได้กับค่าประสิทธิภาพที่เอไอทำได้ ก็ยังคงแสดงผลออกมาเป็นลักษณะเดียวกันกับผลที่ได้จากการเล่นระหว่างผู้เล่นกับเอไอเลียนแบบที่ไม่มีการปรับอัตราการออกท่า ซึ่งสามารถอธิบายได้ด้วยเหตุผลแบบเดียวกัน คือเอไอเรียนรู้การเล่นได้ดีขึ้น ทำให้ผู้เล่นเกิดการปรับรูปแบบวิธีการเล่น ผู้เล่นยังคงเล่นได้ดีกว่าเอไอ อย่างไรก็ตาม ค่าประสิทธิภาพที่เอไอแบบมีการปรับตัวสามารถทำได้นั้น สูงขึ้นใกล้เคียงกับผู้เล่นมากกว่าเอไอแบบที่ไม่มีการปรับตัวอย่างเห็นได้ชัด แสดงให้เห็นว่าเอไอสามารถปรับตัวโดยอาศัยประสบการณ์ที่ได้จากการเล่น และเล่นได้ดีขึ้นจริง

นอกจากผลการทดลองแล้ว ผู้เข้าร่วมการทดลองยังได้ให้ความเห็นต่อเอไอที่ใช้ทำการทดลอง ผู้เข้าทดลองทุกคนเห็นด้วยว่า เอไอที่มีการปรับตัวนั้นสามารถเล่นได้ดีกว่าเอไอที่ไม่มีการ

ปรับตัวอย่างเห็นได้ชัด มีรูปแบบการเล่นที่ดีกว่าและยืดหยุ่นมากกว่าเอไอที่ไม่มีการปรับตัว และให้ความท้าทายในการเล่นที่ดีกว่า

จากผลการทดลองและความคิดเห็นของผู้ทดลอง สามารถสรุปได้ว่า เอไอแบบที่มีการปรับตัวนั้น มีความสามารถในการเล่นดีกว่าเอไอแบบที่ไม่มีการปรับตัว อย่างไรก็ตาม เนื่องจากท่าต่างๆที่เอไอสามารถใช้ได้ยังคงจำกัดอยู่เพียงท่าที่ลอกเลียนมาจากผู้เล่นเท่านั้น ไม่มีการใช้ท่าใหม่อื่นแต่อย่างใด ทำให้รูปแบบการเล่นที่เอไอสามารถเล่นได้ยังคงมีจำกัดอยู่ ผู้เขียนได้นำเอาข้อมูลที่ได้มาเป็นต้นแบบของเอไอที่มีการปรับตัวโดยทั้งเลียนแบบรูปแบบการเล่นของผู้เล่นและพัฒนารูปแบบการเล่นของตัวเองได้และนำเสนอในบทต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

ปัญญาประดิษฐ์แบบปรับตัวได้อัตโนมัติ

4.1 เครื่องมือและสภาพแวดล้อมที่ใช้ในการพัฒนา

วิทยานิพนธ์นี้จะทำการสร้างและทดสอบระบบเอไอสำหรับเกมต่อสู้ที่สามารถปรับตัวตามผู้เล่นได้แบบอัตโนมัติในเวลาจริง ทำการวิจัยทดลองบนระบบทดสอบไอ-เทมโดยใช้เกม Street Fighter Zero 3 Upper (STZ3) (ในทวีปอเมริกาจะใช้ชื่อ Street Fighter Alpha 3 Upper (STA3)) ของเครื่องเกม Nintendo Gameboy Advance (GBA) ซึ่งเป็นเกมที่ได้รับการยอมรับว่ามีระบบการเล่นดีที่สุดเกมหนึ่งในเกมประเภทเกมต่อสู้ของเครื่อง GBA ในการวิจัยทดลองจะทำโดยใช้ข้อมูลตัวละครในเกมชื่อ Ryu เพียงตัวเดียวเพื่อเป็นกรณีศึกษา รูปที่ 11 แสดงตัวอย่างภาพจากเกม

รูปที่ 11 Street Fighter Zero 3 Upper

4.2 ภาพรวมของระบบเอไอ

ระบบเอไอในวิทยานิพนธ์ฉบับนี้นั้นได้แรงบันดาลใจและเป็นการพัฒนาต่อเนื่องมาจากระบบโกสเอไอ (Ghost AI) [9][10] ซึ่งเป็นระบบเอไอที่เอไอจะลอกเลียนรูปแบบการเล่นของผู้เล่นมาใช้งานในการเล่นเกม และ ระบบ Adaptive ghost AI [17] ซึ่งเป็นระบบที่พัฒนามาจากหลักการของโกสเอไออีกทีหนึ่ง โดยพัฒนาให้ตัวปัญญาประดิษฐ์มีความสามารถในการเรียนรู้ปรับ

อัตราการใช้ท่าที่จำมาแบบออนไลน์ได้ จากเดิมที่เป็นเพียงแบบออฟไลน์อย่างเดียว สำหรับระบบเอไอของวิทยานิพนธ์ฉบับนี้ เอไอจะทำการเรียนรู้และปรับปรุงรูปแบบการเล่นของตัวเองให้ดีขึ้นอัตโนมัติตามลักษณะการเล่นของผู้เล่นในเวลาจริง โดยจะนำเอาเทคนิคไดนามิคสคริปติ้งมาใช้ประกอบกับเคสเบสรีซันนิง (Case based reasoning) อย่างง่าย ระบบใหม่นี้จะประกอบไปด้วยส่วนประกอบหลักสองส่วน ส่วนแรกคือส่วนที่ทำการลอกเลียนรูปแบบการเล่นของผู้เล่น ซึ่งจะช่วยให้เอไอสามารถเรียนรู้จากผู้เล่นได้ว่าควรจะทำตัวเช่นไรในสถานการณ์ในเกมแบบต่างๆที่เกิดขึ้นในระหว่างการเล่นเกม เพื่อที่จะสามารถเล่นเกมได้อย่างมีประสิทธิภาพ และส่วนที่สองคือ ส่วนที่จะทดลองการกระทำใหม่อื่นๆ ที่ไม่ได้ลอกเลียนมาจากผู้เล่น เพื่อดูผลที่ได้ และสามารถนำมาใช้ในการเล่นเกมได้อย่างมีประสิทธิภาพ ช่วยให้รูปแบบการเล่นของเอไอ ไม่ถูกจำกัดอยู่แต่เพียงรูปแบบเดิมๆ ที่สามารถสังเกตจุดอ่อนและเอาชนะได้โดยง่าย รูปที่ 12 แสดงองค์ประกอบหลักของระบบเอไอในวิทยานิพนธ์นี้

รูปที่ 12 องค์ประกอบหลักของเอไอ

4.3 รายละเอียดของสถานการณ์ในเกม

ในการเล่นเกมนั้น เอไอจะทำงานโดยใช้การอ้างอิงกับสถานการณ์ที่เกิดขึ้นในเกม ซึ่งรายละเอียดที่นำมาใช้เป็นข้อกำหนดของสถานการณ์ในเกมนั้นมีดังนี้

- ข้อมูลระยะห่างในแกน x ระหว่างตัวละครของฝ่ายเอไอกับตัวละครของฝ่ายผู้เล่น
- ข้อมูลระยะห่างในแกน y ระหว่างตัวละครของฝ่ายเอไอกับตัวละครของฝ่ายผู้เล่น
- ข้อมูลระยะห่างในแกน x ระหว่างตัวละครของฝ่ายเอไอกับกระสุนของฝ่ายใดก็ตาม
- ข้อมูลระยะห่างในแกน y ระหว่างตัวละครของฝ่ายเอไอกับกระสุนของฝ่ายใดก็ตาม

- สถานะของตัวละครฝ่ายเอไอ ได้แก่ สถานะที่สามารถถูกโจมตีได้ สถานะที่ไม่สามารถถูกโจมตีได้ สถานะมึน สถานะที่กำลังถูกทำความเสียหาย และสถานะที่กำลังออกท่าโจมตีอยู่

ข้อมูลทั้งหมดข้างต้นนี้เป็นข้อมูลที่แสดงถึงสถานการณ์ในเกมในเวลาใดๆ ข้อมูลที่แสดงสถานการณ์ในเกมทั้งหมดนี้ จะถูกเข้ารหัสให้กลายเป็นค่า ซึ่งมีชื่อว่า Casekey ซึ่งค่านี้เป็นตัวแทนที่แสดงถึงสถานการณ์ในเกม รายละเอียดการหาค่า Casekey ใช้การเข้ารหัสในรูปแบบเดียวกันกับระบบต้นแบบในบทที่ 3 ซึ่งมีการอธิบายในภาคผนวก ข หัวข้อที่ 1 และ 2

4.4 **โครงสร้างของเคส** (Case)

เคส (case) เป็นข้อมูลที่เอไอจะนำมาใช้ในการเล่นเกม โดยจะเป็นข้อมูลที่บอกถึงสถานการณ์ในเกม และท่าต่างๆที่เอไอสามารถเลือกมาใช้ได้ในสถานการณ์นั้น ซึ่งจะมีโครงสร้างประกอบไปด้วย

- Casekey เป็นหมายเลขตัวแทนของเคสนั้น แสดงถึงข้อมูลสถานการณ์ในเกม
- ข้อมูลท่าที่ทำการลอกแบบจากผู้เล่นมา (Imitated actions) โดยแต่ละท่าจะมีค่าประสิทธิภาพ (performance score) ของตัวเอง
- ข้อมูลท่าที่จะใช้ในการทดลองท่าใหม่ (Experimental actions) โดยแต่ละท่าจะมีค่าประสิทธิภาพ (performance score) ของตัวเอง

รูปที่ 13 แสดงตัวอย่างโครงสร้างของเคส ท่าจะแสดงในรูปแบบของหมายเลขของท่าและตามด้วยค่าประสิทธิภาพของท่านั้นในวงเล็บ

CASEKEY = 7414277
Imitated actions : 1[3], 3[50], 22[35], 63[2]
Experimental actions : 9[1], 17[9], 32[4], 55[39]

รูปที่ 13 โครงสร้างของเคส

4.5 **รายละเอียดการทำงานของเอไอ**

สำหรับรายละเอียดการทำงานของเอไอนั้น เนื่องจากเอไอทำงานโดยอาศัยการตอบสนองต่อสถานการณ์ในเกมตามเวลาจริง จะขออธิบายเป็นส่วนๆ โดยอาศัยการเทียบเคียงกับสิ่งที่เกิดขึ้นในระหว่างการเล่นเกมต่อสู้ เพื่อให้เห็นภาพได้ชัดเจน

4.5.1 สังเกตและบันทึกข้อมูลเบื้องต้นระหว่างความสัมพันธ์ของสถานการณ์ในเกมและการกระทำที่ตัวละครแสดงออก ในช่วงเริ่มต้น เมื่อเริ่มเล่นเกมนั้น เอไอจะไม่มีความรู้ใดๆที่จะใช้ในการเล่นเกมเลย ดังนั้นเอไอจะทำการตรวจสอบการแสดงออกของตัวผู้เล่นและตัวเอไอเองว่ามีการแสดงออกการกระทำแบบใด เมื่อสถานการณ์ในเกมเป็นแบบใด (เนื่องจากเอไอไม่มีความรู้สำหรับใช้เล่นเกมเลย การแสดงออกในขั้นตอนนี้จะเป็นการสุ่มท่ามาแสดงออกเพื่อดูผลของท่านั้นที่มีต่อผู้เล่น)

ตัวค่า Casekey ที่ได้จากการเข้ารหัสและการกระทำที่ตัวละครแสดงออกซึ่งเป็นการแสดงออกโดยอ้างอิงกับสถานการณ์ในเกมในเวลานั้น จะถูกจัดเก็บเป็นคู่อันดับของค่า Casekey และการกระทำ โดยจะแบ่งเป็นสองแบบคือ การกระทำที่เอไอแสดงออกอ้างอิงกับสถานการณ์ในเกมในเวลานั้น และการกระทำที่ผู้เล่นแสดงออกอ้างอิงกับสถานการณ์ในเกมในเวลานั้น ซึ่งทุกครั้งที่ตัวละครฝ่ายเอไอหรือผู้เล่นมีการแสดงออกท่าใดๆ ก็ตาม จะเกิดคู่อันดับของค่า Casekey และการกระทำขึ้นหนึ่งคู่อันดับ ซึ่งจะเป็นของฝ่ายเอไอหรือผู้เล่นก็ได้ แล้วแต่ว่าเป็นการกระทำของตัวละครฝ่ายใด โดยการสร้างข้อมูลคู่อันดับนี้จะถูกสร้างไปเรื่อยๆ และบันทึกลงในฐานข้อมูลอาร์เรย์ตามลำดับของการเกิดก่อนหลังตลอดการเล่นเกม โดยจะบันทึกแยกกันระหว่างส่วนของเอไอและส่วนของผู้เล่น รูปที่ 14 แสดงลักษณะการเก็บข้อมูลคู่อันดับของฝ่ายเอไอและผู้เล่น คู่อันดับ 1 คือคู่อันดับที่เกิดแรกสุด คู่อันดับ X คือคู่อันดับที่เกิดล่าสุด

A_Casekey1 A_Action1	A_Casekey2 A_Action2	A_Casekey3 A_Action3	…….. …….	A_CasekeyX A_ActionX

P_Casekey1 P_Action1	P_Casekey2 P_Action2	P_Casekey3 P_Action3	…….. …….	P_CasekeyX P_ActionX

รูปที่ 14 ข้อมูลคู่อันดับ (Casekey , การกระทำ) ของเอไอ (A) และผู้เล่น (P)

4.5.2 สร้างข้อมูลเคส (case) สำหรับใช้ในการเล่นเกม เมื่อตัวละครของฝ่ายเอไอหรือผู้เล่นมีการแสดงออกใดๆก็ตาม หากการกระทำนั้นก่อให้เกิดความเสียหายขึ้นต่อตัวละครของฝ่ายตรงข้าม เอไอจะทำการสร้างเคสขึ้นเพื่อเป็นข้อมูลสำหรับใช้ในการเล่นเกมซึ่งมีรายละเอียดดังนี้

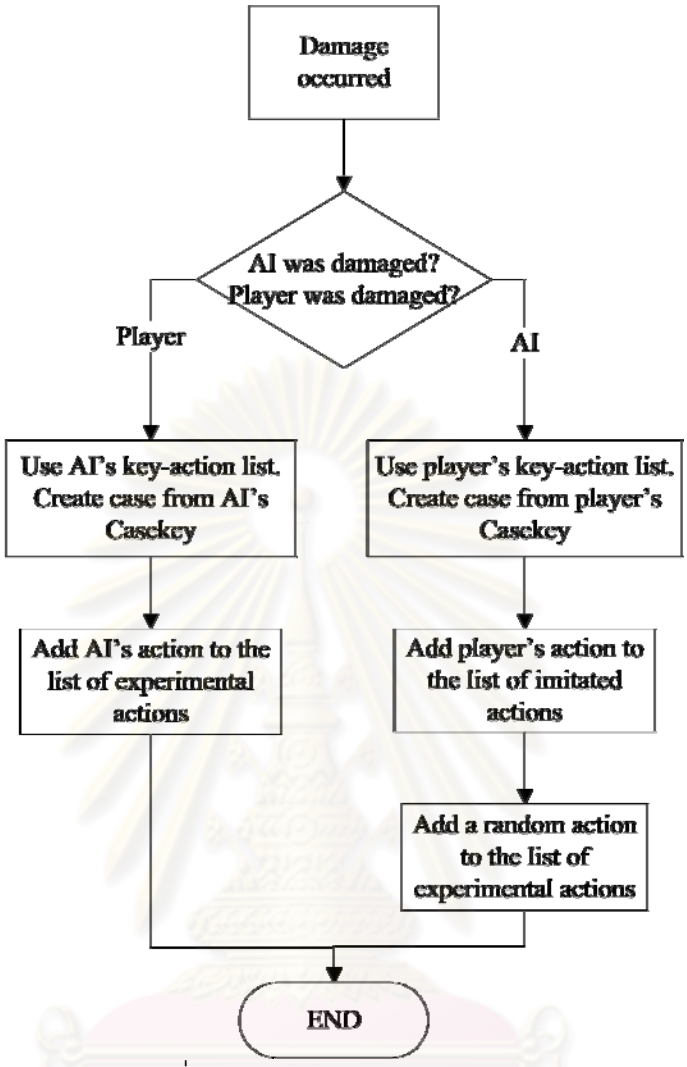
เมื่อเกิดความเสียหายขึ้นกับตัวละครของฝ่ายเอไอหรือผู้เล่นก็ตาม นั่นหมายความว่าความเสียหายนั้นเกิดจากสถานการณ์และการกระทำอันดับล่าสุดของฝ่ายตรงข้ามที่เกิดขึ้นในเกม

ซึ่งได้มีการถูกบันทึกเอาไว้แล้ว เอไอจะนำเอาคู่อันดับของ Casekey และการกระทำอันดับล่าสุดมาตรวจสอบ โดยขึ้นกับว่าความเสียหายนั้นเกิดขึ้นกับฝ่ายใด หากความเสียหายเกิดขึ้นกับฝ่ายเอไอ เอไอจะตรวจสอบคู่อันดับล่าสุดของฝ่ายผู้เล่น ในทางกลับกัน หากความเสียหายเกิดขึ้นกับฝ่ายผู้เล่น เอไอจะตรวจสอบคู่อันดับล่าสุดของตัวเอไอเองแทน ซึ่งจะทำให้ได้ข้อมูลสถานการณ์ที่เกิดความเสียหายและการกระทำที่ทำให้เกิดความเสียหายในสถานการณ์นั้น จากนั้นเอไอจะนำเอาข้อมูลจากคู่อันดับนั้นมาสร้างเป็นเคสขึ้น โดยมีรายละเอียดดังนี้

- สร้างเคสขึ้นโดยมีค่า Casekey เป็นตัวแทนของเคสนั้น
- กรณีที่เอไอเป็นฝ่ายได้รับความเสียหาย ท่าที่ผู้เล่นใช้สร้างความเสียหายให้กับเอไอจะถูกเพิ่มเข้าไปในส่วนเก็บท่าที่ลอกเลียนผู้เล่นมา พร้อมกับค่าประสิทธิภาพเริ่มต้นมีค่าเป็นค่าของคะแนนที่ผู้เล่นได้จากการโจมตีเอไอด้วยท่านั้น นอกจากนั้น จะมีการสุ่มท่าท่าหนึ่งขึ้นมาซึ่งจะยังไม่มีค่าประสิทธิภาพและเพิ่มเข้าไปในส่วนเก็บท่าสำหรับทดลองใช้ เพื่อให้ทางฝ่ายเอไอทดลองใช้กับผู้เล่นในอนาคต เมื่อเจอสถานการณ์นี้อีกในครั้งต่อไป เพื่อที่ฝ่ายเอไอจะหลีกเลี่ยงความเสียหายแบบที่เจอนี้ในอนาคต
- กรณีที่ผู้เล่นเป็นฝ่ายได้รับความเสียหาย ท่าที่เอไอใช้สร้างความเสียหายให้กับผู้เล่น จะถูกเพิ่มเข้าไปในส่วนเก็บท่าที่จะทดลองใช้ พร้อมกับค่าประสิทธิภาพเริ่มต้นเป็นค่าของคะแนนที่เอไอได้จากการโจมตีผู้เล่นด้วยท่านั้น

กระบวนการการสร้างข้อมูลเคสนี้แสดงดังผังงานรูปที่ 15 ข้อมูลเคสที่ถูกสร้างขึ้นมานี้จะถูกบันทึกลงไปในฐานข้อมูลเคส โดยหลังจากกระบวนการนี้เอไอจะเริ่มมีเคสไว้ใช้เป็นข้อมูลในการเล่นเกมแล้ว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 15 กระบวนการสร้างข้อมูลเคส

4.5.3 ปรับปรุงข้อมูลเคส เมื่อเอไอเริ่มมีข้อมูลเคสไว้ใช้สำหรับการเล่นเกมแล้ว ก็จะใช้ข้อมูลนั้นทำการเล่นเกมต่อไป เมื่อเกมดำเนินต่อไปและมีความเสียหายเกิดขึ้นกับตัวละครฝ่ายใดฝ่ายหนึ่ง เอไอจะทำการตรวจสอบคู่อันดับของ Casekey และ การกระทำ โดยอาศัยหลักการเดียวกันกับในขั้นตอนการสร้างข้อมูลเคส โดยก่อนอื่นเอไอจะทำการตรวจสอบ Casekey ก่อนว่ามีเคสของ Casekey นี้ในฐานข้อมูลเคสหรือยัง หากยังไม่มี เอไอจะเข้าสู่กระบวนการสร้างข้อมูลเคสต่อไป หรือหากมีแล้ว เอไอจะทำการปรับปรุงข้อมูลในเคสนั้น โดยมีรายละเอียดดังนี้

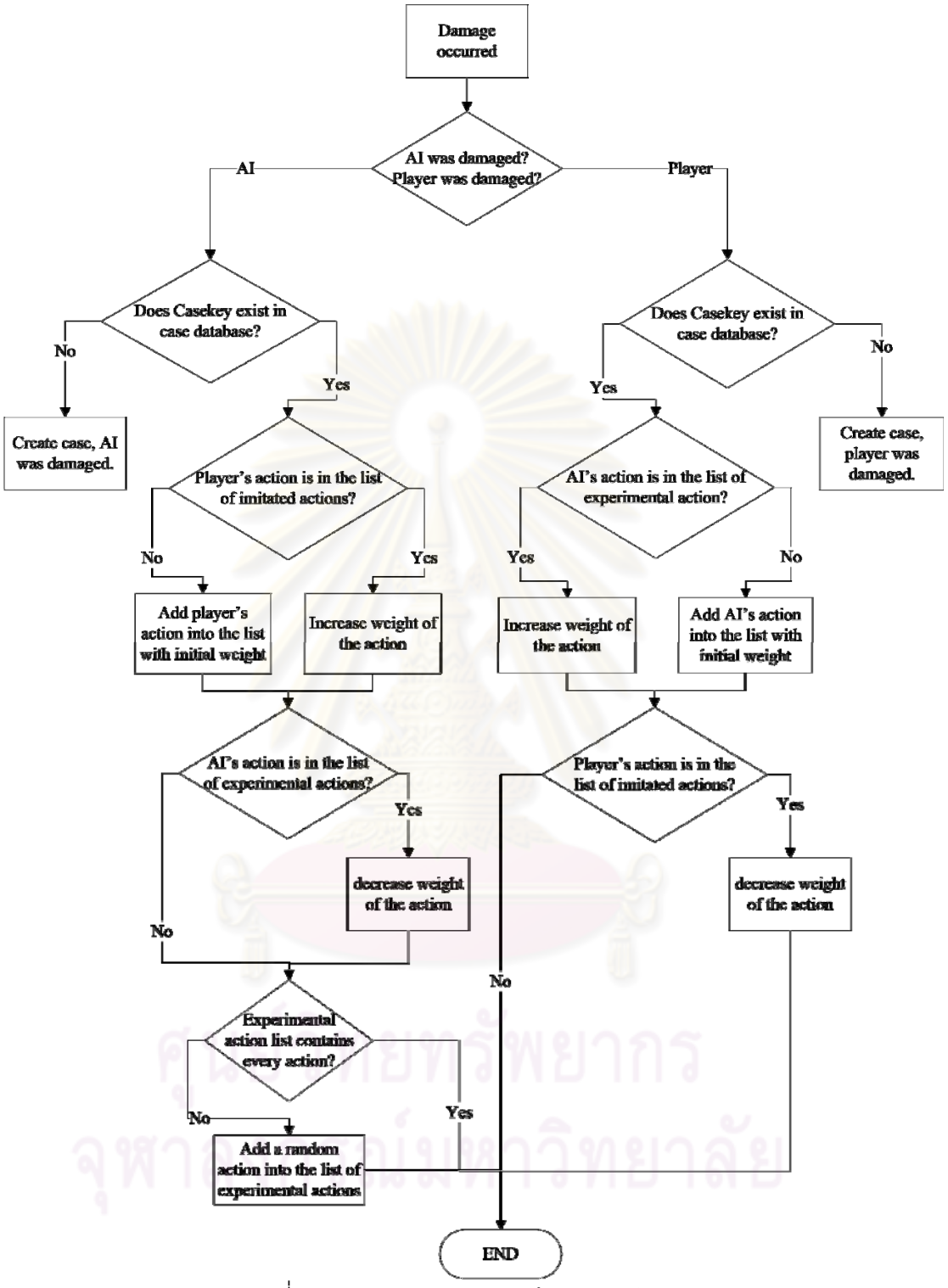
- กรณีเอไอเป็นฝ่ายได้รับความเสียหาย เอไอจะตรวจสอบท่าที่ผู้เล่นใช้สร้างความเสียหายให้กับเอไอว่ามีในส่วนของท่าที่ลอกเลียนมาจากผู้เล่นหรือยัง หากยังไม่มี เอไอจะทำการเพิ่มท่านั้นพร้อมค่าประสิทธิภาพขั้นต้นเข้าไป ในกรณีที่ท่านั้นมีอยู่ในส่วนท่าที่ลอกเลียนมาจากผู้เล่นแล้ว ค่าประสิทธิภาพของท่านั้นจะถูกเพิ่มด้วยค่าคะแนนจากการที่ท่านั้น

สร้างความเสียหายแก่เอไอได้ จากนั้นเอไอจะทำการตรวจสอบท่าที่เอไอใช้แล้วได้รับความเสียหายว่ามีในส่วนของท่าที่จะทดลองใช้หรือไม่ ถ้าไม่มีเอไอจะไม่ทำการบันทึกท่านั้นเพิ่มเข้าไป เนื่องจากเป็นท่าที่ทำให้ตัวเอไอได้รับความเสียหาย แต่หากมี ค่าประสิทธิภาพของท่านั้นจะถูกลดลงด้วยค่าคะแนนจากความเสียหายที่เอไอได้รับ และหากเป็นท่าใหม่ที่ยังไม่มีค่าประสิทธิภาพ ก็จะได้ค่าประสิทธิภาพเท่ากับค่าลบของค่าประสิทธิภาพขั้นต้น นอกจากนั้นหากในส่วนเก็บท่าสำหรับทดลองใช้ยังมีท่าไม่ครบทุกท่า จะมีการสุ่มท่าท่าหนึ่งขึ้นมา โดยจะยังไม่มีค่าประสิทธิภาพและเพิ่มเข้าไปในส่วนเก็บท่าสำหรับทดลองใช้เพื่อทดลองใช้กับผู้เล่นในอนาคตเมื่อเจอสถานการณ์นี้อีกในครั้งต่อไป เพื่อที่จะหลีกเลี่ยงความเสียหายแบบที่เจอนี้ในอนาคต

- กรณีผู้เล่นเป็นฝ่ายได้รับความเสียหาย เอไอจะตรวจสอบท่าที่เอไอใช้สร้างความเสียหายให้กับผู้เล่นว่ามีในส่วนของท่าสำหรับทดลองใช้หรือไม่ หากยังไม่มี เอไอจะทำการเพิ่มท่านั้นพร้อมค่าประสิทธิภาพขั้นต้นเข้าไป ในกรณีที่ท่านั้นมีอยู่ในส่วนท่าสำหรับทดลองใช้แล้ว ค่าประสิทธิภาพของท่านั้นจะถูกเพิ่มด้วยค่าคะแนนจากความเสียหายที่ท่านั้นสร้างแก่ผู้เล่นได้ และหากเป็นท่าใหม่ที่ยังไม่มีค่าประสิทธิภาพ จะได้ค่าประสิทธิภาพเท่ากับค่าประสิทธิภาพขั้นต้น จากนั้นเอไอจะทำการตรวจสอบท่าที่ผู้เล่นใช้แล้วได้รับความเสียหายว่ามีในส่วนของท่าที่ลอกเลียนมาจากผู้เล่นหรือไม่ ถ้าไม่มีเอไอจะไม่บันทึกท่านั้นเพิ่มเข้าไป เนื่องจากเป็นท่าที่ทำให้ตัวผู้เล่นได้รับความเสียหาย แต่หากมี ค่าประสิทธิภาพของท่านั้นจะถูกลดลงด้วยค่าคะแนนจากความเสียหายที่ผู้เล่นได้รับ

การปรับปรุงข้อมูลในเคสนี้จะช่วยให้เคสได้รับการปรับปรุงเรื่องของท่าที่สามารถใช้ได้และค่าประสิทธิภาพของท่า ให้สะท้อนผลลัพธ์ที่สามารถทำได้ในการเล่นเกม รูปที่ 16 แสดงผังงานของกระบวนการปรับปรุงข้อมูลในเคสนี้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 16 กระบวนการปรับปรุงข้อมูลภายในเคส

4.5.4 การเลือกท่ามาใช้ในสถานการณ์ต่างๆ ในหัวข้อข้างต้นนั้น อธิบายถึงกระบวนการสร้างเคส และกระบวนการปรับปรุงข้อมูลในเคส ซึ่งเกิดขึ้นเมื่อมีความเสียหายเกิดขึ้นกับตัวละครของฝ่ายผู้เล่นหรือฝ่ายเอไอฝ่ายใดฝ่ายหนึ่ง ซึ่งในการที่จะเกิดความเสียหายได้นั้น แสดงว่าตัว

ละครของฝ่ายผู้เล่นหรือฝ่ายเอไอจะต้องมีการแสดงออกถึงการกระทำออกมา ในหัวข้อนี้จะอธิบายถึงวิธีการที่เอไอเลือกท่ามาแสดงออกเมื่อพบกับสถานการณ์ต่างๆในเกม โดยมีรายละเอียดดังนี้

ในระหว่างการเล่นเกม เอไอจะทำการพิจารณาสถานการณ์ของเกมจากตัวละครของทั้งฝ่ายเอไอและผู้เล่นในขณะนั้น ว่าตรงกับสถานการณ์ที่เอไอเคยได้บันทึกไว้หรือไม่ โดยอาศัยการเปรียบเทียบค่า Casekey ของสถานการณ์ในเกม ณ เวลานั้น กับค่า Casekey ของเคสที่มีการสร้างเอาไว้ ว่ามีค่าตรงกันบ้างหรือไม่

- กรณีที่สถานการณ์ในเกมในเวลานั้น ไม่ตรงกับสถานการณ์ที่เคยมีการบันทึกไว้เลย เอไอจะทำการสุ่มท่าขึ้นมาท่าหนึ่ง แล้วแสดงการกระทำท่านั้นออกไป
- กรณีที่สถานการณ์ในเกมตรงกับสถานการณ์ที่เคยมีการบันทึกไว้ สามารถแบ่งได้เป็นสามรูปแบบ คือ กรณีที่ตัวละครฝ่ายผู้เล่นอยู่ในสถานการณ์เดียวกับที่มีการบันทึกเอาไว้ กรณีที่ตัวละครฝ่ายเอไออยู่ในสถานการณ์เดียวกับที่มีการบันทึกเอาไว้ กรณีที่ตัวละครทั้งฝ่ายผู้เล่นและเอไออยู่ในสถานการณ์เดียวกับที่มีการบันทึกเอาไว้ โดยจะมีรายละเอียดดังนี้
  - กรณีที่ตัวละครฝ่ายผู้เล่นอยู่ในสถานการณ์เดียวกับที่เคยมีการบันทึกเอาไว้ แสดงว่าฝ่ายผู้เล่นอยู่ในสถานการณ์เดียวกับเอไอตอนที่ถูกผู้เล่นสร้างความเสียหายให้ เอไอจะทำการเลือกท่าออกมาจากส่วนของท่าที่ลอกเลียนมาจากผู้เล่นในเคส โดยในการเลือกท่านั้น จะทำการเลือกออกมาโดยใช้ค่าความน่าจะเป็นที่ได้จากค่าประสิทธิภาพของท่านั้น โดยท่าที่มีค่าประสิทธิภาพมากกว่า จะมีความน่าจะเป็นที่จะถูกเลือกออกมาใช้มากกว่าท่าที่มีค่าประสิทธิภาพน้อยกว่า ซึ่งเหตุผลที่ใช้การเลือกท่าโดยใช้ค่าความน่าจะเป็นแทนที่จะเป็นท่าที่มีค่าประสิทธิภาพมากที่สุดนั้นเป็นเพราะปัจจัยด้านความไม่แน่นอนของตัวเกม รูปแบบการเล่นหนึ่งๆอาจใช้ไม่ได้ผลทุกครั้ง ในทางกลับกันรูปแบบการเล่นที่ลองใช้แล้วไม่ได้ผล ก็ไม่จำเป็นที่จะต้องไม่ได้ผลทุกครั้งเสมอไป ดังนั้นจึงใช้การเลือกท่าโดยอาศัยความน่าจะเป็นแทน เพื่อจะได้มีโอกาสได้ทดลองใช้ท่าอื่นๆ เพื่อขยายรูปแบบการเล่นออกไป โดยเมื่อเลือกท่าได้แล้ว จะทำการแสดงออกท่านั้นต่อไป การแสดงออกในลักษณะนี้จะทำให้เอไอสามารถที่จะเล่นโดยใช้รูปแบบการเล่นเดียวกับที่ผู้เล่นเคยใช้ได้ ส่วนอัตราการออกท่าตามความถี่ที่ผู้เล่นเคยออกท่านั้น ได้ถูกตัดทิ้งไปอย่างสิ้นเชิงเนื่องจากการทดลองกับ AGAI ทำให้ทราบว่า การออกท่าโดยคำนึงถึงค่าประสิทธิภาพในการใช้งานจริงนั้นให้ผลลัพธ์ที่ดีกว่า

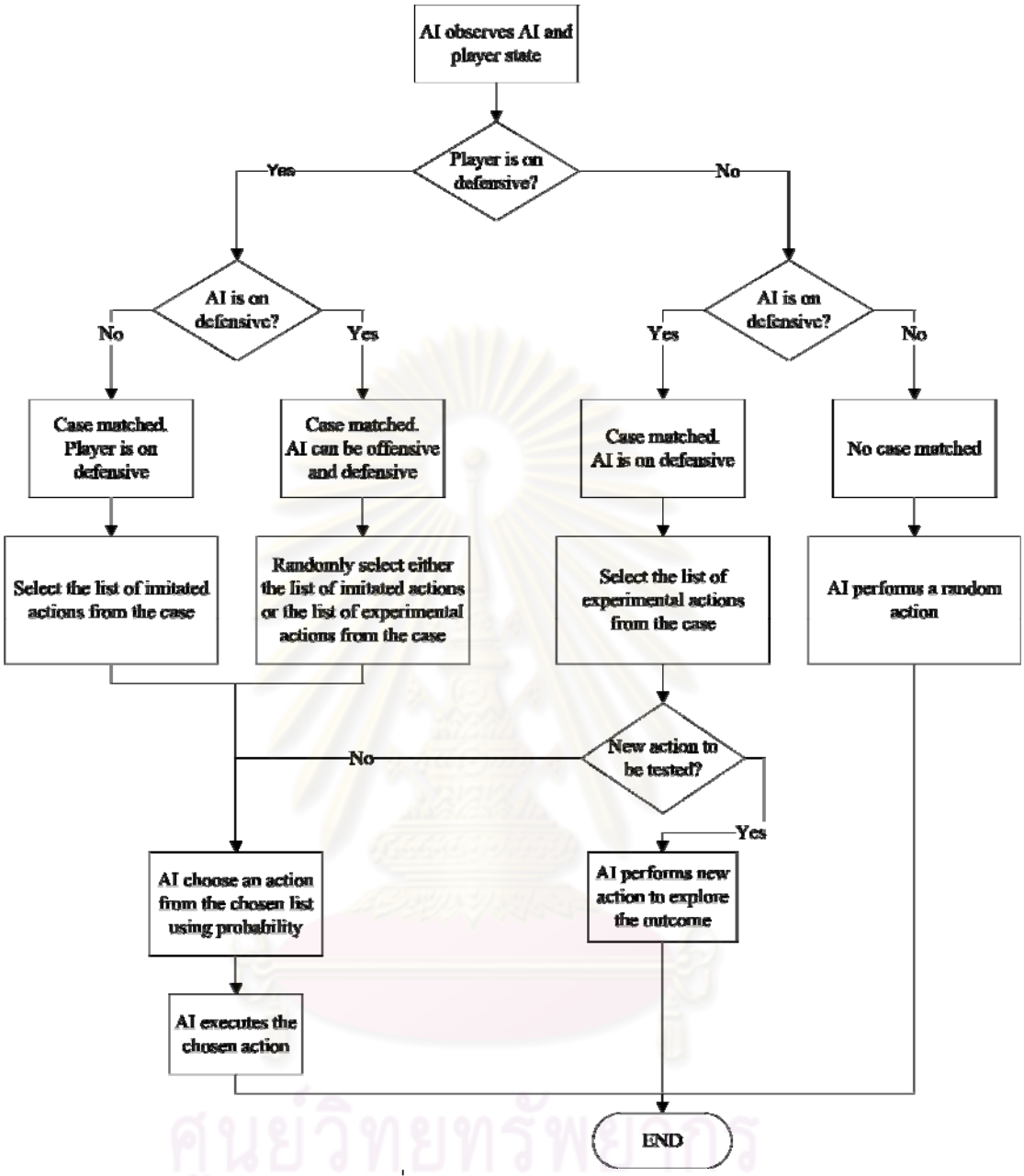
- กรณีที่ตัวละครฝ่ายเอไออยู่ในสถานการณ์เดียวกับที่เคยบันทึกเอาไว้ แสดงว่าฝ่ายเอไออยู่ในสถานการณ์แบบเดียวกับตอนที่ถูกผู้เล่นสร้างความเสียหายให้ ซึ่งโดยธรรมชาติของเกมต่อสู้แล้ว เป็นไปได้สูงมากว่า ผู้เล่นจะโจมตีเข้ามาด้วยลักษณะการโจมตีแบบเดิม เอไอจะทำการเลือกท่าออกมาจากส่วนของท่าสำหรับทดลองใช้ โดยหากมีท่าใหม่สำหรับทดลองใช้(ที่ยังไม่มีค่าประสิทธิภาพ) เอไอจะทดลองใช้ท่านั้นออกไปก่อน เพื่อดูผลที่ได้ หรือหากไม่มีท่าใหม่สำหรับทดลอง เอไอจะทำการเลือกท่าตามความน่าจะเป็นของท่านั้น และทำการแสดงออกท่านั้นออกไป การแสดงออกในลักษณะนี้จะทำให้เอไอสามารถที่จะใช้ท่าใหม่ๆในสถานการณ์เกมที่เคยพบเจอได้ ช่วยให้เอไอสามารถเรียนรู้และปรับตัวขึ้นไปได้

- กรณีที่ทั้งฝ่ายเอไอและผู้เล่นอยู่ในสถานการณ์ที่เคยบันทึกไว้ทั้งคู่ กรณีนี้เอไอจะทำการสุ่มเลือกว่าจะทำการแสดงออกเพื่อลอกเลียนรูปแบบการเล่นของผู้เล่น หรือทำการแสดงออกเพื่อทดลองท่าใหม่ จากนั้นจึงเลือกท่าและทำการแสดงออกท่าที่เลือกต่อไป

ในส่วนของการเลือกท่ามาใช้งานโดยพิจารณาความน่าจะเป็นที่ท่าจะถูกเลือกจากค่าประสิทธิภาพของท่านั้น จะคิดโดยอาศัยหลักการแบบเดียวกันกับที่ใช้ในระบบต้นแบบ ซึ่งได้อธิบายไปในหัวข้อ 3.5.5 โดยจะแตกต่างกันตรงที่เอไอแบบปรับตัวได้อัตโนมัตินั้น ไม่มีการคิดค่า P_a แต่จะคิดค่าร้อยละที่ท่าจะถูกเลือกจากค่าประสิทธิภาพของท่าเลย โดยจะนำเอาค่าประสิทธิภาพของแต่ละท่ามาคิดเทียบกับค่าประสิทธิภาพรวมของทุกท่าในเคสนั้นได้เป็นค่าร้อยละของแต่ละท่าที่จะถูกเลือกออกมา จากนั้นจึงเอาค่าร้อยละที่ได้ไปใช้เลือกท่าด้วยวิธีการเดียวกับในหัวข้อ 3.5.5 ต่อไป

กระบวนการเลือกท่านี้ จะช่วยให้เอไอสามารถเลือกใช้ท่าในสถานการณ์ต่างๆได้อย่างมีประสิทธิภาพ สอดคล้องกับผลลัพธ์ที่ท่านั้นสามารถทำได้ในการเล่นเกม รูปที่ 17 แสดงผังงานของกระบวนการเลือกท่า

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 17 กระบวนการเลือกท่า

4.5.5 การเรียนรู้การกระทำเพื่อหวังผลในระยะยาว (Long term strategy) จากรายละเอียดในขั้นตอนที่ผ่านมา จะเห็นว่าเอไอสามารถเรียนรู้ที่จะเล่นเกมได้ จากการเล่นกับผู้เล่น เอไอสามารถเรียนรู้ข้อมูลสถานการณ์ต่างๆในการเล่นเกมต่อสู้ ท่าทางต่างๆที่สามารถใช้งานได้ เอไอสามารถเรียนรู้ผลลัพธ์ของการใช้ท่าต่างๆในการเล่นเกม รวมถึงสามารถเลือกใช้ท่าที่มีความเหมาะสมต่อสถานการณ์ได้โดยอาศัยข้อมูลจากผลลัพธ์นั้น จากขั้นตอนที่ผ่านมาทั้งหมด เอไอสามารถที่จะเล่นเกมได้โดยมีความสามารถในการปรับตัวตามเวลาจริง โดยใช้ข้อมูลจากการเล่นกับผู้เล่นได้ในระดับหนึ่ง อย่างไรก็ตาม ด้วยการเรียนรู้ของเอไอที่ผ่านมา จะเห็นว่าเอไอใช้เพียง

ข้อมูลสถานการณ์ของสถานการณ์ที่เกิดความเสียหายขึ้นเท่านั้น จากรูปที่ 14 เนื่องจากเวลาที่เกิดความเสียหายนั้น จะเกิดขึ้นจากสถานการณ์ล่าสุดและการกระทำล่าสุดที่เกิดขึ้นภายในเกม ดังนั้นคู่อันดับที่นำมาใช้ในการสร้างข้อมูลเคส หรือ ปรับปรุงข้อมูลในเคส จึงเป็นคู่อันดับล่าสุดที่แสดงถึงสถานการณ์ล่าสุดที่เกิดขึ้นในเกมเพียงคู่อันดับเดียว คือคู่อันดับลำดับที่ X เท่านั้น อย่างไรก็ตามเอไอสามารถเพิ่มความสามารถได้โดยการนำเอาคู่อันดับลำดับอื่น เข้ามาใช้ในการสร้างข้อมูลเคสและปรับปรุงข้อมูลในเคสด้วย ซึ่งจะช่วยให้เอไอมีข้อมูลของสถานการณ์อื่นในเกมที่นำไปสู่สถานการณ์ที่เกิดความเสียหายขึ้นด้วย นอกจากข้อมูลของสถานการณ์ที่ทำให้เกิดความเสียหายขึ้น ซึ่งเป็นการนำเอาหลักการของการเรียนรู้แบบรีนฟอร์สเมนท์ [13] มาใช้งาน

ในธรรมชาติของการเล่นเกมต่อสู้นั้น การกระทำโดยส่วนใหญ่ที่เกิดขึ้นในเกม มักจะเป็นสิ่งที่กระทำลงไป โดยเป็นการตอบสนองต่อการกระทำของฝ่ายตรงข้าม จากรูปที่ 14 ข้อมูลคู่อันดับลำดับที่ X เป็นข้อมูลของสถานการณ์และการกระทำที่ทำให้เกิดความเสียหายขึ้น ซึ่งหมายความว่าข้อมูลในคู่อันดับลำดับก่อนหน้านั้นตั้งแต่ลำดับที่ 1 จนถึง X-1 เป็นข้อมูลของสถานการณ์และการกระทำ ที่เมื่อทำลงไปแล้ว จะนำไปสู่สถานการณ์และการกระทำที่ทำให้เกิดความเสียหายขึ้นได้นั่นเอง ดังนั้นการนำเอาข้อมูลคู่อันดับเหล่านี้มาใช้ในการสร้างและปรับปรุงข้อมูลเคสนอกจากข้อมูลคู่อันดับล่าสุดด้วย จะทำให้เอไอมีความสามารถที่จะแสดงออกการกระทำที่จะนำไปสู่การสร้างความเสียหายให้กับฝ่ายตรงข้ามได้ นอกจากการรอสถานการณ์ที่เคยบันทึกเอาไว้เกิดขึ้นแล้วทำการโจมตีเพียงอย่างเดียว เป็นการเพิ่มความสามารถในการเล่นเกมให้กับเอไอได้ ในการนำเอาข้อมูลคู่อันดับเหล่านี้ไปสร้างและปรับปรุงข้อมูลในเคสนั้น มีรายละเอียดดังนี้

จากกระบวนการสร้างข้อมูลเคสและปรับปรุงข้อมูลในเคส พิจารณารูปที่ 14 เมื่อเอไอทำการสร้างหรือปรับปรุงข้อมูลเคสด้วยข้อมูลคู่อันดับ X เสร็จเรียบร้อยแล้วนั้น เอไอจะนำเอาข้อมูลคู่อันดับ X-1 มาทำการสร้างหรือปรับปรุงข้อมูลในเคสต่อ ด้วยกระบวนการและรายละเอียดแบบเดียวกับการใช้ข้อมูลคู่อันดับ X ทั้งหมด เพียงแต่ค่าประสิทธิภาพที่จะให้หรือลดกับการกระทำในคู่อันดับ X-1 นั้นจะมีค่าน้อยลงไปจากที่ให้กับการกระทำในคู่อันดับ X ซึ่งเมื่อเสร็จแล้วก็จะนำเอาข้อมูลคู่อันดับ X-2, X-3 ย้อนไปเรื่อยๆ มาใช้ต่อไป โดยที่ค่าประสิทธิภาพที่ให้กับการกระทำจะมีค่าลดลงไปเรื่อยๆตามคู่อันดับที่ย้อนกลับไป ในการใช้ข้อมูลคู่อันดับนั้นไม่จำเป็นต้องใช้ข้อมูลทั้งหมดย้อนกลับไปจนถึงลำดับที่ 1 การจะใช้ข้อมูลย้อนกลับไปจนถึงลำดับที่เท่าไรนั้น ขึ้นกับค่าประสิทธิภาพที่ให้กับท่าการกระทำของคู่อันดับ X ซึ่งเป็นท่าที่ทำให้เกิดความเสียหายขึ้น กรณีที่ค่าประสิทธิภาพที่ให้กับการกระทำของคู่อันดับ X มีค่ามาก ก็จะใช้ข้อมูลคู่อันดับย้อนกลับไปมากกว่ากรณีที่ค่าประสิทธิภาพที่ให้กับการกระทำของคู่อันดับ X มีค่าน้อย โดยจะนำเอาข้อมูลคู่อันดับมา

ใช้คิดย้อนกลับไปเรื่อยๆ จนค่าประสิทธิภาพที่ให้มีค่าเป็นศูนย์ เป็นการสะท้อนประสิทธิภาพของการกระทำในคู่อันดับ X ทางหนึ่ง ด้วยการใช้ข้อมูลแบบนี้ ทำให้เอไอสามารถที่จะใช้การกระทำที่หวังผลระยะยาวในการเล่นเกมต่อสู้ ซึ่งปกติแล้วมีแต่การกระทำที่หวังผลระยะสั้นได้

นอกจากนั้นแล้วการเรียนรู้การกระทำเพื่อหวังผลระยะยาวนี้ยังมีข้อดีอีกด้าน คือช่วยในการโจมตีทำท่าต่อเนื่องหรือคอมโบ (combo) ของเอไอได้ การโจมตีทำคอมโบนั้นหมายถึงการโจมตีด้วยท่าท่าเดิมหรือท่าอื่นต่อเนื่องกันไปมากกว่า 1 ครั้ง ซึ่งต่างจากการโจมตีปกติที่เป็นการโจมตีเพียงครั้งเดียว ยกตัวอย่างเช่น มีการโจมตีเป็นคอมโบหนึ่งประกอบด้วยท่า A B และ C ต่อเนื่องกันไป เมื่อท่า C ในเคสหนึ่งได้รับการเพิ่มค่าประสิทธิภาพ ก็จะส่งผลให้ ท่าB และ A ในเคสที่เกิดก่อนหน้าได้รับการเพิ่มค่าประสิทธิภาพไปด้วย ช่วยเพิ่มโอกาสที่เอไอจะสามารถใช้การโจมตีตามลำดับท่า A B C ได้ นอกจากนั้นท่าในสถานการณ์ที่เมื่อแสดงออกแล้ว จะนำไปสู่สถานการณ์ที่สามารถใช้ท่า A ได้ก็จะได้รับการเพิ่มค่าประสิทธิภาพไปด้วยเช่นกัน ซึ่งจะช่วยให้เอไอมีโอกาสเพิ่มมากขึ้นในการทำการกระทำในสถานการณ์หนึ่ง ซึ่งเมื่อทำแล้วจะนำไปสู่สถานการณ์ที่สามารถใช้ท่า A และนำไปสู่การทำการโจมตีเป็นคอมโบโดยลำดับท่า A B C ได้ ซึ่งช่วยให้เอไอสามารถใช้การโจมตีแบบคอมโบซึ่งปกติทำได้ยากสำหรับเอไอเนื่องจากต้องใช้จังหวะและท่าที่สามารถโจมตีต่อเนื่องกันได้ ตารางที่ 3 แสดงตัวอย่างการปรับปรุงค่าประสิทธิภาพของการโจมตีทำ combo โดยใช้ลำดับการกระทำที่ประกอบไปด้วย การเดินเข้าไปหาฝ่ายตรงข้าม (A) ใช้การต่อยเบาโจมตีฝ่ายตรงข้าม (B) ใช้การเตะเบาโจมตีฝ่ายตรงข้าม (C) และ ใช้การยิงกระสุนโจมตีฝ่ายตรงข้าม (D)

ตารางที่ 3 ตัวอย่างการปรับปรุงค่าประสิทธิภาพของท่าต่อเนื่องในเคส

Occurring action	**PERFORMANCE SCORE**				
	Actions before A (if there was any)	**A**	**B**	**C**	**D**
A	-	0	-	-	-
B	Also got weight update	0 + (3 - 1) = 2	3	-	-
C	Also got weight update	2 + (5 - 2) = 5	3 + (5 - 1) = 7	5	-
D	Also got weight update	5 + (15 - 3) = 17	7 + (15 – 2) = 20	5 + (15 - 1) = 19	15

จากตารางที่ 3 เมื่อใช้ท่า A ยังไม่มีความเสียหายเกิดขึ้น ท่า A จึงยังไม่มีค่าประสิทธิภาพ เมื่อมีการทำท่า B และเกิดความเสียหาย ท่า B จะได้รับค่าประสิทธิภาพ และ ท่า A จะได้รับค่าประสิทธิภาพด้วย โดยจะได้จากค่าประสิทธิภาพของท่า B ที่ปรับลดลง ต่อไปเมื่อมีการทำท่า C และเกิดความเสียหายขึ้น ท่า C จะได้รับค่าประสิทธิภาพ ท่า B และ A ก็จะได้รับค่าประสิทธิภาพเช่นกัน โดยจะได้เป็นค่าปรับลดตามลำดับของค่าประสิทธิภาพที่ให้กับท่า C และเมื่อทำท่า D และ

เกิดความเสียหายขึ้น ท่า C, B, A ก็จะได้รับค่าประสิทธิภาพไปด้วย โดยได้ค่าที่ปรับลดลงจากค่าที่ให้กับท่า D ตามลำดับ

หลังจากการใช้คู่อันดับอื่นๆเพื่อสร้างข้อมูลเคสและปรับปรุงข้อมูลในเคสแล้ว เมื่อไม่มีความเสียหายเกิดขึ้นในเกมแล้ว ข้อมูลคู่อันดับของ Casekey และ การกระทำที่ถูกบันทึกเอาไว้ ทั้งของฝ่ายเอไอและผู้เล่นจะถูกลบทิ้งทั้งหมด เนื่องจากเป็นข้อมูลการกระทำที่ไม่ได้ถูกนำไปใช้ และจะมีการสร้างข้อมูลคู่อันดับของทั้งสองฝ่ายนี้ขึ้นใหม่ เมื่อเอไอและผู้เล่นมีการแสดงออกการกระทำใดๆต่อไป

4.5.6 การคิดค่าคะแนนประสิทธิภาพเมื่อท่าใช้สร้างความเสียหายแก่ฝ่ายตรงข้ามได้ สำหรับการคิดค่าคะแนนประสิทธิภาพในระบบปัญญาประดิษฐ์แบบปรับตัวได้อัตโนมัตินี้ จะคิดโดยอาศัยหลักการเดียวกันกับที่ใช้ในระบบต้นแบบ โกสเอไอที่ปรับอัตราการออกท่าที่เรียนรู้ได้ตามเวลาจริง ดังที่ได้อธิบายไปในหัวข้อ 3.5.3 ท่าแต่ละท่าที่สร้างความเสียหายได้ จะได้ค่าคะแนนประสิทธิภาพเป็นค่าความเสียหายที่ท่านั้นสร้างได้เองรวมกับค่าคะแนนของท่าก่อนหน้าที่เกิดขึ้นเป็นการโจมตีต่อเนื่องกัน (ถ้ามีท่าก่อนหน้า) ซึ่งหลังจากมีการคิดคะแนนให้กับแต่ละท่าที่ทำความเสียหายได้แล้ว จะมีการคิดค่าคะแนนย้อนหลังตามขั้นตอนการเรียนรู้การกระทำเพื่อหวังผลในระยะยาวด้วย ถ้ามีการสร้างความเสียหายด้วยท่า A B C และ D ตามลำดับ เช่นเดียวกับที่แสดงในตารางที่ 3 ตัวอย่างการคิดค่าคะแนนของท่าตามลำดับเหตุการณ์ที่เกิดขึ้นในการเล่นเกม ซึ่งขยายรายละเอียดเพิ่มเติมจากตารางที่ 3 แสดงดังตารางที่ 4

ตารางที่ 4 ตัวอย่างการคิดค่าคะแนนประสิทธิภาพของท่า

Occurring action	**PERFORMANCE SCORE**			
	A	**B**	**C**	**D**
Action A	-	-	-	-
Action B damaged opponent	-	3	-	-
Long term strategy	0+(3-1)=2	3	-	-
Action C damaged opponent	2	3	3+2=5	-
Long term strategy	2+(5-2)=5	3+(5-1)=7	5	-
Action D damaged opponent	5	7	5	5+10=15
Long term strategy	5+(15-3)=17	7+(15-2)=20	5+(15-1)=19	15

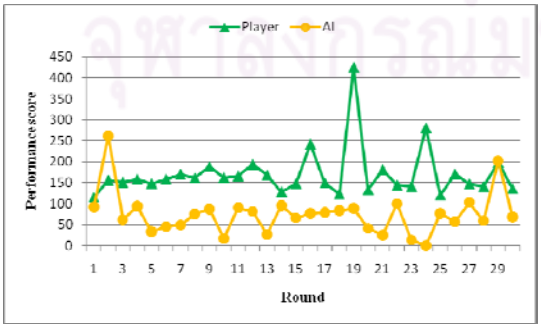
บทที่ 5
การทดลองเอไอและวิเคราะห์ผล

5.1 การทดลอง

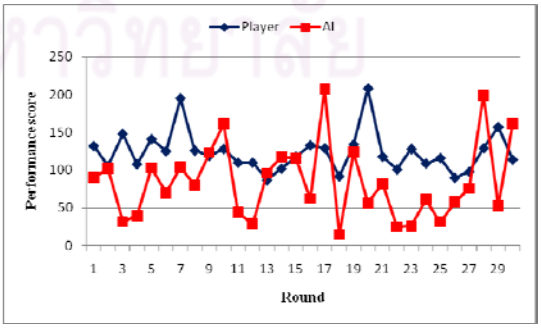
วิทยานิพนธ์ฉบับนี้ได้ทำการสร้างและทดลองเอไอที่สามารถปรับตัวได้แบบอัตโนมัติในเวลาจริง โดยอาศัยข้อมูลที่ได้จากการเล่น ให้ชื่อว่า Automatic Adaptive AI (AAI) ในการทดลองได้ทำการทดลองเอไอเปรียบเทียบกับเอไอแบบ Adaptive Ghost AI (AGAI) ซึ่งเป็นแนวทางในการสร้างเอไอที่ปรับตัวตามข้อมูลการเล่นอีกรูปแบบหนึ่ง เพื่อให้เห็นผลชัดเจนขึ้น โดยทำการทดลองให้ผู้ทดสอบทำการทดลอง เล่นกับเอไอทั้งสองแบบ โดยจะให้ผู้ทดสอบทำการเล่นกับเอไอแบบละ 10 นัด ผู้ทดลองมีประสบการณ์ในการเล่นเกมต่อสู้ที่หลากหลายกันไป ทั้งผู้ที่เพิ่งเริ่มเล่นและผู้เล่นที่เชี่ยวชาญ ทดลองโดยไม่ให้ผู้ทดลองทราบว่ากำลังเล่นกับเอไอชนิดใดอยู่ โดยใน 1 นัด จะต้องเอาชนะฝ่ายตรงข้ามให้ได้ 3 ยก จึงจะถือเป็นการชนะ 1 นัด มีจำนวนผู้ทดลองทั้งหมด 5 คน

5.2 ผลการทดลอง

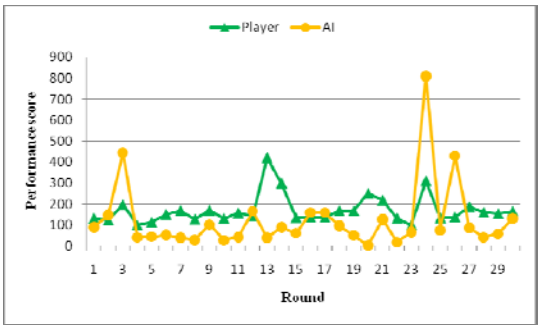
ผลการทดลองแสดงดังรูปที่ 18 ตารางที่ 5 และตารางที่ 6 กราฟผลการทดลองจะแสดงถึงค่าประสิทธิภาพของฝ่ายเอไอและผู้ทดสอบ ที่ได้รับจากการสร้างความเสียหายแก่ฝ่ายตรงข้ามในการเล่นแต่ละยก ตารางที่ 5 แสดงการเปรียบเทียบค่าประสิทธิภาพระหว่าง AGAI กับ AAI ตารางที่ 6 แสดงร้อยละของจำนวนครั้งที่เอไอมีการนำเอาข้อมูลเคสที่เรียนรู้ได้มาใช้งานในแต่ละยกของการเล่นเทียบกับร้อยละของจำนวนครั้งที่เอไอทำการสุ่มออกท่าในแต่ละยกของการเล่น และจำนวนเคสทั้งหมดที่เอไอเรียนรู้ได้



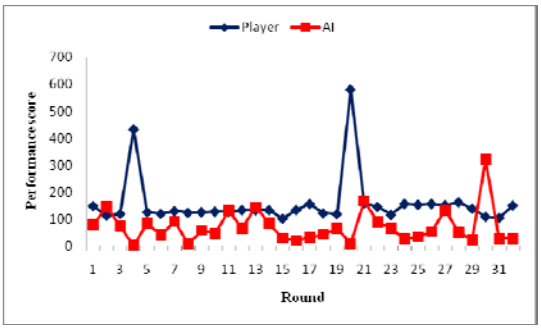
Player 1 – AGAI



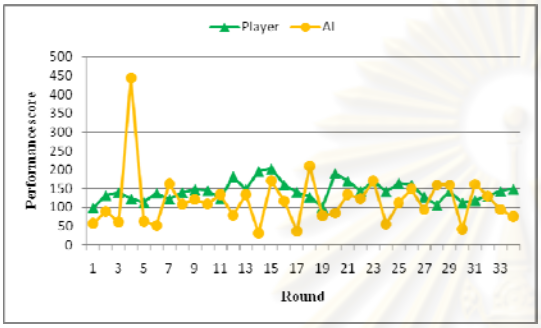
Player 1 – AAI



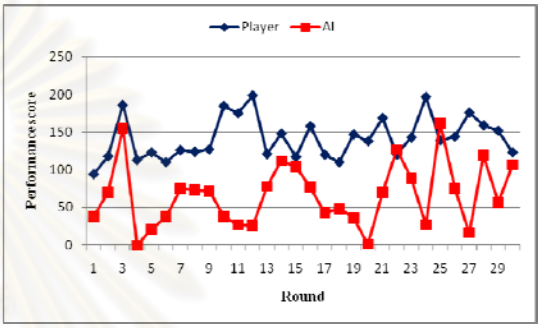
Player 2 – AGAI



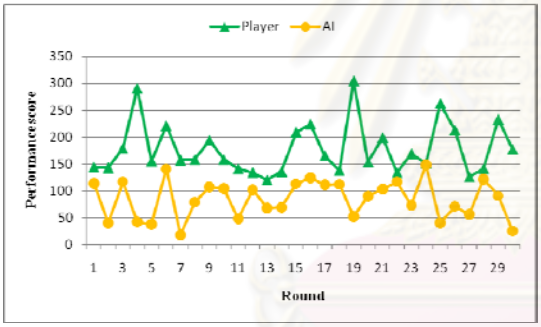
Player 2 – AAI



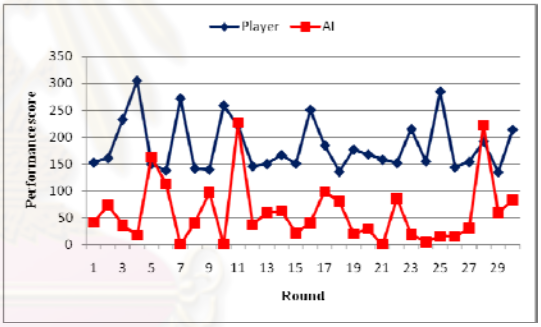
Player 3 – AGAI



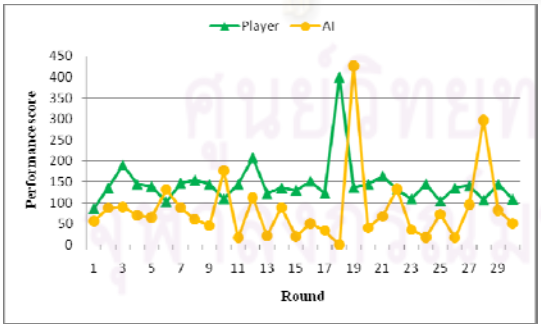
Player 3 – AAI



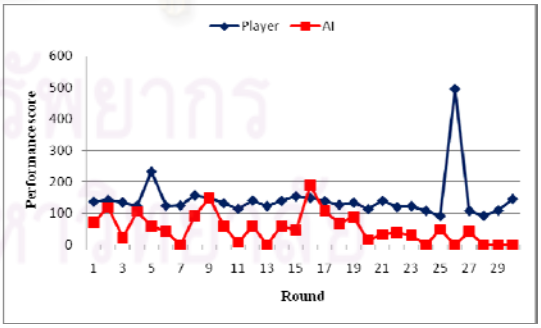
Player 4 – AGAI



Player 4 – AAI



Player 5 – AGAI



Player 5 - AAI

รูปที่ 18 กราฟแสดงผลการทดลอง

ตารางที่ 5 เปรียบเทียบค่าประสิทธิภาพระหว่าง AGAI และ AAI

Player	AVERAGE PERFORMANCE SCORE	
	Player's score – AGAI's score	Player's score – AAI's score
1	95.03	38.90
2	47.10	86.21
3	24.85	76.03
4	93.16	124.03
5	59.46	92.36
AVERAGE	63.92	83.50

ตารางที่ 6 จำนวนครั้งที่มีการนำเคสที่เรียนรู้ได้มาใช้งานเทียบกับจำนวนครั้งที่ออกท่าแบบสุ่มและจำนวนเคสที่เรียนรู้ได้ของ AGAI และ AAI

Player	AVERAGE % OF RANDOM ACTION PER ROUND		AVERAGE % OF CASE UTILIZATION PER ROUND		TOTAL NUMBER OF LEARNED CASE	
	AGAI	AAI	AGAI	AAI	AGAI	AAI
1	9.51	11.23	90.49	88.77	285.00	115.00
2	9.07	12.54	90.93	87.46	293.00	124.00
3	6.10	15.46	93.90	84.54	250.00	102.00
4	6.64	14.31	93.36	85.69	226.00	109.00
5	9.67	22.82	90.33	77.18	156.00	90.00
AVERAGE	8.20	15.27	91.80	84.73	242.00	108.00

5.3 วิเคราะห์ผลการทดลอง

เนื่องจากเกมต่อสู้มีเป้าหมายในการโจมตีให้พลังชีวิตของฝ่ายตรงข้ามหมดเพื่อเอาชนะฝ่ายตรงข้าม ในการทดลองนี้ เมื่อมีการสร้างความเสียหายลดพลังชีวิตแก่ฝ่ายตรงข้าม ฝ่ายที่เป็นผู้กระทำความเสียหายจะได้คะแนนมาเป็นค่าประสิทธิภาพ กำหนดให้ค่าคะแนนที่ทำได้สื่อถึงความสามารถในการเล่นเกม หากทำคะแนนจากฝ่ายตรงข้ามได้มาก หมายความว่าสามารถเล่นได้ดีกว่ากรณีที่ทำคะแนนจากฝ่ายตรงข้ามได้น้อย

จากผลการทดลอง กราฟแสดงค่าประสิทธิภาพที่ฝ่ายเอไอและผู้ทดสอบทำได้จากฝ่ายตรงข้ามในแต่ละยก ในทางด้านซ้ายมือ จะเป็นผลที่ได้จากการที่ผู้ทดลองทำการเล่นกับเอไอแบบ AGAI ซึ่งจากผลจะพบว่าผลที่ได้จากผู้ทดสอบทุกคนออกมาในลักษณะที่คล้ายคลึงกันหมด คือผู้เล่นยังคงเล่นได้ดีกว่าเอไอ แต่ค่าประสิทธิภาพที่เอไอทำได้ในการเล่นกับผู้ทดสอบนั้น มีค่าเข้าใกล้กับค่าที่ผู้ทดสอบทำได้ แสดงให้เห็นว่าเอไอสามารถปรับตัวได้ โดยอาศัยประสบการณ์ที่ได้จากการเล่นเกม ค่าประสิทธิภาพที่ได้ถึงแม้จะมีค่าขึ้นลงบ้าง แต่เมื่อมองลักษณะโดยรวมแล้วจะพบว่ามีลักษณะที่เปลี่ยนแปลงไม่มากค่อนข้างคงที่ ซึ่งค่าประสิทธิภาพที่ผู้ทดสอบทำได้จากเอไอก็ได้ผลออกมาในลักษณะเดียวกัน ซึ่งสามารถอธิบายได้ว่า เมื่อทำการเล่นเกม เอไอจะทำการลอกเลียน

รูปแบบการเล่นของผู้ทดสอบ และนำมาปรับใช้ให้เหมาะสม ซึ่งเมื่อเอไอมีความสามารถในการเล่นดีขึ้น ทำให้ประสิทธิภาพที่ได้รับเพิ่มขึ้น ในขณะที่ผู้ทดสอบ หลังจากได้ประสบการณ์ในการเล่น ก็จะพัฒนารูปแบบการเล่นของตัวเองขึ้นไป ซึ่งทำให้ประสิทธิภาพของเอไอตกลง ส่งผลให้เอไอต้องทำการปรับรูปแบบการเล่นอีก เป็นเช่นนี้ไปเรื่อยๆ ทำให้ค่าประสิทธิภาพแสดงออกมาในลักษณะของกราฟที่มีการแกว่งขึ้นลง แต่มีลักษณะโดยรวมค่อนข้างคงที่ ซึ่งแสดงให้เห็นว่าเอไอสามารถปรับตัวตามผู้ทดสอบได้ อย่างไรก็ตาม เนื่องจาก AGAI สามารถใช้ได้เพียงท่าการกระทำที่ลอกเลียนมาจากทางผู้ทดสอบเท่านั้น ไม่สามารถใช้การกระทำใหม่อื่นๆได้ ทำให้การกระทำที่แสดงออกได้มีจำกัด ซึ่งมักเป็นสาเหตุให้ AGAI ทำการกระทำที่เปิดช่องว่างให้ผู้ทดสอบเข้าไปโจมตีได้โดยง่าย ค่าประสิทธิภาพที่ผู้ทดสอบได้รับในแต่ละยกของการเล่นเกมนั้น โดยส่วนใหญ่ยังคงมีค่ามากกว่าค่าประสิทธิภาพที่ทางฝั่งเอไอได้รับอยู่ ซึ่งแสดงให้เห็นว่า ถึงแม้เอไอแบบ AGAI จะสามารถปรับตัวตามผู้เล่นได้ในระดับหนึ่ง แต่ผู้เล่นที่เป็นมนุษย์จริงๆนั้น ยังคงมีความสามารถในการเรียนรู้และปรับตัวได้เร็วกว่าเอไอแบบ AGAI อยู่ ทำให้ผู้เล่นสามารถเล่นได้ดีกว่า

ในทางด้านขวามือ เป็นผลการทดลองจากการที่ผู้ทดสอบ เล่นกับเอไอแบบ AAI ซึ่งเป็นเอไอที่นำเสนอในวิทยานิพนธ์นี้ จากกราฟจะเห็นได้ว่าผลของผู้ทดสอบทุกคนจะออกมาในลักษณะที่มีการแกว่งขึ้นลง ซึ่งสามารถอธิบายได้ด้วยเหตุผลแบบเดียวกับผลที่ได้จากการทดลองระหว่างผู้ทดสอบกับ AGAI อย่างไรก็ตาม เอไอแบบ AAI นั้น สามารถที่จะปรับตัวโดยแสดงออกซึ่งท่าทางการกระทำใหม่ๆ ที่ได้นำมาทดลองใช้ในสถานการณ์เกมแบบต่างๆได้ ทำให้รูปแบบการเล่นไม่ได้ถูกจำกัด ซึ่งเมื่อพิจารณาที่ค่าประสิทธิภาพที่เอไอแบบ AAI ทำได้จากการเล่นกับผู้ทดสอบและค่าประสิทธิภาพที่ทางฝั่งผู้ทดสอบทำได้ในแต่ละยกแล้วนั้น พบว่า ค่าประสิทธิภาพที่ AAI ทำได้นั้นมีค่าเข้าใกล้กับค่าที่ผู้เล่นทำได้เช่นเดียวกับ AGAI แสดงให้เห็นว่าเอไอสามารถปรับตัวพัฒนาได้โดยอาศัยประสบการณ์ที่ได้จากการเล่นเกมได้ แต่ค่าประสิทธิภาพของ AGAI นั้นมีค่าใกล้เคียงกับค่าของผู้เล่นมากกว่า AAI ซึ่งที่เป็นเช่นนี้นั้น สามารถอธิบายได้ว่า เอไอแบบ AGAI นั้น เนื่องจากท่าต่างๆที่เอไอนำมาใช้นั้น มาจากการลอกเลียนแบบท่าที่ผู้เล่นใช้อย่างเดียว ไม่มีการทดลองท่าใหม่ๆ ซึ่งหมายความว่าท่าที่ใช้ออกมาเป็นท่าที่ผู้เล่นคิดแล้วว่าใช้ได้ผลจึงได้แสดงออกมา เปรียบเสมือนเอไอทำการเรียนรู้โดยอาศัยตัวอย่างที่ดีจากผู้เล่น ท่าต่างๆที่เรียนรู้มาใช้นั้นจึงค่อนข้างใช้ได้ผลดี และได้ค่าประสิทธิภาพที่ดี ในทางกลับกัน เอไอแบบ AAI นั้น ท่าที่ลอกแบบจะมีการบันทึกเป็นเคสก็ต่อเมื่อสามารถทำคะแนนได้(หรือนำไปสู่การทำคะแนนในเวลาไม่นานต่อมา) เท่านั้น ต่างจาก AGAI ที่บันทึกท่าต่างๆจากผู้เล่นมาเป็นเคสเสมอไม่ว่าท่านั้นจะได้ผลหรือไม่ ดังนั้นจำนวนเคสของ AAI จึงมีจำนวนน้อยกว่า การที่มีจำนวนเคสน้อยกว่า ทำให้จำนวน

รูปแบบการกระทำที่ต้องใช้กับการเรียนรู้แบบรีนฟอร์สเม้นท์น้อยลงไปด้วย นอกจากนี้ AAI ยังมีการทดลองท่าใหม่เพื่อแสดงออกถึงรูปแบบการเล่นใหม่ๆด้วย ทำให้ต้องใช้เวลาลองผิดลองถูกสำหรับรูปแบบการเล่นต่างๆ และท่าใหม่จะได้รับการบันทึกเป็นเคสก็ต่อเมื่อใช้ได้คะแนน(หรือนำไปสู่การได้คะแนนในเวลาไม่นานต่อมา) เท่านั้น จากการที่มีจำนวนเคสที่ลอกแบบมาน้อยและต้องทำการทดลองท่าใหม่ๆไปด้วย ทำให้ค่าประสิทธิภาพที่ทำได้ถึงแม้จะมีค่าใกล้เคียงผู้เล่น แต่ก็ยังคงได้น้อยกว่าค่าประสิทธิภาพของ AGAI ซึ่งหากต้องการให้ได้ผลการเล่นที่ดีขึ้น จำเป็นต้องทดลองรูปแบบการเล่นให้ได้รูปแบบที่หลากหลายเพียงพอและใช้เวลาในการเรียนรู้ที่นานขึ้น ตามหลักการของการเรียนรู้แบบรีนฟอร์สเมนท์

อย่างไรก็ตามจากค่าประสิทธิภาพที่เอไอทั้งสองแบบทำได้ในแต่ละยกของการเล่น จะเห็นว่าแม้เอไอจะมีความสามารถในการปรับตัวแล้ว แต่ก็ยังแสดงความสามารถในการเล่นได้เพียงแค่ใกล้เคียงกับผู้ทดสอบเท่านั้น ไม่ถึงกับเหนือกว่าผู้ทดสอบโดยสมบูรณ์ได้ ผู้ทดสอบที่เป็นมนุษย์ก็ยังคงมีความสามารถในการเรียนรู้และปรับตัวพัฒนาได้เร็วกว่าเอไออยู่ ที่เป็นเช่นนี้นั้นสามารถอธิบายได้โดยมีรายละเอียดดังนี้

5.3.1 ความรู้เกี่ยวกับท่าที่จะนำมาใช้งาน ถึงแม้ว่าเอไอจะมีความสามารถในการปรับตัวใช้ท่าตามสถานการณ์โดยใช้ท่าที่อ้างอิงกับผลลัพธ์ของการใช้ท่านั้นที่เคยพบเจอมา ทำให้สามารถเลือกใช้ท่าที่ให้ผลดีตามสถานการณ์ได้แล้วนั้น แต่อย่างไรก็ตามเนื่องจากเอไอไม่มีความรู้ในเรื่องของท่าการกระทำเลย สำหรับ AAI ถ้าไม่ได้มีข้อมูลการลอกแบบบันทึกเป็นเคสไว้ จะต้องทดลองท่าใช้ไปทีละท่าเมื่อพบเจอสถานการณ์ครั้งหนึ่งจนกว่าจะพบท่าที่เหมาะสมให้ผลดี ทำให้ต้องใช้เวลาในการเรียนรู้นาน ทำให้ความสามารถในการปรับตัวยังช้ากว่าผู้เล่นที่เป็นมนุษย์จริง ซึ่งเมื่อเทียบกันแล้ว สามารถรู้ได้ทันที หรือรู้ได้ในการทดลองเพียงไม่กี่ครั้งว่าควรจะใช้ท่าใดในสถานการณ์เกมที่พบเจอนั้นจึงจะให้ผลดีที่สุด ยกตัวอย่างเช่น ในกรณีที่ตัวละครของฝ่ายตรงข้ามทำการโจมตี โดยใช้การกระโดดโจมตีเข้ามาทางด้านบน ผู้เล่นที่เป็นมนุษย์ เมื่อเห็นดังนั้น จะรู้ในทันทีว่าควรจะต้องทำการป้องกัน หลบหลีก หรือใช้การโจมตีที่มีทิศทางการโจมตีไปทางด้านบนเพื่อทำการโจมตีโต้กลับไป ซึ่งมีเพียง 3 ท่า สำหรับตัวละคร Ryu ที่ใช้ทดลองเป็นกรณีศึกษา รวมแล้วมีรูปแบบการกระทำที่เหมาะสมเพียง 5 รูปแบบเท่านั้น ซึ่งเมื่อเทียบกับ AAI แล้ว เมื่อตัวละครฝ่ายตรงข้ามกระโดดโจมตีเข้ามา เอไอจะเลือกท่าการกระทำมาใช้เพื่อโต้ตอบการโจมตีนั้น แต่จากการที่เอไอไม่มีความรู้เกี่ยวกับท่าเลย(นอกเสียจากเคยเห็นผู้เล่นกระทำ) ทำให้เอไอต้องพิจารณารูปแบบการกระทำทั้งหมดที่เป็นไปได้ ซึ่งมีทั้งหมด 77 รูปแบบสำหรับตัวละคร Ryu (สามารถนำมาทดลองใช้ได้ 68 รูปแบบ รูปแบบที่เหลือเป็นรูปแบบที่เอไอไม่สามารถกระทำด้วยตนเองได้หรือ

กระทำได้ยาก) และเลือกเอามาใช้ท่าหนึ่ง ซึ่งท่าที่เลือกมาใช้นั้นอาจเป็นท่าที่ไม่เกี่ยวข้องกับสถานการณ์เลยก็ได้ เช่นท่าก้มต่อยเบา ซึ่งเป็นการโจมตีทางด้านล่างที่มีทิศทางไปข้างหน้า ซึ่งไม่สามารถเอามาใช้หลบหลีก ป้องกัน หรือโจมตีสวนกลับการกระโดดโจมตีจากทางด้านบนได้เลย ทำให้เอไอต้องรับการโจมตีจากฝ่ายตรงข้ามไป แล้วจึงค่อยนำเอาผลลัพธ์นั้นไปทำการปรับค่าน้ำหนักท่าให้รู้ว่าท่าที่ใช้นั้นไม่เหมาะสม ซึ่งกว่าที่เอไอจะสามารถเรียนรู้จนเอาท่าที่เหมาะสมออกมาใช้ให้ถูกกับสถานการณ์นั้น จำเป็นต้องใช้เวลาทดลองไปทีละท่าขึ้นอยู่กับปัจจัยจากการสุ่มท่ามาทดลองใช้ด้วย ว่าจะได้ท่าที่เหมาะสมมาใช้งานได้เร็วหรือไม่ หรือต้องรอเจอสถานการณ์เดิมแล้วถูกโจมตีหลายครั้ง จึงจะสุ่มได้ท่าที่เหมาะสมมาใช้งานได้ ทำให้แม้เอไอจะปรับตัวได้เร็วในเวลาจริง แต่ก็ไม่สามารถที่จะเร็วเทียบเท่ากับมนุษย์จริง ที่เมื่อเห็นสถานการณ์ในเกม ก็สามารถรู้ได้ทันทีโดยไม่ต้องใช้เวลาทำการเรียนรู้นานว่าควรจะทำตัวเช่นไรได้

5.3.2 จังหวะการแสดงออกของท่า อีกสาเหตุหนึ่งก็คือเรื่องของจังหวะการแสดงออกท่า ซึ่งสำหรับเกมต่อสู้แล้วนั้น ช่วงของจังหวะการแสดงออกท่านั้นเป็นเรื่องสำคัญมากในการเล่นเกม โดยส่งผลไปถึงเรื่องของการโจมตีว่าจะโดนฝ่ายตรงข้ามหรือไม่ หรือในการโจมตีทำคอมโบซึ่งจังหวะส่งผลถึงการทำคอมโบว่าจะสามารถทำได้หรือไม่ด้วย ในสถานการณ์การเล่นเกมนั้นถึงแม้ว่าเอไอจะสามารถเลือกการกระทำมาแสดงออก โดยเป็นท่าที่เหมาะสมกับสถานการณ์ดีแล้ว แต่หากจังหวะการแสดงออกของท่านั้นเร็วไปหรือช้าไป ก็อาจทำให้ท่านั้นใช้ไม่ได้ผลจนทำให้ถูกลดค่าน้ำหนักของท่าลงไปได้ และในการโจมตีทำคอมโบหากออกท่าเร็วหรือช้าเกินไป ก็จะทำให้การโจมตีไม่ออกหรือไม่ต่อเนื่องจนไม่สามารถโจมตีนับเป็นคอมโบได้

จังหวะในการออกท่าการกระทำนั้น ขึ้นกับช่วงเวลาของการกดคำสั่งท่านั้นเป็นหลัก โดยเฉลี่ยแล้ว ตัวละครจะเริ่มแสดงอนิเมชันของท่าที่ใส่คำสั่งเข้าไปหลังจากใส่คำสั่งเสร็จสิ้นแล้วเป็นเวลา 6 เฟรม (สำหรับเกมต่อสู้ส่วนใหญ่แล้ว จะมีจำนวนเฟรม 60 เฟรมต่อหนึ่งวินาที) ซึ่งสำหรับเอไอในวิทยานิพนธ์นี้ จะทำการตรวจสอบสถานการณ์ในเกมและเลือกท่ามาส่งคำสั่งกดท่าเพื่อแสดงออกท่านั้นทุกๆ 6 เฟรม (ไม่รวมเฟรมที่ใช้สำหรับกดคำสั่งท่า) เพื่อที่จะสามารถแสดงออกท่าการกระทำได้อย่างต่อเนื่องกันได้ และใช้เวลากดคำสั่งท่าเป็นเวลาคำสั่งละ 3 เฟรม (เป็นเวลาเฉลี่ยสำหรับการกดท่าของมนุษย์ในการเล่น เพื่อไม่ให้มีการได้เปรียบกันในเรื่องของการกดคำสั่งท่าระหว่างเอไอกับผู้เล่น) ซึ่งจะเห็นว่าเอไอใช้เวลาคงที่ตลอดในการส่งคำสั่งและแสดงออกท่าการกระทำ ต่างกับผู้เล่นที่สามารถปรับใช้เวลาทำงานตรงจุดนี้ให้ท่าออกในช่วงเวลาที่ต้องการได้ เนื่องจากตัวเกมมีปัจจัยที่อนุญาตให้ใส่คำสั่งท่าช้าหรือเร็วกว่าปกติได้ช่วงเวลาหนึ่งแล้วแต่ท่าการ

กระทำ ทำให้สามารถกำหนดจังหวะออกท่าได้ดีกว่าเอไอ ส่งผลให้แสดงออกในการเล่นได้ดีกว่าเอไอได้

5.3.3 Short animation นอกจากนั้นแล้วยังมีเรื่องของ short animation ซึ่งเป็นปัญหาที่เกิดขึ้นจากการที่ตัวละครแสดงการกระทำอื่นที่ไม่ได้เป็นการกระทำที่ต้องออกมาในช่วงที่กำลังจะแสดงการกระทำอย่างหนึ่งเช่น ในการปล่อยกระสุนเพื่อโจมตีฝ่ายตรงข้ามนั้น ใช้การกดคำสั่ง ลง ลงเฉียงหน้าและ หน้า แล้วกดปุ่มต่อย ซึ่งจากลักษณะการกดปุ่ม ตัวละครจากท่ายืนปกติ ก่อนที่จะแสดงท่าปล่อยกระสุนออกมา อาจแสดงท่าก้มลงออกมาก่อนระยะหนึ่งได้ จากคำสั่งที่ส่งเข้าไป ซึ่งเมื่อตัวละครแสดงท่าซึ่งไม่ได้ตั้งใจให้เกิดออกมา สามารถทำให้เอไอเข้าใจผิดว่าท่าซึ่งไม่ได้ตั้งใจให้เกิดนั้น เป็นท่าหลักของสถานการณ์และนำเอาไปใช้ ทำให้การเรียนรู้ผิดเพี้ยนไปได้

5.3.4 อนิเมชันที่ซ้ำซ้อนกัน อีกส่วนคือเรื่องความกำกวมของอนิเมชันที่บางท่ามีอนิเมชันในบางส่วนของท่าเหมือนกัน ทำให้เมื่อเวลาที่เอไอทำการตรวจสอบท่า อาจทำให้เข้าใจท่าหนึ่งเป็นอีกท่าหนึ่งได้ อย่างไรก็ตามตรงจุดนี้นั้น ถึงแม้จะเกิดการเข้าใจท่าผิด แต่ก็ยังคงสามารถใช้ท่าที่เข้าใจผิดนั้นได้ในระดับหนึ่ง เพราะท่าที่มีอนิเมชันเหมือนกันนั้น จะเป็นท่าแบบเดียวกัน ที่จะต่างกันไปเพียงระยะและความเร็วของท่าเท่านั้น เอไอจึงสามารถใช้ท่านั้นแทนท่าที่ควรจะเป็นจริงๆได้ในระดับหนึ่ง แม้จะไม่ให้ผลที่เหมือนกันโดยสมบูรณ์ก็ตาม

อย่างไรก็ตาม นอกจากผลการทดลองที่ให้ผู้ทดสอบทดลองเล่นกับเอไอแล้ว ผู้ทดสอบแต่ละคนยังได้ให้ความเห็นเกี่ยวกับเอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้ไว้อีกด้วย โดยผู้ทดสอบทุกคนมีความเห็นไปในทางเดียวกันว่า เอไอแบบ AAI นั้น ถึงแม้จากผลจะได้ค่าประสิทธิภาพเป็นรองเอไอจากระบบต้นแบบ AGAI แต่ความรู้สึกที่ได้จากการเล่นนั้น เอไอแบบ AAI สามารถให้รูปแบบการเล่นที่หลากหลายมีการพัฒนาเหมือนมนุษย์จริงมากกว่าเอไอแบบ AGAI ที่มีการแสดงออกด้วยรูปแบบตายตัวมากกว่า ผู้ทดลองมีความเห็นว่าเอไอแบบ AGAI และ AAI นั้น ทั้งสองแบบมีความสามารถในการปรับตัวโดยอาศัยประสบการณ์ที่ได้จากการเล่นเกมได้ ให้รูปแบบการเล่นการพัฒนาที่ดีกว่า และให้ความท้าทายในการเล่นที่สูงกว่าเอไอแบบ GAI และเอไอปกติของเกมอย่างเห็นได้ชัด

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

จุดมุ่งหมายสำหรับวิทยานิพนธ์ฉบับนี้ คือการสร้างเอไอสำหรับเกมต่อสู้ที่มีความสามารถในการปรับตัวพัฒนาขึ้นแบบอัตโนมัติในเวลาจริง โดยอาศัยข้อมูลที่ได้จากการเล่นเกมมาทำการพัฒนาตนเอง จากแนวคิดและการทดลองสามารถสรุปผลได้ดังนี้

เอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้ทั้ง Adaptive Ghost AI (AGAI) และ Automatic Adaptive AI (AAI) สามารถเรียนรู้และปรับตัวพัฒนาขึ้นได้ในเวลาจริง โดยอาศัยข้อมูลที่ได้จากการเล่นเกม ซึ่งจากผลการทดลองและความคิดเห็นของผู้ทดลอง ได้แสดงให้เห็นว่าเอไอนั้นสามารถปรับตัวพัฒนาได้จริงตามจุดมุ่งหมายของการวิจัย แต่อย่างไรก็ตามเอไอยังสามารถปรับตัวเล่นเกมได้ โดยให้ผลในการเล่นได้ใกล้เคียงผู้เล่นเท่านั้น ไม่ถึงระดับความสามารถที่สูงกว่าผู้เล่นโดยสมบูรณ์ได้ จากปัจจัยที่ได้กล่าวถึงในส่วนวิเคราะห์ผลการทดลอง

6.2 อภิปรายการวิจัยและข้อเสนอแนะ

จากข้อมูลที่ได้นำเสนอในส่วนวิเคราะห์การทดลอง ปัจจัยที่ทำให้เอไอเรียนรู้พัฒนาได้ช้านั้น มาจากปัญหา เรื่องของ short animation ความซ้ำซ้อนของอนิเมชันของท่า จังหวะการแสดงออกท่าของเอไอ ความไม่รู้เกี่ยวกับท่าการกระทำที่ควรนำมาใช้ในสถานการณ์ต่างๆ นอกจากนี้ยังมีปัญหาจำนวนเคสที่น้อยของ AAI อีกด้วย

สำหรับเรื่องของ short animation และความซ้ำซ้อนของแอนิชันนั้น เนื่องจากเอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้นั้น เน้นไปที่การเรียนรู้แบบออนไลน์ ซึ่งจะพิจารณาสิ่งที่เกิดขึ้นในเกมทันทีที่เกิดขึ้น ทำให้ไม่สามารถที่จะตรวจสอบ short animation และ ความซ้ำซ้อนของอนิเมชันได้ เนื่องจากต้องใช้เวลาพิจารณาเป็นจำนวนหลายเฟรม ซึ่งหากเอไอต้องทำการรอเพื่อพิจารณา จะทำให้การเรียนรู้ช้าและไม่ทันกับการเล่นของผู้เล่นได้ อย่างไรก็ตามปัญหานี้สามารถปรับปรุงเพิ่มเติมได้โดยนำการเรียนรู้แบบออฟไลน์ที่ใช้ในงานโกสเอไอ [9] [10] เข้ามาใช้ประกอบการเรียนรู้ด้วย

ในส่วนของจังหวะการแสดงออกท่านั้น อาจถือได้ว่าเป็นข้อจำกัดอย่างหนึ่งของเอไอที่นำเสนอซึ่งแก้ไขได้ยาก เนื่องจากท่าแต่ละท่านั้น สามารถใช้ต่อเนื่องกันได้ด้วยข้อกำหนดที่ต่างกัน

และวิธีการกดคำสั่งแต่ละท่า ก็มีข้อยืดหยุ่นได้หลายรูปแบบ รวมทั้งเวลาที่ใช้ในการกดคำสั่งก็มีปัจจัยด้านการสุ่มและความไม่แน่นอนของตัวเกมเข้ามาเกี่ยวข้องด้วย ผู้เล่นที่เป็นมนุษย์นั้น แม้ไม่สามารถวิเคราะห์ถึงช่วงเวลาและข้อมูลเหล่านี้ได้อย่างแม่นยำถูกต้องทั้งหมด แต่ด้วยการอาศัยประสบการณ์ในการเล่น จึงสามารถที่จะกะเวลาและควบคุมกระบวนการเรื่องจังหวะของท่านี้ได้ในระดับหนึ่ง แต่สำหรับฝ่ายเอไอแล้ว หากต้องมีการวิเคราะห์ถึงข้อมูลด้านนี้โดยละเอียด อาจไม่สามารถทำได้เนื่องด้วยข้อจำกัดของการที่ไม่สามารถเข้าถึงโค้ดของตัวเกมได้ ทำให้ไม่สามารถดึงเอาข้อมูลปัจจัยด้านเวลาของการส่งคำสั่งของตัวเกมทั้งหมดมาพิจารณาได้ นอกจากนั้นแล้ว แม้จะสามารถรู้ข้อมูลที่จำเป็นทั้งหมดได้ แต่ข้อมูลนั้นจะมีสิ่งที่ต้องพิจารณาเป็นจำนวนมาก ทั้งยังมีค่าและรายละเอียดปลีกย่อยอีกมากซึ่งต่างกันไปตามแต่ละตัวละคร รูปแบบการเล่น และตัวเกมต่อสู้เอง ดังนั้นสำหรับวิทยานิพนธ์ฉบับนี้ที่มุ่งเน้นไปที่การนำเสนอเทคนิคเอไอนั้น การแสดงออกการกระทำและส่งคำสั่งการกดท่าโดยใช้ช่วงเวลาเฉลี่ยที่แน่นอนค่าหนึ่ง จึงถือว่าเป็นวิธีการที่ให้ผลดีที่สุดแล้ว ซึ่งหากต้องการที่จะพัฒนาให้เอไอมีความสามารถที่จะยืดหยุ่นจังหวะการออกท่าได้จริงๆนั้น อาจนำเอาวิธีการเปรียบเทียบสัญญาณการกดปุ่มที่มนุษย์เป็นผู้กดที่ใช้ในงานโกสเอไอ [9] [10] มาใช้ เพื่อให้เอไอศึกษาจากวิธีการเล่นของผู้เล่น ว่าควรจะใช้จังหวะการกดปุ่มแบบใดอย่างไรก็ตามวิธีการนี้ใช้ได้กับการเรียนรู้แบบออฟไลน์เท่านั้น

สำหรับปัญหาเรื่องความรู้เรื่องการกระทำที่ควรนำมาใช้ในสถานการณ์แบบต่างๆนั้น เอไอที่ได้นำเสนอนี้ สามารถปรับปรุงให้มีความสามารถเพิ่มขึ้นได้ โดยใช้ข้อมูลฮิวริสติค (heuristic) จากมนุษย์ เข้ามาช่วยในการแนะแนวการเรียนรู้ลักษณะการกระทำที่เหมาะสมกับสถานการณ์ของเอไอให้เป็นไปในแนวทางที่ดีขึ้นได้ อย่างไรก็ตาม ควรจะเป็นการแนะแนวทางการเรียนรู้เท่าที่จำเป็นอย่างเหมาะสมก็เพียงพอแล้ว เนื่องจากหากมีการกำหนดแนวทางให้กับเอไอละเอียดมากจนเกินไป จะทำให้เอไอกลายเป็นเอไอแบบสคริปต์ไปได้ เพราะถูกกำหนดการกระทำไว้ทั้งหมดแล้ว ซึ่งขัดกับจุดมุ่งหมายของงานวิจัยที่มุ่งหาระบบเอไอที่สามารถเรียนรู้ได้ด้วยตัวเองเพื่อลบข้อจำกัดของเอไอระบบสคริปต์ที่ใช้กันอยู่ได้

สำหรับปัญหาจำนวนเคสที่น้อยของ AAI นั้น ต้องแก้ที่การบันทึกเคส อาจใช้การบันทึกเคสแบบ AGAI แล้วค่อยทำการเรียนรู้และทดลองผิดถูกด้วยวิธีของ AAI

รายการอ้างอิง

[1] Capcom Co., Ltd.. Street Fighter [Online]. Available from: http://www.streetfighter.com [2009, 10 Nov]

[2] SNK Playmore. The King of Fighters [Online]. Available from: http://kofaniv.snkplaymore.co.jp/english/index.html [2009, 10 Nov]

[3] Sega Corporation. Virtua Fighter [Online]. Available from: http://www2.sega.com/gamesite/vf5/phase2/index.html [2009, 10 Nov]

[4] Namco Bandai Games Inc.. SOULCALIBUR [Online]. Available from: http://www.soulcalibur.com [2009, 10 Nov]

[5] Namco Bandai Games Inc.. Tekken [Online]. Available from: http://tekken.namco.com [2009, 10 Nov]

[6] Nareyek, A. AI in Computer Games. ACM Queue February 2004.

[7] Thunputtarakul, W and Kotrajaras, V. AI-TEM: Testing AI in commercial fighting game with emulator. International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGames), Lousiville, Kentucky, USA, 2006.

[8] VisualBoyAdvance Team. VisualBoyAdvance [Online]. Available from: http://vba.ngemu.com [2009, 10 Nov]

[9] Thunputtarakul, W and Kotrajaras, V. Data analysis for Ghost AI creation in commercial fighting games. 8th International Conference on Intelligence Games and Simulation (GAME-ON 2007), Bologna, Italy, 2007.

[10] วรพจน์ ธัญภัทรกุล. การพัฒนาโกสเอไอและเทสเบดสำหรับเกมการต่อสู้โดยใช้อีมูเลเตอร์. วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. 2550.

[11] Spronck, P., Sprinkhuizen-Kuyper, I. and Postma, E. Online adaptation of game opponent AI with dynamic scripting. International Journal of Intelligent Games and Simulation (eds. N.E. Gough and Q.H. Mehdi), pp. 45-53. Vol.3, No.1, ISSN: 1477-2043, University of Wolverhampton and EUROSIS, March/April 2004.

[12] Spronck, P., Sprinkhuizen-Kuyper, I. and Postma, E. Enhancing the performance of dynamic scripting in computer games. Entertainment Computing - ICEC 2004 (ed. Matthias Rauterberg), Lecture Notes in Computer Science 3166, pp. 296-307. Springer-Verlag.E, 2004.

[13] Mitchell, T. Machine Learning. McGraw-Hill, 1997.

[14] Graepel, T., Herbrich, R. and Gold, J. Learning to fight. International Conference on Computer Games: Artificial Intelligence, Design and Education, 2004.

[15] Andrade, G., Ramalho, G., Santana, H. and Corruble, V. Automatic computer game balancing: a reinforcement learning approach. 4th International joint conference on Autonomous agents and multiagent systems, pp. 1111–1112. The Netherlands, 2005.

[16] Thawonmas, R. and Osaka, S. A method for online adaptation of computer-game AI rulebase. ACM SIGCHI international conference on Advances in computer entertainment technology, Hollywood, California, USA, 2006.

[17] Lueangrueangroj, S and Kotrajaras, V. Real-time imitation based learning for commercial fighting games. Computer Games, Multimedia & Allied Technology 2009, pp. 67-74. Singapore, 2009.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บรรณานุกรม

[1] Studio BENT STUFF. All About Street Fighter Zero 3: All About series vol.21. Printed in Japan: 1998.

[2] Kolodner, J. Case-Based Reasoning. Morgan Kaufmann Publisher, 1993

[3] Leake, D. Case-Based Reasoning: Experiences, Lessons, and Future Directions. AAAI Press / MIT Press, 1996

[4] Sutton, R. and Barto, A. Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). Cambridge, MA, MIT Press, 1998.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ผลงานการตีพิมพ์

1. Lueangrueangroj, Sarayut and Kotrajaras, Vishnu. Real-Time Imitation Based Learning for Commercial Fighting Games. In *Proc. Computer Games, Multimedia & Allied Technology 2009*, Singapore, 2009, pp. 67-74.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Real-Time Imitation Based Learning for Commercial Fighting Games

Sarayut Lueangrueangroj
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Payathai Road, Patumwan
Bangkok 10330, Thailand
Tel. +66813762763
Sarayut.L@student.chula.ac.th,
yutic@hotmail.com

Vishnu Kotrajaras
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Payathai Road, Patumwan
Bangkok 10330, Thailand
Tel. +66890212323
Vishnu@cp.eng.chula.ac.th,
ajarntoe@gmail.com

Abstract

Ghost AI is an Artificial Intelligence (AI) for fighting games which is capable of observing and imitating any player's style of play. By using Ghost AI, a player can play the game as if playing against another player. Although Ghost AI can simulate a human player, current research only applies it offline. Therefore, a human player can easily learn to beat Ghost AI during a match. We propose a methodology that allows Ghost AI to learn in real-time. The real-time data is also used to vary the frequency of each imitated action, so that the action can only be executed in situations which favor the AI. Our experimental result shows that the enhanced AI appears more human-like and more intelligent than the unmodified Ghost AI and provides a more satisfying experience for human players.

Keywords

Ghost AI, Imitation Based Learning, Dynamic Scripting, Fighting Game.

1. Introduction

Fighting games have a long history along with the history of video games. Fighting games have gone through numerous improvements over the years, but one aspect remains almost unchanged. It is artificial intelligence. Most fighting games utilize pre-scripted AI. It means that, for similar situations, the AI always behaves in the same manner. Commercial fighting games usually allow players to select various difficulty levels according to their skills. However, each difficulty level is still pre-scripted. Consequently, experience gamers find these games repetitive and boring. Most experienced players therefore opt to play against real humans because real humans provide greater challenges. Human players can imitate any useful move performed by an opponent who uses the same player character. They can learn and adapt their playing style during gameplay. Through trial and error, they can also learn how one move can be more effective than other moves for a given situation and select the best move for that situation or similar situations.

Better internet access and online play allow players to find their human opponents much easier than before. However, online play requires all participants to be connected at the same time. A computer controlled AI will still have to be used when online play is unavailable.

Some interesting technique was proposed in order to obtain a better computer controlled AI. Graepel et al. [1] applied reinforcement learning to a real commercial fighting game. It showed that machine learning techniques had good potential for applications in the domain of commercial games.

Another approach was imitation learning. This technique was used in Tekken 5 Dark Resurrection, for Sony PSP game system. When a player was playing, the game produced an imitation data file for him. Other players could then obtain the file in order to play with the imitative AI. There was a research which explored this technique. Ghost AI system [2] was proposed for commercial fighting games running on emulators. By recording actions carried out by a player in various situations, the system generated a probabilistically weighed case based AI that performed comparable to the player it imitated. Players were able to play with a Ghost AI as if playing with the person who trained that AI.

Since players vary in skill and different players play differently, even with the same character, a ghost AI system can provide players with many different challenges from a single game character. However, the Ghost AI system proposed in current research only learned offline. Therefore, its actions and style of play were only limited to what it learned in past matches. It would keep its play style even though the style was not useful in the match it was playing. This allowed human players to eventually spot its weaknesses and beat it using such weaknesses, rendering the same problem as pre-scripted AI. Human players, however, are able to change their play style during play to remove such weaknesses. In order to learn to fight other players effectively, the AI should not just imitate the players' moves from past matches. It should be able to reason when to (or not to) execute each of its learned move in real-time, just like a human player. If there are alternative moves available for a situation, the AI should learn to execute the move that troubles current opponent the most.

This paper proposes a technique to improve an imitation based AI for fighting games. By applying dynamic scripting [3] [4], an efficient unsupervised learning algorithm for a fighting game character is introduced. Our AI agent learns and improves itself from how its opponent plays the game in real-time. The AI observes, imitates its opponent's actions and adjusts how it executes its learned moves in real-time. This leads to a more challenging and more realistic imitation based AI opponent.

2. Testbed: AI-TEM Framework

Our proposed AI operated under AI-TEM (AI-Testbed in Emulator [5]), an efficient testbed environment we created for testing game AI. AI-TEM was created from Visual Boy Advance (VBA), an open source emulator for Nintendo Game Boy Advance (GBA). A commercial Game Boy Advance game ROM could be played on AI-TEM and used for testing AI. By using and controlling values stored in the memory of the emulator, game states and game behaviors could be controlled. Testing on actual commercial games guaranteed the quality of any developed AI. For this paper, we tested our imitation based AI on Street Fighter Zero 3 Upper, which was one of the well known fighting games on the system. The image of the game is shown in Figure 1.

Figure 1. Street Fighter Zero 3 Upper

3. Online Imitation Based AI

3.1 Overview

Our imitation based AI could be broken up into two parts. The first part handled imitation. The AI observed how a player responded to each situation and created cases from the observed data. A case contained the details of a game state and actions the player performed in that state. The AI used cases as its scripts. Therefore it played the game using the same style as the player it recorded its data from. The other part of the AI handled learning. In ordinary Ghost AI, the AI utilized each imitated action with the same frequency as played by its trainer. However, our system worked differently. When an action was executed, our system checked whether the action contributed to the fight score. The probability for the action to be executed in the same situation altered according to the observed fight score. Therefore the AI learned to utilize the play style it had imitated in a way that was most beneficial to itself.

3.2 Data in a Case

A case contained details of a game state, as viewed by the AI. A game state contains:

- The distance between the AI character and its opponent on the x-axis and the y-axis.
- The distance between the AI character and a projectile weapon (if there was any on the screen) on the x-axis and the y-axis.
- AI state.
- The list of actions a human player performed in response to that game state.

Possible values for each parameter are shown in Table 1. For the list of actions performed by the human player, each action stored:

- How often the human player executed the action in that situation.
- The action's weight, which was used in our learning mechanism.

The contents of each list of actions are displayed in Table 2.

Table 1. Game state details

Details	Value	
Delta X	1-9	
Delta Y	1-4	
Delta WX	1-3	
Delta WY	1-4	
AI State	1-5	
Player Actions List [0-76]	Freq	Weight

Table 2. List of actions and values in a case

Actions	Frequency	Weight
Action[0]	x	x
Action[1]	x	x
Action[2]	x	x
…	x	x
…	x	x
…	x	x
Action[76]	x	x

3.3 Imitation Process

When playing against a human player, whenever the player performed an action, the AI determined whether there was a matching case. Parameter values in each state were grouped into ranges in order to allow similar cases to be matched. The list of the player's actions was not used in this matching.

If the case did not exist in the database, the AI would add the case to the database using the game state details. The player's action would be recorded with an initial frequency (the value was 1) and an initial weight (the value was 0). If the case already existed, the AI would check whether the player's action was recorded in the list of actions for that case. If the action was in the list, the AI would then increase the frequency of that action by one. Otherwise, the AI would add that action to the list with the initial frequency and the initial weight. The imitation process is described in Figure 2.

3.4 Learning Process

During gameplay, when a character was dealt damage, the AI updated the weight of the action that created damage in that situation. Since our imitation cases were created when a human player performed actions and any actions the AI could perform came from those cases, it was certain that a case existed for the damage situation. When a character was damaged, it was either the AI character or the player character. If the AI character was damaged, the weight of the action that the AI character performed would decrease. At the same time, the weight of the action performed by the player character would increase. Similar updates took place when the player character took damage. This process is shown in Figure 3.

3.3 Using Cases to Play the Game

When playing against a human opponent, the AI would check if the opponent's state matched the situation details in one of our cases. The AI would check the followings:

- The distance between the player and the AI in the x-axis and the y-axis.
- The distance between the player and a projectile weapon on the x-axis and the y-axis.
- It also checked whether the player's state matched with the AI's state in the case.

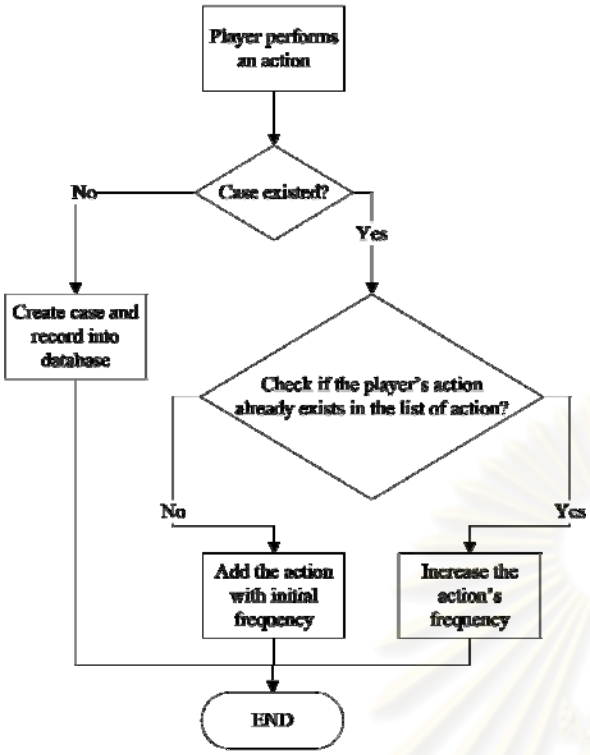


Figure 2. Imitation process

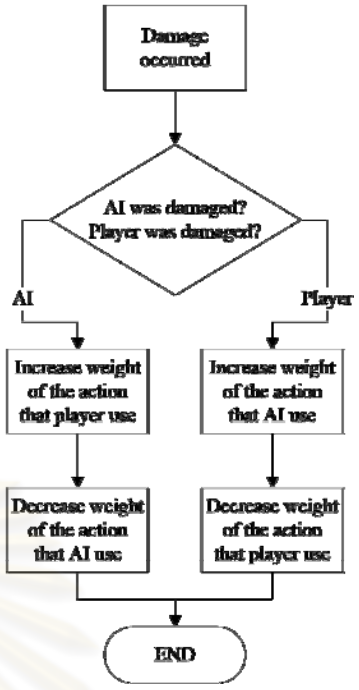


Figure 3. Learning process

If all values matched, it meant the player behaved in the same way as the AI when that case was created. The AI would then be able to choose an action from the player's action list to perform in order to imitate the player. If there was no match, the AI would perform a random action in order to create more chances for a matching case. Any random action that damaged the human-control character would not be used to update the weight because we only wanted to adjust actions obtained from imitation.

A player's action had its frequency and weight. The frequency reflected the imitation aspect and the weight reflected the learning aspect. To choose an action from the list of actions from a case, these two values had to be considered.

Action would be chosen based on probability. This process of choosing an action could be divided into two parts.

- First, the AI used the frequency value of each action to calculate a percentage value for each action. This percentage value was the probability for each action to be chosen based on imitation alone. The frequency based probability was represented by P_f.
- Second, the AI considered the weight value of each action and used it to calculate a percentage value for each action. This percentage value was the probability for each action based on the learned success rate of the action. This probability was represented by P_w.

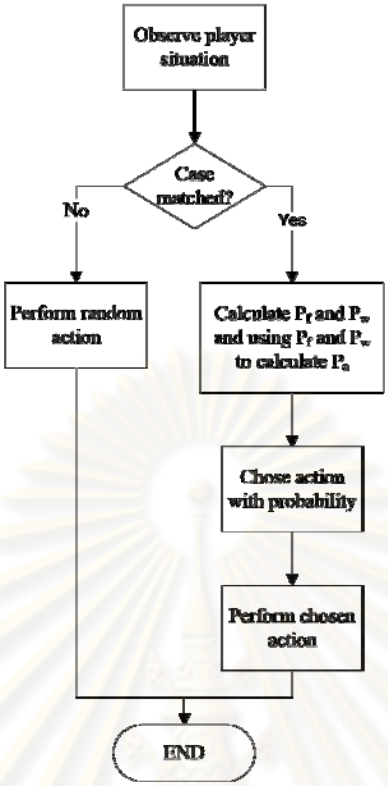


Figure 4. Choosing cases to play the game

The AI then computed an adjusted probability value for each action using P_f as a base value and using P_w to adjust P_f. The adjusted probability for an action was represented by P_a. P_a would be calculated using equation (1).

$$P_a = P_f + (C \times (P_f - P_w)) \quad (1)$$

Where C is a constant value (which is 0.1 in our experiment).

P_a would be used as a probability value for choosing each action in the list. The process of choosing cases is illustrated in Figure 4.

4. Experiment

We conducted an experiment to compare the performance between the Ghost AI (using the code for our AI, but performed only imitation, without learning) and our AI. In this experiment, we tested our AI with 9 human players; each player played against the AI for 10 matches, each match took 3 rounds to win.

First, we had human players play against the Ghost AI. We recorded damage data that each player did to the AI and damage data that AI did to the player for each round. The result is shown in Figure 5. Due to the reason of space, only the result from one person is shown. But all results were similar.

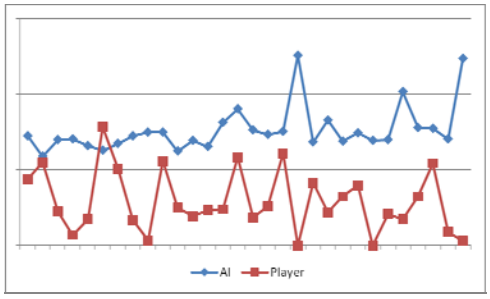


Figure 5. Ghost AI - Player

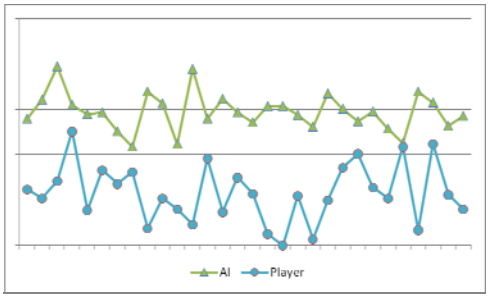


Figure 6. Imitation based learning AI - Player

Then, the experiment was conducted in the same way but human players played against our AI instead. The result of this experiment (for the same person from figure 5) is shown in Figure 6. The line referred to as AI line stands for the amount of damage AI received from a player for each round and the line referred to as Player line stands for the amount of damage a player received from the AI for each round.

4.1 Discussion

Experimental results would be considered in separate parts. The first part's result showed that for most players the amount of damage the Ghost AI received from the players increased over time. This implied that the players learned faster than the ordinary Ghost AI. The amount of damage the players received from the AI increased for a period of time, then suddenly decreased, then increased for another period of time. This was probably caused by the AI imitating players, learning to perform players' moves until players adjust their plays (represented by sudden decrease in damage caused by the AI). The players could exploit weakness from the Ghost AI by adjusting his style.

For the second part of the experiment, we could see that for most players the amount of damage that AI received from them was quite invariant, only slightly increasing at the end. This indicated that the human player still played better, but it was more difficult to see. The amount of damage that players received from the AI appeared to swing between increase and decrease period and appeared to be an invariant pattern similar to the AI. This was probably caused by the AI learning from players, performing better moves until players adjust their plays, similar to how the Ghost AI performed.

From the collected damage results alone, it could be concluded that our imitation based AI could perform online learning in real time, similar to the online Ghost AI. But human players still learned much faster. This caused the result from our AI to appear slightly better than the result from the Ghost AI.

Table 3. Players' opinions on which AI was cleverer and more human-like

Player	Ghost AI (GAI) / Imitation Based Learning AI (IMBL AI)
Player 1	GAI
Player 2	IMBL AI
Player 3	GAI
Player 4	IMBL AI
Player 5	IMBL AI
Player 6	IMBL AI
Player 7	IMBL AI
Player 8	IMBL AI
Player 9	IMBL AI

Besides the damage data, players who tested the AI had given their opinion on how both AI performed. Most players agreed that the imitation based learning AI could perform more clever maneuvers than the pure Ghost AI. They were satisfied with the performance of the imitation based AI. This information is shown in Table 3. From the graphs and players' opinions, our experimental result showed that the imitation based learning AI was able to learn and adapt itself to suit the game situation in real-time. The imitation based AI could perform cleverer moves and appeared more intelligent and human-like than the ordinary Ghost AI.

Conclusion

In this paper, we introduced a new technique for creating an AI in fighting game using Imitation based learning method. This technique had improved the Ghost AI technique, creating an AI that not only simulated the player style of play but also utilized the observed style in an efficient way to favor itself. The experimental results showed that our AI was able to improve itself overtime during gameplay. It also provided more realistic challenges for human players. In our future work, we intend to improve this AI further by introducing a feature for the AI to counter a player's maneuver.

References

[1] Graepel, T. Herbrich, R. and Gold, J. Learning to Fight. *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*. 2004.

[2] Thunputtarakul, W. and Kotrajaras, V. Data Analysis for Ghost AI Creation in Commercial Fighting Games. *8th International Conference on Intelligence Games and Simulation (GAME-ON 2007)*, Bologna, Italy.

[3] Spronck, P. Sprinkhuizen-Kuyper, I. and Postma, E. Online Adaptation of Game Opponent AI with Dynamic Scripting. *International Journal of Intelligent Games and Simulation (eds. N.E. Gough and Q.H. Mehdi), March/April 2004, Vol.3, No.1, ISSN: 1477-2043, pp. 45-53*. University of Wolverhampton and EUROSIS.

[4] Spronck, P. Sprinkhuizen-Kuyper, I. and Postma, E. Enhancing the Performance of Dynamic Scripting in Computer Games. *Entertainment Computing - ICEC 2004 (ed. Matthias Rauterberg), Lecture Notes in Computer Science 3166, pp. 296-307*. Springer-Verlag.

[5] Thunputtarakul, W. and Kotrajaras, V. AI-TEM: Testing AI in Commercial Fighting Game With Emulator. *International Conference on Computer Games: Artificial Intelligence and Mobile Systems*, 24-27 July 2006, Lousiville, Kentucky, USA.

ภาคผนวก ข

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายละเอียดปลีกย่อยเกี่ยวกับเอไอ

1. รายละเอียดของสถานการณ์ในเคส มีดังนี้

1.1. ระยะห่างระหว่างตัวละครฝ่ายเอไอและตัวละครฝ่ายผู้เล่นตามแกน x แบ่งได้เป็น 9 ช่วง โดยใช้การแบ่งช่วงตามระยะท่าของตัวละคร Ryu ที่ใช้เป็นกรณีศึกษา ดังนี้

ตารางที่ 7 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและผู้เล่นตามแนวแกน x

RangeDX_ID	Distance (dot unit)
1	10 – 44
2	45 – 68
3	69 – 78
4	79 – 97
5	98 – 109
6	110 – 117
7	118 – 162
8	163 – 222
9	> 222

1.2. ระยะห่างระหว่างตัวละครฝ่ายเอไอและตัวละครฝ่ายผู้เล่นตามแกน y แบ่งได้เป็น 4 ช่วง ได้แก่ ระยะที่ตัวละครอยู่บนพื้น ระยะที่สูงเป็นครึ่งหนึ่งของความสูงในการกระโดด ระยะที่สูงเท่ากับความสูงในการกระโดด และระยะที่สูงกว่าความสูงของการกระโดด

ตารางที่ 8 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและผู้เล่นตามแนวแกน y

RangeDY_ID	Distance (dot unit)
1	0
2	1 – 43
3	44 – 87
4	> 87

1.3. ระยะห่างระหว่างตัวละครฝ่ายเอไอและกระสุนของฝ่ายใดก็ตาม ตามแกน x แบ่งได้เป็น 3 ช่วง ได้แก่ สถานะที่ไม่มีกระสุนปรากฏอยู่ในฉาก ระยะที่สามารถกระโดดข้ามกระสุนได้ และระยะที่อยู่ไกลกว่าระยะที่สามารถกระโดดข้ามได้

ตารางที่ 9 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและกระสุนของฝ่ายใดก็ตาม ตามแนวแกน x

RangeDWX_ID	Distance (dot unit)
1	0
2	1 – 141
3	> 141

1.4. ระยะห่างระหว่างตัวละครฝ่ายเอไอและกระสุนของฝ่ายใดก็ตาม ตามแกน y แบ่งได้เป็น 4 ช่วง โดยมีรายละเอียดเช่นเดียวกับระยะระหว่างตัวละครของฝ่ายเอไอและตัวละครฝ่ายผู้เล่นตามแนวแกน y ดังนี้

ตารางที่ 10 รายละเอียดระยะห่างระหว่างตัวละครฝ่ายเอไอและกระสุนของฝ่ายใดก็ตาม ตามแนวแกน y

RangeDWY_ID	Distance (dot unit)
1	0
2	1 – 43
3	44 – 87
4	> 87

1.5. สถานะของตัวละครฝ่ายเอไอ แบ่งได้เป็น 5 รูปแบบ ดังนี้

ตารางที่ 11 รายละเอียดสถานะที่เป็นไปได้ของตัวละครฝ่ายเอไอ

Status_ID	Status
1	Attackable
2	Unattackable
3	Dizzy
4	Damaged
5	Performing an attack

จากข้อมูลสถานการณ์ในเคส รวมแล้วมีสถานการณ์ในเกมที่เป็นไปได้ทั้งหมด 9x4x3x4x5 = 2,160 รูปแบบที่เป็นไปได้

นอกจากนี้แล้วยังมีรายละเอียดสถานการณ์ในเกมข้ออื่น ที่มีการตรวจสอบข้อมูล แต่ไม่ได้มีการนำมาใช้เป็นส่วนประกอบของสถานการณ์ในเกมได้แก่

- สถานะติดมุมของตัวละคร แบ่งได้ 2 รูปแบบ ดังนี้

ตารางที่ 12 รายละเอียดแสดงสถานะติดมุมของตัวละคร

Corner_ID	Status
0	Not corner
1	Cornered

- รูปแบบสถานะโดยละเอียดทั้งหมดของตัวละครฝ่ายเอไอ มีทั้งหมด 77 แบบเมื่อแบ่งตามรูปแบบการกระทำที่เป็นไปได้ของตัวละคร Ryu

ตารางที่ 13 รูปแบบการกระทำทั้งหมดของตัวละคร Ryu

Action_ID	Actions
0	Stand
1	StepForward
2	StepBackward
3	Jump
4	Crouch
5	JumpForward
6	JumpBackward
7	Damage
8	Dizzy
9	ThrowDraw
10	MissThrow
11	AutoRoll
12	Unknown
13	b_Block
14	b_CrouchBlock
15	b_AirBlock
16	b_ThrowEscape
17	b_DizzyRecovery
18	b_DefensiveFall
19	b_DefensiveRoll
20	b_ZeroCounter
21	b_DamageReduction
22	b_Taunt
23	s_Jab
24	s_Straight
25	s_StrongJab
26	s_LowKick
27	s_HighKick
28	s_RoundhouseKick
29	cl_ElbowJab
30	cl_BodyBlow
31	cl_Uppercut
32	cl_AxeKick
33	cr_Jab
34	cr_Stright
35	cr_Uppercut
36	cr_ShortKick
37	cr_LongKick
38	cr_SweepKick
39	th_ShoulderToss

Action_ID	Actions
40	th_LegRoll
41	th_AirToss
42	a_DownwardJab
43	a_DownwardPunchM
44	a_UppercutL
45	a_UppercutR
46	a_DownwardPunchH
47	a_PuntKick
48	a_KneeDropL
49	a_KneeDropR
50	a_HighKick
51	a_ThrustKickML
52	a_ThrustKickMR
53	a_Roundhouse
54	a_ThurstKickHL
55	a_ThrustKickHR
56	t_SakotsuWari
57	t_Senpukyaku
58	t_FakeHadoken
59	sp_HadokenL
60	sp_HadokenM
61	sp_HadokenH
62	sp_ShakunetsuHadoukenL
63	sp_ShakunetsuHadoukenM
64	sp_ShakunetsuHadoukenH
65	sp_ShouryukenL
66	sp_ShouryukenM
67	sp_ShouryukenH
68	sp_TatsumakiSenpukyakuL
69	sp_TatsumakiSenpukyakuM
70	sp_TatsumakiSenpukyakuH
71	sp_AirTatsumakiSenpukyakuL
72	sp_AirTatsumakiSenpukyakuM
73	sp_AirTatsumakiSenpukyakuH
74	su_ShinkuHadoken
75	su_ShinkuTatsumakiSenpukyaku
76	su_MetsuShoryuken

2. การเข้ารหัสสถานการณ์ในเกม เพื่อให้ได้ค่า Casekey ซึ่งเป็นค่าตัวแทนของสถานการณ์นั้น

การเข้ารหัสสถานการณ์ในเกมให้เป็นค่า Casekey มีจุดมุ่งหมายเพื่อให้สามารถตรวจสอบและค้นหาสถานการณ์ในเกมได้อย่างรวดเร็ว โดยที่ไม่จำเป็นต้องเปรียบเทียบค่าสถานการณ์ในเกมทีละค่า แต่อาศัยเพียงการเปรียบเทียบตัวเลขชุดเดียวแทน โดยจะเก็บค่าเป็นข้อมูลขนาด 32 บิต (ไม่ได้ทำการใช้ข้อมูลทั้งหมด 32 บิต โดยจะมีบิตเหลือที่ไม่ได้นำมาใช้สามารถ

เพิ่มข้อมูลอื่นๆเข้าไปเพื่อใช้ทำการตรวจสอบได้หากต้องการ) ซึ่งมีขนาดเท่ากับตัวแปรอินทีเจอร์ของคอมไพเลอร์ส่วนมาก โดยมีรายละเอียดการเข้ารหัสดังนี้

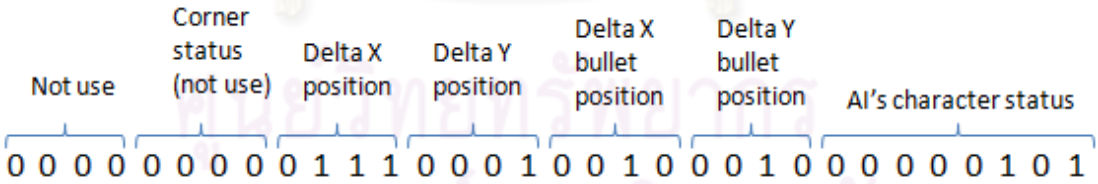
ตารางที่ 14 รายละเอียดการเข้ารหัสสถานการณ์ในเกม

Bit No.	Number of bits	Number of values	Description
0 – 7	8	5	AI's character status
8 – 11	4	4	Delta bullet position in y axis
12 – 15	4	3	Delta bullet position in x axis
16 – 19	4	4	Delta position in y axis
20 – 23	4	9	Delta position in x axis

สำหรับตัวอย่างการเข้ารหัสสถานการณ์ในเกมนั้น ยกตัวอย่างสถานการณ์ในเกมเช่น

- ตัวละครฝ่ายเอไอกำลังทำท่าโจมตีอยู่ จะมี Status_ID เท่ากับ 5
- ตัวละครฝ่ายเอไออยู่ห่างจากกระสุนที่ฝ่ายตรงข้ามปล่อยมาในแนวแกน y เป็นระยะ 25 หน่วย จะมี RangeDWY_ID เท่ากับ 2
- ตัวละครฝ่ายเอไอยู่ห่างจากกระสุนที่ฝ่ายตรงข้ามปล่อยมาในแนวแกน x เป็นระยะ 50 หน่วย จะมี RangeDWX_ID เท่ากับ 2
- ตัวละครทั้งสองฝ่ายอยู่ห่างกันในแนวแกน y เป็นระยะ 0 หน่วย แสดงว่าตัวละครทั้งสองฝ่ายอยู่บนพื้นทั้งคู่ จะมี RangeDY_ID เท่ากับ 1
- ตัวละครทั้งสองฝ่ายอยู่ห่างกันในแนวแกน x เป็นระยะ 135 หน่วยจะมี RangeDX_ID เท่ากับ 7

จากข้อมูลของสถานการณ์ในเกมข้างต้น สามารถนำมาเข้ารหัสเป็นค่า Casekey ได้ตามรูปที่ 19



รูปที่ 19 รายละเอียดการกำหนดค่าของบิตต่างๆในตัวแปรอินทีเจอร์ที่นำเอามาใช้เป็นค่า Casekey

จากรูปแสดงค่าบิตต่างๆของตัวแปรอินทีเจอร์ซึ่งนำมาใช้เป็นค่า Casekey ซึ่งจากค่าต่างๆในบิตจะได้ Casekey มีค่าเท่ากับ 7414277 เป็นค่าตัวแทนของสถานการณ์ในเกมนั้น

ข้อมูลสถานการณ์ในเกมซึ่งไม่ได้มีการเอามาใช้ อันได้แก่สถานะติดมุมของตัวละคร และสถานะของตัวละครแบบละเอียดนั้น หากต้องการนำเอามาใช้ สามารถเอามาเพิ่มใช้เข้ารหัสได้โดยมีรายละเอียดดังนี้

ตารางที่ 15 รายละเอียดการเข้ารหัสที่ไม่มีการนำมาใช้

Bit No.	Number of bits	Number of values	Description
0 – 7	8	77	AI character actions
24 – 27	4	2	Character's corner status

สำหรับเหตุผลที่มีการตัดข้อมูลเรื่องสถานะติดมุมของตัวละครออก และเปลี่ยนการอธิบายสถานะของเอไอจากท่าการกระทำโดยละเอียดมาเป็นลักษณะโดยรวมนั้น เนื่องจากในการทำงานของเอไอนั้น ทำงานโดยใช้การเทียบเคียงกับสถานการณ์ที่เคยมีการบันทึกเอาไว้ รายละเอียดของสถานการณ์ที่เฉพาะเจาะจงมากเกินไปนั้น สามารถก่อให้เกิดปัญหาในการเล่นของเอไอได้ ยกตัวอย่างเช่นในการลอกเลียนการกระทำของผู้เล่น กรณีสถานะติดมุมของตัวละครหากนำเอามาใช้ด้วย จะทำให้ได้สถานการณ์ในเกมที่ต้องคำนึงถึงการที่ตัวละครกำลังติดมุมด้วย ซึ่งในการเล่นจริง เอไอที่ไม่มีความรู้ในการเล่นเกมเลย ในตอนแรกจะถูกฝ่ายตรงข้ามต้อนเข้าติดมุมได้ง่ายมาก ซึ่งเมื่อเป็นเช่นนั้น การกระทำหรือสถานการณ์ใดๆที่เกิดขึ้นต่อไป จะเกิดขึ้นที่มุมของฉากทั้งสิ้น ดังนั้นสถานการณ์ที่เอไอเอามาใช้ในการลอกเลียนการเล่นของฝ่ายตรงข้าม จะใช้ได้ในสถานการณ์ที่ฝ่ายตรงข้ามติดมุมเท่านั้น ซึ่งการที่เอไอจะทำให้ฝ่ายตรงข้ามไปติดมุมได้นั้นเป็นไปได้ยากมากโดยที่ไม่มีความรู้ในการเล่นเกมเลย ทำให้เอไอไม่สามารถใช้ความรู้ที่ได้เรียนรู้มาและโดนโจมตีอยู่ฝ่ายเดียวได้

ในส่วนของสถานะของเอไอก็สามารถอธิบายได้ด้วยเหตุผลเดียวกัน ในการเล่นเกมต่อสู้ที่สถานการณ์มีการเปลี่ยนแปลงอยู่ตลอดเวลานั้น การที่จะให้ตัวละครมีสถานะท่าการกระทำ และระยะทุกอย่างตรงกับที่เคยบันทึกไว้นั้น เป็นไปได้ยาก และต้องใช้เวลานานในการทำให้เกิดขึ้น ส่งผลให้การเรียนรู้ของเอไอต้องใช้เวลานานมากได้ ดังนั้นการเปลี่ยนมาใช้การอธิบายสถานะโดยรวมแทนการแจกแจงเป็นท่าแบบละเอียดจึงเหมาะสมมากกว่า ช่วยลดสถานการณ์ที่เอไอต้องเรียนรู้ไปได้มาก และในการเล่นเกมต่อสู้จริง ส่วนใหญ่แล้ว ผู้เล่นจะสนใจเพียงแค่สถานะโดยรวมของฝ่ายตรงข้ามเท่านั้น เช่น ผู้เล่นสนใจเพียงแค่ว่า ฝ่ายตรงข้ามกำลังอยู่ในสถานะแสดงท่าโจมตีอยู่ แต่ไม่ได้สนใจละเอียดไปถึงว่า กำลังแสดงท่าต่อยเบา หรือต่อยหนักอยู่

3. การเก็บและค้นหาข้อมูลสถานการณ์ในเกม

ค่า Casekey ซึ่งเป็นตัวแทนของข้อมูลสถานการณ์ในเกม จะถูกนำมาใช้เป็นตัวค้นหาในการค้นหาข้อมูลสถานการณ์ในเกมที่ต้องการ โดยข้อมูลเคสสถานการณ์ในเกมเมื่อถูกสร้างขึ้นมานั้น จะถูกจัดเก็บไว้ในฐานข้อมูลซึ่งสร้างจาก STL (Standard Template Library) ของ ภาษา C++ ประเภท map ซึ่งมีลักษณะเป็นต้นไม้ทวิภาคแบบดำแดง โดยให้ความเร็วในการค้นข้อมูลอยู่ที่ O(log n)

4. รายละเอียดเกี่ยวกับท่ากัน (guard)

ท่ากันนั้น เป็นอีกหนึ่งการกระทำที่สำคัญมากในการเล่นเกมต่อสู้ เนื่องจากบทบาทของท่าที่ใช้ในการป้องกันการโจมตีของฝ่ายตรงข้าม สำหรับเอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้นั้น สามารถเรียนรู้เอาท่ากัน มาใช้ในเคสได้สองทางคือ เรียนรู้เอาเองจากการสุ่มเอาท่ากันมาใช้ได้ และอีกวิธีการคือ มีการใส่ท่ากันพร้อมค่าน้ำหนักประสิทธิภาพค่าหนึ่งเข้าไปเป็นท่าพื้นฐานในเคสในตอนสร้างเคสทุกเคส ซึ่งจะมีคำสั่งเมนูที่ใช้เปิดหรือปิดการทำงานในการใส่ท่ากันพร้อมค่าน้ำหนักเข้าไปเป็นท่าพื้นฐานในขณะที่สร้างเคสได้ ซึ่งเอไอที่ทำการทดลองในวิทยานิพนธ์ฉบับนี้จะปิดการทำงานตรงส่วนนี้ไว้ ไม่มีการใส่ท่ากันเป็นท่าพื้นฐานในขณะที่สร้างเคส แล้วให้เอไอทำการเรียนรู้ที่จะใช้ท่ากันเองเอง

5. การเคลื่อนที่เปลี่ยนระยะของตัวละคร

การเคลื่อนที่เพื่อสร้างระยะระหว่างตัวละคร เป็นอีกหนึ่งการกระทำที่มีความสำคัญในการเล่นเกมต่อสู้ ท่าทางต่างๆของตัวละครที่แสดงออกได้นั้น มีระยะที่สามารถใช้งานได้แตกต่างกันไป ดังนั้นระยะระหว่างตัวละครจึงมีผลต่อการแสดงออกท่าทางของตัวละครด้วย การเคลื่อนที่นั้นสามารถทำได้ด้วยการใช้การกระทำที่เป็นการเคลื่อนที่ ซึ่งมีท่าหลักได้แก่การเดินไปข้างหน้า เดินไปข้างหลัง การกระโดดไปข้างหน้า และกระโดดไปข้างหลัง นอกจากนั้นยังมีการกระทำที่เป็นการโจมตีบางท่าที่มีผลในการเคลื่อนที่ของตัวละครด้วย เช่น ท่าหมุนตัวเตะเป็นต้น ในการที่ตัวละครจะเคลื่อนที่ได้นั้น เอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้ สามารถทำได้สองวิธีด้วยกัน วิธีแรกคือให้เอไอทำการเรียนรู้เอาเอง เมื่อมีท่าสำหรับเคลื่อนที่ในเคสแล้ว เอไอก็จะสามารถทำการเคลื่อนที่ได้ แต่ว่าเนื่องจากท่าที่ใช้สำหรับเคลื่อนที่นั้น ไม่สามารถทำความเสียหายฝ่ายตรงข้ามได้ ดังนั้นจะไม่มีการได้คะแนนน้ำหนักความสามารถของท่าจากการสร้างความเสียหายแก่ฝ่ายตรงข้าม จะได้จากการเป็นท่าที่นำไปสู่การกระทำที่สร้างความเสียหายได้เท่านั้น ดังนั้น การที่ตัวละครจะ

เคลื่อนที่ก็ขึ้นอยู่กับคะแนนความสามารถของท่าเคลื่อนที่ และค่าความน่าจะเป็นที่จะถูกเลือกออกมาใช้เท่านั้น

อีกวิธีหนึ่งก็คือ ทุกครั้งที่เอไอได้รับหรือสามารถสร้างความเสียหายกับฝ่ายตรงข้ามได้ เอไอจะมีการบันทึกช่วงระยะที่สามารถทำความเสียหาย หรือถูกทำความเสียหายไว้ ซึ่งช่วงระยะนี้แบ่งตามระยะห่างระหว่างตัวละครของฝ่ายเอไอกับฝ่ายตรงข้ามในแนวแกน x ซึ่งแต่ละช่วงระยะจะมีคะแนนเป็นของตนเอง เมื่อเอไอสร้างความเสียหายให้แกฝ่ายตรงข้ามได้ ช่วงระยะที่เอไออยู่ตอนนั้นจะได้คะแนน ในทางกลับกัน ช่วงระยะที่เอไอถูกฝ่ายตรงข้ามสร้างความเสียหายให้จะถูกลดคะแนนลง เมื่อเอไอจะทำการเลือกท่ามาเพื่อใช้ในสถานการณ์ต่างๆ หากเอไอไม่ได้อยู่ในช่วงระยะที่มีคะแนนดี มีโอกาส 50% ที่เอไอจะทำการเคลื่อนที่แทนการเลือกใช้ท่าจากเคส ตำแหน่งที่เอไอจะทำการเคลื่อนที่ไปเพื่อสร้างช่วงระยะที่เหมาะสมขึ้นนั้น ขึ้นกับคะแนนของช่วงระยะนั้น โดยเอไอจะพยายามทำการเคลื่อนที่ให้เข้าสู่ช่วงระยะที่มีคะแนนดี อย่างไรก็ตามในเกมต่อสู้ที่ตัวละครทั้งสองฝ่ายมีการเคลื่อนที่อยู่ตลอดเวลานั้น การที่จะให้เกิดช่วงระยะที่ต้องการขึ้นนั้น เป็นไปได้ยาก ดังนั้นจึงให้เอไอเคลื่อนที่เข้าสู่ช่วงระยะบวกหรือลบสองช่วงระยะ จากช่วงระยะเป้าหมายแทน ซึ่งจะช่วยให้เอไอไม่ต้องเคลื่อนที่บ่อยเกินไป การเคลื่อนที่ของเอไอแบบนี้นั้น มีเมนูคอยควบคุมการทำงานเปิดปิดได้ โดยเอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้จะปิดการทำงานเอาไว้ แล้วให้เอไอทำการเรียนรู้เองด้วยวิธีแรก

6. ข้อจำกัดของท่าตัวละครที่ใช้ในการทดลอง

ตัวละคร Ryu ที่ใช้ในการทดลองนั้น มีการกระทำทั้งหมด 77 รูปแบบ อย่างไรก็ตามในการทดลองใช้ท่าของเอไอในการเล่นเกมนั้น จะมีท่าซึ่งจัดว่าเป็นท่าที่จะไม่นำมาใช้ทดลองอยู่ 9 รูปแบบ ทำให้เหลือรูปแบบการกระทำที่เอไอจะต้องพิจารณาลองใช้อยู่ 68 รูปแบบ ในส่วนของสาเหตุที่ตัดท่ากระทำ 9 รูปแบบนี้ออกจากท่าที่เอไอจะนำมาลองใช้นั้น เนื่องจากว่าเป็นท่าที่เอไอไม่สามารถกระทำด้วยตัวเองได้ หรือเป็นท่าที่เอไอไม่สามารถทำได้อย่างแน่นอนโดยปราศจากความรู้เบื้องต้นในการเล่นเกม ซึ่งจัดแบ่งตามกลุ่มได้ดังนี้

- ท่าที่เอไอไม่สามารถกระทำได้เอง ซึ่งเป็นท่าที่เกิดจากการกระทำโดยฝ่ายตรงข้าม ได้แก่
    - Damage ท่าแสดงออกถึงการโดนโจมตี
    - Dizzy ท่าแสดงอาการมึนจากการถูกโจมตี
    - ThrowDraw ท่าที่แสดงอาการยกเลิกการจับทุ่มจากการที่ฝ่ายตรงข้ามแสดงการจับทุ่มพร้อมกับฝ่ายเอไอ

- MissThrow ท่าแสดงอาการจับทุ่มพลาด จากการที่ฝ่ายตรงข้ามหลุดจากการจับทุ่ม
- AutoRoll ท่าแสดงอาการกลิ้งหลบ หลังจากถูกฝ่ายตรงข้ามโจมตี
- Unknown เป็นค่าอนิเมชันที่ไม่แสดงท่าการกระทำใดๆ แต่มีค่าปรากฏในเกม

- ท่าที่เอไอไม่สามารถกระทำได้อย่างแน่นอนโดยปราศจากความรู้เบื้องต้นในการเล่นเกม
  - ThrowEscape ท่ากลับตัวลงพื้นหลังจากถูกจับทุ่ม เพื่อไม่ให้เกิดความเสียหาย ต้องใช้จังหวะแม่นยำในการกดปุ่มสูง จำเป็นต้องใช้สคริปต์ในการทำให้แสดงออก
  - DizzyRecovery ท่าที่ช่วยให้ฟื้นตัวจากอาการมึนได้เร็วขึ้น ใช้การกดปุ่มทิศทางหลายครั้งอย่างรวดเร็วขณะตัวละครแสดงอาการมึน ไม่มีตัวบ่งชี้ใดๆแสดงการใช้ท่านี้ของตัวละครเพราะท่าอนิเมชันที่แสดงออกเป็นอนิเมชันของการมึนเพียงอย่างเดียว ดังนั้นการจะออกท่านี้ได้จำเป็นต้องอาศัยสคริปต์ในการสั่งการเท่านั้น
  - DamageReduction ท่าช่วยลดความเสียหายที่เกิดกับตัวละครขณะโดนโจมตี ใช้การกดปุ่มขณะโดนโจมตีด้วยความแม่นยำ ซึ่งอนิเมชันของท่านี้นั้นไม่มีการแสดงออก สิ่งที่ผู้เล่นจะสังเกตได้คือ พลังชีวิตที่ลดอย่างน้อยลงเมื่อถูกโจมตีเท่านั้น ซึ่งหลายๆครั้งไม่สามารถสังเกตเห็นได้ การแสดงออกท่านี้จึงใช้การเรียนรู้ได้ยาก จำเป็นต้องใช้สคริปต์เข้ามาช่วย

7. ท่าที่เป็นกรณีพิเศษ

ท่าในกลุ่มที่เป็นกรณีพิเศษนี้ ได้แก่ท่าในกลุ่มที่มีการเคลื่อนที่ระหว่างการแสดงออกท่า เช่นท่าที่เป็นการกระโดดเข้ามาโจมตี และอีกกลุ่มหนึ่งคือ ท่าในกลุ่มปล่อยกระสุนโจมตี ซึ่งจัดเป็นท่าที่ต้องคำนึงถึงเป็นกรณีพิเศษเนื่องจากสถานการณ์ตอนที่แสดงท่าออกมา และสถานการณ์ตอนที่ถูกโจมตีด้วยท่านั้น สามารถที่จะเป็นคนละสถานการณ์กันได้ ซึ่งต่างจากท่าโดยทั่วไปที่จะเป็นสถานการณ์เดียวกัน

7.1 กรณีท่ากระโดดเข้ามาโจมตี

กรณีท่ากระโดดเข้ามาโจมตี เช่น ท่ากระโดดเข้ามาเตะ ยกตัวอย่างกรณีผู้เล่นใช้ท่ากระโดดเตะเข้ามาโจมตีโดนเอไอ สถานการณ์ตอนที่เริ่มแสดงออกท่านั้น จะเป็นสถานการณ์ที่ตัวละครของทั้งสองฝ่ายอยู่ห่างกันระยะหนึ่ง และเมื่อตัวละครฝ่ายผู้เล่นเริ่มแสดงออกท่า จะแสดงท่ากระโดดเข้ามาก่อนที่จะแสดงท่าเตะเพื่อโจมตีฝ่ายตรงข้าม ซึ่งจะมีการเกิดคู่อันดับของ Casekey

และท่าการกระทำขึ้นมาสองคู่อันดับ คู่อันดับแรกแสดงข้อมูลตอนที่ตัวละครฝ่ายผู้เล่นเริ่มแสดงท่ากระโดด คู่อันดับที่สองแสดงตอนที่ตัวละครฝ่ายผู้เล่นอยู่กลางอากาศและแสดงท่าเตะ(บันทึกเป็นท่า กระโดดเตะ ในการอิมพลีเมนต์) ในตอนที่กระโดดเข้ามาแสดงท่าเตะนี้ จะเห็นว่าระยะระหว่างตัวละครทั้งสองได้เปลี่ยนไปแล้ว ทำให้สถานการณ์ที่เกิดขึ้นตอนที่โดนท่าเตะสร้างความเสียหายขึ้นกลายเป็นอีกสถานการณ์หนึ่งซึ่งไม่ใช่สถานการณ์เดียวกับตอนที่แสดงท่ากระโดด ซึ่งหากบันทึกเคสหรือปรับปรุงข้อมูลเคสแล้วนำไปใช้ทั้งแบบนี้ จะทำให้เกิดความผิดพลาดในการแสดงออกท่าขึ้น เพราะในการบันทึกเคสจะบันทึกในขณะที่เกิดความเสียหายขึ้น เคสที่บันทึกจะเป็นเคสในขณะที่ตัวละครฝ่ายตรงข้ามถูกเตะไปแล้ว ซึ่งมีระยะห่างระหว่างตัวละครค่าหนึ่ง โดยมีท่าหลักของสถานการณ์เป็นท่ากระโดดเตะ ซึ่งหากมีการบันทึกเอาไปใช้จริง แทนที่ตัวละครจะเริ่มกระโดดเข้ามาเตะที่ระยะห่างค่าหนึ่งที่เหมาะสม จะกลายเป็นว่า ตัวละครจะเริ่มกระโดดเตะที่ระยะซึ่งลูกเตะเคยสัมผัสคู่ต่อสู้ไปแล้ว ซึ่งเป็นระยะที่ใกล้เกินไป เมื่อเป็นเช่นนั้นจะทำให้ตัวละครกระโดดข้ามตัวฝ่ายตรงข้ามไปก่อนแล้วจึงแสดงท่าเตะ ซึ่งท่าเตะนั้นจะไม่โดนคู่ต่อสู้ ตรงจุดนี้เอไอที่ได้นำเสนอจะทำการแก้ไข โดยในการบันทึกข้อมูล จะบันทึกเอาสถานการณ์ที่เริ่มแสดงท่ากระโดดที่ถูกต้องเป็นสถานการณ์หลัก พร้อมเอาท่ากระโดดเตะมาเป็นท่าหลักของสถานการณ์นั้นแทนท่าเดิมที่เป็นท่ากระโดดธรรมดา ซึ่งจะทำให้ได้สถานการณ์ในการใช้ท่ากระโดดเตะที่ถูกต้อง

7.2 กรณีท่าปล่อยกระสุนโจมตี

กรณีท่าปล่อยกระสุนโจมตีนี้จะคล้ายกับกรณีท่ากระโดดเข้ามาโจมตีคือมีการเปลี่ยนแปลงเกิดขึ้นกับสถานการณ์ของตัวละคร แต่จะพิเศษกว่ากรณีของท่ากระโดดเข้ามาโจมตีเพราะสามารถมีการเปลี่ยนแปลงสถานการณ์เกิดขึ้นได้หลายแบบ และต้องพิจารณาทั้งส่วนท่าที่ทำให้เกิดความเสียหายและท่าที่ถูกสร้างความเสียหายให้ เนื่องจากท่าปล่อยกระสุนโจมตีนั้น หากไม่โดนฝ่ายตรงข้ามทันทีอาจมีช่วงเวลาที่นานตั้งแต่เริ่มแสดงออกท่าจนถึงท่านั้นแสดงผลได้ ตัวกระสุนที่ปล่อยออกมานั้น ต้องใช้เวลาในการเคลื่อนที่ ซึ่งระหว่างที่กระสุนกำลังเคลื่อนที่อยู่นั้นสถานการณ์ในเกมก็สามารถเปลี่ยนแปลงไปได้ ดังนั้นต้องพิจารณาเป็นสองส่วนคือ สถานการณ์ที่ทำการปล่อยกระสุน และสถานการณ์ที่ถูกกระสุนโจมตีสร้างความเสียหาย

ในส่วนแรกนั้นจะเหมือนกับกรณีท่ากระโดดเข้ามาโจมตี คือขณะที่ปล่อยกระสุนโจมตีเป็นสถานการณ์หนึ่ง แต่เมื่อตัวกระสุนนั้นเดินทางจนไปสร้างความเสียหายแก่ฝ่ายตรงข้ามนั้นเป็นอีกสถานการณ์หนึ่ง ทำให้เคสที่ถูกบันทึกหรือปรับปรุงนั้นผิดเพี้ยนไป ยกตัวอย่าง พิจารณาสถานการณ์จากรูป 20 และ 21

รูปที่ 20 สถานการณ์ขณะตัวละครฝ่ายผู้เล่นทำการปล่อยกระสุนโจมตี

รูปที่ 21 สถานการณ์ขณะที่ตัวละครฝ่ายเอไอโดนโจมตีด้วยกระสุน

จากรูปที่ 20 ในขณะที่ตัวละครฝ่ายผู้เล่นทำการปล่อยกระสุนโจมตีนั้น สถานการณ์ในเกมเป็นสถานการณ์หนึ่ง เรียกว่าสถานการณ์ A มีระยะห่างระหว่างตัวละครทั้งสองค่าหนึ่ง ในรูปที่ 21 เมื่อตัวละครฝ่ายเอไอถูกกระสุนของผู้เล่นโจมตี และระหว่างที่กระสุนของผู้เล่นกำลังเดินทางมานั้น ผู้เล่นได้ทำการกระโดดเข้ามาหาตัวละครฝ่ายเอไอ ตรงจุดนี้เมื่อกระสุนของผู้เล่นโจมตีถูกฝ่ายเอไอ จะเห็นว่าสถานการณ์ในเกมได้เปลี่ยนไปแล้ว ระยะระหว่างตัวละครทั้งสองฝ่ายเปลี่ยนไปกลายเป็นสถานการณ์ B เมื่อทำการคิดค่าคะแนนให้กับท่าที่สร้างความเสียหายได้คือท่าปล่อยกระสุน จากปกติที่จะทำการเพิ่มคะแนนให้กับท่าในเคสที่เป็นอยู่ขณะที่เกิดความเสียหายขึ้น ในกรณีนี้คือสถานการณ์ B ซึ่งจะทำให้เกิดความผิดเพี้ยนในการเรียนรู้ เนื่องจากสถานการณ์ B ไม่ใช่สถานการณ์ที่ทำการใช้ท่าปล่อยกระสุน วิธีการแก้ไขนั้น ทำได้โดยใช้หลักการเดียวกับกรณีท่ากระโดดเข้ามาโจมตี คือบันทึกเอาสถานการณ์ที่ทำการปล่อยกระสุนโจมตีพร้อมท่าปล่อยกระสุนเป็นสถานการณ์หลัก แทนสถานการณ์ที่กระสุนกระทบตัวฝ่ายตรงข้ามแล้ว ทำการเพิ่มค่าคะแนน

ให้กับท่าปล่อยกระสุนในเคสสถานการณ์ A แทน เนื่องจากเป็นเคสที่เกิดการใช้ท่าการปล่อยกระสุนนั้นจริงๆ

อีกส่วนหนึ่งนั้นเป็นส่วนสถานการณ์ที่ถูกโจมตีโดยกระสุนของฝ่ายตรงข้าม พิจารณารูป 20 และ 22

รูปที่ 22 สถานการณ์ขณะตัวละครฝ่ายเอไอกำลังจะถูกโจมตีด้วยกระสุน

จากรูปที่ 20 สถานการณ์ขณะตัวละครฝ่ายผู้เล่นทำการปล่อยกระสุนโจมตีเป็นสถานการณ์ A รูปที่ 22 สถานการณ์ขณะที่ตัวละครฝ่ายเอไอกำลังถูกโจมตีด้วยกระสุนเป็นสถานการณ์ C ซึ่งทั้งสถานะของตัวเอไอและระยะห่างระหว่างตัวละครทั้งสองได้เปลี่ยนไปแล้ว เมื่อทำการคิดค่าคะแนนให้กับท่าที่ใช้แล้วถูกสร้างความเสียหายให้ โดยปกติแล้วท่าที่เมื่อใช้แล้วทำให้ถูกโจมตีจะถูกลดค่าน้ำหนัก ซึ่งในกรณีของการถูกกระสุนโจมตีนี้ จะมาลดค่าน้ำหนักท่าในเคสสถานการณ์ C ตอนที่โดนโจมตีนี้ เนื่องจากกระสุนใช้เวลาในการเดินทาง สถานการณ์ที่ถูกโจมตีด้วยกระสุน (C) นี้จึงเป็นอีกสถานการณ์หนึ่งที่ไม่ใช่สถานการณ์ที่ทำการปล่อยกระสุน (A) ดังนั้นเมื่อโดนกระสุนโจมตีจึงมาคิดลดค่าน้ำหนักของท่าในเคสสถานการณ์ C นี้

8. AI-TEM

สำหรับ testbed AI-TEM ที่ใช้ในวิทยานิพนธ์ฉบับนี้นั้น มีการปรับปรุงเพิ่มเติมจาก [9] สำหรับใช้ทดลองกับเอไอที่ได้นำเสนอในวิทยานิพนธ์ฉบับนี้ โดยในส่วนเมนูควบคุมเอไอนั้น จะมีคำสั่งเพิ่มขึ้นมาไว้สำหรับควบคุมการเรียนรู้ของเอไอดังนี้

- StartDynamicScript

  ใช้เปิดทำงานการเรียนรู้ของเอไอ

- PauseDScriptLearning

  ใช้หยุดการเรียนรู้ของเอไอชั่วคราว

- PauseImitation

  ใช้หยุดการแสดงออกการลอกเลียนรูปแบบการเล่นของผู้เล่นชั่วคราว อย่างไรก็ตามการเรียนรู้การลอกเลียนรูปแบบการเล่นของผู้เล่น ยังคงทำงานอยู่

- PauseCounter

  ใช้หยุดการแสดงออกการทดลองใช้ท่าของเอไอชั่วคราว อย่างไรก็ตามการเรียนรู้การทดลองใช้ท่า ยังคงทำงานอยู่

- PauseRandomMove

  ใช้เปิด/ปิด การสุ่มท่ามาแสดงออกของเอไอเมื่อสถานการณ์ในเกมไม่ตรงกับสถานการณ์ที่มีการบันทึกเอาไว้เลย ในการทำงานของเอไอจะปิดตัวเลือกนี้เอาไว้ นั่นคือให้เอไอทำการสุ่มท่ามาแสดงออกเมื่อสถานการณ์ในเกมไม่ตรงกับที่ได้บันทึกเอาไว้เลย

- NoDefaultGuard

  ใช้เปิด/ปิด การเพิ่มท่ากันพร้อมค่าน้ำหนักค่าหนึ่งเข้าไปเป็นท่าเบื้องต้นของทุกเคสในการสร้างเคส ในการทำงานของเอไอจะเปิดตัวเลือกนี้เอาไว้ ซึ่งจะทำให้ไม่มีการเพิ่มท่ากันเข้าไปเป็นท่าเบื้องต้นของทุกเคสตั้งแต่ตอนที่ทำการสร้างเคส เอไอจะทำการเรียนรู้ที่จะใช้ท่ากันจากการเล่นเอง

- NoMoving

  ใช้เปิด/ปิด การเคลื่อนที่ของเอไอที่จะใช้แทนการออกท่า ในการทำงานของเอไอจะเปิดตัวเลือกนี้เอาไว้ เอไอจะไม่ทำการเคลื่อนที่แทนการออกท่า แต่จะใช้การเรียนรู้เลือกท่าสำหรับใช้เคลื่อนที่มาแสดงออกแทน

- DScriptMode

  ใช้ควบคุมการบันทึกข้อมูลความเสียหายและพลังชีวิตของตัวละครในเกม โดยจะแบ่งเป็นส่วนย่อยได้แก่

  - DScriptTraining

    ใช้สำหรับเอไอ เมื่อทดลองในโหมดฝึกซ้อมของเกม โดยจะบันทึกข้อมูลคะแนนที่เอไอและผู้เล่นทำได้และพลังชีวิตที่เหลือเมื่อสั่งจบการทำงานของเอไอ ทั้งนี้จำเป็นต้องมีคำสั่งในการบันทึกเพราะโหมดนี้ไม่สามารถบันทึกผลตอนเล่นจบยกได้ เนื่องจากไม่มีการจบยก

  - DScriptVersus

ใช้สำหรับการทดลองเล่นกับผู้เล่น จะบันทึกข้อมูลเมื่อจบแต่ละยกของการเล่น

- PlayerTraining
  ใช้สำหรับผู้เล่น เมื่อทดลองในโหมดฝึกซ้อมของเกม โดยจะบันทึกข้อมูลเมื่อสั่งจบการทำงานของเอไอ

ในการทดลองได้ทำการทดลองเปิด/ปิดการทำงานของตัวเลือกต่างๆ โดยลักษณะที่ให้ผลการทำงานของเอไอดีที่สุดคือ จะปิดตัวเลือก PauseRandommove เอาไว้ เพื่อให้เอไอสามารถสุ่มท่ามาทดลองใช้ เพื่อพัฒนาการเรียนรู้ได้ แต่จะเปิด NoDefaultGuard และ NoMoving เอาไว้ ซึ่งสามารถอธิบายได้ดังนี้

สำหรับการปิดไม่ให้เพิ่มท่ากันพร้อมค่าน้ำหนักเข้าเป็นท่าพื้นฐานของทุกเคสนั้น เนื่องจากว่าท่ากันนั้น หากสามารถใช้ได้ผลก็จะได้คะแนนค่าน้ำหนักเพิ่มเติม ซึ่งจะทำให้ค่าน้ำหนักพื้นฐานที่มีอยู่แล้วนั้นเพิ่มขึ้นไปอีก สำหรับเอไอที่ในตอนเริ่มต้นไม่มีความรู้ในการเล่นแล้วนั้น การที่มีท่ากันเป็นท่าพื้นฐาน จะช่วยให้เอไอสามารถป้องกันความเสียหายจากฝ่ายตรงข้ามได้ในระดับหนึ่ง อย่างไรก็ตาม เมื่อสามารถป้องกันความเสียหายได้ ท่ากันจะได้ค่าน้ำหนักเพิ่มขึ้น ทำให้เอไอเอาแต่แสดงออกท่ากันเพียงอย่างเดียวโดยไม่เลือกที่จะแสดงออกอย่างอื่น ทำให้ส่งผลต่อการเรียนรู้ในที่สุด

การปิดไม่ให้เอไอทำการเคลื่อนที่แทนการแสดงออกท่านั้น จากที่ได้อธิบายไปในหัวข้อการเคลื่อนที่เพื่อเปลี่ยนระยะแล้ว แม้จะอนุโลมให้เอไอไม่จำเป็นต้องเคลื่อนที่เข้าสู่ระยะที่ได้ผลการเล่นดีที่สุด โดยให้ใช้ระยะข้างเคียงได้แล้ว แต่จากธรรมชาติของเกมต่อสู้ที่ตัวละครมีการเคลื่อนที่ตลอดเวลา พบจากการทดลองว่าเอไอยังคงทำการเคลื่อนที่มากเกินไป โดยที่ไม่ได้แสดงออกการกระทำอื่นเพื่อเรียนรู้ ทำให้ส่งผลต่อการเรียนรู้ของเอไอ จึงเปิดตัวเลือกนี้แล้วให้เอไอทำการเรียนรู้ที่จะเคลื่อนที่ด้วยตนเองแทน

9. รายละเอียดข้อมูลที่ได้จากอีมูเลเตอร์

ข้อมูลสถานการณ์ในเกมนั้น สามารถตรวจสอบได้จาก testbed AI-TEM ซึ่งข้อมูลที่ได้จากเกม Street Fighter Zero 3 Upper ประกอบไปด้วย

- ID ของตัวละคร
- ประเภท ISM ของตัวละคร
- ค่าพลังชีวิตของตัวละคร
- ค่าพลังการ์ดของตัวละคร

- ค่าพลัง ISM ของตัวละคร
- ตำแหน่งของตัวละครในแนวแกน x
- ตำแน่งของตัวละครในแนวแกน y
- ตำแหน่งของกระสุนในแนวแกน x
- ตำแหน่งของกระสุนในแนวแกน y
- หมายเลขอนิเมชันของตัวละคร
- ค่าแสดงว่ามีกระสุนในฉากหรือไม่
- ค่าความเสียหายของตัวละคร

10. เฟรม (Frame)

สำหรับเกมต่อสู้นั้นจะใช้หน่วยเวลาเป็นเฟรม โดย 1 เฟรมจะเท่ากับ 1 รอบการทำงานของโปรแกรมเกมนั่นเอง โดยส่วนใหญ่แล้ว ในเกมต่อสู้จะมีจำนวนเฟรม 60 เฟรมต่อ 1 วินาที สำหรับเอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้ จะทำการพิจารณาสถานการณ์ในเกมและแสดงออกการกระทำทุกๆ 6 เฟรม ด้วยเหตุผลตามที่อธิบายไปในบทที่ 5 หัวข้อ 5.3.2

11. การเรียนรู้การกระทำเพื่อหวังผลในระยะยาว

ในส่วนการเรียนรู้การกระทำเพื่อหวังผลระยะยาวที่ได้อธิบายไปในบทที่ 4 หัวข้อ 4.5.5 นั้น สำหรับเอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้ ค่าคะแนนประสิทธิภาพที่ให้เพิ่มเติมกับท่าในสถานการณ์รองจะให้ค่าน้ำหนักเท่ากับ ค่าน้ำหนักที่ให้กับท่าในสถานการณ์หลัก – x โดย x จะมีค่าเริ่มที่ 1 และเพิ่มขึ้นทีละ 1 ไปเรื่อยๆตามลำดับสถานการณ์รองที่คิดย้อนกลับไป จนมีค่าสูงสุดเท่ากับค่าน้ำหนักที่ให้กับท่าในสถานการณ์หลัก – 1 เพื่อให้เอไอสามารถนำเอาสถานการณ์รองมาคิดในการเรียนรู้ให้ได้มากที่สุด

12. ข้อจำกัดของเอไอ

เอไอที่นำเสนอในวิทยานิพนธ์ฉบับนี้นั้นมีข้อจำกัดอยู่หนึ่งอย่างคือ ไม่สามารถที่จะแก้ทางรูปแบบการโจมตีแบบดักจังหวะได้เลย การโจมตีแบบดักจังหวะนั้น ยกตัวอย่างเช่น กรณีที่ถูกโจมตีด้วยท่าก้มเตะล่างซึ่งจะมีผลทำให้ตัวละครล้มลงไปนอน ก่อนที่ตัวละครจะกลับขึ้นมายืน ผู้เล่นที่เป็นมนุษย์นั้นสามารถกะจังหวะโจมตีต่อด้วยท่าโจมตีใดๆ เช่นทำการก้มเตะล่างต่อเนื่องอีกครั้งให้โดนตัวละครของเอไอพอดีในช่วงที่ตัวละครลุกขึ้นมายืนพอดีได้ ซึ่งหากฝ่ายที่โดนโจมตีเป็นมนุษย์ด้วยกัน สามารถกดคำสั่งกันล่วงหน้าค้างเอาไว้ก่อนตั้งแต่ตัวละครยังแสดงท่านอนอยู่และ

เมื่อตัวละครลุกขึ้นมาก็จะสามารถแสดงท่ากันป้องกันการโจมตีได้ในทันที แต่ในกรณีที่เอไอเป็นฝ่ายควบคุมตัวละคร จากรายละเอียดเรื่องเฟรมและการกดคำสั่งท่า ตัวละครของฝ่ายเอไอนั้นไม่สามารถที่จะทำแบบผู้เล่นได้ เนื่องจากจำนวนเฟรมระหว่างที่ตัวละครนอนอยู่จนถึงลุกขึ้นมานั้นไม่เพียงพอที่จะให้เอไอสามารถแสดงท่าการกระทำใดๆออกมาได้เลย ดังนั้นจะทำให้เอไอไม่สามารถหลีกเลี่ยงการโจมตีเช่นนี้ได้ ดังนั้นในการทดลองจะมีข้อจำกัดห้ามผู้เล่นทำการโจมตีที่อาศัยจุดอ่อนนี้

13. การสุ่มท่าเพื่อใส่ลงไปในส่วนเก็บท่าสำหรับทดลองใช้ในเคส

ในการสุ่มท่ามาเพื่อใช้สำหรับทดลองนั้น อาจมีคำถามว่าหากท่าที่สุ่มมานั้น ซ้ำกับท่าที่มีอยู่ในส่วนเก็บท่าสำหรับทดลองแล้วจะทำอย่างไร คำตอบคือจะไม่ทำการสุ่มท่าใหม่ โดยจะทำการเพิ่มหมายเลขของท่านั้นเข้าไปอีกหนึ่งเพื่อให้ได้ท่าอื่น ซึ่งหากยังซ้ำอีกก็จะเพิ่มหมายเลขของท่าเข้าไปอีก จนกว่าจะเจอท่าที่ยังไม่เคยบันทึกเอาไว้ แล้วจึงบันทึกท่านั้นเข้าไป ซึ่งจะช่วยให้การเรียนรู้ของเอไอรวดเร็วยิ่งขึ้น เพราะสามารถมีท่าใหม่ๆมาทดลองใช้ได้เร็วขึ้น หากในเคสนั้น ส่วนเก็บท่าสำหรับทดลองมีท่าครบทุกท่าแล้ว ในอนาคตก็ไม่จำเป็นต้องทำการสุ่มท่ามาเพิ่มอีกต่อไป วิธีการทำแบบนี้ ได้รูปแบบมาจากวิธีเก็บข้อมูลวิธีหนึ่งของโครงสร้างข้อมูลแบบตารางแฮช

14. การตรวจสอบสถานการณ์และเลือกใช้ท่า

ในหัวข้อ 4.5.4 อธิบายถึงการเลือกใช้ท่าของเอไอโดยอ้างอิงกับสถานการณ์ในเกม ซึ่งในกรณีที่ทั้งฝ่ายเอไอและผู้เล่นอยู่ในสถานการณ์ที่ตรงกับที่บันทึกเอาไว้ทั้งคู่ จะทำการสุ่มที่จะใช้ท่าเพื่อลอกแบบการเล่นของผู้เล่น หรือทดลองใช้ท่าใหม่อย่างใดอย่างหนึ่ง ตรงจุดนี้ที่ใช้การสุ่มเลือกการแสดงออก เนื่องจากได้ทำการทดลองพบว่าไม่ว่าจะแสดงออกด้านใดเป็นหลักก็ให้ผลที่ไม่แตกต่างกัน ขึ้นอยู่กับท่าการกระทำที่ใช้ออกมามากกว่าว่าสามารถให้ผลได้ดีแค่ไหน ซึ่งขึ้นกับการสุ่มท่าที่ใช้ได้ผลดีออกมา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นาย สรายุทธ เหลืองเรืองโรจน์ เกิดเมื่อวันที่ 5 มีนาคม พ.ศ. 2526 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาวิทยาศาสตรบัณฑิต สาขาฟิสิกส์ประยุกต์ จากภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2547 เมื่อจบการศึกษาแล้วได้เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2549 ระหว่างศึกษาในระดับมหาบัณฑิต มีผลงานทางวิชาการตีพิมพ์ 1 ฉบับในชื่อ Real-time Imitation based Learning for Commercial Fighting Game

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย