

Báo cáo về thời gian chạy và không gian bộ nhớ của thuật toán Alpha-Beta và MiniMax

Nhóm: Đoàn Quốc Đăng và Nguyễn Phương Nhã

Đề tài: Game cờ đam (Checkers)

I. Sơ lược về báo cáo:

- Kết quả thí nghiệm độ phức tạp về thời gian và không gian bộ nhớ được thực hiện ở cả thuật toán MiniMax và Alpha-Beta ở độ sâu (depth) từ 1 đến 7.
- Cấu hình thử nghiệm:
 - o Processor: Intel® Pentium® Gold G6405 2 Cores 4 Threads 4.1Ghz.
 - o RAM: 16GB DDR4.
 - o OS: Windows 11 Pro version 23H2.
 - o IDE: IntelliJ IDEA 2023.2.5.
- Đoạn code thực hiện việc đo đạc thời gian chạy và tiêu thụ bộ nhớ:

```
long begin = System.currentTimeMillis();
long startMemory = Runtime.getRuntime().totalMemory() - Runtime.getRuntime().freeMemory();

CheckerBoard bestMove = ai.getBestMove(checkerBoardController.getCheckerBoard(), gameDifficult);
checkerBoardController.setCheckerBoard(bestMove);

long end = System.currentTimeMillis();
long endMemory = Runtime.getRuntime().totalMemory() - Runtime.getRuntime().freeMemory();

System.out.println("Depth: " + gameDifficult.getDepth());
System.out.println("Memory used: " + (endMemory - startMemory) + " bytes ~ " + (endMemory - startMemory) / 1024 + " KB");
System.out.println("Computed time: " + (end - begin) + "ms");
isBlackTurn = false;
```

- Đoạn code nằm ở phương thức **startGame()** trong lớp **GameController** ở đường dẫn: **com/dangnha/checkers/controller/GameController.java**.
- Trong đó phương thức **ai.getBestMove(CheckerBoard board, GameDifficult gameDifficult)** gọi lại thuật toán Alpha-Beta để tìm ra bàn cờ tối ưu nhất.

- Do game sẽ tự bắt đầu ở mức **GameDifficult** là **EASY** nên chỉ cần thay đổi giá trị độ sâu của **GameDifficult.EASY** để test:

```

ConMuaXaDan *
public enum GameDifficult {
    1 usage
    HARD( depth: 7),
    1 usage
    MEDIUM( depth: 5),
    3 usages
    EASY( depth: 7);

    2 usages
    private int depth;
    6 usages ConMuaXaDan
    private GameDifficult(int depth) { this.depth = depth; }

    2 usages ConMuaXaDan
    public int getDepth() { return depth; }
}

```

- Thuật toán Alpha-Beta: gồm 2 phương thức **minValue()** và **maxValue()**

```

1 usage ConMuaXaDan +1 *
public int minValue(CheckerBoard board, int alpha, int beta, int depth) {
    if (depth == 0) {
        return board.heuristic();
    }

    CheckerBoard cloneBoard = new CheckerBoard(board.getBoardStates(), board.getCheckerList());

    int value = Integer.MAX_VALUE;
    for (CheckerBoard neighbour : cloneBoard.generateNeighbours( isBlackTurn: false)) {
        value = Math.min(value, maxValue(neighbour, alpha, beta, depth: depth - 1));
        if (alpha >= value) return value;
        beta = Math.min(beta, value);
    }
    return value;
}

```

```

3 usages ConMuaXaDan +1 *
public int maxValue(CheckerBoard board, int alpha, int beta, int depth) {
    if (depth == 0)
        return board.heuristic();

    CheckerBoard cloneBoard = new CheckerBoard(board.getBoardStates(), board.getCheckerList());

    int value = Integer.MIN_VALUE;
    for (CheckerBoard neighbour : cloneBoard.generateNeighbours( isBlackTurn: true)) {
        value = Math.max(value, minValue(neighbour, alpha, beta, depth: depth - 1));
        if (value >= beta) return value;
        alpha = Math.max(alpha, value);
    }
    return value;
}

```

- Thuật toán MiniMax sẽ sử dụng 2 phương thức trên nhưng sẽ comment lại 2 dòng sau dòng if ở 2 phương thức **minValue()** và **maxValue()**

II. Kết quả thực nghiệm:

Độ sâu (Depth)	MiniMax	Alpha-Beta
1	<pre>Depth: 1 Memory used: 2082728 bytes ~ 2033 KB Computed time: 58ms</pre>	<pre>Depth: 1 Memory used: 2050760 bytes ~ 2002 KB Computed time: 37ms</pre>
2	<pre>Depth: 2 Memory used: 7246216 bytes ~ 7076 KB Computed time: 64ms</pre>	<pre>Depth: 2 Memory used: 6291432 bytes ~ 6143 KB Computed time: 54ms</pre>
3	<pre>Depth: 3 Memory used: 49915808 bytes ~ 48745 KB Computed time: 121ms</pre>	<pre>Depth: 3 Memory used: 23572832 bytes ~ 23020 KB Computed time: 88ms</pre>
4	<pre>Depth: 4 Memory used: 81778864 bytes ~ 79862 KB Computed time: 359ms</pre>	<pre>Depth: 4 Memory used: 69955408 bytes ~ 68315 KB Computed time: 144ms</pre>
5	<pre>Depth: 5 Memory used: 285943648 bytes ~ 279241 KB Computed time: 1744ms</pre>	<pre>Depth: 5 Memory used: 55686792 bytes ~ 54381 KB Computed time: 269ms</pre>

6	<div>Depth: 6</div> <div>Memory used: 375193848 bytes ~ 366400 KB</div> <div>Computed time: 10691ms</div>	<div>Depth: 6</div> <div>Memory used: 112770128 bytes ~ 110127 KB</div> <div>Computed time: 431ms</div>
7	<div>Depth: 7</div> <div>Memory used: 399554160 bytes ~ 390189 KB</div> <div>Computed time: 78285ms</div>	<div>Depth: 7</div> <div>Memory used: 234188264 bytes ~ 228699 KB</div> <div>Computed time: 932ms</div>