

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

-----\*\*\*-----



**BÁO CÁO BÀI TẬP GIỮA KỲ TRÍ TUỆ NHÂN TẠO**

**CHESS PROGRAM**

**Lớp học phần: INT3401E 5**

**Giảng viên: TS. Nguyễn Văn Vinh**

**Sinh viên: Nguyễn Đình Tuấn Anh - 22028136**

**HANOI - 2024**

# MỤC LỤC

<b>1. Giới thiệu.....</b>	<b>3</b>
<b>2. Các kỹ thuật sử dụng.....</b>	<b>3</b>
2.1. Đánh giá vị trí.....	3
2.2. Cơ chế tạo nước.....	4
2.3. Thuật toán tìm kiếm.....	4
2.4. Giới hạn thời gian mỗi nước đi.....	5
2.5. Sử dụng opening book.....	5
<b>3. Xây dựng chương trình.....</b>	<b>5</b>

# 1. Giới thiệu

Yêu cầu bài toán: Xây dựng một phần mềm cờ vua có 2 chế độ, người với máy và máy với máy, với elo của bot đạt khoảng 2200 với thời gian phản hồi dưới 10s, có GUI hiển thị. Khuyến khích dùng các kỹ thuật như Alpha-Beta, reinforcement learning.

Giải pháp: Xây dựng 1 engine sử dụng nền là alpha-beta và tối ưu, sau đó xây dựng giao diện

## 2. Các kỹ thuật sử dụng

### 2.1. Đánh giá vị trí

Lập bảng điểm đánh giá độ quan trọng của các quân cờ và ma trận đánh giá điểm vị trí của từng loại quân cờ. Đây là nền tảng để đánh giá điểm số của mỗi nước đi, thuận tiện cho việc sử dụng thuật toán tìm kiếm Alpha - Beta sau này.

```
piece = {"P": 100, "N": 280, "B": 320, "R": 479, "Q": 929, "K": 60000}
```

*Điểm đánh giá độ quan trọng của từng quân cờ*

'N':	(	-65,	-55,	-55,	-10,	-10,	-55,	-55,	-65,
		-5,	-5,	60,	0,	0,	60,	-5,	-5,
		10,	65,	15,	75,	75,	15,	65,	10,
		20,	25,	40,	35,	35,	40,	25,	20,
		0,	5,	30,	20,	20,	30,	5,	0,
		-15,	10,	15,	20,	20,	15,	10,	-15,
		-20,	-15,	0,	0,	0,	0,	-15,	-20,
		-70,	-25,	-25,	-20,	-20,	-25,	-25,	-70),

*Điểm đánh giá vị trí của quân mã*

## 2.2. Cơ chế tạo nước

```
directions = {  
    "P": (N, N+N, N+W, N+E),  
    "N": (N+N+E, E+N+E, E+S+E, S+S+E, S+S+W, W+S+W, W+N+W, N+N+W),  
    "B": (N+E, S+E, S+W, N+W),  
    "R": (N, E, S, W),  
    "Q": (N, E, S, W, N+E, S+E, S+W, N+W),  
    "K": (N, E, S, W, N+E, S+E, S+W, N+W)  
}
```

Để tối ưu cho bảng vị trí, engine có cơ chế xoay chiều để luôn nhìn bàn cờ từ vị trí trắng, tính toán việc xoay ở phần sau.

## 2.3. Thuật toán tìm kiếm

Thuật toán được sử dụng là một biến thể của Alpha - Beta, tương ứng với thuật toán Negamax.

Engine đánh giá từng trạng thái bàn cờ bằng một hàm đánh giá, sử dụng điểm số đánh giá giá trị quân cờ và vị trí ở trên, trả về giá trị biểu thị mức độ lợi thế của một bên (dương nếu trắng có lợi, âm nếu đen có lợi). Các nước đi cũng được sắp xếp theo giá trị tiềm năng để tối ưu cho việc cắt tỉa.

```
def search(self, history):  
    self.nodes = 0  
    self.history = set(history)  
    self.tp_score.clear()  
  
    gamma = 0  
    for depth in range(1, 1000):  
        lower, upper = -MATE_LOWER, MATE_LOWER  
        while lower < upper - EVAL_ROUGHNESS:  
            score = self.bound(history[-1], gamma, depth, can_null=False)  
            if score >= gamma:  
                lower = score  
            if score < gamma:  
                upper = score  
            yield depth, gamma, score, self.tp_move.get(history[-1])  
            gamma = (lower + upper + 1) // 2
```

Với việc sử dụng negamax, alpha và beta được thay thế bằng gamma, phục vụ cho việc cắt tỉa.

```
entry = self.tp_score.get((pos, depth, can_null), Entry(-MATE_UPPER, MATE_UPPER))
if entry.lower >= gamma: return entry.lower
if entry.upper < gamma: return entry.upper
```

Tìm kiếm tĩnh (Quiescent Search) cũng được áp dụng, để tránh horizon effect (cắt đúng trước khi xảy ra ăn quân lớn).

## 2.4. Giới hạn thời gian mỗi nước đi

Trong thuật toán tìm kiếm có sử dụng Iterative Deepening, tìm và trả về nước đi tốt nhất ở mỗi độ sâu, đảm bảo luôn có nước đi tốt nhất có thể trong giới hạn thời gian.

```
for depth, gamma, score, move in searcher.search([self.position]):
    if timer_expired[0] == True:
        print("Stop searching")
        break
```

## 2.5. Sử dụng opening book

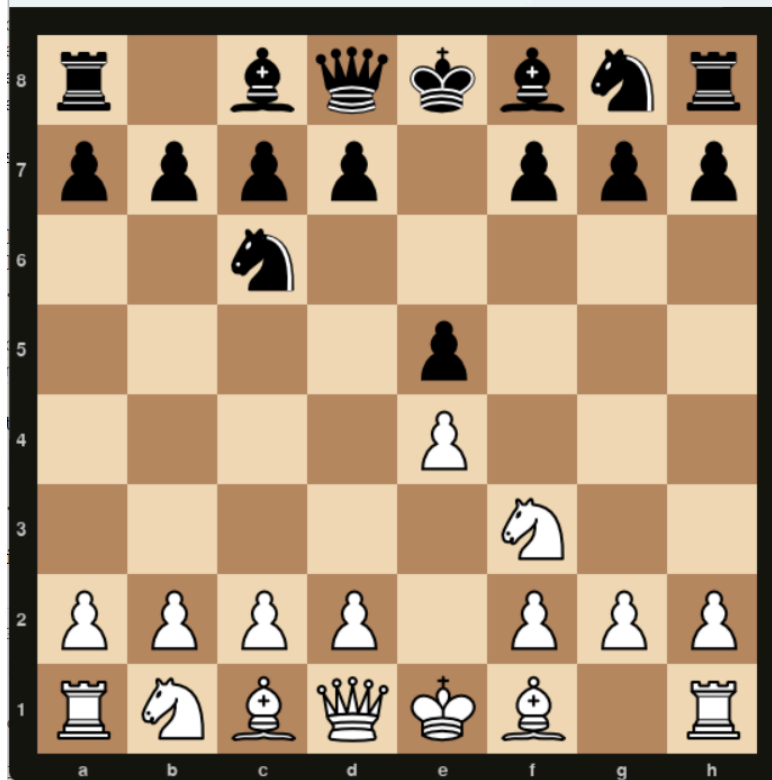
Opening Book chứa các nước đi khai cuộc đã được nghiên cứu và đánh giá từ các ván đấu của kỳ thủ chuyên nghiệp. Tích hợp thêm opening book giúp engine thực hiện tốt các nước đi đầu tiên và tăng độ chính xác khi sử dụng các khai cuộc đã được kiểm chứng. Ở đây, opening book được sử dụng là Titans, một opening book phổ biến.

# 3. Xây dựng chương trình

Chương trình được xây dựng theo yêu cầu, gồm 2 chế độ người với máy và máy với máy. Với nền tảng đồ họa sử dụng pygame, chương trình sẽ quản lý luật chơi, xác định nước đi hợp lệ, xác định chiếu tướng hòa cờ.

- **Chế độ người với máy:**

Được thiết lập với Trắng là người chơi, Đen là máy.



- **Chế độ máy với máy:**

Sử dụng Auto Play để kích hoạt việc chơi cờ giữa 2 máy. Có phân chia giới hạn thời gian cho từng bên.



Kết quả, engine được xây dựng đảm bảo giới hạn thời gian phản hồi dưới 10s, tuy nhiên do thời gian hạn chế và làm dự án cá nhân, Engine chưa thể đảm bảo có Elo > 2200, ước tính khoảng 1800.