Ονοματεπώνυμο: Ορφανίδης Ελευθέριος
Α.Μ.: 1115201400133

Pacman Project 1: Εχω υλοποιήσει όλα τα ερωτήματα του pacman project με τον τρόπο που τα ήθελε ο autograder ο οποίος μου δίνει βαθμολόγηση 26/25.

Ερωτήματα 1-4: Στα ερωτήματα αυτά χρησιμοποίησα κυρίως τις διαφάνειες με μόνη διαφορά το πότε γίνεται ο έλεγχος αν ένα στοιχείο είναι κόμβος- στόχος (πριν εισαχθεί ή μόλις γίνει pop()). Τον έλεγχο τον πραγματοποιώ μόλις ένα στοιχείο γίνει pop() από το fringe , όπως θέλει και ο autograder, και όχι πριν τον εισάγω στο fringe όπως προτείνεται στις διαφάνειες.

Ερώτημα 5: Στο CornersProblem θέτω ως αρχικό state το ((x,y) , ()) όπου x,y οι συντεταγμένες του Pacman(το StartingPosition). Ως goal state θέτω την κατάσταση της οποίας το state[1] (η λίστα που στην αρχική κατάσταση είναι κενή) περιέχει και τις 4 γωνίες του προβλήματος.

Ερώτημα 6: Στο cornerHeuristic για κάθε σημείο για το οποίο καλείται η ευριστική υπολογίζω αρχικά ποιες γωνίες δεν έχουν επισκεφθεί ακόμα. Στην συνέχεια ελέγχω ποια είναι η πιο μακρινή γωνία από το τωρινό position (χρησιμοποιώντας την manhattan distance) και επιστρέφω αυτή την τιμή. Έτσι ο A* επιλέγει σε κάθε περίπτωση τον κόμβο με την μικρότερη απόσταση από την πιο μακρινή γωνία. Έχω μέσα σε σχόλια και άλλη μια έκδοση της CornerHeuristic η οποία υπολογίζει το κόστος μέχρι την κοντινότερη γωνία και μετά, θέτοντας αυτή την γωνία ως starting position, το κόστος μέχρι να επισκεφθούν όλες οι γωνίες διαλέγοντας κάθε φορα την κοντινότερη γωνία. Η δεύτερη ευριστική είναι πιο ακριβής καθώς κάνει expand λιγότερους κόμβους για να βρει την λύση , αλλά έχει μεγαλύτερη πολυπλοκότητα για αυτό και την άφησα σε σχόλια.

Ερώτημα 7: Στο FoodHeuristic υπολογίζω για κάθε τροφή που υπάρχει στο grid:
1)το κόστος από το position μέχρι την τροφή
2)το κόστος από την τροφή μέχρι τις όλες τις υπόλοιπες τροφές διαλέγοντας κάθε φορά την κοντινότερη τροφή
και στο τέλος διαλέγω ως score το μικρότερο άθροισμα των 1) και 2). Τα κόστη τα υπολογίζω με την manhattan distance.
Η ευριστική υπολογίζει την λύση στο trickySearch problem σε 4.2 δευτερόλεπτα κάνοντας expand 6761 κόμβους, με path cost 60.

Ερώτημα 8: Ως isGoal στο AnyFoodSearchProblem θέτω το state == foodpos, όπου foodpos η θέση της κοντινότερης τροφής.