**PROXIMITY**

ΜΠΑΛΤΖΗΣ ΕΥΡΙΠΙΔΗΣ 8196

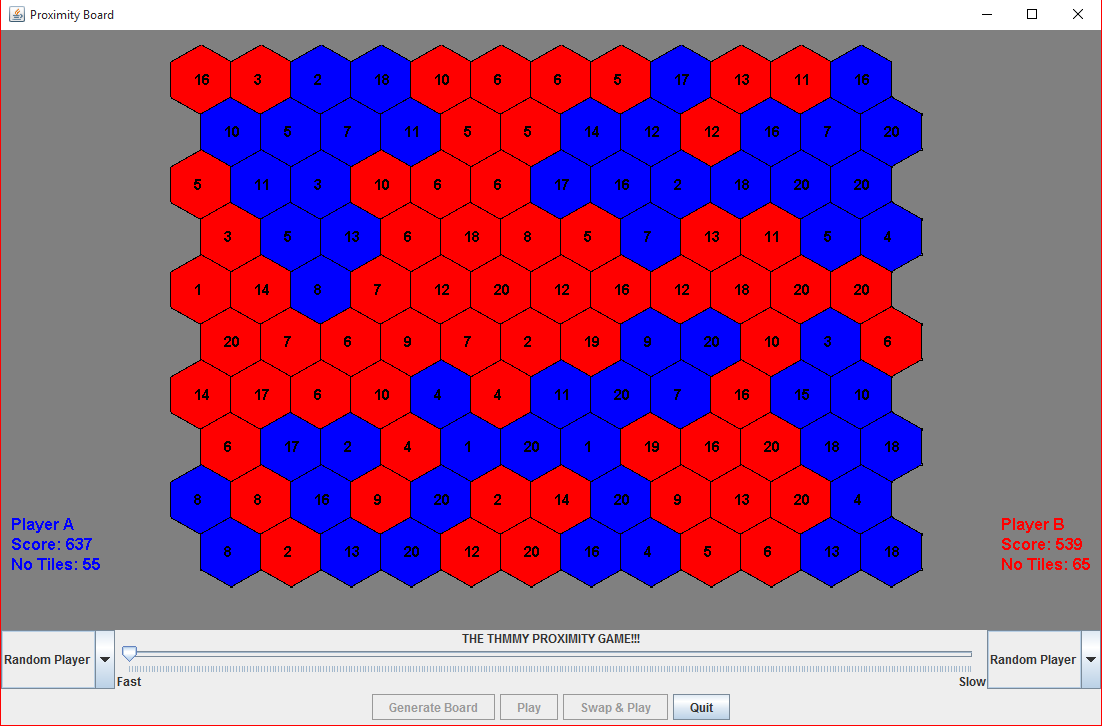
ΡΟΠΟΥΤΗΣ ΔΗΜΗΤΡΗΣ 8233

Στο **πρώτο** μέρος της εργασίας κληθήκαμε να συμπληρώσουμε την κλάση RandomPlayer με όλες τις απαραίτητες συναρτήσεις, ώστε αυτή:

Α) να δημιουργεί έναν καινούριο παίκτη, ο οποίος θα επιλέγει με τυχαίο τρόπο μια θέση (x,y), αφού πρώτα θα έχει ελέγxει αν η θέση αυτή δεν είναι κατειλημμένη (από τον ίδιο ή τον αντίπαλο παίχτη) και είναι εντός του ταμπλό και

Β) να περιέχει τη συνάρτηση getNeighborsCoordinates, η οποία θα επιστρέφει έναν πίνακα με όλα τα εφαπτόμενα στην επιλεγμένη θέση πλακίδια (σε περίπτωση που κάποιο από τα εφαπτόμενα πλακίδια βρίσκεται εκτός ταμπλό, οι συντεταγμένες του ορίζονται ως (-1,-1)).

Μετά την υλοποίση της κλάσης αυτής θα πρέπει να τρέξουμε το παιχνίδι επιλέγοντας και οι 2 παίκτες να είναι της μορφής Random Player και περιμένουμε ως αποτέλεσμα τη διεξαγωγή ενός παιχνιδιού μεταξύ των τυχαίων παικτών που δημιουργεί η κλάση που υλοποιήσαμε, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 1: Το περιβάλλον του παιχνιδιού Proximity μετά το πέρας ενός επιτυχημένου παιχνιδιού μεταξύ δύο τυχαίων παικτών.

**Class RandomPlayer**

Παρακάτω παρουσιάζεται η υλοποίηση της ζητούμενης κλάσης και περιγράφονται οι μεταβλητές, μέθοδοι και συναρτήσεις που την αποτελούν.

Αρχικά, οι μεταβλήτές ενός αντικειμένου της κλάσης αυτές είναι:

* **int id:** η μεταβλητή αυτή παίρνει τις τιμές 1 ή 2 και καθορίζει αν ο παίκτης που δημιουργήθηκε είναι ο μπλε ή ο κόκκινος.
* **String name:** περιέχει το όνομα του εκάστοτε παίκτη
* **int score:** περιέχει το συνολικό σκορ του παίκτη, το οπολίο προκύπτει από το άθροισμα όλων των τιμών εντός των πλακιδίων που ανήκουν στον παίκτη
* **int numOfTiles:** περιέχει τον αριθμό τον πλακιδίων που ανήκουν στον παίκτη

Στη συνέχεια αναλύονται οι συναρτήσεις και μέθοδοι της κλάσης.

* **Constructor RandomPlayer(Integer pid):** ο πρώτος εκ των δύο constructor της κλάσης, ο οποίος δέχεται ως όρισμα του έναν integer και περνάει την τιμή του στη μεταβλητή id του αντικειμένου που δημιουργείται.
* **Constructor RandomPlayer(int pid, String onoma, id score, int plakidia):** ο δεύτερος constructor της κλάσης, ο οποίος αντιστοιχεί τις τιμές των ορισμάτων με τις αντίστοιχες μεταβλητές της κλάσης

id = pid;

**this**.score = score;

numOfTiles = plakidia;

name = onoma;

(στην περίπτωση της μεταβλητής score χρησιμοποιείται το keyword this καθώς τα ονόματα του ορίσματος και της μεταβλητής της κλάσης ταυτίζονται).

* **Όλες οι συναρτήσεις της μορφής typos\_metavlhths getOnomaMetavlhths():** οι συναρτήσεις αυτές επιστρέφουν τις αντίστοιχες τιμές των μεταβλητών που περιέχονται στο όνομα της συνάρτησης (π.χ. η συνάρτηση int getId() επιστρέφει την τιμή της μεταβλητής id του αντικειμένου της κλάσης που την καλεί).
* **Όλες οι συναρτήσεις της μορφής void setOnomaMetavlhths(typos\_metavhths orisma):** οι συναρτήσεις αυτές δέχονται ως όρισμά τους μια μεταβλητή τύπου ανάλογου του τύπου της μεταβλητής που περιέχεται στο όνομα της συνάρτησης και περνάνε την τιμή του ορίσματος στην αντίστοιχη μεταβλητή του αντικειμένου της κλάσης που τις καλεί(π.χ. η συνάρτηση setId(int pid) περνάει την τιμή του ορίσματος pid στη μεταβλητή id του αντικειμένου της κλάσης που την καλεί).
* **Συνάρτηση int getNextMove(Board board):** η συνάρτηση αυτή δέχεται σαν όρισμα ένα αντικείμενο της κλάσης Board, δηλ. δέχεται ως όρισμα ολόκληρο το ταμπλό, και καλείται να επιστρέψει ένα μονοδιάστατο πίνακα 2 θέσεων, που θα περιέχει τις συντεταγμένες του πλακιδίου που θα επιλέξει ο παίκτης να χρησιμοποιήσει για την επόμενη κίνησή του. Η επιλογή του κάθε πλακιδίου γίνεται τυχαία με τη χρήση ενός αντικειμένου της κλάσης Random και της αντίστοιχης μεθόδου *nextInt(bound).* Η μέθοδος αυτή παράγει με τυχαίο τρόπο έναν αριθμό στο διάστημα [0, bound), όπου bound είναι μια μεταβλητή της μορφής int. Στη περίπτωση της τετμημένης (συντεταγμένη x), ως bound χρησιμοποιείται ο αριθμός των στηλών του ταμπλό που είναι περασμένος στη στατική μεταβλητή *NUMBER\_OF\_COLUMNS*, ενώ αντίστοιχα για την τεταγμένη (συντεταγμένη y), ως bound χρησιμοποιείται ο αριθμός των γραμμών του ταμπλό που είναι περασμένος στη στατική μεταβλητή *NUMBER\_OF\_ROWS.* Φυσικά, για να είναι αποδεκτές οι συντεταγμένες που παράγονται από τη Random θα πρέπει πρώτα να ελεγχθεί ότι το πλακίδιο με τις συντεταγμένες αυτές δεν ανήκει σε κάποιον παίκτη. Αυτό επιτυγχάνεται με τη χρήση μιας δομής επανάληψης do-while με κριτήριο ελέγχου τη συνθήκη *board.getTile(coordinates[0],coordinates[1])).getPlayerId() != 0*. Τελικά, η συνάρτηση επιστρέφει τον πίνακα με τις συντεταγμένες που θα χρησιμοποιήσει ο παίκτης στην επόμενη κίνησή του.
* **Συνάρτηση static int[][] getNeighborsCoordinates(Board board,int x, int y):** η συνάρτηση αυτή δέχεται ως ορίσματα το ταμπλό ( όπως και στην παραπάνω συνάρτηση), καθώς και τις συντεταγμένες (x,y) του πλακιδίου που χρησιμοποίησε ο παίκτης στην κίνησή του και επιστρέφει ένα διδιάστατο πίνακα 6x2 που περιέχει όλους τους γείτονες του, δηλ. όλα τα εφαπτόμενα σε αυτό πλακίδιο. Προφανώς, εφόσον το ταμπλό είναι ένα εξαγωνικό πλέγμα, τα εφαπτόμενα στο (x,y) πλακίδια μπορούν να είναι μέχρι έξι. Επίσης, ζητείται ο επιστρεφόμενος πίνακας να έχει την παρακάτω μορφή:

|  |  |
| --- | --- |
| **x.EAST** | **y.EAST** |
| **x.SOUTHEAST** | **y.SOUTHEAST** |
| **x.SOUTHWEST** | **y.SOUTHWEST** |
| **x.WEST** | **y.WEST** |
| **x.NORTHWEST** | **y.NORTHWEST** |
| **Xx.NORTHEAST** | **y.NORTHEAST** |

Ανάλογα με το αν το πλακίδιο βρίσκεται σε άρτια ή περιττή σειρά του ταμπλό, η σειρά με την οποία εμφανίζονται οι γείτονες διαφέρει. Φυσικά, οι τιμές που πάιρνουν οι σειρές του ταμπλό βρίσκονται στο διάστημα [0, *NUMBER\_OF\_ROWS – 1*] και όχι στο [1, *NUMBER\_OF\_ROWS*], οπότε οι όροι άρτιος και περιττός διαφέρουν.

Έτσι, αν το πλακίδιο βρίσκεται σε περιττή σειρά (δηλ. σε άρτια στο [0, *NUMBER\_OF\_ROWS – 1*]), τότε ο επιστρεφόμενος πίνακας παίρνει τιμές:

|  |  |
| --- | --- |
| x+1 | y |
| x | y+1 |
| x-1 | y+1 |
| x-1 | y |
| x-1 | y-1 |
| x | y-1 |

Αντίστοιχα, αν το πλακίδιο βρίσκεται σε άρτια σειρά (δηλ. σε περιττή στο [0, *NUMBER\_OF\_ROWS – 1*]), τότε ο επιστρεφόμενος πίνακας παίρνει τιμές:

|  |  |
| --- | --- |
| x+1 | y |
| x+1 | y+1 |
| X | y+1 |
| x-1 | y |
| x | y-1 |
| x+1 | y-1 |