

ΕΡΓΑΣΙΑ ΣΤΟΝ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

ΤΡΙΤΟ ΠΑΡΑΔΟΤΕΟ

Υλοποίηση κληρονομικότητας και συναρτήσεων δεικτών σε C++

Το τρίτο παραδοτέο της εργασίας απαιτεί τη δημιουργία των κληρονομημένων κλάσεων των παικτών του παιχνιδιού του Blokus. Συγκεκριμένα, θα υλοποιηθούν οι παρακάτω κλάσεις **που βρίσκονται όλες στα αρχεία player.h και player.cpp (και υλοποιημένες στην playerpieces.cpp)**:

1. Κλάση Player

Η κλάση αυτή δημιουργεί τα αντικείμενα των παικτών του παιχνιδιού. Θα πρέπει να περιλαμβάνει μεταβλητές για το id, το όνομα και το σύμβολο (χαρακτήρα) του παίκτη, καθώς επίσης και τα κομμάτια του παίκτη και το πλήθος τους. Ο παίκτης με id 0 θα έχει όνομα Player 1 και χαρακτήρα #, ενώ ο παίκτης με id 1 θα έχει όνομα Player 2 και χαρακτήρα O. Επίσης η κλάση αυτή θα πρέπει να περιλαμβάνει συναρτήσεις για την αρχικοποίηση και την καταστροφή του αντικειμένου (constructor και destructor), συναρτήσεις επιστροφής μεταβλητών (getters), καθώς επίσης και συναρτήσεις που επιστρέφουν το κάθε κομμάτι και το πλήθος των κομματιών του παίκτη που είναι διαθέσιμα/δεν έχουν τοποθετηθεί. Τέλος, περιλαμβάνονται ήδη υλοποιημένες συναρτήσεις για τη δημιουργία των κομματιών του παίκτη και την εμφάνισή τους.

2. Κλάση HumanPlayer

Η κλάση αυτή δημιουργεί τα αντικείμενα των παικτών-ανθρώπων. Κληρονομεί την κλάση Player. Για την αρχικοποίηση του παίκτη, θα πρέπει να περιλαμβάνει μια συνάρτηση αρχικών συνθηκών (constructor) με παράμετρο το id του, καθώς και μια συνάρτηση αρχικών συνθηκών με παραμέτρους το id και το όνομα του παίκτη ώστε να είναι εφικτό να δοθεί κάποιο άλλο όνομα στον παίκτη (πέρα από τα Player 1, Player 2). Επίσης, περιλαμβάνει μια ήδη υλοποιημένη συνάρτηση για την τοποθέτηση των κομματιών του παίκτη.

3. Κλάση ComputerPlayer

Η κλάση αυτή δημιουργεί τα αντικείμενα των παικτών-υπολογιστών. Κληρονομεί την κλάση Player. Για την αρχικοποίηση του παίκτη, θα πρέπει να περιλαμβάνει μια συνάρτηση αρχικών συνθηκών (constructor) με παράμετρο το id του. Επίσης, περιλαμβάνει μια ήδη υλοποιημένη συνάρτηση για την τοποθέτηση των κομματιών του παίκτη. Ο παίκτης παίζει τυχαία, οπότε η κλάση θα πρέπει να περιλαμβάνει τρεις συναρτήσεις για τη δημιουργία μιας τυχαίας κίνησης, μια συνάρτηση που επιστρέφει τυχαία το id ενός κομματιού που θα παιχτεί (από το 1 μέχρι και το 21), μια συνάρτηση που επιστρέφει τυχαία μία περιστροφή για το κομμάτι (UP, RIGHT, DOWN, LEFT) και μια συνάρτηση που επιστρέφει τυχαία αν το κομμάτι θα αναστραφεί οριζόντια (NO, YES).

Σημείωση: Δε χρειάζεται να βάλετε τυχαίο seed, καθώς θα το ορίσουμε αργότερα στη main.

Σημειώστε επίσης ότι κάποιες από τις μεθόδους των παραπάνω κλάσεων έχουν ήδη υλοποιηθεί στο αρχείο playerpieces.cpp. Εσείς θα πρέπει να υλοποιήσετε τις υπόλοιπες στο αρχείο player.cpp.

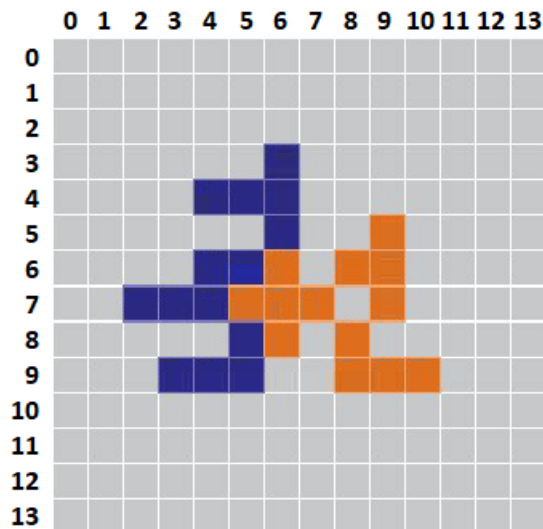
Επιπλέον, θα πρέπει να υλοποιήσετε τις παρακάτω **συναρτήσεις της κλάσης Board στο αρχείο boardrules.cpp**:

- Συνάρτηση pieceCanBePlaced

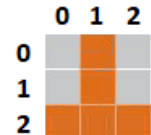
Η συνάρτηση αυτή αναλαμβάνει να ελέγξει αν ένα κομμάτι μπορεί να τοποθετηθεί σε μια συγκεκριμένη τοποθεσία στο ταμπλό (board). Δέχεται ως είσοδο το κομμάτι που πρόκειται να τοποθετηθεί καθώς και την τετμημένη και την τεταγμένη της τοποθέτησης (άνω αριστερά σημείο όπως τοποθετείται το κομμάτι). **Επιστρέφει true εφόσον ισχύουν οι παρακάτω συνθήκες**:

- Αν είναι η πρώτη κίνηση του παίκτη (δεν έχει τοποθετηθεί ακόμα κανένα κομμάτι του παίκτη), τότε το κομμάτι πρέπει να βρίσκεται οπωσδήποτε πάνω στο τετράγωνο (4, 4) για τον πρώτο παίκτη ή στο τετράγωνο (9, 9) για τον δεύτερο παίκτη.
- Αν δεν είναι η πρώτη κίνηση του παίκτη, το κομμάτι πρέπει να ακουμπάει μία τουλάχιστον γωνία κάποιου κομματιού του ίδιου παίκτη.
- Το κομμάτι πρέπει να τοποθετηθεί μέσα στα όρια του ταμπλό.
- Το κομμάτι δεν πρέπει να τοποθετηθεί πάνω σε κάποιο άλλο κομμάτι, δηλαδή δεν πρέπει να τοποθετηθεί σε ήδη κατειλημμένα τετράγωνα του ταμπλό.
- Το κομμάτι δεν πρέπει να ακουμπάει τις πλευρές κάποιου κομματιού του ίδιου παίκτη.

Σε κάθε άλλη περίπτωση η συνάρτηση επιστρέφει false. Ένα παράδειγμα για την τοποθέτηση ενός κομματιού φαίνεται στην παρακάτω εικόνα:



Για να τοποθετήσουμε το κομμάτι στο ταμπλό, ως όρισμα δίνεται η θέση του άνω αριστερά τετραγώνου (εκεί που θα μπει το 0, 0 του κομματιού)



Παραδείγματα τοποθέτησης:

- (2, 6): λάθος τοποθέτηση, το κομμάτι επικαλύπτει ένα άλλο κομμάτι στο ταμπλό
- (3, 10): λάθος τοποθέτηση, το κομμάτι ακουμπάει σε πλευρά κομματιού του ίδιου παίκτη
- (10, 5): λάθος τοποθέτηση, το κομμάτι δεν ακουμπά κάποια γωνία κομματιού του ίδιου παίκτη
- (12, 5): λάθος τοποθέτηση, το κομμάτι βγαίνει εκτός ταμπλό
- (6, 11): σωστή τοποθέτηση

- Συνάρτηση computeScore της κλάσης Player

Η συνάρτηση αυτή μετράει τους πόντους που έχει μαζέψει ο κάθε παίκτης στο τέλος του παιχνιδιού. **Οι πόντοι υπολογίζονται με βάση τους παρακάτω κανόνες:**

- Ο παίκτης παίρνει ένα πόντο για κάθε τετράγωνο που είναι κατειλημμένο από αυτόν (δηλαδή από κάποιο κομμάτι του) στο ταμπλό.
- Αν ο παίκτης έχει τοποθετήσει όλα του τα κομμάτια, τότε κερδίζει ακόμα 15 πόντους.
- Αν ο παίκτης έχει τοποθετήσει όλα του τα κομμάτια και επιπλέον το τελευταίο κομμάτι που τοποθετήθηκε είναι το μικρότερο (δηλαδή το πρώτο κομμάτι που είναι το 1 τετράγωνο), τότε κερδίζει ακόμα 5 πόντους.

Εφόσον γίνουν οι υπολογισμοί, η συνάρτηση επιστρέφει το άθροισμα των πόντων του παίκτη.

Εγκατάσταση

Για να βάλετε το project στο CodeBlocks, αρκεί να αποσυμπιέσετε 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.cbp από τον φάκελο που μόλις αποσυμπιέσατε 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 επεμβείτε 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 player.cpp και boardrules.cpp.

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 Παρασκευής 27 Μαΐου**. Καμία παρέκκλιση δε θα γίνει από την παραπάνω προθεσμία.