Project 2: Multi-Agent Search

→ Q1 Reflex Agent

Στο πρώτο ερώτημα φτιάχνω την evaluationFunction για την επιλογή του καλύτερου απογόνου της τρέχουσας κατάστασης. Αρχικά εξετάζω εάν είμαι σε θέση νίκης ή ήττας επιστρέφοντας κατάλληλα νούμερα(πολύ υψηλά για νίκη-πολύ χαμηλά για ήττα).

Έπειτα βρίσκω την ελάχιστη απόσταση του pacman από το φάντασμα και από τη τροφή του, με χρήση της manhattanDistance. Εάν ο pacman βρίσκεται πολύ κοντά επιστρέφω μία μεγάλη αρνητική τιμή προκειμένου να το αποφύγει.

Κλείνοντας η συνάρτηση επιστρέφει το άθροισμα του getScore με το λόγο της απόστασης από το κοντινότερο φάντασμα προς την απόσταση από την κοντινότερη τροφή μιας και ιδανικά θέλουμε ο παρονομαστής να είναι πολύ μικρός(δηλαδή να βρίσκεται κοντά στη τροφή) και ο αριθμητής μεγάλος(δηλαδή μακριά από το φάντασμα)

→ Q2 Minimax

Για την υλοποίηση του Minimax βασίστηκα αρκετά στις διαφάνειες του μαθήματος καθώς και στους αλγορίθμους που παρουσιάστηκαν σε αυτό.

Αρχικά υλοποιήθηκαν οι συναρτήσεις MaxValue και MinValue, με την getAction να καλεί τη MaxValue μιας και ο pacman παίζει πρώτος.

Στο MaxValue εξετάζω εάν είμαστε σε τερματική κατάσταση όπου και καλεί την evaluationFunction. Μετά για κάθε απόγονο του **MAX** με τη βοήθεια της MinValue υπολογίζω το σκορ του καθενός και επιστρέφω τη μεγαλύτερη τιμή των απογόνων καθώς και την ενέργεια για να φτάσει το pacman εκεί.

Στο MinValue ελέγχω εάν έχουν παίξει όλα τα φαντάσματα, εφόσον αυτό δεν ισχύει καλώ την MinValue για το επόμενο μέχρι να τελειώσουν τα φαντάσματα που επρόκειτο να παίξουν. Αντίστοιχα επιστρέφει το μικρότερο σκορ(**successor_score<v** αντί **successor_score>v** που είναι στη MaxValue) και την ενέργεια. Αντίθετα στο τελευταίο **MIN** παίκτη καλεί την MaxValue για την ανάθεση του σκορ μεταβαίνοντας στο επόμενο βάθος(depth+1).

→ Q3 Alpha-Beta Pruning

Στο Q3 επεκτείνω την υλοποίηση του minimax ώστε να μην εξετάζονται όλοι οι κόμβοι.

Ειδικότερα, οι τιμές alpha και beta μας βοηθάνε να αποκόψουμε καταστάσεις. Στη Max Value εάν είναι μεγαλύτερη από την καλύτερη επιλογή του **MIN** τότε επιστρέφει το μέγιστο που έχει βρεθεί μέχρι τότε ενώ το alpha ανανεώνεται κάθε φορά που βρίσκεται καλύτερη από την προηγούμενη επιλογή.

Όσο αφορά το MinValue γίνεται η ανάποδη διαδικασία δηλαδή για σκορ που είναι μικρότερο της καλύτερης επιλογής του **MAX** σταματάει και επιστρέφει το ελάχιστο που έχει βρεί ενώ το beta ανανεώνεται σε κάθε νέο μικρότερο σκορ που υπολογίζεται.

→ Q4 Expectimax

Με τη συνάρτηση ExpectimaxValue βρίσκω την προβλεπόμενη τιμή για όλες τις 'νόμιμες' κινήσεις του pacman. Στα φαντάσματα καλείται αναδρομικά η expectimaxvalue πέρα του τελευταίου που καλεί τη MaxValue, η οποία έχει την ίδια συμπεριφορά με τα προηγούμενα ερωτήματα με μοναδική εξαίρεση τη κλήση της ExpectimaxValue -αφού δεν υπάρχειMinValue-.

Σε κάθε πιθανή κίνηση του φαντάσματος προστίθεται η πιθανότητα επί το σκορ στη συνολική τιμή .

→ Q5 Evaluation Function

Μια βελτιωμένη εκδοχή του Q1 που αξιολογεί καταστάσεις αντί για ενέργειες. Αρχικά εξετάζω εάν βρίσκομαι σε state νίκης ή ήττας επιστρέφοντας πάλι ανάλογες τιμές.

Επιπλέον αποθηκεύω την ελάχιστη απόσταση του pacman από τη τροφή, την κάψουλα και από τα φαντάσματα στις closest_food , closest_capsule, closest_ghost αντίστοιχα με την manhattanDistance.

Η συνάρτηση αξιολόγησης θα επιστρέφει το τρέχων score ως άθροισμα του λόγου closest_ghost προς closest_food και της αντίστοφς τιμής της απόστασης από τη κοντινότερη κάψουλα.