Εργασία 2 στη Τεχνητή Νοημοσύνη
Μυλωνόπουλος Δημήτριος – Α.Μ: 1115201600112

Πρόβλημα 4:(Pacman Project 2)

Στο πρόγραμμα μου απαντήθηκαν όλα τα ερωτήματα και συλλέγει όλες τις μονάδες απ' το πρόγραμμα αξιολόγησης του pacman autograder.py. Για την επίλυση των ερωτημάτων του project βασίστηκα στην θεωρία του μαθήματος και στις διαλέξεις. Για την υλοποίηση του προγράμματος έγιναν αλλαγές στο multiAgents.py.

Question 1: Reflex Agent

Σκοπός του συγκεκριμένου ερωτήματος είναι να αξιολογεί ζευγάρια state-action και να σου επιστρέφει μεγαλύτερο score στα ζευγάρια που θεωρούνται καλύτερες κινήσεις ώστε να μπορούμε να πετύχουμε μεγαλύτερο score.

Η λογική στην οποία βασίστηκα είναι σχετικά απλή:

- Υπολογίζω τις αποστάσεις των φαντασμάτων και αν κάποιο φάντασμα είναι σε απόσταση 0 από τον pacman, τότε επιστρέφω αρνητικό score εφόσον δεν υπάρχει περεταίρω λόγος αξιολόγησης της κίνησης, εφόσον σε αυτή την κίνηση ο Pacman χάνει.
- Έπειτα υπολογίζω την μικρότερη απόσταση του PacMan από κάποιο φαγητό. Χρησιμοποιώ για αυτό ManhattanDistance. Επειδή όμως θέλω να αξιολογεί τις κοντινότερες αποστάσεις ως καλύτερες έφτιαξα τον λόγο α/απόσταση. Ο συντελεστής α = 10.0 επιλέχθηκε ύστερα από δοκιμές επειδή έδινε καλύτερο score.
- Επιπλέον επειδή ο pacman είχε την τάση να παραμένει ακίνητος μέσα στην πίστα, κάτι το οποίο έβλαπτε το score έβαλα πως αν το state μετά την κίνηση είναι ίδιο με πριν (δηλαδή ο pacman παρέμεινε στάσιμος) να τον τιμωρεί μειώνοντας το score κάτα -7 (Ο αριθμός κρίθηκε κατάλληλος δεδομένου των υπόλοιπων τιμών)
- Τέλος, επειδή ο pacman πολλές φορές κόλλαγε μπροστά από φαγητό αν η θέση του pacman μετά την κίνηση περιέχει φαγητό τότε το score του αυξάνεται κατά 10.

Question 2: Minimax Agent

Σε αυτό το ερώτημα υλοποιήθηκε η συνάρτηση MiniMax. Στην οποία στο pacman project ο Pacman ανταγωνίζεται τα φαντάσματα. Γι' αυτό το λόγο ο pacman αντιστοιχεί σε κόμβους Max του minimax ενώ τα φαντάσματα αντιστοιχούν σε κόμβους Min. Επειδή το πρόβλημα όπως το έχουμε διδαχτεί έχει εναλλάξ τους Max-Min κόμβους χρειάστηκε να τροποποιηθεί για να ανταποκρίνεται στις απαιτήσεις του προβλήματος. Συγκεκριμένα είχαμε την ποσότητα agentIndex η οποία αντιστοιχούσε στο 0 για τον pacman και στις μεγαλύτερες τιμές για τα φαντάσματα. Έτσι όταν γινόταν κλήση της Min_Value συνάρτησης και τις Max_Value συνάρτησεις μπορούσαμε να ξέρουμε αν μετά έπρεπε να καλέσουμε αναδρομικά την Min_Value ή τη Max_Value αντίστοιχα. Οι Min_Value και Max_Value επιστρέφουν τις τιμές των κόμβων Min και Max του προβλήματος Minimax τροποποιημένες όπως είπαμε στις απαιτήσεις του προβληματος. Το οποίο βασίζεται στα legal_actions που δίνεται από το παρόν state.

Συγκεκριμένα στην Max_Value(curr_state, agentIndex, depth) αν το depth στο οποίο βρισκόμαστε είναι ίδιο με το βάθος του δέντρου του προβλήματος τότε επιστρέφομουμε την evaluationFucntion όπως και αν δεν υπάρχουν νόμιμες κινήσεις για την συγκεκριμένη κατάσταση. Και στις δύο περιπτώσεις βρισκόμαστε σε φύλλο του δέντρου.

Έπειτα στο MaxValue υπολογίζω απ τα legal actions αυτό που δίνει την μεγαλύτερη τιμή και καταγράφω και την αντίστοιχη κίνηση. Για αυτό καλώ αναδρομικά την Min_Value προφανώς αυξάνοντας το agent. Αν δεν ειναι το βάθος 0 δηλαδή δεν βρισκόμαστε στον κόμβο ρίζα η συνάρτηση επιστρέφει την τιμή της καλύτερης κίνησης αλλιώς την καλύτερη κίνηση εφόσον είναι αυτό που μας ενδιαφέρει από την minimax.

Στην Min_Value(curr_state, agentIndex, depth) αν δεν υπάρχουν legal_actions τότε επιστρέφω το evaluation function. Έπειτα υπολογίζω την κίνηση με το μικρότερη αυτή τη φορά τιμη. Για αυτό έχω δύο περιπτώσεις: Είτε να είμαστε στο τελευταίο φάντασμα άρα ο επόμενος κόμβος να είναι pacman που σημαίνει οτι πρέπει να καλέσουμε την Max_Value με το depth αυξημένο κατά 1 και το agentIndex =0. Αλλιώς ο επόμενος κόμβος είναι φάντασμα άρα θα καλέσουμε την Min_value με ίδιο depth και αυξημένο το agentIndex κατά 1. Τέλος επιστρέφουμε τη μικρότερη τιμή που βρήκαμε.

Επιστρέφω την κλήση της συνάρτησης Max_Value(gameState, 0, 0)

Question 3: AlphaBetaAgent

Στην συνάρτηση αυτή βασίστηκα σε μεγάλο βαθμό στην συνάρτηση Minimax, επομένως δε θα ξανααναφερθώ στον τρόπο λειτουργίας της, παρά μόνο στις “βελτιώσεις” που έχουν γίνει για να πετυχαίνουμε κλάδεμα κόμβων με το A-B pruning.

Έχω προσθέσει στις συναρτήσεις Max_Value και Min_Value δύο ορίσματα τα a, b που παριστάνουν τις τιμές a,b που έχουμε στον αλγόριθμος a-b pruning. Για την υλοποίηση του αλγορίθμου βασίστηκα στα πλαίσια που δόθηκαν μαζί με την ερώτηση στη σελίδα του project τα οποία συμφωνούν με τον αλγόριθμο που δίνεται στο υλικό του μαθήματος.

Στην Max_Value όταν ψάχνω να βρω την κίνηση που δινει το μεγαλύτερο value ελεγχω αν η τιμή του καλύτερη μέχρι στιγμής value είναι μεγαλύτερη του b και αν είναι τότε την επιστρέφω. Αλλιώς θέτω ως a το μεγαλύτερο απ τα a και την καλύτερη τιμή και τέλος επιστρέφω την μεγαλύτερη τιμή που βρέθηκε.

Στην Min_Value όταν ψάχνω να βρώ την κίνηση που δίνει το μικρότερο value ελέγχω αν αυτή είναι μικρότερη του a αν είναι την επιστρέφω, αλλιώς θέτω κάθε φορά που ελέγχω μια κίνηση το b ως το ελάχιστο απ τα v,b και τέλος επιστρέφω την μεγαλύτερη τιμή που βρέθηκε.

Στην συνάρτηση μου επιστρέφω το max_value με depth=agentIndex =0 και a = -oo και b = +oo.

Question 4: ExpectimaxAgent

Για την συνάρτηση αυτή βασίστηκα πάλι στο Minimax αλγόριθμο. Ο max_value παραμένει ίδιος, παραμόνο ότι καλεί την chance_value (περισσότερα για αυτή στη συνέχεια) αντί για τον min, όμως πλέον δεν έχουμε Min κόμβους αλλά Chance κόμβους και γι αυτό έφτιαξα την συνάρτηση chance_value, η οποία αντί να υπολογίζει την κίνηση με το μικρότερο value, ουσιαστικά αθροίζει τις τιμές των legal_actions λαμβάνοντας υπόψιν στην κάθε μια και μια πιθανότητα. Επειδή όμως θεωρήθηκαν όλες οι κινήσεις ισοπίθανες, βασιζόμενος στο ότι sum pr * r όπου r ισούτε με την τιμή και pr ισούτε με την πιθανότητα της δεδομένου ότι το pr ειναι σταθερό και ισούτε με

1/αριθμό_νόμιμων_κινήσεων βγαίνει εκτός αθροίσματος και τελικά έχουμε pr *sum r. Και για λόγους ευκολίας τροποποίησα τη συνάρτηση μου ώστε να υπολογίζει την τιμή με αυτόν τον τρόπο.

Καλώ την συνάρτηση Max_Value με agentIndex=depth=0.

Question 5:betterEvaluationFunction

Για τη συνάρτηση αυτή βασίστηκα αρκετά στο ερώτημα 1 όμως έκανα κάποιες αλλαγές και πρόσθεσα κάποια πράγματα ώστε και αυτή να επιστρέφει καλό score δεδομένου πως εδώ συγκρίνουμε states χωρίς actions που τους αντιστοιχούν.

Η συνάρτηση μου λειτουργει ως εξής:
Υπολογίζω το φαγητό που απέχει λιγότερο από τον pacman μας και αυτό το μετατρέπω σε α/αποσταση_φαγητού_που_απέχει_λιγότερο έτσι ώστε να επιβραβεύει περισσότερο τα κοντινότερα. Επίσης αξιολόγησα τα states με βάση και το πόσα φαγητά υπάρχουν στην πίστα έχοντας παρόμοια λογική με προηγουμένως, δηλαδή όσο λιγότερα τόσο το καλύτερο, με συντελεστή α =400. Αν παρουσιαζει προβλημα ο αριθμός των φαγητών (σε εμένα τουλάχιστον δεν δημιουργούσε) παρακαλώ μην τον λαβεται υπόψιν σας στο αποτέλεσμα (Μεταβλητή food).

Κάτι που έχει προστεθεί είναι όταν τα φαντάσματα δεν είναι όλα τρομαγμένα ο pacman να κυνηγά το κοντινότερο pellet με 2.5 φορές μεγαλύτερη προτεραιότητα, απ’ ό,τι το φαγητό. Συγκεκριμένα επιλέχθηκαν οι συντελεστες 10 και 25, ύστερα από δοκιμές. Άν δεν έχω όλα τα φαντάσματα τρομαγμένα θα επιστρέψω το score του state + τον αριθμό που εφτιαξα για τον αριθμό των φαγητών + τον αριθμό για την μικρότερη απόσταση των φαγητού + τον αριθμό για την μικρότερη απόσταση pellet.

Αν όμως τα φαντάσματα είναι όλα τρομαγμένα τότε βρίσκω την μικρότερη απόσταση από φαντασμα και δημιουργώ τον αριθμό α/μικρότερη_αποσταση με α=30, δηλαδή 3 φορές μεγαλύτερη προτεραιότητα απ’ το κοντινότερο φαγητό, γιατί τα τρομαγμένα φαντάσματα δίνουν αρκετούς πόντους αν τα φάει ο pacman. Άρα σε αυτή την περίπτωση θα επιστρέψω το score του state + αριθμό που έφτιαξα για τον αριθμό των φαγητών + τον αριθμό για την μικρότερη απόσταση φαγητού + τον αριθμό για την μικρότερη απόσταση φαντάσματος. Δεν λαμβάνω υπόψιν τα pellets.

Επέλεξα να κυνηγά φαντάσματα μόνο όταν είναι όλα τρομαγμένα με την λογική ότι αν δεν είναι όλα ο timer των υπολοιπώμενων έχει πέσει αρκετά άρα μας συμφέρει καλύτερα να κυνηγήσουμε pellet για να τα επαναφέρουμε σε φοβισμένη κατάσταση. Τέλος δεν υπάρχει λόγος να προσπαθούμε να αποφύγουμε φαντάσματα, γι’ αυτό και δεν το λαμβάνω υπόψιν μου. (Έβλαπτε το score)