



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΤΟΜΕΑΣ ΜΕΛΕΤΗΣ ΠΛΟΙΟΥ & ΘΑΛΑΣΣΙΩΝ ΜΕΤΑΦΟΡΩΝ

Παραμετροποίηση και Υλοποίηση Καθοδηγητικών  
Συμπεριφορών στη Μοντελοποίηση Πλήθους με τη Χρήση  
Πυρήνα Εικονικής Πραγματικότητας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Δημητρίου Π. Φουντουκίδη

Επιβλέπων: Παναγιώτης Δ. Κακλής  
Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος 2008

(Η σελίδα αυτή έμεινε κενή)



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΤΟΜΕΑΣ ΜΕΛΕΤΗΣ ΠΛΟΙΟΥ & ΘΑΛΑΣΣΙΩΝ ΜΕΤΑΦΟΡΩΝ

Παραμετροποίηση και Υλοποίηση Καθοδηγητικών  
Συμπεριφορών στη Μοντελοποίηση Πλήθους με τη Χρήση  
Πυρήνα Εικονικής Πραγματικότητας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Δημητρίου Π. Φουντουκίδη

Επιβλέπων: Παναγιώτης Δ. Κακλής  
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την \_\_\_/\_\_\_/2008.

---

Α. Παπανικολάου  
Καθηγητής ΕΜΠ

Π. Κακλής  
Καθηγητής ΕΜΠ

Α. Γκίνης  
Λέκτορας ΕΜΠ

Αθήνα, Σεπτέμβριος 2008

.....  
(Δημήτριος, Π. Φουντουκίδης)  
(Διπλωματούχος Ναυπηγός Μηχανολόγος Μηχανικός)

© (2008) Εθνικό Μετσόβιο Πολυτεχνείο. All rights reserved.

# Περιεχόμενα

Ευχαριστίες	xiii
Πρόλογος	xv
Περίληψη	xvii
<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Η Εικονική Πραγματικότητα	1
1.1.1 Ορισμός	2
1.1.2 Σημαντικοί Παράγοντες των Συστημάτων Ε.Π.	3
1.1.3 Είδη Συστημάτων Ε.Π.	5
1.1.4 Γενικές Εφαρμογές ΕΠ	7
1.1.5 Εφαρμογές ΕΠ στη Ναυπηγική	14
1.2 Πεδίο Έρευνας	15
1.2.1 Μοντελοποίηση πλήθους	15
1.2.2 Αυτόνομοι Χαρακτήρες (Autonomous Characters)	16
1.2.2.1 Εισαγωγή στους Αυτόνομους Πράκτορες	16
1.2.2.2 Ταξινόμηση Πρακτόρων	17
1.2.2.3 Διάφοροι Ορισμοί Πρακτόρων	18
1.2.2.4 Χαρακτηριστικά Πρακτόρων	20
1.2.2.5 Ορισμός των Αυτόνομων Χαρακτήρων	21
1.2.3 Καθοδηγητικές συμπεριφορές	22
<b>2 Λογισμική Πλατφόρμα Εικονικής Πραγματικότητας</b>	<b>25</b>
2.1 Πυρήνας Εικονικής Πραγματικότητας (VRKernel)	25

2.2	Μοντελοποίηση 2Δ/3Δ Γεωμετρικών Σχημάτων .....	27
2.2.1	Open Inventor .....	27
2.2.1.1	Αντικείμενα .....	29
2.2.1.2	Open Inventor και OpenGL .....	30
2.3	Υλοποίηση νόμων Φυσικής .....	30
2.3.1	Ανίχνευση Σύγκρουσης ή Παρεμβολής .....	30
2.4	Μοντελοποίηση πλήθους .....	31
2.4.1	Πράκτορες .....	33
2.4.2	Εμπόδια .....	34
2.4.3	Καθοδηγητικές Συμπεριφορές .....	34
2.4.4	Αυτόνομη Νοημοσύνη Πρακτόρων .....	37
2.5	Προσομοιώσεις VRKernel .....	38
2.5.1	Προσομοίωση Κίνησης Πλήθους .....	39
2.5.2	Εισαγωγή Στοιχείων Προσομοίωσης .....	41
2.6	Στόχοι της Εργασίας .....	42
2.6.1	Σημεία Έμφασης .....	42
2.6.2	Σημεία Πρωτοτυπίας .....	43
<b>3</b>	<b>Αλγόριθμοι Εύρεσης Διαδρομών</b> .....	<b>45</b>
3.1	Εισαγωγή - Ορισμός .....	45
3.2	Είδη Αλγορίθμων .....	46
3.2.1	Μέθοδος Παρεμβολής Γραμμής .....	47
3.2.2	Μέθοδοι Γραφήματος Βαρών .....	48
3.2.2.1	Μέθοδος Dijkstra και Best First Search .....	49
3.2.2.2	Μέθοδος Εύρεσης A* .....	51
3.2.3	Άλλες Μέθοδοι .....	53
<b>4</b>	<b>Ανάπτυξη Καθοδηγητικών Συμπεριφορών</b> .....	<b>55</b>
4.1	Αποφυγή .....	56
4.1.1	Περιγραφή .....	56
4.1.2	Ψευδοκώδικας .....	56

4.1.3	Υλοποίηση .....	58
4.2	Καταδίωξη .....	62
4.2.1	Περιγραφή .....	62
4.2.2	Ψευδοκώδικας .....	63
4.2.3	Υλοποίηση .....	65
4.3	Διαφυγή .....	69
4.3.1	Περιγραφή .....	69
4.3.2	Ψευδοκώδικας .....	69
4.3.3	Υλοποίηση .....	71
4.4	Αποφυγή Εμποδίων .....	72
4.4.1	Περιγραφή .....	72
4.4.2	Ψευδοκώδικας .....	80
4.4.3	Υλοποίηση .....	91
4.5	Ακολουθία Διαδρόμου .....	91
4.5.1	Περιγραφή .....	91
4.5.2	Ψευδοκώδικας .....	95
4.5.3	Υλοποίηση .....	99
4.6	Ευθυγράμμιση .....	100
4.6.1	Περιγραφή .....	100
4.6.2	Ψευδοκώδικας .....	101
4.6.3	Υλοποίηση .....	102
4.7	Άφιξη .....	106
4.7.1	Περιγραφή .....	106
4.7.2	Ψευδοκώδικας .....	106
4.7.3	Υλοποίηση .....	112
<b>5</b>	<b>Συμπεράσματα</b> .....	<b>113</b>
5.1	Αποφυγή .....	113
5.2	Καταδίωξη & Διαφυγή .....	114
5.3	Αποφυγή Εμποδίων .....	115
5.4	Ακολουθία Διαδρόμου .....	118

5.5	Ευθυγράμμιση .....	118
5.6	Άφιξη .....	120
<b>6</b>	<b>Προτάσεις</b>	<b>121</b>



# Κατάλογος Σχημάτων

1.1	Χρήση Προσομοιωτή Αλεξιπτώτου από Πιλότο .....	4
1.2	Μία Ταξινόμηση των Πρακτόρων.....	17
1.3	Ο βασικός πράκτορας κατά Russel & Norvig.....	18
1.4	Ο βασικός πράκτορας κατά Maes.....	19
1.5	Ο βασικός πράκτορας κατά Hayes-Roth.....	19
1.6	Ο βασικός πράκτορας κατά Coen.....	20
1.7	Διαστρωμάτωση της Κίνησης.....	22
2.1	Η Αρχιτεκτονική του Open Inventor.....	28
2.2	Ιεραρχία Κλάσεων του VRkernel στα Αντικείμενα Σκηνης, Μοντελοποίηση Νου, και Καθοδηγητικών Συμπεριφορών.....	33
3.1	Εύρεση Βέλτιστης Διαδρομής vs Ελεύθερης Κίνησης .....	46
3.2	Εύρεση Βέλτιστης Διαδρομής vs Κυρτής Επιφάνειας .....	47
3.3	Εύρεση Βέλτιστης Διαδρομής με τον αλγόριθμο Dijkstra.....	49
3.4	Εύρεση Βέλτιστης Διαδρομής με τον αλγόριθμο Best First Search .....	50
3.5	Εύρεση Βέλτιστης Διαδρομής με τον αλγόριθμο A*.....	51
4.1	Αναζήτηση & Αποφυγή .....	58
4.2	Εισαγωγή Στοιχείων Συμπεριφοράς "Αποφυγή" .....	59
4.3	Καθοδηγητική Συμπεριφορά "Αποφυγή" - Έναρξη.....	60
4.4	Καθοδηγητική Συμπεριφορά "Αποφυγή" - Προσέγγιση.....	60
4.5	Καθοδηγητική Συμπεριφορά "Αποφυγή" - Ενεργοποίηση.....	61
4.6	Καθοδηγητική Συμπεριφορά "Αποφυγή" - Ελάχιστη Προσέγγιση .....	61
4.7	Καθοδηγητική Συμπεριφορά "Αποφυγή" - Απομάκρυνση .....	62

4.8	Εισαγωγή Στοιχείων Συμπεριφοράς "Καταδίωξη" .....	65
4.9	Καταδίωξη & Διαφυγή .....	66
4.10	Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Έναρξη .....	67
4.11	Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Τυχαία Κίνηση .....	67
4.12	Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Ενεργοποίηση .....	68
4.13	Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Ενδιάμεση Προσέγγιση.....	68
4.14	Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Τέλος Χειρισμού.....	69
4.15	Εισαγωγή Στοιχείων Συμπεριφοράς "Διαφυγή" .....	71
4.16	Καθοδηγητική Συμπεριφορά "Διαφυγή" - Έναρξη .....	73
4.17	Καθοδηγητική Συμπεριφορά "Διαφυγή" - Τυχαία Κίνηση .....	73
4.18	Καθοδηγητική Συμπεριφορά "Διαφυγή" - Ενεργοποίηση .....	74
4.19	Καθοδηγητική Συμπεριφορά "Διαφυγή" - Ενδιάμεση Αποφυγή .....	74
4.20	Καθοδηγητική Συμπεριφορά "Διαφυγή" - Τέλος Χειρισμού .....	75
4.21	Αποφυγή Εμποδίων .....	76
4.22	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Έναρξη & Τυχαία Κίνηση .....	81
4.23	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενεργοποίηση.....	82
4.24	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Αρχικός Χειρισμός .....	83
4.25	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός I.....	84
4.26	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός II.....	85
4.27	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός III.....	86
4.28	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός IV.....	87
4.29	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός V.....	88
4.30	Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Τέλος Χειρισμού .....	89
4.31	Εισαγωγή Στοιχείων Συμπεριφοράς "Αποφυγή Εμποδίου" .....	90
4.32	Εύρεση Βέλτιστης Διαδρομής με αλγόριθμο A* και μεγέθυνσης εμποδίου .....	94
4.33	Εισαγωγή Στοιχείων Συμπεριφοράς "Ακολουθία Διαδρόμου".....	96
4.34	Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Έναρξη .....	96
4.35	Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Παράκαμψη 1ου Εμποδίου ...	97
4.36	Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Αρχή Διαγώνιας Κίνηση .....	97
4.37	Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Τέλος Διαγώνιας Κίνησης ..	98

4.38	Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Παράκαμψη 2ου Εμποδίου ...	98
4.39	Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Τέλος Διαδρομής .....	99
4.40	Έννοια της Γειτονίας .....	100
4.41	Ευθυγράμμιση .....	101
4.42	Εισαγωγή Στοιχείων Συμπεριφοράς "Ευθυγράμμιση" .....	102
4.43	Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Έναρξη Κίνησης .....	103
4.44	Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Ενεργοποίηση .....	103
4.45	Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Έναρξη Στροφής Ευθυγράμμισης ...	104
4.46	Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Ενδιάμεσο Σημείο Στροφής I .....	104
4.47	Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Ενδιάμεσο Σημείο Στροφής II .....	105
4.48	Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Τέλος Χειρισμού .....	105
4.49	Άφιξη .....	107
4.50	Εισαγωγή Στοιχείων Συμπεριφοράς "Άφιξη" .....	108
4.51	Καθοδηγητική Συμπεριφορά "Άφιξη" - Έναρξη .....	109
4.52	Καθοδηγητική Συμπεριφορά "Άφιξη" - Ενεργοποίηση .....	109
4.53	Καθοδηγητική Συμπεριφορά "Άφιξη" - Έναρξη Χειρισμού .....	110
4.54	Καθοδηγητική Συμπεριφορά "Άφιξη" - Ενδιάμεσος Χειρισμός I .....	110
4.55	Καθοδηγητική Συμπεριφορά "Άφιξη" - Ενδιάμεσος Χειρισμός II .....	111
4.56	Καθοδηγητική Συμπεριφορά "Άφιξη" - Τέλος Χειρισμού .....	111



# Ευχαριστίες

Η διπλωματική μου εργασία σηματοδοτεί το τέλος μίας μεγάλης προσπάθειας για την ολοκλήρωση ενός παιδικού μου ονείρου, την ονομασία μου ως Ναυπηγό Μηχανολόγο Μηχανικό του Ε.Μ.Π. Έχοντας ήδη αποφοιτήσει από τη Ναυτική των Δοκίμων Σχολή και όντας παντρεμένος κατά το διάστημα των σπουδών μου, βρήκα αμέριστη συμπαράσταση και κατανόηση στη δύσκολή πορεία μου, όλο αυτό το διάστημα, στο πρόσωπο της γυναίκας μου Βαλεντίνας την οποία και ευχαριστώ πρώτα και κύρια και της αφιερώνω την εργασία μου αυτή. Επίσης στο υπέροχο πλάσμα που μου χάρισε και η κατανόηση της οποίας εκφραζόταν με το αθώο "Ησυχία, ο μπαμπάς διαβάζει για το σχολείο του", τη Ματίνα.

Ιδιαίτερα θα πρέπει να αναφερθώ στον Καθηγητή του Τμήματος Ναυπηγών Μηχανολόγων Μηχανικών του Ε.Μ.Π. κ. Παναγιώτη Κακλή, στον οποίο τρέφω ιδιαίτερη εκτίμηση τόσο για τον ανθρώπινη όσο και για την επιστημονική του πλευρά, επειδή στάθηκε πολύτιμος αρωγός και με στήριξε στη πορεία μου.

Ευχαριστώ ολόθερμα το Διδάκτορα του Τμήματος Ναυπηγών Μηχανολόγων Μηχανικών του Ε.Μ.Π. κ. Κωνσταντίνο Κώστα για τη σημαντική και ουσιαστική βοήθεια που μου προσέφερε με τις εποικοδομητικές του παρατηρήσεις και υποδείξεις, ξοδεύοντας πολύ από τον προσωπικό του χρόνο. Η παρούσα διπλωματική εργασία δεν είναι παρά ένα λιθαράκι στο πρωτότυπο και ανεκτίμητο έργο του, τον Πυρήνα Εικονικής Πραγματικότητας για την ανάπτυξη ναυπηγικών εφαρμογών.

Θα πρέπει να ευχαριστήσω θερμά το Διδακτικό Προσωπικό της Σχολής για τη βοήθειά και την κατανόησή τους. Ορισμένοι εξάντλησαν την επιείκειά τους και πιστεύω όχι άδικα, βοηθώντας με τα μέγιστα όλο αυτό το διάστημα.

Επίσης ευχαριστώ όλους τους φίλους και συμφοιτητές μου που με βοήθησαν ο καθένας

με το δικό του τρόπο, στην ολοκλήρωση της δύσκολης πορείας μου.

*Δημήτριος Π. Φουντουκίδης*

*Αθήνα, Σεπτέμβριος 2008*

# Πρόλογος

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η υλοποίηση συγκεκριμένων καθοδηγητικών συμπεριφορών στη μοντελοποίηση πλήθους με χρήση πυρήνα εικονικής πραγματικότητας. Ο πυρήνας εικονικής πραγματικότητας αναπτύχθηκε στα πλαίσια της διδακτορικής διατριβής του κ. Κωνσταντίνου Κώστα, Σχολή Ναυπηγών Μηχανολόγων Μηχανικών, Ε.Μ.Π. και είναι το βασικό συστατικό μίας πλατφόρμας Εικονικής Πραγματικότητας με δυνατότητες δικτυακής εξυπηρέτησης. Ουσιαστικά πρόκειται για μία βιβλιοθήκη λογισμικού, αντικειμενοστραφής και επεκτάσιμη. Αποτελείται από ένα σύνολο υποσυστημάτων που επιτρέπουν τη μεμονωμένη ή και συνδυασμένη χρήση τους. Ένα από τα κύρια συστατικά του πυρήνα αυτού είναι και η μοντελοποίηση πλήθους, δηλαδή αυτόνομοι χαρακτήρες τεχνητής νοημοσύνης με δυνατότητες δράσης συγκεκριμένων χαρακτηριστικών (καθοδηγητικών συμπεριφορές) και μοντελοποίηση φυσικών νόμων, όπως είναι η πρόληψη και η αποφυγή σύγκρουσης και η εξεύρεση διαδρόμων βέλτιστης κυκλοφορίας.

Στο πρώτο κεφάλαιο επιχειρείται μία εισαγωγή στην έννοια της Εικονικής Πραγματικότητας, τα συστήματα και οι παράγοντες που επιδρούν για την όσο το δυνατόν ρεαλιστική της απεικόνιση καθώς και τις εφαρμογές της, γενικές και εξειδικευμένες στη Ναυπηγική Τεχνολογία. Επίσης παρατίθενται βασικές γνώσεις τεχνητής νοημοσύνης και ειδικότερα για τη δράση αυτόνομων χαρακτήρων με βάση καθοδηγητικές συμπεριφορές που μπορούν να μοντελοποιήσουν τη συμπεριφορά ενός πλήθους ανθρώπων.

Κατόπιν στο δεύτερο κεφάλαιο θα γίνει μία σύντομη ανάλυση του πυρήνα Εικονικής Πραγματικότητας και των κύριων συστατικών και υποσυστημάτων του. Επίσης θα παρουσιασθούν τα τμήματα του πυρήνα που θα επηρεασθούν από την υλοποίηση συγκεκριμένων καθοδηγητικών συμπεριφορών καθώς και τα σημεία της μοντελοποίησης των φυσικών νόμων

που θα βελτιωθούν περαιτέρω.

Στο τρίτο κεφάλαιο γίνεται καταρχήν μία παρουσίαση της αναγκαιότητας του αλγόριθμου σχεδίασης και εύρεσης διαδρόμων βέλτιστης κυκλοφορίας. Επίσης παρουσιάζονται ορισμένες από τις πιθανές υλοποιήσεις και τελικά γίνεται μία παραμετροποίηση των αποτελεσμάτων με στόχο την επιλογή του βέλτιστου δυνατού αλγόριθμου.

Το τέταρτο κεφάλαιο είναι και το πιο ουσιαστικό της εργασίας αυτής. Περιγράφει την ανάπτυξη των Καθοδηγητικών Συμπεριφορών και συγκεκριμένα των Flee, Pursuit, Evasion, Obstacle Avoidance, Alignment, Arrival και Path Following. Ειδικότερα και για κάθε Συμπεριφορά παρατίθεται μία μικρή περιγραφή, κατόπιν αναλύεται ο ψευδοκώδικας εφαρμογής και τέλος η λογισμική υλοποίησή της και τα οπτικά αποτελέσματα ενός χαρακτηριστικού σεναρίου.

Στο πέμπτο κεφάλαιο παρατίθενται τα βασικά συμπεράσματα που προέκυψαν κατά την υλοποίηση και παραμετροποίηση των συμπεριφορών ενώ το τελευταίο κεφάλαιο, το έκτο, περιέχει προτάσεις για τη συνέχεια της έρευνας στο συγκεκριμένο ερευνητικό πεδίο.

*Δημήτριος Π. Φουντουκίδης*

*Αθήνα, Σεπτέμβριος 2008*



# Περίληψη

Η συμπεριφορά ενός πλήθους μπορεί με κάποιες παραδοχές να προσομοιωθεί με ένα μοντέλο Εικονικής Πραγματικότητας. Το μοντέλο αυτό διαθέτει ένα πυρήνα με δυνατότητες φυσικής κίνησης αλλά και συμπεριφοράς. Αν δεχθούμε ότι το πλήθος μπορεί να αναλυθεί σε κάποιες βασικές συμπεριφορές οι οποίες με τη σειρά τους μπορούν να συνδυάζονται τότε μπορούμε να προσομοιώνουμε με μεγάλη ακρίβεια μία πολυσύνθετη κατάσταση όπως αυτή της κίνησης μίας ομάδας ανθρώπων σε περιορισμένο περιβάλλον.

Η υλοποίηση των βασικών συμπεριφορών που ονομάζονται και καθοδηγητικές έγινε καταρχήν διαισθητικά και κατόπιν ακολούθησε μία παραμετροποίηση της κάθε υλοποίησης με στόχο τη βελτιστοποίηση της προσομοίωσης αλλά και του κάθε αλγόριθμου.

Τέλος, υλοποιήθηκε μία προσέγγιση εύρεσης βέλτιστων διαδρόμων και βελτιστοποίηση του αλγόριθμου αποφυγής συγκρούσεως.

(Η σελίδα αυτή έμεινε κενή)

# Κεφάλαιο 1

## Εισαγωγή

Οι έννοιες της Εικονικής Πραγματικότητας, της Μοντελοποίησης ενός Πλήθους και η Τεχνολογία των Συμπεριφορών θεωρούνται θεμελιώδης και απαραίτητες στη κατανόηση του κειμένου και για το λόγο αυτό κρίθηκε σκόπιμο να προηγηθεί ένα εισαγωγικό κεφάλαιο το οποίο και ουσιαστικά θα καθορίσει το εννοιολογικό πλαίσιο της εργασίας αυτής.

Ακολούθως θα αναφερθούμε στην έννοια της Εικονικής Πραγματικότητας, στους παράγοντες που επηρεάζουν την υλοποίηση ενός ρεαλιστικού εικονικού περιβάλλοντος, στα είδη συστημάτων και στις εφαρμογές της στη Ναυπηγική και άλλες επιστήμες. Επιπλέον θα αναφερθούμε στη μοντελοποίηση πλήθους ανθρώπων ή ζώων, στη τεχνολογία των αυτόνομων πρακτόρων και στην ανάπτυξη των καθοδηγητικών συμπεριφορών.

### 1.1 Η Εικονική Πραγματικότητα

Η Εικονική Πραγματικότητα είναι μια ευρύτατα χρησιμοποιούμενη έννοια, η οποία ανάλογα με τη χρήση που αποσκοπεί, ορίζεται με ποικίλους τρόπους. Κάποιος μπορεί να υποθέσει ότι αφού οι εφαρμογές της εκτείνονται σε ένα ευρύ φάσμα τότε δικαιολογείται και η ποικιλία των χρήσεών της. Ακόμα και έτσι όμως, υπάρχουν περιπτώσεις όπου η χρήση του όρου γίνεται σε τομείς και εφαρμογές που δεν είναι συναφείς με αυτόν.

### 1.1.1 Ορισμός

Η **Εικονική Πραγματικότητα, ΕΠ (Virtual Reality, VR)** είναι μία σχετικά νέα τεχνολογία και ο ορισμός της δεν έχει ακόμα οριστικοποιηθεί. Οι ερευνητές και οι χρήστες της ΕΠ βλέπουν την έννοια αυτή από τη δική τους οπτική γωνία ο καθένας. Ο ορισμός που θα δώσουμε στην εργασία αυτή είναι αυτός που χρησιμοποιείται περισσότερο από τη διεθνή πανεπιστημιακή κοινότητα, ο οποίος όμως δεν είναι και εκείνος που συνήθως χρησιμοποιείται από τα μέσα μαζικής επικοινωνίας και τους εμπόρους διαφόρων προϊόντων.

Το λεξικό της αγγλικής γλώσσας [22] ορίζει τη λέξη *Virtual* δηλαδή *Εικονικός* ως "το υπάρχειν κατ' ανάγκη ή αποτέλεσμα, αλλά όχι ως γεγονός"<sup>1</sup>. Ο όρος έχει ήδη χρησιμοποιηθεί και σε άλλους τομείς της επιστήμης των υπολογιστών, όπως για παράδειγμα στην περίπτωση που ένα υπολογιστικό σύστημα απαιτεί τη χρήση περισσότερης μνήμης RAM από αυτή που του διατίθεται από το υλικό (hardware) και μπορεί να δεσμεύσει χώρο στο σκληρό δίσκο οπότε και επεκτείνει *εικονικά* τη μνήμη του.

Η σημασία της έννοιας *Πραγματικότητα* είναι πιο πολυσύνθετη και τυχόν προσπάθεια να την ορίσουμε πλήρως μπορεί να οδηγήσει σε περίπλοκες φιλοσοφικές συζητήσεις. Θα αρκεστούμε στον ορισμό της αγγλικής λέξης *Reality* από το ίδιο λεξικό [22], ως "τη κατάσταση ή την ποιότητα του πραγματικού. Κάτι που υπάρχει ανεξάρτητα από ιδέες που το αφορούν. Κάτι που αποτελεί ένα πραγματικό ή υπαρκτό αντικείμενο και ξεχωρίζει από κάτι που είναι απλώς προφανές"<sup>2</sup>. Θα απλοποιήσουμε το ορισμό χωρίς βλάβη της γενικότητας και θα ορίσουμε *Πραγματικότητα* ως κάτι που υπάρχει και μπορούμε να το αισθανθούμε.

Ένας ορισμός της ΕΠ που είναι αρκετά στενός ώστε να αποκλείει αρκετές παραπλανητικές χρήσεις του όρου αλλά και αρκετά ευρύς ώστε να περιλαμβάνει αρκετές συσκευές και μέσα που χρησιμοποιούν οι ερευνητές του θέματος είναι ο εξής [22]:

**Εικονική Πραγματικότητα, ΕΠ (Virtual Reality, VR)** είναι ένα μέσο που απαρτίζεται από μία σειρά διαδραστικών προσομοιώσεων που υλοποιούνται

---

<sup>1</sup>"being in essence or effect, but not in fact"

<sup>2</sup>"the state or quality of being real. Something that exists independently of ideas concerning it. Something that constitutes a real or actual thing as distinguished from something that is merely apparent"

μέσω υπολογιστικών συστημάτων, αισθάνεται τη θέση και πράξεις του συμμετέχοντα και αντικαθιστά ή προσθέτει μία νέα ανάδραση σε μία ή περισσότερες αισθήσεις του, δίνοντας στο χρήστη την αίσθηση ή την αίσθηση ότι είναι εμβυθισμένος ή παρών στη υπό προσομοίωση πραγματικότητα (δηλ. σε ένα εικονικό κόσμο).

### 1.1.2 Σημαντικοί Παράγοντες των Συστημάτων Ε.Π.

Υπάρχουν πολλοί παράγοντες που μπορούν να συνεισφέρουν σε ένα ρεαλιστικό και όσο το δυνατόν πιστευτό εικονικό περιβάλλον. Αρκετοί από τους παράγοντες αυτούς παρατίθενται παρακάτω:

#### 1. Οπτικός ρεαλισμός (Visual Realism)

Το επίπεδο του ρεαλισμού στη σκηνή που περιγράφεται βοηθά στο να γίνει όσο το δυνατόν πιο πιστευτό το περιβάλλον. Συστήματα κινηματικών γραφικών (Animation Systems) επαγγελματικού επιπέδου και ανιχνευτές ακτίδων (Ray Tracers) σε γραφικό περιβάλλον παράγουν απίστευτα ρεαλιστικές εικόνες και μπορούν κάλλιστα να χρησιμοποιηθούν ως ειδικά εφέ στη βιομηχανία του κινηματογράφου.

#### 2. Ανάλυση Εικόνας (Image Resolution)

Η ανάλυση εικόνας είναι ένας άλλος παράγοντας που συνδέεται στενά με τον οπτικό ρεαλισμό. Οι εικόνες που δημιουργούνται με τον Η/Υ αποτελούνται από διακριτά στοιχεία, τα εικονοστοιχεία (pixels), το μέγεθος και ο αριθμός των οποίων εξαρτάται από το μέγεθος της προβολής και την ανάλυσή της. Σε υψηλές αναλύσεις, η διακριτή φύση της προβολής γίνεται λιγότερο προφανής αλλά ο αριθμός των εικονοστοιχείων γίνεται όλο και μεγαλύτερος έως τεράστιος για τις υπολογιστικές δυνατότητες ενός συστήματος.

#### 3. Ρυθμός Ανανέωσης Καρέ (Frame Rate)

Ο ρυθμός ανανέωσης των καρέ είναι ένα αποτέλεσμα της διακριτής φύσης των γραφικών και κινηματικών γραφικών που παράγονται με ένα υπολογιστικό σύστημα. Για να

δώσουμε την εντύπωση της κίνησης, το υπολογιστικό σύστημα ανανεώνει την προβολή με μία ακολουθία εικόνων και σε μία συχνότητα που θα μας δώσει την ψευδαίσθηση της συνέχειας. Μία ακολουθία εικόνων με ρυθμό ανανέωσης τουλάχιστον 25 καρέ ανά δευτερόλεπτο (25 frames per second - fps) θεωρείται ότι δίνει την εντύπωση μίας φυσιολογικής κίνησης.

#### 4. Καθυστέρηση Ανάδρασης (Latency)

Η καθυστέρηση ανάδρασης είναι πιθανώς ο πιο σημαντικός παράγοντας ενός συστήματος ΕΠ και πρέπει να δώσουμε αρκετή σημασία αν θέλουμε να δώσουμε ένα προφανή ρεαλισμό στη προσομοίωσή μας. Η καθυστέρηση της ανάδρασης εισάγεται από τα διάφορα υποσυστήματα ενός συστήματος ΕΠ και είναι ο χρόνος μεταξύ της δράσης του χρήστη και της αντίδρασης του συστήματος. Εμφανίζεται συνήθως με τη μορφή μιας απώλειας της συνέχειας στην προβολή των γραφικών.



*Σχήμα 1.1: Προσωπικό του Πολεμικού Ναυτικού των Η.Π.Α. καθώς χρησιμοποιεί έναν προσομοιωτή πτώσης με αλεξίπτωτο*

### 1.1.3 Είδη Συστημάτων Ε.Π.

Μία μεγάλη διαφοροποίηση των συστημάτων ΕΠ είναι σύμφωνα με το τρόπο που αυτά συνδέονται και αλληλεπιδρούν με το χρήστη. Το εδάφιο αυτό περιγράφει τους πιο συχνούς τρόπους διασύνδεσης που βρίσκουμε σε συστήματα ΕΠ.

#### 1. Συστήματα τ. Παράθυρο στο Κόσμο (Window on World).

Κάποια συστήματα χρησιμοποιούν μία κοινή οθόνη Η/Υ για να απεικονίσουν τον εικονικό κόσμο. Αυτό ορισμένες φορές καλείται και Επιφανειακή ΕΠ (Desktop VR) ή Παράθυρο στο Κόσμο (Window on World). Αυτή η ιδέα υφίσταται από τις πρώτες ημέρες σχεδόν των Υπολογιστικών Γραφικών (Computer Graphics). Το 1965, ο Ivan Sutherland ανέπτυξε ένα ερευνητικό πρόγραμμα για τα Υπολογιστικά Γραφικά και με μία δημοσίευση με τον τίτλο "The Ultimate Display", οδήγησε τις εξελίξεις τα χρόνια που ακολούθησαν. Ένα φημισμένο σημείο από τη δημοσίευση αυτή που επεξηγεί και τον όρο αυτό αναφέρει:

"Κάποιος πρέπει να βλέπει την οθόνη ενός υπολογιστή σαν ένα παράθυρο μέσω του οποίου βλέπει έναν εικονικό κόσμο. Η πρόκληση στα υπολογιστικά γραφικά είναι να κάνουν την εικόνα του παραθύρου αυτού να φαίνεται αληθινή, να ακούγεται αληθινή και τα αντικείμενα της να δρουν αληθινά."

#### 2. Αντιστοίχιση Κινούμενης Εικόνας (Video Mapping)

Μία παραλλαγή της προηγούμενης εκδοχής συγχωνεύει την είσοδο της ψηφιοποιημένης εκδοχής του χρήστη με την είσοδο υπολογιστικών γραφικών δύο διαστάσεων. Ο χρήστης παρακολουθεί στην οθόνη του υπολογιστή το δικό του εικονικό σώμα, που μπορεί να έχει διάφορα σχήματα, να αλληλεπιδρά με τον κόσμο. Ο Myron Krueger, ο οποίος είναι από τους πρωτοπόρους στη μορφή αυτή της ΕΠ από τα τέλη της δεκαετίας των '60 αναφέρεται διεξοδικά στα Artificial Reality [16] και Artificial Reality II [17].

#### 3. Συστήματα Εμβυθισμένης Πραγματικότητας (Immersive Systems)

Το απόλυτο σύστημα ΕΠ βυθίζει κυριολεκτικά τον χρήστη μέσα σε ένα εικονικό κόσμο. Αυτά τα "εμβυθισμένα" συστήματα ΕΠ διαθέτουν συχνά Head Mounted Display (HMD), ένα κράνος ή μία μάσκα προσώπου που έχει προσαρμοσμένα πάνω του/της μία μικρή οθόνη και ακουστικά. Το κράνος μπορεί να κινείται ελεύθερα ή να παίρνει κίνηση από τον υπολογιστή μέσω μηχανικών συνδέσεων.

Μία παραλλαγή των εμβυθιζόμενων συστημάτων χρησιμοποιεί πολλαπλές οθόνες προβολής για να δημιουργήσει ένα Cave<sup>3</sup>, ένα δωμάτιο δηλαδή όπου ο χρήστης μπορεί να σταθεί μέσα και να "ζήσει" την εμπειρία του εικονικού κόσμου στους περιβάλλοντες τοίχους.

#### 4. Τηλεπαρουσία (Telepresence)

Η Τηλεπαρουσία και η μεταφορά εικόνας θεωρούνται σαν τα πιο πολλά υποσχόμενα εργαλεία για την οπτικοποίηση και μεταφορά σύνθετων πληροφοριών. Η Τηλεπαρουσία ποικίλει από τη παραδοσιακή οπτικοποίηση ΕΠ που δημιουργήθηκε με υπολογιστικά γραφικά με τη προοπτική ο κόσμος που δημιουργείται να προσεγγίσει τον πραγματικό. Η τεχνολογία αυτή συνδέει απομακρυσμένους αισθητήρες του πραγματικού κόσμου με τις αισθήσεις του χρήστη. Οι απομακρυσμένοι αισθητήρες μπορεί να είναι τοποθετημένοι σε ένα ρομπότ ή και σε ένα όργανο που τηλεχειρίζεται από άνθρωπο.

Κάθε σύστημα τηλεπαρουσίας περιέχει απαραίτητα τρία υποσυστήματα: την υποδομή που συνδέει το χρήστη και το κανάλι επικοινωνίας, το κανάλι επικοινωνίας από μόνο του και την υποδομή που συνδέει το κανάλι επικοινωνίας με τον απομακρυσμένο χώρο ή τον απομακρυσμένο χρήστη.

#### 5. Ανάμικτη Πραγματικότητα (Mixed Reality)

Αν συγχωνεύσουμε τις τεχνολογίες της Τηλεπαρουσίας με αυτή των παραδοσιακών συστημάτων ΕΠ παράγουμε τα συστήματα Ανάμικτης Πραγματικότητας ή Ενιαίας Προσομοίωσης. Υπάρχει ένας πιο διάσημος όρος, εναλλακτικός της "Ανάμικτης" και είναι

---

<sup>3</sup>Το 1992, η Electronic Visualization Laboratory (EVL, Chicago II, USA) εφηύρε το θέατρο εικονικής πραγματικότητας CAVE.



αυτός της Επαυξημένης Πραγματικότητας (Augmented Reality). Στη τεχνολογία αυτή τα εξαγόμενα από ένα υπολογιστικό σύστημα αναμειγνύονται με στοιχεία από ένα σύστημα τηλεπαρουσίας και/ή την οπτική του χρήστη για τον πραγματικό κόσμο. Ένας χειριστής μαχητικού αεροσκάφους μπορεί και βλέπει χάρτες και στοιχεία που παρήχθησαν με τη χρήση υπολογιστή σε προβολή επί του ανεμοθώρακιου ή σε ειδικό προβολικό εξάρτημα του κράνους του ταυτόχρονα με την οπτική που έχει ο ίδιος για τον περιβάλλοντα φυσικό κόσμο.

#### 1.1.4 Γενικές Εφαρμογές ΕΠ

Ακολούθως θα παρουσιάσουμε τους κυρίως τομείς εφαρμογών των τεχνολογιών ΕΠ μαζί με μία μικρή περιγραφή παραδειγμάτων.

##### 1. Προσομοιώσεις Πτήσεων (Flight Simulation)

Η ανάπτυξη συστημάτων Εμβυθισμένης Προσομοίωσης συνεισέφερε τα μέγιστα στην έρευνα συστημάτων ΕΠ. Ένας τομέας όπου η ανάπτυξη τέτοιων συστημάτων ήταν όχι μόνο καθοριστική αλλά και απαραίτητη ήταν αυτός της προσομοίωσης συνθηκών πτήσης αεροσκαφών και γενικά ιπτάμενων μέσων. Με το τρόπο αυτό, οι χειριστές μπορούν να χρησιμοποιήσουν τον προσομοιωτή για να εκπαιδευθούν σε κάθε είδους σενάριο που μπορεί κάποιος να φαντασθεί και το οποίο δε είναι δυνατόν να υλοποιηθεί κατά τη διάρκεια μίας πραγματικής πτήσης.

Τα υψηλής ισχύος υπολογιστικά συστήματα γραφικών επέτρεψαν τους προσομοιωτές πτήσεως να γίνουν περισσότερο ρεαλιστικοί και ευέλικτοι. Αντί του μοντελοποιημένου πεδίου, οι σύγχρονοι προσομοιωτές διαθέτουν μία τράπεζα δεδομένων με γεωγραφικά δεδομένα πολλών πεδίων. Έτσι οι σύγχρονοι προσομοιωτές επιτρέπουν την προσομοίωση συνθηκών πτήσεως και διαδικασιών ανάγκης που αλλιώς θα κατέληγαν σε επικίνδυνες ή θανατηφόρες καταστάσεις αν πραγματοποιούνταν σε πτήση.

##### 2. Μηχανολογία

Η οπτική αναπαράσταση δεν είναι κάτι καινούργιο στη μηχανολογία. Η τεχνολογία

της ΕΠ χαρακτηρίζεται περισσότερο ως συμπλήρωμα στην τεχνολογία της οπτικής αναπαράστασης. Έτσι η ΕΠ προσθέτει απλώς χαρακτηριστικά κατευθυνόμενης κίνησης εντός των μοντέλων CAD αλλά και διάδρασης με το σχεδιαστή και χρήστη. Αυτό βέβαια μπορεί να έχει ως φυσικό επακόλουθο, τη εξάλειψη της ανάγκης δημιουργίας ενός πρωτότυπου σε συγκεκριμένες περιπτώσεις στο προσεχές μέλλον.

Κάποιες τυπικές εφαρμογές της ΕΠ στο πεδίο της Μηχανολογίας αναφέρονται παρακάτω:

#### (α') Σχεδίαση και Κατασκευή Στροβίλων

Σήμερα, αν και όλοι οι τύποι των στροβίλων σχεδιάζονται αποκλειστικά με συστήματα CAD/CAM, προκύπτει ορισμένες φορές η ανάγκη για την κατασκευή πρωτοτύπων αφού ορισμένες λειτουργικές παράμετροι αλλά και θέματα συντήρησης, επιθεωρήσεων και προσβασιμότητας δεν μπορούν να επιλυθούν μόνο με τη χρήση των σχεδιαστικών συστημάτων. Επειδή το κόστος κατασκευής ενός πρωτότυπου πλήρους κλίμακας μαζί με τις διαδικασίες επιθεώρησης, ολοκλήρωσης, τελικής ανάλυσης και εκτίμησης είναι τεράστιο, ορισμένες μεγάλες κατασκευαστριες εταιρίες όπως η McDonnell Douglas<sup>4</sup> οδηγήθηκαν στη λύση της δημιουργίας ενός μοντέλου εμπυθισμένης πραγματικότητας.

#### (β') Σχεδίαση Υποβρυχίων

Στη περίπτωση αυτή χρησιμοποιείται και για την επίλυση θεμάτων εργονομίας. Η εταιρεία VSEL<sup>5</sup> έχει εγκαταστήσει ένα σύστημα ΕΠ πλήρους εμπύθισης που προσομοιώνει ένα πλήρες ωκεάνιο περιβάλλον όπου ο χρήστης μπορεί να εκτιμήσει τις ανάγκες ενός υποβρυχίου λαμβάνοντας υπόψη όλες τις επιχειρησιακές του απαιτήσεις.

#### (γ') Αρχιτεκτονική

Η Αρχιτεκτονική είναι ένα πεδίο όπου η ΕΠ έχει βρει ένα ευρύ φάσμα εφαρμογών

---

<sup>4</sup>Αεροναυπηγική εταιρεία των Η.Π.Α. που χρησιμοποιεί ένα πλήρως εμπυθισμένο σύστημα, το Provision 100 VPX.

<sup>5</sup>Vickers Shipbuilding and Engineering Ltd.

από απλά σχεδιαστικά προγράμματα μέχρι συστήματα πλήρους εμπύθισης ([10]). Η μεγάλη πλειοψηφία των εφαρμογών αυτών παρέχουν τα απαραίτητα εργαλεία για τη μελέτη της προσβασιμότητας κτηρίων και άλλων πολυσύνθετων κατασκευών και χώρων.

#### (δ') Μοντελοποίηση Ανθρώπινων Παραγόντων

Η ΕΠ χρησιμοποιείται όλο και περισσότερο για να μοντελοποιήσει την ανθρώπινη συμπεριφορά και σκέψεις (ίδη [5] για παράδειγμα) στο σχεδιασμό νέων προϊόντων και κτηρίων. Για παράδειγμα, ένα τέτοιο σύστημα μπορεί να προσομοιώσει μία πυρκαγιά σε ένα κτήριο και ο χρήστης να δει με ρεαλισμό τις πιθανές αντιδράσεις σε μία επικίνδυνη κατάσταση αλλά και να επιδείξει τους πιο σημαντικούς παράγοντες που συμβάλλουν όπως είναι τακτικές διαφυγής, μοντέλα πυρκαγιών, ανθρώπινη συμπεριφορά και αντίληψη του χώρου σε πολυσύνθετα κτήρια.

Στο Κέντρο Ερευνών Υπολογιστικών Γραφικών του Πανεπιστημίου της Pennsylvania των Η.Π.Α. (Center for Computer Graphics Research at the University of Pennsylvania) αναπτύχθηκε μία μελέτη που είχε σαν αποτέλεσμα τη δημιουργία του "Jack" (ίδη [28], [19], [30] και [34]), ενός εικονικού ανθρώπου που μπορεί να χρησιμοποιηθεί για να εκτιμηθούν πολλές περιπτώσεις ανθρώπινων παραγόντων σε νέες σχεδιάσεις.

Ο "Jack" είναι εξοπλισμένος με ένα εκλεπτυσμένο σύστημα κινηματικής που "του" επιτρέπει μέσω μηχανικών αντιδράσεων να προσομοιώνει διάφορες ανθρώπινες αντιδράσεις όπως περπάτημα, ανέβασμα σκαλοπατιών και πιάσιμο αντικειμένων χωρίς να χάνει την ισορροπία του. Ο "Jack" μπορεί να χρησιμοποιηθεί και σε άλλα συστήματα ΕΠ με σκοπό να επιτρέψει τη μελέτη της ανθρώπινης συμπεριφοράς και σε άλλες καταστάσεις.

#### (ε') Βιομηχανικές Εφαρμογές

Η επιτυχία μίας νέας τεχνολογίας αντικατοπτρίζεται στο βαθμό αποδοχής της από το βιομηχανικό τομέα. Οι εταιρείες θα υιοθετήσουν και θα χρησιμοποιήσουν μία τεχνολογία μόνον όταν αυτή θα τους λύσει προβλήματα καλύτερα και πιο

αποδοτικά από οποιαδήποτε άλλη εναλλακτική λύση. Οι εφαρμογές της ΕΠ στη βιομηχανία είναι πολύ συχνές και παρακάτω θα δούμε δύο αντιπροσωπευτικές:

- i. Η BMW χρησιμοποιεί ένα σύστημα ΕΠ για να εξετάσει τα αποτελέσματα προσομοιώσεων συγκρούσεων. Ένα αυτοκίνητο μπορεί να συγκρούεται σε ένα τοίχο με ή χωρίς συγκεκριμένα φορτία και επιβάτες αλλά και να παρεμβάλλεται ένα δέντρο εμπρός από το τοίχο ή ο τοίχος να αποκτά τις ιδιότητες του πέτρινου, του σαθρού ή του σιδερένιου.
- ii. Η Fraunhofer IGD<sup>6</sup> χρησιμοποιεί ένα εργαλείο βασισμένο στην ΕΠ για την εκμετάλλευση μεταλλευμάτων το οποίο το ανέπτυξε από κοινού με μία εταιρεία εξορύξεων, την Wismut GmbH. Ο σκοπός ήταν να μπορούν να πείθουν του κατοίκους μίας περιοχής που έφεραν αντίδραση στα αναπτυξιακά τους σχέδια για μία περιοχή, πώς θα έμοιαζε η περιοχή μετά την επέμβασή τους. Έτσι οι κάτοικοι/χρήστες θα μπορούσαν να "περιδιαβούν" το εικονικό τοπίο και να "δουν" με τα μάτια τους αλλά και να "ζήσουν" τα αποτελέσματα των επεμβάσεων της εταιρείας.

#### (ε') Μοντελοποίηση Δικτύων Υπολογιστών

Έχουν αναπτυχθεί πολλές εφαρμογές ΕΠ με σκοπό την εξέταση της τοπολογίας ενός δικτύου και της αντίστοιχης διαδικασίας ελέγχου του. Βέβαια οι εφαρμογές αυτές απευθύνονται σε τοπολογίες πολύ εκτεταμένες και περίπλοκες όπου ο χρήστης της εφαρμογής μπορεί πράγματι να επωφεληθεί από τη χρήση τους.

### 3. Βιομηχανία Κινηματογράφου

Η Βιομηχανία του Κινηματογράφου είναι ένα από τα πεδία που έχουν επωφεληθεί τα μέγιστα από τη χρήση της ΕΠ, αν και ορισμένες προσπάθειες δημιουργίας ταινιών αποκλειστικά με εικονικούς ηθοποιούς δεν έχουν στεφθεί με μεγάλη επιτυχία. Παρόλα αυτά, η τεχνολογία της ΕΠ έχει χρησιμοποιηθεί εκτεταμένα στη δημιουργία ταινιών (π.χ. όταν πια βλέπουμε στις ταινίες πλήθη, αυτά δεν είναι τίποτα άλλο από εικονικούς

<sup>6</sup>Δικτυακός Τόπος: <http://www.igd.fhg.de/>

πράκτορες<sup>7</sup>), αναπαραστάσεις μουσείων για τα παιδιά, υπολογιστικά παιχνίδια (Computer Games) και παιχνίδια τηλεόρασης (TV Games). Παρακάτω θα παρουσιάσουμε σε συντομία τη σύνδεση της ΕΠ με τα Υπολογιστικά Γραφικά (Computer Graphics) και την Υπολογιστική Κινηματική (Animation) αλλά και παραδείγματα από τη βιομηχανία των Μέσων Μαζικής Επικοινωνίας.

**(α') Υπολογιστικά Γραφικά και Κινηματική Αναπαράσταση (Computer Graphics & Animation).**

Το πεδίο των Υπολογιστικών Γραφικών είναι πολύ καλά εδραιωμένο και χρησιμοποιεί αμέτρητες εφαρμογές σε πάμπολλες περιστάσεις. Η ΕΠ επικεντρώνεται στα Γραφικά 3Δ<sup>8</sup>, ένα πεδία σχετικά εξειδικευμένο αλλά πολλά υποσχόμενο στο μέλλον και συνεχώς αναπτυσσόμενο.

Τα Υπολογιστικά Γραφικά βασίζονται στις ίδιες αρχές με αυτές των συστημάτων CAD. Χρησιμοποιούν μία τράπεζα δεδομένων για να αποθηκεύσουν τη γεωμετρική αναπαράσταση των αντικειμένων μέσα στο εικονικό περιβάλλον. Μία Μηχανή Γραφικών (Graphics Engine) κατασκευάζει μία εικόνα της σκηνής που είναι ορατή από οποιαδήποτε γωνία θέασης ή θέση μέσα στη σκηνή και προβάλλει την εικόνα στις δύο διαστάσεις σαν να παρατηρούσαμε μέσα από το φακό μίας φωτογραφικής μηχανής. Ο όρος που χρησιμοποιείται για την δημιουργία της εικόνας είναι Ψηφιακή Αναπαράσταση (Rendering).

Βέβαια θα υπάρχει πάντα ένα αντιστάθμισμα μεταξύ του ρεαλισμού και της διάρκειας. Όσο πιο ρεαλιστική φαίνεται μία εικόνα τόσο πιο πολύ χρόνο απαιτεί η ψηφιακή της αναπαράσταση άρα τόσο πιο αργά θα ανανεώνεται το εικονικό περιβάλλον.

**(β') Υπολογιστικά Παιχνίδια (Computer Games)**

Η τεχνολογία που χρησιμοποιείται στην ανάπτυξη των προσομοιωτών αεροσκαφών και παρόμοιων εκπαιδευτικών εγκαταστάσεων έχουν υιοθετηθεί για χρήση

---

<sup>7</sup>Η έννοια του "πράκτορα" και μάλιστα του νοήμονα θα αναλυθεί παρακάτω.

<sup>8</sup>τρισεπίσταντα

σε Υπολογιστικά Παιχνίδια. Επιπλέον, η έρευνα πάνω στη μοντελοποίηση χαρακτήρων έχει ωφεληθεί σημαντικά αν όχι καθοδηγηθεί από τη βιομηχανία υπολογιστικών παιχνιδιών καθώς η ανάγκη για ρεαλιστικότερους χαρακτήρες είναι μία αναγκαία προϋπόθεση στα σύγχρονα παιχνίδια.

#### 4. Επιστημονική Έρευνα

Οι χρήσεις της ΕΠ είναι πολλές και μπορούμε να τις συναντήσουμε σε όλο το εύρος της έρευνας της επιστημονικής κοινότητας. Ακολούθως, θα παρουσιάσουμε κάποιες από τις πιο ασυνήθεις:

##### (α) Υπολογιστική Νευρολογία

Η Υπολογιστική Νευρολογία είναι ένας κλάδος της νευρολογίας που ερευνά πώς λειτουργεί το νευρικό σύστημα με την ανάπτυξη μοντέλων προσομοίωσης ενός νευρώνα ή δικτύου νευρώνων που βασίζονται σε πειραματικά ευρήματα. Η οπτικοποίηση των στοιχείων της προσομοίωσης από την έρευνα, συχνά υλοποιείται μέσω εξειδικευμένων[20] λογισμικών πακέτων ΕΠ. Τέλος, πολλά ακαδημαϊκές και επιστημονικές μελέτες έχουν συμπεριλάβει ένα βιολογικά αποδεκτό μοντέλο της υπολογιστικής νευρολογίας για να βελτιώσουν συστήματα 3Δ ήχου και διασυνδέσεις συστημάτων ΕΠ.

##### (β') Χημεία και Μοντελοποίηση Μοριακή Δομών

Οι τελευταίες εξελίξεις στην οπτικοποίηση δεδομένων και στην διερεύνηση των αρχών της χημείας περιλαμβάνουν περιβάλλοντα πλήρως εμβυθισμένα όπου ο χημικός περιβάλλεται από στοιχεία και χημικές ενώσεις σε ένα 3Δ χώρο. Υπάρχει ένα ευρύ ερευνητικό έργο (π.χ. [40], [24] και [27]) και μία ανάπτυξη συστημάτων που είναι αφιερωμένα στην εφαρμογή των τεχνικών και της τεχνολογίας της ΕΠ στη μοντελοποίηση μοριακών δομών.

##### (γ') Τηλεπαρουσία

Η Τηλεπαρουσία έχει παρουσιασθεί νωρίτερα και είναι ένα από τα λίγα συστήματα που οπτικοποιούν τον πραγματικό κόσμο με πιθανές εισαγωγές στοιχείων ΕΠ που

δημιουργούνται με υπολογιστικά συστήματα.

Ορισμένες τυπικές εφαρμογές της τηλεπαρουσίας ποικίλουν από πυροσβέστες που χρησιμοποιούν τηλεκατευθυνόμενα οχήματα για να διαχειριστούν επικίνδυνες καταστάσεις έως χειρουργούς που χρησιμοποιούν πολύ μικρά εργαλεία με καλώδια χωρίς να χρειάζεται να κάνουν μεγάλες τομές στους ασθενείς. Η NASA χρησιμοποιεί ήδη μικρά τηλεκατευθυνόμενα ρομπότ για την εξερεύνηση του διαστήματος και των εγγύς πλανητών όπως του Άρη και της Αφροδίτης.

#### (δ') Εξέταση με Υπερήχους

Πρόσφατα έχουν αναπτυχθεί μία μεγάλη ποικιλία από συστήματα Ενισχυμένης ΕΠ για την εξέταση ασθενών με υπερήχους ([8], [9], [15], [31] και [32]). Στις περισσότερες περιπτώσεις, οι εφαρμογές αυτές είναι 3Δ συστήματα λήψης υπερήχων σε συνδυασμό με τη ταυτόχρονη προβολή εικόνας βίντεο του αντικειμένου υπό εξέταση. Συχνά, τέτοια συστήματα είναι εξοπλισμένα με HMD, μηχανικά χέρια και άλλες συσκευές που χρησιμοποιούνται σε εφαρμογές τηλεπαρουσίας.

### 5. Εκπαίδευση

Διακρίνουμε τρία είδη εμπειριών που μπορούμε να εκμεταλλευτούμε στη διαδικασία της μάθησης. Συγκεκριμένα, η πλήρως εμβυθισμένη ΕΠ επιτρέπει να ζήσουμε εμπειρίες που δεν είναι δυνατόν να ζήσουμε στο πραγματικό κόσμο και δεν μπορούν να επιτευχθούν με κάποιο τρόπο. Ο χρήστης μπορεί να γίνει τόσο μικρός ώστε να παίζει με άτομα ή τόσο μεγάλος που να διασχίζει πλανήτες και σύμπαντα. Έπειτα ο σχεδιαστής των συστημάτων ΕΠ μπορεί να επηρεάσει κατάλληλα τις ανθρώπινες αισθήσεις με κάποιο τρόπο ώστε να γίνει αντιληπτή η επιρροή μίας δράσης που σε φυσιολογικές συνθήκες αυτό είναι αδύνατο. Έτσι μπορεί να μεταβάλλει την ένταση και συχνότητα του ήχου για να δείξει την επίδραση μίας ραδιενεργούς ακτινοβολίας ή να επιβάλλει διαφορετικά χρώματα σε περιοχές ενός κτηρίου για να δείξει τις διαφορετικές θερμοκρασίες που επικρατούν. Τέλος, η ΕΠ επιτρέπει τη δημιουργία και οπτικοποίηση ιδεών και γεγονότων, που είναι γενικά αφηρημένες, συνδυάζοντας χαρακτηριστικά από τις δύο προηγούμενες τεχνικές.

### 1.1.5 Εφαρμογές ΕΠ στη Ναυπηγική

Οι τεχνικές της ΕΠ, όπως αναφέρθηκε και στην προηγούμενη ενότητα, έχουν χρησιμοποιηθεί με επιτυχία σε πολλές εφαρμογές που ποικίλουν από τη μηχανολογία και τη διασκέδαση μέχρι την επιστημονική έρευνα και την εκπαίδευση. Η ναυπηγική βιομηχανία είναι πολυσύνθετη και έχει οδηγήσει πολλούς ερευνητές στη χρήση τεχνικών ΕΠ σε αρκετές φάσεις της μελέτης, κατασκευής και επιχειρησιακής εκμετάλλευσης ενός πλοίου.

Η ΕΠ μπορεί να χρησιμοποιηθεί στη διερεύνηση πολλών προβλημάτων και θεμάτων που συχνά ανακύπτουν στη ναυπηγική βιομηχανία και το πεδίο της Ναυπηγίας και Ναυτικής Μηχανολογίας. Ειδικά στη Ναυτική Μηχανολογία, η προκαταρκτική εξέταση της χωροταξίας του μηχανοστασίου και του επακόλουθου βαθμού δυσκολίας στη συντήρηση λόγω προσβασιμότητας του προσωπικού μπορεί να βελτιωθεί σημαντικά με τη χρήση τέτοιων τεχνικών. Ένας πρόχειρος και σίγουρα όχι πλήρης κατάλογος πιθανών χρήσεων παρατίθεται παρακάτω:

#### 1. Κανονισμοί και Ασφάλεια

- (α') Προσομοίωση Ατυχημάτων και Αξιολόγηση Κανονισμών.
- (β') Προσομοίωση Φόρτωσης.
- (γ') Αξιολόγηση νέων Κανονισμών.

#### 2. Σχεδίαση

- (α') Κίνηση εντός Σχεδιαζόμενων Χώρων και Κατασκευών, Επιθεώρηση νέων Σχεδίων.
- (β') Άμεση εφαρμογή Αλλαγών στη Σχεδίαση.
- (γ') Επιθεώρηση Εξοπλισμού και Ενδιαίτησης Επιβατών.

#### 3. Υδροδυναμική

- (α') Προσομοίωση Προτύπων σε Δεξαμενή.
- (β') Προσομοίωση Ελιγμών και Ελικτικών Στοιχείων.



## 1.2 Πεδίο Έρευνας

Η παρούσα εργασία πάνω στην ΕΠ θα επικεντρωθεί σε τρεις κατευθύνσεις:

- Μοντελοποίηση Πλήθους.
- Αυτόνομοι Χαρακτήρες.
- Καθοδηγητικές Συμπεριφορές.

### 1.2.1 Μοντελοποίηση πλήθους

Το Πλήθος ως φαινόμενο έχει μελετηθεί από τα τέλη το 19ου αιώνα [18]. Βέβαια με την έλευση των υπολογιστικών συστημάτων έγινε δυνατή όχι μόνο η παρατήρηση του πλήθους ανθρώπων αλλά και η προσομοίωση ορισμένων φαινομένων συλλογικής συμπεριφοράς σε εικονικό περιβάλλον. Οι συλλογικές συμπεριφορές έχουν μελετηθεί και μοντελοποιηθεί για διάφορους σκοπούς. Εκτός από ορισμένα σπάνια παραδείγματα που αφορούν τη προσομοίωση πλήθους στο ευρύτερο πλαίσιο [26], οι περισσότερες προσομοιώσεις αφορούσαν συγκεκριμένες εφαρμογές της συμπεριφοράς πλήθους. Συνεπώς, υπάρχει ένα μεγάλο εύρος τεχνικών μοντελοποίησης που ποικίλουν από μοντέλα που αφορούν δίκτυα και ροή και όπου δεν υπάρχει σαφής διαχωρισμός του ατόμου που συμμετέχει στο πλήθος έως μοντέλα που περιγράφουν το κάθε άτομο ξεχωριστά και βασίζονται σε φυσικούς νόμους και καθοδηγητικές συμπεριφορές. Η μοντελοποίηση πλήθους περιλαμβάνεται στις εφαρμογές που χρησιμοποιούνται στην βιομηχανία του κινηματογράφου και συγκεκριμένα στην κινηματική αναπαράσταση (Animation). Επίσης χρησιμοποιούνται ευρέως σε προσομοιώσεις στρατιωτικού [36] και αστυνομικού [39] προσωπικού, προσομοιώσεις αρχιτεκτονικών εφαρμογών όπου μας ενδιαφέρει η ροή και η κυκλοφορία του πλήθους εντός μίας καινούργιας ή υφιστάμενης κατασκευής ή κτηρίου [4] αλλά και σε περιπτώσεις μελέτης διαφυγής από κτήρια σε καταστάσεις ανάγκης [33], [35], [37] και [2]. Υπάρχουν όμως εφαρμογές όπου επιχειρείται η προσομοίωση της δυναμικής του πλήθους και των φυσικών της συνεπειών [12] αλλά και κοινωνιολογικές προσομοιώσεις συμπεριφορών [21].

*Υλοποίηση Καθοδηγητικών Συμπεριφορών*

Στην περίπτωση της παρούσας διπλωματικής εργασίας εξετάζεται κατά κύριο λόγο η μοντελοποίηση του πλήθους διαμέσου ενός περιβάλλοντος ΕΠ. Οι τεχνικές που χρησιμοποιούνται, βασίζονται σε πρόσφατα αποτελέσματα που αφορούν τη μοντελοποίηση πλήθους μέσω ενός περιβάλλοντος ΕΠ. Βέβαια, οι τεχνικές αυτές βελτιώνονται με σκοπό να διευκολύνουν την περαιτέρω εφαρμογή τους στον ναυπηγικό τομέα<sup>9</sup>.

## 1.2.2 Αυτόνομοι Χαρακτήρες (Autonomous Characters)

Πριν προχωρήσουμε στην παράθεση των κύριων χαρακτηριστικών ενός Αυτόνομου Χαρακτήρα, κρίνεται σκόπιμο να γίνει μία παρουσίαση της έννοιας του Αυτόνομου Πράκτορα δεδομένου ότι η έννοια αυτή επιδέχεται πολλών ερμηνειών και έχει χρησιμοποιηθεί πολλές φορές ακόμα και σε εφαρμογές που δεν έχουν καμία λογική σύνδεση.

### 1.2.2.1 Εισαγωγή στους Αυτόνομους Πράκτορες

Οι νοήμονες πράκτορες (Intelligent Agents) είναι ένας από τους πιο πρόσφατους και με μεγαλύτερο ενδιαφέρον κλάδους της Τεχνητής Νοημοσύνης. Ένας πράκτορας (Agent) είναι μία οντότητα που αντιλαμβάνεται το περιβάλλον μέσα στο οποίο βρίσκεται με τη βοήθεια αισθητήρων (sensors), είναι μέρος του περιβάλλοντος αυτού, κάνει συλλογισμούς για το περιβάλλον και δρα πάνω σε αυτό με τη βοήθεια μηχανισμών δράσης (effectors), για την επίτευξη κάποιων στόχων.

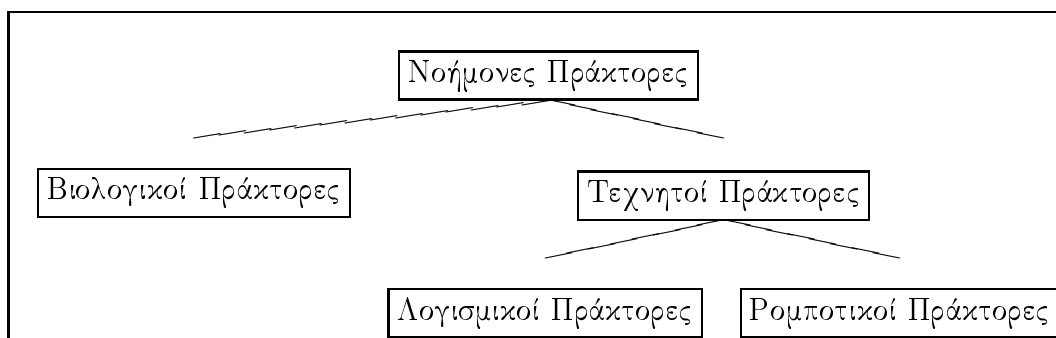
Ο παραπάνω ορισμός εμπεριέχει την έννοια της αυτονομίας ενός πράκτορα, δηλαδή της αυτενέργειας για την επίτευξη του σκοπού του. Αυτό το χαρακτηριστικό είναι ίσως ο πιο κοινός παρονομαστής όλων των ειδών των πρακτόρων, και υποχρεώνει την ύπαρξη νοημοσύνης τουλάχιστον σε κάποιο βαθμό. Οι πράκτορες αποτελούν ίσως την πιο "ανθρωπόμορφη" αρχιτεκτονική λογισμικού στην οποία αποδίδονται συχνά και ανθρωποκεντρικά χαρακτηριστικά όπως θα δείξουμε παρακάτω.

---

<sup>9</sup>Ιδιαίτερα στη περίπτωση της εγκατάλειψης του πλοίου από τους επιβάτες.

### 1.2.2.2 Ταξινόμηση Πρακτόρων

Σε ένα πιο αφηρημένο επίπεδο οι πράκτορες μπορούν να διαχωριστούν σε δύο μεγάλες κατηγορίες: τους βιολογικούς και τους τεχνητούς πράκτορες (Σχήμα 1.2).



Σχήμα 1.2: Μία Ταξινόμηση των Πρακτόρων.

Οι βιολογικοί πράκτορες χρησιμοποιούν τις αισθήσεις τους για να αντιληφθούν το περιβάλλοντα κόσμο, τις γνώσεις τους για να βγάλουν συμπεράσματα για αυτόν και τα μέρη του σώματός τους για να εφαρμόσουν τις ενέργειες που προκύπτουν από τη συλλογιστική τους.

Οι τεχνητοί πράκτορες (*Artificial Agents*) λειτουργούν με παρόμοιο τρόπο και χωρίζονται σε δύο υποκατηγορίες:

- Τους ρομποτικούς πράκτορες (*Romprotic Agents* ή *Robots*), οι οποίοι έχουν σαν αισθητήρες και μηχανισμούς δράσης μηχανικά ή ηλεκτρονικά μέρη και δρουν στον πραγματικό κόσμο.
- Τους λογισμικούς πράκτορες (*Software Agents* ή *Softbots*), οι οποίοι είναι προγράμματα και δρουν σε ένα υπολογιστικό σύστημα.

Και τα δύο είδη πρακτόρων εμπεριέχουν μία συλλογιστική διαδικασία (*Reasoning*), μέσω της οποίας επεξεργάζονται τα ερεθίσματα του περιβάλλοντός τους και εφαρμόζουν τα αποτελέσματα της συλλογιστικής στο περιβάλλον αλλάζοντας έτσι την κατάστασή του.

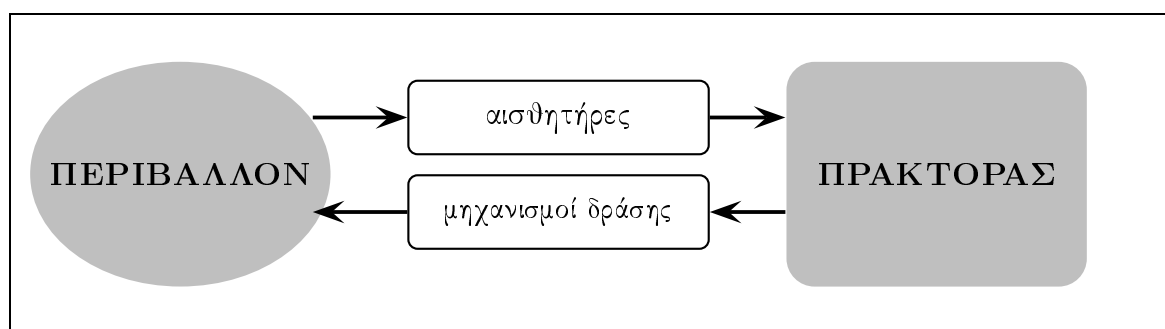
Οι πράκτορες αποτελούν ένα κοινό πεδίο ενδιαφέροντος για πολλές περιοχές της Επιστήμης των Υπολογιστών, καθώς μπορούν να εξεταστούν από διαφορετικές για κάθε πεδίο σκοπιές. Έτσι για τη Τεχνητή Νοημοσύνη οι πράκτορες θεωρούνται ευφυείς οντότητες,

για τον αντικειμενοστραφή προγραμματισμό θεωρούνται οντότητες (αντικείμενα) που επικοινωνούν μεταξύ τους, για τον παράλληλο προγραμματισμό είναι οντότητες που εκτελούνται παράλληλα και για το πεδίο διασύνδεσης ανθρώπου - μηχανής είναι οντότητες που παίρνουν πρωτοβουλία σε συνεργασία με το χρήστη για να επιλέξουν τα καθήκοντά τους.

Οι επιθετικοί προσδιορισμοί συχνά χαρακτηρίζουν την κύρια λειτουργία των πρακτόρων, όπως πράκτορες αναζήτησης, αναφοράς, παρουσίασης, προσανατολισμού, διαχείρισης κλπ. Η παρούσα εργασία αφορά κυρίως *νοήμονες λογισμικούς πράκτορες (intelligent software agents)*. Η κατηγορία περιλαμβάνει ένα μεγάλο εύρος υπολογιστικών οντοτήτων, στο οποίο περιέχονται από σχετικά απλά συστήματα, όπως είναι ο "συνδετήρας" που παρέχει συμβουλές στο πρόγραμμα επεξεργασίας κειμένου WORD<sup>10</sup>, μέχρι κατανεμημένα συστήματα ελέγχου βιομηχανικών εγκαταστάσεων, όπως είναι το ARCHON - 3M Motor Vehicle Systems<sup>11</sup>

### 1.2.2.3 Διάφοροι Ορισμοί Πρακτόρων

Η έλλειψη ενιαίου ορισμού σε καμία περίπτωση δεν αποτελεί ανασταλτικό παράγοντα για τη διάδοση και χρήση της τεχνολογίας αυτής σε πολλές εφαρμογές, καθώς επίσης και της έρευνας που διεξάγεται ολοένα και από περισσότερους επιστήμονες στον κόσμο. Το μόνο πρόβλημα που δημιουργείται είναι ότι η λέξη "πράκτορας" χρησιμοποιείται πολλές φορές αδικαιολόγητα για πολλά από τα συστήματα που αναπτύσσονται.



Σχήμα 1.3: Ο βασικός πράκτορας κατά Russel & Norvig.

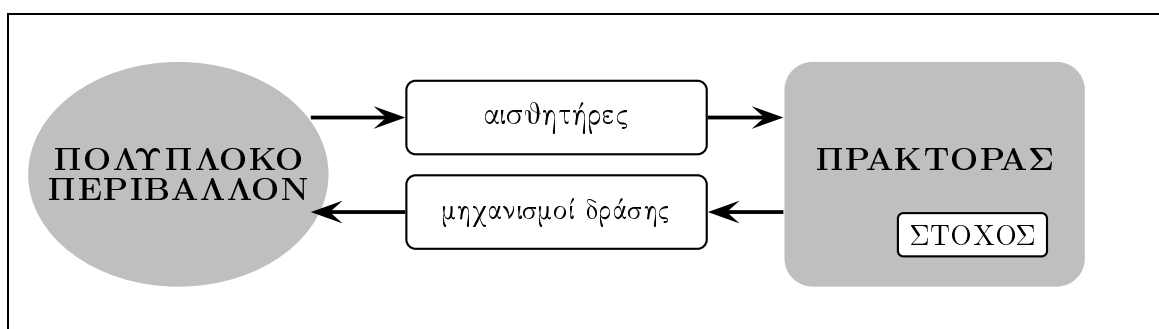
Οι Russel & Norvig δίνουν έμφαση στην αλληλεπίδραση πρακτόρων με το περιβάλλον

<sup>10</sup>Προϊόν λογισμικού της Microsoft Corporation.

<sup>11</sup>Η εταιρεία ARCHON εξαγοράστηκε τον Αύγουστο του 2006 από την 3M Motor Vehicle Systems Corp.

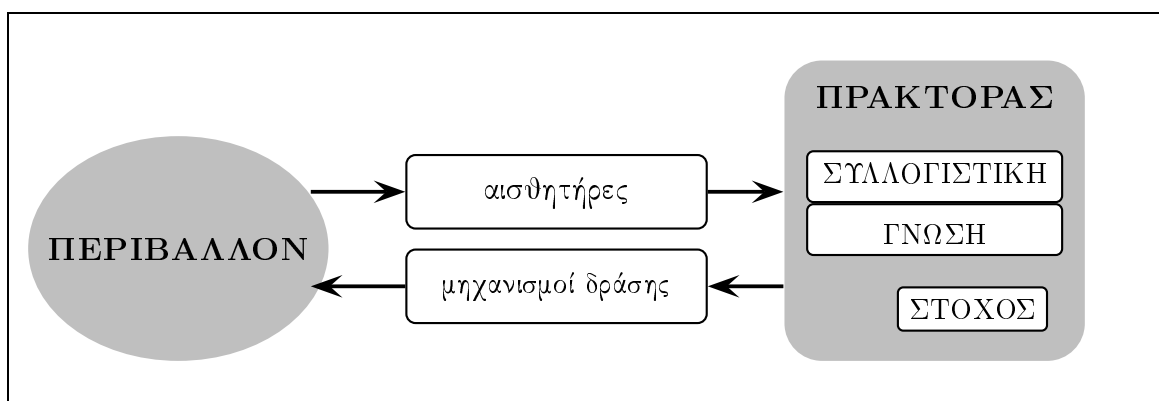
τους, αλλά από τον ορισμό τους αφήνεται να εννοηθεί ότι κάθε πρόγραμμα είναι και πράκτορας: "Πράκτορας είναι οτιδήποτε μπορεί να αντιληφθεί το περιβάλλον του μέσω αισθητήρων και να αντιδράσει πάνω στο περιβάλλον μέσω μηχανισμών δράσης" (Σχήμα 1.3).

Ο *Virdhagrishwaran* δίνει έμφαση στην αυτόνομη εκτέλεση: "Ο όρος πράκτορας χρησιμοποιείται για να αναπαραστήσει δύο ορθογώνιες έννοιες: την αυτόνομη εκτέλεση και τη συλλογιστική πάνω σε κάποιο πεδίο".



Σχήμα 1.4: Ο βασικός πράκτορας κατά Maes.

Η *Maes* εκτός της αυτονομίας, δίνει έμφαση και στο πολύπλοκο και δυναμικό περιβάλλον: "Οι πράκτορες είναι υπολογιστικά συστήματα που δρουν σε ένα πολύπλοκο περιβάλλον, αντιλαμβάνονται και δρουν αυτόνομα πάνω σε αυτό, πετυχαίνοντας έτσι ένα σύνολο από στόχους για τους οποίους έχουν κατασκευαστεί" (Σχήμα 1.4).

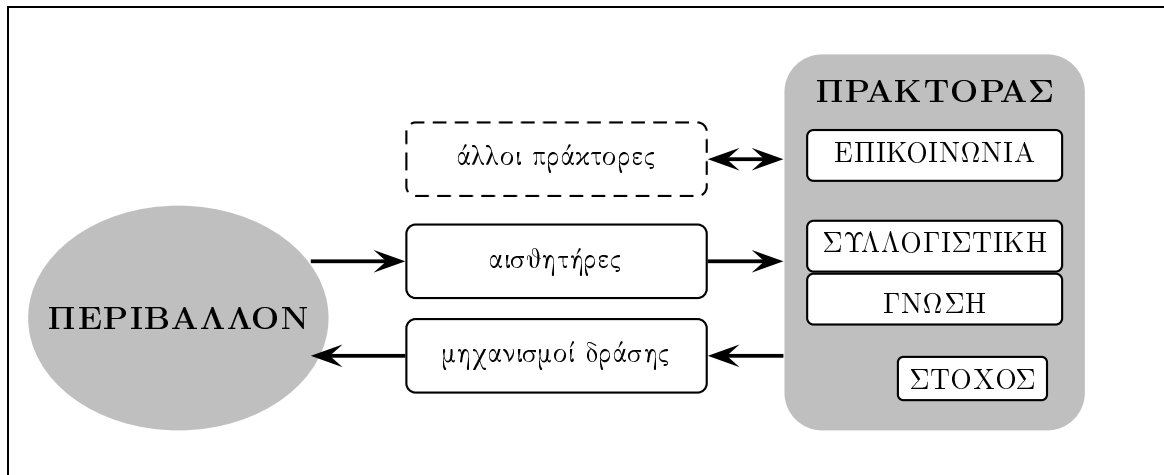


Σχήμα 1.5: Ο βασικός πράκτορας κατά Hayes-Roth.

Η *Hayes-Roth* δίνει έμφαση στη συλλογιστική: "Οι ευφυείς πράκτορες κάνουν συνεχώς τις εξής λειτουργίες:

*Υλοποίηση Καθοδηγητικών Συμπεριφορών*

- αντιλαμβάνονται τις δυναμικές συνθήκες του περιβάλλοντος,
- δρουν πάνω στο περιβάλλον ώστε να το αλλάξουν και
- συλλογίζονται ώστε να ερμηνεύσουν αυτά που αντιλαμβάνονται, να λύσουν προβλήματα, να συμπεράνουν και να καθορίσουν τη θέση τους" (Σχήμα 1.5).



Σχήμα 1.6: Ο βασικός πράκτορας κατά Coen.

Τέλος ο Coen δίνει έμφαση στη διαδραστικότητα (*interactivity*): "Λογισμικοί πράκτορες είναι προγράμματα που διενεργούν διάλογο, διαπραγματεύονται και συντονίζουν τη ροή των πληροφοριών" (Σχήμα 1.6).

#### 1.2.2.4 Χαρακτηριστικά Πρακτόρων

Ένα ερώτημα που τίθεται σε κάθε εισαγωγή στο πεδίο των πρακτόρων είναι σε τι διαφέρουν οι πράκτορες (κυρίως οι λογισμικοί) από τα συμβατικά προγράμματα. Η απάντηση δεν είναι απλή, καθώς από τη μία τα όρια είναι πολλές φορές δυσδιάκριτα και από την άλλη ο όρος πράκτορας περιλαμβάνει ένα πλήθος συστημάτων με διαφορές τόσο στη πολυπλοκότητα όσο και στα επιμέρους χαρακτηριστικά.

Τα κύρια χαρακτηριστικά ενός νοήμονα πράκτορα, τα οποία αποτελούν και τις βασικές διαφορές μεταξύ των πρακτόρων και προγραμμάτων, παρατίθενται στο [], όπου Πράκτορας είναι ένα σύστημα υλικού ή λογισμικού που έχει τις παρακάτω ιδιότητες:

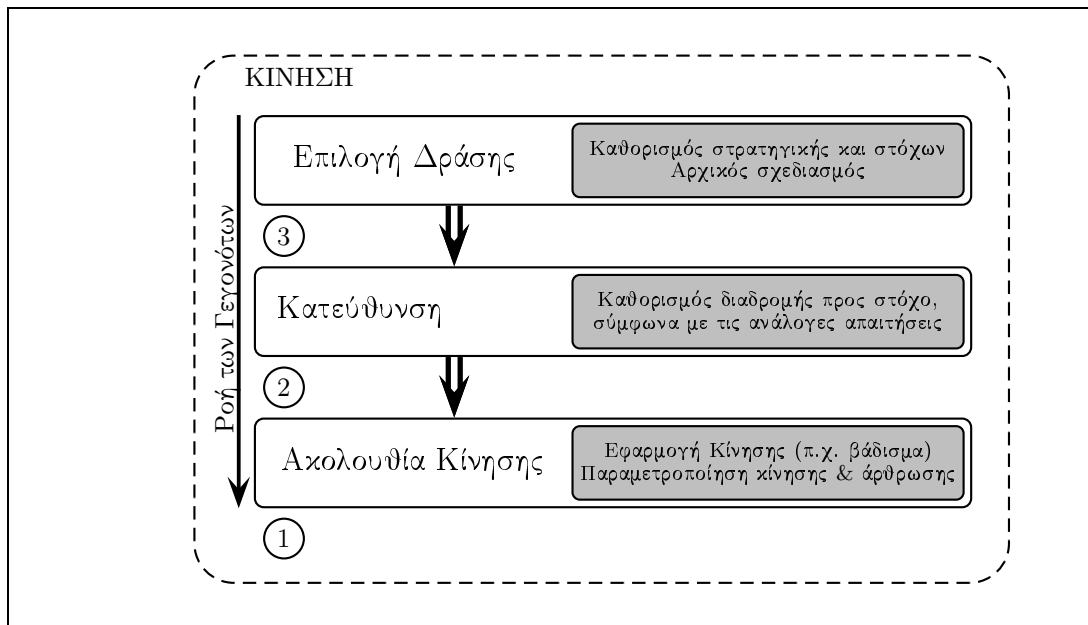
- **Αυτονομία (Autonomy):** Οι πράκτορες λειτουργούν χωρίς την άμεση παρέμβαση των χρηστών ή άλλων πρακτόρων και έχουν αυτοέλεγχο, δηλαδή έλεγχο της εσωτερικής τους κατάστασης και αυτενέργεια. Αυτό σημαίνει ότι οι πράκτορες έχουν τη δυνατότητα να επιδιώκουν τους στόχους τους χωρίς να δέχονται συνεχώς εντολές από το χρήστη ή κάποια άλλη εξωτερική πηγή, το οποίο έχει σαν αποτέλεσμα να αφήνει στο χρήστη μόνο τον προσδιορισμό του γενικού στόχου και να τον απελευθερώνει από το βάρος της λήψης των επιμέρους αποφάσεων.
- **Κοινωνικότητα (Socialability):** Οι πράκτορες επικοινωνούν με άλλους πράκτορες και χρήστες μέσω μίας κοινά κατανοητής γλώσσας, έτσι ώστε να μπορούν να συνεργαστούν για την επίτευξη των στόχων τους. Υπάρχει δηλαδή αλληλεπίδραση μεταξύ των πρακτόρων είτε για την επίτευξη των ανεξάρτητων στόχων τους είτε για την επίτευξη ενός κοινού στόχου.
- **Δυνατότητα Αντίδρασης (Reactiveness):** Οι πράκτορες αντιλαμβάνονται το περιβάλλον τους και αντιδρούν μέσα σε συγκεκριμένα χρονικά πλαίσια στις αλλαγές που επέρχονται σε αυτό.
- **Προνοητικοί (Pro-Activeness):** Οι πράκτορες δεν αντιδρούν απλά στο περιβάλλον αλλά είναι ικανοί να επιδείξουν και συμπεριφορά που βασίζεται σε στόχους, λαμβάνοντας ουσιαστικά κάποια πρωτοβουλία ανάλογα με τις συνθήκες οι οποίες εμφανίζονται στο περιβάλλον τους. Τόσο η προνοητικότητα όσο και η δυνατότητα αντίδρασης απαιτούν σε κάποιο βαθμό τη δυνατότητα συλλογισμού από τον πράκτορα.

#### 1.2.2.5 Ορισμός των Αυτόνομων Χαρακτήρων

Οι Αυτόνομοι Χαρακτήρες είναι ένα είδος *Αυτόνομων Πρακτόρων* που προορίζονται για χρήση στην Κινηματική Αναπαράσταση με Υπολογιστικά Συστήματα αλλά και διαδραστικά πολυμέσα όπως τα παιχνίδια και η ΕΠ. Οι Πράκτορες αυτοί αναπαριστούν ένα Χαρακτήρα σε ένα σενάριο ή ένα παιχνίδι και έχουν τη δυνατότητα να προοδεύουν και να βελτιώνουν τις αντιδράσεις τους. Αυτό έρχεται σε αντίθεση με ένα χαρακτήρα σε

μία κινηματογραφική ταινία με γραφικά, του οποίου οι διάλογοι και οι αντιδράσεις είναι προδιαγεγραμμένα και σε ένα ανδρείκελο (Avatar) σε ένα παιχνίδι ή μία ΕΠ, όπου οι πράξεις του υπαγορεύονται από το χρήστη ή τον παίκτη σε πραγματικό χρόνο. Στα παιχνίδια οι Αυτόνομοι Χαρακτήρες καλούνται ορισμένες φορές και *Χαρακτήρες Εκτός Παιδιάς (Non-Player Character)*

Ένας Αυτόνομος Χαρακτήρας πρέπει να συνδυάζει διάφορα χαρακτηριστικά ενός αυτόνομου ρομπότ και τις ικανότητες ενός ανθρώπου-ηθοποιού σε ένα θέατρο πειραματικής σκηνης. Βέβαια αυτοί οι χαρακτήρες δεν μπορούν να ονομασθούν ούτε ρομπότ αλλά ούτε άνθρωποι, αλλά μάλλον ότι συνδυάζουν κάποια χαρακτηριστικά και από τους δύο.



Σχήμα 1.7: Διαστρωμάτωση της Κίνησης.

### 1.2.3 Καθοδηγητικές συμπεριφορές

Ο όρος "Συμπεριφορά" (Behavior) έχει πολλές σημασίες. Μπορεί να αναπαραστήσει τη σύνθετη δράση ενός ανθρώπου ή ενός ζώου βασισμένη σε σκέψη ή ένστικτο. Μπορεί να ανταποκριθεί σε προβλέψιμες δράσεις ενός απλού μηχανικού συστήματος ή τη σύνθετη δράση ενός χαωτικού συστήματος. Στην ΕΠ και στις εφαρμογές πολυμέσων, χρησιμοποιείται



ορισμένες φορές και σαν συνώνυμο της υπολογιστικής αναπαράστασης. Στην εργασία αυτή ο όρος θα χρησιμοποιηθεί για να περιγράψει τη κίνηση ενός αυτόνομου πράκτορα ή χαρακτήρα που προσομοιάζει περισσότερο σε αυτή του ανθρώπου.

Επιπλέον, η συμπεριφορά ενός αυτόνομου χαρακτήρα μπορεί να γίνει περισσότερο κατανοητή όταν διαιρέσουμε το μηχανισμό της κίνησης σε περισσότερα από ένα επίπεδα. Το σχήμα 1.7 δείχνει τη διαίρεση του μηχανισμού της κίνησης στα αντίστοιχα συστατικά της συμπεριφοράς σε τρία επίπεδα: επιλογή δράσης, κατεύθυνση και ακολουθία κίνησης. Βέβαια περαιτέρω διαιρέσεις σε περισσότερα υποστρώματα και επίπεδα είναι δυνατή αλλά θα αρκεστούμε στη θεώρηση αυτή. Μία παρόμοια ιεράρχηση περιγράφεται από τους Blumberg & Galyen στο [3], όπου τα επίπεδα αναφέρονται σαν: διέγερση, σκοπός και μηχανή. Θα πρέπει να σημειωθεί ότι αν και η ιεράρχηση αυτή θεωρείται ότι είναι εφαρμόσιμη στη περίπτωση της μοντελοποίησης συμπεριφορών ενός αυτόνομου πράκτορα, εντούτοις άλλες συμπεριφορές, όπως αυτή της συνομιλίας ενός "chatterbot"<sup>12</sup>, απαιτεί εντελώς διαφορετική δομή.

□

---

<sup>12</sup>λογισμικός πράκτορας που χρησιμοποιείται για να διεξάγει συνομιλίες με ανθρώπους, ρεαλιστικά και με ευρηματικό τρόπο πολλές φορές.



## Κεφάλαιο 2

# Λογισμική Πλατφόρμα Εικονικής Πραγματικότητας

Το λογισμικό της εργασίας αυτής είναι βασισμένο πάνω στη πλατφόρμα ΕΠ VRSystem. Το VRSystem, με τη σειρά της, είναι μία δικτυακή εφαρμογή ΕΠ και αναπτύσσεται με βάση έναν κεντρικό εξυπηρετητή (Server) και πολλούς πελάτες (Clients). Έχει τη δυνατότητα εναγκαλισμού υπολογιστικών προγραμμάτων ναυπηγικού ενδιαφέροντος για τις ανάγκες της φυσικής αναπαράστασης της κίνησης αλλά και της αναπαράστασης του πλοίου.

Η πλατφόρμα αυτή είναι πολυχρηστική<sup>1</sup> και πολυπρακτορική<sup>2</sup> και μπορεί να χρησιμοποιηθεί ως μέσο διερεύνησης της κίνησης επιβατών ενός πλοίου σε διάφορες καταστάσεις ανάγκης ή κίνησης του πλοίου σε πραγματικό χρόνο.

### 2.1 Πυρήνας Εικονικής Πραγματικότητας (VRKernel)

Ένα από τα βασικά συστατικά του VRSystem είναι ο VRKernel, ο πυρήνας υπολογισμών της πλατφόρμας, όπου ουσιαστικά γίνεται:

- η υλοποίηση του συνθετικού κόσμου λαμβάνοντας υπόψη γεωμετρικές αναπαραστάσεις των αντικειμένων που λαμβάνουν μέρος στην ΕΠ του VRSystem,

---

<sup>1</sup>μπορεί να χρησιμοποιηθεί ταυτόχρονα από πολλούς χρήστες

<sup>2</sup>περιλαμβάνει τη δράση πολλών πρακτόρων τεχνητής νοημοσύνης ταυτόχρονα

- η μοντελοποίηση του πλήθους που λαμβάνει μέρος στην ΕΠ,
- η υλοποίηση των μηχανισμών αποφυγής σύγκρουσης των κινούμενων χαρακτήρων και πρακτόρων,
- η διαχείριση των γεγονότων δράσης (Event Handling) και
- η εκτέλεση όλων των υπόλοιπων λειτουργιών που σχετίζονται με την οπτικοποίηση και το γενικότερο σχεδιασμό και οργάνωση της εκάστοτε σκηνής.

Ο *VRKernel* είναι το κύριο τμήμα της πλατφόρμας *VRSystem* στις εφαρμογές του εξυπηρετητή και του πελάτη. Θα μπορούσαμε να χαρακτηρίσουμε τον πυρήνα αυτόν ως μία κεντρική βιβλιοθήκη αντικειμένων<sup>3</sup> και λειτουργιών κατάλληλες για την ανάπτυξη εφαρμογών ΕΠ που σχετίζονται με τη μοντελοποίηση του πλήθους σε καταναμημένο περιβάλλον.

Ο πυρήνας περιλαμβάνει ένα αριθμό στοιχείων που είτε έχουν σχεδιαστεί είτε τροποποιηθεί για να επιτρέπουν τη συνδυασμένη τους χρήση. Συνεπώς, δεν είναι μία απλή συλλογή αλγορίθμων αλλά ένας ολοκληρωμένος πυρήνας ΕΠ που είναι αρκετά εστιασμένος για να εξυπηρετήσει τις ανάγκες της διερεύνησης της συμπεριφοράς ενός πλήθους αλλά και αρκετά ευρύς σε δυνατότητες ώστε να αποτελεί το πλαίσιο ανάπτυξης λογισμικών ΕΠ.

Τα κύρια συστατικά του Πυρήνα και οι τεχνολογίες που εφαρμόστηκαν για την ολοκλήρωσή του είναι:

1. Μοντελοποίηση 2Δ/3Δ γεωμετρικών σχημάτων (Geometric/Solid Modeler).
2. Μοντελοποίηση Εικονικής Πραγματικότητας (VR Modeler).
3. Μοντελοποίηση νόμων Φυσικής (Physics Modelling).
4. Υλοποίηση μοντέλου πλήθους (Crowd Modelling).
5. Προσομοίωση κίνησης περιβάλλοντος (Environment Motion Simulation).
6. Μηχανή γραφικών (Graphics Engine)

---

<sup>3</sup>Η όλη εφαρμογή είναι σχεδιασμένη σε αντικειμενοστραφή προγραμματισμό

7. Διαστρωμάτωση δικτύου (Network Layer)

8. Θεμελίωση περιβάλλοντος χρήστη (User Interface Foundations)

Στις ακόλουθες ενότητες θα γίνει μία συνοπτική παρουσίαση των επιμέρους αναφερθέντων τμημάτων του Πυρήνα Εικονικής Πραγματικότητας με μόνο γνώμονα την κατανόηση από τον αναγνώστη του περιβάλλοντος εντός του οποίου κινήθηκε η παρούσα διπλωματική εργασία.

## 2.2 Μοντελοποίηση 2Δ/3Δ Γεωμετρικών Σχημάτων

Η καρδιά του VRKernel είναι μία βιβλιοθήκη 2Δ/3Δ μοντέλων γεωμετρικών σχημάτων που βασίζεται στο λογισμικό πακέτο Open Inventor. Η βιβλιοθήκη αυτή μπορεί και υποστηρίζει το σύστημα ΕΠ που έχει αναπτυχθεί, δεδομένου ότι οι απαιτούμενες τεχνικές υλοποίησης αλλά και οι αλγόριθμοι γραφικών που εφαρμόζονται σε γεωμετρικά αντικείμενα μπορούν και ορίζονται με τη βοήθειά του.

Το πακέτο Open Inventor παρέχεται σε πολλές εκδόσεις. Η αρχική του έκδοση είχε αναπτυχθεί από την SGI<sup>4</sup>, αλλά σήμερα είναι διαθέσιμες τόσο εμπορικές εκδόσεις όπως αυτή της Mercury αλλά και ελεύθερου λογισμικού όπως αυτή της Coin. Το λογισμικό της παρούσας εργασίας όπως και του VRKernel κάνει χρήση και των δύο αυτών εκδόσεων, δεδομένου ότι οι απαιτήσεις μας καλύπτονται μόνο με επικάλυψη των δύο αυτών πακέτων.

Θεωρείται απαραίτητη μία συνοπτική περιγραφή του λογισμικού πακέτου Open Inventor αφού πρόκειται για τη βάση της υλοποίησης των γραφικών στην παρούσα εργασία.

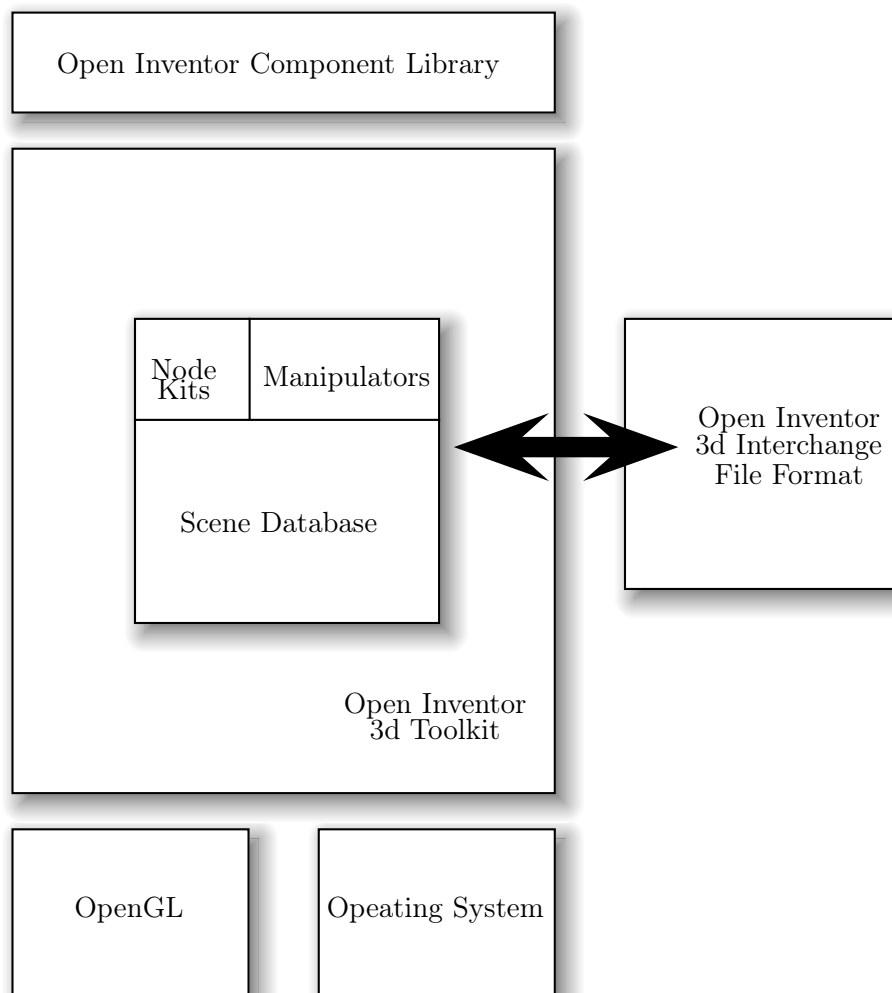
### 2.2.1 Open Inventor

Το λογισμικό πακέτο Open Inventor είναι μία βιβλιοθήκη αντικειμενοστραφούς προγραμματισμού και μπορεί να χρησιμοποιηθεί για τη δημιουργία διαδραστικών εφαρμογών με

---

<sup>4</sup>Silicon Graphics Incorporation

3Δ γραφικά (βλ. [38]). Παρέχει μία βιβλιοθήκη αντικειμένων που εμπεριέχουν μη αντικειμενοστραφή κώδικα γλώσσας προγραμματισμού C++ για την περιγραφή και δημιουργία γραφικών σε πρότυπο OpenGL ver. 1.0 και μπορούν να χρησιμοποιηθούν όπως είναι ή να επεκταθούν και μεταβληθούν για να ανταποκριθούν στις ανάγκες μας. Με το τρόπο αυτό παρέχει την οικονομία και αποτελεσματικότητα ενός πλήρους αντικειμενοστραφούς συστήματος με όλες τις δυνατότητες της OpenGL. Επιπλέον εξασφαλίζει την μεταφορά των αντικειμένων μεταξύ εφαρμογών μέσω της μορφοποίησης 3D Interchange File Format. Έτσι οι τελικοί χρήστες των 3Δ εφαρμογών που βασίζονται στο Open Inventor, μπορούν να μεταφέρουν 3Δ αντικείμενα με την ευκολία της αντιγραφής & επικόλλησης (cut & paste).



Σχήμα 2.1: Η Αρχιτεκτονική του Open Inventor.

Όπως φαίνεται και στο σχήμα 2.1, ο Open Inventor θεμελιώνεται πάνω στην OpenGL και το εν χρήση λειτουργικό σύστημα. Αναπαριστά μία συλλογή αντικειμενοστραφών εφαρμογών και παρέχει μία διεπιφάνεια προγραμματισμού και επικοινωνίας με τα προγράμματα της OpenGL. Η βιβλιοθήκη ανάπτυξης είναι ανεξάρτητη του λειτουργικού συστήματος αφού χρησιμοποιείται και μία βιβλιοθήκη προσαρμογής στο αντίστοιχο λειτουργικό σύστημα.

### 2.2.1.1 Αντικείμενα

Ο Open Inventor επικεντρώνεται στη δημιουργία 3Δ αντικειμένων. Όλες οι πληροφορίες για τα αντικείμενα αυτά (σχήμα, μέγεθος, χρώμα, υφή επιφάνειας, θέση στο χώρο) αποθηκεύονται στη τράπεζα δεδομένων της σκηνής. Οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν με διάφορους τρόπους αλλά ο πιο συνηθισμένος είναι να απεικονίσει στην οθόνη το 3Δ αντικείμενο. Βέβαια για πολλά 3Δ γραφικά λογισμικά πακέτα, η εικόνα αυτή είναι και ο αντικειμενικός τους σκοπός - μία φωτορεαλιστική απεικόνιση στην οθόνη της 3Δ σκηνής. Αλλά τι θα γίνει όταν ο χρήστης θα θελήσει να μετακινήσει ένα από τα αντικείμενα σε μία διαφορετική θέση και ίσως να δει ένα άλλο αντικείμενο από διαφορετική γωνία; Τι θα γίνει όταν ο χρήστης θελήσει να πειραματιστεί με μία διαφορετική ποικιλία χρωμάτων για τα αντικείμενα και για το φόντο της σκηνής; Αν το αντικείμενο υφίσταται μόνο σαν ένα σχέδιο στην οθόνη, ο προγραμματιστής πρέπει να γράψει ένα πεπλεγμένο κώδικα για να υλοποιήσει τις πράξεις αυτές. Με τον Open Inventor, η δυνατότητα να υλοποιηθούν οι αλλαγές αυτές ενυπάρχουν στο προγραμματιστικό μοντέλο. Η αλλαγή των αντικειμένων της σκηνής, η προσθήκη αντικειμένων και η διάδραση με αυτά είναι μία από τις δυνατότητες και τις βασικές προδιαγραφές του λογισμικού αυτού.

Επειδή οι πληροφορίες που τηρούνται στη βάση δεδομένων του Open Inventor έχουν σχέση με την 3Δ ύπαρξή τους στο χώρο και όχι με μία απλή απεικόνισή τους στην οθόνη σαν εικονοστοιχεία (pixels), μπορούν να υλοποιηθούν επιπρόσθετες λειτουργίες από μία απλή αναπαράσταση. Συνεπώς μπορούμε να επιλέξουμε (pick) τα αντικείμενα στην οθόνη, να τα τονίσουμε περιμετρικά (highlight) και γενικά να τα διαχειριστούμε σαν διακριτές οντότητες. Μπορούμε επίσης να εκτελέσουμε υπολογισμούς περιβάλλουσας επιφάνειας (bounding box), να τα εκτυπώσουμε, να τα αποθηκεύσουμε σε ένα φάκελο ή να τα διαβάσουμε από ένα φάκελο.

### 2.2.1.2 Open Inventor και OpenGL

Ο Open Inventor χρησιμοποιεί τη μη αντικειμενοστραφή βιβλιοθήκη γραφικών OpenGL για την απεικόνιση στην οθόνη. Όμως στον Open Inventor η απεικόνιση μαζί με άλλες λειτουργίες όπως επιλογή, ανάγνωση, εγγραφή και υπολογισμοί περιβάλλοντος κουτιού, εμπεριέχονται σε κλάσεις και αντικείμενα. Υπάρχουν βέβαια και άλλες διαφορές που βασίζονται κυρίως στο ότι ο Open Inventor είναι μία αντικειμενοστραφής έκδοση γραφικών που αν και βασίζεται στην OpenGL, εν τούτοις παρέχει πρόσβαση σε αντικείμενα και λειτουργίες τους που χαρακτηρίζονται υψηλότερου επιπέδου από αυτήν καθαυτή την OpenGL. Έτσι ο Open Inventor δεν έχει πρόσβαση στην εκάστοτε μνήμη αναπαράστασης (frame buffer) αλλά απλά παριστάνει στην οθόνη τα αντικείμενα μόνο όταν του ζητηθεί κατά την ροή του προγράμματος.

## 2.3 Υλοποίηση νόμων Φυσικής

Στο σημείο αυτό θα πρέπει να σημειώσουμε ότι δεν θα παρουσιάσουμε κάποια βιβλιοθήκη χρήσης νόμων φυσικής (κίνηση, επιτάχυνση, ταχύτητα ή βαρύτητα) αλλά ένα σύνολο αλγορίθμων που έχουν μία καλά καθορισμένη φυσική σημασία και παρουσίαση στην οθόνη.

### 2.3.1 Ανίχνευση Σύγκρουσης ή Παρεμβολής

Υπάρχει ένα εκτεταμένο ερευνητικό έργο που αντιμετωπίζει αποτελεσματικά το πρόβλημα της ανίχνευσης σύγκρουσης κινούμενων πολυεδρικών αντικειμένων (βλ. [11], [23]).

Η κλάση `SoRayPickAction` επιλέγει αντικείμενα δημιουργώντας μία ακτίδα με αρχή το βασικό αντικείμενο και κατεύθυνση τη σκηνή και ελέγχοντας τη διασταύρωσή της με όλα τα αντικείμενα. Η ακτίδα αυτή μπορεί να πάρει και τη μορφή 3Δ αντικειμένου όταν πρέπει να ελέγξει τη διασταύρωσή της με 2Δ αντικείμενο. Κάθε αντικείμενο που διασταυρώνεται επιστρέφεται σαν δείγμα κλάσης `SoPickedPoint`.

Η ακτίδα επιλογής μπορεί να καθορισθεί στο χώρο είτε ως έχουσα αρχή τη θέση της κάμερας διαμέσου ενός εικονοστοιχείου του οπτικοποιημένου χώρου είτε ως έχουσα γεωμε-



τρικές συντεταγμένες στο χώρο.

Όταν καλούμε τη συγκεκριμένη κλάση, μπορούμε να επιλέξουμε αν θα υπολογίσει όλες τις διασταυρώσεις κατά μήκος της ακτίδας (κατατάσσοντας τα αντικείμενα από το εγγύτερο προς το απώτερο) ρυθμίζοντας κατάλληλα μία μεταβλητή της κλάσης. Εξ' ορισμού η κλάση υπολογίζει μόνο την εγγύτερη διασταύρωση αλλά μπορεί να επιστρέψει μία λίστα με όλα τα αντικείμενα που διασταυρώνει. Επίσης διαθέτει μεθόδους για την εύρεση του σημείου διασταύρωσης άνω στην επιφάνεια του αντικειμένου ή και του ορθοκανονικού διανύσματος στο σημείο αυτό.

## 2.4 Μοντελοποίηση πλήθους

Η μοντελοποίηση του πλήθους είναι το κύριο μέρος του VRkernel και του VRsystem και στη περίπτωση της εφαρμογής της εγκατάλειψης ενός πλοίου, η μοντελοποίηση αυτή είναι το πιο σημαντικό από τα υπόλοιπα συστατικά του. Στο σημείο αυτό θα αρκестούμε να αναλύσουμε τον τρόπο με τον οποίο αυτή ενσωματώνεται στο VRkernel και πως χρησιμοποιείται από το VRsystem.

Η μοντελοποίηση του πλήθους βασίζεται σε πράκτορες, στα αντικείμενα της σκηνής (όπως π.χ.εμπόδια) και στη τεχνολογία των καθοδηγητικών συμπεριφορών. Η τεχνολογία των καθοδηγητικών συμπεριφορών είναι ο πυρήνας της μοντελοποίησης πλήθους που χρησιμοποιείται στο VRkernel και θα αναφερθούμε με λεπτομέρεια στο κεφάλαιο 4. Οι πράκτορες που χρησιμοποιούνται στο VRkernel είναι στη πραγματικότητα *Αυτόνομοι Πράκτορες (Autonomous Agents)*, στους οποίους έχουμε αναφερθεί αρχικά στη παράγραφο 1.2.2 και θα αναφερθούμε με περισσότερη λεπτομέρεια στη παράγραφο 2.4.3.<sup>5</sup>

Πριν προχωρήσουμε στην παρουσίαση των επιμέρους συστατικών της μοντελοποίησης του πλήθους, θα αναφερθούμε σε γενικές γραμμές στη δομή του προγραμματιστικού κώδικα και συγκεκριμένα θα σκιαγραφήσουμε τις κλάσεις που χρησιμοποιούμε και την ιεράρχησή τους<sup>6</sup>. Στο σχήμα 2.2 έχουμε παραθέσει ένα απόσπασμα του συνολικού διαγράμματος ιερα-

<sup>5</sup>Στην εργασία αυτή, αν και ο όρος είναι πιο γενικός, όπου αναφέρουμε την έννοια *πράκτορας* θα εννοούμε *Αυτόνομος Πράκτορας* εκτός αν αναφέρεται διαφορετικά.

<sup>6</sup>Η εφαρμογή είναι εξολοκλήρου αντικειμενοστραφής και σε γλώσσα προγραμματισμού C++ με ευρεία

χίας των κλάσεων. Στο σχήμα αυτό μπορούμε να ξεχωρίσουμε τρεις επιμέρους ιεραρχίες:

1. Η πράσινη διακεκομμένη γραμμή περιέχει κλάσεις που ανήκουν στην ιεραρχία των αντικειμένων της σκηνής (περιλαμβανομένου των πρακτόρων και των εμποδίων).
2. Η γαλάζια γραμμή είναι για την ιεραρχία της μοντελοποίησης του νου.
3. Η μοβ γραμμή περιλαμβάνει τις κλάσεις των καθοδηγητικών συμπεριφορών.

Η πατρική κλάση για όλα τα αντικείμενα της σκηνής είναι η `VR_Geometry`, η οποία εφαρμόζει όλες τις παραμέτρους και δυνατότητες που χρειάζεται μία γεωμετρική οντότητα για να αναπαρασταθεί σε ένα εικονικό περιβάλλον. Θα πρέπει να αναφέρουμε εδώ ότι η `VR_Geometry` διαθέτει ένα κόμβο σχήματος (shape node) του Open Inventor, κόμβους για υλικά επικάλυψης και υφής (textures) και όλες τις απαραίτητες δομές που απαιτούνται για τον έλεγχο της θέσης και των γεωμετρικών μετασχηματισμών της όπως αναλογική παραμόρφωση (scaling) και περιστροφή (rotation). Η παράμετρος `mind` που διαθέτει η `VR_Geometry` χρησιμοποιείται σε ορισμένα αντικείμενα της σκηνής που διαθέτουν ικανότητες Τεχνητής Νοημοσύνης (Artificial Intelligence, AI), όπως είναι πράκτορες της κλάσης `VR_SimpleVehicle` και οι κληρονόμοι τους<sup>7</sup>. Για όλα τα άλλα αντικείμενα της σκηνής, το `mind` είναι `null`. Εκτός από τους πράκτορες και τα αντικείμενα, που εκπροσωπούνται από τις κλάσεις `VR_SimpleVehicle` και `VR_Obstacle` αντίστοιχα, υφίσταται μία άλλη σημαντική κλάση, η `VR_SceneLevel` που είναι κληρονόμος της κλάσης `VR_Geometry` και μαζί με τους διαδόχους της χρησιμοποιούνται για να ορίσουν την εικονική επιφάνεια τους εδάφους (terrain).

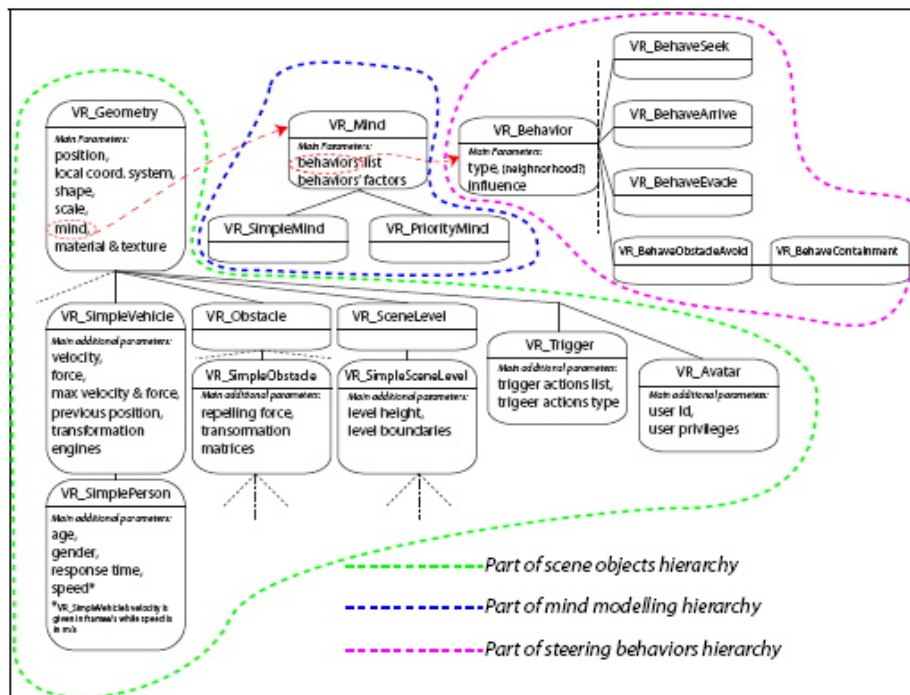
Η ιεραρχία της μοντελοποίησης του νου διαθέτει μία εικονική κλάση, τη `VR_Mind`, η οποία είναι και η πατρική κλάση για όλες τις εφαρμογές του νου. Οι κύριες παράμετροι της `VR_Mind` είναι μία λίστα συμπεριφορών και οι αντίστοιχοι συντελεστές επίδρασης.

Η πατρική κλάση των κλάσεων που αντιπροσωπεύουν τις καθοδηγητικές συμπεριφορές είναι η `VR_Behavior` που διαθέτει κάποιες απλές παραμέτρους ελέγχου. Όλες οι κλάσεις των καθοδηγητικών συμπεριφορών είναι κληρονόμοι της `VR_Behavior` ενώ κάποιες πιο πεπλεγμένες κληρονομούν τη δομή τους από πιο απλές συμπεριφορές (π.χ. η συμπεριφορά

---

χρήση της βιβλιοθήκης δημιουργίας παραθύρων Microsoft Foundation Classes Library (MFC Library)

<sup>7</sup>όρος του αντικειμενοστραφούς προγραμματισμού ως μετάφραση του αγγλικού όρου inheritance



Σχήμα 2.2: Ιεραρχία Κλάσεων του VRkernel στα Αντικείμενα Σκηνης, Μοντελοποίηση Νου, και Καθοδηγητικών Συμπεριφορών

"Περιστολή" (Containment) που εκπροσωπείται από την κλάση VR\_BehaveContainment είναι διάδοχος της κλάσης VR\_BehaveObstacleAvoid που εκπροσωπεί την αποφυγή εμποδίου).

### 2.4.1 Πράκτορες

Η τεχνολογία των αυτόνομων πρακτόρων υλοποιείται στη κλάση VR\_Simple Vehicle του VRkernel. Προφανώς οι πράκτορες κάνουν χρήση των χαρακτηριστικών και της δομής της παραμέτρου mind της κλάσης VR\_Geometry και έτσι διαθέτουν μία ή περισσότερες καθοδηγητικές συμπεριφορές. Βέβαια εκτός από αυτό, υφίσταται ειδικές παράμετροι των πρακτόρων που ελέγχουν τη ταχύτητα και τη δύναμη απόκρισης τους. Αλλά υπάρχουν και αντίστοιχα άνω όρια στην επιλογή των παραμέτρων αυτών που δίνονται από τις παραμέτρους max\_speed και max\_force.

Ουσιαστικά, η μοντελοποίηση του πλήθους υλοποιείται με την παραγωγή αντιγράφων ενός πράκτορα, τα οποία ως αντικείμενα της σκηνης είναι εξοπλισμένα με τις κατάλληλες

καθοδηγητικές συμπεριφορές που ελέγχονται και συνδυάζονται με την ιδιαίτερη λογική (τεχνητή νοημοσύνη) που διαθέτει και μπορεί να κινείται, συμπεριφέρεται και δρα στο εικονικό περιβάλλον του VRkernel με ένα όσο το δυνατό φυσιολογικό ή/και ανθρώπινο τρόπο.

### 2.4.2 Εμπόδια

Η κλάση `VR_Obstacle` είναι μία άλλη κλάση που χρησιμοποιείται στο VRkernel μαζί με την κλάση `VR_SimpleVehicle` για την αναπαράσταση της σκηνής. Κάθε αντικείμενο της σκηνής που δεν είναι πράκτορας ή δεν ανήκει σε κάποια ειδική κλάση, είναι ένα αντίγραφο της κλάσης `VR_Obstacle` ή της υποκλάσης `VR_SimpleObstacle`.

Τα αντίγραφα της κλάσης `VR_Obstacle` μπορεί να είναι απλά εμπόδια της σκηνής και να μην διαθέτουν τη "νοημοσύνη" των πρακτόρων της σκηνής, αλλά όμως διαθέτουν παραμέτρους και λειτουργίες που τα κάνουν να διαφέρουν από τυχόν απλές αναπαραστάσεις στην οθόνη. Επειδή είναι επίσης και αντίγραφα της κλάσης `VR_Geometry`, είναι εφοδιασμένα και με δυνατότητες επικάλυψης τους με ψηφιακές επιδερμίδες ώστε να προσομοιάζουν διάφορα υλικά. Διαθέτουν επίσης ιδιαίτερες δυνατότητες που τους επιτρέπουν να τα σπρώχνουμε ή να τα ανοίγουμε, όπως είναι μία πόρτα ή και παραμέτρων που αντιπροσωπεύουν δυνάμεις αντίδρασης σε σύγκρουση.

### 2.4.3 Καθοδηγητικές Συμπεριφορές

Στο σημείο αυτό και για λόγους πληρότητας του θέματος της προσομοίωσης της κίνησης του πλήθους, θα προβούμε σε μία συνοπτική περιγραφή των καθοδηγητικών συμπεριφορών αν και θα αναφερθούμε πιο εμπειριστατωμένα στο κεφάλαιο 4.

Οι καθοδηγητικές συμπεριφορές είναι η κινητήριος δύναμη των αυτόνομων χαρακτήρων και τους παρέχει τη δυνατότητα να κινούνται στο συνθετικό τους χώρο με ένα τρόπο που προσομοιάζει την ανθρώπινη κίνηση. Ο αυτόνομος χαρακτήρας που χρησιμοποιείται στο VRkernel βασίζεται στη τεχνολογία των πρακτόρων και είναι ο τύπος του χαρακτήρα που συνήθως χρησιμοποιείται στην υπολογιστική αναπαράσταση και στα διαδραστικά μέσα ενός περιβάλλοντος εικονικής πραγματικότητας. Ένας αυτόνομος χαρακτήρας πρέπει να συνδυάζει

χαρακτηριστικά ενός αυτόνομου μηχανισμού (όπως ένα ρομπότ) με κάποιες δυνατότητες μίας ανθρώπινης συμπεριφοράς. Αυτοί είναι και οι λόγοι που χρησιμοποιούμε τον όρο *συμπεριφορά* όταν αναφερόμαστε στο τρόπο κίνησης των αυτόνομων χαρακτήρων στο περιβάλλον εικονικής πραγματικότητας. Η κάθε συμπεριφορά γίνεται περισσότερο αντιληπτή με τη διαίρεσή της σε τρία θεωρητικά ανεξάρτητα μεταξύ τους επίπεδα λήψης απόφασης:

1. **Επιλογή Δράσης:** εδώ τίθενται οι στόχοι και γίνονται οι σχεδιασμοί για την υλοποίηση της δράσης.
2. **Καθοδήγηση:** εδώ καθορίζεται ο τελικός διάδρομος κίνησης.
3. **Κίνηση:** εδώ γίνεται πράξη η αρθρωτή κίνηση και οι λεπτομέρειες της ψηφιακής αναπαράστασης της κίνησης.

Οι καθοδηγητικές συμπεριφορές καλύπτουν το μεσαίο επίπεδο λήψης απόφασης αν και το πρώτο και τρίτο επίπεδο προσδίδουν τον χαρακτήρα της δημιουργικότητας της πορείας του στον αυτόνομο πράκτορα. Μία ανάλογη περίπτωση παρμένη από την πραγματικότητα είναι η εξής: Ας θεωρήσουμε ένα τροχονόμο που ελέγχει τη κυκλοφορία σε ένα δρόμο. Λόγω της κυκλοφοριακής συμφόρησης στο δρόμο αυτό, ο τροχονόμος υποχρεώνει τα αυτοκίνητα να χρησιμοποιήσουν παρακαμπτήριους ή εναλλακτικούς δρόμους. Ο τροχονόμος υποχρεώνει έτσι έναν οδηγό να ακολουθήσει μία αριστερή στροφή σε ένα περιφερειακό δρόμο. Ο οδηγός υπακούει, βάζει πρώτη ταχύτητα, επιταχύνει και στρίβει το τιμόνι αριστερά οδηγώντας το αυτοκίνητό του στο περιφερειακό δρόμο. Στο παράδειγμα αυτό, ο τροχονόμος αντιπροσωπεύει την επιλογή δράσης: παρατηρεί ότι η κατάσταση του δρόμου έχει αλλάξει, θέτει στόχους για τον έλεγχο της κυκλοφορίας. Το καθοδηγητικό επίπεδο, εκπροσωπείται από τον οδηγό που υλοποιεί τον στόχο (οδηγεί προς το περιφερειακό δρόμο) μέσω μίας σειράς από μικρούς στόχους που ολοκληρώνουν τη προσπάθεια (εμπλοκή ταχύτητας, επιτάχυνση, στρέψη τιμονιού, οδήγηση, αποφυγή άλλων αυτοκινήτων, τήρηση ορίου ταχύτητας κλπ). Χρησιμοποιεί διάφορα μέσα (τιμόνι, πεντάλ επιτάχυνσης, φρένο) και οδηγεί το αυτοκίνητό του προς το στόχο που έχει οριστεί από το επίπεδο επιλογής. Το αυτοκίνητο υλοποιεί το τελικό επίπεδο κίνησης (η μηχανή θέτει σε κίνηση τους τροχούς και κινείται το αυτοκίνητο).

Στο παράδειγμα αυτό, οι μικροί στόχοι που ορίζονται από τον οδηγό είναι πράξεις που παρόλο που είναι απλές στη περιγραφή και στην εφαρμογή, μπορεί όταν συνδυασθούν να δημιουργήσουν σύνθετες συμπεριφορές όπως την οδήγηση του αυτοκινήτου. Αυτοί οι μικροί στόχοι και η ανάμιξή τους που δημιουργεί μία σύνθετη τεχνητή νοημοσύνη είναι το κεντρικό ζήτημα της τεχνολογίας των καθοδηγητικών συμπεριφορών. Πριν αρχίσουμε την αναφορά μας σε κάποιες από τις καθοδηγητικές συμπεριφορές, θα πρέπει να ορίσουμε κάποιες από τις αφηρημένες έννοιες που τις συνθέτουν.

1. **Όχημα (Vehicle)**: Είναι η αφηρημένη αναπαράσταση του κινούμενου αντικειμένου. Περιέχει πληροφορίες για τη θέση του, τη πορεία, τη ταχύτητά του κλπ.
2. **Νους (Mind)**: Ο κώδικας της τεχνητής νοημοσύνης που αποφασίζει για τους ορισμούς των μικρών στόχων και τις εσωτερικές διαδικασίες που εκτελούνται για την υλοποίηση των επιλεγέντων στόχων.
3. **Συμπεριφορά (Behavior)**: Ο κώδικας που υλοποιεί τις αποφάσεις του Νου και τελικά δημιουργεί το τελικό άνυσμα κίνησης.

Στην εργασία αυτή, υποθέτουμε ότι η συνεχής κίνηση υλοποιείται από ένα απλό μοντέλο οχήματος. Το μοντέλο παραμετροποιείται από ένα απλό διάνυσμα καθοδηγητικής δύναμης και είναι το τελικό εξαγόμενο του συνδυασμού των ανεξάρτητων διανυσμάτων των καθοδηγητικών δυνάμεων των ανεξάρτητων συμπεριφορών του μηχανισμού του νου του. Παραδείγματα καθοδηγητικών συμπεριφορών είναι:

- Αναζήτηση (Seek) και Αποφυγή (Flee).
- Καταδίωξη (Pursuit) και Διαφυγή (Evasion).
- Άφιξη (Arrival).
- Έκκεντρη Καταδίωξη (Offset Pursuit) και Σχηματισμοί (Formations).
- Αποφυγή Εμποδίων (Obstacle Avoidance).

- Περιπλάνηση (Wander) και Εξερεύνηση (Explore).
- Ακολουθία Διαδρόμου (Path Following).
- Ακολουθία Πεδίου Ροής (Flow Field Following).
- Μη Ευθυγραμμισμένη Αποφυγή Σύγκρουσης (Unaligned Collision Avoidance).
- Συγκροτημένη Κίνηση (Cohesion) και Ευθυγραμμισμένη Κίνηση (Alignment).
- Ακολουθία Αρχηγού (Leader Following).

#### 2.4.4 Αυτόνομη Νοημοσύνη Πρακτόρων

Το τελευταίο θέμα που θα μας απασχολήσει και είναι άμεσα συνδεδεμένο με τη θεωρία του Νου ή εναλλακτικά με το συστατικό του πράκτορα που αναφέρεται στη Τεχνητή Νοημοσύνη, είναι ο τρόπος ανάμιξης των συμπεριφορών και του συνδυασμού τους έτσι ώστε να μπορεί να προκύπτει μία πιο σύνθετη αλλά και πιο φυσιολογική συμπεριφορά για τον αυτόνομο χαρακτήρα μας.

Ο πιο απλός τρόπος ανάμιξης συμπεριφορών είναι ένας απλός *μεσοβαρικός (weighted average)* όπου όλες οι συμπεριφορές συνεισφέρουν συνεχώς στην τελική συμπεριφορά ανάλογα με το συγκεκριμένο ειδικό βάρος τους. Αν και ο τρόπος αυτός είναι πολύ απλός στη σύλληψη και εφαρμογή, εντούτοις αναλώνει υπολογιστική ισχύ αφού οι απαιτούμενοι υπολογισμοί πρέπει να γίνονται για κάθε στιγμιότυπο, κάθε καθοδηγητικό διάνυσμα που παράγεται από κάθε επιμέρους συμπεριφορά και για όλους τους αυτόνομους χαρακτήρες. Επίσης είναι φανερό ότι η διαδικασία ζύγισης - σημασίας της κάθε συμπεριφοράς αλλά και το αποτέλεσμα της μίξης μπορεί να μην αποκρίνονται στα αναμενόμενα αποτελέσματα. Αυτό σημαίνει ότι η διαδικασία της ανάθεσης βαρών μπορεί τελικά να αποδειχθεί πολύ επίπονη καθώς περιλαμβάνει διορθώσεις με τη μέθοδο "Δοκιμή - Λάθος" ώστε να συσχετισθούν τελικά τα βάρη της κάθε συμπεριφοράς με το τελικό επιθυμητό εξαγόμενο.

Μία άλλη πιθανή προσέγγιση είναι η *Ανάμιξη Προτεραιότητας (prioritized dithering)* που προτείνεται και από τον Reynolds στο [29] όπου σε γενικές γραμμές ο αλγόριθμος που εφαρμόζεται περιγράφεται με τα παρακάτω βήματα:

*Υλοποίηση Καθοδηγητικών Συμπεριφορών*

Βήμα 1ο: δημιουργία τυχαίου αριθμού  $r$ .

Βήμα 2ο: εάν το  $r$  είναι μεγαλύτερο από ένα κατώφλι,

εκτέλεση της συμπεριφοράς με τη μεγαλύτερη προτεραιότητα  
(συνήθως συμπεριφορές αποφυγής) και άφησε τον αλγόριθμο

αλλιώς

Βήμα 3ο: εάν η προηγούμενη συμπεριφορά δεν ήταν ενεργή

(δηλ. δεν υπάρχει τίποτα να αποφύγει)

εκτέλεσε την επόμενη συμπεριφορά στην ουρά προτεραιότητας

Βήμα 4ο: επανέλαβε το βήμα 3

μέχρις ότου εκτελεσθούν όλες οι συμπεριφορές στην ουρά ή

μία συμπεριφορά βρεθεί ενεργή

Σε αυτό το παράδειγμα ψευδοκώδικα τυχαιοποιείται μόνο η πρώτη συμπεριφορά αλλά θα μπορούσαμε κάλλιστα να τυχαιοποιήσουμε περισσότερες συμπεριφορές.

Προφανώς αυτές οι δύο προσεγγίσεις δεν είναι και οι μοναδικές δυνατές επιλογές του νου αλλά μας βοηθούν να κατανοήσουμε τη λειτουργία του νου. Μία πιο ενδελεχής περιγραφή της μοντελοποίησης του νου και των συνδυασμών των συμπεριφορών παρουσιάζεται στη παράγραφο 2.5.1.

## 2.5 Προσομοιώσεις VRKernel

Ο VRKernel είναι εξοπλισμένος με μία ιεραρχία προσομοίωσης με τις αντίστοιχες κλάσεις ελέγχου που επιτρέπουν τον ανεξάρτητο αλλά και γενικό έλεγχο όλων των επιμέρους προσομοιώσεων και των διασυνδέσεών τους που εκτελούνται στο εσωτερικό του.

Η βασική κλάση της ιεραρχίας των προσομοιώσεων είναι η `VR_Simulation` η οποία και ορίζει ένα κοινό πρωτόκολλο για όλες τις παραγόμενες από αυτή κλάσεις. Με σκοπό την ευελιξία του συστήματος, η βασική κλάση έχει εξοπλισθεί με δύο διατάξεις για τις προ- (pre-)



και μετά- (post-) προσομοιώσεις.

Μία προ-προσομοίωση καλείται πριν την έναρξη της προσομοίωσης. Αυτή η τεχνική χρησιμοποιείται στις περιπτώσεις όπου πρέπει να αρχικοποιηθούν κάποια δεδομένα πριν οποιοδήποτε στοιχείο του VRKernel, όπως μία συμπεριφορά, ενεργοποιηθεί και προσπαθήσει να αποκτήσει πρόσβαση σε αυτά. Ένα παράδειγμα προ-προσομοίωσης είναι η κλάση *VRNeighborhood* που πρέπει να αρχικοποιήσει τον πίνακα αποστάσεων της πριν μπορέσουμε να την χρησιμοποιήσουμε. Η κλάση *VRNeighborhood* είναι ένα αντικείμενο της προσομοίωσης του VRKernel που παρέχει τον ορισμό της περιοχής της γειτονίας ενός οχήματος αλλά και τις απαραίτητες δοσοληψίες και θα μπορέσει να χρησιμοποιηθεί σε συμπεριφορές όπως "Ευθυγράμμιση", "Συνοχή" και "Διαχωρισμός".

Μία μετά-προσομοίωση χρησιμοποιείται σε διαδικασίες εκκαθαρισμού της μνήμης αφού έχει ολοκληρωθεί η προσομοίωση. Ένα παράδειγμα μπορεί να είναι το στοιχείο της διεπιφάνειας που εμφανίζει τις δυναμικές παραμέτρους της κάθε συμπεριφοράς.

Συνοπτικά, τα βήματα ενός κύκλου προσομοίωσης είναι τα ακόλουθα:

Βήμα 1ο: το ρολόι της προσομοίωσης αρχίζει να μετρά το χρόνο  $t$  της προσομοίωσης,

Βήμα 2ο: αρχίζει ο κύκλος της προσομοίωσης πλήθους,

Βήμα 3ο: τελειώνει ο κύκλος της προσομοίωσης πλήθους.

### 2.5.1 Προσομοίωση Κίνησης Πλήθους

Αφού τελειώσει η διάταξη των αντικειμένων της προσομοίωσης (πράκτορες, εμπόδια κ.α.) και η απαραίτητη ρύθμιση των παραμέτρων του συστήματος, είναι έτοιμος να εκτελεσθεί ο κύκλος της προσομοίωσης. Παρακάτω μπορούμε να παραθέσουμε έναν τυπικό κύκλο εκτέλεσης μίας προσομοίωσης του πλήθους στο VRKernel και συνεπώς στο VRSystem.

Τα κυριότερα βήματα στην προσομοίωση του πλήθους είναι τα ακόλουθα:

Βήμα 1ο: Έναρξη της προσομοίωσης κίνησης πλήθους,

Βήμα 2ο: Εκτέλεση όλων των προ-προσομοιώσεων (όπως είναι η κατασκευή γειτονιών για

*Υλοποίηση Καθοδηγητικών Συμπεριφορών*

τα οχήματα),

Βήμα 3ο: Ενημέρωση της θέσης κάθε οχήματος,

Βήμα 4ο: Για κάθε πράκτορα της προσομοίωσης:

- Αποθηκεύεται η παρούσα θέση και ταχύτητα στο τοπικό σύστημα συντεταγμένων
- Οι πράκτορες που είναι εξοπλισμένοι με νου, προχωρούν στο υπολογισμό των καθοδηγητικών συμπεριφορών
- Αναλόγως ανανεώνεται η θέση, η ταχύτητα και το τοπικό σύστημα συντεταγμένων του κάθε πράκτορα

Βήμα 5ο: Εκτελούνται όλες οι μετά-προσομοιώσεις,

Θα πρέπει να εξετάσουμε το βήμα 4 με μεγαλύτερη λεπτομέρεια:

Προφανώς ο τύπος του νου που είναι προσαρμοσμένος σε κάθε πράκτορα μεταβάλλει τον τρόπο υπολογισμού των καθοδηγητικών συμπεριφορών και τις κάνει έτσι ιδιαίτερες για τον κάθε ένα. Έτσι σε ένα υπολογισμό που ο νους είναι απλός, η κάθε συμπεριφορά υπολογίζεται ανεξάρτητα και το τελικό διάνυσμα κίνησης εξάγεται με τη συνολική διανυσματική πρόσθεση των επιμέρους υπολογισμένων σύμφωνα πάντα με τους επιμέρους συντελεστές.

Έτσι το βήμα 4 μπορεί να αναλυθεί λεπτομερώς ως εξής για κάθε πράκτορα:

1. Αποθήκευση της κατάστασης του πράκτορα. Αυτό το στάδιο μπορεί να χρησιμοποιηθεί για την εξομάλυνση συμπεριφορών όπως η "Διαφυγή" και η "Καταδίωξη":
  - Αποθήκευση της παρούσας θέσης ως `previous_position`.
  - Αποθήκευση της παρούσας ταχύτητας ως `previous_velocity`
  - Αποθήκευση του παρόντος τοπικού συστήματος συντεταγμένων ως `previous_lcs`
2. Ανάλογα με τον τύπο του νου του κάθε πράκτορα, ο τρόπος υπολογισμού θα καλείται:
  - Περίπτωση Απλού Νου:

- Υπολογίζουμε για κάθε συμπεριφορά ένα καθοδηγητικό διάνυσμα και το αναγάγουμε αναλόγως με το συντελεστή επιμέρισης.
  - Προσθέτουμε διανυσματικά όλα τα διανύσματα καθοδήγησης.
- Περίπτωση Νου Προτεραιότητας:
    - Ο υπολογισμός των συμπεριφορών γίνεται ανάλογα με τη θέση που έχει η κάθε συμπεριφορά στη λίστα προτεραιότητας και αναλόγως γίνεται και ο υπολογισμός του τελικού διανύσματος.
    - Εάν το τελικό διάνυσμα ξεπερνά την μέγιστη δυνατή τιμή του, τότε ο υπολογισμός σταματά εδώ και αδιαφορούμε για τις υπόλοιπες συμπεριφορές.
3. Περικόπτουμε εάν χρειάζεται το μέγεθος του διανύσματος στο μέγιστο δυνατό.
  4. Προσθέτουμε το τελικό διάνυσμα στην παρούσα ταχύτητα και περικόπτουμε ανάλογα το μέγεθος στο μέγιστο δυνατό.
  5. Ανανέωση της θέσης, της ταχύτητας και του τοπικού συστήματος συντεταγμένων.

### 2.5.2 Εισαγωγή Στοιχείων Προσομοίωσης

Η εισαγωγή των στοιχείων της προσομοίωσης στο συγκεκριμένο περιβάλλον χρήστη γίνεται με δύο τρόπους:

- Εισαγωγή αρχείου τύπου XML όπου περιγράφονται σε κώδικα, που γίνεται αντιληπτός από το κέλυφος χειρισμού του πυρήνα, οι τυχόντες πράκτορες και οι καθοδηγητικές συμπεριφορές με τα χαρακτηριστικά που διαθέτουν. Η έναρξη της προσομοίωσης είναι δυνατή με την επιλογή του κατάλληλου χειριστηρίου.
- Χρήση του γραφικού περιβάλλοντος και ιδιαίτερας της προβολής δένδρου Tree-view των στοιχείων της προσομοίωσης για την σταδιακή εισαγωγή των πρακτόρων και των χαρακτηριστικών τους.

## 2.6 Στόχοι της Εργασίας

Η παρούσα εργασία θα εστιασθεί στην υλοποίηση των παρακάτω συμπεριφορών καθώς και στη παραμετροποίησή τους:

- Αποφυγή (Flee)
- Καταδίωξη (Pursuit)
- Διαφυγή (Evasion)
- Αποφυγή Εμποδίων (Obstacle Avoidance)
- Ακολουθία Διαδρόμου (Path-Following)
- Ευθυγράμμιση (Alignment)
- Άφιξη (Arrive)

### 2.6.1 Σημεία Έμφασης

Οι καθοδηγητικές συμπεριφορές βασίζονται σε κάποιες προδιαγραφές που είναι σχεδόν κοινές σε όλες τις προσπάθειες αναπαράστασης μίας εικονικής πραγματικότητας:

- Αληθοφάνεια της κίνησης. Κατά την εναλλαγή των διαδοχικών στιγμιότυπων απεικόνισης παρατηρείται συχνά μία απότομη μεταπήδηση των αντικειμένων. Σύμφωνα με τις ελάχιστες απαιτήσεις παραπλάνησης του ανθρώπινου εγκεφάλου, η ελάχιστη ταχύτητα εναλλαγής στιγμιότυπων είναι 25 frames per second και αυτός ο περιορισμός θα καταβληθεί προσπάθεια να διατηρηθεί στα πλαίσια εφαρμογής του λογισμικού σε ένα μέτριο δυνατοτήτων υπολογιστικό σύστημα.
- Διατήρηση της φυσικότητας στις κινήσεις. Η κίνηση των πρακτόρων θα πρέπει να διέπεται από τους ανάλογους φυσικούς νόμους ώστε να προσεγγίζεται με τον καλύτερο τρόπο η πραγματικότητα. Συνεπώς, η υλοποίηση θα βασισθεί στη θεωρία των διανυσματικής ανάλυσης που παρέχει την επαρκή ακρίβεια της κίνησης

των σωμάτων. Βέβαια, όπου είναι απαραίτητη η οποιαδήποτε ψηφιοποίηση (απεικόνιση αντικειμένων σε εσχάρα) αυτή θα γίνεται με τον βέλτιστο τρόπο που δεν θα παραβιάζει την ελάχιστη απαίτηση αληθοφάνειας της εφαρμογής.

- Όλες οι καθοδηγητικές συμπεριφορές που θα υλοποιηθούν θα έχουν ως βασικό συστατικό ένα συντελεστή ανάμιξης που θα βασίζεται στην αρχή της αναλογικότητας (το μέγεθος των αποτελεσμάτων θα είναι αναλογικό με την αύξησή του). Ο συντελεστής αυτός θα δίνει τη δυνατότητα ανάμιξης των συμπεριφορών σε μία περίπτωση προσομοίωσης ενός αυτόνομου πράκτορα με πολυσύνθετο τρόπο αντίδρασης ανάλογο ενός νοήμονα ανθρώπου.

### 2.6.2 Σημεία Πρωτοτυπίας

Οι καθοδηγητικές συμπεριφορές που θα υλοποιηθούν δίνουν ορισμένες δυνατότητες πρωτοτυπίας που αν και περιορισμένες μπορούν να χαρακτηρίσουν την εκπόνηση της παρούσας διπλωματικής εργασίας όπως είναι οι εξής:

- Προσπάθεια ομαλοποίησης της κίνησης ενός πράκτορα κατά τη διάρκεια της αποφυγής ενός εμποδίου. Η κίνηση αποφυγής θα πρέπει να λαμβάνει υπόψη ορισμένους περιορισμούς και προϋποθέσεις όπως είναι η απόσταση ενεργοποίησης του αλγόριθμου, η ελάχιστη απόσταση αποφυγής του εμποδίου, η ταχύτητα κίνησης του πράκτορα, η έκταση του εμποδίου και η έκταση του πράκτορα. Θα καταβληθεί προσπάθεια ώστε υπό ορισμένες απλουστεύσεις και παραδοχές που δεν βλάψουν την γενικότητα να αντιμετωπισθεί ένα βασικό πρόβλημα αναπαράστασης στο πεδίο της εικονικής πραγματικότητας
- Ενσωμάτωση αλγορίθμων εξεύρεσης βέλτιστων διαδρομών στον πυρήνα της εικονικής πραγματικότητας και στα πλαίσια της υλοποίησης της συμπεριφοράς "Ακολουθία Διαδρόμου". Βέβαια, στο επίπεδο μίας διπλωματικής εργασίας η εμβάθυνση θα είναι περιορισμένη όμως θα δίνει δυνατότητες για περαιτέρω συνέχεια της έρευνας.

□



## Κεφάλαιο 3

# Αλγόριθμοι Εύρεσης Διαδρομών

### 3.1 Εισαγωγή - Ορισμός

Κατά τη σχεδίαση και εκτέλεση των κυριότερων καθοδηγητικών συμπεριφορών ανακύπτει ένα βασικό πρόβλημα που στη περίπτωση της ανθρώπινης συμπεριφοράς είναι αυτονόητο: *Ποια πορεία πρέπει να ακολουθήσει ένας αυτόνομος πράκτορας κατά την κίνησή του προς κάποιο συγκεκριμένο προορισμό;* . Το πρόβλημα γίνεται δυσκολότερο όταν ο πράκτορας δεν έχει αντίληψη του χωρίου όπου θα κινηθεί και ιδιαίτερα δεν έχει αντίληψη της θέσης και της έκτασης των πιθανών εμποδίων και του βαθμού που αυτά μπορούν να επηρεάσουν την κίνησή του.

Το πρόβλημα αυτό καλείται "Εύρεση Βέλτιστης Διαδρομής<sup>1</sup> (ΕΒΔ)" και οι αλγόριθμοι που έχουν αναπτυχθεί για την αντιμετώπιση και βέλτιστη επίλυσή του καλούνται "Αλγόριθμοι Εύρεσης Βέλτιστης Διαδρομής<sup>2</sup> (ΕΒΔ)" .

Σε γενικές γραμμές, αυτοί οι αλγόριθμοι προσφέρουν τη δυνατότητα στον αυτόνομο πράκτορα να σχεδιάζει τις επόμενες κινήσεις του ώστε να μεταβεί υπό προϋποθέσεις μεταξύ δύο σημείων. Οι αυτόνομοι πράκτορες χρησιμοποιούν τις τεχνικές ΕΒΔ ιδιαίτερα στην περίπτωση που κινούνται σε περιβάλλον με πεπλεγμένες διατάξεις εμποδίων, όπου η χρήση μίας καθοδηγητικής συμπεριφοράς, όπως αυτής της αποφυγής εμποδίων, δεν επαρκεί για

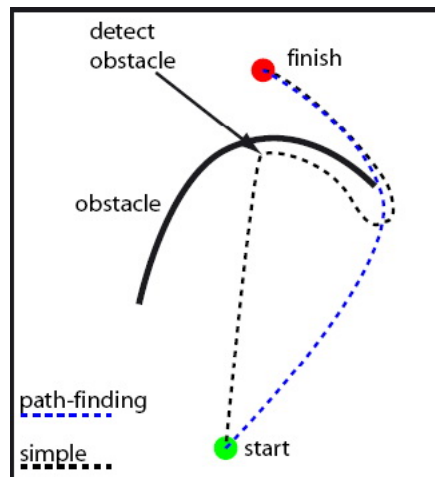
---

<sup>1</sup>Path Finding Problem

<sup>2</sup>Path Finding Algorithms

την επίτευξη του αντικειμενικού τους σκοπού. Το Σχήμα 3.1 αναπαριστά τη σύγκριση της κίνησης ενός πράκτορα με χρήση μίας τεχνικής ΕΒΔ με τη κίνηση που θα εκτελούσε με χρήση μίας άλλης καθοδηγητικής συμπεριφοράς.

Η χρήση των αλγόριθμων ΕΒΔ προϋποθέτει την επακριβή γνώση και χωροθέτηση του χωρίου που θα κινηθεί ο αυτόνομος πράκτορας και για το λόγο αυτό δεν μπορούν να χρησιμοποιηθούν στην περίπτωση που κατά τη διάρκεια της κίνησής του η διάταξη του χωρίου αυτού αλλάξει. Η χρήση τεχνικών παράκαμψης της κυρτότητας<sup>3</sup> (βλ. Σχήμα 3.2) προσεγγίζει την ανθρώπινη αντιμετώπιση του προβλήματος αλλά οι αλγόριθμοι ΕΒΔ μπορούν και επιδέχονται κατάλληλες ρυθμίσεις ώστε να προσομοιάζουν περισσότερο τον "ανθρώπινο" χαρακτήρα της κίνησης του αυτόνομου πράκτορα.



Σχήμα 3.1: Εύρεση Βέλτιστης Διαδρομής vs Ελεύθερης Κίνησης

## 3.2 Είδη Αλγορίθμων

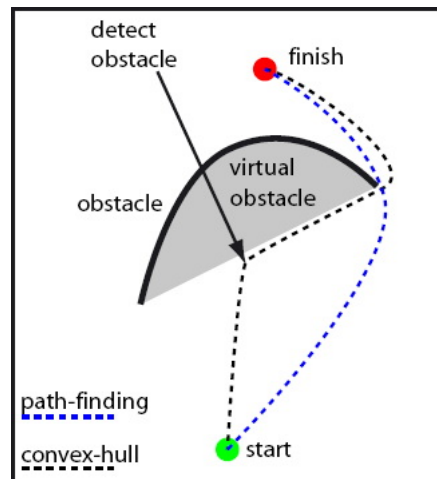
Το πρόβλημα της εύρεσης βέλτιστης διαδρομής σε ένα χώρο που περιλαμβάνει εμπόδια αποτελεί το αντικείμενο έρευνας πολλών ερευνητών από διάφορους επιστημονικούς κλάδους και έχει οδηγήσει σε μία ποικιλία λύσεων που επιλύουν το πρόβλημα τουλάχιστον στη αντίστοιχη περιοχή της έρευνάς τους. Παρόλα αυτά, οι περισσότεροι από τους προτεινόμενους

<sup>3</sup>Convex Hull Techniques



αλγόριθμους οδηγούν σε ένα πρόβλημα έρευνας γραφήματος. Έτσι, έχουμε τρεις κύριες κατηγορίες μεθόδων ΕΒΔ:

- Μέθοδος παρεμβολής γραμμής,
- Μέθοδος γραφημάτων βαρών,
  - Μέθοδοι Dijkstra και Best First Search
  - Μέθοδος εύρεσης A\*
- Άλλες μέθοδοι.



Σχήμα 3.2: Εύρεση Βέλτιστης Διαδρομής vs Κυρτής Επιφάνειας

### 3.2.1 Μέθοδος Παρεμβολής Γραμμής

Η μέθοδος παρεμβολής γραμμής είναι μία πολύ κοινή προσέγγιση όταν το πεδίο του προβλήματος μπορεί να χαρακτηριστεί δυαδικού τύπου (binary type). Με τον όρο δυαδικού τύπου εννοούμε ότι το πεδίο μπορεί να επιμερισθεί σε τέτοιο βαθμό ώστε να χαρακτηρίζεται είτε ως προσβάσιμο (ελεύθερο) είτε ως ένα αδιαπέραστο εμπόδιο. Επιπλέον δεν θεωρούμε μεταβολές της ταχύτητας ή άλλα εμπόδια στη κίνηση από την ίδια τη φύση του πράκτορα-οχήματος. Στην περίπτωση αυτή, η μικρότερη διαδρομή - ευκλείδειος απόσταση - είναι και εξ' ορισμού η βέλτιστη διαδρομή. Η βασική ιδέα σε αυτή τη κατηγορία αλγορίθμων μπορεί να συνοψισθεί ως ακολούθως:

*Υλοποίηση Καθοδηγητικών Συμπεριφορών*

1. Επιβάλλουμε συνθήκες κυρτότητας σε όλα τα αντικείμενα. Οι κορυφές όλων των πολυγώνων που κατασκευάζονται με τον τρόπο αυτό ορίζουν ένα σύνολο κορυφών.
2. Συνδέουμε όλες τις κορυφές και ορίζουμε ακμές (δημιουργούμε το αρχικό γράφημα).
3. Επιλέγουμε κορυφές με βάση συγκεκριμένα κριτήρια που βασίζονται στη συγκεκριμένη εφαρμογή ή είναι πιο γενικευμένα.
4. Έρευνα για τη συντομότερη έγκυρη διαδρομή μεταξύ δύο σημείων αρχής και τέλους με τη χρήση συνηθισμένων αλγορίθμων γραφήματος.

Ο σκοπός της επιλογής συγκεκριμένων κορυφών είναι να απομακρύνουμε αυτές που δεν ανήκουν στη συντομότερη διαδρομή και έτσι να μειώσουμε τον αριθμό των πιθανών διαδρομών που θα εξετάσουμε αργότερα. Έτσι οι διαδρομές που τέμνουν το περίγραμμα του κελύφους μπορούν να απορριφθούν άμεσα. Η μέθοδος που συνήθως προτείνεται περιλαμβάνει τρόπους υπολογισμού αποστάσεων και ευριστικές μεθόδους που συνδυάζονται για να αποκλειστούν πιθανές ακμές που παραμένουν μη ανιχνεύσιμες. Κάποιες συνηθισμένες εκδόσεις της μεθόδου παρεμβολής γραμμής παρουσιάζονται στα [13] και [25].

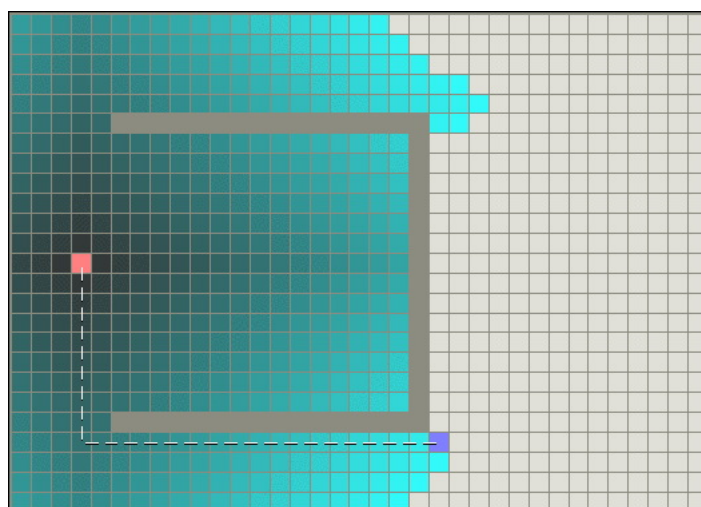
### 3.2.2 Μέθοδοι Γραφήματος Βαρών

Αν και οι μέθοδοι γραφήματος βαρών είναι ουσιαστικά μέθοδοι έρευνας γραφήματος όπως και οι μέθοδοι παρεμβολής γραμμής, οι μέθοδοι αυτοί έχουν μία ριζοσπαστικά διαφορετική και κατά μία έννοια αντίθετη προσέγγιση στην κατασκευή του γραφήματος. Η βασική ιδέα είναι η διαίρεση του χώρου σε διακριτές περιοχές, που συνήθως καλούνται κελιά, ώστε να περιοριστούν οι πιθανές κινήσεις από ένα συγκεκριμένο κελί σε ένα γειτονικό του. Κατόπιν κατασκευάζουμε ένα γράφημα κατεύθυνσης λαμβάνοντας τα κελιά σαν κορυφές του γραφήματος και τις πιθανές κινήσεις στα γειτονικά τους κελιά σαν κατευθυνόμενες ακμές μεταξύ κορυφών. Μία συνάρτηση γραφήματος βαρών ορίζεται με την ανάθεση μίας τιμής σε κάθε κορυφή που αντιστοιχεί στο κόστος μετάβασης κατά μήκος της διερευνόμενης ακμής λαμβάνοντας υπόψη το χρόνο μετάβασης, το μήκος της ακμής ή οποιαδήποτε συνάρτηση που έχουμε υπόψη μας. Η διαίρεση του χώρου, ο ορισμός των γειτονικών κελιών και ο ορισμός

του κόστους μετάβασης μπορούν να διαφέρουν από μέθοδο σε μέθοδο της κατηγορίας αυτής. Η μέθοδος  $A^*$  που χρησιμοποιείται στο VRKernel ανήκει στη προσέγγιση αυτή και θα αναλυθεί με λεπτομέρεια αργότερα.

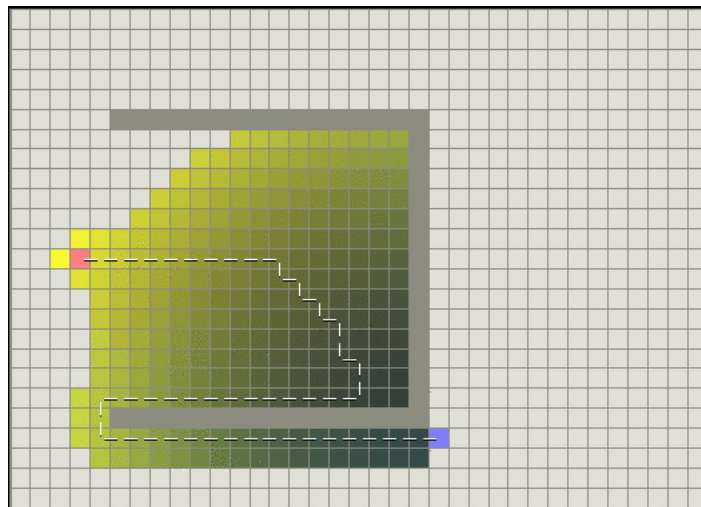
### 3.2.2.1 Μέθοδος Dijkstra και Best First Search

Ο αλγόριθμος Dijkstra (βλ. [6] και [7]) ουσιαστικά επισκέπτεται κορυφές σε ένα γράφημα βαρών, αρχίζοντας από τη θέση του οχήματος. Έπειτα εξετάζει επαναληπτικά τη πλησιέστερη κορυφή που δεν έχει διερευνηθεί ακόμα, προσθέτοντας τις κορυφές στο σύνολο των κορυφών που θα εξετασθούν μελλοντικά. Επεκτείνεται από ένα σημείο έναρξης μέχρι να ανακαλύψει το καταληκτικό σημείο. Ο αλγόριθμος Dijkstra εγγυάται να ανακαλύψει το συντομότερο διάδρομο μεταξύ ενός σημείου αφετηρίας και ενός σημείου τερματισμού, όσο καμία από τις ακμές που ενώνουν τα κελιά της διαδρομής δεν έχει αρνητικό κόστος. Εδώ θα πρέπει να σημειωθεί ότι ο αλγόριθμος αυτός ανακαλύπτει το πρώτο από τους δυνατούς διαδρόμους ελάχιστης απόστασης αφού υπάρχουν πολλοί διάδρομοι με ισοδύναμη ελάχιστη διαδρομή αλλά διαφορετική διαδρομή. Στο σχήμα 3.3, η αφετηρία είναι το ροζ τετραγωνάκι, το γαλάζιο είναι το κελί τερματισμού και τα τυρκουάζ κελιά είναι τα κελιά που διερευνήθηκαν από τον αλγόριθμο. Οι ελαφρές αποχρώσεις του τυρκουάζ δείχνουν τις πιο απομακρυσμένες περιοχές από την αφετηρία και σχηματίζουν τα όρια της εξερεύνησης.



Σχήμα 3.3: *Εύρεση Βέλτιστης Διαδρομής με τον αλγόριθμο Dijkstra*

Ο αλγόριθμος Best First Search (BFS) εργάζεται με ένα παρόμοιο τρόπο εκτός του ότι με ένα ευριστικό τρόπο εκτιμά την απόστασή από τον τερματισμό του τρέχοντος σημείου. Αντί να επιλέγει σταδιακά το σημείο που βρίσκεται πλησιέστερα στην αφετηρία, επιλέγει το σημείο που βρίσκεται πλησιέστερα στο τερματισμό. Ο αλγόριθμος BFS δεν εγγυάται την ανεύρεση της συντομότερης διαδρομής αλλά είναι ταχύτερος από τον αλγόριθμο Dijkstra επειδή χρησιμοποιεί μία ευριστική συνάρτηση για να οδηγηθεί προς τον τερματισμό πολύ γρήγορα. Ως παράδειγμα, αν ο τερματισμός είναι νότια της αφετηρίας, ο αλγόριθμος BFS θα εστιασθεί σε διαδρόμους που οδηγούν νότια. Στο σχήμα 3.4, το κίτρινο χρώμα αναπαριστά τα κελιά με υψηλή ευριστική τιμή (υψηλό κόστος προς τον τερματισμό) και το μαύρο χρώμα αναπαριστά τα κελιά με χαμηλή ευριστική τιμή (χαμηλό κόστος προς τον τερματισμό). Δείχνει επίσης ότι ο αλγόριθμος μπορεί να βρει διαδρόμους πολύ γρήγορα, αν υπολογίσει κανείς τα επισκεπτόμενα κελιά, σε σύγκριση με τον αλγόριθμο Dijkstra αλλά ο διάδρομος που προτείνει τελικά δεν είναι και ο βέλτιστος.



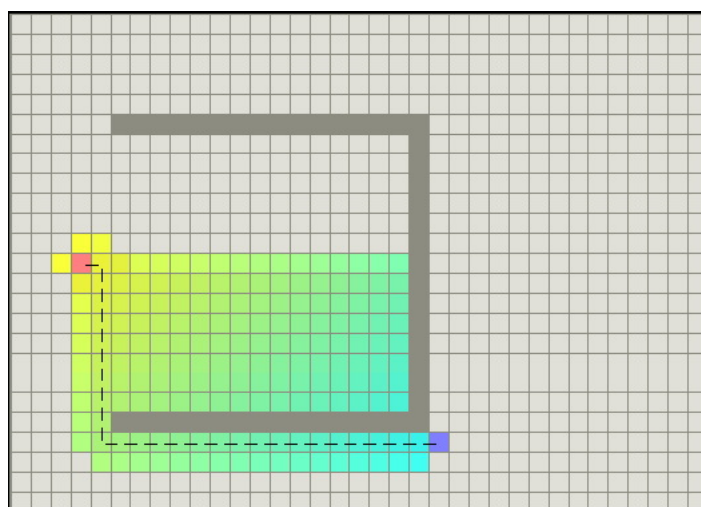
**Σχήμα 3.4:** Εύρεση Βέλτιστης Διαδρομής με τον αλγόριθμο Best First Search

Το γενικότερο πρόβλημα του αλγόριθμου BFS είναι ότι είναι "άπληστος" και προσπαθεί να κινείται πάντα προς τη διεύθυνση του τερματισμού ακόμα και όταν δεν βρίσκεται στον ορθό διάδρομο. Επειδή λαμβάνει υπόψη του μόνο το κόστος για να φθάσει στο στόχο του και αδιαφορεί για το κόστος του μέχρι εκείνου του σημείου διανυθέντος διαδρόμου, συνεχίζει να κινείται πάνω σε ένα διάδρομο που μπορεί και να έχει γίνει ήδη πολύ μακρύς.

### 3.2.2.2 Μέθοδος Εύρεσης $A^*$

Ο αλγόριθμος  $A^*$  αναπτύχθηκε το 1968 με σκοπό να συνδυάσει ευριστικές προσεγγίσεις όπως αυτή του BFS και μαθηματικά ορθές προσεγγίσεις όπως αυτή του Dijkstra. Είναι βέβαια λίγο παράδοξο ότι οι ευριστικές προσεγγίσεις όπως αυτή του BFS παρέχουν μία προσεγγιστική επίλυση του προβλήματος χωρίς καμία εγγύηση για τη βέλτιστη λύση. Παρόλα αυτά, ο αλγόριθμος αυτός είναι από τους κορυφαίους των ευριστικών μεθόδων και αν και οι ευριστικοί δεν εγγυώνται την ανεύρεση της βέλτιστης λύσης, ο  $A^*$  μπορεί και εγγυάται το συντομότερο διάδρομο μεταξύ δύο σημείων.

Ο  $A^*$  είναι η πιο κοινή επιλογή για το πρόβλημα εύρεσης βέλτιστου διαδρόμου γιατί είναι αρκετά ευέλικτος και μπορεί να χρησιμοποιηθεί σε ένα αρκετά μεγάλο εύρος εφαρμογών. Όπως έχουμε αναφέρει προηγουμένως, ο  $A^*$  είναι παρόμοιος με τους άλλους αλγόριθμους έρευνας γραφήματος λόγω του γεγονότος ότι μπορεί να εξερευνήσει μία πιθανώς δυσανάλογα μεγάλη περιοχή της εσχάρας. Όμως είναι παρόμοιος με τον αλγόριθμο Dijkstra λόγω του ότι μπορεί να χρησιμοποιηθεί για την ανεύρεση της συντομότερης διαδρομής. Όμως είναι παρόμοιος με τον αλγόριθμο BFS λόγω του ότι κατευθύνεται μόνος του και συνεχώς προς τον τερματισμό. Τελικά είναι αρκετά γρήγορος όσο και ο BFS σε σχέση με το χώρο εξερεύνησης αλλά και εγγυάται ότι θα ανακαλύψει το συντομότερο και ασφαλέστερο διάδρομο μεταξύ δύο σημείων (βλ. σχήμα 3.5).



Σχήμα 3.5: Εύρεση Βέλτιστης Διαδρομής με τον αλγόριθμο  $A^*$

Η επιτυχία του αλγόριθμου  $A^*$  έγκειται στο γεγονός ότι συνδυάζει πληροφορίες που χρησιμοποιεί και ο αλγόριθμος Dijkstra (προτίμηση κορυφών που είναι πλησιέστερα στην αφετηρία) αλλά και ο αλγόριθμος BFS (προτίμηση κορυφών που βρίσκονται πλησιέστερα στον τερματισμό). Όταν αναφερόμαστε στον αλγόριθμο  $A^*$ , χρησιμοποιούμε το μέγεθος  $g(n)$  που αντιπροσωπεύει το κόστος της διαδρομής από την αφετηρία μέχρι ένα συγκεκριμένο κελί-σημείο  $n$  και το μέγεθος  $h(n)$  που αντιπροσωπεύει το ευριστικό εκτιμώμενο κόστος από το κελί-σημείο  $n$  μέχρι το τερματισμό. Στο σχήμα 3.4 τα κίτρινα κελιά (υψηλές τιμές  $h$ ) αντιπροσωπεύουν τα κελιά που απέχουν πολύ από τον τερματισμό και οι τυρκουάζ (υψηλές  $g$  τιμές) αντιπροσωπεύουν κελιά μακριά από την αφετηρία. Ο αλγόριθμος  $A^*$  ισορροπεί τις δύο τιμές καθώς κινείται από την αφετηρία προς τον τερματισμό. Κάθε φορά και μέσω του κύριου επαναληπτικού βρόγχου, εξετάζει το κελί  $n$  που διαθέτει το ελάχιστο  $f(n) = g(n) + h(n)$ . Η ευριστική συνάρτηση  $h(n)$  οδηγεί τον αλγόριθμο με δεδομένη μίας εκτίμησης του ελάχιστου κόστους από το συγκεκριμένο κελί  $n$  προς το τερματισμό. Προφανώς είναι πολύ σημαντικό να επιλεγεί η κατάλληλη ευριστική συνάρτηση.

Ακολούθως θα δούμε πώς η ευριστική συνάρτηση ελέγχει τη συμπεριφορά του  $A^*$ :

- Στην οριακή περίπτωση όπου η  $h(n)$  είναι 0, τότε σημασία έχει μόνο η  $g(n)$  και ο  $A^*$  εκφυλίζεται σε αλγόριθμο Dijkstra, ο οποίος και εγγυάται την εύρεση της μικρότερης διαδρομής.
- Αν η  $h(n)$  είναι πάντοτε μικρότερη ή ίση με το κόστος της μετακίνησης από το κελί  $n$  προς το κελί τερματισμού, τότε ο  $A^*$  εγγυάται την εύρεση της μικρότερης διαδρομής. Όσο μικρότερη είναι η τιμή της  $h(n)$ , τόσο ο  $A^*$  επεκτείνεται σε διερεύνηση περισσότερων κελιών και γίνεται βραδύτερος υπολογιστικά.
- Αν η  $h(n)$  είναι ίση με το κόστος της μετακίνησης από το κελί  $n$  προς το κελί τερματισμού, τότε ο  $A^*$  θα ακολουθήσει τη βέλτιστη διαδρομή χωρίς επέκταση σε περιττά κελιά, κάνοντάς τον έτσι ταχύτερο υπολογιστικά. Αν και αυτό δεν είναι κατορθωτό σε όλες τις περισσότερες, μπορούμε να το πετύχουμε τουλάχιστον στις περισσότερες. Πάντως θα πρέπει να σημειώσουμε ότι αν  $A^*$  τροφοδοτείται με "τέλειες" πληροφορίες, τότε θα συμπεριφέρεται και "τέλεια".

- Αν η  $h(n)$  είναι μερικές φορές μεγαλύτερη από το κόστος της μετακίνησης από το κελί  $n$  προς τον τερματισμό, τότε ο  $A^*$  δεν εγγυάται την εύρεση της βέλτιστης διαδρομής αλλά μπορεί να είναι ταχύτερος υπολογιστικά.
- Στην οριακή περίπτωση που η  $h(n)$  είναι πολύ μεγαλύτερη από την  $g(n)$ , τότε παίζει ρόλο μόνο η  $h(n)$  και ο  $A^*$  εκφυλίζεται σε BFS.

Η ικανότητα του  $A^*$  να μεταβάλλει τη συμπεριφορά του βασιζόμενος στις ευριστικές και συναρτησείς κόστους μπορεί να αποδειχθεί πολύ χρήσιμη στη παρούσα μελέτη. Μπορούμε να εκμεταλλευτούμε το αντιστάθμισμα μεταξύ ταχύτητας και ακρίβειας είτε για να επιταχύνουμε την εκτέλεση της εφαρμογής είτε για να διαφοροποιήσουμε τις διαδρομές που ακολουθούν τα διάφορα οχήματα.

### 3.2.3 Άλλες Μέθοδοι

Πολλοί ερευνητές έχουν προσπαθήσει να προσεγγίσουν το ζήτημα και με άλλες διαφορετικές μεθόδους. Ως παράδειγμα αναφέρεται το [14], όπου χρησιμοποιείται μία διωνυμική συνάρτηση για να υποτυπώνει την εξέλιξη των καμπυλών ίσων αποστάσεων πάνω σε μία επιφάνεια και εξάγεται ένα σχήμα από την κατασκευή αυτή που χρησιμοποιείται για την εύρεση διαδρομών ελάχιστης απόστασης πάνω σε πολλές επιφάνειες.

Μία άλλη ιδέα είναι να ορίσουμε επιφάνειες συγκεκριμένου κόστους και να ακολουθήσουμε τη διεύθυνση της τοπικής παραγωγής π.χ. η μέθοδος της πιο απόκρημνης καθόδου. Πάντως, παρόλο που η βασική ιδέα είναι αρκετά απλή, η διαχείριση του τοπικού ελάχιστου δεν είναι αρκετά δύσκολη. Ας θεωρήσουμε για παράδειγμα ένα ποτάμι που ρέει κατά τη διεύθυνση μίας αρνητικής κλίσης του ύψους του εδάφους. Στη πραγματικότητα, θα δημιουργηθούν λίμνες και λεκάνες στις περιοχές των τοπικών ελαχίστων. Το πρόβλημα είναι ότι αν δεν διαχειριστούμε σωστά τα τοπικά ελάχιστα, τότε είναι δυνατόν το όχημα να "κολλήσει" πολύ εύκολα σε ένα από αυτά. Έχουν προταθεί διάφοροι αλγόριθμοι και τεχνικές όπως για παράδειγμα η τεχνική της "πλήρωσης του τοπικού ελαχίστου" που περιγράφεται στο [1].

Πάνω στο θέμα έχουν προταθεί προσεγγίσεις που βασίζονται σε ευριστικές μεθόδους. Στη πραγματικότητα, οι μέθοδοι αυτοί δεν εγγυώνται παρά μόνο τη βελτιστοποίηση ορισμέ-

νων μεγεθών βασισμένες σε απλές θεωρήσεις για το υπό εξέταση πεδίο και για αυτό πολλές φορές καταλήγουν σε αδιέξοδο ή στη συγκεκριμένη περίπτωση σε τέλμα. Σε ορισμένες δε περιπτώσεις, δεν υπάρχει ούτε καν εγγύηση εύρεσης έστω και μίας διαδρομής. Πάντως οι ευριστικές μέθοδοι είναι μία καλή προσέγγιση, ιδιαίτερα όταν χρησιμοποιούνται σαν δευτερεύοντα συστατικά για την επιτάχυνση πιο εξονυχιστικών μεθόδων, όπως αυτές που ανήκουν στις δύο προηγούμενες κατηγορίες που περιγράφηκαν προηγουμένως.

□



## Κεφάλαιο 4

# Ανάπτυξη Καθοδηγητικών Συμπεριφορών

Η διπλωματική εργασία έχει ως κεντρικό θέμα την ανάπτυξη και υλοποίηση συγκεκριμένων καθοδηγητικών συμπεριφορών με χρήση αντικειμενοστραφούς προγραμματισμού σε γλώσσα C++ ώστε να αποτελέσουν λειτουργικό τμήμα και επέκταση του πυρήνα ΕΠ VRKernel. Οι καθοδηγητικές συμπεριφορές που θα αναπτυχθούν είναι οι παρακάτω:

- Αποφυγή (Flee)
- Καταδίωξη (Pursuit)
- Διαφυγή (Evasion)
- Αποφυγή Εμποδίων (Obstacle Avoidance)
- Ακολουθία Διαδρομής (Path-Following)
- Ευθυγράμμιση (Alignment)
- Άφιξη(Arrive)

Όπως αναφέρθηκε και στα εισαγωγικά κεφάλαια, οι συμπεριφορές που θα εξετασθούν βασίζονται πάνω σε ένα κοινό μοτίβο: την παρουσία ενός αυτόνομου πράκτορα στόχου και

ενός αυτόνομου πράκτορα που κινείται προς τον στόχο για την ικανοποίηση ενός αντικειμενικού σκοπού. Οι δύο πράκτορες διαθέτουν μία ταχύτητα και μία πορεία κίνησης αναλόγως της κάθε συμπεριφοράς<sup>1</sup>.

Θα πρέπει να σημειωθεί ότι πέραν της λογισμικής ανάπτυξης των καθοδηγητικών συμπεριφορών, παρατίθεται και μία παραμετροποίηση των δεδομένων τους με στόχο την όσο το δυνατόν πιο ρεαλιστική υλοποίησή τους.

## 4.1 Αποφυγή

### 4.1.1 Περιγραφή

Η πιο απλή συμπεριφορά είναι η "Αποφυγή", η οποία όμως εξετάζεται σε συνδυασμό με τη συζυγή της, την "Αναζήτηση". Οι συμπεριφορές αυτές επιτρέπουν ένα κινούμενο πράκτορα να προσεγγίζει ή να απομακρύνεται από έναν ακίνητο στόχο με μία σταθερή ταχύτητα.

### 4.1.2 Ψευδοκώδικας

Θα εξετάσουμε καταρχήν το ψευδοκώδικα υλοποίησης της συμπεριφοράς "Αναζήτηση" και στη συνέχεια θα τον τροποποιήσουμε κατάλληλα για να περιγράψουμε την συμπεριφορά "Αποφυγή".

Η συμπεριφορά "Αναζήτηση" προσαρμόζει τη θέση του οχήματος έτσι ώστε το διάνυσμα της ταχύτητάς του να παραμένει ακτινικά ευθυγραμμισμένο με το στόχο. Η προσέγγιση του πράκτορα στόχου επιτυγχάνεται σε τρία στάδια:

- Αναζήτηση της επιθυμητής πορείας του κινούμενου πράκτορα με την εξεύρεση του ανύσματος `desired_velocity` δηλαδή του ανύσματος που συνδέει τη θέση του με τη θέση του ακίνητου πράκτορα στόχου:

$$desired\_velocity_M = position_M - position_T$$

---

<sup>1</sup>Εφεξής, τα ανύσματα που αναφέρονται στον πράκτορα κίνησης θα φέρουν το διακριτικό M ή m ενώ του πράκτορα στόχου T ή t

- Κανονικοποίηση του ανύσματος της επιθυμητής πορείας. Το μήκος του ανύσματος μπορεί να είναι η μέγιστη ταχύτητα που διαθέτει ο κινούμενος πράκτορας `max_speed` ή η ταχύτητα χειρισμού του κινούμενου πράκτορα αναλόγως της εφαρμογής:

$$desired\_velocity_M = normalize(position_M - position_T) * max\_speed_M$$

- Πορεία χειρισμού του κινούμενου πράκτορα `seek_steering`:

$$seek\_steering_M = desired\_velocity_M - current\_velocity_M$$

Υπάρχει βέβαια η πιθανότητα το όχημα να περάσει κοντά από τη θέση του στόχου, να μη σταματήσει και να τον προσπεράσει. Στη περίπτωση αυτή, το όχημα θα επανακυκλώσει και θα προσεγγίσει τον ακίνητο στόχο με τον ίδιο τρόπο. Η τελική κίνηση σε απόλυτες συντεταγμένες, προσομοιάζει πολύ τη κίνηση που κάνει ένα κουνούπι γύρω από μία λάμπα φωτισμού.

Θα πρέπει να σημειωθεί ότι ο κινούμενος πράκτορας δεν κινείται υπό την επήρεια σταθερής δύναμης έλξης από τον πράκτορα στόχο γιατί στην περίπτωση αυτή θα είχε μετατραπεί σε δορυφόρο του πράκτορα στόχου.

Η συμπεριφορά "Αποφυγή" είναι το αντίστροφο της "Αναζήτηση" και προσαρμόζει τη θέση του οχήματος ώστε το διάνυσμα της ταχύτητάς του να είναι ευθυγραμμισμένο και ακτινικά αντίθετο με τον ακίνητο στόχο. Η αποφυγή του πράκτορα στόχου επιτυγχάνεται σε τρία στάδια:

- Αναζήτηση της επιθυμητής πορείας του κινούμενου πράκτορα με την εξεύρεση του ανύσματος `desired_velocity` δηλαδή του ανύσματος που συνδέει τη θέση του με τη θέση του ακίνητου πράκτορα στόχου:

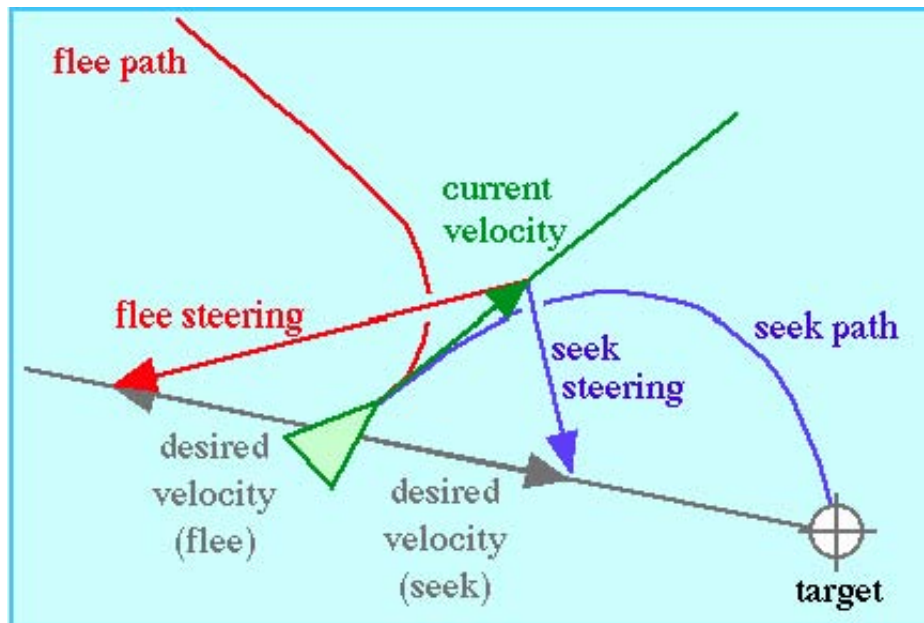
$$desired\_velocity_M = position_T - position_M)$$

- Κανονικοποίηση του ανύσματος της επιθυμητής πορείας. Το μήκος του ανύσματος μπορεί να είναι η μέγιστη ταχύτητα που διαθέτει ο κινούμενος πράκτορας `max_speed` ή η ταχύτητα χειρισμού του κινούμενου πράκτορα αναλόγως της εφαρμογής:

$$desired\_velocity_M = normalize(position_T - position_M) * max\_speed_M$$

- Πορεία χειρισμού του κινούμενου πράκτορα `flee_steering`:

$$flee\_steering_M = desired\_velocity_M - current\_velocity_M$$



Σχήμα 4.1: Αναζήτηση & Αποφυγή

### 4.1.3 Υλοποίηση

Η καθοδηγητική συμπεριφορά "Αποφυγή" υλοποιήθηκε με βάση τον ψευδοκώδικα που περιγράφηκε στο προηγούμενο εδάφιο. Κατόπιν διαμορφώθηκε σε μία κλάση αντικειμενοστραφούς προγραμματισμού γλώσσας C++ και προσαρμόστηκε στο κέλυφος χειρισμού του πυρήνα VRKernel αφού έγιναν οι κατάλληλες αλλαγές στο γραφικό περιβάλλον αλληλεπίδρασης του χρήστη<sup>2</sup>.

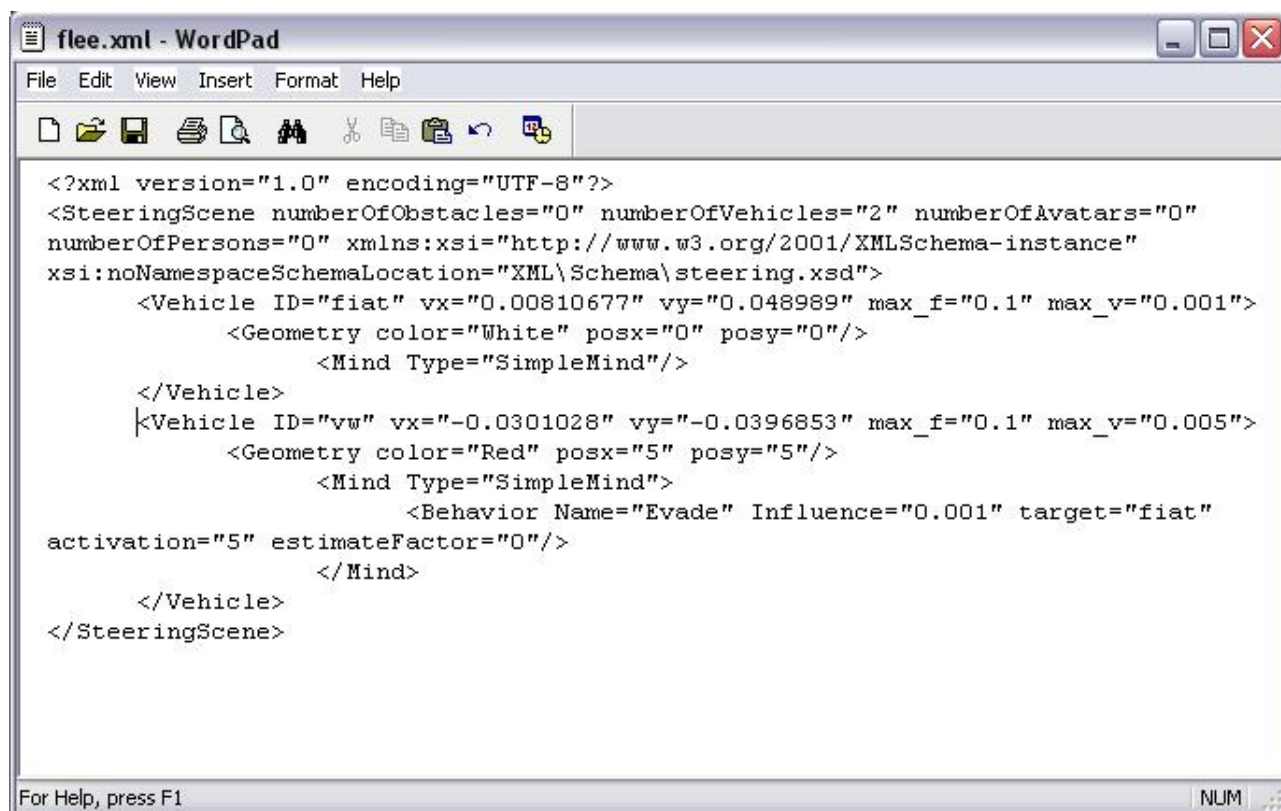
Κατόπιν έγινε η εισαγωγή των βασικών ρυθμίσεων των πρακτόρων που θα λάβουν μέρος στη προσομοίωση σύμφωνα με το αρχείο τύπου XML όπως αυτό παρουσιάζεται στο Σχήμα 4.2.

<sup>2</sup>Graphic User Interface, GUI

Στα Σχήματα 4.3 έως 4.7 παρουσιάζεται η διαδοχική εκτέλεση ενός τυπικού σεναρίου με τη χρήση της συμπεριφοράς "Αποφυγή". Σύμφωνα με το σενάριο αυτό, στο κέντρο ενός καρτεσιανού συστήματος συντεταγμένων ευρίσκεται ο λευκός πράκτορας στόχος, ο οποίος και στη προκειμένη περίπτωση είναι ακίνητος. Ο ερυθρός κινούμενος πράκτορας ευρίσκεται αρχικά στη θέση  $(x, y) = (5.0, 5.0)$  και το άνωσμα της ταχύτητάς του είναι:

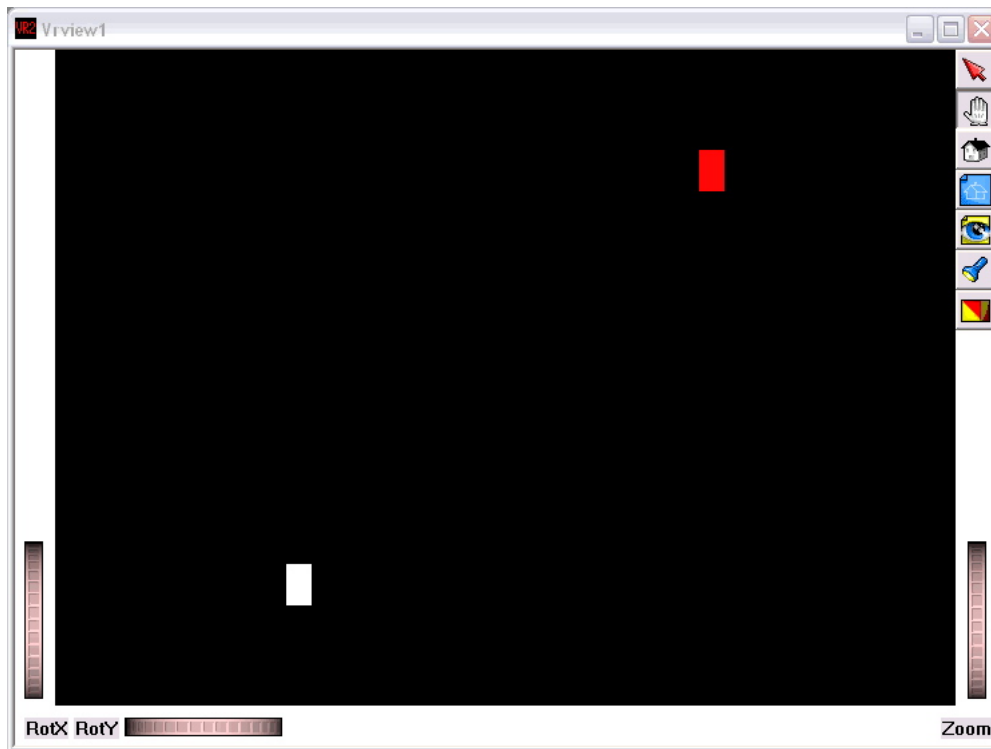
$$\vec{V}_M = \{-0.00301028, -0.00399269\}, \text{ όπου } \|\vec{V}_M\|_{max} = 0.005$$

Ο κινούμενος πράκτορας κατευθύνεται αρχικά προς τον λευκό ακίνητο πράκτορα (Σχήματα 4.3 και 4.4) αλλά όταν φθάσει σε απόσταση ενεργοποίησης του αλγόριθμου "Αποφυγή", που στη συγκεκριμένη περίπτωση είναι  $r = 5$  μήκη, τότε αρχίζει τον ελιγμό αποφυγής (Σχήμα 4.5) και εκτελεί μία παραβολοειδή κίνηση απομακρυνόμενος σταδιακά του πράκτορα στόχου (Σχήματα 4.6 και 4.7).

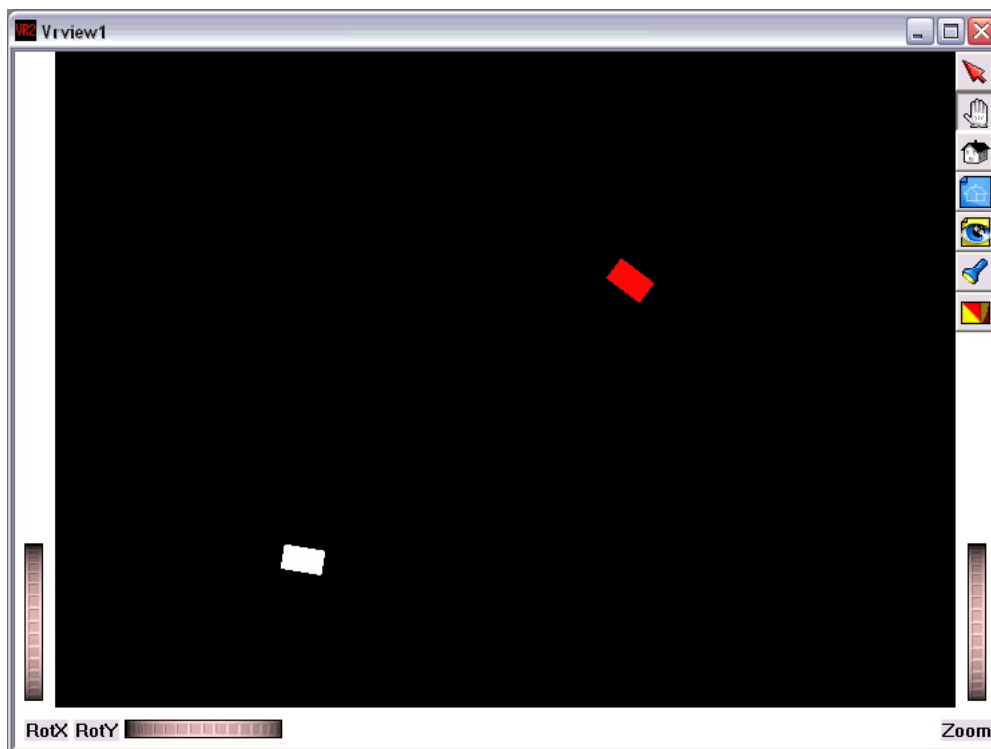


```
<?xml version="1.0" encoding="UTF-8"?>
<SteeringScene numberOfObstacles="0" numberOfVehicles="2" numberOfAvatars="0"
numberOfPersons="0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XML\Schema\steering.xsd">
  <Vehicle ID="fiat" vx="0.00810677" vy="0.048989" max_f="0.1" max_v="0.001">
    <Geometry color="White" posx="0" posy="0"/>
    <Mind Type="SimpleMind"/>
  </Vehicle>
  <Vehicle ID="vw" vx="-0.0301028" vy="-0.0396853" max_f="0.1" max_v="0.005">
    <Geometry color="Red" posx="5" posy="5"/>
    <Mind Type="SimpleMind">
      <Behavior Name="Evade" Influence="0.001" target="fiat"
activation="5" estimateFactor="0"/>
    </Mind>
  </Vehicle>
</SteeringScene>
```

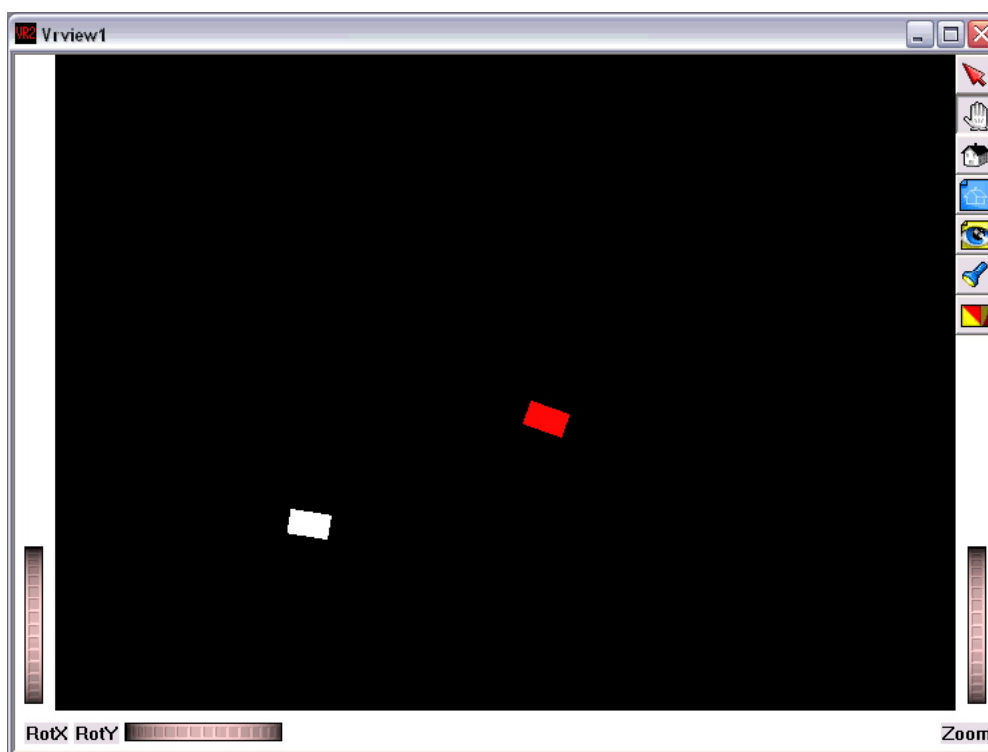
Σχήμα 4.2: Εισαγωγή Στοιχείων Συμπεριφοράς "Αποφυγή"



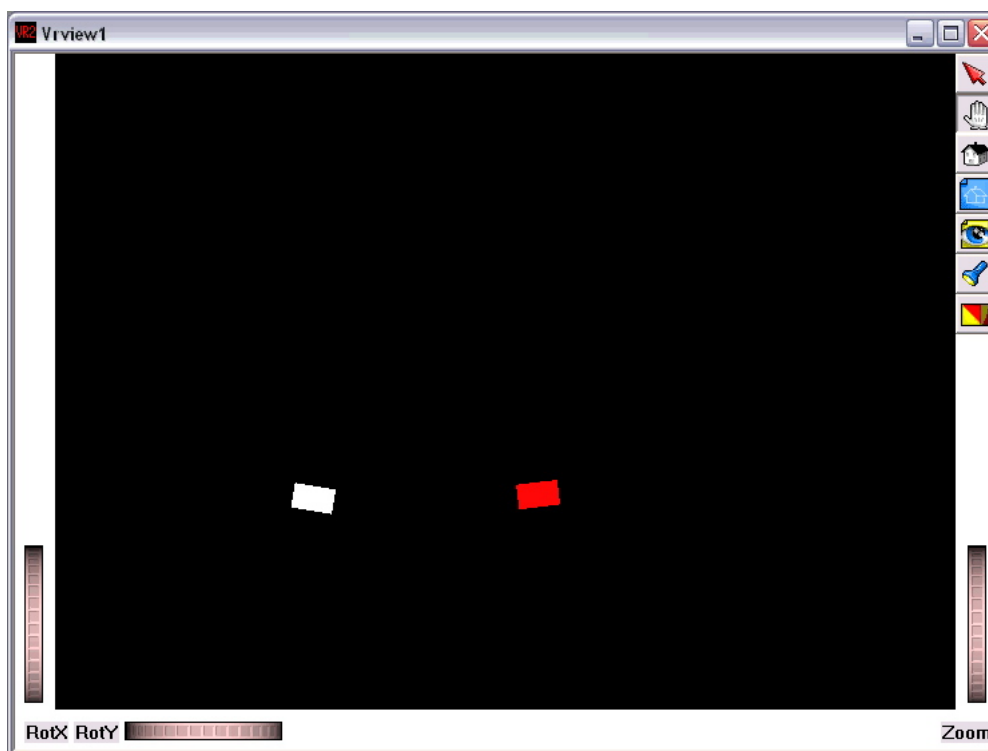
Σχήμα 4.3: Καθοδηγητική Συμπεριφορά "Αποφυγή" - Έναρξη



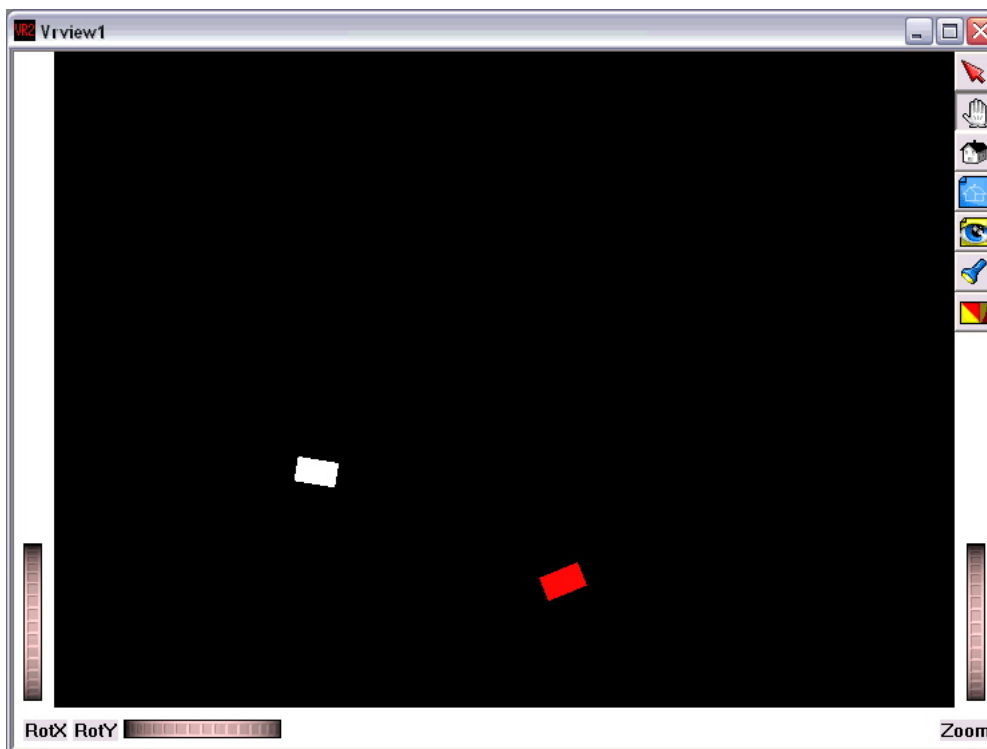
Σχήμα 4.4: Καθοδηγητική Συμπεριφορά "Αποφυγή" - Προσέγγιση



Σχήμα 4.5: Καθοδηγητική Συμπεριφορά "Αποφυγή" - Ενεργοποίηση



Σχήμα 4.6: Καθοδηγητική Συμπεριφορά "Αποφυγή" - Ελάχιστη Προσέγγιση



Σχήμα 4.7: Καθοδηγητική Συμπεριφορά "Αποφυγή" - Απομάκρυνση

## 4.2 Καταδίωξη

### 4.2.1 Περιγραφή

Η "Καταδίωξη" είναι παρόμοια με την "Αναζήτηση" εκτός από το γεγονός ότι ο στόχος είναι κινούμενος και μία αποτελεσματική "Καταδίωξη" απαιτεί τον προϋπολογισμό της επόμενης θέσης του στόχου. Μία σωστή προσέγγιση του προβλήματος απαιτεί την ακριβή γνώση του χρονικού διαστήματος που παρεμβάλλεται από την ενεργοποίηση της συμπεριφοράς μέχρι τη στιγμή που ο κινούμενος πράκτορας προσεγγίσει το πράκτορα στόχο. Όμως κάτι τέτοιο είναι σχεδόν αδύνατον να εκτιμηθεί αφού οι πράκτορες στόχοι θα εκτελούν κατά πάσα πιθανότητα συνεχείς ελιγμούς.

Στη παρούσα μελέτη, επιλέγουμε να εξετάσουμε τη διαδοχική προσέγγιση του πράκτορα στόχου ανά κάθε χρονικό στιγμιότυπο (frame). Θα χρησιμοποιήσουμε συνεπώς μία αρχική εκτίμηση του χρόνου προσέγγισης του στόχου και στη συνέχεια θα προβούμε σε συνεχείς αναθεωρήσεις του. Με το τρόπο αυτό, υποθέτουμε ότι ο πράκτορας στόχος έχει ευθύγραμμη



και ισοταχή κίνηση στο χώρο και δεν θα εκτελέσει κάποιο ελιγμό, τουλάχιστον για το χρονικό διάστημα που εξετάζουμε. Η υπόθεση δύναται να θεωρηθεί επαρκής εφόσον η χρονική στιγμή που εξετάζουμε συνήθως έχει διάρκεια  $1/25$  του δευτερολέπτου (όσο διαρκεί ένα στιγμιότυπο της προσομοίωσης). Θα χρησιμοποιήσουμε, συνεπώς, την αναμέτρηση για χρόνο ίσο με τη διάρκεια ενός στιγμιότυπου της προσομοίωσης και θα μεταβάλλουμε ανάλογα τη θέση των δύο πρακτόρων της προσομοίωσης σύμφωνα με το διάνυσμα της ταχύτητάς τους. Κατόπιν θα προβούμε στην απαραίτητη διόρθωση αναλόγως με τη νέα θέση, πορεία και ταχύτητα του στόχου οπότε και θα αναθεωρήσουμε τη πορεία διόρθωσης του κινούμενου πράκτορα.

Βέβαια ένας σημαντικός παράγοντας στο πρόβλημά μας είναι και η εκάστοτε απόσταση μεταξύ των πρακτόρων αλλά και η διαφορά των ανυσμάτων των ταχυτήτων τους. Αυτό σημαίνει ότι ο κινούμενος πράκτορας θα πρέπει να εκτελεί πιο δραστικούς ελιγμούς όσο πιο απομακρυσμένος είναι από το στόχο αλλά και όσο πιο αποκλίνουντα είναι τα ανύσματα των ταχυτήτων τους. Συνεπώς, θα προσπαθήσουμε να επιλύσουμε τις προϋποθέσεις του προβλήματος προσέγγισης ενός κινούμενου στόχου με την κατάλληλη επεξεργασία όχι των πραγματικών ανυσμάτων ταχύτητάς τους αλλά των ανυσμάτων της σχετικής τους απόστασης και ταχύτητας. Ο λόγος των δύο αυτών μεγεθών μας δίνει ένα χρονικό διάστημα  $\delta T$  που είναι ανάλογο της μεταξύ τους απόστασης και της διαφοράς των ταχυτήτων των πρακτόρων. Κατόπιν, θα εκτιμήσουμε την νέα θέση του στόχου για χρονικό διάστημα  $\delta T$  οπότε και θα προβούμε στην κατάλληλη διόρθωση του ανύσματος της ταχύτητας του κινούμενου πράκτορα. Είναι προφανές ότι όσο μεγαλύτερη είναι η απόσταση των δύο πρακτόρων και όσο μεγαλύτερη η διαφορά των ανυσμάτων των ταχυτήτων τους, τόσο δραστικότερη θα είναι η αλλαγή της ταχύτητας του κινούμενου πράκτορα. Το ανάλογο θα συμβεί στην περίπτωση που ο στόχος είναι εγγύς και κινείται σχεδόν παράλληλα.

#### 4.2.2 Ψευδοκώδικας

Συνεπώς, η συμπεριφορά "Καταδίωξη" θα προσπαθεί να προσαρμόζει τη θέση του κινούμενου πράκτορα έτσι ώστε το διάνυσμα της ταχύτητάς του να παραμένει όσο το δυνατόν ακτινικά ευθυγραμμισμένο με τη μελλοντική θέση του πράκτορα στόχου. Η προσέγγιση του

πράκτορα στόχου επιτυγχάνεται σε επτά στάδια:

- Αναζήτηση της σχετικής ταχύτητας μεταξύ του πράκτορα στόχου και του κινούμενου πράκτορα με την εξεύρεση του ανύσματος *relative\_velocity* δηλαδή του ανύσματος που αντιπροσωπεύει την προσέγγιση μεταξύ των δύο πρακτόρων:

$$relative\_velocity = current\_velocity_T - current\_velocity_M$$

- Αναζήτηση της σχετικής θέσης μεταξύ του πράκτορα στόχου και του κινούμενου πράκτορα με την εξεύρεση του ανύσματος *relative\_position*:

$$relative\_position = position_T - position_M$$

- Εξαγωγή του χρονικού διαστήματος *interval* που εκτιμάται ότι θα παρέλθει μέχρι τη στιγμή που ο κινούμενος πράκτορας θα συναντήσει τον πράκτορα στόχο:

$$interval = relative\_position / relative\_velocity$$

- Εξαγωγή της μελλοντικής θέσης του πράκτορα στόχου *future\_position* με βάση την παρούσα θέση του και τη θέση που προκύπτει από αναμέτρηση με βάση την ταχύτητα που διαθέτει και το χρονικό διάστημα που θα παρέλθει μέχρι να συναντηθεί με τον κινούμενο πράκτορα:

$$future\_position_T = position_T + velocity_T * interval$$

- Αναζήτηση της επιθυμητής πορείας του κινούμενου πράκτορα με την εξεύρεση του ανύσματος *desired\_velocity* δηλαδή του ανύσματος που συνδέει τη θέση του με τη μέλλουσα θέση του κινούμενου πράκτορα στόχου:

$$desired\_velocity_M = future\_position_T - position_M$$

- Κανονικοποίηση του ανύσματος της επιθυμητής πορείας. Το μήκος του ανύσματος μπορεί να είναι η μέγιστη ταχύτητα που διαθέτει ο κινούμενος πράκτορας  $max\_speed$  ή η ταχύτητα χειρισμού του κινούμενου πράκτορα αναλόγως της εφαρμογής:

$$desired\_velocity_M = normalize(future\_position_T - position_M) * max\_speed_M$$

- Πορεία χειρισμού του κινούμενου πράκτορα  $pursuit\_steering$ :

$$pursuit\_steering_M = desired\_velocity_M + current\_velocity_M$$



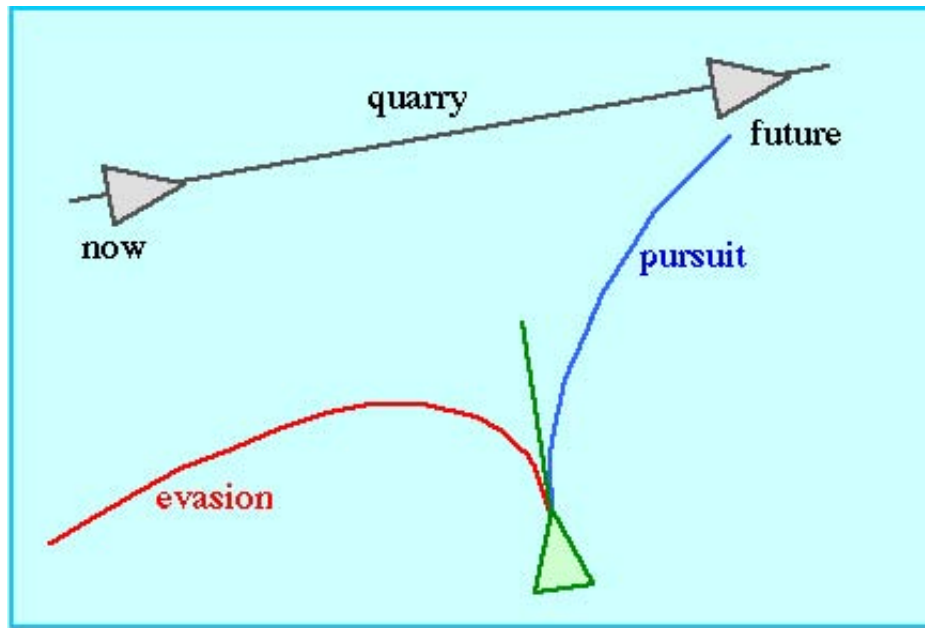
```
<?xml version="1.0" encoding="UTF-8"?>
<SteeringScene numberOfObstacles="0" numberOfVehicles="2" numberOfAvatars="0"
numberOfPersons="0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XML\Schema\steering.xsd">
  <Vehicle ID="fiat" vx="0.001" vy="0.001" max_f="0.1" max_v="0.001">
    <Geometry color="White" posx="0" posy="0"/>
    <Mind Type="SimpleMind"/>
  </Vehicle>
  <Vehicle ID="vw" vx="-0.0301028" vy="-0.0396853" max_f="0.1" max_v="0.005">
    <Geometry color="Red" posx="5" posy="0"/>
    <Mind Type="SimpleMind"/>
  </Vehicle>
</SteeringScene>
```

Σχήμα 4.8: Εισαγωγή Στοιχείων Συμπεριφοράς "Καταδίωξη"

### 4.2.3 Υλοποίηση

Η καθοδηγητική συμπεριφορά "Καταδίωξη" υλοποιήθηκε με βάση τον ψευδοκώδικα που περιγράφηκε στο προηγούμενο εδάφιο. Κατόπιν διαμορφώθηκε σε μία κλάση αντικειμενοστραφούς προγραμματισμού γλώσσας C++ και προσαρμόστηκε στο κέλυφος χειρισμού του

*Υλοποίηση Καθοδηγητικών Συμπεριφορών*



Σχήμα 4.9: Καταδίωξη & Διαφυγή

πυρήνα VRKernel αφού έγιναν οι κατάλληλες αλλαγές στο γραφικό περιβάλλον αλληλεπίδρασης του χρήστη. Η εισαγωγή των βασικών ρυθμίσεων των πρακτόρων που θα λάβουν μέρος στη προσομοίωση έγινε σύμφωνα με το αρχείο τύπου XML όπως αυτό παρουσιάζεται στο Σχήμα 4.8.

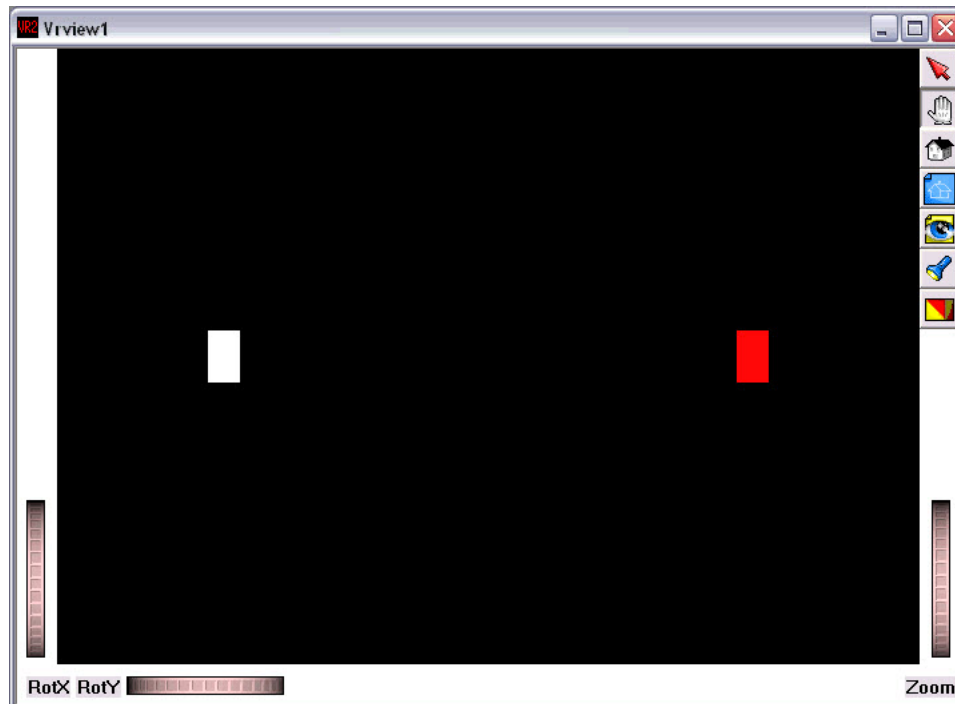
Στα Σχήματα 4.10 έως 4.14 παρουσιάζεται η διαδοχική εκτέλεση ενός τυπικού σεναρίου με τη χρήση της συμπεριφοράς "Καταδίωξη". Σύμφωνα με το σενάριο αυτό, στο κέντρο ενός καρτεσιανού συστήματος συντεταγμένων ευρίσκεται ο λευκός πράκτορας στόχος, ο οποίος και στη προκειμένη περίπτωση το άνυσμα της ταχύτητάς του είναι:

$$\vec{V}_T = \{0.0035355, 0.0035355\}, \text{ όπου } \|\vec{V}_T\|_{max} = 0.005$$

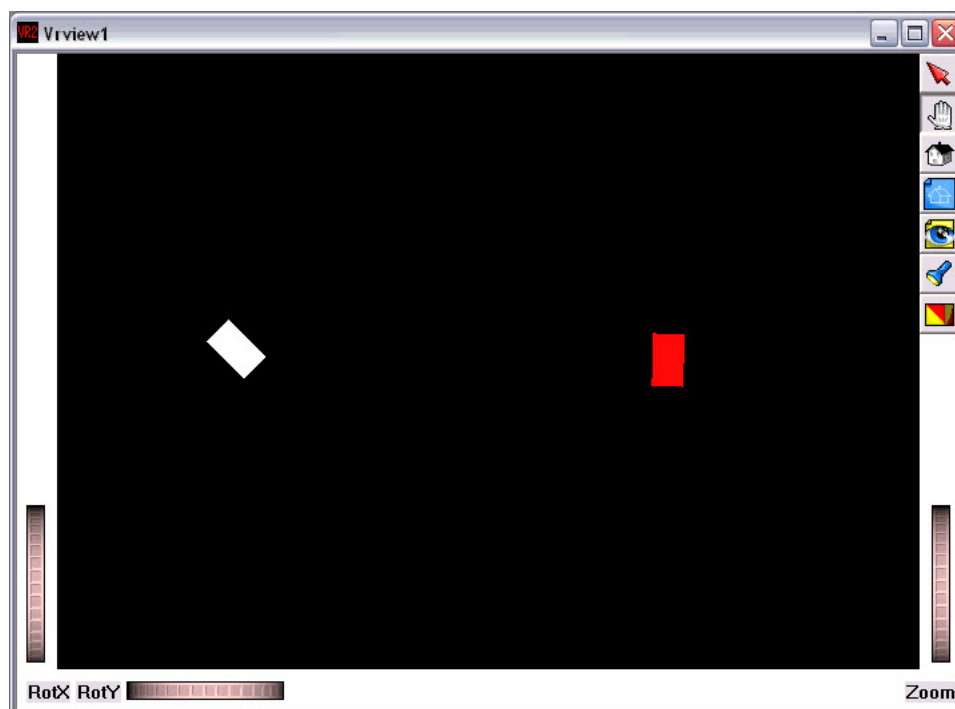
Ο ερυθρός κινούμενος πράκτορας ευρίσκεται αρχικά στη θέση  $(x, y) = (5.0, 0.0)$  και το άνυσμα της ταχύτητάς του είναι:

$$\vec{V}_M = \{-0.005, 0.000\}, \text{ όπου } \|\vec{V}_M\|_{max} = 0.005$$

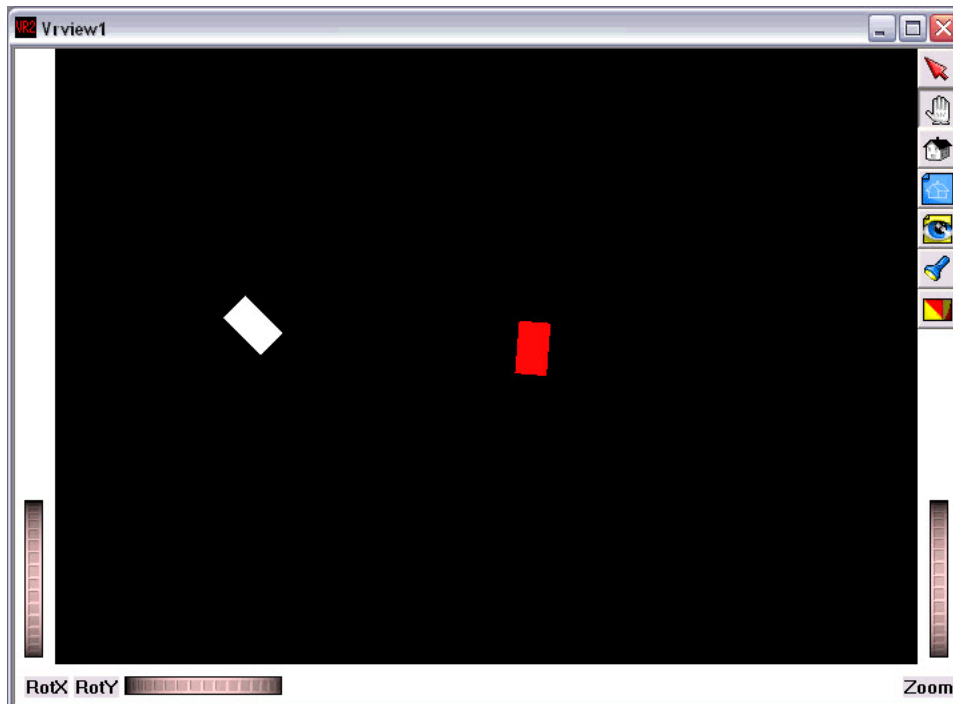
Ο κινούμενος πράκτορας κατευθύνεται αρχικά με σταθερή πορεία και ταχύτητα και ασύμβατα με το άνυσμα της ταχύτητας του στόχου, αλλά όταν φθάσει σε απόσταση ενεργοποίησης



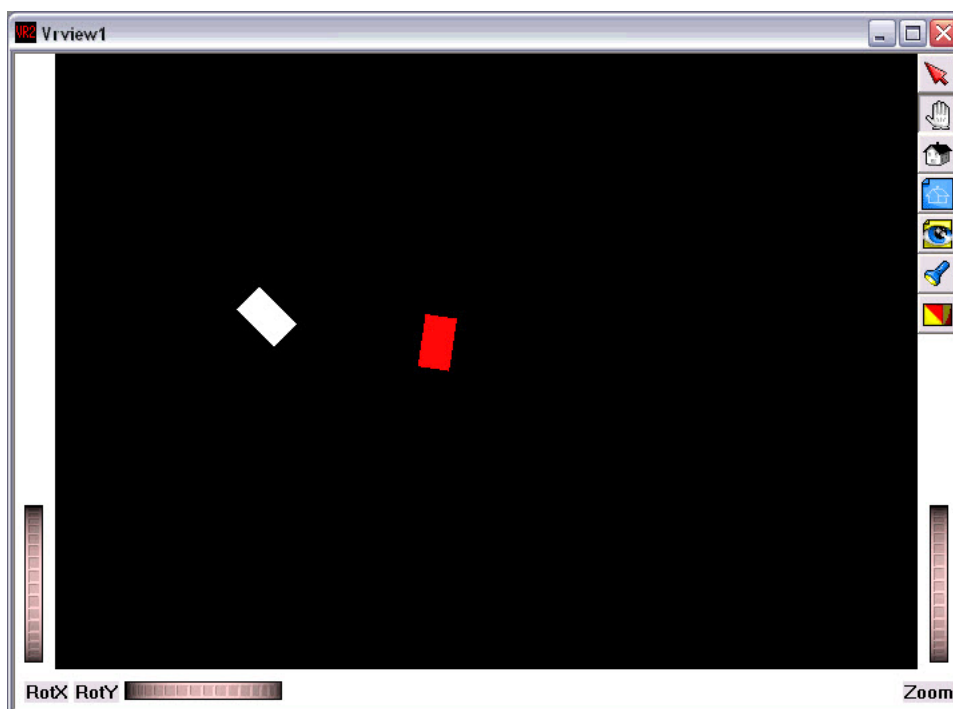
Σχήμα 4.10: Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Έναρξη



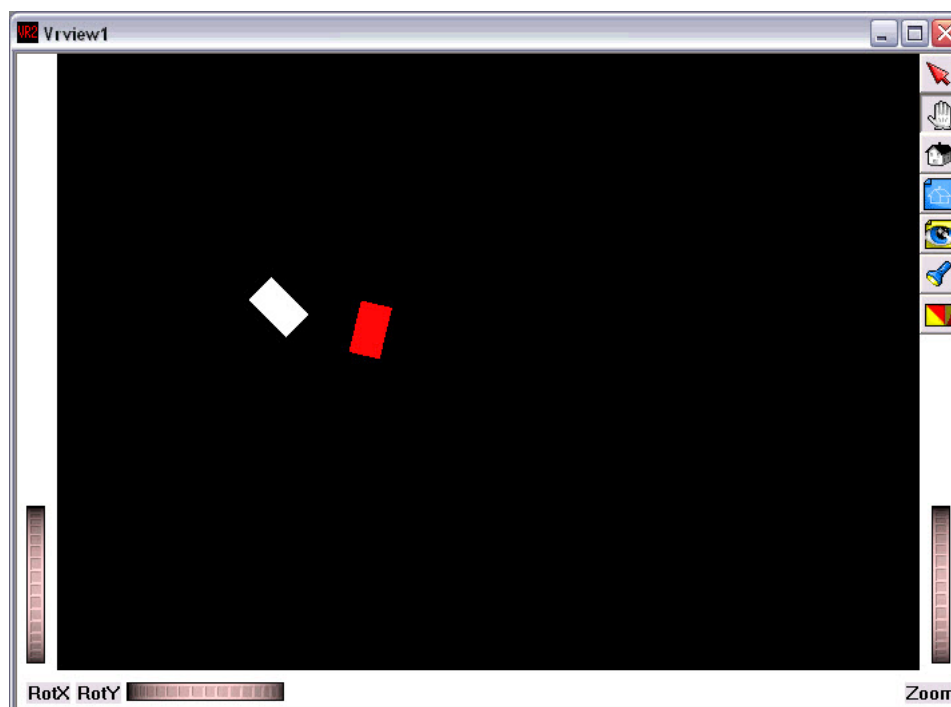
Σχήμα 4.11: Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Τυχαία Κίνηση



Σχήμα 4.12: Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Ενεργοποίηση



Σχήμα 4.13: Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Ενδιάμεση Προσέγγιση



Σχήμα 4.14: Καθοδηγητική Συμπεριφορά "Καταδίωξη" - Τέλος Χειρισμού

του αλγόριθμου "Καταδίωξη" που στη συγκεκριμένη περίπτωση είναι  $r = 5$  μήκη, τότε αρχίζει τον ελιγμό καταδίωξης (Σχήμα 4.12) εκτελώντας μία δεξιά στροφή και προσεγγίζει σταδιακά τον πράκτορα στόχο (Σχήματα 4.13 έως 4.14).

## 4.3 Διαφυγή

### 4.3.1 Περιγραφή

Η "Διαφυγή" είναι ανάλογη με την "Καταδίωξη" με τη διαφορά ότι η "Διαφυγή" χρησιμοποιείται για την καθοδήγηση του οχήματος μακριά από τη μέλλουσα θέση του κινούμενου στόχου.

### 4.3.2 Ψευδοκώδικας

Η συμπεριφορά "Διαφυγή" είναι ουσιαστικά το αντίστροφο της συμπεριφοράς "Καταδίωξη" και προσαρμόζει τη θέση του οχήματος ώστε το διάνυσμα της ταχύτητάς του να είναι ευ-

θυγραμμισμένο και αντίθετο με τη μελλοντική θέση του πράκτορα στόχου. Η διαφυγή του κινούμενου πράκτορα μακριά του πράκτορα στόχου επιτυγχάνεται σε επτά στάδια:

- Αναζήτηση της σχετικής ταχύτητας μεταξύ του πράκτορα στόχου και του κινούμενου πράκτορα με την εξεύρεση του ανύσματος `relative_velocity` δηλαδή του ανύσματος που αντιπροσωπεύει την προσέγγιση μεταξύ των δύο πρακτόρων:

$$relative\_velocity = current\_velocity_T - current\_velocity_M$$

- Αναζήτηση της σχετικής θέσης μεταξύ του πράκτορα στόχου και του κινούμενου πράκτορα με την εξεύρεση του ανύσματος `relative_position`:

$$relative\_position = position_T - position_M$$

- Εξαγωγή του χρονικού διαστήματος `interval` που εκτιμάται ότι θα παρέλθει μέχρι τη στιγμή που ο κινούμενος πράκτορας θα συναντήσει τον πράκτορα στόχο:

$$interval = relative\_position / relative\_velocity$$

- Εξαγωγή της μελλοντικής θέσης του πράκτορα στόχου `future_position` με βάση την παρούσα θέση του και τη θέση που προκύπτει από αναμέτρηση με βάση την ταχύτητα που διαθέτει και το χρονικό διάστημα που θα παρέλθει μέχρι να συναντηθεί με τον κινούμενο πράκτορα:

$$future\_position_T = position_T + velocity_T * interval$$

- Αναζήτηση της επιθυμητής πορείας του κινούμενου πράκτορα με την εξεύρεση του ανύσματος `desired_velocity` δηλαδή του ανύσματος που συνδέει τη θέση του με τη μέλλουσα θέση του κινούμενου πράκτορα στόχου:

$$desired\_velocity_M = future\_position_M - position_T$$

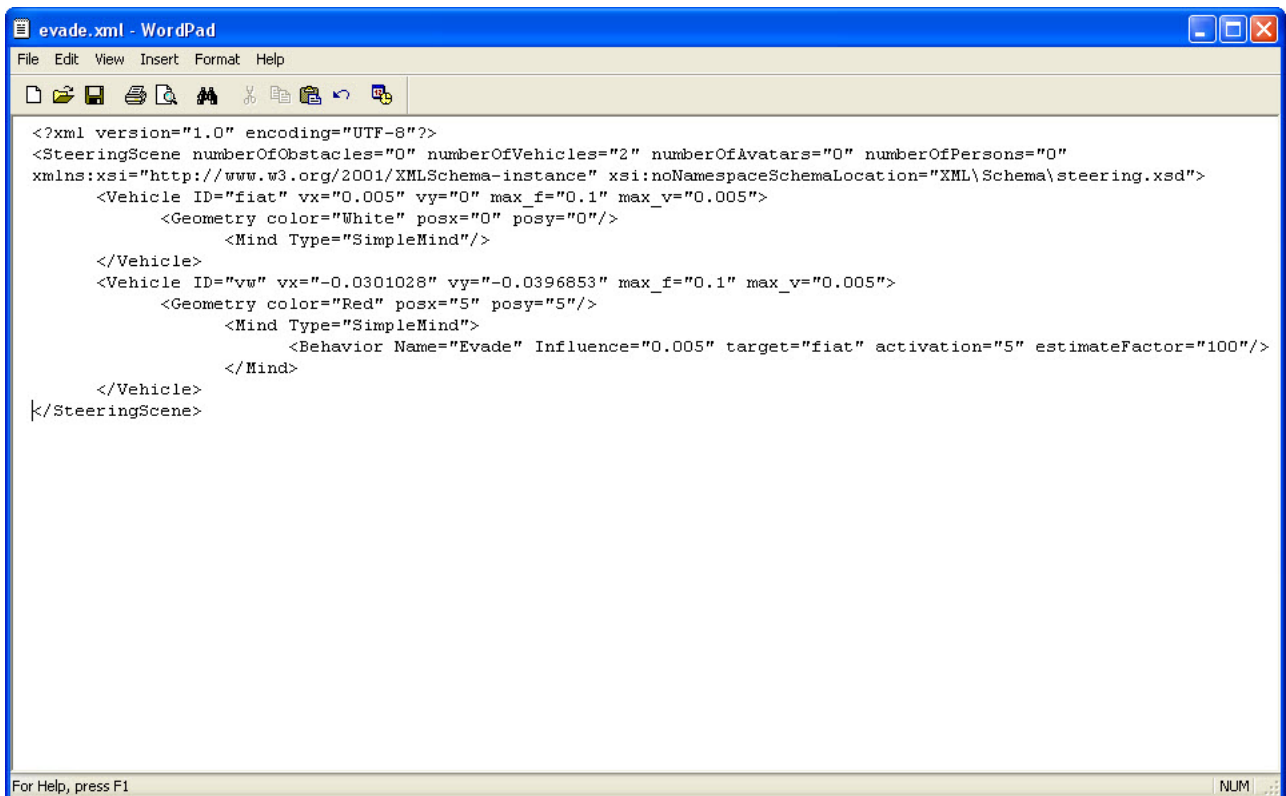


- Κανονικοποίηση του ανύσματος της επιθυμητής πορείας. Το μήκος του ανύσματος μπορεί να είναι η μέγιστη ταχύτητα που διαθέτει ο κινούμενος πράκτορας  $max\_speed$  ή η ταχύτητα χειρισμού του κινούμενου πράκτορα αναλόγως της εφαρμογής:

$$desired\_velocity_M = normalize(future\_position_M - position_T) * max\_speed_M$$

- Πορεία χειρισμού του κινούμενου πράκτορα  $evasion\_steering$ :

$$evasion\_steering_M = desired\_velocity_M - current\_velocity_M$$



```
<?xml version="1.0" encoding="UTF-8"?>
<SteeringScene numberOfObstacles="0" numberOfVehicles="2" numberOfAvatars="0" numberOfPersons="0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XML\Schema\steering.xsd">
  <Vehicle ID="fiat" vx="0.005" vy="0" max_f="0.1" max_v="0.005">
    <Geometry color="White" posx="0" posy="0"/>
    <Mind Type="SimpleMind"/>
  </Vehicle>
  <Vehicle ID="vw" vx="-0.0301028" vy="-0.0396853" max_f="0.1" max_v="0.005">
    <Geometry color="Red" posx="5" posy="5"/>
    <Mind Type="SimpleMind">
      <Behavior Name="Evade" Influence="0.005" target="fiat" activation="5" estimateFactor="100"/>
    </Mind>
  </Vehicle>
</SteeringScene>
```

Σχήμα 4.15: Εισαγωγή Στοιχείων Συμπεριφοράς "Διαφυγή"

### 4.3.3 Υλοποίηση

Η καθοδηγητική συμπεριφορά "Διαφυγή" υλοποιήθηκε με βάση τον ψευδοκώδικα που περιγράφηκε στο προηγούμενο εδάφιο. Κατόπιν διαμορφώθηκε σε μία κλάση αντικειμενοστραφούς προγραμματισμού γλώσσας C++ και προσαρμόστηκε στο κέλυφος χειρισμού του

Υλοποίηση Καθοδηγητικών Συμπεριφορών

πυρήνα VRKernel αφού έγιναν οι κατάλληλες αλλαγές στο γραφικό περιβάλλον αλληλεπίδρασης του χρήστη. Η εισαγωγή των βασικών ρυθμίσεων των πρακτόρων που θα λάβουν μέρος στη προσομοίωση σύμφωνα με το αρχείο τύπου XML όπως αυτό παρουσιάζεται στο Σχήμα 4.15.

Στα Σχήματα 4.16 έως 4.20 παρουσιάζεται η διαδοχική εκτέλεση ενός τυπικού σεναρίου με τη χρήση της συμπεριφοράς "Διαφυγή". Σύμφωνα με το σενάριο αυτό, στο κέντρο ενός καρτεσιανού συστήματος συντεταγμένων ευρίσκεται ο λευκός πράκτορας στόχος, ο οποίος και στη προκειμένη περίπτωση το άνυσμα της ταχύτητάς του είναι:

$$\vec{V}_T = \{0.0035355, 0.0035355\}, \text{ όπου } \|\vec{V}_T\|_{max} = 0.005$$

Ο ερυθρός κινούμενος πράκτορας ευρίσκεται αρχικά στη θέση  $(x, y) = (5.0, 0.0)$  και το άνυσμα της ταχύτητάς του είναι:

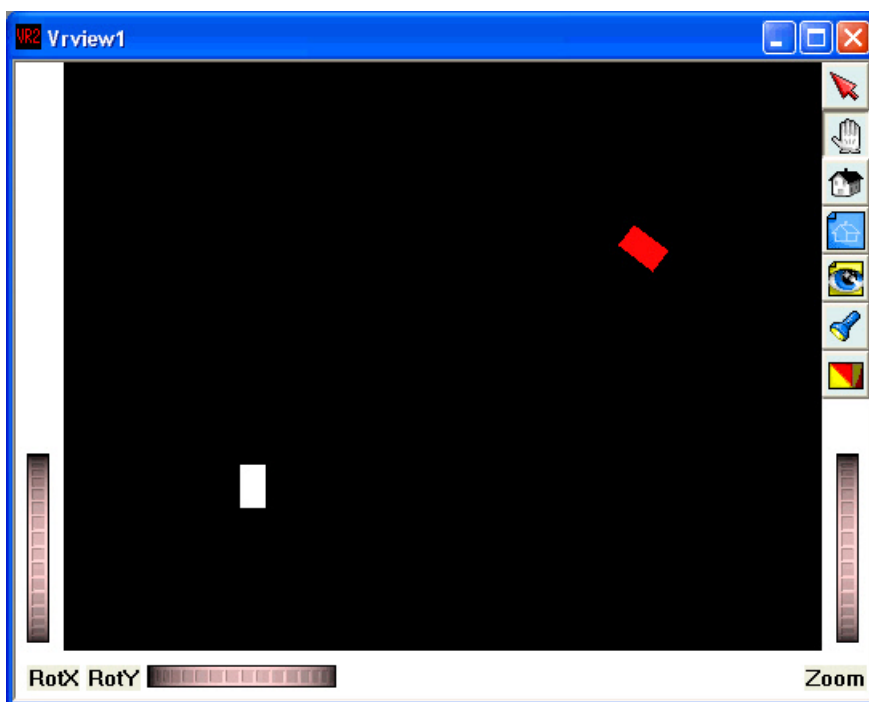
$$\vec{V}_M = \{-0.005, 0.000\}, \text{ όπου } \|\vec{V}_M\|_{max} = 0.005$$

Ο κινούμενος πράκτορας κατευθύνεται αρχικά προς τον λευκό ακίνητο πράκτορα (Σχήματα 4.16 και 4.17) αλλά όταν φθάσει σε απόσταση ενεργοποίησης του αλγόριθμου "Διαφυγή", που στη συγκεκριμένη περίπτωση είναι  $r = 5$  μήκη, τότε αρχίζει τον ελιγμό αποφυγής (Σχήμα 4.18) εκτελώντας μία παραβολοειδή κίνηση απομακρυνόμενος σταδιακά του πράκτορα στόχου (Σχήματα 4.19 έως 4.20).

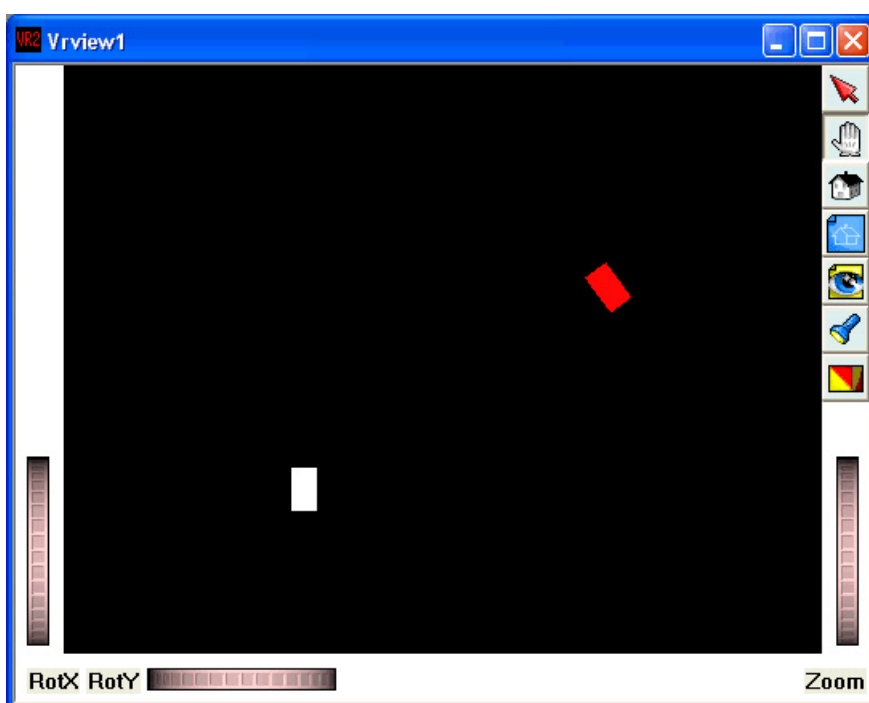
## 4.4 Αποφυγή Εμποδίων

### 4.4.1 Περιγραφή

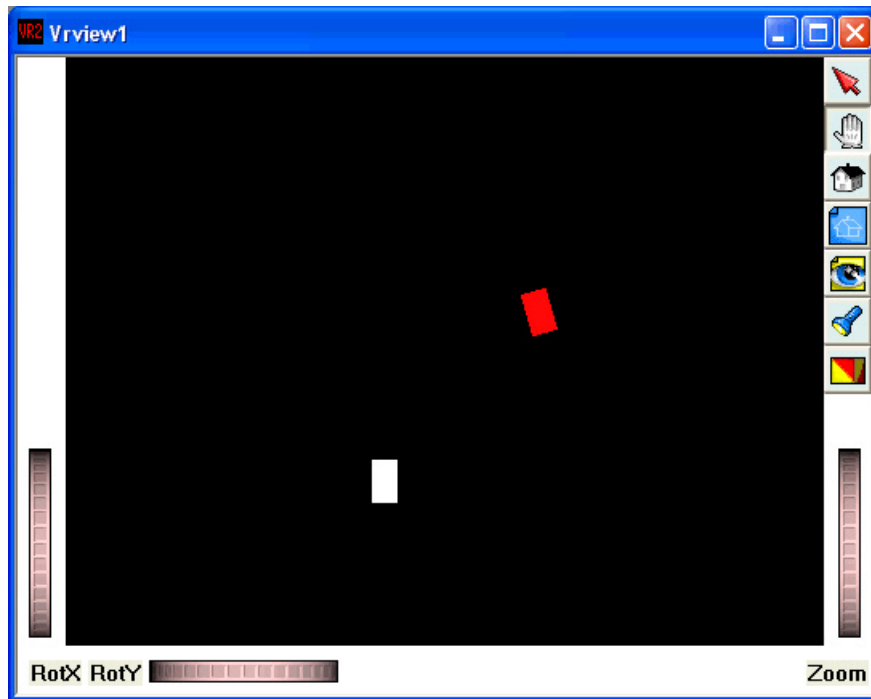
Η "Αποφυγή Εμποδίων" δίνει σε ένα πράκτορα τη δυνατότητα να ελιχθεί σε ένα περιβάλλον διάσπαρτο εμποδίων χωρίς να προσκρούει στα εμπόδια ή σε έτερα κινούμενα οχήματα. Οι συμπεριφορές "Αποφυγή" και "Αποφυγή Εμποδίων" διαφέρουν στο ότι η "Αποφυγή" αναγκάζει τον πράκτορα να χειρίσει μακριά από μία δεδομένη θέση ενώ η "Αποφυγή Εμποδίων" λαμβάνει διορθωτικές κινήσεις μόνο όταν τα γειτονικά εμπόδια κείνται εντός της διεύθυνσης κινήσεως του ή έχουν πορεία συγκρούσεως με αυτόν.



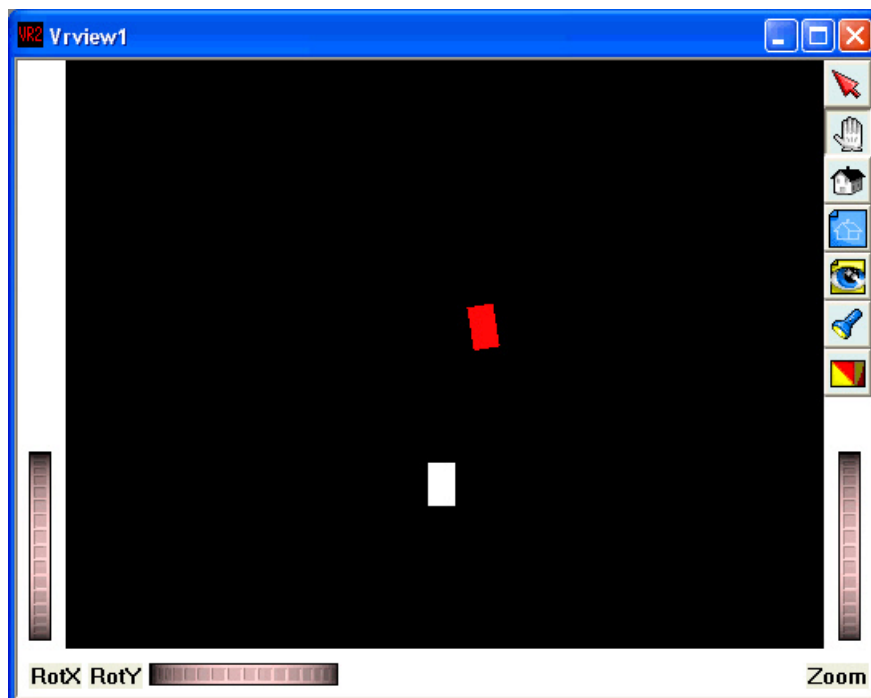
Σχήμα 4.16: Καθοδηγητική Συμπεριφορά "Διαφυγή" - Έναρξη



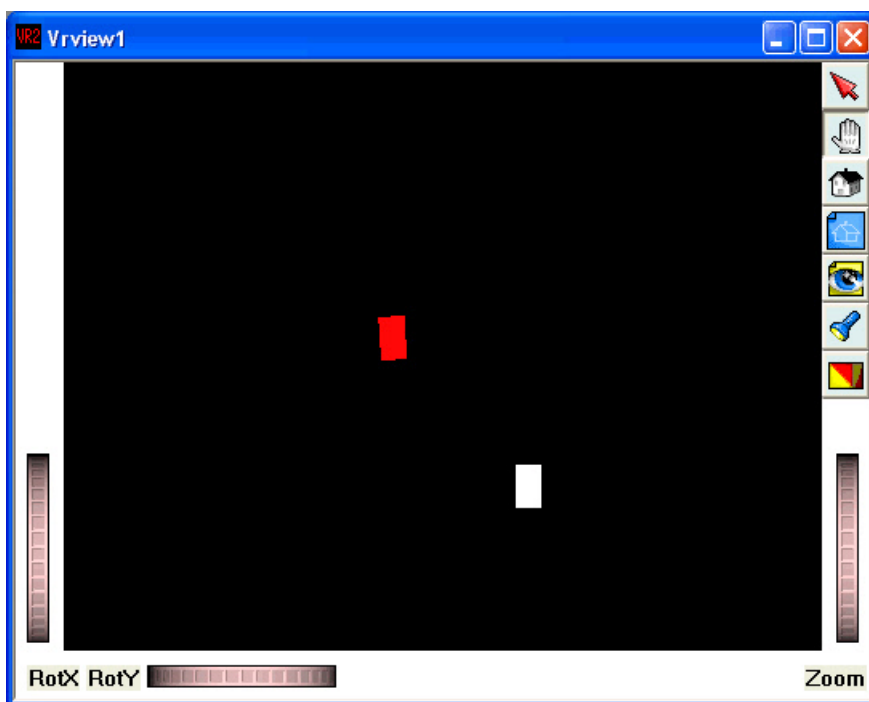
Σχήμα 4.17: Καθοδηγητική Συμπεριφορά "Διαφυγή" - Τυχαία Κίνηση



Σχήμα 4.18: Καθοδηγητική Συμπεριφορά "Διαφυγή" - Ενεργοποίηση



Σχήμα 4.19: Καθοδηγητική Συμπεριφορά "Διαφυγή" - Ενδιάμεση Αποφυγή



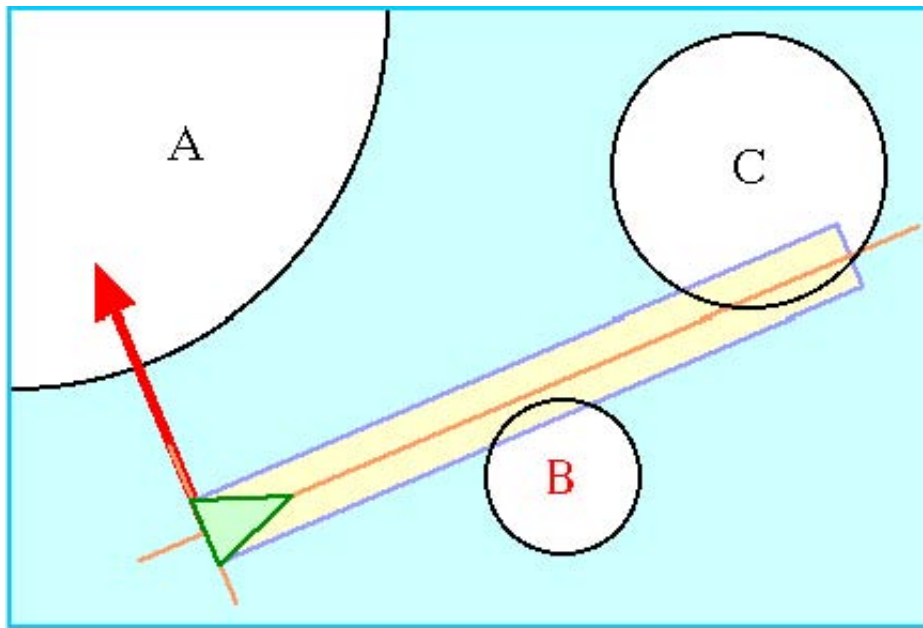
Σχήμα 4.20: Καθοδηγητική Συμπεριφορά "Διαφυγή" - Τέλος Χειρισμού

Η συμπεριφορά που εξετάζουμε μπορεί να αναλυθεί σε δύο ειδών επιμέρους προβλήματα:

- Ο πράκτορας που κινείται εντός ενός συγκεκριμένου πεδίου δεν γνωρίζει εκ των προτέρων τη χωροταξία του, δηλαδή τη τοποθεσία τους στο χώρο τυχόν εμπόδια, ούτε τις διαστάσεις τους αλλά κυρίως δεν γνωρίζει αν εντός του χωρίου που κινείται κυκλοφορούν άλλοι πράκτορες με συγκεκριμένες διαστάσεις και ανύσματα ταχύτητας.
- Έστω ότι γίνεται γνωστή με κάποια μέθοδο η θέση και διάσταση τυχόν εμποδίου (κινητού ή ακίνητου), τότε ποιος πρέπει να είναι ο χειρισμός του πράκτορα ώστε να αποκτήσει καταρχήν μία αληθοφανή πορεία ανάσχεσης και κυρίως να προλάβει να μην προσκρούσει με οποιοδήποτε σημείο του όγκου του με το εμπόδιο;

Το πρώτο πρόβλημα επιλύεται με τη χρήση της βιβλιοθήκης Coin3d και συγκεκριμένα με τη κλάση SoRayPickAction. Η συγκεκριμένη κλάση δημιουργεί μία ακτίνα (κάτι σαν ακτίνα laser με συγκεκριμένο μήκος). Έχει τη δυνατότητα να "προσκολλάται" σε ένα κινούμενο αντικείμενο<sup>3</sup> και να επιστρέφει στον κάτοχό του τυχόν συγκρούσεις του με άλλα αντικείμενα

<sup>3</sup>έννοια αντικειμενοστραφούς προγραμματισμού (object)



Σχήμα 4.21: Αποφυγή Εμποδίων

στο χώρο. Η ιδέα είναι πολύ απλή στη σκέψη και εφαρμογή της και έχει ήδη εφαρμοστεί σε πληθώρα εφαρμογών γραφικών. Μπορούμε να την προσομοιώσουμε με ένα άνθρωπο που κινείται στο σκοτάδι και κρατά ένα φακό με πολύ στενό λοβό φωτισμού και ανιχνεύει μόνο τα αντικείμενα που η φωτεινή δέσμη του φακού προσπίπτει επάνω τους. Βέβαια παραμένει το πρόβλημα του όγκου του κινούμενου πράκτορα και συγκεκριμένα το ερωτηματικό που ανακύπτει ευθύς με την εφαρμογή της ιδέας της ακτίδας: σε ποιο σημείο του όγκου του κινούμενου πράκτορα πρέπει να προσαρμοσθεί η ακτίδα ώστε να καλύψουμε κάθε πιθανή περίπτωση πρόσκρουσης του με ένα εμπόδιο. Αν προσαρμόσουμε την ακτίδα στο κέντρο βάρους του κινούμενου πράκτορα τότε σε περίπτωση που αυτός δεν είναι σημειακός αλλά έχει κάποιες διαστάσεις, υπάρχει μεγάλη πιθανότητα μία από τις πλευρές του να προσκρούσουν σε κάποιο εμπόδιο.

Στην περίπτωση που εξετάζουμε θεωρήθηκε ως ικανοποιητική προσέγγιση η τοποθέτηση μία ακτίδας στο κέντρο του όγκου του πράκτορα και από μία ακίδα στις δύο άκρες του πράκτορα που είναι παράλληλες με την πορεία του. Αυτό γίνεται γιατί οι πράκτορες στην περίπτωση της προσομοίωσης που εξετάζουμε είναι πρισματικού σχήματος, ορθογώνια παραλληλεπίπεδα με τη μικρή τους πλευρά παράλληλη με την εκάστοτε πορεία τους. Συνεπώς, μπορούμε να

προσαρμόσουμε από μία ακίδα σε κάθε πλευρά του πράκτορα και να εξασφαλίσουμε κατά κάποιο τρόπο την αποφυγή της σύγκρουσής του με κάποιο εμπόδιο.

Το δεύτερο πρόβλημα είναι πιο σύνθετο και στην έκταση της διπλωματικής εργασίας που εξετάζουμε προτάθηκαν τρία μοντέλα που έχουν σαν σκοπό την όσο πιο δυνατόν ομαλή πορεία ανάσχεσης αλλά και πιο δραστική ταυτόχρονα.

Το πρόβλημα διερευνήθηκε καταρχήν για την ύπαρξη οριακών συνθηκών που θα μπορούσαν να το καταστήσουν δυσεπίλυτο και προέκυψαν οι ακόλουθες περιπτώσεις που δύναται να χαρακτηριστούν οριακές:

- Ο κινούμενος πράκτορας κινείται με πορεία που είναι εντελώς κάθετη σε ένα εμπόδιο και θα πρέπει να χειρίσει ομαλά για την αποφυγή του<sup>4</sup>.
- Οι αλλαγές πορείας και γενικώς οι χειρισμοί του κινούμενου πράκτορα θα πρέπει να είναι πιο δραστικοί όταν η απόσταση με το εμπόδιο είναι μικρή και λιγότερο όταν η απόσταση είναι μεγαλύτερη. Συνεπώς, καθώς ο πράκτορας πλησιάζει ένα εμπόδιο με πορεία συγκρούσεως και ξεπεράσει την απόσταση ενεργοποίησης της συμπεριφοράς, θα πρέπει να αρχίζει να χειρίζεται μεταβάλλοντας ελάχιστα τα στοιχεία της κίνησής του και να συνεχίζει να τα μεταβάλλει σε διαρκώς αυξανόμενο ποσοστό μέχρι να εξαλειφθεί ο κίνδυνος σύγκρουσης.
- Ένας πράκτορας κινείται σχεδόν εφαπτόμενα με έναν τοίχο-εμπόδιο και προς το παρόν δεν ισχύουν τα κριτήρια συγκρούσεως. Όμως ο πράκτορας αποφασίζει να στρέψει προς το εμπόδιο χωρίς να υπάρχουν τα απαραίτητα χρονικά περιθώρια χειρισμού με αποτέλεσμα η σύγκρουση να είναι αναπόφευκτη.

Οι οριακές συνθήκες που μόλις περιγράψαμε θα οριοθετήσουν και το πρόβλημά μας εφόσον είναι προφανές ότι οποιαδήποτε ενδιάμεση κατάσταση μπορεί και αντιμετωπίζεται στα πλαίσια όχι μόνο της αληθοφάνειας αλλά και των κανόνων της φυσικής. Οι λύσεις που προτείνονται εξετάστηκαν στη πράξη με συγκεκριμένα σενάρια κίνησης και χωροθέτησης

---

<sup>4</sup>Η έννοια "ομαλά" περιγράφει την αποφυγή δραστικών αλλαγών πορείας (άνω των 30-40 μοιρών) μεταξύ δύο χρονικών στιγμιότυπων που χαρακτηρίζονται στη γλώσσα των γραφικών ως ανώμαλες μεταπτώσεις (jitter)

εμποδίων και τα αποτελέσματα κρίθηκαν ικανοποιητικά και ως προς την αληθοφάνεια αλλά και ως προς τη φυσική τους αντιμετώπιση. Βέβαια, η πλατφόρμα εικονικής πραγματικότητας της οποίας ένα μικρό τμήμα περιλαμβάνει τις συγκεκριμένες συμπεριφορές, είναι αρκετά ευέλικτη και διαθέτει πληθώρα ρυθμίσεων που δύνανται να φέρουν την κίνηση του κινούμενου πράκτορα σε υψηλά επίπεδα αληθοφάνειας.

- Η πρώτη οριακή συνθήκη αντιμετωπίστηκε επαρκώς με την μεταβάλλοντας το ήδη υπάρχον άνυσμα της ταχύτητας του κινούμενου πράκτορα με την πρόσθεση ενός ανύσματος ελάχιστου μέτρου αλλά με διαφορά 90 μοιρών. Με την ενέργεια αυτή ο πράκτορας αποκλίνει ελάχιστα με την κάθετη στην επιφάνεια του εμποδίου οπότε από εδώ και πέρα ο χειρισμός του εμπίπτει στο γενικότερο πλαίσιο της θεώρησης της συμπεριφοράς "Αποφυγή Εμποδίων". Η κίνηση στην περίπτωση αυτή γίνεται ασυμπτωτική με την επιφάνεια και το άνυσμα της ταχύτητας μεταβάλλεται σταδιακά κατά 90 μοίρες αριστερά ή δεξιά. Βέβαια η κίνηση του κινούμενου πράκτορα εξαρτάται πολύ από το μέτρο της ταχύτητάς του και κυρίως από την απόσταση που η συγκεκριμένη συμπεριφορά ενεργοποιείται. Η λογισμική εφαρμογή της συμπεριφοράς δεν περιλαμβάνει τον αυτόματο συσχετισμό των μεταβλητών που αναφέρθηκαν για την ολοκληρωτική εξάλειψη του προβλήματος με κατάλληλες ενέργειες (στα πλαίσια μία αυξημένης τεχνητής νοημοσύνης του πράκτορα) καθόσον κάτι τέτοιο ξεφεύγει των χρονικών περιορισμών και απαιτήσεων εμβάθυνσης της παρούσας μελέτης.
- Οι αλλαγές πορείας του πράκτορα σε σχέση με την απόσταση εντοπισμού του εμποδίου καθώς και ο κατάλληλος βαθμός αποφασιστικότητας αντιμετωπίστηκαν με την υιοθέτηση τριών συντελεστών που μεταβάλλουν κατά διαφορετικό τρόπο ο καθένας τους την διόρθωση πορείας που υπολογίστηκε για την συγκεκριμένη κατάσταση. Οι συντελεστές αυτοί είναι:
  - $1/closest\_obstacle\_Distance$  όπου  $closest\_obstacle\_Distance$  είναι η απόσταση του κινούμενου στόχου από το εμπόδιο. Εδώ αντιμετωπίζεται επιτυχώς το πρόβλημα της αποφασιστικότητας με βαθμό αντιστρόφως ανάλογο της απόστασης



αλλά παραμένει έως ένα βαθμό η ανωμαλία στις μεταπτώσεις της θέσης του πράκτορα (jitter).

- $1/closest\_obstacle\_Distance^2$  όπου  $closest\_obstacle\_Distance$  είναι η απόσταση του κινούμενου στόχου από το εμπόδιο. Εδώ αντιμετωπίζεται επιτυχώς το πρόβλημα της αποφασιστικότητας με βαθμό αντιστρόφως ανάλογο του τετραγώνου της απόστασης και σε σημαντικό βαθμό η ανωμαλία στις μεταπτώσεις της θέσης του πράκτορα (jitter).
- $(max\_threshold - closest\_obstacle\_Distance) / (max\_threshold - min\_threshold)$  όπου εισάγονται δύο επιπλέον έννοιες, η  $max\_threshold$  και  $min\_threshold$  που αντίστοιχα οριοθετούν την μέγιστη απόσταση για την ενεργοποίηση της συμπεριφοράς "Αποφυγή Εμποδίου" και την ελάχιστη απόσταση αντίστοιχα για την λήψη ενός πολύ αποφασιστικού χειρισμού όπως η αλλαγή πορείας έως και 90 μοιρών για την αποφυγή του εμποδίου την τελευταία στιγμή. Στην περίπτωση αυτή η ομαλότητα στην απεικόνιση της θέσης και κατεύθυνσης του πράκτορα θυσιάζεται προς όφελος της αναγκαιότητας της ελάχιστης απόστασης και περίπου ισοδυναμεί με αυτήν της πρώτης περίπτωσης.

- Στην περίπτωση όπου ένας πράκτορας κινείται σχεδόν εφαπτόμενα με την επιφάνεια ενός εμποδίου τότε ο χειρισμός αποφυγής του εμποδίου επιτυγχάνεται με το συνδυασμό δύο μεθόδων που ακολουθήθηκαν προηγουμένως για την επίλυση άλλων προβλημάτων. Καταρχήν υιοθετούμε ως συντελεστή διόρθωσης τον:

$$(max\_threshold - closest\_obstacle\_Distance) / (max\_threshold - min\_threshold)$$

όπου εξασφαλίζεται ότι θα υπάρξει χειρισμός ασφαλείας για την περίπτωση της ελάχιστης απόστασης εντοπισμού αλλά και τοποθέτηση σε κάθε όριο του κινούμενου πράκτορα από μία ακτίδα εντοπισμού ώστε να έχω και την ταχύτερη συνέγερση σε περίπτωση που ο πράκτορας πλησιάζει το εμπόδια με πολύ μικρή γωνία προσέγγισης.

#### 4.4.2 Ψευδοκώδικας

Συνεπώς, η συμπεριφορά "Αποφυγή Εμποδίων" θα προσπαθεί να προσαρμόζει τη θέση του κινούμενου πράκτορα έτσι ώστε το διάνυσμα της ταχύτητάς του να μη διέρχεται καθ' οιονδήποτε τρόπο διαμέσου ενός εμποδίου. Η αποφυγή των εμποδίων στόχων επιτυγχάνεται σε επτά στάδια:

- Αναζήτηση πιθανόν εμποδίου κατά την έμπροσθεν κίνηση του κινούμενου πράκτορα με τη χρήση της κλάσης `SoRayPickAction`:

$$closest\_obstacle = SoRayPickAction_M()$$

- Αναζήτηση της επιθυμητής πορείας του κινούμενου πράκτορα με την εξεύρεση του ανύσματος `desired_velocity` δηλαδή του ανύσματος που συνδέει τη θέση του με τη θέση του ακίνητου πράκτορα στόχου:

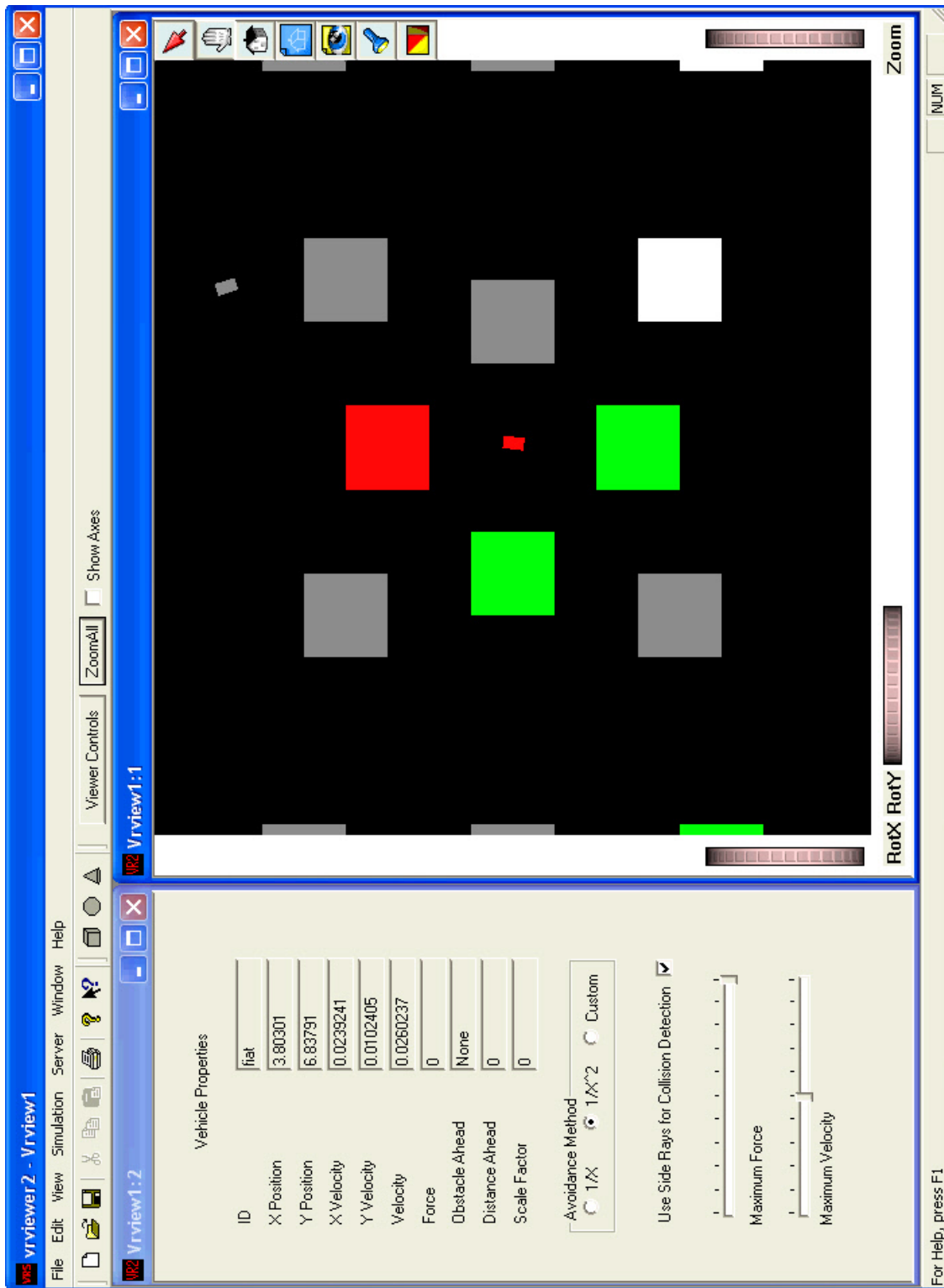
$$desired\_velocity_M = position_M - position_{obstacle}$$

- Κανονικοποίηση του ανύσματος της επιθυμητής πορείας. Το μήκος του ανύσματος μπορεί να είναι η μέγιστη ταχύτητα που διαθέτει ο κινούμενος πράκτορας `max_speed` ή η ταχύτητα χειρισμού του κινούμενου πράκτορα αναλόγως της εφαρμογής πολλαπλασιασμένη με ένα συντελεστή που δείχνει τη δραστηριότητα του χειρισμού:

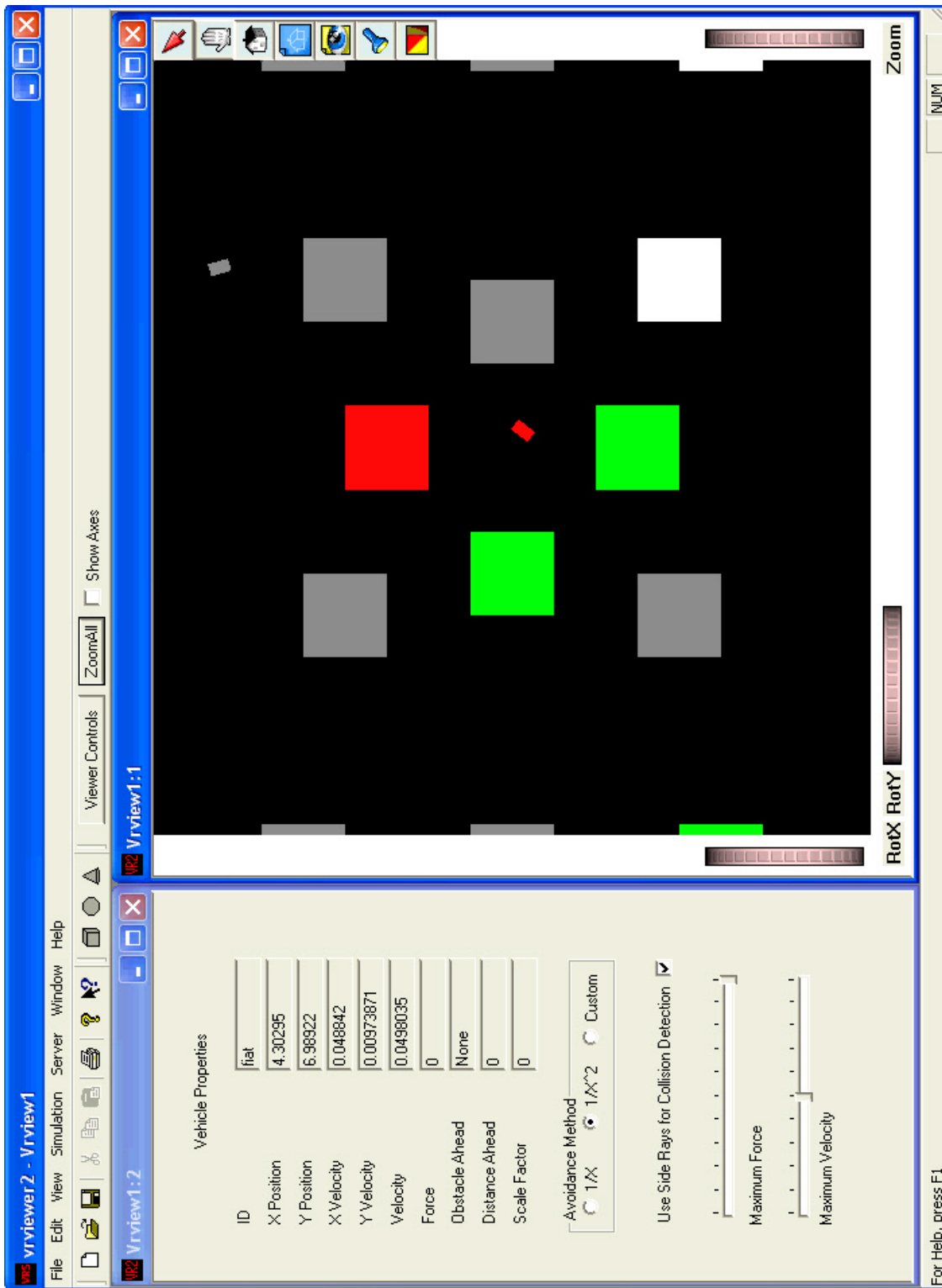
$$desired\_velocity_M = normalize(position_M - position_{obstacle}) * max\_speed_M * coef$$

- Πορεία χειρισμού του κινούμενου πράκτορα `obstacle_avoidance_steering`:

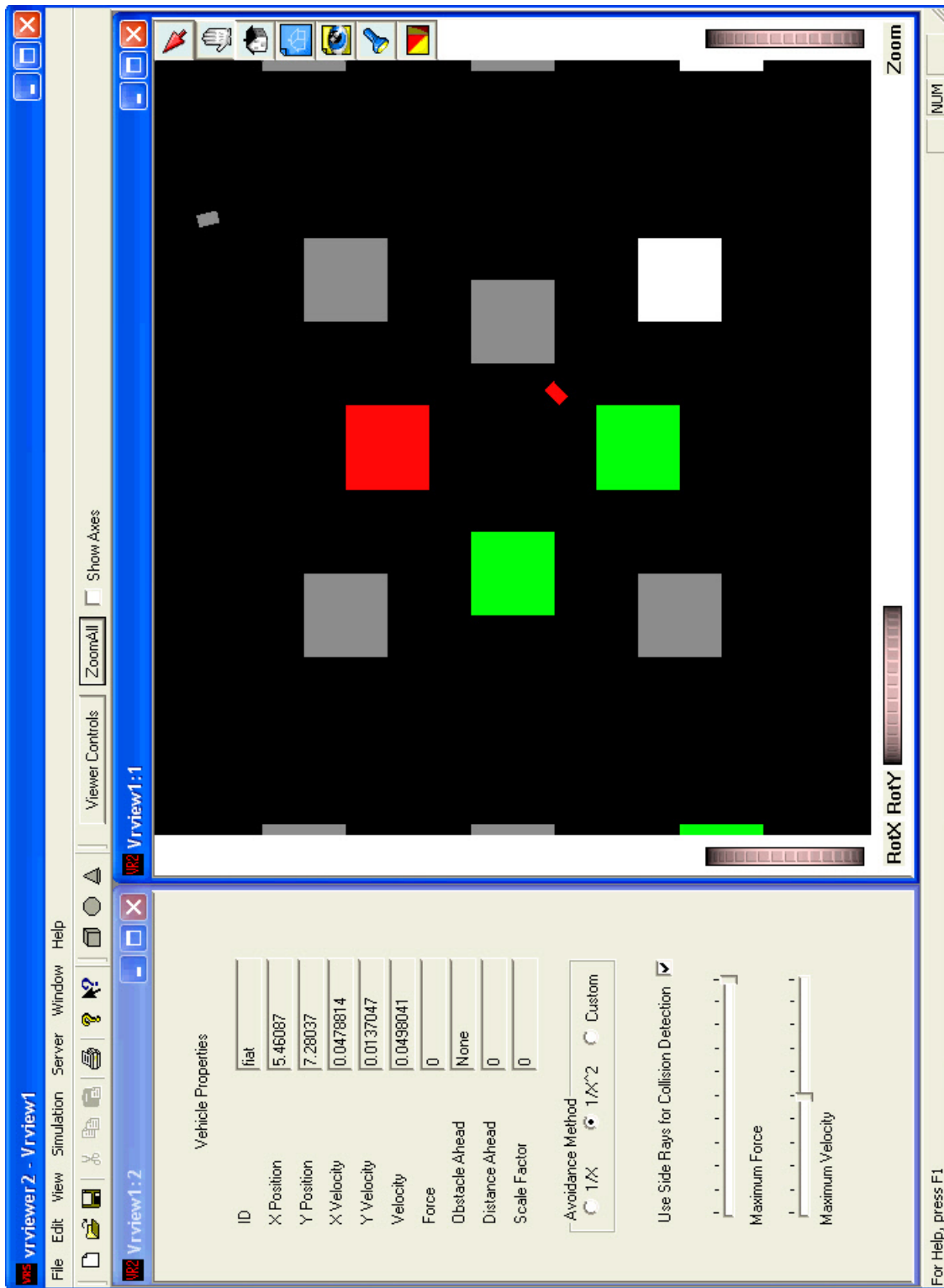
$$obstacle\_avoidance\_steering_M = desired\_velocity_M - current\_velocity_M$$



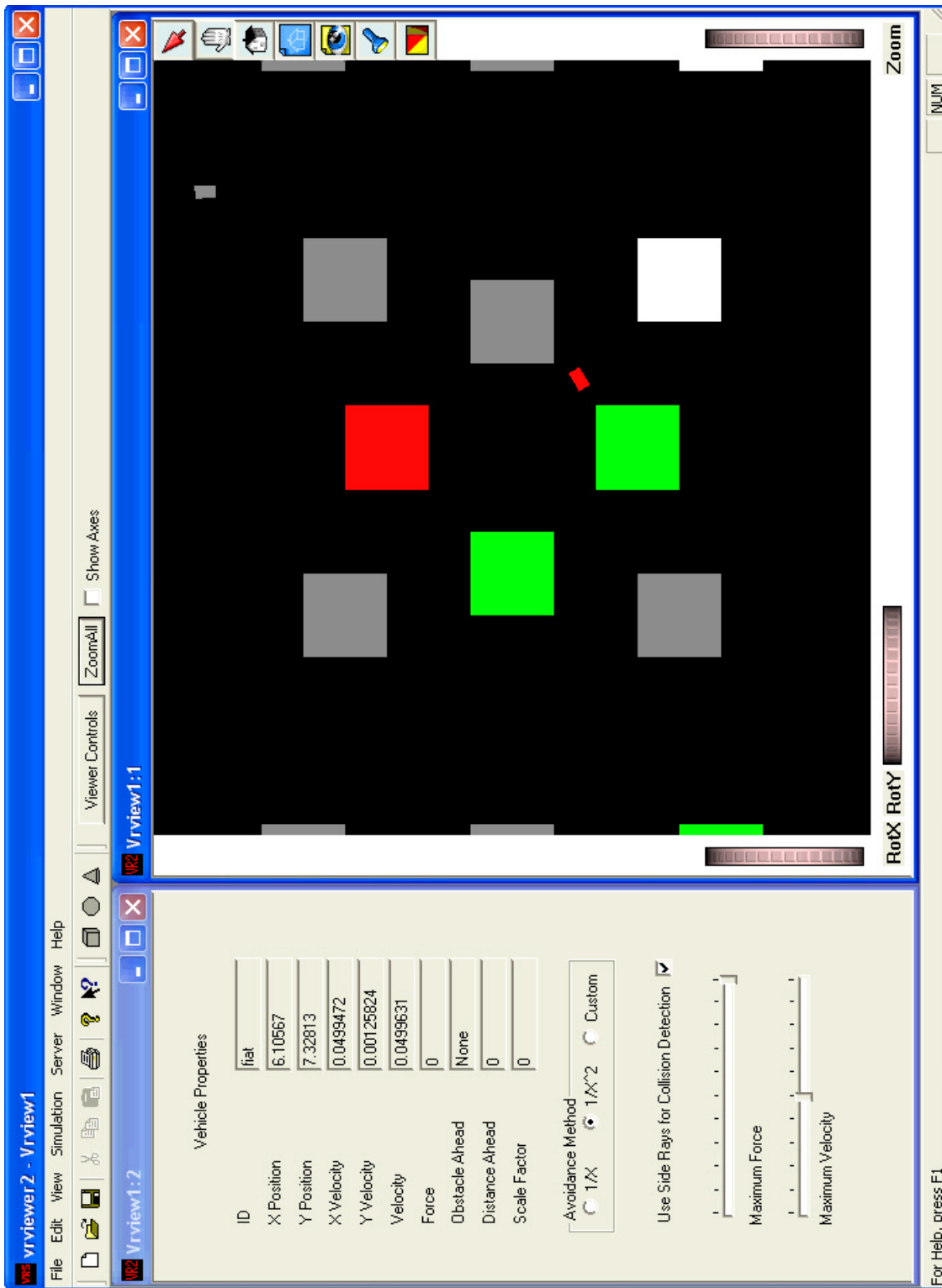
Σχήμα 4.22: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Έναρξη & Τυχαία Κίνηση



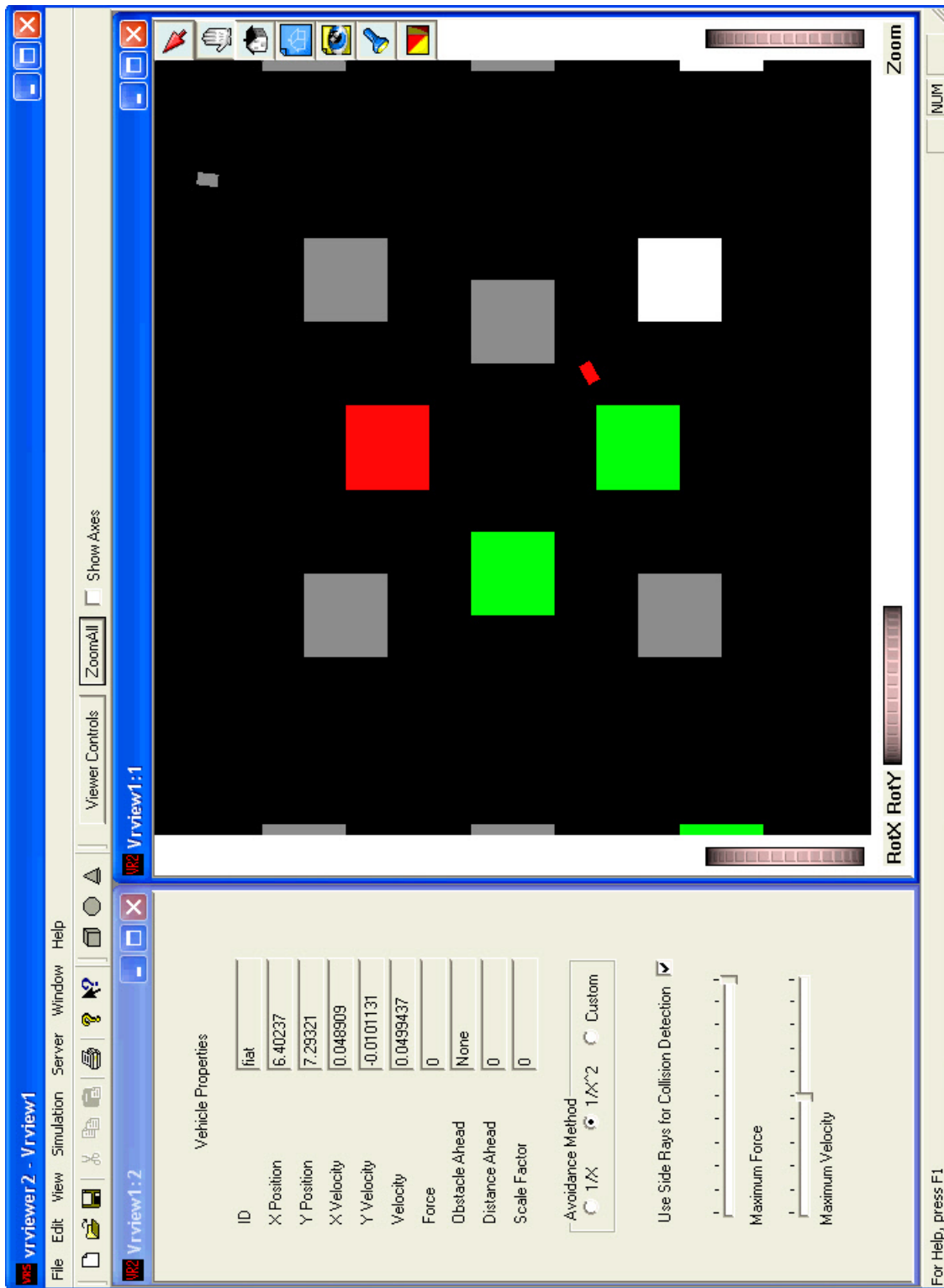
Σχήμα 4.23: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενεργοποίηση



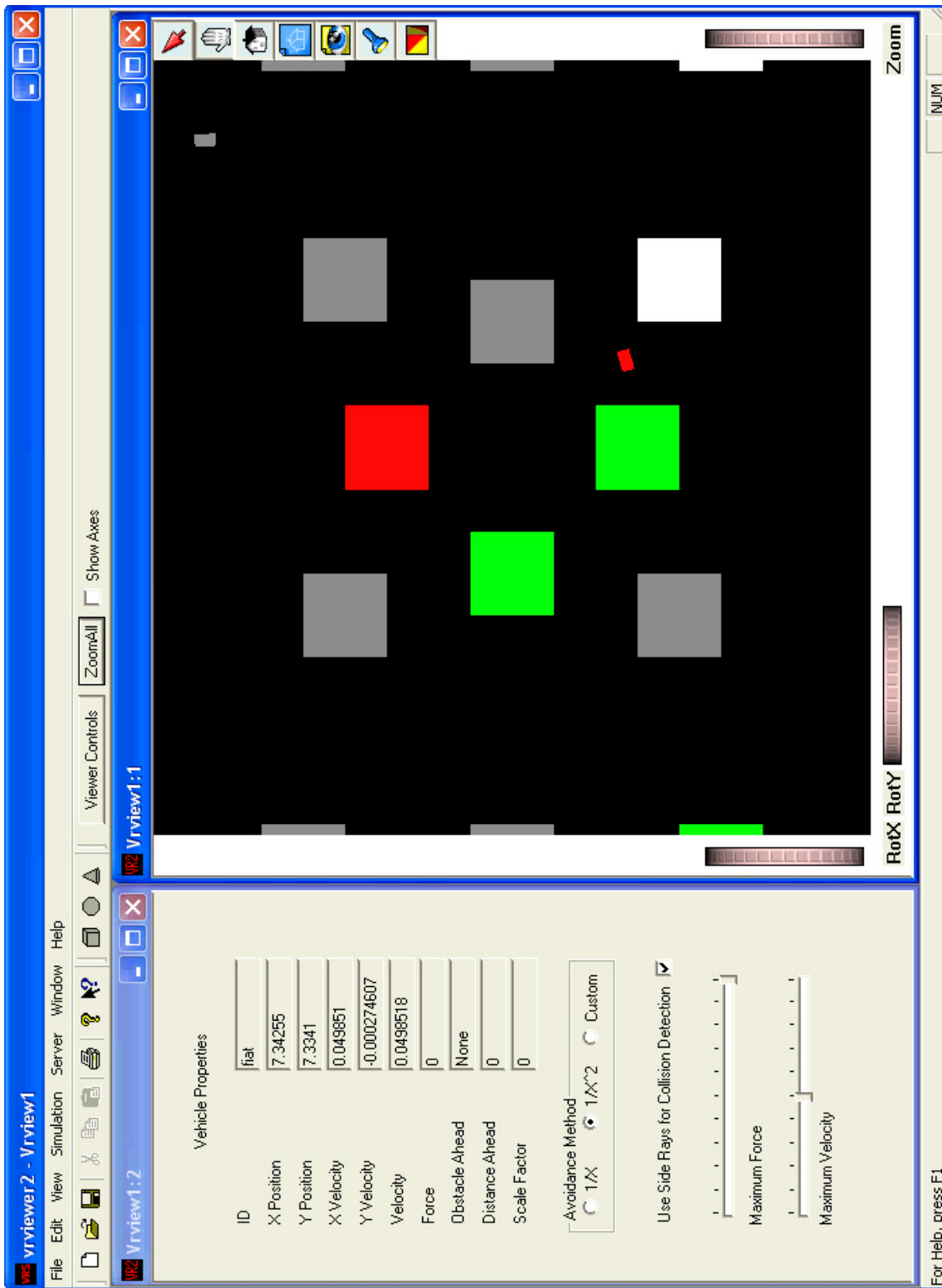
Σχήμα 4.24: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Αρχικός Χερισιός



Σχήμα 4.25: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός 1

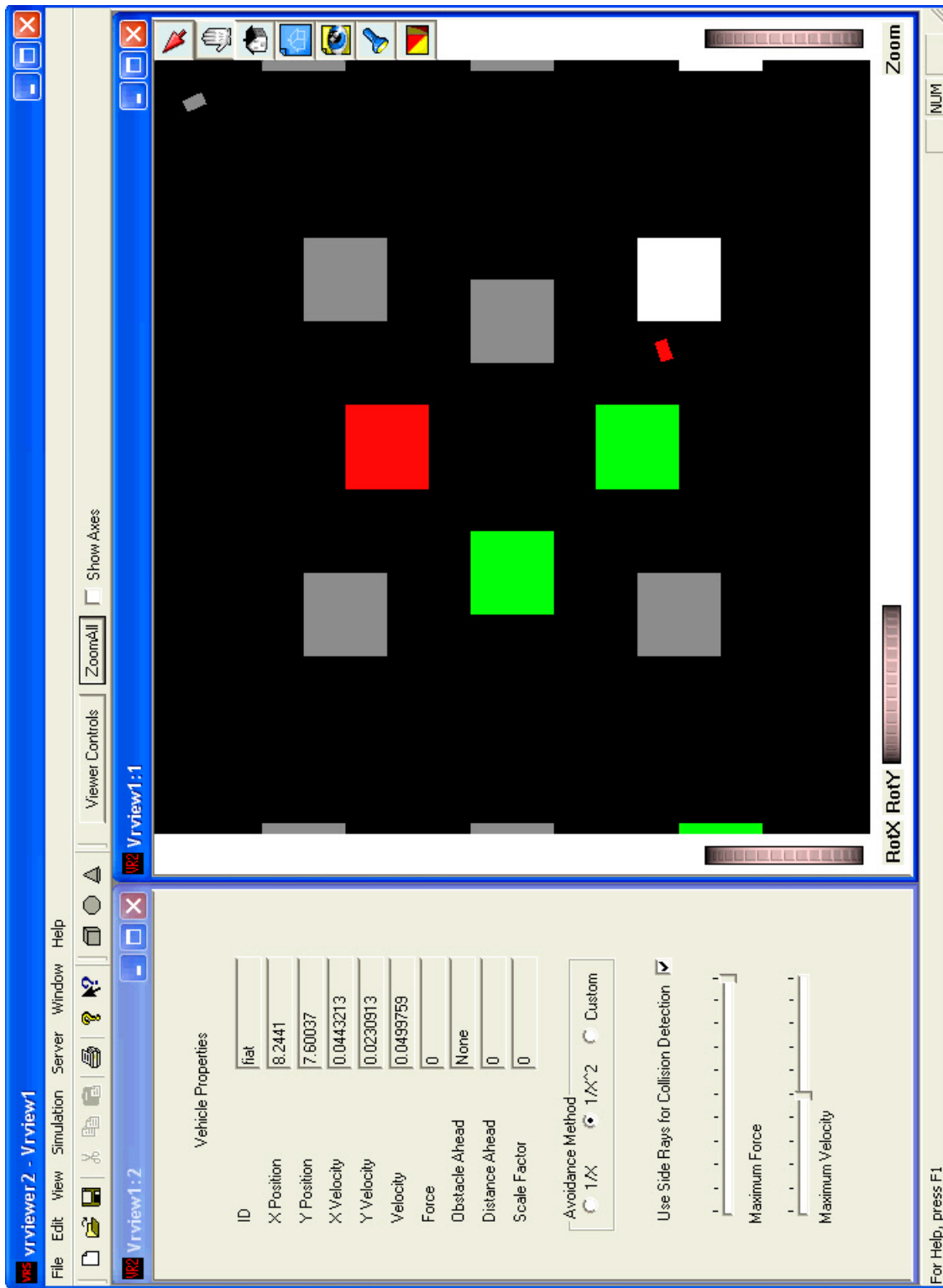


Σχήμα 4.26: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χερσιγίος II

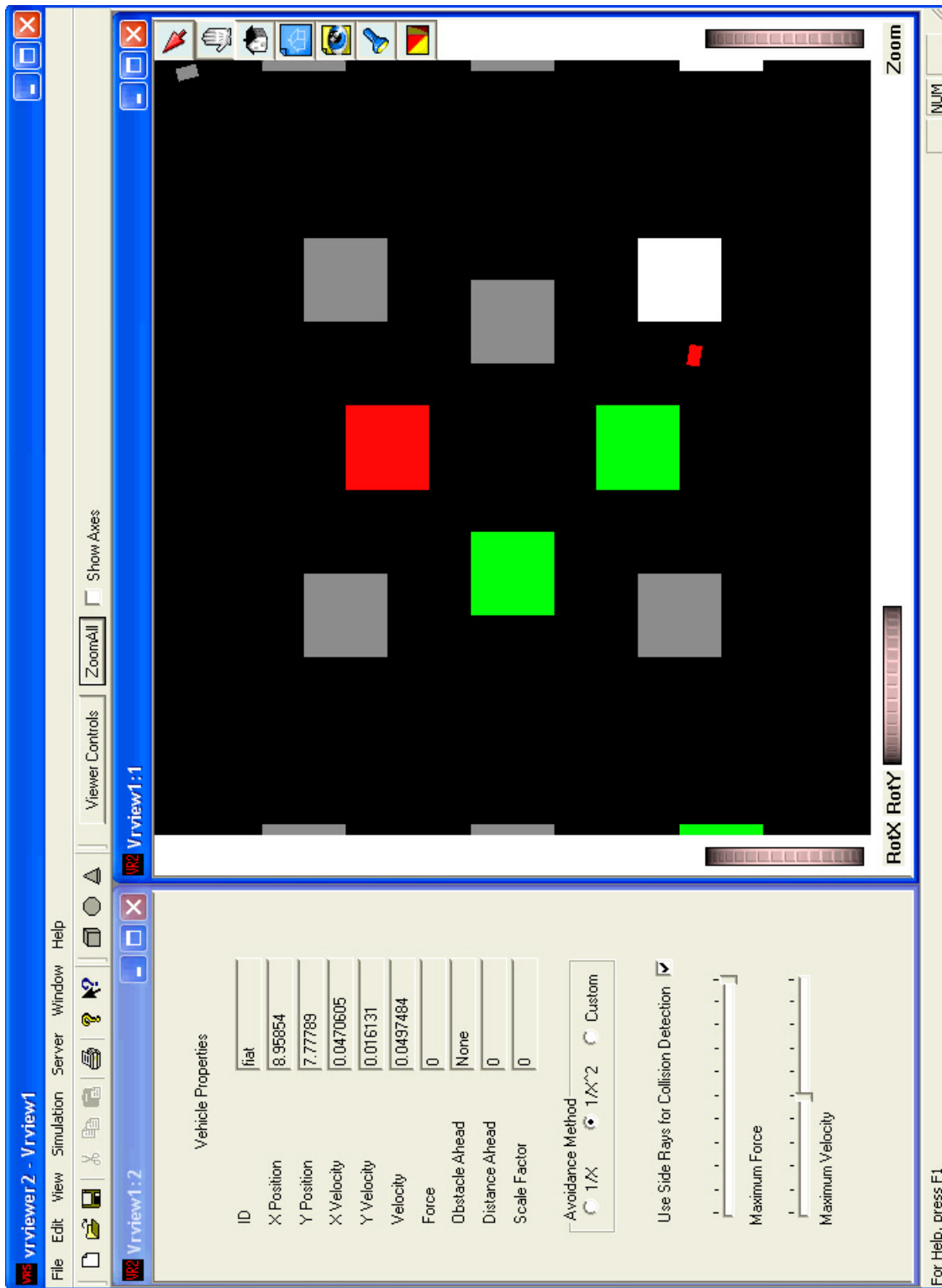


Σχήμα 4.27: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός III

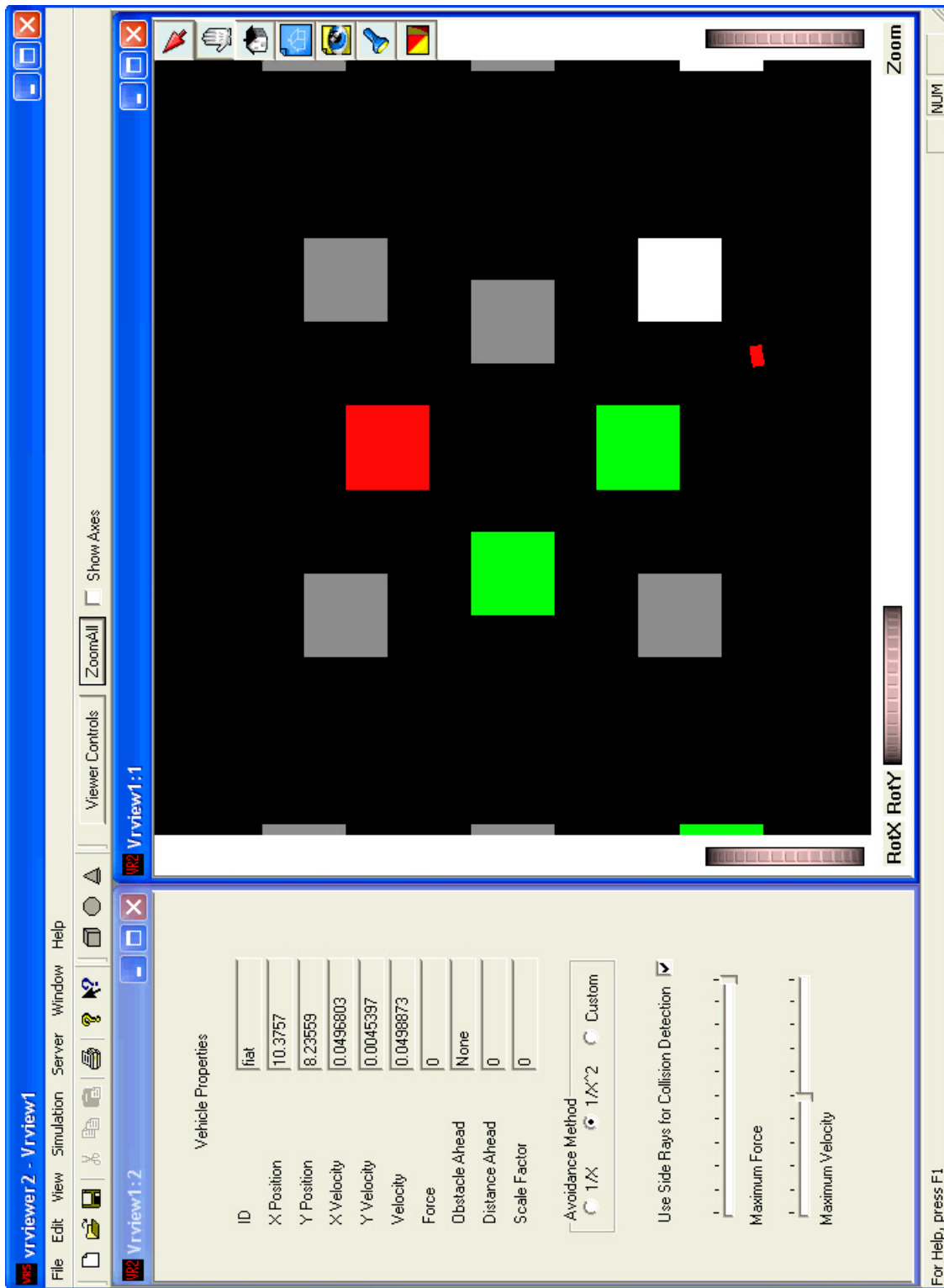




Σχήμα 4.28: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χειρισμός IV



Σχήμα 4.29: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Ενδιάμεσος Χερσιμιάς V



Σχήμα 4.30: Καθοδηγητική Συμπεριφορά "Αποφυγή Εμποδίων" - Τέλος Χειρισμού



### 4.4.3 Υλοποίηση

Η καθοδηγητική συμπεριφορά "Αποφυγή Εμποδίων" υλοποιήθηκε με βάση τον ψευδοκώδικα που περιγράφηκε στο προηγούμενο εδάφιο. Κατόπιν διαμορφώθηκε σε μία κλάση αντικειμενοστραφούς προγραμματισμού γλώσσας C++ και προσαρμόστηκε στο κέλυφος χειρισμού του πυρήνα VRKernel αφού έγιναν οι κατάλληλες αλλαγές στο γραφικό περιβάλλον αλληλεπίδρασης του χρήστη. Η εισαγωγή των βασικών ρυθμίσεων των πρακτόρων που θα λάβουν μέρος στη προσομοίωση έγινε σύμφωνα με το αρχείο τύπου XML όπως αυτό παρουσιάζεται στο Σχήμα 4.31.

Στα Σχήματα 4.22 έως 4.30 παρουσιάζεται η διαδοχική εκτέλεση ενός τυπικού σεναρίου με τη χρήση της συμπεριφοράς "Αποφυγή Εμποδίων". Σύμφωνα με το σενάριο αυτό, σε ένα πλέγμα εμποδίων ευρίσκεται ο ερυθρός κινούμενος πράκτορας, του οποίου το άνωσμα της ταχύτητας είναι:

$$\vec{V}_M = \{0.0035355, 0.0035355\}, \text{ όπου } \|\vec{V}_M\|_{max} = 0.005$$

Ο κινούμενος πράκτορας κατευθύνεται αρχικά με σταθερή πορεία και ταχύτητα εντός του πλέγματος των εμποδίων, αλλά όταν φθάσει σε απόσταση ενεργοποίησης του αλγόριθμου "Αποφυγή Εμποδίων" που στη συγκεκριμένη περίπτωση είναι  $r = 2$  μήκη, τότε αρχίζει τον ελιγμό αποφυγής (Σχήμα 4.23) εκτελώντας διαδοχικές αλλαγές πορείας μέχρι να αποφύγει τον κίνδυνο της σύγκρουσης (Σχήματα 4.25 έως 4.30).

## 4.5 Ακολουθία Διαδρόμου

### 4.5.1 Περιγραφή

Η "Ακολουθία Διαδρόμου" δίνει τη δυνατότητα στο όχημα να κινηθεί κατά μήκος προκαθορισμένων διαδρόμων. Αυτό βέβαια δεν σημαίνει ότι το όχημα θα κινείται αυστηρά επάνω σε συγκεκριμένη τροχιά αλλά θα δίνει την εντύπωση ανθρώπων που κινούνται σε ένα διάδρομο: οι μεμονωμένοι διάδρομοι κίνησης του κάθε ανθρώπου παραμένουν εγγύς και συχνά παράλληλοι με τη κεντρική γραμμή του διαδρόμου αλλά ταυτόχρονα έχουν τη δυνατότητα

παρέκκλισης. Παραλλαγές της συμπεριφοράς αυτής είναι η "Ακολουθία Τοιχώματος Διαδρόμου" (Wall Following) και η "Περιστολή" (Containment). Η "Ακολουθία Τοιχώματος Διαδρόμου" υλοποιείται με την "Έκκεντρη Καταδίωξη" ενώ η "Περιστολή" αποτελεί γενίκευση της "Ακολουθίας Διαδρόμου" (η "Ακολουθία Διαδρόμου" είναι ο τύπος αυτός της "Περιστολής" όπου η επιτρεπόμενη περιοχή διακίνησης είναι ένας κύλινδρος εκ περιστροφής με άξονα τον άξονα του διαδρόμου). Οι αλγόριθμοι εύρεσης βέλτιστης διαδρομής βρίσκουν μεγάλη εφαρμογή στην περίπτωση της ανάπτυξης της συμπεριφοράς Ακολουθίας Διαδρόμου.

Κατά τη διερεύνηση της συμπεριφοράς αυτής και στα πλαίσια της παρούσας εργασίας, η τυχαιότητα της επιλογής των διαδρόμων περιορίσθηκε και προσαρμόσθηκε στην απλή μετάβαση του πράκτορα από ένα σημείο που θα καλείται "Αφετηρία" προς ένα άλλο σημείο που θα καλείται "Τερματισμός". Ο περιορισμός της έννοιας "Ακολουθία Διαδρόμου" έγινε χωρίς βλάβη της γενικότητας και του σκοπού για τον οποίο διερευνώνται οι συμπεριφορές ενός πράκτορα αφού η συμπεριφορά αυτή προσομοιάζει την κατάσταση όπου ένας επιβάτης ενημερώνεται από ένα διάγραμμα για την συντομότερη διαδρομή από το χώρο όπου βρίσκεται προς ένα άλλο σημείο όπως είναι ο χώρος συγκέντρωσης επιβατών στο κατάστρωμα και μπορεί να εξάγει τη συντομότερη ή ευχερέστερη διαδρομή προς το τερματισμό του έχοντας υπόψη τυχόν περιορισμούς από κινδύνους κατά τη διαδρομή αυτή. Στην περίπτωση αυτή, ο πράκτορας επιλέγει την συντομότερη διαδρομή η οποία ταυτόχρονα πληροί και μία σειρά κριτηρίων που ποικίλουν από το να είναι ελεύθερη εμποδίων έως και το να προτιμηθεί μία διαδρομή που ναι μεν διαθέτει εμπόδια αλλά είναι προτιμότερη μίας άλλης που όμως εγκυμονεί κινδύνους.

Στην εφαρμογή μας δε χρειαζόμαστε πάντοτε τη βέλτιστη διαδρομή μεταξύ δύο σημείων αφού αυτό μπορεί να σημαίνει ότι όλοι οι επιβάτες είναι τόσο εξοικειωμένοι με τη χωροταξία του πλοίου ώστε να γνωρίζουν ανά πάσα στιγμή τη βέλτιστη διαδρομή μεταξύ δύο σημείων στο πλοίο. Στις περισσότερες περιπτώσεις μία απλά καλή ή ανεκτή διαδρομή θα είναι ικανοποιητική. Γενικά η επιλογή μίας διαδρομής μπορεί επίσης να βασίζεται στα διάφορα σενάρια που μπορούμε να εφαρμόσουμε στην εφαρμογή και στη συμπεριφορά που αναμένεται να επικρατήσει τελικά ή ακόμα και πόσο γρήγορος είναι ο φέρων υπολογιστής.

Η υλοποίηση της Καθοδηγητικής Συμπεριφοράς Ακολουθίας Διαδρόμου συνεπώς θα

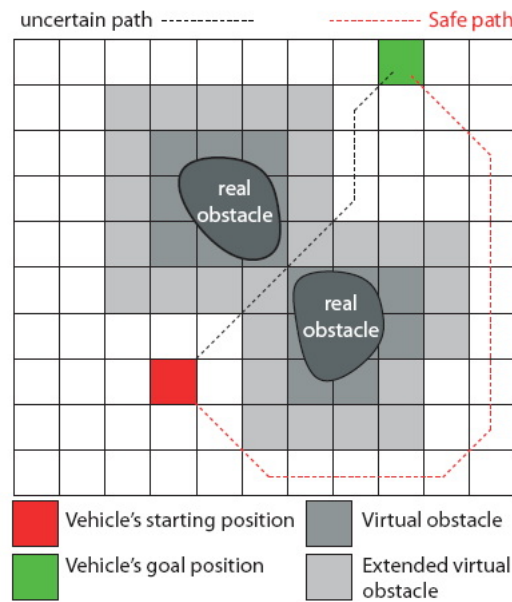
περιοριστεί σε ένα πολύ περιορισμένο πεδίο το οποίο είναι η μετάβαση ενός επιβάτη από ένα σημείο προς ένα άλλο. Στο Κεφάλαιο 3 αναπτύχθηκαν οι μέθοδοι εξεύρεσης της βέλτιστης διαδρομής μεταξύ δύο σημείων που παρεμβάλλονται εμπόδια αδιαπέραστα και επιλέχθηκε η μέθοδος  $A^*$  ως η πιο βέλτιστη μέθοδος.

Ας εξετάσουμε την εφαρμογή του αλγόριθμου σε μία περίπτωση εγκατάλειψης ενός πλοίου όπου ταυτόχρονα έχει εκδηλωθεί πυρκαγιά σε διάφορα σημεία του με ταυτόχρονη εξάπλωση καπνού: Έστω ότι έχουμε δύο τύπους κελιών που αντιπροσωπεύουν ένα καθαρό χωρίο και ένα χωρίο γεμάτο με καπνούς και ας υποθέσουμε ότι η μετακίνηση σε ένα καθαρό χωρίο κοστίζει 1 μονάδα και προς ένα χωρίο με καπνούς 3 μονάδες. Αυτό επιτρέπει στους πράκτορες της εφαρμογής να ερευνήσουν για ένα διάδρομο διαμέσου των καθαρών χωρίων που θα κινείται εκτός των χωρίων με καπνό. Μπορούμε να αλλάξουμε τις προτιμήσεις μας για την επιλογή του διαδρόμου, απλά με το να αλλάξουμε το κοστολόγιο μετακίνησης προς μία καθαρή περιοχή. Με το τρόπο αυτό αν υποθέσουμε ότι η μετακίνηση σε καθαρή περιοχή κοστίζει 1.5 μονάδες τότε ο αλγόριθμος δεν θα βλέπει ως δραστική τη διαφορά μεταξύ δύο κελιών καθαρού και με καπνό οπότε δεν θα επιμένει πια τόσο πολύ για την εύρεση διαδρόμου που να κινείται εκτός των περιοχών καπνού και έτσι θα επιταχύνει τη λύση του. Εναλλακτικά μπορούμε να επιταχύνουμε την έρευνα του  $A^*$  με το να μειώσουμε το κόστος μετακίνησης σε καθαρές περιοχές από 3 σε 2. Τώρα ο  $A^*$  θα διερευνήσει μόνο δύο φορές παραπάνω κατά μήκος καθαρών περιοχών από ότι διερχόμενος περιοχών καπνού. Αυτές οι δύο προσεγγίσεις του προβλήματος είναι ισοδύναμες όσον αφορά τη συμπεριφορά του  $A^*$ .

Η επιλογή μεταξύ ταχύτητας και ακρίβειας αποτελεσμάτων μπορεί να είναι και δυναμική και να βασίζεται στην ταχύτητα της CPU<sup>5</sup>, το κλάσμα του χρόνου μεταξύ σκημών κατά την εύρεση του βέλτιστου διαδρόμου, ο αριθμός των οχημάτων (πρακτόρων ή επιβατών) που παίρνουν μέρος στη διαδικασία, το φύλο των επιβατών, το μέγεθος των ομάδων ή οποιαδήποτε άλλο παράγοντα. Αυτή η δυναμική υπεισέρχεται με μία ευριστική συνάρτηση που υποθέτει για παράδειγμα ότι το ελάχιστο κόστος μετακίνησης από ένα κελί προς ένα άλλο είναι 1 και κατόπιν το κόστος μπορεί να δοθεί από μία συνάρτηση του τύπου

---

<sup>5</sup>Central Processing Unit (Κεντρική Μονάδα Επεξεργασίας του Η/Υ)



Σχήμα 4.32: Εύρεση Βέλτιστης Διαδρομής με αλγόριθμο  $A^*$  και μεγέθυνση εμποδίου

$$g'(n) = 1 + \alpha * (g(n) - 1)$$

Εάν η παράμετρος  $\alpha$  ισούται με το 0 τότε η μεταβλητή συνάρτηση κόστους θα εκτιμάται πάντα σε 1. Στο σημείο αυτό το κόστος του κάθε χωρίου αγνοείται τελείως και ο  $A^*$  εργάζεται στο επίπεδο των προσβάσιμων και μη προσβάσιμων χωρίων (όπως στη περίπτωση των μεθόδων Παρεμβολής Γραμμής). Εάν το  $\alpha$  ισούται με 1 τότε θα χρησιμοποιηθεί η αρχική συνάρτηση κόστους και ο αλγόριθμος θα αποδώσει τα μέγιστα. Στη πραγματικότητα, το  $\alpha$  μπορεί να πάρει οποιαδήποτε τιμή μεταξύ 0 και 1.

Μία άλλη πιθανή θεώρηση είναι να υπολογίζουμε το ευριστικό εκτιμώμενο ελάχιστο κόστος αντί το απόλυτο ελάχιστο κόστος. Ως παράδειγμα έχουμε την περίπτωση όπου ο χώρος είναι γεμάτος από εμπόδια (καρέκλες, εμπόδια κλπ) με ένα κόστος μετακίνησης έστω 2 και ο υπόλοιπος χώρος είναι διάδρομοι με κόστος μετακίνησης έστω 1, οπότε κάποιος μπορεί να θεωρήσει ότι η ευριστική συνάρτηση δεν λαμβάνει υπόψη τους διαδρόμους αλλά επιστρέφει  $g = 2 * dist$  όπου  $dist$  είναι η τυχαία απόσταση.

Βέβαια, προϋπόθεση εφαρμογής του αλγόριθμου αυτού είναι ο επιμερισμός του πεδίου με τέτοιο τρόπο ώστε τα σημεία - κελιά να μπορούν να περιλαμβάνουν με μεγάλη ακρίβεια την πληροφορία των υπάρχοντων εμποδίων που σε τελική ανάλυση αναπαριστούν και την

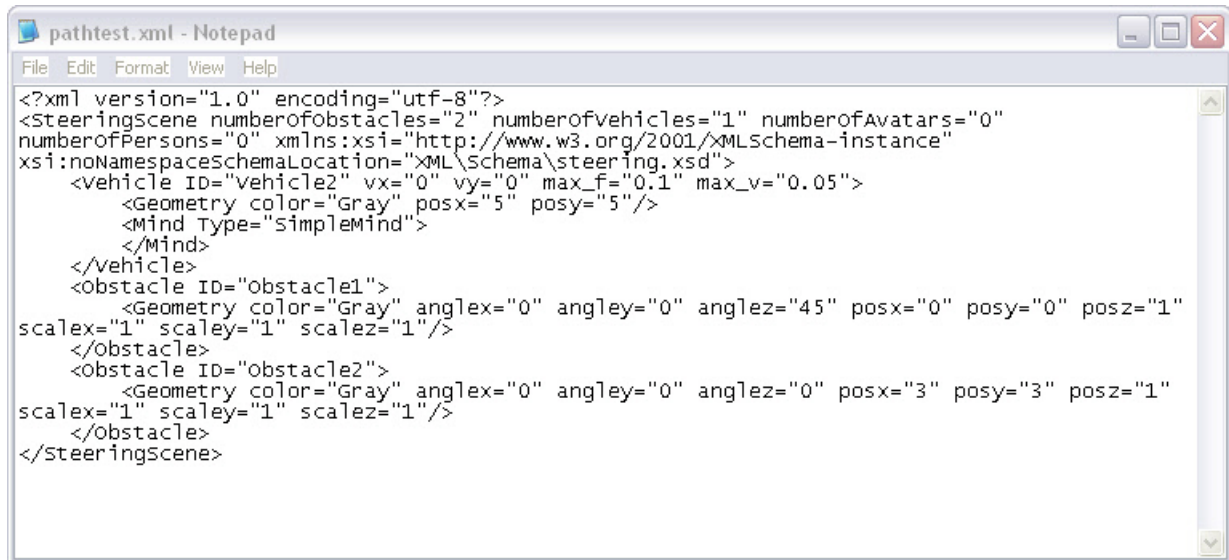


μορφολογία του πεδίου. Ο επιμερισμός του πεδίου θα πρέπει να είναι τέτοιος ώστε να δίνει την ακριβέστερη μορφολογία τηρούμενης της ελάχιστης απαίτησης κατάλληλης υπολογιστικής ισχύος ώστε να δύναται η προσομοίωση να είναι ακόμα αληθοφανής. Με τον τρόπο αυτό, ο επιμερισμός θα πρέπει να είναι η χρυσή τομή μεταξύ της ποιότητας και της ποσότητας.

#### 4.5.2 Ψευδοκώδικας

Η κυκλοφορία του κινούμενου πράκτορα επί μίας διαδρομής από μία "Αφετηρία" προς ένα "Τερματισμό" επιτυγχάνεται σε πέντε στάδια:

- Οριοθέτηση του πεδίου κίνησης. Η εφαρμογή περιορίζει την έκταση των υπολογισμών που θα λάβουν μέρος με την οριοθέτηση του πεδίου κίνησης του πράκτορα σε ένα τετράγωνο του οποίου οι κορυφές της μίας διαγωνίου του καταλαμβάνουν η αφετηρία και ο τερματισμός. Αν από τη χάραξη του τετραγώνου αυτού η περίμετρος του τέμνει αδιαπέραστα εμπόδια, τότε η περίμετρος επεκτείνεται αυτόματα μέχρις ότου αυτή είναι ελεύθερη εμποδίων.
- Επιμερισμός και ψηφιοποίηση του χωρίου σε τέτοιο βαθμό ώστε ανάλογα τη μορφή των εμποδίων (τετράγωνο, κύκλος ή πολύγωνο) να επιτυγχάνεται η ανάλογη λεπτομέρεια περιγραφής των εμποδίων.
- Εφαρμογή του αλγόριθμου  $A^*$  για την εξαγωγή της βέλτιστης διαδρομής από την αφετηρία προς τον τερματισμό.
- Καταγραφή του αποτελέσματος του αλγόριθμου  $A^*$  σε μία λίστα διαδοχικών σημείων μετάβασης προς τον τερματισμό.
- Εφαρμογή της συμπεριφοράς "Αναζήτηση" για κάθε σημείο της λίστας με αποτέλεσμα τη διαδοχική μετάβαση του πράκτορα από την αφετηρία στον τερματισμό παρακάμπτοντας τα εμπόδια.

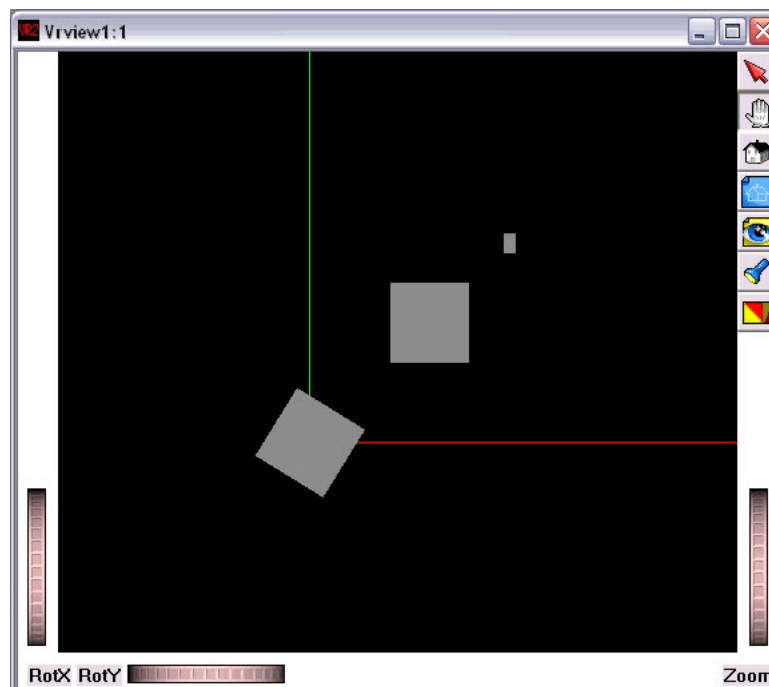


```

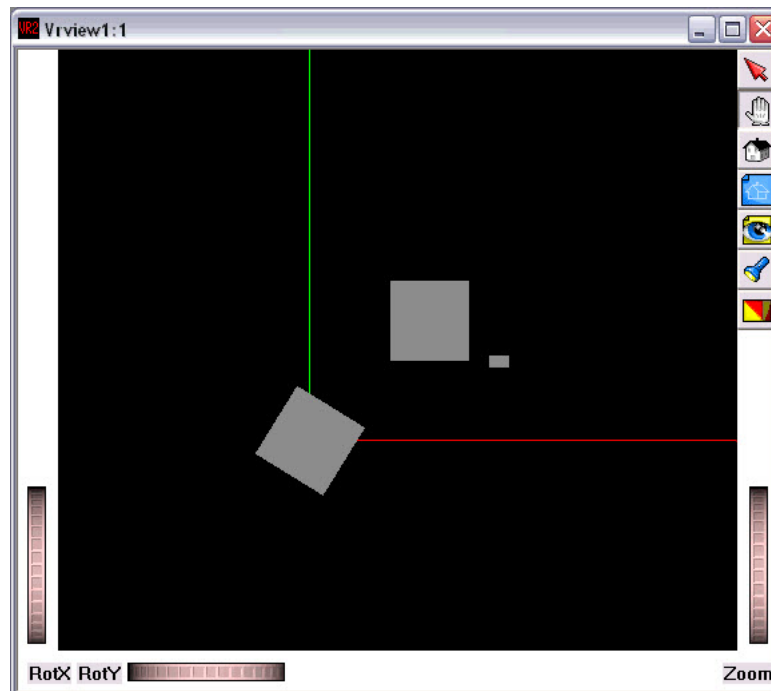
pathstest.xml - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<SteeringScene numberOfObstacles="2" numberOfVehicles="1" numberOfAvatars="0"
numberOfPersons="0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XML\Schema\steering.xsd">
  <vehicle ID="vehicle2" vx="0" vy="0" max_f="0.1" max_v="0.05">
    <Geometry color="Gray" posx="5" posy="5"/>
    <Mind type="simplemind">
    </Mind>
  </vehicle>
  <Obstacle ID="obstacle1">
    <Geometry color="Gray" anglex="0" angley="0" anglez="45" posx="0" posy="0" posz="1"
scalex="1" scaley="1" scalez="1"/>
  </Obstacle>
  <Obstacle ID="obstacle2">
    <Geometry color="Gray" anglex="0" angley="0" anglez="0" posx="3" posy="3" posz="1"
scalex="1" scaley="1" scalez="1"/>
  </Obstacle>
</SteeringScene>

```

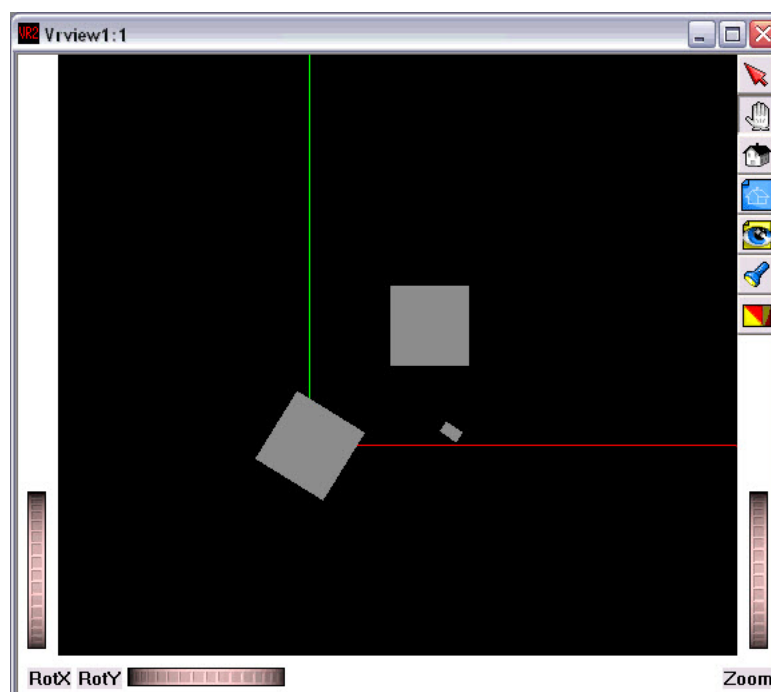
Σχήμα 4.33: Εισαγωγή Στοιχείων Συμπεριφοράς "Ακολουθία Διαδρόμου"



Σχήμα 4.34: Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Έναρξη

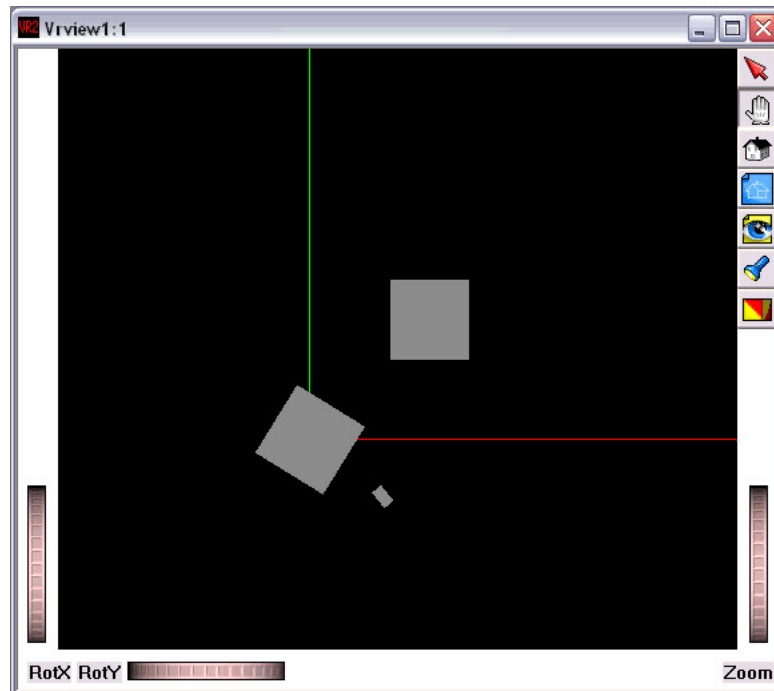


Σχήμα 4.35: Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Παράκαμψη 1ου Εμποδίου

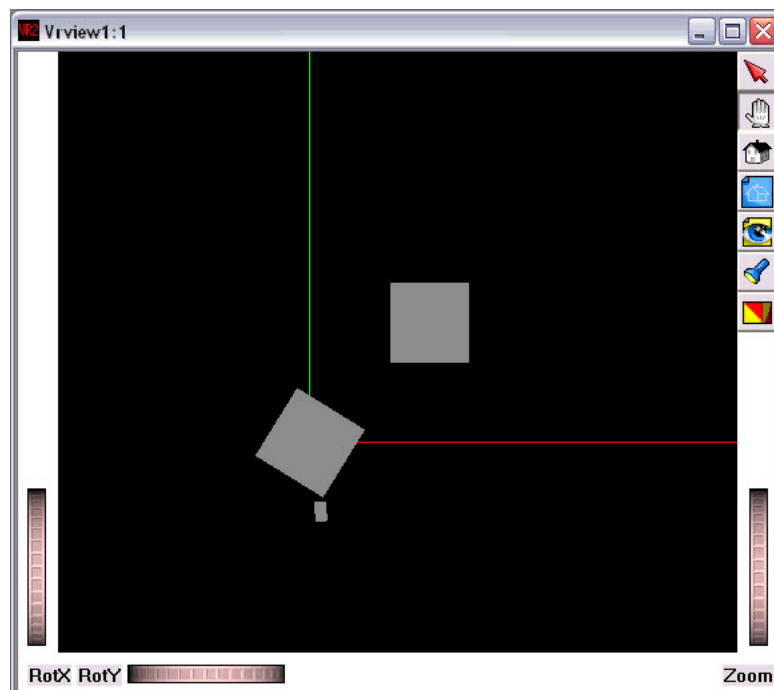


Σχήμα 4.36: Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Αρχή Διαγώνιας Κίνηση

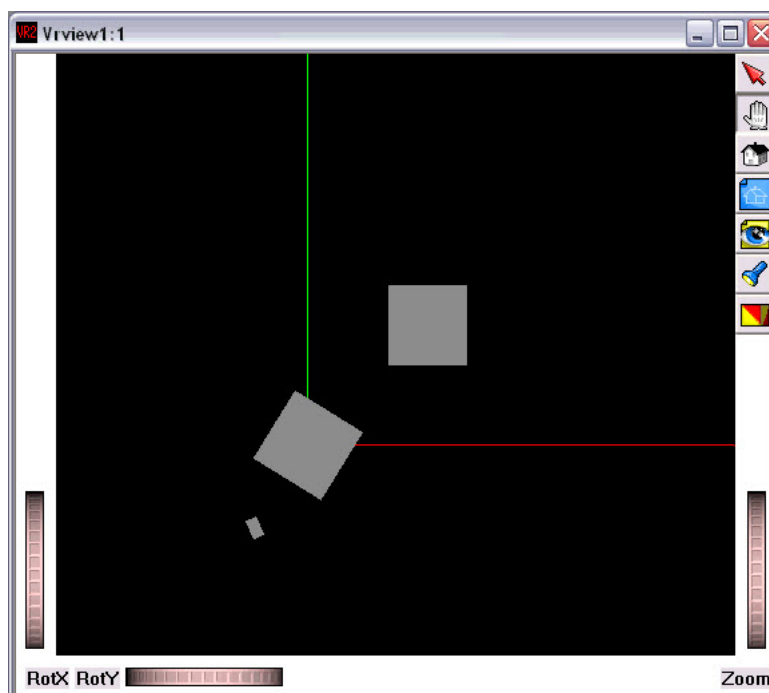
Υλοποίηση Καθοδηγητικών Συμπεριφορών



Σχήμα 4.37: Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Τέλος Διαγώνιας Κίνησης



Σχήμα 4.38: Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Παράκαμψη 2ου Εμποδίου



Σχήμα 4.39: Καθοδηγητική Συμπεριφορά "Ακολουθία Διαδρόμου" - Τέλος Διαδρομής

### 4.5.3 Υλοποίηση

Η καθοδηγητική συμπεριφορά "Ακολουθία Διαδρόμου" υλοποιήθηκε με βάση τον ψευδοκώδικα που περιγράφηκε στο προηγούμενο εδάφιο. Κατόπιν διαμορφώθηκε σε μία κλάση αντικειμενοστραφούς προγραμματισμού γλώσσας C++ και σε συνδυασμό με μία κλάση που εφαρμόζει τον αλγόριθμο A\* προσαρμόστηκαν στο κέλυφος χειρισμού του πυρήνα VRKernel αφού έγιναν οι κατάλληλες αλλαγές στο γραφικό περιβάλλον αλληλεπίδρασης του χρήστη. Η εισαγωγή των βασικών ρυθμίσεων των πρακτόρων που θα λάβουν μέρος στη προσομοίωση έγινε σύμφωνα με το αρχείο τύπου XML όπως αυτό παρουσιάζεται στο Σχήμα 4.33.

Στα Σχήματα 4.34 έως 4.39 παρουσιάζεται η διαδοχική εκτέλεση ενός τυπικού σεναρίου με τη χρήση της συμπεριφοράς "Ακολουθία Διαδρόμου". Σύμφωνα με το σενάριο αυτό, ο φαιός κινούμενος πράκτορας ευρίσκεται σε μία αρχική θέση  $(x, y) = (3.0, 3.0)$  και στη γειτονία ενός πλέγματος εμποδίων ενώ το άνωσμα της αρχικής του ταχύτητας είναι:

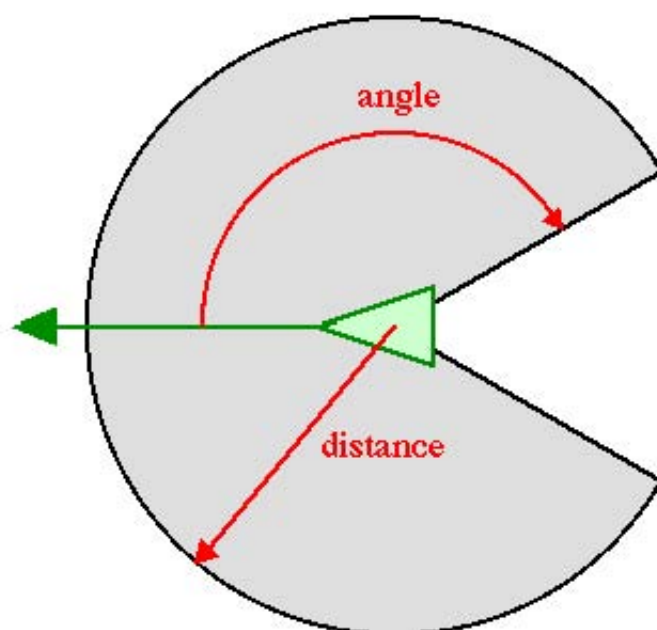
$$\vec{V}_M = \{0.0035355, 0.0035355\}, \text{ όπου } \|\vec{V}_M\|_{max} = 0.005$$

Ο κινούμενος πράκτορας κατευθύνεται προς τη θέση  $(x, y) = (-2.0, -2.0)$  με σταθερή

πορεία και ταχύτητα διαμέσου του πλέγματος των εμποδίων. Γίνεται προφανές ότι ακολουθείται η βέλτιστη διαδρομή η οποία σε χαρακτηριστικά σημεία της παρακάμπτεi με ασφάλεια τα υφιστάμενα εμπόδια (Σχήματα 4.35 και 4.38).

## 4.6 Ευθυγράμμιση

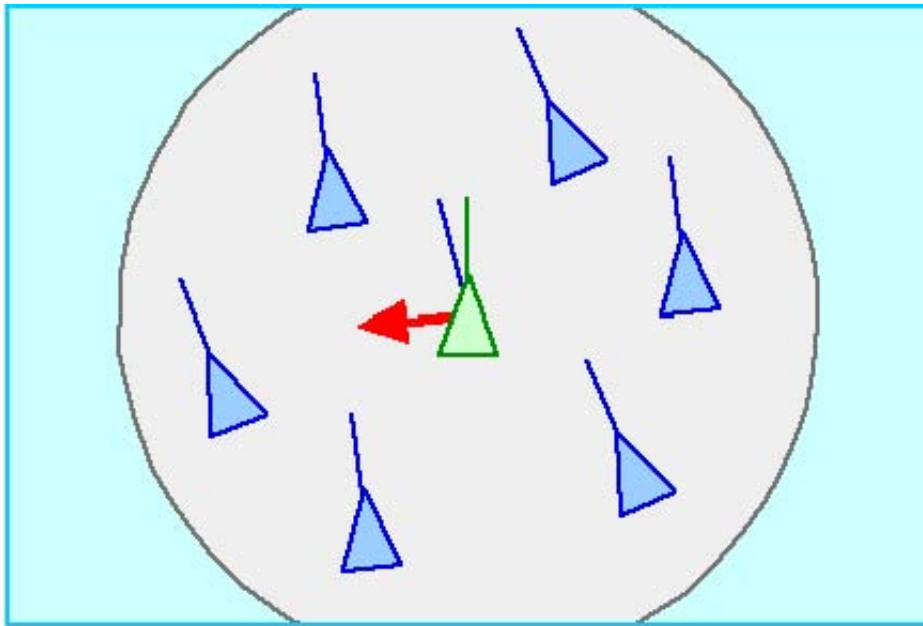
### 4.6.1 Περιγραφή



Σχήμα 4.40: Έννοια της Γειτονίας

Η "Συγκροτημένη Κίνηση" και η "Ευθυγράμμιση" μαζί με τον "Διαχωρισμό" (Separation) εφαρμόζονται σε γενικές γραμμές σε ομάδες οχημάτων. Πιο συγκεκριμένα, η "Συγκροτημένη Κίνηση" οδηγεί το όχημα σε μία θέση που ορίζεται ως το κέντρο βάρους της θέσης των γειτονικών οχημάτων ενώ στην "Ευθυγράμμιση" ο πράκτορας ευθυγραμμίζεται (πορεία και/ή ταχύτητα) με τα γειτονικά οχήματα.

Όταν εξετάζουμε ομαδικές συμπεριφορές όπως η "Ευθυγράμμιση" θα πρέπει να αναφέρουμε την "Γειτονία (Neighborhood)" ως μία εισαγωγική έννοια που θα μας βοηθήσει να κατανοήσουμε το μηχανισμό λειτουργίας και εφαρμογής τους. Όπως φαίνεται και στο Σχήμα



Σχήμα 4.41: Ευθυγράμμιση

4.40 η "Γειτονία" περιγράφει τον πληθυσμό των κινητών πρακτόρων που περιβάλλονται σε μία περιοχή έμπροσθεν της κίνησης του πράκτορα που μας ενδιαφέρει με συγκεκριμένο εύρος και απόσταση. Όταν ο πράκτορας εντοπίσει άλλον ή άλλους κινητούς πράκτορες εντός του τομέα ερεύνης του τότε ενεργοποιεί την κατάλληλη συμπεριφορά που θα του επιτρέψει ουσιαστικά να συμμετάσχει σε μία μετακινούμενη ομάδα ή αγέλη ή σμήνος.

#### 4.6.2 Ψευδοκώδικας

Η κυκλοφορία του κινούμενου πράκτορα σε "Ευθυγράμμιση" προς ένα πληθυσμό γειτονικών πρακτόρων επιτυγχάνεται σε τρία στάδια:

- Διερεύνηση σε μία περιοχή έμπροσθεν της κίνησης του μετακινούμενου πράκτορα για την ύπαρξη άλλων πρακτόρων `neighborhood` που διαθέτουν πορεία και ταχύτητα.

$$\text{near\_agents}[] = \text{findNeighborhood}()$$

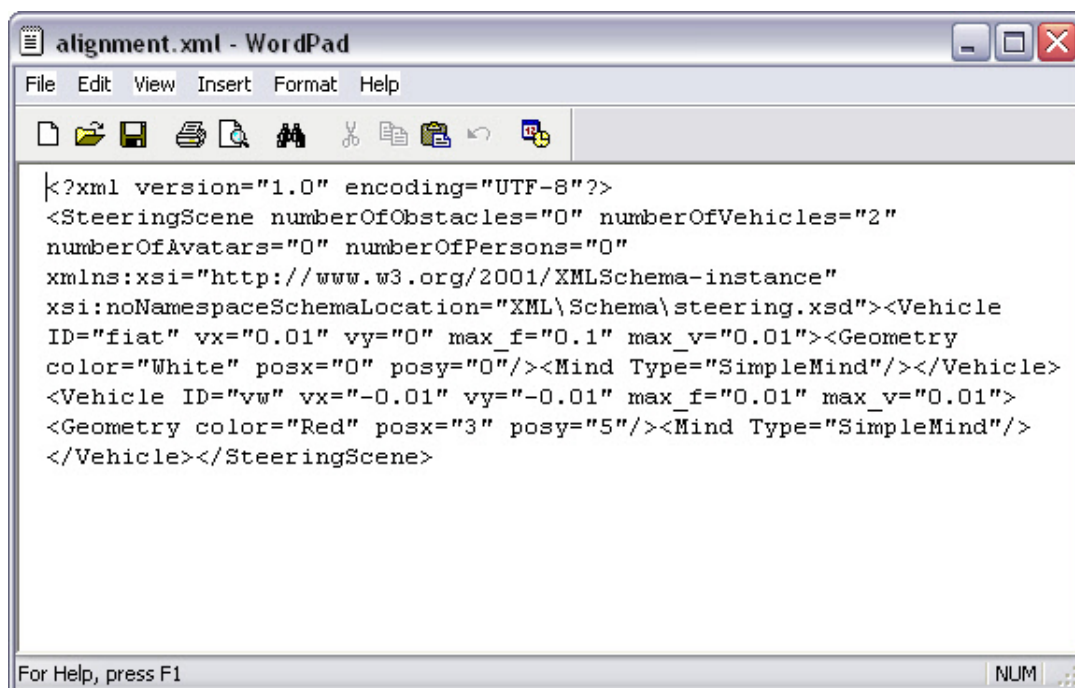
- Εξαγωγή του μέτρου του ανύσματος του μέσου όρου των ταχυτήτων των πρακτόρων που περιέχονται στην γειτονία του κινούμενου πράκτορα `neighborhood_velocity`:

Υλοποίηση Καθοδηγητικών Συμπεριφορών

$$neighborhood\_velocity = average(near\_agents\_velocity[])$$

- Εξαγωγή της διορθωτικής ταχύτητας `alignment_steering` που εκτιμάται ως η διαφορά της υφιστάμενης πορείας και της πορείας που πρέπει να αποκτήσει ο κινούμενος πράκτορας ώστε να ευθυγραμμισθεί με τους υπολοίπους:

$$alignment\_steering_M = neighborhood\_velocity - current\_velocity_M$$

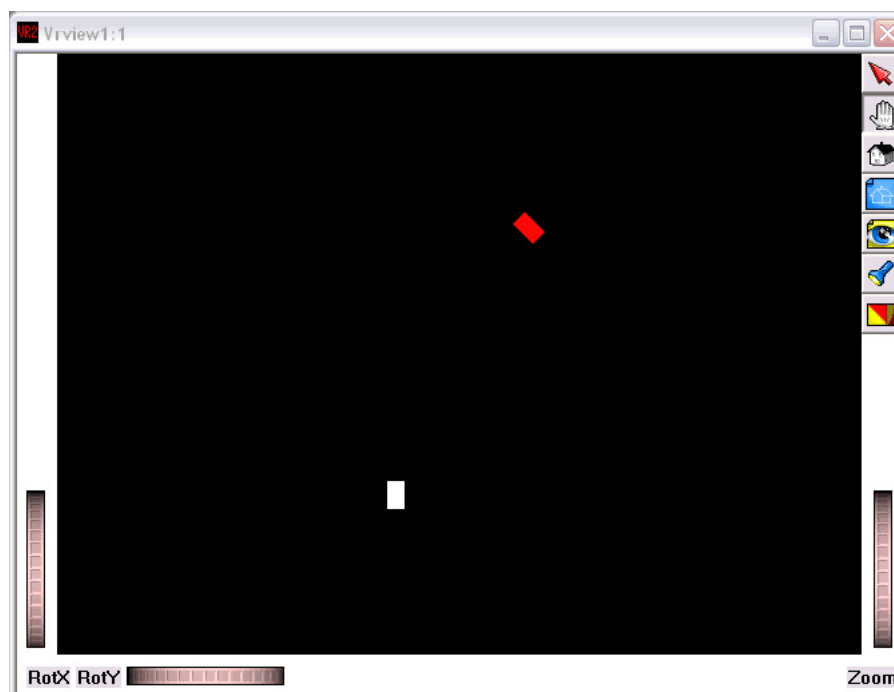


Σχήμα 4.42: Εισαγωγή Στοιχείων Συμπεριφοράς "Ευθυγράμμιση"

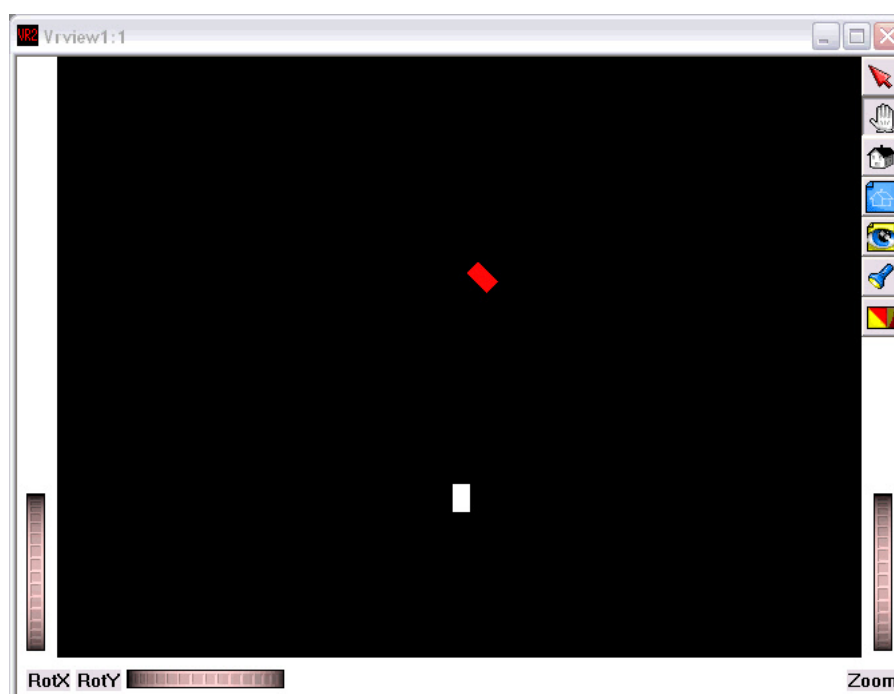
### 4.6.3 Υλοποίηση

Η καθοδηγητική συμπεριφορά "Ευθυγράμμιση" υλοποιήθηκε με βάση τον ψευδοκώδικα που περιγράφηκε στο προηγούμενο εδάφιο. Κατόπιν διαμορφώθηκε σε μία κλάση αντικειμενοστραφούς προγραμματισμού γλώσσας C++ και προσαρμόστηκε στο κέλυφος χειρισμού του πυρήνα VRKernel αφού έγιναν οι κατάλληλες αλλαγές στο γραφικό περιβάλλον αλληλεπίδρασης του χρήστη. Η εισαγωγή των βασικών ρυθμίσεων των πρακτόρων που θα λάβουν

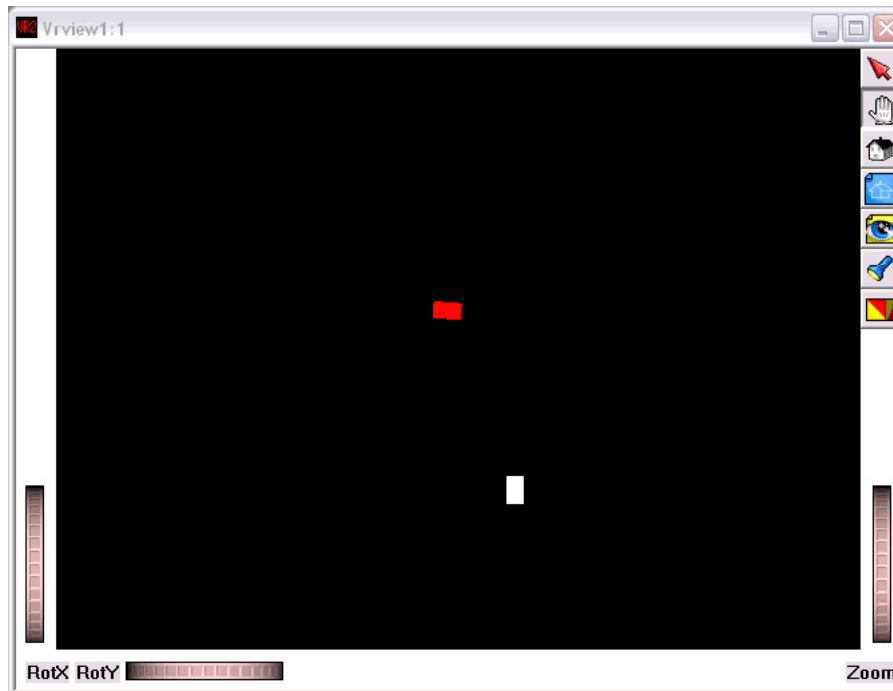




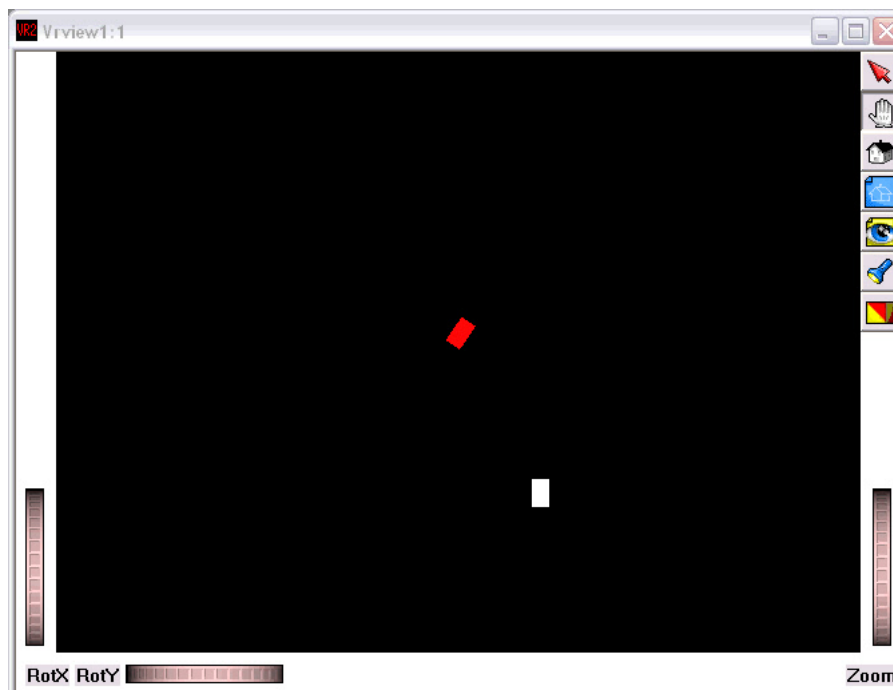
Σχήμα 4.43: Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Έναρξη Κίνησης



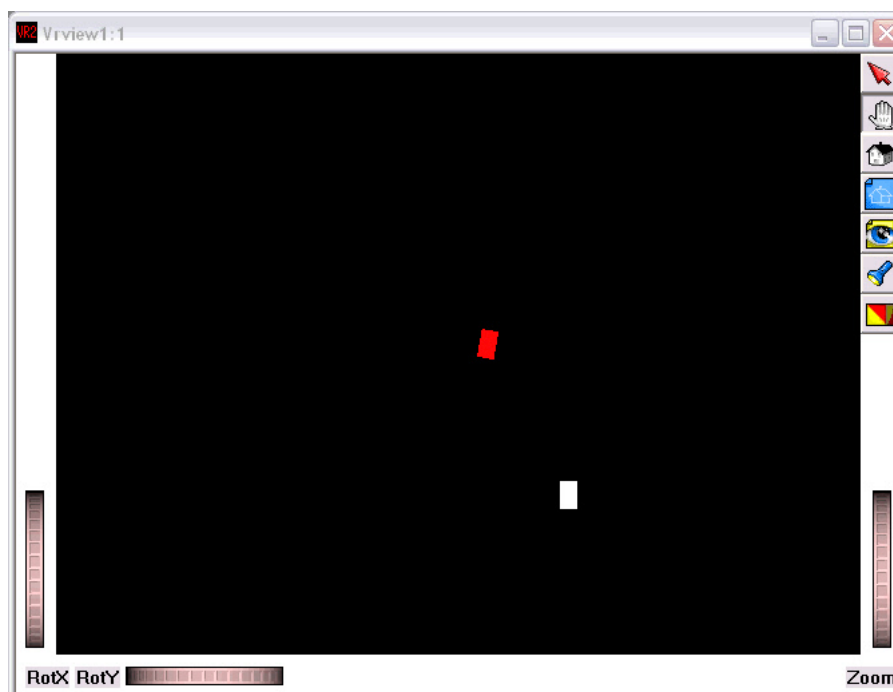
Σχήμα 4.44: Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Ενεργοποίηση



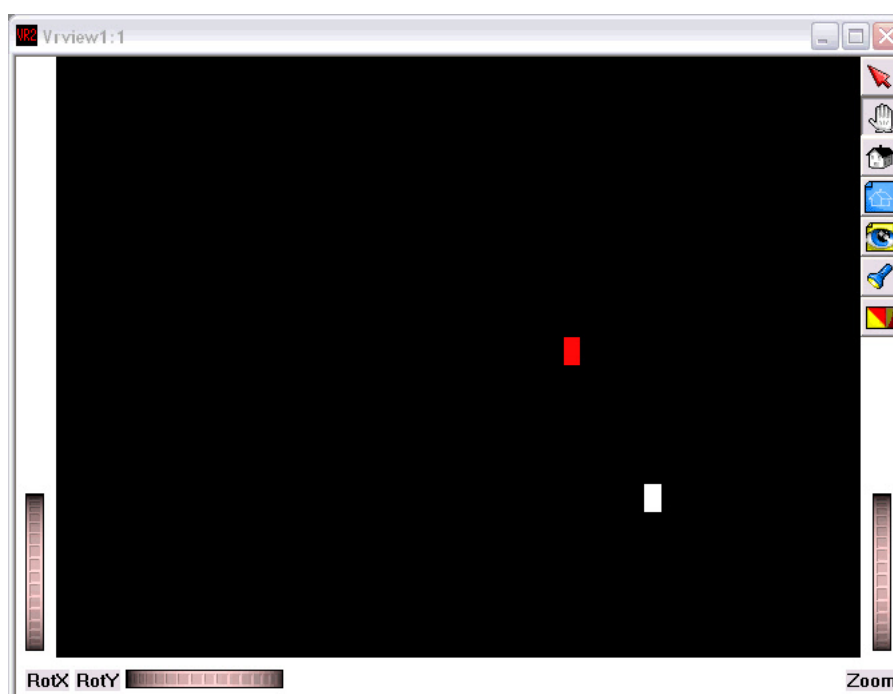
Σχήμα 4.45: Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Έναρξη Στροφής Ευθυγράμμισης



Σχήμα 4.46: Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Ενδιάμεσο Σημείο Στροφής I



Σχήμα 4.47: Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Ενδιάμεσο Σημείο Στροφής II



Σχήμα 4.48: Καθοδηγητική Συμπεριφορά "Ευθυγράμμιση" - Τέλος Χειρισμού

μέρος στη προσομοίωση έγινε σύμφωνα με το αρχείο τύπου XML όπως αυτό παρουσιάζεται στο Σχήμα 4.42.

Στα Σχήματα 4.43 έως 4.48 παρουσιάζεται η διαδοχική εκτέλεση ενός τυπικού σεναρίου με τη χρήση της συμπεριφοράς "Ευθυγράμμιση". Σύμφωνα με το σενάριο αυτό, ο ερυθρός κινούμενος πράκτορας ευρίσκεται σε μία αρχική θέση  $(x, y) = (3.0, 5.0)$  και το άνωσμα της αρχικής του ταχύτητας είναι:

$$\vec{V}_M = \{0.0035355, 0.0035355\}, \text{ όπου } \|\vec{V}_M\|_{max} = 0.005$$

Ο κινούμενος πράκτορας κατευθύνεται αρχικά με σταθερή πορεία και ταχύτητα και όταν αντιληφθεί την ύπαρξη ενός κινούμενου πράκτορα σε απόσταση μικρότερη από  $r = 3.4$  μήκη, τότε αρχίζει τον ελιγμό ευθυγράμμισης οπότε γίνεται προφανές ότι ξεκινάει η διαδικασία της ευθυγράμμισης του (Σχήμα 4.44) με την πορεία του κινούμενου πράκτορα που απεικονίζεται στα Σχήματα 4.45 έως 4.48.

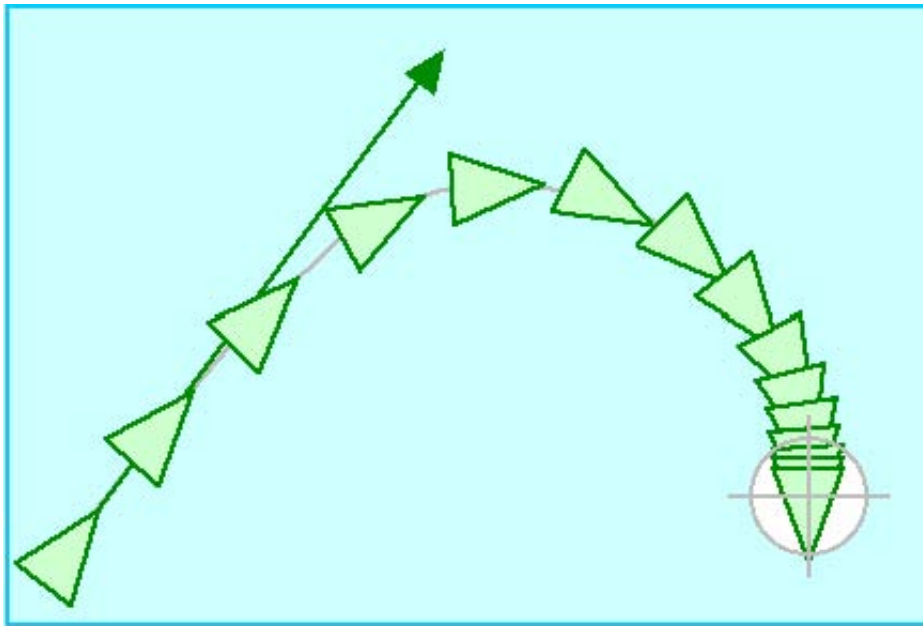
## 4.7 Άφιξη

### 4.7.1 Περιγραφή

Η συμπεριφορά "Αναζήτηση" έχει το χαρακτηριστικό της πλήρους ταχύτητας κατά την άφιξη του οχήματος στη περιοχή του στόχου. Αυτό έχει σαν αποτέλεσμα μία κίνηση του οχήματος που προσομοιάζει τη κίνηση ενός εκκρεμούς περί τη θέση ισορροπίας. Με τη συμπεριφορά "Άφιξη" έχουμε τον ίδιο τρόπο προσέγγισης προς το στόχο με αυτόν της "Αναζήτηση" με τη διαφορά ότι εδώ το όχημα επιβραδύνει καθώς πλησιάζει στη περιοχή του στόχου ώστε να έλθει σε πλήρη ακινησία επί του στόχου. Η ταχύτητα προσέγγισης του οχήματος στο στόχο είναι ανάλογη της απόστασής τους.

### 4.7.2 Ψευδοκώδικας

Η άφιξη του κινούμενου πράκτορα στο πράκτορα στόχο επιτυγχάνεται σε έξι στάδια:



Σχήμα 4.49: Άφιξη

- Αναζήτηση του ανύσματος της διαφοράς των θέσεων του πράκτορα στόχου και του κινούμενου πράκτορα  $target\_offset$  δηλαδή του ανύσματος που αντιπροσωπεύει την κατεύθυνση προσέγγισης μεταξύ των δύο πρακτόρων:

$$target\_offset = position_T - position_M$$

- Εξαγωγή του μέτρου του ανύσματος της διαφοράς των θέσεων του πράκτορα στόχου και του κινούμενου στόχου  $distance$ :

$$distance = length(target\_offset)$$

- Εξαγωγή της ταχύτητας επιβράδυνσης  $ramped\_speed$  που εκτιμάται ως συνάρτηση της μέγιστης ταχύτητας του κινούμενου στόχου, της απόστασης μεταξύ των θέσεων κινούμενου και πράκτορα στόχου και της απόστασης έναρξης λειτουργίας του αλγόριθμου:

$$ramped\_speed_M = max\_speed_M * (distance / slowing\_distance)$$

- Εξαγωγή της νέας ταχύτητας του κινούμενου στόχου `clipped_speed` ως η ελάχιστη ταχύτητα μεταξύ της μέγιστης ταχύτητας του κινούμενου στόχου και της εξαγόμενης ως ταχύτητας επιβράδυνσης:

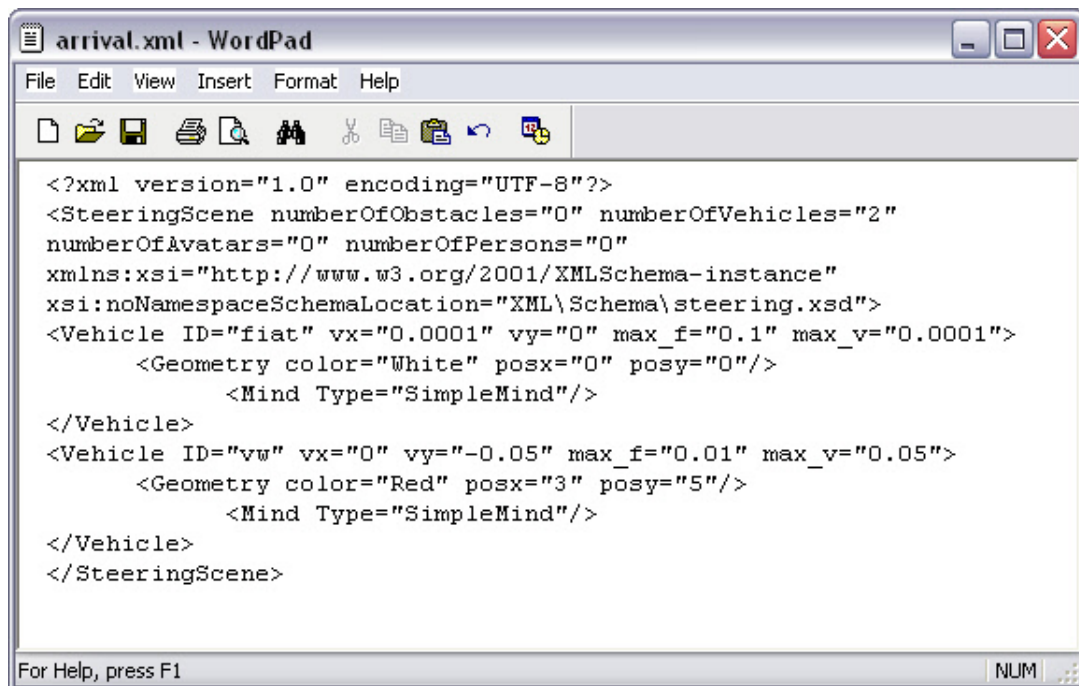
$$clipped\_speed_M = \text{minimum}(ramped\_speed_M, max\_speed_M)$$

- Αναζήτηση της επιθυμητής πορείας του κινούμενου πράκτορα με την εξεύρεση του ανύσματος `desired_velocity` δηλαδή του ανύσματος που συνδέει τη θέση του με τη μέλλουσα θέση του κινούμενου πράκτορα στόχου:

$$desired\_velocity_M = target\_offset * (clipped\_speed_M / distance)$$

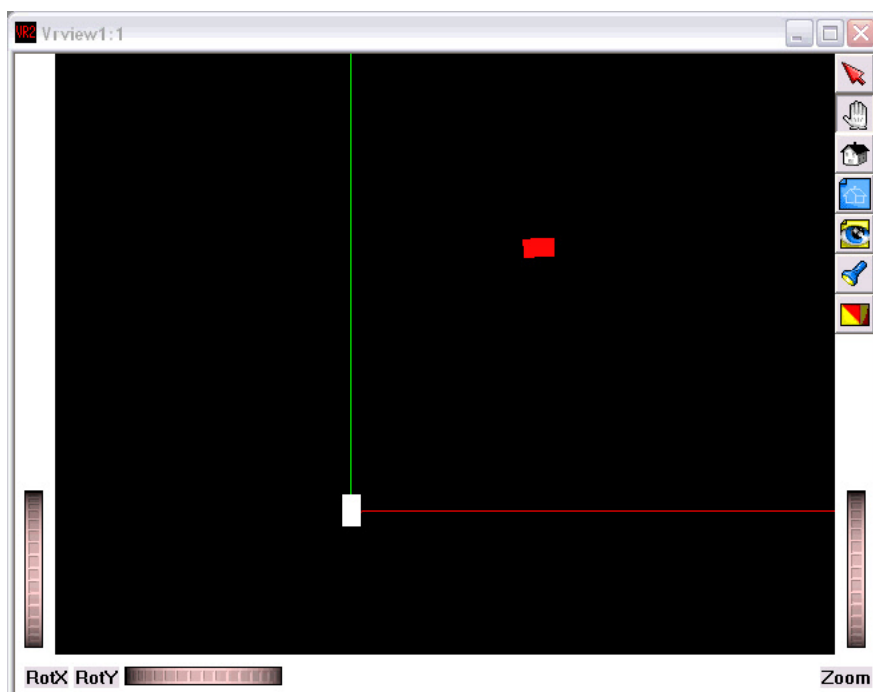
- Πορεία χειρισμού του κινούμενου πράκτορα `arrival_steering`:

$$arrival\_steering_M = desired\_velocity_M - current\_velocity_M$$

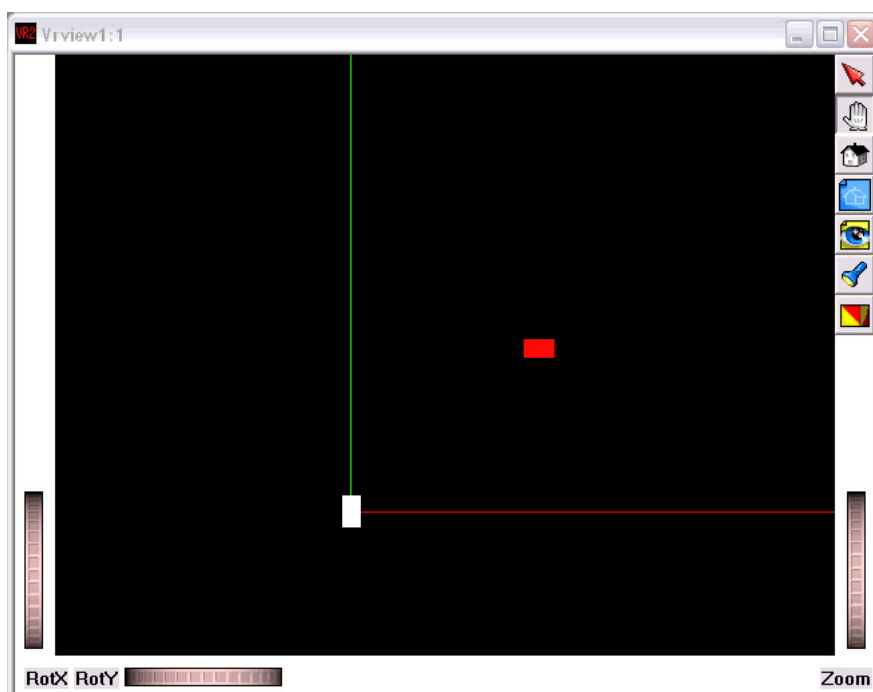


```
<?xml version="1.0" encoding="UTF-8"?>
<SteeringScene numberOfObstacles="0" numberOfVehicles="2"
numberOfAvatars="0" numberOfPersons="0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XML\Schema\steering.xsd">
  <Vehicle ID="fiat" vx="0.0001" vy="0" max_f="0.1" max_v="0.0001">
    <Geometry color="White" posx="0" posy="0"/>
    <Mind Type="SimpleMind"/>
  </Vehicle>
  <Vehicle ID="vw" vx="0" vy="-0.05" max_f="0.01" max_v="0.05">
    <Geometry color="Red" posx="3" posy="5"/>
    <Mind Type="SimpleMind"/>
  </Vehicle>
</SteeringScene>
```

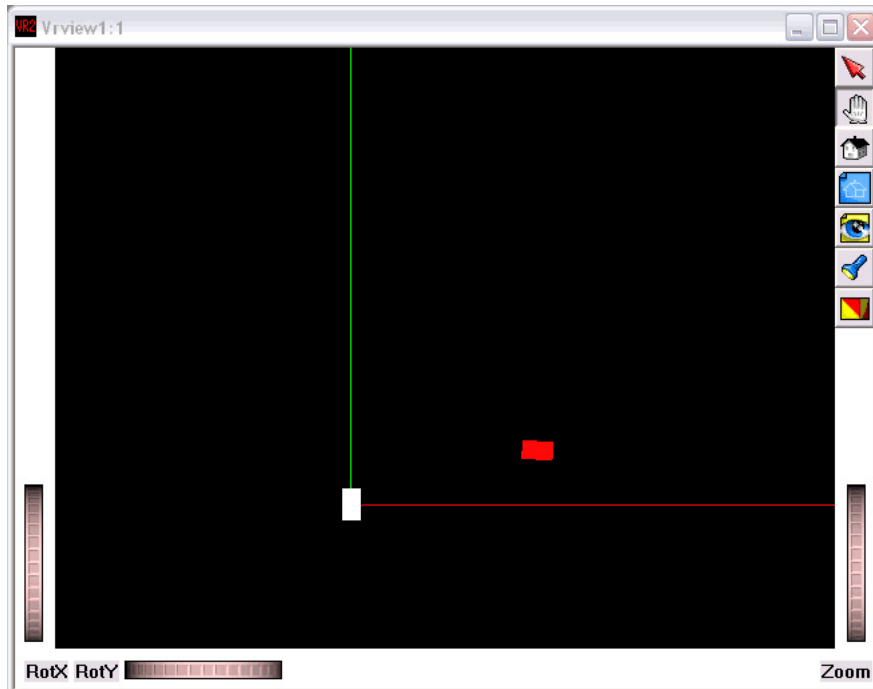
Σχήμα 4.50: Εισαγωγή Στοιχείων Συμπεριφοράς "Αφιξη"



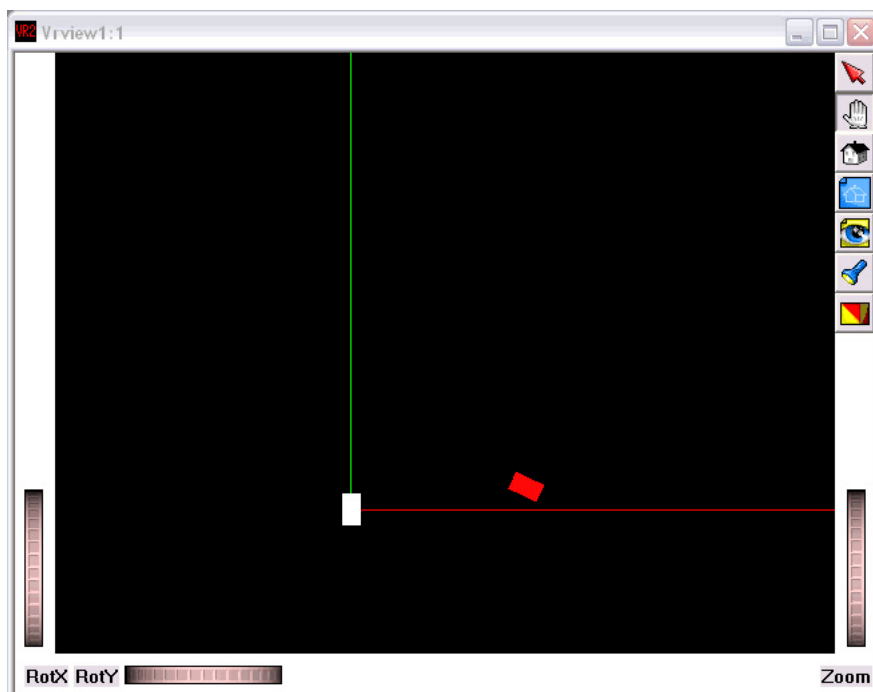
Σχήμα 4.51: Καθοδηγητική Συμπεριφορά "Άφιξη" - Έναρξη



Σχήμα 4.52: Καθοδηγητική Συμπεριφορά "Άφιξη" - Ενεργοποίηση



Σχήμα 4.53: Καθοδηγητική Συμπεριφορά "Αφιξη" - Έναρξη Χειρισμού



Σχήμα 4.54: Καθοδηγητική Συμπεριφορά "Αφιξη" - Ενδιάμεσος Χειρισμός I





Σχήμα 4.55: Καθοδηγητική Συμπεριφορά "Άφιξη" - Ενδιάμεσος Χειρισμός II



Σχήμα 4.56: Καθοδηγητική Συμπεριφορά "Άφιξη" - Τέλος Χειρισμού

### 4.7.3 Υλοποίηση

Η καθοδηγητική συμπεριφορά "Αφιξη" υλοποιήθηκε με βάση τον ψευδοκώδικα που περιγράφηκε στο προηγούμενο εδάφιο. Κατόπιν διαμορφώθηκε σε μία κλάση αντικειμενοστραφούς προγραμματισμού γλώσσας C++ και προσαρμόστηκε στο κέλυφος χειρισμού του πυρήνα VRKernel αφού έγιναν οι κατάλληλες αλλαγές στο γραφικό περιβάλλον αλληλεπίδρασης του χρήστη. Η εισαγωγή των βασικών ρυθμίσεων των πρακτόρων που θα λάβουν μέρος στη προσομοίωση έγινε σύμφωνα με το αρχείο τύπου XML όπως αυτό παρουσιάζεται στο Σχήμα 4.50.

Στα Σχήματα 4.51 έως 4.56 παρουσιάζεται η διαδοχική εκτέλεση ενός τυπικού σεναρίου με τη χρήση της συμπεριφοράς "Καταδίωξη". Σύμφωνα με το σενάριο αυτό, στο κέντρο ενός καρτεσιανού συστήματος συντεταγμένων ευρίσκεται ο λευκός πράκτορας στόχος, ο οποίος και στη προκειμένη περίπτωση το άνυσμα της ταχύτητάς του είναι μηδενικό.

Ο ερυθρός κινούμενος πράκτορας ευρίσκεται αρχικά στη θέση  $(x, y) = (3.0, 5.0)$  και το άνυσμα της ταχύτητάς του είναι:

$$\vec{V}_M = \{0.000, -5.000\}, \text{ όπου } \|\vec{V}_M\|_{max} = 0.005$$

Ο κινούμενος πράκτορας κατευθύνεται αρχικά με σταθερή πορεία και ταχύτητα αλλά όταν φθάσει σε απόσταση ενεργοποίησης του αλγόριθμου "Αφιξη" που στη συγκεκριμένη περίπτωση είναι  $r = 3.4$  μήκη, τότε αρχίζει τον ελιγμό άφιξης (Σχήμα 4.53) εκτελώντας μία δεξιά στροφή και προσεγγίζει σταδιακά τον πράκτορα στόχο (Σχήματα 4.54 έως 4.56).

□

# Κεφάλαιο 5

## Συμπεράσματα

Η ανάπτυξη των καθοδηγητικών συμπεριφορών στο προηγούμενο κεφάλαιο 4 έγινε κατά ένα τρόπο που επιτρέπει την εκτέλεσή τους από έναν Η/Υ χαμηλών επιδόσεων για τα σημερινά δεδομένα. Επιπλέον όμως καταβλήθηκε προσπάθεια ώστε να προσομοιάζουν όσο το δυνατόν την πραγματικότητα με τις ελάχιστες δυνατές παραδοχές. Η υλοποίησή τους ανέδειξε κάποια επιμέρους προβλήματα τα οποία αντιμετωπίστηκαν άλλοτε επιτυχώς και άλλοτε στα πλαίσια του δυνατού. Σε γενικές γραμμές όμως εξήχθησαν πολύτιμα συμπεράσματα που είναι δυνατόν να αποτελέσουν και τη σκυτάλη για την διεύρυνση και συνέχεια της έρευνας.

Τα συμπεράσματα της εργασίας αυτής θα παρατεθούν παρακάτω ομαδοποιημένα κατά τη θεματική ενότητα της κάθε συμπεριφοράς που υλοποιήθηκε και στη συνέχεια θα αναφερθούν γενικότερες παρατηρήσεις και συμπεράσματα.

### 5.1 Αποφυγή

Κατά την ανάπτυξη της συμπεριφοράς "Αποφυγή" δεν αντιμετωπίστηκαν ιδιαίτερα προβλήματα στην υλοποίηση του ψευδοκώδικα. Θα πρέπει να σημειώσουμε όμως τα παρακάτω όσον αφορά την παραμετροποίηση των μεταβλητών της συμπεριφοράς:

- Η παράμετρος της ταχύτητας του κινούμενου στόχου είναι βαρύνουσα σημασίας κατά την υλοποίηση της συμπεριφοράς αυτής. Η απόκριση της συμπεριφοράς αυξάνεται με το μέγεθος της ταχύτητας του κινούμενου πράκτορα. Δεν παρατηρήθηκε ιδιαίτερη

επιβάρυνση της απασχόλησης του υπολογιστικού συστήματος παρά μονάχα δραστικές μεταβολές στην απεικόνιση του κινούμενου πράκτορα σε κάθε νέο στιγμιότυπο.

- Η εξέταση της παραμέτρου της απόστασης ενεργοποίησης της συμπεριφοράς εξετάζεται σε συνδυασμό με την ταχύτητα του στόχου. Η ρύθμιση του συνδυασμού των δύο αυτών παραμέτρων θα πρέπει να ρυθμίζεται ανάλογα για να έχουμε αληθοφανή αποτελέσματα. Έτσι στην περίπτωση της μεγάλης ταχύτητας του κινούμενου πράκτορα, η απόσταση ενεργοποίησης του αλγόριθμου δύνανται να λάβει και χαμηλές τιμές (ιδιαίτερα εγγύς του ακίνητου στόχου) ενώ στην περίπτωση της μικρής ταχύτητας κίνησης τότε η απόσταση ενεργοποίησης του αλγόριθμου θα πρέπει να είναι αυξημένη ώστε να δίνεται το επαρκές περιθώριο αντίδρασης.
- Η παράμετρος πρόσμιξης της υπολογισμένης αντίδρασης του πράκτορα στη συνολική του κίνηση δεν μελετήθηκε σε βάθος αφού ο πυρήνας δίνει τη δυνατότητα πληθώρας ρυθμίσεων που δύνανται να καταστήσουν οποιαδήποτε προσομοίωση αληθοφανή αναλόγως των απαιτήσεών της. Σε γενικές γραμμές όμως η επίδραση της παραμέτρου αυτής είναι πανομοιότυπη με αυτή της ταχύτητας του κινούμενου πράκτορα.

## 5.2 Καταδίωξη & Διαφυγή

Κατά την ανάπτυξη των συμπεριφορών "Καταδίωξη" & "Διαφυγή" δεν αντιμετωπίστηκαν ιδιαίτερα προβλήματα στην υλοποίηση του ψευδοκώδικα. Θα πρέπει να σημειώσουμε όμως τα παρακάτω όσον αφορά την παραμετροποίηση των μεταβλητών των συμπεριφορών:

- Η παράμετρος της ταχύτητας του κινούμενου στόχου και η παράμετρος της ταχύτητας του πράκτορα στόχου είναι βαρύνουσας σημασίας κατά την υλοποίηση των συμπεριφορών αυτών. Η απόκριση των συμπεριφορών αυξάνεται με το μέγεθος της ταχύτητας των δύο πρακτόρων. Θα πρέπει όμως να σημειωθεί όμως ότι για την επιτυχή ολοκλήρωση της συμπεριφοράς θα πρέπει ο κινούμενος πράκτορας να έχει μεταβλητή ταχύτητα κατά τη διάρκεια του αλγόριθμου. Αυτό σημαίνει ότι ο κινούμενος πράκτορας θα πρέπει να διαθέτει μία ταχύτητα που θα μπορεί να αποκαλείται και ταχύτητα χειρισμού και

θα είναι σε κάθε περίπτωση μεγαλύτερη αυτή που διαθέτει ο πράκτορας στόχος. Ο κινούμενος πράκτορας θα χρησιμοποιεί την ταχύτητα χειρισμού μέχρις ότου έλθει σε μία κατάλληλη απόσταση όπου μπορεί να τη μειώσει ή να την αυξήσει ανάλογα κατάλληλα. Αυτό γίνεται γιατί σε περίπτωση σταθερής ταχύτητας του κινούμενου πράκτορα που είναι όμως μικρότερη από τη ταχύτητα του στόχου τότε στην πλειάδα των εφαρμογών η συμπεριφορά δεν ολοκληρώνεται επιτυχώς. Στο σημείο αυτό υπεισέρχεται ο παράγοντας της σχετικής ταχύτητας και ο παράγοντας της σύγκλισης της κίνησης των δύο πρακτόρων.

- Η εξέταση της παραμέτρου της απόστασης ενεργοποίησης της συμπεριφοράς εξετάζεται σε συνδυασμό με την σχετική ταχύτητα των δύο πρακτόρων. Η ρύθμιση της σχετικής ταχύτητας με τη σειρά της δεν είναι απευθείας δυνατή αφού αυτή με τη σειρά της εξαρτάται από τις υφιστάμενες για κάθε στιγμή τιμές των ταχυτήτων των δύο πρακτόρων. Έτσι στην περίπτωση της συγκλίνουσας σχετικής ταχύτητας προσέγγισης των δύο πρακτόρων, η απόσταση ενεργοποίησης του αλγόριθμου δύνανται να λάβει και χαμηλές τιμές (ιδιαίτερα εγγύς του ακίνητου στόχου) ενώ στην περίπτωση της αποκλίνουσας σχετικής ταχύτητας τότε η απόσταση ενεργοποίησης του αλγόριθμου θα πρέπει να είναι αυξημένη ώστε να δίνεται το επαρκές περιθώριο αντίδρασης και ο κινούμενος πράκτορας να είναι συνεχώς εντός του πεδίου δράσης του αλγόριθμου.
- Η παράμετρος πρόσμιξης της υπολογισμένης αντίδρασης του πράκτορα στη συνολική του κίνηση δεν μελετήθηκε σε βάθος αφού ο πυρήνας δίνει τη δυνατότητα πληθώρας ρυθμίσεων που δύνανται να καταστήσουν οποιαδήποτε προσομοίωση αληθοφανή αναλόγως των απαιτήσεών της. Σε γενικές γραμμές όμως η επίδραση της παραμέτρου αυτής είναι πανομοιότυπη με αυτή της ταχύτητας των δύο πρακτόρων.

### 5.3 Αποφυγή Εμποδίων

Η υλοποίηση της συμπεριφοράς αυτής κατέδειξε την πληθώρα των διαφορετικών προσεγγίσεων για την επίλυση του προβλήματος της "Αποφυγής Εμποδίων". Στην περίπτωση

της συμπεριφοράς αυτής υπεισέρχονται και προβλήματα αντίληψης όγκου αλλά και επιτυχίας του αλγόριθμου εντός στενών χρονικών πλαισίων. Ένα άλλο πρόβλημα που παρουσιάστηκε κατά την εκτέλεση του αλγόριθμου είναι ο υπό συνθήκες εγκλωβισμός του πράκτορα σε αντιδράσεις που δημιουργούν παλινδρομήσεις και επιφέρουν ουσιαστικά την ακινητοποίησή του.

- Το σημαντικότερο πρόβλημα κατά την επιτυχή υλοποίησή της συμπεριφοράς αυτής είναι η αντίληψη από τη μεριά του πράκτορα όχι μόνο της ύπαρξης σε κάποια απόσταση ενός εμποδίου αλλά και της πλήρους μορφής του. Η σωστή λύση στο πρόβλημά μας θα ήταν η ολοκληρωτική ψηφιοποίηση των εξωτερικών διαστάσεων του εμποδίου αλλά και του πράκτορα της προσομοίωσης όμως κάτι τέτοιο είναι ανέφικτο λόγω περιορισμών στην υπολογιστική ισχύ των Η/Υ. Βέβαια στην περίπτωση της παρούσας διπλωματικής εργασίας περιορίστηκε η μορφή των εμποδίων σε απλά πρισματικά χωρίς την ύπαρξη ασυνέχειας στην εξωτερική τους επιφάνεια και συνεπώς εύκολων στην ψηφιοποίηση και αντιμετώπισή τους. Το πρόβλημα εστιάστηκε περισσότερο στην μορφή του κινούμενου πράκτορα και στον τρόπο με τον οποίο αυτό αντιλαμβάνεται τα εμπόδια. Δεδομένου ότι έγινε χρήση μίας ιδιαίτερης κλάσης της βιβλιοθήκης `Coin3d` και συγκεκριμένα της κλάσης `SoRayPickAction`, το πρόβλημα έγινε ιδιαίτερα εμφανές όταν ο πράκτορας αποκτούσε μορφή διαφορετική της σημειακής. Υπήρχε περίπτωση όπου ενώ το κέντρο βάρους του πράκτορα μπορούσε να διέλθει ασφαλώς από ένα εμπόδιο, δεν συνέβαινε το ίδιο για τις κορυφές της πρισματικής του επιφάνειας. Το πρόβλημα περιορίστηκε με τη μορφοποίηση του πράκτορα ως ένα πρίσμα οπότε και έγινε δυνατή η χρήση ανιχνευτών εμποδίων στις εξωτερικές κορυφές του όγκου του.
- Η παράμετρος της ταχύτητας του κινούμενου στόχου ήταν ιδιαίτερα σημαντική όχι μόνο ως προς το μέτρο αλλά και ως προς τη διεύθυνση της ως προς την επιφάνεια του εμποδίου. Στην περίπτωση που την εξετάσουμε σε συνδυασμό με την απόσταση ενεργοποίησης του αλγόριθμου τότε καταλήγουμε σε προβλήματα επάρκειας αντίδρασης του αλγόριθμου όταν η ταχύτητα είναι μεγάλη σε σχέση με μικρή απόσταση ενεργοποίησης. Επίσης στην περίπτωση όπου ο πράκτορας εισέλθει σε περίκλειστο χωρίο

τότε υπάρχει περίπτωση σε υπό συνθήκες ακινητοποίησή του λόγω παλινδρομικών κινήσεων που αναιρούν μακροσκοπικά η μία την άλλη. Στην περίπτωση που η ταχύτητα του στόχου είναι κάθετη στην επιφάνεια του εμποδίου τότε δόθηκε και ο κανονικός αλγόριθμος οδηγεί σε υπερβολική ασυνέχεια στην κίνηση του πράκτορα (αλλαγή πορείας έως και 180 μοίρες) τότε επιχειρήθηκε μία τυχαία απειροελάχιστη αλλαγή της πορείας, με στόχο την διαφοροποίησή της από την κάθετο και την τελική αληθοφάνεια της αντίδρασης με ικανοποιητικά αποτελέσματα.

- Η εξέταση της παραμέτρου της απόστασης ενεργοποίησης της συμπεριφοράς εξετάζεται σε συνδυασμό με την έκταση της αντίδρασης του πράκτορα και την αληθοφάνεια του χειρισμού. Η απόσταση ενεργοποίησης του αλγόριθμου δύνανται να λάβει και χαμηλές τιμές (ιδιαίτερα εγγύς του εμποδίου) οπότε και υφίσταται ανάγκη για ένα δραστικό χειρισμό άσχετα με την πιθανή ασυνέχεια της απεικόνισης της κίνησης αλλά και να λάβει μεγάλες τιμές οπότε και παρέχεται η δυνατότητα στον πράκτορα για μία πιο αληθοφανή κίνηση. Βέβαια στην περίπτωση της έγκαιρης αντίληψης του εμποδίου αντιμετωπίστηκε ένα νέο πρόβλημα, αυτό της ομαλής αλλαγής της πορείας του πράκτορα (smooth change of course όχι μόνο σε εμφάνιση αλλά και υπό το βάρος του χρονικού περιορισμού για την επιτυχή ολοκλήρωση του αλγόριθμου. Προτάθηκαν τρία μοντέλα για την αντιμετώπιση του προβλήματος αυτού, εκ των οποίων μόνο το τελευταίο έδωσε ικανοποιητικά αποτελέσματα τόσο οπτικά όσο και επιτυχούς αντιμετώπισης των εμποδίων. Το μοντέλο περιγράφει έναν συντελεστή

$$(max\_threshold - closest\_obstacle\_Distance)/(max\_threshold - min\_threshold)$$

όπου εισάγονται δύο επιπλέον έννοιες, η *max\\_threshold* και *min\\_threshold* που αντίστοιχα οριοθετούν την μέγιστη απόσταση για την ενεργοποίηση της συμπεριφοράς "Αποφυγή Εμποδίου" και την ελάχιστη απόσταση αντίστοιχα για την λήψη ενός πολύ αποφασιστικού χειρισμού όπως η αλλαγή πορείας έως και 90 μοιρών για την αποφυγή του εμποδίου την τελευταία στιγμή. Στην περίπτωση αυτή η ομαλότητα στην απεικόνιση της θέσης και κατεύθυνσης του πράκτορα υψισιάζεται προς όφελος της αναγκαιότητας της ελάχιστης απόστασης και παρέχει ικανοποιητικά αποτελέσματα αληθοφάνειας.

- Η παράμετρος πρόσμιξης της υπολογισμένης αντίδρασης του πράκτορα στη συνολική του κίνηση δεν μελετήθηκε σε βάθος αφού ο πυρήνας δίνει τη δυνατότητα πληθώρας ρυθμίσεων που δύνανται να καταστήσουν οποιαδήποτε προσομοίωση αληθοφανή αναλόγως των απαιτήσεών της. Σε γενικές γραμμές όμως η επίδραση της παραμέτρου αυτής είναι πανομοιότυπη με αυτή της ταχύτητας των δύο πρακτόρων.

## 5.4 Ακολουθία Διαδρομού

Η υλοποίηση της συμπεριφοράς αυτής έγινε υπό σημαντικούς περιορισμούς όσον αφορά την κίνηση του πράκτορα αφού αυτός περιορίσθηκε στην κίνησή του επί μίας διαδρομής μεταξύ δύο σημείων που είναι και η εγγύτερη αν προσπαθήσει να παρακάμψει και τα εμπόδια που παρεμβάλλονται.

Το πιο σημαντικό πρόβλημα είναι η αντίληψη της μορφής των εμποδίων σε συνδυασμό με την ψηφιοποίηση του πεδίου κίνησης του πράκτορα. Το πεδίο κίνησης είναι δύο διαστάσεων οπότε και ο βαθμός απεικόνισης μειώνεται αλλά η ανάλυση της παράστασης του πεδίου παραμένει ως σημαντικό πρόβλημα που επιβαρύνει σημαντικά τους υπολογισμούς. Χρησιμοποιήθηκε ως ελάχιστη διάσταση ψηφιοποίησης ένα μικρό ποσοστό της διάστασης του πράκτορα με ικανοποιητικά αποτελέσματα απεικόνισης και ψηφιοποίησης των εμποδίων αλλά με σχετική επιβάρυνση της υπολογιστικής ισχύος και με μεγάλη επιβράδυνση κατά την αναπαράσταση της κίνησης του πράκτορα.

## 5.5 Ευθυγράμμιση

Κατά την ανάπτυξη της συμπεριφορών "Ευθυγράμμιση" δεν αντιμετωπίσθηκαν ιδιαίτερα προβλήματα στην υλοποίηση του ψευδοκώδικα. Θα πρέπει να σημειώσουμε όμως τα παρακάτω όσον αφορά την παραμετροποίηση των μεταβλητών της συμπεριφοράς:

- Η παράμετρος της ταχύτητας του κινούμενου στόχου και η παράμετρος της ταχύτητας του πράκτορα στόχου είναι βαρύνουσας σημασίας κατά την υλοποίηση της συμπεριφοράς αυτής. Η απόκριση της συμπεριφοράς αυξάνεται με το μέγεθος της ταχύτητας των



δύο πρακτόρων. Θα πρέπει όμως να σημειωθεί όμως ότι για την επιτυχή ολοκλήρωση της συμπεριφοράς θα πρέπει ο κινούμενος πράκτορας να έχει μεταβλητή ταχύτητα κατά τη διάρκεια του αλγόριθμου. Αυτό σημαίνει ότι ο κινούμενος πράκτορας θα πρέπει να διαθέτει μία ταχύτητα που θα μπορεί να αποκαλείται και ταχύτητα χειρισμού και θα είναι σε κάθε περίπτωση μεγαλύτερη αυτή που διαθέτει ο πράκτορας στόχος. Ο κινούμενος πράκτορας θα χρησιμοποιεί την ταχύτητα χειρισμού μέχρις ότου έλθει σε μία κατάλληλη απόσταση όπου μπορεί να τη μειώσει ή να την αυξήσει ανάλογα κατάλληλα. Αυτό γίνεται γιατί σε περίπτωση σταθερής ταχύτητας του κινούμενου πράκτορα που είναι όμως μικρότερη από τη ταχύτητα του στόχου τότε στην πλειάδα των εφαρμογών η συμπεριφορά δεν ολοκληρώνεται επιτυχώς. Στο σημείο αυτό υπεισέρχεται ο παράγοντας της σχετικής ταχύτητας και ο παράγοντας της σύγκλισης της κίνησης των δύο πρακτόρων.

- Η εξέταση της παραμέτρου της απόστασης ενεργοποίησης της συμπεριφοράς εξετάζεται σε συνδυασμό με την σχετική ταχύτητα των δύο πρακτόρων. Η ρύθμιση της σχετικής ταχύτητας με τη σειρά της δεν είναι απευθείας δυνατή αφού αυτή με τη σειρά της εξαρτάται από τις υφιστάμενες για κάθε στιγμή τιμές των ταχυτήτων των δύο πρακτόρων. Έτσι στην περίπτωση της συγκλίνουσας σχετικής ταχύτητας προσέγγισης των δύο πρακτόρων, η απόσταση ενεργοποίησης του αλγόριθμου δύναται να λάβει και χαμηλές τιμές (ιδιαίτερα εγγύς του ακίνητου στόχου) ενώ στην περίπτωση της αποκλίνουσας σχετικής ταχύτητας τότε η απόσταση ενεργοποίησης του αλγόριθμου θα πρέπει να είναι αυξημένη ώστε να δίνεται το επαρκές περιθώριο αντίδρασης και ο κινούμενος πράκτορας να είναι συνεχώς εντός του πεδίου δράσης του αλγόριθμου.
- Η παράμετρος πρόσμιξης της υπολογισμένης αντίδρασης του πράκτορα στη συνολική του κίνηση δεν μελετήθηκε σε βάθος αφού ο πυρήνας δίνει τη δυνατότητα πληθώρας ρυθμίσεων που δύναται να καταστήσουν οποιαδήποτε προσομοίωση αληθοφανή αναλόγως των απαιτήσεών της. Σε γενικές γραμμές όμως η επίδραση της παραμέτρου αυτής είναι πανομοιότυπη με αυτή της ταχύτητας των δύο πρακτόρων.

## 5.6 Άφιξη

Κατά την ανάπτυξη της συμπεριφοράς "Άφιξη" δεν αντιμετωπίστηκαν ιδιαίτερα προβλήματα στην υλοποίηση του ψευδοκώδικα. Θα πρέπει να σημειώσουμε όμως τα παρακάτω όσον αφορά την παραμετροποίηση των μεταβλητών της συμπεριφοράς:

- Η παράμετρος της ταχύτητας του κινούμενου πράκτορα είναι βαρύνουσας σημασίας κατά την υλοποίηση της συμπεριφοράς αυτής. Η απόκριση της συμπεριφοράς είναι αντιστρόφως ανάλογη με το μέγεθος της ταχύτητας του κινούμενου πράκτορα. Δεν παρατηρήθηκε ιδιαίτερη επιβάρυνση της απασχόλησης του υπολογιστικού συστήματος παρά μονάχα δραστικές μεταβολές στην απεικόνιση του κινούμενου πράκτορα σε κάθε νέο στιγμιότυπο.
- Η εξέταση της παραμέτρου της απόστασης ενεργοποίησης της συμπεριφοράς εξετάζεται σε συνδυασμό με την ταχύτητα του κινούμενου πράκτορα. Η ρύθμιση του συνδυασμού των δύο αυτών παραμέτρων θα πρέπει να ρυθμίζεται ανάλογα για να έχουμε αληθοφανή αποτελέσματα. Έτσι στην περίπτωση της μεγάλης ταχύτητας του κινούμενου πράκτορα, η απόσταση ενεργοποίησης του αλγόριθμου δύνανται να λάβει και χαμηλές τιμές (ιδιαίτερα εγγύς του ακίνητου στόχου) ενώ στην περίπτωση της μικρής ταχύτητας κίνησης τότε η απόσταση ενεργοποίησης του αλγόριθμου θα πρέπει να είναι αυξημένη ώστε να δίνεται το επαρκές περιθώριο αντίδρασης.
- Η παράμετρος πρόσμιξης της υπολογισμένης αντίδρασης του πράκτορα στη συνολική του κίνηση δεν μελετήθηκε σε βάθος αφού ο πυρήνας δίνει τη δυνατότητα πληθώρας ρυθμίσεων που δύνανται να καταστήσουν οποιαδήποτε προσομοίωση αληθοφανή αναλόγως των απαιτήσεών της. Σε γενικές γραμμές όμως η επίδραση της παραμέτρου αυτής είναι πανομοιότυπη με αυτή της ταχύτητας του κινούμενου πράκτορα.

□

# Κεφάλαιο 6

## Προτάσεις

Οι περιορισμοί που επιβάλλονται κατά την διάρκεια της εκπόνησης μίας διπλωματικής εργασίας, τόσο χρονικοί όσο και εμβάθυνσης, είναι οι κυριότερη αιτία της περιορισμένης και υπό συνθήκες εξέτασης της υλοποίησης και παραμετροποίησης όλων των πιθανών καθοδηγητικών συμπεριφορών ενός αυτόνομου πράκτορα. Με δεδομένη τη σπουδαιότητα του ερευνητικού αυτού πεδίου στην ολοκλήρωση του πυρήνα της εικονικής πραγματικότητας κρίνεται αναγκαία η επέκταση και συνέχεια της έρευνας σε θέματα όπως τα παρακάτω:

- Υλοποίηση και ανάπτυξη και των υπολοίπων καθοδηγητικών συμπεριφορών που χαρακτηρίζονται ως πιο σύνθετες και θα ολοκληρώσουν τον αυτόνομο χαρακτήρα ενός νοήμονα πράκτορα. Χαρακτηριστικά αναφέρονται οι εξής:

### **Μη Ευθυγραμμισμένη Αποφυγή Σύγκρουσης (Unaligned Collision Avoidance)**

Τα οχήματα κινούνται που κινούνται σε τυχαίες διευθύνσεις, ρυθμίζονται με τη συμπεριφορά αυτή ώστε να μην συγκρούονται. Η "Αποφυγή Συγκρούσεως" στη περίπτωση αυτή περιλαμβάνει την πρόβλεψη των πιθανών συγκρούσεων και την δυναμική λήψη αποφάσεων για τις αλλαγές στη πορεία και τη ταχύτητα του οχήματος. Εάν υφίσταται μία ομάδα οχημάτων και είναι ευθυγραμμισμένα τότε μπορεί να εφαρμοσθεί μία απλή τακτική που καλείται "Διαχωρισμός" (Separation). Ο "Διαχωρισμός" παρέχει τη δυνατότητα της διατήρησης μίας απόστασης διαχωρισμού από τα γειτονικά οχήματα.

### Ακολουθία Αρχηγού (Leader Following)

Η "Ακολουθία Αρχηγού" (Leader Following) βασίζεται σε μεγάλο βαθμό στην ανάμιξη της συμπεριφοράς "Αφιξη" με την "Έκκεντρη Καταδίωξη" (σε θέση λίγο πιο πίσω από τον αρχηγό). Επιπλέον ο "Διαχωρισμός" χρησιμοποιείται για να περιορίσει το φαινόμενο της πύκνωσης μεταξύ των οχημάτων που ακολουθούν τον Αρχηγό.

- Αναγωγή της υλοποίησης των συμπεριφορών από το χώρο των δύο διαστάσεων στο χώρο των τριών διαστάσεων. Η αναγωγή αυτή θα μπορούσε να έχει άμεσα και εφαρμόσιμα αποτελέσματα σε ένα πλήρες μοντέλο ενός πλοίου τριών διαστάσεων όπου οι πράκτορες θα μπορούν να κινούνται άνετα από κατάστρωμα σε κατάστρωμα αλλά και από την πρύμνη προς την πλώρη μέσω διαφορετικών καταστρωμάτων.
- Βελτίωση του αλγόριθμου της ακολουθίας διαδρόμου με την δυνατότητα κινήσεως επί καμπύλων γραμμών ή και παραμετρικών ευθειών που προσεγγίζουν καμπύλες διαδρομές. Επίσης θα μπορούσε να βελτιωθεί η ψηφιοποίηση του χωρίου κίνησης του πράκτορα με την προσθήκη βελτιωμένων αλγόριθμων και τεχνικών προβολής των εμποδίων από το χώρο των τριών διαστάσεων στο επίπεδο.
- Διερεύνηση της βελτιστοποίησης πρόσμιξης των διαφόρων συμπεριφορών ώστε ο πράκτορας να αποκτήσει συμπεριφορές που ταιριάζουν και αναλογούν σε ανθρώπινες συμπεριφορές, όπως είναι αυτή του ενήλικα, του μεσήλικα, του παιδιού, του ηλικιωμένου αλλά και του άντρα ή της γυναίκας. Η προσέγγιση αυτή θα προσδώσει αληθινό χαρακτήρα στην επιχειρούμενη προσομοίωση εφόσον γίνει με αυτοματοποιημένο τρόπο που θα βασίζεται σε τυχόν αναλογίες και δυνατότητες.

□

# Βιβλιογραφία

- [1] Baert S. Motion planning using potential fields. *www.gamedev.net/reference/articles/article1125.asp*, July 2000.
- [2] Bles W., Nooy S., and Boer L. C. Influence of Ship Listing and Ship Motion on Walking Speed. *Conference on Pedestrian and Evacuation Dynamics, April 4-6, 2001, Duisburg*, 2001.
- [3] Blumberg, Bruce and Galyean, Tinsley. Multi-Level Direction of Autonomous Creature for Real-Time Virtual Environments. *Proceedings of SIGGRAPH 95, in Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, σελίδες 47–54, 1995.
- [4] Bouvier E. and Guilloteau P. Crowd Simulation in Immersive Space Management. *Proc. Eurographics Workshop on Virtual Environments and Scientific Visualization '96*, σελίδες 104–110, 1996. Springer-Verlag.
- [5] Chryssolouris G., Mavrikios D., Karabatsou V., Fragos D., Sarris I., and Mourtzis D. *A Virtual Reality Based Approach for the Verification of Human Related Factors in Assembly and Maintenance Processes*. In Proceedings of the Computer-Aided Ergonomics and Safety Conference, CAES '99, 1999.
- [6] Dantzig G. B. On the shortest route through a network. *Management Science*, 17:187–190, 1960.
- [7] Dijkstra E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

- [8] Emmelkamp P. M. G. Technological Innovations in Clinical Assessment and Psychotherapy. *Psychotherapy & Psychosomatics*, 74:336–343, 2005.
- [9] Emmelkamp P. M. G. Virtual Reality in Psychotherapy: Commentary. *CyberPsychology & Behavior*, 8:235–237, 2005.
- [10] Good M. and Tan L. VR in Architecture: Today's use and Tomorrow's Promise. *Virtual Reality World*, 2(6):58–64, 1994.
- [11] Gottshalk S., Lin M., and Manocha D. Obb-tree: A hierarchical structure for rapid interference detection. *Proceedings of ACM Siggraph '96*, σελίδες 171–180, 1996.
- [12] Helbing D., Farkas I., and Viscek T. Simulating Dynamical Features of Escape Panic. *Letters to Nature*, 407:487–490, 2000.
- [13] Holmes P. and Jungert E. R. A. Symbolic and geometric connectivity graph methods for route planning in digitized maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(5):549–565, 1992.
- [14] Kimmel R., Amir A., and Bruckstein A. M. Finding shortest paths on surfaces using level sets propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):635–640, 1995.
- [15] Krijn M. *Virtual Reality and Specific Phobias - Welcome to the Real World*. Διαποικιακή Διατριβή, University of Amsterdam, 2006.
- [16] Krueger M. . *Artificial Reality I*. Addison-Wesley, New York, 1983.
- [17] Krueger M. . *Artificial Reality II*. Addison-Wesley, New York, 1991.
- [18] LeBonG. . *Psychologie des Foules*. Alcan, Paris, 1895.
- [19] Lee P., Phillips C., Otani E., and Badler N. I. The Jack Interactive Human Model. *In Concurrent Engineering of Mechanical Systems*, 1:179–198, 1989. First Annual Symposium on Mechanical Design in a Concurrent Engineering Environment, Univ. of Iowa, Iowa City.

- [20] Leigh J., Vasilakis C., DeFanti T., Grossman R., Assad C., Rasnow B., Protopapas A., DeSchutter E., and Bower J. Virtual Reality in Computational Neuroscience. *Proceedings of the Conference on Applications of Virtual Reality*, 1994. British Computer Society.
- [21] McPhail C., Powers W. T., and Tucker C. W. Simulating Individual and Collective Actions in Temporary Gatherings. *Social Science Computer Review*, 10(1):1–28, 1992.
- [22] Merriam T., Webster K. *Webster's New Universal Unabridged Dictionary*. Merriam-Webster Inc., New York, 1989.
- [23] Mirtich B. V-clip: Fast and robust polyhedral collision detection. *ACM Transactions On graphics*, 17(3), 1998.
- [24] Moeckel G., Keil M., Spiegelhalter B., and Brickmann J. Vml in Cancer Research Local Molecular Properties of the P53 Tumor Suppressor Protein-DNA Interface. *Journal of Molecular Modeling*, 2(9):370–372, 1996. British Computer Society.
- [25] Montgomery M. et al. Navigation algorithm for a nested hierarchical system of robot path planning among polyhedral obstacles. *Proceedings IEEE International conference on Robotics and Automation*, σελίδες 1616–1622, 1987.
- [26] Musse S. *Human Crowd Modelling with Various Levels of Behaviour Control*. Διδακτορική Διατριβή, EPFL, Lausanne, 2000.
- [27] Parkinson C., Cooper M., Hewitt W., and Hillier I. Mavis: An Interactive Visualization Tool for Computational Chemistry Calculations in a Distributed Networked Environment. *Pacific Symposium on Biocomputing*, 3:189–200, 1998. British Computer Society.
- [28] Phillips C. and Badler N. I. *Jack: A toolkit for Manipulating Articulated Figures*. In ACM/SIGGRAPH Symposium on User Interface Software, Banff, Canada, 1988.

- [29] Reynolds C. W. Steering behaviors for autonomous characters. *GDC'99 (Game Developers Conference)*, 1999.
- [30] Shi J., Smith T. J., Granieri J., and Badler N. *Smart avatars in jackmoo*. In Proc. Virtual Reality '99, IEEE Computer Society, Houston, TX, 1999.
- [31] State A., Hirota G., Chen D. T., Garrett W. F., and Livingston M. A. Superior Augmented-Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. *Proceedings of SIGGRAPH 96 (New Orleans, LA, August 4-9, 1996)*, σελίδες 429–438, 1996. ACM SIGGRAPH.
- [32] State A., Livingston M. A., Hirota G., Garrett W. F., Whitton M. C., Fuchs H., and Pisano E. D. Technologies for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. *Proceedings of SIGGRAPH 96 (New Orleans, LA, August 4-9, 1996)*, σελίδες 439–446, 1996. ACM SIGGRAPH.
- [33] Still G. K. *Crowd Dynamics*. Διδακτορική Διατριβή, Warwick University, 2000.
- [34] Szalavari Z., Eckstein E., and Gervautz M. Collaborative Gaming in Augmented Reality. In *Proceedings of VRST'98*, σελίδες 195–204, 1998.
- [35] Thompson P. A. and Marchant E. W. A Computer-model for the Evacuation of Large Building Population. *Fire Safety Journal*, 24(2):131–148, 1995.
- [36] Varner D. et al. USMC Small Unit Leader Non-Lethal Trainer. *Proc. ITEC'98*, 1998.
- [37] Vassalos D., Kim H., Christiansen G., and Majumder J. A Mesoscopic Model for Passenger Evacuation in a Virtual Ship-Sea Environment and Performance-based Evaluation. *Conference on Pedestrian and Evacuation Dynamics, April 4-6, 2001, Duisburg*, 2001.
- [38] Wernecke J. and O. I. A. Group. *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor*. Addison-Wesley Professional, New York, 1994.



- [39] Williams J. R. *A Simulation Environment to Support Training for Large Scale Command and Control Tasks*. Διδακτορική Διατριβή , University of Leeds, 1995.
- [40] Winfield A. A Virtual Laboratory Notebook for Simulation Models. *Pacific Symposium on Biocomputing*, 3:177–188, 1998. British Computer Society.