

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Εργασία 1

- ✓ Μάριος Γόρδιος (3180036)
- ✓ Ηλίας-Θεοφάνης Γραββάνης (3200248)

Reversi Αλγόριθμος

Για την υλοποίηση του Reversi χρησιμοποιήθηκε ο Minimax αλγόριθμος. Η υλοποίηση του έγινε με την χρήση των Helpers directory(Move.java, Heuristic.java) και του Models(Board.java, Player.java) και της κλάσης Main.java όπου γίνεται η χρήση των προηγούμενων κλάσεων και τρέχει το παιχνίδι.

Main.java: Στην κλάση αυτή ζητείται από την χρήστη να επιλέξει το μέγιστο βάθος αναζήτησης του αλγορίθμου Minimax και εάν θέλει να παίξει πρώτος. Έπειτα, εμφανίζει τον αρχικό πίνακα και σύμφωνα με τα όσα έχει επιλέξει ο χρήστης είτε εμφανίζει μήνυμα να εισαγάγει την κίνησή του, είτε εμφανίζει ποια κίνηση έπαιξε, έως ότου φτάσει στον τερματισμό. Η κλάση ελέγχει αν η καταχώρηση x,y είναι ορθή, είτε λάθους input του χρήστη(πχ. 10,10) είτε βάλει σε θέση εντός του παιχνιδιού που δεν είναι διαθέσιμο.

Board.java: Εδώ «δημιουργείται» ο πίνακας και γίνονται οι έλεγχοι του παιχνιδιού. Αρχικά τοποθετούνται δύο X και δύο O, σύμφωνα με τους κανόνες του Reversi. Με την getChildren παίρνει τις αποδέκτες κινήσεις και τις προσθέτει σε μια Arraylist (children). Η makeMove παίρνει τις αποδεκτές κινήσεις και βάση του λεκτικού που παίρνει όταν γίνεται στον έλεγχο υλοποιεί την κίνηση. Στην isValidMove γίνεται έλεγχος αν είναι αποδεκτή η κίνηση. Αρχικά, ελέγχει αν η κίνηση είναι στα όρια του gameboard και αν το κελί που επιλέξαμε είναι κενό. Έπειτα ελέγχει αν στα κελιά που ελέγχουμε υπάρχει αντίπαλος και αν υπάρχει αλλάζει την τιμή της μεταβλητής, που έχουμε δημιουργήσει για αυτό τον λόγο, σε true. Η μέθοδος αυτή επιστρέφει τις διαθέσιμες κινήσεις ή το λεκτικό "INVALID" αν βρεθεί σφάλμα. Η acceptableMoves η οποία δημιουργεί μία λίστα με όλες τις διαθέσιμες κινήσεις. Η makeMove είναι η κίνηση του παίκτη, δηλαδή η σειρά και η στήλη στην οποία πάει ο παίκτης. Η countLetters μετράει πόσα X ή O έχουμε ανάλογα με ποιόν παίκτη του έχουμε περάσει σαν όρισμα. Η heuristic μας επιστρέφει την τιμή της ευρετικής. Η συνάρτηση getGameWinner βρίσκει τον νικητή του παιχνιδιού με βάση πόσα X. Η print την εικόνα του gameBoard εκείνη την ώρα.

Player.java: Στην κλάση αυτή υλοποιείται ο αλγόριθμος Minimax. Ο χρήστης με το σύμβολο X μεγιστοποιεί τον αλγόριθμό ενώ ο χρήστης με το O τον ελαχιστοποιεί. Και στις δύο

περιπτώσεις ελέγχει αν η κίνηση που παίχτηκε είναι τερματική ή αν έχει φτάσει στο μέγιστο βάθος αναζήτησης σύμφωνα με όσα έχει καταχωρήσει ο χρήστης. Σε διαφορετική περίπτωση παίρνει την λίστα των κινήσεων που μπορεί να υλοποιηθούν και σύμφωνα με την τιμή που παίρνει από την ευρετική επιλέγει την κίνηση που συμφέρει για να μεγιστοποιήσει ή ελαχιστοποιήσει αντίστοιχα τον αλγόριθμο.

Move.java: Η κλάση αυτή «δημιουργεί» την κίνηση με βάση την γραμμή, τη στήλη και την τιμή της ευρετικής. Υλοποιήσαμε την toString για να εμφανίζουμε την κίνηση που έπαιξε ο αλγόριθμος.

Heuristic.java: Στην κλάση αυτή η κάθε κίνηση βαθμολογείται (value) με βάση κάποιες τιμές. Αυτές είναι: 1. Ισοτιμία (Coin Parity) 2. Αριθμός δυνατών κινήσεων (Mobility) 3. Βάρος Γωνιών (Corners Captured): Οι γωνίες έχουν μεγαλύτερη βαρύτητα από τις άλλες θέσεις καθόσον εφόσον καταληφθούν δεν αλλάζουν. 4. Σταθερότητα (Stability): Δίνουμε σε κάθε θέση μία βαρύτητα, ανάλογα με την πιθανότητα κάθε παίκτη να αλλάξει.

* Στον φάκελο demo θα βρείτε ένα πειραματικό παιχνίδι στο οποίο ηττηθήκαμε. Συγκεκριμένα στο Demo Full Game1.txt είναι καταγεγραμμένο όλο το Reversi game όπως φαίνεται στο terminal. Επιπλέον έχουμε και δύο screenshot την αρχή και το τέλος του παιχνιδιού img_game_finish.png και img_game_start.png.

Συνεχίστε με την ανάγνωση του Readme.md , εκτελέστε τον κώδικα και ευχαριστηθείτε το παιχνίδι!!!