***ΜΕΛΗ ΟΜΑΔΑΣ***

Αλέξανδρος Μελής ΑΜ:3150102

Κωνσταντίνου Βασίλειος ΑΜ:3150085

Φουρφουρής Ιωάννης ΑΜ:3150190

**ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ**

**1η ΕΡΓΑΣΙΑ**

Για την υλοποίηση της εργασίας δημιουργήσαμε τις εξής κλάσεις:

•Main.java

•Play.java

•State.java

•GamePlayer.java

•Move.java

Η εργασία μας προσομοιώνει το παιχνίδι Reversi. Ο ένας παίκτης είναι ο χρήστης ο οποίος δίνει της εντολές μέσα από το πληκτρολόγιο ενώ ο αντίπαλος παίκτης είναι ο υπολογιστής επιλέγει τις κινήσεις του με την χρήση του αλγορίθμου minimax.

Όσο αναφορά την αρχιτεκτονική του προγράμματος, το παιχνίδι ξεκινά από την κλάση Μain.java. Στη συνέχεια, η Main.java καλεί την κλάση Play.java όπου δίνει την δυνατότητα στον χρήστη να επιλέξει το χρώμα του πιονιού, το αν θα παίξει πρώτος ή όχι και το μέγιστο βάθος αναζήτησης του minimax( αντιστοιχεί στο βαθμό δυσκολίας του παιχνιδιού). Όταν ο παίκτης παίζει καλείται η μέθοδος PlayerTurn() η οποία επιτρέπει στον χρήστη να επιλέγει τη θέση του πιονιού του. Ενώ όταν παίζει ο υπολογιστής καλείται η μέθοδος AITurn() όπου η θέση του πιονιού επιλέγεται βάση του αλγόριθμου minimax ο οποίος βρίσκεται στην κλάση GamePlayer.java . Ο πίνακας για την αναπαράσταση του παιχνιδιού βρίσκεται στην κλάση State.java η οποία περιέχει όλες τις απαραίτητες λειτουργίες για την εκτέλεση του παιχνιδιού. Μερικές από τις λειτουργίες είναι οι εξής:

•Initializer(): αρχικοποιεί τον πίνακα

•Print(): εμφανίζει τον πίνακα

•addElement(x,y,color): προσθέτει πιόνι χρώματος color στη θέση x,y

•Predict(color):εμφανίζει όλες τις πιθανές κινήσεις που μπορούν να εκτελεστούν στη σειρά του παίκτη που διαθέτει το χρώμα color

•FlipElement(x,y,color): αφού τοποθετηθεί το πιόνι χρώματος color στη θέση x,y αλλάζει το χρώμα των πιονιών του αντιπάλου που περιέχονται μεταξύ της θέσης x,y και ενός πιονιού αντίστοιχου χρώματος.

•isFull(): ελέγχει αν ο πίνακας είναι γεμάτος.

Στην κλάση GamePlayer υλοποιείται η λειτουργία του minimax. Αφού έχουν δημιουργηθεί αντίγραφα των επόμενων πιθανών κινήσεων του πιονιού χρώματος color μέσω της μεθόδου getChildren() της κλάσης State.java καλείται η μέθοδος min() και max() για πιόνι χρώματος Χ και πιόνι χρώματος Ο αντίστοιχα και με τη βοήθεια της κλάσης Move.java. Κάθε παραγόμενο παιδί αντιστοιχίζεται σε μία κίνηση και χρειάζεται να κρατάμε τις κινήσεις όπως και τις καταστάσεις. Στην συνέχεια αφού φτάσουμε στο max depth του αλγορίθμου minimax υπολογίζουμε το score κάθε κατάστασης χρησιμοποιώντας την μέθοδο evaluate() η οποία περιέχει τις εξής ευρετικές:

•heuristic1(): απαριθμεί τα πιόνια του κάθε παίχτη

•heuristic2(): δίνει μεγαλύτερο score στον παίχτη που έχει πιάσει γωνίες του πίνακα

•heuristic3(): δίνει μεγαλύτερο score στα άκρα του πίνακα(εκτός γωνιών)

•heuristic4(): μειώνει το score σε καταστάσεις που ευνοούν τον αντίπαλο να τοποθετήσει πιόνι στις γωνίες

•heuristic5(): μειώνει το score σε καταστάσεις που ευνοούν τον αντίπαλο να τοποθετήσει πιόνι στις άκρες(εκτός γωνιών)