โครงการ
การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ ปัญญาประดิษฐ์ในเกมโปเกมอน
Artificial Intelligence in Pokémon Game

ชื่อนิสิต นายกวิน ไวทยวงศ์สกุล 6033602023

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2563

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปัญญาประดิษฐ์ในเกมโปเกมอน

นาย กวิน ไวทยวงศ์สกุล

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Artificial Intelligence in Pokémon Game

Kawin Waitayawongsakul

A Project Submitted in Partial Fulfillment of the Requirements

for the Degree of Bachelor of Science Program in Computer Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2020

Copyright of Chulalongkorn University

หัวข้อโครงงาน	ปัญญาประดิษฐ์เกมโปเกมอน
โดย	นายกวิน ไวทยวงศ์สกุล
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงงาน	อาจารย์ โชติรส สุรพลชัย

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

.....

(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงงาน

.....

(อาจารย์ โชติรส สุรพลชัย) อาจารย์ที่ปรึกษาโครงงาน

.....

(ผู้ช่วยศาสตราจารย์ ดร.กิติพร พลายมาศ) กรรมการ

.....

(อาจารย์ ดร.วุฒิชัย จงจิตเมตต์) กรรมการ

นายกวิน ไวทยวงศ์สกุล: ปัญญาประดิษฐ์ในเกมโปเกมอน (Artificial Intelligence in Pokémon Game) อ.ที่ปรึกษาโครงงาน : อ.โชติรส สุรพลชัย, 66 หน้า.

โครงงานเรื่อง “ปัญญาประดิษฐ์ในเกมโปเกมอน” มีวัตถุประสงค์ คือ เพื่อพัฒนาปัญญาประดิษฐ์ที่สามารถเล่นเกมโปเกมอนโชว์ดาวน์ได้ที่สามารถใช้เล่นแข่งขันกับมนุษย์ได้ มีจุดมุ่งหมายเพื่อสร้างความบันเทิงและช่วยในการทดสอบกลยุทธ์ในการเล่น ขอบเขตโครงงานภายใต้เกมโปเกมอนในรุ่นที่ 7 วิธีการทดลองจะใช้การเรียนรู้แบบเสริมกำลังโดยการสร้างโครงข่ายประสาทเทียมสำหรับการเรียนรู้แบบคิว ฝึกฝนโดยการเล่นกับตัวเอง และวัดผลโดยการเล่นกับการเล่นแบบสุ่ม ผลการทดลองที่ได้แสดงให้เห็นว่ามีอัตราชนะกับการเล่นแบบสุ่ม 67%

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต กวิน ไวทยวงศ์สกุล

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงาน

ปีการศึกษา 2563

##6033602023 : MAJOR COMPUTER SCIENCE

KEYWORDS : ARTIFICIAL INTELLIGENCE / NEURAL NETWORK / REINFORCEMENT LEARNING

KAWIN WAITAYAWONGSAKUL : ARTIFICIAL INTELLIGENCE IN POKÉMON GAME ADVISOR : CHOTIROS SURAPHOLCHAI, 66 pp.

The topic of this senior project is "Artificial Intelligence in Pokémon Game". The objective of this project is to develop an artificial intelligence that can play Pokémon Showdown with humans. The purposes of this project are for entertaining and testing strategies. The scope of this project is that the artificial intelligence can only play in the generation 7th format. The methods of this research are using reinforcement learning by creating deep Q-learning, training by self-playing, and evaluating by playing with a random agent. The result of this project shows that the artificial intelligence agent has the win-rate of 67 percents against the random agent.

Department :Mathematics and Computer Science Student's Signature กวิน ไวทยวงศ์สกุล

Field of Study : Computer Science Advisor's Signature

Academic Year : 2020...

กิตติกรรมประกาศ

โครงงานในหัวข้อเรื่อง “ปัญญาประดิษฐ์ในเกมโปเกมอน” ได้รับการสนับสนุนอย่างเต็มที่จากครอบครัว ในด้านการดูแลเอาใจใส่ระหว่างช่วงการทำโครงงานและจัดหาอุปกรณ์ที่มีประสิทธิภาพสูงไว้ให้ใช้ในการทำโครงงาน และคณาจารย์ในภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ในด้านองค์ความรู้เรื่องปัญญาประดิษฐ์และการเขียนรูปเล่มรายงาน อาทิ เช่น อ.โชติรส สุรพลชัย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ผศ.ดร.กิติพร พลายมาศ และ อ.ดร.วุฒิชัย จงจิตเมตต์ ซึ่งเป็นกรรมการ เป็นต้น รวมไปถึงเพื่อน ๆ ที่คอยช่วยให้คำแนะนำเกี่ยวกับความรู้เฉพาะด้านในเกมโปเกมอนและช่วยเหลือในด้านอื่น ๆ ซึ่งทำให้โครงงานนี้สำเร็จลุล่วงไปด้วยดี ข้าพเจ้าจึงใคร่ขอขอบพระคุณเป็นอย่างยิ่งสำหรับความช่วยเหลือในทุก ๆ ด้าน ที่ได้รับจากครอบครัว คณาจารย์ และเพื่อน ๆ และหวังว่าผลของโครงงานนี้จะเป็นประโยชน์ในการต่อยอดงานวิจัยด้านปัญญาประดิษฐ์ได้ต่อไป

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญภาพ ..... ญ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและเหตุผลการวิจัย 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตการวิจัย 2
1.4 ขั้นตอนการวิจัย 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 โครงสร้างของรายงาน 3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4
2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับโครงข่ายประสาทเทียมและวิธีการเรียนรู้แบบเสริมกำลัง 4
2.2 ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับปัญญาประดิษฐ์ในเกมโปเกมอน 5
2.3 การประยุกต์ใช้และทฤษฎีที่ใช้เพิ่มเติม 9
บทที่ 3 การออกแบบระบบ 11
3.1 ขั้นตอนวิธีที่ใช้ 11
3.2 การเชื่อมต่อเข้ากับเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์ 11
3.3 การสร้างตัวแบบโครงข่ายประสาทเทียมสำหรับการเรียนรู้แบบคิว 12
3.4 วิธีการฝึกฝนปัญญาประดิษฐ์ 17

3.5 วิธีการวัดผลปัญญาประดิษฐ์ 18
3.6 ภาพรวมระบบของโครงงาน 31
บทที่ 4 เทคนิคการเขียนโปรแกรมและอภิปรายผลการทดลอง 33
4.1 เทคนิคการเขียนโปรแกรม 33
4.2 ผลของการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่เหมือนกัน 33
4.3 ผลของการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่ต่างกัน 34
4.4 การอภิปรายผลการทดลอง 36
บทที่ 5 ข้อสรุปและข้อเสนอแนะ 38
5.1 ข้อสรุป 38
5.2 ปัญหาที่พบ 38
5.3 วิธีการแก้ปัญหา 38
5.4 ข้อเสนอแนะ 38
รายการอ้างอิง 40
ภาคผนวก ก แบบเสนอหัวข้อโครงงาน 43
ภาคผนวก ข คำศัพท์และกติกาที่เกี่ยวกับเกมโปเกมอน 48
ประวัติผู้เขียน 56

สารบัญตาราง

หน้า

ตารางที่ 3.1 ตารางข้อมูลที่จะป้อนเข้าสำหรับท่าโจมตีในโครงข่ายประสาทเทียม 12
ตารางที่ 3.2 ตารางข้อมูลที่จะป้อนเข้าสำหรับโปเกมอนในโครงข่ายประสาทเทียม 14
ตารางที่ 3.3 ตารางข้อมูลที่จะป้อนเข้าสำหรับสถานะเกมในโครงข่ายประสาทเทียม 15
ตารางที่ 3.4 ตารางค่าตอบแทน 16
ตารางที่ 3.5 ตารางจำนวนเกมที่ฝึกฝน 18
ตารางที่ 3.6 ตารางทีมที่นำมาใช้วัดผล 20
ตารางที่ 4.1 ผลการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่เหมือนกัน 34
ตารางที่ 4.2 ผลการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่ต่างกัน 35
ตารางที่ ข.1 ตารางแสดงการแพ้ชนะของประเภทแต่ละประเภทในเกมโปเกมอน 49
ตารางที่ ข.2 ตารางแสดงการเพิ่มลดค่าพลังตามลักษณะนิสัยของโปเกมอน 52

สารบัญภาพ

หน้า

ภาพที่ 3.1 แผนภาพกระแสข้อมูลระดับ 0 31
ภาพที่ 3.2 แผนภาพกระแสข้อมูลระดับ 1 31
ภาพที่ 3.3 แผนภาพกระแสข้อมูลระดับ 2 ในกระบวนการทำงานที่ 3 31
ภาพที่ 3.4 แผนภาพกระแสข้อมูลระดับ 2 ในกระบวนการทำงานที่ 4 32
ภาพที่ 3.5 แผนภาพกระแสข้อมูลระดับ 3 ในกระบวนการทำงานที่ 4.4 32
ภาพที่ 4.1 ตัวอย่างซอร์สโค้ดการคำนวณคะแนนมาตรฐาน 33
ภาพที่ 4.2 อัตราการชนะเมื่อเล่นกับการเล่นแบบสุ่มโดยใช้ทีมที่เหมือนกัน 34
ภาพที่ 4.3 อัตราการชนะเมื่อเล่นกับการเล่นแบบสุ่มโดยใช้ทีมที่ต่างกัน 35

บทที่ 1
บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาและเหตุผลการวิจัย วัตถุประสงค์ของการวิจัย ขอบเขตการวิจัย ขั้นตอนการวิจัย ประโยชน์ที่คาดว่าจะได้รับ และโครงสร้างของรายงาน

1.1 ความเป็นมาและเหตุผลการวิจัย

โปเกมอน [1] เป็นเกมเล่นตามบทบาทแบบเทิร์นเบส (Turn-based) ที่เกิดขึ้นในปี ค.ศ. 1996 มีการจัดการแข่งขันมากมาย โดยผู้เล่นจะต้องเตรียมทีมซึ่งประกอบไปด้วยโปเกมอน 6 ตัวเพื่อต่อสู้กับทีมของฝ่ายตรงข้าม โปเกมอนแต่ละตัวสามารถเรียนรู้ท่าโจมตีได้หลากหลายโดยผู้เล่นจะต้องเลือกมาเพียง 4 ท่าโจมตี อีกทั้งยังมีการให้โปเกมอนถือสิ่งของเพื่อเพิ่มความสามารถต่าง ๆ ทำให้โปเกมอนเป็นเกมที่มีแผนการเล่นที่หลากหลาย เนื่องจากโปเกมอนเป็นเกมที่มีกลยุทธ์ที่หลากหลายทำให้การพัฒนาปัญญาประดิษฐ์ในเกมโปเกมอนเป็นสิ่งที่น่าสนใจ

โปเกมอนโชว์ดาวน์ (Pokémon Showdown) [2] เป็นแพลตฟอร์ม (Platform) ที่ใช้ในการจำลองเกมโปเกมอนซึ่งจะเน้นไปที่การต่อสู้ของโปเกมอนเพียงอย่างเดียวเพื่อที่จะให้ผู้เล่นสามารถจัดทีมได้สะดวก เนื่องจากว่าผู้เล่นไม่ต้องทำการจับโปเกมอนและฝึกฝนโปเกมอนเองทำให้ผู้เล่นเลือกวางแผนกลยุทธ์ที่หลากหลายได้มากขึ้น นอกจากนั้นโปเกมอนโชว์ดาวน์เป็นแพลตฟอร์มที่เป็นโปรแกรมประยุกต์บนเว็บสามารถเชื่อมต่อเข้าถึงได้ง่ายทำให้สามารถเปิดให้ผู้ทดสอบและผู้ที่ต้องการจะเล่นกับปัญญาประดิษฐ์สามารถเล่นผ่านเครือข่ายอินเทอร์เน็ตได้ ดังนั้นผู้พัฒนาจึงเลือกใช้โปเกมอนโชว์ดาวน์เป็นแพลตฟอร์มสำหรับการพัฒนาปัญญาประดิษฐ์

โดยทั่วไปการพัฒนาปัญญาประดิษฐ์ในเกมที่เคยเอาชนะผู้เล่นที่เป็นผู้ชนะเลิศของการแข่งขันเกมนั้น ๆ มักจะใช้วิธีการเรียนรู้แบบเสริมกำลัง [3][4] ซึ่งเป็นการเรียนรู้ที่ดูผลตอบรับจากระบบของการกระทำ หากผลตอบรับระบุว่าดีจะส่งเสริมการกระทำดังกล่าวให้เลือกใช้มากขึ้น แต่ถ้าผลตอบรับไม่ดีจะลดการเลือกใช้การกระทำดังกล่าว เนื่องจากว่าเกมเป็นสิ่งที่มีผลลัพธ์ของการเล่นชัดเจนแต่ไม่สามารถบ่งบอกได้ชัดเจนว่า การตัดสินใจดังกล่าวถูกหรือผิดซึ่งเหมาะกับการใช้โมเดลการเรียนรู้แบบเสริมกำลัง การพัฒนาปัญญาประดิษฐ์ในโครงงานนี้จึงเลือกใช้โมเดลการเรียนรู้แบบเสริมกำลัง

โครงงานนี้จึงต้องการพัฒนาปัญญาประดิษฐ์ที่สามารถเล่นได้หลากหลายกลยุทธ์โดยเลือกใช้ตัวแบบการเรียนรู้แบบเสริมกำลังซึ่งผ่านการฝึกฝนที่สามารถเล่นกับผู้เล่น เพื่อสร้างความบันเทิงให้กับผู้เล่นและสามารถใช้ให้ผู้เล่นวางแผนการเล่นและทดสอบกลยุทธ์ได้ การฝึกฝนตัวแบบดังกล่าวจะทำโดยการให้ปัญญาประดิษฐ์ต่อสู้กับตัวเองซึ่งจะใช้ทีมจากระบบการสุ่มของโปเกมอนโชว์ดาวน์ซึ่งใช้วิธีการสุ่มแบบตัดกรณีที่ได้โปเกมอนที่ไม่เหมาะสม เช่น ได้ท่าโจมตี 2 ท่ามีลักษณะคล้ายกัน และเพิ่ม

ลำดับความสำคัญบางอย่าง เช่นการให้โปเกมอนถือสิ่งของเฉพาะตัวจะได้พลังที่เพิ่มขึ้นมาก การวัดผลตัวแบบดังกล่าวจะใช้การให้ต่อสู้กับการเล่นแบบสุ่มเพื่อให้สอดคล้องกับงานวิจัยอื่น ๆ [5][6] โดยจะเป็นการสุ่มแบบยูนิฟอร์ม (Uniform) เลือกจากการกระทำทั้งหมดที่เป็นไปได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาปัญญาประดิษฐ์ที่สามารถเล่นเกมโปเกมอนโชว์ดาวน์ได้ดีกว่าการเล่นแบบสุ่มบนระบบปฏิบัติการวินโดวส์ (Windows)

1.3 ขอบเขตการวิจัย

1. ปัญญาประดิษฐ์จะเล่นได้ในรูปแบบ [Gen 7] Anything Goes ซึ่งจะเป็นแบบฝ่ายละ 1 คนและใช้โปเกมอน 6 ตัว โดยใช้โปเกมอนได้ทีละ 1 ตัว
2. ปัญญาประดิษฐ์จะเล่นได้หลายทีมโดยไม่จำเป็นต้องฝึกฝนโมเดลใหม่
3. ปัญญาประดิษฐ์จะเล่นตามกติกาของโปเกมอนโชว์ดาวน์
4. ปัญญาประดิษฐ์จะทำงานได้ก็ต่อเมื่อการเชื่อมต่อกับเซิร์ฟเวอร์ (Server) ไม่เกิดข้อผิดพลาดและข้อความที่ได้รับจากเซิร์ฟเวอร์ครบถ้วนและเรียงลำดับถูกต้อง
5. ปัญญาประดิษฐ์จะไม่สามารถเล่นได้หากมีโปเกมอนที่มีความสามารถ Illusion
6. ปัญญาประดิษฐ์จะไม่เลือกใช้ Z-Move และไม่สามารถเล่นได้ถ้าฝ่ายตรงข้ามเลือกใช้สิ่งนั้น
7. ปัญญาประดิษฐ์จะเล่นได้ในระบบปฏิบัติการวินโดวส์ 10 64-bit
8. ปัญญาประดิษฐ์จะเล่นโดยใช้เซิร์ฟเวอร์จำลองที่เปิดขึ้นในคอมพิวเตอร์เครื่องเดียวกันโดยแพลตฟอร์มที่ใช้จะแสดงผลเป็นภาษาอังกฤษ และเชื่อมต่อกันผ่านเว็บซ็อกเก็ต (websocket) [7] โดยใช้ภาษาไพธอน (Python) เซิร์ฟเวอร์จำลองดังกล่าวจะใช้การโคลน (Clone) โอเพนซอร์ส (Open source) ของโปเกมอนโชว์ดาวน์ [2]
9. การวัดผลจะวัดผลเทียบอัตราแพ้ชนะโดยวิธีการเล่นแบบสุ่ม

1.4 ขั้นตอนการวิจัย

1. พัฒนาโปรแกรมส่วนเชื่อมต่อเข้ากับตัวเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์ [2] ที่จำลองในคอมพิวเตอร์เครื่องเดียวกันบนระบบปฏิบัติการ Windows 10 64-bit
2. ออกแบบตัวแบบที่ใช้ในงานปัญญาประดิษฐ์
3. พัฒนาโปรแกรมส่วนตัวปัญญาประดิษฐ์
4. ออกแบบวิธีการวัดผลและทดสอบการทำงานของปัญญาประดิษฐ์
5. วัดผลและทดสอบการทำงานของปัญญาประดิษฐ์

6. จัดทำเอกสารประกอบโครงงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ได้รับจากการวิจัยในครั้งนี้มีดังนี้

1. ประโยชน์ต่อผู้พัฒนาโปรแกรม
   1.1. ได้ฝึกฝนทักษะในการเขียนโปรแกรมภาษาไพธอนในการเชื่อมต่อกับเซิร์ฟเวอร์และการทำปัญญาประดิษฐ์
   1.2. ได้ฝึกฝนทักษะในการวางแผนตัวแบบที่ใช้ในปัญญาประดิษฐ์
2. ประโยชน์ต่อผู้ใช้
   2.1. ได้รับความบันเทิงในการรับชมหรือในการเล่นกับปัญญาประดิษฐ์
   2.2. ได้กลยุทธ์ในการเล่นใหม่ ๆ

1.6 โครงสร้างของรายงาน

บทที่ 2 จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับโครงข่ายประสาทเทียมและวิธีการเรียนรู้แบบเสริมกำลังและปัญญาประดิษฐ์ในเกมโปเกมอนที่มีการใช้วิธีการเรียนรู้แบบเสริมกำลัง

บทที่ 3 จะกล่าวถึงวิธีการพัฒนาปัญญาประดิษฐ์ในเกมโปเกมอนซึ่งจะประกอบไปด้วยการเชื่อมต่อเข้ากับเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์ การออกแบบตัวแบบสำหรับปัญญาประดิษฐ์ วิธีการฝึกฝนและการวัดผลปัญญาประดิษฐ์

บทที่ 4 จะกล่าวถึงเทคนิคการเขียนโปรแกรมและอภิปรายผลของการแข่งขันระหว่างปัญญาประดิษฐ์ในเกมโปเกมอนที่พัฒนาขึ้นในโครงงานนี้กับการเล่นแบบสุ่ม

และบทที่ 5 จะกล่าวถึงสรุปผลจากข้อมูลผลการทดลองในบทที่ 4 ข้อสรุปโครงงานปัญญาประดิษฐ์ในเกมโปเกมอนที่พัฒนาขึ้น ปัญหาที่พบและวิธีการแก้ปัญหา รวมถึงข้อเสนอแนะอื่น ๆ ที่ได้จากการโครงงาน

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับโครงข่ายประสาทเทียมและวิธีการเรียนรู้แบบเสริมกำลังและปัญญาประดิษฐ์ในเกมโปเกมอนที่มีการใช้วิธีการเรียนรู้แบบเสริมกำลัง

2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับโครงข่ายประสาทเทียมและวิธีการเรียนรู้แบบเสริมกำลัง

การสร้างตัวแบบโครงข่ายประสาทเทียมนั้นเป็นส่วนหนึ่งของการเรียนรู้ของเครื่องที่ใช้ในงานปัญญาประดิษฐ์ หลักการทำงานเบื้องต้นของโครงข่ายประสาทเทียมคือให้แต่ละโหนด (Node) แทนด้วยจำนวนจริง 1 จำนวน ในตัวแบบโครงข่ายประสาทเทียมจะประกอบไปด้วยหลายชั้น ซึ่งแต่ละชั้นประกอบด้วยหลายโหนด โดยโหนดในชั้นแรกสุดจะเป็นข้อมูลที่ป้อนเข้าให้โครงข่ายประสาทเทียม ในที่นี้จะเป็นสถานะของเกมซึ่งจะต้องเป็นในรูปแบบของจำนวนจริงหลาย ๆ จำนวน และโหนดในชั้นอื่น ๆ จะมาจากการคำนวณค่าจากค่าของโหนดในชั้นก่อนหน้าและค่าน้ำหนักของเส้นเชื่อมจากแต่ละโหนดในชั้นก่อนหน้า วิธีคำนวณจะใช้สมการดังนี้

$$x_j = f\left(\sum_{(i,j)\in E} w_{ij}x_i + b_j\right) \tag{1}$$

โดยตัวแปรต่าง ๆ มีความหมายดังนี้

x_i หมายถึง ค่าของโหนดที่ i

w_{ij} หมายถึง ค่าน้ำหนักจากโหนดที่ i ไปยังโหนดที่ j

b_j หมายถึง ค่าไบแอส (Bias) ของโหนดที่ j ใช้เพื่อทำให้สมการเส้นตรงสามารถสร้างโดยไม่ผ่านจุดกำเนิดได้

f หมายถึง ฟังก์ชันกระตุ้นเพื่อทำให้สมการไม่เป็นเส้นตรงซึ่งสามารถนำไปใช้แก้ปัญหาที่ไม่สามารถใช้สมการเส้นตรงได้

E หมายถึง เซ็ตของเส้นเชื่อมทั้งหมดในโครงข่ายประสาทเทียม

การเรียนรู้ของตัวแบบโครงข่ายประสาทเทียมจะทำได้โดยการเปรียบเทียบค่าของโหนดในชั้นสุดท้ายซึ่งเป็นชั้นที่เป็นผลลัพธ์ในการตัดสินใจของตัวแบบกับค่าผลลัพธ์ที่ต้องการ แล้วคิดค่าเฉลี่ยความผิดพลาดกำลังสอง หลังจากนั้นจะทำการปรับค่าน้ำหนักและไบแอสทั้งหมดด้วยการแทนค่าความผิดพลาดนี้ลงไปในอนุพันธ์ของโหนดในชั้นสุดท้ายเทียบกับข้อมูลที่ป้อนเข้า ทำให้สามารถทำนายค่าได้ใกล้เคียงกับผลลัพธ์ที่ต้องการมากขึ้น

งานวิจัยของ Berner และคณะฯ [3] ใช้การเรียนรู้แบบเสริมกำลังโดยการสร้างโครงข่ายประสาทเทียมสำหรับการเล่นเกม Dota 2 แม้ว่าแนวการเล่นของเกมจะต่างจากโปเกมอนก็ตามแต่ก็สามารถนำวิธีการทำข้อมูลนำเข้าสำหรับเวกเตอร์สถานะและการทำตารางค่าตอบแทนได้ โดยเริ่มจากการเลือกข้อมูลนำเข้าสำหรับป้อนเข้าโครงข่ายประสาทเทียมจะแบ่งข้อมูลเป็น 3 ประเภทคือ

1. ข้อมูลตัวเลข จะแปลงเป็นเป็นคะแนนมาตรฐานก่อนที่จะป้อนเข้าโครงข่ายประสาทเทียมโดยใช้วิธีเก็บข้อมูลค่าเฉลี่ยและส่วนเบี่ยนเบนมาตรฐานของข้อมูลทั้งหมดไว้ แล้วนำค่าที่สังเกตได้มาลบด้วยค่าเฉลี่ยแล้วนำมาหารด้วยส่วนเบี่ยงเบนมาตรฐานเพื่อต้องการให้มีค่าอยู่ในช่วง (-5, 5)
2. ข้อมูลที่จัดเป็นหมวดหมู่ จะนำไป Embed ก่อน
3. ข้อมูลที่เป็นค่าความจริง จะแปลงเป็น 0 และ 1 ก่อนจากนั้นจะแปลงค่าเป็นคะแนนมาตรฐานเช่นเดียวกับข้อมูลตัวเลข

นอกจากนี้ยังมีการระบุว่ามีการเก็บข้อมูลที่เป็นตัวละครที่ฝ่ายเดียวกันมองเห็นได้ 189 ตัว ในกรณีที่มีไม่สามารถมองเห็นถึง 189 ตัวจะแทนค่าเป็น 0 ทั้งหมดสำหรับตัวละครตัวนั้น

การทำตารางค่าตอบแทนนั้น งานวิจัยนี้จะเลือกค่าตอบแทนที่รวมกันสองฝ่ายมีค่าเป็น 0 เนื่องจากมีเพียงฝ่ายเดียวที่จะชนะได้ ดังนั้นทุกอย่างที่ทีมหนึ่งได้ประโยชน์จะต้องถือว่าอีกทีมหนึ่งเสียผลประโยชน์

สิ่งที่ต้องการนำมาปรับใช้คือการจัดการกับข้อมูลโดยใช้วิธีแบบเดียวกัน แต่สำหรับโครงงานนี้จะมีการปรับปรุงในเรื่องของข้อมูลที่เป็นค่าความจริงซึ่งจะไม่แปลงค่าเป็นคะแนนมาตรฐานเนื่องจากคาดว่าจะไปเพิ่มระยะเวลาคำนวณมากขึ้น นอกจากนี้ก็จะใช้ระบบที่ค่าตอบแทนที่ทั้งสองฝ่ายรวมกันได้ 0 เช่นกันเนื่องจากเป็นเกมที่มีผู้ชนะเพียงฝ่ายเดียว

2.2 ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับปัญญาประดิษฐ์ในเกมโปเกมอน

เว็บไซต์ Bulbepedia [8] เป็นสารานุกรมสำหรับเกมโปเกมอน ซึ่งมีการระบุสูตรการคำนวณค่าพลังของโปเกมอนไว้ดังนี้

$$HP = \left\lfloor \frac{\left(2 \times\ Base + IV + \left\lfloor \frac{EV}{4} \right\rfloor\right) \times Level}{100} \right\rfloor + Level + 10 \quad (2)$$

$$OtherStat = \left\lfloor \left(\left\lfloor \frac{\left(2 \times\ Base + IV + \left\lfloor \frac{EV}{4} \right\rfloor\right) \times Level}{100} \right\rfloor + 5 \right) \times Nature \right\rfloor \quad (3)$$

โดยตัวแปรต่าง ๆ มีความหมายดังนี้

HP หมายถึง ค่าพลังชีวิตของโปเกมอน

$OtherStat$ หมายถึง ค่าพลังอื่น ๆ ของโปเกมอน

$Base$ หมายถึง ค่าพลังพื้นฐานสำหรับโปเกมอนสปีชีส์นั้น ๆ

IV หมายถึง ค่าเฉพาะตัวของโปเกมอนตัวนั้น

EV หมายถึง ค่าการฝึกฝนโปเกมอน

$Level$ หมายถึง ค่าเลเวลของโปเกมอน มีค่าระหว่าง 1 ถึง 100

$Nature$ หมายถึง ค่าที่เปลี่ยนแปลงจากนิสัยของโปเกมอน

สูตรคำนวณนี้สามารถใช้เพื่อทำนายค่าพลังโดยประมาณของฝ่ายตรงข้ามได้เนื่องจากสามารถทราบข้อมูลค่าพลังพื้นฐานและเลเวลของโปเกมอนฝ่ายตรงข้ามขณะต่อสู้ได้และจากซอร์สโค้ด (Source code) ของโปเกมอนโชว์ดาวน์ระบุว่าการสุ่มทีมมักจะใช้ค่า IV เป็น 31, EV เป็น 85 และค่า Nature เป็น 1 ทั้งหมด นอกจากนี้ Bulbapedia ยังมีการระบุสูตรคำนวณความเสียหายของการโจมตีไว้ดังนี้

$$Damage = \left(\frac{\left(\frac{2 \times Level}{5} + 2 \right) \times Power \times A/D}{50} + 2 \right) \times Modifier$$
$$Modifier = Targets \times Weather \times Badge \times Critical \times random \times STAB \times Type \times Burn \times other \quad (4)$$

โดยตัวแปรต่าง ๆ มีความหมายดังนี้

$Damage$ หมายถึง ความเสียหายต่อพลังชีวิตที่เกิดขึ้นในการโจมตี

$Level$ หมายถึง ค่าเลเวลของโปเกมอน

$Power$ หมายถึง ค่าพลังของท่าโจมตี

A หมายถึง ค่าพลังโจมตีของโปเกมอนของฝ่ายที่โจมตี

D หมายถึง ค่าพลังป้องกันของโปเกมอนของฝ่ายที่รับความเสียหาย

$Modifier$ หมายถึง ค่าที่เปลี่ยนแปลงค่าความเสียหายซึ่งเกิดจากหลายปัจจัยรวมกัน

$Targets$ หมายถึง ค่าที่เปลี่ยนแปลงตามจำนวนเป้าหมาย ซึ่งในโครงงานนี้จะใช้โปเกมอนทีละ 1 ตัว ค่านี้จึงเป็น 1 ตลอด

$Weather$ หมายถึง ค่าความเหมาะสมกับสภาพอากาศ

$Badge$ หมายถึง ค่าพลังเสริมเมื่อมีเหรียญตราของประเภทการโจมตีซึ่งในโครงงานนี้จะเป็น 1 ตลอด เพราะรูปแบบเกมไม่ใช่ [Gen 2]

$Critical$ หมายถึง ค่าความเสียหายเพิ่มเติมเมื่อสามารถโจมตีติดคริติคอล

$random$ หมายถึง ค่าที่เกิดจากการสุ่ม มีค่าระหว่าง 0.85 ถึง 1

$STAB$ หมายถึง ค่าพลังเสริมเมื่อโปเกมอนใช้ท่าโจมตีที่มีประเภทตรงกับประเภทของตัวเองจะมีค่าเป็น 1.5 แต่ถ้าไม่ก็จะมีค่าเป็น 1

$Type$ หมายถึง ค่าที่นำมาคูณตามการโจมตีตามประเภทของท่าโจมตีและประเภทของโปเกมอนที่ถูกโจมตี

$Burn$ หมายถึง หากผู้โจมตีติดสถานะเผาไหม้และใช้ท่าโจมตีกายภาพจะมีค่าเป็น 0.5 แต่ใน

กรณีอื่นจะมีค่าเป็น 1

$other$ หมายถึง ค่าอื่น ๆ ซึ่งขึ้นกับความสามารถของโปเกมอนแต่ละตัว ท่าโจมตี และสิ่งของถือของโปเกมอน

สำหรับสูตรคำนวณนี้จะใช้เพื่อกำหนดข้อมูลที่ใช้ในเวกเตอร์สถานะว่าควรมีข้อมูลอะไรบ้าง นอกจากนั้นจะสังเกตว่าสูตรคำนวณดังกล่าวจะเน้นผลคูณของตัวแปรเป็นส่วนมาก ทำให้นำไปปรับใช้โดยตั้งสมมติฐานว่าการใช้ลอการิทึมน่าจะช่วยในการคำนวณโดยเปลี่ยนเป็นผลบวกของตัวแปรซึ่งคาดว่าน่าจะเหมาะสมกับตัวแบบที่เป็นโครงข่ายประสาทเทียม

$$\log(\prod_{x\in X} x) = \sum_{x\in X} (\log x) \tag{5}$$

งานวิจัยจาก Kalose และคณะฯ [5] ใช้วิธีการเรียนรู้แบบเสริมกำลังโดยการใช้การเรียนรู้แบบคิว (Q-Learning) ซึ่งมีการสร้างตารางค้นหาเพื่อคำนวณหาคุณภาพของการกระทำของสถานะนั้น ๆ และนำไปเปรียบเทียบกับการกระทำอื่น ๆ เพื่อเลือกการกระทำที่ดีที่สุด ในงานวิจัยดังกล่าวระบุขอบเขตไว้ว่าใช้โปเกมอนเพียง 151 ตัวแรกและท่าโจมตี 165 ท่าโจมตีแรกซึ่งได้ผลของการวิจัยว่าปัญญาประดิษฐ์มีอัตราการชนะการเล่นแบบสุ่ม 60% หลังจากการฝึกฝนกับตัวเอง 5,000 เกมเมื่อใช้การเลือกการกระทำด้วย Epsilon-greedy และอัตราการชนะการเล่นแบบสุ่ม 65% เมื่อใช้การเลือกการกระทำด้วย Softmax นอกจากนั้นยังมีการทดลองว่าหากฝึกฝนกับตัวเองมากขึ้นเป็น 20,000 เกมจะมีอัตราชนะเพิ่มขึ้นเป็น 70% และมีการทดลองใช้อัลกอริทึม Minimax ซึ่งได้อัตราชนะมากกว่าการใช้การเรียนรู้แบบคิวคือมีอัตราการชนะการเล่นแบบสุ่ม 90% จากงานวิจัยดังกล่าวมีการระบุสูตรการปรับปรุงค่าของ Q ซึ่งหมายถึงคุณภาพของการกระทำไว้ดังนี้

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\left(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right) \tag{6}$$

ซึ่งจะนำงานวิจัยมาประยุกต์ใช้กับโครงงานในเรื่องของการเลือกการกระทำด้วย Softmax และนำสูตรดังกล่าวมาปรับใช้โดยวิเคราะห์ว่าตัวแปร Q จะหยุดปรับค่าเมื่อสมการทั้งสองข้างมีค่าเท่ากัน ซึ่งสมการทั้งสองข้างจะมีค่าเท่ากันเมื่อ

$$Q(s_t, a_t) = r_t + \gamma \max_a Q(s_{t+1}, a) \tag{7}$$

สาเหตุที่ต้องแปลงวิธีการคำนวณค่าเป็นสมการดังกล่าวเพราะว่าการใช้ตัวแบบที่เป็นโครงข่ายประสาทเทียมจะต้องคำนึงถึงการใช้อัตราการเรียนรู้เพื่อปรับค่าพารามิเตอร์ต่าง ๆ อยู่แล้วจึงต้องตัดตัวแปรดังกล่าวออกเพื่อไม่ให้เกิดการซ้ำซ้อนในการคำนวณค่า

งานวิจัยของ Simões และคณะฯ [6] แสดงให้แห็นว่าปัญญาประดิษฐ์ที่ใช้วิธีการเรียนรู้แบบเสริมกำลังโดยการใช้การเรียนรู้แบบคิวนั้นมีศักยภาพที่พัฒนาตัวเองไปเหนือกว่าอัลกอริทึมที่ตายตัว และยังสามารถเอาชนะวิธีการเล่นแบบสุ่มได้ทั้งหมดแม้ว่าจะเริ่มต้นด้วยสถานการณ์ที่เสียเปรียบก็ตาม แต่เนื่องจากงานวิจัยนี้ระบุขอบเขตว่าผู้เล่นสามารถใช้โปเกมอนได้เพียง 2 ตัวเท่านั้น ซึ่งในการเล่นทั่วไปผู้เล่นจะใช้โปเกมอน 6 ตัว จึงตั้งสมมติฐานไว้ว่าหากใช้โปเกมอน 6 ตัวจะได้ผลลัพธ์ไปในทิศทางเดียวกัน อย่างไรก็ตามงานวิจัยนี้มีขนาดค่อนข้างใหญ่เพราะใช้การฝึกฝน 2,000,000 เกม สิ่งที่น่าจะนำมาปรับใช้จากงานวิจัยคือวิธีการคำนวณค่าตอบแทน โดยงานวิจัยนี้ระบุค่าตอบแทนไว้ดังนี้

$$R = R_d + R_f - R_t \quad (8)$$

โดยตัวแปรต่าง ๆ มีความหมายดังนี้

R หมายถึง ผลตอบแทนรวม

R_d หมายถึง ร้อยละความเสียหายต่อโปเกมอนของฝ่ายตรงข้ามเทียบกับพลังชีวิตสูงสุดของโปเกมอนตัวนั้น

R_f หมายถึง ค่าเพิ่มเติมหากสามารถทำให้โปเกมอนฝ่ายตรงข้ามไม่สามารถสู้ต่อได้จะมีค่าเป็น
1 ในกรณีอื่น ๆ ให้มีค่าเป็น 0

R_t หมายถึง ร้อยละความเสียหายที่โปเกมอนของฝ่ายผู้เล่นได้รับเทียบกับพลังชีวิตสูงสุดของโปเกมอนฝ่ายผู้เล่น

สำหรับโครงงานนี้จะนำวิธีคำนวณค่าตอบแทนมาใช้บางส่วนเนื่องจากต้องการให้ค่าตอบแทนสมมาตรสำหรับทั้งสองฝ่าย

งานวิจัยของ Lee และคณะฯ [9] มีการใช้การเรียนรู้แบบคิวที่ใช้โครงข่ายประสาทเทียมเช่นกันแต่ไม่มีรายละเอียดมากนัก งานวิจัยนี้จะเน้นไปที่การแข่งขันระหว่างปัญญาประดิษฐ์ 2 ตัวที่ใช้อัลกอริทึมต่างกันโดยจัดการแข่งขัน 15 รอบ รอบละ 6 เกม และในแต่ละรอบจะเป็นการสุ่มทีมมา 2 ทีม สลับกันเล่นรอบละ 3 เกมต่อ 1 ทีม โดยข้อมูลที่น่าสนใจคือมีการระบุว่าตัวแบบที่เป็นเพอร์เซปตรอนชั้นเดียวชนะการเล่นแบบสุ่ม 6 รอบจาก 15 รอบคิดเป็น 40% และ 53 เกมจาก 90 เกมคิดเป็น 58.9% และตัวแบบที่เป็นเพอร์เซปตรอนหลายชั้นชนะการเล่นแบบสุ่ม 10 รอบจาก 15 รอบคิดเป็น 66.7% และ 57 เกมจาก 90 เกมคิดเป็น 63.3% จากข้อมูลนี้สามารถสรุปได้ว่าการใช้ตัวแบบที่เป็นเพอร์เซปตรอนหลายชั้นมีประสิทธิภาพมากกว่าเพอร์เซปตรอนชั้นเดียวเนื่องจากจำนวนรอบที่ชนะมากกว่าแม้ว่าจำนวนเกมที่ชนะใกล้เคียงกัน ซึ่งอาจตีความได้ว่าการชนะเกิดขึ้นในรอบเดียวกันมากกว่าการชนะหลาย ๆ เกมซึ่งเกิดในรอบที่ต่างกันเป็นการแสดงให้เห็นว่าสามารถเอาชนะได้แม้ว่าจะได้เล่นทีมที่เสียเปรียบ สิ่งที่น่าสนใจคืออีกอัลกอริทึมที่ระบุในตัวงานวิจัยชื่อว่า OTL มีวิธีเลือกการกระทำคือดูว่าท่าโจมตีที่สร้างความเสียหายต่อโปเกมอนฝ่ายคู่ต่อสู้ขณะปัจจุบันมากที่สุดอยู่กับโปเก

มอนฝ่ายผู้เล่นตัวไหน ถ้าเป็นโปเกมอนบนสนามก็ให้เลือกท่าโจมตีนั้นได้เลย แต่ถ้าไม่ก็ให้เลือกคำสั่งเปลี่ยนตัวโปเกมอนที่มีท่าโจมตีนั้นออกมา ผลลัพธ์ของอัลกอริทึมดังกล่าวถือว่ามีประสิทธิภาพสูงมาก เพราะมีอัตราการชนะสูงจากการแข่งขันกับอัลกอริทึมอื่น ๆ หลายตัว ซึ่งคิดว่าอาจจะใช้ในการทดสอบด้วยได้หากต้องการนำมาพัฒนาหรือต่อยอด สิ่งที่นำมาปรับใช้ในโครงงานนี้คือการเลือกใช้ตัวแบบโครงข่ายประสาทเทียมที่เป็นเพอร์เซปตรอนหลายชั้นและการตั้งขอบเขตที่ว่าไม่อนุญาตให้ผู้เล่นใช้โปเกมอนที่ชื่อว่า Zoroark ที่มีความสามารถ Illusion ซึ่งเป็นความสามารถที่ฝ่ายคู่ต่อสู้จะมองเห็นเป็นโปเกมอนอีกตัวหนึ่งจนกว่าจะได้รับความเสียหายจากการโจมตี เนื่องจากมีโอกาสที่ปัญญาประดิษฐ์จะไม่สามารถรับรู้ได้ถูกต้องและอาจจะติดปัญหาเช่นการใช้ท่าโจมตีประเภทพลังจิตจะไม่สามารถสร้างความเสียหายใส่ Zoroark ได้เนื่องจากเป็นโปเกมอนประเภทมืด เมื่อไม่ได้รับความเสียหายจากการโจมตีก็จะไม่สูญเสียความสามารถก็จะทำให้เกิดปัญหาเดิมซ้ำ ๆ

2.3 การประยุกต์ใช้และทฤษฎีที่ใช้เพิ่มเติม

จากงานวิจัยดังกล่าว [5][6] จะสังเกตได้ว่าเลือกใช้วิธีการเรียนรู้แบบเสริมกำลัง และการวัดผลโดยการแข่งขันกับการเล่นแบบสุ่ม โครงงานนี้จึงต้องการที่จะทดลองทำปัญญาประดิษฐ์ที่ใช้ในสถานการณ์การเล่นจริงซึ่งจะต่อยอดจากงานวิจัยก่อนหน้าโดยการใช้วิธีการเรียนรู้แบบเสริมกำลังและวัดผลโดยการแข่งขันกับการเล่นแบบสุ่ม

โดยปกติแล้วฟังก์ชันหลัก ๆ ที่ใช้ในการแปลงข้อมูลสำหรับงานนี้คือฟังก์ชันลอการิทึมตามที่กล่าวไว้ในหัวข้อที่ 2.2 แต่ฟังก์ชันลอการิทึมนั้นจะใช้ได้เฉพาะจำนวนจริงบวกเท่านั้น และข้อมูลบางอย่างอาจมีค่าติดลบได้จึงกำหนดฟังก์ชัน rlog เพิ่มเติมโดยจะทำให้ใช้กับจำนวนจริงได้โดยที่คิดว่าไม่เสียคุณสมบัติด้านการแปลงผลคูณเป็นผลบวกมากนัก

$$\mathrm{rlog}(x) = \begin{cases} \log(x+1) & ; \quad x \geq 0 \\ -\log(-x+1) & ; \quad x < 0 \end{cases} \tag{9}$$

การเก็บข้อมูลสถิติทั้งหมดอาจเปลืองพื้นที่ในการจัดเก็บและใช้เวลาอ่านเขียนไฟล์เพิ่มขึ้น นอกจากนี้ยังต้องมีการคำนวณค่าสถิติใหม่ทุกรอบ ทำให้ความซับซ้อนเชิงเวลาเป็น O(n) จึงเปลี่ยนเป็นการเก็บเพียงค่าสถิติซึ่งประกอบด้วย จำนวนค่า ค่าเฉลี่ย และค่าความแปรปรวน เพื่อลดพื้นที่ที่ใช้ในการจัดเก็บข้อมูลและลดความซับซ้อนเชิงเวลาเป็น O(1) โดยใช้สูตรคำนวณดังนี้

$$\begin{aligned} n_{new} &= n_{old} + |X_{new}| \\ \mu_{new} &= \frac{\mu_{old} n_{old} + \sum_{x \in X_{new}} x}{n_{new}} \\ (\sigma^2)_{new} &= \frac{n((\sigma^2)_{old} + (\mu_{old})^2) + \sum_{x \in X_{new}} x^2}{n_{new}} - (\mu_{new})^2 \end{aligned} \tag{10}$$

การเลือกการกระทำจะใช้ฟังก์ชัน Softmax แต่ดัดแปลงเพื่อให้ความน่าจะเป็นที่เลือกการกระทำที่เล่นไม่ได้เป็น 0 และป้องกันไม่ให้ปัญญาประดิษฐ์สลับตัวไปมาโดยไม่โจมตีกันเลยระหว่างการฝึกฝนโดยการเพิ่มความน่าจะเป็นที่จะเลือกใช้ท่าโจมตีตามจำนวนเทิร์นที่ผ่านไป

$$q_t(a_i) = \begin{cases} \min(10, Q(s_t, a_i) + \varepsilon t^d) & ; \quad a_i \in A_{aval} \wedge a_i \in A_{move} \\ Q(s_t, a_i) & ; \quad a_i \in A_{aval} \wedge a_i \notin A_{move} \\ -\infty & ; \quad a_i \notin A_{aval} \end{cases} \tag{11}$$

$$P_t(a_i) = \frac{e^{q_t(a_i)}}{\sum_{a \in A_{aval}} e^{q_t(a)}}$$

โดยตัวแปรต่าง ๆ มีความหมายดังนี้

$Q(s_t, a_i)$ หมายถึงค่า Q ของการกระทำ a_i เมื่ออยู่ในสถานะ s_t

t หมายถึงหมายเลขเทิร์น

A_{aval} หมายถึงเซ็ตของการกระทำที่ใช้งานได้

A_{move} หมายถึงเซ็ตของการกระทำที่เป็นท่าโจมตี

$P_t(a_i)$ หมายถึงความน่าจะเป็นที่จะใช้การกระทำ a_i

ε และ d เป็นไฮเพอร์พารามิเตอร์ที่เลือกกำหนดเอาเอง

วิธีการฝึกฝนตัวแบบจะทำการสั่งให้ตัวแบบทำนายค่า Q ของสถานะนั้น ๆ และสถานะถัดไป แต่เนื่องจากพบว่าปัญญาประดิษฐ์มีปัญหาเรื่องการที่ว่าในช่วงเริ่มต้นของการฝึกฝนนั้นค่า Q ของสถานะถัดไปอาจจะถูกทำนายไว้สูงเกินไปทำให้เมื่อคำนวณค่าออกมาแล้วค่า Q จะสูงขึ้นเรื่อย ๆ ถึงแม้ว่าจะบีบค่าในตัวโครงข่ายประสาทเทียมแล้วก็ตามก็จะได้ค่าเข้าใกล้ 10 เกือบทุกการกระทำ จึงต้องมีการบีบค่าประมาณค่าตอบแทนในอนาคตให้อยู่ในช่วง -5 ถึง 5

$$\begin{gathered} Q'(s_t) = \min\left(5, \max\left(-5, \max_{a \in A_{aval}} Q(s_{t+1}, a)\right)\right) \\ Q(s_t, a_i) = \begin{cases} Q(s_t, a_i) & ; \quad a_i \neq a_t \\ r + \gamma Q'(s_t) & ; \quad a_i = a_t \end{cases} \end{gathered} \tag{12}$$

บทที่ 3
การออกแบบระบบ

ในบทนี้จะกล่าวถึงวิธีการพัฒนาปัญญาประดิษฐ์ในเกมโปเกมอนซึ่งจะประกอบไปด้วยการเชื่อมต่อเข้ากับเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์ การออกแบบตัวแบบสำหรับปัญญาประดิษฐ์ วิธีการฝึกฝนและการวัดผลปัญญาประดิษฐ์

3.1 ขั้นตอนวิธีที่ใช้

จากการศึกษางานวิจัยต่าง ๆ ทำให้โครงงานนี้เลือกใช้ตัวแบบเป็นโครงข่ายประสาทเทียมมีวิธีการเรียนรู้แบบเสริมกำลังโดยใช้การเรียนรู้แบบคิว ใช้การเลือกการกระทำโดยใช้ฟังก์ชัน Softmax ซึ่งมีการปรับปรุงเพิ่มเติมเพื่อคำนวณความน่าจะเป็นของการกระทำออกมาแล้วจึงสุ่มด้วยความน่าจะเป็นนั้น เลือกใช้วิธีการฝึกฝนโดยการเล่นกับตัวเอง และเลือกใช้การวัดผลโดยการเล่นแข่งขันกับการเล่นแบบสุ่ม

3.2 การเชื่อมต่อเข้ากับเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์

วิธีการเชื่อมต่อกับเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์จะใช้การโคลนเซิร์ฟเวอร์ดังกล่าวมาเปิดที่คอมพิวเตอร์ จากนั้นใช้เว็บซ็อกเก็ตเพื่อเชื่อมต่อโปรแกรมเข้ากับเซิร์ฟเวอร์ที่เปิดไว้และทำการล็อกอินผ่านเว็บไซต์ของโปเกมอนโชว์ดาวน์จะทำให้เซิร์ฟเวอร์สามารถส่งข้อความมาที่โปรแกรม แล้วนำมาประมวลผลว่าข้อความต่าง ๆ ที่ส่งมามีความหมายว่าอย่างไรโดยเอกสารในซอร์สโค้ดของโปเกมอนโชว์ดาวน์ [2] จะระบุโปรโตคอลเอาไว้ ซึ่งจะแยกจัดการโดยเน้นข้อความที่เกี่ยวกับการต่อสู้เป็นหลักเพราะว่าข้อความเกี่ยวกับการต่อสู้นั้นค่อนข้างมีความสำคัญต่อการทำปัญญาประดิษฐ์และมีความซับซ้อนสูง เนื่องจากว่าโปเกมอนโชว์ดาวน์ไม่ได้ส่งสถานะของเกมมาตรง ๆ เหมือนที่ปรากฏบนส่วนประสานกับผู้ใช้แต่จะคอยระบุการเหตุการณ์ที่เกิดขึ้นในการต่อสู้แทน ความซับซ้อนอีกอย่างก็คือโปรแกรมไม่สามารถรับรู้ข้อมูลของโปเกมอนได้นอกจากชื่อและไฟล์ที่เป็นฐานข้อมูลของโปเกมอนโชว์ดาวน์ใช้เป็นไฟล์โปรแกรมภาษาจาวาสคริปต์ แต่โครงงานนี้พัฒนาโปรแกรมในภาษาไพธอนจึงต้องใช้ฐานข้อมูลจากแหล่งอื่นแทน

ฐานข้อมูลที่ใช้จะเลือกฐานข้อมูลของ veekun [10] เนื่องจากเป็นไฟล์ csv สามารถนำมาใช้ได้ง่ายและใช้ได้กับภาษาไพธอน นอกจากนี้ฐานข้อมูลนี้ยังผ่านกระบวนการลดความซ้ำซ้อนแล้ว แต่ก็จำเป็นต้องพัฒนาโปรแกรมสำหรับการอ่านข้อมูลและเปรียบเทียบข้อมูลเนื่องจากข้อมูลในฐานข้อมูลบางอย่างมีค่าต่างกับค่าที่ส่งมาจากเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์

3.3 การสร้างตัวแบบโครงข่ายประสาทเทียมสำหรับการเรียนรู้แบบคิว

ไลบรารีสำเร็จรูปที่ใช้เกี่ยวกับการจัดการข้อมูลจะใช้ NumPy [11] และการสร้างโครงข่ายประสาทเทียมจะใช้ TensorFlow [12] และ Keras [13] การใช้ฟังก์ชันลอการิทึมจะใช้ NumPy เป็นหลักเพื่อที่จะแปลงให้ลอการิทึมของ 0 มีค่าเป็นติดลบอนันต์แทนที่จะเกิดข้อผิดพลาด โครงข่ายประสาทเทียมจะใช้อัตราการเรียนรู้ 0.001 ใช้ออพติไมเซอร์เป็น Adam ใช้การตั้งค่าน้ำหนักตอนเริ่มด้วย GlorotUniform

ตัวแบบโครงข่ายประสาทเทียมจะเริ่มจากชั้นที่เป็นการรับข้อมูลนำเข้าจะมีฟังก์ชันการสร้างชั้นที่เป็นท่าโจมตีและโปเกมอนแยกออกจากฟังก์ชันหลักสำหรับสถานะเกมเพื่อทำให้สามารถสร้างฟังก์ชันหลักได้สะดวกยิ่งขึ้น เนื่องจากว่าสถานะเกมจะประกอบไปด้วยโปเกมอน 12 ตัวจากทั้ง 2 ฝ่าย ฝ่ายละ 6 ตัวและโปเกมอน 1 ตัวจะมีท่าโจมตีทั้งหมด 4 ท่าโจมตี ดังนั้นข้อมูลนำเข้าจะแยกเป็น 3 ส่วนคือส่วนของท่าโจมตี 1 ท่าโจมตี ส่วนของโปเกมอน 1 ตัว และส่วนของสถานะเกม นอกจากนี้ข้อมูลที่ใช้ป้อนเข้าจะมีความแตกต่างกันขึ้นอยู่กับ 2 ปัจจัยคือเป็นโปเกมอนตัวนั้นเป็นของฝ่ายใดและโปเกมอนตัวนั้นเป็นโปเกมอนปัจจุบันบนสนามหรือไม่ ข้อมูลที่จะป้อนเข้าโครงข่ายประสาทเทียมมีดังนี้

ตารางที่ 3.1 ตารางข้อมูลที่จะป้อนเข้าสำหรับท่าโจมตีในโครงข่ายประสาทเทียม

ข้อมูล	ประเภท	วิธีการจัดการ	สำหรับโปเกมอน
ไอดีของท่าโจมตี	หมวดหมู่	Embedding (729 -> 32)	ทั้งหมด
ไอดีของประเภทของท่าโจมตี	หมวดหมู่	One-Hot Encoding	ทั้งหมด
พลังของท่าโจมตี	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	ทั้งหมด
ความแม่นยำของท่าโจมตี	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	ทั้งหมด
ลำดับความสำคัญในการโจมตี	ตัวเลข	นำมาบวก 6 จากนั้นคำนวนคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
จำนวนครั้งที่สามารถใช้ท่าโจมตีได้	ตัวเลข	คำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
ไอดีของหมวดหมู่ของการโจมตี	หมวดหมู่	One-Hot Encoding	ทั้งหมด

ข้อมูล	ประเภท	วิธีการจัดการ	สำหรับโปเกมอน
จำนวนครั้งการโจมตีที่น้อยที่สุด	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
จำนวนครั้งการโจมตีที่มากที่สุด	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
จำนวนเทิร์นที่การโจมตีมีผลน้อยที่สุด	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
จำนวนเทิร์นที่การโจมตีมีผลมากที่สุด	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
อัตราการฟื้นฟูพลังชีวิตจากการโจมตี (เทียบกับความเสียหายฝ่ายตรงข้าม)	ตัวเลข	ผ่านฟังก์ชันกำหนดเอง rlog จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
อัตราการฟื้นฟูพลังชีวิตจากการใช้ท่าโจมตี (เทียบกับพลังชีวิตสูงสุดของผู้ใช้)	ตัวเลข	ผ่านฟังก์ชันกำหนดเอง rlog จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
ท่าโจมตีนี้มีอัตราโจมตีติดคริติคอลมากกว่าปกติ	ค่าความจริง	แปลงเป็น 0 และ 1	โปเกมอนปัจจุบัน
โอกาสที่จะเกิดเอฟเฟคจากการโจมตีขึ้น	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
โอกาสที่เกิดสถานะผิดปกติขึ้นจากการโจมตี	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	ทั้งหมด
โอกาสที่ทำให้อีกฝ่ายโจมตีไม่ได้ในเทิร์นนั้นจากการโจมตี	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
การเปลี่ยนแปลงค่าพลังต่าง ๆ จากการโจมตี	ตัวเลข 7 ตัว	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
โอกาสการเปลี่ยนแปลงค่าพลังจากการใช้ท่าโจมตี	ตัวเลข	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
ข้อมูลระบุเพิ่มเติม	หมวดหมู่ 0 – 21 ค่า	One-Hot Encoding ทุกค่า	โปเกมอนปัจจุบัน
ไอดีระบุเป้าหมายการโจมตี	หมวดหมู่	Embedding (15 -> 2)	โปเกมอนปัจจุบัน
ไอดีของเอฟเฟคที่เกิดจากท่าโจมตี	หมวดหมู่	Embedding (421 -> 16)	โปเกมอนปัจจุบัน

ข้อมูล	ประเภท	วิธีการจัดการ	สำหรับโปเกมอน
ไอดีของหมวดหมู่ของเอฟเฟคที่เกิดจากท่าโจมตี	หมวดหมู่	Embedding (15 -> 8)	โปเกมอนปัจจุบัน
ไอดีของสถานะผิดปกติที่เกิดขึ้นจากท่าโจมตี	หมวดหมู่	Embedding (27 -> 8)	ทั้งหมด

ตารางที่ 3.2 ตารางข้อมูลที่จะป้อนเข้าสำหรับโปเกมอนในโครงข่ายประสาทเทียม

ข้อมูล	ประเภท	วิธีการจัดการ	สำหรับโปเกมอน
ไอดีของโปเกมอน	หมวดหมู่	Embedding (10158 -> 32)	ทั้งหมด
ร้อยละพลังชีวิตที่เหลือของโปเกมอน	ตัวเลข	-	ทั้งหมด
ไอดีของประเภทของโปเกมอน	หมวดหมู่ 1 – 2 ค่า	One-Hot Encoding ทุกค่า	ทั้งหมด
ค่าพลังแต่ละค่าของโปเกมอน	ตัวเลข 8 ตัว	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	ทั้งหมด
อัตราส่วนของค่าพลังที่เปลี่ยนแปลงชั่วคราวของโปเกมอน	ตัวเลข 7 ตัว	ผ่านฟังก์ชันลอการิทึม จากนั้นคำนวณคะแนนมาตรฐาน	โปเกมอนปัจจุบัน
ไอดีของสิ่งของถือของโปเกมอน	หมวดหมู่	Embedding (1006 -> 16)	โปเกมอนปัจจุบัน
ไอดีของความสามารถของโปเกมอน	หมวดหมู่	Embedding (234 -> 32)	โปเกมอนปัจจุบัน
ไอดีของสถานะผิดปกติแบบถาวรของโปเกมอน	หมวดหมู่	Embedding (6 -> 4)	ทั้งหมด
ไอดีของสถานะผิดปกติแบบชั่วคราวของโปเกมอน	หมวดหมู่ 0 – 32 ค่า	Embedding (256 -> 8) ทั้งหมด แล้วนำมาหาผลบวก	โปเกมอนปัจจุบัน
ท่าโจมตี 4 ท่า	ดูตาราง 3.2		ทั้งหมดยกเว้นโปเกมอนปัจจุบันฝ่ายเดียวกัน

ตารางที่ 3.3 ตารางข้อมูลที่จะป้อนเข้าสำหรับสถานะเกมในโครงข่ายประสาทเทียม

ข้อมูล	ประเภท	วิธีการจัดการ
โปเกมอนปัจจุบันของฝ่ายเดียวกัน	ดูตาราง 3.2	
โปเกมอนปัจจุบันของฝ่ายตรงข้าม	ดูตาราง 3.2	
ท่าโจมตี 4 ท่าของโปเกมอนปัจจุบันของฝ่ายเดียวกัน	ดูตาราง 3.1	
โปเกมอนตัวอื่นของฝ่ายเดียวกัน 5 ตัว	ดูตาราง 3.2	
โปเกมอนตัวอื่นของฝ่ายตรงข้าม 5 ตัว	ดูตาราง 3.2	
ไอดีของสภาพพิเศษที่มีผลเฉพาะฝ่ายเดียวกัน	หมวดหมู่ 0 – 16 ค่า	Embedding (128 -> 8) ทั้งหมด แล้วนำมาหาผลบวก
ไอดีของสภาพพิเศษที่มีผลเฉพาะฝ่ายตรงข้าม	หมวดหมู่ 0 – 16 ค่า	Embedding (128 -> 8) ทั้งหมด แล้วนำมาหาผลบวก
ไอดีของสถาพพิเศษที่มีผลทั้งสนาม	หมวดหมู่ 0 – 8 ค่า	Embedding (128 -> 4) ทั้งหมด แล้วนำมาหาผลบวก

จากสมการ (4) จะเห็นว่าข้อมูลส่วนใหญ่จะใช้การคูณกัน การใช้ฟังก์ชันลอการิทึมเพื่อแปลงผลคูณต่าง ๆ ให้เป็นผลบวกตามสมการ (5) จะทำให้โครงข่ายประสาทเทียมทำงานได้ง่ายขึ้นตามที่กล่าวไว้แล้วในหัวข้อที่ 2.2

ในเรื่องของการคำนวณคะแนนมาตรฐานซึ่งใช้การคาดการว่าข้อมูลที่มีทั้งค่าบวกและลบน่าจะเหมาะสมกับฟังก์ชัน ReLU มากกว่าข้อมูลที่มีค่าบวกเพียงอย่างเดียว และน่าจะทำให้การแทนค่าที่ไม่รู้ด้วย 0 ได้ดีขึ้น

อัตราการฟื้นฟูพลังชีวิตจากการใช้ท่าโจมตีทั้ง 2 แบบที่บันทึกในฐานข้อมูลนั้นมีโอกาสเป็นค่าติดลบเนื่องจากท่าโจมตีบางท่าจะเป็นท่าโจมตีที่รุนแรงมากแต่จะมีข้อเสียคือจะเสียพลังชีวิตบางส่วนตามค่าความเสียหายที่สร้างได้เพื่อเป็นสมดุลของเกม ดังนั้นจึงอาจมีค่าติดลบและไม่สามารถผ่านฟังก์ชันลอการิทึมได้ จึงต้องใช้ฟังก์ชัน rlog ที่กำหนดเองตามสมการ (9) ที่ระบุไว้ในหัวข้อที่ 2.3

การคำนวณคะแนนมาตรฐานจะใช้วิธีการเทียบกับค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของข้อมูลแบบเดียวกันทั้งหมดที่เคยเจอรวมถึงค่าใหม่ด้วย นอกจากนั้นจะมีการคัดกรองโดยค่า 0 ที่ผ่านฟังก์ชันลอการิทึมจะได้ติดลบอนันต์ จะทำการแปลงเป็น -5 เพราะคิดว่าค่าติดลบอนันต์น่าจะไม่เหมาะแก่การป้อนเข้าในโครงข่ายประสาทเทียม

ในบางครั้งข้อมูลจะไม่สามารถทราบค่าได้เช่นโดยปกติผู้เล่นจะไม่ทราบท่าโจมตีของโปเกมอนฝ่ายตรงข้ามจนกว่าจะใช้งานซึ่งข้อมูลเหล่านั้นที่เป็นตัวเลขให้ข้ามการจัดการค่าไปเลย นอกจากนี้สภาพพิเศษอาจมีไม่ถึง 8 สภาพพร้อมกัน สิ่งที่ทำคือตั้งเอาไว้ให้มากกว่าหรือเท่ากับจำนวนข้อมูลที่ประมาณไว้ว่าเป็นไปได้ซึ่งหากไม่มีสภาพเกิดขึ้นก็จะจัดการค่าแบบเดียวกันคือให้เป็น 0 ทั้งหมด ข้อมูลที่ระบุเป็นสีน้ำเงินจะแทนที่ข้อมูลที่ไม่ทราบค่าด้วย -5 เนื่องจากสันนิษฐานว่าท่าโจมตีอาจไม่มีความสามารถเพิ่มเติมทำให้โอกาสเป็น 0 เมื่อผ่านฟังก์ชันลอการิทึมจะได้ค่าเป็นติดลบอนันต์จากนั้นเมื่อนำมาคำนวณคะแนนมาตรฐานด้วยวิธีดังกล่าวจะได้ค่าเป็น -5 ยกเว้นข้อมูลบางอย่างที่ใช้การคาดคะเนเช่นเราไม่สามารถมองเห็นค่าพลังฝ่ายคู่ต่อสู้แต่เราสามารถใช้สมการ (2) และ (3) ตามหัวข้อที่ 2.2 เพื่อคำนวณหาค่าพลังโดยประมาณได้

การ Embedding จะใช้ของ Keras ซึ่งเสมือนกับการทำ One-Hot Encoding แล้วสร้าง Dense 1 ชั้น และข้อมูลแบบเดียวกันจะใช้การ Embedding เหมือนกันทั้งหมดซึ่งจะทำเป็นชั้นแรกของโครงข่ายประสาทเทียมก่อนนำมารวมกับข้อมูลอื่น ๆ

สำหรับท่าโจมตีของโปเกมอนที่ไม่ใช่โปเกมอนบนสนามและโปเกมอนฝ่ายตรงข้ามที่ไม่ใช่โปเกมอนบนสนามจะผ่านชั้นการคำนวณที่เป็น Dense ก่อนโดยมีขนาดของผลลัพธ์เป็น 8 และ 16 ตามลำดับจากนั้นนำข้อมูลทั้งหมดมาต่อกันแล้วต่อกับ Dense ขนาด 128 จำนวน 2 ชั้นแล้วนำมาต่อกับ Dense ที่ได้ผลลัพธ์ที่มีขนาด 9 คือแต่ละโหนดระบุค่า Q ของแต่ละการกระทำ โดย Dense ทุกชั้นยกเว้นชั้นสุดท้ายจะใช้ฟังก์ชัน ReLU เพื่อปรับไม่ให้เป็นสมการเส้นตรง ส่วน Dense ชั้นสุดท้ายจะใช้ฟังก์ชันไฮเพอร์โบลิกแทนเจนท์ตามด้วยคูณ 10 เพื่อให้มีค่าติดลบได้และต้องการควบคุมให้ค่าอยู่ระหว่าง -10 ถึง 10

กำหนดค่าตอบแทนไว้ดังตารางที่ 3.4

ตารางที่ 3.4 ตารางค่าตอบแทน

เหตุการณ์	ค่าตอบแทน
ชนะ	+4
แพ้	-4
ทำให้โปเกมอนฝ่ายตรงข้ามไม่สามารถสู้ต่อได้	+1
โปเกมอนฝ่ายเดียวกันไม่สามารถสู้ต่อได้	-1
โปเกมอนฝ่ายตรงข้ามได้รับความเสียหาย 1%	+0.01
โปเกมอนฝ่ายเดียวกันได้รับความเสียหาย 1%	-0.01

เหตุการณ์	ค่าตอบแทน
โปเกมอนฝ่ายเดียวกันฟื้นฟูพลังชีวิต 1%	+0.01
โปเกมอนฝ่ายตรงข้ามฟื้นฟูพลังชีวิต 1%	-0.01

การเลือกการกระทำจะใช้ฟังก์ชัน Softmax ที่ดัดแปลงแล้วตามที่กล่าวไปในหัวข้อที่ 2.3 เลือกการกระทำโดยใช้สมการ (11) และเลือกกำหนดไฮเพอร์พารามิเตอร์ดังนี้

$$\varepsilon = 10^{-5}, d = 3$$

นอกจากนี้หากปัญญาประดิษฐ์สามารถใช้ Mega Evolution จะกำหนดให้ใช้เมื่อเลือกท่าโจมตีเป็นครั้งแรกโดยทันที

3.4 วิธีการฝึกฝนปัญญาประดิษฐ์

การฝึกฝนตัวแบบโครงข่ายประสาทเทียมจะใช้วิธีการตอนที่ถึงเวลาให้เลือกการกระทำ จะให้ปัญญาประดิษฐ์จดจำ 4 สิ่งคือ สถานะของเกมในเทิร์นก่อนหน้า (จะถือว่าเป็นสถานะปัจจุบัน) การกระทำที่เลือกในเทิร์นก่อนหน้า ค่าตอบแทนที่ได้รับ และค่า Q สูงสุดของสถานะของเกมขณะนั้น (จะถือว่าเป็นสถานะถัดไป) โดยจะเก็บข้อมูลนี้ไว้ 1024 เทิร์นล่าสุดในการฝึกฝน เมื่อจบเกมจะสั่งให้ปัญญาประดิษฐ์สุ่มเอาข้อมูลดังกล่าวมา 32 ชุด มาเพื่อฝึกฝน โดยกำหนดค่าป้อนเข้าเป็นสถานะของเกมและค่าผลลัพธ์เป็นเวกเตอร์ของค่า Q การปรับปรุงค่า Q จะใช้สมการ (12) และใช้ไฮเพอร์พารามิเตอร์ดังนี้

$$\gamma = 0.95$$

ซึ่งจะฝึกฝนเพียงฝ่ายเดียวเท่านั้นเพราะคิดว่าถ้าฝึกฝนพร้อม ๆ กันทั้งสองฝ่ายน่าจะมีปัญหาเรื่องการเรียกใช้และแก้ไขไฟล์ในเวลาเดียวกันสำหรับ 2 โปรเซสและแก้ไขได้ยาก จึงใช้วิธีดังกล่าวแล้วให้ฝ่ายที่ไม่ได้ฝึกฝนคัดลอกค่าน้ำหนักของฝ่ายที่ฝึกฝนมาเลย การสุ่มชุดข้อมูลทำไปเพื่อที่จะไม่ให้ข้อมูลที่ป้อนเข้าเรียงติดกันทำให้ตัวแบบมองว่าลำดับของข้อมูลมีผลต่อการฝึกฝนจากคู่มืออธิบายเรื่องโครงข่ายประสาทเทียมที่ใช้ในการเรียนรู้แบบคิว [14]

ทีมที่ใช้ในการฝึกฝนจะเป็นทีมที่สุ่มจากโปเกมอนโชว์ดาวน์โดยใช้รูปแบบเกมของ [Gen 7] Random Battle ซึ่งจะมีข้อแตกต่างใน 2 เรื่องคือผู้เล่นจะส่งโปเกมอนตัวแรกออกมาทันทีแทนที่จะเลือกจาก 6 ตัวและผู้เล่นจะไม่สามารถมองเห็นโปเกมอนที่ฝ่ายตรงข้ามยังไม่เคยใช้ได้ แม้ว่าจะต่างกับที่ใช้ในการวัดผลก็ตามแต่ก็คิดว่าน่าจะพอเรียนรู้ได้ นอกจากนี้จะให้ปัญญาประดิษฐ์ยอมแพ้ทันทีเมื่อมีโปเกมอนที่มีความสามารถ Illusion อยู่ในทีมเพื่อลดปัญหาที่อาจเกิดขึ้น

จำนวนเกมที่ใช้การฝึกฝนจะใช้วิธีการประเมินเวลาที่จะใช้ฝึกฝนในแต่ละวัน แต่ในบางครั้งเกมที่สูสีกันมากอาจใช้เวลานานและใช้เวลาการฝึกฝนในแต่ละวันไม่เท่ากันทำให้ได้จำนวนเกมไม่เท่ากัน จึงทำการบันทึกจำนวนเกมที่ฝึกฝนได้ดังนี้

ตารางที่ 3.5 ตารางจำนวนเกมที่ฝึกฝน

วันที่	จำนวนเกม
15 มีนาคม 2564	250
16 มีนาคม 2564	250
17 มีนาคม 2564	300
18 มีนาคม 2564	200
19 มีนาคม 2564	250
รวมทั้งหมด	**1250**

3.5 วิธีการวัดผลปัญญาประดิษฐ์

การวัดผลปัญญาประดิษฐ์จะใช้ทีมที่ต่างกัน 10 ทีม โดยเล่นกับการเล่นแบบสุ่ม โดยฝ่ายที่เล่นแบบสุ่มก็จะใช้ขอบเขตเดียวกันกับที่ระบุในโครงงาน ทีมที่ใช้จะเลือกทีมของโปเกมอนแชมป์เปียน 8 คนในเกมโปเกมอนเนื่องจากถือว่าเป็นตัวละครที่เก่งที่สุดในเกมโปเกมอนและบางคนยังเลือกใช้ทีมที่มีโปเกมอนหลากหลายประเภทและใช้โปเกมอนครบ 6 ตัวซึ่งถือเป็นตัวละครส่วนน้อยของในเกม นอกจากนี้ในเกมโปเกมอนภาค Black 2 & White 2 ซึ่งอยู่ใน [Gen 5] จะมีระบบ PWT ที่รวบรวมโปเกมอนแชมป์เปียนจากเกมโปเกมอนภาคอื่น ๆ ไว้ด้วยแล้วจัดทีมที่เป็นทีมที่เหมาะกับการต่อสู้ไว้แล้วเช่นโปเกมอนทุกตัวถือสิ่งของที่ช่วยเพิ่มพลังโจมตีหรือลดความเสียหายจากการโจมตีที่เสียเปรียบส่วนโปเกมอนแชมป์เปียนที่ชื่อ Iris ซึ่งเป็นโปเกมอนแชมป์เปียนในเกมโปเกมอนภาค Black 2 & White 2 อยู่แล้วจึงใช้ทีมที่ใช้เมื่อต่อสู้เป็นครั้งที่ 2 ในเกมระดับยาก ซึ่งทีมต่าง ๆ มีการระบุไว้ในเว็บไซต์ Bulbapedia [8] ในบางครั้งความสามารถของโปเกมอนที่เขียนไว้จะไม่ชัดเจนซึ่งจะทำการเลือกความสามารถที่คิดว่าดีที่สุดให้ นอกจากนี้จะไม่มีระบุว่าโปเกมอนแต่ละตัวจะมี EV และ IV เท่าไรจึงใช้วิธีแทนค่า EV ทั้งหมดเป็น 85 และ IV ทั้งหมดเป็น 31 ให้เหมือนตอนสุ่มทีมของโปเกมอนโชว์ดาวน์

แต่เนื่องจากต้องการทดสอบในทีมที่ใช้โปเกมอนจนถึงโปเกมอนที่อยู่ใน [Gen 7] รวมถึงการใช้ Mega Evolution และต้องการทดสอบการใช้งานในการเล่นจริงจึงใช้ทีมของโปเกมอนแชมป์เปียนในโปเกมอนภาค XY และทีมของศาสตราจารย์ในเกมโปเกมอนภาค Sun & Moon ซึ่งเป็นผู้เล่นที่ต้อง

สู้เป็นคนสุดท้ายก่อนฉากจบเกมครั้งแรก โดย 2 ทีมนี้จะใช้การปรับปรุงคือเพิ่มสิ่งของถือที่เหมาะสมไปในโปเกมอนที่ไม่มีสิ่งของถือหรือแทนที่สิ่งของถือที่อนุญาตให้ผู้เล่นใช้ Z-Move ของแต่ละตัว

ทำการวัดผลตามอัตราชนะและจำนวนโปเกมอนที่เหลืออยู่ในแต่ละรอบ และวิเคราะห์ผลเชิงลึกตามวัตถุประสงค์หลัก ๆ ที่เป็นความรู้เฉพาะด้านของเกมโปเกมอนดังนี้

1. การเลือกท่าโจมตีที่สร้างความเสียหายได้ดีที่สุดซึ่งอาจมีความหมายได้ 2 แบบคือการเลือกสร้างความเสียหายต่อโปเกมอนบนสนามของฝ่ายตรงข้ามที่มากที่สุด และการวิเคราะห์ว่าฝ่ายตรงข้ามจะเปลี่ยนตัวจึงเลือกท่าโจมตีที่ได้เปรียบโปเกมอนที่น่าจะเปลี่ยนตัวออกมา
2. การเปลี่ยนตัวเพื่อหลบการโจมตีที่รุนแรงของฝ่ายตรงข้าม โดยวิเคราะห์ว่าฝ่ายตรงข้ามน่าจะใช้ท่าโจมตีที่ได้เปรียบโปเกมอนฝ่ายเราแล้วอาจสร้างความเสียหายมาก จึงสลับโปเกมอนบนสนามที่เสียเปรียบอยู่กับโปเกมอนอีกตัวที่ต้านทานต่อการโจมตีนั้นได้
3. การใช้ท่าโจมตีที่มีผลระยะยาวเช่นท่า Swords Dance จะช่วยเพิ่มพลังโจมตีสามารถทำให้เอาชนะโปเกมอนฝ่ายตรงข้ามได้หลาย ๆ ตัวโดยที่ฝ่ายตรงข้ามไม่มีโอกาสได้โจมตีโต้ตอบ หรือการใช้ท่า Stealth Rock ซึ่งไม่มีผลในขณะที่ใช้แต่จะช่วยสร้างความเสียหายทุกครั้งเมื่อฝ่ายตรงข้ามเปลี่ยนตัวออกมา นอกจากนี้สถานะผิดปกติหลาย ๆ อย่างจะช่วยส่งผลไปในระยะยาวเช่นการทำให้ติดสถานะอัมพาตจะทำให้ไม่สามารถโจมตีในบางครั้ง
4. การเลือกใช้ท่าโจมตีที่มีลำดับความสำคัญมากกว่าจะทำให้ได้โจมตีก่อนฝ่ายตรงข้ามแม้ว่าค่าพลังความเร็วจะน้อยกว่าฝ่ายตรงข้ามก็ตาม แต่มีข้อเสียเรื่องของท่าโจมตีเหล่านี้มักจะสร้างความเสียหายได้น้อย จึงมีวิธีการเล่นที่ว่าให้ใช้เวลาที่พลังชีวิตของฝ่ายตรงข้ามเหลือน้อยเท่านั้น
5. การใช้ท่าโจมตีที่ผสมผสานกันเป็นสิ่งที่น่าสนใจ เช่นการเลือกใช้ท่า Trick-or-Treat ก่อนแล้วตามด้วย Phantom Force หรือ Shadow Sneak จะทำให้สามารถโจมตีได้ชนะทางฝ่ายตรงข้ามแล้วสร้างความเสียหายได้มากขึ้นและท่า Rest ที่จะช่วยฟื้นฟูพลังชีวิตจนเต็มแต่จะทำให้ติดสถานะหลับเป็นเวลา 2 เทิร์นทำให้โจมตีไม่ได้ แต่การใช้ Sleep Talk จะใช้ได้เฉพาะเมื่อติดสถานะหลับเท่านั้น โดยจะสุ่มท่าโจมตีอื่น ๆ แทนที่จะโจมตีไม่ได้

ตารางที่ 3.6 ตารางทีมที่นำมาใช้วัดผล

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
1	Red (PWT)	Pikachu	50	Electric	Static	Light Ball	Volt Tackle	Electric
							Iron Tail	Steel
							Brick Break	Fighting
							Fake Out	Normal
		Venusaur	50	Grass Poison	Overgrow	White Herb	Leaf Storm	Grass
							Sludge Bomb	Poison
							Earthquake	Ground
							Sleep Powder	Grass
		Charizard	50	Fire Flying	Blaze	Focus Sash	Fire Blast	Fire
							Focus Blast	Fighting
							Air Slash	Flying
							Dragon Pulse	Dragon
		Blastoise	50	Water	Torrent	Choice Scarf	Water Spout	Water
							Hydro Pump	Water
							Blizzard	Ice
							Focus Blast	Fighting
		Snorlax	50	Normal	Thick Fat	Quick Claw	Body Slam	Normal
							Earthquake	Ground
							Crunch	Dark
							Seed Bomb	Grass
		Lapras	50	Water Ice	Water Absorb	Sitrus Berry	Ice Beam	Ice
							Hydro Pump	Water
							Ice Shard	Ice
							Thunderbolt	Electric

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
2	Blue (PWT)	Aerodactyl	50	Rock Flying	Pressure	Choice Band	Stone Edge	Rock
							Earthquake	Ground
							Ice Fang	Ice
							Fire Blast	Fire
		Exeggutor	50	Grass Psychic	Chlorophyll	Life Orb	Leaf Storm	Grass
							Wood Hammer	Grass
							Zen Headbutt	Psychic
							Leech Seed	Grass
		Gyarados	50	Water Flying	Intimidate	King's Rock	Waterfall	Water
							Earthquake	Ground
							Ice Fang	Ice
							Outrage	Dragon
		Alakazam	50	Psychic	Synchronize	Focus Sash	Psychic	Psychic
							Focus Blast	Fighting
							Shadow Ball	Ghost
							Reflect	Psychic
		Arcanine	50	Fire	Intimidate	Expert Belt	Flare Blitz	Fire
							Close Combat	Fighting
							Wild Charge	Electric
							Extreme Speed	Normal
		Machamp	50	Fighting	No Guard	White Herb	Superpower	Fighting
							Stone Edge	Rock
							Fire Punch	Fire
							Bullet Punch	Steel

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
3	Lance (PWT-6)	Dragonite	64	Dragon Flying	Inner Focus	Focus Sash	Extreme Speed	Normal
							Ice Punch	Ice
							Fire Punch	Fire
							Draco Meteor	Dragon
		Salamence	64	Dragon Flying	Intimidate	Expert Belt	Dragon Claw	Dragon
							Crunch	Dark
							Earthquake	Ground
							Stone Edge	Rock
		Kingdra	64	Water Dragon	Sniper	Scope Lens	Surf	Water
							Ice Beam	Ice
							Dragon Pulse	Dragon
							Flash Cannon	Steel
		Hydreigon	64	Dark Dragon	Levitate	White Herb	Draco Meteor	Dragon
							Fire Blast	Fire
							Earth Power	Ground
							Dark Pulse	Dark
		Haxorus	64	Dragon	Mold Breaker	Choice Scarf	Outrage	Dragon
							Superpower	Fighting
							Earthquake	Ground
							Rock Slide	Rock
		Flygon	64	Ground Dragon	Levitate	Power Herb	Solar Beam	Grass
							Draco Meteor	Dragon
							Earth Power	Ground
							U-turn	Bug

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
4	Steven (PWT)	Metagross	50	Steel Psychic	Clear Body	Occa Berry	Zen Headbutt	Psychic
							Hammer Arm	Fighting
							Earthquake	Ground
							Bullet Punch	Steel
		Aggron	50	Steel Rock	Rock Head	Air Balloon	Head Smash	Rock
							Avalanche	Ice
							Earthquake	Ground
							Metal Burst	Steel
		Excadrill	50	Ground Steel	Sand Rush	Focus Sash	Earthquake	Ground
							Rock Slide	Rock
							X-Scissor	Bug
							Sandstorm	Rock
		Archeops	50	Rock Flying	Defeatist	Sitrus Berry	Head Smash	Rock
							Acrobatics	Flying
							Earthquake	Ground
							Quick Attack	Normal
		Cradily	50	Rock Grass	Suction Cups	Expert Belt	Stone Edge	Rock
							Seed Bomb	Grass
							Earthquake	Ground
							Sandstorm	Rock
		Armaldo	50	Rock Bug	Battle Armor	White Herb	X-Scissor	Bug
							Rock Blast	Rock
							Earthquake	Ground
							Superpower	Fighting

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
5	Wallace (PWT)	Milotic	50	Water	Marvel Scale	Rocky Helmet	Scald	Water
							Icy Wind	Ice
							Rest	Psychic
							Sleep Talk	Normal
		Sharpedo	50	Water Dark	Rough Skin	Focus Sash	Hydro Pump	Water
							Crunch	Dark
							Zen Headbutt	Psychic
							Aqua Jet	Water
		Walrein	50	Ice Water	Thick Fat	Leftovers	Surf	Water
							Yawn	Normal
							Blizzard	Ice
							Sheer Cold	Ice
		Ludicolo	50	Water Grass	Swift Swim	Life Orb	Surf	Water
							Focus Blast	Fighting
							Giga Drain	Grass
							Rain Dance	Water
		Swampert	50	Water Ground	Torrent	Rindo Berry	Muddy Water	Water
							Earth Power	Ground
							Ice Beam	Ice
							Focus Blast	Fighting
		Starmie	50	Water Psychic	Natural Cure	Expert Belt	Surf	Water
							Psychic	Psychic
							Thunderbolt	Electric
							Signal Beam	Bug

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
6	Cynthia (PWT)	Garchomp	50	Dragon Ground	Sand Veil	Focus Sash	Outrage	Dragon
							Stone Edge	Rock
							Earthquake	Ground
							Swords Dance	Normal
		Spiritomb	50	Ghost Dark	Pressure	Rocky Helmet	Sucker Punch	Dark
							Protect	Normal
							Will-O-Wisp	Fire
							Pain Split	Normal
		Roserade	50	Grass Poison	Natural Cure	White Herb	Leaf Storm	Grass
							Sludge Bomb	Poison
							Shadow Ball	Ghost
							Sleep Powder	Grass
		Togekiss	50	Fairy Flying	Serene Grace	Leftovers	Air Slash	Flying
							Aura Sphere	Fighting
							Shadow Ball	Ghost
							Grass Knot	Grass
		Lucario	50	Fighting Steel	Inner Focus	Life Orb	Close Combat	Fighting
							Dark Pulse	Dark
							Stone Edge	Rock
							Extreme Speed	Normal
		Glaceon	50	Ice	Snow Cloak	Choice Scarf	Ice Beam	Ice
							Shadow Ball	Ghost
							Signal Beam	Bug
							Water Pulse	Water

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
7	Alder (PWT)	Volcarona	59	Bug Fire	Flame Body	Charti Berry	Quiver Dance	Bug
							Heat Wave	Fire
							Bug Buzz	Bug
							Psychic	Psychic
		Conkeldurr	59	Fighting	Guts	Life Orb	Hammer Arm	Fighting
							Mach Punch	Fighting
							Payback	Dark
							Stone Edge	Rock
		Reuniclus	59	Psychic	Magic Guard	Leftovers	Light Screen	Psychic
							Reflect	Psychic
							Toxic	Poison
							Psychic	Psychic
		Krookodile	59	Ground Dark	Moxie	Expert Belt	Earthquake	Ground
							Crunch	Dark
							Stone Edge	Rock
							Outrage	Dragon
		Chandelure	59	Ghost Fire	Flame Body	Choice Scarf	Flamethrower	Fire
							Shadow Ball	Ghost
							Energy Ball	Grass
							Psychic	Psychic
		Braviary	59	Normal Flying	Sheer Force	Choice Band	Brave Bird	Flying
							Superpower	Fighting
							Rock Slide	Rock
							U-turn	Bug

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
8	Iris (B2W2)	Hydreigon	81	Dark Dragon	Levitate	Wise Glasses	Fire Blast	Fire
							Focus Blast	Fighting
							Dragon Pulse	Dragon
							Surf	Water
		Salamence	81	Dragon Flying	Intimidate	Life Orb	Fire Blast	Fire
							Earthquake	Ground
							Draco Meteor	Dragon
							Crunch	Dark
		Archeops	81	Rock Flying	Defatist	None	Acrobatics	Flying
							Stone Edge	Rock
							Dragon Claw	Dragon
							Endeavor	Normal
		Aggron	81	Steel Rock	Rock Head	Muscle Band	Earthquake	Ground
							Double-Edge	Normal
							Head Smash	Rock
							Autotomize	Steel
		Lapras	81	Water Ice	Water Absorb	Zoom Lens	Hydro Pump	Water
							Blizzard	Ice
							Thunder	Electric
							Sing	Normal
		Haxorus	83	Dragon	Mold Breaker	Focus Sash	Earthquake	Ground
							Guillotine	Normal
							Outrage	Dragon
							Dragon Dance	Dragon

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
9	Diantha (XY)	Hawlucha	64	Fighting Flying	Limber	Focus Sash	Swords Dance	Normal
							Flying Press	Fighting
							X-Scissor	Bug
							Poison Jab	Poison
		Tyrantrum	65	Rock Dragon	Strong Jaw	Choice Band	Head Smash	Rock
							Earthquake	Ground
							Dragon Claw	Dragon
							Crunch	Dark
		Aurorus	65	Rock Ice	Refrigerate	Zoom Lens	Thunder	Electric
							Blizzard	Ice
							Light Screen	Psychic
							Reflect	Psychic
		Gourgeist	66	Ghost Grass	Pickup	Life Orb	Trick-or-Treat	Ghost
							Phantom Force	Ghost
							Seed Bomb	Grass
							Shadow Sneak	Ghost
		Goodra	66	Dragon	Sap Sipper	Assault Vest	Dragon Pulse	Dragon
							Muddy Water	Water
							Fire Blast	Fire
							Focus Blast	Fighting
		Gardevoir	68	Psychic Fairy	Trace	Gardevoirite	Moonblast	Fairy
							Psychic	Psychic
							Shadow Ball	Ghost
							Thunderbolt	Electric

ทีมที่	ชื่อทีม	โปเกมอน	LV	ประเภทโปเกมอน	ความสามารถ	สิ่งของถือ	ท่าโจมตี	ประเภทท่าโจมตี
10	Kukui (SM)	Lycanroc (Midday)	57	Rock	Keen Eye	Life Orb	Stone Edge	Rock
							Accelerock	Rock
							Crunch	Dark
							Stealth Rock	Rock
		Ninetales (Alola)	56	Ice Fairy	Snow Cloak	Focus Sash	Dazzling Gleam	Fairy
							Blizzard	Ice
							Ice Shard	Ice
							Safeguard	Normal
		Braviary	56	Normal Flying	Keen Eye	Leftovers	Crush Claw	Normal
							Brave Bird	Flying
							Tailwind	Flying
							Whirlwind	Normal
		Magnezone	56	Electric Steel	Sturdy	Wiki Berry	Thunderbolt	Electric
							Flash Cannon	Steel
							Thunder Wave	Electric
							Mirror Coat	Psychic
		Snorlax	56	Normal	Thick Fat	Assault Vest	Body Slam	Normal
							Crunch	Dark
							Heavy Slam	Steel
							High Horsepower	Ground
		Incineroar	58	Fire Dark	Blaze	Choice Band	Flare Blitz	Fire
							Darkest Lariat	Dark
							Outrage	Dragon
							Cross Chop	Fighting

ตัวอักษรสีม่วงระบุว่าในสมัย [Gen 5] นั้น Togekiss เคยเป็นโปเกมอนประเภท Normal/Flying แต่หากใช้รูปแบบการเล่นที่กำหนดในขอบเขตไว้คือ [Gen 7] ก็จะเป็น Fairy/Flying
ตัวอักษรสีต่าง ๆ ในชื่อของท่าโจมตีระบุหมวดหมู่ของการโจมตีตามนี้

- ตัวอักษรสีส้มหมายถึงท่าโจมตีหมวดหมู่ Physical
- ตัวอักษรสีน้ำเงินหมายถึงท่าโจมตีหมวดหมู่ Special
- ตัวอักษรสีดำหมายถึงท่าโจมตีหมวดหมู่ Status

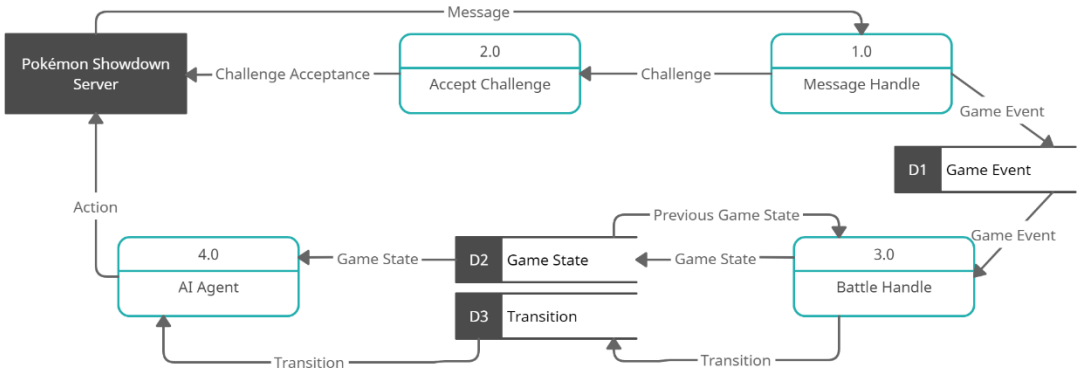
สีแดงหมายถึงความสามารถที่เลือกให้เองและสิ่งที่มีการเปลี่ยนแปลงเอง โดยกรณีของ Iris คือ Flying Gem ซึ่งใช้ไม่ได้ใน [Gen 7] Anything Goes จึงเอาออกแทน เพื่อไม่ให้กระทบต่อท่าโจมตี Acrobatics ที่จะสร้างความเสียหายมากขึ้นเมื่อไม่มีสิ่งของถือ และเลเวลของ Hydreigon ไม่สามารถตั้งไว้เป็น 50 ได้ เนื่องจากผิดกติกาของโปเกมอนโชว์ดาวน์เพราะ Zweilous จะวิวัฒนาการเป็น Hydreigon ในเลเวล 64 จึงต้องปรับทั้งทีมเป็นเลเวล 64 แทน และกรณีของ Alder ก็มีปัญหาแบบเดียวกันจึงปรับเลเวลเพิ่มขึ้นทั้งทีม

การวัดผลจะแบ่งออกเป็น 2 ส่วน คือ การวัดผลโดยใช้ทีมเหมือนกันทั้งสองฝ่ายซึ่งจะมีการเล่นทีมละ 10 เกม และการวัดผลโดยใช้ทีมที่ต่างกันซึ่งจะจัดการแข่งขันทั้งหมด 10 รอบ ในแต่ละรอบจะเลือก 2 ทีมมาจับคู่เพื่อต่อสู้กัน โดยแต่ละทีมจะผลัดกันเล่นฝ่ายละ 5 เกม นอกจากนี้จะกำหนดไว้ว่าแต่ละทีมจะถูกนำมาเล่น 2 รอบ และเนื่องจากแต่ละทีมมีโปเกมอนที่มีเลเวลต่างกันมาก จึงปรับสมดุลโดยการแก้ไขโปเกมอนทุกตัวให้มีเลเวล 100 ทั้งหมดในการวัดผลส่วนที่ 2

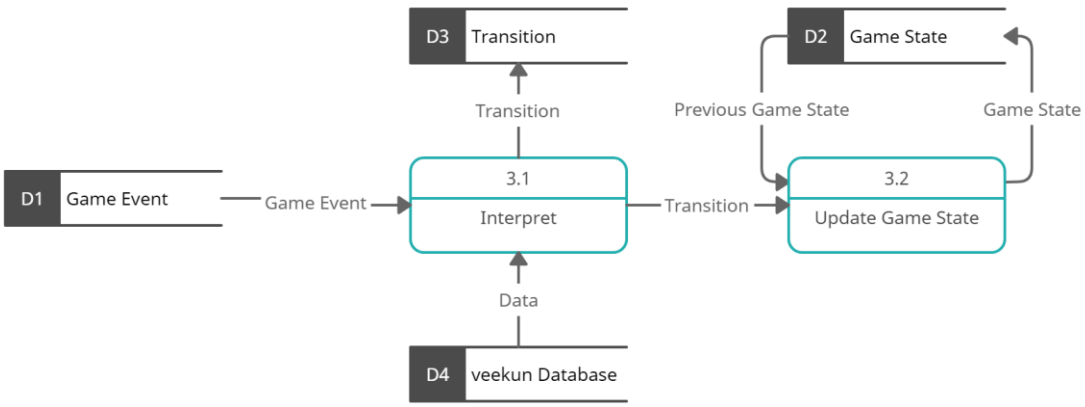
3.6 ภาพรวมระบบของโครงงาน

แผนภาพกระแสข้อมูลต่อไปนี้จะแสดงภาพรวมของโปรแกรมโดยคร่าว ๆ ที่พัฒนาขึ้นในโครงงานโดยสร้างจากเว็บไซต์ creately.com [15]

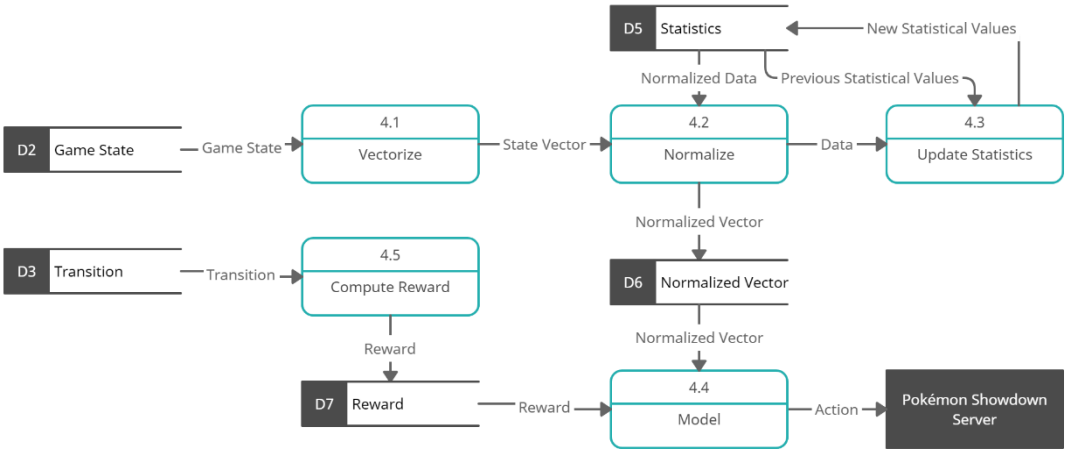
ภาพที่ 3.1 แผนภาพกระแสข้อมูลระดับ *0*



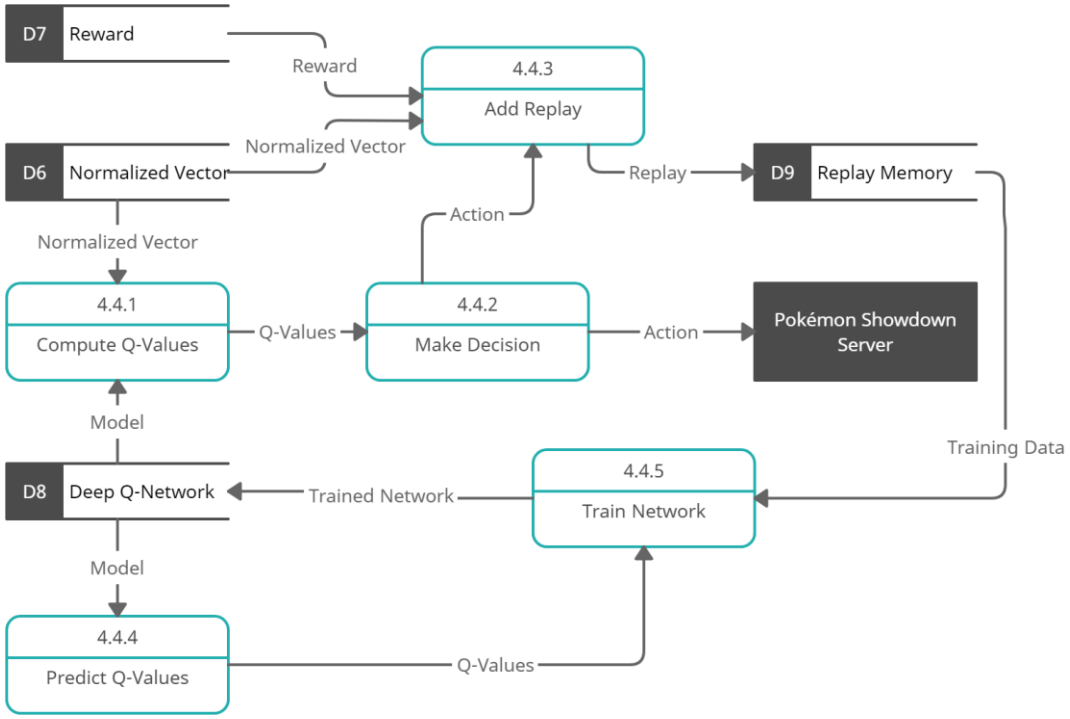
ภาพที่ 3.2 แผนภาพกระแสข้อมูลระดับ 1



ภาพที่ 3.3 แผนภาพกระแสข้อมูลระดับ 2 ในกระบวนการทำงานที่ 3



ภาพที่ 3.4 แผนภาพกระแสข้อมูลระดับ 2 ในกระบวนการทำงานที่ 4



ภาพที่ 3.5 แผนภาพกระแสข้อมูลระดับ 3 ในกระบวนการทำงานที่ 4.4

บทที่ 4
เทคนิคการเขียนโปรแกรมและอภิปรายผลการทดลอง

ในบทนี้จะกล่าวถึง เทคนิคการเขียนโปรแกรมและอภิปรายผลของการแข่งขันระหว่างปัญญาประดิษฐ์ในเกมโปเกมอนที่พัฒนาขึ้นในโครงงานนี้กับการเล่นแบบสุ่ม

4.1 เทคนิคการเขียนโปรแกรม

ดังที่กล่าวไปในบทที่ 2.3 การคำนวณคะแนนมาตรฐานสามารถใช้วิธีดังต่อไปนี้เพื่อช่วยลดระยะเวลาการคำนวณและพื้นที่ในการจัดเก็บได้

```
    def memorize(self, index, array):
        if index not in self.statistics_df.index:
            df = pd.DataFrame([[0, 0, 0]],
                              index = [index],
                              columns = ['n', 'mu', 'var'])
            self.statistics_df = self.statistics_df.append(df)
        n   = self.statistics_df['n'][index]
        mu  = self.statistics_df['mu'][index]
        var = self.statistics_df['var'][index]
        n_new = n + array.size
        mu_new = (mu * n + np.sum(array)) / n_new
        var_new = ((n * (var + mu ** 2) + np.sum(array ** 2)) / n_new) -
(mu_new ** 2)
        self.statistics_df.update(pd.DataFrame([[n_new, mu_new, var_new]],
                                               index = [index],
                                               columns = ['n', 'mu', 'var']))

    def normalize_array(self, index, array):
        if self.training_mode:
            self.memorize(index, array[array > self.MINIMUM_VALUE])
        mu  = self.statistics_df['mu'][index]
        var = self.statistics_df['var'][index]
        if var <= 0:
            return np.zeros_like(array)
        arr = (array - mu) / (var ** 0.5)
        arr[arr < self.MINIMUM_VALUE] = self.MINIMUM_VALUE
        return arr
```

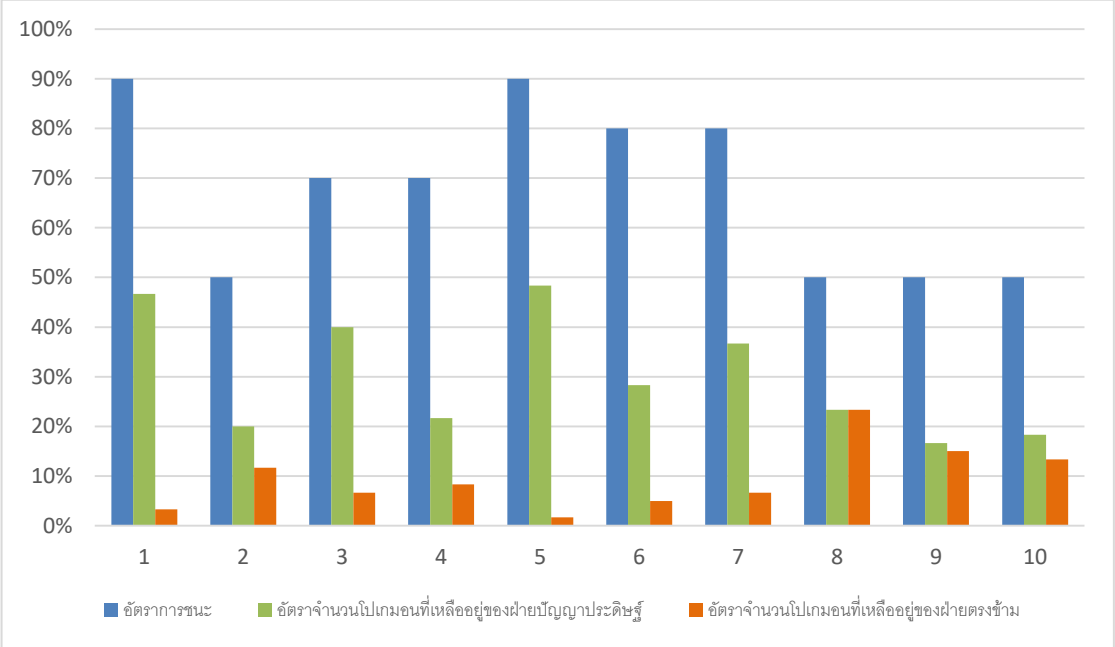
ภาพที่ 4.1 ตัวอย่างซอร์สโค้ดการคำนวณคะแนนมาตรฐาน

4.2 ผลของการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่เหมือนกัน

ผลการแข่งขันระหว่างปัญญาประดิษฐ์กับการเล่นแบบสุ่มโดยใช้ทีมที่จัดไว้ให้ซึ่งจะเป็นทีมที่เหมือนกันทั้งสองฝ่าย

ตารางที่ 4.1 ผลการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่เหมือนกัน

ทีมที่	ชื่อทีม	จำนวนครั้งการชนะของฝ่ายปัญญาประดิษฐ์	จำนวนโปเกมอนที่เหลืออยู่	
			ฝ่ายปัญญาประดิษฐ์	ฝ่ายการเล่นแบบสุ่ม
1	Red (PWT)	9	28	2
2	Blue (PWT)	5	12	7
3	Lance (PWT-6)	7	24	4
4	Steven (PWT)	7	13	5
5	Wallace (PWT)	9	29	1
6	Cynthia (PWT)	8	17	3
7	Alder (PWT)	8	22	4
8	Iris (B2W2)	5	14	14
9	Diantha (XY)	5	10	9
10	Kukui (SM)	5	11	8
รวมทั้งหมด		**68**	**180**	**57**



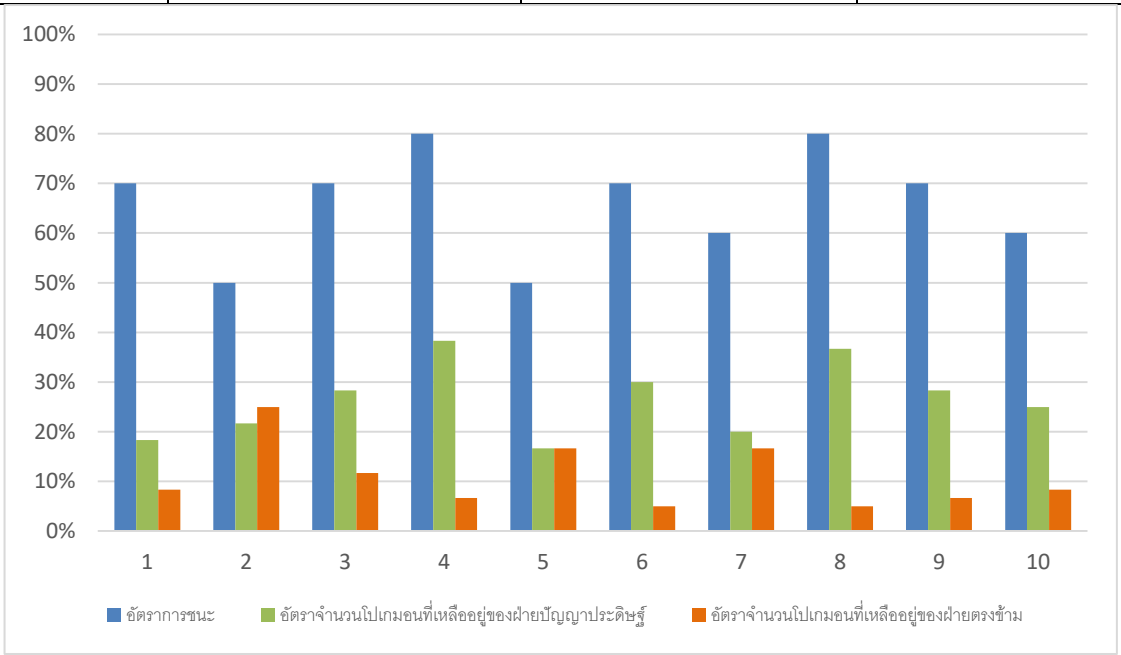
ภาพที่ 4.2 อัตราการชนะเมื่อเล่นกับการเล่นแบบสุ่มโดยใช้ทีมที่เหมือนกัน

4.3 ผลของการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่ต่างกัน

ผลการแข่งขันระหว่างปัญญาประดิษฐ์กับการเล่นแบบสุ่มโดยใช้ทีมที่จัดไว้ให้ซึ่งจะเป็นทีมที่ต่างกัน

ตารางที่ 4.2 ผลการแข่งขันกับการเล่นแบบสุ่มโดยใช้ทีมที่ต่างกัน

คู่ที่	ทีมที่		จำนวนครั้งการชนะของฝ่ายปัญญาประดิษฐ์		จำนวนโปเกมอนที่เหลืออยู่			
					ฝ่ายปัญญาประดิษฐ์		ฝ่ายการเล่นแบบสุ่ม	
			ทีมด้านซ้าย	ทีมด้านขวา	ทีมด้านซ้าย	ทีมด้านขวา	ทีมด้านซ้าย	ทีมด้านขวา
1	1	2	4	3	6	5	3	2
2	3	4	5	0	13	0	15	0
3	5	6	3	4	7	10	1	6
4	7	8	5	3	15	8	4	0
5	9	10	1	4	0	10	1	9
6	1	6	2	5	5	13	0	3
7	2	7	2	4	4	8	2	8
8	3	8	5	3	13	9	3	0
9	4	9	4	3	12	5	3	1
10	5	10	2	4	6	9	1	4
รวมทั้งหมด			**66**		**158**		**66**	



ภาพที่ 4.3 อัตราการชนะเมื่อเล่นกับการเล่นแบบสุ่มโดยใช้ทีมที่ต่างกัน

4.4 การอภิปรายผลการทดลอง

การวัดผลในส่วนที่ 1 ซึ่งใช้ทีมที่เหมือนกันทั้งสองฝ่าย ปัญญาประดิษฐ์ชนะ 68 เกมจาก 100 เกมซึ่งคิดเป็นอัตราการชนะ 68% และในส่วนที่ 2 ซึ่งใช้ทีมที่ต่างกัน ปัญญาประดิษฐ์ชนะ 66 เกมจาก 100 เกมซึ่งคิดเป็นอัตราการชนะ 66% จากการวัดผลทั้ง 2 ส่วนจะได้ว่าปัญญาประดิษฐ์ชนะ 134 เกมจาก 200 เกมซึ่งคิดเป็นอัตราการชนะ 67%

จากอัตราการชนะและจำนวนโปเกมอนที่เหลืออยู่ จะพบว่าในแต่ละทีมปัญญาประดิษฐ์สามารถเล่นได้เทียบเท่าดีกว่าการเล่นแบบสุ่มเล็กน้อย ยกเว้นการวัดผลในส่วนที่ 2 คู่ที่ 2 ที่เล่นได้แย่กว่าเล็กน้อย แต่โดยรวมแล้วก็ยังมีอัตราที่ชนะและจำนวนโปเกมอนที่เหลืออยู่ที่สูงกว่าเล็กน้อยซึ่งอาจเป็นตัวชี้วัดได้ว่าปัญญาประดิษฐ์อาจเรียนรู้ได้

จากการสังเกตการเล่นของปัญญาประดิษฐ์แล้วพบว่าปัญญาประดิษฐ์สามารถทำตามวัตถุประสงค์หลัก ๆ ที่กำหนดในหัวข้อที่ 3.5 ได้บางส่วนคือ

1. ในส่วนของการโจมตีก็สามารถเลือกท่าโจมตีที่ชนะทางได้เป็นบางครั้ง แต่ก็พบว่ายังมีการเลือกที่ไม่ค่อยดีนัก เช่นทีมที่ 3 นั้นจะใช้โปเกมอนประเภทมังกรทุกตัวซึ่งจะแพ้การโจมตีประเภทมังกรทั้งหมดและแพ้การโจมตีประเภทน้ำแข็งเกือบทั้งหมด เพื่อทดสอบว่าปัญญาประดิษฐ์จะเลือกท่าโจมตีประเภทมังกรหรือน้ำแข็งตลอดเวลาหรือไม่ พบว่าปัญญาประดิษฐ์เลือกการโจมตีดังกล่าวเป็นบางครั้ง
2. การเลือกสลับตัวนั้นค่อนข้างทำได้ไม่ดี คือปัญญาประดิษฐ์ทำการสลับตัวบ่อยเกินไปทำให้เปิดโอกาสให้ฝ่ายตรงข้ามโจมตีได้ต่อเนื่องและไม่สามารถโจมตีโต้ตอบได้ อย่างไรก็ตามผู้พัฒนาได้ลองให้แสดงผลความน่าจะเป็นของการเล่นในเทิร์นแรกที่เลือกตัวไม่เสียเปรียบฝ่ายตรงข้ามก็พบว่าความน่าจะเป็นของการสลับตัวน้อยกว่าการใช้ท่าโจมตีเล็กน้อย
3. ปัญญาประดิษฐ์จะเลือกใช้การโจมตีที่ส่งผลระยะยาวในบางครั้ง เช่นการทำให้ฝ่ายตรงข้ามติดสถานะผิดปกติ แต่อาจไม่สามารถเรียนรู้กติกาของเกมที่ว่าสถานะผิดปกติถาวรนั้นไม่สามารถซ้อนทับกันได้ เนื่องจากสังเกตว่าปัญญาประดิษฐ์พยายามใช้สถานะผิดปกติถาวรในโปเกมอนที่ติดสถานะผิดปกติถาวรอยู่แล้ว และการสลับตัวหลังจากใช้ท่าที่เพิ่มพลังซึ่งจะทำให้การเพิ่มพลังดังกล่าวนั้นสูญเปล่า
4. การเลือกท่าโจมตีที่มีลำดับความสำคัญสูงเป็นสิ่งที่พบว่าปัญญาประดิษฐ์สามารถทำได้ดีพอสมควร พบว่าในหลาย ๆ ครั้งที่โปเกมอนฝ่ายตรงข้ามมีพลังชีวิตเหลือน้อยก็จะใช้เพื่อที่จะมั่นใจได้ว่าฝ่ายตรงข้ามไม่ได้โจมตีก่อน หรือแม้แต่ขณะที่ฝ่ายเดียวกันมีพลังชีวิตเหลือน้อยก็จะใช้เพื่อที่จะได้โอกาสในการโจมตีครั้งสุดท้าย

5. การใช้ท่าโจมตีที่ผสมผสานกัน ถือว่าปัญญาประดิษฐ์ทำไม่ได้เลย คือการพยายามใช้ท่าโจมตีอื่นนอกเหนือจาก Sleep Talk ขณะติดสถานะหลับ หรือไม่ก็ใช้ Sleep Talk ขณะที่ไม่ติดสถานะหลับ

นอกจากนี้ ผู้พัฒนายังได้ทดลองเล่นกับตัวปัญญาประดิษฐ์แล้วโดยใช้วิธีเดียวกับการวัดผล แต่พบว่าปัญญาประดิษฐ์ยังไม่สามารถเอาชนะมนุษย์ได้แม้แต่เกมเดียว

บทที่ 5
ข้อสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึง สรุปผลจากข้อมูลผลการทดลองในบทที่ 4 ข้อสรุปโครงงานปัญญาประดิษฐ์ในเกมโปเกมอนที่พัฒนาขึ้น ปัญหาที่พบและวิธีการแก้ปัญหา รวมถึงข้อเสนอแนะอื่น ๆ ที่ได้จากโครงงาน

5.1 ข้อสรุป

จากการฝึกฝนทั้งหมด 1,250 เกม จะได้ว่าปัญญาประดิษฐ์มีอัตราชนะเมื่อสู้กับการเล่นแบบสุ่ม 67% ซึ่งเล่นได้ดีกว่าการเล่นแบบสุ่มเล็กน้อย แต่ยังไม่สามารถที่จะเอาชนะมนุษย์ได้ อย่างไรก็ตามผลลัพธ์ที่ได้อาจเป็นตัวชี้วัดได้ว่าปัญญาประดิษฐ์อาจเรียนรู้ได้

5.2 ปัญหาที่พบ

ปัญหาหลัก ๆ ที่เกิดขึ้นคือมีข้อจำกัดทางด้านเวลา จึงฝึกฝนและวัดผลตัวแบบได้น้อย และไม่สามารถปรับปรุงตัวแบบได้มากนัก ส่วนปัญหาอื่น ๆ ที่พอแก้ไขได้มีดังนี้

1. ปัญญาประดิษฐ์พยายามสลับตัวโปเกมอนไปมาทำให้เกมไม่มีวันจบ
2. การทำงานของโปรแกรมใช้เวลานานมากเมื่อทำการฝึกฝนโดยการสู้กับตัวเอง
3. ค่า Q ที่ได้จากโครงข่ายประสาทเทียมมีค่าผิดปกติและเกินจากค่าสูงสุดที่เป็นไปได้

5.3 วิธีการแก้ปัญหา

จากปัญหาที่พบดังกล่าวผู้พัฒนาได้แก้ปัญหาด้วยวิธีดังนี้

1. ใช้การเพิ่มความน่าจะเป็นในการเลือกท่าโจมตีเมื่อเทิร์นผ่านไป
2. หลีกเลี่ยงการแตกเธรดในภาษาไพธอน แต่ให้สั่งทำงาน 2 โปรเซสแทน
3. ใช้การบีบค่าด้วยฟังก์ชันไฮเพอร์โบลิกแทนเจนท์ และควบคุมค่าตอบแทนที่คาดว่าจะได้รับในอนาคต

5.4 ข้อเสนอแนะ

สิ่งที่น่าจะทำต่อยอดได้น่าจะเป็นการปรับปรุงตัวแบบโดยการเพิ่มความซับซ้อนให้มากขึ้น การฝึกฝนตัวแบบด้วยจำนวนเกมที่มากขึ้นโดยอาจจัดการไฟล์ให้ดีขึ้นเพื่อที่ปัญญาประดิษฐ์จะฝึกฝนได้ทีละหลาย ๆ เกมพร้อมกัน การปรับปรุงวิธีการจัดการข้อมูล และการปรับปรุงการเลือกการกระทำให้มีค่าต่างกันมากขึ้น นอกจากนี้ยังมีเรื่องของการปรับค่าไฮเพอร์พารามิเตอร์ที่เหมาะสมอีกด้วย

สิ่งที่อาจจะพัฒนาต่อไปได้คือเรื่องของการทำนายข้อมูลที่มองไม่เห็น เช่นการทำนายท่าโจมตีของโปเกมอนฝ่ายตรงข้าม จริง ๆ แล้วผู้พัฒนาได้วางแผนจะใช้การเก็บข้อมูลว่าโปเกมอนตัวใดมักมีท่าโจมตีแบบใดแล้วหาความน่าจะเป็นเพื่อที่จะทำนายท่าโจมตี แต่ด้วยข้อจำกัดทางด้านเวลาจึงจำเป็นต้องตัดส่วนนี้ออกไป

การพัฒนาวิธีการคำนวณค่าตอบแทนก็เป็นสิ่งที่น่าสนใจ เพราะในโครงงานนี้อาจมีปัญหาเรื่องปัญญาประดิษฐ์ไม่สามารถแยกค่าตอบแทนที่กำกวมได้ เช่นการที่โปเกมอนของฝ่ายตรงข้ามใช้ Memento ซึ่งทำให้ตัวเองสูญเสียพลังชีวิตทั้งหมด เมื่อคำนวณค่าตอบแทนแล้ว ไม่ว่าจะเลือกการกระทำใด ๆ ก็จะได้ค่าตอบแทนเป็น 1 ซึ่งอาจจะทำให้ไม่สามารถเรียนรู้ได้อย่างถูกต้อง

รายการอ้างอิง

[1] Nintendo, Game Freak, Creature Inc. 1996. Pokémon [Online]. Available from: https://www.pokemon.com/

[2] Guangcong Luo. 2011. Pokémon Showdown [Online]. Available from: https://github.com/smogon/pokemon-showdown

[3] Christopher Berner, Greg Brockman, Brooke Chan, et al. Dota 2 with Large Scale Deep Reinforcement Learning arXiv preprint arXiv:1912.06680, 2019.

[4] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, et al. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model arXiv preprint arXiv:1911.08265, 2020.

[5] Akshay Kalose, Kris Kaya, and Alvin Kim. 2018. Optimal Battle Strategy in Pokémon using Reinforcement Learning [Online]. Available from: https://web.stanford.edu/class/aa228/reports/2018/final151.pdf

[6] David Simões, Simão Reis, Nuno Lau, and Luís Paulo Reis. Competitive Deep Reinforcement Learning over a Pokémon Battling Simulator. 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 2020, pp. 40-45.

[7] liris. 2019. websocket_client [Online]. Available from: https://pypi.org/project/websocket_client/

[8] Archaic. 2005. Bulbapedia [Online]. Available from: https://bulbapedia.bulbagarden.net/wiki/Main_Page

[9] Scott Lee and Julian Togelius. Showdown AI Comptition. IEEE Conference on Computational Intelligence and Games 2017, New York, NY, 2017, pp. 191-198.

[10] veekun. 1999. pokedex [Online]. Available from: https://github.com/veekun/pokedex

[11] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357-362 (2020). DOI: 0.1038/s41586-020-2649-2.

[12] Martín Abadi, Ashish Agarwal, Paul Barham, et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous system [Online]. Available from: https://tensorflow.org

[13] François Chollet. 2015. Keras [Online]. Available from: https://keras.io

[14] Chris Yoon. 2019. Vanilla Deep O Networks [Online]. Available from: https://towardsdatascience.com/dqn-part-1-vanilla-deep-q-networks-6eb4a00febfb

[15] Cinergix Pty Ltd. 2008. Creately [Online]. Available from: app.creately.com

ภาคผนวก

ภาคผนวก ก
แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal
ปีการศึกษา 2563

ชื่อโครงงาน (ภาษาไทย) ปัญญาประดิษฐ์ในเกมโปเกมอน
ชื่อโครงงาน (อังกฤษ) Artificial Intelligence in Pokémon Game
อาจารย์ที่ปรึกษา อาจารย์ โชติรส สุรพลชัย
ผู้ดำเนินการ นาย กวิน ไวทยวงศ์สกุล เลขประจำตัวนิสิต 6033602023
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

---

1. **หลักการและเหตุผล**

โปเกมอนเป็นเกมเล่นตามบทบาทแบบเทิร์นเบส (Turn-based) ที่เกิดขึ้นในปี ค.ศ. 1996 มีการจัดการแข่งขันมากมาย [1] โดยผู้เล่นจะต้องเตรียมทีมซึ่งประกอบไปด้วยโปเกมอน 6 ตัวเพื่อต่อสู้กับทีมของฝ่ายตรงข้าม โปเกมอนแต่ละตัวสามารถเรียนรู้ท่าโจมตีได้หลากหลายโดยผู้เล่นจะต้องเลือกมาเพียง 4 ท่าโจมตี อีกทั้งยังมีการให้โปเกมอนถือสิ่งของเพื่อเพิ่มความสามารถต่าง ๆ ทำให้โปเกมอนเป็นเกมที่มีแผนการเล่นที่หลากหลาย เนื่องจากโปเกมอนเป็นเกมที่มีกลยุทธ์ที่หลากหลายทำให้การพัฒนาปัญญาประดิษฐ์ในเกมโปเกมอนเป็นสิ่งที่น่าสนใจ

โปเกมอนโชว์ดาวน์ (Pokémon Showdown) [2] เป็นแพลตฟอร์ม (Platform) ที่ใช้ในการจำลองเกมโปเกมอนซึ่งจะเน้นไปที่การต่อสู้ของโปเกมอนเพียงอย่างเดียวเพื่อที่จะให้ผู้เล่นสามารถจัดทีมได้สะดวก เนื่องจากว่าผู้เล่นไม่ต้องทำการจับโปเกมอนและฝึกฝนโปเกมอนเองทำให้ผู้เล่นเลือกวางแผนกลยุทธ์ที่หลากหลายได้มากขึ้น นอกจากนั้นโปเกมอนโชว์ดาวน์เป็นแพลตฟอร์มที่เป็นโปรแกรมประยุกต์บนเว็บสามารถเชื่อมต่อเข้าถึงได้ง่ายทำให้สามารถเปิดให้ผู้ทดสอบและผู้ที่ต้องการจะเล่นกับปัญญาประดิษฐ์สามารถเล่นผ่านเครือข่ายอินเทอร์เน็ตได้ ดังนั้นผู้พัฒนาจึงเลือกใช้โปเกมอนโชว์ดาวน์เป็นแพลตฟอร์มสำหรับการพัฒนาปัญญาประดิษฐ์

โดยทั่วไปการพัฒนาปัญญาประดิษฐ์ในเกมที่เคยเอาชนะผู้เล่นที่เป็นผู้ชนะเลิศของการแข่งขันเกมนั้น ๆ มักจะใช้วิธีการเรียนรู้แบบเสริมกำลัง [3][4] ซึ่งเป็นการเรียนรู้ที่ดูผลตอบรับจากระบบของการกระทำ หากผลตอบรับระบุว่าดีจะส่งเสริมการกระทำดังกล่าวให้เลือกใช้มากขึ้น แต่ถ้าผลตอบรับไม่ดีจะลดการเลือกใช้การกระทำดังกล่าว เนื่องจากว่าเกมเป็นสิ่งที่มีผลลัพธ์

ของการเล่นชัดเจนแต่ไม่สามารถบ่งบอกได้ชัดเจนว่า การตัดสินใจดังกล่าวถูกหรือผิดซึ่งเหมาะกับการใช้โมเดลการเรียนรู้แบบเสริมกำลัง การพัฒนาปัญญาประดิษฐ์ในโครงงานนี้จึงเลือกใช้โมเดลการเรียนรู้แบบเสริมกำลัง

โครงงานนี้จึงต้องการพัฒนาปัญญาประดิษฐ์ที่สามารถเล่นได้หลากหลายกลยุทธ์ โดยเลือกใช้ตัวแบบการเรียนรู้แบบเสริมกำลังซึ่งผ่านการฝึกฝนที่สามารถเล่นกับผู้เล่น เพื่อสร้างความบันเทิงให้กับผู้เล่นและสามารถใช้ให้ผู้เล่นวางแผนการเล่นและทดสอบกลยุทธ์ได้ การฝึกฝนตัวแบบดังกล่าวจะทำโดยการให้ปัญญาประดิษฐ์ต่อสู้กับตัวเองซึ่งจะใช้ทีมจากระบบการสุ่มของโปเกมอนโชว์ดาวน์ซึ่งใช้วิธีการสุ่มแบบตัดกรณีที่ได้โปเกมอนที่ไม่เหมาะสม เช่น ได้ท่าโจมตี 2 ท่า มีลักษณะคล้ายกัน และเพิ่มลำดับความสำคัญบางอย่าง เช่นการให้โปเกมอนถือสิ่งของเฉพาะตัวจะได้พลังที่เพิ่มขึ้นมาก การวัดผลตัวแบบดังกล่าวจะใช้การให้ต่อสู้กับการเล่นแบบสุ่มเพื่อให้สอดคล้องกับงานวิจัยอื่น ๆ [5][6] โดยจะเป็นการสุ่มแบบยูนิฟอร์ม (Uniform) เลือกจากการกระทำทั้งหมดที่เป็นไปได้

2. **วัตถุประสงค์** เพื่อพัฒนาปัญญาประดิษฐ์ที่สามารถเล่นเกมโปเกมอนโชว์ดาวน์ได้ดีกว่าการเล่นแบบสุ่มบนระบบปฏิบัติการวินโดวส์ (Windows)

3. **ขอบเขตของโครงงาน**

3.1 ปัญญาประดิษฐ์จะเล่นได้ในรูปแบบ [Gen 7] Anything Goes ซึ่งจะเป็นแบบฝ่ายละ 1 คน และใช้โปเกมอน 6 ตัว โดยใช้โปเกมอนได้ทีละ 1 ตัว

3.2 ปัญญาประดิษฐ์จะเล่นได้หลายทีมโดยไม่จำเป็นต้องฝึกฝนโมเดลใหม่

3.3 ปัญญาประดิษฐ์จะเล่นตามกติกาของโปเกมอนโชว์ดาวน์

3.4 ปัญญาประดิษฐ์จะทำงานได้ก็ต่อเมื่อการเชื่อมต่อกับเซิร์ฟเวอร์ (Server) ไม่เกิดข้อผิดพลาดและข้อความที่ได้รับจากเซิร์ฟเวอร์ครบถ้วนและเรียงลำดับถูกต้อง

3.5 ปัญญาประดิษฐ์จะไม่สามารถเล่นได้หากมีโปเกมอนที่มีความสามารถ Illusion

3.6 ปัญญาประดิษฐ์จะไม่เลือกใช้ Z-Move และไม่สามารถเล่นได้ถ้าฝ่ายตรงข้ามเลือกใช้สิ่งนั้น

3.7 ปัญญาประดิษฐ์จะเล่นได้ในระบบปฏิบัติการวินโดวส์ 10 64-bit

3.8 ปัญญาประดิษฐ์จะเล่นโดยใช้เซิร์ฟเวอร์จำลองที่เปิดขึ้นในคอมพิวเตอร์เครื่องเดียวกัน โดยแพลตฟอร์มที่ใช้จะแสดงผลเป็นภาษาอังกฤษ และเชื่อมต่อกันผ่านเว็บซ็อกเก็ต (websocket) [7] โดยใช้ภาษาไพธอน (Python) เซิร์ฟเวอร์จำลองดังกล่าวจะใช้การโคลน (Clone) โอเพนซอร์ส (Open source) ของโปเกมอนโชว์ดาวน์ [2]

3.9 การวัดผลจะวัดผลเทียบอัตราแพ้ชนะโดยวิธีการเล่นแบบสุ่ม

4. **ขั้นตอนการดำเนินการ**

4.1 พัฒนาโปรแกรมส่วนเชื่อมต่อเข้ากับตัวเซิร์ฟเวอร์ของโปเกมอนโชว์ดาวน์ [2] ที่จำลองในคอมพิวเตอร์เครื่องเดียวกันบนระบบปฏิบัติการ Windows 10 64-bit

4.2 ออกแบบตัวแบบที่ใช้ในงานปัญญาประดิษฐ์

4.3 พัฒนาโปรแกรมส่วนตัวปัญญาประดิษฐ์

4.4 ออกแบบวิธีการวัดผลและทดสอบการทำงานของปัญญาประดิษฐ์

4.5 วัดผลและทดสอบการทำงานของปัญญาประดิษฐ์

4.6 จัดทำเอกสารประกอบโครงงาน

ตารางเวลาการดำเนินการ

ขั้นตอนการดำเนินการ	เดือน / ปีการศึกษา 2563								
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. พัฒนาโปรแกรมส่วนเชื่อมต่อเข้ากับตัวเซิร์ฟเวอร์									
2.ออกแบบตัวแบบ									
3.พัฒนาโปรแกรมส่วนตัวปัญญาประดิษฐ์									
4.ออกแบบวิธีการวัดผลและทดสอบ									
5.วัดผลและทดสอบ									
6.จัดทำเอกสารประกอบโครงงาน									

5. **ประโยชน์ที่คาดว่าจะได้รับ**

5.1 ประโยชน์ต่อผู้พัฒนาโปรแกรม

5.1.1 ได้ฝึกฝนทักษะในการเขียนโปรแกรมภาษาไพธอนในการเชื่อมต่อกับเซิร์ฟเวอร์และการทำปัญญาประดิษฐ์

5.1.2 ได้ฝึกฝนทักษะในการวางแผนตัวแบบที่ใช้ในปัญญาประดิษฐ์

5.2 ประโยชน์ต่อผู้ใช้

5.2.1 ได้รับความบันเทิงในการรับชมหรือในการเล่นกับปัญญาประดิษฐ์

5.2.2 ได้กลยุทธ์ในการเล่นใหม่ ๆ

6. **อุปกรณ์และเครื่องมือที่ใช้**

6.1 ฮาร์ดแวร์

6.1.1 แล็ปท็อปคอมพิวเตอร์

6.1.1.1 Intel® Core™ i7-7500 CPU @ 2.70 GHz

6.1.1.2 RAM 8.00 GB

6.1.1.3 ระบบปฏิบัติการ Windows 10 64-bit

6.1.1.4 GPU: NVIDIA GeForce 940MX

6.2 ซอฟท์แวร์

6.2.1 Anaconda Python 3 [8]

6.2.2 Spyder [9]

6.2.3 Google Chrome [10] (ใช้ในการทดสอบระบบบน Local Host)

6.2.4 เซิร์ฟเวอร์ Pokémon Showdown [2]

7. **เอกสารอ้างอิง**

[1] Nintendo, Game Freak, Creature Inc. 1996. Pokémon Championship Series [Online]. Available from: https://www.pokemon.com/us/play-pokemon/pokemon-events/pokemon-tournaments/

[2] Guangcong Luo. 2011. Pokémon Showdown [Online]. Available from: https://github.com/smogon/pokemon-showdown

[3] Christopher Berner, Greg Brockman, Brooke Chan, et al. Dota 2 with Large Scale Deep Reinforcement Learning arXiv preprint arXiv:1912.06680, 2019.

[4] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, et al. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model arXiv preprint arXiv:1911.08265, 2020.

[5] Akshay Kalose, Kris Kaya, and Alvin Kim. 2018. Optimal Battle Strategy in Pokémon using Reinforcement Learning [Online]. Available from: https://web.stanford.edu/class/aa228/reports/2018/final151.pdf

[6] David Simões, Simão Reis, Nuno Lau, and Luís Paulo Reis. Competitive Deep Reinforcement Learning over a Pokémon Battling Simulator. 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 2020, pp. 40-45.

[7] liris. 2019. websocket_client [Online]. Available from: https://pypi.org/project/websocket_client/

[8] Python Software Foundation. 1995. Python Language Reference [Online]. Available from: http://www.python.org

[9] Pierre Raybaut. 2009. Spyder [Online]. Available from: https://www.spyder-ide.org/

[10] Google LLC. 2008. Google Chrome [Online]. Available from: https://www.google.com/chrome/

ภาคผนวก ข
คำศัพท์และกติกาที่เกี่ยวกับเกมโปเกมอน

คำศัพท์หลัก ๆ ที่ใช้ โดยจะวงเล็บต่อท้ายเป็นชื่อภาษาอังกฤษที่ใช้ในเกม ซึ่งบางคำใช้ทับศัพท์

ข.1 โปเกมอน (Pokémon)

ในที่นี้หมายถึงตัวโปเกมอน หากจะพูดถึงชื่อเกมจะพยายามใช้คำว่าเกมโปเกมอน
โปเกมอน 1 ตัวจะมีประเภทอยู่ 1 - 2 ประเภทขึ้นอยู่กับสปีชีส์ ซึ่งจะเรียนรู้ท่าโจมตีได้สูงสุด 4 ท่าในเวลาเดียวกัน จะสามารถถือสิ่งของได้ 1 ชิ้น โปเกมอนแต่ละสปีชีส์จะมีความสามารถ 1 - 3 แบบ แต่ว่าโปเกมอน 1 ตัวจะมีความสามารถเพียงแค่ 1 แบบจาก 1 – 3 แบบนั้น
โปเกมอนจะมีค่า IV และ EV รวมไปถึงลักษณะนิสัยที่จะปรับค่าพลังได้

ข.2 ประเภท (Type)

จะไว้กำกับประเภทของโปเกมอนและท่าโจมตี โดยในเกมโปเกมอนจะมี 18 ประเภทดังนี้

1. ธรรมดา (Normal)
2. ไฟ (Fire)
3. น้ำ (Water)
4. ไฟฟ้า (Electric)
5. พืช (Grass)
6. น้ำแข็ง (Ice)
7. ต่อสู้ (Fighting)
8. พิษ (Poison)
9. ดิน (Ground)
10. บิน (Flying)
11. พลังจิต (Psychic)
12. แมลง (Bug)
13. หิน (Rock)
14. ผี (Ghost)
15. มังกร (Dragon)
16. มืด (Dark)
17. เหล็ก (Steel)
18. แฟรี่ (Fairy)

โดยประเภทแต่และประเภทจะได้เปรียบและเสียเปรียบประเภทอื่น ๆ แตกต่างกันไป เมื่อผู้เล่นสามารถโจมตีด้วยท่าโจมตีที่ชนะทางโปเกมอนของอีกฝ่าย (Super Effective) ความเสียหายจะแรงขึ้นเป็น 2 เท่า แต่ถ้าหากโปเกมอนอีกฝ่ายมีประเภทที่ต้านทานประเภทของท่าโจมตี (Resisted / Not Very Effective) ความเสียหายจะลดลงเหลือเพียงครึ่งเดียว และหากโปเกมอนอีกฝ่ายมีประเภทที่ไม่รับการโจมตีประเภทดังกล่าว (Immune) ก็จะไม่สร้างความเสียหายใด ๆ การคำนวณความเสียหายนั้นจะใช้ผลคูณทั้งหมด เช่น Camerupt เป็นโปเกมอนที่มีประเภทไฟและดิน ถ้าหากโดนท่าโจมตีประเภทพืชจะได้รับการคูณ 2 จากการที่โปเกมอนประเภทดินแพ้ท่าโจมตีประเภทพืช แต่จะกลับมา

เป็น 1 เท่าตามเดิมเพราะโปเกมอนไฟจะต้านทานการโจมตีพืช ตารางต่อไปนี้จะแสดงการแพ้ชนะของประเภททั้งหมด

ตารางที่ ข.1 ตารางแสดงการแพ้ชนะของประเภทแต่ละประเภทในเกมโปเกมอน

Atk \ Def	Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	Dark	Steel	Fairy
Normal													R	I			R	
Fire		R	R		S	S						S	R		R		S	
Water		S	R		R				S				S		R			
Electric			S	R	R				I	S					R			
Grass		R	S		R			R	S	R		R	S		R		R	
Ice		R	R		S	R			S	S					S		R	
Fighting	S					S		R		R	R	R	S	I		S	S	R
Poison					S			R	R				R	R			I	S
Ground		S		S	R			S		I		R	S				S	
Flying				R	S		S					S	R				R	
Psychic							S	S			R					I	R	
Bug		R			S		R	R		R	S			R		S	R	R
Rock		S				S	R		R	S		S					R	
Ghost	R										S			S		R		
Dragon															S		R	I
Dark							R				S			S		R		R
Steel		R	R	R		S							S				R	S
Fairy		R					S	R							S	S	R	

โดย Atk ระบุว่าเป็นประเภทของท่าโจมตีของฝ่ายที่โจมตี ส่วน Def ระบุว่าเป็นประเภทของโปเกมอนที่ถูกโจมตี

ช่องสีเขียว S หมายถึง Super Effective

ช่องสีแดง R หมายถึง Resisted

ช่องสีเทา I หมายถึง Immune

ข.3 ค่าพลัง (Stats)

โปเกมอน 1 ตัวจะมีค่าพลังถาวร 6 ค่าดังนี้

1. ค่าพลังชีวิตสูงสุด (Max HP / Max Hit Points)
   โปเกมอนโชว์ดาวน์จะเริ่มการต่อสู้ด้วยโปเกมอนที่มีพลังชีวิตเป็นค่าพลังชีวิตสูงสุดทั้งหมด โดยหากโปเกมอนมีค่าพลังชีวิตเหลือ 0 จะถือว่าสู้ต่อไม่ได้
2. ค่าพลังโจมตี (Attack)
   เป็นค่าที่กำหนดความเสียหายตามสูตรในหัวข้อที่ 2.2 ใช้กับท่าโจมตีหมวดหมู่ Physical เท่านั้น
3. ค่าพลังป้องกัน (Defense)
   เป็นค่าที่กำหนดความเสียหายตามสูตรในวข้อที่ 2.2 ใช้กับท่าโจมตีหมวดหมู่ Physical เท่านั้น
4. ค่า Special Attack
   เป็นค่าที่กำหนดความเสียหายตามสูตรในวข้อที่ 2.2 ใช้กับท่าโจมตีหมวดหมู่ Special เท่านั้น
5. ค่า Special Defense
   เป็นค่าที่กำหนดความเสียหายตามสูตรในวข้อที่ 2.2 ใช้กับท่าโจมตีหมวดหมู่ Special เท่านั้น
6. ค่าความเร็ว (Speed)
   เป็นค่าที่กำหนดว่าโปเกมอนตัวไหนจะเริ่มโจมตีก่อนหรือหลัง

นอกจากนี้ยังมีค่าพลังชั่วคราวดังนี้

7. ค่าความแม่นยำ (Accuracy)
   เป็นค่าที่ใช้ระบุตัวคูณความแม่นยำของการโจมตี
8. ค่าการหลบหลีก (Evasion)
   เป็นค่าที่ใช้ระบุตัวหารความแม่นยำของการโจมตีของฝ่ายตรงข้าม

ค่าพลังถาวรจะมีค่าพื้นฐานแตกต่างกันออกไปสำหรับโปเกมอนแต่ละสปีชีส์ แต่ค่าพลังชั่วคราวจะไม่ต่างกันคือ 1

การเพิ่มหรือลดค่าพลังจะใช้การเรียกว่าขั้น (Stage) โดยจะเริ่มที่ขั้นที่ 0 เพิ่มได้มากที่สุดคือ +6 และลดให้น้อยที่สุดคือ -6 การเปลี่ยนแปลงค่าพลังนี้จะมีผลชั่วคราวจนกว่าผู้เล่นจะเปลี่ยนตัวโปเกมอน โดยค่าพลังถาวรจะใช้สูตรในการเปลี่ยนแปลงต่างจากค่าพลังชั่วคราว ผลคูณของค่าพลังที่เปลี่ยนแปลงจะมีสูตรดังนี้

$$mult(stat) = \begin{cases} \dfrac{2 + \max(0,\ stage_{stat})}{2 - \min(0, stage_{stat})} & ;\quad stat \in Stat_{perma} \\ \dfrac{3 + \max(0, stage_{stat})}{3 - \min(0, stage_{stat})} & ;\quad stat \in Stat_{temp} \end{cases} \tag{13}$$

ซึ่งตัวแปรต่าง ๆ มีความหมายดังนี้

$mult(stat)$ หมายถึงผลคูณของค่าพลัง

$stage_{stat}$ หมายถึงจำนวนขั้นที่เปลี่ยนแปลงไปสำหรับค่าพลังนั้น ๆ

$Stat_{perma}$ หมายถึงเซ็ตของค่าพลังถาวร

$Stat_{temp}$ หมายถึงเซ็ตของค่าพลังชั่วคราว

สิ่งที่เปลี่ยนแปลงค่าพลังที่น่าสนใจมีดังนี้

1. ค่าเฉพาะตัว (IV / Individual Value)
   เป็นค่าที่จะสุ่มเมื่อเจอโปเกมอนนั้นเป็นครั้งแรกจะมีค่าระหว่าง 0 – 31 ซึ่งในการแข่งขันมักจะเลือกใช้ค่านี้เป็น 31 ทั้งหมด ยกเว้นบางกรณีที่ต้องการให้ค่าพลังน้อยเช่น Trick Room จะมีผลทำให้โปเกมอนที่ความเร็วน้อยกว่าจะได้โจมตีก่อน
2. ค่าการฝึกฝนของโปเกมอน (EV / Effort Value)
   เป็นค่าที่จะได้จากการเอาชนะโปเกมอนตัวอื่น ๆ ในการเล่นเกมปกติ มีค่าระหว่าง 0 – 252 และรวมกันสำหรับค่าพลังถาวรทั้ง 6 แบบจะไม่เกิน 510
3. ลักษณะนิสัยของโปเกมอน (Nature)
   จะเพิ่มค่าพลัง 1 ค่าและลดค่าพลัง 1 ค่า 10% ซึ่งหากเพิ่มและลดค่าพลังเดียวกันจะถือว่าหักล้างกันและไม่เปลี่ยนแปลงค่าพลังใด ๆ ซึ่งมีทั้งหมด 25 แบบดังนี้

ตารางที่ ข.2 ตารางแสดงการเพิ่มลดค่าพลังตามลักษณะนิสัยของโปเกมอน

ลด / เพิ่ม	Attack	Defense	Sp. Atk.	Sp. Def.	Speed
Attack	Hardy	Lonely	Adamant	Naughty	Brave
Defense	Bold	Docile	Impish	Lax	Relaxed
Sp. Atk.	Modest	Mild	Bashful	Rash	Quiet
Sp. Def.	Calm	Gentle	Careful	Quirky	Sassy
Speed	Timid	Hasty	Jolly	Naive	Serious

ข.4 ท่าโจมตี (Moves)

นอกจากประเภทแล้ว ท่าโจมตี 1 ท่าจะประกอบด้วย

1. พลังความเสียหายหรือความรุนแรง (Power)
   เป็นค่าที่กำหนดความเสียหายตามสูตรในหัวข้อที่ 2.2
2. ค่าความแม่นยำ (Accuracy)
   โอกาสที่ท่าโจมตีจะโจมตีโดน
3. หมวดหมู่ของท่าโจมตี (Category)
   จะแบ่งเป็น 3 หมวดหมู่คือ
   3.1. Physical จะคำนวณความเสียหายด้วย Attack และ Defense
   3.2. Special จะคำนวณความเสียหายด้วย Special Attack และ Special Defense
   3.3. Status จะเป็นท่าที่ไม่สร้างความเสียหายโดยตรงหรืออาจไม่สร้างความเสียหายเลย
4. จำนวนครั้งที่ใช้ได้ (PP / Power Point)
   เมื่อเหลือ 0 ท่าโจมตีนั้นจะใช้ไม่ได้
5. ลำดับความสำคัญ (Priority)
   ท่าโจมตีที่ลำดับความสำคัญสูงกว่าจะได้โจมตีก่อน โดยไม่คิดความเร็วของโปเกมอน ถ้าหากลำดับความสำคัญเท่ากันจึงจะคิดความเร็วของโปเกมอน

6. เอฟเฟคเพิ่มเติม (Additional effect)
   ผลข้างเคียงจากการใช้ท่าโจมตี ซึ่งอาจจะมีผลลัพธ์หลักจากการใช้ท่าโจมตีหมวดหมู่ Status

ท่าโจมตีมีโอกาสติดคริติคอลได้ มีโอกาสเกิดขึ้น 4.17% โดยความเสียหายจะเพิ่มขึ้นเป็น 1.5 เท่า โปเกมอนแต่ละสปีชีส์จะมีเซ็ตของท่าโจมตีที่เรียนรู้ได้แตกต่างกัน

ข.5 สิ่งของถือ (Held Item)

สิ่งของที่โปเกมอนถือ โดยจะเพิ่มบางอย่างให้กับโปเกมอน สิ่งของบางอย่างจะใช้งานครั้งเดียวแล้วหายไปเลย บางอย่างจะมีผลต่อเนื่องตามเงื่อนไขที่กำหนด

ข.6 ความสามารถของโปเกมอน (Ability)

ความสามารถติดตัวของโปเกมอน บางครั้งอาจจะทำงานอัตโนมัติเมื่อมีเงื่อนไขที่กำหนด

ข.7 สถานะผิดปกติ (Status condition)

โปเกมอนจะสามารถติดสถานะผิดปกติได้ สถานะผิดปกติจะไม่ทำงานหากไม่ใช่โปเกมอนบนสนาม โดยสถานะผิดปกติจะแบ่งเป็น 2 แบบใหญ่ ๆ คือ

1. สถานะผิดปกติถาวร (Non-volatile status)
   จะมีผลตลอดทั้งเกมจนกว่าจะรักษา และโปเกมอนไม่สามารถติดสถานะผิดปกติถาวรมากกว่า 1 สถานะได้ในเวลาเดียวกัน มีทั้งหมด 6 สถานะดังนี้
   1.1. สถานะเผาไหม้ (Burn)
      โปเกมอนจะสูญเสียพลังชีวิต 6.25% ของพลังชีวิตสูงสุดในทุก ๆ เทิร์น และการโจมตีแบบ Physical จะสร้างความเสียหายน้อยลงครึ่งนึง
   1.2. สถานะแช่แข็ง (Freeze)
      โปเกมอนจะไม่สามารถใช้ท่าโจมตีใด ๆ ได้ยกเว้นท่าโจมตีบางท่าจะสามารถละลายน้ำแข็งได้ ทุก ๆ เทิร์นจะมีโอกาสหลุดจากสถานะนี้ 20% และยังอาจจะหลุดจากท่าโจมตีบางท่าของฝ่ายตรงข้ามอีกด้วย
   1.3. สถานะอัมพาต (Paralysis)
      โปเกมอนจะถูกลดความเร็ว 25% และจะมีโอกาส 25% ที่จะใช้ท่าโจมตีไม่ได้
   1.4. สถานะติดพิษ (Poison)
      โปเกมอนจะสูญเสียพลังชีวิต 12.5% ของพลังชีวิตสูงสุดในทุก ๆ เทิร์น

1.5. สถานะติดพิษรุนแรง (Badly poisoned / Toxic)

โดยเริ่มต้นโปเกมอนจะสูญเสียพลังชีวิต 6.25% ของพลังชีวิตสูงสุดในทุก ๆ เทิร์น นอกจากนี้ค่าดังกล่าวจะเพิ่มขึ้นอีก 6.25% ในแต่ละเทิร์น จนกว่าผู้เล่นจะสลับตัวออกจากสนามจึงจะรีเซ็ตค่ากลับเป็น 6.25%

1.6. สถานะหลับ (Asleep)

โปเกมอนจะโจมตีไม่ได้โดยจะสุ่มระหว่าง 1 – 3 เทิร์นจึงจะตื่น

2. สถานะผิดปกติชั่วคราว (Volatile status)

จะมีผลชั่วคราวเพียงแค่ขณะเป็นโปเกมอนบนสนาม หากผู้เล่นทำการสลับโปเกมอน สถานะเหล่านี้ก็จะหายไป เนื่องจากว่ามีหลากหลายรูปแบบมากจึงขอยกตัวอย่างที่จำเป็นแค่บางสถานะเท่านั้นดังนี้

2.1. ห้ามเปลี่ยนตัว (Can’t escape / Trapped)

ผู้เล่นจะไม่สามารถสลับโปเกมอนตัวนี้ได้เป็นระยะเวลาที่กำหนด บางครั้งอาจจะห้ามสลับตัวจนกว่าฝ่ายตรงข้ามจะสลับตัวหรือโปเกมอนบนสนามฝ่ายใดฝ่ายหนึ่งสู้ต่อไม่ได้

2.2. มึนงง (Confusion)

โปเกมอนจะมีโอกาสโจมตีตนเอง 33% แทนที่จะใช้ท่าโจมตี โดยการโจมตีจะนับว่าเป็นการโจมตี Physical ความรุนแรง 40 จะมีโอกาสหายไปเองเมื่อครบ 1 – 4 เทิร์น

2.3. ฟื้นฟูพลัง (Recharge)

โปเกมอนจะไม่สามารถโจมตีได้ 1 เทิร์นซึ่งมักจะเป็นผลของท่าโจมตีของตนเองที่ใช้ก่อนหน้ามักจะมีความรุนแรงสูงมาก

ข.8 กติกาของเกมโปเกมอน

ผู้เล่นจะใช้โปเกมอนในทีมได้สูงสุด 6 ตัว เมื่อเริ่มเกมผู้เล่นจะต้องส่งโปเกมอนออกมา 1 ตัว โดยโปเกมอนตัวนั้นจะถือว่าเป็นโปเกมอนปัจจุบันบนสนาม (Active) ส่วนโปเกมอนที่เหลือก็จะเก็บไว้ก่อน มีเพียงแค่โปเกมอนปัจจุบันเท่านั้นที่จะต่อสู้กันได้

ในแต่ละเทิร์น ผู้เล่นจะต้องทำการเลือกคำสั่งว่าเลือกใช้ท่าโจมตีใดหรือผู้เล่นมีสิทธิ์ที่จะสลับโปเกมอนปัจจุบันของตัวเองกับโปเกมอนตัวอื่นที่เก็บไว้ได้โดยจะถือว่ามีลำดับความสำคัญ +6 ผู้เล่นจะเลือกทำได้เพียงแค่อย่างใดอย่างหนึ่งเท่านั้นหมายความว่าถ้าเลือกที่จะโจมตีก็จะสลับตัวไม่ได้ การเลือกคำสั่งนั้นจะต้องรอให้ผู้เล่นทั้งหมดเลือกคำสั่งก่อนจึงจะแสดงผลการเล่น

โปเกมอนแต่ละตัวจะมีค่าพลังชีวิตอยู่ เมื่อโปเกมอนได้รับความเสียหายโปเกมอนจะสูญเสียพลังชีวิตตามค่าความเสียหายนั้น และหากโปเกมอนตัวนั้นมีพลังชีวิตเหลือ 0 ก็จะไม่สามารถสู้ต่อได้

และจะบังคับให้ผู้เล่นต้องเปลี่ยนเอาโปเกมอนตัวอื่นออกมาแทน หากผู้เล่นไม่มีโปเกมอนที่ยังสามารถต่อสู้ได้ตัวอื่นเหลืออยู่แล้วจะถือว่าแพ้โดยทันที

ถ้าหากโปเกมอนไม่มีท่าโจมตีที่ใช้ได้เหลืออยู่ ก็ยังนับว่าสามารถสู้ต่อได้ แต่ถ้าเลือกคำสั่งใช้ท่าโจมตีจะกลายเป็นท่า Struggle แทน ซึ่งจะเป็นการโจมตีแบบ Physical มีพลังความรุนแรง 50 และยังสร้างความเสียหายให้กับตนเอง 25% ของพลังชีวิตสูงสุดอีกด้วย

โปเกมอนที่มี Mega Stone เฉพาะตัวสามารถทำการ Mega Evolve ได้โดยจะแปลงร่างเป็นโปเกมอนที่แข็งแกร่งขึ้นและจะอยู่จนกว่าจะจบเกม แต่ใน 1 เกม ผู้เล่นจะใช้ Mega Evolution ได้แค่เพียงครั้งเดียว

ประวัติผู้เขียน

นาย กวิน ไวทยวงศ์สกุล
เกิดวันที่ 25 พฤศจิกายน 2541 จังหวัดภูเก็ต
จบมัธยมศึกษาตอนปลายที่โรงเรียนภูเก็ตวิทยาลัย
กำลังศึกษาปริญญาตรีที่ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย