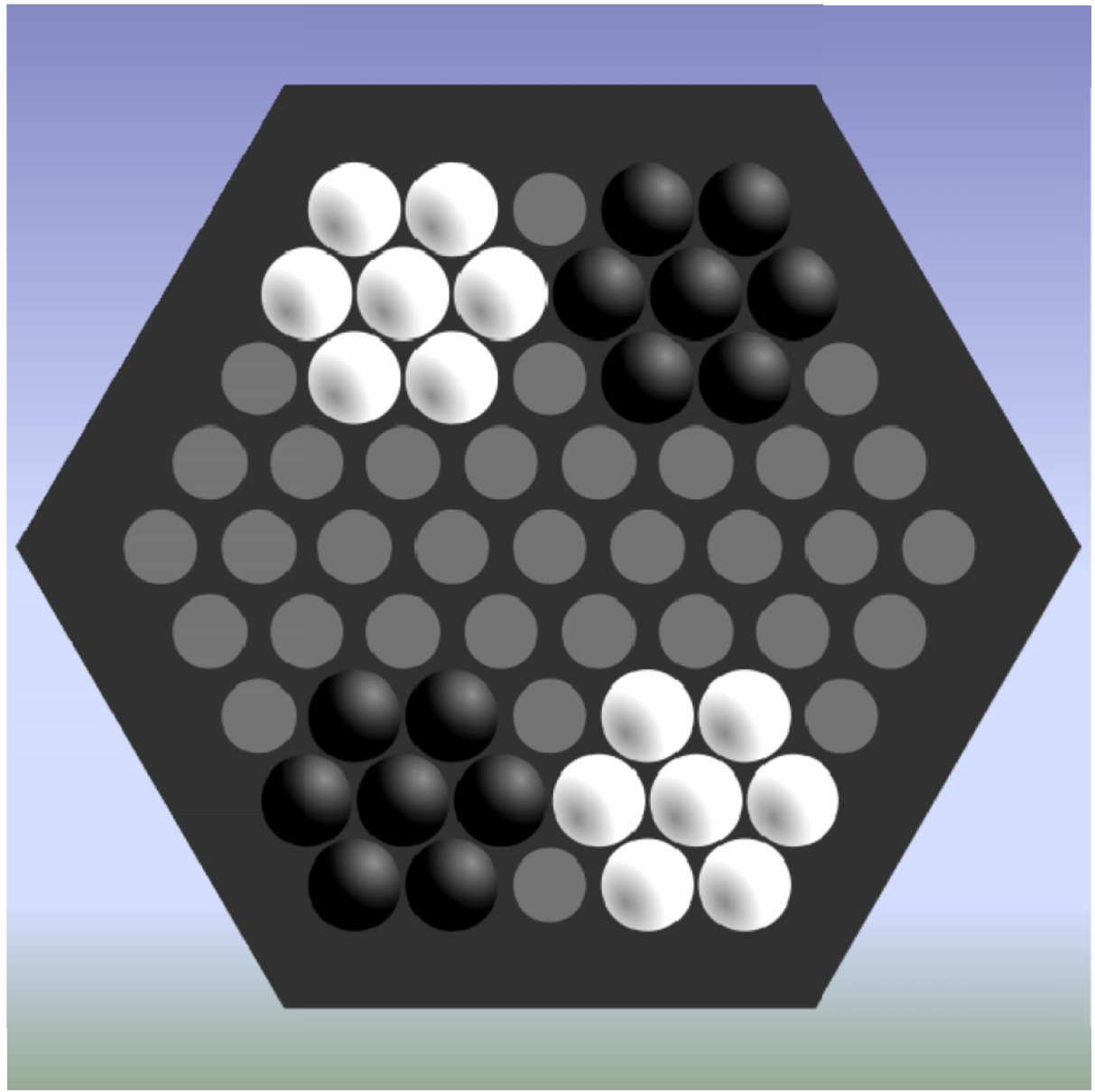


Δομές Δεδομένων – Εργασία Β΄

ABALONE

*Belgian daisy starting position



Περιγραφή του προβλήματος:

Σκοπός του προβλήματος ήταν η δημιουργία των μεθόδων «public vector <Move>findAllMoves(Game game)» και «public double evaluate(Game game)» οι οποίες βρίσκονται μέσα στην κλάση Player. Οι συγκεκριμένες μέθοδοι, σε συνδυασμό με τις υπόλοιπες που μας δοθήκαν θα υλοποιεί έναν αυτόματο παίχτη.

Περιγραφή του αλγορίθμου:

Για την επίλυση του προβλήματος δημιουργήσαμε τις μεθόδους
«*public vector <Move>findAllMoves(Game game)*» και
«*public double evaluate(Game game)*».

public vector <Move>findAllMoves(Game game):

Σκοπός της μεθόδου είναι η επιστροφή ενός πίνακα που περιέχει
όλες τις διαθέσιμες κινήσεις του παίχτη που έχει σειρά.

Αρχικά, δημιουργήσαμε τον πίνακα «moves», ο οποίος θα δέχεται
όλες τις πιθανές κινήσεις.

Έχουμε τρία είδη κινήσεων :

i) 1 πούλι

Αναζητώντας το from1 ψάχνουμε όλα τα κελιά του ταμπλό
που έχουν πούλι ίδιου χρώματος με αυτό του παίχτη που
έχει σειρά. Για κάθε τέτοιο πούλι ελέγχουμε όλα τα
γειτονικά του. Αν κάποιο από αυτά είναι κενό και
βρίσκεται μέσα στο ταμπλό, τότε αυτή η θέση μπορεί να
θεωρηθεί ως πιθανός προορισμός του πουλιού (to2).

ii) 2 πουλιά μαζί

Αναζητώντας το from2 ψάχνουμε όλα τα κελιά του ταμπλό που έχουν πούλι ίδιου χρώματος με αυτό του παίχτη που έχει σειρά. Για κάθε τέτοιο πούλι που μπορεί να είναι from2, ελέγχουμε τα μισά γειτονικά του ώστε να αποφύγουμε τις διπλότυπες κινήσεις. Αν κάποιο από αυτά είναι κενό, βρίσκεται μέσα στο ταμπλό και η μετακίνηση του from2 σε συνδυασμό με το from1 είναι εφικτή, τότε αυτή η θέση μπορεί να θεωρηθεί ως πιθανός προορισμός του from2 (to2).

iii) 3 πούλια μαζί

Αναζητώντας το from3 ψάχνουμε όλα τα κελιά του ταμπλό που έχουν πούλι ίδιου χρώματος με αυτό του παίχτη που έχει σειρά. Για κάθε τέτοιο πούλι που μπορεί να είναι from3, ελέγχουμε τα μισά γειτονικά του ώστε να αποφύγουμε τις διπλότυπες κινήσεις. Αν κάποιο από αυτά είναι κενό, βρίσκεται μέσα στο ταμπλό και η μετακίνηση του from3 σε συνδυασμό με το from1 και το from2 είναι εφικτή, τότε αυτή η θέση μπορεί να θεωρηθεί ως πιθανός προορισμός του from3 (to3).

Όλες αυτές οι πιθανές κινήσεις προστίθενται στον πίνακα «moves», οποίος και επιστρέφεται.

public double evaluate(Game game):

Σκοπός της μεθόδου είναι η αξιολόγηση της τρέχουσας θέσης του ταμπλό.

Έχουμε 3 κριτήρια αξιολόγησης:

i) Αριθμός πουλιών εκτός ταμπλό

Με την χρήση μιας εντολής ελέγχου Switch και της μεθόδου «public int getDeadPieces()», μετράμε για κάθε παίχτη τον αριθμό των πουλιών που βρίσκονται εκτός ταμπλό και του δίνουμε μια βαθμολογία.

Συγκεκριμένα:

<u>Χαμένα Πούλια</u>	<u>Βαθμολογία</u>
0	100
1	90
2	75
3	55
4	30
5	10

ii) Θέση πουλιού στο ταμπλό

Για κάθε παίχτη ελέγχουμε τη θέση του κάθε πουλιού και ανάλογα με το πόσο κοντά στο κέντρο βρίσκεται, τόσο μεγαλύτερη βαθμολογία παίρνει.

Συγκεκριμένα:

<u>Θέσεις από το κέντρο</u>	<u>Βαθμολογία</u>
4	-1
3	-0.5
2	+0.5
1	+1
0	+3

iii) Συνεκτικότητα πουλιών

Για κάθε πούλι του κάθε παίχτη ελέγχουμε το κάθε γειτονικό του. Εάν αυτό είναι κενό ή κατειλημμένο, το βαθμολογούμε ανάλογα.

Συγκεκριμένα:

<u>Γειτονικό κελί</u>	<u>Βαθμολογία</u>
Κενό	0.5
Πούλι αντιπάλου	0.25
Δικό μας πούλι	1

Στη συνέχεια, για κάθε παίχτη, προσθέτουμε τις επιμέρους βαθμολογίες του. Έτσι προκύπτει μια γενική βαθμολογία για τον κάθε παίχτη.

Τέλος, για να μην είναι μεροληπτική η αξιολόγηση, αφαιρούμε τη γενική βαθμολογία του παίχτη που έχει σειρά από αυτή του αντιπάλου. Η τελική βαθμολογία είναι η ζητούμενη και επιστρέφεται.