

AM: 1115201800036

Ον/νυμο: Άρης Γρηγορόπουλος

Πρόβλημα 5)

Q1) Στο evaluation function χρησιμοποιώ 5 κριτήρια για τον υπολογισμό του σκορ.

- 1) Την απόσταση από το κοντινότερο food
- 2) Την απόσταση από το κοντινότερο ghost
- 3) Το πόσα foods απομένουν στο παιχνίδι
- 4) Το ο pacman μετακινείται ή όχι (έτσι ώστε να μην μένει ακίνητος για πολύ ώρα
- 5) Το αν ο pacman μετακινείται σε θέση όπου υπάρχει food

Η απόσταση του pacman από το food και τα ghosts υπολογίζεται με Manhattan distance. Όσο πιο κοντά βρίσκεται ο pacman σε food, τόσους περισσότερους πόντους δίνει η κίνηση αυτή. Το αντίθετο προφανώς ισχύει για τα φαντάσματα. Αν η κίνηση οδηγεί ακριβώς πάνω σε ένα food δίνεται ένα μικρό bonus σε αυτή την κίνηση (200) ενώ αν η κίνηση οδηγεί ακριβώς πάνω σε ένα ghost δίνεται ένα πολύ μεγάλο penalty στην κίνηση (-10000) αφού αυτό θα οδηγήσει σε σίγουρη ήττα. Λαμβάνεται υπόψιν το πόσα foods απομένουν έτσι ώστε ο pacman να λαμβάνει μεγαλύτερα ρίσκα προς το τέλος του παιχνιδιού, το οποίο με tests βρήκα ότι οδηγεί σε πιο γρήγορα παιχνίδια συνολικά. Επίσης έβαλα το μικρό penalty (-500) για την ακινησία έτσι ώστε ο pacman να αποφεύγει να μένει ακίνητος, αφού αυτό δεν είναι πολύ ρεαλιστικό στο κανονικό παιχνίδι.

Τα νούμερα που χρησιμοποιώ, τα έβαλα μετά από δοκιμές, ώστε να είναι κατάλληλα για τα tests.

Q2) Για το MinimaxAgent, έφτιαξα την minimax συνάρτηση, η οποία δουλεύει αναδρομικά και επιστρέφει το score που χρησιμοποιείται από τους υπολογισμούς με το evaluationFunction(). Η συνάρτηση αφού ελέγξει το αν έχουν «παίξει» όλοι οι agents και ότι δεν βρίσκεται σε τερματική κατάσταση, ελέγχει το σκορ κάθε νόμιμης κίνησης. Αν ο τρέχων agent είναι ο 0 (δηλαδή ο pacman, δηλαδή ο max παίκτης) τότε κρατάει την κίνηση με το μεγαλύτερο σκορ. Αλλιώς κρατάει την κίνηση με το χαμηλότερο, αφού είναι η περίπτωση min.

Q3) Όπως υποδείκνυε η εκφώνηση της άσκησης, πέρα από την «κύρια» συνάρτηση minimax, έχω υλοποιήσει και τις συναρτήσεις max_value και min_value. Θα μπορούσα και αυτές να τις υλοποιήσω παρόμοια με το προηγούμενο ερώτημα (δηλαδή min και max να μην είναι διαφορετικές συναρτήσεις και απλά να γίνεται ένας έλεγχος για τον τρέχων agent μέσα στη συνάρτηση minimax), αλλά αποφάσισα να ακολουθήσω τον ψευδοκώδικα που έδινε η άσκηση.

Πέρα από αυτό, ο κώδικας είναι σχεδόν ίδιος με αυτόν του Q2), με τη διαφορά ότι χρησιμοποιούνται τα a και b για το «κλάδεμα». Μέσα στο loop του for, που ελέγχει την κάθε πιθανή κίνηση, αν στο max βρεθεί σκορ μεγαλύτερο του b τότε επιστρέφεται αμέσως αυτή η κίνηση (το αντίθετο ισχύει προφανώς για το min).

Q4) Πάλι ο κώδικας της Expectimax είναι σχεδόν ίδιος με αυτόν της Q2) με την εξής διαφορά:

Αν ο agent δεν είναι 0 (δηλαδή είναι ghost), τότε αντί να διαλέγει την κίνηση με το minimum score, παίρνει ως score το evaluation της κάθε κίνησης διαιρεμένο με το πλήθος των πιθανών κινήσεων, αφού τα ghosts δεν παίζουν βέλτιστα.

Q5) Είναι σχεδόν ίδιος με τον κώδικα της Q1) και δίνει 5/6 στον autograder. Η μόνη διαφορά είναι ότι χρησιμοποιεί το current game state αντί για το successor game state. Δυστυχώς λόγω έλλειψης χρόνου δεν είχα την ευκαιρία να τον βελτιώσω περαιτέρω.

Για κάθε ερώτημα έχω γράψει επίσης σχόλια στον κώδικα που βοηθούν με την κατανόηση των υλοποιήσεών μου.