

ΤΙΤΛΟΣ ΔΙΑΤΡΙΒΗΣ

ΕΝΑΣ ΕΥΦΥΗΣ ΠΟΛΥΤΡΟΠΙΚΟΣ ΠΡΑΚΤΟΡΑΣ ΓΙΑ ΤΟ ΠΑΙΧΝΙΔΙ ANGRY BIRDS

Η ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

Υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύνοψης
του Τμήματος Πληροφορικής Εξεταστική Επιτροπή

από τον

Γεώργιο Παπαγιάννη

ως μέρος των Υποχρεώσεων

για τη λήψη του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ
ΜΕ ΕΙΔΙΚΕΥΣΗ ΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ-ΕΦΑΡΜΟΓΕΣ

Σεπτέμβριος 2014

ΑΦΙΕΡΩΣΗ

Στον πατέρα μου Σπύρο, τη μητέρα μου Ελένη, τον αδερφό μου Θανάση και τους πραγματικούς φίλους μου.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω από καρδιάς τον επιβλέποντα καθηγητή μου κ. Κωνσταντίνο Μπλέκα για τις πολύτιμες συμβουλές, την υπομονή και την άριστη επιστημονική καθοδήγηση καθ' όλη τη διάρκεια εκπόνησης της παρούσας διατριβής. Επίσης θα ήθελα να ευχαριστήσω θερμά τον υποψήφιο διδάκτορα Νικόλαο Τζιωρτζιώτη για την εξαιρετική συνεργασία και την πολύπλευρη συνεισφορά του στην προσέγγιση του θέματος της διατριβής.

Επίσης νιώθω την ανάγκη να ευχαριστήσω τους γονείς και τον αδερφό μου, που όλα αυτά τα χρόνια στέκονται δίπλα μου και στηρίζουν κάθε μου προσπάθεια. Τέλος, ένα μεγάλο ευχαριστώ σε όλους τους φίλους που ήταν δίπλα μου όλα αυτά τα χρόνια και συνέβαλαν με το δικό τους ξεχωριστό τρόπο όλο αυτόν τον καιρό, δείχνοντας υπομονή και κατανόηση όποτε χρειάστηκε.

ΠΕΡΙΕΧΟΜΕΝΑ

	Σελ
ΑΦΙΕΡΩΣΗ	ii
ΕΥΧΑΡΙΣΤΙΕΣ	iii
ΠΕΡΙΕΧΟΜΕΝΑ	iv
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	vi
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	vii
ΕΠΕΞΗΓΗΣΕΙΣ ΣΥΜΒΟΛΙΣΜΩΝ	viii
ΠΕΡΙΛΗΨΗ	ix
EXTENDED ABSTRACT IN ENGLISH	xi
ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ	1
1.1. Η Έννοια του Πράκτορα	1
1.1.1. Ευφυείς Πράκτορες	3
1.1.2. Μέτρα Απόδοσης/Αποτελεσματικότητας και Λογική	4
1.1.3. Κλάσεις Ευφυών Πρακτόρων	5
1.1.4. Είδη Περιβάλλοντος και Ιδιότητες	10
1.2. Εφαρμογές Ευφυών Πρακτόρων	11
1.2.1. Ρομποτική (Robotics)	12
1.2.2. Παιχνίδια (Games)	14
1.2.3. Non-Player Characters σε Computer Games	16
1.2.4. Υλοποίηση Πρακτορικών NPCs	17
1.2.5. Γραφικά (Animation)	18
1.3. Αντικείμενο Διατριβής	19
1.4. Διάρθρωση Διατριβής	20
ΚΕΦΑΛΑΙΟ 2. ΤΟ ΠΑΙΧΝΙΔΙ “ANGRY BIRDS”	22
2.1. Ιστορική Αναδρομή και Εξέλιξη	22
2.2. Περιγραφή του Παιχνιδιού	23
2.3. Η Σκηνή του Παιχνιδιού	24
2.3.1. Τύποι Πουλιών	27
2.3.2. Τύποι Γουρουνιών	30
2.4. Το Angry Birds ως Πεδίο Εφαρμογής Μεθόδων Τεχνητής Νοημοσύνης	30
ΚΕΦΑΛΑΙΟ 3. ΠΡΟΤΕΙΝΟΜΕΝΗ ΜΕΘΟΔΟΣ	34
3.1. Εισαγωγή	34
3.2. Γενική Περιγραφή της Μεθόδου	35
3.3. Δενδρική Δομή Αναπαράστασης	37
3.3.1. Τύποι Αντικειμένων ως Κόμβοι του Δένδρου	38
3.3.2. Κατασκευή του Δένδρου	39
3.3.3. Συγχώνευση/Συνένωση Κόμβων	41
3.3.4. Εξαγωγή χαρακτηριστικών κόμβων	42
3.3.5. Παραδείγματα Κατασκευής Δένδρου	43

3.4. Εξαγωγή Υποψήφιων Στόχων	47
3.5. Επιλογή Στόχου Συνδυάζοντας Μοντέλα Παλινδρόμησης	50
3.5.1. Γραμμικά Μοντέλα Παλινδρόμησης	51
3.5.2. Κατασκευή Συναρτήσεων Βάσης	52
3.5.3. Bayesian Ensemble Regression	54
3.5.4. Μηχανισμός UCB για την Επιλογή Regressor	56
3.6. Επιλογή tap time	57
3.7. Online Διαδικασία Μάθησης	59
ΚΕΦΑΛΑΙΟ 4. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	61
4.1. Εισαγωγή	61
4.2. Πρώτη Σειρά Πειραμάτων – Επίπεδα 1-21	62
4.3. Δεύτερη Σειρά Πειραμάτων – Επίπεδα 22-42	63
4.4. Angry Birds AI Competition	65
4.5. Φάσεις του Angry Birds AI Competition	66
4.5.1. Προκριματικές φάσεις QR1, QR2	66
4.5.2. Προημιτελική φάση - Quarter Finals	66
4.5.3. Ημιτελική Φάση – Semi Finals	66
4.5.4. Τελικός - Grand Final	66
4.6. Αποτελέσματα Angry Birds AI Competition 2014	67
ΚΕΦΑΛΑΙΟ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	69
ΑΝΑΦΟΡΕΣ	71
ΔΗΜΟΣΙΕΥΣΕΙΣ ΣΥΓΓΡΑΦΕΑ	79
ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ	80

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας	Σελ
Πίνακας 3.1: Χαρακτηριστικά Κόμβων Δενδρικής Δομής του Σχήματος 3.6	45
Πίνακας 4.1: Στατιστικά του Πράκτορα “AngryBER” για 1-21 Επίπεδα του Επεισοδίου “Poached Eggs”	62
Πίνακας 4.2: Στατιστικά του Πράκτορα “AngryBER” για τα Επίπεδα 22-42 του Επεισοδίου “Poached Eggs”	64
Πίνακας 4.3: Αποτελέσματα Προκριματικής Φάσης	67
Πίνακας 4.4: Αποτελέσματα Προημιτελικής Φάσης	68
Πίνακας 4.5: Αποτελέσματα Ημιτελικής Φάσης και Τελικού	68
Πίνακας Π.1: Benchmarks Angry Birds AI Competition 2013-2014 – Επίπεδα 1-21 του “Poached Eggs”	75
Πίνακας Π.2: Benchmarks Angry Birds AI Competition 2013-2014 - Επίπεδα 22-42 του “Poached Eggs”	77
Πίνακας Π.3: Αθροιστικά Αποτελέσματα των Πινάκων Π.1 και Π.2	78

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχήμα	Σελ
Σχήμα 1.1: Τυπική Σχέση Πράκτορα – Περιβάλλοντος	2
Σχήμα 1.2: Simple Reflex Agent	6
Σχήμα 1.3: Model-based Reflex Agent	7
Σχήμα 1.4: Goal-based Agent	7
Σχήμα 1.5: Utility-based Agent	8
Σχήμα 1.6: Learning Agent	9
Σχήμα 1.7: Finite State Machine Ενός NPC-Στρατιώτη σε Ένα Παιχνίδι Δράσης	17
Σχήμα 3.1: Στάδια Προτεινόμενης Μεθόδου	37
Σχήμα 3.2: Αναπαράσταση Αντικειμένων της Σκηνης	38
Σχήμα 3.3: Παράδειγμα Συνένωσης Κόμβων του Δένδρου	42
Σχήμα 3.4: Κατασκευή Δένδρου – 4 Βήματα	43
Σχήμα 3.5: Κατασκευή Δένδρου – Βήματα 5-9 - Ολοκληρωμένη Δενδρική Δομή	44
Σχήμα 3.6: Ολοκληρωμένη Δενδρική Δομή 9 ^{ου} Επιπέδου	46
Σχήμα 4.1: Το client/server Πρωτόκολλο	65

ΕΠΕΞΗΓΗΣΕΙΣ ΣΥΜΒΟΛΙΣΜΩΝ

- *direct shot*: όταν η γωνία εκτόξευσης είναι μικρότερη των 45° , τότε πρόκειται για *direct shot*
- *high-arching shot*: όταν η γωνία εκτόξευσης είναι μεγαλύτερη των 45° , τότε πρόκειται για *high-arching shot*
- *Naive Agent*: ο πράκτορας που παρέχεται στο [18] ως βάση για την κατασκευή ευφυών αυτόνομων πρακτόρων. Επιλέγει τυχαία κάποιο από τα γουρούνια της σκηνής και στοχεύει σε περιοχές κοντά σε αυτά. Είναι γενικά αποδεκτό από την επιστημονική κοινότητα που ασχολείται με το παιχνίδι “Angry Birds”, ότι ο *Naive Agent* προσφέρει ένα καλό μέτρο σύγκρισης της επίδοσης άλλων πρακτόρων στα επίπεδα του παιχνιδιού

ΠΕΡΙΛΗΨΗ

Γεώργιος Παπαγιάννης του Σπυρίδωνα και της Ελένης, MSc, Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Σεπτέμβριος, 2014. Ένας Ευφυής Πράκτορας για το Παιχνίδι Angry Birds.

Επιβλέπωντας: Κωνσταντίνος Μπλέκας.

Το παιχνίδι “Angry Birds” (Rovio™) αποτελεί ένα από τα δημοφιλέστερα παιχνίδια της εποχής μας. Σκοπός του παιχνιδιού είναι η εξολόθρευση των γουρουνιών, τα οποία συνήθως είναι προστατευμένα σε δομές, οι οποίες αποτελούνται από διάφορους τύπους υλικών. Ο παίκτης έχει στη διάθεσή του ένα περιορισμένο αριθμό πουλιών, τα οποία εκτοξεύει προς το στόχο μέσω μίας σφεντόνας. Διάφοροι τύποι πουλιών είναι διαθέσιμοι, μερικοί εκ των οποίων έχουν μεγαλύτερη επίδραση σε συγκεκριμένα υλικά αλλά και επιπλέον χαρακτηριστικά. Λόγω της φύσης του (μεγάλος χώρος καταστάσεων και ενεργειών, θορυβώδης ανίχνευση αντικειμένων, ανακρίβειες Φυσικών μοντέλων), το παιχνίδι “Angry Birds” αποτελεί μία ανοιχτή πρόκληση για τον τομέα της Τεχνητής Νοημοσύνης. Τα τελευταία χρόνια έχουν προταθεί αρκετές εργασίες, οι οποίες επικεντρώνονται στην ανάπτυξη ευφύων πρακτόρων (*AI agents*) με στόχο να προσεγγίσουν ή ακόμη και να ξεπεράσουν τις ανθρώπινες επιδόσεις.

Στην παρούσα διατριβή, προτείνουμε ένα *Ensemble Μπεϋζιανό μηχανισμό Παλινδρόμησης* για την ανάπτυξη ενός ευφυούς πράκτορα (*AngryBER – Bayesian Ensemble Regression*). Μία καινοτόμα δενδρική δομή προτείνεται για την αναπαράσταση της σκηνής του παιχνιδιού. Η δομή αυτή παρέχει πληθώρα πληροφοριών, ενώ ταυτόχρονα είναι σχετικά απλή, γεγονός που μειώνει το υπολογιστικό κόστος και καθιστά δυνατή την εφαρμογή της προτεινόμενης μεθόδου σε πραγματικό χρόνο. Ταυτόχρονα, διευκολύνει την εξαγωγή χαρακτηριστικών, τα οποία χρησιμοποιούνται στα μοντέλα Παλινδρόμησης για την πρόβλεψη της

αναμενόμενης ανταμοιβής (σκορ) κάθε πιθανού στόχου. Πιο συγκεκριμένα, για κάθε δυνατό ζεύγος “*material – bird*” χρησιμοποιείται ένα διαφορετικό μοντέλο Μπεϋζιανής Γραμμικής Παλινδρόμησης (*Bayesian Ensemble Regression*), ενώ εφαρμόζεται ο αλγόριθμος *UCB (Upper Confidence Bound)* για τη λήψη της τελικής απόφασης. Τέλος, οι παράμετροι του μοντέλου που αντιστοιχεί στο στόχο που έχει επιλεγεί, ενημερώνονται εφαρμόζοντας μία ακολουθιακή (*online*) διαδικασία μάθησης.

Για την αξιολόγηση του πράκτορα *AngryBER*, παρουσιάζουμε τα αριθμητικά αποτελέσματα των πειραμάτων που έγιναν σε ένα μεγάλο αριθμό επιπέδων του παιχνιδιού. Επιπλέον, παρουσιάζουμε τα αποτελέσματα του διεθνούς διαγωνισμού *Angry Birds AI Competition (ECAI, 2014)*, όπου συμμετείχε ο πράκτορας *AngryBER*, κατακτώντας τη **2^η θέση**. Λαμβάνοντας μέρος στο συγκεκριμένο διαγωνισμό είχαμε την ευκαιρία να αξιολογήσουμε την αποδοτικότητα του πράκτορα *AngryBER* σε άγνωστα έως τώρα επίπεδα, καθώς επίσης και να συγκρίνουμε με αντικειμενικότητα την απόδοσή αυτού σε σχέση με τους υπόλοιπους συμμετέχοντες πράκτορες.

EXTENDED ABSTRACT IN ENGLISH

Papagiannis Georgios, S. MSc, Computer Science Department, University of Ioannina, Greece. September, 2014. An Advanced Intelligent Multi-modal Agent for the Angry Birds Game.

Thesis Supervisor: Konstantinos Blekas.

“Angry birds” was first launched five years ago by Rovio™, and since then it has become one of the most popular games nowadays. The objective is to get rid of the pigs, which are usually protected in structures made of different kinds of building materials, by killing them. This is achieved by taking control of a limited number of various birds’ types, which the player launches to the targets (e.g. building blocks or pigs) via a slingshot. It must be noted that different types of birds are available with some of them being more effective against particular materials, while some other have special features. The received return at each level is calculated according to the number of pigs killed, the number of the unused birds as well as to the destruction on the structure that achieved.

Due to its nature (e.g. large state and action spaces, continuous tap timing, various objects’ properties, noisy object detection, inaccurate physical models), “Angry Birds” constitute a really challenging task. During the last two years, a number of works have been proposed which are focused on the development of AI agents with playing capabilities similar to those exhibited by human players. The Angry Birds competitions pose several challenges for building various AI approaches.

In this work, we propose a Bayesian ensemble regression framework for designing an intelligent agent (*AngryBER – Bayesian Ensemble Regression*) for the Angry Bird domain. The main advantages of our approach lies on two aspects: Firstly, a novel

tree structure is proposed for mapping scenes of game levels, where the nodes represent different material of solid objects. This state representation is informative as it incorporates all the necessary knowledge about game snapshots, and simultaneously abstract so as to reduce the computational cost and accelerate the learning procedure. This tree representation allows the construction of an efficient and powerful feature space that can be used next for the prediction. Secondly, an ensemble learning approach is designed where every possible pair of “material - bird” has its own Bayesian linear regression model for calculating the expected reward. An ensemble integration framework based on the UCB algorithm is employed using the predictions to obtain the final ensemble prediction. Then, an online estimation procedure is performed in order to adjust the regression model parameters. Finally, an appropriate Gaussian kernel space has been constructed.

A series of experiments has been conducted in an attempt to analyze the performance of the proposed agent in the Angry Birds domain. Our analysis was concentrated mainly on the first 21 levels of the freely available “Poached Eggs” episode of Angry Birds. Finally, we present the results of the last Angry Birds AI Competition, where AngryBER agent participated. This event provided us an ideal opportunity to test the proposed agent’s generalization ability.

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

- 1.1 Η Έννοια του Πράκτορα
 - 1.2 Εφαρμογές Ευφύων Πρακτόρων
 - 1.3 Αντικείμενο Διατριβής
 - 1.4 Διάρθρωση της Διατριβής
-

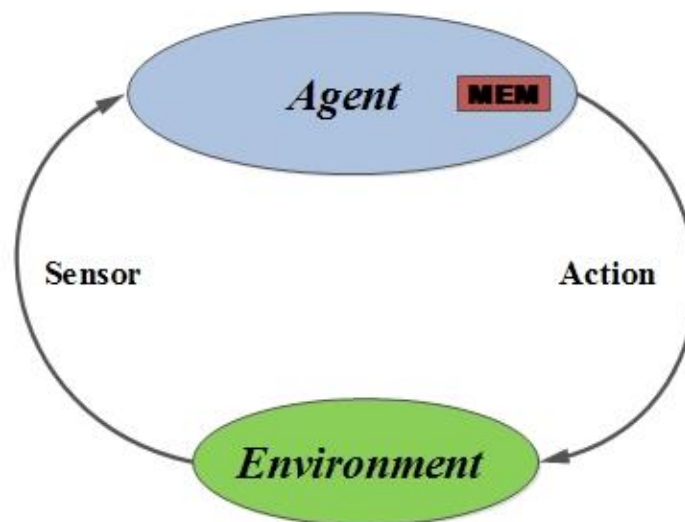
1.1. Η Έννοια του Πράκτορα

Ο όρος πράκτορας (*agent*) έχει πολλούς ορισμούς και έννοιες. Συνήθως συγχέεται με τον όρο ρομπότ (*robot*), αλλά με τον όρο αυτό δεν αναφερόμαστε μόνο σε *hardware*. Το ρομπότ για να πραγματοποιήσει κάποιες ενέργειες στο *περιβάλλον* (*environment*) που δρα στηρίζεται σε ένα *λογισμικό* (*software*). Επομένως, το ρομπότ αποτελεί την *αρχιτεκτονική* (*architecture*) και το λογισμικό με βάση το οποίο είναι προγραμματισμένο αποτελεί το *πρόγραμμα* (*program*). Ο συνδυασμός των δύο παραπάνω μας δίνουν έναν ολοκληρωμένο πράκτορα, ο οποίος δρα μέσα σε ένα περιβάλλον από το οποίο μπορεί να αντλεί γνώση μέσω διαφόρων ειδών αισθητήρων. Συνοψίζοντας, καταλήγουμε στο εξής [10]:

$$\mathbf{agent = architecture + program}$$

Ο όρος *program* καλείται και *agent function*, και πρόκειται για μία αφηρημένη έννοια που μπορεί να οριστεί μαθηματικά ως: η συνάρτηση η οποία *απεικονίζει* κάθε δυνατή ακολουθία αντίληψης (*percept sequence*) σε ενέργεια που θα εκτελέσει ο πράκτορας, ή σε κάποιο πραγματικό αριθμό που ενδεχομένως δείχνει τη χρησιμότητα που αποφέρουν οι ενέργειές του (*utility*), ή και σε κάποια άλλη συνάρτηση ή σταθερά που επιδρά σε ενδεχόμενες ενέργειές του.

Η παραπάνω εισαγωγή έγινε για να γίνει ξεκάθαρο ότι όταν αναφερόμαστε στην έννοια του πράκτορα, δεν εννοούμε πάντα κάποιο ρομποτικό σύστημα. Εννοούμε κάποια *οντότητα (entity)*, η οποία μπορεί να αντιλαμβάνεται το περιβάλλον της μέσω αισθητήρων (*sensors*), και έπειτα μπορεί να δρα σε αυτό (*actions*) χρησιμοποιώντας κάποιους μηχανισμούς του (*effectors*). Επιπλέον, μπορεί να διαθέτει και *μνήμη (memory)* για να μπορεί να αποθηκεύει τις δράσεις του και να μαθαίνει από αυτές. Η περιγραφή που έγινε δίνεται στο Σχήμα 1.1 που ακολουθεί.



Σχήμα 1.1: Τυπική Σχέση Πράκτορα – Περιβάλλοντος

Με τον όρο *sensor* εννοούμε συνήθως κάποιο μηχανισμό, ο οποίος παρέχει στον πράκτορα όλες τις πληροφορίες σχετικά με το περιβάλλον του. Οι ενέργειες εκτελούνται αφού έχει προηγηθεί μία διαδικασία απόφασης, κατά τη διάρκεια της οποίας λαμβάνονται υπόψη όλες οι πληροφορίες σχετικά με την κατάσταση του περιβάλλοντος, αλλά και του πράκτορα.

Επίσης, ο όρος *agent* όταν αναφέρεται σε *software agents* χρησιμοποιείται για να περιγράψει με απλό και αφηρημένο τρόπο ένα *πολύπλοκο software*, το οποίο ενεργεί *αυτόνομα* προκειμένου να επιτύχει κάποιο στόχο, ο οποίος έχει τεθεί από τον σχεδιαστή του. Ουσιαστικά, πρόκειται για *λογισμικό (computer program)* που είναι προγραμματισμένο να ενεργεί με *πρακτορικό* τρόπο, δηλαδή για λογαριασμό του σχεδιαστή του, σε ένα συγκεκριμένο περιβάλλον ώστε να επιτύχει ένα στόχο.

1.1.1. Ευφυείς Πράκτορες

Μέχρι σήμερα δεν έχει διατυπωθεί κάποιος ορισμός για τον όρο του *πράκτορα* που να είναι καθολικά αποδεκτός, γεγονός που δείχνει την πολύπλευρη διάσταση του συγκεκριμένου όρου. Στο [17] ο όρος *πράκτορας* δίνεται ως εξής: «*Πράκτορας είναι ένα υπολογιστικό σύστημα το οποίο σχετίζεται με κάποιο περιβάλλον, και είναι ικανό να δρα αυτόνομα μέσα σε αυτό με στόχο να ικανοποιήσει τους στόχους για τους οποίους σχεδιάστηκε*». Ο παραπάνω ορισμός αποτυπώνεται και στο Σχήμα 1.1 που δόθηκε προηγουμένως. Εισάγει όμως και μία άλλη έννοια, αυτή της αυτονομίας (*autonomy*), χαρακτηριστικό των *ευφύων πρακτόρων* (*intelligent agents*).

Κατά το [16] ένας *ευφύης πράκτορας* ορίζεται ως ένα υπολογιστικό σύστημα που συνήθως βασίζεται στο λογισμικό (ή και στο υλικό) και έχει τις ακόλουθες ιδιότητες:

- **autonomy:** οι πράκτορες λειτουργούν χωρίς κάθε είδους παρέμβαση (είτε ανθρώπινη, είτε από κάποια άλλη υπολογιστική οντότητα). Οι ενέργειές τους αποφασίζονται με βάση την κατάσταση τους, αλλά και την κατάσταση του περιβάλλοντος στο οποίο βρίσκονται
- **social ability:** αλληλεπιδρούν με άλλους πράκτορες ή και με ανθρώπινες οντότητες μέσω κάποιας *οικείας γλώσσας* (*agent-communication language*)
- **reactivity:** λαμβάνουν πληροφορίες σχετικά με το περιβάλλον τους και δίνουν άμεση απόκριση, λαμβάνοντας υπόψη τις αλλαγές που έχουν παρατηρήσει σε αυτό
- **pro-activeness:** δρουν όχι μόνο σε σχέση με το περιβάλλον τους, αλλά μπορούν να εκδηλώσουν ορισμένες *πρωτοβουλίες*, προκειμένου να φτάσουν στον στόχο για τον οποίο έχουν σχεδιαστεί.

Οι παραπάνω ιδιότητες ενός *πράκτορα* καθορίζουν εν τέλει τη συμπεριφορά του (*behavior*). Η συμπεριφορά ενός *πράκτορα* πρέπει να βασίζεται στην εμπειρία που έχει αποκτήσει ο ίδιος σύμφωνα με προηγούμενες δράσεις του, αλλά και στην γνώση που του έχει *εμφυτεύσει* ο κατασκευαστής του σχετικά με το περιβάλλον του. Αν ένας *πράκτορας* δρα αποκλειστικά με βάση την αρχική *εμφυτευμένη* γνώση του, τότε λέμε ότι *στερείται* αυτονομίας. Πιο γενικά, ένας *πράκτορας* χαρακτηρίζεται *αυτόνομος* στο βαθμό που η συμπεριφορά του καθορίζεται από την γνώση που έχει αποκτήσει μέσα

από τη δράση του στο συγκεκριμένο περιβάλλον [10]. Βέβαια, δεν αναμένουμε ένας πράκτορας να ενεργεί αυτόνομα από την πρώτη στιγμή που θα βρεθεί στο περιβάλλον του. Όταν η εμπειρία του (*experience*) είναι ελάχιστη έως μηδαμινή πρέπει να ενεργεί με σχετική *τυχειότητα*, ή λαμβάνοντας κάποια *βοήθεια* μέχρι να αποκτήσει τη στοιχειώδη εμπειρία που θα του επιτρέψει να *εξελιξεί* τη συμπεριφορά του σε αυτόνομη.

1.1.2. Μέτρα Απόδοσης/Αποτελεσματικότητας και Λογική

Κάθε πράκτορας κατασκευάζεται για να εξυπηρετήσει ένα σκοπό και σύμφωνα με αυτό θα κριθεί για το κατά πόσο είναι αποτελεσματικός στην επίτευξη του στόχου. Αυτό συνεπάγεται ότι τα κριτήρια αποτελεσματικότητας είναι διαφορετικά για κάθε πράκτορα. Έστω ότι έχουμε ένα αυτόματο ταξί [10]. Αυτό σημαίνει ότι υπάρχει μία αρχιτεκτονική και ένα λογισμικό τα οποία συνθέτουν έναν πράκτορα, ο οποίος είναι σε θέση να μεταφέρει κάθε πελάτη στον προορισμό του. Με βάση το περιβάλλον, αλλά και τον στόχο του πράκτορα μπορούν να εξαχθούν κάποια κριτήρια αποτελεσματικότητας. Ένα από αυτά μπορεί να είναι ο χρόνος στο οποίο φτάνει ο πελάτης στον προορισμό του. Όσο πιο γρήγορα, τόσο πιο αποτελεσματικός και αποδοτικός κρίνεται ο πράκτορας. Ένα άλλο μέτρο απόδοσης μπορεί να είναι το κόστος της διαδρομής, δηλαδή τα καύσιμα, ενδεχομένως και τα διόδια της διαδρομής. Ένα ακόμη πιο χρήσιμο μέτρο μπορεί να είναι ο συνδυασμός των δύο παραπάνω, δηλαδή να φτάσει το ταχύτερο δυνατό το ταξί στον προορισμό του, έχοντας καταναλώσει όσο το δυνατόν λιγότερα καύσιμα με αποφυγή ενδεχόμενων διοδίων. Επιπλέον, σημαντικό κριτήριο αποτελεί η ασφάλεια του πελάτη. Ο συνδυασμός όλων των παραπάνω θα μπορούσε να είναι το γενικό μέτρο απόδοσης του πράκτορα που υλοποιεί το αυτόματο ταξί.

Επίσης, είναι σημαντικό το *πότε* αξιολογούμε την απόδοση του πράκτορα. Στο προηγούμενο παράδειγμα, ο πράκτορας θα έχει κακή απόδοση εάν το ταξί περάσει μία διασταύρωση με κόκκινο φανάρι, διότι η ασφάλεια του πελάτη τίθεται σε κίνδυνο. Άρα ο πράκτορας πρέπει να έχει *πολύ καλή γνώση του περιβάλλοντος* με το οποίο αλληλεπιδρά και αυτό εξαρτάται αρχικά από τον σχεδιαστή του, και έπειτα από τον ίδιο τον πράκτορα, υπό την έννοια ότι οι *ενέργειές* του πρέπει να του *προσδίδουν*

επιπλέον γνώση και πληροφορία σχετικά με το περιβάλλον στο οποίο λειτουργεί. Όπως γίνεται αντιληπτό, το μέτρο αποτελεσματικότητας για κάθε πράκτορα διαφέρει ανάλογα με τον στόχο και το περιβάλλον στο οποίο δραστηριοποιείται, αλλά και τις ενέργειες του για συγκεκριμένες *κομβικές* χρονικές στιγμές κατά τη δράση του.

Με βάση τα παραπάνω, γίνεται φανερό ότι κάθε πράκτορας είναι σχεδιασμένος ώστε να μεγιστοποιεί την απόδοσή του. Αν και ακούγεται απλό, επί της ουσίας είναι αρκετά δύσκολο. Για να μεγιστοποιήσει την απόδοσή του ένας πράκτορας πρέπει να εκτελεί τη *σωστή ενέργεια* τη *σωστή χρονική στιγμή*, δηλαδή να ενεργεί με βάση τη *λογική (rationality)* που έχει αποκτήσει μέχρι εκείνη τη χρονική στιγμή. Είναι δύσκολο να οριστεί η τέλεια λογική σε ένα τέτοιου είδους σύστημα, κυρίως γιατί είναι πολύ δύσκολο να έχουν προβλεφθεί όλες οι δυνατές καταστάσεις του περιβάλλοντος, ώστε ο πράκτορας να τις συνδυάσει και να εκτελέσει την *τέλεια* ενέργεια. Μπορούμε να πούμε ότι ένας πράκτορας ενεργεί με βάση τη *λογική* σε μία δεδομένη χρονική στιγμή λαμβάνοντας υπόψη:

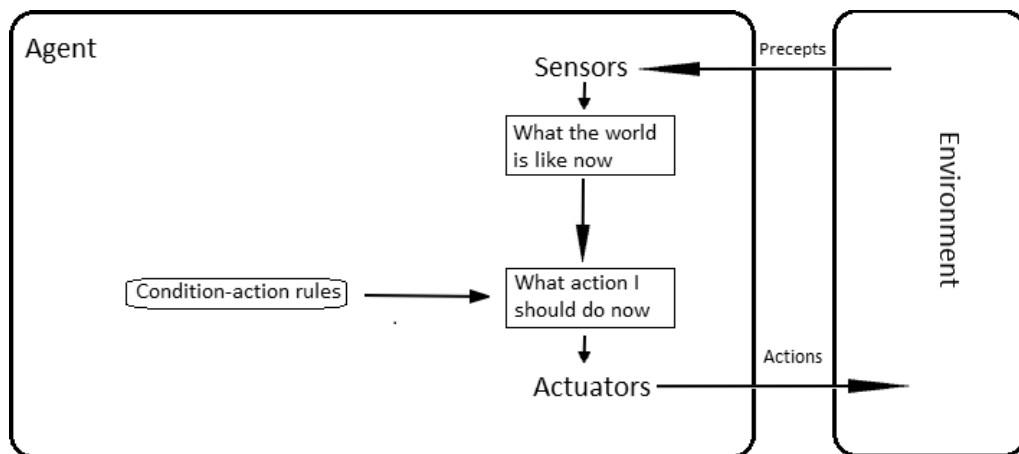
- το *μέτρο* απόδοσης/αποτελεσματικότητας που έχει οριστεί
- τη *γνώση* για το περιβάλλον του
- οτιδήποτε έχει *αντιληφθεί* μέχρι την παρούσα χρονική στιγμή (*percept sequence*)
- τις *ενέργειες* που μπορεί να εκτελέσει

Χρησιμοποιώντας τις παραπάνω διαπιστώσεις μπορούμε να ορίσουμε να ως *ideal rational agent*, τον πράκτορα που εκτελεί την ενέργεια εκείνη που αναμένεται να *μεγιστοποιήσει* το μέτρο αποτελεσματικότητάς του, με βάση τη γνώση που του παρέχει το επίπεδο αντίληψής του για το περιβάλλον του και τη γνώση που του έχει *εμφυτεύσει* ο σχεδιαστής του.

1.1.3. Κλάσεις Ευφυών Πρακτόρων

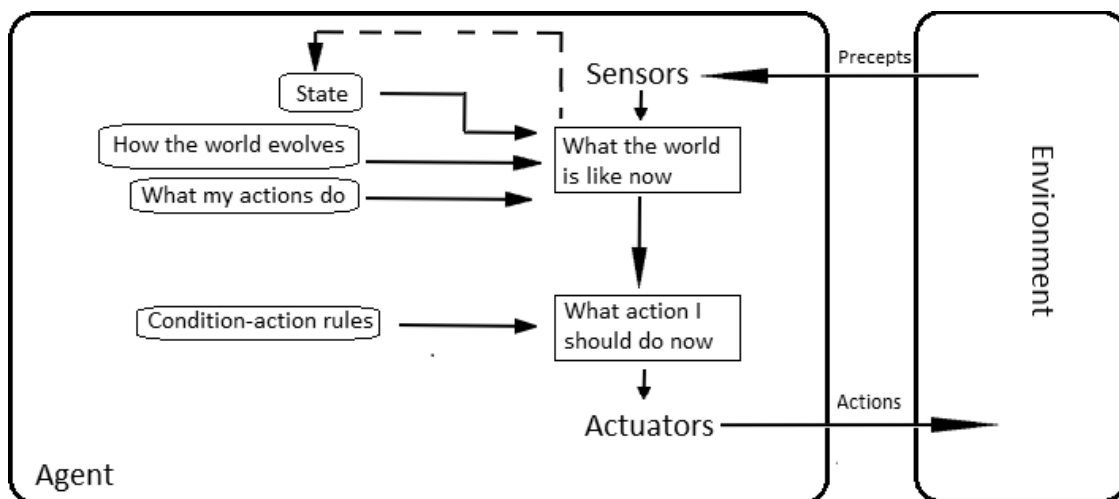
Στο [10] οι ευφυείς πράκτορες κατηγοριοποιούνται σε πέντε *κλάσεις* ανάλογα με το επίπεδο αντίληψής τους και τον τρόπο που επιλέγουν την κάθε τους ενέργεια:

1. **Simple reflex agents:** Οι πράκτορες αυτής της κλάσης ενεργούν σύμφωνα με την πιο πρόσφατη αντίληψή τους, αγνοώντας τι έχει συμβεί σε προηγούμενες χρονικές στιγμές. Επιλέγουν κάθε τους ενέργεια μέσω κανόνων (*if-then rules* ή *condition-action rule*). Όπως φαίνεται και στο Σχήμα 1.2, η συνάρτηση *agent function* τέτοιου είδους πρακτόρων μπορεί να λειτουργήσει μόνο όταν το περιβάλλον έχει παρατηρηθεί πλήρως. Αν υπάρχουν κάποιες πτυχές του που δεν έχουν παρατηρηθεί τότε οι κανόνες μπορεί να οδηγήσουν σε ατέρμονους βρόγχους ενεργειών (*infinite loops*). Αυτό μπορεί να αποφευχθεί εάν κάποιες από τις ενέργειες του πράκτορα εκτελεστούν με τυχαιότητα και αγνοώντας τους κανόνες.



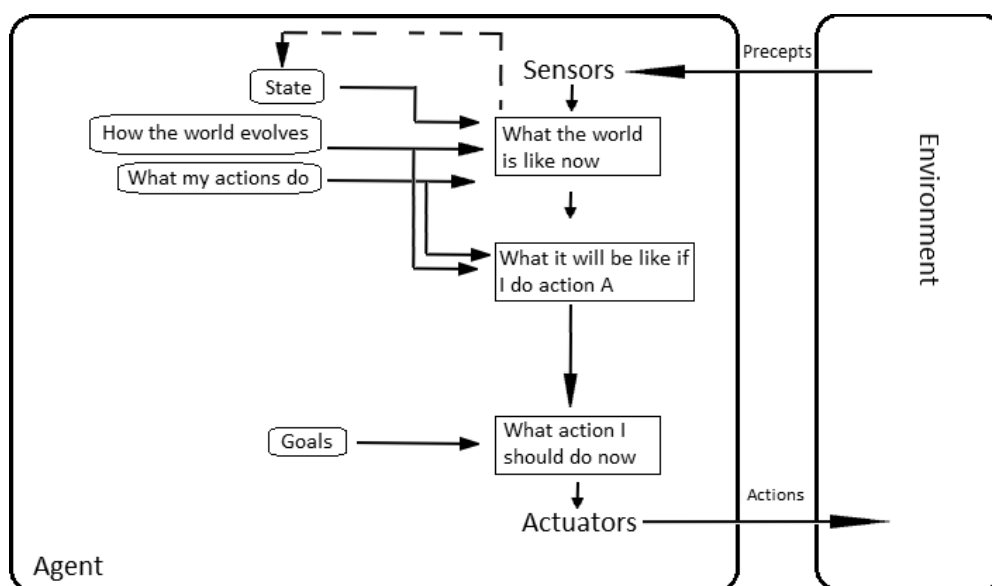
Σχήμα 1.2: Simple Reflex Agent

2. **Model-based reflex agents:** Διαφέρουν σε σχέση με πράκτορες της προηγούμενης κλάσης στη μνήμη που διαθέτουν και μπορούν να εξάγουν κάποια συμπεράσματα για το μέρος του περιβάλλοντος που δεν έχει παρατηρηθεί ακόμη. Τα συμπεράσματα αυτά ενσωματώνονται στους υπάρχοντες κανόνες και τελικά ενεργούν ακολουθώντας τη μέθοδο της προηγούμενης κλάσης πρακτόρων (*if-then rules*). Ονομάζεται *model-based*, διότι βασίζεται στη γνώση για το μέρος του περιβάλλοντος που έχει παρατηρηθεί (*world model*).



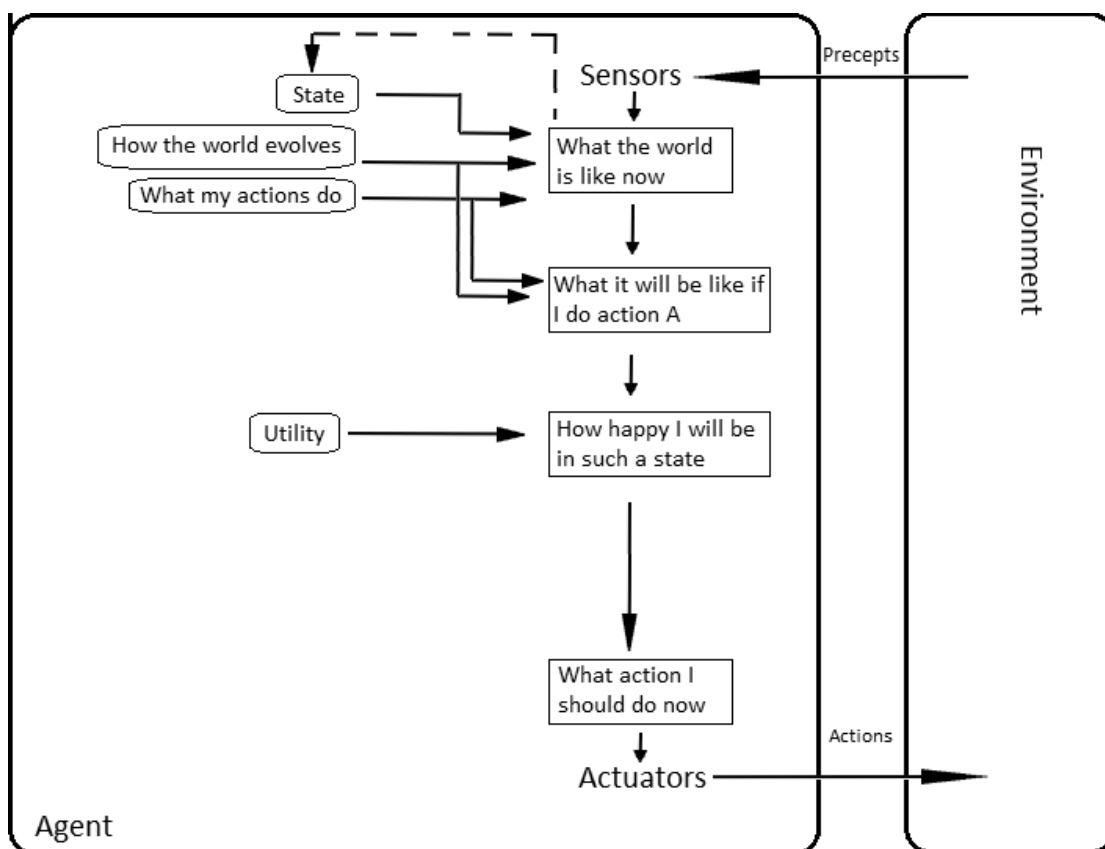
Σχήμα 1.3: Model-based Reflex Agent

3. **Goal-based agents:** Επεκτείνουν τις δυνατότητες των *model-based agents* (Σχήμα 1.4) χρησιμοποιώντας εκ των προτέρων γνώση σχετικά με τον στόχο τους. Η γνώση αυτή περιγράφει ποιες είναι οι επιθυμητές καταστάσεις. Αυτό δίνει τη δυνατότητα στον πράκτορα να επιλέξει ανάμεσα από πολλές ενέργειες, εκείνη που θα τον οδηγήσει στο στόχο του (*searching and planning*). Η γνώση που έχουν οι *goal-based agents* και τελικά τους οδηγούν στις ανάλογες ενέργειες αναπαριστώνται ρητά και μπορούν να τροποποιηθούν ανάλογα με το περιβάλλον και τον στόχο τους.



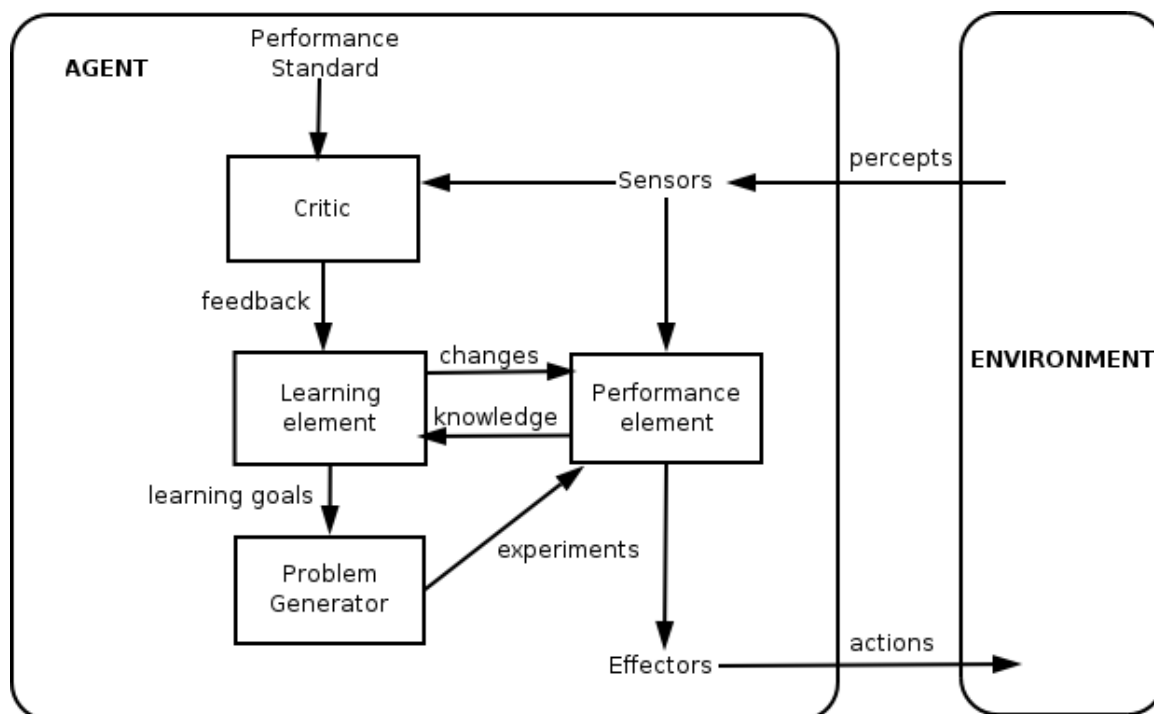
Σχήμα 1.4: Goal-based Agent

4. **Utility-based agents:** Οι *goal-based agents* μπορούν να διακρίνουν αν βρίσκονται στην κατάσταση κατά την οποία έχουν πετύχει το στόχο τους (*goal state*) ή σε κάποια άλλη (*non-goal state*). Οι *utility-based agents* (Σχήμα 1.5) μέσω της *utility function* που διαθέτουν να αντιληφθούν τη χρησιμότητα μίας κατάστασης, με άλλα λόγια πόσο επιθυμητή είναι η κατάσταση αυτή. Ένας πράκτορας αυτής της κλάσης, καλείται να επιλέξει την ενέργεια που θα *μεγιστοποιήσει* την αναμενόμενη χρησιμότητα της επόμενης κατάστασής του. Τέλος, ένας *utility-based agent* πρέπει να παρακολουθεί το περιβάλλον του, εισάγοντας πλέον τις έννοιες της *αντίληψης* (*perception*), της *αναπαράστασης* (*representation*) και της *μάθησης* (*learning*).



Σχήμα 1.5: Utility-based Agent

5. **Learning Agents:** Η μάθηση (*learning*) επιτρέπει στους πράκτορες που χρησιμοποιούν τεχνικές μάθησης να λειτουργούν σε άγνωστα για αυτούς περιβάλλοντα και στη συνέχεια, αλληλεπιδρώντας με το περιβάλλον τους να εμπλουτίζουν την αρχική τους γνώση. Όπως φαίνεται στο Σχήμα 1.6, το *learning element* λαμβάνει την απόκριση από το *critic element*, το οποίο κρίνει τα αποτελέσματα των ενεργειών του πράκτορα, και καθορίζει πως πρέπει να τροποποιηθεί το *performance element*, ώστε οι μελλοντικές ενέργειες να οδηγήσουν σε καλύτερες καταστάσεις. Πρέπει να σημειώσουμε ότι, το *learning element* είναι υπεύθυνο για τη βελτίωση της απόδοσης του πράκτορα, ενώ το *performance element* είναι υπεύθυνο για την επιλογή των ενεργειών αυτών.



Σχήμα 1.6: Learning Agent

1.1.4. Είδη Περιβάλλοντος και Ιδιότητες

Οι πράκτορες αλληλεπιδρούν με το περιβάλλον τους. Ως αποτέλεσμα, οι ιδιότητες του περιβάλλοντος παίζουν σημαντικό ρόλο κατά τη σχεδίαση και ανάπτυξη πρακτόρων. Στο [17] τα διάφορα περιβάλλοντα που μπορεί να συναντήσει ένας πράκτορας κατηγοριοποιούνται ως εξής:

- ***fully observable vs partially observable (accessible vs inaccessible)***: Σε ένα *fully observable/accessible* περιβάλλον ο πράκτορας έχει πλήρη, ακριβή και άμεση γνώση σχετικά με την κατάσταση του περιβάλλοντος. Όσο περισσότερες από τις παραπάνω ιδιότητες πληροί ένα περιβάλλον, τόσο πιο εύκολη είναι η σχεδίαση ενός πράκτορα που θα λειτουργεί σε αυτό. Όμως, τα περισσότερα περιβάλλοντα (όπως το *Internet*) είναι *partially observable/inaccessible*
- ***deterministic vs non-deterministic***: Σε *ντετερμινιστικά* περιβάλλοντα δεν υπάρχει *αβεβαιότητα*, οποιαδήποτε κίνηση έχει ένα προκαθορισμένο αποτέλεσμα. *Μη-ντετερμινιστικά* περιβάλλοντα θέτουν περισσότερες δυσκολίες στον σχεδιαστή του πράκτορα, καθώς μία ενέργεια μπορεί να αποφέρει πληθώρα αποτελεσμάτων
- ***episodic vs non-episodic***: Σε ένα *episodic* περιβάλλον, η απόδοση του πράκτορα εξαρτάται από κάθε επεισόδιο ξεχωριστά. Δεν υπάρχει σύνδεση μεταξύ διαφορετικών επεισοδίων για την απόδοση του πράκτορα. Τα *episodic* περιβάλλοντα είναι πιο απλά από την πλευρά του σχεδιαστή του πράκτορα, διότι ο πράκτορας πρέπει να αποφασίζει για την κάθε ενέργειά του βασιζόμενος στο τρέχον επεισόδιο
- ***static vs dynamic***: Ένα περιβάλλον χαρακτηρίζεται *στατικό (static)* αν παραμένει αμετάβλητο, εκτός και αν ο πράκτορας εκτελέσει κάποια ενέργεια που θα οδηγήσει σε μεταβολή του. Ένα *δυναμικό (dynamic)* περιβάλλον επηρεάζεται από πολλές λειτουργίες που επιδρούν σε αυτό και οι μεταβολές του δεν οφείλονται αποκλειστικά στις ενέργειες του πράκτορα. Π.χ. σε ένα δυναμικό περιβάλλον μπορούν να δρουν πολλοί πράκτορες

- ***discrete vs continuous***: Ένα περιβάλλον χαρακτηρίζεται *διακριτό (discrete)* εάν υπάρχει πεπερασμένος αριθμός ενεργειών που μπορούν να εκτελεστούν σε αυτό (π.χ. μία παρτίδα σκάκι), ενώ χαρακτηρίζεται *συνεχές (continuous)* εάν οι ενέργειες που μπορούν να εκτελεστούν λαμβάνουν συνεχείς τιμές (π.χ. για ένα αυτοκίνητο, η ταχύτητα και η τοποθεσία του).

Όπως γίνεται αντιληπτό, η πιο πολύπλοκη περίπτωση για ένα περιβάλλον είναι *inaccessible, non-deterministic, non-episodic, static, continuous*. Ένα τέτοιο πολύπλοκο παράδειγμα περιβάλλοντος είναι ο *φυσικός κόσμος (“Real World”)*.

1.2. Εφαρμογές Ευφών Πρακτόρων

Στις μέρες μας, η χρήση ευφών πρακτόρων είναι αρκετά διαδεδομένη, καλύπτοντας ένα μεγάλο εύρος εφαρμογών, όπως:

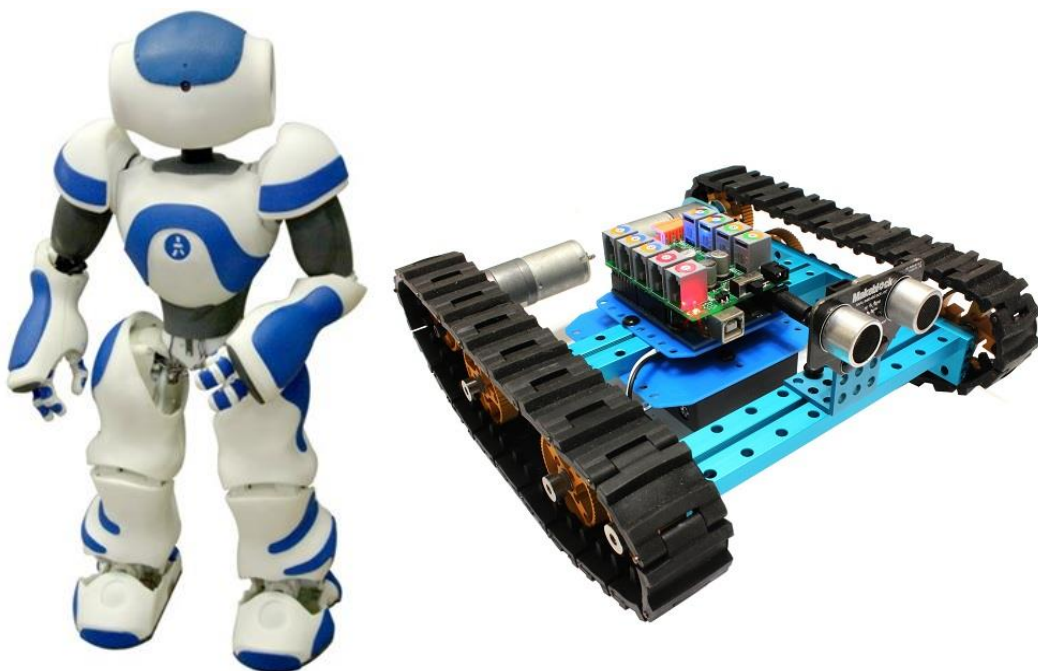
- *Ρομποτική (robotics)*
- *Παιχνίδια (computer games)*
- *Λογισμικό (Software)*
- *Γραφικά (animation)*

Στις παραπάνω εφαρμογές οι πράκτορες είναι σε θέση να *αντιλαμβάνονται* το περιβάλλον τους, να *σκέφτονται* και να *σχεδιάζουν* την επόμενη κίνησή τους, είτε πρόκειται για πράκτορες που ελέγχουν κάποιο ρομποτικό σύστημα, είτε πρόκειται για πράκτορες που δρουν σε ένα υπολογιστικό περιβάλλον (π.χ. σε έναν προσομοιωτή (*simulator*), ή σε ένα περιβάλλον παιχνιδιού όπως το σκάκι, ή το τάβλι). Οι δυνατότητες αυτές απορρέουν από τεχνικές και αλγορίθμους *Τεχνητής Νοημοσύνης* που υλοποιούν οι πράκτορες των παραπάνω εφαρμογών και καθιστούν τις οντότητες στις οποίες είναι ενσωματωμένος ο εκάστοτε πράκτορας *ευφείς*.

1.2.1. Ρομποτική (Robotics)

Ένα ρομποτικό σύστημα (Εικόνα 1.1) αποτελεί μία *ευφυή οντότητα (intelligent entity)*, η οποία είναι ικανή να δρα *αυτόνομα* σε κάποιο περιβάλλον. Είναι σε θέση να αντιλαμβάνεται το περιβάλλον του μέσω *αισθητήρων (sensors)* ή κάποιας κάμερας, να σκέφτεται και να ενεργεί ανάλογα με την κατάσταση του και την κατάσταση του περιβάλλοντος. Συνήθως, ένα ρομποτικό σύστημα πρέπει να πραγματοποιήσει τις ενέργειες που απαιτούνται, προκειμένου να επιτύχει ένα στόχο. Με βάση τα παραπάνω μπορούμε να πούμε ότι αποτελείται από τρία βασικά στοιχεία:

- *μηχανισμός αντίληψης* του περιβάλλοντος ή *αισθητήρες*
- *λογισμικό μέρος (code)*, που αποθηκεύει την πληροφορία που λαμβάνει μέσω των αισθητήρων και χρησιμοποιεί μεθόδους Τεχνητής Νοημοσύνης για να καθορίσει την επόμενη ενέργεια που θα πραγματοποιηθεί. Στο κομμάτι αυτό ενσωματώνεται η έννοια του *πράκτορα*, ο οποίος τελικά αποτελεί την καρδιά του ρομποτικού συστήματος, αφού καθορίζει την κάθε του ενέργεια
- *μηχανικό μέρος* που εκτελεί ενέργεια που καθορίζεται από λογισμικό μέρος



Εικόνα 1.1: Ρομποτικά Συστήματα

Με βάση τον παραπάνω διαχωρισμό αντιλαμβανόμαστε ότι το *κέντρο των αποφάσεων* ενός ρομποτικού συστήματος είναι το λογισμικό μέρος, στο οποίο είναι ενσωματωμένος ο πράκτορας που δρα σε συνεργασία με τον μηχανισμό αντίληψης. Η επόμενη ενέργεια που θα εκτελεστεί από το σύστημα καθορίζεται μέσω μεθόδων Τεχνητής Νοημοσύνης. Οι μέθοδοι αυτοί καθιστούν το σύστημα *αυτόνομο* και *ευφές*, υπό την έννοια ότι *αντιλαμβάνεται* και *προσαρμόζεται* στο περιβάλλον του *μαθαίνοντας* από τις ενέργειές του.

Στις μέρες μας, τα ρομποτικά συστήματα έχουν ευρεία εφαρμογή σε πολλούς τομείς. Σε βιομηχανικό επίπεδο χρησιμοποιούνται για λόγους *ταχύτητας*, *ακρίβειας* και *οικονομίας*. Επιπλέον, χρησιμοποιούνται σε στρατιωτικές επιχειρήσεις, διαστημικές αποστολές και γενικά σε εργασίες οι οποίες είναι επικίνδυνες για τον άνθρωπο. Ενδιαφέρον δεν παρουσιάζουν τα συστήματα που ελέγχονται από τον άνθρωπο, αλλά τα *ευφή* συστήματα που είναι προγραμματισμένα να ενεργούν *αυτόνομα* σε *δυναμικά* περιβάλλοντα, δηλαδή σε περιβάλλοντα που συνεχώς μεταβάλλονται. Στο πεδίο αυτό εστιάζεται και η έρευνα που διεξάγεται από την επιστημονική κοινότητα που ασχολείται με την Ρομποτική. Δημιουργούνται και κατασκευάζονται ρομπότ που αλληλεπιδρούν μεταξύ τους ή και με ανθρώπους (π.χ. ρομπότ που παίζουν πινγκ πονγκ). Επίσης, κατασκευάζονται ρομποτικά συστήματα που είναι σε θέση να *πλοηγούν* μη-επανδρωμένες διαστημικές αποστολές σε άγνωστα μέρη του Σύμπαντος για τον άνθρωπο. Γενικά, τα *ευφή αυτόνομα ρομποτικά συστήματα* προσφέρουν πληθώρα εφαρμογών και χρησιμοποιούνται κατά κόρον στις μέρες μας, κυρίως για ερευνητικούς σκοπούς.



Εικόνα 1.2: Space Robotics

1.2.2. Παιχνίδια (Computer Games)

Τα παιχνίδια αποτελούν ένα ενδιαφέρον πεδίο εφαρμογής της έννοιας του *ευφυούς* πράκτορα. Η σχεδίαση και ανάπτυξη αυτόνομων πρακτόρων που μπορούν να ανταποκριθούν ικανοποιητικά στις απαιτήσεις ενός παιχνιδιού, έχει αποκτήσει μεγάλη διάσταση τα τελευταία χρόνια. Εξάλλου, τα παιχνίδια είναι κατάλληλα για τέτοιου είδους εφαρμογές καθώς:

- προσφέρουν ιδανικά περιβάλλοντα δράσης, στα οποία η έννοια της επιτυχημένης ή αποτυχημένης ενέργειας είναι καλά ορισμένη (συνήθως αυτό καθορίζεται με μία *ανταμοιβή* (reward) ως αποτέλεσμα μίας συγκεκριμένης ενέργειας)
- είναι ιδανικά για την *ανάδειξη* και τη *βελτίωση* των δυνατοτήτων των μεθόδων της Τεχνητής Νοημοσύνης
- τα τελευταία χρόνια έχει παρατηρηθεί ραγδαία ανάπτυξη των παιχνιδιών. Η μελέτη και ανάπτυξη ευφών αυτόνομων πρακτόρων που ενεργούν στο περιβάλλον του παιχνιδιού δίνει την κατεύθυνση για τον εμπλουτισμό των δυνατοτήτων των παιχνιδιών, αναδεικνύοντας την σημαντική επίδραση της Τεχνητής Νοημοσύνης στον τομέα αυτό.

Για τη *σχεδίαση* και *ανάπτυξη* ενός ευφυούς πράκτορα που θα μπορεί να ενεργεί αυτόνομα μέσα σε ένα περιβάλλον παιχνιδιού, πρέπει να ληφθεί υπόψη η φύση του παιχνιδιού, με άλλα λόγια οι ιδιαιτερότητες και οι δυσκολίες που παρουσιάζει το παιχνίδι, οι οποίες έχουν τεθεί κατά τη σχεδίασή του. Με άλλα λόγια, ένας πράκτορας πρέπει:

- να ενεργεί με *αποδοτικό* τρόπο, εκμεταλλευόμενος όλους τους διαθέσιμους πόρους του παιχνιδιού
- να λαμβάνει υπόψη τις *απαιτήσεις του παιχνιδιού σε υπολογιστικούς πόρους* και το *κόστος διεξαγωγής πειραμάτων* να μην είναι ιδιαίτερα υψηλό
- να *αντιλαμβάνεται* το περιβάλλον του, να *λειτουργεί* και να *παίρνει αποφάσεις* σε πραγματικό χρόνο
- να παρουσιάζει *ευρωστία* και *προσαρμοστικότητα* σε *μεταβολές* του περιβάλλοντος (*εξαιρετικά συχνό φαινόμενο σε παιχνίδια*)

Όπως γίνεται αντιληπτό, η σχεδίαση και ανάπτυξη ευφυών πρακτόρων αποτελεί μία ιδιαίτερα δύσκολη και πολύπλοκη διαδικασία. Παρόλα αυτά, η δομή τους ακολουθεί ένα συγκεκριμένο σχήμα, το οποίο επαναλαμβάνεται για κάθε ενέργεια του πράκτορα: **ΑΝΤΙΛΗΨΗ – ΑΠΟΦΑΣΗ – ΕΚΤΕΛΕΣΗ – ΜΑΘΗΣΗ**.

- **ΑΝΤΙΛΗΨΗ:** Αρχικά, ο πράκτορας πρέπει να *αντιλαμβάνεται* το περιβάλλον του και να *αναπαριστά* με αποδοτικό τρόπο τη γνώση αυτή. Ανάλογα με τον τύπο του περιβάλλοντος, η αναπαράσταση καταστάσεων του πρέπει να οδηγεί στη **μέγιστη αξιοποίηση της πληροφορίας** που απορρέει από αυτό. Όμως, αυτό δεν είναι πάντα εφικτό. Σε κάποιες περιπτώσεις είναι δύσκολη η εξαγωγή της σωστής πληροφορίας, σε άλλες υπολογιστικά δαπανηρή διαδικασία. Με άλλα λόγια, αν και το περιβάλλον προσφέρει πληθώρα πληροφορίας είναι δύσκολη η διαδικασία εξαγωγής των χρήσιμων για τον πράκτορα πληροφοριών. Επίσης, *ζωτικής σημασίας* είναι η επιλογή της **δομής αναπαράστασης**
- **ΑΠΟΦΑΣΗ:** Αξιολογώντας τις πληροφορίες που απορρέουν από την παραπάνω *αναπαράσταση*, υπάρχει ένας *μηχανισμός απόφασης* που δίνει την ενέργεια που κρίνει ότι πρέπει να πραγματοποιηθεί, ώστε να πλησιάσει ο πράκτορας στον επιθυμητό στόχο. Ο μηχανισμός αυτός μπορεί να βασίζεται είτε σε κανόνες, είτε σε μεθόδους και αλγορίθμους μάθησης
- **ΕΚΤΕΛΕΣΗ:** Αποτελεί το στάδιο εκτέλεσης της ενέργειας που υπέδειξε ο μηχανισμός απόφασης του προηγούμενου βήματος. Το αποτέλεσμα αυτό είναι ορατό στον σχεδιαστή και δείχνει το επίπεδο ευφυΐας που έχει αναπτύξει ο πράκτορας μέχρι εκείνη τη χρονική στιγμή
- **ΜΑΘΗΣΗ:** Αποτελεί το στάδιο προσαρμογής και αυτο-οργάνωσης της γνώσης του πράκτορα. Με βάση την ανταμοιβή που απέδωσε η εκτέλεση της ενέργειας, εφαρμόζεται η διαδικασία μάθησης του μηχανισμού αποφάσεων. Με αυτόν τον τρόπο, ο πράκτορας ανακαλύπτει *πολιτικές (policies)* και *στρατηγικές (strategies)* που τον οδηγούν στην επίτευξη του στόχου του.

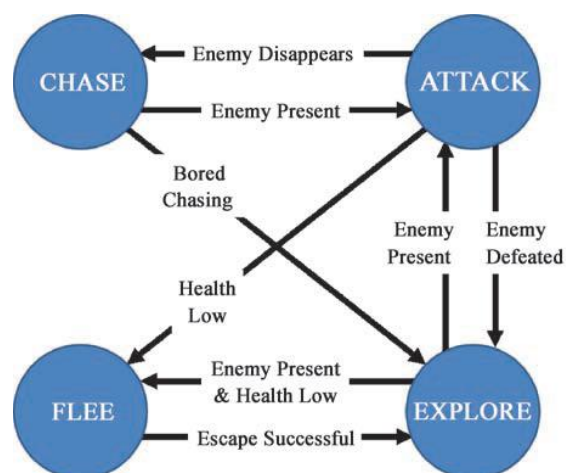
Τα τελευταία χρόνια έχουν προταθεί αρκετές εργασίες σχετικά την ανάπτυξη ευφυών πρακτόρων που δρουν αυτόνομα σε διάφορα περιβάλλοντα παιχνιδιών. Για παράδειγμα, στο [13] προτείνεται ένας ευφυής αυτόνομος πράκτορας, ο οποίος χρησιμοποιεί αλγορίθμους *Ενισχυτικής Μάθησης (Reinforcement Learning)* προκειμένου να λύνει επιτυχώς επίπεδα του δημοφιλούς *arcade* παιχνιδιού *Ms. Pac-Man*. Τέλος, στο [14] περιγράφεται ο πράκτορας *AngryBER* της παρούσας διατριβής, ο οποίος χρησιμοποιεί τεχνικές *Μηχανικής Μάθησης*, πιο συγκεκριμένα *Μάθησης με επίβλεψη (Supervised Learning)*, προκειμένου να καταφέρει να ολοκληρώνει επιτυχώς επίπεδα του παιχνιδιού *Angry Birds*.

1.2.3. *Non-Player Characters σε Computer Games*

Τα τελευταία χρόνια έχει παρατηρηθεί μία ραγδαία ανάπτυξη των *computer games*, κάτι που έχει ως συνέπεια οι απαιτήσεις των παικτών να μεγαλώνουν. Ο παίκτης αναμένει ότι οι *εικονικοί χαρακτήρες (virtual characters)* ενός παιχνιδιού ενεργούν ρεαλιστικά σε μεγάλο βαθμό, παίρνοντας ερεθίσματα από το περιβάλλον του παιχνιδιού. Οι περισσότεροι από τους εικονικούς χαρακτήρες ενός παιχνιδιού μοντελοποιούνται ως πράκτορες που λειτουργούν μέσα στο περιβάλλον του παιχνιδιού. Οι χαρακτήρες αυτοί πρέπει να αλληλεπιδρούν μεταξύ τους αλλά και με τον παίκτη. Ένα ακόμη στοιχείο που εξιτάρει τους παίκτες είναι η *απρόβλεπτη συμπεριφορά* των χαρακτήρων του παιχνιδιού. Ο συνδυασμός των δύο προαναφερθέντων στοιχείων αποτελεί μία μεγάλη πρόκληση στο σχεδιασμό πρακτόρων που ελέγχουν τους χαρακτήρες του παιχνιδιού που ονομάζονται *non-player characters (NPCs)* και δρουν αντιλαμβανόμενοι το περιβάλλον του παιχνιδιού, αλληλεπιδρώντας μεταξύ τους αλλά και με τον ίδιο τον παίκτη. Πράκτορες ελέγχουν τους *NPCs* σε πολλά είδη παιχνιδιών (*action games, strategy games, role playing games, sports games*), τα οποία όμως πρέπει να τρέχουν σε πραγματικό χρόνο. Αυτό σημαίνει ότι η υπολογιστική ισχύς και πολυπλοκότητα που απαιτούνται για τη δράση των *NPCs* του παιχνιδιού πρέπει να κυμαίνονται σε ασφαλή επίπεδα, καθώς ήδη πολλούς από τους υπολογιστικούς πόρους που απαιτούνται για ένα παιχνίδι καταναλώνονται από τα γραφικά [6].

1.2.4. Υλοποίηση Πρακτορικών NPCs

Στην πράξη, η συμπεριφορά και η δράση των πρακτόρων που ελέγχουν τους NPCs ενός παιχνιδιού πρέπει να είναι πολύπλευρη και πολυδιάστατη. Μία ευρέως διαδομένη τεχνική η οποία μπορεί να καθορίσει αποτελεσματικά τη συμπεριφορά ενός NPC μέσα σε ένα παιχνίδι είναι η *Finite State Machine* (Σχήμα 1.7). Μία FSM είναι ένας απλός *συνδεδεμένος γράφος* με πεπερασμένο αριθμό κόμβων και ακμών. Οι κόμβοι αντιστοιχούν στις *καταστάσεις* στις οποίες μπορεί να βρεθεί ο πράκτορας, ενώ οι ακμές δίνουν τη *μετάβαση* σε μία άλλη κατάσταση, δείχνοντας ποιες ενέργειες του πράκτορα ή αλλαγές του περιβάλλοντος είναι ικανές να αλλάξουν την κατάσταση του πράκτορα. Η FSM είναι ευρέως διαδεδομένη τεχνική, καθώς είναι πολύ απλή στον προγραμματισμό, κατανοητή, και συγχρόνως καθόλου απαιτητική σε σχέση με τους υπολογιστικούς πόρους που καταναλώνει. Επίσης ευρεία εφαρμογή έχουν οι *παράλληλες FSMs* (*parallel FSMs*), οι οποίες αποτελούνται από επίπεδα καθένα από τα οποία λειτουργεί ως FSM και σε κάποιες περιπτώσεις συνεργάζεται και με τα υπόλοιπα επίπεδα [6]. Οι FSMs όμως έχουν και τα μειονεκτήματά τους. Κατά την κατασκευή FSMs, οι σχεδιαστές πρέπει να προβλέψουν κάθε δυνατή κατάσταση με την οποία θα βρεθεί αντιμέτωπος ένας NPC στα πλαίσια του παιχνιδιού, κάτι που δεν είναι εφικτό για όλα τα παιχνίδια, λόγω πολύπλοκων καταστάσεων που έχει ως συνέπεια την αύξηση της πολυπλοκότητας του γράφου. Αυτό έρχεται να ενισχύσει την πεποίθηση ότι βασιζόμενοι μόνο σε FSMs, η αλληλεπίδραση μεταξύ των NPCs και του παίκτη γίνεται *μονότονη* καθώς φαίνεται να βασίζεται σε κανόνες εκδηλώνοντας *ντετερμινιστική συμπεριφορά*, στοιχείο που απωθεί τον παίκτη από τέτοιου είδους παιχνίδια.



Σχήμα 1.7: Finite State Machine Ενός NPC-Στρατιώτη σε Ένα Παιχνίδι Δράσης

Λαμβάνοντας υπόψη τα παραπάνω γίνεται αντιληπτό ότι η συμπεριφορά και η αλληλεπίδραση των *NPCs* μεταξύ τους και με τον παίκτη σε κάθε παιχνίδι μπορούν να κάνουν το παιχνίδι δημοφιλές. Η ντετερμινιστική συμπεριφορά κουράζει τους παίκτες, ενώ η απρόβλεπτη συμπεριφορά των *NPCs* τους εξιτάρει. Αλγόριθμοι και μέθοδοι Τεχνητής Νοημοσύνης κατά τη σχεδίαση και υλοποίηση των *NPCs* μπορούν να εισάγουν το *απρόβλεπτο* στη συμπεριφορά των *NPCs*, με την έννοια ότι θα μπορούν να αντιλαμβάνονται το περιβάλλον τους και να δρουν με βάση τη γνώση που θα οικοδομήσουν λαμβάνοντας υπόψη το αποτέλεσμα των ενεργειών τους, αλλά και τις αλλαγές του περιβάλλοντος. Βέβαια, απαιτείται η υλοποίησή τους σε πραγματικό χρόνο, άρα υπολογιστική πολυπλοκότητα σε λογικά επίπεδα.

1.2.5. Γραφικά (Animation)

Σε κάθε ταινία, κυρίως σε πολεμικές όπου υπάρχουν εκατοντάδες χιλιάδες χαρακτήρες, η μοντελοποίηση κάθε χαρακτήρα με ένα πράκτορα που δρα με βάση τη γνώση του, προσφέρει σημαντικές λύσεις, κυρίως στον οικονομικό τομέα. Τέτοιου είδους πράκτορες λαμβάνουν γνώση για το περιβάλλον της ταινίας μέσω απλών τεχνικών προσομοίωσης όρασης, ακοής και αφής [6]. Χρησιμοποιώντας τα *ερεθίσματα* αυτά αποφασίζουν ποια είναι η καλύτερη ενέργεια (*action*) που μπορούν να εκτελέσουν. Μπορεί η σχεδίαση τέτοιων πρακτόρων να φαίνεται πολύπλοκη, αλλά λαμβάνοντας υπόψη την ευκολία με την οποία προγραμματίζονται για διαφορετικές σκηνές της ταινίας (π.χ. για σκηνές μάχης ή για σκηνές κάποιας γιορτής) αντιλαμβανόμαστε το λόγο για τον οποίο χρησιμοποιούνται ευρέως σε ταινίες με πολλούς εικονικούς χαρακτήρες.

Τα επιτεύγματα της χρήσης πρακτόρων σε ταινίες για την μοντελοποίηση εικονικών χαρακτήρων είναι πράγματι εντυπωσιακά, αλλά πρέπει να σημειώσουμε ότι δεν μπορούν να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου (*real-time applications*). Για παράδειγμα, η σκηνή της μάχης που διεξάγεται στην αρχή της ταινίας “*Lord of the Rings: The Fellowship of the Ring*” σχεδιάστηκε από υπολογιστικά συστήματα μεγάλης ισχύος, αλλά για την παραγωγή του τελικού αποτελέσματος της σκηνής απαιτήθηκε μία ολόκληρη εβδομάδα [6].

1.3. Αντικείμενο Διατριβής

Από τη δημιουργία του μέχρι σήμερα, το παιχνίδι *Angry Birds* αποτελεί αντικείμενο συζήτησης για πολλούς ανθρώπους σε όλο τον κόσμο. Είναι γενικά αποδεκτό ότι αποτελεί ένα σημαντικό πεδίο μελέτης και έρευνας, κυρίως σε ότι αφορά την ανάπτυξη ευφυών αυτόνομων πρακτόρων (*AI agents*) που είναι τελικά σε θέση να παίζουν το παιχνίδι *Angry Birds* χωρίς ανθρώπινη παρέμβαση. Τέτοιου είδους ευφυείς πράκτορες μπορούν να κατασκευαστούν χρησιμοποιώντας στοιχεία από διάφορους τομείς της *Τεχνητής Νοημοσύνης*, όπως είναι η *Υπολογιστική Όραση* (*Computer vision*), η *Αναπαράσταση Γνώσης* (*Knowledge Representation*), ο *Σχεδιασμός* (*Planning*), η *Μηχανική Μάθηση* (*Machine Learning*) κ.α.

Όλα τα παραπάνω, σε συνδυασμό με την πολύπλευρη και πολύπλοκη φύση του παιχνιδιού *Angry Birds* (πολύπλοκος χώρος καταστάσεων και δράσεων, πολλαπλές ιδιότητες των αντικειμένων της σκηνής, ειδικές ικανότητες του κάθε πουλιού με βάση το *tap*, εναρμονισμένοι νόμοι της Φυσικής με τα επίπεδα του παιχνιδιού, μη ακριβής αντίχνευση των αντικειμένων και πουλιών του παιχνιδιού) οδήγησε στη δημιουργία του διαγωνισμού *Angry Birds AI Competition*¹, όπου τα τελευταία 2-3 χρόνια έχουν προταθεί αρκετές ενδιαφέρουσες μελέτες και δουλειές σχετικά με την δημιουργία *AI agents* που μπορούν να δώσουν καλύτερα αποτελέσματα από τον ίδιο τον άνθρωπο, αντιμετωπίζοντας βέβαια άγνωστα σε αυτόν επίπεδα.

Από τους διοργανωτές του διαγωνισμού παρέχεται ένα ελεύθερο *basic playing game software* [18], το οποίο είναι υλοποιημένο σε *Java* και προσφέρει τις ακόλουθες δυνατότητες:

- *ανάλυση της σκηνής του κάθε επιπέδου*
- *ανάλυση και υπολογισμό της τροχιάς των πουλιών για ένα συγκεκριμένο στόχο από τη σκηνή*
- *εκτέλεση ενεργειών και λήψης στιγμιότυπων της σκηνής για περαιτέρω ανάλυση*
- *plugin για αλληλεπίδραση με τον browser Chrome της Google*

¹ <https://aibirds.org/>

Στην παρούσα διατριβή, και έχοντας ως βάση το λογισμικό που περιγράφηκε πριν, γίνεται μία μελέτη του περιβάλλοντος του δημοφιλούς παιχνιδιού *Angry Birds* και προτείνεται μία μέθοδος επίλυσής του που βασίζεται στην υλοποίηση του πράκτορα *AngryBER* (*Angry-Bayesian Ensemble Regression*), ο οποίος χρησιμοποιώντας τεχνικές Μηχανικής Μάθησης (*Machine Learning*) καλείται να κατανοήσει το συγκεκριμένο περιβάλλον και στη συνέχεια με μία *online* διαδικασία μάθησης να ανακαλύψει τις ιδιότητες των αντικειμένων του παιχνιδιού αλλά και την αλληλεπίδραση που έχουν μεταξύ τους.

Η αναπαράσταση της σκηνής του παιχνιδιού κρίνεται *ιδιαίτερα κομβική*, καθώς ο χώρος καταστάσεων του παιχνιδιού είναι *συνεχής* και η εύρεση μίας δομής αναπαράστασης που θα εκμεταλλεύεται πλήρως τη σκηνή είναι *ζωτικής σημασίας*. Η δενδρική αναπαράσταση που προτείνουμε στα πλαίσια της παρούσας διατριβής για την απεικόνιση της σκηνής του παιχνιδιού, χτίζεται *ιεραρχικά* (*bottom-up*) και εκμεταλλεύεται τις τοπολογικές εξαρτήσεις μεταξύ των αντικειμένων, ενώ ταυτόχρονα έχει χαμηλή πολυπλοκότητα, η οποία επισπεύδει τη διαδικασία επιλογής του κατάλληλου στόχου και την μετέπειτα διαδικασία μάθησης.

Μετά την κατασκευή της δενδρικής αναπαράστασης της σκηνής ακολουθεί η επιλογή του στόχου. Ο στόχος επιλέγεται με ένα μηχανισμό *Μπεϋζιανής Ensemble Παλινδρόμησης*, ο οποίος επιλέγει το στόχο συνδυάζοντας πολλαπλά μοντέλα μάθησης, τα οποία δρουν ανταγωνιστικά μεταξύ τους (*competitive fashion*) [14] και δίνουν την αναμενόμενη πρόβλεψη για κάθε δυνατό στόχο. Επιλέγεται ο στόχος που αναμένεται να δώσει την *μεγαλύτερη ανταμοιβή* (*σκορ*). Τέλος, λαμβάνει χώρα η *online* διαδικασία μάθησης για τον πράκτορα *AngryBER*, όπου ενημερώνονται οι παράμετροι του μοντέλου που υπέδειξε τον στόχο.

1.4. Διάρθρωση Διατριβής

Η παρούσα διατριβή αποτελείται από πέντε κεφάλαια. Στο παρόν κεφάλαιο, αρχικά δόθηκαν οι έννοιες του ευφυούς πράκτορα, του περιβάλλοντός του. Επίσης, περιγράφηκαν δημοφιλή πεδία εφαρμογής των ευφών αυτόνομων πρακτόρων. Στο Κεφάλαιο 2 δίνεται μία αναλυτική περιγραφή του παιχνιδιού *Angry Birds*, το οποίο

αποτελεί πεδίο εφαρμογής μεθόδων Τεχνητής Νοημοσύνης. Επίσης, παρουσιάζονται κάποιες εργασίες που έχουν εκπονηθεί και αφορούν τη μελέτη του περιβάλλοντος του παιχνιδιού και την ανάπτυξη ευφυών αυτόνομων πρακτόρων για το περιβάλλον αυτό. Στο Κεφάλαιο 3, δίνεται αναλυτικά η μέθοδος που προτείνουμε στα πλαίσια της παρούσας διατριβής. Αρχικά, περιγράφεται λεπτομερώς η δενδρική αναπαράσταση της σκηνής του παιχνιδιού που προτείνουμε, στη συνέχεια η διαδικασία επιλογής του στόχου για κάθε βολή μέσα από ένα μηχανισμό *Μπεϋζιανής Ensemble Παλινδρόμησης* και τέλος, η *online διαδικασία μάθησης (εκπαίδευσης)* του πράκτορα *AngryBER*. Στο Κεφάλαιο 4 υλοποιούνται δύο σειρές πειραμάτων στην προσπάθειά μας να αξιολογήσουμε την επίδοση του πράκτορα *AngryBER*, ενώ δίνονται τα αποτελέσματα του *Angry Birds AI Competition 2014*, στον οποίο ο πράκτορας *AngryBER* κατάφερε να φτάσει μέχρι την Τελική Φάση του διαγωνισμού. Κλείνοντας, στο Κεφάλαιο 5 με βάση τα πειραματικά αποτελέσματα αλλά και την προσεκτική παρατήρηση της συμπεριφοράς του πράκτορα *AngryBER* κατά τη διεξαγωγή τους, προσπαθούμε να εξάγουμε κάποια συμπεράσματα αναφορικά με την επίδοση και τις ιδιαιτερότητές του. Τέλος, δίνουμε κάποιες κατευθύνσεις για μελλοντική επέκταση και βελτίωση της προτεινόμενης μεθόδου.

ΚΕΦΑΛΑΙΟ 2. ΤΟ ΠΑΙΧΝΙΔΙ “ANGRY BIRDS”

2.1 Ιστορική Αναδρομή και Εξέλιξη

2.2 Περιγραφή του Παιχνιδιού

2.3 Η Σκηνή του Παιχνιδιού

2.4 Το Angry Birds ως Πεδίο Εφαρμογής Μεθόδων Τεχνητής Νοημοσύνης

Στο κεφάλαιο αυτό παρουσιάζεται το παιχνίδι “*Angry Birds*”. Αρχικά γίνεται μία ιστορική αναδρομή και παρουσιάζεται η σημαντική εξέλιξη του παιχνιδιού μέχρι τις μέρες μας. Στη συνέχεια δίνεται η περιγραφή του παιχνιδιού, κατά την οποία παρουσιάζονται οι ιδιαιτερότητές του και γίνεται ανάλυση του περιβάλλοντός του παρουσιάζοντας παραδείγματα και τους κύριους χαρακτήρες του παιχνιδιού. Ακολούθως, γίνεται μία αναφορά στις προκλήσεις που απορρέουν από τη φύση του παιχνιδιού και αφορούν την σχεδίαση και ανάπτυξη ευφώνων αυτόνομων πρακτόρων για το συγκεκριμένο περιβάλλον. Τέλος, γίνεται μία αναφορά σε ενδιαφέρουσες μελέτες που πραγματεύονται θέματα σχετικά με τη φύση του παιχνιδιού “*Angry Birds*”.

2.1. Ιστορική Αναδρομή και Εξέλιξη

Το *Angry Birds* είναι ένα δημοφιλές βιντεοπαιχνίδι, το οποίο δημιουργήθηκε από την φιλανδική εταιρεία *Rovio Entertainment (TM)*. Η ιδέα κατασκευής του παιχνιδιού ξεκίνησε από ένα σχέδιο πουλιών, τα οποία πετούν χάρη στη σφεντόνα που υπάρχει σε κάθε επίπεδο του παιχνιδιού. Αρχικά, κυκλοφόρησε για το λογισμικό *iOS* της Apple πριν από πέντε περίπου χρόνια, συγκεκριμένα το Δεκέμβριο του 2009. Από τότε, το ηλεκτρονικό κατάστημα *App Store* της Apple έχει διαθέσει στην αγορά πάνω από δώδεκα (12) εκατομμύρια αντίγραφα του παιχνιδιού, γεγονός που οδήγησε την

εταιρεία να σχεδιάσει και να κατασκευάσει αντίστοιχες εκδόσεις του παιχνιδιού και για smartphones τα οποία χρησιμοποιούν το λειτουργικό *Android*.

Το *Angry Birds* έγινε ιδιαίτερα δημοφιλές λόγω του συνδυασμού εθισμού, κωμικού χαρακτήρα και χαμηλής τιμής που το χαρακτηρίζει, ενώ είναι αξιοσημείωτο ότι προσελκύει ανθρώπους όλων των ηλικιών. Η διασημότητα και η μεγάλη αναγνωρισιμότητα του παιχνιδιού «ανάγκασε» την εταιρεία να προβεί στη σχεδίαση εκδόσεων του παιχνιδιού για ηλεκτρονικούς υπολογιστές και παιχνιδοκονσόλες, ενώ ήδη υπάρχει μία τηλεοπτική σειρά βασισμένη στα *Angry Birds*, με μακροπρόθεσμα σχέδια για τη δημιουργία και προβολή μίας κινηματογραφικής ταινίας. Όλα τα παραπάνω σε συνδυασμό με το ένα και πλέον δισεκατομμύριο *downloads* του παιχνιδιού για κινητά τηλέφωνα, ηλεκτρονικούς υπολογιστές και παιχνιδοκονσόλες, έχουν ως συνέπεια το παιχνίδι *Angry Birds* να χαρακτηρίζεται ως *το πιο διάσημο βιντεοπαιχνίδι* στις μέρες μας, ενώ εμπλουτίζεται συνεχώς με καινούριο περιεχόμενο και νέες αναβαθμίσεις καθώς και ανεξάρτητες εποχιακές εκδόσεις του παιχνιδιού ³.

2.2. Περιγραφή του Παιχνιδιού

Στο παιχνίδι *Angry Birds*, σκοπός του παίκτη είναι να εξολοθρεύσει όλα τα γουρούνια κάθε επιπέδου, τα οποία βρίσκονται προστατευμένα σε διάφορες πολύπλοκες δομές που αποτελούνται από διάφορα υλικά. Για να το πετύχει αυτό, ο παίκτης έχει στη διάθεσή του ένα συγκεκριμένο αριθμό πουλιών σε κάθε επίπεδο, καθώς και μία σφεντόνα. Χρησιμοποιώντας τη σφεντόνα, εκτοξεύει τα πουλιά προς τον στόχο που αυτός κρίνει ότι είναι κατάλληλος, είτε πρόκειται για γουρούνι που μπορεί χτυπηθεί άμεσα, είτε πρόκειται για κάποιο άλλο αντικείμενο της σκηνής. Η καταστροφή πολλαπλών αντικειμένων μπορεί να προκαλέσει την εξολόθρευση των γουρουνιών μέσα από την κατάρρευση πολύπλοκων δομών. Επίσης, σε πολλά επίπεδα του παιχνιδιού βρίσκονται επιπρόσθετα αντικείμενα, όπως εκρηκτικά κιβώτια (TNTs) και στρόγγυλες πέτρες ή ξύλα που βρίσκονται ανάμεσα στις πολύπλοκες δομές που προστατεύουν τα γουρούνια και μπορούν να χρησιμοποιηθούν σε συνδυασμό με τα διαθέσιμα πουλιά ώστε να εξολοθρευτούν τα προστατευμένα γουρούνια.

³ <http://en.wikipedia.org/>

Υπάρχουν πουλιά με διαφορετικές μορφές και ιδιότητες που χρησιμοποιούνται κατά τη διάρκεια του παιχνιδιού. Στα αρχικά επίπεδα λίγοι τύποι πουλιών είναι διαθέσιμοι. Όμως, όσο ο παίκτης προχωρά ολοκληρώνοντας επιτυχώς κάθε επίπεδο του παιχνιδιού, όλο και περισσότερα πουλιά γίνονται διαθέσιμα. Είναι απαραίτητο να σημειωθεί, ότι κάθε τύπος πουλιού έχει συγκεκριμένες ιδιότητες, οι οποίες ενεργοποιούνται από τον παίκτη τη στιγμή που κρίνει ο ίδιος, και έχει μεγαλύτερη επίδραση σε συγκεκριμένα υλικά. Επίσης υπάρχουν συγκεκριμένοι τύποι γουρουνιών, των οποίων η αντοχή έχει να κάνει με το κυρίως με το μέγεθός τους.

Τέλος, κάθε επίπεδο ξεκινά με προκαθορισμένο αριθμό τύπο και σειρά πουλιών. Εάν όλα τα γουρούνια εξολοθρευτούν μέχρι τη χρησιμοποίηση του τελευταίου πουλιού τότε το επίπεδο ολοκληρώνεται επιτυχώς και το επόμενο επίπεδο ξεκλειδώνεται. Η εξολόθρευση των γουρουνιών, καθώς και η καταστροφή αντικειμένων της σκηνής δίνουν πολλούς πόντους στον παίκτη, ενώ για κάθε πουλί που δεν έχει χρησιμοποιηθεί ενώ το επίπεδο έχει ολοκληρωθεί επιτυχώς, ο παίκτης καρπώνεται +10000 πόντους στο σκορ του για το συγκεκριμένο επίπεδο ². Αξίζει να σημειωθεί ότι κάθε επίπεδο έχει διαφορετικές δομές, αριθμό γουρουνιών και αντικειμένων άρα και διαφορετικό μέγιστο αριθμό πόντων που μπορεί να συγκεντρώσει ο παίκτης. Ο παίκτης μπορεί να ξαναπαίξει όσες φορές θέλει ένα επίπεδο το οποίο έχει προηγουμένως ολοκληρώσει είτε για να το ολοκληρώσει επιτυχώς, είτε για να βελτιώσει το σκορ του στο συγκεκριμένο επίπεδο.

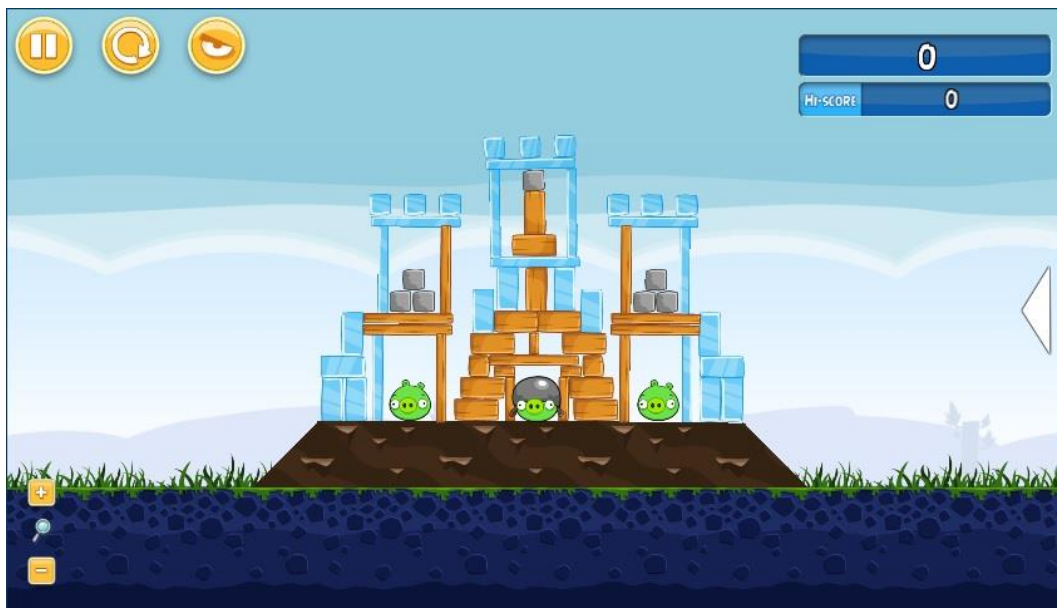
2.3. Η Σκηνή του Παιχνιδιού

Στην ενότητα αυτή περιγράφεται αναλυτικότερα η *σκηνή* του παιχνιδιού, ενώ δίνονται κάποιες εικόνες με χαρακτηριστικά παραδείγματα επιπέδων του παιχνιδιού που αναδεικνύουν την πολύπλοκη φύση του, και δείχνουν ότι απαιτείται αυξημένη δεξιότητα αλλά και λογική για ολοκληρώσει κάποιος επιτυχώς ένα επίπεδο, πόσο μάλλον ένα επεισόδιο. Τα βασικά υλικά που χρησιμοποιούνται για την κατασκευή των δομών που προστατεύουν τα γουρούνια είναι πέτρες, διαφόρων μεγεθών και σχημάτων, ξύλα, πάγος, ενώ υπάρχουν και οι λόφοι, *hills*, που προσδίδουν ακόμη μεγαλύτερη δυσκολία εάν προστατεύουν τις δομές στις οποίες είναι κρυμμένα τα γουρούνια, καθώς τα πουλιά δεν έχουν καμία επίδραση πάνω τους. Όμως, υπάρχουν

² <http://www.angrybirdsnest.com/>

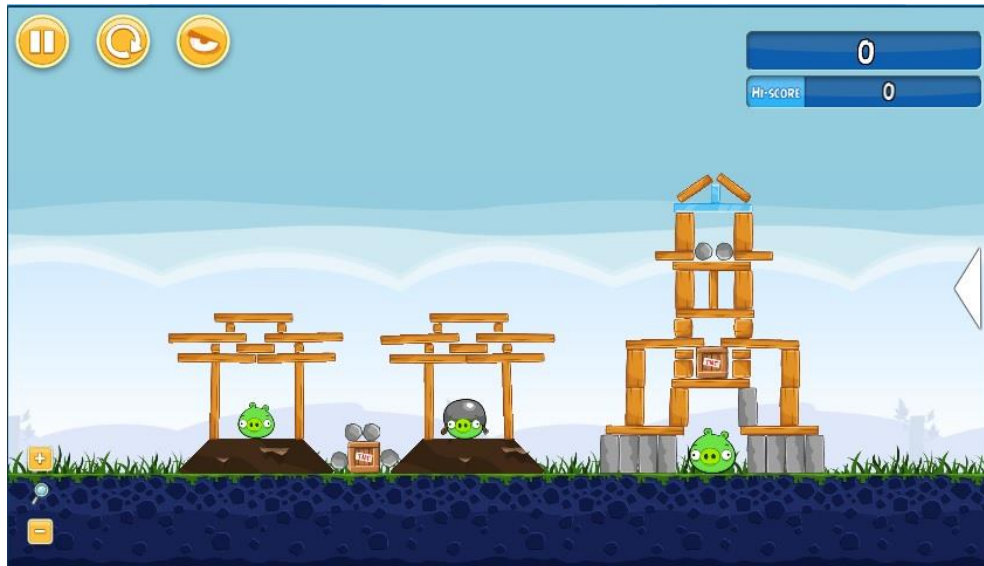
και υλικά που βοηθούν τον παίκτη, όπως τα *εκρηκτικά κουτιά*, ή αλλιώς *TNTs*, τα οποία είναι συνήθως τοποθετημένα σε καίρια σημεία και ενδεχόμενη πρόσκρουση ενός πουλιού πάνω τους προκαλεί ένα καίριο χτύπημα σε κάθε υλικό που βρίσκεται κοντά του.

Όλα τα επίπεδα είναι σχεδιασμένα έτσι ώστε να δυσκολεύουν στο μέγιστο δυνατό τον παίκτη, ενώ η φύση του παιχνιδιού με τη γωνία εκτόξευσης από τη σφεντόνα, τις ταχύτητες (οριζόντιες και κάθετες) που αναπτύσσονται κατά τη διάρκεια της βολής κάνουν το *Angry Birds* ένα εκρηκτικό μίγμα *λογικής, δεξιάτητας και Φυσικής*. Στην Εικόνα 2.1 παρουσιάζεται το επίπεδο 21 του επεισοδίου “*Poached Eggs*”.



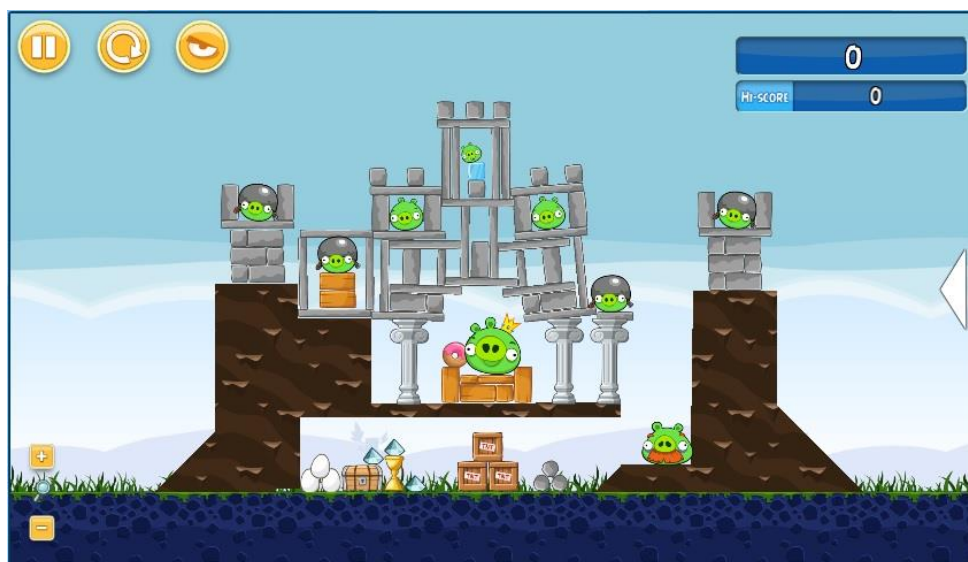
Εικόνα 2.1: Το Επίπεδο 21 του Επεισοδίου “*Poached Eggs*”

Όπως μπορούμε να παρατηρήσουμε, η σκηνή αυτή είναι πολύ προσεκτικά δομημένη έτσι ώστε τα γουρούνια να μένουν εκτεθειμένα μόνο με προσεκτικές και στοχευμένες βολές, χρησιμοποιώντας βέβαια πάντα και τις ειδικές ικανότητες του κάθε πουλιού. Στην Εικόνα 2.1 μπορούμε να παρατηρήσουμε τα ξύλα, τον πάγο, τις πέτρες καθώς και τον τύπο γουρουνιού με το πέτρινο κράνος. Ένα άλλο παράδειγμα ενός επιπέδου του επεισοδίου “*Poached Eggs*” δίνεται στην Εικόνα 2.2.



Εικόνα 2.2: Το Επίπεδο 37 του Επεισοδίου “Poached Eggs”

Όπως βλέπουμε υπάρχουν δύο TNTs σε καίριες θέσεις της σκηνής που μπορούν να προκαλέσουν μεγάλη καταστροφή με την έκρηξή τους και να αφήσουν τα γουρούνια εκτεθειμένα στον παίκτη. Επιπλέον, στην Εικόνα 2.3 παρουσιάζεται το τελευταίο επίπεδο του επεισοδίου “Poached Eggs”.



Εικόνα 2.3: Το Επίπεδο 63 του Επεισοδίου “Poached Eggs”

Τέλος, αξίζει να σημειωθεί ότι εκτός από την πολύπλοκη φύση των δομών που προστατεύουν τα γουρούνια, ιδιαίτερα σημαντικό ρόλο παίζει το είδος των πουλιών που έχει στη διάθεσή του ο παίκτης, όπως και η σειρά με την οποία αυτά είναι διαθέσιμα για την εκτέλεση της εκάστοτε βολής.

2.3.1. Τύποι Πουλιών

Υπάρχουν πέντε τύποι πουλιών που χρησιμοποιούνται στα επίπεδα με τα οποία ασχοληθήκαμε στα πλαίσια αυτής της εργασίας. Κάθε τύπος πουλιού έχει συγκεκριμένα χαρακτηριστικά και ειδικές ικανότητες και αφορούν την επίδραση που έχουν σε αντικείμενα της σκηνής του παιχνιδιού. Σημαντικό ρόλο παίζει το λεγόμενο *tap*, στην περίπτωση των smartphones ένα άγγιγμα της οθόνης, ενώ στην περίπτωση των ηλεκτρονικών υπολογιστών ένα κλικ του ποντικιού, και αποτελεί την ενέργεια που εκτελεί ο παίκτης ώστε να ενεργοποιηθεί η ειδική ικανότητα του πουλιού που βρίσκεται στη σφεντόνα. Αναλυτικότερα:

- *Red bird or “Red”*: Αποτελεί το πιο αναγνωρίσιμο πουλί του παιχνιδιού, καθώς κοσμεί και το λογότυπό του. Είναι διαθέσιμο από τα πρώτα επίπεδα του παιχνιδιού, όπου μπορεί να προκαλέσει μεγάλη ζημιά σε αδύναμες δομές που προστατεύουν τα γουρούνια. Δεν έχει σημαντικές ιδιότητες, ούτε μεγάλη επίδραση σε συγκεκριμένα υλικά και αυτό φαίνεται όσο προχωρά το παιχνίδι και οι δομές που προστατεύουν τα γουρούνια γίνονται πιο περίπλοκες. Πρέπει να χρησιμοποιείται έξυπνα από τον παίκτη για να καταστρέφει μέρος των πολύπλοκων δομών και να απλοποιεί την δυσκολία τους για τα υπόλοιπα πουλιά.



Εικόνα 2.4: “The Red”

- *Blue birds or “The Blues”*: Ο συγκεκριμένος τύπος πουλιού είναι διαθέσιμος στον παίκτη από το 10^ο επίπεδο του επεισοδίου “*Poached Eggs*” και έχει την ιδιότητα να χωρίζεται σε τρία πουλιά μόλις ο παίκτης κάνει tap. Είναι ιδιαίτερα αποτελεσματικός σε δομές που περιέχουν πάγο, ενώ αξίζει να σημειωθεί ότι το tap πρέπει να γίνεται κοντά στον στόχο όταν ο παίκτης θέλει να καταστρέψει συμπαγείς και ογκώδεις δομές, ενώ μακριά από τον στόχο όταν πρέπει να καταστρέψει δομές που καταλαμβάνουν μεγάλο μέρος της σκηνής.



Εικόνα 2.5: “The Blues”

- *Yellow bird or “Chuck”*: Ο συγκεκριμένος τύπος πουλιού είναι διαθέσιμος στον παίκτη από το 16^ο επίπεδο του επεισοδίου “*Poached Eggs*” του παιχνιδιού και έχει την ιδιότητα να επιταχύνει, μόλις ο παίκτης εκτελέσει το tap. Είναι ιδανικός στην καταστροφή ξύλινων δομών, ειδικά όταν πρόκειται για ξύλα με μικρό πάχος και μεγάλο ύψος ακόμη και όταν βρίσκονται τέσσερις ή και πέντε τέτοιου είδους δομές στη σειρά. Δεν έχει μεγάλη επίδραση σε δομές πάγου, ενώ έχει μηδαμινή επίδραση σε πέτρινες δομές. Επίσης έχει την ιδιότητα να ακολουθεί ευθεία πορεία από τη στιγμή που ο παίκτης εκτελέσει το tap και μέχρι να προσκρούσει σε κάποιο αντικείμενο της σκηνής.



Εικόνα 2.6: “The Chuck”

- *Black bird or “The Bomb”*: Πρόκειται για το πιο δυνατό τύπο πουλιού του παιχνιδιού και γίνεται διαθέσιμος στον παίκτη από το 23^ο επίπεδο του επεισοδίου “*Poached Eggs*”. Έχει την ιδιότητα να εκρήγνυται, είτε μόλις ο παίκτης εκτελέσει το tap, είτε δύο περίπου δευτερόλεπτα μετά την πρόσκρουσή του σε κάποιο αντικείμενο της σκηνής. Επειδή πρόκειται για βαρύ και ογκώδες πουλί, έχει καταστροφική επίδραση σε πέτρινες και ογκώδεις δομές, ενώ η αντίστοιχη επίδραση μειώνεται αισθητά σε ξύλινες ή δομές πάγου.



Εικόνα 2.7: “The Bomb”

- *White bird or “Matilda”*: Πρόκειται για ένα ιδιαίτερο τύπου πουλιού, με παράξενες ιδιότητες. Γίνεται διαθέσιμος στον παίκτη από το 35^ο επίπεδο του επεισοδίου “*Poached Eggs*”. Έχει την ιδιότητα να πετά *Egg bombs* μόλις ο παίκτης εκτελέσει το tap για να καταστρέφει αντικείμενα της σκηνής, και αμέσως μετά εκσφενδονίζεται με κατεύθυνση διαγώνια πάνω δεξιά από το σημείο που έγινε το tap. Πρόκειται για ογκώδες και βαρύ τύπο πουλιού, μπορεί να συγκριθεί με το *Black bird* σε μέγεθος και βάρος, και είναι ιδιαίτερα αποτελεσματικό σε πέτρινες δομές. Μπορεί να χρησιμοποιηθεί κατάλληλα, ώστε η δύναμη που αποκτά αμέσως μετά το tap να έχει επίδραση σε ογκώδεις δομές της σκηνής.



Εικόνα 2.8: “Matilda”

2.3.2. Τύποι Γουρουνιών

Τέλος, υπάρχουν συγκεκριμένοι τύποι γουρουνιών, των οποίων η αντοχή έχει να κάνει με το κυρίως με το μέγεθός τους ή με συγκεκριμένα χαρακτηριστικά τους όπως το *πέτρινο κράνος*, το *μουστάκι* ή το *στέμμα* που μερικά από αυτά έχουν τοποθετημένα στο κεφάλι τους και τα κάνει πιο ανθεκτικά σε συγκρούσεις τόσο με τα πουλιά όσο και με αντικείμενα της σκηνής. Στην Εικόνα 2.9 παρουσιάζονται οι τύποι γουρουνιών που συναντήσαμε στα πλαίσια αυτής της εργασίας.



Εικόνα 2.9: Τα Γουρούνια του Παιχνιδιού με Αύξουσα Σειρά Ανθεκτικότητας

2.4. Το Angry Birds ως Πεδίο Εφαρμογής Μεθόδων Τεχνητής Νοημοσύνης

Τα τελευταία χρόνια, το “Angry Birds” αποτελεί αντικείμενο συζήτησης στην επιστημονική κοινότητα που ασχολείται με θέματα *Τεχνητής Νοημοσύνης*. Η *σχεδίαση* και *ανάπτυξη* ευφών πρακτόρων για το παιχνίδι *Angry Birds* αποτελεί μία επίπονη και χρονοβόρα διαδικασία, κυρίως λόγω της πολύπλοκης φύσης του. Πιο συγκεκριμένα, κατά τη σχεδίαση ενός πράκτορα πρέπει να ληφθούν υπόψη τα εξής σημαντικά ζητήματα, τα οποία, κατά κύριο λόγο, εστιάζουν στο *περιβάλλον* του παιχνιδιού:

- *πολύπλοκος χώρος καταστάσεων και ενεργειών*
- *θορυβώδης αντίληψη αντικειμένων (πιθανότητα σφαλμάτων)*
- *εύρεση αποδοτικής και κατάλληλης αναπαράστασης της σκηνής*
- *πολλές στρατηγικές για την επιλογή του tap time, το οποίο εξαρτάται από τον τύπο του πουλιού και της τροχιάς*
- *διαφορετικές ιδιότητες και ειδικά χαρακτηριστικά αντικειμένων και πουλιών*
- *ανακρίβειες που διέπουν τα Φυσικά μοντέλα κατά τη διάρκεια των βολών που προσομοιώνονται στα πλαίσια του παιχνιδιού*
- *διαφορετικές δομές και ιδιαιτερότητες κάθε επιπέδου του παιχνιδιού.*

Τα παραπάνω σημαντικά στοιχεία καθιστούν το *Angry Birds* μία μεγάλη πρόκληση για τον τομέα της Τεχνητής Νοημοσύνης στα πλαίσια της ανάπτυξης *ευφρών αυτόνομων πρακτόρων* για το παιχνίδι. Για το λόγο αυτό, τα τελευταία χρόνια διεξάγεται ο διαγωνισμός *Angry Birds AI Competition*. Οι διοργανωτές του προτρέπουν επιστημονικές ομάδες από όλο τον κόσμο να αναπτύξουν και να παρουσιάσουν πράκτορες, με στόχο:

- σε πρώτο στάδιο να ανταποκρίνονται ικανοποιητικά στις αυξημένες απαιτήσεις του παιχνιδιού
- ενώ, μακροπρόθεσμα να προσεγγίσουν ή και να ξεπεράσουν τις ανθρώπινες επιδόσεις.

Κυρίως στα πλαίσια του παραπάνω διαγωνισμού, πολλές μελέτες και εργασίες έχουν προταθεί πάνω στο παιχνίδι. Οι εργασίες αυτές δεν πραγματεύονται μόνο την ανάπτυξη *ευφρών πρακτόρων*. Μερικές από αυτές επικεντρώνονται στην ακριβή και αποδοτική ανίχνευση της σκηνής του παιχνιδιού.

Στο [8] χρησιμοποιούνται δύο διαφορετικές τεχνικές Μηχανικής Μάθησης (*μάθησης με επίβλεψη (supervised learning)*), οι αλγόριθμοι *Weighted Majority* και *Naive Bayesian Network*. Οι αλγόριθμοι αυτοί αποφασίζουν ποια είναι η καταλληλότερη βολή σε κάθε βήμα του παιχνιδιού, αφού αρχικά έχουν εκπαιδευτεί με βάση δείγματα βολών, τα οποία προέρχονται από τη δράση διάφορων πρακτόρων (*Naive, RandomAngle agents*). Στη συνέχεια, τα δείγματα αυτά κατηγοριοποιούνται ως *goodShots* (βολές σε επίπεδα που ολοκληρώθηκαν επιτυχώς) ή *badShots* (\neg *goodShot*). Επιπλέον, η αναπαράσταση της σκηνής του παιχνιδιού υλοποιείται με ένα 7×7 *grid* για κάθε υποψήφια βολή, παίρνοντας ως είσοδο το αντικείμενο που αποτελεί τον στόχο και τη γωνία της βολής για το συγκεκριμένο στόχο. Έπειτα, το *grid* τοποθετείται γύρω από τον στόχο και δίνει ως έξοδο χαρακτηριστικά του στόχου της συγκεκριμένης βολής (σχετικές θέσεις στη σκηνή, πόσα αντικείμενα βρίσκονται στο κάθε κελί, τη θέση των γουρουνιών σε σχέση με τον στόχο της βολής κ.α.). Τελικά, αφού οι παραπάνω αλγόριθμοι έχουν εκπαιδευτεί και είναι σε θέση να παίζουν το παιχνίδι, λαμβάνουν την πληροφορία από το *grid* για τον στόχο κάθε

βολής (*impact point*) και αποφασίζουν ποια από αυτές θα δώσει τη μεγαλύτερη ανταμοιβή (*σκορ*).

Στο [5] δίνεται έμφαση στην ανίχνευση των αντικειμένων με αποδοτικό και ακριβή τρόπο, διαχωρίζοντας τα αντικείμενα σε *στατικά* (*static - Hough Transform*) και *δυναμικά* (*dynamic - Bounding Convex Polygon BCP*) και εφαρμόζοντας τις κατάλληλες τεχνικές για την αναπαράστασή τους. Στο [4] η σχεδίαση και υλοποίηση του πράκτορα που προτείνεται, βασίζεται στην *ποιοτική χωρική αναπαράσταση* (*qualitative spatial representation*) της σκηνής του παιχνιδιού. Επίσης χρησιμοποιείται μία συνάρτηση χρησιμότητας (*utility function*) για την εξαγωγή του στόχου, ενώ λαμβάνεται υπόψη η πιθανότητα κάθε δυνατής τροχιάς να φτάσει στον στόχο. Αρχικά, δίνεται μεγάλη έμφαση στη τοπολογική σχέση των αντικειμένων (πάνω, αριστερά από κ.α.). Στη συνέχεια, μία συνάρτηση χρησιμότητας, η οποία ορίζεται μόνο για αντικείμενα τύπου *Pig, Ice, Wood, Rock* και απεικονίζει την κατάσταση της σκηνής σε ένα πραγματικό αριθμό. Έπειτα, επιλέγεται ο στόχος με τη μεγαλύτερη χρησιμότητα. Τέλος, για να αντιμετωπιστεί το πρόβλημα της ενδεχόμενης *αστοχίας* (βολές που δεν πετυχαίνουν τον στόχο τους λόγω των Νόμων της Φυσικής), επιλέγεται η τροχιά με τη μεγαλύτερη πιθανότητα να πετύχει το στόχο.

Στο [15] δύο τύποι αντικειμένων λαμβάνονται υπόψη ως υποψήφιοι στόχοι. Ο πρώτος αφορά τα γουρούνια και ο δεύτερος δομές που τα προστατεύουν (*shelters*). Σε κάθε τέτοιο τύπο αντικειμένου, με την προϋπόθεση ότι μπορεί να χτυπηθεί απευθείας από μία βολή, ανατίθεται μία πραγματική τιμή με βάση την επίδραση που έχουν στα υπόλοιπα αντικείμενα της σκηνής. Η επίδραση διαχωρίζεται σε οριζόντια και κατακόρυφη, ενώ έχουν υλοποιηθεί πολλές μετρικές σχετικά με την *σταθερότητα* (*stability*), την *σχετική θέση* (*above*) και την *επαφή* μεταξύ των αντικειμένων. Τέλος, η τιμή που *τελικά* θα ανατεθεί σε κάθε εφικτό τύπο αντικειμένου εξαρτάται από τον τύπο του πουλιού της σφεντόνας αλλά και από το *είδος της τροχιάς* (*high or low parabola*).

Επιπρόσθετα, στο [19] προτείνεται ένας *ποιοτικός λογισμός αναπαράστασης* και ανάλυσης της σκηνής που στηρίζεται στις *δομικές ιδιότητες* των αντικειμένων. Χρησιμοποιείται ένα είδος *άλγεβρας* (*Interval Algebra*), η οποία εκφράζει εξαρτήσεις

μεταξύ των αντικειμένων, ενώ επεκτείνεται κατάλληλα (*Extended Rectangle Algebra*) για να εισαχθεί η έννοια της *σταθερότητας* (*stability*) των αντικειμένων στο πλαίσιο αυτό. Τελικά, η επιλογή του στόχου καθορίζεται από κανόνες (*rules*), οι οποίοι αφορούν τις δομικές ιδιότητες των αντικειμένων που δεν μπορούν να εκφραστούν επαρκώς με την προαναφερθείσα άλγεβρα.

Τέλος, στο [9] προτείνεται ένας μηχανισμός προσομοίωσης, ο οποίος υλοποιεί ένα μοντέλο τριών αφηρημένων επιπέδων. Ο πράκτορας στον οποίο ενσωματώνεται αυτός ο μηχανισμός, αρχικά προσπαθεί να αντιληφθεί το περιβάλλον του παιχνιδιού, στη συνέχεια να προβλέψει το αποτέλεσμα της επόμενης βολής προσομοιώνοντας διάφορες δυνατές τροχιές. Τελικά, επιλέγεται η τροχιά η οποία δίνει την μεγαλύτερη δυνατή επίδραση. Το αποτέλεσμα αυτό εξάγεται από μία διαδικασία μάθησης.

ΚΕΦΑΛΑΙΟ 3. ΠΡΟΤΕΙΝΟΜΕΝΗ ΜΕΘΟΔΟΣ

- 3.1 Εισαγωγή
 - 3.2 Γενική Περιγραφή της Μεθόδου
 - 3.3 Δενδρική Δομή Αναπαράστασης
 - 3.4 Εξαγωγή Υποψήφιων Στόχων
 - 3.5 Επιλογή Στόχου Συνδυάζοντας Μοντέλα Παλινδρόμησης
 - 3.6 Επιλογή tap time
 - 3.7 Online Διαδικασία Μάθησης
-

3.1. Εισαγωγή

Όπως τονίσαμε και σε προηγούμενα κεφάλαια της παρούσας διατριβής, το παιχνίδι “Angry Birds” αποτελεί μία μεγάλη πρόκληση για την Τεχνητή Νοημοσύνη, στα πλαίσια ανάπτυξης ευφυών αυτόνομων πρακτόρων που, μακροπρόθεσμα, θα ανταγωνίζονται τις ανθρώπινες επιδόσεις στο παιχνίδι αυτό. Στην παρούσα διατριβή προτείνουμε έναν *ευφυή αυτόνομο πράκτορα* για το παιχνίδι “Angry Birds”, ο οποίος έχει *δύο βασικά προτερήματα*:

- πρώτον, *αποδοτική και χαμηλής πολυπλοκότητας αναπαράσταση της σκηνής σε μία δενδρική δομή*, της οποίας οι κόμβοι (*nodes*) αναπαριστούν αντικείμενα της σκηνής του παιχνιδιού. Η δομή αυτή προσφέρει πληθώρα πληροφορίας σχετικά με τη σκηνή του παιχνιδιού. Ταυτόχρονα, διευκολύνει την εξαγωγή χαρακτηριστικών (*features*), τα οποία χρησιμοποιούνται για την πρόβλεψη της αναμενόμενης ανταμοιβής κάθε πιθανού στόχου

- δεύτερον, μία πολυτροπική διαδικασία επιλογής του τελικού στόχου, λαμβάνοντας υπόψη τον τύπο πουλιού-αντικειμένου, με άλλα λόγια μία *ensemble διαδικασία μάθησης* [7], η οποία συνδυάζει πολλαπλά μοντέλα μάθησης. Για κάθε τύπο χρησιμοποιείται ένα ξεχωριστό **Μπεϋζιανό Μοντέλο Γραμμικής Παλινδρόμησης (Bayesian linear regression model)**, το οποίο δίνει την πρόβλεψη για την αναμενόμενη ανταμοιβή του κάθε στόχου.

Το πρόβλημα επιλογής του μοντέλου μετατρέπεται σε ένα *multi-armed bandit problem* και η επιλογή του μοντέλου που θα δώσει τον τελικό στόχο γίνεται με τον αλγόριθμο *UCB* [4], σε συνδυασμό με την πρόβλεψη του μοντέλου για την αναμενόμενη ανταμοιβή. Τέλος, η ενημέρωση των παραμέτρων του μοντέλου που αντιστοιχεί στον στόχο που επιλέχθηκε, γίνεται με μία *online διαδικασία μάθησης*.

Στον παρόν κεφάλαιο παρουσιάζεται αναλυτικά η μέθοδος που προτείνουμε για την κατασκευή ενός πράκτορα στα πλαίσια αυτής της διατριβής. Στην ενότητα 3.2 δίνουμε τη γενική περιγραφή της μεθόδου. Στην ενότητα 3.3 αναλύεται η δενδρική δομή αναπαράστασης, η οποία αποτελεί *αναπόσπαστο κομμάτι* της μεθόδου που προτείνουμε, και δίνεται ο αλγόριθμος υλοποίησής της. Στην ενότητα 3.4 περιγράφεται αναλυτικά η εξαγωγή των εφικτών στόχων της σκηνής. Στη συνέχεια, στην ενότητα 3.5 παρουσιάζεται η διαδικασία επιλογής του στόχου, μέσα από την επιλογή του κατάλληλου *regressor* (*γραμμικού παλινδρομητή*), ενώ στην ενότητα 3.6 δίνονται λεπτομέρειες σε σχέση με το *tap time*. Κλείνοντας, στην ενότητα 3.7 παρουσιάζεται η *online* διαδικασία μάθησης, με την οποία ενημερώνονται οι παράμετροι του μοντέλου που επιλέχθηκε για την εξαγωγή του τελικού στόχου.

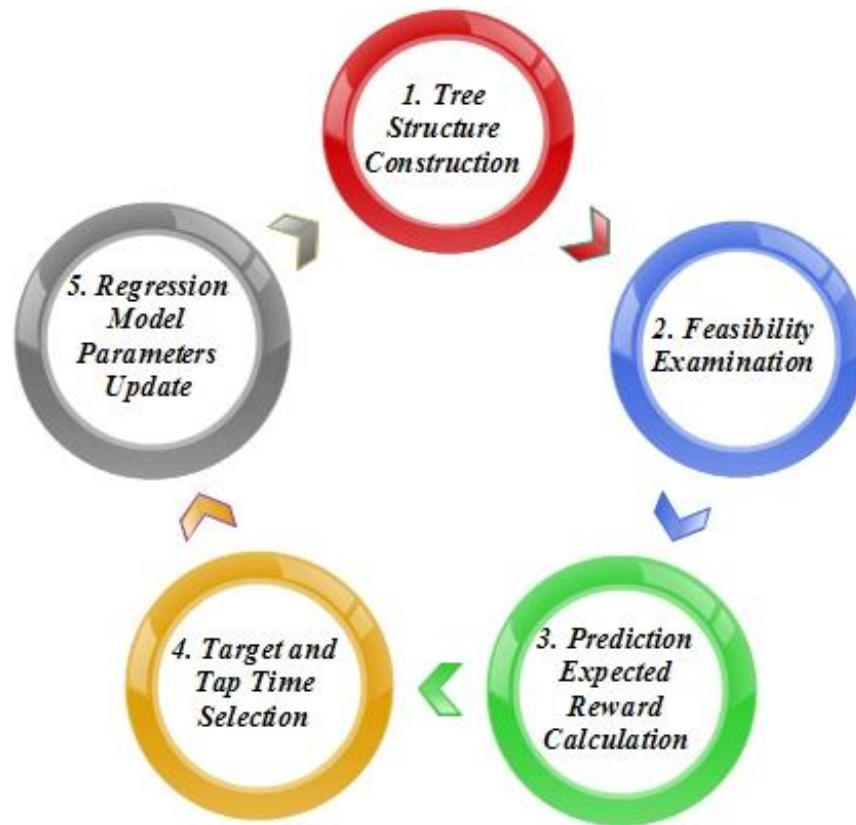
3.2. Γενική Περιγραφή της Μεθόδου

Στα πλαίσια αυτής της διατριβής βασιστήκαμε στο λογισμικό *Angry Birds Game Playing Software (version 1.31)* [18], το οποίο είναι υλοποιημένο σε *Java* και παρέχεται από τους διοργανωτές του *Angry Birds AI Competition*. Η προτεινόμενη μέθοδος δίνει μία αποδοτική αναπαράσταση του χώρου καταστάσεων του παιχνιδιού αξιοποιώντας όλες τις πληροφορίες αναγνώρισης των αντικειμένων της σκηνής που παρέχονται από το *Vision component* του προαναφερθέντος software. Στη συνέχεια, ο

στόχος επιλέγεται από ένα μηχανισμό, ο οποίος χρησιμοποιεί *Μπεϋζιανή Ensemble Παλινδρόμηση* για να κάνει την πρόβλεψη σχετικά με την αναμενόμενη ανταμοιβή που θα επιφέρει ο στόχος αυτός.

Η μέθοδος αποτελείται από τα παρακάτω στάδια, όπως φαίνεται και στο Σχήμα 3.1:

1. **ΑΝΑΠΑΡΑΣΤΑΣΗ (Tree Structure Construction):** Κατασκευή του δένδρου αναπαράστασης της σκηνής.
2. **ΥΠΟΨΗΦΙΟΙ ΣΤΟΧΟΙ (Feasibility Examination):** Για κάθε κόμβο του δένδρου εξετάζουμε εάν μπορεί να αποτελέσει υποψήφιο στόχο, δηλαδή αν μπορεί να χτυπηθεί *άμεσα*. Με άλλα λόγια, ελέγχουμε αν παρεμβάλλεται οτιδήποτε στην πορεία της τροχιάς από τη σφεντόνα μέχρι τον στόχο, ή αν μπορεί να χτυπηθεί με βάση τους νόμους της Φυσικής. Άρα, αν μπορεί να χτυπηθεί *άμεσα* τότε είναι ένας υποψήφιος στόχος και ο κόμβος χαρακτηρίζεται *εφικτός (feasible)*, διαφορετικά χαρακτηρίζεται ως *μη εφικτός (infeasible)*.
3. **ΕΠΙΛΟΓΗ ΣΤΟΧΟΥ (Prediction: Expected Reward Calculation):** Υπολογισμός της αναμενόμενης ανταμοιβής καθενός από τους εφικτούς στόχους χρησιμοποιώντας ένα *μηχανισμό Πολυτροπικής Μπεϋζιανής Ensemble Παλινδρόμησης*, ο οποίος λαμβάνει υπόψη τον τύπο πουλιού-αντικειμένου. Τελικά, ο στόχος με τη μεγαλύτερη αναμενόμενη ανταμοιβή επιλέγεται.
4. **ΕΚΤΕΛΕΣΗ (Target and Tap Time Selection):** Εκτέλεση της βολής λαμβάνοντας υπόψη τον τύπο πουλιού που βρίσκεται στη σφεντόνα για τον καθορισμό του *tap time*.
5. **ΕΚΠΑΙΔΕΥΣΗ (Regression Models Parameters Update):** Ενημέρωση των παραμέτρων του μοντέλου παλινδρόμησης (*regressor*) που αντιστοιχεί στον τελικό στόχο που επιλέχθηκε.



Σχήμα 3.1: Στάδια Προτεινόμενης Μεθόδου

3.3. Δενδρική Δομή Αναπαράστασης

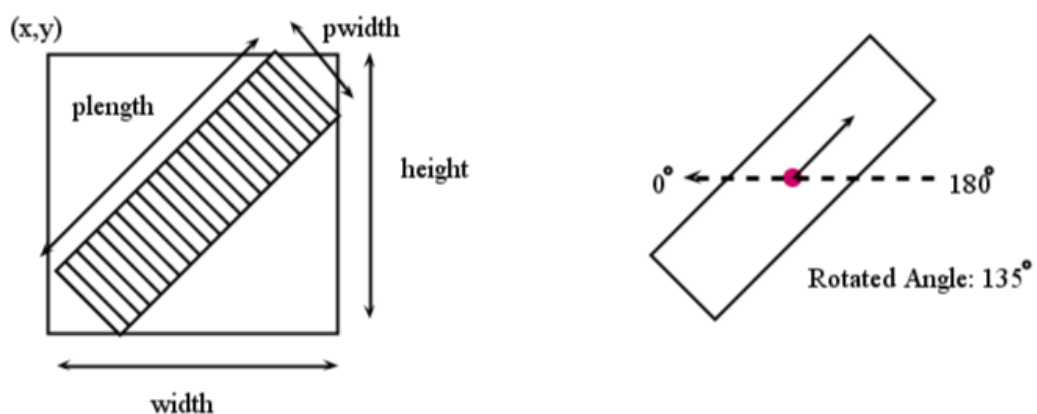
Η πολύπλοκη και πολύπλευρη φύση του παιχνιδιού “Angry Birds” οδηγεί στην εύρεση μίας κατάλληλης δομής για την αναπαράσταση του χώρου καταστάσεων, η οποία θα πρέπει:

- να είναι *αποδοτική*
- να έχει χαμηλή πολυπλοκότητα
- να εκμεταλλεύεται πλήρως όλες τις πληροφορίες που παρέχονται από το *Vision Component* του *software* και
- να επιτρέπει την εξαγωγή χρήσιμων *χαρακτηριστικών (features)*, ώστε να οδηγηθούμε σε ένα αριθμητικό χώρο καταστάσεων.

Η προτεινόμενη δενδρική δομή που θα παρουσιαστεί στη συνέχεια συνδυάζει σε μεγάλο βαθμό όλα τα παραπάνω.

3.3.1. Τύποι Αντικειμένων ως Κόμβοι του Δένδρου

Αρχικά, τα αντικείμενα της σκηνής επιστρέφονται σε λίστες της *Java* με την κλήση των κατάλληλων συναρτήσεων. Ουσιαστικά, αυτές οι λίστες αποτελούν την είσοδο της διαδικασίας κατασκευής του δένδρου της κάθε σκηνής. Κάθε στοιχείο των λιστών αυτών αναπαριστά ένα αντικείμενο της σκηνής και παρέχει πληροφορία σχετικά με **τον τύπο του** (πέτρα, ξύλο, πάγος κλπ.), **τη θέση του στη σκηνή** (συντεταγμένες της πάνω αριστερής γωνίας το αντικειμένου), **τις διαστάσεις του** (πλάτος, ύψος), **το εμβαδόν του** και **τη γωνία** με την οποία βρίσκεται τοποθετημένη πάνω στη σκηνή και το σχήμα του αντικειμένου, όπως φαίνεται στο Σχήμα 3.2 [18].



Σχήμα 3.2: Αναπαράσταση Αντικειμένων της Σκηνής

Συνδυάζοντας όλα τα παραπάνω στοιχεία, διακρίνουμε επτά (7) τύπους αντικειμένων, οι οποίοι εμφανίζονται σε επίπεδα του παιχνιδιού. Πρέπει να σημειωθεί ότι η διάκριση αυτή λαμβάνει υπόψη το υλικό και το σχήμα των αντικειμένων. Η διάκριση αυτή γίνεται ως ακολούθως:

- *Ice* (I)
- *Wood* (W)
- *Stone* (S)
- *Rolling Stone* (RS)
- *Rolling Wood* (RW)
- *Pig* (P)
- *TNTs* (T)

οι οποίοι αποτελούν κόμβους (*nodes*) της δενδρικής δομής της προτεινόμενης μεθόδου. Οι κόμβοι του δένδρου απαιτούν περισσότερη πληροφορία από εκείνη των αρχικών αντικειμένων, που έχει κυρίως να κάνει με τη σύνδεσή τους με άλλους κόμβους του δένδρου (πατέρας ή παιδί σε σχέση με κάποιον άλλο κόμβο του δένδρου), με την ιδιότητα τους να μπορούν να θεωρηθούν εφικτοί στόχοι ή όχι και με διάφορες αποστάσεις από σημαντικά στοιχεία της σκηνής (π.χ. κοντινότερα σε αυτούς γουρούνια, σφεντόνα). Οι πληροφορίες αυτές είναι διαθέσιμες για κάθε κόμβο μετά την κατασκευή του δένδρου. Χρησιμοποιώντας τις παραπάνω σημαντικές πληροφορίες προχωράμε στην εξαγωγή χαρακτηριστικών (*features*) για κάθε κόμβο, τα οποία θα χρησιμοποιηθούν στην πρόβλεψη για την αναμενόμενη ανταμοιβή κάθε κόμβου, ανάλογα βέβαια με τον τύπο πουλιού στη σφεντόνα για τη συγκεκριμένη βολή.

3.3.2. Κατασκευή του Δένδρου

Η διαδικασία κατασκευής της δενδρικής δομής αναπαράστασης δίνεται από τον Αλγόριθμο 1. Η κατασκευή του δένδρου γίνεται *ιεραρχικά* από το έδαφος προς τα επάνω (*bottom-up fashion*) [14]. Οι *ακμές/συνδέσεις* μεταξύ των κόμβων του δένδρου αναπαριστούν τη σχέση γονέα-παιδιού (ή πιο γενικά παιδιού-προγόνων). Αξίζει να σημειώσουμε ότι ακμές υπάρχουν μόνο μεταξύ κόμβων *διαφορετικών επιπέδων*, που σημαίνει ότι δεν μπορεί να υπάρχει σύνδεση μεταξύ κόμβων του ίδιου επιπέδου. Με άλλα λόγια, κόμβοι του ίδιου επιπέδου *δεν μπορούν* να έχουν σχέση παιδιού-προγόνων (*βήμα 5*). Επίσης, στο *βήμα 2* δημιουργείται ένας εικονικός κόμβος *Root* (*virtual node*), ο οποίος έχει παιδιά όσους από τους κόμβους δεν έχουν κάποιον άλλο κόμβο από πάνω τους, δηλαδή δεν έχουν πατέρα. Με τη δημιουργία του κόμβου αυτού δε μένουν *ορφανοί* κόμβοι (*orphan nodes*) και τελικά το δένδρο είναι συνεκτικό, δηλαδή αποτελείται από μία συνεκτική συνιστώσα, τον εαυτό του. Επιπρόσθετα, στο *βήμα 3* είναι απαραίτητη η ταξινόμηση των κόμβων του δένδρου από τον αριστερότερο προς τον δεξιότερο, έτσι ώστε να αποτυπώνεται στο δένδρο η άποψη της σκηνής χωρικά. Τέλος, πρέπει να σημειώσουμε ότι αντικείμενα τύπου *hill* δεν εισάγονται ως κόμβοι στην δενδρική δομή και λαμβάνονται υπόψη μόνο ως προς την εξέταση της εφικτότητας του κάθε κόμβου (*βλ. ενότητα 3.4*).

Αλγόριθμος 1 – Κατασκευή Δένδρου

Είσοδος: Λίστα (*list*) που περιέχει τα αντικείμενα της σκηνής

Έξοδος: Δενδρική αναπαράσταση της σκηνής

Start_Algorithm
1. Κατασκευή επιπέδων{

k = 1;

While (! *list.isEmpty()*){ // όσο η λίστα έχει αντικείμενα

α. Δημιουργία επιπέδου *k*

β. Εύρεση του κοντινότερου στο έδαφος αντικειμένου της προσωρινής λίστας

γ. Χάραξη «νοητής» οριζόντιας ευθείας που διέρχεται από το κέντρο του επιλεγμένου αντικειμένου

δ. Εισαγωγή των αντικειμένων που τέμνονται από τη «νοητή» ευθεία στο επίπεδο *k*

ε. Διαγραφή των αντικειμένων αυτών από την αρχική λίστα και ενημέρωση της λίστας

στ. *k* ++;

} **end_While**

}

2. Δημιουργία εικονικού κόμβου *Root* και εισαγωγή του στο επίπεδο *k* (ρίζα του δένδρου)
3. Ταξινόμηση των κόμβων κάθε επιπέδου με βάση τη θέση τους στη σκηνή, από τον αριστερότερο προς τον δεξιότερο
4. Συγχώνευση/Συνένωση κόμβων από το ίδιο ή διαφορετικό επίπεδο (βλέπε 3.3.3)
5. Δημιουργία Ακμών/Σχέσεων Παιδιού και Γονέα κόμβου{

For each level *l*{

For each Node *v*{

α. Αναζήτηση κόμβων από υψηλότερα επίπεδα που μέρος τους βρίσκεται πάνω από τον *v* (είτε εφάπτονται, είτε όχι)

β. Αν υπάρχει τέτοιου είδους κόμβος ή κόμβοι του ίδιου επιπέδου, τότε γίνεται ή γίνονται πατέρας (-ες) του κόμβου *v*, διαφορετικά ο κόμβος *v* είναι *ορφανός* (*orphan node*) και πατέρας του είναι ο εικονικός κόμβος *Root*.

} **End_For**

} **End_For**

}

End_Algorithm

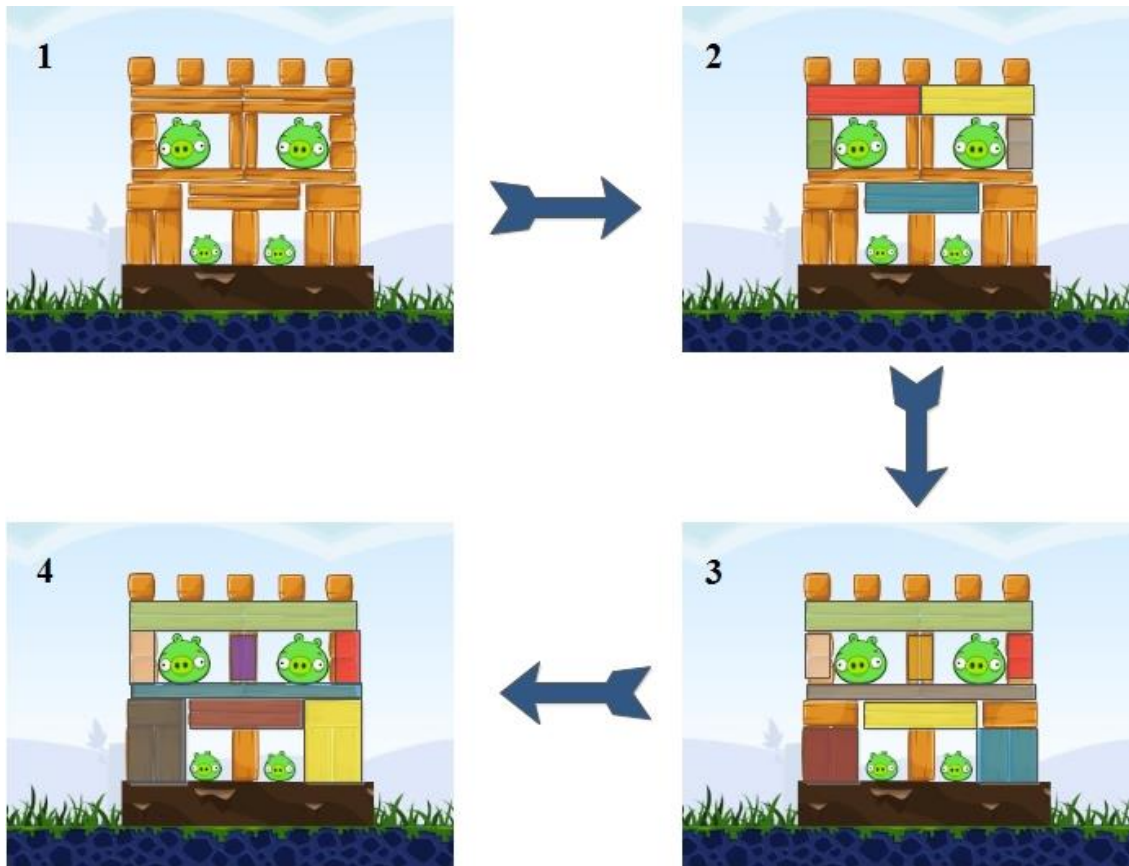
3.3.3. Συγχώνευση/Συνένωση Κόμβων

Πρόκειται για μία λειτουργία η οποία μειώνει τη χωρική πολυπλοκότητα του δένδρου και εκτελείται στο βήμα 4 του αλγορίθμου. Για την εκτέλεσή της πρέπει να ισχύουν κάποιες προϋποθέσεις που αφορούν κυρίως τον *τύπο*, τη *θέση* και τις *διαστάσεις* των κόμβων που πρόκειται να συγχωνευθούν. Πιο συγκεκριμένα, σε όλες τις περιπτώσεις **οι κόμβοι πρέπει να είναι του ίδιου τύπου για να γίνει συνένωση**. Δηλαδή, δεν επιτρέπουμε να συγχωνευθούν δύο κόμβοι με τύπους *Wood* και *Stone*. Επίσης, αν πρόκειται για κόμβους του *ίδιου επιπέδου* πρέπει ο ένας να βρίσκεται *ακριβώς δίπλα* στον άλλο, και να *έχουν περίπου ίδιο ύψος*. Τότε είναι εφικτή η συνένωση. Αν πρόκειται για κόμβους *διαφορετικού επιπέδου*, τότε πρέπει να βρίσκεται ο ένας *ακριβώς πάνω από τον άλλο* και να *έχουν περίπου το ίδιο πλάτος*. Τότε είναι εφικτή η συνένωση. Έχει διαπιστωθεί πειραματικά ότι ο καλύτερος τρόπος για να επιτευχθεί η αποδοτικότερη συνένωση κόμβων είναι ο εξής:

- αρχικά, συνένωση κόμβων *διαφορετικών επιπέδων*
- στη συνέχεια συνένωση κόμβων του *ίδιου επιπέδου*
- και τέλος, συνένωση κόμβων *διαφορετικών επιπέδων*

Τέλος, αξίζει να σημειωθεί ότι δεν εφαρμόζεται η συνένωση σε κόμβους που είναι τύπου *Rig* ή *TNT*. Η λειτουργία αυτή του Αλγορίθμου 1 κρίνεται ιδιαίτερα σημαντική σε πολλές περιπτώσεις, κυρίως σε επίπεδα με πολύπλοκες σκηνές, όπου υπάρχουν πολλά αντικείμενα με ίδιες ιδιότητες και μπορούν να συγχωνευθούν. Με αυτόν τον τρόπο γίνεται περισσότερο αποδοτική η δενδρική δομή αναπαράστασης, χωρίς ωστόσο να χάνεται πληροφορία για τα αντικείμενα της σκηνής. Παράδειγμα της συγκεκριμένης λειτουργίας δίνεται στο Σχήμα 3.3.

Στο Σχήμα 3.3 αποτυπώνεται η μέθοδος που περιγράφηκε προηγουμένως για τη *Συγχώνευση/Συνένωση κόμβων* του δένδρου. Παρατηρούμε ότι για το συγκεκριμένο επίπεδο πρέπει να δημιουργηθούν 30 κόμβοι (31 με τον κόμβο *Root*). Εφαρμόζοντας την λειτουργία της *Συνένωσης* όπως περιγράφηκε, ο αριθμός των κόμβων του δένδρου περιορίζεται σημαντικά. Συγκεκριμένα, απαιτείται η δημιουργία 18 κόμβων (19 με τον κόμβο *Root*) για την αναπαράσταση της συγκεκριμένης σκηνής, γεγονός που μειώνει σε σημαντικό βαθμό την πολυπλοκότητα της δομής που προτείνουμε.



Σχήμα 3.3: Παράδειγμα Συνένωσης Κόμβων του Δένδρου

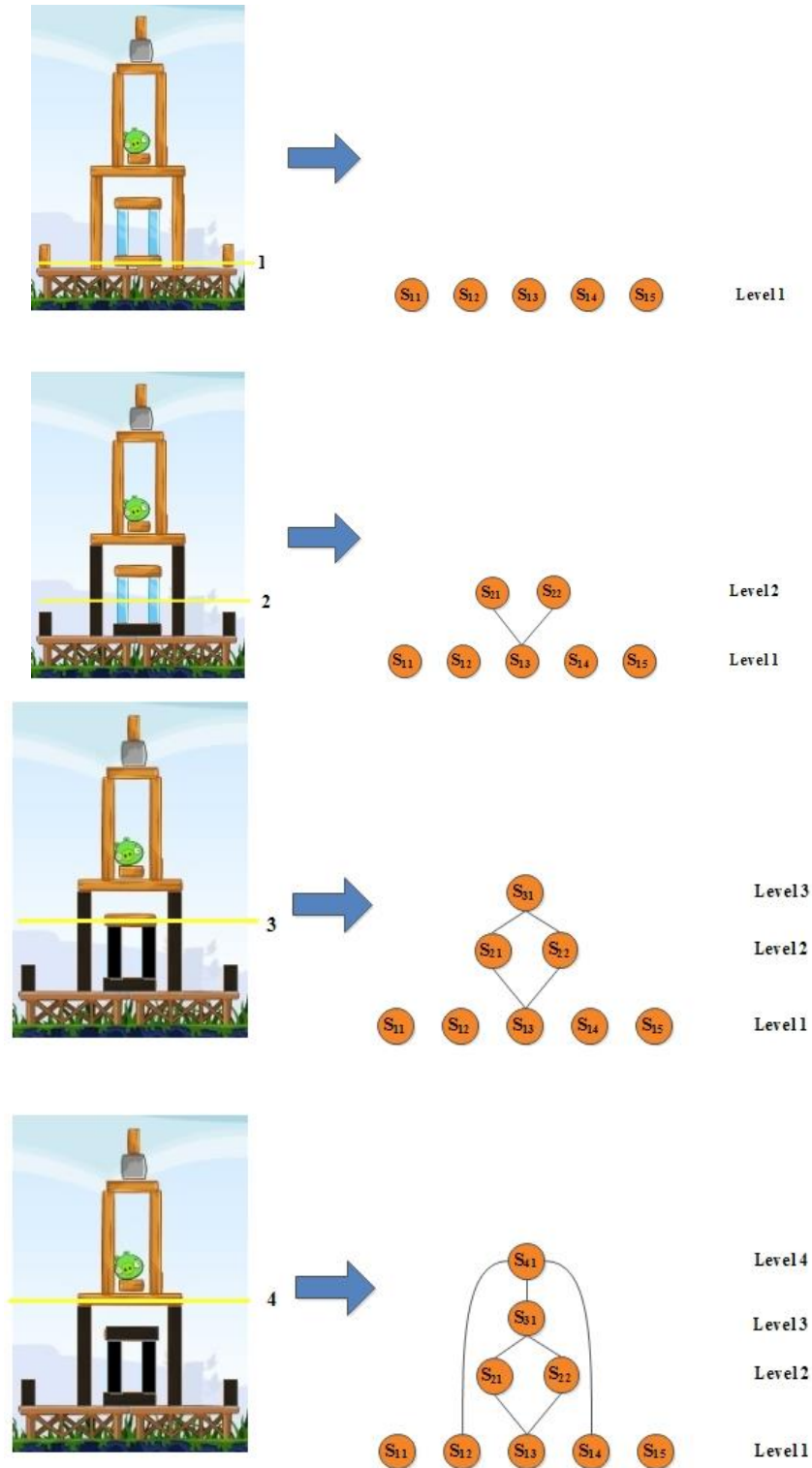
3.3.4. Εξαγωγή χαρακτηριστικών κόμβων

Μετά την κατασκευή του δένδρου για κάθε κόμβο s μπορούμε να εξάγουμε τα τρία ακόλουθα χαρακτηριστικά (*features*) με *ιεραρχικό* τρόπο, αλλά σε αυτή την περίπτωση από τη ρίζα (*Root*) του δένδρου προς τα φύλλα του (*top-down fashion*):

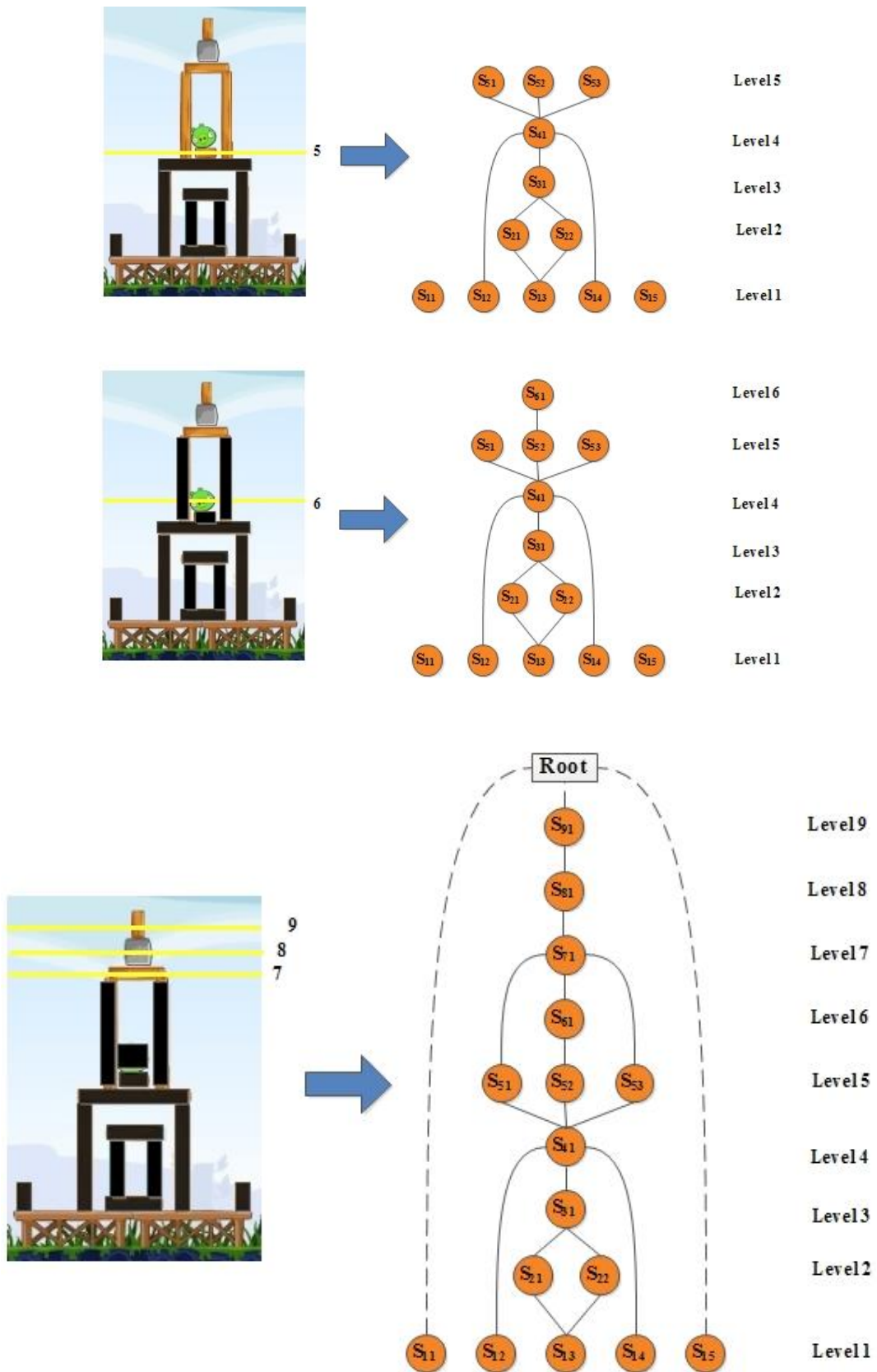
- $x_1(s)$: *Ατομικό βάρος (personal weight)* του κόμβου s , το οποίο ισούται με το εμβαδόν που καταλαμβάνει το αντικείμενο του s .
- $x_2(s)$: *Αθροιστικό βάρος των προγόνων $\mathcal{P}(s)$* του κόμβου s , $x_2(v) = \sum_{s' \in \mathcal{P}(s)} x_1(s')$.
- $x_3(s)$: *Η απόσταση (distance in pixels)* του κόμβου s από το κοντινότερο σε αυτόν γουρούνι, η οποία είναι κανονικοποιημένη στο $[0,1]$. Αυτό επιτυγχάνεται διαιρώντας την αρχική απόσταση με 100, έχοντας ως εμπειρικό κριτήριο ότι αντικείμενα με απόσταση μικρότερη από 100 μπορούν να επηρεάσουν άμεσα τον πραγματικό στόχο

3.3.5. Παραδείγματα Κατασκευής Δένδρου

Στην ενότητα αυτή δίνονται παραδείγματα κατασκευής της δενδρικής δομής που προτείνουμε στα πλαίσια της παρούσας διατριβής.



Σχήμα 3.4: Κατασκευή Δένδρου – 4 Βήματα



Σχήμα 3.5: Κατασκευή Δένδρου – Βήματα 5-9 - Ολοκληρωμένη Δενδρική Δομή

Πίνακας 3.1: Χαρακτηριστικά Κόμβων της Δενδρικής Δομής του Σχήματος 3.6

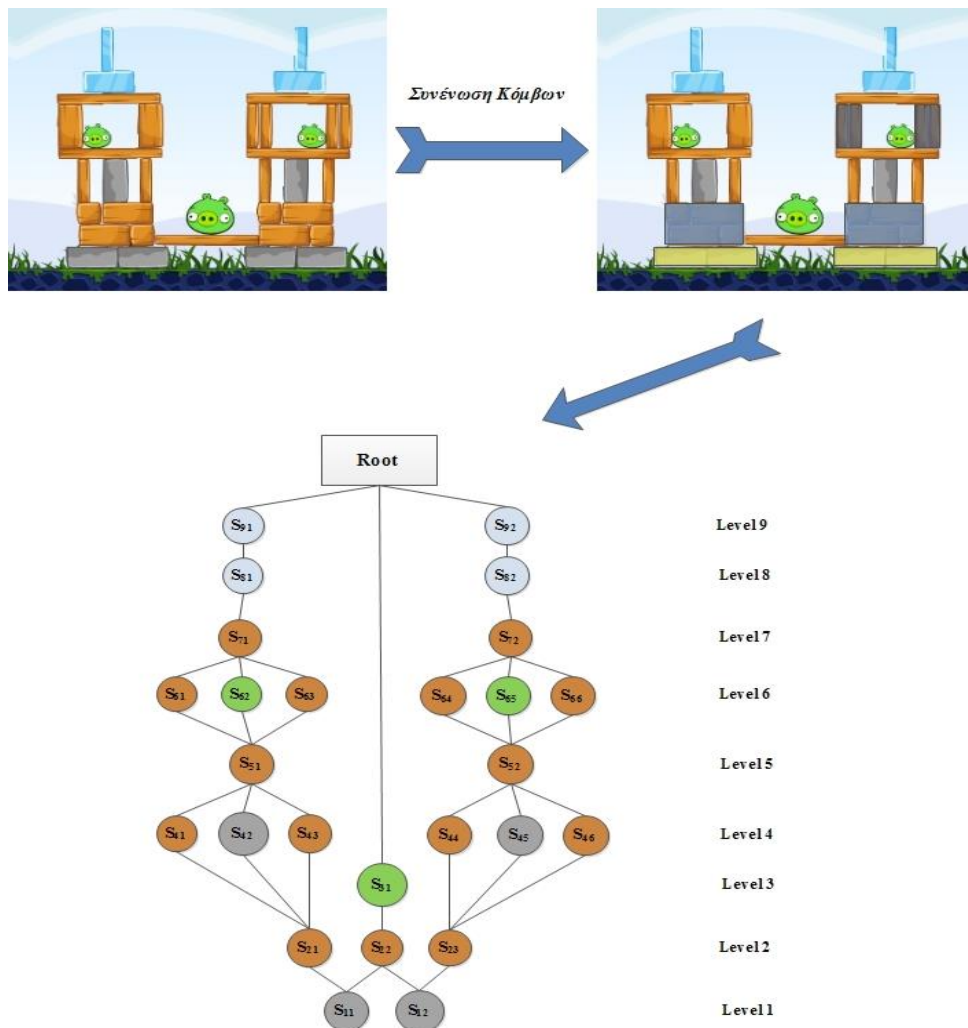
<i>Nodes</i>	<i>Features of each Node</i>					
	<i>Level</i>	<i>Type</i>	<i>Feasible</i>	<i>Personal Weight</i> (x_1)	<i>Above Weight</i> (x_2)	<i>Distance from Nearest Pig</i> (x_3)
S₁₁	1	<i>Wood</i>	True	65	0	0.818
S₁₂	1	<i>Wood</i>	True	312	3557	0.501
S₁₃	1	<i>Wood</i>	False	156	7656	0.660
S₁₄	1	<i>Wood</i>	False	312	3557	0.501
S₁₅	1	<i>Wood</i>	False	65	0	0.818
S₂₁	2	<i>Ice</i>	False	162	3682	0.504
S₂₂	2	<i>Ice</i>	False	130	3682	0.504
S₃₁	3	<i>Wood</i>	False	125	3557	0.341
S₄₁	4	<i>Wood</i>	False	318	3239	0.151
S₅₁	5	<i>Wood</i>	True	318	377	0.164
S₅₂	5	<i>Wood</i>	False	72	1777	0.082
S₅₃	5	<i>Wood</i>	False	318	377	0.198
S₆₁	6	<i>Pig</i>	True	1400	377	0.170
S₇₁	7	<i>Wood</i>	True	156	221	0.431
S₈₁	8	<i>Stone</i>	True	156	65	0.521
S₉₁	9	<i>Wood</i>	True	65	0	0.651

Στα παραπάνω Σχήματα (3.4 και 3.5) παρουσιάζεται βήμα προς βήμα η κατασκευή της δενδρικής δομής για το πρώτο επίπεδο του επεισοδίου “Poached Eggs”. Πρόκειται για μία απλή σκηνή του παιχνιδιού, που όμως αποτυπώνει και δίνει με τον καλύτερο τρόπο γραφικά την δενδρική δομή που προτείνουμε.

Τέλος, αμέσως μετά την ολοκλήρωση της κατασκευής της δενδρικής δομής γίνεται η εξαγωγή των χαρακτηριστικών των κόμβων. Η δομή που προτείνουμε παραπάνω

καθιστά την εξαγωγή των χαρακτηριστικών ιδιαίτερα απλή και εύκολη διαδικασία. Στον Πίνακα 3.1 παρουσιάζονται οι τιμές των χαρακτηριστικών καθώς και η εφικτότητα (*feasibility*) για κάθε κόμβο του δένδρου. Η εξέταση για την εφικτότητα κάθε κόμβου αναλύεται στην ενότητα 3.4 που ακολουθεί.

Ολοκληρώνοντας την παρούσα ενότητα, θα δώσουμε ένα ακόμη παράδειγμα κατασκευής του δένδρου της μεθόδου, παρουσιάζοντας απευθείας το δένδρο για μία συγκεκριμένη σκηνή του παιχνιδιού. Στο Σχήμα 3.6 παρουσιάζεται το δένδρο για το 9^ο επίπεδο του επεισοδίου “Roached Eggs”, δείχνοντας αρχικά τη διαδικασία Συγχώνευσης/Συνένωσης, και στη συνέχεια παρουσιάζοντας τη δενδρική δομή για αυτή τη σκηνή του παιχνιδιού, όπου το χρώμα κάθε κόμβου αντιστοιχεί στο χρώμα του υλικού που αυτός αντιπροσωπεύει.



Σχήμα 3.6: Ολοκληρωμένη Δενδρική Δομή 9^ο Επιπέδου

Παρατηρώντας προσεκτικά τη σκηνή του επιπέδου 9 του Σχήματος 3.7, βλέπουμε ότι υπάρχουν δύο όμοιες δομές αριστερά και δεξιά του κεντρικού γουρουνιού. Αυτό αποτυπώνεται και στη δενδρική δομή μετά και τη διαδικασία συνένωσης των κόμβων.

3.4. Εξαγωγή Υποψήφιων Στόχων

Το επόμενο στάδιο είναι η εξέταση κάθε κόμβου του δένδρου σχετικά με το αν μπορεί να αποτελέσει *υποψήφιο στόχο*, οπότε χαρακτηρίζεται εφικτός (*feasible*) ή ανέφικτος (*infeasible*) κόμβος με βάση τα ακόλουθα κριτήρια:

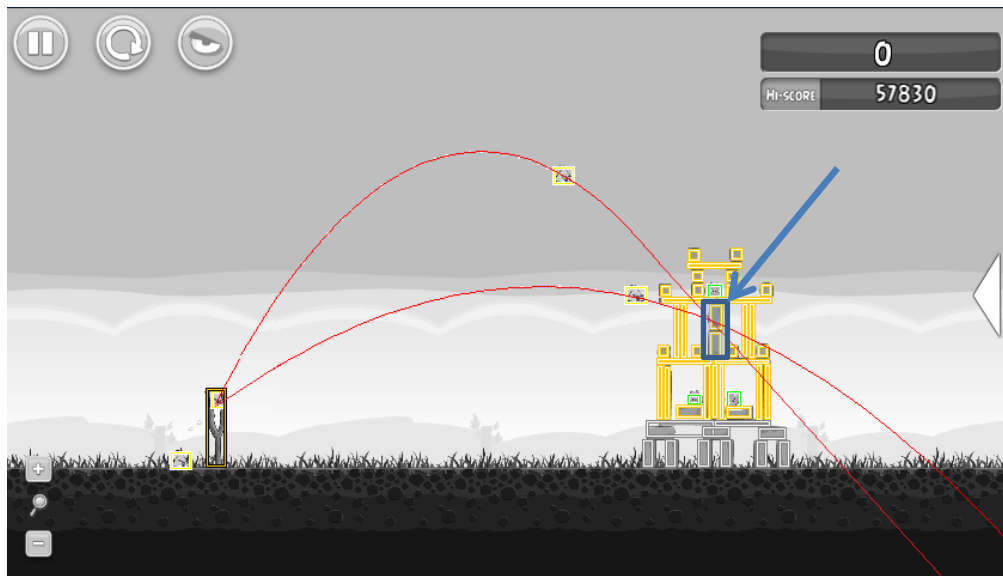
- **Πρώτον**, αν πράγματι μπορεί να χτυπηθεί ο εξεταζόμενος κόμβος με βάση τους κανόνες και τους νόμους της Φυσικής. Όπως φαίνεται και στην Εικόνα 3.1, οι τρεις κόμβοι τύπου *Pig*, εντός του διαφανούς περιγράμματος, είναι αδύνατο να χτυπηθούν, καθώς η σφεντόνα βρίσκεται πολύ μακριά από αυτά και δεν υπάρχει τροχιά ικανή να τα προσεγγίσει.



Εικόνα 3.1: Παράδειγμα Ανέφικτων (*infeasible*) Κόμβων

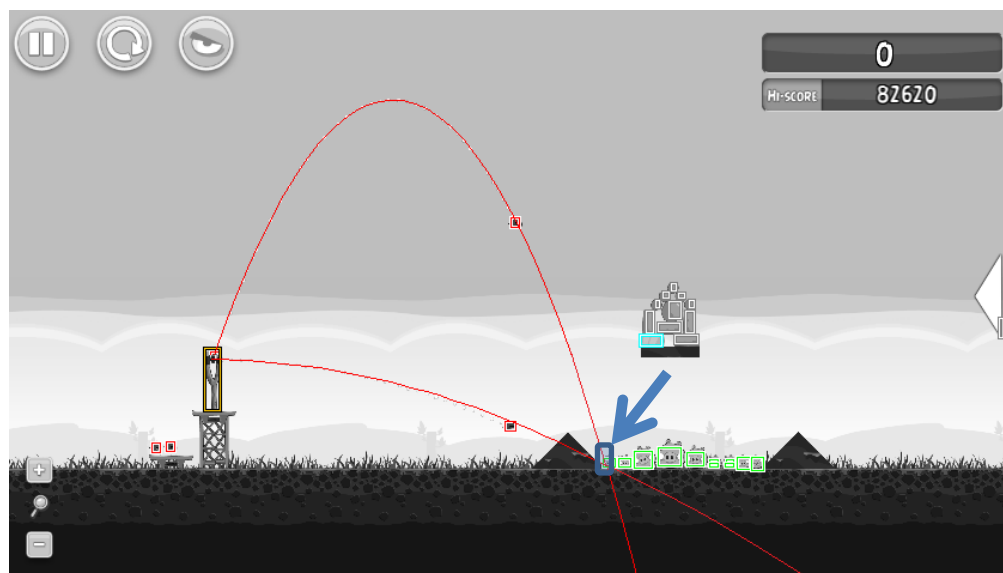
- **Δεύτερον**, αν κάποιος άλλος κόμβος του δένδρου παρεμβάλλεται στην πορεία της τροχιάς, όπως φαίνεται στην Εικόνα 3.2 για τον κόμβο τύπου *Wood*, που βρίσκεται εντός του διαφανούς πλαισίου, όπου και τέμνονται οι δύο τροχιές (*direct* και *high-arching*). Όπως παρατηρούμε ο κόμβος αυτός δεν μπορεί να

είναι υποψήφιος στόχος, διότι και για τις δύο δυνατές τροχιές το πρώτο σημείο πρόσκρουσης δεν ανήκει σε αυτόν τον κόμβο, αλλά σε διαφορετικό. Και για τις δύο βολές ισχύει το παραπάνω, οπότε ο κόμβος αυτός χαρακτηρίζεται *ανέφικτος* και δεν λαμβάνεται ως υποψήφιος στόχος, αφού δεν μπορεί να χτυπηθεί άμεσα.



Εικόνα 3.2: Ανέφικτος Κόμβος Τύπου Wood λόγω της Δενδρικής Δομής

- **Τρίτον**, αν υπάρχει κάποιος λόφος (*hill*) που παρεμβάλλεται στην πορεία της τροχιάς, όπως φαίνεται στην Εικόνα 3.3.



Εικόνα 3.3: Εφικτός Κόμβος Pig

Όπως φαίνεται στην Εικόνα 3.3, ο κόμβος τύπου *Pig* μπορεί να χτυπηθεί άμεσα από την *high-arching τροχιά*, ενώ η *direct τροχιά* προσκρούει πάνω στο λόφο που βρίσκεται στα αριστερά του. Τελικά, υπάρχει μία τουλάχιστον τροχιά που μπορεί να χτυπήσει άμεσα τον κόμβο αυτό, οπότε ο κόμβος χαρακτηρίζεται *εφικτός (feasible)* στόχος.

Αξίζει να σημειώσουμε ότι για κάθε κόμβο ανάλογα με τον *τύπο*, το *σχήμα*, τις *διαστάσεις* του και τη *γωνία* που σχηματίζει με το δάπεδο έχει *υποψήφια σημεία* στα οποία εξετάζουμε αν μπορεί να χτυπηθεί είτε με *direct*, είτε με *high-arching* τροχιές. Αν τελικά μετά την εξέταση παραπάνω περιορισμών για κάθε τέτοιο σημείο ενός κόμβου υπάρχει έστω και μία τροχιά η οποία τους ικανοποιεί, τότε λέμε ότι ο κόμβος αποτελεί *εφικτό στόχο (feasible)*. Αν υπάρχουν δύο εφικτές τροχιές, *direct* και *high-arching*, τότε προτιμάται η *direct* τροχιά λόγω της μεγαλύτερης επίδρασης που έχει πάνω στη σκηνή σε σχέση με την *high-arching*.

Σε ότι αφορά την εφικτότητα κόμβων τύπου *Pig* και *TNT* πρέπει να σημειωθεί ότι εξετάζονται ως προς το **πρώτο κριτήριο** (νόμοι Φυσικής) και το **τρίτο κριτήριο** (*hills*). Δηλαδή, είναι εφικτοί στόχοι στην περίπτωση που πράγματι μπορούν να χτυπηθούν και δεν προστατεύονται από λόφους. Αυτή η παραδοχή έγινε με βάση την παρατήρηση ότι οι δύο αυτοί τύποι κόμβων έχουν σημαντική επίδραση στην ολοκλήρωση ενός επιπέδου αν δεν προστατεύονται από *σημαντικά* εμπόδια. Όπως θα δούμε παρακάτω, ο πράκτορας έχει τη δυνατότητα να *μάθει* ότι αν τέτοιου είδους κόμβοι προστατεύονται από σημαντικά εμπόδια δεν είναι δυνατό να χτυπηθούν και να επιστρέψουν μεγάλη ανταμοιβή.

Επιπλέον, ιδιαίτερη προσοχή χρειάζεται στην περίπτωση των *White Birds* σχετικά με το ποιοι κόμβοι είναι εφικτοί και μπορούν να αποτελέσουν *υποψήφιους στόχους*. Η ιδιότητα των *White Birds* να αφήνουν *egg bombs* όταν εκτελείται το *tap*, οδήγησε στην διαπίστωση ότι υποψήφιοι στόχοι μπορούν να είναι όσοι από τους κόμβους έχουν πατέρα τον εικονικό κόμβο *Root*, δηλαδή δεν έχουν κάποιον άλλο κόμβο από πάνω τους. Επίσης, ελέγχονται το **πρώτο** και το **τρίτο κριτήριο** σε σχέση με τη θέση που πρέπει να πάρει το *White Bird* πάνω από τον στόχο. Ουσιαστικά, ελέγχεται εάν

το *White Bird* μπορεί να πάρει την κατακόρυφη θέση που απαιτείται πάνω από τον υποψήφιο στόχο ώστε τελικά να μπορέσει να αφήσει το *egg bomb*.

Τέλος, ακόμη μία παρατήρηση μας οδήγησε στη διαπίστωση ότι δεν έχει νόημα να είναι εφικτοί στόχοι κόμβοι, οι οποίοι βρίσκονται *πιο δεξιά από το δεξιότερο κόμβο τύπου Pig* της σκηνής. Αυτό δεν ισχύει για κόμβους οι οποίοι αν χτυπηθούν μπορούν να επηρεάσουν την υπόλοιπη σκηνή, όπως για παράδειγμα οι κόμβοι τύπου *Rolling Woods* ή *Rolling Stones* (Εικόνα 3.4).



Εικόνα 3.4 Εφικτοί Κόμβοι Τύπου Rolling Stone

3.5. Επιλογή Στόχου Συνδυάζοντας Μοντέλα Παλινδρόμησης

Στις προηγούμενες ενότητες του Κεφαλαίου 3 δόθηκαν αναλυτικά η δενδρική αναπαράσταση της σκηνής του παιχνιδιού, καθώς και τα κριτήρια με τα οποία ένας κόμβος του δένδρου χαρακτηρίζεται *εφικτός (feasible)* ή *ανέφικτος* και με βάση αυτόν τον χαρακτηρισμό μπορεί να αποτελέσει υποψήφιο στόχο. Στην παρούσα ενότητα παρουσιάζεται η προσέγγιση με την οποία επιλέγεται ο *τελικός στόχος*, δηλαδή ο κόμβος ο οποίος θα χτυπηθεί με την εκτέλεση της βολής. Η προσέγγιση για την επιλογή του τελικού στόχου που προτείνεται στα πλαίσια της παρούσας διατριβής, αφορά την μετατροπή του προβλήματος της επιλογής του στόχου σε ένα *μηχανισμό ensemble παλινδρόμησης (ensemble regression framework)*, δηλαδή ένα *μηχανισμό απόφασης που συνδυάζει πολλαπλά μοντέλα παλινδρόμησης (multi-model regression framework)*, καθένα από τα οποία έχει τις δικές του παραμέτρους και επιλέγεται

λαμβάνοντας υπόψη την πρόβλεψη για την αναμενόμενη ανταμοιβή που δίνουν οι παράμετροί του και έναν επιπλέον όρο. Ο επιπλέον όρος αυτός προκύπτει χρησιμοποιώντας ένα μηχανισμό (*UCB – Upper Confidence Bound*) [1] που μετατρέπει το πρόβλημα επιλογής του κατάλληλου *regressor* (γραμμικού παλινδρομητή) σε ένα *multi-armed bandit problem*, ενώ συγχρόνως προσφέρει εξισορρόπηση (*trade-off*) μεταξύ της εξερεύνησης (*exploration*) και της εκμετάλλευσης (*exploitation*) κατά τη διαδικασία μάθησης.

3.5.1. Γραμμικά Μοντέλα Παλινδρόμησης

Στην ενότητα αυτή δίνονται βασικά στοιχεία που αφορούν την παλινδρόμηση (*regression*). Η παλινδρόμηση χρησιμοποιείται ως μία μεθοδολογία πρόβλεψης της τιμής μίας ή περισσότερων συνεχών μεταβλητών t , δεδομένου ενός D -διάστατου διανύσματος x μεταβλητών εισόδου. Δηλαδή, δοθέντος ενός συνόλου N παρατηρήσεων $\{x_n\}$, με $n = 1, \dots, N$ με αντίστοιχες αποκρίσεις $\{t_n\}$, στόχος είναι η πρόβλεψη της τιμής t για μία νέα είσοδο x . Με μία απλή προσέγγιση, αυτό μπορεί να επιτευχθεί κατασκευάζοντας μία κατάλληλη συνάρτηση $y(x)$, της οποίας οι αποκρίσεις για νέα x αποτελούν τις προβλέψεις για τις αντίστοιχες τιμές των μεταβλητών t . Η πιθανοτική (*probabilistic*) προσέγγιση επικεντρώνεται στη μοντελοποίηση της υπό-συνθήκη κατανομής $p(t|x)$ (*conditional predictive distribution*). Η αβεβαιότητα εκφράζεται με τον θόρυβο ϵ των δεδομένων ως εξής:

$$t = y(x) + \epsilon \quad (3.1)$$

Η $y(x)$ της Σχέσης (3.1) καλείται μοντέλο. Στο πιο απλό γραμμικό μοντέλο παλινδρόμησης, οι μεταβλητές εισόδου εκφράζονται ως γραμμικός συνδυασμός των παραμέτρων του μοντέλου ως εξής:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_D x_D \quad (3.2)$$

όπου $\mathbf{x} = (x_1, \dots, x_D)^T$. Το παραπάνω μοντέλο είναι γνωστό ως απλή γραμμική παλινδρόμηση. Η κύρια ιδιότητα του μοντέλου αυτού είναι η γραμμικότητά του σε σχέση με τις παραμέτρους w_0, \dots, w_D . Όμως, είναι γραμμικό και σε σχέση με τις

μεταβλητές x_i , γεγονός το οποίο προσδίδει σημαντικούς περιορισμούς στο μοντέλο. Για το λόγο αυτό, καταλήγουμε στο μη-γραμμικό μοντέλο:

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^M w_j \phi_j(\mathbf{x}) \quad (3.3)$$

όπου $\mathbf{w} = (w_1, \dots, w_M)^T$ και $\boldsymbol{\phi} = (\phi_1, \dots, \phi_M)^T$. Οι $\phi_j(\mathbf{x})$ είναι γνωστές ως *συναρτήσεις βάσης*, ή αλλιώς *basis functions*. Κύρια ιδιότητα του μοντέλου της Σχέσης (3.3) είναι η γραμμικότητα ως προς τις παραμέτρους \mathbf{w} , με την ταυτόχρονη μη-γραμμικότητα ως προς τις μεταβλητές εισόδου \mathbf{x} .

Στην παρούσα διατριβή, για το πρόβλημα της *επιλογής στόχου* θεωρούμε τις επιστρεφόμενες ανταμοιβές (*σκορ*) t_n των *δειγμάτων* \mathbf{x}_n (*feature vectors*), τα οποία παίρνουμε ακολουθιακά, βολή προς βολή για κάθε εφικτό κόμβο του δένδρου. Θεωρούμε το γραμμικό μοντέλο M -τάξης για τις ανταμοιβές t_n ως ακολούθως:

$$t_n = \sum_{i=1}^M w_i \phi_i(\mathbf{x}_n) + \epsilon_n = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) + \epsilon_n \quad (3.4)$$

όπου $\epsilon_n \sim \mathcal{N}(0, \beta^{-1})$ ονομάζεται *Γκαουσιανός προσθετικός θόρυβος* (*Gaussian additive noise*) και $\mathbf{w} = (w_1, \dots, w_M)^T$ είναι οι M άγνωστοι παράμετροι του μοντέλου. Η Σχέση (3.4) εκφράζει την ανταμοιβή t_n ως ένα σταθμισμένο γραμμικό άθροισμα M συναρτήσεων βάσης, με $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x}))^T$.

3.5.2. Κατασκευή Συναρτήσεων Βάσης

Αφού ορίσαμε το μη-γραμμικό μοντέλο της Σχέσης (3.4), θεωρήσαμε ως συναρτήσεις βάσης *Γκαουσιανές συναρτήσεις πυρήνα* (*Gaussian kernels*). Αρχικά, συλλέξαμε δεδομένα (*feature vectors*) από διάφορα επίπεδα του παιχνιδιού, προκειμένου να αποκτήσουμε μία γενική μορφή των δεδομένων. Για την κατασκευή των συναρτήσεων βάσης ακολουθήσαμε δύο στρατηγικές:

1. **Ομαδοποίηση (clustering)** χρησιμοποιώντας τον αλγόριθμο ομαδοποίησης *k-means* για το διαχωρισμό του χώρου σε M ομάδες, για προκαθορισμένο M . Τα χαρακτηριστικά (μ, σ^2) των ομάδων που παράγονται, αποτελούν τις τιμές των παραμέτρων των Γκαουσιανών συναρτήσεων βάσης.
2. **Διαχωρισμός πλέγματος (gridding)** χωρίζοντας τον χώρο με ένα πλέγμα σε περιοχές, δηλαδή κατασκευάζοντας ένα ***d*-διάστατο grid**, όπου d η διάσταση των *feature vectors*. Αυτό επιτυγχάνεται χωρίζοντας κάθε διάσταση (j) σε k_j διαστήματα $\forall j = 1, \dots, d$. Να σημειωθεί ότι το *grid* δεν έχει διαμορφωθεί ομοιόμορφα, ενώ το πλήθος k_j των διαστημάτων εξαρτάται από το *έυρος τιμών* που μπορεί να πάρει το συγκεκριμένο χαρακτηριστικό. Έτσι, ο αριθμός συναρτήσεων βάσης που προκύπτουν είναι $M = k_1 * k_2 * \dots * k_d$. Κάθε συνάρτηση βάσης έχει μέσο μ_j το κέντρο κάθε τμήματος. Η τιμή της διακύμανσης σ_j^2 επιλέγεται με τέτοιο τρόπο ώστε να υπάρχει **επικάλυψη (overlapping)** μεταξύ **γειτονικών** συναρτήσεων βάσης, όπως προέκυψε και από την πειραματική μελέτη όπου διαπιστώσαμε ότι η μέθοδος παρουσιάζει **καλύτερη συμπεριφορά**.

Τελικά, ακολουθήσαμε τη δεύτερη στρατηγική για τον προσδιορισμό των συναρτήσεων βάσης. Οι συναρτήσεις βάσης που θεωρήσαμε για το μοντέλο μας έχουν την μορφή:

$$\phi_i(\mathbf{x}) = \exp\left\{-\left(\sum_{k=1}^3 \frac{(x_k - m_{ik})^2}{2\sigma_{ik}^2}\right)\right\} \quad (3.5)$$

Αξίζει να σημειώσουμε ότι το πλήθος M των συναρτήσεων βάσης δεν επηρεάζει άμεσα την απόδοση της μεθόδου, παρά μόνο σε επίπεδο ρυθμού μάθησης. Κατά την πειραματική μας μελέτη διαπιστώσαμε ότι $M = 150$ συναρτήσεις βάσης αποτελούν ικανοποιητικό πλήθος για το μοντέλο μας.

3.5.3. Bayesian Ensemble Regression

Έστω μία ακολουθία παρατηρήσεων, πιο συγκεκριμένα *input vectors*, $\{\mathbf{x}_k\}_{k=1}^n$ με τις αντίστοιχες αποκρίσεις $t_{1:n} = \{t_k\}_{k=1}^n$. Στη συνέχεια, δοθέντων των παραμέτρων παλινδρόμησης \mathbf{w}, β μπορούμε να κατασκευάσουμε την *από-κοινού υπό-συνθήκη συνάρτηση πυκνότητας πιθανότητας (conditional probability density)* για τις αποκρίσεις $t_{1:n}$, λαμβάνοντας υπόψη τη Σχέση (3.4):

$$p(t_{1:n}|\mathbf{w}, \beta) = \mathcal{N}(t_{1:n}|\Phi_n \mathbf{w}, \beta^{-1} I_n) \quad (3.6)$$

Όπου $\Phi_n = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \dots, \boldsymbol{\phi}(\mathbf{x}_n)]^T$, καλείται *πίνακας σχεδίασης (design matrix)* μεγέθους $n \times n$ και I_n είναι ο μοναδιαίος πίνακας n -τάξης.

Ένα ακόμη σημαντικό ζήτημα που πρέπει να λάβουμε υπόψη όταν χρησιμοποιούμε μοντέλα παλινδρόμησης είναι η πολυπλοκότητα, ή αλλιώς η *τάξη* M του μοντέλου, στην περίπτωσή μας ο αριθμός των συναρτήσεων βάσης. Γενικά, η πολυπλοκότητα του μοντέλου πρέπει να εξαρτάται και να εξάγεται ανάλογα με το μέγεθος του διαθέσιμου συνόλου δεδομένων. Όλα τα παραπάνω καθιστούν την επιλογή της τάξης M του μοντέλου αρκετά πολύπλοκη και δύσκολη υπόθεση. Η επιλογή μικρού M συνήθως οδηγεί σε *underfitting*, ενώ μεγάλες τιμές για το M οδηγούν σε *overfitting*. Δηλαδή, επιλέγοντας μικρό M υπάρχει η πιθανότητα το μοντέλο μας να είναι τόσο απλό, ώστε να μην είναι σε θέση να *κατανοήσει* βασικές ιδιότητες του συνόλου δεδομένων και ως αποτέλεσμα να μην έχει καλή ικανότητα γενίκευσης (*generalization*). Επίσης, επιλέγοντας μεγάλο M μπορεί να οδηγηθούμε σε πολύπλοκο μοντέλο, το οποίο πιθανότατα θα *υπερεκπαιδευτεί*, δηλαδή θα προσπαθεί να *απομνημονεύσει* το σύνολο των δεδομένων παρουσιάζοντας μικρή γενικευτική ικανότητα σε δεδομένα για τα οποία δεν έχει εκπαιδευτεί. Μία προσέγγιση η οποία αντιμετωπίζει το παραπάνω πρόβλημα εφαρμόζεται στα [12], [2] και αφορά την *Μπεϋζιανή αντιμετώπιση* της γραμμικής παλινδρόμησης, όπου τελικά η πολυπλοκότητα του μοντέλου παλινδρόμησης ορίζεται από τα ίδια τα δεδομένα, υπό την έννοια ότι παράμετροι του μοντέλου οι οποίοι δεν *υποστηρίζονται* από τα δεδομένα *εξαλείφονται* αυτόματα.

Η Μπεϋζιανή προσέγγιση προτείνει μία *εκ των προτέρων κατανομή (prior)* για τις παραμέτρους \mathbf{w} , που θεωρούμε ότι είναι η Γκαουσιανή:

$$p(\mathbf{w}|a) = \mathcal{N}(\mathbf{w}|\vec{\mathbf{0}}, \alpha^{-1}I_M) \quad (3.7)$$

όπου η *υπερπαραμέτρος (hyperparameter)* α είναι η *ακρίβεια (precision)* των παραμέτρων του μοντέλου, ενώ I_M είναι ο μοναδιαίος πίνακας. Συνεχίζοντας, παίρνουμε την *εκ των υστέρων κατανομή (posterior)* για τις παραμέτρους \mathbf{w} , η οποία είναι επίσης Γκαουσιανή:

$$p(\mathbf{w}|t_{1:n}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_n, \Sigma_n) \quad (3.8)$$

όπου η μέση τιμή $\boldsymbol{\mu}_n$ και ο πίνακας διασπορών Σ_n ορίζονται ως εξής:

$$\boldsymbol{\mu}_n = \beta \Sigma_n \Phi_n^T t_{1:n} \quad (3.9)$$

και

$$\Sigma_n = (\beta \Phi_n^T \Phi_n + aI)^{-1} \quad (3.10)$$

Κατά την διαδικασία επιλογής του στόχου πρέπει για κάθε εφικτό κόμβο να πάρουμε την πρόβλεψη για την αναμενόμενη ανταμοιβή που θα δώσει αν χτυπηθεί. Έστω ότι εξετάζουμε ένα κόμβο \mathbf{x}_* του δένδρου, μπορούμε να κάνουμε την *πρόβλεψη* για την αναμενόμενη ανταμοιβή (σκορ) t_* που θα σημειωθεί με τη βολή, σύμφωνα με την *εκ των υστέρων κατανομή (predictive distribution)*:

$$p(t_*|t_{1:n}, \alpha, \beta) = \mathcal{N}(t_*|\boldsymbol{\mu}_n^T \boldsymbol{\phi}(\mathbf{x}_*), \beta_*) \quad (3.11)$$

όπου

$$1/\beta_* = 1/\beta + \boldsymbol{\phi}(\mathbf{x}_*)^T \Sigma_n \boldsymbol{\phi}(\mathbf{x}_*) \quad (3.12)$$

Ο πρώτος όρος της Σχέσης (3.12) αντιστοιχεί στον *θόρυβο* που υπάρχει στα δεδομένα, ενώ ο δεύτερος όρος εκφράζει την *αβεβαιότητα* που υπάρχει όσον αφορά τις παραμέτρους \mathbf{w} του μοντέλου. Τέλος, αξίζει να σημειώσουμε ότι συνεχώς καταφθάνουν νέα δεδομένα, δηλαδή καθώς $n \rightarrow \infty$, και ο δεύτερος όρος της Σχέσης (3.12) τείνει στο μηδέν και τελικά η *διακύμανση* (*variance*) της (3.11) εξαρτάται αποκλειστικά από τον θόρυβο β των δεδομένων.

3.5.4. Μηχανισμός UCB για την Επιλογή Regressor

Η προσέγγιση για την επιλογή του τελικού στόχου που προτείνεται στα πλαίσια της παρούσας διατριβής, αφορά την μετατροπή του προβλήματος της επιλογής του στόχου σε ένα *μηχανισμό ensemble παλινδρόμησης* (*ensemble regression framework*), δηλαδή ένα μηχανισμό απόφασης που συνδυάζει *πολλαπλά μοντέλα παλινδρόμησης* (*multi-model regression framework*), καθένα από τα οποία έχει τις δικές του παραμέτρους $\theta_j = \{\mathbf{w}_j, \beta_j\}$. Σε κάθε βολή λαμβάνοντας υπόψη τον τύπο πουλιού-αντικειμένου γίνεται μία πρόβλεψη για την αναμενόμενη ανταμοιβή κάθε αντικειμένου της σκηνης. Πιο συγκεκριμένα, στην παρούσα μέθοδο θεωρούμε $5 \times 7 = 35$ *ξεχωριστά μοντέλα παλινδρόμησης* (*regressors*) για κάθε δυνατό τύπο πουλιού-αντικειμένου. Σε κάθε βολή, οι κατάλληλοι *regressors* δίνουν την πρόβλεψη για την αναμενόμενη ανταμοιβή κάθε κόμβου του δένδρου που αποτελεί εφικτό στόχο.

Ο τελικός στόχος προκύπτει από την *επιλογή του κατάλληλου regressor*. Στη μέθοδο που προτείνεται στα πλαίσια της παρούσας διατριβής, μετατρέψαμε τον μηχανισμό απόφασης για την επιλογή του regressor σε ένα *multi-armed bandit problem*, γεγονός που προσφέρει εξισορρόπηση (*trade-off*) μεταξύ της εξερεύνησης (*exploration*) και εκμετάλλευσης (*exploitation*) κατά τη διαδικασία της μάθησης. Με άλλα λόγια, κρίνεται απαραίτητο ο πράκτορας να αξιοποιεί κατάλληλα την *γνώση* που έχει αποκτήσει κατά τη διάρκεια της εκπαίδευσης, αλλά είναι επίσης αναγκαίο να *ανακαλύψει* και άλλες ενέργειες που δεν έχουν χρησιμοποιηθεί σε προηγούμενα βήματα, ώστε να *μάθει* τελικά τις ιδιότητες των πουλιών και των αντικειμένων της σκηνης.

Ειδικότερα, εφαρμόσαμε τον αλγόριθμο *UCB* (*Upper Confidence Bound*) [1] για την επιλογή του regressor που θα δώσει τον τελικό στόχο. Επιπλέον, πρέπει να σημειώσουμε ότι η επιλογή του regressor περιορίζεται στους κόμβους του δένδρου, οι οποίοι αποτελούν εφικτούς στόχους. Σύμφωνα με τον *UCB* αλγόριθμο, κάθε βραχίονας (*arm*) διατηρεί τη συχνότητα επιλογής του, η οποία συμβολίζεται με $n_{f(q)}$, όπου $f(q)$ αντιστοιχεί στον τύπο του μοντέλου παλινδρόμησης για το συγκεκριμένο κόμβο q σε σχέση με τον τύπο πουλιού της σφεντόνας. Τελικά, ο αλγόριθμος επιλέγει *greedily* τον βραχίονα $f(j^*)$ με βάση τον τύπο της Σχέσης (3.13):

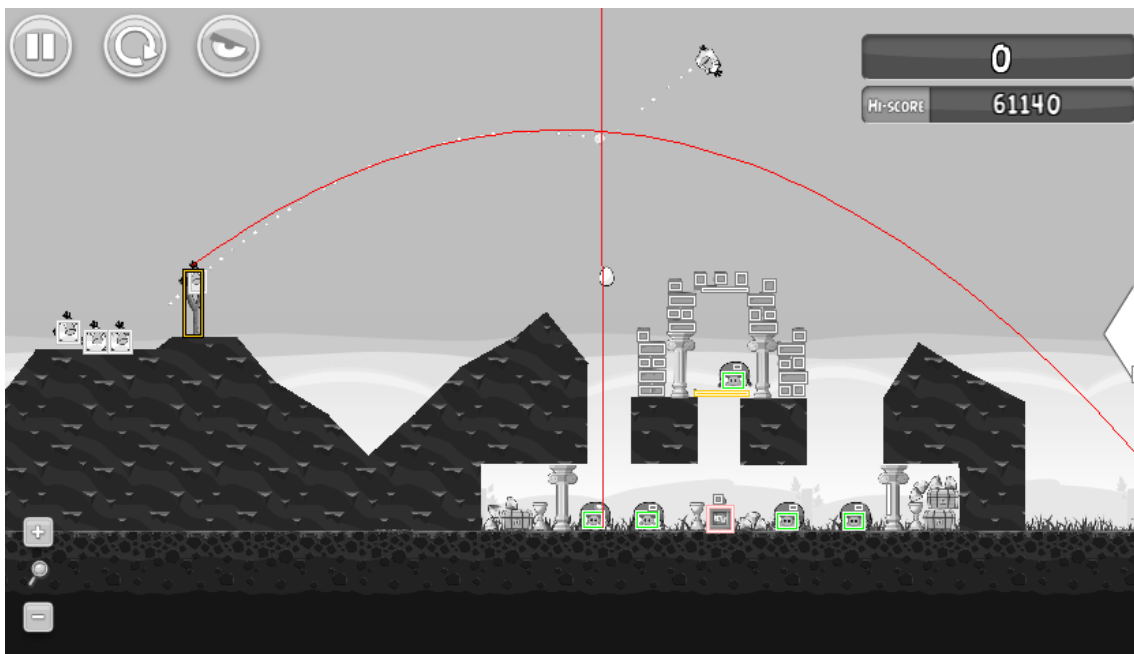
$$j^* = \arg \max_q \left\{ \left(\mu_{n_{f(q)}}^{f(q)} \right)^T \phi(x_q) + C \sqrt{\frac{2 \ln N}{n_{f(q)}}} \right\} \quad (3.13)$$

όπου με N συμβολίζουμε τον συνολικό αριθμό βολών που έχουν πραγματοποιηθεί μέχρι αυτή τη στιγμή, ενώ με x_q το *feature vector* του κόμβου q . Ο όρος $\mu_{n_{f(q)}}^{f(q)}$ δίνει την τρέχουσα εκτίμηση των συντελεστών του μοντέλου που αντιστοιχεί στο συγκεκριμένο τύπου πουλιού-αντικειμένου της βολής. Τέλος, ο όρος C είναι μία σταθερά, η οποία καθ' όλη τη διάρκεια των πειραμάτων μας είχε μία προκαθορισμένη τιμή ($C = 3000$). Η τιμή της σταθεράς C καθορίζει τη βαρύτητα του κάθε όρου της Σχέσης (3.13) για την επιλογή του regressor, δηλαδή αν ο regressor θα επιλεγεί βάση της πρόβλεψης που δίνουν οι παράμετροί του, ή αν θα επιλεγεί λόγω της συχνότητας επιλογής του σε προηγούμενα βήματα. Η τιμή της σταθεράς C σταθμίζει τη βαρύτητα των δύο αυτών όρων.

3.6. Επιλογή tap time

Αφού έχει γίνει η επιλογή του στόχου, με βάση την επιλογή του regressor του προηγούμενου βήματος, το επόμενο βήμα είναι η επιλογή του tap που θα εκτελέσει ο πράκτορας. Ειδικότερα, για κάθε τύπο πουλιού έχουμε διαφορετικό *tap interval* (ποσοστό της τροχιάς που θα εκτελεστεί το tap) και η ακριβής στιγμή που θα εκτελεστεί το tap υπολογίζεται από το *Trajectory Planner Component* δοθέντος του *tap interval*. Πιο συγκεκριμένα:

- για τα *Red* και *Black Birds* δεν εκτελείται tap
- για τα *Blue Birds* το tap εκτελείται στο 75% της τροχιάς από τη σφεντόνα μέχρι το πρώτο σημείο σύγκρουσης
- για τα *Yellow Birds* το tap εκτελείται ανάλογα με τη φύση της βολής. Ειδικότερα, στην περίπτωση των *direct shots* το tap εκτελείται στο 90%, ενώ στην περίπτωση των *high-arching shots* το tap εκτελείται στο 95%
- κλείνοντας, για τα *White Birds* το tap εκτελείται ακριβώς όταν το πουλί βρίσκεται πάνω από τον στόχο, προκειμένου να αφήσει στον καταλληλότερο χρόνο το *egg bomb*, όπως φαίνεται στην Εικόνα 3.5.



Εικόνα 3.5: Tap Timing Διαδικασία για το White Bird

3.7. Online Διαδικασία Μάθησης

Το τελικό στάδιο της προτεινόμενης μεθόδου αφορά τη διαδικασία μάθησης, δηλαδή την ενημέρωση των παραμέτρων του γραμμικού παλινδρομητή (*regressor*) που επιλέχθηκε για την πραγματοποίηση της βολής. Πιο συγκεκριμένα, χρησιμοποιούμε μία *online αναδρομική διαδικασία μάθησης* [2], λόγω της ακολουθιακής φύσης των δεδομένων (μετά από κάθε βολή επιστρέφεται η ανταμοιβή της). Έτσι, αμέσως μετά την εκτέλεση της βολής και την επιστροφή της πραγματικής ανταμοιβής, πραγματοποιείται η διαδικασία μάθησης. Η **βασική ιδέα** είναι ότι για κάθε εκτίμηση λαμβάνουμε υπόψη την *εκ των υστέρων* κατανομή του προηγούμενου βήματος, ως την *εκ των προτέρων* κατανομή του τρέχοντος βήματος.

Έστω ότι έχει επιλεγεί ο regressor $k \triangleq f(j^*)$, για να γίνει πρόβλεψη της ανταμοιβής που θα επιστραφεί αν ο στόχος είναι ένας κόμβος με *feature vector* \mathbf{x}_{n_k+1} . Μετά την εκτέλεση της βολής επιστρέφεται μία ανταμοιβή t_{n_k+1} (σκορ). Για την ενημέρωση των παραμέτρων του regressor k χρησιμοποιούμε την *εκ των υστέρων κατανομή* δοθέντων των ανταμοιβών $t_{1:n_k}$ των προηγούμενων n_k επιλογών του regressor k , δηλαδή διατηρούμε *εικονικά* ένα *trace παρατηρήσεων* για κάθε γραμμικό παλινδρομητή:

$$p(\mathbf{w}_k | t_{1:n_k}) = \mathcal{N}(\mathbf{w}_k | \boldsymbol{\mu}_{n_k}^k, \boldsymbol{\Sigma}_{n_k}^k) \quad (3.14)$$

Η μεταβλητή της νέας ανταμοιβής μετά την εκτέλεση της βολής ακολουθεί την κανονική κατανομή:

$$p(t_{n_k+1} | \mathbf{w}_k) = \mathcal{N}(t_{n_k+1} | \mathbf{w}_k^T \boldsymbol{\phi}(x_{n_k+1}), \beta_k) \quad (3.15)$$

Τελικά, παίρνουμε την *εκ των υστέρων κατανομή* για τις παραμέτρους \mathbf{w} του regressor k ως εξής:

$$\begin{aligned} p(\mathbf{w}_k | t_{1:n_k+1}) &= p(t_{n_k+1} | \mathbf{w}_k) p(\mathbf{w}_k | t_{1:n_k}) \\ &= \mathcal{N}(\mathbf{w}_k | \boldsymbol{\mu}_{n_k+1}^k, \boldsymbol{\Sigma}_{n_k+1}^k) \end{aligned} \quad (3.16)$$

όπου οι Γκαουσιανές παράμετροι μπορούν να γραφούν αναδρομικά ως ακολούθως:

$$\Sigma_{n_k+1}^k = [(\Sigma_{n_k}^k)^{-1} + \beta_k \boldsymbol{\phi}(\mathbf{x}_{n_k+1})^T \boldsymbol{\phi}(\mathbf{x}_{n_k+1})]^{-1} \quad (3.17)$$

και

$$\boldsymbol{\mu}_{n_k+1}^k = \Sigma_{n_k+1}^k [\beta_k \boldsymbol{\phi}^T(\mathbf{x}_{n_k+1}) t_{n_k+1} + (\Sigma_{n_k}^k)^{-1} \boldsymbol{\mu}_{n_k}^k] \quad (3.18)$$

Οι παραπάνω σχέσεις αποτελούν μία αναδρομική διαδικασία εκτίμησης των παραμέτρων του μοντέλου παλινδρόμησης. Στο αρχικό στάδιο (βήμα 0) της διαδικασίας οι παράμετροι \mathbf{w}_k του μοντέλου ακολουθούν την εκ των προτέρων $p(\mathbf{w}_k)$ κατανομή με μέση τιμή $\boldsymbol{\mu}_0^k = 0$ και πίνακα διασπορών $\Sigma_0^k = a^{-1}I$. Τέλος, πρέπει να σημειωθεί ότι η ακολουθιακή ενημέρωση των παραμέτρων των μοντέλων που χρησιμοποιούνται κατά της διάρκεια της διαδικασίας μάθησης, μας επιτρέπει να ελέγξουμε την επίδραση της διαδικασίας στις παραμέτρους του συνολικού μοντέλου (*ensemble regression model*), επιλέγοντας τα κατάλληλα επίπεδα του παιχνιδιού ανάλογα με την εκπαίδευση που κρίνουμε ότι πρέπει να πραγματοποιηθεί.

ΚΕΦΑΛΑΙΟ 4. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

- 4.1 Εισαγωγή
 - 4.2 Πρώτη Σειρά Πειραμάτων – Επίπεδα 1-21
 - 4.3 Δεύτερη Σειρά Πειραμάτων – Επίπεδα 22-42
 - 4.4 Angry Birds AI Competition
 - 4.5 Φάσεις του Angry Birds AI Competition
 - 4.6 Αποτελέσματα Angry Birds AI Competition 2014
-

4.1. Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται δύο σειρές πειραμάτων που διεξήχθησαν στα πλαίσια αυτής της εργασίας, σε μία προσπάθεια ανάλυσης της συμπεριφοράς και της απόδοσης του πράκτορα *AngryBER*. Ειδικότερα, δίνονται πίνακες με αριθμητικά στοιχεία για τα σκορ που επιτεύχθηκαν σε κάθε επίπεδο. Και για τις δύο σειρές πειραμάτων έγινε σύγκριση των αποτελεσμάτων του πράκτορα *AngryBER* με εκείνων του *Naive*, ο οποίος αποτελεί ένα πολύ καλό μέτρο σύγκρισης. Επιπλέον στην 1^η σειρά γίνεται σύγκριση της επίδοσης του πράκτορα *AngryBER* με εκείνες των πρακτόρων που συμμετείχαν στον διαγωνισμό *Angry Birds AI Competition 2013*. Στα πλαίσια της εργασίας εστίασαμε την προσοχή μας στο επεισόδιο “*Poached Eggs*” για τα πειράματα και την ανάλυση της επίδοσης του πράκτορα *AngryBER*, καθώς τα προβλήματα στην ανίχνευση και ανάλυση της σκηνής ήταν λιγότερα από ότι σε άλλα επεισόδια του παιχνιδιού, κυρίως λόγω του *background* της σκηνής. Τέλος, πρέπει να τονιστεί ότι οι πίνακες που παρουσιάζονται στη συνέχεια του συγκεκριμένου κεφαλαίου συμπληρώθηκαν αφού έγινε η *εκπαίδευση (training)* του πράκτορα *AngryBER*, δηλαδή μετά από ικανό αριθμό περασμάτων των πρώτων 42 επιπέδων του επεισοδίου “*Poached Eggs*”.

4.2. Πρώτη Σειρά Πειραμάτων – Επίπεδα 1-21

Στον Πίνακα 4.1 παρουσιάζονται τα αποτελέσματα της επίδοσης του πράκτορα *AngryBER* στα πρώτα 21 επίπεδα του επεισοδίου “*Poached Eggs*”. Τα αποτελέσματα αυτά προκύπτουν μετά από δέκα (10) ανεξάρτητα περάσματα που έκανε ο πράκτορας στα επίπεδα αυτά, μετά τη φάση της εκπαίδευσής του.

Πίνακας 4.1: Στατιστικά του Πράκτορα “*AngryBER*” για 1-21 Επίπεδα του Επεισοδίου “*Poached Eggs*”

Level	AngryBER Agent			Naive Agent	High Scores of IJCAI
	Mean Scores	Max Scores	Min Scores		2013 Angry Birds Competition
1	28740 ± 165.60	28940	28400	29510	31210
2	51370 ± 2875.06	52360	43190	52230	60400
3	41917 ± 9.49	41920	41890	40620	42240
4	27049 ± 3485.57	29110	20350	20680	36770
5	65483 ± 2272.90	69800	63350	55160	65850
6	33961 ± 2859.97	35200	26020	16070	36180
7	26449 ± 7767.84	45650	20430	21590	49120
8	53191 ± 8782.17	57110	28240	25730	57780
9	36053 ± 7392.18	52320	24410	35490	51480
10	50547 ± 11221.86	65560	37980	32600	68740
11	55211 ± 7756.42	60030	33490	46760	59070
12	50151 ± 5502.49	54800	36530	54070	58600
13	43945 ± 7214.32	50920	25200	49470	50360
14	70181 ± 7176.07	79330	56620	50590	65640
15	43185 ± 3998.41	51620	38460	46430	55300
16	60430 ± 3295.07	63650	53680	55210	66550
17	48242 ± 3745.76	52050	39760	48140	54750
18	42975 ± 3145.83	48480	40210	49430	54500
19	30622 ± 4533.63	39110	21130	37920	38460
20	45523 ± 5643.80	54370	38870	36790	56050
21	66012 ± 5911.54	78100	58760	54240	75870
Total	971237 ± 14647.36	991370	943250	858730	1134920

Παρατηρώντας προσεκτικά τον Πίνακα 4.1 και βάση της εμπειρίας που αποκτήσαμε στο πλαίσιο της διατριβής, βλέπουμε ότι ο πράκτορας *AngryBER* καταφέρνει να ολοκληρώνει επιτυχώς κάθε επίπεδο με το πολύ δύο προσπάθειες, ενώ αξιοσημείωτη είναι η ικανότητά του να επιτυγχάνει αρκετά μεγάλα σκορ συγκρινόμενα με τα αποτελέσματα που έχουμε στη διάθεσή μας. Είναι αρκετά ενδιαφέρον το γεγονός ότι ο πράκτορας *AngryBER* κατάφερε να ξεπεράσει τα *highscores* των πρακτόρων του *Angry Birds AI Competition 2013* σε επτά (7) επίπεδα, τα οποία φαίνονται εντονότερα στη στήλη *Max Scores* του Πίνακα 4.1. Επίσης ένα πέρασμα όλων των επιπέδων του επεισοδίου γίνεται σε περίπου σαράντα (40) λεπτά, ενώ υπάρχει το υπάρχει το χρονικό μέτρο σύγκρισης των τριών (3) λεπτών κατά μέσο όρο ανά επίπεδο, το οποίο αντιστοιχεί σε 63 λεπτά για ολόκληρο το επεισόδιο. Το γεγονός αυτό φανερώνει την ικανότητα του πράκτορα να ολοκληρώνει σε μικρό χρονικό διάστημα επίπεδα και παράλληλα να επιτυγχάνει μεγάλα σκορ.

Ένα επιπλέον εντυπωσιακό γεγονός για τον πράκτορα *AngryBER* αποτελεί η ικανότητά του να *μαθαίνει* αρκετά γρήγορα, και με μικρό αριθμό περασμάτων των επιπέδων κατά την εκπαίδευση να ανακαλύπτει σύντομα *καλές πολιτικές* για το συγκεκριμένο παιχνίδι. Στο γεγονός αυτό βοηθάει η *καινοτόμα δενδρική αναπαράσταση* της σκηνης του παιχνιδιού σε συνδυασμό με το *μηχανισμό Πολλαπλής Ensemble Παλινδρόμησης* που προτείνουμε στο Κεφάλαιο 3. Ειδικότερα, τα δύο αυτά σημαντικά στοιχεία βοηθούν τον πράκτορα να ανακαλύψει αρκετά γρήγορα την επίδραση του κάθε πουλιού σε κάθε τύπο αντικειμένου της σκηνης.

4.3. Δεύτερη Σειρά Πειραμάτων – Επίπεδα 22-42

Στον Πίνακα 4.2 παρουσιάζονται τα στατιστικά αποτελέσματα της απόδοσης του πράκτορα *AngryBER* για τα επίπεδα 22-42 του επεισοδίου “*Poached Eggs*”. Τα αποτελέσματα αυτά προκύπτουν μετά από 10 ανεξάρτητα περάσματα που έκανε ο πράκτορας επίπεδα αυτά, μετά τη φάση της εκπαίδευσής του, όπως ακριβώς και στην προηγούμενη ενότητα αυτού του Κεφαλαίου.

Πίνακας 4.2: Στατιστικά του Πράκτορα “AngryBER” για τα Επίπεδα 22-42 του Επεισοδίου “Poached Eggs

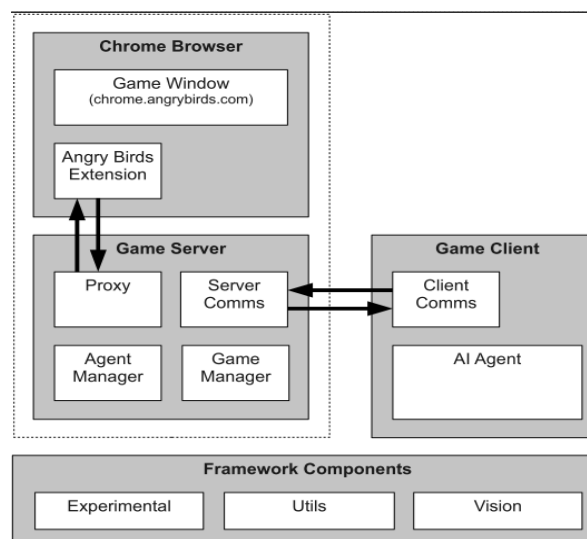
Level	AngryBER Agent			Naive Agent
	Mean Scores	Max Scores	Min Scores	
22	59084 ± 3551.70	62780	53680	59480
23	52325 ± 4412.11	60330	47140	0
24	95448 ± 4033.42	100740	89660	91300
25	50860 ± 7387.37	59850	33800	0
26	77688 ± 6297.01	85890	70140	0
27	44854 ± 4419.29	51320	40250	58270
28	47591 ± 7250.37	56480	38890	51040
29	51827 ± 4967.85	57500	43770	50300
30	24108 ± 2873.02	29710	19870	46240
31	38354 ± 4761.77	41340	28210	42330
32	90102 ± 3555.70	96720	85610	0
33	49104 ± 9467.97	63710	36860	32650
34	71964 ± 6021.88	79190	62310	71970
35	45316 ± 1591.72	47550	42420	0
36	36932 ± 5771.77	48350	29890	0
37	63031 ± 2579.42	66500	58840	0
38	32323 ± 3927.16	39970	25000	0
39	56497 ± 3762.06	61750	49410	0
40	39421 ± 7216.52	52210	30960	0
41	45120 ± 3141.78	46350	36390	0
42	68828 ± 7318.44	82330	61290	0
Total	1140777 ± 22035.21	1178350	1102820	503580

Τα πειράματα που έλαβαν χώρα στα παραπάνω επίπεδα ακολούθησαν το ίδιο μοτίβο διεξαγωγής με εκείνα της Ενότητας 4.2, μετά βέβαια από τη φάση εκπαίδευσης του πράκτορα, η οποία ήταν ανάλογη με εκείνη της προηγούμενης ενότητας. Στα πειράματα αυτά, ο πράκτορας κατάφερε να περιοριστεί στο χρονικό αυτό όριο, ολοκληρώνοντας επιτυχώς κάθε του πέρασμα σε λιγότερο από 63 λεπτά. Χρειάστηκε όμως περισσότερες από δύο προσπάθειες σε κάποια από τα επίπεδα. Αυτό μπορούμε να το αποδώσουμε στην αυξημένη δυσκολία των επιπέδων αυτών, καθώς και στην

ύπαρξη πουλιών με ειδικές ικανότητες, των οποίων ο χειρισμός από τον πράκτορα απαιτεί μεγάλη ακρίβεια. Εξάλλου, από την συνολική μας εμπειρία στο πλαίσιο της εργασίας αυτής αντιληφθήκαμε γρήγορα πως τα *White Birds* απαιτούν ιδιαίτερα λεπτούς και προσεκτικούς χειρισμούς. Παρά τις δυσκολίες, ο πράκτορας *AngryBER* ανταποκρίθηκε αρκετά καλά και έδωσε ικανοποιητικά αποτελέσματα, που αναδεικνύουν σε ακόμη μεγαλύτερο βαθμό την καλή απόδοση της μεθόδου που προτείνεται στο Κεφάλαιο 3. Αξιοσημείωτο είναι ότι ο *Naive Agent* δεν ολοκλήρωσε επιτυχώς κανένα επίπεδο μετά το 34°. Η δυσκολία του αυτή έγκειται στην μη-ανίχνευση των λόφων της κάθε σκηνής, αλλά και στο περίεργο χειρισμό των *White Birds*, κυρίως λόγω του *tap time* που έχει επιλεγεί για αυτόν τον τύπο πουλιού.

4.4. Angry Birds AI Competition

Οι ομάδες που συμμετέχουν στον εν λόγω διαγωνισμό καλούνται να παρουσιάσουν έναν πράκτορα που θα διαγωνιστεί σε άγνωστα για αυτόν επίπεδα. Για να εξασφαλιστεί η αξιοπιστία του διαγωνισμού οι συμμετέχοντες πράκτορες τρέχουν παράλληλα και επικοινωνούν μόνο με τον *server* των διοργανωτών, χρησιμοποιώντας ένα πρωτόκολλο *client/server* (Σχήμα 4.1), ο οποίος τους παρέχει πληροφορίες σχετικά με τα *screenshots*, τους πόντους των αντιπάλων κ.α. Κάθε πράκτορας λαμβάνοντας τα στοιχεία που ζητά από τον *server*, αποφασίζει ποιος είναι ο στόχος για την επόμενη βολή και στέλνει την πληροφορία αυτή στον *server*, ο οποίος εκτελεί τις βολές και καταγράφει τα σκορ που επιτυγχάνει ο καθένας.



Σχήμα 4.1: Το client/server Πρωτόκολλο

4.5. Φάσεις του Angry Birds AI Competition

Ο διαγωνισμός χωρίζεται στις εξής φάσεις: Δύο προκριματικές φάσεις, *qualification rounds* (QR1 και QR2), Προημιτελική Φάση (*quarter-finals*), Ημιτελική Φάση (*semi-finals*) και Τελικός (*Grand-Final*).

4.5.1. Προκριματικές φάσεις QR1, QR2

Αρχικά, όλες οι ομάδες δίνουν δύο προκριματικές φάσεις διαγωνιζόμενες στα ίδια επίπεδα. Οι πράκτορες των ομάδων καλούνται να λύσουν ένα προκαθορισμένο αριθμό επιπέδων, έχοντας και ένα χρονικό όριο. Με το πέρας των δύο αυτών φάσεων, οι οκτώ καλύτερες ομάδες, με βάση τον συνολικό αριθμό πόντων που συγκέντρωσαν, προκρίνονται στη φάση των «8».

4.5.2. Προημιτελική φάση - Quarter Finals

Στη φάση αυτή δημιουργούνται δύο *groups* των τεσσάρων ομάδων, έχοντας ως κριτήριο την κατάταξη των ομάδων μετά το πέρας της προηγούμενης προκριματικής φάσης. Μετά τον σχηματισμό των *group*, οι ομάδες κάθε *group* διαγωνίζονται παράλληλα στα ίδια επίπεδα του παιχνιδιού, έχοντας πάντα ένα προκαθορισμένο χρονικό όριο, και οι δύο καλύτεροι από κάθε *group*, με βάση τους συνολικούς πόντους που συγκέντρωσαν, προκρίνονται στην επόμενη φάση.

4.5.3. Ημιτελική Φάση – Semi Finals

Οι ομάδες διαγωνίζονται παράλληλα στα ίδια επίπεδα και οι δύο καλύτεροι προκρίνονται στον Τελικό.

4.5.4. Τελικός - Grand Final

Συμμετέχουν δύο ομάδες και νικήτρια είναι η ομάδα που θα συγκεντρώσει τους περισσότερους πόντους στα επίπεδα που θα κληθούν να αντιμετωπίσουν οι πράκτορές τους.

4.6. Αποτελέσματα Angry Birds AI Competition 2014

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα του *Angry Birds AI Competition 2014*, όπου συμμετείχε και ο πράκτορας *AngryBER* που προτείνουμε στα πλαίσια αυτής της εργασίας και κατάφερε να φτάσει μέχρι τον Τελικό και να κατακτήσει τη **2^η θέση** του διαγωνισμού. Πρέπει να τονιστεί ότι οι ομάδες που έφτασαν στον *Τελικό* συμμετείχαν για *πρώτη φορά* στο διαγωνισμό, ενώ η νικήτρια ομάδα του περσινού διαγωνισμού (συμμετείχε χωρίς αλλαγές στον πράκτορά της) δεν κατάφερε να προκριθεί στα προημιτελικά, γεγονός που φανερώνει ότι κάθε χρονιά οι δυνατότητες των πρακτόρων βελτιώνονται και εξελίσσονται σημαντικά. Παρακάτω παρουσιάζονται αναλυτικότερα τα αποτελέσματα του εν λόγω διαγωνισμού. Στον Πίνακα 4.3 δίνεται η κατάταξη των προκριματικών φάσεων, στον Πίνακα 4.4 τα group και η κατάταξη της προημιτελικής φάσης και τέλος στον Πίνακα 4.5 η κατάταξη της ημιτελικής φάσης και του Τελικού.

Πίνακας 4.3: Αποτελέσματα Προκριματικής Φάσης

Rank	Team	Total Score
1	DataLab Birds	423280
2	PlanA+	372810
3	s-Birds Avengers	361770
4	AngryDragons	317300
5	Impact Vactor	298390
6	AngryHEX	294170
7	IHSEV	292380
8	AngryBER	253820
9	BeauRivage	238080
10	RMIT RedBacks	188890
11	Auto Lilienthal	0

Πίνακας 4.4: Αποτελέσματα Προημιτελικής Φάσης

Quarter Final – Group 1			Quarter Final – Group 2		
Rank	Team	Score	Rank	Team	Score
1	DataLab Birds	346260	1	PlanA+	360920
2	AngryBER	224860	2	IHSEV	277530
3	Impact Vactor	173710	3	AngryHEX	129610
4	s-Birds Avengers	167860	4	AngryDragons	78970

Πίνακας 4.5: Αποτελέσματα Ημιτελικής Φάσης και Τελικού

Semi Final			Grand Final		
Rank	Team	Score	Final Results	Team	Score
1	DataLab Birds	232790	WINNER	DataLab Birds	406340
2	AngryBER	206680			
3	PlanA+	206620	RUNNER-UP	AngryBER	243880
4	IHSEV	93100			

ΚΕΦΑΛΑΙΟ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στην παρούσα διατριβή ασχοληθήκαμε με την σχεδίαση και την κατασκευή ενός ευφυούς πράκτορα (*AngryBER*) για το δημοφιλές παιχνίδι “*Angry Birds*”, η λειτουργία του οποίου βασίζεται σε πολλαπλά μοντέλα παλινδρόμησης. Η κατασκευή του βασίστηκε στο λογισμικό [18] που παρέχεται από τους διοργανωτές του *Angry Birds AI Competition*, το οποίο διαθέτει δυνατότητες ανίχνευσης των αντικειμένων της σκηνής του παιχνιδιού και σχεδιασμού των βολών. Αρχικά, προτείνουμε μία καινοτόμα και αποδοτική *δενδρική αναπαράσταση* της σκηνής του παιχνιδιού, η οποία εκμεταλλεύεται στο έπακρο τις τοπολογικές εξαρτήσεις μεταξύ των αντικειμένων της σκηνής και διευκολύνει τη διαδικασία εξαγωγής χαρακτηριστικών. Η *δενδρική δομή* χτίζεται *ιεραρχικά (bottom-up)*, ενώ η πολυπλοκότητά της είναι χαμηλή, στοιχείο που κρατά την γενική πολυπλοκότητα της μεθόδου σε χαμηλά επίπεδα. Στη συνέχεια για την επιλογή του στόχου, παρουσιάσαμε ένα μηχανισμό απόφασης που βασίζεται σε πολλαπλά μοντέλα *Μπεϋζιανής Ensemble Παλινδρόμησης*. Κάθε μοντέλο από αυτά ενεργοποιείται και δίνει μία πρόβλεψη για τον εκάστοτε στόχο λαμβάνοντας υπόψη τον τύπο πουλιού-αντικειμένου. Τα μοντέλα αυτά δρουν *ανταγωνιστικά* μεταξύ τους και τελικά επιλέγεται το μοντέλο που δίνει την καλύτερη πρόβλεψη. Τέλος, μετά την εκτέλεση της βολής και την απολαβή της πραγματικής ανταμοιβής (*σκορ*) εκτελείται ένας *online μηχανισμός μάθησης*, όπου ενημερώνονται οι παράμετροι του μοντέλου που έδωσε την πρόβλεψη για τον στόχο [14].

Πραγματοποιήθηκαν σειρές πειραμάτων για τον έλεγχο της απόδοσης του πράκτορα *AngryBER* σε επίπεδα του παιχνιδιού για τα οποία υπάρχουν ήδη καταγεγραμμένα αποτελέσματα και με τον τρόπο αυτό συγκρίναμε την απόδοση του πράκτορα

AngryBER και με άλλες προσεγγίσεις στο πεδίο αυτό. Τα πειραματικά αποτελέσματα έδειξαν ότι η μέθοδος που προτείνουμε έχει δυνατότητες και απόδοση που ανταγωνίζεται με τον καλύτερο δυνατό τρόπο τα αποτελέσματα των υπολοίπων προσεγγίσεων. Το παραπάνω στοιχείο έγινε φανερό με την κατάκτηση της 2^{ης} θέσης του *Angry Birds AI Competition 2014*, όπου συμμετείχε ο πράκτορας *AngryBER* και έντεκα (11) ακόμη ομάδες από όλο τον κόσμο. Κατά τη διάρκεια του διαγωνισμού, ο πράκτορας *AngryBER* κατάφερε να ολοκληρώσει επιτυχώς επίπεδα αυξημένης δυσκολίας, ανάμεσα σε αυτά και κάποια από τα επίπεδα του *Τελικού* του *Angry Birds AI Competition 2013*, τα οποία κλήθηκε να αντιμετωπίσει.

Σε μία μελλοντική προοπτική για την παρούσα διατριβή, θα ήταν ενδιαφέρον να αναλυθεί η απόδοση της μεθόδου και σε άλλα διαθέσιμα επίπεδα του παιχνιδιού, με στόχο να μελετηθεί η ικανότητα γενίκευσης του πράκτορα *AngryBER* πιο συστηματικά και έχοντας στη διάθεσή μας περισσότερα πειραματικά αποτελέσματα. Επιπλέον, διακρίνοντας την ικανότητα της *δενδρικής* δομής να μοντελοποιεί και να εκμεταλλεύεται στο έπακρο τις ιδιότητες της σκηνής του παιχνιδιού, μία ακόμη ενδιαφέρουσα μελλοντική μελέτη θα μπορούσε να είναι ο εμπλουτισμός των χαρακτηριστικών των αντικειμένων της σκηνής με άλλες *τοπολογικές μετρικές*, οι οποίες μπορούν εύκολα να εξαχθούν από την προτεινόμενη *δενδρική* δομή αναπαράστασης.

Τέλος, άλλη μία μελλοντική επέκταση σχετίζεται με την επιλογή του αριθμού των συναρτήσεων βάσης, με άλλα λόγια την επιλογή της τάξης του μοντέλου. Η *Bayesian Sparse Παλινδρόμηση* προσφέρει μία λύση στο πρόβλημα αυτό με την εισαγωγή *sparse priors* για τις παραμέτρους του μοντέλου [12], [11], [3]. Κατά τη διάρκεια της εκπαίδευσης, οι συντελεστές που δεν θεωρούνται σημαντικοί εξαλείφονται και τελικά μόνο λίγοι από αυτούς διατηρούνται από το μοντέλο, εκείνοι που θεωρούνται σημαντικοί για το σύνολο εκπαίδευσης. Η παραπάνω διαπίστωση αποτελεί μία σημαντική κατεύθυνση για μελλοντική μελέτη και υλοποίηση, η οποία ενδεχομένως θα βελτιώσει τις δυνατότητες και την απόδοση της προτεινόμενης μεθόδου.

ΑΝΑΦΟΡΕΣ

- [1] P. Auer and N. Cesa-Bianchi, "Finite-time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, pp. 235-256, 2002.
- [2] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [3] K. Blekas and A. Likas, "Sparse regression mixture modeling with the multi-kernel relevance," *Knowledge and Information Systems (KAIS)*, pp. 241-264, 2014.
- [4] L. A. Ferreira, G. A. W. Lopes and P. E. Santos, "Combining Qualitative Spatial Representation Utility Function and Decision Making Under Uncertainty on the Angry Birds Domain," in *International Joint Conference on Artificial Intelligence*, 2013.
- [5] S. Lin, Q. Zhang and H. Zhang, "Object Representation in Angry Birds Game," in *International Joint Conference on Artificial Intelligence*, 2013.
- [6] B. MacNamee, "Agent Based Modeling in Computer Graphics and Games," in *Encyclopedia of Complexity and Systems Science*, Springer, 2009, pp. 1335-1352.
- [7] J. Mendes-Moreira, C. Soares, A. M. Jorge and J. Freire de Sousa, "Ensemble Approaches for Regression: A Survey," *ACM Computing Surveys*, pp. 1-10, 2012.

- [8] A. Narayan-Chen, L. Xu and J. Shavlik, "An empirical evaluation of machine learning approaches for angry birds," in *International Joint Conference on Artificial Intelligence*, 2013.
- [9] M. Polceanu and C. Buche, "Towards A Theory-Of-Mind-Inspired Generic Decision-Making Framework," in *International Joint Conference on Artificial Intelligence*, 2013.
- [10] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2009.
- [11] M. Seeger, "Bayesian and Optimal Design for the Sparse Linear Model," *Journal of Machine Learning Research*, pp. 759-813, 2008.
- [12] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, pp. 211-244, 2001.
- [13] N. Tziortziotis, K. Tziortziotis and K. Blekas, "Play Ms. Pac-Man using an advanced reinforcement learning agent," in *8th Hellenic Conference on AI - SETN*, 2014.
- [14] N. Tziortziotis , Papagiannis Georgios and K. Blekas, "A Bayesian Ensemble Regression Framework on the Angry Birds Game," in *ECAI Symposium on Artificial Intelligence on Angry Birds*, 2014.
- [15] P. Walega, T. Lechowski and M. Zawidzki, "Qualitative Physics in Angry Birds: first results," in *ECAI Symposium on Artificial Intelligence in Angry Birds*, 2014.
- [16] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *Knowledge Engineering Review*, pp. 115-152, 1995.

- [17] M. Wooldridge , An Introduction to MultiAgent Systems - Second Edition, John Wiley & Sons, 2009.
- [18] Xiao Yu Ge, S. Gould , J. Renz, S. Abenyasinghe, P. Zhang , J. Keys and A. Wang, "Angry birds game playing software, version 1.31," in *Technical Report*, Research School of Computer Science, The Australian National University, 2014.
- [19] P. Zhang and J. Renz, "Qualitative Spatial Representation and Reasoning in Angry Birds: The Extended Rectangle Algebra," in *Proceedings of Knowledge Representation and Reasoning*, p. to appear, 2014.

ΠΑΡΑΡΤΗΜΑ

Στο παράρτημα της παρούσας διατριβής δίνονται σε πίνακες τα συγκεντρωτικά αποτελέσματα των Angry Birds AI Competitions 2013-2014.

Πίνακας Π.1: Benchmarks Angry Birds AI Competition 2013-2014 – Επίπεδα 1-21 του “Poached Eggs”

<i>Rank</i>	<i>Team</i>	<i>Total</i>	<i>1-1</i>	<i>1-2</i>	<i>1-3</i>	<i>1-4</i>	<i>1-5</i>	<i>1-6</i>	<i>1-7</i>	<i>1-8</i>	<i>1-9</i>	<i>1-10</i>	<i>1-11</i>	<i>1-12</i>	<i>1-13</i>	<i>1-14</i>	<i>1-15</i>	<i>1-16</i>	<i>1-17</i>	<i>1-18</i>	<i>1-19</i>	<i>1-20</i>	<i>1-21</i>
1	PlanA+	1002380	30480	62370	40620	29000	69440	36970	32020	47320	26440	56830	47240	58210	34010	65640	54910	57530	51190	52120	39440	45980	64620
2	DataLab Birds	981120	31620	52000	41890	19790	70320	15700	45720	43190	50420	56790	50650	53420	32010	55640	46450	57380	48570	45730	35470	54680	73680
3	AngryHex (2013)	974670	30390	54160	41890	28000	64440	24990	36900	24860	49570	50570	53510	57750	42010	58190	46550	63430	46820	50020	38460	46970	65190
4	WISC	963160	29030	52180	41950	27850	65810	24640	25520	55810	33680	39170	50000	54980	32450	65060	49200	63430	54750	44740	37370	47980	67560
5	AngryHex (2014)	960320	32660	52580	41910	19690	68090	25180	24680	43330	42740	54890	53570	54860	41200	57150	41100	61470	50260	48050	36780	39010	71120
6	Angry Concepts	954030	29410	52250	41320	28160	64160	15660	24630	47170	48670	48290	56130	58600	50360	58050	43340	55000	45990	44030	37880	43730	61200
7	Beau Rivage	952390	29760	43160	40500	28680	61000	33540	45780	49150	33110	42200	57420	55600	35190	62770	47270	55890	49880	38820	29780	37020	75870
8	HungryBirds	951440	31210	53860	42040	20980	65700	28430	40580	27020	50710	53070	56260	55410	30130	59780	41040	55310	43530	46770	32690	56050	60870
9	AngryBER	935330	28510	43420	41910	19390	62880	35610	31980	45050	48670	51980	51550	54720	42510	45640	47090	50000	47740	44110	36450	35990	70130
10	RMIT RedBacks	933120	32220	53860	30480	19380	70350	33470	45740	29180	33320	59190	45250	56910	42300	45640	41540	66570	39370	48530	30460	40760	68600
11	Luabab	894840	29600	52180	40760	28030	66100	16860	32180	48330	41840	54110	45580	53040	41110	65640	28910	65670	39550	37680	16820	36970	53880
12	Dan	893370	29210	42760	41610	27990	63840	25700	45990	23390	45930	49570	38570	54990	32270	57550	47280	63000	42770	48290	22040	36910	53710
13	IHSEV (2014)	891590	29520	53070	41440	21860	56470	33470	29920	36630	40620	32300	31570	61070	41330	59250	34830	54160	47830	42840	40100	40470	62840
14	Impact Vector	886990	29710	60480	42040	19700	65440	33870	36880	25720	34770	46870	57490	55230	49620	0	45860	58870	47070	48210	25040	37650	66470
15	Black Forest Cuckoos	874650	29880	52390	41600	28980	65380	0	32380	56310	23050	52360	54360	57750	29770	57560	55300	59940	52920	41410	36940	46370	0
16	IHSEV (2013)	861830	29360	52780	41240	36810	51860	33790	30130	57570	24310	29860	59070	40960	42840	65640	30700	51730	47940	47320	30940	0	56980
17	Naïve Agent 2013	858730	29510	52230	40620	20680	55160	16070	21590	25730	35490	32600	46760	54070	49470	50590	46430	55210	48140	49430	37920	36790	54240

18	Sniper	785520	29760	42720	40500	18620	65850	35490	28780	48170	26960	35540	51150	49190	26180	65640	0	56530	37980	40460	30390	55610	0
19	Wanderer	774190	30840	60400	41900	36770	63550	26910	22070	57780	51480	68740	30620	42320	20050	58200	25520	38960	37560	35220	25300	0	0
20	Naive Agent (2014)	756680	29760	43250	40180	10590	62490	14980	22150	35840	36050	52570	39310	48660	30000	45640	44190	52300	39530	39590	29460	40140	0
21	s-birds	754190	30080	26240	42240	19050	38160	36180	49120	38340	41670	54940	57950	47460	22550	63560	42990	65750	38790	39120	0	0	0
22	S-birds Avengers	706110	28520	51770	35380	27950	60920	26000	0	48010	49050	0	48320	45150	22130	53760	41240	0	45000	45750	38200	38960	0
23	Lambdaers	705010	30970	52390	42070	19580	63750	26400	45880	57600	49370	61820	44120	38390	29350	54340	44040	0	44940	0	0	0	0
24	A.Wang	680640	22400	43320	31910	28080	48440	24770	23960	27580	42590	57610	47410	46980	27490	65640	36160	66550	39750	0	0	0	0
25	Akbaba	641080	28640	35540	41910	19120	47470	0	23880	45380	36960	56000	48460	52310	0	59420	0	47230	44260	54500	0	0	0
26	AngryDragons	620390	28120	33440	30430	28820	65650	0	0	29340	25590	0	39580	51340	30840	55640	33920	53800	45660	38390	29830	0	0
27	ObjectS	569000	29670	43200	40620	11720	65850	15570	21310	37170	22610	51250	57930	45200	26570	65640	34690	0	0	0	0	0	0
28	AngryPKU	476100	18420	33280	24070	10120	35650	26450	0	27840	24660	30160	31740	45870	0	45640	32790	55890	0	0	33520	0	0
29	JulyPlayer	352340	29360	52520	35360	20690	46300	17400	21810	45370	48670	34860	0	0	0	0	0	0	0	0	0	0	0
30	FEI2	211870	28360	34160	30470	27650	64710	26520	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	High Scores	1146160	32660	62370	42240	36810	70350	36970	49120	57780	51480	68740	59070	61070	50360	65640	55300	66570	54750	54500	38460	56050	75870
	2013 Highscores	1134920	31210	60400	42240	36810	65850	36180	49120	57780	51480	68740	59070	58600	50360	65640	55300	66550	54750	54500	38460	56050	75870
	3 stars	1042000	32000	60000	41000	28000	64000	35000	45000	50000	50000	55000	54000	45000	47000	70000	41000	64000	53000	48000	35000	50000	75000

Πίνακας Π.2: Benchmarks Angry Birds AI Competition 2013-2014 - Επίπεδα 22-42 του “Poached Eggs”

<i>Rank</i>	<i>Team</i>	<i>Total</i>	<i>2-1</i>	<i>2-2</i>	<i>2-3</i>	<i>2-4</i>	<i>2-5</i>	<i>2-6</i>	<i>2-7</i>	<i>2-8</i>	<i>2-9</i>	<i>2-10</i>	<i>2-11</i>	<i>2-12</i>	<i>2-13</i>	<i>2-14</i>	<i>2-15</i>	<i>2-16</i>	<i>2-17</i>	<i>2-18</i>	<i>2-19</i>	<i>2-20</i>	<i>2-21</i>
1	DataLab Birds	876090	64050	53130	108510	52620	78810	46040	45690	45090	23390	40620	0	31570	0	0	31120	56140	29440	55800	45580	0	68490
2	PlanA+	703650	0	96180	99490	0	73460	51280	53450	49200	20640	43600	53470	46720	70930	45230	0	0	0	0	0	0	0
3	AngryBER	682530	48710	0	100570	53240	74940	57350	44700	0	21910	40720	90540	0	77000	44620	28230	0	0	0	0	0	0
4	AngryHex (2014)	541690	47530	0	98940	48580	76530	0	0	47170	25060	36600	87580	0	73700	0	0	0	0	0	0	0	0
5	Naive Agent (2014)	503580	59480	0	91300	0	0	58270	51040	50300	46240	42330	0	32650	71970	0	0	0	0	0	0	0	0
6	Angry Dragons	465740	59990	0	0	0	69900	35820	49210	57300	30470	23800	0	41140	65530	32580	0	0	0	0	0	0	0
7	IHSEV (2014)	440110	63350	45790	96480	0	0	56750	40880	50490	18680	26490	0	0	0	41200	0	0	0	0	0	0	0
8	RMIT RedBacks	405000	0	0	0	0	64980	0	49460	47510	23680	34470	0	0	67620	37970	0	0	0	41540	37770	0	0
9	Impact Vactor	327340	46110	0	0	49190	0	37240	48020	0	0	0	0	35880	67230	43670	0	0	0	0	0	0	0
10	S-birds Avengers	181310	0	0	96270	56550	0	0	0	0	0	0	0	0	0	28490	0	0	0	0	0	0	0
	High Scores	1209020	64050	96180	108510	56550	78810	58270	53450	57300	46240	43600	90540	46720	77000	45230	31120	56140	29440	55800	45580	0	68490
	3-stars score	1216000	60000	60000	102000	50000	80000	62000	50000	53000	28000	40000	69000	60000	70000	50000	50000	62000	36000	60000	47000	52000	75000

Πίνακας Π.3: Αθροιστικά Αποτελέσματα των Πινάκων Π.1 και Π.2

<i>Rank</i>	<i>Team</i>	<i>Total Score</i>
1	DataLab Birds	1857210
2	PlanA+	1706030
3	AngryBER	1617860
4	AngryHex (2014)	1502010
5	RMIT RedBacks	1338120
6	IHSEV (2014)	1331700
7	Naïve Agent (2014)	1260260
8	Impact Vactor	1214330
9	Angry Dragons	1086130
10	S-birds Avengers	887420

ΔΗΜΟΣΙΕΥΣΕΙΣ ΣΥΓΓΡΑΦΕΑ

- [1] N. Tziortziotis , Papagiannis Georgios and K. Blekas, "A Bayesian Ensemble Regression Framework on the Angry Birds Game," in *ECAI Symposium on Artificial Intelligence on Angry Birds*, 2014.

ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ

Ο Γεώργιος Παπαγιάννης γεννήθηκε στη Λάρισα το 1989. Αποφοίτησε το 2007 από το Γενικό Λύκειο Καλαμπάκας και εισήχθη στο Τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων από το οποίο αποφοίτησε τον Φεβρουάριο του 2012. Οι σπουδές του συνεχίστηκαν σε μεταπτυχιακό επίπεδο στο ίδιο τμήμα, από το Μάρτιο του 2012 έως το Σεπτέμβριο του 2014, όταν και αποφοίτησε αποκτώντας ειδίκευση στις «Τεχνολογίες - Εφαρμογές».