

## รายงานสรุปการเขียน Final Project โปรแกรมเล่นเกม XO

### 1. ที่มาและความสำคัญ

ในตอนแรกผู้เขียนจะทำการวิเคราะห์ประโยชน์จาก Card Against Humanity และจะสร้าง AI ขึ้นมาเพื่อเล่นเกมนี้ แต่ว่าพบปัญหาเรื่องข้อมูล จะต้องทำข้อมูลขึ้นมาใหม่ ซึ่งใช้เวลานานมาก ผู้เขียนที่สนใจการสร้างโปรแกรม AI ที่เล่นบอร์ดเกม จึงเขียนโปรแกรมเล่นเกมใหม่ คือเกม XO โดยอาศัยการดูข้อมูลจากคลิป “โค้ดชีว : หัดเขียน A.I. แบบ AlphaGo กันแบบชิว” ตามลิงก์ที่แนบมา (<https://www.youtube.com/watch?v=rYMa4qXhzbk&t=4343s>) โดยในคลิปนี้สอนการใช้ Machine Learning แบบ Reinforce ซึ่งไม่ได้เรียนมาในห้อง ผู้เขียนจึงลองเขียนโค้ดง่ายๆตามระบบคิดในคลิป โดยไม่ใช้ Machine Learning แต่ใช้เป็นเขียนกฎ และเรียงลำดับความสำคัญ ใส่เช็คไปที่ละกฎ ถ้าลงได้ก็ให้ลงเลย โปรแกรมนี้ก็จะเป็นพื้นฐานการจัดระบบคิดของคอมพิวเตอร์ ซึ่งเหตุผลที่เลือก xo เพราะ 1. มีทางเลือกน้อย ไม่ต้องเขียนเงื่อนไขยาวมาก 2. เป็นเกมที่มีกติกาการชนะ (win condition) ง่ายไม่ซับซ้อน

### 2. วิธีการดำเนินการ

ผู้เขียนเขียนฟังก์ชัน สำหรับเช็ค win condition โดยจะกำหนดในตาของผู้เล่น รับ input จากผู้เล่นว่าจะลงช่องไหน และให้ลงเครื่องหมายหนึ่งช่อง จากนั้นใช้ฟังก์ชันสำหรับเช็ค win condition ซึ่งถ้าใช่ ผู้เล่นก็ชนะและจบเกม ถ้าไม่ใช่ก็จะเป็นตาของคอมพิวเตอร์ โดยให้คอมพิวเตอร์เลือกลงหนึ่งช่อง โดยประเมินตามกฎที่ผู้เล่นเขียน ผู้เขียนจึงสร้างกฎพื้นฐาน โดยใช้คอมสร้างก๊อปปี้ของตาราง แล้วเอาไปประเมินดูว่าเข้าเงื่อนไขไหนก่อน โดยให้คอมพิวเตอร์ เช็คทีละเงื่อนไข ดังนี้

ลงแล้วชนะ --> ลงกันไม่ให้อีกฝ่ายชนะ --> ลงตรงกลาง --> สุ่มลงตรงมุม --> และสุ่มลงข้างๆ

ซึ่งถ้าตาต่อไปมีช่องที่ลงแล้วชนะ (ลงแล้วทำให้เข้ากับฟังก์ชัน win condition) ให้ลงเลย ถ้าไม่มี ให้ดูกฎข้อต่อไป คือ ถ้าอีกฝั่งจะลงแล้วชนะ ให้ลงกัน แต่ถ้าไม่มีก็ ให้เลือกลงตรงกลาง แล้วค่อยลงมุม และค่อยลง ด้านข้าง หลังจากลงแล้ว ก็ใช้ฟังก์ชัน สำหรับเช็ค win condition ซึ่งถ้าใช้เกมก็จะจบและคอมพิวเตอร์เป็นฝ่ายชนะ ถ้าไม่ใช่ก็จะกลับไปตาผู้เล่น และวนแบบนี้ไปเรื่อยๆ ซึ่งผู้เขียนยังใช้ฟังก์ชันเพื่อเช็ค ว่ากระดานเต็มแล้วหรือไม่ ซึ่งถ้าเต็มก็จะจบเกม และผลเป็นเสมอ

### 3. ผลการดำเนินการ

การดำเนินการใช้งานได้ แต่ผลลัพธ์การเล่นค่อนข้างอ่อน ถ้าผู้เล่นลองลงเล่นมั่วๆ คอมพิวเตอร์สามารถชนะได้บ้าง แต่เมื่อเล่นกับคนเล่นเป็นสถิติการแพ้สูง (เกือบจะร้อยเปอร์เซ็นต์) คอมพิวเตอร์ เนื่องจากโปรแกรมที่เขียนมา คอมพิวเตอร์คิดได้แค่ขั้นเดียว (คิดได้แค่ตาถัดไปเท่านั้น) ไม่สามารถอ่านเกมได้สองหรือสามขั้น ทำให้ผู้เล่นสามารถลงแบบ ดักสองทางเพื่อชนะได้ และจะทำให้คอมพิวเตอร์ไม่สามารถกันได้ ซึ่งถือว่าเป็นการเริ่มต้นที่ดีในการไปเขียนโปรแกรมเพื่อเล่นเกมให้ คอมพิวเตอร์คิดได้ซับซ้อนมากขึ้น และทำให้เข้าใจคอนเซปของการใช้ Machine learning มากขึ้น

### 4. ปัญหาและอุปสรรค

ในตอนแรกจะทำให้คอมพิวเตอร์คิดสองขั้นได้ แต่ต้องเขียนวนวายมาก จึงล้มเลิกไป ซึ่งการล้มเหลวนี้ก็ทำให้เข้าใจถึงประโยชน์ของ machine learning และเข้าใจระบบมันมากขึ้น

ปัญหาต่อมาคือเรื่อง ประเภทของข้อมูล และชื่อตัวแปร เนื่องจากการเรียกใช้หลายฟังก์ชัน ทำให้สับสนเรื่องประเภทข้อมูล และชื่อตัวแปร ต้องคอยติดตามตลอด ถ้าเป็นโค้ดที่สมบูรณ์ควรจัดระบบฟังก์ชันให้ดีกว่านี้ ใช้class เข้ามาช่วย

#### 5. สรุปผล และประเมิน

ถึงแม้ว่าโค้ดจะไม่ได้ยาก และประสิทธิภาพของโปรแกรมจะไม่สูงมาก (เล่นแป๊บเดียว) แต่สิ่งที่ได้จากการจัดระบบคิดให้คอมพิวเตอร์ ในตอนทำ machine learning ของ PA ผู้เขียนจะไม่เข้าใจระบบอย่างชัดเจน แต่การมาทำโปรเจกต์นี้ทำให้เห็นภาพได้มากขึ้น ซึ่งเป็นการต่อยอดการใช้ machine learning ได้ดี