

Περιγραφή Παιγνίου και Υλοποίησης MINIMAX – Άσκηση 2

Η κατάσταση του παιγνιδιού ορίζεται ως ένας πίνακας 4x3 χαρακτήρων (`char[][] board`), όπου κάθε κελί μπορεί να περιέχει:

- τον χαρακτήρα `'.'` για κενό κελί,
- `'X'` για τον υπολογιστή (παίκτης MAX),
- `'O'` για τον χρήστη (παίκτης MIN).

Το παιχνίδι παίζεται εναλλάξ, ξεκινώντας από μια αρχική κατάσταση όπου υπάρχουν ήδη τοποθετημένα ένα `'X'` και ένα `'O'` σε μη διαδοχικές θέσεις. Οι παίκτες βάζουν σύμβολά τους σε κενά κελιά, μέχρι να βρεθεί κάποιο από τα δύο μοτίβα νίκης ή να γεμίσει το ταμπλό.

Η αξία των τελικών καταστάσεων προσδιορίζεται ως εξής:

- Αν σχηματιστεί το μοτίβο `"XOX"` (οριζόντια, κάθετα ή διαγώνια), τότε ο MAX κερδίζει και η κατάσταση παίρνει τιμή `+1`.
- Αν σχηματιστεί το μοτίβο `"OXO"`, τότε ο MIN κερδίζει και η κατάσταση παίρνει τιμή `-1`.
- Αν το ταμπλό γεμίσει χωρίς να έχει σχηματιστεί κάποιο νικητήριο μοτίβο, τότε έχουμε ισοπαλία και η κατάσταση παίρνει τιμή `0`.

Η υλοποίηση γίνεται με χρήση του αλγορίθμου **MINIMAX**, ο οποίος υλοποιείται αναδρομικά στη μέθοδο `minimax(...)`. Ο αλγόριθμος υπολογίζει όλα τα δυνατά μελλοντικά σενάρια κινήσεων, εναλλάσσοντας τη σειρά MAX και MIN, και επιστρέφει τη βέλτιστη έκβαση (`1`, `0` ή `-1`) κάθε φορά. Ο MAX προσπαθεί να μεγιστοποιήσει το αποτέλεσμα (προτιμά νίκες), ενώ ο MIN προσπαθεί να το ελαχιστοποιήσει (προτιμά να αποφύγει ήττα).

Η μέθοδος `findBestMove(...)` καλεί τη `minimax(...)` για κάθε διαθέσιμη κίνηση του υπολογιστή και επιλέγει εκείνη που οδηγεί στη βέλτιστη δυνατή έκβαση, ακόμα και αν ο παίκτης παίζει άριστα. Οι κινήσεις αξιολογούνται όλες, χωρίς ευρετικές ή κλάδεμα (πλήρες δέντρο απόφασης).

Τέλος, η μέθοδος `checkWin(...)` ελέγχει για την ύπαρξη των μοτίβων `"XOX"` και `"OXO"` σε οριζόντια, κάθετη και διαγώνια μορφή, ενώ η `isGameOver(...)` εντοπίζει αν το παιχνίδι πρέπει να τερματιστεί λόγω νίκης ή ισοπαλίας.

Η προσέγγιση αυτή εξασφαλίζει ότι ο υπολογιστής παίζει πάντα βέλτιστα, οδηγώντας σε νίκη ή ισοπαλία αν δεν κάνει λάθος ο χρήστης.