

Q1: Reflex Agent

Χρησιμοποίησα την απόσταση Manhattan για να υπολογίσω την απόσταση από την πλησιέστερη τροφή. Υπολόγισα την απόσταση από τα φαντάσματα και προσθέσα ποινή αν κάποιο φάντασμα είναι πολύ κοντά και δεν είναι φοβισμένο από το power-up. Συνδύασα τις αποστάσεις με το τρέχον σκορ για να υπολογίσω την τελική αξιολόγηση.

Q2: Minimax Agent

Χρησιμοποίησα αναδρομική συνάρτηση minimax (τα δύο helper functions, min/maxValue) για να υπολογίσω τις βέλτιστες κινήσεις. Εναλλάξα μεταξύ των επιπέδων max (Pacman) και min (φαντάσματα) για να υπολογίσω τις τιμές minimax. Τέλος, επέκτεινα το δέντρο παιχνιδιού σε αυθαίρετο βάθος και χρησιμοποίησα την self.evaluationFunction για την αξιολόγηση των φύλλων.

Q3: Alpha-Beta Pruning

Πρόσθεσα παραμέτρους alpha και beta στη συνάρτηση minimax για να εκτελέσω το κλάδεμα alpha-beta. Αρχικοποίησα τις μεταβλητές alpha και beta σε αρνητικό και θετικό άπειρο αντίστοιχα. Για κάθε νόμιμη κίνηση του Pacman, δημιούργησα την κατάσταση διαδόχου και κάλεσα τη συνάρτηση alphaBeta για να υπολογίσω το σκορ. Ενημέρωσα το καλύτερο σκορ και την καλύτερη κίνηση με βάση το αποτέλεσμα της alphaBeta και επέστρεψα την καλύτερη κίνηση. Η συνάρτηση alphaBeta ελέγχει αν η κατάσταση είναι νίκη, ήττα ή αν έχει φτάσει στο μέγιστο βάθος και επιστρέφει την τιμή αξιολόγησης. Για τον Pacman (maximizer), η συνάρτηση ενημερώνει την τιμή v με το μέγιστο μεταξύ της τρέχουσας τιμής v και της τιμής που επιστρέφει η alphaBeta, και αν η τιμή v είναι μεγαλύτερη από την beta, επιστρέφει την τιμή v (κλάδεμα). Για τα φαντάσματα (minimizer), η συνάρτηση ενημερώνει την τιμή v με το ελάχιστο μεταξύ της τρέχουσας τιμής v και της τιμής που επιστρέφει η alphaBeta, και αν η τιμή v είναι μικρότερη από την alpha, επιστρέφει την τιμή v (κλάδεμα).

Q4: Expectimax Agent

Χρησιμοποίησα τη συνάρτηση expectimax για να υπολογίσω τις προσδοκώμενες τιμές αντί για τις ελάχιστες τιμές. Υπέθεσα ότι τα φαντάσματα επιλέγουν τις κινήσεις τους τυχαία και ομοιόμορφα. Στη συνάρτηση getAction, αρχικοποίησα τις μεταβλητές bestScore και

bestAction. Για κάθε νόμιμη κίνηση του Pacman, δημιούργησα την κατάσταση διαδόχου και κάλεσα τη συνάρτηση expectimax για να υπολογίσω το σκορ. Ενημέρωσα το καλύτερο σκορ και την καλύτερη κίνηση με βάση το αποτέλεσμα της expectimax και επέστρεψα την καλύτερη κίνηση. Η συνάρτηση expectimax ελέγχει αν η κατάσταση είναι νίκη, ήττα ή αν έχει φτάσει στο μέγιστο βάθος και επιστρέφει την τιμή αξιολόγησης. Για τον Pacman (maximizer), η συνάρτηση ενημερώνει την τιμή maxValue με το μέγιστο μεταξύ της τρέχουσας τιμής maxValue και της τιμής που επιστρέφει η expectimax. Για τα φαντάσματα (expectation node), η συνάρτηση υπολογίζει τη συνολική τιμή totalValue προσθέτοντας τις τιμές που επιστρέφει η expectimax για κάθε νόμιμη κίνηση και επιστρέφει τη μέση τιμή averageValue διαιρώντας τη συνολική τιμή με τον αριθμό των νόμιμων κινήσεων.

Q5: Evaluation Function

Αρχικά, υπολόγισα το τρέχον σκορ της κατάστασης. Στη συνέχεια, εξήγαγα τη θέση του Pacman, τις θέσεις των τροφών, τις καταστάσεις των φαντασμάτων και τις θέσεις των καψουλών. Υπολόγισα την απόσταση Manhattan από την πλησιέστερη τροφή και την πλησιέστερη κάψουλα. Για τα φαντάσματα, υπολόγισα την απόσταση από κάθε φάντασμα και πρόσθεσα ποινή αν κάποιο φάντασμα είναι πολύ κοντά και δεν είναι φοβισμένο. Αν το φάντασμα είναι φοβισμένο, πρόσθεσα θετική τιμή ανάλογα με την απόσταση. Συνδύασα αυτούς τους παράγοντες με το τρέχον σκορ για να υπολογίσω την τελική αξιολόγηση. Η τελική συνάρτηση αξιολόγησης λαμβάνει υπόψη την απόσταση από την πλησιέστερη τροφή, την απόσταση από τα φαντάσματα (φοβισμένα ή μη), την απόσταση από τις κάψουλες, τον αριθμό των υπολειπόμενων τροφών και καψουλών, και το τρέχον σκορ.