

1. Οι κλάσεις - Αρχιτεκτονική

-diskTile

Η κλάση diskTile χρησιμοποιείται για αναπαράσχησε ένα δίσκο που έχει ήδη τοποθετηθεί στο ταμπλό.
X=γραμμή δίσκου, Y=στήλη δίσκου

-freeMoveTile

Αποτελεί παιδί της diskTile το οποίο αναπαριστά ένα ελεύθερο τετράγωνο στο ταμπλό, στο οποίο όμως μπορεί να τοποθετηθεί δίσκος (ως έγκυρη κίνηση) στον γύρο που εξετάζεται εκείνη τη στιγμή. Σε περίπτωση που ένα τέτοιο αντικείμενο επιλεγεί ως κίνηση, ο πίνακας directions (μεγέθους 8) δίνει την πληροφορία για τις κατευθύνσεις προς τις οποίες θα αναζητήσουμε να αντιστρέψουμε αντίπαλους δίσκους.

Το πεδίο value χρησιμοποιείται αποκλειστικά από το CPU για να αποθηκεύσει τις τιμές των φύλλων του MiniMax (evaluation των ταμπλό μεγίστου βάθους), ώστε να μπορούν να μεταφερθούν στο υπόλοιπο δέντρο μέσω των αναδρομών.

-BoardState

- **Πεδία**

board: ένας απλός δισδιάστατος πίνακας, που αποτελεί όμως την κύρια δομή πάνω στην οποία βασίζεται η εισαγωγή/εξαγωγή αντικειμένων στις υπόλοιπες δομές, και η οποία χρησιμοποιείται για την εκτύπωση του ταμπλό μετά από κάθε κίνηση

bDisks: μια λίστα που περιέχει αντικείμενα diskTile που αναπαριστούν μαύρους δίσκους στο ταμπλό

wDisks: η αντίστοιχη λίστα για τους άσπρους δίσκους

lastMove: το πρώην ελεύθερο τετράγωνο στο οποίο τοποθετήθηκε δίσκος και οδήγησε στην τρέχουσα κατάσταση του ταμπλό

- **Constructors**

Ένας για απλή αρχικοποίηση των άνω πεδίων στην καθιερωμένη αρχική κατάσταση του ταμπλό σε ένα παιχνίδι Othello, και ένας copy constructor για deep copy ενός ταμπλό σε άλλο αντικείμενο (χρήση στον MiniMax).

- **Μέθοδοι**

print: Εκτυπώνει τα περιεχόμενα του board με τρόπο που αντιπροσωπεύει ταμπλό Othello. Οι μαύροι δίσκοι αναπαρίστανται με X, ενώ οι άσπροι με O.

ΕΡΓΑΣΙΑ - OTHELLO

validMoves: Με βάση το χρώμα του τρέχοντα παίχτη (παράμετρος player), ελέγχει κάθε δίσκο του στο ταμπλό για να καθορίσει τα έγκυρα ελεύθερα τετράγωνα που βρίσκονται σε μονοπάτι που ξεκινά από τον τρέχοντα δίσκο. Ουσιαστικά, όταν εξετάζεται ένας δίσκος, ελέγχεται για κάθε μία από τις 8 κατευθύνσεις αν υπάρχει μονοπάτι αποκλειστικά αντίπαλων δίσκων που οδηγεί σε ελεύθερο τετράγωνο. Αν βρεθεί με αυτόν τον τρόπο ένα ελεύθερο τετράγωνο, δημιουργείται το αντίστοιχο freeMoveTile με μαρκαρισμένη τη σωστή (αντίθετη από αυτή που ακολουθήσαμε για να φτάσουμε σε αυτό) κατεύθυνση στον πίνακα directions, και τοποθετείται στη λίστα των διαθέσιμων κινήσεων. Αν υπάρχει ήδη στη λίστα λόγω προηγούμενου μονοπατιού, αναζητείται το index του, και απλά ενημερώνεται ο πίνακας directions.

move: Δέχεται ως ορίσματα ένα έγκυρο ελεύθερο τετράγωνο, και το χρώμα του παίχτη που έκανε την κίνηση. Στη συγκεκριμένη υλοποίηση του προγράμματος, είναι σίγουρο πως το freeMoveTile που δίνεται σαν όρισμα έχει προκύψει από κλήση της validMoves με όρισμα το ίδιο χρώμα. Έτσι:

α) Τοποθετείται Χ ή Ο στην αντίστοιχη θέση του freeMoveTile στο board

β) Δημιουργείται diskTile με τις συντεταγμένες του ελεύθερου τετραγώνου και προστίθεται στη σωστή λίστα δίσκων του τρέχοντα παίχτη

γ) Με βάση τον πίνακα directions αντιστρέφεται στο board κάθε αντίπαλος δίσκος του μονοπατιού που εξετάζεται, αφαιρείται το diskTile του από τη λίστα του αντιπάλου (αφού βρεθεί το index του) και προστίθεται στη λίστα του τρέχοντα παίχτη

evaluate: Αναλύεται παρακάτω στην ευρετική

isTerminal: 0 διαθέσιμες κινήσεις για τον Μαύρο && 0 διαθέσιμες κινήσεις για τον Άσπρο == τέλος παιχνιδιού

iOfTile, iOfTile2: Ουσιαστικά η indexOf προσαρμοσμένη στον τύπο των ArrayList που χρησιμοποιούνται και των αντικειμένων που περιέχουν

freeTilesAround: Επιστρέφει true αν υπάρχει έστω και μία κενή θέση γύρω από το diskTile που δίνεται ως όρισμα

printScore: Μέγεθος λίστας Μαύρου == score Μαύρου, αντίστοιχα για τον Άσπρο

-CPU

Είναι το αντικείμενο που αναπαριστά το bot που έχουμε για αντίπαλο. Δέχεται ως ορίσματα το χρώμα του παίκτη που αντιπροσωπεύει και το μέγιστο βάθος του MiniMax που χρησιμοποιεί. Ο Constructor του κάνει μια απλή αρχικοποίηση τιμών, ενώ ο MiniMax (και οι max, min) θα αναλυθούν παρακάτω.

Υποσημείωση: Το πεδίο counter χρησιμοποιείται καθαρά για debugging και φροντίζει να μετρά και να εκτυπώνει το πλήθος των καταστάσεων που εξετάζονται από τον MiniMax κάθε φορά που καλείται (κυρίως για σύγκριση απλού MiniMax, MiniMax με α-β, MiniMax με άλλες βελτιώσεις κ.τ.λ.).

2.Η main – Χρήση

Τα αρχικά βήματα της main είναι τα εξής:

- I. Δημιουργία του αρχικού ταμπλό
- II. Προτροπή για επιλογή χρώματος και εισαγωγή από το χρήστη με scanner. Η main ελέγχει αν η είσοδος είναι έγκυρη.
- III. Προτροπή για επιλογή μεγίστου βάθους MiniMax και εισαγωγή από το χρήστη με scanner. Η main ελέγχει αν η είσοδος είναι έγκυρη.
- IV. Η αντίθετη της 1^{ης} εισόδου, και το βάθος της 2^{ης} δίνονται ως ορίσματα για τη δημιουργία ενός αντικειμένου CPU – bot, ενώ με τη χρήση ενός ακεραίου counter και την αρχικοποίηση του σε 1 ή 2, καθορίζεται αν θα παίξει πρώτος ο χρήστης ή το bot.
- V. Εκτύπωση του αρχικού ταμπλό.

Τα αρχικά βήματα ακολουθούνται από την loop που αναπαριστά τη διαρκή ροή του παιχνιδιού. Κάθε επανάληψη είναι και ένας γύρος παιχνιδιού:

- I. Έλεγχος του ακεραίου endGame: Το endGame ξεκινά με τιμή 0 και αυξάνεται κατά ένα κάθε φορά που ένας απ' τους 2 παίκτες δεν έχει κίνηση. Αν πραγματοποιηθεί μια κίνηση, επιστρέφει στο 0. Αν δεν πραγματοποιηθεί κίνηση για 2 γύρους συνεχόμενα, σημαίνει πως κανείς δεν έχει έγκυρη κίνηση και το παιχνίδι τελειώνει.
- II. Έλεγχος του ακεραίου counter: Αν ο counter είναι περιττός αριθμός, είναι η σειρά του χρήστη. Αν είναι άρτιος, είναι η σειρά του bot.
- III. **ΣΕΙΡΑ ΧΡΗΣΤΗ**

Αρχικά καλείται η validMoves στο τρέχον ταμπλό με όρισμα το χρώμα που έχει επιλέξει ο παίκτης, προκειμένου να ανακτηθεί η λίστα των διαθέσιμων κινήσεων που έχει, και εξετάζεται αν είναι κενή (endGame++) ή όχι (endGame=0). Αν είναι κενή, εμφανίζεται το σχετικό μήνυμα και γίνεται επιστροφή στην αρχή της loop για να δοθεί ο έλεγχος στο CPU ή να τερματίσει το παιχνίδι.

Δεδομένου ότι η λίστα δεν είναι κενή, τυπώνονται στην οθόνη οι συντεταγμένες Othello του κάθε ελεύθερου τετραγώνου που αποτελεί έγκυρη κίνηση, μαζί με το index που έχουν στη λίστα αυτή.

Ο χρήστης προτρέπει να διαλέξει την κίνηση που επιθυμεί, εισάγοντας το αντίστοιχο index. Ελέγχεται εάν ο ακεραίος που δόθηκε ως είσοδος είναι εκτός του πλήθους των διαθέσιμων κινήσεων.

Αφού δοθεί έγκυρη είσοδος, το αντικείμενο του ταμπλό πραγματοποιεί με την move την κίνηση που υπάρχει στο index που επέλεξε ο χρήστης (είσοδος χρήστη-1 λόγω αρίθμησης θέσεων λίστας από 0), και τυπώνει στην οθόνη το νέο ταμπλό και το score που προέκυψαν από την κίνηση.

Το counter αυξάνεται και επιστρέφουμε στην αρχή της loop.

IV. **ΣΕΙΡΑ CPU**

Αποθηκεύεται σε ένα αντικείμενο ελεύθερου τετραγώνου το αντίστοιχο ελεύθερο τετράγωνο που επιστρέφει ο MiniMax και εξετάζεται εάν αυτό είναι κενό (0 διαθέσιμες κινήσεις -> endGame++) ή όχι (endGame=0). Αν είναι κενή, ακολουθεί αντίστοιχο σενάριο με αυτό του χρήστη.

ΕΡΓΑΣΙΑ - OTHELLO

Δεδομένου ότι το αντικείμενο δεν είναι κενό, η έξοδος του MiniMax δίνεται ως είσοδος στο ταμπλό και την move, πραγματοποιείται η κίνηση και τυπώνονται το νέο ταμπλό και score.

Το counter αυξάνεται και επιστρέφουμε στην αρχή της loop.

V. Αφού το πρόγραμμα εξέλθει από τη loop (τέλος παιχνιδιού), προσμετρώνται οι μαύροι και οι άσπροι δίσκοι στο ταμπλό και τυπώνεται το αποτέλεσμα του παιχνιδιού.

#μαύρων δίσκων > #άσπρων δίσκων == Νίκη Μάυρου

#μαύρων δίσκων < #άσπρων δίσκων == Νίκη Άσπρου

#μαύρων δίσκων = #άσπρων δίσκων == Ισοπαλία

3. MiniMax με α-β – Μέθοδοι τεχνητής νοημοσύνης

Ο MiniMax εδώ έχει ως παραμέτρους ένα αντικείμενο-ταμπλό BoardState και το χρώμα που έχει ανατεθεί στο CPU.

Αρχικά εξετάζει αν το χρώμα του CPU έχει διαθέσιμες κινήσεις στο ταμπλό. Αν ναι, τότε ξεκινά με την κλήση του max αν το CPU είναι ο Μαύρος (X) παίχτης ή του min αν είναι ο Άσπρος (O). Τα ορίσματα που περνιούνται στους αλγορίθμους max και min είναι το αρχικό ταμπλό, το αρχικό βάθος που είναι 0, η τιμή -άπειρο για τον α, και η τιμή +άπειρο για τον β.

Μια κλήση ενός εκ' των 2, ας πούμε του max, λειτουργεί με τον εξής τρόπο:

- I. Παίρνει ως παραμέτρους ένα ταμπλό, το τρέχων βάθος που βρίσκεται ο max κόμβος, η τιμή του α και η τιμή του β
- II. Εξετάζει εάν το ταμπλό που του δόθηκε αποτελεί τερματικό ταμπλό (0 διαθέσιμες κινήσεις), ή αν το βάθος που του δόθηκε είναι ίσο με το μέγιστο βάθος του CPU. Αν ναι, δημιουργεί και επιστρέφει ένα αντικείμενο freeMoveTile (αντικείμενο διαθέσιμης κίνησης) με τις συντεταγμένες της τελευταίας κίνησης που οδήγησε στο ταμπλό αυτό και την αποτίμηση του ταμπλό.
- III. Εάν το ταμπλό δεν είναι τερματικό, ανακτάται η λίστα των διαθέσιμων κινήσεων που έχει ο Μαύρος.

Κενή λίστα σημαίνει ότι πρέπει να παίξει πάλι ο Άσπρος, οπότε καλείται ο min για το ίδιο ακριβώς ταμπλό με +1 βάθος.

Δεδομένου ότι η λίστα δεν είναι κενή, δημιουργείται ένα freeMoveTile που τελικά θα αντιστοιχισθεί στην καλύτερη δυνατή για τον Μαύρο κίνηση από το τρέχον ταμπλό.

Ο max λοιπόν ξεκινάει να ελέγχει μία-μία τις διαθέσιμες κινήσεις που έχει λάβει. Η κίνηση πραγματοποιείται με την move σε ένα αντίγραφο του ταμπλό που έχει δοθεί ως όρισμα στον max, και καλείται ο min εντός της εξέτασης της συγκεκριμένης κίνησης. Με αυτόν τον τρόπο, αναδρομικά με αναζήτηση κατά βάθος, ο MiniMax θα φτάσει στο μέγιστο βάθος (ή σε τερματικό κόμβο) του δέντρου, θα αποτιμήσει το ταμπλό μεγίστου βάθους, και θα επιστρέψει ένα freeMoveTile με την αποτίμηση αυτή.

ΕΡΓΑΣΙΑ - OTHELLO

Όταν ο έλεγχος τελικά επιστρέψει στον άνω max, θα του έχει επιστραφεί και ένα freeMoveTile με την αποτίμηση ενός τελικού (για τον MiniMax) ταμπλό.

α-β κόψιμο

Το επόμενο βήμα είναι να εξετασθεί εάν κάποιος απόγονος min σε ψηλότερο επίπεδο (μικρότερος αριθμός = ψηλότερο επίπεδο) έχει ήδη δεχθεί τιμή μικρότερη της τρέχουσας αποτίμησης. Αυτό γίνεται με τον έλεγχο του β που έχει περασθεί ως όρισμα στον max. Εάν η τρέχουσα αποτίμηση είναι $\geq \beta$, ο max αγνοεί των έλεγχο των υπόλοιπων κινήσεων που του έχουν μείνει και επιστρέφει επί τόπου την αποτίμηση (η τιμή θα έχει σημασία στην περίπτωση που ισούται με το β, αλλά εξοικονομείται αρκετός χρόνος εάν διακοπεί εκεί η εξέταση διαθέσιμων κινήσεων).

Η ιδέα είναι ότι έχει βρεθεί μια τιμή την οποία είτε θα επιστρέψει την ίδια ο max, είτε κάποια άλλη μεγαλύτερή της (υπόλοιπες κινήσεις που έχουν μείνει να εξετασθούν), στα άνω επίπεδα. Επομένως ακόμη και στην περίπτωση που η ίδια τιμή του συγκεκριμένου max συνεχίσει να ανεβαίνει επίπεδα, κάποια στιγμή θα φτάσει στον min που έχει θέσει το β βάσει του οποίου απορρίφθηκε. Έτσι είναι σίγουρο πως αυτός ο min δε θα επιλέξει κάποια από τις κινήσεις του max που χρησιμοποιείται εδώ ως παράδειγμα και ο περεταίρω έλεγχος κινήσεων καθίσταται ανούσιος.

Εάν δεν κοπεί

Ουσιαστικά ακολουθεί η διαδικασία μιας τυπικής max μεθόδου. Δεδομένου ότι η καλύτερη για τον τρέχων max κίνηση, maxMove, έχει αρχική τιμή $-\infty$, συγκρίνεται η τιμή της κίνησης που επιστράφηκε με την τρέχουσα maxMove. Αν οι δύο τιμές είναι ίσες επιλέγεται μία εκ' των 2 κινήσεων για maxMove με πιθανότητα 50%. Αν η νέα τιμή είναι μεγαλύτερη αυτής του maxMove, η maxMove αντικαθίσταται με την τρέχουσα κίνηση. Αν είναι μικρότερη, η maxMove παραμένει.

Η απόκλιση από μια τυπική max γίνεται στην αντικατάσταση της maxMove. Οι συντεταγμένες της γίνονται ίδιες με αυτές της κίνησης που εξετάζεται, όμως η τιμή της θα είναι η αποτίμηση ενός τερματικού ταμπλό, και όχι αυτή του τρέχοντος. Η μεταφορά των αποτιμήσεων των φύλλων του MiniMax καθώς και οι συγκρίσεις σε όλους τους κόμβους γίνεται με τη χρήση των προσωρινών/βοηθητικών freeMoveTile tmp, όπως φαίνεται στον κώδικα.

Η τελευταία ενέργεια που κάνει ο max στον έλεγχο μιας συγκεκριμένης κίνησης, είναι η σύγκρισή της με τον α, και επομένως η πιθανή ενημέρωσή του. Ο α θα λειτουργήσει για τους min κατώτερων επιπέδων με τρόπο ανάλογο του β που εξετάστηκε πιο πάνω (τιμή κίνησης του min $\leq \alpha \rightarrow$ πρόωρος τερματισμός του min)

IV. Τέλος, εάν δε διακοπεί πιο νωρίς, ο max επιστρέφει την κίνηση maxMove

Ο τρόπος με τον οποίο λειτουργεί ο min είναι εντελώς ανάλογος, όπου Μαύρος \rightarrow Άσπρος, β \rightarrow α και αντίστροφα.

ΕΡΓΑΣΙΑ - OTHELLO

4.Δυνατότητες

-Ταχύτητα – Μέγιστο βάθος

Το πρόγραμμα φαίνεται να ανταποκρίνεται αρκετά γρήγορα κατά τον γύρο του CPU για πλήθος ελεγμένων ταμπλό μέχρι και 60.000, που δείχνει να είναι λίγο κάτω από τη χειρότερη περίπτωση του MiniMax με βάθος 5 και α-β. Για βάθη 6,7 οι χρόνοι απόκρισης είναι και πάλι σχετικά χαμηλοί σε σύγκριση με βάθος 5, ενώ για βάθος 8 όταν το πλήθος των διαθέσιμων κινήσεων κάθε παίχτη αυξάνεται 5-6 γύρους μετά την έναρξη (≥ 300.000 ταμπλό), η ταχύτητά του είναι σημαντικά μικρότερη και μόνο φθίνει για τους επόμενους γύρους.

Σίγουρα δε βοηθά το μεγάλο πλήθος ελέγχων με τους οποίους υλοποιούνται οι διάφορες καταστάσεις και ενέργειες του Othello, και σε ένα βαθμό η ευρετική συνάρτηση που δεν μειώνει τις διαθέσιμες κινήσεις των παικτών, και άρα ούτε και το μέγιστο παράγοντα διακλάδωσης του MiniMax.

Αξίζει πάντως να σημειωθεί η βελτίωση που υπάρχει με τη χρήση του α-β κοψίματος σε σχέση με τον MiniMax χωρίς αυτό. Σαν τυπικό παράδειγμα που επιδεικνύει τη σημαντική διαφορά (ο βαθμός της βελτίωσης φυσικά εξαρτάται από πολλούς παράγοντες), δίνονται εικόνες από παιχνίδι με βάθος 5, όπου το CPU είναι ο Άσπρος, στην κίνηση που του αναλογεί στον 2^ο και 12^ο γύρο, για την ίδια ακολουθία κινήσεων.

Με α-β

```
CPU's turn
*****
States checked: 661
  a b c d e f g h
1 | | | | | | | 1
2 | | | | | | | 2
3 | | | | | | | 3
4 | | | | | | | 4
5 | | | | | | | 5
6 | | | | | | | 6
7 | | | | | | | 7
8 | | | | | | | 8
  a b c d e f g h

SCORE
-----
Black: 3
White: 3
-----
```

```
CPU's turn
*****
States checked: 1289
  a b c d e f g h
1 | | | | | | | 1
2 | | | | | | | 2
3 | | | | | | | 3
4 | | | | | | | 4
5 | | | | | | | 5
6 | | | | | | | 6
7 | | | | | | | 7
8 | | | | | | | 8
  a b c d e f g h

SCORE
-----
Black: 12
White: 4
-----
```

Έλεγχος α-β σε σχόλια /**/

```
CPU's turn
*****
States checked: 2478
  a b c d e f g h
1 | | | | | | | 1
2 | | | | | | | 2
3 | | | | | | | 3
4 | | | | | | | 4
5 | | | | | | | 5
6 | | | | | | | 6
7 | | | | | | | 7
8 | | | | | | | 8
  a b c d e f g h

SCORE
-----
Black: 3
White: 3
-----
```

```
CPU's turn
*****
States checked: 14918
  a b c d e f g h
1 | | | | | | | 1
2 | | | | | | | 2
3 | | | | | | | 3
4 | | | | | | | 4
5 | | | | | | | 5
6 | | | | | | | 6
7 | | | | | | | 7
8 | | | | | | | 8
  a b c d e f g h

SCORE
-----
Black: 12
White: 4
-----
```

-Δυσκολία – Ευρετική

Ευάλωτοι δίσκοι -> έχουν γύρω τους τουλάχιστον ένα κενό τετράγωνο και είναι πιο εύκολο (αν δεν το κάνουν ήδη, σε επόμενους γύρους) να δώσουν έγκυρες κινήσεις στον αντίπαλο και να αντιστραφούν

Δίσκοι πλευράς -> a3 έως a6, h3 έως h6, c1 έως f1, c8 έως f8

C Δίσκοι -> a2, a7, h2, h7, b1, g1, b8, g8 (διπλανοί στις γωνίες)

X Δίσκοι -> b2, b7, g2, g7 (πρώτοι διαγώνια από τις γωνίες)

Στις διαφορές πάντα ισχύει (Μαύρος – Άσπρος), ενώ τα αρνητικά βάρη στους C και X δίσκους λόγω της πιθανής ευκαιρίας που δίνεται στον αντίπαλο να κερδίσει την κοντινή σε αυτούς γωνία.

I. Για συνολικό πλήθος κινήσεων < 30 (νωρίς στο παιχνίδι)

Διαφορά πλήθους κινήσεων - Διαφορά «ευάλωτων» δίσκων + 2*Διαφορά δίσκων πλευράς
-10*Διαφορά C Δίσκων – 25*Διαφορά X Δίσκων + 100*Διαφορά Δίσκων γωνίας

II. Για συνολικό πλήθος κινήσεων ≥30 και <54 (μέσα παιχνιδιού)

Διαφορά πλήθους κινήσεων – 20*Διαφορά «ευάλωτων» δίσκων + 10*Διαφορά δίσκων πλευράς
-10*Διαφορά C Δίσκων – 25*Διαφορά X Δίσκων + 100*Διαφορά Δίσκων γωνίας

III. Για τις τελευταίες 10 (ή λιγότερες) κινήσεις (τελικές κινήσεις)

5*Διαφορά πλήθους δίσκων – 2*Διαφορά «ευάλωτων» δίσκων + Διαφορά πλήθους κινήσεων +
2*Διαφορά δίσκων γωνίας

-Το σκεπτικό εδώ είναι στην αρχή του παιχνιδιού το πρόγραμμα να στοχεύει στην αύξηση της κινητικότητάς (πλήθος κινήσεων) του σε σχέση με αυτή του χρήστη και στην απόκτηση γωνιών, ενώ ταυτόχρονα να αποφεύγει κινήσεις που δίνουν ευκαιρίες στον χρήστη να πάρει μία γωνία.

-Όσο το παιχνίδι προχωράει, το πρόγραμμα καλό είναι να εστιάζει στις πλευρές και στην «ασφάλεια» των δίσκων του, ώστε να μην μπορεί ο χρήστης να τα αντιστρέψει.

-Τέλος, έχει πολύ μεγάλη σημασία η μεγιστοποίηση του πλήθους των δίσκων του προγράμματος.

Δυστυχώς μετά από αρκετά παιχνίδια με βάθος 6-7, φαίνεται πως το πρόγραμμα είναι ευάλωτο στην αρχική φάση του παιχνιδιού, επιλέγοντας σε μερικές περιπτώσεις κινήσεις που δίνουν την ευκαιρία στο χρήστη να αποκτήσει γωνίες του ταμπλό. Φαίνεται όμως πως εάν δεν δημιουργηθεί αυτό το πρόβλημα στην αρχή του παιχνιδιού και το CPU αποκτήσει πλεονέκτημα γωνίας, κινείται αρκετά αποτελεσματικά προς την απόκτηση θέσεων που βοηθούν στη μεγιστοποίηση τις διαφορές δίσκων υπέρ του στις τελικές κινήσεις.

ΕΡΓΑΣΙΑ - OTHELLO

-Επιπλέον βελτιώσεις – Κλειστό Σύνολο – Επέκταση Βάθους

Στα σχόλια του αρχείου με τον κώδικα του CPU, καθώς και σε αυτά του αρχείου του BoardState, υπάρχει η υλοποίηση του MiniMax με κλειστό σύνολο και η μέθοδος σύγκρισης ταμπλό αντίστοιχα. Παρόλο που με κλειστό σύνολο έχουμε και πάλι αξιοσημείωτη μείωση στο πλήθος καταστάσεων που ελέγχει ο MiniMax, φαίνεται πως για τις συγκεκριμένες υλοποιήσεις των ταμπλό, του τρόπου σύγκρισής τους και του ελέγχου του κλειστού συνόλου, ο χρόνος του MiniMax είναι τελικά χειρότερος.

Επίσης στα σχόλια του αρχείου με τον κώδικα του CPU υπάρχει η υλοποίηση του ελέγχου μεγίστου βάθους – τελικής κατάστασης των min και max με επέκταση βάθους. Ο κώδικας αυτός δηλαδή αντικαθιστά τον έλεγχο `if(board.isTerminal() || depth == maxDepth)` και τον κώδικα που περιέχεται σε αυτόν. Παρ' όλα αυτά πάλι δημιουργείται σημαντική καθυστέρηση στην εκτέλεση του MiniMax ακόμα και για επέκταση βάθους 2 επιπέδων. Επίσης φαίνεται δύσκολο να διακριθούν κρίσιμες/μη κρίσιμες καταστάσεις για επέκταση/αποκοπή από αυτή μόνο την ευρετική, δεδομένων των διαφόρων παραγόντων που παίζουν ρόλο στην επιλογή της «καλύτερης» κίνησης σε ένα γύρο Othello.

5.Τυπικά παραδείγματα χρήσης

Επιλογή Χρώματος και μεγίστου βάθους MiniMax – Αρχικό ταμπλό και προτροπή κίνησης

```
Choose 1 for X(Black) or 2 for O(White). Black always plays first.
3
Please insert either 1 or 2
1
Insert the maximum depth of the CPU's MiniMax tree (Minimum: 1, Maximum: 60)
-1
Please insert a valid maximum depth
62
Please insert a valid maximum depth
5

GAME START

  a b c d e f g h
1 | | | | | | | | 1
2 | | | | | | | | 2
3 | | | | | | | | 3
4 | | | | O | X | | | 4
5 | | | | X | O | | | 5
6 | | | | | | | | 6
7 | | | | | | | | 7
8 | | | | | | | | 8
  a b c d e f g h

Your turn
*****
Choose one from the current valid moves:
1)d3 2)f5 3)c4 4)e6
```


ΕΡΓΑΣΙΑ - ΟΤHELLO

Ζευγάρι κινήσεων χρήστη και CPU

```
a b c d e f g h
1 | | | | | | | 1
2 | | | | | | | 2
3 | | | X | | | 3
4 | | | X | X | | 4
5 | | X | X | O | | 5
6 | X | O | O | | | 6
7 | | | | O | | | 7
8 | | | | | | | 8
a b c d e f g h

-----
SCORE

Black: 6
White: 6
-----

Your turn
*****
Choose one from the current valid moves:
1)e8 2)g6 3)f6 4)c7 5)f8 6)g5 7)b7 8)d7 9)f7
1
a b c d e f g h
1 | | | | | | | 1
2 | | | | | | | 2
3 | | | X | | | 3
4 | | | X | X | | 4
5 | | X | X | X | O | 5
6 | X | O | O | X | | 6
7 | | | | X | | | 7
8 | | | | X | | | 8
a b c d e f g h

-----
SCORE

Black: 10
White: 3
-----

CPU's turn
*****
States checked: 11049
a b c d e f g h
1 | | | | | | | 1
2 | | | | | | | 2
3 | | | X | | | 3
4 | | | X | X | | 4
5 | | X | X | X | O | 5
6 | X | O | O | X | | 6
7 | | | | O | | | 7
8 | | | | X | O | | 8
a b c d e f g h

-----
SCORE

Black: 9
White: 5
-----
```

Μήνυμα μη διαθέσιμων κινήσεων

```
Your turn
*****
Choose one from the current valid moves:
1)h2
1
a b c d e f g h
1 | O | O | O | O | | | O | O | 1
2 | O | O | O | O | X | X | X | X | 2
3 | O | O | O | O | O | O | O | | 3
4 | O | O | O | O | O | O | O | | 4
5 | O | X | O | O | O | O | O | X | 5
6 | O | O | O | O | O | O | O | O | 6
7 | O | O | O | O | O | O | O | O | 7
8 | O | O | O | O | O | O | O | O | 8
a b c d e f g h

-----
SCORE

Black: 6
White: 54
-----

CPU's turn
*****
States checked: 33
a b c d e f g h
1 | O | O | O | O | | O | O | O | 1
2 | O | O | O | O | O | O | X | X | 2
3 | O | O | O | O | O | O | O | | 3
4 | O | O | O | O | O | O | O | | 4
5 | O | X | O | O | O | O | O | X | 5
6 | O | O | O | O | O | O | O | O | 6
7 | O | O | O | O | O | O | O | O | 7
8 | O | O | O | O | O | O | O | O | 8
a b c d e f g h

-----
SCORE

Black: 4
White: 57
-----

Your turn
*****
You have no valid moves.

CPU's turn
*****
States checked: 7
a b c d e f g h
1 | O | O | O | O | | O | O | O | 1
2 | O | O | O | O | O | O | O | O | 2
3 | O | O | O | O | O | O | O | O | 3
4 | O | O | O | O | O | O | O | | 4
5 | O | X | O | O | O | O | O | X | 5
6 | O | O | O | O | O | O | O | O | 6
7 | O | O | O | O | O | O | O | O | 7
8 | O | O | O | O | O | O | O | O | 8
a b c d e f g h
```

Τέλος παιχνιδιού

```
a b c d e f g h
1 | O | O | O | O | | O | O | O | 1
2 | O | O | O | O | O | O | O | O | 2
3 | O | O | O | O | O | O | O | O | 3
4 | O | O | O | O | O | O | O | O | 4
5 | O | X | O | O | O | O | O | O | 5
6 | O | O | O | O | O | O | O | O | 6
7 | O | O | O | O | O | O | O | O | 7
8 | O | O | O | O | O | O | O | O | 8
a b c d e f g h
```

```
-----
SCORE

Black: 1
White: 62
-----

Your turn
*****
You have no valid moves.

CPU's turn
*****
CPU has no valid moves.

*****
WHITE WINS
*****
```