



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
1^η Εργασία Τεχνητής Νοημοσύνης
Ακαδημαϊκό Έτος 2022-2023
Μαρία Κονταράτου – 3200078
Γεώργιος Κουμουνδούρος- 3200083

1.1 Εκτέλεση προγράμματος

Στο Command Line εκτελούμε τις εντολές:

```
-javac Main.java  
-java Main
```

1.2 Τεχνικές που χρησιμοποιήθηκαν

Όπως ζητήθηκε και στην εκφώνηση, χρησιμοποιήθηκε ο αλγόριθμος MiniMax με πριόνισμα α-β. Η ευρετική συνάρτηση που χρησιμοποιήσαμε ουσιαστικά έδινε πόντους ανάλογους με την εκάστοτε θέση του παίκτη. Αυτό σημαίνει ότι αν ο παίκτης βρισκόταν στις γωνίες του ταμπλό λάμβανε περισσότερους πόντους, λιγότερους αν βρισκόταν στις πλευρές και ακόμα λιγότερες στις υπολοιπούσες θέσεις.

Βασίσαμε τον κώδικά μας στο 3^ο εργαστήριο, δημιουργώντας 4 classes: Board, Player, Move και Main. Οι Player και Move χρησιμοποιήθηκαν σχεδόν αυτούσιες.

Στην Board.java, έχουμε τους βασικούς constructors, getters και setters, την print() που τυπώνει την κατάσταση του πίνακα και την printResult() η οποία βάσει της τελικής κατάστασης του πίνακα εκτυπώνει το αποτέλεσμα.

Ως προς την makeMove(), έχει σκοπό να τοποθετήσει ανάλογα το ποιος παίζει στο ταμπλό το αντίστοιχο γράμμα. Κάθε φορά που καλείται, θα χρειάζεται να αδειάζουμε τη λίστα με τις πιθανές κινήσεις (του προηγούμενου παίκτη) που έχουμε δημιουργήσει και αν η θέση είναι πράγματι έγκυρη για κάθε κίνηση, καλούμε την createLine(). Στην createLine(), η κάθε κίνηση έστω v εάν παρουσιάζει ήδη το αντίστοιχο γράμμα απλά δεν επιστρέφουμε κάτι, ενώ σε άλλη περίπτωση πρέπει να αλλάξουμε με το αντίστοιχο του παίκτη και να την ξανακαλέσουμε αναδρομικά στην Delta Direction. Η deltaR είναι η αύξηση κατά 1 row γύρω από ένα τετράγωνο και αντίστοιχα η deltaC η αύξηση κατά 1 column γύρω από το τετράγωνο. Έτσι, μέσω της αναδρομής μπορούμε να ακολουθήσουμε την κίνηση είτε αυτή είναι διαγώνια είτε οριζόντια είτε κάθετη.

Στην isValid(), έχουμε 4 περιπτώσεις που κάποια κίνηση δεν είναι έγκυρη:

- (α) Αν το τετράγωνο είναι κενό γιατί δεν θα αποτελεί κίνηση
- (β) Αν βρίσκεται εκτός ορίων του ταμπλό
- (γ) Αν δεν υπάρχουν «γείτονες» αντίθετου γράμματος: Χρησιμοποιούμε μια λίστα opposites στην οποία τοποθετούμε όλα τα γειτονικά τετράγωνα που έχουν αντίθετο γράμμα. Ελέγχουμε ένα-ένα πλαίσιο γύρω από την θέση που θέλουμε να πάμε. Αν τελικά η λίστα μας είναι άδεια, τότε δεν υπάρχουν τέτοιοι γείτονες και επιστρέφουμε false

(δ) Αν δεν υπάρχει γραμμή μεταξύ του γράμματος που θα προσθέσουμε και ενός υπάρχοντος ίδιου χρώματος στην ίδια γραμμή. Έχουμε από το (γ) τη λίστα με τα opposites στην οποία με παρόμοια λογική χρησιμοποιώντας delta directions και την Line() ελέγχουμε εάν υπάρχει η προαναφερθείσα νοητή γραμμή.

Για την ευρετική evaluate(), δίνουμε περισσότερους πόντους στον παίκτη που έχει γράμμα στις γωνίες, λιγότερους στις πλευρές και ελάχιστες στις υπόλοιπες θέσεις του ταμπλό

Στην Main(), σύμφωνα και με την εκφώνηση της εργασίας, σε πρώτο στάδιο ζητάμε από τον χρήστη να μας δηλώσει αν θέλει να παίξει πρώτος ή όχι καθώς και το μέγιστο βάθος αναζήτησης και ελέγχουμε αν είναι valid ή όχι. Στη συνέχεια, αρχικοποιούμε τους παίκτες, το ταμπλό και το εμφανίζουμε στον χρήστη.

Ακολουθήσαμε το πρότυπο του εργαστηρίου και χτίσαμε το πρόγραμμα με βάση 3 cases, βάση του ποιος έπαιξε τελευταίος: το Board.X (που έπαιξε τελευταίος ο X και τώρα παίζει ο O), το Board.O (που έπαιξε τελευταίος ο O και τώρα παίζει ο X) και το default.

Board.X:

Δημιουργούμε τη λίστα moves στην οποία τοποθετούμε όλες τις έγκυρες κινήσεις που έχει ο χρήστης σύμφωνα με την isValid(). Στη συνέχεια ζητάμε από τον χρήστη τη σειρά και τη στήλη ελέγχοντας πάντα τα στοιχεία του και αν συνάδει κάθε κίνηση απ' τη λίστα moves με το input του. Μετά τον έλεγχο, καλούμε για το ταμπλό την makeMove() – την οποία έχουμε παραλλάξει για να επιστρέφει Boolean τιμές και ανάλογα με το αποτέλεσμα τυπώνουμε εάν έχει ο χρήστης έγκυρες κινήσεις (όπου εμφανίζουμε και μήνυμα με τις συντεταγμένες της) ή όχι.

Board.O:

Εδώ το μόνο που χρειάζεται να κλανουμε για τον υπολογιστή είναι να δημιουργήσουμε μια μεταβλητή moveX που καλεί την MinMaxAB(). Μετά ελέγχουμε πάλι για έγκυρες κινήσεις μέσω της makeMove() όπως στο Board.X.

Σε κάθε περίπτωση μετά από κάθε κίνηση εμφανίζουμε το ταμπλό μέσω της print(). Στο τέλος του παιχνιδιού εκτυπώνουμε τον νικητή.

1.3 Παράδειγμα παρτίδας

```
-----REVERSI-----

You will be playing against a computer

Computer plays with X
Player plays with 0

To play first, enter 1, else enter 2
1
Enter the depth of search:
1
* 1 2 3 4 5 6 7 8 *
1 - - - - - - - *
2 - - - - - - - *
3 - - - - - - - *
4 - - - X 0 - - - *
5 - - - 0 X - - - *
6 - - - - - - - *
7 - - - - - - - *
8 - - - - - - - *
* * * * * * * *

It's player's 0 turn
You can move to (3,4)
You can move to (4,3)
You can move to (5,6)
You can move to (6,5)
Enter the number of row:
3
Enter the number of column:
4
You played: (3,4)
* 1 2 3 4 5 6 7 8 *
1 - - - - - - - *
2 - - - - - - - *
3 - - - 0 - - - *
4 - - - 0 0 - - - *
5 - - - 0 X - - - *
6 - - - - - - - *
7 - - - - - - - *
8 - - - - - - - *
* * * * * * * *

It's player's X turn
Player X played: (3,5)
* 1 2 3 4 5 6 7 8 *
1 - - - - - - - *
2 - - - - - - - *
3 - - - 0 X - - - *
4 - - - 0 X - - - *
5 - - - 0 X - - - *
6 - - - - - - - *
7 - - - - - - - *
8 - - - - - - - *
* * * * * * * *

It's player's 0 turn
You can move to (2,6)
You can move to (3,6)
You can move to (4,6)
You can move to (5,6)
You can move to (6,6)
Enter the number of row:
2
Enter the number of column:
6
You played: (2,6)
* 1 2 3 4 5 6 7 8 *
1 - - - - - - - *
2 - - - - - 0 - - *
```

Εικόνα 1/Στιγμιότυπο αρχικού σταδίου παιχνιδιού

You played: (2,6)
 * 1 2 3 4 5 6 7 8 *
 1 - - - - - - - *
 2 - - - - 0 - - *
 3 - - - 0 0 - - - *
 4 - - - 0 X - - - *
 5 - - - 0 X - - - *
 6 - - - - - - - *
 7 - - - - - - - *
 8 - - - - - - - *
 * * * * * * * * *

It's player's X turn
 Player X played: (6,3)
 * 1 2 3 4 5 6 7 8 *
 1 - - - - - - - *
 2 - - - - 0 - - *
 3 - - - 0 0 - - - *
 4 - - - 0 X - - - *
 5 - - - X X - - - *
 6 - - X - - - - *
 7 - - - - - - - *
 8 - - - - - - - *
 * * * * * * * * *

It's player's 0 turn
 You can move to (4,6)
 You can move to (5,6)
 You can move to (6,4)
 You can move to (6,5)
 You can move to (6,6)
 Enter the number of row:
 4
 Enter the number of column:
 6
 You played: (4,6)
 * 1 2 3 4 5 6 7 8 *
 1 - - - - - - - *
 2 - - - - 0 - - *
 3 - - - 0 0 - - - *
 4 - - - 0 0 0 - - *
 5 - - - X X - - - *
 6 - - X - - - - *
 7 - - - - - - - *
 8 - - - - - - - *
 * * * * * * * * *

It's player's X turn
 Player X played: (2,5)
 * 1 2 3 4 5 6 7 8 *
 1 - - - - - - - *
 2 - - - - X 0 - - *
 3 - - - 0 X - - - *
 4 - - - 0 X 0 - - *
 5 - - - X X - - - *
 6 - - X - - - - *
 7 - - - - - - - *
 8 - - - - - - - *
 * * * * * * * * *

It's player's 0 turn
 You can move to (1,6)
 You can move to (2,4)
 You can move to (3,6)
 You can move to (5,6)
 You can move to (6,4)
 You can move to (6,6)
 Enter the number of row:
 1
 Enter the number of column:
 6
 You played: (1,6)

7 X X 0 0 0 X - - *
 8 X X X X - X X - *
 * * * * * * * * *

It's player's 0 turn
 You can move to (7,7)
 You can move to (7,8)
 You can move to (8,5)
 Enter the number of row:
 7
 Enter the number of column:
 7
 You played: (7,7)
 * 1 2 3 4 5 6 7 8 *
 1 X 0 0 0 0 0 0 0 *
 2 X 0 0 0 0 0 0 0 *
 3 X 0 X 0 0 0 0 0 *
 4 X 0 X X 0 0 0 0 *
 5 X 0 X X 0 0 0 0 *
 6 X 0 X X X 0 0 X *
 7 X X 0 0 0 0 0 - *
 8 X X X X - X X - *
 * * * * * * * * *

It's player's X turn
 Player X played: (8,8)
 * 1 2 3 4 5 6 7 8 *
 1 X 0 0 0 0 0 0 0 *
 2 X 0 0 0 0 0 0 0 *
 3 X 0 X 0 0 0 0 0 *
 4 X 0 X X 0 0 0 0 *
 5 X 0 X X X 0 0 0 *
 6 X 0 X X X X 0 X *
 7 X X 0 0 0 0 X - *
 8 X X X X - X X X *
 * * * * * * * * *

It's player's 0 turn
 You can move to (7,8)
 Enter the number of row:
 7
 Enter the number of column:
 8
 You played: (7,8)
 * 1 2 3 4 5 6 7 8 *
 1 X 0 0 0 0 0 0 0 *
 2 X 0 0 0 0 0 0 0 *
 3 X 0 X 0 0 0 0 0 *
 4 X 0 X X 0 0 0 0 *
 5 X 0 X X X 0 0 0 *
 6 X 0 X X X X 0 0 *
 7 X X 0 0 0 0 0 0 *
 8 X X X X - X X X *
 * * * * * * * * *

It's player's X turn
 Player X played: (8,5)
 * 1 2 3 4 5 6 7 8 *
 1 X 0 0 0 0 0 0 0 *
 2 X 0 0 0 0 0 0 0 *
 3 X 0 X 0 0 0 0 0 *
 4 X 0 X X 0 0 0 0 *
 5 X 0 X X X 0 0 0 *
 6 X 0 X X X X 0 0 *
 7 X X 0 X X 0 0 0 *
 8 X X X X X X X *
 * * * * * * * * *

Player 0, you win!

-----GAME OVER-----