

Τεχνητή Νοημοσύνη

Εργασία 2 - Pacman project

Στέφανος Βάβουλας - 1115201800014

(Question 1)

Για την Evaluation Function του Reflex Agent:

Για κάθε successor state που προκύπτει από την κίνηση του pacman, βρίσκω το αρχικό σκορ της και σε αυτό προσθέτω:

Αν υπάρχει φαγητό στην καινούρια θέση: 10 πόντους

Αν όχι: 10 πόντους / απόσταση μέχρι το κοντινότερο φαγητό και αφαιρώ:

25 πόντους / απόσταση από το(τα) φάντασμα(τα)

Διάλεξα τον αριθμό 25 ώστε όταν η απόσταση μέχρι το φάντασμα είναι λιγότερη από 2.5, η αφαίρεση σκορ ξεπερνάει την πρόσθεση(μαζ 10), επομένως το pacman αποφεύγει επικίνδυνες θέσεις κοντά σε φαντάσματα ακόμα και αν υπάρχει φαγητό σε αυτές.

(Question 2)

Για τον Minimax Agent:

Τρέχω μια recursive minimax συνάρτηση για κάθε successor state της θέσης του pacman και επιστρέφω την κίνηση που αντιστοιχεί στην successor state με το υψηλότερο σκορ(max).

Όσον αφορά την minimax:

Ελέγχω αν το παιχνίδι έχει τελειώσει, ή έχω φτάσει στο depth που ζητείται. Αν ναι, επιστρέφω την evaluation function για αυτό το state. Διαφορετικά:

-Αν είναι η σειρά του agent 0 (pacman), τρέχω την minimax συνάρτηση για κάθε successor state και της δίνω agent index = 1 (1st ghost) και επιστρέφω το υψηλότερο(max) σκορ από τα successor states

-Αν είναι η σειρά ενός ghost agent:

-Αν δεν είναι ο τελευταίος ghost agent που κινείται στο γύρο, τρέχω την minimax συνάρτηση για κάθε successor state και της δίνω agent index = agent index + 1 (next ghost) και επιστρέφω το χαμηλότερο (min) σκορ από τα successor states

-Αν είναι ο τελευταίος, τρέχω την minimax συνάρτηση για κάθε successor state και της δίνω agent index = 0 (pacman), depth = depth + 1(καθώς θέλω να προχωρήσω στο επόμενο βάθος) και επιστρέφω το χαμηλότερο (min) σκορ από τα successor states

(Question 3)

Για τον Alpha-Beta Agent:

Ακολουθήσα τον αλγόριθμο που δίνεται στην εκφώνηση για τις συναρτήσεις minVal, maxVal. Έτσι:

Τρέχω την minVal συνάρτηση με $\alpha = -inf$, $\beta = inf$ για κάθε successor state της θέσης του pacman και επιστρέφω την κίνηση που αντιστοιχεί στην successor state με το υψηλότερο σκορ(max).

Όσον αφορά την minVal:

Ελέγχω αν το παιχνίδι έχει τελειώσει, ή έχω φτάσει στο depth που ζητείται. Αν ναι, επιστρέφω την evaluation function για αυτό το state. Διαφορετικά:

Είναι η σειρά ενός ghost agent:

–Αν δεν είναι ο τελευταίος ghost agent που κινείται στο γύρο, τρέχω την minVal συνάρτηση(ακομά είναι σειρά των φαντασμάτων) για κάθε successor state και της δίνω $\text{agent index} = \text{agent index} + 1$ (next ghost), τα ανανεωμένα alpha, beta βάσει του αλγορίθμου της εκφώνησης και επιστρέφω το χαμηλότερο (min) σκορ από τα successor states

–Αν είναι ο τελευταίος, τρέχω την maxVal συνάρτηση(σειρά του pacman) για κάθε successor state και της δίνω $\text{agent index} = 0$ (pacman), $\text{depth} = \text{depth} + 1$ (καθώς θέλω να προχωρήσω στο επόμενο βάθος), τα ανανεωμένα alpha, beta βάσει του αλγορίθμου της εκφώνησης και επιστρέφω το χαμηλότερο (min) σκορ από τα successor states

Όσον αφορά την maxVal:

Ελέγχω αν το παιχνίδι έχει τελειώσει, ή έχω φτάσει στο depth που ζητείται. Αν ναι, επιστρέφω την evaluation function για αυτό το state. Διαφορετικά:

Είναι η σειρά του agent 0 (pacman). Τρέχω την minimax συνάρτηση για κάθε successor state και της δίνω $\text{agent index} = 1$ (1st ghost), τα ανανεωμένα alpha, beta βάσει του αλγορίθμου της εκφώνησης και επιστρέφω το υψηλότερο(max) σκορ από τα successor states.

(Question 4)

Για τον Expectimax Agent, η υλοποίηση μου είναι ίδια με τον minimax agent(Q2) με τη διαφορά ότι όταν είναι η σειρά των φαντασμάτων, δεν παίρνουν την ελάχιστη τιμή από τις expectimax των successor states, αλλά παίρνουν τον μέσο όρο των τιμών αυτών, σύμφωνα με τον αλγόριθμο expectimax.

Για τη σειρά του pacman δεν αλλάζει παρά μόνο το όνομα της recursive συνάρτησης.

(Question 5)

Για την Better Evaluation Function, η υλοποίηση μου είναι παρόμοια με την evaluation function (Q1), προσαρμοσμένη στο δεδομένο ότι πλέον εξετάζουμε το current state και όχι τα actions όπως στο πρώτο ερώτημα. Έτσι:

Για κάθε state στην οποία βρίσκεται το pacman, βρίσκω το αρχικό σκορ της και σε αυτό προσθέτω:

10 πόντους / απόσταση μέχρι το κοντινότερο φαγητό

και αφαιρώ:

7 πόντους / απόσταση από το(τα) φάντασμα(τα)

Είχα αρχικά τον αριθμό 25 όπως στο πρώτο ερώτημα, ωστόσο θέλησα να το μειώσω για λόγους ταχύτητας ώστε να αυξήσω το σκορ. Επομένως έκανα δοκιμές με μικρότερες τιμές και κατέληξα στην τιμή 7 ως μια τιμή που:

–Κρατάει το pacman ασφαλές από τα φαντάσματα και

–Είναι αρκετά μικρή ώστε το pacman να 'ρискάρει' παραπάνω να μαζέψει το φαγητό παρότι μπορεί να υπάρχουν φαντάσματα κοντά.