

Μάθημα: Τεχνητή Νοημοσύνη

Άσκηση 2: Κατασκευή παιγνίου

Ονοματεπώνυμο: Κωνσταντίνος Βαρδάκας

AM: 522

Email: [pcs0522@uoi.gr](mailto:pcs0522@uoi.gr)

## Εισαγωγή

Η παρούσα αναφορά περιγράφει την υλοποίηση ενός προγράμματος σε γλώσσα C για το παίγνιο δύο παικτών (MAX και MIN) σε πλέγμα 4x3. Το πρόγραμμα επιτρέπει στον παίκτη MAX (υπολογιστής) να παίζει βέλτιστα εναντίον του παίκτη MIN (χρήστης) χρησιμοποιώντας τον αλγόριθμο MINIMAX με αναδρομή.

## Ορισμός της Κατάστασης του Παιγνίου

Η κατάσταση του παιγνίου αναπαρίσταται από ένα πλέγμα 4x3, υλοποιημένο ως δισδιάστατος πίνακας `char board[ROWS][COLS]`, όπου `ROWS=4` και `COLS=3`. Κάθε κελί του πλέγματος μπορεί να περιέχει:

- X: Τοποθετήθηκε από τον παίκτη MAX.
- O: Τοποθετήθηκε από τον παίκτη MIN.
- \_ : Κενή θέση.

Η αρχική κατάσταση του παιγνίου ορίζεται ώστε να περιλαμβάνει ένα «X» και ένα «O» σε τυχαίες, μη διαδοχικές θέσεις (οριζόντια, κάθετα ή διαγώνια), μειώνοντας τον χώρο καταστάσεων. Η συνθήκη μη διαδοχικότητας ελέγχεται με τη συνάρτηση `is_adjacent`, η οποία χρησιμοποιεί τη απόσταση Chebyshev ( $\max(|x_1, x_2|), \max(|y_1, y_2|)$ ) για να εξασφαλίσει ότι οι θέσεις δεν είναι γειτονικές.

Οι κινήσεις των παικτών εναλλάσσονται, με τον MAX να παίζει πρώτος, τοποθετώντας το αντίστοιχο σύμβολο σε οποιαδήποτε κενή θέση. Το παίγνιο τερματίζει

όταν σχηματίζεται τουλάχιστον μία τριάδα XOX ή OXO (οριζόντια, κάθετα ή διαγώνια) ή όταν το πλέγμα γεμίσει χωρίς τριάδες (ισοπαλία).

### Αποτίμηση Τελικών Καταστάσεων

Η αποτίμηση των τελικών καταστάσεων στο παίγνιο βασίζεται στον αριθμό των τριάδων XOX και OXO που σχηματίζονται στο πλέγμα. Συγκεκριμένα, η κατάσταση θεωρείται νίκη για τον παίκτη MAX όταν ο αριθμός των τριάδων XOX υπερβαίνει τον αριθμό των τριάδων OXO, με τιμή +10. Αντίθετα, η κατάσταση θεωρείται νίκη για τον παίκτη MIN όταν ο αριθμός των τριάδων OXO είναι μεγαλύτερος από τον αριθμό των τριάδων XOX, με τιμή -10. Σε περίπτωση ισοπαλίας, δηλαδή όταν ο αριθμός των τριάδων XOX ισούται με τον αριθμό των τριάδων OXO ή όταν το πλέγμα είναι πλήρες χωρίς καμία τριάδα, η τιμή αποτίμησης είναι 0.

Η διαδικασία αποτίμησης υλοποιείται μέσω της συνάρτησης `is_game_over`, η οποία εξετάζει το πλέγμα για να μετρήσει όλες τις τριάδες XOX και OXO. Η συνάρτηση επιστρέφει 1 για νίκη του MAX, -1 για νίκη του MIN, 2 για ισοπαλία, ή 0 αν το παίγνιο συνεχίζεται (δηλαδή δεν υπάρχουν τριάδες και το πλέγμα δεν είναι πλήρες). Οι τιμές αποτίμησης +10, -10 και 0 επιλέχθηκαν ώστε να διαχωρίζουν σαφώς τις καταστάσεις νίκης, ήττας και ισοπαλίας, ενώ είναι συμβατές με τον αλγόριθμο MINIMAX, ο οποίος επιδιώκει τη μεγιστοποίηση της τιμής για τον MAX και την ελαχιστοποίηση για τον MIN.

### Υλοποίηση του MiniMax

Η υλοποίηση του αλγορίθμου Minimax βασίζεται στη συνάρτηση `minimax(int is_max)`, η οποία υλοποιεί αναδρομική εξερεύνηση όλων των δυνατών καταστάσεων του πλέγματος για να επιλέξει τη βέλτιστη κίνηση. Στην αρχή κάθε κλήσης, καλείται η `is_game_over`, η οποία ελέγχει αν η παρτίδα έχει τερματιστεί, επιστρέφοντας αντίστοιχα θετική τιμή για νίκη του παίκτη MAX, αρνητική για νίκη του MIN και μηδέν για ισοπαλία. Αυτές οι τιμές χρησιμοποιούνται για την αξιολόγηση των φύλλων στο δέντρο του παιχνιδιού.

Ο αλγόριθμος λειτουργεί με εναλλαγή ρόλων μεταξύ του παίκτη που μεγιστοποιεί το σκορ (MAX, προγράμματος) και του παίκτη που ελαχιστοποιεί (MIN, χρήστη). Σε κάθε κίνηση, η συνάρτηση δοκιμάζει όλες τις διαθέσιμες θέσεις, τοποθετεί προσωρινά το αντίστοιχο σύμβολο, καλεί αναδρομικά τον minimax για την επόμενη κατάσταση και στη συνέχεια αναιρεί την κίνηση με την επανατοποθέτηση του χαρακτήρα «\_» στο συγκεκριμένο κελί. Ο MAX επιλέγει τη μέγιστη τιμή από τις διαθέσιμες κινήσεις, ενώ ο MIN επιλέγει τη μικρότερη, διαμορφώνοντας έτσι βέλτιστη στρατηγική για κάθε παίκτη.

Η συνάρτηση `computer_move()` αξιοποιεί τον αλγόριθμο Minimax για να επιλέξει την καλύτερη δυνατή κίνηση για τον υπολογιστή, ο οποίος παίζει ως παίκτης «X». Για κάθε διαθέσιμη θέση στο πλέγμα, τοποθετεί προσωρινά το σύμβολο «X» και καλεί τη συνάρτηση `minimax(0)` ώστε να υπολογίσει την απόδοση αυτής της πιθανής κίνησης, θεωρώντας ότι ο επόμενος παίκτης (ο χρήστης, MIN) θα προσπαθήσει να ελαχιστοποιήσει το σκορ. Μετά την αξιολόγηση, η κίνηση αναιρείται και διατηρείται η καλύτερη από πλευράς σκορ. Τελικά, ο υπολογιστής τοποθετεί το «X» στη θέση με το υψηλότερο σκορ, εξασφαλίζοντας βέλτιστη στρατηγική με βάση τις προβλέψεις του Minimax.

### Πιθανές τερματικές καταστάσεις

Ακολουθούν τα πιθανά σενάρια τελικής κατάστασης:

- 1) Νίκη του παίκτη (MIN)

```
  0 1 2
0 X X 0
1 X X 0
2 0 _ _
3 0 _ _

0 XOX counts (MAX) vs 1 OXO counts (MIN)
You (MIN) win!
```

- 2) Νίκη του προγράμματος (MAX)

```

  0 1 2
0 X 0 0
1 X 0 X
2 - - X
3 - - -

2 XOX counts (MAX) vs 0 OXO counts (MIN)
Computer (MAX) wins!

```

3) Ισοπαλία λόγω ισάριθμων XOX και OXO

```

  0 1 2
0 X 0 0
1 X - X
2 0 - 0
3 0 X X

1 XOX counts (MAX) vs 1 OXO counts (MIN)
It's a draw!

```

4) Ισοπαλία λόγω απουσίας XOX ή OXO

```

  0 1 2
0 0 X X
1 0 X X
2 0 0 X
3 0 0 X

0 XOX counts (MAX) vs 0 OXO counts (MIN)
It's a draw!

```