



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΠΡΑΚΤΟΡΑΣ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΟΣ
ΣΕ ΕΙΚΟΝΙΚΟ ΠΕΡΙΒΑΛΛΟΝ**

Διπλωματική Εργασία

Ζαπάρας Πασχάλης

Επιβλέπουσα: Δασκαλοπούλου Ασπασία

Βόλος 2020



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΠΡΑΚΤΟΡΑΣ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΟΣ
ΣΕ ΕΙΚΟΝΙΚΟ ΠΕΡΙΒΑΛΛΟΝ**

Διπλωματική Εργασία

Ζαπάρας Πασχάλης

Επιβλέπουσα: Δασκαλοπούλου Ασπασία

Βόλος 2020



UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**PROBLEM SOLVING AGENTS
IN VIRTUAL ENVIRONMENTS**

Diploma Thesis

Zaparas Paschalis

Supervisor: Daskalopoulou Aspasia

Volos 2020

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων	Δασκαλοπούλου Ασπασία Επίκουρος καθηγήτρια, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας
Μέλος	Βασιλακόπουλος Μιχαήλ Αναπληρωτής καθηγητής, , Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας
Μέλος	Μπαργιώτας Δημήτριος Αναπληρωτής καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 5-10-2020

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτρια κ. Δασκαλοπούλου Ασπασία για την ευκαιρία που μου πρόσφερε να συνεργαστώ μαζί της στο συγκεκριμένο θέμα και την καθοδήγηση της κατά την εκπόνηση της παρούσας εργασίας. Επίσης θα ήθελα να εκφράσω την ευγνωμοσύνη μου για την εμπιστοσύνη που μου έδειξε εξ αρχής, για την ενθάρρυνση, τις υποδείξεις και τις πολύτιμες συμβουλές της καθώς και για την συνεχή συμπαράστασή της.

Εκτιμώ την υποστήριξη όλων των καθηγητών/τριών μου και τους ευχαριστώ για τον χρόνο και την ενέργεια που μου αφιέρωσαν, ώστε να με εφοδιάσουν με γνώσεις και δεξιότητες που θα είναι οδηγοί μου στην επαγγελματική μου πορεία.

Κλείνοντας, ευχαριστώ εγκάρδια την οικογένειά μου, που με στηρίζει και είναι δίπλα μου όχι μόνο κατά τη διάρκεια των σπουδών μου, αλλά και σε κάθε βήμα μου, εμπυχώνοντάς με ώστε να προσπαθώ για το καλύτερο.

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο Δηλών

(Υπογραφή)
Ζαπάρας Πασχάλης
Σεπτέμβριος 2020

ΠΕΡΙΛΗΨΗ

Στην σημερινή εποχή η τεχνολογία συνεχίζει να προχωρά και να εξελίσσεται με ραγδαίους ρυθμούς στους περισσότερους επιστημονικούς τομείς. Σε αυτό το περιβάλλον μια τεχνολογία η οποία έχει ανθίσει και εμπλέκεται και σε ποικίλες άλλες επιστήμες καθώς και σε πλήθος πτυχών στην καθημερινή μας ζωή είναι η τεχνητή νοημοσύνη. Τα συστήματα τεχνητής νοημοσύνης καλύπτουν ένα πολύ μεγάλο εύρος λειτουργιών και καθένα έχει τα δικά του χαρακτηριστικά και δυνατότητες, κάποια κοινά κάποια όχι. Στα πλαίσια αυτής της διπλωματικής εργασίας μελετήθηκαν τα βασικά της τεχνητής νοημοσύνης. Επιπλέον ολοκληρώθηκε ένα απλό, αλλά λειτουργικό και επεκτάσιμο εργαλείο δημιουργίας σχεδίου δράσης, βασισμένο στην PDDL. Το PDDL σημαίνει Planning Domain Definition language είναι μια γλώσσα σχεδιασμού για τεχνητή νοημοσύνη, εμπνευσμένη από την STRIPS και την ADL. Το εργαλείο αυτό λειτουργεί με μια υλοποίηση του ευρετικού αλγορίθμου αναζήτησης κατά πλάτος σε δέντρα αποφάσεων. Συμπληρωματικά δημιουργήθηκαν μερικά σενάρια σχεδιασμένα να αναδείξουν την προτεραιότητα των εντολών στη συγκεκριμένη υλοποίηση.

ABSTRACT

Today technology continues to advance and evolve with a rapid pace at most scientific areas. In this environment there is one technology which is constantly involved in many of these evolutions as well as in many aspects of our daily lives. This technology is artificial intelligence. The systems which employ artificial intelligence cover a wide variety of functions and every system has each one characteristics and capabilities, some shared some not. In this diploma thesis we studied the basics of artificial intelligence. Additionally, we completed in python a simple, but functional and expandable action planning PDDL based tool, which employs a breadth first search to solve a scenario. PDDL stands for Planning Domain Definition Language and is Artificial intelligence planning language, inspired by STRIPS and ADL. The tool can work on custom scenarios or on one of the pre-created scenarios, which were designed to highlight the difference on command priority based or the commands relative positions.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	xiii
ABSTRACT	xv
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	xvii
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	xix
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	xix
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	xix
ΚΕΦΑΛΑΙΟ 1	1
ΕΙΣΑΓΩΓΗ	1
Τεχνητή νοημοσύνη σήμερα	1
ΚΕΦΑΛΑΙΟ 2	3
ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	3
2.1 Κατηγοριοποιήσεις τεχνητής νοημοσύνης	3
2.2 Συχνά εμπλεκόμενοι όροι	4
2.3 Ανάλυση κατηγοριών υλοποίησης	5
2.4 Βασικές γνώσεις	6
ΚΕΦΑΛΑΙΟ 3	7
ΠΕΙΡΑΜΑ	7
3.1 Μια σύντομη περιγραφή	7
3.2 Δομή και επεξήγηση κώδικα	9
3.3 Σύμβολα στο πλαίσιο της παρούσας διπλωματικής	13
3.4 Παράμετροι του πειράματος.....	14
ΚΕΦΑΛΑΙΟ 4	17
ΣΥΜΠΕΡΑΣΜΑΤΑ	17
4.1 Σύνοψη.....	17
4.2 Συμπεράσματα	17

4.3 Μελλοντικές Επεκτάσεις	18
ΒΙΒΛΙΟΓΡΑΦΙΑ	19
ΠΑΡΑΡΤΗΜΑ	21
ΕΞΟΔΟΙ ΕΚΤΕΛΕΣΗΣ ΣΕΝΑΡΙΩΝ.....	21
Έξοδος Σεναρίου 1 μέσω PDDL.py	21
Έξοδος Σεναρίου 1 μέσω planner.py.....	23
Έξοδος Σεναρίου 1 μέσω launcher.py	24
Έξοδος Σεναρίου 2 μέσω launcher.py	25
Έξοδος Σεναρίου 3 μέσω launcher.py	26
Έξοδος Σεναρίου 4 μέσω launcher.py	27
Έξοδος Σεναρίου 5 μέσω launcher.py	29
Έξοδος Σεναρίου 6 μέσω launcher.py	30

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 3.1 Σχηματική απεικόνιση project structure.....	8
Σχήμα 3.2 Σχηματικό διάγραμμα του code structure	11
Σχήμα 3.3 Σχηματική απεικόνιση διαφορών των σεναρίων.....	16

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1 Αντιστοίχιση συμβόλων - εννοιών	13
---	----

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1 Στιγμιότυπο «Έξοδος Σεναρίου 1 μέσω PDDL.py» part1.....	21
Εικόνα 2 Στιγμιότυπο «Έξοδος Σεναρίου 1 μέσω PDDL.py» part2.....	22
Εικόνα 3 Στιγμιότυπο «Έξοδος Σεναρίου 1 μέσω planner.py»	23
Εικόνα 4 Στιγμιότυπο «Έξοδος Σεναρίου 1 μέσω launcher.py».....	24
Εικόνα 5 Στιγμιότυπο «Έξοδος Σεναρίου 2 μέσω launcher.py».....	25
Εικόνα 6 Στιγμιότυπο «Έξοδος Σεναρίου 3 μέσω launcher.py».....	26
Εικόνα 7 Στιγμιότυπο «Έξοδος Σεναρίου 4 μέσω launcher.py» part1.....	27
Εικόνα 8 Στιγμιότυπο «Έξοδος Σεναρίου 4 μέσω launcher.py» part2.....	28
Εικόνα 9 Στιγμιότυπο «Έξοδος Σεναρίου 5 μέσω launcher.py».....	29
Εικόνα 10 Στιγμιότυπο «Έξοδος Σεναρίου 6 μέσω launcher.py»	30

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Τεχνητή νοημοσύνη σήμερα

Η ζωή μας στην σημερινή εποχή είναι γεμάτη από ευκολίες, εργαλεία και αυτοματισμούς, που επιτεύχθηκαν μέσω τεχνολογιών που προέκυψαν από την ραγδαία ανάπτυξη των τελευταίων δεκαετιών σε πολλούς επιστημονικούς τομείς. Ένας τέτοιος επιστημονικός τομέας είναι αυτός της τεχνητής νοημοσύνης. Αν και η συνεισφορά της ίσως να μην είναι πάντα εμφανής με μια αρχική ματιά, η τεχνητή νοημοσύνη έχει πληθώρα εφαρμογών σε πολλούς τομείς, όπως στην οικονομία [1], στην υγεία [2], στην αντιμετώπιση κρίσεων [3], σε αυτοματισμούς [4], [5]. Η τεχνητή νοημοσύνη επηρεάζει και εμπεριέχεται σε πολλές ποικίλες πτυχές της ζωής μας. Πάντα συνδυαζόμενη με άλλες επιστήμες και μεθόδους, η τεχνητή νοημοσύνη παίζει δυνητικά κρίσιμο ρόλο στην επίτευξη τεχνολογιών όπως των ψηφιακών βοηθών και αυτο-οδηγούμενων οχημάτων καθώς και στην περαιτέρω πρόοδο επιστημών [6].

Ωστόσο, ο ορισμός της τεχνητής νοημοσύνης είναι σε αρκετές περιπτώσεις μια ενδιαφέρουσα πρόκληση. Ο ορισμός της τεχνητής νοημοσύνης είναι δύσκολο να οριστεί από την φύση του, αλλά είναι κάτι το οποίο κεντρίζει το επιστημονικό ενδιαφέρον πλέον εδώ και τουλάχιστον 70 χρόνια από το 1950 με τον Turing [7]. Ένας από τους πρώτους τρόπους για τον έλεγχο ύπαρξης τεχνητής νοημοσύνης είναι ο έλεγχος του Turing. Επίσης ένας από τους πρώτους ορισμούς που διατυπώθηκαν από τους Barr και Feigenbaum αναφέρει ότι, «ΤΝ είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά» [8].

Ο ορισμός της τεχνητής νοημοσύνης έχει εξελιχτεί πολλές φορές από τότε. Σήμερα οι περισσότεροι ορισμοί κατατάσσονται σε μια από τις εξής κατηγορίες [7]:

- ✓ συστήματα που μιμούνται την ανθρώπινη συμπεριφορά,
- ✓ συστήματα που σκέφτονται όπως οι άνθρωποι,
- ✓ συστήματα που σκέφτονται λογικά και

✓ συστήματα που πράττουν λογικά.

Σχετικά με το θέμα αυτό έχει μέχρι σήμερα διεξαχθεί μεγάλος πλήθος ερευνών και κατά συνέπεια έχει δημιουργηθεί ήδη πολύ πλούσια βιβλιογραφία. Υπάρχει πλέον τεράστιο εύρος ορισμών [9], [10], [11], ωστόσο υπάρχουν ακόμα περιπτώσεις όπου τα όρια είναι δυσδιάκριτα και κάποιοι ορισμοί πιθανόν αντιφατικοί.

Ένας ακόμα παράγοντας που συνεισφέρει στη δυσκολία εύρεσης ενός οικουμενικού ορισμού για την τεχνητή νοημοσύνη είναι το μεγάλο εύρος το προβλημάτων που καλείται να αντιμετωπίσει και το πλήθος συνθηκών στις οποίες καλείται να λειτουργήσει. Όσον αφορά τα προβλήματα, στα οποία εφαρμόζεται η τεχνητή νοημοσύνη, αποτελούνται από διάφορους συνδυασμούς, που απαρτίζονται από ποικίλες φάσεις. Η ποικιλομορφία αυτή των προβλημάτων σημαίνει πως υπάρχει ένα ευρύ φάσμα στόχων.

Λόγω της ποικιλίας των στόχων υπάρχει και μεγάλη ποικιλία θεωρητικών και πρακτικών μεθόδων υλοποίησής της τεχνητής νοημοσύνης προς την επίτευξη των στόχων αυτών. Οι διάφορες μέθοδοι μπορούν και συχνά αξιοποιούνται για την εκτέλεση διαφόρων λειτουργιών. Οι λειτουργίες αυτές συμπεριλαμβάνουν, αλλά δεν περιορίζονται μόνο σ' αυτές, τις εξής: image recognition, disease diagnosis [12], DDOS filtering, chip placement for heat distribution, material sciences [13], robotics, νανο-τεχνολογία [14], στην δημιουργία ενεργειακά αποδοτικών κτηρίων [15].

Επίσης η δυσχέρεια στη δημιουργία ενός οικουμενικού ορισμού ενισχύεται από το εκτενές φάσμα μεθόδων με τις οποίες μπορεί να εφαρμοστεί η τεχνητή νοημοσύνη. Η τεχνητή νοημοσύνη μεταξύ άλλων βασίζεται κατά κόρον σε αλγορίθμους ταξινόμησης - Classification Algorithms, επαναληπτικούς αλγορίθμους - Regression Algorithms και αλγόριθμους ομαδοποίησης - Clustering Algorithms.

Συνοψίζοντας, η τεχνητή νοημοσύνη είναι ένας επιστημονικός κλάδος, με μεγάλο πλάτος και βάθος, βαθιά ενσωματωμένος στην καθημερινότητα μας και με πλήθος εφαρμογών, θεωρητικών και υπαρχόντων, σε πολλούς και διαφορετικούς άλλους επιστημονικούς κλάδους.

ΚΕΦΑΛΑΙΟ 2

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Κατηγοριοποιήσεις τεχνητής νοημοσύνης

Η τεχνητή νοημοσύνη παίρνει ποικίλες μορφές συχνά ανάλογα με τις λειτουργίες που καλείται να επιτελέσει. Παρόλα αυτά όλες οι μορφές της τεχνητής νοημοσύνης σε έναν αφαιρετικό βαθμό έχουν κάποια κοινά χαρακτηριστικά στην μορφή τους. Όλες οι εφαρμογές της τεχνητής νοημοσύνης έχουν έναν τουλάχιστον πράκτορα, ο οποίος χρησιμοποιεί μια βάση γνώσης και έναν αλγόριθμο για να παράγει ένα αποτέλεσμα στην μορφή σχεδίου ή δράσης. Η παραπάνω διαδικασία μπορεί να συμβαίνει μια φορά ή να τρέχει σε έναν κύκλο όπου το αποτέλεσμα του πράκτορα επηρεάζει το περιβάλλον, το οποίο επιδρά στη βάση του πράκτορα και κατά συνέπεια το επόμενο σχέδιο του είτε μέχρι να ικανοποιηθεί κάποια συνθήκη είτε επαναλαμβάνεται περιοδικά για όσο συνεχίζεται η εκτέλεση.

Στην παραπάνω περιγραφή εμπίπτουν έως ένα βαθμό όλα τα μοντέλα, που εφαρμόζουν τεχνητή νοημοσύνη, και χωρίζονται σε πλήθος κατηγοριών με βάση ποικίλα κριτήρια. Αρχικά, με βάση το εύρος της νοημοσύνης του πράκτορα [16] :

1. Artificial narrow intelligence, έχει συγκεκριμένο εύρος δυνατοτήτων. Οι περισσότερες εφαρμογές της τεχνητής νοημοσύνης σήμερα εμπίπτουν σε αυτή την κατηγορία.
2. Artificial general intelligence, έχει εύρος δυνατοτήτων παρόμοιο με τον άνθρωπο. Αυτός ο τύπος τεχνητής νοημοσύνης δεν είναι βιώσιμος ακόμα, είναι ωστόσο πιθανόν να προκύψει στο σχετικά άμεσο μέλλον.
3. Artificial superintelligence, έχει μεγαλύτερο εύρος δυνατοτήτων από έναν άνθρωπο.

Επιπλέον υπάρχουν πολλοί αλγόριθμοι τεχνητής νοημοσύνης και κάθε ένας από αυτούς έχει ποικίλες παραλλαγές. Ωστόσο σε γενικές γραμμές όλοι οι αλγόριθμοι είναι δυνατόν να ταξινομηθούν σε μια από τις εξής τρεις κατηγορίες , με βάση τον αλγόριθμο υλοποίησης που χρησιμοποιούν :

1. Classification Algorithms – Αλγόριθμοι ταξινόμησης
2. Regression Algorithms – Αλγόριθμοι επαναληπτικοί

3. Clustering Algorithms – Αλγόριθμοι ομαδοποίησης

Επιπλέον οι περισσότεροι αλγόριθμοι γενικότερα μπορούν να υλοποιηθούν με τρόπο ο οποίος επιτρέπει έστω σε κάποιον βαθμό παράλληλη εκτέλεση. Κατά συνέπεια κάθε αλγόριθμος μπορεί να χαρακτηριστεί είτε ως σειριακός είτε ως παραλληλοποιήσιμος.

Ως προς το πλήθος των πρακτόρων, το σύστημα ορίζεται ως πολυπρακτορικό ή μη, όταν αντίστοιχα αποτελείται από πολλούς ή έναν πράκτορα.

Τέλος, υπάρχουν δύο ακόμα προσεγγίσεις στην τεχνητή νοημοσύνη, η χρήση νευρωνικών δικτύων και η χρήση μηχανικής μάθησης. Αυτές οι δύο προσεγγίσεις μπορεί να εφαρμοστούν μαζί ή μόνες τους ανάλογα με την περίπτωση και αναλύονται περισσότερο στο επόμενο υποκεφάλαιο.

Οι μέθοδοι εφαρμογής της τεχνητής νοημοσύνης μπορούν σε γενικές γραμμές ανάλογα με την προσέγγιση υλοποίησής τους να ταξινομηθούν σε κάποιες από τις προαναφερθείσες κατηγορίες. Φυσικά κάθε μέθοδος, ανεξαρτήτως σε ποια κατηγορία ανήκει, έχει τα δικά της πλεονεκτήματα, μειονεκτήματα, απαιτήσεις και ιδιαιτερότητες, που μπορεί να είναι κοινά με αυτές άλλων μεθόδων ιδίων κατηγοριών.

2.2 Συχνά εμπλεκόμενοι όροι

Ο όρος μηχανική μάθηση περιγράφει ένα σύστημα με δυνατότητα μάθησης που μεταβάλλεται διαρκώς προς το «καλύτερο». Η βασική παραδοχή είναι ότι η λειτουργία του συστήματος επιφέρει μεταβολές στη βάση γνώσης του [8]. Επιπλέον οι προσεγγίσεις μηχανικής μάθησης χωρίζονται σε δυο ακόμα βασικές κατηγορίες, με επιτήρηση – Supervised και χωρίς επιτήρηση – Unsupervised.

Σχετικά με τα νευρωνικά δίκτυα μια συνοπτική και αφαιρετική περιγραφή είναι πως «Τα νευρωνικά δίκτυα είναι μια ιδιαίτερη προσέγγιση στη δημιουργία συστημάτων με νοημοσύνη καθώς αποφεύγουν να αναπαραστήσουν ρητά τη γνώση και να υιοθετήσουν ειδικά σχεδιασμένους αλγόριθμους αναζήτησης. Αντίθετα, βασίζονται σε λογικά πρότυπα καθώς χρησιμοποιούν δομές και διαδικασίες που μιμούνται τις αντίστοιχες του ανθρώπινου εγκεφάλου.» [8]

2.3 Ανάλυση κατηγοριών υλοποίησης

Κάθε κατηγορία και προσέγγιση τεχνητής νοημοσύνης έχει, όπως είναι φυσικό, πλεονεκτήματα καθώς και μειονεκτήματα.

Όσον αφορά την μηχανική μάθηση υπάρχουν αρκετοί απλοί αλγόριθμοι μηχανικής μάθησης οι οποίοι μπορούν να προσαρμοστούν και να ρυθμιστούν σχετικά γρήγορα ώστε να παράγουν ικανοποιητικά αποτελέσματα για προβλήματα γραμμικά διαχωρίσιμα. Ωστόσο λειτουργούν μόνο για γραμμικά διαχωρίσιμα προβλήματα και εφόσον ρυθμιστούν είναι δύσκολο να επικυρωθεί το αποτέλεσμα τους [17] .

Συγκεκριμένα σχετικά με τα νευρωνικά δίκτυα [18] η ικανότητα τους να αποδίδουν καλύτερα από τις παραδοσιακές προσεγγίσεις, όταν το πρόβλημα είναι αρκετά περίπλοκο (υπάρχουν δηλαδή πολλά δεδομένα προς επεξεργασία καθώς και πολλές μεταβλητές εισόδου), τα καθιστά ιδιαίτερα αποτελεσματικά σε προβλήματα όπου η λύση (ή και οι κανόνες που καθορίζουν ποιες λύσεις είναι βιώσιμες και ποιες όχι) δεν είναι γνωστή. Επιπλέον αυτό το χαρακτηριστικό τους σημαίνει πως είναι ικανά να διαχειριστούν μεγαλύτερο εύρος εισόδων από τις παραδοσιακές προσεγγίσεις. Παράλληλα, τα νευρωνικά δίκτυα εν μέρη και λόγω του τρόπου λειτουργίας τους μπορούν να επεξεργαστούν γρήγορα μεγάλο όγκο δεδομένων.

Βέβαια η φύση λειτουργίας των νευρωνικών δικτύων έχει ως αποτέλεσμα και αρκετές αρνητικές συνέπειες οι οποίες πιθανώς να τα καθιστούν ακατάλληλα για κάποιες εφαρμογές. Αρχικά η δημιουργία και η εκπαίδευση των νευρωνικών δικτύων προϋποθέτει την ύπαρξη μεγάλων όγκων δεδομένων σχετικά με το αντικείμενο. Επιπρόσθετα η λειτουργία τους γίνεται εσωτερικά από κανόνες που δημιουργήθηκαν κατά την εκπαίδευση τους, αλλά δεν είναι γνωστοί ωστόσο σε κάποιον εξωτερικό παρατηρητή. Αυτό βέβαια δυσχεραίνει την διόρθωση τους καθώς δεν γνωρίζει κανείς γιατί το σύστημα έλαβε μια απόφαση. Επιπλέον το γεγονός πως χρειάζεται να διαχειρίζονται μεγάλο όγκο δεδομένων σημαίνει πως είναι υπολογιστικά απαιτητικές μέθοδοι, και περαιτέρω κατά την φάση της εκπαίδευσης συνήθως είναι ακόμα πιο δύσκολη η κατανομή του υπολογιστικού κόστους καθιστώντας την διαδικασία εκπαίδευσης αργή.

Σε σχέση με την λήψη αποφάσεων και την δημιουργία σχεδίου οι ευρετικοί αλγόριθμοι συγκριτικά με τις προαναφερθείσες προσεγγίσεις επιδεικνύουν πιο προβλέψιμες

συμπεριφορές καθώς λειτουργούν με γνωστούς και προκαθορισμένους κανόνες είναι παρατηρήσιμοι και κατά συνέπεια πιο εύκολοι στην διατήρηση και διόρθωση τους. Επιπλέον στα αντικείμενα όπου είναι δυνατό να εφαρμοστούν συνήθως είναι λιγότερο απαιτητικοί από άποψη υπολογιστικής ισχύος. Από την άλλη πλευρά η δημιουργία μιας κατάλληλης βάσης γνώσης απαιτεί και χρόνο, ανάλογα με το θέμα, καθώς και σωστή κατανόηση του προβλήματος και των σχετικών παραμέτρων που χρειάζεται να ληφθούν υπόψη κατά την διαδικασία δημιουργίας του σχεδίου δράσης.

2.4 Βασικές γνώσεις

Παρόλα αυτά ωστόσο, αναλύοντας όλες τις προηγούμενες προσεγγίσεις αφαιρετικά υπάρχει μια κοινή βασική γενική δομή την οποία σε κάποιο βαθμό ή μορφή μοιράζονται όλα τα συστήματα τα οποία εφαρμόζουν την τεχνητή νοημοσύνη. Η βασική δομή αποτελείται από έναν πράκτορα, ο οποίος έχει μια αντίληψη για το περιβάλλον του είτε άμεσα είτε μέσω μια βάσης γνώσης. Ο πράκτορας θα βρει είτε μέσω ενός προγραμματισμένου αλγορίθμου όπως της αναζήτησης κατά βάθος είτε μέσω κάποιου άλλου αλγορίθμου ένα σχέδιο δράσης είτε ένα άλλο κατάλληλο αποτέλεσμα.

Στο πλαίσιο αυτής της διπλωματικής εργασίας πράκτορα θεωρούμε μια οντότητα η οποία αντιλαμβάνεται το περιβάλλον της μέσω της εισόδου της, είτε μέσω αισθητήρων, είτε μέσω μιας βάσης γνώσης και σχεδιάζει ως έξοδο ένα σχέδιο δράσεων [8].

Η μέθοδος με την οποία πραγματοποιήθηκε ο σχεδιασμός είναι η αναζήτηση κατά πλάτος (Breadth First Search [BFS]), ένας ευρετικός αλγόριθμος ο οποίος λειτουργεί χρησιμοποιώντας μια βάση γνώσεων δομημένη με την Planning Domain Definition Language(PDDL). Ο πράκτορας λειτουργεί σε ένα ντετερμινιστικό περιβάλλον με τέλεια γνώση.

ΚΕΦΑΛΑΙΟ 3

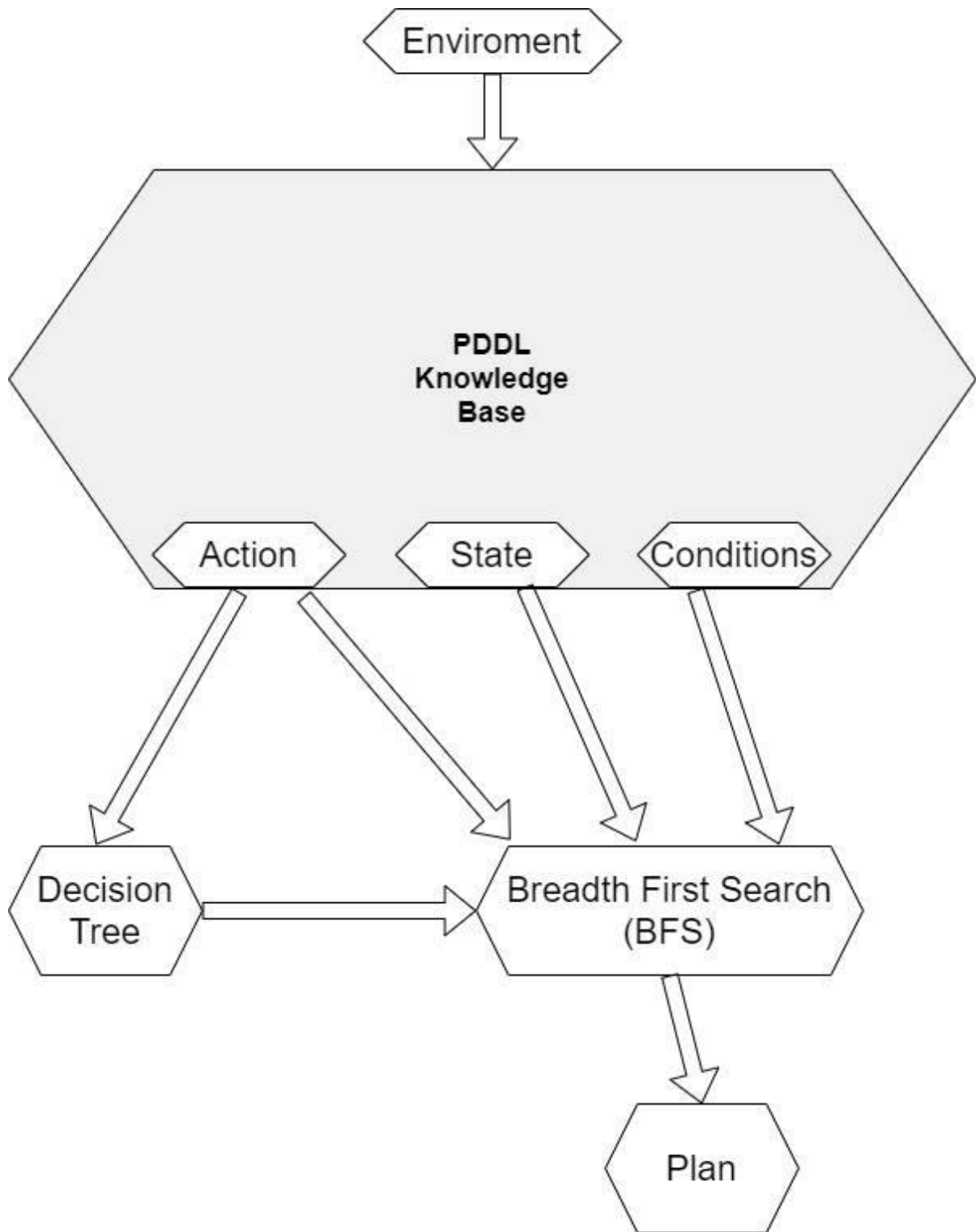
ΠΕΙΡΑΜΑ

3.1 Μια σύντομη περιγραφή

Για την παρούσα διπλωματική σχεδιάστηκε και υλοποιήθηκε σε ρυθμον μια αναζήτηση κατά πλάτος σε ένα δέντρο αποφάσεων υπό την υποστήριξη της Planning Domain Definition Language (PDDL) προς την δημιουργία ενός σχεδίου δράσης. Επίσης δημιουργήθηκαν τροποποιήσεις βασικών σεναρίων τα οποία είναι σχεδιασμένα για να αναδείξουν την διαφορά στη προτεραιότητα των εντολών υπό διαφορετικές αρχικές συνθήκες με σταθερό στόχο.

Η εφαρμογή των εκτελέσιμων της εργασίας αυτής σε ένα πρόβλημα ή σενάριο λειτουργεί όπως φαίνεται στο Σχήμα 3.1. Αρχικά πρέπει να εισαχθεί στο σύστημα με την μορφή PDDL η πληροφορία σχετικά με το περιβάλλον δηλαδή τα εμπόδια, τα εργαλεία, τις συνθήκες, τις αρχικές καταστάσεις, τις πράξεις με τις επιδράσεις τους, τις απαιτήσεις τους, και τους παράγοντες που αποτρέπουν την χρήση τους. Οι σχετικές πληροφορίες χωρίζονται σε δύο αρχεία το `domain.pddl` και το `problem.pddl`. Το πρώτο, το `domain.pddl`, εμπεριέχει πληροφορίες σχετικά με όλες τις διαθέσιμες δράσεις, με τις προαπαιτούμενες συνθήκες, καθώς και τις επιδράσεις των δράσεων, και τις αναγνωρίσιμες καταστάσεις, που χρησιμοποιούνται για την περιγραφή προαπαιτήσεων και επιδράσεων. Το δεύτερο αρχείο, το `problem.pddl`, εμπεριέχει τις πληροφορίες σχετικά με την αρχική κατάσταση και την επιθυμητή κατάσταση ή κατάσταση στόχο.

Στη συνέχεια εκτελείται η υλοποίηση της αναζήτησης κατά πλάτος και έμμεσα κατά την εκτέλεση δημιουργούνται τα απαραίτητα κομμάτια του δέντρου αποφάσεων όταν γίνονται απαραίτητα. Εάν ο συνδυασμός αρχικών συνθηκών, δράσεων και τελικών στόχων έχει τουλάχιστον ένα βιώσιμο σχέδιο, τότε επιστρέφεται το σχέδιο αυτό. Εναλλακτικά επιστρέφεται κατάλληλο μήνυμα σχετικά με την έλλειψη σχεδίου όταν εξερευνηθούν όλες οι πράξεις και καταλήξουν σε αδιέξοδο.



Σχήμα 3.1 Σχηματική απεικόνιση project structure

3.2 Δομή και επεξήγηση κώδικα

Ο κώδικας είναι σχεδιασμένος ώστε να μπορεί να εκτελεστεί με διαφορετικούς τρόπους. Τα εκτελέσιμα αρχεία για την παρούσα διπλωματική εργασία μπορούν να βρεθούν στον σύνδεσμο:

https://github.com/Zaparas/AI_planning-PDDL-DT-BFS

Η δομή του κώδικα και οι εξαρτήσεις μεταξύ των αρχείων φαίνεται στο Σχήμα 3.2. Υπάρχουν συνολικά 4 συνεργατικά αρχεία κώδικα `python` τα οποία είναι εκτελέσιμα.

Ο πιο εύκολος τρόπος να εκτελεστούν ωστόσο τα προκατασκευασμένα σενάρια είναι η χρήση του `launcher`. Η χρήση του είναι σχετικά απλή: κατά την εκκίνηση θα ζητηθεί ο αριθμός του σεναρίου που είναι προς εκτέλεση. Στην παρούσα υπάρχουν έξι (6) προκαθορισμένα σενάρια. Το πρόγραμμα επιπλέον αναγνωρίζει ως αποδεκτή είσοδο το μηδέν (0) στο οποίο στην συνέχεια θα επιτρέψει στο χρήστη να ορίσει δικά του αρχεία ως `domain.pddl` και `problem.pddl`, που στην συνέχεια θα επιχειρήσει να λύσει. Άρα συνολικά υπάρχουν επτά (7) αποδεκτές εισόδους, 1-6 για τις προκαθορισμένες εκτελέσεις και το μηδέν (0) για την εκτέλεση αρχείων επιλεγμένων από τον χρήστη. Σε όλες τις υπόλοιπες εισόδους το πρόγραμμα θα υποχωρήσει στην προκαθορισμένη `default` επιλογή και θα εκτελέσει το σενάριο (1).

Το `launcher.py` εκτελείται χωρίς ορίσματα, απλά με την χρήση της εντολής

```
python launcher.py
```

και είναι υπεύθυνο για την επιλογή των κατάλληλων αρχείων για την εκτέλεση. Ανεξαρτήτως της εισόδου το `launcher.py`, είτε μέσω των προκαθορισμένων αρχείων είτε μέσω των επιλεγμένων αρχείων του χρήστη, θα καλέσει το `planner.py` για να συνεχίσει την εκτέλεση.

Το `planner.py` χρησιμοποιείται έμμεσα μέσω του `launcher.py` αλλά επιπλέον είναι δυνατόν να χρησιμοποιηθεί κατευθείαν με μία εντολή της μορφής:

```
python planner.py Domain.pddl Problem.pddl
```

όπου το όρισμα 1, `Domain.pddl`, είναι το αρχείο που περιέχει τα δεδομένα για τον κόσμο και το όρισμα 2, `Problem.pddl`, είναι το αρχείο που περιέχει τις πληροφορίες για την αρχική κατάσταση και την κατάσταση στόχο. Το `planner.py` διαχειρίζεται έμμεσα την

μετατροπή των αρχείων σε διαχειρίσιμα δεδομένα, μέσω του PDDL.py και στην συνέχεια εκτελεί την αναζήτηση κατά πλάτος με σκοπό την εύρεση ενός σχεδίου δράσης. Σε περίπτωση που δεν υπάρχει συνδυασμός δράσεων, σχέδιο δράσης, που μας οδηγεί στην επιθυμητή κατάσταση το πρόγραμμα τυπώνει κατάλληλο μήνυμα στο τερματικό και τερματίζει. Το πρόγραμμα επίσης τερματίζει όταν βρει έστω και ένα βιώσιμο σχέδιο, βασικά το πρώτο που θα βρει.

Το PDDL.py είναι το πρόγραμμα το οποίο ανεξαρτήτως από τον τρόπο εκτέλεσης είναι υπεύθυνο για την μετατροπή των αρχείων .pddl σε εκμεταλλεύσιμα δεδομένα. Επιπρόσθετα είναι δυνατό να κληθεί ανεξάρτητα με χρήση της εντολής

```
python PDDL.py Target.pddl
```

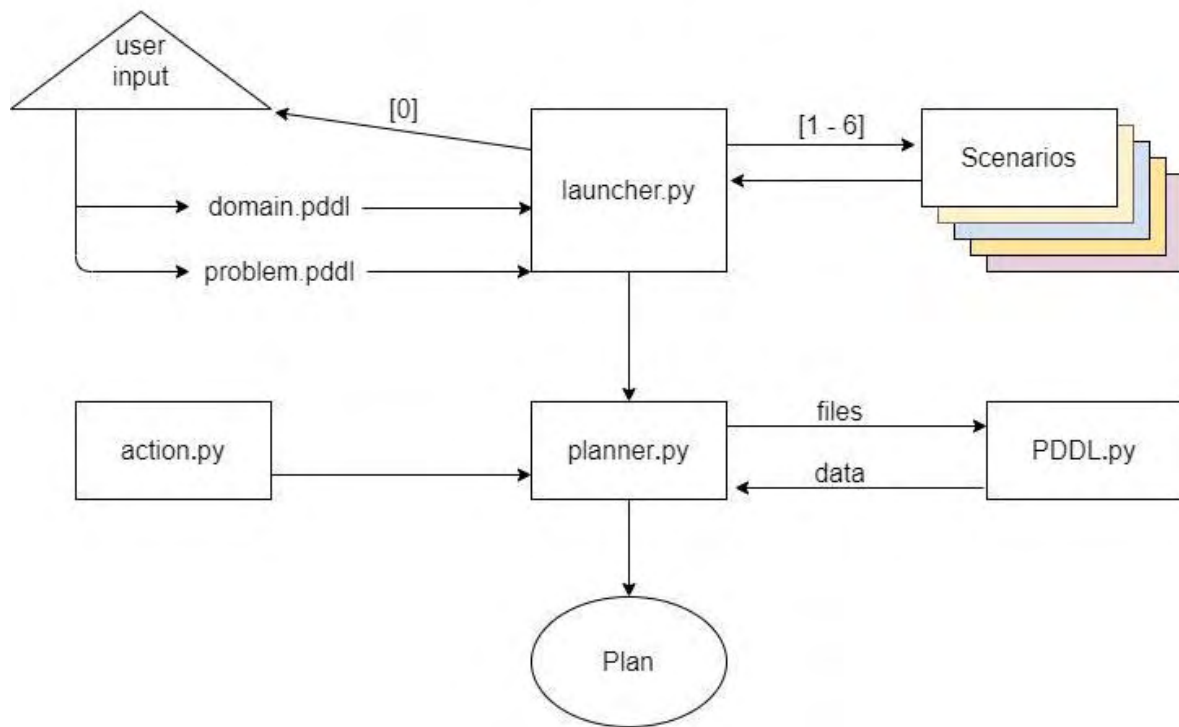
όπου το όρισμα 1, Target.pddl είναι το αρχείο το οποίο θα διαβαστεί και του οποίου τα δεδομένα θα τυπωθούν στο τερματικό. Το PDDL.py επίσης χρησιμοποιεί το action.py έμμεσα για να δημιουργήσει τις μεταβλητές όπου αποθηκεύει τα δεδομένα που διαβάζει.

Το action.py είναι το πρόγραμμα, το οποίο είναι υπεύθυνο για την δημιουργία των μεταβλητών που αποθηκεύουν τα δεδομένα των δράσεων και όλων των χρήσεων των καταστάσεων. Χρησιμοποιείται τόσο από το planner.py όσο και από το PDDL.py. Όταν το action.py καλείται προς εκτέλεση με την εντολή

```
python action.py
```

τότε απλά τυπώνει ένα σύνολο προκαθορισμένων δεδομένων με σκοπό να αναδείξει τις δυνατότητες που υποστηρίζει.

Code structure



1. start - get input
2. a. get selected scenario [1 - 6] files
b. get custom scenario files [0]
3. give data files to planner.py through launcher.py
4. convert data files to variables through PDDL.py
5. search using action.py functionality via BFS algorithm to plan generate

Σχήμα 3.2 Σηματικό διάγραμμα του code structure

Ο κώδικας αποτελείται από τέσσερα (4) αρχεία python και μπορούν να χρησιμοποιηθούν το καθένα για το δικό του σκοπό και με το δικό του τρόπο.

Σχετικά με το αρχείο PDDL.py, η πιο απλή και βασική εκτέλεση γίνεται με την χρήση μιας εντολή της μορφής:

```
python PDDL.py PathToTargetDomainFile.pddl PathToTargetProblemFile.pddl
```

όπου το PathToTargetFile.pddl το μονοπάτι με το αρχείο .pddl το οποίο θα διαβαστεί. Για παράδειγμα όπως υπάρχει καταγεγραμμένο στο README.md η εντολή

```
python PDDL.py scenarios\Scenario-1\KitchenDomain.pddl scenarios\Scenario-1\KitchenProblem.pddl
```

παράγει την έξοδο σύμφωνα με την «Έξοδο σεναρίου 1 μέσω PDDL.py» στο παράρτημα.

Σχετικά με το αρχείο `planner.py` η πιο άμεση εκτέλεση είναι δυνατή με την χρήση μιας εντολής της μορφής:

```
python planner.py PathToTargetDomainFile.pddl PathToTargetProblemFile.pddl
```

όπου το `PathToTargetFile.pddl` είναι το μονοπάτι με το αρχείο `.pddl` το οποίο θα διαβαστεί. Για παράδειγμα όπως υπάρχει καταγεγραμμένο στο `README.md`

```
python planner.py scenarios\Scenario-1\KitchenDomain.pddl scenarios\Scenario-1\KitchenProblem.pddl
```

η έξοδος της εκτέλεσης του σεναρίου 1 μέσω του `planner.py` υπάρχει στο « Έξοδος Σεναρίου 1 μέσω `planner.py`» του παραρτήματος.

Σχετικά με το αρχείο `launcher.py` η εκτέλεση είναι δυνατή με την χρήση της απλής εντολής της μορφής:

```
python launcher.py
```

και στην συνέχεια την επιλογή του επιθυμητού προς εκτέλεση σεναρίου. Οι έξοδοι για τα προκαθορισμένα σενάρια βρίσκονται στο παράρτημα, στους αντίστοιχους τίτλους του.

3.3 Σύμβολα στο πλαίσιο της παρούσας διπλωματικής

Στο πλαίσιο αυτής της διπλωματικής θα αναφερόμαστε σε κάθε πράκτορα-Operative/Agent i ως O_i άρα ο πράκτορας 1 θα αντιστοιχίζεται από το σύμβολο O_1 . Κάθε πράκτορας εκτελεί τον κώδικα που έχει διαθέσιμο προσπαθώντας να βρει την λύση στο δικό του σενάριο.

Σχετικά με τα σενάρια, κάθε σενάριο-Scenario i θα αναφέρεται ως S_i άρα το σενάριο 1 θα αντιστοιχίζεται από το σύμβολο S_1 . Κάθε σενάριο εμπεριέχει δράσεις και καταστάσεις. Οι δράσεις από εδώ και πέρα θα αναφέρονται ως A_i , άρα η δράση-Action 1 θα συμβολίζεται από το A_1 . Ομοίως οι καταστάσεις θα συμβολίζονται και με το C_i κατά συνέπεια η κατάσταση-Condition 1 θα συμβολίζεται με το C_1 . Οι καταστάσεις λειτουργούν ενίοτε ως επιδράσεις και άλλοτε ως απαιτήσεις. Όταν γίνεται αναφορά σε μια επίδραση-Effect i τότε θα αναφέρεται και ως E_i και όταν γίνεται σε μια απαίτηση-Requirement i τότε θα αναφέρεται και ως R_i . Τέλος, σκοπός κάθε πράκτορα O_i κάθε φορά που τρέχει κάποιο σενάριο S_i είναι η δημιουργία ενός σχεδίου δράσεων. Στα πλαίσια αυτής της εργασίας θα αναφέρονται τα σχέδια-Plan i ως P_i , άρα το σχέδιο 1 θα αναπαρίσταται από το P_1 .

Συνοπτικά η αντιστοιχία των διαφόρων στοιχείων με τα σύμβολα που τα αντιπροσωπεύουν φαίνεται στο Πίνακα 1 που ακολουθεί:

Πίνακας 1 Αντιστοίχιση συμβόλων - εννοιών

Στοιχείο-Element	Σύμβολο-Symbol
Σενάριο – Scenario	S_i
Πράκτορας – Agent / Operative	O_i
Δράση – Action	A_i
Επίδραση – Effect	E_i
Κατάσταση – Condition	C_i
Απαίτηση – Requirement	R_i
Σχέδιο – Plan	P_i

3.4 Παράμετροι του πειράματος

Για την εκπόνηση της παρούσας διπλωματικής δημιουργήθηκαν διαφορετικές εκδόσεις ενός κεντρικού σεναρίου. Η αναφορά στις εκδόσεις αυτές γίνεται με τον όρο «Σενάριο 1», όπου 1 ο αριθμός της έκδοσης στην οποία αναφερόμαστε. Στο Σχήμα 3.3 «Απεικόνιση διαφορών σεναρίων» απεικονίζονται συνοπτικά οι διαφορές μεταξύ των σεναρίων. Το domain του S_1 αναγνωρίζει πέντε (5) καταστάσεις: Clean, dinner, quiet, present, garbage. Επίσης προσφέρει στον πράκτορα τέσσερις (4) λειτουργίες – δράσεις: cook, wrap, carry, dolly. Σχετικά με τις δράσεις που μπορεί να λάβει ο πράκτορας:

1. Η δράση cook απαιτεί την κατάσταση clean και ως αποτέλεσμα προκύπτει η κατάσταση dinner.
2. Η δράση wrap απαιτεί την κατάσταση quiet και ως αποτέλεσμα προκύπτει η κατάσταση present.
3. Η δράση carry απαιτεί την κατάσταση garbage και ως αποτέλεσμα ακυρώνονται οι καταστάσεις garbage και clean.
4. Η δράση dolly απαιτεί την κατάσταση garbage και ως αποτέλεσμα ακυρώνονται οι καταστάσεις garbage και clean.

Από το S_1 στο S_2 προστίθεται μετά την δράση cook, η δράση order_delivery_food με την ίδια επίδραση, αλλά χωρίς προαπαιτούμενες καταστάσεις. Άρα είναι πιο ευέλικτη από την σχετική δράση cook. Παρατηρώντας τις σχετικές εξόδους, «Έξοδος Σεναρίου 1 μέσω launcher.py» και «Έξοδος Σεναρίου 2 μέσω launcher.py» στο παράρτημα, φαίνεται ότι δεν αλλάζει το σχέδιο δράσης λόγω της θέσης της καινούριας εντολής.

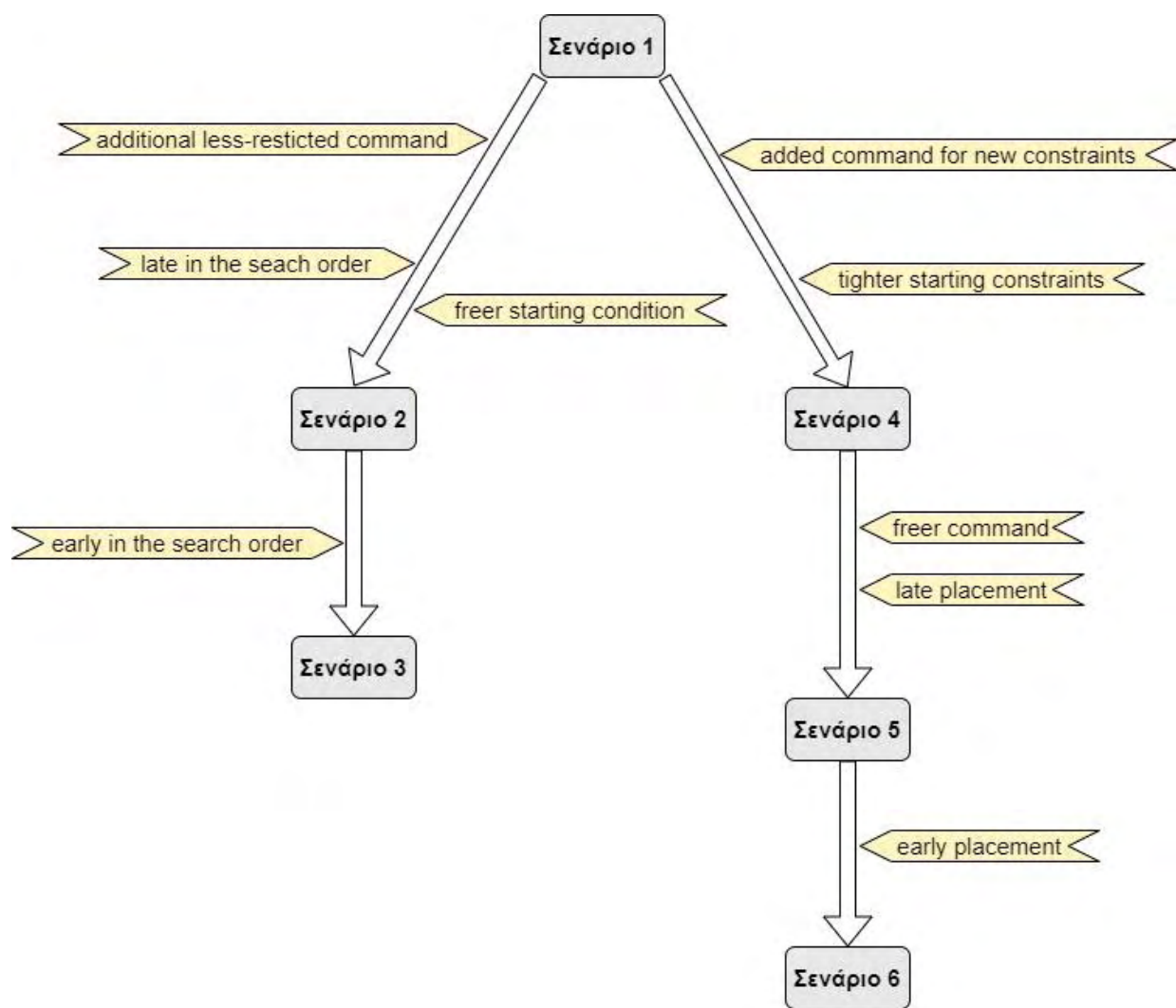
Από το S_2 στο S_3 μετατίθεται η δράση order_delivery_food νωρίτερα από την δράση cook. Μελετώντας τις σχετικές εξόδους, «Έξοδος Σεναρίου 2 μέσω launcher.py» και «Έξοδος Σεναρίου 3 μέσω launcher.py» στο παράρτημα, φαίνεται ότι αλλάζει το σχέδιο δράσης λόγω της νέας θέσης της δράσης order_delivery_food σε σχέση με την cook.

Από το S_1 στο S_4 αφαιρείται μια κατάσταση από τις αρχικές συνθήκες και προστίθεται η δράση με το όνομα santify, που προαπαιτεί την απουσία της καταστασης garbage και δίνει την κατάσταση clean . Συγκρίνοντας τις σχετικές εξόδους, «Έξοδος Σεναρίου 1 μέσω launcher.py» και «Έξοδος Σεναρίου 4 μέσω launcher.py» στο παράρτημα, φαίνεται ότι

αλλάζει το σχέδιο δράσης για να προσαρμοστεί στις καινούριες αρχικές συνθήκες ενσωματώνοντας την νέα δράση `santify` στο σχέδιο δράσης.

Από το S_4 στο S_5 προστίθεται η δράση `order_delivery_food`, όπως στο σεναρίου 2 μετά την δράση `cook`. Με βάση τις σχετικές εξόδους, «Έξοδος Σεναρίου 4 μέσω `launcher.py`» και «Έξοδος Σεναρίου 5 μέσω `launcher.py`» στο παράρτημα, το σχέδιο δράσης προσαρμόζεται και αξιοποιεί την δράση `order_delivery_food` παρόλο που βρίσκεται μετά την δράση `cook` ως μέρος του συντομότερου σχεδίου.

Από το S_5 στο S_6 μετατίθεται η δράση `order_delivery_food`, πριν την δράση `cook`. Παρατηρώντας τις σχετικές εξόδους, «Έξοδος Σεναρίου 5 μέσω `launcher.py`» και «Έξοδος Σεναρίου 6 μέσω `launcher.py`» στο παράρτημα, βλέπουμε ότι το σχέδιο δράσης δεν αλλάζει. Ο πρώτος λόγος είναι πως η δράση `cook` με τις προαπαιτούμενες συνθήκες της θα οδηγούσε σε μεγαλύτερο σχέδιο. Ο δεύτερος λόγος είναι η τοποθέτησή της μετά την δράση `order_delivery_food`.



Σχήμα 3.3 Σχηματική απεικόνιση διαφορών των σεναρίων

ΚΕΦΑΛΑΙΟ 4

ΣΥΜΠΕΡΑΣΜΑΤΑ

4.1 Σύνοψη

Στο πλαίσιο αυτής της διπλωματικής μελετήθηκαν τα βασικά της τεχνητής νοημοσύνης και δημιουργήθηκε ένα πρότυπο εργαλείο για την εκτέλεση αρχικώς ενός αλγορίθμου με μια βάση γνώσης. Εν πρώτοις πραγματοποιήθηκε μια γενική επισκόπηση της τεχνητής νοημοσύνης, σχετικά με τον ορισμό της, και άλλες βασικές πληροφορίες σχετικά με τις κατηγορίες και άλλους συχνά εμπλεκόμενους όρους (Κεφάλαιο 2). Επιπρόσθετα ολοκληρώθηκε μια υλοποίηση στη γλώσσα `python` ενός συστήματος το οποίο αξιοποιεί έναν ευρετικό αλγόριθμο, αναζήτησης κατά πλάτος, και της γλώσσας αναπαράστασης PDDL, μέσω ενός `parser`, για την εύρεση βασικών σχεδίων δράσεων σε διαμορφώσιμα καθώς και προκαθορισμένα σενάρια (Κεφάλαιο 3). Επίσης δημιουργήθηκαν μερικά προκαθορισμένα σενάρια τα όποια είναι διαθέσιμα προς εκτέλεση, μέσω των οποίων μελετήθηκε η επίδραση των διαφορετικών θέσεων εντολών στην διαμόρφωση ενός σχεδίου δράσης (Κεφάλαια 3 & 4).

4.2 Συμπεράσματα

Σχετικά με το θεωρητικό κομμάτι της διπλωματικής αυτής, το εύρος των μεθόδων υλοποίησης της τεχνητής νοημοσύνης είναι μεγάλο και προσφέρει μεγάλη ελευθερία. Υπάρχει πλήθος μεθόδων, διαφοροποιήσεων και συνδυασμών με νευρωνικά δίκτυα και μηχανική μάθηση που καλύπτουν όλα τα μήκη και βάρη λειτουργιών με διαφορετικά φυσικά βάρη στην υλοποίηση. Η χρήση των απλών εργαλείων – μεθόδων έχει τα δικά της πλεονεκτήματα και ιδιότητες. Η πρόκληση είναι πάντα να επιλεγεί η κατάλληλη μέθοδος για την κατάλληλη δουλειά. Σε πιο πολύπλοκα συστήματα η ιδανική λύση πιθανώς να εμπλέκει συνδυασμό διαφόρων μεθόδων για διάφορες λειτουργίες.

Τελικά με βάση τις παρατηρήσεις που προκύπτουν στα πλαίσια της διπλωματικής αυτής μέσω των σεναριών που αναφέρθηκαν στο κεφάλαιο 3 καταλήγουμε στο συμπέρασμα πως οι εντολές που βρίσκονται πιο πάνω στο αρχείο και κατά συνέπεια πιο αριστερά στο δέντρο αποφάσεων ελέγχονται πρώτες όταν η αναζήτηση γίνεται με βάση τον αλγόριθμο

αναζήτησης κατά πλάτος και άρα έχουν προτεραιότητα στην δημιουργία σχεδίων εάν υπάρχουν πολλαπλά σχέδια ίδιου μήκους.

4.3 Μελλοντικές Επεκτάσεις

Η υλοποίηση που πραγματοποιήθηκε ήταν από τις πιο απλές με στόχο την ελαχιστοποίηση των παραγόντων που επηρέαζαν τα αποτελέσματα και έχοντας υπόψη την βιωσιμότητα της εργασίας στο διαθέσιμο χρονικό περιθώριο, με τους διαθέσιμους πόρους. Υπάρχει ωστόσο περιθώριο για μελλοντικές επεκτάσεις με βάση την εργασία αυτή.

Πιθανές μελλοντικές επεκτάσεις συμπεριλαμβάνουν:

1. Δημιουργία γραφικού περιβάλλοντος διεπαφής με τον χρήστη
2. Δημιουργία εργαλείου δημιουργίας και διαχείρισης στοιχείων της βάσης γνώσης
3. Δημιουργία επιπρόσθετων σεναρίων
4. Επέκταση λειτουργικότητας του συστήματος με βεβαρυμένους αλγορίθμους
5. Επέκταση λειτουργικότητας του συστήματος με εναλλακτικούς αλγορίθμους

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] G. Rampersad, «Robot will take your job: Innovation for an era of artificial intelligence,» *Journal of Business Research*, τόμ. 116, pp. 68-74, August 2020.
- [2] L. BoYang, «Procedia Computer Science,» *Application of Artificial Intelligence in Electrical Automation Control*, τόμ. 166, pp. 292-295, 2020.
- [3] R. Vaishya, M. Javaid, I. H. Kha και A. Haleem, «Artificial Intelligence (AI) applications for COVID-19 pandemic,» *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, τόμ. 14, αρ. 4, pp. 337-339, Juli 2020.
- [4] G. Liang, W. Fan, H. Luo και X. Zhu, «Biomedicine & Pharmacotherapy,» *The emerging roles of artificial intelligence in cancer drug development and precision therapy*, τόμ. 128, August 2020.
- [5] D. Wiljer και Z. Hakim, «Developing an Artificial Intelligence–Enabled Health Care Practice: Rewiring Health Care Professions for Better Care,» *Journal of Medical Imaging and Radiation Sciences*, τόμ. 50, αρ. 4, pp. S8-S14, December 2019.
- [6] J. N. Kok, E. J. W. Boers, W. A. Kusters και P. v. d. Putten, *Artificial Intelligence: Definition, Trends, Techniques and Cases*.
- [7] G. M. Sacha και P. Varona, «Artificial intelligence in nanotechnology,» IOPSIENCE, 2013.
- [8] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης και Η. Σακελαρίου, Τεχνητή Νοημοσύνη, Πανεπιστήμιο Μακεδονίας, 2006.
- [9] J. Ferber και G. Weiss, «Multi-Agent System: An Introduction to Distributed Artificial Intelligence,» *Journal of Artificial Societies and Social Simulation*, τόμ. 1, 1999.
- [10] B. Goertzel και P. Wang, *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms*, IOS Press, 2007.
- [11] A. M. Turing, «OXFORD ACADEMIC,» τόμ. LIX, αρ. 236, pp. 433-460, 1 October 1950.
- [12] J.-S. Chou και D.-K. Bui, «Energy and Buildings,» *Modeling heating and cooling loads by artificial intelligence for energy-efficient building design*, τόμ. 82, pp. 437-446, October 2014.
- [13] P. Szolovits, R. S. Patil και W. B. Schwartz, «Artificial Intelligence in Medical Diagnosis,» *AcpJournals*, 1988.
- [14] M.Z.Naser, «Deriving temperature-dependent material models for structural steel through artificial intelligence,» *Construction and Building Materials*, τόμ. 191, pp. 56-68, 10 December 2018.
- [15] Jui-ShengChou και D.-K. Bui, «Energy and Buildings,» *Modeling heating and cooling loads by artificial intelligence for energy-efficient building design*, τόμ. 82, pp. 427-446, October 2014.
- [16] G. Gurkaynak, I. Yilmaz και G. Hakseverb, «Computer Law & Security Review,» *Stifling artificial intelligence: Human perils*, τόμ. 32, αρ. 5, pp. 749-758, October 2016.
- [17] M. Seidl, «Ausarbeitung zum Hauptseminar Machine Learning,» 2004.

- [18] N. Hezel, «DATA SCIENCE BLOG,» 12 September 2015. [Ηλεκτρονικό]. Available: <https://data-science-blog.com/blog/2015/09/12/knn-vorteile-und-nacheile/>. [Πρόσβαση 2020].
- [19] [Ηλεκτρονικό]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S1871402120300771>.

ΠΑΡΑΡΤΗΜΑ

ΕΞΟΔΟΙ ΕΚΤΕΛΕΣΗΣ ΣΕΝΑΡΙΩΝ

Έξοδος Σεναρίου 1 μέσω PDDL.py

```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python PDDL.py scenarios\Scenario-1\KitchenDomain.pddl scenarios\Scenario-1\KitchenProblem.pddl
-----
['define',
 ['domain', 'dinner'],
 [':requirements', ':strips'],
 [':predicates', ['clean'], ['dinner'], ['quiet'], ['present'], ['garbage']],
 [':action', 'cook', ':precondition', ['clean'], ':effect', ['dinner']],
 [':action', 'wrap', ':precondition', ['quiet'], ':effect', ['present']],
 [':action',
 'carry',
 ':precondition',
 ['garbage'],
 ':effect',
 ['and', ['not', ['garbage']], ['not', ['clean']]]],
 [':action',
 'dolly',
 ':precondition',
 ['garbage'],
 ':effect',
 ['and', ['not', ['garbage']], ['not', ['quiet']]]]]
-----
['define',
 ['problem', 'pb1'],
 [':domain', 'dinner'],
 [':init', ['garbage'], ['clean'], ['quiet']],
 [':goal', ['and', ['dinner'], ['present'], ['not', ['garbage']]]]]
-----
Domain name: dinner
action: cook
parameters: []
positive_preconditions: [['clean']]
negative_preconditions: []
add_effects: [['dinner']]
```

Εικόνα 1 Στιγμιότυπο « Έξοδος Σεναρίου 1 μέσω PDDL.py» part1


```
Administrator: Command Prompt
Domain name: dinner
action: cook
  parameters: []
  positive_preconditions: [['clean']]
  negative_preconditions: []
  add_effects: [['dinner']]
  del_effects: []

action: wrap
  parameters: []
  positive_preconditions: [['quiet']]
  negative_preconditions: []
  add_effects: [['present']]
  del_effects: []

action: carry
  parameters: []
  positive_preconditions: [['garbage']]
  negative_preconditions: []
  add_effects: []
  del_effects: [['garbage'], ['clean']]

action: dolly
  parameters: []
  positive_preconditions: [['garbage']]
  negative_preconditions: []
  add_effects: []
  del_effects: [['garbage'], ['quiet']]

-----
Problem name: pb1
Objects: {}
State: [['garbage'], ['clean'], ['quiet']]
Positive goals: [['dinner'], ['present']]
Negative goals: [['garbage']]
```

Εικόνα 2 Στιγμιότυπο «Έξοδος Σεναρίου 1 μέσω PDDL.py» part2

Έξοδος Σεναρίου 1 μέσω planner.py



```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python planner.py scenarios\Scenario-1\KitchenDomain.pddl scenarios\Scenario-1\KitchenProblem.pddl
Time: 0.0009989738464355469s
plan:
action: cook
  parameters: []
  positive_preconditions: [['clean']]
  negative_preconditions: []
  add_effects: [['dinner']]
  del_effects: []

action: wrap
  parameters: []
  positive_preconditions: [['quiet']]
  negative_preconditions: []
  add_effects: [['present']]
  del_effects: []

action: carry
  parameters: []
  positive_preconditions: [['garbage']]
  negative_preconditions: []
  add_effects: []
  del_effects: [['garbage'], ['clean']]

C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>
```

Εικόνα 3 Στιγμιότυπο «Έξοδος Σεναρίου 1 μέσω planner.py»

Έξοδος Σεναρίου 1 μέσω launcher.py

```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python launcher.py
<<Launcher Message>> Enter a number to select a scenario: 1
<<Launcher Message>> Scenario 1 initiated:

Time: 0.0019254684448242188s
plan: {step count: 3 }
step: 1
    action: cook
    parameters: []
    positive_preconditions: [['clean']]
    negative_preconditions: []
    add_effects: [['dinner']]
    del_effects: []

step: 2
    action: wrap
    parameters: []
    positive_preconditions: [['quiet']]
    negative_preconditions: []
    add_effects: [['present']]
    del_effects: []

step: 3
    action: carry
    parameters: []
    positive_preconditions: [['garbage']]
    negative_preconditions: []
    add_effects: []
    del_effects: [['garbage'], ['clean']]

<<Launcher Message>> Execution Terminated successfully

C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>
```

Εικόνα 4 Στιγμιότυπο «Έξοδος Σεναρίου 1 μέσω launcher.py»

Έξοδος Σεναρίου 2 μέσω launcher.py

```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python launcher.py
<<Launcher Message>> Enter a number to select a scenario: 2
<<Launcher Message>> Scenario 2 initiated:

Time: 0.0019986629486083984s
plan: {step count: 3 }
step: 1
    action: cook
    parameters: []
    positive_preconditions: [['clean']]
    negative_preconditions: []
    add_effects: [['dinner']]
    del_effects: []

step: 2
    action: wrap
    parameters: []
    positive_preconditions: [['quiet']]
    negative_preconditions: []
    add_effects: [['present']]
    del_effects: []

step: 3
    action: carry
    parameters: []
    positive_preconditions: [['garbage']]
    negative_preconditions: []
    add_effects: []
    del_effects: [['garbage'], ['clean']]

<<Launcher Message>> Execution Terminated successfully

C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>_
```

Εικόνα 5 Στιγμιότυπο «Έξοδος Σεναρίου 2 μέσω launcher.py»

Έξοδος Σεναρίου 3 μέσω launcher.py

```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python launcher.py
<<Launcher Message>> Enter a number to select a scenario: 3
<<Launcher Message>> Scenario 3 initiated:

Time: 0.0016064643859863281s
plan: {step count: 3 }
step: 1
    action: order_delivery_food
    parameters: []
    positive_preconditions: []
    negative_preconditions: []
    add_effects: [['dinner']]
    del_effects: []

step: 2
    action: wrap
    parameters: []
    positive_preconditions: [['quiet']]
    negative_preconditions: []
    add_effects: [['present']]
    del_effects: []

step: 3
    action: carry
    parameters: []
    positive_preconditions: [['garbage']]
    negative_preconditions: []
    add_effects: []
    del_effects: [['garbage'], ['clean']]

<<Launcher Message>> Execution Terminated successfully

C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>_
```

Εικόνα 6 Στιγμιότυπο «Έξοδος Σεναρίου 3 μέσω launcher.py»

Έξοδος Σεναρίου 4 μέσω launcher.py

```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python launcher.py
<<Launcher Message>> Enter a number to select a scenario: 4
<<Launcher Message>> Scenario 4 initiated:

Time: 0.0018665790557861328s
plan: {step count: 4 }
step: 1
    action: wrap
    parameters: []
    positive_preconditions: [['quiet']]
    negative_preconditions: []
    add_effects: [['present']]
    del_effects: []

step: 2
    action: carry
    parameters: []
    positive_preconditions: [['garbage']]
    negative_preconditions: []
    add_effects: []
    del_effects: [['garbage'], ['clean']]

step: 3
    action: santify
    parameters: []
    positive_preconditions: []
    negative_preconditions: [['garbage']]
    add_effects: [['clean']]
    del_effects: []

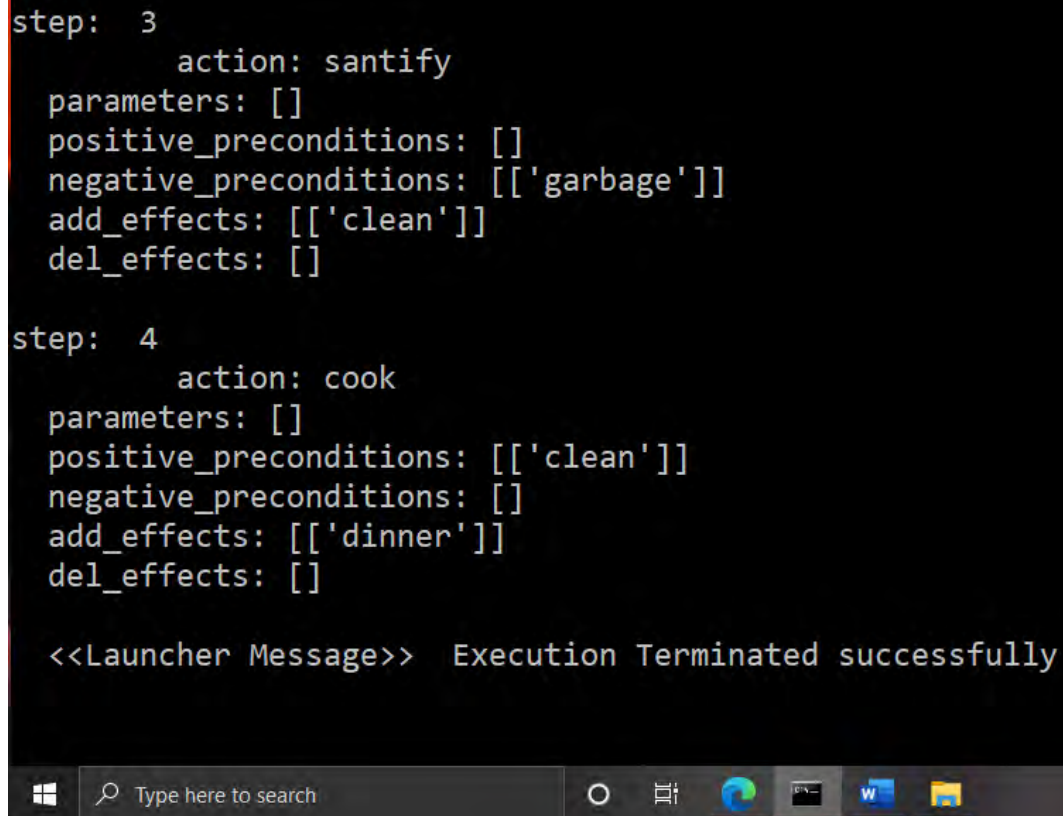
step: 4
    action: cook
    parameters: []
```

Εικόνα 7 Στιγμιότυπο «Έξοδος Σεναρίου 4 μέσω launcher.py» part1

```
step: 3
      action: santify
      parameters: []
      positive_preconditions: []
      negative_preconditions: [['garbage']]
      add_effects: [['clean']]
      del_effects: []

step: 4
      action: cook
      parameters: []
      positive_preconditions: [['clean']]
      negative_preconditions: []
      add_effects: [['dinner']]
      del_effects: []

<<Launcher Message>> Execution Terminated successfully
```



Εικόνα 8 Στιγμιότυπο «Έξοδος Σεναρίου 4 μέσω launcher.py» part2

Έξοδος Σεναρίου 5 μέσω launcher.py

```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python launcher.py
<<Launcher Message>> Enter a number to select a scenario: 5
<<Launcher Message>> Scenario 5 initiated:

Time: 0.0009984970092773438s
plan: {step count: 3 }
step: 1
    action: order_delivery_food
    parameters: []
    positive_preconditions: []
    negative_preconditions: []
    add_effects: [['dinner']]
    del_effects: []

step: 2
    action: wrap
    parameters: []
    positive_preconditions: [['quiet']]
    negative_preconditions: []
    add_effects: [['present']]
    del_effects: []

step: 3
    action: carry
    parameters: []
    positive_preconditions: [['garbage']]
    negative_preconditions: []
    add_effects: []
    del_effects: [['garbage'], ['clean']]

<<Launcher Message>> Execution Terminated successfully

C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>
```

Εικόνα 9 Στιγμιότυπο «Έξοδος Σεναρίου 5 μέσω launcher.py»

Έξοδος Σεναρίου 6 μέσω launcher.py

```
Administrator: Command Prompt
C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>python launcher.py
<<Launcher Message>> Enter a number to select a scenario: 6
<<Launcher Message>> Scenario 6 initiated:

Time: 0.002001047134399414s
plan: {step count: 3 }
step: 1
    action: order_delivery_food
    parameters: []
    positive_preconditions: []
    negative_preconditions: []
    add_effects: [['dinner']]
    del_effects: []

step: 2
    action: wrap
    parameters: []
    positive_preconditions: [['quiet']]
    negative_preconditions: []
    add_effects: [['present']]
    del_effects: []

step: 3
    action: carry
    parameters: []
    positive_preconditions: [['garbage']]
    negative_preconditions: []
    add_effects: []
    del_effects: [['garbage'], ['clean']]

<<Launcher Message>> Execution Terminated successfully

C:\Users\Zaparas\Google Drive\Academics\Projects\DiplomaPaperFinal\Diploma Draft>
```

Εικόνα 10 Στιγμιότυπο «Έξοδος Σεναρίου 6 μέσω launcher.py»