Τεχνητή Νοημοσύνη
Εργασία 2 - Berkeley Pacman Project 2: Multi-Agent Search
Χειμερινό εξάμηνο 2021-2022

Ευάγγελος Δημητριάδης 1115201700287

1 Reflex Agent

Από ένα state μπορούμε να εξάγουμε τις εξής χρήσιμες πληροφορίες:

- Την θέση του pacman
- Την θέση των pellets
- Τα states των ghosts που συμπεριλαμβάνουν την θέση τους
- Τον αριθμό κινήσεων που τα ghosts θα παραμείνουν scared

Στην reflexAgent.evaluationFunction() αυτές οι πληροφορίες είναι διαθέσιμες για το τωρινό state και για το επόμενο state που θα προκύψει αν ο pacman κάνει μια συγκεκριμένη κίνηση.

Το evaluation function που έχω δημιουργήσει διαμορφώνει το score ως εξής:
Αν η λίστα με τα pellets στο νέο state είναι άδεια επιστρέφεται το μεγαλύτερο δυνατό score, αφού σε αυτήν την περίπτωση ο Pacman κερδίζει.
Αν το pacman στο νέο state βρίσκεται εντός του range στο οποίο μπορεί να τον φάει ένα φάντασμα τότε επιστρέφεται το μικρότερο δυνατό score , αφού σε αυτήν την περίπτωση ο Pacman θα χάσει.
Αν ο αριθμός των pellets στο νέο state είναι μικρότερος από τον αριθμό των pellets στο προηγούμενο state ,το score αυξάνεται κατά 1.
Μετά βρίσκεται η απόσταση του κοντινότερου pellet στο νέο state και στο score προσθέτεται το 1 δια την ελάχιστη απόσταση (γίνεται διαίρεση του ένα ώστε η αύξηση να είναι ανάλογη με την αύξηση αν φαγωθεί ένα pellet ,όσο μικρότερη η απόσταση τόσο μεγαλύτερο το score).

Έτσι ένα state επιλέγεται σίγουρα αν είναι winning,απορρίπτεται σίγουρα αν είναι losing. Αν δεν ισχύει τίποτα από τα δύο έχουν προτεραιότητά τα states που τρώνε κάποιο pellet και από αυτά έχουν προτεραιότητα αυτά που βρίσκονται πιο κοντά σε κάποιο pellet.

Στο mediumClassic maze με 1 και 2 ghost ο pacman κάνει clear το maze.

2 Minimax

Για την υλοποίηση του minimax χρησιμοποιείται η συνάρτηση max που αντιπροσωπεύει μια κίνηση του pacman και η συνάρτηση min που αντιπροσωπεύει την κίνηση των φαντασμάτων.
Η συνάρτηση MAX βρίσκει το score για όλες τις πιθανές κινήσεις του pacman και επιλέγει αυτήν με το καλύτερο score.
Η συνάρτηση MIN αντιπροσωπεύει μια κίνηση από κάθε φάντασμα. Έτσι για κάθε κίνηση του πρώτου ghost παίρνουμε κάθε κίνηση του δεύτερου ghost κ.ο.κ. Για να επιτευχθεί αυτό χρησιμοποιείται εμφώλευση, η συνάρτηση min έχει ένα argument ghostmove που σηματοδοτεί ποιανού ghost είναι η σειρά να κινηθεί, έτσι όταν καλείται η min από την max βρίσκει όλες τις κινήσεις του πρώτου ghost και μετά για καθεμία από αυτές καλεί τον ευατό της με ghostmove=2

κ.ο.κ.
Η συνάρτηση min παίρνει το score για κάθε δυνατή κίνηση ενός ghost και επιλέγει την κίνηση με το μικρότερο score.

Με τον minimax ο αλγόριθμος υποθέτει ότι τα ghosts θα κάνουν σε κάθε γύρο την καλύτερη δυνατή κίνηση ενώ στην πραγματικότητα κινούνται κάπως τυχαία. Έτσι στο παράδειγμα που δόθηκε ο pacman λαμβάνει ως δεδομένο ότι και τα 2 ghosts που τον περικυκλώνουν θα κινηθούν προς την κατεύθυνση του για να τον φάνε αφού αυτή είναι η καλύτερη δυνατή κίνηση. Έτσι έχει να κάνει την επιλογή ανάμεσα σε 3 κινήσεις (stop,east,west) που και οι τρεις έχουν πάρα πολύ χαμηλό score αφού και στις τρεις ο pacman χάνει (σύμφωνα με την εκτίμηση του minimax) , επιλέγεται η κίνηση προς τα δεξιά επειδή ο pacman χάνει σε λίγο λιγότερες κινήσεις οπότε περνάει λίγος λιγότερος χρόνος που ο pacman κινείται χάνοντας score. Ο pacman θα μπορούσε να πάρει το ρίσκο να κινηθεί προς το μπλε φάντασμα καθώς υπάρχει η πιθανότητα αυτό να κινηθεί προς την αντίθετη κατεύθυνση δίνοντας στον pacman την ευκαιρία να φάει pellets αυξάνοντας δραματικά το score του.

3 Alpha-Beta Pruning

Η βασική λειτουργία του αλγόριθμου είναι ίδια με αυτήν του minimax.
Η διαφορά είναι ότι στις συναρτήσεις min και max περνιούνται ως ορίσματα το alpha και το beta. Ο αλγόριθμος ξεκινάει από max με alpha=+∞ και beta=−∞.
Για τον max αλλάζει το alpha για να είναι η μεγαλύτερη τιμή που έχει βρεθεί μέχρι στιγμής.Όταν βρεθεί τιμή μεγαλύτερη του β σημαίνει ότι δεν υπάρχει περίπτωση ο κόμβος max να επιλεχθεί από τον κόμβο min που είναι πατέρας του (αν επιλεχθεί κάποια άλλη τιμή μπορεί να είναι μόνο μεγαλύτερη), οπότε επιστρέφεται η τιμή (γίνεται κλάδεμα των υπόλοιπων κόμβων).
Για τον min αλλάζει το beta για να είναι η μικρότερη τιμή που έχει βρεθεί μέχρι στιγμής.Όταν βρεθεί τιμή μικρότερη του alpha σημαίνει ότι δεν υπάρχει περίπτωση ο κόμβος min να επιλεχθεί από τον κόμβο max που είναι πατέρας του (αν επιλεχθεί κάποια άλλη τιμή μπορεί να είναι μόνο μικρότερη), οπότε επιστρέφεται η τιμή (γίνεται κλάδεμα των υπόλοιπων κόμβων).
Η max και min καλούν η μια την άλλη δίνοντας ως ορίσματα τα alpha και beta που έχουν μέχρι στιγμής.

Για το παράδειγμα που δόθηκε οι αποφάσεις που παίρνει ο pacman είναι οι ίδιες ,όπως ήταν και αναμενόμενο,αλλά λόγο της χρήσης του pruning ο χρόνος που χρειάστηκε για να ληφθούν αυτές οι αποφάσεις ήταν μικρότερος. Για depth=3 ο minimax χρειάστηκε 6,296s ενώ ο alpha-beta 5,219s.

4 Expectimax

Ο αλγόριθμος έχει την ίδια λειτουργία με αυτή του minimax με την διαφορά να είναι ότι η συνάρτηση min από το επιστρέφει την ελάχιστη τιμή από τα παιδιά του κόμβου επιστρέφει τον μέσο όρο των τιμών. Επιστρέφεται ο μέσος όρος των scores επειδή δεν είναι εγγυημένο ότι ο αντίπαλος θα παίξει την βέλτιστη κίνηση και είναι πιθανό να παίξει οποιαδήποτε από τις δυνατές κινήσεις.

Στο παράδειγμα που δόθηκε ο pacman κερδίσει τις μισές φορές με τον expectimax ενώ με τον minimax χάνει συνέχεια.Αυτό συμβαίνει γιατί ο minimax υποθέτει ότι για κάθε γύρο τα ghosts θα κάνουν την καλύτερη δυνατή κίνηση, έτσι ακόμα και αν υπάρχει κάποιο άνοιγμα που του επιτρέπει να φάει ένα pellet και πιθανόν να κερδίσει ο pacman θα το αγνοήσει καθώς θα υποθέσει ότι κάποιο ghost θα τον κλείσει χάνοντας έτσι το παιχνίδι. Αντίθετα ο expectimax θα πάρει το ρίσκο να περάσει το άνοιγμα για να φάει τα pellets καθώς μπορεί τα ghost να κάνουν κάποιο λάθος. Τις μισές φορές κάνουν όντως , μη λαμβάνοντας την καλύτερη κίνηση, και ο

pacman κερδίζει.

5 Evaluation Function

Αν ένα state είναι winning τότε επιστρέφεται το μεγαλύτερο δυνατό score.
Αν ένα state είναι losing τότε επιστρέφεται το μικρότερο δυνατό score.
Για το state βρίσκεται η απόσταση προς το κοντινότερο pellet και προσθέτεται στο score 1 δια την απόσταση.
Για κάθε power pellet που βρίσκεται εντός απόστασης 2 το score αυξάνεται κατά 1.
Τέλος στο score προσθέτεται το 1 δια το 1 δια το Score του state (όσο μεγαλύτερο το score τόσο μεγαλύτερη η αύξηση).