



# การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ การสร้างปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรม  
Creating Artificial Intelligence for Abstract Strategy Board Game

ชื่อนิสิต นายพีรพล อยู่ไพศาล 603 36437 23

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์  
สาขาวิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2563

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

การสร้างปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรม

นายพีรพล อยู่ไพศาล

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต  
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2563

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

# Creating Artificial Intelligence for Abstract Strategy Board Game

Peeraphol Youpaisarn

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

หัวข้อโครงการ	การสร้างปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรม
โดย	นายพีรพล อยู่ไพศาล
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงการหลัก	รองศาสตราจารย์ ดร.ศุภกานต์ พิมลธเรศ
อาจารย์ที่ปรึกษาโครงการร่วม	ผู้ช่วยศาสตราจารย์ ศศิภา พันธุ์ดีธธร

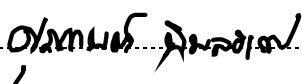
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้รับโครงการฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงการวิทยาศาสตร์ (Senior Project)



.....  
(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี)

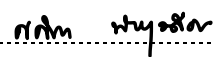
หัวหน้าภาควิชาคณิตศาสตร์  
และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงการ



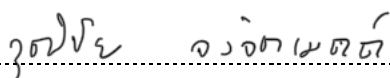
.....  
(รองศาสตราจารย์ ดร.ศุภกานต์ พิมลธเรศ)

อาจารย์ที่ปรึกษาโครงการหลัก



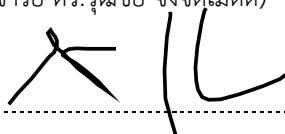
.....  
(ผู้ช่วยศาสตราจารย์ ศศิภา พันธุ์ดีธธร)

อาจารย์ที่ปรึกษาโครงการร่วม



.....  
(อาจารย์ ดร.วุฒิชัย จงจิตเมตต์)

กรรมการ



.....  
(อาจารย์ โชติรส สุรพลชัย)

กรรมการ

นายพีรพล อยู่ไพศาล: การสร้างปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรม. (Creating Artificial Intelligence for Abstract Strategy Board Game) อาจารย์ที่ปรึกษาโครงการหลัก: รองศาสตราจารย์ ดร. ศุภกานต์ พิมลธเรศ, อาจารย์ที่ปรึกษาโครงการร่วม: ผู้ช่วยศาสตราจารย์ ศศิภา พันธุ์วดีธร, 40 หน้า

โครงการนี้มีจุดประสงค์เพื่อพัฒนาปัญญาประดิษฐ์ที่ระดับความยากปรับตัวได้ด้วยตัวเองสำหรับเกมกระดานเชิงกลยุทธ์ เนื่องจากเกมที่ยากเกินไปทำให้ผู้เล่นเกิดความเครียดได้ ในทางกลับกันหากสร้างเกมที่มีความง่ายเกินไปทำให้ผู้เล่นรู้สึกเบื่อหน่าย ปัญญาประดิษฐ์ที่ปรับตัวได้สองแบบ ได้แก่ ปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ และปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกได้ถูกนำเสนอในการศึกษานี้ นอกจากนี้โครงการนี้ยังเสนอตัวชี้วัดที่ใช้วัดความใกล้เคียงของระดับความยากระหว่างผู้เล่นสองคนโดยคำนวณจากค่าอัตราส่วนของผลต่างระหว่างจำนวนหมากของปัญญาประดิษฐ์ที่มีระดับความยากที่เท่ากันต่อจำนวนหมากทั้งหมดบนกระดานที่เวลาหนึ่ง หลังจากนั้นแล้วปัญญาประดิษฐ์ที่ปรับตัวได้ทั้งสองถูกนำมาเปรียบเทียบโดยทดลองกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่เพื่อเลือกตัวแบบที่เหมาะสมสำหรับการทดสอบกับมนุษย์ ในท้ายที่สุดปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับได้ถูกเลือกและผลการทดลองพบว่าปัญญาประดิษฐ์สามารถควบคุมเกมได้ใน 30 ตาเดินหลังจากสถานะเริ่มต้นของเกม อย่างไรก็ตามผลลัพธ์โดยรวมมีค่าอยู่ในเกณฑ์ที่เทียบเท่ากับการนำปัญญาประดิษฐ์ที่มีระดับความยากใกล้เคียงกันมาทดสอบ

ภาควิชา.....คณิตศาสตร์และวิทยาการคอมพิวเตอร์.....ลายมือชื่อนิสิต..... *พีรพล อยู่ไพศาล*

สาขาวิชา.....วิทยาการคอมพิวเตอร์.....ลายมือชื่อ อ.ที่ปรึกษาโครงการหลัก..... *อ.ศุภกานต์ พิมลธเรศ*

ปีการศึกษา .....2563.....ลายมือชื่อ อ.ที่ปรึกษาโครงการร่วม..... *อ.ศศิภา พันธุ์วดีธร*

603364372323: MAJOR COMPUTER SCIENCE

KEYWORDS: MINIMAX/ ARTIFICIAL INTELLIGENCE

PEERAPHOL YOUPAISARN: CREATING ARTIFICIAL INTELLIGENCE FOR ABSTRACT STRATEGY BOARD GAME. ADVISOR: ASSO. PROF.SUPHAKANT PHIMOLTARES, PH.D., CO-ADVISOR: ASST. PROF. SASIPA PANTHUWADEETHORN, 40 pages

The purpose of this project is to develop artificial intelligence with self-adjustable difficulty level for abstract strategy board game since too hard game can make a player feel stress whereas too easy game can make player feel boredom. Two types of adaptive artificial intelligence, which are rank-based adaptive artificial intelligence and depth-based adaptive artificial intelligence were proposed in this study. Additionally, this project proposed the indicator that can measure similarity of difficulty levels between two players by calculating the ratio of the difference between the number of pieces of both artificial intelligences with the same difficulty level to the total number of pieces on the board at a time. Subsequently, both adaptive artificial intelligences were compared by testing with the artificial intelligence with static difficulty level in order to select the appropriate model for testing with human. Finally, the rank-based adaptive artificial intelligence was chosen and the result found that the artificial intelligence can control the game in 30 moves after the beginning state of the game. However, the overall result is equivalent to the result from competition of two artificial intelligences with the same difficulty level.

Department: Mathematics and Computer Science.....

Student's Signature

Field of Study: Computer Science.....

Advisor's Signature

Academic Year: 2020.....

Co-advisor's Signature

สารบัญ

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
สารบัญตาราง.....	ซ
สารบัญภาพ.....	ฅ
กิตติกรรมประกาศ.....	ญ
บทที่ 1 บทนำ.....	1
1.1 หลักการและเหตุผล.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินงาน.....	2
ก. แผนการศึกษา.....	2
ข. ตารางการดำเนินงาน.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
ก. ประโยชน์ด้านความรู้และประสบการณ์ก่อนนิต.....	3
ข. ประโยชน์ที่ได้จากโครงการที่พัฒนาขึ้น.....	3
1.6 โครงสร้างรายงาน.....	3
บทที่ 2.....	4
2.1 หลักการและทฤษฎีที่เกี่ยวข้อง.....	4
2.1.1 โอเทลโล.....	4
2.1.2 ขั้นตอนวิธี Minimax.....	5
2.2 งานวิจัยที่เกี่ยวข้อง.....	8
2.2.1 การพัฒนาปัญญาประดิษฐ์รูปแบบต่าง ๆ สำหรับโอเทลโล.....	8

2.2.2 การปรับความยากแบบพลวัตในเกมคอมพิวเตอร์ (Dynamic Difficulty Adjustment in Computer Game).....	10
บทที่ 3.....	13
3.1 ปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ.....	14
3.2 ปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึก.....	17
3.3 การทดลองเพื่อหาตัวชี้วัดความใกล้เคียงของระดับความยาก.....	18
บทที่ 4.....	19
4.1 สมมติฐานของการวิจัย.....	19
4.2 ผลการวิจัย.....	21
4.2.1 การทดลองกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่.....	21
4.2.2 การทดลองกับมนุษย์.....	22
4.3 การอภิปรายผล.....	25
บทที่ 5.....	26
5.1 ข้อเสนอ.....	26
5.2 ข้อเสนอแนะ.....	27
รายการอ้างอิง.....	28
ภาคผนวก ก.....	29
ประวัติผู้เขียน.....	34



## สารบัญตาราง

ตารางที่ 2.1 ผลลัพธ์ของการเล่นโอเทลโลระหว่าง Minimax และ Monte Carlo Tree Search.....	9
ตารางที่ 2.2 ผลลัพธ์ของการเล่นโอเทลโลระหว่างมนุษย์กับ Minimax.....	9
ตารางที่ 2.3 ผลลัพธ์ของการเล่นโอเทลโลระหว่างมนุษย์ กับ Monte Carlo Tree Search .....	10
ตารางที่ 3.1 ตารางแสดงเงื่อนไขของการเลือกตาเดินของปัญญาประดิษฐ์.....	17
ตารางที่ 3.2 ตารางแสดงเงื่อนไขการปรับความลึกของปัญญาประดิษฐ์ .....	18

## สารบัญภาพ

ภาพที่ 2.1 กระดานตอนเริ่มเกม .....	4
ภาพที่ 2.2 เกมจบโดยที่ไม่สามารถวางหมากเพิ่มได้.....	5
ภาพที่ 2.3 เกมจบโดยที่หมากเต็มกระดาน .....	5
ภาพที่ 2.4 ภาพแสดงระดับความสนุกของตัวแทนสองตัวแทน .....	11
ภาพที่ 2.5 ภาพแสดงระดับความสนุกของผู้เข้าร่วมสี่คน .....	11
ภาพที่ 3.1 เหตุการณ์ตัวอย่างประกอบการคำนวณคะแนน .....	14
ภาพที่ 3.2 โครงสร้างของปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ .....	15
ภาพที่ 3.3 ภาพแสดงโครงสร้างของปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึก .....	17
ภาพที่ 4.1 ผลการทดลองระหว่างปัญญาประดิษฐ์ที่มีระดับความยากคงที่ (Static difficulty AI) ซึ่งมีระดับความยากเท่ากัน .....	19
ภาพที่ 4.2 ผลลัพธ์ของการทดสอบระหว่างปัญญาประดิษฐ์ที่มีระดับความยากคงที่ (Static difficulty AI) กับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ซึ่งเพิ่มระดับความยากขึ้น 1 ระดับ.....	20
ภาพที่ 4.3 ผลลัพธ์ของการทดสอบปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ทั้ง 6 ระดับ .....	21
ภาพที่ 4.4 ผลลัพธ์ของการทดสอบปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ทั้ง 6 ระดับ .....	22
ภาพที่ 4.5 ผลการทดสอบปัญญาประดิษฐ์กับมนุษย์คนที่หนึ่ง.....	23
ภาพที่ 4.6 ผลการทดสอบปัญญาประดิษฐ์กับมนุษย์คนที่สอง .....	23
ภาพที่ 4.7 ผลการทดสอบปัญญาประดิษฐ์กับมนุษย์คนที่สาม .....	24
ภาพที่ 4.8 การเปรียบเทียบค่าความได้เปรียบเฉลี่ยของปัญญาประดิษฐ์จากการเล่นกับมนุษย์ 3 คน.....	24
ภาพที่ 5.1 ภาพประกอบการคำนวณจำนวนตาเดินก่อนถึงช่วงมุมกระดาน .....	27

## กิตติกรรมประกาศ

ในงานวิจัย “การสร้างปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรม” นี้ได้รับการสนับสนุนและความช่วยเหลืออย่างเต็มที่จาก รองศาสตราจารย์ ดร.ศุภกานต์ พิมลธเรศ อาจารย์ที่ปรึกษาโครงการหลัก และผู้ช่วยศาสตราจารย์ ศศิภา พันธุ์ดิษฐ์ อาจารย์ที่ปรึกษาโครงการร่วม ในการเอาใจใส่ให้คำปรึกษา ชี้แนะแนวทางอันเป็นประโยชน์ในงานวิจัย ตรวจสอบแก้ไขข้อผิดพลาด รวมถึงคอยให้กำลังใจตั้งแต่เริ่มดำเนินงานวิจัย จนกระทั่งสำเร็จไปได้ด้วยดี

ขอขอบพระคุณอาจารย์ ดร.วุฒิชัย จงจิตเมตต์ และอาจารย์ โชติรส สุรพลชัย กรรมการสอบโครงการ ซึ่งได้ช่วยชี้แนะให้โครงการมีความสมบูรณ์มากขึ้น

ขอขอบพระคุณอาจารย์ท่านอื่นที่มีได้กล่าวนามไว้ ณ ที่นี้ที่ได้ถ่ายทอดความรู้ให้ผู้วิจัยได้มี ความรู้ และความเข้าใจในทฤษฎีต่าง ๆ อันเป็นประโยชน์ในการดำเนินงานวิจัยนี้

ขอขอบพระคุณภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ที่ได้จัดสถานที่ให้นิสิตในการดำเนินงานวิจัย รวมถึงงบประมาณค่าใช้จ่ายในการดำเนินงานวิจัยนี้

ขอขอบพระคุณบิดา มารดา และญาติมิตรทุกท่านที่ได้สนับสนุน ให้คำปรึกษา และให้กำลังใจเสมอตลอดการดำเนินงานวิจัยนี้

# บทที่ 1

## บทนำ

### 1.1 หลักการและเหตุผล

เกมกระดานเชิงกลยุทธ์หรือหมากกระดานนั้นเป็นเกมประเภทหนึ่งที่อยู่คู่กับมนุษย์มายาวนาน และมีความแตกต่างกันในแต่ละประเทศ อีกทั้งยังมีส่วนช่วยในการพัฒนาทักษะในด้านต่าง ๆ ของสมอง เช่น การวางแผน การวิเคราะห์สถานการณ์ ความรอบคอบ ซึ่งในปัจจุบันมีการพัฒนาปัญญาประดิษฐ์ที่มุ่งเน้นไปที่ความเก่งของตัวปัญญาประดิษฐ์เพื่อนำมาเล่นหมากกระดานแข่งกับมนุษย์อย่างมากมาย เช่น AlphaGo[1] แต่เนื่องจากหมากกระดานเป็นเกมประเภทหนึ่งที่ต้องอาศัยการคิดวิเคราะห์และการวางแผนเป็นอย่างมาก ดังนั้นการที่ผู้เล่นจะพัฒนาตนเองไปได้จึงต้องมาจากการเล่นซ้ำ ๆ ซึ่งถ้าหากเล่นกับผู้เล่นที่เก่งกว่ามาก ๆ นั้นจะส่งผลให้เกิดความเครียด และหากเล่นกับผู้เล่นที่มีมือด้อยกว่า จะทำให้เกิดความเบื่อหน่ายได้ตามแนวคิด flow channel ของ Csikszentmihalyi[2] ดังนั้นผู้พัฒนาจึงต้องการพัฒนาปัญญาประดิษฐ์ที่สามารถปรับระดับความยากให้เหมาะสมกับผู้เล่นแต่ละคนได้อย่างอัตโนมัติ

ในการศึกษาเกี่ยวกับการปรับระดับความยากแบบพลวัต (Dynamic Difficulty Adjustment) จากบทความปริทัศน์ของ Mohammad Zohaib[3] ได้รวบรวมวิธีการพัฒนาปัญญาประดิษฐ์ที่สามารถปรับระดับความยากแบบพลวัตได้ในรูปแบบต่าง ๆ สำหรับเกมหลากหลายประเภท ซึ่งเกมแต่ละประเภทนั้นมีแนวคิดที่เหมือนกันคือ การรับข้อมูลต่าง ๆ ที่ผู้เล่นเป็นคนป้อนให้ในเกมแล้วนำมาวิเคราะห์ระดับทักษะของผู้เล่น จากนั้นจึงปรับพฤติกรรมของตนเองให้เหมาะสมกับผู้เล่นที่เผชิญหน้าอยู่ ณ ขณะนั้น ในกระบวนการปรับระดับความยากของเกม ระบบจะต้องปรับความยากของเกมด้วยความต่อเนื่องและนุ่มนวลที่สุด เพื่อไม่ให้ผู้เล่นทราบดีว่าเกมที่กำลังเล่นอยู่นั้นได้ถูกปรับระดับความยากให้เหมาะสม เช่น ระบบพิจารณาสถานะของผู้เล่นเพื่อนำไปสร้างความสามารถของศัตรูในเกมยิงมุมมองบุคคลที่หนึ่ง (First Person Shooter)[4] กล่าวคือ หากผู้เล่นนั้นมีค่าพลังชีวิตต่ำ หรือมีทรัพยากรที่จะต้องใช้ในการฝ่าอุปสรรคอยู่น้อย ระบบจะช่วยเหลือผู้เล่นให้สามารถผ่านอุปสรรคไปได้ง่ายขึ้นในทางกลับกัน หากผู้เล่นมีทรัพยากรมาก ระบบก็จะออกแบบให้อุปสรรคมีความยากและท้าทายมากกว่าปกติ เพื่อเพิ่มอรรถรสในการเล่น ซึ่งผลลัพธ์ที่ได้จากการสร้างระบบการปรับความยากแบบพลวัตพบว่า ปัญญาประดิษฐ์นั้นสามารถปรับระดับความยากได้ด้วยตนเองและผู้เล่นมีส่วนร่วมกับเกมที่มีการปรับความยากแบบพลวัตมากกว่าเกมที่ออกแบบระดับความยากแบบปกติ โดยพิจารณาผ่านตัวชี้วัดต่าง ๆ เช่น จำนวนการเล่นเกมต่อวัน และอัตราการแพ้ชนะของปัญญาประดิษฐ์

ในการศึกษาวิธีการพัฒนาปัญญาประดิษฐ์สำหรับหมากกระดานนั้น โครงการงานของ Owen Shevlin[5] ได้มีการพัฒนาและทดสอบประสิทธิภาพของปัญญาประดิษฐ์หลากหลายประเภทสำหรับเกมโอเทลโล ได้แก่ Minimax, Monte-Carlo Tree Search และ Deep Learning ซึ่งการพัฒนาปัญญาประดิษฐ์แต่ละประเภทจะส่งผลให้ตัว

ปัญญาประดิษฐ์มีความสามารถที่ไม่เท่ากัน แต่ผู้พัฒนาได้สังเกตเห็นว่า เมื่อนำปัญญาประดิษฐ์ทั้งหมดมาทดลองเล่นแข่งกับมนุษย์พบว่า อัตราการชนะมีค่าที่ต่างกันไม่มาก ผู้พัฒนาจึงเลือกวิธีการพัฒนาที่ง่ายต่อการนำมาดัดแปลงเพื่อสร้างปัญญาประดิษฐ์ที่สามารถปรับตัวได้ โดยจะนำวิธีการ Minimax[6] มาดัดแปลง เพราะ Minimax คือการสร้างต้นไม้ที่มีค่าผลรวมสูงสุดขั้นต่ำ เนื่องจากโครงสร้างของหมากกระดานนั้นสามารถมองให้อยู่ในรูปของต้นไม้ได้ โดยที่ต้นไม้หนึ่งต้นจะมีโหนดเก็บค่าความได้เปรียบของผู้เล่นหนึ่งคนที่สนใจ กล่าวคือการเลือกตาเดิน (move) ที่ดีที่สุดจะเปรียบเสมือนการค้นหาโหนดที่มีค่าความได้เปรียบมากที่สุด นอกจากนี้ในการเล่นหมากกระดานผู้เล่นทั้งสองฝ่ายจะต้องผลัดกันเล่นคนละหนึ่งตาเดิน ดังนั้นตาเดินที่ดีที่สุดของผู้เล่นฝั่งตรงข้ามก็คือการค้นหาโหนดที่มีค่าความได้เปรียบของเราน้อยที่สุด

จากที่กล่าวมาข้างต้น ทำให้ผู้พัฒนาที่มีความสนใจที่จะศึกษาและพัฒนาปัญญาประดิษฐ์ที่สามารถปรับตัวให้เหมาะสมต่อผู้เล่นทุกระดับ เพื่อให้ผู้เล่นได้สนุกกับการเล่นเกมหมากกระดานมากขึ้น อีกทั้งยังสามารถนำปัญญาประดิษฐ์นี้ไปเป็นคู่มือเพื่อพัฒนาฝีมือและทักษะทางกระบวนการคิดได้

## 1.2 วัตถุประสงค์

เพื่อพัฒนาปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรมที่สามารถปรับตัวให้เหมาะสมต่อผู้เล่นทุกระดับ

## 1.3 ขอบเขตของโครงการ

- 1.3.1 โปรแกรมสามารถทำงานได้ในระบบปฏิบัติการ Windows เท่านั้น
- 1.3.2 เลือกพัฒนาปัญญาประดิษฐ์ที่เล่นเฉพาะเกมโอเทลโลเท่านั้น
- 1.3.3 เกมโอเทลโลจะจบลงด้วยผลลัพธ์ที่จำนวนหมากขาวกับจำนวนหมากดำมีค่าต่างกันไม่เกิน 38 ตัว
- 1.3.4 ปัญญาประดิษฐ์จะต้องเป็นฝ่ายที่เดินทีหลังเสมอ

## 1.4 วิธีการดำเนินงาน

### ก. แผนการศึกษา

1. ศึกษาวิธีการปรับความยากแบบพลวัต
2. ศึกษางานวิจัยที่เกี่ยวข้องกับการพัฒนาปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์
3. ออกแบบและพัฒนาปัญญาประดิษฐ์ที่สามารถปรับตัวได้
4. ทดสอบประสิทธิภาพปัญญาประดิษฐ์ที่สามารถปรับตัวได้
5. วิเคราะห์และอภิปรายผล
6. จัดทำเอกสาร

## ข. ตารางการดำเนินงาน

ขั้นตอน	ปีการศึกษา 2563								
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. ศึกษาวิธีการปรับความยากแบบ พลวัต									
2. ศึกษางานวิจัยที่เกี่ยวข้องกับการ พัฒนาปัญญาประดิษฐ์สำหรับเกม กระดานเชิงกลยุทธ์									
3. ออกแบบและพัฒนา ปัญญาประดิษฐ์ที่สามารถปรับตัวได้									
4. ทดสอบประสิทธิภาพ ปัญญาประดิษฐ์ที่สามารถปรับตัวได้									
5. วิเคราะห์และอภิปรายผล									
6. จัดทำเอกสาร									

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

#### ก. ประโยชน์ด้านความรู้และประสบการณ์ต่อนิสิต

1. ได้รับความรู้ ความเข้าใจเกี่ยวกับการพัฒนาปัญญาประดิษฐ์
2. ได้ฝึกการคิดวิเคราะห์ การทำงานอย่างเป็นระบบ
3. ได้รับความรู้เกี่ยวกับขั้นตอนวิธีที่ใช้กับเกมกระดานเชิงกลยุทธ์

#### ข. ประโยชน์ที่ได้จากโครงการที่พัฒนาขึ้น

1. ได้เกมกระดานที่มีปัญญาประดิษฐ์ที่มีความสามารถเหมาะสมกับผู้เล่น

### 1.6 โครงสร้างรายงาน

บทที่ 2 กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้อง

บทที่ 3 กล่าวถึงวิธีการวิจัยในการพัฒนาปัญญาประดิษฐ์ทั้งสองชุด ได้แก่ ปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ และ ปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึก

บทที่ 4 กล่าวถึงกระบวนการทดลอง และผลของการดำเนินการวิจัยของการทดสอบปัญญาประดิษฐ์ที่สามารถปรับตัวได้ รวมถึงการอภิปรายผลการทดลอง

บทที่ 5 กล่าวถึงการสรุปผลการวิจัยในการพัฒนาปัญญาประดิษฐ์ที่สามารถปรับตัวได้

## บทที่ 2

### งานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีรวมถึงงานวิจัยที่เกี่ยวข้องกับการพัฒนาปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์ที่สามารถปรับตัวได้

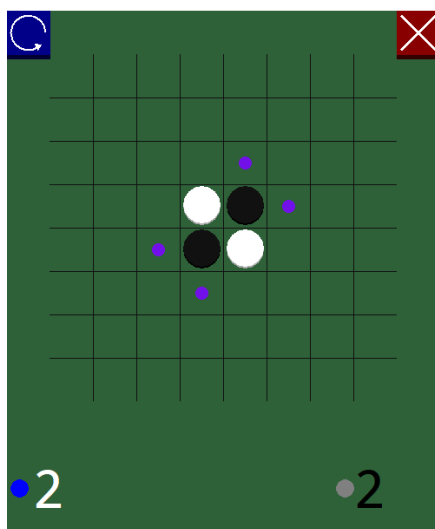
#### 2.1 หลักการและทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 โอเทลโล

โอเทลโลคือเกมกระดานที่ถูกคิดค้นในช่วงศตวรรษที่ 19 โดยมีแนวคิดจากการวางหมากสีให้เกิดการพลิกกันบนกระดานขนาด  $8 \times 8$  ช่อง ซึ่งวิธีการพลิกคือการวางหมากสีของผู้เล่นให้ไปครอบหมากสีของผู้เล่นฝั่งตรงข้ามตามแนวเส้นตรง (แนวตั้ง แนวนอน แนวทแยง) เมื่อวางครบแล้วจะทำให้หมากสีของผู้เล่นฝั่งตรงข้ามที่ถูกครอบทุกชิ้นซึ่งอยู่ระหว่างหมากตัวเดิมของผู้เล่นกับหมากที่ถูกวางเพิ่มเกิดการพลิกสีเป็นหมากของผู้เล่น

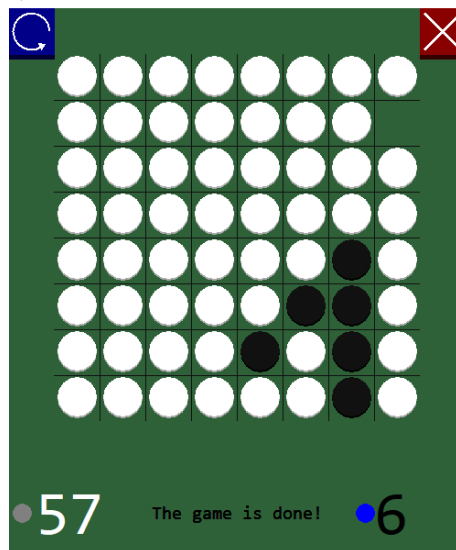
โอเทลโลจะเริ่มโดยการวางหมากสีขาวและหมากสีดำอย่างละ 2 ชิ้นไว้ที่ 4 ช่องตรงกลางซึ่งจะวางหมากสีเดียวกันตามแนวทแยงมุม จากนั้นจึงเริ่มเล่นโดยฝ่ายสีขาวจะเป็นฝ่ายที่เริ่มก่อนเสมอ ดังภาพที่

2.1

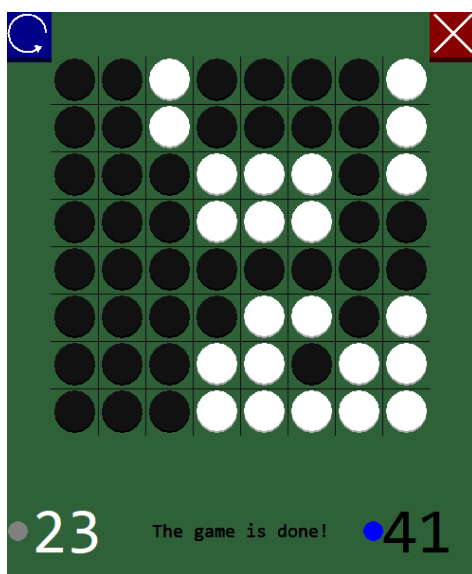


ภาพที่ 2.1 กระดานตอนเริ่มเกม

เมื่อเกิดเหตุการณ์ที่ไม่สามารถวางหมากให้เกิดการพลิกสีได้ ผู้เล่นจำเป็นจะต้องผ่านตาเล่นนั้นให้ผู้เล่นฝ่ายตรงข้ามเป็นผู้วางหมากแทน โดยเกมจะจบก็ต่อเมื่อไม่มีผู้เล่นสามารถวางได้ หรือ ตัวหมากเต็มกระดาน



ภาพที่ 2.2 เกมจบโดยที่ไม่สามารถวางหมากเพิ่มได้



ภาพที่ 2.3 เกมจบโดยที่หมากเต็มกระดาน

### 2.1.2 ขั้นตอนวิธี Minimax

เนื่องจากโอเทลโล่นั้นเป็นเกมกระดาน จึงสามารถแปลงสภาพให้อยู่ในรูปของต้นไม้ตัดสินใจได้ ซึ่งทำให้ปัญญาประดิษฐ์สามารถคำนวณคะแนนความได้เปรียบของทางเลือกในทุก ๆ ตาเดินได้



ขั้นตอนวิธี Minimax เป็นวิธีการที่นำมาใช้คำนวณคะแนนดังกล่าวของต้นไม้ตัดสินใจให้ได้ผลลัพธ์ที่ดีที่สุดตามความลึกของต้นไม้ที่ต้องการพิจารณา โดยการใช้ maximizer สลับกับ minimizer ผ่านการตั้งสมมติฐานว่าผู้เล่นทุกฝ่ายจะเลือกทางเลือกที่ดีที่สุด เช่น กำหนดให้ Minimax สนใจผู้เล่นสีขาว และสามารถคิดลึกลงไปได้ถึง 4 ชั้น Minimax จะกำหนดให้ตาเดินของสีขาวใช้ maximizer และตาเดินของสีดำใช้ minimizer เพราะทางเลือกที่ดีที่สุดของสีดำคือทางเลือกที่ทำให้สีขาวมีคะแนนน้อยที่สุดและทางเลือกที่ดีที่สุดของสีขาวคือทางเลือกที่ทำให้สีขาวมีคะแนนมากที่สุด จึงกำหนดวิธีการประเมินต้นไม้ตัดสินใจในชั้นที่ 1 - 4 ได้เป็น maximizer minimizer maximizer minimizer ตามลำดับ จากนั้นจะเริ่มคำนวณคะแนนย้อนกลับขึ้นไปจากชั้นที่ 4 ไปสู่ชั้นที่ 1 ซึ่งมี pseudocode ดังนี้

```
function minimax(node, depth, maximizingPlayer)
    if depth = 0 or node is a terminal node
        return the utility of the node
    if maximizingPlayer
        bestValue := -infinity
        for each child of node
            v := minimax(child, depth - 1, FALSE)
            bestValue := max(bestValue, v)
        return bestValue
    else (* minimizing player *)
        bestValue := +infinity
        for each child of node
            v := minimax(child, depth - 1, TRUE)
            bestValue := min(bestValue, v)
        return bestValue
```

จาก pseudocode ข้างต้นจะพบว่าขั้นตอนวิธีนี้จะคำนวณค่าของโหนดทุกโหนดที่ความลึกมากที่สุดที่กำหนดไว้ ดังนั้นขั้นตอนวิธีนี้จะมีประสิทธิภาพเชิงเวลาอยู่ที่  $O(b^d)$  โดยที่  $b$  คือจำนวนโหนดลูกที่มากที่สุดของโหนดหนึ่งโหนด (branch factor) และ  $d$  คือค่าความลึกในการพิจารณา (depth) แต่ขั้นตอนวิธีนี้สามารถปรับปรุงให้มีประสิทธิภาพเชิงเวลาที่ดีขึ้นด้วยการใช้แนวคิด Alpha-Beta pruning

Alpha-Beta pruning คือการเพิ่มตัวแปรสองตัวเพื่อช่วยในการลดจำนวนการท่องต้นไม้ของขั้นตอนวิธี minimax ลง โดยให้ชื่อตัวแปรทั้งสองว่า Alpha และ Beta ซึ่ง Alpha จะเก็บค่าสูงสุดของโหนดลูกในชั้น maximizer และ Beta จะเก็บค่าต่ำสุดของโหนดลูกในชั้น minimizer ซึ่งจะแสดงผ่าน pseudocode ดังนี้

```
function alphaBeta(node, alpha, beta, depth)
    if depth = 0 or node is a leaf
        return the utility value of node
    if node is a minimizing node
        for each child of node
            beta = min (beta, alphaBeta(child, alpha, beta, depth - 1))
            if beta <= alpha
                return beta
        return beta
    if node is a maximizing node
        for each child of node
            alpha = max (alpha, alphaBeta(child, alpha, beta, depth - 1))
            if alpha >= beta
                return alpha
        return alpha
```

จาก pseudocode ข้างต้นจะมีการเพิ่มเงื่อนไขเพื่อจบการทำงานแบบวนซ้ำนั่นคือเงื่อนไข  $\beta \leq \alpha$  ซึ่งเงื่อนไขดังกล่าวจะมีความหมายที่แตกต่างกันในชั้น maximizer และ minimizer เนื่องจากขั้นตอนวิธีนี้ใช้การค้นหาตามแนวลึกและค่า  $\alpha/\beta$  จะถูกส่งย้อนกลับจากโหนดลูกขึ้นไปโหนดแม่ โดยการทำงานในชั้น maximizer เมื่อเกิดเงื่อนไข  $\alpha \geq \beta$  จะแสดงว่าค่าที่ชั้น maximizer เลือกจะมีค่ามากกว่าค่าที่มากที่สุดของโหนดทางซ้ายในระดับความลึกเดียวกันอย่างแน่นอน ซึ่งทำให้โหนดแม่ที่เป็นโหนดชั้น minimizer จะไม่มีทางเลือกโหนดลูกตัวนี้เป็นทางเดินที่ดีที่สุด ในทางกลับกันชั้น minimizer เมื่อเกิดเงื่อนไข  $\beta \leq \alpha$  จะแสดงว่าค่าที่ชั้น minimizer เลือกจะเป็นค่าที่น้อยกว่าค่าที่น้อยที่สุดของโหนดทางซ้ายในระดับความลึกเดียวกันอย่างแน่นอน ดังนั้นโหนดแม่ที่เป็นโหนดชั้น maximizer จะไม่มีทางเลือกโหนดลูกตัวนี้เป็นทางเดินที่ดีที่สุด

จากการปรับปรุงขั้นตอนวิธีข้างต้นทำให้สามารถลดการคำนวณลงได้โดยจะทำให้ประสิทธิภาพเชิงเวลาเปลี่ยนแปลงไปตามการเรียงกันของข้อมูลในต้นไม้ โดยจะแบ่งออกเป็นสองกรณีคือ

กรณีที่ดีที่สุดจะมีประสิทธิภาพเชิงเวลาเท่ากับ  $O(b^{\frac{d}{2}})$  เพราะสามารถลดภาระการคำนวณทั้งหมดลงได้ครึ่งหนึ่งของข้อมูลทั้งหมด แต่กรณีที่แย่ที่สุดจะไม่สามารถลดภาระการคำนวณได้ซึ่งมีประสิทธิภาพเชิงเวลาอยู่ที่  $O(b^d)$

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 การพัฒนาปัญญาประดิษฐ์รูปแบบต่าง ๆ สำหรับโอเทลโล

จากการศึกษาโครงงาน “Creating an AI to play Othello” ของ Owen Shevlin ทำให้เห็นวิธีการพัฒนาปัญญาประดิษฐ์สำหรับโอเทลโลในรูปแบบต่าง ๆ มากมาย ได้แก่ Minimax, Monte Carlo Tree Search และ Artificial Neural Network โดยแต่ละวิธีจะมีประสิทธิภาพที่ต่างกัน อีกทั้งยังมีการนำปัญญาประดิษฐ์แต่ละประเภทมาทดลองเล่นกับปัญญาประดิษฐ์ด้วยกัน และทดลองเล่นกับมนุษย์

ตารางที่ 2.1 ผลลัพธ์ของการเล่นโอเทลโลระหว่าง Minimax และ Monte Carlo Tree Search

Minimax vs Monte Carlo Tree Search				Player 1	Player 2
Decider				FixedMinimax	MCTS
Evaluator				Positional	Positional
Victories				3	17
Win Percent				15%	85%
Total Games	20	Draws	0		

จากตารางที่ 2.1 เมื่อนำปัญญาประดิษฐ์ที่ถูกพัฒนาโดย Minimax มาเล่นกับ Monte Carlo Tree Search เป็นจำนวน 20 เกม ผลที่ได้คือ Monte Carlo Tree Search ชนะทั้งหมด 17 เกม แสดงถึงความสามารถของ Monte Carlo Tree Search ที่สามารถเลือกตาเดินที่มีโอกาสชนะมากกว่า Minimax

ตารางที่ 2.2 ผลลัพธ์ของการเล่นโอเทลโลระหว่างมนุษย์กับ Minimax

Human vs Minimax				Player 1	Player 2
Decider				Human	FixedMinimax
Evaluator				-	Positional
Victories				4	20
Win Percent				16.7%	83.3%
Total Games	24	Draws	0		

### ตารางที่ 2.3 ผลลัพธ์ของการเล่นโอเทลระหว่างมนุษย์ กับ Monte Carlo Tree Search

Human vs Monte Carlo Tree Search				Player 1	Player 2
Decider				Human	MCTS
Evaluator				-	Positional
Victories				3	21
Win Percent				12.5%	87.5%
Total Games	24	Draws	0		

เมื่อพิจารณาผลการทดลองจากตารางที่ 2.2 และตารางที่ 2.3 จะเห็นว่าในการนำปัญญาประดิษฐ์ที่ถูกพัฒนาโดย Minimax และ Monte Carlo Tree Search มาทดสอบกับมนุษย์ทั้งหมด 24 เกม ผลที่ได้คือปัญญาประดิษฐ์ชนะจำนวน 20 เกม และ 21 เกม ตามลำดับ กล่าวคือถึงแม้ว่า Monte Carlo Tree Search จะมีประสิทธิภาพดีกว่า Minimax แต่ในการทดสอบกับมนุษย์นั้นให้ผลที่ไม่ต่างกันมาก จึงทำให้สามารถเลือกพัฒนาปัญญาประดิษฐ์เพื่อนำไปเล่นกับมนุษย์ด้วยวิธีใดก็ได้

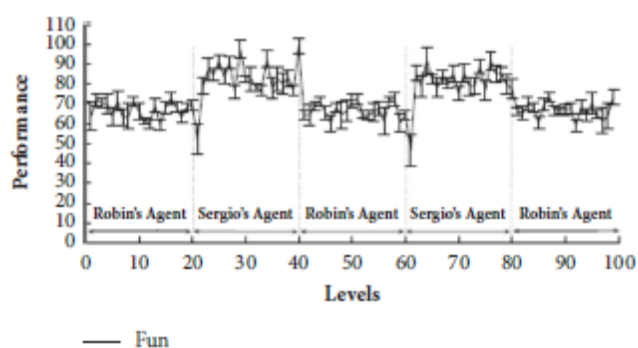
#### 2.2.2 การปรับความยากแบบพลวัตในเกมคอมพิวเตอร์ (Dynamic Difficulty Adjustment in Computer Game)

จากการศึกษาบทความปริทัศน์เกี่ยวกับการปรับความยากแบบพลวัตในเกมคอมพิวเตอร์ (Dynamic Difficulty Adjustment in Computer Game) [3] พบว่าการปรับความยากแบบพลวัต หรือ Dynamic Difficulty Adjustment (DDA) คือวิธีการปรับเปลี่ยนเหตุการณ์ พฤติกรรม และพีเจอร์ภายในเกมให้เหมาะสมกับทักษะของผู้เล่นอย่างอัตโนมัติ เนื่องจากถ้าเกมที่เล่นนั้นง่ายเกินไปจะทำให้ผู้เล่นรู้สึกเบื่อ แต่ถ้ามีความยากมากเกินไปจะทำให้ผู้เล่นรู้สึกหงุดหงิดได้ ซึ่ง DDA จะมีส่วนช่วยให้ผู้เล่นสามารถเล่นเกมได้จนจบ นอกจากนี้ยังทำให้ผู้เล่นได้รับประสบการณ์การเล่นที่ท้าทาย ซึ่งในบทความนี้ได้กล่าวถึงการทำให้ DDA ในหลากหลายรูปแบบ โดยผู้จัดทำจะยกมาเพียงหนึ่งวิธีดังนี้

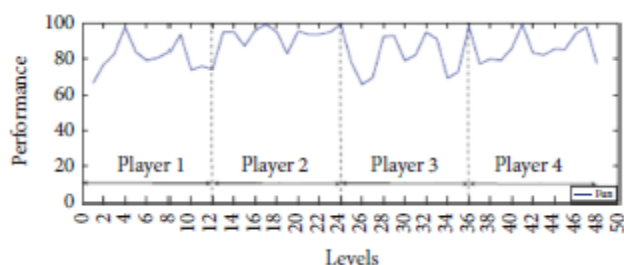
##### Single and Multilayered Perceptrons

จากงานวิจัย “Towards automatic personalized content generation for platform games” [7] Shaker, Yannakakis, และ Togelius ได้สร้าง ตัวกำเนิดฉากส่วนบุคคลอย่างอัตโนมัติสำหรับ platform games แรกเริ่มพวกเขาได้ใช้ single layer perceptrons (SLP) ในการประเมินระดับอารมณ์ของผู้เล่นโดยประมาณ ซึ่งเลือกอินพุตจากการเลือกพีเจอร์ตามลำดับ พีเจอร์ที่ควบคุมได้นั้นเป็นสิ่งที่จำเป็นในการสร้างสิ่งต่าง ๆ ให้เหมาะสมกับประสบการณ์ของผู้เล่นอย่างอัตโนมัติตามเวลาจริงและการทำนายอารมณ์ของผู้เล่น

ในงานวิจัยนี้ ได้มีการใช้การปรับตัวแบบพลวัตเพื่อที่จะเปลี่ยนแนวทางการเล่น ตัวแบบนี้มีประสิทธิภาพที่จะสรุปผลการทดสอบจากผู้เล่นหลากหลายประเภทได้ จึงมีการสร้างปัญญาประดิษฐ์มาสองตัวมาเป็นตัวแทนสำหรับเล่น ในขณะที่กำลังติดตามการเติบโตของค่า performance การทดลองนั้นเริ่มจากการสร้างด่านแบบสุ่ม โดยตัวแทนนั้นเล่น 100 ด่านโดยผลัดกันเล่นคนละ 20 ด่าน โดยผลลัพธ์จะแสดงในภาพที่ 2.4



ภาพที่ 2.4 ภาพแสดงระดับความสนุกของตัวแทนสองตัวแทน [7]



ภาพที่ 2.5 ภาพแสดงระดับความสนุกของผู้เข้าร่วมสี่คน [7]

จากภาพที่ 2.4 จะเห็นได้ว่า เมื่อตัวแทนคนแรกเริ่มเล่น ค่าความสนุกจะอยู่ที่ประมาณ 70% ในช่วง 20 ด่านแรก เมื่อตัวแทนคนที่สองเริ่มเล่น ค่าความสนุกเพิ่มขึ้นเป็น 80% ในช่วง 20 ด่านถัดมา จากนั้นเมื่อตัวแทนคนแรกกลับมาเล่นอีกครั้ง ค่าความสนุกลดลงมาที่ 70% นั้นแสดง

ได้อย่างชัดเจนว่าตัวแบบมีประสิทธิภาพในการปรับตัวตามลักษณะของผู้เล่น จึงมีการทดลองต่อไปกับผู้เข้าร่วม 4 คน แต่ละจำนวนด้านล่างเหลือคนละ 12 ด้าน โดยผลลัพธ์จะแสดงในภาพที่ 2.5 ซึ่งจะเห็นว่าได้ผลลัพธ์ที่คล้ายคลึงกับผลที่ได้จากการทดลองกับปัญญาประดิษฐ์ นั่นชี้ชัดว่าตัวแบบสามารถปรับตัวให้เข้ากับผู้เล่นแต่ละคนได้อย่างดี

### บทที่ 3

#### วิธีการวิจัย

ในบทนี้จะกล่าวถึงวิธีการวิจัยการพัฒนาปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์ที่สามารถปรับตัวได้ซึ่งจะพัฒนาจากสองแนวคิด คือปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ และปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึก นอกจากนี้จะกล่าวถึงวิธีการทดลองเพื่อหาตัวชี้วัดความใกล้เคียงของระดับความยาก โดยที่ปัญญาประดิษฐ์ดังกล่าวทั้งสองชุดจะถูกกำหนดให้ใช้ขั้นตอนวิธี minimax ที่มีความลึก 4 ชั้น และมีการกำหนดแต้ม [8] ดังนี้

หมากธรรมดา (piece) มีค่า 1 แต้ม

หมากที่อยู่มุมกระดาน (cornered piece) มีค่า 25 แต้ม

หมากที่อยู่ข้างหมากสีเดียวกันที่อยู่มุมกระดาน (ally cornered adjacent piece) มีค่า 5 แต้ม

หมากที่อยู่ข้างหมากคนละสีที่อยู่มุมกระดาน (opponent cornered adjacent piece) มีค่า -5 แต้ม

ซึ่งใช้คำนวณคะแนนฝั่งที่กำลังพิจารณาหรือ  $player_{score}$  ดังนี้

$$player_{score} = 1 \times p + 25 \times cp + 5 \times acap + (-5) \times ocap$$

โดยที่  $p$  คือจำนวนหมากธรรมดา

$cp$  คือจำนวนหมากที่อยู่มุมกระดาน

$acap$  คือจำนวนหมากที่อยู่ข้างหมากสีเดียวกันที่อยู่มุมกระดาน

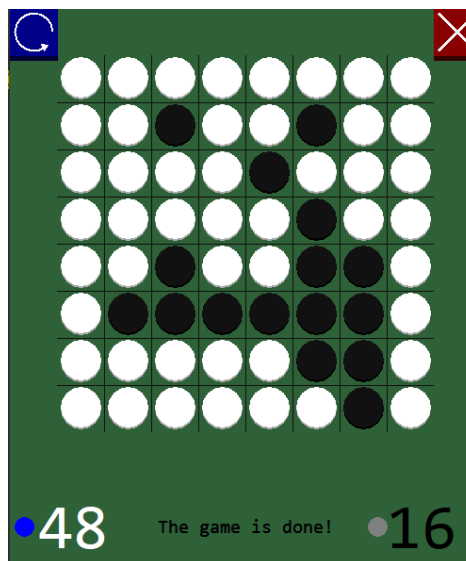
$ocap$  คือจำนวนหมากที่อยู่ข้างหมากคนละสีที่อยู่มุมกระดาน

เช่นเดียวกับ  $opponent_{score}$  ก็สามารถคำนวณได้ด้วยสมการดังกล่าวจากฝั่งตรงข้ามทั้ง  $player_{score}$  และ  $opponent_{score}$  จะนำไปใช้กับฟังก์ชันฮิวริสติกดังนี้

$$player_{advantage} = player_{score} - opponent_{score}$$

โดยสรุปการคิดคะแนนของกระดานจากฟังก์ชันฮิวริสติกนี้จะนำสีของหมากเข้ามาคิดด้วย ถ้าหากหมากบนกระดานที่พิจารณาเป็นฝ่ายเดียวกัน ค่าคะแนนจะมีค่าเป็นบวก แต่ถ้าหมากบนกระดานที่พิจารณาเป็นของฝ่ายตรงข้าม ค่าคะแนนจะมีค่าเป็นลบ โดยจะแสดงวิธีการคำนวณคะแนนจากเหตุการณ์ตัวอย่างในภาพที่ 3.1





ภาพที่ 3.1 เหตุการณ์ตัวอย่างประกอบการคำนวณคะแนน

จากภาพที่ 3.1 สามารถสรุปจำนวนหมากในกระดานทั้งหมดตามเงื่อนไขฟังก์ชันฮิวริสติก โดยให้หมากสีขาวเป็น player และหมากสีดำเป็น opponent ได้ว่า หมากสีดำ : หมากธรรมดา 15 ชิ้น หมากที่อยู่ข้างหมากคนละสีที่อยู่มุมกระดาน 1 ชิ้น หมากสีขาว : หมากที่อยู่มุมกระดาน 4 ชิ้น หมากที่อยู่ข้างหมากสีเดียวกันที่อยู่มุมกระดาน 7 ชิ้น หมากธรรมดา 37 ชิ้น ดังนั้น คะแนนของสีขาวมีค่า

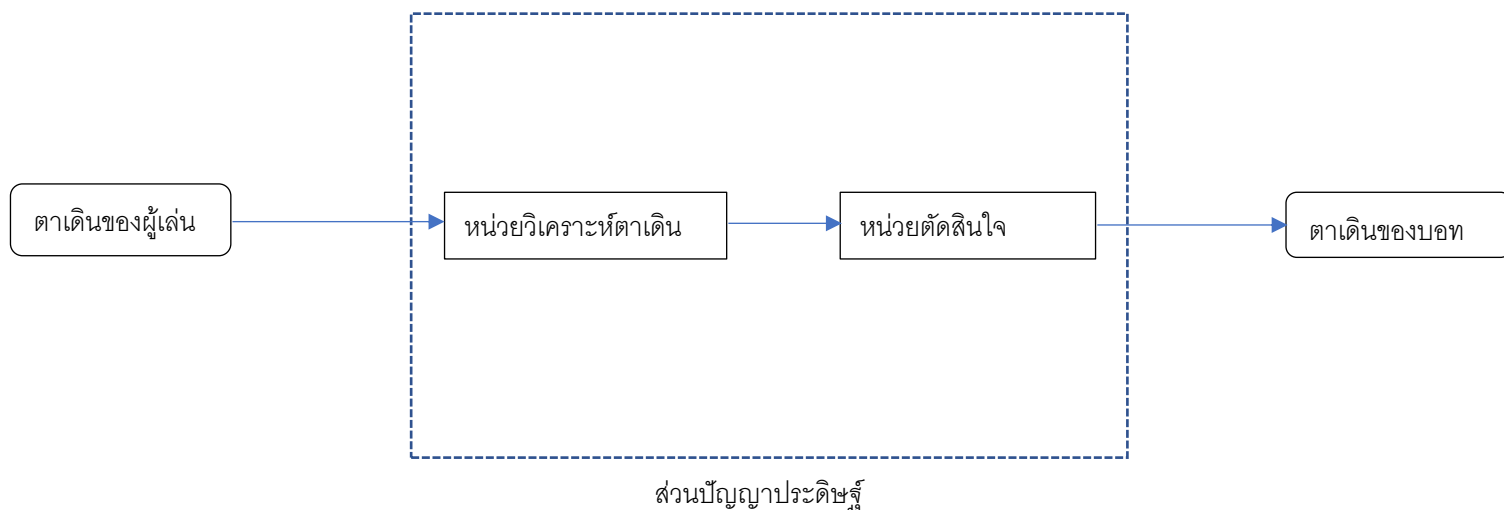
$$White_{score} = 37 \times 1 + 7 \times 5 + 4 \times 25 = 172$$

$$Black_{score} = 15 \times 1 + (-5) \times 1 = 10$$

$$White_{advantage} = 172 - 10 = 162$$

### 3.1 ปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ

ปัญญาประดิษฐ์ชุดนี้ได้ถูกพัฒนาภายใต้แนวคิดการเลือกตาเดินให้มีอันดับตาเดินที่ใกล้เคียงกับอันดับตาเดินของผู้เล่น เพื่อให้ผู้เล่นรู้สึกถึงความใกล้เคียงกันของระดับความสามารถ โดยโครงสร้างของปัญญาประดิษฐ์ชุดนี้ จะถูกแสดงดังภาพที่ 3.2



ภาพที่ 3.2 โครงสร้างของปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ

ปัญญาประดิษฐ์ชุดนี้จะประกอบไปด้วย 2 ส่วน ได้แก่

1. หน่วยวิเคราะห์ตาเดิน

หน่วยวิเคราะห์ตาเดินจะทำหน้าที่คำนวณคะแนนของกระดานที่เกิดจากตาเดินทุกตาที่สามารถเดินได้ผ่านขั้นตอนวิธี Minimax ที่มีความลึก 4 ชั้น โดยมีการตัดแปลงขั้นตอนวิธีให้สามารถส่งค่าคะแนนของทุก ๆ ตาเดินพร้อมทั้งจัดอันดับของตาเดินจากคะแนนได้ โดยมี pseudocode ดังนี้

```

global aiChoicelist := dict()

function minimax(node, depth, max_depth, maximizingPlayer)
  if depth = max_depth
    for each child of node
      aiChoicelist.update({child:-infinity})
  if depth = 0 or node is a terminal node
    return the utility of the node
  if maximizingPlayer
    bestValue := -infinity
    for each child of node
  
```

```

v := minimax(child, depth - 1, max_depth, FALSE)

bestValue := max(bestValue, v)

else (* minimizing player *)

    bestValue := +infinity

    for each child of node

        v := minimax(child, depth - 1, max_depth, TRUE)

        bestValue := min(bestValue, v)

    if child is in aiChoicelist.keys()

        if bestValue > aiChoicelist[child]

            aiChoicelist[child] := bestValue

aiChoicelist.sort(descend)

```

## 2. หน่วยตัดสินใจ

หน่วยตัดสินใจจะตัดสินใจเลือกตาเดินของบอทโดยใช้ค่าความได้เปรียบของเกม ณ ตาเดินนั้น ๆ มาสร้างเป็นเงื่อนไข ซึ่งคำนวณจากค่าอัตราส่วนของผลต่างของจำนวนหมากในกระดาน และจำนวนหมากในกระดานทั้งหมด โดยคิดจากสูตรดังนี้

$$\text{ความได้เปรียบ} = \frac{\text{จำนวนหมากของปัญญาประดิษฐ์} - \text{จำนวนหมากของผู้เล่น}}{\text{จำนวนหมากในกระดานทั้งหมด}}$$

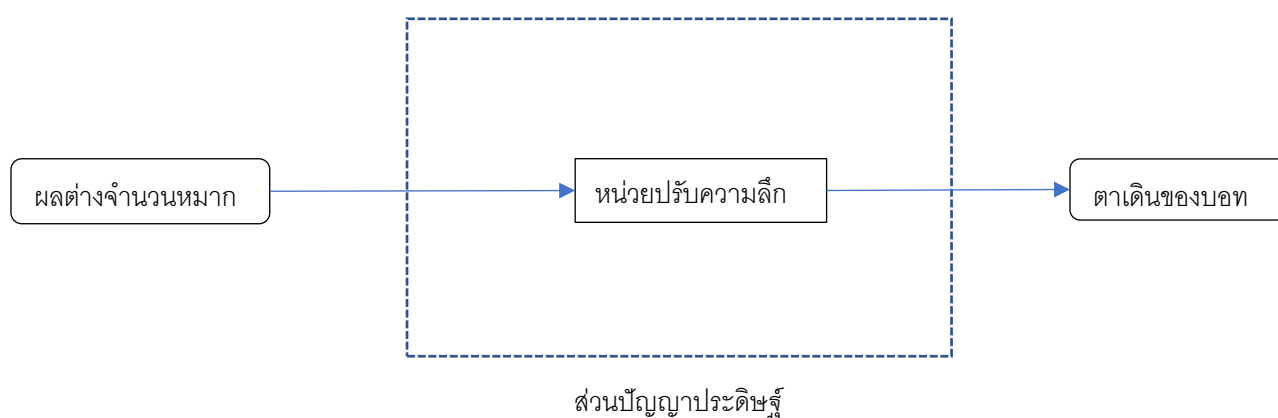
โดยที่การตัดสินใจจะอ้างอิงจากตาเดินของผู้เล่นเป็นหลัก ซึ่งจะคิดจากอันดับของตาเดินของผู้เล่นที่ได้มาจากหน่วยวิเคราะห์ตาเดิน แล้วนำมาปรับทางเลือกที่จะเดินตามเงื่อนไขที่กำหนดไว้ ดังตารางที่ 3.1

ตารางที่ 3.1 ตารางแสดงเงื่อนไขของการเลือกตาเดินของปัญญาประดิษฐ์

ความได้เปรียบ	ทางเลือก
มากกว่า 0.5	เลือกตาเดินที่ดึ้น้อยที่สุด
0.5 ถึง 0.4	เลือกตาเดินที่ดึ้น้อยกว่าผู้เล่น 2 ระดับ
0.4 ถึง 0.1875	เลือกตาเดินที่ดึ้น้อยกว่าผู้เล่น 1 ระดับ
0.1875 ถึง -0.1875	เลือกตาเดินที่ดีเท่ากับผู้เล่น
-0.1875 ถึง -0.4	เลือกตาเดินที่ดีมากกว่าผู้เล่น 1 ระดับ
-0.4 ถึง -0.5	เลือกตาเดินที่ดีมากกว่าผู้เล่น 2 ระดับ
น้อยกว่า -0.5	เลือกตาเดินที่ดีมากที่สุด

### 3.2 ปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึก

ปัญญาประดิษฐ์ชุดนี้ถูกพัฒนาภายใต้แนวคิดการปรับระดับความยากของปัญญาประดิษฐ์จากการปรับระดับความลึกในการคำนวณคะแนนของต้นไม้จากขั้นตอนวิธี Minimax โดยโครงสร้างของปัญญาประดิษฐ์จะถูกแสดงดังภาพที่ 3.3



ภาพที่ 3.3 ภาพแสดงโครงสร้างของปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึก

เนื่องจากในเกมกระดานเชิงกลยุทธ์ มนุษย์จะเลือกทางเลือกที่ดีที่สุดผ่านการจำลองเหตุการณ์ในเกมตามความสามารถของแต่ละคน ยิ่งคิดไปได้ตาเดินที่ลึกจะทำให้ได้ตาเดินที่ดีมากขึ้น ดังนั้นผู้วิจัยจึงเลือกที่จะปรับความลึกในการคำนวณของขั้นตอนวิธี Minimax ที่ปรับปรุงด้วย alpha-beta pruning โดยเริ่มต้นระดับความลึกไว้ที่ 4

ชั้น และกำหนดช่วงความลึกไว้ตั้งแต่ 1 ชั้น ถึง 6 ชั้น เพื่อไม่ให้ใช้เวลานานจนเกินไป จากนั้นจะปรับระดับความลึกตามเงื่อนไขโดยคิดจากผลต่างของจำนวนหมากในกระดานของทั้งสองฝ่ายโดยใช้จำนวนหมากของปัญญาประดิษฐ์ที่เดินทีหลังเป็นตัวตั้งเสมอ ดังตารางที่ 3.2

ตารางที่ 3.2 ตารางแสดงเงื่อนไขการปรับความลึกของปัญญาประดิษฐ์

ผลต่างของจำนวนหมากในกระดาน	การปรับความลึก
มากกว่า 5	เพิ่มขึ้น 1 ชั้น
น้อยกว่า 5	ลดลง 1 ชั้น

### 3.3 การทดลองเพื่อหาตัวชี้วัดความใกล้เคียงของระดับความยาก

ก่อนที่จะนำปัญญาประดิษฐ์ที่ปรับตัวได้ทั้งสองชุดไปทดสอบ การทดสอบจำเป็นจะต้องมีตัวชี้วัดเป็นหลักเกณฑ์ให้ยึดว่า ถ้าผู้เล่นทั้งสองฝ่ายมีความสามารถที่อยู่ในระดับใกล้เคียงกันแล้ว เมื่อเล่นจบเกมจะต้องมีค่าความได้เปรียบอยู่ในช่วงหนึ่งเสมอ โดยจะจำลองผู้เล่นทั้งสองฝ่ายโดยใช้ปัญญาประดิษฐ์ที่มีระดับความยากคงที่จากขั้นตอนวิธี Minimax แล้วนำมาเล่นโอเทลโล่แข่งกัน และนำผลลัพธ์มาวิเคราะห์ซึ่งจะกล่าวถึงในบทที่ 4 ต่อไป

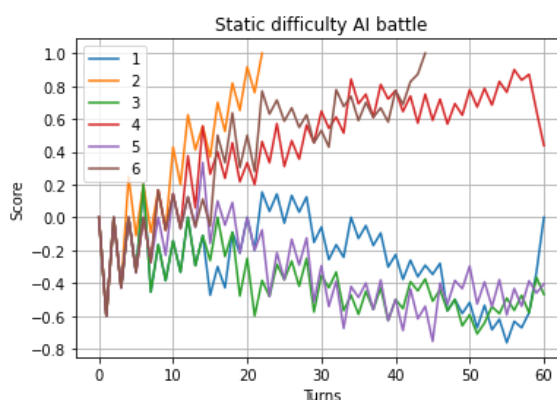
## บทที่ 4

### ผลการวิจัย

ในบทนี้จะกล่าวถึงสมมติฐานของการวิจัย ผลการวิจัย และการอภิปรายผลการทดลองนำปัญญาประดิษฐ์ที่ปรับตัวได้สำหรับเกมกระดานเชิงกลยุทธ์มาเล่นกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ทั้ง 6 ระดับ และนำมาทดสอบกับผู้เล่น 3 คน

#### 4.1 สมมติฐานของการวิจัย

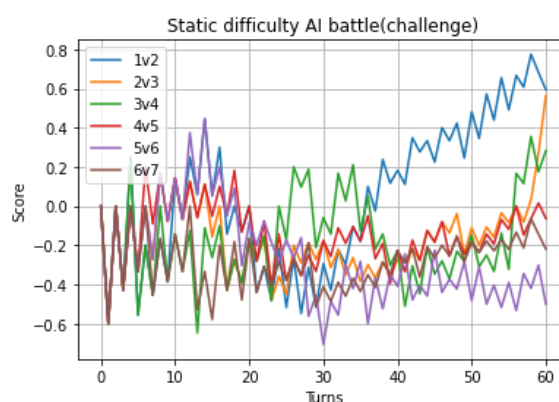
การวิจัยนี้ตั้งอยู่บนสมมติฐานว่า “ถ้าผู้เล่นทั้งสองฝ่ายมีความสามารถที่อยู่ในระดับใกล้เคียงกันแล้ว เมื่อเล่นจบเกมจะต้องมีค่าความได้เปรียบอยู่ในช่วงหนึ่งเสมอ” แต่สมมติฐานข้างต้นนี้มีปัญหาอยู่ที่ว่าช่วงของความได้เปรียบนั้นควรจะอยู่ที่ใด ดังนั้นผู้วิจัยจึงทำการนำปัญญาประดิษฐ์ที่มีระดับความยากเท่ากันมาทดสอบเล่นด้วยตัวเองเสียก่อน โดยจะมีความยากตั้งแต่ Minimax 1 ชั้น ถึง Minimax 6 ชั้น ซึ่งจะแสดงผลลัพธ์ในรูปแบบของกราฟแสดงค่าความได้เปรียบตลอดทั้งเกม โดยค่าความได้เปรียบจะคิดจากปัญญาประดิษฐ์ที่เดินทีหลัง ดังภาพที่ 4.1



ภาพที่ 4.1 ผลการทดลองระหว่างปัญญาประดิษฐ์ที่มีระดับความยากคงที่ (Static difficulty AI) ซึ่งมีระดับความยากเท่ากัน

จากภาพที่ 4.1 จะสังเกตเห็นได้ว่า เมื่อนำปัญญาประดิษฐ์ที่มีความยากเท่ากันมาทดลองแล้วพบว่าผลของการทดลองของ Minimax 2 ชั้น และ Minimax 6 ชั้น จะเล่นจบก่อนตาเดินที่ 60 และค่าความได้เปรียบอยู่ที่ 1.0 แสดงว่าผลแพ้ชนะออกมาโดยที่หมากทั้งกระดานมีเพียงแค่สีดำเพียงอย่างเดียว เมื่อพิจารณาที่ผลการทดลองของ Minimax 3 ชั้น Minimax 4 ชั้น และ Minimax 5 ชั้น จะพบว่าตอนจบเกมจะมีค่าความได้เปรียบอยู่ที่ -0.46875 0.4375 และ -0.40625 ตามลำดับ และผลการทดลองของ Minimax 1 ชั้น มีค่าความได้เปรียบอยู่ที่ 0.0 ซึ่งแสดงว่าเสมอกัน

จากผลการทดลองดังกล่าวจะเห็นว่าการที่นำปัญญาประดิษฐ์ที่มีระดับความยากเท่ากันส่วนใหญ่จะมีค่าความได้เปรียบในตอนจบเกมอยู่ในช่วงประมาณ 0.45 ถึง -0.45 แต่มีถึง 2 เกมที่มีการชนะอย่างสมบูรณ์แบบตั้งนั้นผู้วิจัยจึงมีแนวคิดที่จะปรับการทดลองเล็กน้อย โดยให้ปัญญาประดิษฐ์ที่เดินทีหลังมีระดับความยากเพิ่มขึ้น 1 ระดับ เนื่องจากโอเทลโลมีลักษณะของกระดานที่สมมาตร ทำให้การเริ่มเล่นในตาเดินที่ 1 มีคะแนนจาก heuristic function เท่ากันทุกตาเดิน ดังนั้นผู้เล่นที่เดินทีหลังจะเป็นผู้ดำเนินแผนของตนเองก่อน และไม่สามารถล่วงรู้ความคิดของผู้ที่เดินก่อนได้ในตาเดินแรก ดังนั้นผู้ที่เดินทีหลังจะมีความเสียเปรียบอยู่เล็กน้อย ซึ่งจะแสดงผลลัพธ์ดังภาพที่ 4.2



ภาพที่ 4.2 ผลลัพธ์ของการทดสอบระหว่างปัญญาประดิษฐ์ที่มีระดับความยากคงที่ (Static difficulty AI) กับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ซึ่งเพิ่มระดับความยากขึ้น 1 ระดับ

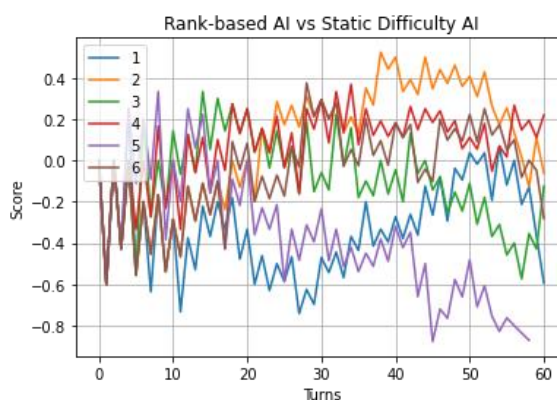
จากภาพที่ 4.2 จะสังเกตได้ว่า ค่าความได้เปรียบในตอนจบเกมจะมีค่าอยู่ระหว่าง 0.6 ถึง -0.6 ซึ่งแสดงถึงขอบเขตที่สามารถนำไปใช้เป็นตัวชี้วัดความใกล้เคียงกันของระดับความยากของผู้เล่นทั้งสองฝ่ายได้ ซึ่งเมื่อคำนวณออกมาเป็นจำนวนหมากในกระดานจะได้ว่าผลต่างตัวหมากของทั้งสองฝ่ายในช่วงจบเกมควรมีค่าอยู่ที่ 38.4 ตัว หรือประมาณ 38 ตัว

## 4.2 ผลการวิจัย

ผลการวิจัยจะแบ่งออกเป็นสองส่วนคือ การทดลองกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ และ การทดลองกับมนุษย์

### 4.2.1 การทดลองกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่

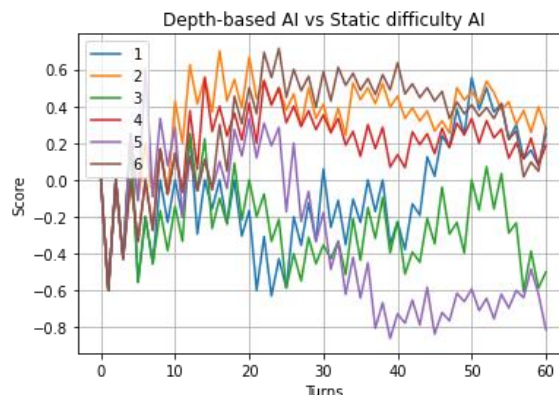
หลังจากที่ปัญญาประดิษฐ์ได้ผ่านการพัฒนาตามที่ได้กล่าวไปในบทที่ 3 ผู้วิจัยได้นำปัญญาประดิษฐ์ที่สามารถปรับตัวได้ทั้งสองชุดมาทดสอบกับปัญญาประดิษฐ์ที่มีความยากคงที่ทั้ง 6 ระดับ โดยผลลัพธ์จะแสดงไว้ดังภาพที่ 4.3 และภาพที่ 4.4



ภาพที่ 4.3 ผลลัพธ์ของการทดสอบปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ทั้ง 6 ระดับ

จากภาพที่ 4.3 จะเห็นได้ว่าปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับสามารถเล่นได้คะแนนความได้เปรียบอยู่ในช่วงที่กำหนดไว้เกือบทั้งหมด ยกเว้นแค่ระดับที่ 5 ที่มีค่าความได้เปรียบต่ำกว่า -0.8 และจบก่อนตาเดินที่ 60 นอกจากนี้ยังมีระดับที่ 1 ที่มีค่าความได้เปรียบอยู่ที่ประมาณ -0.6 เนื่องจากเกิดเหตุการณ์ที่ปัญญาประดิษฐ์ที่ปรับตัวได้ไม่สามารถลงหมากเล่นได้ ทำให้ถูกข้ามตาเดินไป 1 ตาเดิน จึงส่งผลให้คะแนนถูกทิ้งห่างไปได้





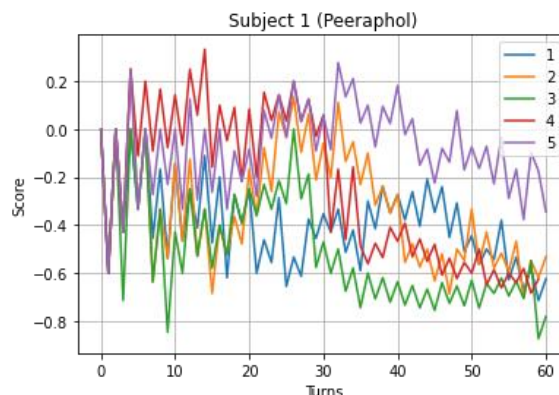
ภาพที่ 4.4 ผลลัพธ์ของการทดสอบปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ทั้ง 6 ระดับ

จากภาพที่ 4.4 จะเห็นได้ว่าปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกสามารถเล่นได้คะแนนความได้เปรียบอยู่ในช่วงที่กำหนดไว้เกือบทั้งหมด ยกเว้นแค่ระดับที่ 5 ที่มีค่าความได้เปรียบประมาณ -0.8 นอกจากนี้ค่าความได้เปรียบช่วงจบเกมจากการแข่งกับปัญญาประดิษฐ์ที่มีความยากระดับที่ 1 ระดับที่ 2 ระดับที่ 4 และระดับที่ 6 มีค่าใกล้เคียงกันโดยอยู่ในช่วง 0.2 ถึง 0.4

จากการทดสอบข้างต้นพบว่า ปัญญาประดิษฐ์ทั้งสองชุดมีความสามารถในการปรับตัวที่ใกล้เคียงกัน แต่ในทางปฏิบัติ การเพิ่มความลึกในการคำนวณของขั้นตอนวิธี Minimax ส่งผลให้ปัญญาประดิษฐ์ใช้เวลาคำนวณที่มากขึ้น นอกจากนี้การพัฒนาปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกใช้การปรับระดับความลึกทีละหนึ่งชั้น อาจทำให้เกิดเหตุการณ์ที่มนุษย์ต้องเล่นกับปัญญาประดิษฐ์ที่มีระดับความลึกมากเป็นเวลานาน จึงอาจทำให้ผู้เล่นรอนานเกินไปได้ ดังนั้นผู้วิจัยจึงเลือกปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดอันดับมาทดสอบกับมนุษย์

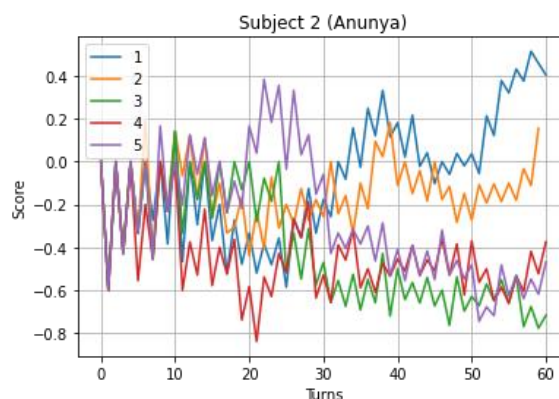
#### 4.2.2 การทดลองกับมนุษย์

หลังจากผ่านการทดสอบกับปัญญาประดิษฐ์มาแล้วได้ผลลัพธ์ที่น่าพึงพอใจ ผู้วิจัยจึงดำเนินการนำปัญญาประดิษฐ์ตัวนี้มาทดสอบกับมนุษย์บ้าง ซึ่งการทดสอบครั้งนี้จะมีผู้เข้าร่วมทดสอบ 3 คน และทดสอบคนละ 5 ครั้ง โดยจะแสดงผลดังภาพที่ 4.5 ถึงภาพที่ 4.8



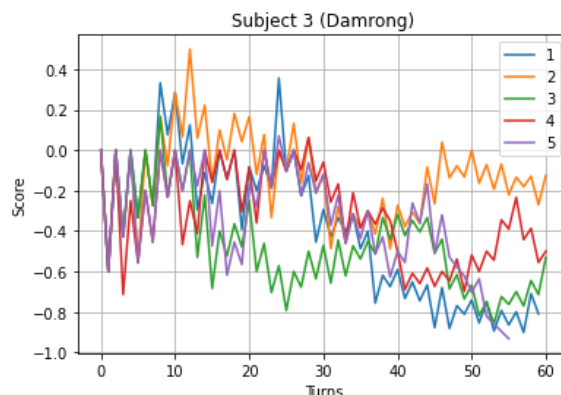
ภาพที่ 4.5 ผลการทดสอบปัญญาประดิษฐ์กับมนุษย์คนที่หนึ่ง

จากการทดสอบกับผู้เล่นคนแรกพบว่าค่าความได้เปรียบของปัญญาประดิษฐ์ในช่วง 30 ตาเดินแรกมีลักษณะที่ค่อนข้างแกว่ง ซึ่งแสดงถึงความพยายามที่จะทำให้การเล่นในแต่ละรอบมีความใกล้เคียงกันของระดับความสามารถของทั้งสองฝ่าย แต่เมื่อเกมเริ่มผ่านเข้าสู่ช่วงตาเดินที่ 32 ปัญญาประดิษฐ์ไม่สามารถที่จะควบคุมสถานการณ์ไว้ได้ และค่าความได้เปรียบตอนจบเกมส่วนใหญ่มีค่าประมาณ -0.6 ซึ่งอยู่ในเกณฑ์ที่เทียบเท่ากับการนำปัญญาประดิษฐ์ที่มีระดับความยากใกล้เคียงกันมาทดสอบในภาพที่ 4.2



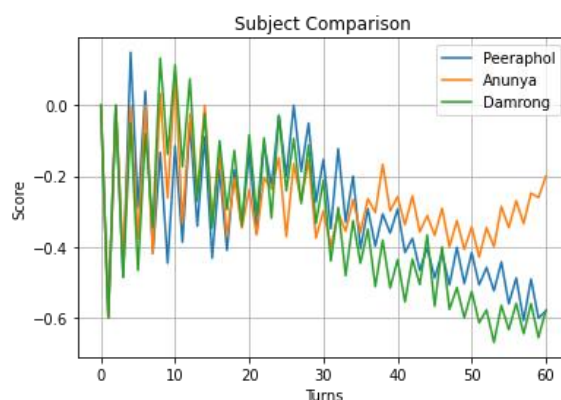
ภาพที่ 4.6 ผลการทดสอบปัญญาประดิษฐ์กับมนุษย์คนที่สอง

จากการทดสอบกับผู้เล่นคนที่สองพบว่าในเกมที่ 1 และเกมที่ 2 ปัญญาประดิษฐ์สามารถชนะผู้เล่นโดยมีค่าความได้เปรียบตอนจบเกมไม่เกิน 0.4 ซึ่งอยู่ในเกณฑ์ที่ตั้งไว้ นอกจากนี้ในเกมที่ 3 ถึงเกมที่ 5 ปัญญาประดิษฐ์เล่นแพ้ผู้เล่นทั้งหมดโดยมีค่าความได้เปรียบอยู่ที่ -0.71875 -0.375 และ -0.46875 ตามลำดับ ผลการเล่นทั้ง 5 ครั้ง ได้แสดงออกถึงความใกล้เคียงกันของระดับความสามารถของผู้เล่นและปัญญาประดิษฐ์ชุดนี้ได้เป็นอย่างดี



ภาพที่ 4.7 ผลการทดสอบปัญญาประดิษฐ์กับมนุษย์คนที่สาม

จากการทดสอบกับผู้เล่นคนที่สามพบว่าผลการเล่นส่วนใหญ่จะมีค่าความได้เปรียบตอนจบเกมอยู่ในช่วง  $-0.6$  โดยค่าความได้เปรียบตอนจบเกมตั้งแต่รอบที่ 1 ถึงรอบที่ 5 มีค่า  $-0.8095238095238095$   $-0.125$   $-0.53125$   $-0.5$  และ  $-0.9322033898305084$  ตามลำดับ ซึ่งมีเพียงรอบที่ 2 ที่ปัญญาประดิษฐ์สามารถเล่นกับมนุษย์ได้อย่างสูสี



ภาพที่ 4.8 การเปรียบเทียบค่าความได้เปรียบเฉลี่ยของปัญญาประดิษฐ์จากการเล่นกับมนุษย์ 3 คน

จากภาพที่ 4.8 จะเห็นว่าใน 30 ตาเดินแรก ปัญญาประดิษฐ์สามารถควบคุมค่าความได้เปรียบไว้ให้ใกล้เคียงกันได้ดี นอกจากนี้ค่าความได้เปรียบเฉลี่ยตลอดทั้งเกมของเส้นสีส้มมีค่าอยู่ระหว่าง  $0.4$  ถึง  $-0.4$  แสดงถึงความใกล้เคียงของระดับความสามารถระหว่างผู้เล่นและปัญญาประดิษฐ์ได้เป็นอย่างดี อีกทั้งยังสามารถทำให้ค่าความได้เปรียบตอนจบเกมทั้งสองเส้นที่เหลือมีค่าไม่ต่ำกว่า  $-0.6$  ได้ ซึ่งอยู่ในเกณฑ์ที่พอใช้

### 4.3 การอภิปรายผล

จากการศึกษาตัวชี้วัดความใกล้เคียงของระดับความสามารถของทั้งสองฝ่ายโดยการนำปัญญาประดิษฐ์ที่มีระดับความยากเท่ากันมาทดสอบดังภาพที่ 4.1 และ 4.2 ทำให้กำหนดตัวชี้วัดได้โดยมีค่าตั้งแต่ 0.6 ถึง -0.6

จากการทดลองกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ทั้ง 6 ระดับดังภาพที่ 4.3 และ 4.4 พบว่าปัญญาประดิษฐ์ที่สามารถปรับตัวได้มีค่าความได้เปรียบส่วนใหญ่อยู่ในช่วง 0.6 ถึง -0.6 ทำให้สรุปได้ว่าปัญญาประดิษฐ์ที่ปรับตัวได้ทั้งสองชุดน่าจะสามารถปรับระดับความยากของตนเองให้เหมาะสมกับผู้เล่นส่วนมากได้ แต่ปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกมีปัญหาในด้านของเวลา ทำให้ผู้วิจัยเลือกนำปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับมาทดสอบกับมนุษย์

ผลจากการทดสอบกับมนุษย์ทั้งสามคนดังภาพที่ 4.5 4.6 และ 4.7 จะเห็นได้ว่าปัญญาประดิษฐ์ชุดนี้สามารถเล่นให้มีระดับความสามารถที่ใกล้เคียงกับมนุษย์ได้ในระดับที่เทียบเท่ากับการนำปัญญาประดิษฐ์ที่มีความสามารถเท่ากันมาทดสอบ แต่ก็ยังไม่สามารถทำให้ค่าความได้เปรียบอยู่ในช่วงที่กำหนดไว้ได้โดยตลอด

## บทที่ 5

### ข้อสรุปและข้อเสนอแนะ

ในบทนี้กล่าวถึง สรุปผลการวิจัยการพัฒนาปัญญาประดิษฐ์แบบนามธรรมสำหรับเกมกระดานเซิงกลยุทธ์ และข้อเสนอแนะ โดยมีรายละเอียดดังนี้

#### 5.1 ข้อสรุป

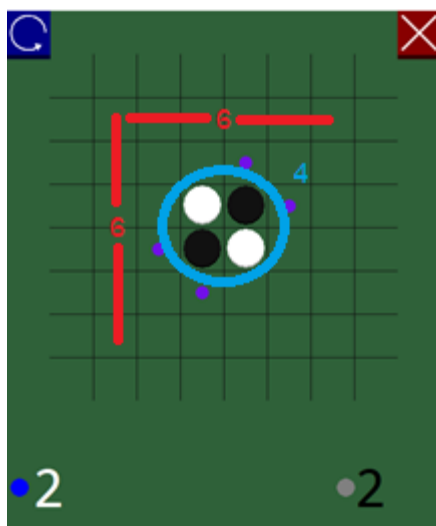
ในงานวิจัยเล่มนี้มีวัตถุประสงค์เพื่อพัฒนาปัญญาประดิษฐ์สำหรับเกมกระดานเซิงกลยุทธ์แบบนามธรรมที่สามารถปรับตัวให้เหมาะสมต่อผู้เล่นทุกระดับ โดยพิจารณาจากอันดับตาเดินของผู้เล่น และค่าความได้เปรียบ ณ ขณะนั้น ซึ่งแบ่งขั้นตอนในงานวิจัยออกเป็นสองขั้นตอน คือ การพัฒนาปัญญาประดิษฐ์ที่สามารถปรับตัวได้ และการทดสอบปัญญาประดิษฐ์

สำหรับการพัฒนาปัญญาประดิษฐ์ที่สามารถปรับตัวได้ ผู้วิจัยได้พัฒนาปัญญาประดิษฐ์สองชุด คือ ปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับ และปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึก ซึ่งปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับมีโครงสร้างสองส่วน คือ หน่วยวิเคราะห์ตาเดิน และหน่วยตัดสินใจ โดยหน่วยวิเคราะห์ตาเดินมีหน้าที่วิเคราะห์ตาเดินทุกตาของผู้เล่นผ่านขั้นตอนวิธี Minimax ที่มีความลึก 4 ชั้น เมื่อผู้เล่นทำการเดินหมาก หน่วยวิเคราะห์ตาเดินจะให้อันดับการเดินของผู้เล่นแล้วส่งข้อมูลไปให้หน่วยตัดสินใจ โดยหน่วยตัดสินใจทำหน้าที่รับข้อมูลอันดับการเดินของผู้เล่น และคำนวณค่าความได้เปรียบจากจำนวนหมากในกระดาน เพื่อเป็นข้อมูลประกอบการเลือกตาเดินของปัญญาประดิษฐ์ ส่วนปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกจะมีโครงสร้างเพียงแค່หน่วยปรับความลึก โดยหน่วยปรับความลึกจะปรับความลึกโดยมีเงื่อนไขจากผลต่างของจำนวนหมากในกระดาน

สำหรับการทดสอบปัญญาประดิษฐ์จะแบ่งเป็นสองส่วน คือการทดสอบกับปัญญาประดิษฐ์ที่มีความยากคงที่ 6 ระดับ และการทดสอบกับมนุษย์ ซึ่งผลการทดสอบกับปัญญาประดิษฐ์ที่มีความยากคงที่ 6 ระดับพบว่าปัญญาประดิษฐ์ทั้งสองชุดมีค่าความได้เปรียบตอนจบเกมอยู่ในช่วง 0.6 ถึง -0.6 เป็นส่วนมาก แต่ปัญญาประดิษฐ์ที่ปรับตัวด้วยความลึกมีข้อจำกัดด้านเวลา จึงทำให้ผู้วิจัยจำเป็นต้องเลือกปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับมาทดสอบกับมนุษย์

จากการนำปัญญาประดิษฐ์ที่ปรับตัวด้วยการจัดลำดับมาทดสอบกับมนุษย์พบว่าค่าความได้เปรียบโดยเฉลี่ยจากการเล่น 5 ครั้งของผู้เล่นแต่ละคนมีค่าอยู่ในช่วง -0.2 ถึง -0.6 ซึ่งปัญญาประดิษฐ์ไม่สามารถควบคุมค่าความได้เปรียบให้อยู่ในช่วงที่กำหนดได้ ถึงแม้ว่าจะสามารถควบคุมค่าความได้เปรียบในช่วง 30 ตาเดินแรกของเกมได้ เพราะโอเทลโลเป็นเกมที่ดำเนินด้วยการวางหมากคร่อมหมากของฝ่ายตรงข้าม ทำให้หมากที่อยู่มุมกระดานเป็นหมากที่สร้างความได้เปรียบได้มากที่สุด เพราะเป็นตำแหน่งที่ไม่สามารถถูกคร่อมได้อย่างแน่นอน นอกจากนี้โอเทลโลเป็นเกมกระดานที่เริ่มต้นด้วยกระดาน 64 ช่อง และหมาก 4 ตัวบนกระดาน ดังนั้นผู้เล่นทั้งสองฝ่ายจะ

สามารถวางหมากในช่วงก่อนที่จะถูกบังคับให้วางมุมกระดานรวมกันอย่างมาก  $6 \times 6 - 4 = 32$  ตาเดิน ดังภาพที่ 5.1



ภาพที่ 5.1 ภาพประกอบการคำนวณจำนวนตาเดินก่อนถึงช่วงมุมกระดาน

## 5.2 ข้อเสนอแนะ

จากการทดสอบปัญญาประดิษฐ์ที่สามารถปรับตัวได้กับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ และทดสอบกับมนุษย์ ผู้วิจัยมีข้อเสนอแนะดังต่อไปนี้

1. ในการพัฒนาปัญญาประดิษฐ์ที่สามารถปรับตัวได้ ควรมีการปรับวิธีการเลือกตาเดินให้มีความยืดหยุ่นมากขึ้น เช่น การเพิ่มความน่าจะเป็นสำหรับการเลือกตาเดินอื่น ๆ เนื่องจากการออกแบบด้วยขั้นตอนวิธี Minimax แบบดัดแปลง ทำให้ตัวปัญญาประดิษฐ์นั้นทำการตัดสินใจด้วยขั้นตอนวิธีที่มีแนวคิดคล้ายคลึงกับขั้นตอนวิธีเชิงละโมภ (Greedy Algorithm) ซึ่งส่งผลให้ปัญญาประดิษฐ์ตัดสินใจได้ไม่ยืดหยุ่น และทำให้การทดสอบกับปัญญาประดิษฐ์ที่ออกแบบมาด้วยขั้นตอนวิธี Minimax มีผลลัพธ์ที่ตายตัว
2. ในการทดสอบปัญญาประดิษฐ์ที่สามารถปรับตัวได้กับมนุษย์ ควรให้มนุษย์ทำการทดสอบความสามารถของตนเองก่อน โดยให้มนุษย์ทดลองเล่นกับปัญญาประดิษฐ์ที่มีระดับความยากคงที่ทั้ง 6 ระดับก่อน เพื่อให้ข้อมูลผลการทดสอบปัญญาประดิษฐ์ที่สามารถปรับตัวได้ นำมาใช้วิเคราะห์ได้ชัดเจนขึ้น
3. การวัดความใกล้เคียงของระดับความสามารถของผู้เล่นทั้งสองฝ่าย ควรจะพิจารณาไปที่อัตราการทำชนะ ประกอบด้วย นอกจากค่าอัตราส่วนของผลต่างของจำนวนหมากในกระดาน เพื่อทำให้ผู้เล่นที่เล่นด้วยไม่รู้สึกว่าปัญญาประดิษฐ์ที่ผู้เล่นกำลังเผชิญอยู่กำลังเล่นเพื่อเสมอ

## รายการอ้างอิง

- [1] AlphaGo. [<https://deepmind.com/research/case-studies/alphago-the-story-so-far>]. เข้าถึงเมื่อ 26 ตุลาคม พ.ศ. 2563
- [2] Mihaly. Csikszentmihalyi, Flow: The Psychology of Optimal Experience, Harper Row, NewYork, NY, USA, 2009.
- [3] Mohammad Zohaib, "Dynamic Difficulty Adjustment (DDA) in Computer Games", Advances in Human-Computer Interaction Volume 2018, 2018.
- [4] Robin Hunicke and Vernell Chapman, "AI for Dynamic Difficulty Adjustment in Games," in Proceedings of the Challenges in Game Artificial Intelligence AAAIWorkshop, pp. 91–96, San Jose, Calif, USA, 2004.
- [5] Owen Shevlin, "Creating an AI to play Othello", 2017 เข้าถึงเมื่อ 26 ตุลาคม พ.ศ. 2563
- [6] Mini-Max Algorithm in Artificial Intelligence. [<https://www.javatpoint.com/mini-max-algorithm-in-ai>]. เข้าถึงเมื่อ 26 ตุลาคม พ.ศ. 2563
- [7] Noor Shaker, Georgios Yannakakis and Julian Togelius, "Towards automatic personalized content generation for platform games," in Proceedings of the 6th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2010, pp. 63–68, Stanford, Calif, USA, October 2010.
- [8] John Fish. "Othello", 2015 เข้าถึงเมื่อ 26 ตุลาคม พ.ศ. 2563

## ภาคผนวก ก

### แบบเสนอหัวข้อโครงการ รายวิชา 2301399 Project Proposal ปีการศึกษา 2563

ชื่อโครงการ (ภาษาไทย)	การสร้างปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรม
ชื่อโครงการ (ภาษาอังกฤษ)	Creating Artificial Intelligence for Abstract Strategy Board Game
อาจารย์ที่ปรึกษา	1. รองศาสตราจารย์ ดร.ศุภกานต์ พิมลธเรศ 2. ผู้ช่วยศาสตราจารย์ ศศิภา พันธวุฒิชัย
ผู้ดำเนินการ	นายพีรพล อยู่ไพศาล เลขประจำตัวนิสิต 6033643723 สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

#### หลักการและเหตุผล

เกมกระดานเชิงกลยุทธ์หรือหมากกระดานนั้นเป็นเกมประเภทหนึ่งที่อยู่คู่กับมนุษย์มายาวนาน และมีความแตกต่างกันในแต่ละประเทศ อีกทั้งยังมีส่วนช่วยในการพัฒนาทักษะในด้านต่าง ๆ ของสมอง เช่น การวางแผน การวิเคราะห์สถานการณ์ ความรอบคอบ ซึ่งในปัจจุบันมีการพัฒนาปัญญาประดิษฐ์ที่มุ่งเน้นไปที่ความเก่งของตัวปัญญาประดิษฐ์เพื่อนำมาเล่นหมากกระดานแข่งกับมนุษย์อย่างมากมาย เช่น AlphaGo[1] แต่เนื่องจากหมากกระดานเป็นเกมประเภทหนึ่งที่ต้องอาศัยการคิดวิเคราะห์และการวางแผนเป็นอย่างมาก ดังนั้นการที่ผู้เล่นจะพัฒนาตนเองไปได้จึงต้องมาจากการเล่นซ้ำ ๆ ซึ่งถ้าหากเล่นกับผู้เล่นที่เก่งกว่ามาก ๆ นั้นจะส่งผลให้เกิดความเครียด และหากเล่นกับผู้เล่นที่มีมือด้อยกว่า จะทำให้เกิดความเบื่อหน่ายได้ตามแนวคิด flow channel ของ Csikszentmihalyi[2] ดังนั้นผู้พัฒนาจึงต้องการพัฒนาปัญญาประดิษฐ์ที่สามารถปรับระดับความยากให้เหมาะสมกับผู้เล่นแต่ละคนได้อย่างอัตโนมัติ

ในการศึกษาเกี่ยวกับการปรับระดับความยากแบบพลวัต (Dynamic Difficulty Adjustment) จากบทความปริทัศน์ของ Mohammad Zohaib[3] ได้รวบรวมวิธีการพัฒนาปัญญาประดิษฐ์ที่สามารถปรับระดับความยากแบบพลวัตได้ในรูปแบบต่าง ๆ สำหรับเกมหลากหลายประเภท ซึ่งเกมแต่ละประเภทยังมีแนวคิดที่



เหมือนกันคือ การรับข้อมูลต่าง ๆ ที่ผู้เล่นเป็นคนป้อนให้ในเกมแล้วนำมาวิเคราะห์ระดับทักษะของผู้เล่น จากนั้นจึงปรับพฤติกรรมของตนเองให้เหมาะสมกับผู้เล่นที่เผชิญหน้าอยู่ ณ ขณะนั้น ในกระบวนการปรับระดับความยากของเกม ระบบจะต้องปรับความยากของเกมด้วยความต่อเนื่องและนุ่มนวลที่สุด เพื่อไม่ให้ผู้เล่นทราบว่าเกมที่กำลังเล่นอยู่นั้นได้ถูกปรับระดับความยากให้เหมาะสม เช่น ระบบพิจารณาสถานะของผู้เล่นเพื่อนำไปสร้างความสามารถของศัตรูในเกมยิงมุมมองบุคคลที่หนึ่ง (First Person Shooter)[4] กล่าวคือ หากผู้เล่นนั้นมีค่าพลังชีวิตต่ำ หรือมีทรัพยากรที่จะต้องใช้ในการฝ่าอุปสรรคอยู่น้อย ระบบจะช่วยเหลือผู้เล่นให้สามารถผ่านอุปสรรคไปได้ง่ายขึ้น ในทางกลับกัน หากผู้เล่นมีทรัพยากรมาก ระบบก็จะออกแบบให้อุปสรรคมีความยากและท้าทายมากกว่าปกติ เพื่อเพิ่มรรถรสในการเล่น ซึ่งผลลัพธ์ที่ได้จากการสร้างระบบการปรับความยากแบบพลวัตพบว่า ปัญญาประดิษฐ์นั้นสามารถปรับระดับความยากได้ด้วยตนเองและผู้เล่นมีส่วนร่วมกับการปรับความยากแบบพลวัตมากกว่าเกมที่ออกแบบระดับความยากแบบปกติ โดยพิจารณาผ่านตัวชี้วัดต่าง ๆ เช่น จำนวนการเล่นเกมนต่อวัน ระดับความสนุกของผู้เล่นทดสอบโดยวัดจากค่าประสิทธิภาพของผู้เล่น และอัตราการแพ้ชนะของปัญญาประดิษฐ์

ในการศึกษาวิธีการพัฒนาปัญญาประดิษฐ์สำหรับหมากระดานนั้น โครงการของ Owen Shevlin[5] ได้มีการพัฒนาและทดสอบประสิทธิภาพของปัญญาประดิษฐ์หลากหลายประเภทสำหรับเกมโอเทลโล ได้แก่ Minimax, Monte-Carlo Tree Search และ Deep Learning ซึ่งการพัฒนาปัญญาประดิษฐ์แต่ละประเภทจะส่งผลให้ตัวปัญญาประดิษฐ์มีความสามารถที่ไม่เท่ากัน แต่ผู้พัฒนาได้สังเกตเห็นว่า เมื่อนำปัญญาประดิษฐ์ทั้งหมดมาทดลองเล่นแข่งกับมนุษย์พบว่า อัตราการชนะมีค่าที่ต่างกันไม่มาก ผู้พัฒนาจึงเลือกวิธีการพัฒนาที่ง่ายต่อการนำมาดัดแปลงเพื่อสร้างปัญญาประดิษฐ์ที่สามารถปรับตัวได้ โดยจะนำวิธีการ minimax[6] มาดัดแปลง เพราะ minimax คือการสร้างต้นไม้ที่มีค่าผลรวมสูงสุดขึ้นต่ำ เนื่องจากโครงสร้างของหมากระดานนั้นสามารถมองให้อยู่ในรูปของต้นไม้ได้ โดยที่ต้นไม้หนึ่งต้นจะมีโหนดเก็บค่าความได้เปรียบของผู้เล่นหนึ่งคนที่สนใจ กล่าวคือการเลือกตาเดิน (move) ที่ดีที่สุดจะเปรียบเสมือนการค้นหาโหนดที่มีค่าความได้เปรียบมากที่สุด นอกจากนี้ในการเล่นหมากระดานผู้เล่นทั้งสองฝ่ายจะต้องผลัดกันเล่นคนละหนึ่งตาเดิน ดังนั้นตาเดินที่ดีที่สุดของผู้เล่นฝั่งตรงข้ามก็คือการค้นหาโหนดที่มีค่าความได้เปรียบของเราน้อยที่สุด

จากที่กล่าวมาข้างต้น ทำให้ผู้พัฒนาที่มีความสนใจที่จะศึกษาและพัฒนาปัญญาประดิษฐ์ที่สามารถปรับตัวให้เหมาะสมต่อผู้เล่นทุกระดับ เพื่อให้ผู้เล่นได้สนุกกับการเล่นเกมหมากระดานมากขึ้น อีกทั้งยังสามารถนำปัญญาประดิษฐ์นี้ไปเป็นคู่มือเพื่อพัฒนาฝีมือและทักษะทางกระบวนการคิดได้

## วัตถุประสงค์

เพื่อพัฒนาปัญญาประดิษฐ์สำหรับเกมกระดานเชิงกลยุทธ์แบบนามธรรมที่สามารถปรับตัวให้เหมาะสมต่อผู้เล่นทุกระดับ



## ประโยชน์ที่คาดว่าจะได้รับ

1. ประโยชน์ด้านความรู้และประสบการณ์ต่ออนิสิต
  - 1.1 ได้รับความรู้ ความเข้าใจเกี่ยวกับการพัฒนาปัญญาประดิษฐ์
  - 1.2 ได้ฝึกการคิดวิเคราะห์ การทำงานอย่างเป็นระบบ
  - 1.3 ได้รับความรู้เกี่ยวกับขั้นตอนวิธีที่ใช้กับเกมกระดานเชิงกลยุทธ์
2. ประโยชน์ที่ได้จากโครงการที่พัฒนาขึ้น
  - 2.1 ได้เกมกระดานที่มีปัญญาประดิษฐ์ที่มีความสามารถเหมาะสมกับผู้เล่น

## อุปกรณ์และเครื่องมือที่ใช้

1. ฮาร์ดแวร์
 

คอมพิวเตอร์ส่วนตัวที่มีหน่วยประมวลผลกลางความเร็วไม่ต่ำกว่า 3.0 GHz หน่วยความจำหลัก ความจุไม่ต่ำกว่า 16 GB และหน่วยความจำสำรองความจุไม่ต่ำกว่า 1 TB
2. ซอฟต์แวร์
  - 2.1 ระบบปฏิบัติการ Windows
  - 2.2 ไลบรารี Python สำหรับการทำงานและคำนวณข้อมูล เช่น numpy, pandas

## งบประมาณ

- |                                       |            |                        |
|---------------------------------------|------------|------------------------|
| 1. Magic Keyboard with Numeric Keypad | 1 ชิ้น     | 4790 บาท               |
|                                       | <b>รวม</b> | <b><u>4790 บาท</u></b> |

## เอกสารอ้างอิง

- [1] AlphaGo. [<https://deepmind.com/research/case-studies/alphago-the-story-so-far>]. เข้าถึงเมื่อ 26 ตุลาคม พ.ศ. 2563
- [2] Mihaly. Csikszentmihalyi, Flow: The Psychology of Optimal Experience, Harper Row, NewYork, NY, USA, 2009.
- [3] Mohammad Zohaib, "Dynamic Difficulty Adjustment (DDA) in Computer Games", Advances in Human-Computer Interaction Volume 2018, 2018.
- [4] Robin Hunicke and Vernell Chapman, "AI for Dynamic Difficulty Adjustment in Games," in Proceedings of the Challenges in Game Artificial Intelligence AAAIWorkshop, pp. 91–96, San Jose, Calif, USA, 2004.
- [5] Owen Shevlin, "Creating an AI to play Othello", 2017 เข้าถึงเมื่อ 26 ตุลาคม พ.ศ. 2563
- [6] Mini-Max Algorithm in Artificial Intelligence. [<https://www.javatpoint.com/mini-max-algorithm-in-ai>]. เข้าถึงเมื่อ 26 ตุลาคม พ.ศ. 2563

## ประวัติผู้เขียน



นายพีรพล อยู่ไพศาล

รหัสนิสิต 6033643723

วันเดือนปีเกิด 16 เมษายน 2542

ภูมิลำเนา จังหวัดสมุทรปราการ

กำลังศึกษาในสาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย