

ΕΡΓΑΣΙΑ ΣΤΟΝ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

ΤΕΤΑΡΤΟ ΠΑΡΑΔΟΤΕΟ

Υλοποίηση εικονικών συναρτήσεων και πρότυπων συναρτήσεων σε C++

Το τέταρτο παραδοτέο της εργασίας απαιτεί τη δημιουργία των κληρονομημένων κλάσεων των **παικτών για διάφορες στρατηγικές του παιχνιδιού** και των ενεργειών τους στο παιχνίδι Blokus. Συγκεκριμένα, θα υλοποιηθούν οι παρακάτω συναρτήσεις που βρίσκονται όλες στα αρχεία **computerplayer.cpp** και **algorithms.h**:

1. Εικονική συνάρτηση evaluateBoard

Η συνάρτηση αυτή αφορά τη στρατηγική του ComputerPlayer. Όταν καλείται, η συνάρτηση επιστρέφει ένα score για την κατάσταση του ταμπλό, το οποίο μπορεί να χρησιμοποιηθεί από τον παίκτη για να αποφασίσει την κίνησή του (εφόσον καλέστηκε από την evaluateMove). Το score υπολογίζεται ως το πλήθος των τετραγώνων του ταμπλό που είναι κατειλημμένα από τον παίκτη μείον το πλήθος των τετραγώνων που είναι κατειλημμένα από τον αντίπαλό του. Η συνάρτηση evaluateBoard είναι μια **εικονική συνάρτηση** που όμως **δεν είναι γνήσια**, οπότε υλοποιείται μόνο για τον ComputerPlayer και όχι για τον RandomPlayer.

Υπόδειξη: μπορείτε να χρησιμοποιήσετε την συνάρτηση squareBelongsToPlayer της κλάσης Board για την υλοποίησή σας.

2. Πρότυπη συνάρτηση shuffle

Η συνάρτηση αυτή αφορά τη στρατηγική του RandomPlayer. Καλείται ώστε να ανακατευτούν οι πιθανές κινήσεις του παίκτη ώστε στη συνέχεια να επιλεγεί μία από αυτές τυχαία. Η συνάρτηση παίρνει ως είσοδο ένα δυναμικό πίνακα δυναμικών αντικειμένων (διπλό pointer) και ανακατεύει τα αντικείμενα στις θέσεις του πίνακα. Ο αλγόριθμος ανακατέματος είναι ο παρακάτω:

```
ΑλγόριθμοςΑνακατέματοςΣτοιχείωνΕνόςΠίνακα(Πίνακας, Μέγεθος){
    Για το i (αριθμοδείκτη πίνακα) όπου i από το τελευταίο έως και το δεύτερο στοιχείο{
        j = τυχαίος ακέραιος αριθμός στο διάστημα [0, i]
        αντιμετάθεση του στοιχείου Πίνακας[i] με το στοιχείο Πίνακας[j]
    }
}
```

Υπόδειξη: για το loop μην ξεχάσετε ότι οι αριθμοδείκτες πινάκων ξεκινούν από το 0, οπότε το πρώτο στοιχείο είναι το 0, το δεύτερο το 1, κ.ο.κ. Επίσης, διευκρινίζεται ότι το παραπάνω loop θα πρέπει να εκτελεστεί κανονικά όταν το i δείχνει στο δεύτερο στοιχείο (δηλαδή το i πάει **μέχρι και** το δεύτερο στοιχείο).

3. Πρότυπη συνάρτηση getElementWithMaxScore

Η συνάρτηση αυτή αφορά τη στρατηγική του ComputerPlayer. Καλείται ώστε να βρεθεί η κίνηση με το μέγιστο score από αυτές που μπορεί να παίξει ο παίκτης. Η συνάρτηση παίρνει ως είσοδο ένα δυναμικό πίνακα δυναμικών αντικειμένων (διπλό pointer) και έναν δυναμικό πίνακα από ακεραίους. Κάθε στοιχείο του πρώτου πίνακα έχει το αντίστοιχο score στο δεύτερο πίνακα. Η συνάρτηση επιστρέφει το στοιχείο του πρώτου πίνακα που βρίσκεται στη θέση που αντιστοιχεί στο μέγιστο score. Αν το μέγιστο score εμφανίζεται για περισσότερα από ένα στοιχεία, τότε επιστρέφεται το πρώτο στοιχείο για το οποίο εμφανίζεται. Ένα παράδειγμα είναι το παρακάτω:

```
Πίνακας myArray: [A, B, C, D, E, F, G, H]
Πίνακας myArrayScore: [3, 5, 2, 7, 1, 6, 7, 6]
```

Στο παράδειγμα το μέγιστο score στον myArrayScore είναι το 7. Οπότε η συνάρτηση πρέπει να επιστρέψει το πρώτο στοιχείο του myArray που αντιστοιχεί σε αυτό το score, δηλαδή το D.

Εγκατάσταση

Για να βάλετε το project στο CodeBlocks, αρκεί να αποσυμπίεσετε το αρχείο Blokus.zip σε μια τοποθεσία του σκληρού σας δίσκου, και στη συνέχεια από το περιβάλλον του CodeBlocks να επιλέξετε File → Open... και να επιλέξετε το Blokus.crb από τον φάκελο που μόλις αποσυμπίεσατε το project.

Παρατηρήσεις

Η υλοποίηση θα πρέπει να γίνει στη C++ και να μπορεί να ανοίξει με το CodeBlocks, **με τις εκδόσεις που χρησιμοποιούμε** στο πλαίσιο του μαθήματος. Ο κώδικάς σας θα πρέπει να είναι καλά τεκμηριωμένος, ώστε να είναι παντού σαφείς οι λεπτομέρειες υλοποίησης.

Για την υλοποίηση, σας δίνονται τα αρχεία κεφαλίδων .h των κλάσεων/συναρτήσεων που πρέπει να υλοποιήσετε καθώς και κάποιες βοηθητικές κλάσεις/συναρτήσεις. Επιπλέον, σας δίνεται ο κώδικας της συνάρτησης main (main.cpp). **Σε καμία περίπτωση δεν επιτρέπεται να επέμβετε στον κώδικα των κλάσεων και των συναρτήσεων αυτών. Σε περίπτωση που το κάνετε, η εργασία σας αυτομάτως θεωρείται λανθασμένη και μηδενίζεται. Θα πρέπει να γράψετε κώδικα μόνο στα αρχεία computerplayer.cpp και algorithms.h.**

Παραδοτέο

Το παραδοτέο θα είναι ένα αρχείο zip με όνομα Blokus.zip που θα περιλαμβάνει **όλο το project (τα αρχεία που θα υλοποιήσετε αλλά και αυτά που σας έχουν δοθεί)**, δηλαδή ακριβώς ίδιο με το αρχείο Blokus.zip που δίνεται, φυσικά με τον κώδικα υλοποιημένο. Επιπλέον, προτείνεται πριν δημιουργήσετε το αρχείο zip, να κάνετε Clean το project (που γίνεται από το Codeblocks επιλέγοντας στο μενού Build → Clean).

Προθεσμία υποβολής

Το παραδοτέο πρέπει να παραδοθεί μέχρι τις **23:59 της Τετάρτης 8 Ιουνίου**. Καμία παρέκκλιση δε θα γίνει από την παραπάνω προθεσμία.