ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΕΡΓΑΣΙΑ 1

Τεχνητή Νοημοσύνη

Αναστασιάδου Κωνσταντίνα

Γεωργιάδη Δέσποινα

Λέσι Άννα

Εισαγωγή

Η εργασία μας αφορά το δεύτερο πρόβλημα της δοσμένης εκφώνησης, δηλαδή την υλοποίηση του παιχνιδιού Reversi. Η υλοποίησή μας επιτρέπει σε έναν χρήστη να παίζει Reversi με αντίπαλο τον υπολογιστή με την δυνατότητα επιλογής σειράς (turn) που μπορεί να παίξει και επιλογής επιπέδου δυσκολίας. Το πρόγραμμα μας γράφτηκε σε java, και μέρος του βασίστηκε στον κώδικα των εργαστηρίων του μαθήματος.

Ο αλγόριθμος τεχνητής νοημοσύνης που χρησιμοποιήσαμε για την υλοποίηση του αντιπάλου AI στο παιχνίδι μας είναι ο Minimax με πριόνισμα α-β.

Επεξήγηση Αρχείων Κώδικα

Move

Ο κώδικας είναι πανομοιότυπος με αυτόν του ομώνυμου αρχείου του εργαστηρίου 4. Χρησιμοποιείται για τα παιδία του δένδρου και την προηγούμενη κίνηση που εκτελέστηκε.

GamePlayer

Περιέχει τις μεθόδους Minimax, min, max, getPlayer, setPlayer, getMaxDepth, setMaxDepth.

Οι setMaxDepth και getMaxDepth χρησιμοποιούνται για τον καθορισμό της «δυσκολίας» του παιχνιδιού, και αντίστοιχα για την επιστροφή του μέγιστου βάθους αναζήτησης της Minimax.

Οι setPlayer και getPlayer χρησιμοποιούνται για τον καθορισμό και αντίστοιχα την επιστροφή του player.

Η Minimax καλεί την max εφόσον το AI θέλει πάντα να μεγιστοποιεί το value και επιστρέφει την κίνηση που θα κάνει εν τέλει το AI μας.

Οι min και max καλούνται εναλλάξ όσο αλλάζουμε επίπεδο. Η max καλεί την min για το επόμενο επίπεδο του δέντρου καταστάσεων, και, αντίστοιχα, η min καλεί την max για το παρά-επόμενο κοκ.

Γενικότερα, η υλοποίηση μας βασίζεται σε αυτή του εργαστηρίου με την προσθήκη πριονίσματος α-β.

Board

Μεταβλητές

Οι μεταβλητές X, O, EMPTY κατ' αντιστοιχία αναπαριστούν τους μαύρους, άσπρους δίσκους και τις κενές θέσεις στον πίνακα. Οι μεταβλητές X_disks, O_disks κρατάνε το πόσους δίσκους έχει στον πίνακα ο αντίστοιχος παίκτης (X, O) κατά τη διάρκεια του παιχνιδιού.

Μέθοδοι

Περιέχει τις μεθόδους makeMove, isValidMove, getChildren, evaluate, canPlay, possibleMoves, isTerminal, printMoves, print, print_available, getLastMove, setLastMove, getLastColorPlayed, setLastColorPlayed, getGameBoard, setGameBoard, getOScore και getXScore.

Οι setLastMove και getLastMove χρησιμοποιούνται για τον καθορισμό και αντίστοιχα την επιστροφή της τελευταίας κίνησης.

Οι setLastColorPlayed και getLastColorPlayed χρησιμοποιούνται για τον καθορισμό και αντίστοιχα την επιστροφή του τελευταίου παίκτη που είχε σειρά.

Οι setGameBoard και getGameBoard χρησιμοποιούνται για τον καθορισμό και αντίστοιχα την επιστροφή του δισδιάστατου πίνακα που αναπαριστά τον πίνακα του παιχνιδιού μας.

Οι getOScore και getXScore χρησιμοποιούνται για την επιστροφή του score του παίκτη που έχει τους άσπρους δίσκους και του παίκτη που έχει τους μαύρους δίσκους, αντίστοιχα.

Η makeMove διατρέχει τον δισδιάστατό μας πίνακα για δοσμένες συντεταγμένες κίνησης ενός παίκτη και γυρνάει τους δίσκους που πρέπει να γυριστούν, δηλαδή τους δίσκους του αντιπάλου που θα εγκλωβιστούν με αυτή την κίνηση. Ελέγχει προς κάθε κατεύθυνση απ' αυτή την κίνηση που κάνει (πάνω, κάτω, δεξιά, αριστερά, διαγώνια πάνωδεξιά, κάτω-δεξιά, πάνω-αριστερά, κάτω-αριστερά) αν βρίσκονται δίσκοι του αντιπάλου και σταματάμε όταν βρούμε δικό μας δίσκο σε αυτή την κατεύθυνση. Αν δεν βρούμε κανένα δίσκο δικό μας σε αυτή την κατεύθυνση, τότε δεν γυρνάμε τους δίσκους του αντιπάλου, αντίθετα, αν ικανοποιηθεί η συνθήκη (να βρούμε δικό μας δίσκο) τους γυρνάμε. Επιπλέον, αυξομειώνονται οι μεταβλητές που αποθηκεύουν το πόσους δίσκους έχει ο κάθε παίκτης, δηλαδή μειώνονται οι δίσκοι του αντιπάλου όταν γυρνάμε δίσκους του και αυξάνονται οι δικοί μας δίσκοι ανάλογα.

Η isValidMove ελέγχει αν είναι αποδεκτή η κίνηση που επιθυμεί ο παίκτης να κάνει. Δηλαδή επιτρέπουμε στον παίκτη να κάνει κίνηση μόνο εάν παγιδεύει έστω και 1 δίσκο του αντιπάλου του. Επιπλέον, ένας καινούργιος δίσκος μπορεί να τοποθετηθεί μόνο σε κενή θέση.

Η getChildren επιστρέφει έναν ArrayList με τα παιδιά (αντικείμενα τύπου Board) της τρέχουσας κατάστασης και αυτό γίνεται διατρέχοντας τον πίνακά μας και βρίσκοντας όλες τις πιθανές κινήσεις που μπορούν να γίνουν από τον ζητούμενο παίκτη (color) και για καθεμία απ' αυτές δημιουργείται ένα παιδί-κατάσταση όπου έχει πραγματοποιηθεί η συγκεκριμένη κίνηση.

Η evaluate βρίσκει το value για μία δοσμένη κατάσταση με βάση τον παίκτη (color). Αρχικά, έχουμε τις συνθήκες για κίνηση που τερματίζει το παιχνίδι («killer move»), δηλαδή όταν στον πίνακα υπάρχουν μόνο δικοί μας δίσκοι και κανένας του αντιπάλου. Σε αυτή την

περίπτωση, προσαυξάνεται το value (sum) κατά 1000. Έπειτα διατρέχεται ο πίνακας (gameBoard) και για κάθε δίσκο στον πίνακα προσαυξάνουμε -εάν ανήκει στον παίκτη- και μειώνουμε -εάν ανήκει στον αντίπαλο- το κόστος της κατάστασης με βάρος ίσο με:

- 16, όταν έχουμε ένα δίσκο σε γωνία, καθώς οι γωνίες είναι σημαντικές για το reversi.
- 4, όταν ένας δίσκος βρίσκεται στα περιθώρια του πίνακα (εκτός των γωνιών).
- 1, για οποιαδήποτε άλλη θέση που βρίσκεται ένας δίσκος.

Η canPlay ελέγχει αν υπάρχουν possibleMoves για τον παίκτη (player) και επιστρέφει false αν δεν υπάρχουν, true αν υπάρχουν.

Η possibleMoves γυρνάει έναν ArrayList με όλες τις πιθανές κινήσεις που μπορεί να κάνει ο παίκτης (color) σε μορφή πίνακα δύο θέσεων, διατρέχοντας τον πίνακά μας και βλέποντας για κάθε θέση του πίνακα αν οι συντεταγμένες της θέσης αυτής είναι valid move για τον παίκτη.

Η isTerminal ελέγχει αν ο πίνακας δεν έχει πλέον διαθέσιμες θέσεις και αν ισχύει αυτό επιστρέφουμε true, αν όχι false.

Η printMoves εκτυπώνει την λίστα των συντεταγμένων των διαθέσιμων κινήσεων που μπορούν να γίνουν από τον παίκτη (color) και αντίστοιχα τον ενημερώνει αν δεν υπάρχει καμία διαθέσιμη κίνηση.

Η print εκτυπώνει την αναπαράσταση του πίνακα του παιχνιδιού.

Η print_available εκτυπώνει τον πίνακα του παιχνιδιού επισημαίνοντας τις διαθέσιμες κινήσεις του χρήστη με αστερίσκο (*).

Η score εμφανίζει τα αποτελέσματα της παρτίδας.

Main

Το πρόγραμμα μας εκτελείται σύμφωνα με τις οδηγίες του αρχείου README.txt. Τρέχουμε ένα αντικείμενο τύπου Window, το οποίο αναπαριστά το παιχνίδι μας και τις λειτουργίες του.

Window

Το πρόγραμμα μας εκτελείται σύμφωνα με τις οδηγίες του αρχείου README.txt.
Η κλάση αυτή είναι το κύριο μέρος για την υλοποίηση του προγράμματος με gui.

Μεταβλητές gamer : είναι ένα στιγμιότυπο της κλάσης GamePlayer που θα λειτουργήσει ως το AI μας. turn : αν παίζει 1ος ή 2ος
AI : είναι ένα στιγμιότυπο της κλάσης Move που θα λειτουργήσει ως την κίνηση που θα κάνει το AI μας κάθε φορά.

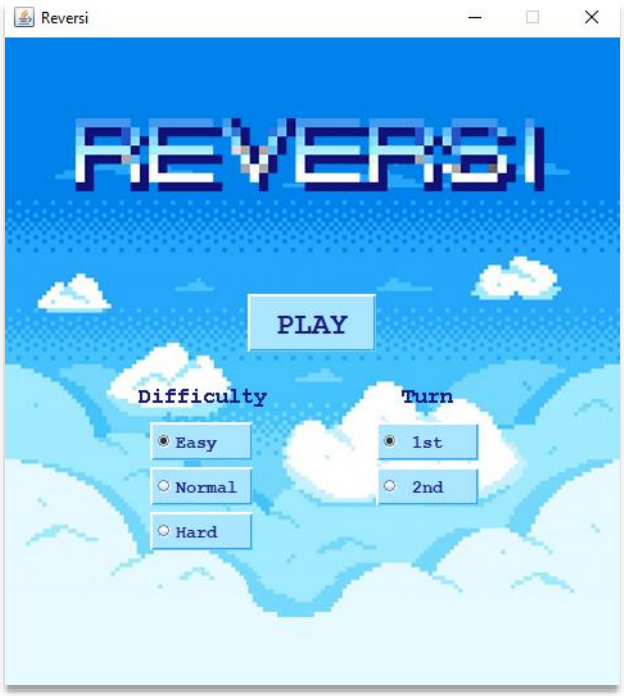
board : είναι ένα στιγμιότυπο της κλάσης Board που θα λειτουργήσει ως ο πίνακας του παιχνιδιού. data : ο δισδιάστατος πίνακας που αντιπροσωπεύει τον πίνακα του παιχνιδιού μας.
frame : κεντρικό παράθυρο του παιχνιδιού που προστίθενται τα panels που εναλλάσσονται.
startingScreen : είναι το panel εκκίνησης του παιχνιδιού όπου ο χρήστης μπορεί να επιλέξει την δυσκολία του παιχνιδιού και αν θέλει να παίξει 1ος ή όχι.
panel : είναι το κεντρικό panel του παιχνιδιού στο οποίο εμφανίζεται ο πίνακας και ο χρήστης παίζει.
endScreen : είναι το τελευταίο panel, όπου εμφανίζουμε τα αποτελέσματα και ποιος κέρδισε και δίνεται στον χρήστη η επιλογή για έναρξη καινούργιας παρτίδας ή έξοδο από την εφαρμογή.

Μέθοδοι
Window() : στον κατασκευαστή της κλάσης γίνεται μορφοποίηση και αρχικοποίηση όλων των απαραίτητων panel, των κουμπιών τους και των λειτουργιών των κουμπιών. Για τις λειτουργίες των κουμπιών, μπορείτε να δείτε στις γραμμές του κώδικα από την σειρά 135 έως 204, όπου έχουμε καθορίσει τις λειτουργίες των κουμπιών.
mousePressed : Για κάθε κλικ που γίνεται στο panel παίρνουμε την τοποθεσία του κέρσορα και με βάση αυτή ελέγχουμε αν ο παίκτης μπορεί να κινηθεί ή όχι και εφόσον μπορεί και είναι σειρά του, εκτελούμε κίνηση για τον ίδιο εκεί που πάτησε και αλλάζουμε τον πίνακα αναλόγως, ενώ αν δεν είναι σειρά του να παίξει τότε περιμένουμε κλικ από τον χρήστη για να προχωρήσει σε κίνηση το AI μας. Όταν το παιχνίδι φτάσει σε κατάσταση end game το κλικ του χρήστη εκείνη την στιγμή θα αλλάξει το panel σε endScreen και θα του εμφανίσει την τελική οθόνη του παιχνιδιού.
AIplays : Μέθοδος που εφόσον η παράμετρος της είναι αληθής, πραγματοποιεί κίνηση για τον AI.
Available : Τοποθετεί την τιμή 2 στον πίνακα στα κελιά των διαθέσιμων κινήσεων (κλήση της possibleMoves) για τον χρήστη. Χρησιμεύει στην εκτύπωση του panel.
RemoveTwos : Αφαιρεί τις τιμές 2 από τον πίνακα, τις οποίες πρόσθεσε η Available, αφού γίνεται εκτύπωση.

Minimax με a-b pruning

Στο πρόγραμμά μας χρησιμοποιείται ο αλγόριθμος Minimax. Ο αλγόριθμος Minimax παρέχει τον τρόπο για να βρεθεί η βέλτιστη κίνηση στο παιχνίδι μας, καθώς είναι παιχνίδι 2 αντιπάλων. Στο δέντρο αναζήτησης που προκύπτει για τους 2 παίκτες, υπάρχουν 2 είδη κόμβων, κόμβοι που αναπαριστούν τις κινήσεις του AI και κόμβοι που αναπαριστούν τις κινήσεις του αντιπάλου του, χρήστη. Επίσης για την αξιολόγηση μίας κατάστασης υπάρχει η συνάρτηση οφέλους evaluate που δίνει έναν ακέραιο αριθμό στην κατάσταση ως αξία. Οι κόμβοι του AI είναι max κόμβοι, δηλαδή στόχος τους είναι να μεγιστοποιήσουν την αξία του υπο-δέντρου τους. Για να επιτευχθεί αυτό, ο max κόμβος επιλέγει το παιδί με την μεγαλύτερη αξία και αυτή η αξία γίνεται η αξία του max κόμβου. Αντίστοιχα, οι κόμβοι του χρήστη είναι min κόμβοι, στόχος τους είναι να ελαχιστοποιήσουν την αξία του υποδέντρου τους. Ο min κόμβος επιλέγει το παιδί με την μικρότερη αξία και αυτή η αξία γίνεται η αξία του min κόμβου.
Το πριόνισμα a-b παρέχει 2 όρια το άλφα και βήτα που περιορίζουν το σύνολο πιθανών λύσεων με βάση το μέρος του δέντρου αναζήτησης που έχει ήδη αναπτυχθεί. Αποφεύγεται η παραγωγή υπο-δέντρων που ούτως ή άλλως δεν θα επιλέγονταν. Το άλφα είναι το μέγιστο χαμηλότερο όριο αξίας για τις πιθανές λύσεις και αντίστοιχα το βήτα είναι το ελάχιστο άνω όριο. Παραπάνω για την υλοποίηση της Minimax σε σχέση με το πρόγραμμα
στην αναφορά στο κομμάτι της κλάσης Gameplayer και στα σχόλια του κώδικά μας στην κλάση Gameplayer μεθόδους Minimax, max, min.

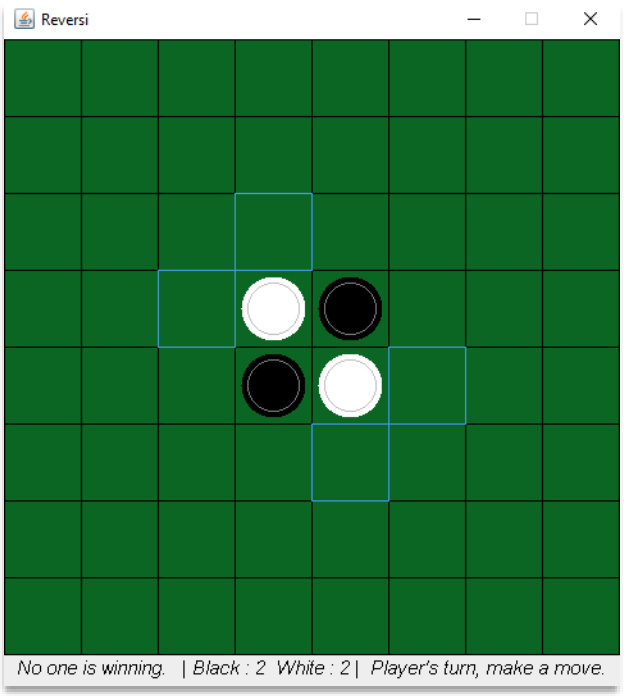
Χρήση Προγράμματος – Παραδείγματα



Starting Screen

Η αρχική οθόνη του παραθύρου που εμφανίζεται με την εκτέλεση του κώδικα έχει τις επιλογές δυσκολίας και σειράς που προτιμάει ο παίκτης. Είναι ήδη προεπιλεγμένα τα Easy και 1st, εφόσον αντίστοιχη είναι και η αρχικοποίηση των τιμών turn και maxDepth.

Για την έναρξη του παιχνιδιού πρέπει να πατηθεί το κουμπί PLAY. Οι επιλογές του χρήστη παραμένουν εάν πατηθεί το κουμπί Play Again στην τελική οθόνη.

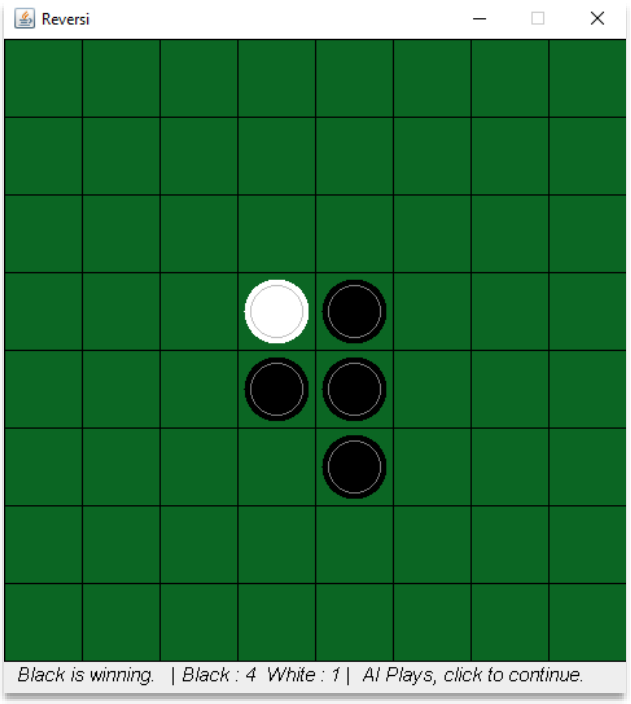


Game Screen

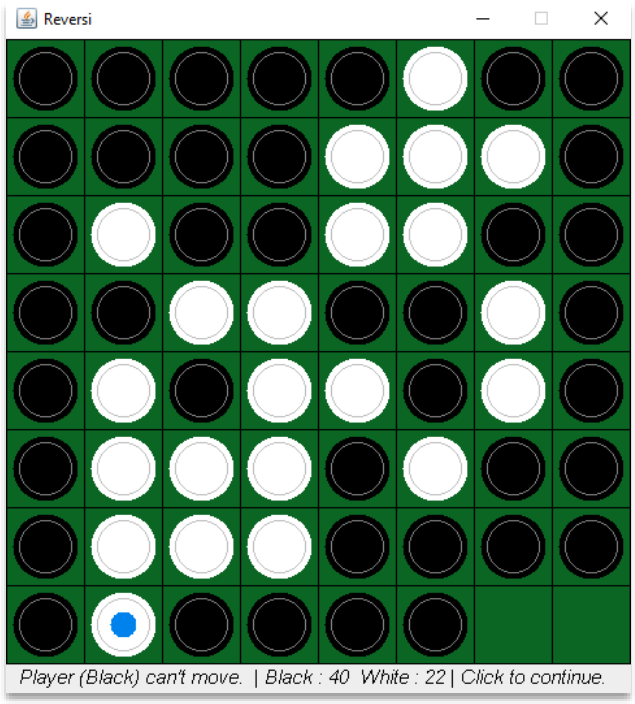
Η οθόνη στην οποία λαμβάνει μέρος το παιχνίδι, εάν ο παίκτης επιλέξει να παίξει πρώτος, εμφανίζονται απευθείας οι διαθέσιμες κινήσεις που μπορεί να επιλέξει πατώντας στο αντίστοιχο υπογραμμισμένο πλαίσιο.

Το AI παίζει στη σειρά του όταν ο χρήστης πατήσει οπουδήποτε πάνω στο ταμπλό. Εάν ο παίκτης επιλέξει να παίξει δεύτερος, πρέπει να πατήσει οπουδήποτε προκειμένου το AI να εκτελέσει την κίνηση του για να συνεχίσει μετά ο παίκτης κοκ.

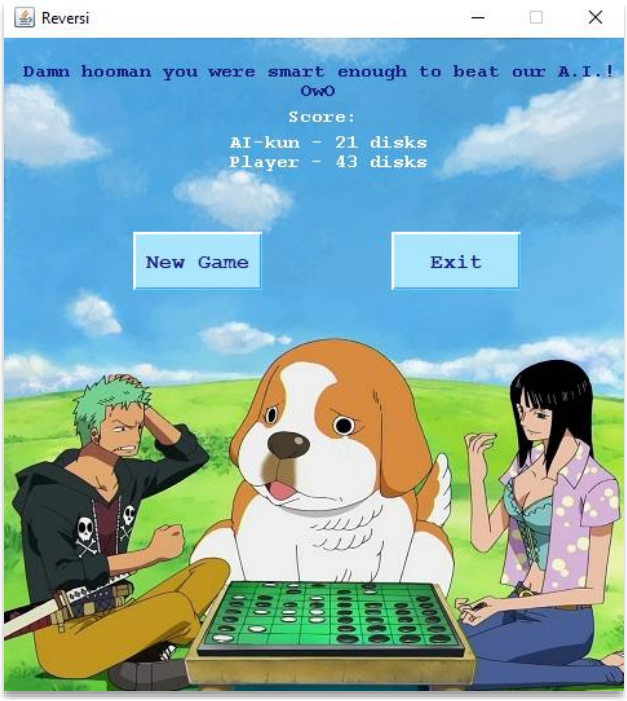
Στο κάτω μέρος του παραθύρου εμφανίζονται ποιος έχει το «προβάδισμα», το σκορ και το status του παιχνιδιού.



Ο χρήστης (μαύρο, πάνω αριστερά) παίζει και η κίνηση εμφανίζεται στο ταμπλό καθώς το προβάδισμα και το σκορ ανανεώνονται. Είναι η σειρά του AI, ο χρήστης πρέπει τώρα να πατήσει οπουδήποτε ώστε αυτό να παίξει.



Ο χρήστης (μαύρο, πάνω δεξιά) δεν έχει διαθέσιμη κίνηση να εκτελέσει. Πρέπει να πατήσει οπουδήποτε για να παίξει το AI. Η τελευταία κίνηση που έκανε το AI φαίνεται με τον μικρό μπλε κύκλο.



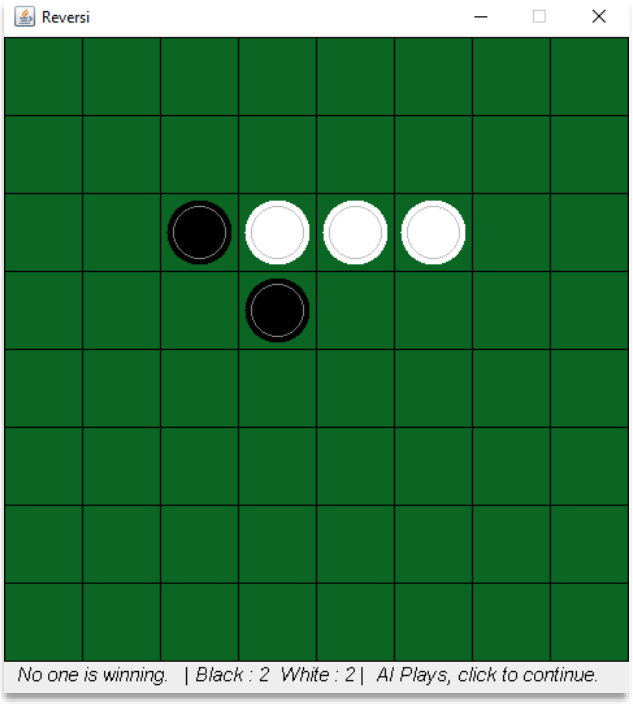
Ending Screen

Η τελική οθόνη του παραθύρου που εμφανίζεται αφού πατήσει στο ταμπλό ο χρήστης όταν δεν υπάρχουν διαθέσιμες κινήσεις ούτε για τον παίκτη ούτε για το AI.

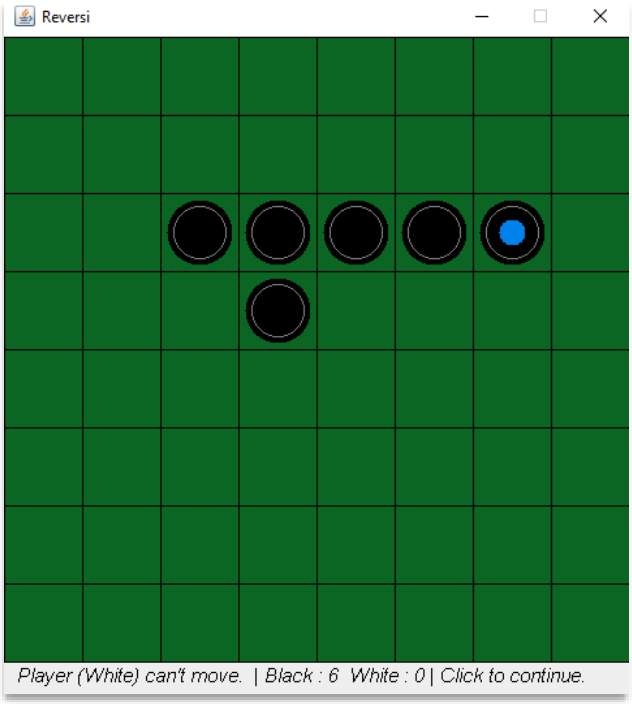
Για να ξαναπαίζει ο χρήστης πρέπει να πατήσει το κουμπί New Game, το οποίο ξανά αρχικοποιεί όλα τα στοιχεία του προγράμματος .

Το πάτημα του κουμπιού Exit τερματίζει την εκτέλεση του προγράμματος κλείνοντας το παράθυρο.

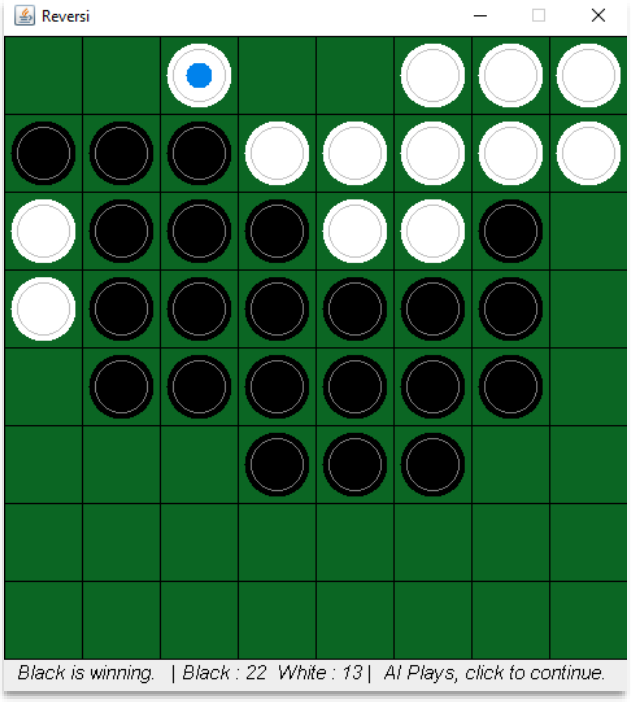
Το τελικό σκορ παίκτη, AI και αντίστοιχα μηνύματα τυπώνονται στο παράθυρο.



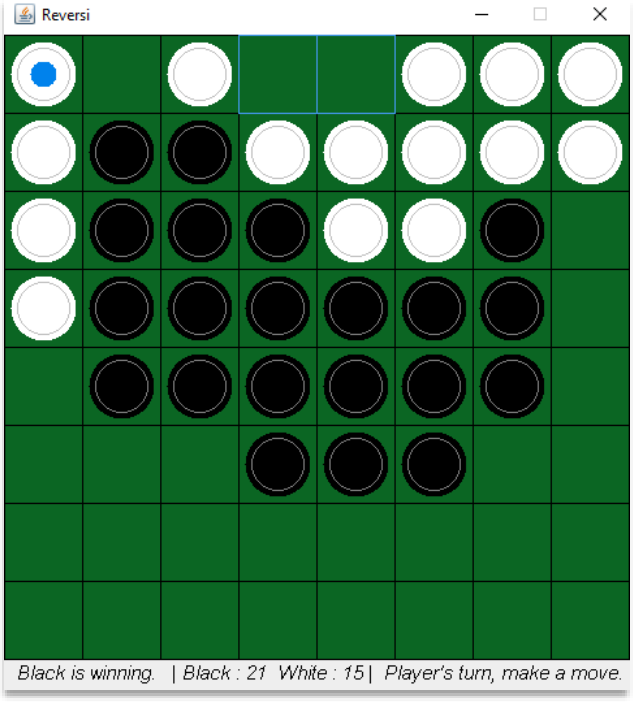
Στην κατάσταση αυτή (πάνω αριστερά) το AI (μαύρο) αναμένεται να κάνει την κίνηση που εγκλωβίζει όλα τα λευκά δισκάκια.



Όντως έτσι και έγινε, εφόσον η κίνηση αυτή θεωρείται «killer move», με σαφώς μεγαλύτερη βαρύτητα από τις άλλες.



Στην κατάσταση αυτή (πάνω αριστερά) το AI (λευκό) αναμένεται να πάει στην γωνία αντί να παγιδεύσει πχ 4 μαύρα δισκάκια.



Όντως έτσι και έγινε, εφόσον η κίνηση αυτή είναι κίνηση μεγαλύτερης βαρύτητας, λόγω γνωστής στρατηγικής του Reversi.