

ลำดับการทำงาน

1. Check turn ของฝั่งเราว่า ณ ตอนนั้นเป็น turn หรือว่าเป็น turn ของฝั่งตรงข้าม ถ้าเป็นฝั่งเราให้ทำการ move or pass หรือถ้าเป็นฝั่งตรงข้ามให้ทำการรอจนกว่าฝั่งตรงข้ามจะเดินจบและวนกลับมาที่ turn ของเรา
 2. หลังจากที่เรา check turn แล้วถ้าเป็น turn ของเราให้ทำการ moveable ของหมากแต่ละตัวของเราออกมาเป็น filed(json) จาก board stage of my number
if piece['Owner'] == player_number:
 3. เมื่อเราทำการดึง moveable ของเราออกมาแล้วเมื่อถึง turn ของฝั่งตรงข้าม ให้ทำการดึง moveable ของฝั่งตรงข้ามออกทั้งหมดจาก board stage
if piece['Owner'] != player_number:
 4. ทำการหา priority ของหมากฝ่ายเราทั้งหมด
โดยจะหาเรียงหมากไปที่ละตัวจาก movable field ใน ฟังก์ชัน process_board
-

การใช้ทีแต่ละตัว

5. Pawn

โดย PAWN default priority = 5

โดยเราจะทำการสร้าง List เพื่อเก็บการเดินทั้งหมดของหมากบนบอร์ด จาก from field to field ที่จะเดินด้วย list moves โดยจะ set ค่า priority ก็นี้

```
pawn_values = {"pawn": 50, "queen": 120, "knight": 110, "rook": 110, "bishop": 105, "king": 150}
```

หลังจากที่เช็คค่าแล้วจะเป็นการไม่ให้หมากของฝั่งเรานั้นเดินไปกินฝั่งตรงข้ามมากจนเกินไปด้วยเราทำการกำหนด หากหมากเดินไปสุดทางของแต่ละฝั่งแล้ว ให้ทำการหัก priority การเดินนั้นๆ ทิ้ง โดยในส่วนนี้ยังไม่สามารถเดินไปยังฝั่งตรงข้ามได้ เราจึงกำหนดค่าให้เดินทะลุเส้นขอบไปยังฝั่งตรงข้ามได้ดังนี้

```
player1_beware = ['GA4', 'GB4', 'GC4', 'GD4', 'GE4', 'GF4', 'GG4', 'GH4', 'RH4', 'RG4', 'RF4', 'RE4', 'RD4', 'RC4', 'RB4', 'RA4']  
player2_beware = ['BA4', 'BB4', 'BC4', 'BD4', 'BE4', 'BF4', 'BG4', 'BH4', 'RH4', 'RG4', 'RF4', 'RE4', 'RD4', 'RC4', 'RB4', 'RA4']  
player3_beware = ['GA4', 'GB4', 'GC4', 'GD4', 'GE4', 'GF4', 'GG4', 'GH4', 'BA4', 'BB4', 'BC4', 'BD4', 'BE4', 'BF4', 'BG4', 'BH4']  
beware_list = player1_beware if player_number == "Player1" else  
(player2_beware if player_number == "Player2" else player3_beware)
```

หลังจากนั้นเราจะทำการเช็คตำแหน่งของการเดินของหมากเราว่า จะเดินไปตรงที่หมากของฝ่ายตรงข้ามเดินไปหรือป่าว โดยเราจะเช็คส่วนนี้ได้จากการดึง enemy move ในข้อที่ (3)

หลังจากนั้นแล้วทำการเช็คต่อว่า ตำแหน่งที่หมากเราเดินไปนั้น หากหมากของฝั่งตรงข้ามเดินได้ดึง priority จาก price_values ในข้อ (5)โดยให้ ความสำคัญตามนี้จากมากไปน้อย

```
king > queen > rook > knight> bishop > pawn
```

จากนั้นให้ทำการเช็คค่าว่า หาก moveableของหมากเรานั้น ไปตรงเส้นขอบของฝั่งตรงข้ามแล้วให้หัก priority

6. Rook

โดย Rook default priority = 4

โดยเราจะทำการสร้าง List เพื่อเก็บการเดินทั้งหมดของหมากบนบอร์ด จาก from field to field ที่จะเดินด้วย list moves โดยจะ set ค่า priority กัน ดังนี้

```
piece_values = {"pawn": 2, "queen": 115, "knight": 110, "rook": 110, "bishop": 100, "king": 150}
```

โดยเราจะให้ Rook หากเจอ pawn ค่า priority = จะเท่ากับ pawn เพื่อให้ไปสู้ร่วมกับ pawn ตอนเดินปกติเพื่อไม่ให้ Rook ไปกิน pawn ก่อนแล้วเสียหมากไปให้กับ rook ฝ่ายตรงข้าม แต่ที่ทำแบบนี้เพราะจะรอให้ Pawn เราเหลือน้อยๆก่อนถึงจะมีโอกาสเดินมากขึ้น

ถ้าฝ่ายตรงข้าม สามารถกิน rook ได้ แล้ว rook เราก็กินได้ จะทำการ Counter attack ฝ่ายตรงข้าม โดยจะเซต Priority เป็น +=100 เพื่อให้ Counter attack ก่อน

แต่หากเดินไปจริงที่ฝ่ายตรงข้ามเดินได้จะหัก Priority -=100

หลังจากนั้นแล้วทำการเช็คต่อว่า ตำแหน่งที่หมากเราเดินไปนั้น หากหมากของฝั่งตรงข้าม โดยจะให้กินอย่างเดียวโดยไม่สนใจอะไรเลย โดยลำดับความสำคัญการกินแบบนี้

```
King > Queen > Knight = Rook > Bishop > Pawn
```

แต่หากฝ่ายตรงข้ามกินเราได้ แต่เรากินตัวไหนไม่ได้เลยจะทำการเซต priority = 50 เพื่อให้หนี

7. Knight default priority = 4

โดยเราจะทำการสร้าง List เพื่อเก็บการเดินทั้งหมดของหมากบนบอร์ด จาก from field to field ที่จะเดินด้วย list moves โดยจะ set ค่า priority ก็นี้

```
piece_values = {"pawn": 5, "queen": 98, "knight": 12, "rook": 12, "bishop": 10, "king": 150}
```

อันดับแรกจะเช็คหมากที่กินฝ่ายตรงข้ามได้

หากหมากฝ่ายตรงข้ามกินเราได้ และเรากินฝ่ายตรงข้ามได้เช่นกัน จะเช็ค

Priority = 100 + piece_values เพื่อให้ Counter attack ก่อน

แต่หากหมากเรากินได้แต่ไม่อยู่ใน enemy movable จะทำการ get priority จาก

piece_values ตามชื่อหมาก

แต่หาก กินตัวไหนไม่ได้เลยจะทำการ คิด priority แบบหากตำแหน่งเดินอยู่ใน enemy

move จะทำการ -priority ด้วย 100 ตามนี้

```
priority = 4 - 100 * any(move_square in enemy_moves for enemy_moves in enemy_movable_fields.values())
```

แต่หากฝ่ายตรงข้ามกินเราได้ แต่เรากินตัวไหนไม่ได้เลยจะทำการเช็ค priority = 50 จุด

ที่ปลอดภัยเพื่อให้หนี

8. Bishop default priority = 3

โดยเราจะทำการสร้าง List เพื่อเก็บการเดินทั้งหมดของหมากบนบอร์ด จาก from field to field ที่จะเดินด้วย list moves โดยจะ set ค่า priority กัน ดังนี้

```
piece_values = {"pawn": 4, "queen": 99, "knight": 12, "rook": 13, "bishop": 12, "king": 150}
```

อันดับแรกจะเช็คหมากที่กินฝ่ายตรงข้ามได้

หากหมากฝ่ายตรงข้ามกินเราได้ และเรากินฝ่ายตรงข้ามได้เช่นกัน จะเช็ค

Priority = 100 + piece_values เพื่อให้ Counter attack ก่อน

แต่หากหมากเรากินได้แต่ไม่อยู่ใน enemy movable จะทำการ get priority จาก

piece_values ตามชื่อหมาก

แต่หาก กินตัวไหนไม่ได้เลยจะทำการ คัด priority แบบหากตำแหน่งเดินอยู่ใน enemy

move จะทำการ -priority ด้วย 100 ตามนี้

```
priority = 3 - 100 * any(move_square in enemy_moves for enemy_moves in enemy_movable_fields.values())
```

แต่หากฝ่ายตรงข้ามกินเราได้ แต่เรากินตัวไหนไม่ได้เลยจะไม่หนีกันกรณีมีตัวมาไล่แล้วหากหลบไปมีโอกาสดูโดนกินฟรี

9. Queen default priority = 2

โดยเราจะทำการสร้าง List เพื่อเก็บการเดินทั้งหมดของหมากบนบอร์ด จาก from field to field ที่จะเดินด้วย list moves โดยจะ set ค่า priority ก็นี้ ดังนี้

```
piece_values = {"pawn": 1, "queen": 94, "knight": 10, "rook": 80, "bishop": 10, "king": 150}
```

โดย Queen เราจะเช็คเลยว่าไม่ให้กิน Pawn เด็ดขาดกันไปกิน แล้วเสียหมากฟรี แต่จะเช็คทำให้ เน้นกิน Rook เป็นพิเศษกันโดนได้

อันดับแรกจะเช็คหมากที่กินฝ่ายตรงข้ามได้

หากหมากฝ่ายตรงข้ามกินเราได้ และเรากินฝ่ายตรงข้ามได้เช่นกัน จะเช็ค

Priority = 200 + piece_values (เช็คมากกว่า Knight bishop rook เพื่อให้หนีก่อนหมู)

แต่หากหมากเรากินได้แต่ไม่อยู่ใน enemy movable จะทำการ get priority จาก

piece_values ตามชื่อหมาก

แต่หาก กินตัวไหนไม่ได้เลยจะทำการ คัด priority แบบหากตำแหน่งเดินอยู่ใน enemy

move จะทำการ -priority ด้วย 100 ตามนี้

```
priority = 2 - 100 * any(move_square in enemy_moves for enemy_moves in enemy_movable_fields.values())
```

แต่หากฝ่ายตรงข้ามกินเราได้ แต่เรากินตัวไหนไม่ได้เลยจะทำการเช็ค priority = 60

เพื่อให้หนี

10. King default priority = -10000

โดยเราจะทำการสร้าง List เพื่อเก็บการเดินทั้งหมดของหมากบนบอร์ด จาก from field to field ที่จะเดินด้วย list moves โดยจะ set ค่า priority ก็นี

```
piece_values = {"pawn": 500, "queen": 500, "knight": 500, "rook": 500,
                "bishop": 500, "king": 500}
```

King เราจะ ไม่เดินเลยหากไม่มีฝ่ายตรงข้ามกินเราได้ หรือมีฝ่ายตรงข้ามมาอยู่หน้าเราด้วยการเช็ค Priority ต่ำๆ

อันดับแรกจะเช็คหมากที่กินฝ่ายตรงข้ามได้

หากหมากฝ่ายตรงข้ามกินเราได้ และเรากินฝ่ายตรงข้ามได้เช่นกัน จะเช็ค

Priority = 1000 + piece_values (เช็คมากกว่าทุกตัวเพื่อกันคิงไม่ให้แพ้ฟรีๆ)

หากกินได้แต่ฝ่ายตรงข้ามกินเราไม่ได้ก็จะดึงค่า Priority จาก piece_values มา

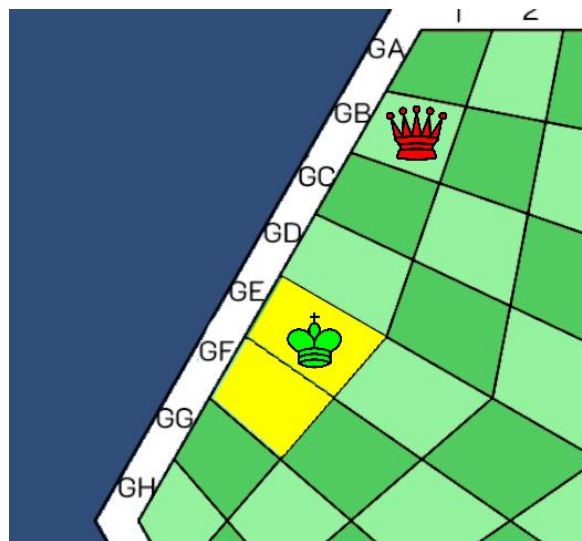
แต่หาก กินตัวไหนไม่ได้เลยจะทำการ คัด priority แบบหากตำแหน่งเดินอยู่ใน enemy move จะทำการ -priority ด้วย 100 ตามนี้

```
priority = -10000
```

11.พอได้ทั้ง Priority มาทั้งหมดแล้ว อันดับแรกเราจะ CheckKing ก่อนว่าปลอดภัยไหม

หากปลอดภัยจะข้ามไปในข้อ (12) แต่หากไม่ปลอดภัยจะ หาตำแหน่งที่ปลอดภัยโดยการดูจาก enemy move ณ ตอนนั้นว่าเดินไปช่องไหนได้บ้าง และเอามาเทียบกับ King หาก ตำแหน่งที่ King เดินได้ ศัตรูกินไม่ได้จะเดินไปช่องนั้น แล้วทำการสุ่มอีกทีใน list ที่เดินได้

แต่จะมีกรณี อย่างหากเดินไปในช่องที่ศัตรู ตอนแรกไปไม่ได้ แต่เราเดินไปแล้วศัตรูก็ยังเดินได้ก็จะแพ้ไปแบบฟรีๆ แบบนี้



12.หากคิง ปลอดภัยจะทำการ select move จากฟังก์ชัน

`select_move_from_processed_board`

จากที่ Max priority มาก แต่หาก Max priority เท่ากันจะทำการสุ่ม move

13. หากหมากเข้าสู่ Promote จะปรับเป็น Queen อัตโนมัติ

14. หากไม่มีที่เดินแล้วจะ Pass ทันที

ภาพรวมคะแนน ที่ให้ Priority จากการที่หมากไปกินฝ่ายตรงข้ามได้

```
pawn_values = {"pawn": 50, "queen": 120, "knight": 110, "rook": 110, "bishop": 105, "king": 150}
rook_values = {"pawn": 2, "queen": 115, "knight": 100, "rook": 110, "bishop": 100, "king": 150}
knight_values = {"pawn": 5, "queen": 98, "knight": 12, "rook": 12, "bishop": 10, "king": 150}
bishop_values = {"pawn": 4, "queen": 99, "knight": 12, "rook": 13, "bishop": 12, "king": 150}
queen_values = {"pawn": 1, "queen": 94, "knight": 10, "rook": 80, "bishop": 10, "king": 150}
king_values = [{"pawn": 500, "queen": 500, "knight": 500, "rook": 500, "bishop": 500, "king": 500}]
```

ภาพรวมเลยหากจะเรียงการที่มีโอกาสส่งไปกินมากที่สุด หาก สามารถกินได้หากทุกหมากกิน
ได้หมด

Pawn > Rook > Bishop > Knight > Queen > King