

**PROXIMITY**

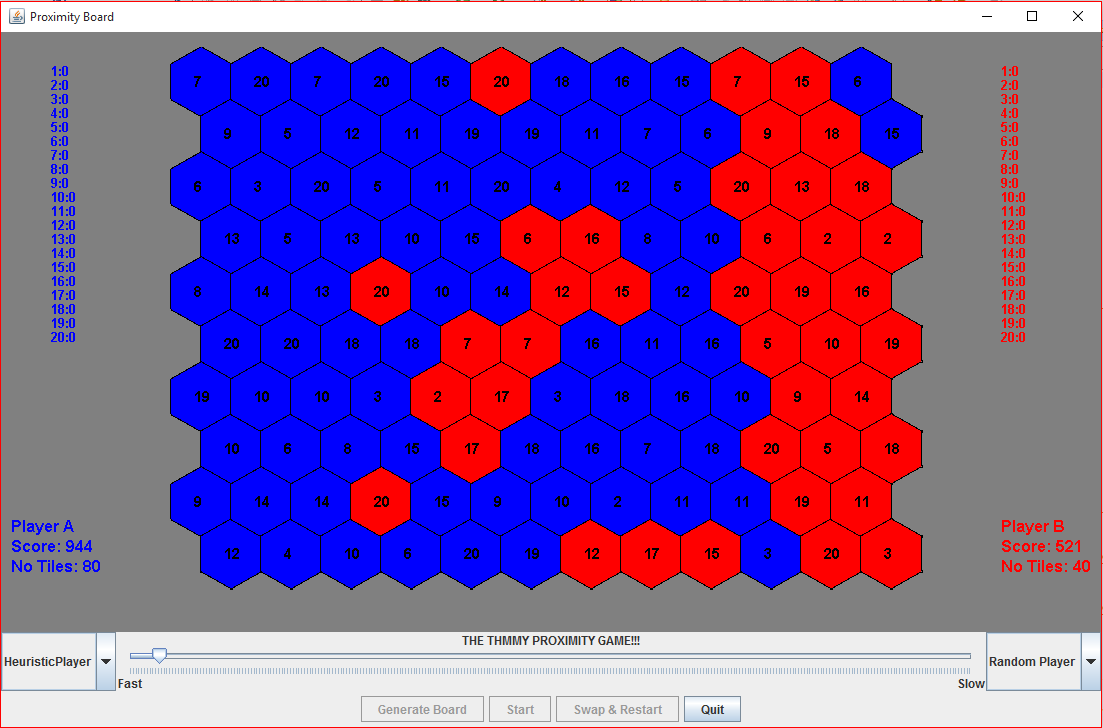
ΜΠΑΛΤΖΗΣ ΕΥΡΙΠΙΔΗΣ 8196

ΡΟΠΟΥΤΗΣ ΔΗΜΗΤΡΗΣ 8233

Στο **δεύτερο** μέρος της εργασίας κληθήκαμε να συμπληρώσουμε την κλάση HeuristicPlayer και συγκεκριμένα τις συναρτήσεις:

* **getEvaluation**, η οποία θα παρέχει ένα αλγόριθμο αξιολόγησης της προεπιλεγμένης κίνησης ακολουθώντας μια σειρά κριτηρίων ορισμένων από το χρήστη, καθώς και τη
* **getNextMove**, η οποία θα ελέγχει όλες τις διαθέσιμες κινήσεις, θα τις περνάει από τη συνάρτηση αξιολόγησης και στη συνέχεια θα επιλέγει την καλύτερη.

Μετά την υλοποίση της κλάσης αυτής θα πρέπει να τρέξουμε το παιχνίδι επιλέγοντας τον ένα παίκτη να είναι της μορφής Heuristic Player και τον άλλο να είναι της μορφής Random Player και περιμένουμε ως αποτέλεσμα τη διεξαγωγή ενός παιχνιδιού μεταξύ του παίκτη που δημιουργεί η κλάση που υλοποιήσαμε και του τυχαίου παίκτη, καθώς και τη νίκη του παίκτη που δημιουργήσαμε, όπως φαίνεται στην παρακάτω εικόνα.



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

**Class HeuristicPlayer**

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

Όλες οι συνάρτήσεις **set** και **get** των μεταβλητών αυτών παραμένουν ίδιες με την πρώτη εργασία, όπως και ο αντίστοιχος **constructor HeuristicPlayer(Integer pid)**. Παρακάτω παρουσιάζονται οι προς υλοποίηση συναρτήσεις:

* **int getEvaluation(Board board, int randomNumber, Tile tile)**: η συνάρτηση αυτή δέχεται σαν όρισμα ένα αντικείμενο της κλάσης Board, δηλ. δέχεται ως όρισμα ολόκληρο το ταμπλό, ένα αντικέιμενο της κλάσης Tile, δηλ. ένα συγκεκριμένο πλακίδιο πάνω στο ταμπλό, και τον αριθμό randomNumber, που αντιπροσωπεύει το σκορ του πλακιδίου που πρόκειται να τοποθετηθεί στο ταμπλό από τον παίκτη. Σύμφωνα με τους κανόνες του παιχνιδίου, κάθε εφαπτομενικό πλακίδιο που ανήκει στον παίκτη αυξάνει στο σκορ του κατά 1, ενώ κάθε αντίπαλο εφαπτομενικό με μικρότερο σκορ αλλάζει κατοχή και ανήκει πλέον στον παίκτη, χωρίς να αλλάξει το σκορ του. Έτσι, μια κίνηση προσθέτει στον παίκτη συνολικό σκορ ίσο με το παρακάτω άθροισμα: *ΣΥΝΟΛΙΚΟ ΣΚΟΡ = ΣΚΟΡ\_ΠΛΑΚΙΔΙΟΥ + ΣΚΟΡ\_ΚΕΡΔΙΣΜΕΝΩΝ\_ΑΝΤΙΠΑΛΟΥ + ΣΚΟΡ\_ΑΥΞΗΣΗΣ\_ΠΑΙΚΤΗ*. Η μέγιστη τιμή αυτού του σκορ είναι 131, δεδομένου ότι όλα τα γειτονικά πλακίδια είναι εντός ταμπλό, ανήκουν στον αντίπαλο και έχουν ανά 3 σκορ 18 και 19 αντίστοιχα, ενώ το προς τοποθέτηση πλακίδιο έχει σκορ 20, δηλ. 20 + (3x18 + 3x19) + 0 = 131. Αντίστοιχα προκύπτει ότι το μέγιστο σκορ που μπορεί ένας παίκτης να αφαιρέσει από τον αντίπαλό του με μια κίνηση ισούται με 3x18 + 3x19 = 111. Έτσι, η συνάρτηση υπολογίζει το συνολικό σκορ που προστίθεται στον παίκτη και το συνολικό σκορ που αφαιρείται από τον αντίπαλο και τα διαιρεί με τις μέγιστες δυνατές τιμές που μπορούν να πάρουν αυτά τα σκορ σαν αναγωγή στη μονάδα. Στη συνέχεια, το αποτέλεσμα που προκύπτει πολλαπλασιάζεται με τον αντίστοιχο συντελεστή βαρύτητας που ορίσαμε (40 για το προστιθέμενο σκορ και 30 για το αφαιρούμενο από τον αντίπαλο). Ωστόσο επειδή συνήθως τα σκορ αυτά δεν παίρνουν μεγάλες τιμές, αποφασίσαμε στην περίπτωση που το προστιθέμενο σκορ ξεπερνά τα 60 να αυξάνουμε το συντελεστή βαρύτητας κατά 15, ενώ στην περίπτωση που το αφαιρούμενο ξεπερνά τα 40 να αυξάνουμε τον αντίστοιχο συντελεστή κατά 10. Ο τελευταίος συντελεστής που ελήφθη υπόψην είναι ο αριθμός των εγαπτομενικών κελιών που είναι “πιασμένα” προς τον αριθμό των εφαπτομενικών κελιών εντός ταμπλό, δηλ. όσο λιγότερα διαθέσιμα γειτονικά κελιά για τον αντίπαλο, τόσο λιγότερες πιθανότητες να χαθούν γειτονικά κελιά στις επόμενες κινήσεις του αντιπάλου. Ο αντίστοιχος συντελεστής βαρύτητας στην προκειμένη περίπτωση είναι 15. Τελικά, επιστρέφει το αντίστοιχο άθροισμα των τριών αξιολογητικών κριτηρίων σε μια μεταβλητή double.
* **int getNextMove(Board board, int randomNumber)**: η συνάρτηση αυτή δέχεται σαν όρισμα ένα αντικείμενο της κλάσης Board, δηλ. δέχεται ως όρισμα ολόκληρο το ταμπλό, και τον αριθμό randomNumber, που αντιπροσωπεύει το σκορ του πλακιδίου που πρόκειται να τοποθετηθεί στο ταμπλό από τον παίκτη. Στη συνέχεια, σαρώνει ολόκληρο το ταμπλό και για όλες τις διαθέσιμες θέσεις δημιουργεί ένα νέο αντικείμενο της κλάσης Tile, το οποίο το περνά μέσα από την παραπάνω συνάρτηση αξιολόγησης και κρατάει το αποτέλεσμά της. Οι συντεταγμένες του πλακιδίου Tile, σε συνδυασμό με το αντίστοιχο αποτέλεσμα της αξιολόγησης περνάν στη συνέχεια σε ένα Hashmap της μορφής <Tile, Double> ως συνδυασμός κλειδιού-τιμής. Έπειτα, το Hashmap αυτό σαρώνεται και επιλέγεται εκείνο το κλειδί (Tile) για το οποιό εμφανίζεται η μέγιστη τιμή αξιολόγησης Double, το οποίο αποτελεί και την επόμενη κίνηση του παίκτη. Τελικά, η συνάρτηση επιστρέφει έναν πίνακα 3 θέσεων που περιέχει την τετμημένη και την τεταγμένη του πλακιδίου που επιλέχτηκε, καθώς και το σκορ του πλακιδίου που πρόκειται να τοποθετηθεί από τον παίκτη.