

(1) Όσον αφορά την τιμή την οποία η evaluation function επιστρέφει, λαμβάνονται υπόψιν οι θέσεις φαντασμάτων, capsule, foods, καθώς και άμα η κίνηση μου είναι stop. Αρχικά, άμα ένα φάντασμα είναι ενεργό, μπορεί δηλαδή να μας φάει, δεν θέλουμε αυτό να πλησιάσει πάρα πολύ κοντά μας, οπότε άμα η απόσταση μας από κάποιο τέτοιο φάντασμα γίνει μικρότερη του 3 αφαιρείται από την τελική τιμή του evaluation ένας αρκετά μεγάλος αριθμός ( μεγαλύτερος όσο η απόσταση ελαττώνεται). Επίσης δεν θέλουμε ο pacman να παραμένει σταματημένος καθώς και επιθυμούμε η θέση στη οποία βρισκόμαστε να περιέχει φαγητό άρα και η τιμή evaluation μειώνεται και αυξάνεται αντίστοιχα. Τέλος βρίσκω τη μικρότερη απόσταση από capsule, καθώς και αυτή από το πλησιέστερο φαγητό και προσθέτω για κάθε μια  $10/\text{mindistance}$  στην τελική τιμή του evaluation, όπου mindistance η ελάχιστη αυτή απόσταση. Με τον τρόπο αυτό υποδηλώνεται πως όσο πιο κοντά βρίσκεται το κοντινότερο φαγητό ή capsule τόσο το καλύτερο καθώς όσο ελαττώνεται ο παρονομαστής του κλάσματος  $10/\text{mindist}$  τόσο μεγαλύτερη θα είναι η ποσότητα που τελικά θα προστεθεί άρα και η τελική τιμή επιστροφής της evaluation function.

(2 – 3 – 4) Στα ερωτήματα αυτά υλοποιείται ο minimax αλγόριθμος ( alpha beta pruning και expectimax ουσιαστικά βασίζονται στον minimax με μερικές διαφοροποιήσεις). Η βασική διαφορά σε σχέση με την κλασσική υλοποίηση που παρουσιάζεται στις διφάνιες είναι πως min και max δεν παίζουν εναλλάξ μια και μια φορά διαδοχικά αλλά παίζει μια φορά ο pacman και μετά από μια φορά ένα ένα τα φάντασματα. Η αλληλουχία αυτή αποτελεί ουσιαστικά και μια μονάδα βάθους depth, δηλαδή το βάθος όταν έχουν παίξει όλοι οι min και ο max θα είναι 1. Βάσει λοιπόν της υλοποίησης αυτής έχουμε την παραδοχή οι παίκτες να παίζουν κυκλικά δηλαδή, πρώτα ο max μετά όλοι οι min έπειτα ο max κοκ. Για την επιλογή του άμα θα κληθεί min η max έχει χρησιμοποιηθεί η minimax συνάρτηση η οποία καλείται και από τον min και από τον max και αποφασίζει ανάλογα με το agent\_index με το οποίο κλήθηκε αλλά και δεδομένου πως κινούμαστε κυκλικά (με τη χρήση του mod ( $\text{id\_τρέχοντα\_πράκτορα} \% \text{αριθμός\_πρακτόρων}$ ) όπως είπαμε παραπάνω), άμα ο επόμενος παίκτης θα είναι min η max. Η evaluation function επιστρέφεται όταν έχουμε νίκη, ήττα η φτάσουμε ενα μέγιστο βάθος. Το alpha beta pruning βασίζεται στην ίδια λογική, μόνο που στον min και στον max πραγματοποιούνται συγκρίσεις των alpha, beta που προσθέτουμε ως ορίσματα, με την τιμή value ώστε να αποφασιστεί η όχι το κλάδεμα κάποιου υποδένδρου, ενώ στον expectimax ο min μετονομάζεται σε chance και η τιμή value προκύπτει ως μέσος όρος των τιμών των παιδιών του κόμβου η αλλιώς το άθροισμα των τιμών των παιδιών, με πολλαπλασιασμό η κάθε επιμέρους τιμής που αθροίζεται, με την σταθερή πιθανότητα  $1/\text{πλήθος\_παιδιών}$ , που έχουμε υπολογίσει παραπάνω.

(5) Για το σχεδιασμό της evaluation function αυτής έχει ληφθεί υπ' όψιν αρχικά πόσο κοντά βρισκόμαστε σε φαγητό η σε capsule ( αποστάσεις που επιθυμούμε να είναι όσο το δυνατόν μικρότερες). Το σκεπτικό και η υλοποίηση γι αυτά είναι ίδια με του πρώτου ερωτήματος και έχει αναλυθεί πιο πάνω. Το επιπλέον στοιχείο το οποίο έχω προσθέσει είναι πως εξετάζω τη δυνατότητα του pacman να φάει το φάντασμα στην περίπτωση που βρίσκεται κοντά σε capsule οπότε και η συμπεριφορά του είναι να καταναλώσει την capsule και έπειτα να φάει και το φάντασμα, ενέργεια επιθυμητή που αποδίδει πολλούς πόντους στο τελικό σκόρ. Επίσης είναι επιθυμητό το φάντασμα να είναι φοβισμένο, καθώς και δεν θέλουμε να βρισκόμαστε πάρα πολύ κοντά όταν αυτό είναι ενεργό και δεν έχουμε κοντά μας ( σε μικρότερη απόσταση από ότι βρίσκεται το φάντασμα) capsule, ώστε να την καταναλώσουμε και έπειτα να έχουμε την ευκαιρία να το καταστρέψουμε κερδίζοντας πόντους. Όλες οι παραπάνω ενέργειες ανάλογα με το άμα είναι θετικές οι αρνητικές όπως ορίζονται παραπάνω προσθέτουν η αφαιρούν από την τελική τιμή της evaluation function μια ποσότητα, ανάλογη του πόσο θετική η αρνητική είναι η κάθε περίπτωση.

Στην περίπτωση για παράδειγμα που θέλουμε να βρισκόμαστε κοντά σε φαί η capsule προσθέτουμε  $10/\text{mindist}$  ( όπου mindist η ελάχιστη απόσταση ) έτσι όσο μικρότερη είναι η απόσταση αυτή τόσο μειώνεται ο παρονομαστής του κλάσματος άρα και αυξάνεται η ποσότητα που προσθέτουμε στο αποτέλεσμα αποτυπώνοντας τη θετική συμπεριφορά του να βρισκόμαστε το δυνατότερο κοντά σε food η capsule που θέλουμε.