

Ονοματεπώνυμο: Ορφανίδης Ελευθέριος
Α.Μ.: 1115201400133

Pacman project 2: Έχω υλοποιήσει όλα τα ερωτήματα του pacman project με τον τρόπο που τα ήθελε ο autograder, ο οποίος μου δίνει βαθμολόγηση 25/25.

Question 1: Για το ερώτημα 1, στην evaluationFunction δημιουργώ το successorGameState με την χρήση της συνάρτησης generatePacmanSuccessor. Απο το successorGameState παίρνω τις μεταβλητές newPos, newFood και newGhostStates. Για κάθε φάντασμα που υπάρχει στο παιχνίδι, υπολογίζω την απόσταση τους απο τον pacman (manhattan distance) και προσθετω στο score την ελαχιστη απόσταση. Με αυτο τον τροπο αν σε ενα state το κοντινότερο φάντασμα είναι αρκετα μακρια απο τον pacman θα προστεθει στο score μεγαλη τιμη (θα είναι πιο πιθανο να προτιμηθει), ενώ αν βρισκεται αρκετα κοντα σε καποιο φάντασμα τοτε θα προστεθει στο score μια πολυ μικρη τιμη (κατι που θα δυσκολεψει την επιλογη αυτου του action απο τα υπολοιπα που θα μπορει να κανει ο pacman). Επίσης υπολογίζω την απόσταση του pacman απο την κοντινότερη τροφή, την οποια την αφαιρω απο το score , ετσι αν βρισκεται κοντα σε μια τροφη θα αφαιρεθει πολυ μικρη τιμη , ενώ αν βρισκεται μακρια θα αφαιρεθει μεγαλη τιμη (κανοντας δυσκολη την επιλογη αυτου του κομβου λογω μικρου score).

Question 2: Στον minimax αλγοριθμο θεωρώ βαθος 1 την κινηση του pacman και τις κινήσεις όλων των φαντασμάτων. Οπότε αν στο grid εχουμε τον pacman και 2 φαντασματα το depth θα το μειωσω κατα μια μοναδα όταν εκτελεί τον minimax ο agent με agentnum == totalagents - 1 (τελευταίο φάντασμα, μετα κανει κινηση ο pacman). Άλλος τροπος που μπορουσα να υλοποιησω αυτο το προβλημα ηταν με την χρηση των layers αντι του depth (δηλ. layers = self.depth * totalagents) οπου σε καθε αναδρομικη κληση θα εθετα minimax(.. , .. , layer-1). Γενικα max παιχτης θεωρείται μονο ο pacman (agentnum == 0) και min παιχτες θεωρουνται τα φαντασματα.

Question 3: Για το 3ο ερωτημα χρησιμοποiehσα την minimax συναρτηση του προηγουμενου ερωτηματος με μονες διαφοροποιησεις τα alpha, beta που τα χρησιμοποiehσα οπως ακριβως χρησιμοποiehσονται στο παραδειγμα του αλγοριθμου στην εκφωνηση του ερωτηματος 3.

Question 4: Στο ερωτημα του expectimax λειτουργω με τον ιδιο τροπο με τα προηγουμενα 2 ερωτηματα (σε σχεση με το depth και το ποιος είναι max παιχτης), αλλα στην περιπτωση των φαντασμάτων δεν έχω max κομβους , αλλα chance. Πρακτικα αντι να επιστρεψω το ελαχιστο απο τις τιμές των παιδιων του κομβου (που αποτελει την βελτιστη κινηση για ενα ghost , επιστρεφω τον μεσο ορο των τιμων των παιδιων του (αφου ολες οι κινήσεις που κανει ενα ghost είναι ισοπιθανες). Οποτε αθροizω τις τιμες της λιστας values και διαιρω το αθροισμα με τον αριθμο των κομβων.

Question 5: Η υλοποιηση μου σε αυτο το προβλημα θετει βαρυτητες σε καποιες αποστασεις , αναλογα με το ποσο καλη/επικινδυνη είναι η τιμη καποιας αποστασης. Συγκεκριμενα εχω δωσει μεγαλυτερη προτεραιοτητα στην αποσταση απο ενα ghost που μπορει να γινει consume απο τον pacman καθως αυτο θα ωθησει τον pacman στο να κυνηγησει το ghost δινοντας του ετσι προβαδισμα. Στην συνεχεια, στην μεση των προτεραιοτητων είναι η αποσταση του pacman απο καποιο φάντασμα (αν είναι μικρη η αποσταση απο το φάντασμα αφαιρω μεγαλο ποσο απο το score) ενώ μικροτερη προτεραιοτητα εχει η αποσταση απο το κοντινότερο φαγητο. Έχω θεσει ουσιαστικα προτεραιοτητα στην επιβιωση και μετα την καταναλωση τροφης που θα οδηγησει πιο γρηγορα στον τερματισμο του παιχνιδιου.