Τεχνητή Νοημοσύνη
(Project 2 Pacman)

Ονοματεπώνυμο: Δημήτριος Σιταράς
Αριθμός μητρώου: 1115201800178
Εξάμηνο: 5ο

Πρόβλημα 5

multiAgents.py:

*(*class ReflexAgent) *evaluationFunction:*
Υλοποίησα μια απλή συνάρτηση αξιολόγησης η οποία επιστρέφει μια εκτίμηση της αναμενόμενης χρησιμοτητας της κατάστασης του pacman με βάση το κοντινότερο φαγητό. Αναλυτικότερα, είναι κακό για τον pacman να μην τρώει το κοντινότερο φαγητό, διότι δεν θα φτάσει σε μια καλύτερη κατάσταση, οπότε πρέπει να το φαει, γι' αυτό η χρησιμότητα της κατάστασης που επιστρέφει η συνάρτηση αξιολόγησης είναι: το σκορ αυτης της καταστασης μείον την απόσταση του κοντινότερου φαγητού (το ορίζω δηλαδη ως penalty). Επίσης, αν ο pacman έχει πολύ κοντά του ghost πρέπει να το αποφύγει αλλιως θα χάσει, επομένως, η συνάρτηση αξιολόγησης σε αυτή την περίπτωση επιστρέφει ως χρησιμότητα καταστασης την πιο μεγαλη αρνητικη τιμή (-float('inf')).

(class MinimaxAgent) getAction,minimax,maxvalue,minvalue:
Υλοποίησα τον αλγόριθμο minimax όπως ακριβώς βρίσκεται στις διαφάνειες του μαθήματος (χρησιμοποιώντας τις συναρτήσεις minimax, maxvalue, minvalue). Ο MAX (maximazer) είναι ο pacman, τα ghosts ειναι οι MIN (minimizers), στην υλοποίηση μου ο MAX δηλαδή ο pacman "παίζει πρώτος". Η τιμή minimax ενός κόμβου είναι η καλύτερη τιμή χρησιμότητας που μπορεί να επιτευχθεί από έναν πράκτορα στον κόμβο αυτό, με την προϋπόθεση ότι όλοι οι πρακτορες (δηλαδή ο pacman και τα ghosts) παίζουν βέλτιστα από εκείνο το σημείο μέχρι το τέλος του παιχνιδιού. Τέλος, η συνάρτηση minimax επιστρέφει την minimax κίνηση δηλαδή την καλύτερη επόμενη κινηση για τον πρακτορα MAX (pacman) που μπορεί να κάνει από την τρέχων κατάσταση στην οποία βρίσκεται.

(class AlphaBetaAgent) alphabeta,maxvalue,minvalue:

Υλοποίησα τον αλγόριθμο alpha-beta όπως ακριβώς βρίσκεται στις διαφάνειες του μαθήματος (χρησιμοποιώντας τις συναρτήσεις alphabeta, maxvalue, minvalue). Ουσιαστικά, είναι η βελτιωμένη εκδοχή του minimax αλγορίθμου καταφέρνοντας να μειώσει στο μισό τον αριθμο τον καταστάσεων που εξετάζει ο minimax και αυτό το πετυχαίνει με το κλάδεμα (pruning) των κομβων μέσω των παραμέτρων α και β που προσδιορίζουν φράγματα για τις τιμές χρησιμότητας που αντιγράφονται κατά μήκος μιας διαδρομής στο δέντρο παιχνιδιού (η παράμετρος α αντιστοιχεί στην τιμης της καλύτερης επιλογής για τον MAX που έχουμε βρει μέχρι στιγμής σε οποιοδήποτε κόμβο κατά μήκος της διαδρομής και ομοίως η παράμετρος β για το MIN). Τελικά, (όπως και η συνάρτηση minimax) η συναρτηση alphabeta επιστρέφει την καλύτερη επόμενη κινηση για τον πρακτορα MAX (pacman) που μπορεί να κάνει από την τρέχων κατάσταση στην οποία βρίσκεται.

(class ExpectimaxAgent) expectiminimax,maxvalue,minvalue:

Υλοποίησα τον αλγόριθμο expectiminimax όπως περιγράφεται στις διαφάνειες του μαθήματος (χρησιμοποιώντας τις συναρτήσεις expectiminimax, maxvalue, minvalue). Ουσιαστικά, βασίζεται στον minimax αλγόριθμο μόνο που οι καταστάσεις δεν είναι οριστικές τιμές minimax, διότι στο συγκεκριμένο πρόβλημα τα ghosts κινουνται με τυχαία κινήσεις (κάνοντας το παιχνίδι “στοχαστικο”). Επομένως, ο expectiminimax αλγόριθμος υπολογίζει τις αναμενόμενες τιμές minimax ως το μέσο όρο των τιμών για όλα τα δυνατά αποτελέσματα των κόμβων τύχης (αυτο γίνεται στην συνάρτηση expvalue, οπου και ορίζω τους κομβους τύχης ως την πιθανότητα που προκυπτει απο το σύνολο των κινήσεων που μπορεί να κινηθεί ένας πράκτορα ghost). Τελικά, η συνάρτηση expectiminimax επιστρέφει την καλύτερη επόμενη κινηση για τον πρακτορα MAX (pacman) που μπορεί να κάνει από την τρέχων κατάσταση στην οποία βρίσκεται, δεδομένου ότι οι πρακτορες ghosts αυτήν την φορά κινούνται τυχαία.

betterEvaluationFunction:

Υλοποίησα μια καλύτερη συνάρτηση αξιολόγησης (συγκριτικά με την *evaluationFunction)* η οποία επιστρέφει μια βελτιωμένη εκτίμηση της αναμενόμενης χρησιμότητας της καταστασης του pacman με βάση το κοντινότερο φαγητό, το πληθος των φαγητών που απομένουν, τη κοντινότερη κάψουλα, το πληθος των καψουλών που απομένουν και το κοντινότερο φάντασμα. Αναλυτικότερα, οπως ειπα και πριν, πρέπει ο pacmac να φαει το κοντινότερο φαγητο για να φτάσει σε μια καλύτερη κατάσταση, κοντινότερη στην κατάσταση νίκης. Επειτα, οσο πιο λιγα φαγητά εχει να φαει ο pacman τοσο πιο κοντά είναι στην κατάσταση νίκης, οπότε ειναι κακο να του εχουν απομείνει πολλα φαγητα. Ο pacman, επίσης, κερδίζει πολλούς πόντους όταν τρώει τα ghosts οπότε είναι προς συμφέρον του (πρέπει) να τρώει την κοντινότερη καψουλα και μαλιστα οσο πιο λίγες καψουλες του απομένουν τοσο το καλύτερο (σε αντίθετη περίπτωση που οι κάψουλες είναι πολλές είναι “κακό” γιατί έχει δοθεί λίγες φορές ή καθόλου η δυνατότητα στον pacman να φαει τα ghosts κερδίζοντας παραπάνω πόντους), αφού ενδέχεται να κερδίσει περισσότερους πόντους. Επιπλέον, ο pacman είναι σε πλεονεκτική θεση οταν το κοντινότερο φαντασμα βρίσκεται σε όσο το δυνατόν μεγαλύτερη απόσταση από εκείνον (και σε μειονεκτικη θεση αν αυτο βρίσκεται κοντά του). Άρα, τελικά η συνάρτηση αξιολόγησης επιστρέφει ως εκτίμηση της αναμενόμενης χρησιμότητας της κατάστασης το score της κατάστασης που βρίσκεται ο pacman στο οποίο προσθέτω επιπλέον την απόσταση από το κοντινότερο φάντασμα διαιρεμένο με το κατάλληλο συντελεστή βαρύτητας (ώστε να λαμβάνεται υπόψη ως “κέρδος” μονο οταν ο pacman είναι αρκετά μακρια απο αυτό) και “τα ποσοστά” του κοντινότερου φαγητου,του πληθος των φαγητων που απομένουν, της κοντινότερης κάψουλας, του πλήθους των καψουλών, πολλαπλασιασμενα επίσης με τους αντίστοιχους συντελεστές βαρύτητας που προκύπτουν σύμφωνα με την σημαντικότητα του κριτηρίου, για παράδειγμα το πληθος των φαγητων εχουν αμεση σχεση με το πόσο κοντά είναι ο pacman στην κατάσταση νικης επομένως είναι πολύ σημαντικό κριτήριο και γι’ αυτο πολλαπλασιάζω το αντίστοιχο ποσοστο με τον μεγαλύτερο συντελεστή βαρύτητας σε σχέση με τα υπόλοιπα ποσοστά.

Στις περιπτώσεις που η καταστασηειναι καταστασηνίκης, ήττας η συνάρτηση αξιολόγησης επιστρεφει ως χρησιμότητα καταστασης την πιο μεγαλη θετική τιμή (float("inf") και την πιο μικρή αρνητική τιμή (-float("inf")) αντίστοιχα. Ακόμα, στην περίπτωση που σε μια κατάσταση ο pacman έχει δίπλα του ένα φάντασμα τότε η συνάρτηση αξιολόγησης προκειμένου ο pacman να “καταλάβει” πως πρέπει να αποφύγει το φαντασμα επιστρέφει ως χρησιμότητα καταστασης την πιο μεγαλη αρνητικη τιμή (-float('inf')).