



ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
Τ: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

Μάθημα: ΜΥΥ602 – Τεχνητή Νοημοσύνη
Ακαδημαϊκό έτος: 2023 – 2024
Διδάσκων: Α. Λύκας
Ημερομηνία παράδοσης: 19 Μαΐου 2024

ΟΝΟΜΑΤΕΠΩΝΥΜΑ, ΑΜ ΟΜΑΔΑΣ:

ΜΠΟΥΡΑΝΤΑΣ ΛΑΖΑΡΟΣ, 5303

ΜΠΟΛΟΤΣΗΣ ΣΤΥΛΙΑΝΟΣ, 5429

ΧΑΝΛΑΡΙΔΟΥ ΔΑΝΑΗ, 5386



Εργαστηριακή Άσκηση 2 (Κατασκευή Παιγνίου) (10%)

Αλγόριθμος MINMAX

Ο αλγόριθμος Minmax είναι ένας αλγόριθμος αναζήτησης που χρησιμοποιείται σε παιχνίδια απόφασης με δύο παίκτες, για τη λήψη της βέλτιστης κίνησης. Στην προκειμένη περίπτωση, έχουμε μόνο δύο παίκτες: τον ανθρώπινο παίκτη και τον υπολογιστή. Ο αλγόριθμος Minmax εξετάζει το δέντρο των κινήσεων με αποκοπή σε κάθε βάθος, όπου το βάθος είναι η απόσταση από την αρχική κατάσταση του παιχνιδιού. Σε κάθε επίπεδο του δέντρου, αλλάζει ο παίκτης που κάνει την κίνηση, και ο αλγόριθμος εξετάζει όλες τις πιθανές κινήσεις που μπορεί να κάνει ο παίκτης αυτός σε αυτό το βήμα. Στο επόμενο επίπεδο, ο αλγόριθμος εξετάζει όλες τις πιθανές απαντήσεις του αντιπάλου παίκτη και ούτω καθεξής. Κάθε φορά που φτάνει σε ένα φύλλο του δέντρου, δηλαδή σε μια τελική κατάσταση του παιχνιδιού, υπολογίζει μια "εκτίμηση" του αποτελέσματος του παιχνιδιού από την σκοπιά του τρέχοντος παίκτη. Στην περίπτωση μας, αυτή η εκτίμηση μπορεί να είναι οι νίκες, οι ισοπαλίες και οι ήττες. Στη συνέχεια, ο αλγόριθμος Minmax επιστρέφει το καλύτερο αποτέλεσμα που μπορεί να επιτευχθεί για τον τρέχοντα παίκτη, λαμβάνοντας υπόψη τις εκτιμήσεις των παιδιών του κόμβου. Ο αλγόριθμος προχωρά αναδρομικά, αλλάζοντας τη σειρά των παικτών σε κάθε επίπεδο, με στόχο να επιλέξει την κίνηση που θα οδηγήσει στο βέλτιστο αποτέλεσμα για τον πρώτο παίκτη και στο χειρότερο αποτέλεσμα για τον δεύτερο παίκτη.

Ορισμός σταθερών

Ορίζονται οι σταθερές MAX, MIN, N, TIE, GAME_OVER, και CONTINUE, που αντιπροσωπεύουν τις διάφορες καταστάσεις του παιχνιδιού.

Δομή inform

Αυτή αποτελείται από:

- Έναν πίνακα (board) που αποθηκεύει την παρούσα κατάσταση του 3*3 πίνακα (N=3 στο enum).
- Έναν πίνακα με δείκτες (kinisis) ο οποίος παίρνει σαν όρισμα που είναι οι πιθανές μελλοντικές κινήσεις του minmax αλγόριθμου. (Ο minmax αλγόριθμος δηλαδή η βέλτιστη απάντηση του υπολογιστή θα έρθει όταν στον πίνακα υπάρχει ήδη το αρχικό S αλλά και η πρώτη κίνηση του χρήστη,



συνεπώς υπάρχουν 7 κενές θέσεις κάθε μία εκ των οποίων μπορεί να πάρει 3 πιθανά σύμβολα άρα $3 \cdot 7 = 21$).

- Έναν ακέραιο (next) που αποθηκεύει τον δείκτη του καλύτερου μονοπατιού στο δέντρο.
- Έναν ακέραιο (vacancies) που αποθηκεύει τον αριθμό των κινήσεων που έχει το παιχνίδι μέχρι να τερματίσει.

Συναρτήσεις

- **finalStateCheck:** Η συνάρτηση αυτή ελέγχει αν βρισκόμαστε σε κάποια τελική κατάσταση, δηλαδή αν υπάρχει το CSE ή το ESC σε κάποια διαγώνια παράλληλη ή κάθετη σειρά του πίνακα και τερματίζει το πρόγραμμα (GAME_OVER*υπάρχει σε ανάλογο enum) αλλιώς το πρόγραμμα συνεχίζει (CONTINUE*υπάρχει σε ανάλογο enum).
- **user_move:** Η συνάρτηση αυτή είναι υπεύθυνη να ζητά από τον χρήστη την γραμμή, την στήλη αλλά και το γράμμα το οποίο θέλει να εισάγει στον πίνακα. Ζητάμε ξεχωριστά τις γραμμές και τις στήλες ώστε να είναι πιο εύκολη η κατανόηση από τον χρήστη. Στόχος μας επίσης είναι στην περίπτωση που ο χρήστης βάλει συντεταγμένες που είναι εκτός πίνακα ή σε σημείο όπου είναι ήδη καλυμμένο, το πρόγραμμα να μην το επιτρέψει και να του ζητήσει να τις εισάγει ξανά. Με το ίδιο τρόπο σκέψης δουλεύουμε και όταν θέλουμε ο χρήστης να εισάγει το σύμβολο, του συμβόλου κάνοντας έλεγχο εγκυρότητας και σε περίπτωση σφάλματος δίνουμε την δυνατότητα να το εισάγει ξανά. Τέλος, ενημερώνεται ο πίνακας και μειώνεται ο αριθμός των κενών θέσεων κατά μία.



Δημιουργία παιδιών κόμβων

- **makeChildren** : Η συνάρτηση αυτή είναι υπεύθυνη για την δημιουργία του δέντρου των παιδιών. Αρχικά με την `for` διατρέχει όλα τα κελιά και όπου βρίσκει “?”, δημιουργεί ακόμα μια επανάληψη ώστε να εξετάσει και για E και για S αλλά και για C. Για κάθε μια από αυτές τις επιλογές, δημιουργεί ένα νέο παιδί. Έπειτα δημιουργεί και καλεί τη συνάρτηση `copy_board`, η οποία απλά αντιγράφει τον πίνακα στο νέο παιδί. Αφού έχει συμβεί αυτό μειώνουμε τον αριθμό των κενών κελιών κατά ένα και τοποθετούμε την εκάστοτε επιλογή στον πίνακα. Τέλος, καλούμε την `tree_MinMax_tree` για να αξιολογήσει αν το νέο παιδί είναι βέλτιστο και αυξάνουμε την θέση κατά μία.
- **tree_MinMax_tree**: Ελέγχει αν ισχύουν οι προϋποθέσεις για να τερματίσει το πρόγραμμα με τελικές καταστάσεις, την νίκη του υπολογιστή, την νίκη του χρήστη ή την ισοπαλία. Στη συνέχεια, είναι υπεύθυνη στο να αλλάζει μετά από κάθε κίνηση το ποιος παίζει. πχ. αν παίζει ο χρήστης και ολοκληρώσει την κίνηση του έπειτα, θα ακολουθήσει η κίνηση του υπολογιστή. Στην συγκεκριμένη συνάρτηση, αρχικά καλούμε τις δυο παραπάνω οι οποίες είναι υπεύθυνες για το αν έχει τελειώσει το παιχνίδι και ποιος παίζει μετά. Έπειτα, καλούμε την `makeChildren` που δημιουργεί τα παιδιά (θα δούμε παρακάτω πιο αναλυτικά την λειτουργία της). Τέλος, στόχος του αλγορίθμου είναι η εύρεση της βέλτιστης κίνησης για τον υπολογιστή, κάτι που γίνεται όταν ο αλγόριθμος διατρέχει κάθε παιδί ξεχωριστά και αποθηκεύει ποιο είναι, ανανεώνοντας το όταν βρίσκεται ένα βέλτιστο, μέχρι να τελειώσει το δέντρο.

Εκκίνηση του παιχνιδιού

- **game_start**: Αρχικά, αφού εκχωρίσουμε την απαραίτητη μνήμη δυναμικά, καλούμε την συνάρτηση `game_preparation` ώστε εν τέλει στην `main` να υπάρχει μόνο η `game_start`. Έπειτα δημιουργούμε έναν βρόγχο που ελέγχει αν κέρδισες κάποιος ή ήρθε ισοπαλία και τερματίζει το παιχνίδι αν ισχύει κάτι από αυτά. Αλλιώς εναλλάσσουμε την σειρά μεταξύ υπολογιστή και χρήστη, καλώντας τις ανάλογες συναρτήσεις που έχουμε ήδη δημιουργήσει και αναφέρει.
- **print_board**: Η συνάρτηση αυτή εκτυπώνει τον πίνακα.
- **game_preparation**: Στην συνάρτηση αυτή αρχικά ενημερώνει το σύστημα ότι υπάρχουν 9 κενά τετραγωνάκια και τα γεμίζει με ?. Έπειτα ζητά από τον χρήστη να διαλέξει που θέλει να βάλει το αρχικό S ανάμεσα στην δεξιά και την αριστερή της κεντρικής (βάση εκφώνησης)



Αυτές οι συναρτήσεις συνθέτουν τον βασικό μηχανισμό του παιγνίου, ο οποίος χρησιμοποιεί τον αλγόριθμο Minmax για να αποφασίζει την καλύτερη δυνατή κίνηση για το πρόγραμμα και να αναλύει τις πιθανές καταστάσεις του παιγνίου μέχρι την τελική κατάσταση δεδομένου ότι το παίγνιο ξεκινά από αρχική κατάσταση όπου υποχρεωτικά υπάρχει το S είτε στην αριστερή είτε στη δεξιά θέση της μεσαίας γραμμής.

```
-----
The table should start with S in one of the positions left or right of the middle one.
Press 1 for left or 2 for right: 2
GET READY. THE GAME HAS JUST STARTED!!!

? ? ?
? ? S
? ? ?

USER TURN
-----
Choose the row for the first letter (0-first row/1-second row/2-third row): 0
Choose the col for the first letter (0-first col/1-second col/2-third col): 1
Choose between C , S , E: E
? E ?
? ? S
? ? ?

COMPUTER TURN
-----
E E ?
? ? S
? ? ?

USER TURN
-----
Choose the row for the first letter (0-first row/1-second row/2-third row): 2
Choose the col for the first letter (0-first col/1-second col/2-third col): 2
Choose between C , S , E: C
E E ?
? ? S
? ? C

COMPUTER TURN
-----
E E E
? ? S
? ? C

COMPUTER WON
-----
GAME OVER
-----

...Program finished with exit code 0
Press ENTER to exit console.
```



ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
Τ: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

```
*****
-----
The table should start with S in one of the positions left or right of the middle one.
Press 1 for left or 2 for right: 1
GET READY. THE GAME HAS JUST STARTED!!!

? ? ?
S ? ?
? ? ?

USER TURN

Choose the row for the first letter (0-first row/1-second row/2-third row): 2
Choose the col for the first letter (0-first col/1-second col/2-third col): 2
Choose between C , S , E: C
? ? ?
S ? ?
? ? C

COMPUTER TURN

? ? ?
S ? ?
? ? C

USER TURN

Choose the row for the first letter (0-first row/1-second row/2-third row): 2
Choose the col for the first letter (0-first col/1-second col/2-third col): 0
Choose between C , S , E: C
? ? ?
S ? ?
C ? C

USER WON

GAME OVER

*****
-----
...Program finished with exit code 0
Press ENTER to exit console.
```