Όνομα: Δημήτριος
Επώνυμο: Φούντας
Α.Μ:1112201600236

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

(PROJECT 2)

ΕΡΩΤΗΣΗ1:

Στο πρώτο κομμάτι της εργασίας προσπαθούμε να φτίαξουμε μια συνάρτηση αξιολόγησης η οποία επικεντρώνεται στο να μπορεί ο pacman να αντιδρά σε μελλοντικές καταστάσεις. Προφανώς προσπαθούμε να χρησιμοποιήσουμε όσο το δυνατόν περισσότερες πληροφορίες απο αυτές που μας δίνονται από την κλάση Gamestate. Για αρχή υπολογίζουμε όλες τις αποστάσεις που έχει η θέση μας από κάθε ένα μήλο που προσπαθούμε να κάνουμε τον pacman να φάει, και υπολογίζουμε την μικρότερη απόσταση. Αυτή η απόσταση όσο μεγαλύτερη τιμή έχει , τόσο πολυ θα μεγαλώσει και η τιμή χρησιμότητας. Στην συνέχεια υπολογίζουμε τις αποστάσεις που έχει ο pacman απο κάθε ένα φάντασμα, και βρίσκουμε την απόσταση από το πιο απειλητικό εξ αυτών , δηλαδή αυτό με την μικρότερη απόσταση.Αυτή η απόσταση όσο μεγαλύτερη είναι τόσο το καλύτερο για μας , οπότε θα δώσουμε καλύτερη τιμή χρησιμότητας στις θέσεις με την μεγαλύτερη απόσταση. Ύψιστης σημασίας για να δουλέψει αποτελεσματικά το πρόγραμμα μας είναι όσο ελέγχουμε τις αποστάσεις που έχει ο pacman με τα φαντάσματα , να δώσουμε πολύ μικρή τιμή στις θέσεις στις οποίες τα φαντάσματα πέφτουν πάνω μας και μας κάνουν να χάσουμε το παιχνίδι.

Είναι λογικό στην συνέχεια να ζητάμε απο τον pacman να μην σταματάει ποτέ να κινείται, καθώς οταν μένει ακίνητος περνάει ο χρόνος περνάει και το σκόρ μας γίνεται όλο και μικρότερο. Αυτό όμως δεν είναι αρκετό, καθώς επειδή δίνουμε πολύ μεγάλες τιμές χρησιμότητας στις θέσεις που είναι δίπλα στην κοντινότερη τροφή, ο pacman κινδυνεύει να παγιδευτεί σε αυτές και να μην πάει ποτέ να φάει το μήλο που είναι ακριβώς δίπλα του. Οπότε του ζητάμε, κάθε φορά που μπορεί να βρεθεί σε μια κατάσταση η οποία έχει περισσότερα μήλα από την προηγούμενη, να κάνει αυτή την κίνηση, εκτός αν αυτή θα τον κάνει να χάσει το παιχνίδι.Σε αυτή την περίπτωση προτεραιότητα παίρνει να αποφύγουμε το φάντασμα.Τέλος καλό είναι να προσθέτουμε στην τιμή χρησιμότητας το σκορ που θα έχει ο pacman κάνοντας την συγκεκριμένη κίνηση. Η επιλογή μας να το προσθέσουμε ως παράμετρο θα κάνει τον pacman να παίρνει κάθε φορά απόφασεις που θα μεγιστοποιούν το σκορ του ,όπως είναι το να φάει το σφαιρίδιο ή να φάει το φάντασμα ή να τελειώσει το παιχνίδι όσο πιο σύντομα μπορεί.

Θα παρατηρήσετε ότι έχω πολλαπλασιάσει με μεγάλο βάρος την ελάχιστη απόσταση από το φάντασμα. Αυτό συμβαίνει γιατί αυτή η απόσταση τείνει να είναι πολύ μεγαλύτερη από την απόσταση απο την κοντινότερη τροφή .Οπότε κινδυνεύουμε ο pacman να είναι τόσο τρομοκρατημένος από το φάντασμα που να περιορίζεται για πολύ ώρα σε σημεία του χάρτη που δεν έχουν τροφή , χαλώντας μας έτσι πολίτιμο χρόνο.

Στην δεύτερη ερώτηση μας δίνεται η δοκιμασία του να εφαρμόσουμε τον αλγόριθμο minimax στις ανάγκες του συγκεκριμένου προβλήματος. Αρχικά ξεκινάμε φτίαχνοντας μια συνάρτηση minimax η οποία έχει ως μοναδικό σκοπό να υπολογίζει αν είναι η σειρά του pacman να παίξει ή αν είναι η σειρά κάποιου φαντάσματος. Η συνάρτηση αυτή παίρνει ως είσοδο την κατάσταση του παιχνιδιού, εναν συντελεστή agent που μας βοηθάει να καταλαβαίνουμε ποιανού είναι η σειρά να παίξει και το βάθος που έχουμε εξερευνήσει στο δέντρο μέχρι αυτή την στιγμή. Αν ο συντελεστής agent είναι 0 τότε η συνάρτηση minimax θα θεωρήσει ότι είναι η σειρά του pacman να παίξει και θα στειλει την ροή του προγραμματος στην συνάρτηση max_value.Αντίθετα στην περίπτωση που ο συντελεστής agent είναι διαφορετικός του μηδενός και μικρότερος από τον αριθμό των φαντασματων +1, τότε γνωρίζουμε πως είναι η σειρά του φαντάσματος με σειρά agent να παίξει. Κάθε φορά που ολοκληρώνεται ένας κύκλος και έχει παίξει την σειρά του ο pacman και όλα τα φαντάσματα, τότε ο συντελεστής agent θα έχει φτάσει να γίνει ίσος με τον αριθμό των πρακτόρων. Σε αυτή την περίπτωση η συνάρτηση value θα μηδενίσει το συντελεστή agent με σκοπό να ξεκινήσει τον κύκλο από την αρχή, και ταυτόχρονα θα ενημερώσει τον αλγόριθμο ότι έχουμε προχωρήσει ένα επίπεδο πιο βαθεία στο δέντρο.

Συνεχίζουμε δημιουργώντας την συνάρτηση max_value η οποία δέχετε τις παραμέτρους gameState και depth (η οποία υπολογίζει το βάθος του δέντρου που έχουμε εξερευνήσει) και επιστρέφει είτε την τιμή χρησιμότητας ενός φύλλου είτε την μεγαλύτερη τιμή χρησιμότητας που μπορεί να επιστρέψει ο αλγόριθμος μας , μαζί με την δράση που θα μας οδηγήσει σε αυτό το αποτέλεσμα. Αρχικά αυτή η συνάρτηση ελέγχει αν έχουμε φτάσει στα φύλλα του δέντρου ή αν δεν υπάρχει πλέον καμία κίνηση που επιτρέπεται να κάνει ο pacman ,δηλαδή αν έχει παγιδευτεί, και σε αυτή την περίπτωση επιστρέφει την τιμή χρησιμότητας της συγκεκριμένης θέσης . Στην συνέχεια εφαρμόζουμε τον αλγόριθμο όπως μας δίνεται στις σημειώσεις μας, με την διαφορά πως τώρα αντί να επιστρέφουμε απλά την τελική τιμή του αλγορίθμου. Τωρα επιστρέφουμε μια λίστα που περιέχει την τελική τιμή χρησιμότητας του αλγορίθμου καθώς και το μονοπάτι κινήσεων που μας οδήγησε σε αυτή . Για κάθε επιτρεπόμενη κίνηση του pacman βρίσκουμε τον κόμβο που δημιουργείται από την συγκεκριμένη κίνηση και καλούμε πάλι τον αλγόριθμο minimax με σκοπό μετά από αρκετές αναδρομικές κλήσεις να βρώ την βέλτιστη τιμή χρησιμότητας για τον αλγόριθμο.

Ακριβώς την αντίθετη λειτουργία επιτελεί ο αλγόριθμος min_value. Ο αλγόριθμος max_value, στους κόμβους πάνω στους οποίους δρα προσπαθεί να επιλέξει το παιδί του του που έχει αποτιμηθεί με την μεγαλύτερη τιμή ,ενώ ο min_value στους κόμβους πάνω στους οποίους καλείτε προσπαθει να τους αποτιμήσει με την μικρότερη δυνατή τιμή που μπορεί να βρει στα αποτιμημένα παιδια τους.
Στην ερώτηση 3 απλά γενικεύσαμε τον αλγόριθμο όπως μας ζητήθηκε στην εκφώνηση του project ώστε να αποφύγουμε περιττούς υπολογισμούς, δημιουργήσαμε θέσεις σε όλες τις συναρτήσεις για τις μεταβλητές a,b οι οποιες μας βοηθήσαν να κάνουμε κλάδεμα τον κόμβων που δεν χρειάζεται να υπολογίσουμε την τιμή τους για να πάρουμε το βέλτιστο μονοπάτι και την βέλτιστη τιμή που μπορούμε να πάρουμε.

Στην ερώτηση 4 η υλοποίηση του expectimax ήταν εξαιρετικά ενδιαφέρουσα. Η υλοποίηση διέφερε ελάχιστα από την υλοποίηση του minimax . Η βασική αλλαγή είναι πως τα φαντάσματα πλέον δεν κινούνται με βέλτιστη στρατηγική αλλά κινούνται τυχαία στον χάρτη.Αυτό σημαίνει πως ενώ η συνάρτηση max_value θα μείνει ως έχει. Η συνάρτηση min_value δεν έχει θέσει στον αλγόριθμο μας πλέον. Την συνάρτηση αυτή θα αντικαταστήσει η συνάρτηση exp_value. Αυτή η συνάρτηση στους κόμβους του δέντρου πάνω στους οποίους καλείτε κάθε φορά θα τους αποτιμάει με την μέση τιμή των τιμών των παιδιών του. Οπότε προκειμένου να τον υλοποιήσουμε στο συγκεκριμένο πρόβλημα που η μέση τιμή ταυτίζεται με τον μέσο όρο, υπολογίζουμε την τιμή του κάθε παιδιού ,τις προσθέτουμε και τέλος διαιρούμε με το πλήθος των παιδιών για να βγάλουμε τον μέσο όρο των τιμών των παιδιών του συγκεκριμενου κόμβου.

Τέλος χρησιμοποιώντας την κλάση currentGameState προσπαθήσαμε να φτιάξουμε μια ικανοποιητική συναρτηση χρησιμότητας για το πρόβλημα μας. Είναι παραπάνω από λογικό να ξεκινήσουμε ελέγχοντας αν η συγκεκριμένη κατάσταση που βρισκόμαστε ειναι κατασταση νίκης ,ώστε να της δώσουμε την μέγιστη τιμή και αντίθετα να ελέγχξουμε αν είναι κατάσταση ήττας ώστε να της δώσουμε την ελάχιστη τιμή έτσι ώστε ο pacman να αποφεύγει με κάθε κόστος τα φαντάσματα. Ακριβώς όπως και στην προηγούμενη συνάρτηση χρησιμότητας θα υπολογίσουμε την μικρότερη δυνατή απόσταση του pacman από την κοντινότερη τροφή. Αυτή η απόσταση όσο πιο μεγάλη τόσο πιο κακή είναι και η θέση που βρισκόμαστε. Για τον παραπάνω λόγο θα πολλαπλασιαστεί με αρνητικό βάρος όταν την χρησιμοποιήσουμε για να πάρουμε τιμή χρησιμότητας της θέσης που βρισκόμαστε. Όμοια θα υπολογίσω και την μικρότερη απόσταση (και μετά την μεγαλύτερη αν υπάρχουν στο παιχνίδι δύο σφαιρίδια) από το πιο κοντινότερο στην θέση μας σφαιρίδιο και αντίστοιχα θα το πολλαπλασιασω με αρνητικό βάρος.Τέλος υπολογίζω την ελάχιστη απόσταση του pacman από τα φαντάσματα. Γνωρίζοντας πως όσο μεγαλύτερη είναι η απόσταση αυτή ,τόσο καλύτερη είναι και η θέση του pacman καθώς δεν απειλείτε από φαντάσματα όσο αυτη είναι μεγάλη, οπότε θα την προσθέσουμε με θετικό βάρος. Τελειώνοντας εξαιρετικά μεγάλη πρόοδο στην βελτίωση της συναρτησης χρησιμότητας δημιουργείτε όταν επιλέγεις να εντάξεις στον γραμμικό συνδιασμό της τιμής χρησιμότητας την μεταβλητή του σκορ που έχεις σε εκείνη την κατάσταση τον συγκεκριμένο χρόνο καθώς η επιλογή μας αυτή θα κάνει τον pacman να παίρνει κάθε φορά απόφασεις που θα μεγιστοποιούν το σκορ του ,όπως είναι το να φάει το σφαιρίδιο ή να φάει το φάντασμα ή να τελειώσει το παιχνίδι όσο πιο σύντομα μπορεί.