Εργασία CandyCrush-Part B

Δομές Δεδομένων 5ο Εξάμηνο

Δημήτρης-Νεκτάριος Ευαγγελόπουλος ΑΕΜ:8561

Μπεκιάρης Θεοφάνης ΑΕΜ:8200

Περιγραφή προβλήματος

Σε αυτό το μέρος της εργασίας ασχολούμαστε με μια συνάρτηση αξιολόγησης των διαθέσιμων κινήσεων που έχει κάθε παίκτης σε ένα γύρο παιχνιδιού, καθώς και με την επιλογή της καλύτερης κίνησης.Ο στόχος είναι να εντοπίσουμε τους παράγοντες που καθορίζουν την πίο ευνοική εξέλιξη του παιχνιδιού για εμάς και να συνυπολογίσουμε την αξία του κάθε παράγοντα για την δημιουργία της τελικής συνάρτησης αξιολόγησης.

Περιγραφή αλγορίθμου

Η λογική με την οποία αναπτύξαμε τον κώδικα ήταν να διαλέξουμε την κίνηση η οποία θα μας επιφέρει την μεγαλύτερη απομάκρυνση ζαχαρωτών, υπολογίζοντας τα ζαχαρωτά που θα διαγραφούν άμεσα απο την κίνηση καθώς και την απομάκρυνση ζαχαρωτών που προκύπτει απο την νέα κατάσταση του ταμπλό μετά την κίνηση.Επίσης για να μην είναι μεροληπτική η αξιολόγηση, λαμβάνουμε υπόψη μας και την κίνηση την οποία θα κάνει ο αντίπαλος στον επόμενο γύρο,δηλαδή επιλέγουμε την κίνηση μας έτσι ώστε να μήν αφήνουμε στον αντίπαλο συνδιασμούς με μεγάλο σκόρ.Επιπλέον για κινήσεις με ίση αξιολόγηση συνυπολογίζουμε και το ύψος του πλακιδίου που θα μετακινήσουμε ώστε να βρίσκεται όσο πιο χαμηλά γίνεται στο ταμλό όπως περιγράφεται παρακάτω.

Περιγραφή Κώδικα

Έχουμε μια κλαση την HeuirsticPlayer μέσα στην οποία υπάρχουν δύο συναρτήσεις που καλούμαστε να υλοποιήσουμε,τις moveEvaluation() και της findBestMoveIndex().Επιπλέον ορίσαμε μέσα στην κλάση και μία νέα συνάρτηση με όνομα calculateEarnedPoints.

calculateEarnedPoints():Η λογική της στην ουσία είναι:
"κάνω μια κίνηση--> πόσα πλακίδια θα μου δώσει συνολικά αυτή η κίνηση"
Η συνάρτηση αυτή δέχεται σαν όρισμα την υποψήφια κίνηση και ένα αντικείμενο τύπου Board και επιστρέφει τον αριθμό των συνολικών πλακιδίων που θα διαγραφούν απο την κίνηση,δηλαδή και πλακίδια που θα διαγραφούν απο τους συνδιασμούς που θα προκύψουν μετά απο την κίνηση.

moveEvaluation():Η συνάρτηση αυτή δέχεται σαν όρισμα την επιθυμητή κίνηση(int[] move),ένα αντικείμενο τύπου Board και μία μεταβλητή int depthTwo η οποία χρησιμοποιείται ως βοηθητική για να αξιολογούμε και την αμέσως επόμενη κίνηση του αντιπάλου,στην ουσία σταματάει την αναδρομή μετά απο την αξιολόγηση της αμέσως επόμενης κίνησης του αντιπάλου ώστε η αξιολόγηση να είναι αμερόληπτη.Η συνάρτηση υπολογίζει τον αριθμό των ζαχαρωτών που θα πάρουμε απο την κίνηση μας με την βοήθεια της συνάρτησης **calculateEarnedPoints**,έπειτα υπολογίζει τα ζαχαρωτά που θα κερδίσει ο αντίπαλος απο την αμέσως επόμενη καλύτερη κίνηση του,δεδομένου οτι εμείς επιλέξαμε την συγκεκριμένη κίνηση.Τέλος επιστρέφει την διαφορά των δύο παραπάνω αποτελεσμάτων,έτσι αποφεύγεται μία κίνηση που φαινομενικά θα μας έδινε πολλούς πόντους αλλά θα μπορούσε να δώσει πολύ περισσότερους στον αντίπαλο.

findBestMoveIndex():Η συνάρτηση υπολογίζει μέσα απο μία λίστα επιτρεπτών κινήσεων την καλύτερη κίνηση με βάση την συνάρτηση αξιολόγησης **moveEvaluation**.Επιπλέον φροντίζει να επιλέξει ανάμεσα απο κινήσεις με ίδια αξιολόγηση εκείνη η οποία θα βρίσκεται πιο χαμηλά στο ταμπλό.Η λογική για την επιλογή χαμηλού ύψους ειναι οτι αφού υπάρχουν κινήσεις στις οποίες θα παίξω εγω και αμέσως μετά ο αντίπαλος και θά επιφέρουν το ίδιο αποτέλεσμα θα διαλέξω εκείνη η οποία θα έχει μεταβάλει όσο το δυνατόν περισσότερο γίνεται το ταμπλό ώστε να δίνει σε εμένα περισσότερες ευκαιρίες για συνδιασμούς όταν ξανα έρθει η σειρά μου.

Σχόλιο:Η συνάρτηση αξιολόγησης έχει υλοποιηθεί με το σκεπτικό ότι ο αντίπαλος παίκτης θα παίξει την καλύτερη του κίνηση στο επόμενο γύρο,κάτι το οποίο δεν είναι απόλυτο ούτε πάντα σωστό καθώς τον αντίπαλο παίκτη ίσος να μην τον συμφέρει να παίξει την καλύτερη του κίνηση.Μια καλύτερη υλοποιήση θα απαιτούσε τον έλεγχο όλων των δυνατών κινήσεων του αντιπάλου για κάθε μία απο τις κινήσεις μας,με τίμημα περισσότερο χρόνο για τον υπολογισμό όλων αυτών των περιπτώσεων.