

Προσομοίωση Υπολογιστή Βασισμένου σε Μυρμήγκια

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Αναστάσιος Γιαννακίδης

Επιβλέπων: Δρ. Λοΐζος Μιχαήλ
Καθηγητής Α.Π.ΚΥ.

Λευκωσία, Ιούλιος 2011

Ευχαριστίες

Η παρούσα μεταπτυχιακή διατριβή εκπονήθηκε κατά το ακαδημαϊκό έτος 2010 - 2011 στα πλαίσια του μεταπτυχιακού προγράμματος στα Πληροφοριακά Συστήματα του Ανοικτού Πανεπιστημίου Κύπρου.

Θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή κ. Λοΐζο Μιχαήλ για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου την εργασία αυτή και για τη δυνατότητα που μου έδωσε να ασχοληθώ με το συγκεκριμένο ενδιαφέρον αντικείμενο καθώς και για τις ατέλειωτες ώρες πολύτιμων συμβουλών και υποδείξεων. Να ευχαριστήσω τέλος την οικογένεια μου και τους φίλους μου, που με στήριξαν και με βοήθησαν, ο καθένας με τον τρόπο του, κατά τη διάρκεια αυτού του μεταπτυχιακού προγράμματος.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Αντικείμενο της μελέτης	1
1.2 Σκοπός της εργασίας	1

Κεφάλαιο 2

Μοντελοποίηση

2.1 Μυρμήγκια και φερομόνες	3
2.2 Αλγόριθμος κίνησης μυρμηγκιών	7
2.3 Χτίζοντας ένα αντιστροφή	9
2.4 Βασικές λογικές πύλες	14

Κεφάλαιο 3

Προσομοίωση

3.1 Περιβάλλον ανάπτυξης	17
3.2 Δομές δεδομένων	19
3.3 Σχεδιαστικές αποφάσεις	24
3.4 Σημαντικοί αλγόριθμοι	26
3.5 Βελτιώσεις	35

Κεφάλαιο 4

Γραφικό Περιβάλλον

4.1 Περιβάλλον σχεδίασης	39
4.2 Δομή αρχείων	39
4.3 Σχεδιαστικές αποφάσεις	41
4.4 Διεπαφή με το χρήστη	47

Κεφάλαιο 5

5.1 Προσομοιωτές με μυρμήγκια	53
5.2 Συμπεράσματα	56
5.3 Μελλοντικές Προσθήκες Βελτιώσεις	56

Βιβλιογραφία	58
---------------------	----

Παράρτημα	59
------------------	----

Εγχειρίδιο χρήσης	68
--------------------------	----

Κεφάλαιο 1

Εισαγωγή

1.1 Αντικείμενο της μελέτης

Αλγόριθμοι που είναι εμπνευσμένοι από τη φύση βασισμένοι στην ευφυΐα του σμήνους [2] (swarm intelligence) καθώς και η δυνατότητα αυτό-οργάνωσης (self organized behavior) των κοινωνικών εντόμων μπορούν να λύσουν πολλά πολύπλοκα προβλήματα όπως το πρόβλημα του περιοδεύοντος πωλητή καθώς και την επαναδρομολόγηση (rerouting) της κίνησης σε ένα δίκτυο με μεγάλη κυκλοφορία. Αυτοί οι τύποι των αλγορίθμων είναι μόνο ένα μικρό κομμάτι από τους αλγόριθμους που είναι εμπνευσμένοι από την βιολογία (biology inspired). Οι αλγόριθμοι που είναι εμπνευσμένοι από τη βιολογία περιέχουν γενετικούς αλγόριθμους, νευρωνικά δίκτυα και άλλους αλγόριθμους.

Έρευνες που έχουν γίνει σε πραγματικά μυρμήγκια έχουν δείξει ότι αυτά αντιδρούν και κινούνται στο περιβάλλον τους με πολύ συγκεκριμένο τρόπο χωρίς αυτά να κάνουν κάποιους πολύπλοκους υπολογισμούς.

Συγκεκριμένα τα μυρμήγκια μπορούν να εκτελέσουν πολύ απλούς αλγόριθμους χωρίς την χρήση στοιχείων μνήμης. Η επικοινωνία μεταξύ τους είναι πολύ περιορισμένη και ο τρόπος επικοινωνίας γίνεται με έμμεσο τρόπο αντίδρασης τους με το περιβάλλον τους.

1.2 Σκοπός της εργασίας

Στο σύγγραμμα αυτό μελετάται ένα βιολογικό μοντέλο [1] κίνησης των μυρμηγκιών που προτείνεται από τη βιβλιογραφία. Μελέτες έχουν δείξει ότι τα μυρμήγκια μπορούν να λύνουν δύσκολα προβλήματα, όπως εύρεση της πιο σύντομης διαδρομής μεταξύ δύο σημείων [3] ή ταξινόμηση διαφόρων πραγμάτων σε σωρούς [4]. Παρόλο που τα μυρμήγκια μπορούν να λύσουν τόσο δύσκολα προβλήματα, εντούτοις αυτά δεν μπορούν να κάνουν υπολογισμούς. Σε αυτή τη εργασία θα επιχειρήσουμε τη δημιουργία ενός προσομοιωτή του βιολογικού μοντέλου που προτείνεται από τη βιβλιογραφία. Θα δείξουμε ότι εάν εφαρμόσουμε τον αλγόριθμο κίνησης τους σε ένα ειδικά διαμορφωμένο περιβάλλον τότε μπορούμε να

κτίσουμε ένα υπολογιστή από μυρμήγκια. Στα κεφάλαια που ακολουθούν παρουσιάζεται το μοντέλο κίνησης των μυρμηγκιών στο οποίο κτίστηκε ο προσομοιωτής μας. Περιγράφονται οι βασικές αρχές του και τα συστατικά του. Ακολούθως γίνεται μια λεπτομερής ανάπτυξη του πως μπορούμε να αναπαραστήσουμε το μοντέλο αυτό σε ηλεκτρονικό υπολογιστή. Περιγράφονται προβλήματα που προέκυψαν, λύσεις που δόθηκαν καθώς και σημεία εμπλουτισμού του δοσμένου μοντέλου. Τέλος ακολουθεί περιγραφή του γραφικού περιβάλλοντος.

Κεφάλαιο 2

Μοντελοποίηση

Σε αυτό το κεφάλαιο περιγράφουμε αναλυτικά ένα συγκεκριμένο μοντέλο κίνησης των μυρμηγκιών, που προτείνεται από τη βιβλιογραφία. Πρόκειται για το Ant Based Computer [1]. Το μοντέλο αυτό προτείνει ένα συγκεκριμένο τρόπο καθοδήγησης των μυρμηγκιών εκμεταλλευόμενο την ιδιότητα τους να έλκονται από τη φερομόνη. Προτείνει τη δημιουργία βασικών κυκλωμάτων χρησιμοποιώντας ως πηγή τα μυρμήγκια αντί το ηλεκτρικό ρεύμα και επεκτείνεται σε πολυπλοκότερα κυκλώματα ώστε να είμαστε σε θέση να αναφερόμαστε σε υπολογιστή που δουλεύει με αυτά. Ακολουθεί λεπτομερής περιγραφή του μοντέλου αναλύοντας τον τρόπο με τον οποίο η φερομόνη επηρεάζει τη επιλογή κατεύθυνσης των μυρμηγκιών και πως αυτή διαδίδεται. Επίσης αναλύονται τα θεμελιώδη συστατικά που προτείνονται από το μοντέλο και παρουσιάζονται παραδείγματα κυκλωμάτων υλοποιημένων με αυτό.

2.1 Μυρμήγκια και φερομόνες

Η φερομόνη είναι μια χημική ουσία την οποία εκκρίνουν τα ζώα και τα έντομα για να επικοινωνήσουν έμμεσα με τα άλλα μέλη του είδους τους. Υπάρχουν φερομόνες κινδύνου, εντοπισμού τροφής, σεξουαλικές και πολλές άλλες που επηρεάζουν τη συμπεριφορά και τη φυσιολογία των ζώων και εντόμων. Η ύπαρξη και η χρήση τους στα έντομα έχει μελετηθεί σε μεγάλο βαθμό από την επιστημονική κοινότητα. Μελέτες που έχουν διεξαχθεί από τον Deneubourg [6], [7], [8] σε πραγματικά μυρμήγκια δείχνουν ότι τα περισσότερα είδη μυρμηγκιών κατά την αναζήτηση και συγκομιδή τροφής εναποθέτουν φερομόνη στο πέρασμά τους. Η φερομόνη αυτή είναι ανιχνεύσιμη και από τα υπόλοιπα μυρμήγκια της αποικίας τους τα οποία αποφασίζουν με κάποια πιθανότητα αν θα ακολουθήσουν το ίχνος αυτό.

Μυρμήγκια

Τα βασικά στοιχεία δημιουργίας ενός υπολογιστή μυρμηγκιών είναι τα μυρμήγκια και οι φερομόνες. Θα παρουσιάσουμε εδώ το μοντέλο πάνω στο οποίο θα βασίσουμε τον προσομοιωτή καθώς και τυχόν παραδοχές που θα ισχύουν για όλη την εργασία.

Ένα μυρμήγκι χαρακτηρίζεται από τη θέση και την κατεύθυνση που έχει μια συγκεκριμένη χρονική στιγμή. Ονομάζουμε την τρέχουσα θέση του μυρμηγκιού L_c (Current Location) και την κατεύθυνση του D_c (Current Direction). Οι προσπελάσιμες γειτονικές θέσεις $R(L_c, D_c)$ ενός μυρμηγκιού ορίζονται ως οι θέσεις που μπορεί να κινηθεί ένα μυρμήγκι από την τρέχουσα του θέση προς αυτές σε μια χρονική στιγμή. Τα μυρμήγκια αλληλεπιδρούν με το περιβάλλον τους με το να ανιχνεύουν τη συσσώρευση φερομόνης στην παρούσα αλλά και στις γειτονικές τους θέσεις. Το μυρμήγκι επιλέγει αν πρέπει να εκκρίνει ποσότητα φερομόνης στην παρούσα του θέση και επιλέξει μία από τις γειτονικές προσπελάσιμες θέσεις για να κατευθυνθεί προς αυτή την επόμενη χρονική στιγμή. Το πως το μυρμήγκι επιλέγει μια από τις προσπελάσιμες γειτονικές του θέσεις γίνεται βάση μιας κατανομής πιθανοτήτων που είναι η ακόλουθη.

$$P(LN)^n / \sum_{L \in R(L_c, D_c)} P(L)^n$$

Όπου $P(LN)^n$ η συγκέντρωση φερομόνης της κάθε υπό εξέταση γειτονικής θέσης και $\sum_{L \in R(L_c, D_c)} P(L)^n$ το άθροισμα συγκέντρωσης της φερομόνης των γειτονικών προσπελάσιμων σημείων. Η παράμετρος n καθορίζει το βαθμό μη γραμμικότητας της συνάρτησης επιλογής. Μεγάλη τιμή του n συνεπάγεται ότι μικρή διαφοροποίηση στα επίπεδα φερομόνης μεταξύ υποψήφιων γειτονικών προσπελάσιμων θέσεων, αυξάνει κατά πολύ την πιθανότητα επιλογής της θέσης με την μεγαλύτερη συγκέντρωση.

Φερομόνες

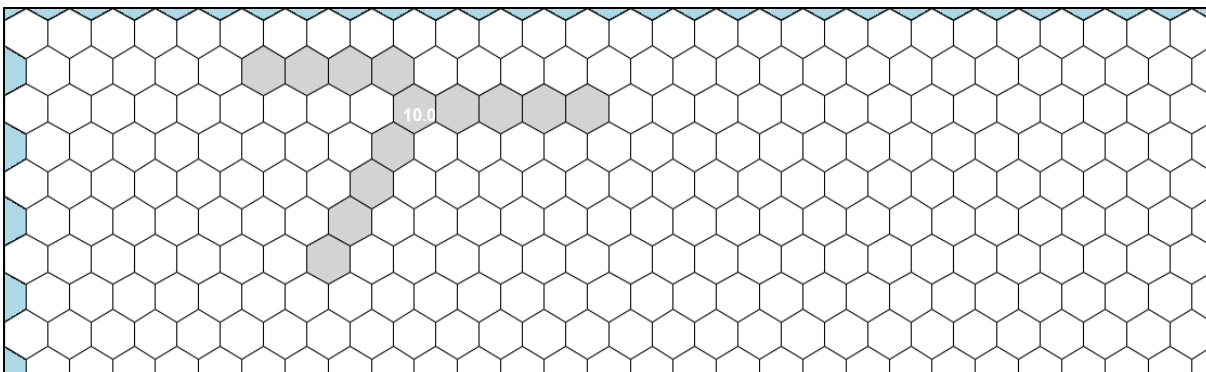
Τα μυρμήγκια έλκονται από τη φερομόνη. Εκτός του ότι τείνουν να κατευθύνονται προς τα σημεία που υπάρχει μεγαλύτερη συγκέντρωση φερομόνης μπορούν επίσης να αισθανθούν εάν η ποσότητα συγκέντρωσης είναι μικρότερη ή μεγαλύτερη μιας σταθερής ποσότητας T (Threshold). Τα μυρμήγκια μπορούν να αναγνωρίσουν με επιτυχία εάν μια ποσότητα συγκέντρωσης είναι χαμηλή ή ψηλή και αν αυτή είναι αρκετά μακριά από το Threshold. Θα ονομάσουμε την απόσταση αυτή ϵ .

Στο μοντέλο που παρουσιάζουμε, αν ένα μυρμήγκι ανιχνεύσει μεγάλη ποσότητα συγκέντρωσης φερομόνης στην παρούσα του θέση, μεγαλύτερη από $T+\epsilon$ και χαμηλή συγκέντρωση μικρότερη $T+\epsilon$ από σε μια από τις προσπελάσιμες θέσεις τότε το μυρμήγκι θα αφήσει μια ποσότητα φερομόνης Ph_i στην παρούσα του θέση.

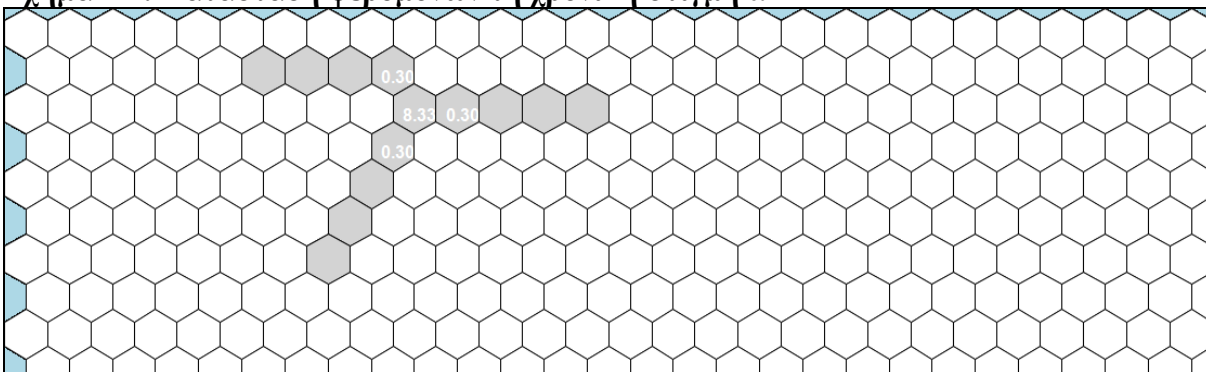
Λόγω διάχυσης η φερομόνη θα προχωρήσει προς τις γειτονικές θέσεις. Η συγκέντρωση της φερομόνης θα αυξηθεί σε κάποια σημεία και θα κινηθεί από τα σημεία που έχει υψηλή συγκέντρωση στα σημεία που υπάρχει χαμηλότερη. Λαμβάνουμε επίσης υπόψη μας ότι κάθε χρονική στιγμή μια ποσότητα φερομόνης εξατμίζεται στο περιβάλλον.

Παρουσιάζουμε αναλυτικότερα τον τρόπο διάδοσης της φερομόνης με ένα παράδειγμα: Έστω ότι είμαστε στη χρονική στιγμή t και έχουμε ένα σημείο με συγκέντρωση φερομόνης ίση με 10 μονάδες. Όπως έχει ειπωθεί η φερομόνη θα κινείται από τα σημεία που έχει μεγαλύτερη συγκέντρωση προς τα σημεία που έχουν μικρότερη.

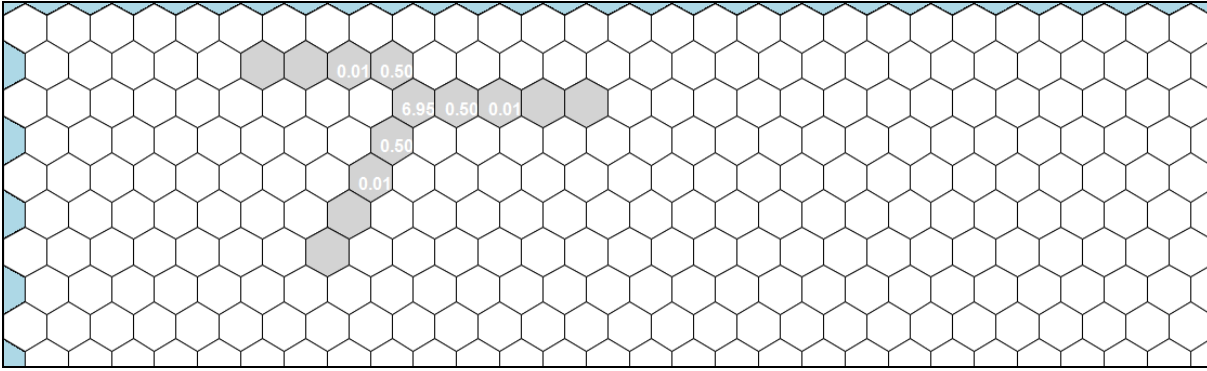
Στα πιο κάτω σχήματα φαίνεται η διάδοση της φερομόνης στις γειτονικές θέσεις. Παρατηρούμε ότι η φερομόνη μεταβάλλεται ανά χρονική στιγμή και κινείται από τις ψηλότερες τιμές συγκέντρωσης προς τις χαμηλότερες. Είναι προφανές ότι σε πεπερασμένο αριθμό χρονικών στιγμών η φερομόνη θα κινείται ομοιόμορφα προς όλα τα γειτονικά σημεία. Λόγω εξάτμισης ένα ποσοστό της θα διαλύεται στην ατμόσφαιρα έως ότου εξαφανισθεί η φερομόνη από το σύστημα.



Σχήμα 2-1. Κατάσταση φερομόνων τη χρονική στιγμή t .



Σχήμα 2-2. Κατάσταση φερομόνων τη χρονική στιγμή $t+1$.



Σχήμα 2-3. Κατάσταση φερομόνων τη χρονική στιγμή $t+2$.

Εκτός από τα μυρμήγκια συγκεκριμένη ποσότητα φερομόνης τροφοδοτούν στο σύστημα κάθε χρονική στιγμή και κάποιες αντλίες. Διαφορετικές αντλίες είναι δυνατό να τροφοδοτούν διαφορετική ποσότητα φερομόνης το σύστημα. Μια αντλία τροφοδοτεί το σύστημα μας με συγκεκριμένη ποσότητα φερομόνης στο σημείο που βρίσκεται ανά μονάδα χρόνου.

Η διάδοση της φερομόνης περιγράφεται με την πιο κάτω εξίσωση.

$$P^t(L) = (1 - d) \cdot [(1 - f) \cdot P^{t-1}(L) + f \cdot W^{t-1}(L)] + s^{t-1}(L) + p(L)$$

Όπου

- **$P^t(L)$** : Η συγκέντρωση φερομόνης στη θέση L τη χρονική στιγμή t .
- **W^{t-1}** : Η μέση συγκέντρωση φερομόνης των γειτονικών θέσεων της θέσης L και της παρούσας θέσης την προηγούμενη χρονική στιγμή.
- **s^{t-1}** : Ποσότητα φερομόνης που εκκρίνεται στη θέση L από ένα μυρμήγκι που ήταν σε αυτή τη θέση την προηγούμενη χρονική.
- **p** : Ποσότητα φερομόνης που τροφοδοτείται στο σύστημα ανά μονάδα χρόνου από αντλία που μπορεί να βρίσκεται στη θέση L .
- **f** : Ρυθμός διάχυσης φερομόνης στο σύστημα.
- **d** : Ποσοστό φερομόνης που εξατμίζεται στο περιβάλλον.

Παρατηρούμε ότι η διάδοση της φερομόνης είναι συνδεδεμένη χρονικά με τις τιμές της φερομόνης στην προηγούμενη κατάσταση. Ο χρόνος που χρειάζεται να μετακινηθεί ένα μυρμήγκι προς μια γειτονική θέση

έχει διάρκεια μια χρονική στιγμή. Ορίζουμε ως χρονική στιγμή t την παρούσα κατάσταση δηλ. την νέα θέση και ως $t-1$ την προηγούμενη κατάσταση. Οι τιμές των φερομόνων υπολογίζονται για όλες τις διαθέσιμες θέσεις του συστήματος μας κάθε μια χρονική στιγμή.

Επομένως η συγκέντρωση της φερομόνης σε μια θέση L εξαρτάται από:

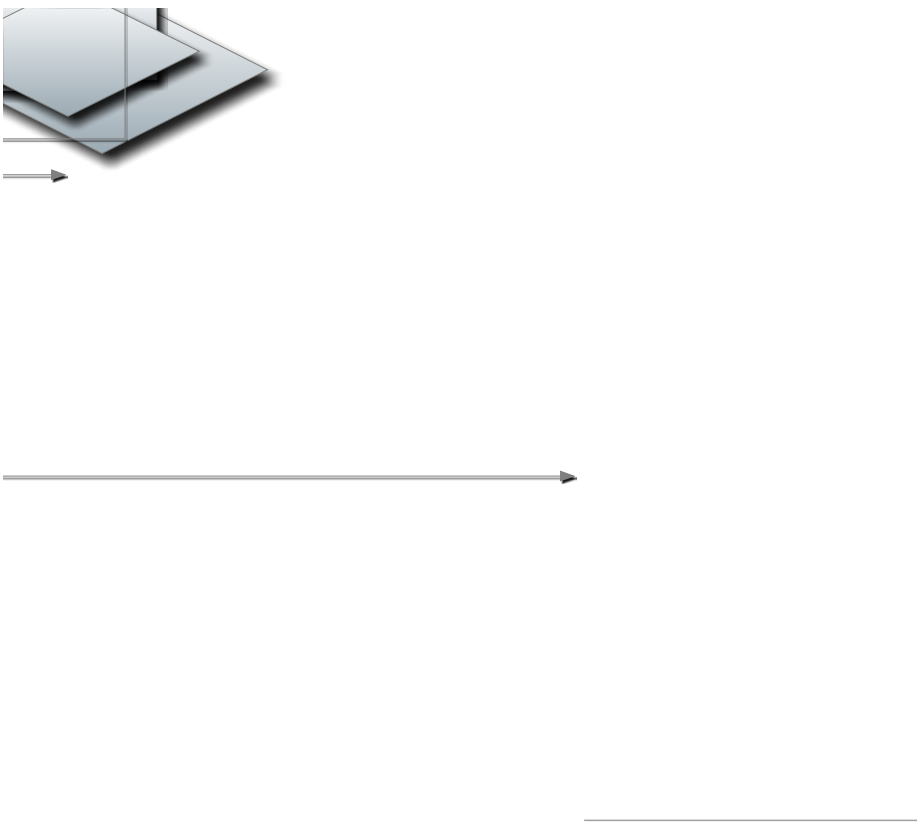
- Τη συγκέντρωση φερομόνης της θέσης αυτής και των γειτονικών της θέσεων την προηγούμενη χρονική στιγμή.
- Την απόφαση ενός μυρμηγκιού που βρίσκεται στο σημείο L την προηγούμενη χρονική στιγμή να εκκρίνει φερομόνη.
- Το ρυθμό διάχυσης της φερομόνης.
- Την πτητικότητα της.
- Την ποσότητα που τροφοδοτεί μια αντλία που βρίσκεται στη θέση L ανά μονάδα χρόνου.

2.2 Αλγόριθμος κίνησης μυρμηγκιών

Στο προηγούμενο υποκεφάλαιο εξετάσαμε το πώς ένα μυρμήγκι επιλέγει να κινηθεί προς τις γειτονικές του θέσεις, πόσο τυχαίο είναι αυτό το γεγονός, πότε εκκρίνει φερομόνη και τον τρόπο που διαδίδεται η φερομόνη στο σύστημα. Στο υποκεφάλαιο αυτό θα δώσουμε μια αναλυτική επεξήγηση του πώς κινούνται τα μυρμήγκια σύμφωνα με το μοντέλο μας και τι επιτυγχάνουμε με αυτό. Επίσης θα προσπαθήσουμε να τοποθετήσουμε χρονικά τα συμβάντα για την καλύτερη κατανόηση της λειτουργίας του μοντέλου κίνησης.

Ας υποθέσουμε ότι έχουμε μια επιφάνεια χωρισμένη σε μικρά τμήματα όπου κάθε σημείο μπορεί να φιλοξενήσει ακριβώς ένα μυρμήγκι. Τοποθετούμε σε αυτή μυρμήγκια σε μερικές θέσεις τέτοιες ώστε στην επόμενη χρονική στιγμή να μην έρθουν σε σύγκρουση μεταξύ τους. Ο αλγόριθμος κίνησης θα λειτουργήσει ως εξής. Θα εντοπίσει τις θέσεις στις οποίες υπάρχουν μυρμήγκια και θα εξετάσει για κάθε μυρμήγκι ποιες είναι οι υποψήφιες θέσεις στις οποίες μπορεί να κινηθεί. Οι θέσεις που μπορεί να κινηθεί ένα μυρμήγκι είναι αυτές που μπορεί να προσεγγίσει με τη πάροδο μιας χρονικής στιγμής, δηλαδή οι εφαιπόμενες θέσεις του σημείου στο οποίο βρίσκεται. Αν το μυρμήγκι για οποιοδήποτε λόγο δεν μπορεί να κινηθεί προς

γειτονική θέση τότε παραμένει ακίνητο. Αν υπάρχει έστω και μια γειτονική θέση με ποσότητα φερομόνης μικρότερη από τη θέση στην οποία βρίσκεται τότε εκκρίνει φερομόνη στη θέση την οποία βρίσκεται. Έπειτα επιλέγεται η θέση στην οποία θα κινηθεί βάση πιθανότητας δίνοντας περισσότερο βάρος στα σημεία που έχουν μεγαλύτερη συγκέντρωση φερομόνης. Κατόπιν υπολογίζεται η διάδοση της φερομόνης για όλες τις θέσεις της επιφάνειας. Ακολουθεί διάγραμμα ροής το οποίο περιγράφει τον αλγόριθμο κίνησης των μυρμηγκιών όπως αυτό περιγράφεται στο βιολογικό μοντέλο που εξετάζουμε.



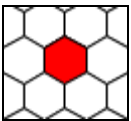
Σχήμα 2-4 Αλγόριθμος κίνησης μυρμηγκιών.

2.3 Χτίζοντας ένα Αντιστροφέα

Σύμφωνα με τη λογική των λογικών κυκλωμάτων δύο είδη λογικών πυλών χρειάζονται για να αναπαραστήσουμε οποιοδήποτε λογικό κύκλωμα. Οι πύλες αυτές είναι οι αντιστροφείς (NOT) και οι πύλες OR. Η λογική πύλη OR υλοποιείται σχετικά απλά στο μοντέλο μας για αυτό θα επικεντρωθούμε στον Αντιστροφέα. Ο Αντιστροφέας στα λογικά κυκλώματα είναι το κύκλωμα που δίνει ως έξοδο το αντίστροφο της εισόδου του. Χρησιμοποιώντας τον σε συνδιασμό με πύλες οι μπορούμε να αναπαραστήσουμε πολύπλοκα λογικά κυκλώματα και αν αυτά συνδυαστούν μπορούμε να φτάσουμε στην παραγωγή ενός υπολογιστή. Επομένως, αναπαριστώντας τη λειτουργία ενός Αντιστροφέα χρησιμοποιώντας μυρμήγκια λύνουμε κατά το ήμισυ το πρόβλημα της δημιουργίας ενός υπολογιστή που να βασίζει την λειτουργία του σε αυτά.

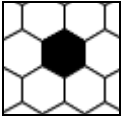
Πριν ξεκινήσουμε με την αναλυτική περιγραφή του Αντιστροφέα είναι αναγκαίο να περιγράψουμε τα θεμελιώδη συστατικά του Ant Based Computer καθώς και το περιβάλλον του.

Ας θεωρήσουμε ένα επίπεδο στρωμένο με εξάγωνα το οποίο από τώρα και στο εξής θα καλούμε επιφάνεια εργασίας ή board. Τα εξάγωνα αυτά τα καλούμε κελιά ή cells. Κάθε κελί εκτός από μυρμήγκια μπορεί να φιλοξενήσει τα εξής:



Σχήμα 2-5 Pump ή Αντλία.

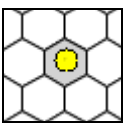
Η αντλία αφήνει συγκεκριμένη ποσότητα φερομόνης στο κελί που βρίσκεται ανά χρονική στιγμή.



Σχήμα 2-6. Brick ή Τουβλάκι.

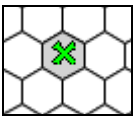
Είναι ένα τουβλάκι σε σχήμα εξαγώνου και λειτουργεί ως φυσικό εμπόδιο στην διέλευση ενός μυρμηγκιού.

Χρησιμοποιείται για να αποτρέψουμε την επίσκεψη ενός μυρμηγκιού σε συγκεκριμένο κελί ή για να οριοθετήσουμε το χώρο στο οποίο κινείται ένα μυρμήγκι ή στο να το κατευθύνουμε να χαράξει μια συγκεκριμένη πορεία στην επιφάνεια εργασίας μας.



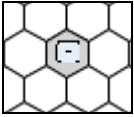
Σχήμα 2-7. Source ή Πηγή.

Η πηγή είναι το μέσο με το οποίο εισάγονται νέα μυρμήγκια στο σύστημα μας. Εισάγει ένα νέο μυρμήγκι στο κελί στο οποίο βρίσκεται ανά μονάδα χρόνου. Η κατεύθυνση που έχει ένα μυρμήγκι μόλις εισαχθεί στο σύστημα είναι μια τυχαία επιλογή από τις διαθέσιμες γειτονικές θέσεις.



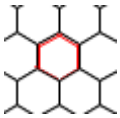
Σχήμα 2-8. Cross ή Γέφυρα.

Η γέφυρα επιτρέπει την ταυτόχρονη είσοδο μυρμηγκιών σε ένα κελί σε μια συγκεκριμένη χρονική στιγμή χωρίς αυτά να συγκρούονται. Η κατεύθυνση εισόδου ενός μυρμηγκιού σε μια γέφυρα είναι η ίδια με την έξοδο του στη γέφυρα δηλαδή το μυρμήγκι κινείται στην απέναντι πλευρά. Μπορούμε να φανταστούμε τη γέφυρα σαν τρία ανεξάρτητα μονοπάτια τα οποία το κάθε ένα συνδέει τις απέναντι κατευθύνσεις.



Σχήμα 2-9. Sink ή Έξοδος.

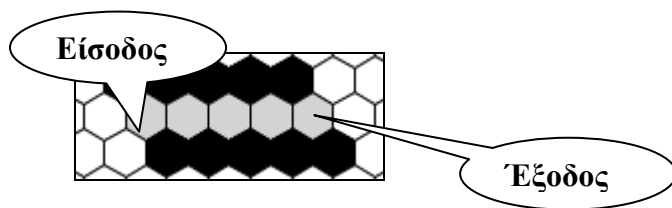
Το μέσο με το οποίο τα μυρμήγκια εξάγονται από το σύστημα μας. Αν ένα μυρμήγκι καταλήξει στην έξοδο μια συγκεκριμένη χρονική στιγμή, στην επόμενη χρονική στιγμή το μυρμήγκι βρίσκεται εκτός επιφάνειας εργασίας και συστήματος γενικότερα.



Σχήμα 2-10. Wall ή Τοίχος.

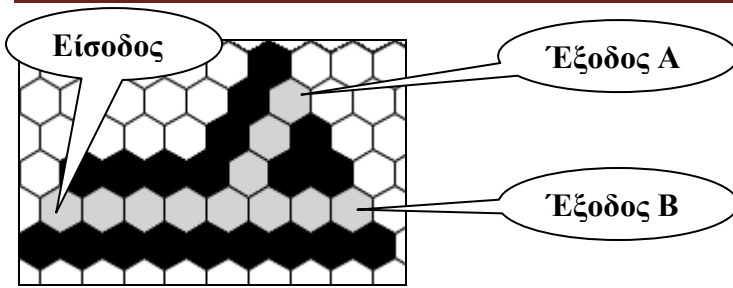
Είναι λεπτοί τοίχοι οι οποίοι απαγορεύουν τη διέλευση ενός μυρμηγκιού προς μια γειτονική του θέση χωρίς να απαγορεύει την διέλευση στα μυρμήγκια που βρίσκονται σε γειτονικές θέσεις να κατευθυνθούν σε αυτή. Ο Τοίχος μπορεί να τοποθετηθεί σε έξι πιθανές θέσεις, όσες και οι πλευρές ενός κελιού. Το θεμελιώδες συστατικό τοίχος δεν υπήρχε στο αρχικό βιολογικό μοντέλο. Κρίθηκε αναγκαία η χρήση του κατά τη φάση της υλοποίηση του θεμελιώδες συστατικού Συνένωσης. Περισσότερες λεπτομέρειες που αφορούν την απόφαση της εισαγωγής του Τοίχου αναπτύσσονται στο κεφάλαιο της Προσομοίωσης.

Συνδυάζοντας Θεμελιώδη συστατικά μπορούμε να δημιουργήσουμε τα εξής:



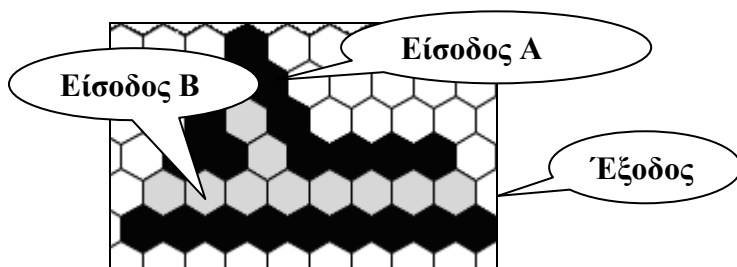
Σχήμα 2-11. Wire ή Καλώδιο.

Είναι το αντίστοιχο του καλωδίου που χρησιμοποιούμε στα λογικά κυκλώματα. Όταν βρεθεί σε αυτό ένα μυρμήγκι είναι αναγκασμένο να ακολουθήσει τη διεύθυνση του μέχρι την έξοδο από αυτό.



Σχήμα 2-12. Split ή Διακλάδωση.

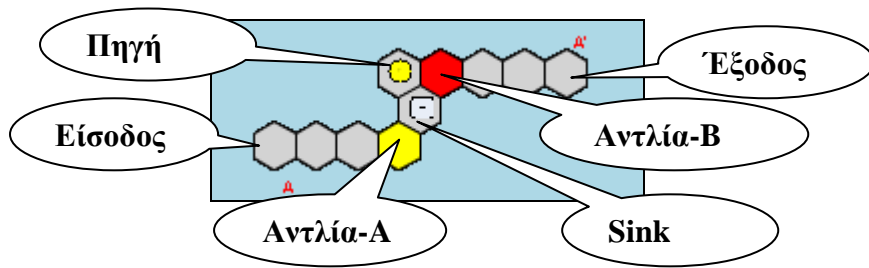
Είναι ουσιαστικά μια διακλάδωση. Ένα μυρμήγκι μπορεί να μπει στην είσοδο και να ακολουθήσει την διαδρομή προς την Έξοδο A ή την Έξοδο B.



Σχήμα 2-13. Merge ή Συνένωση.

Οδηγεί δύο εισόδους σε μια έξοδο. Λόγω της φύσης της λειτουργεί σαν μια πύλη or. Εάν οι συνθήκες είναι τέτοιες ώστε δύο μυρμήγκια διεκδικούν να κατευθυνθούν στην ίδια θέση τότε ένα από τα δύο καταλαμβάνει τη θέση αυτή ενώ το άλλο παραμένει ακίνητο. Η ταυτόχρονη διεκδίκηση μιας θέσης οδηγεί σε σύγκρουση. Περισσότερα περί συγκρούσεων και μηχανισμών αποφυγής σύγκρουσης θα αναπτύξουμε στο επόμενο κεφάλαιο κατά την υλοποίηση του συγκεκριμένου θεμελιώδες συστατικού.

Ο αντιστροφέας μπορεί να υλοποιηθεί παρατάσσοντας τα θεμελιώδη συστατικά του Ant Computer στην επιφάνεια εργασίας μας όπως φαίνεται στο σχήμα 2-13. Συγκεκριμένα ο αντιστροφέας αποτελείται από ένα σύνολο θέσεων διατεταγμένων εξαγώνων. Τα γκρίζα κελιά δηλώνουν το μονοπάτι που τα μυρμήγκια μπορούν να ακολουθήσουν και μέσω των οποίων η φερομόνη μπορεί να διαχέεται. Το κόκκινο και κίτρινο κελί είναι αντλίες οι οποίες εισάγουν συγκεκριμένη ποσότητα φερομόνης στο σύστημα σε κάθε χρονική στιγμή. Σημειώνονται με διαφορετικό χρώμα γιατί η ποσότητα φερομόνης που εισάγεται από τις αντλίες σε κάθε χρονική στιγμή είναι διαφορετική.



Σχήμα 2-14 Αντιστροφείας.

Λειτουργία του Αντιστροφεία

Τα μυρμήγκια μπαίνουν στον αντιστροφεία από την είσοδο και βγαίνουν από την έξοδο. Στην πηγή εμφανίζεται ένα νέο μυρμήγκι κάθε χρονική στιγμή, ενώ οι αντλίες τροφοδοτούν το σύστημα με συγκεκριμένη ποσότητα φερομόνης ανά χρονική στιγμή. Τα μυρμήγκια που παρουσιάζονται στην έξοδο είναι αυτά που εισάγονται στον αντιστροφεία από την πηγή και όχι από την είσοδο.

Όπως αναφέραμε, τα μυρμήγκια έλκονται από τη φερομόνη. Η τροφοδοσία του αντιστροφεία με φερομόνη μέσω της αντλίας-A και της αντλίας-B βοηθά τα μυρμήγκια να ανιχνεύσουν φερομόνη και η φερομόνη με τη σειρά της διαχέεται στις γειτονικές θέσεις. Τα μυρμήγκια ανάλογα της συγκέντρωσης φερομόνης αποφασίζουν κατά πόσο πρέπει να εκκρίνουν φερομόνη στη θέση που βρίσκονται. Όσο υπάρχουν μυρμήγκια στην είσοδο, αυξάνουν την φερομόνη στο συγκεκριμένο μονοπάτι και λόγω διάχυσης αυξάνεται η φερομόνη στο sink. Όσο αυξάνεται η φερομόνη στο sink έλκει τα μυρμήγκια από την πηγή και έτσι δεν οδηγούνται μυρμήγκια στην έξοδο, έχουμε δηλαδή λογικό μηδέν. Στην περίπτωση που δεν έχουμε μυρμήγκια στην είσοδο δηλαδή τροφοδοτούμε τον αντιστροφεία μας με λογικό μηδέν, τα μυρμήγκια που μπαίνουν στο σύστημα μέσω της πηγής, κινούνται προς την αντλία-B διότι η ποσότητα φερομόνης που τροφοδοτείται στο σύστημα από την αντλία-A δεν είναι αρκετή ώστε να τα κάνει να κινηθούν προς αυτή. Έχουμε δηλαδή λογική έξοδο ένα.

2.4 Βασικές λογικές πύλες

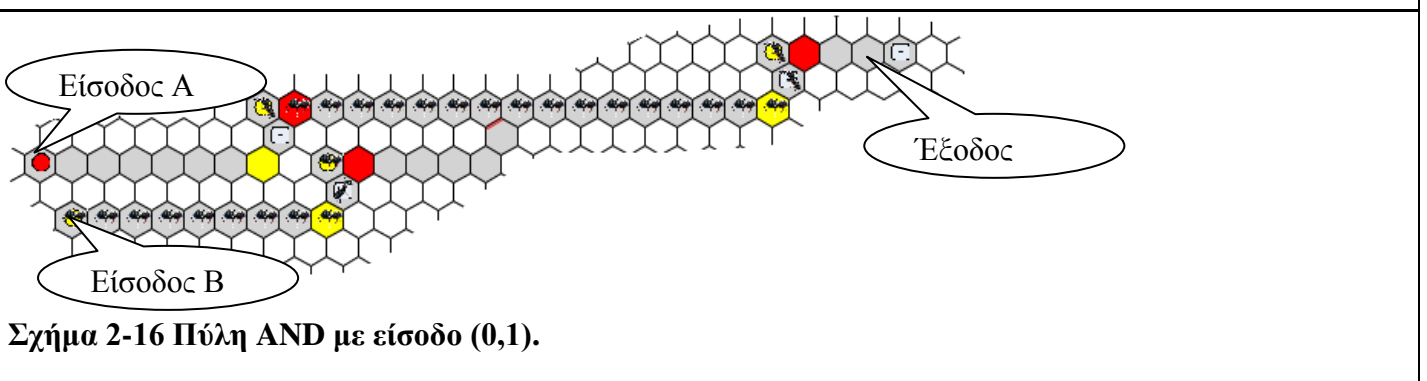
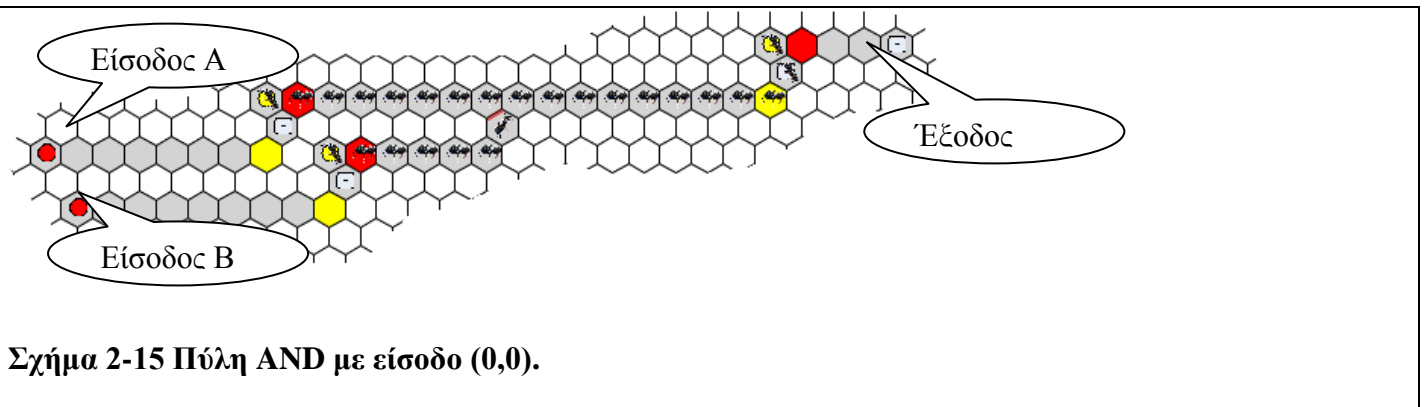
Για τη δημιουργία πυλών στο Ant Based computer ισχύει ότι και στην λογική σχεδίαση συνδυαστικών κυκλωμάτων. Κάθε είσοδος αντιστοιχεί με πηγή και κάθε έξοδος αντιστοιχεί με sink. Ακολουθούν παραδείγματα μερικών βασικών πυλών.

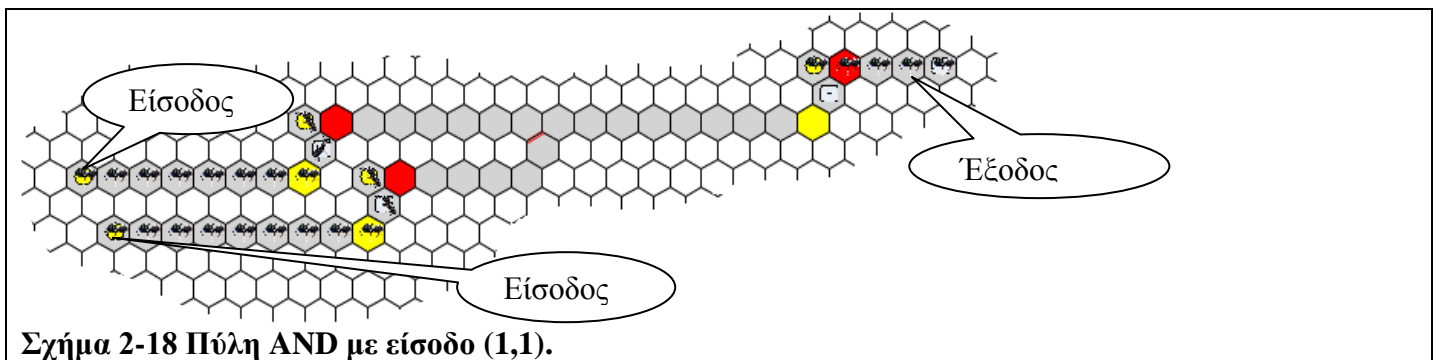
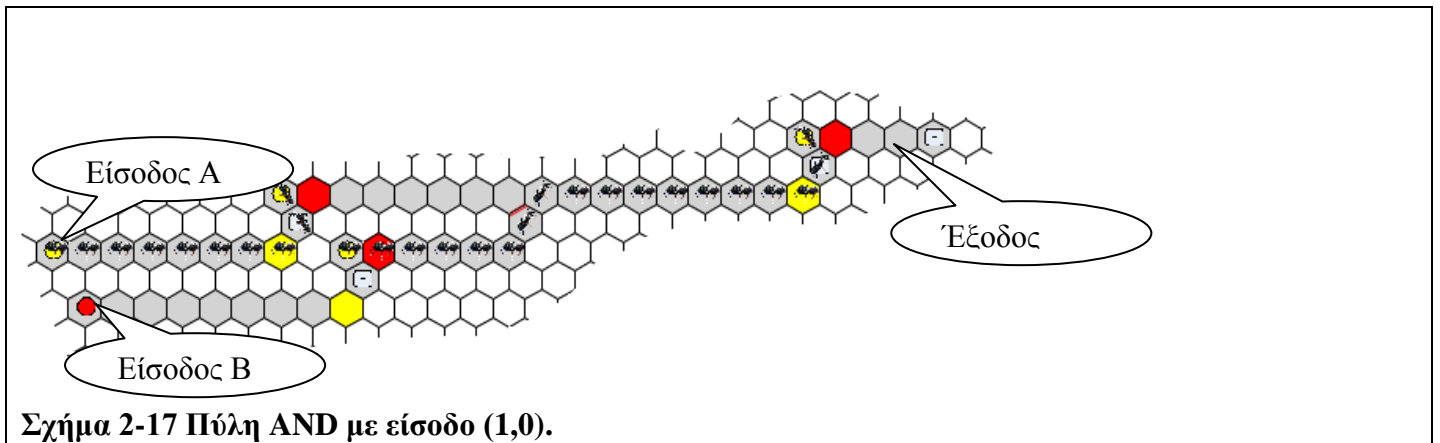
AND GATE Πίνακας Αληθείας		
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Στη παρακάτω υλοποίηση της πύλης AND χρησιμοποιήθηκαν τρεις αντιστροφείς και μια συνένωση.

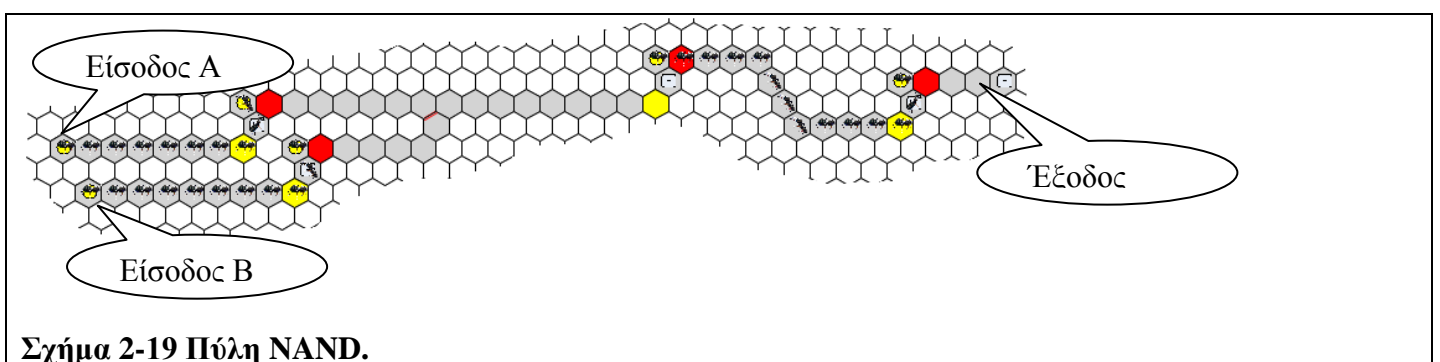
Το λογικό κύκλωμα που αναπαριστάται είναι $(A' + B')' = AB$ (Νόμος De Morgan)

Υλοποιήσεις με Ant Computer

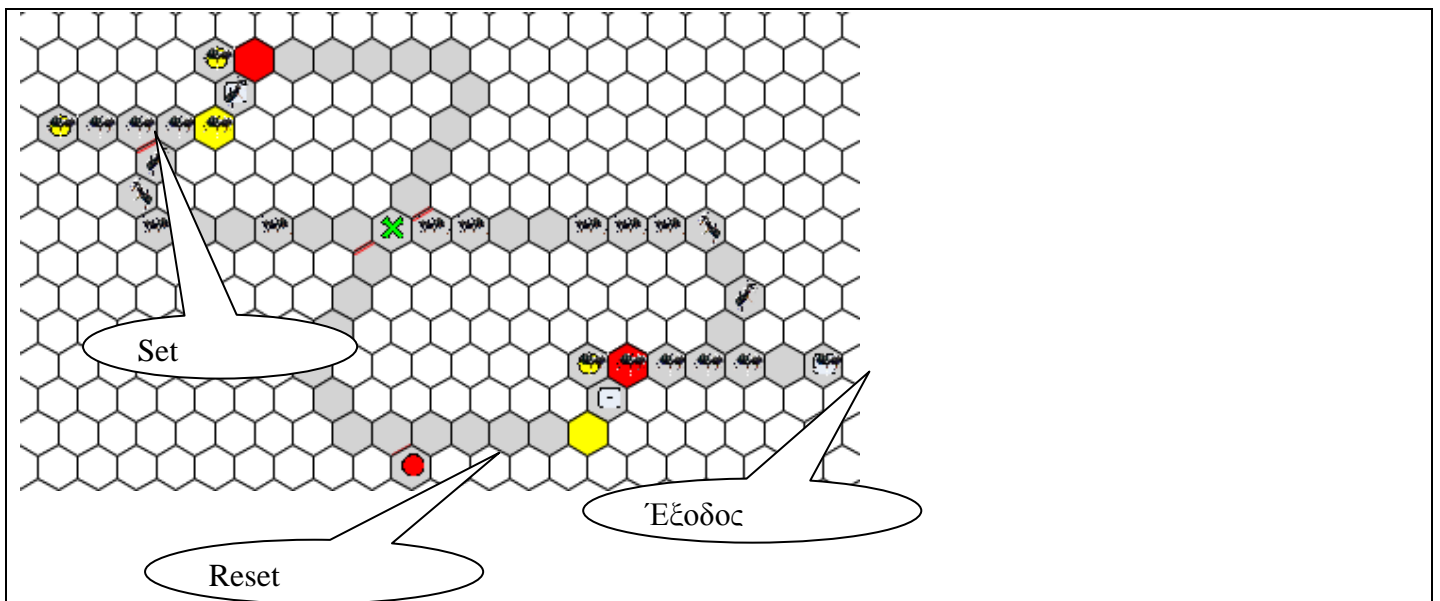




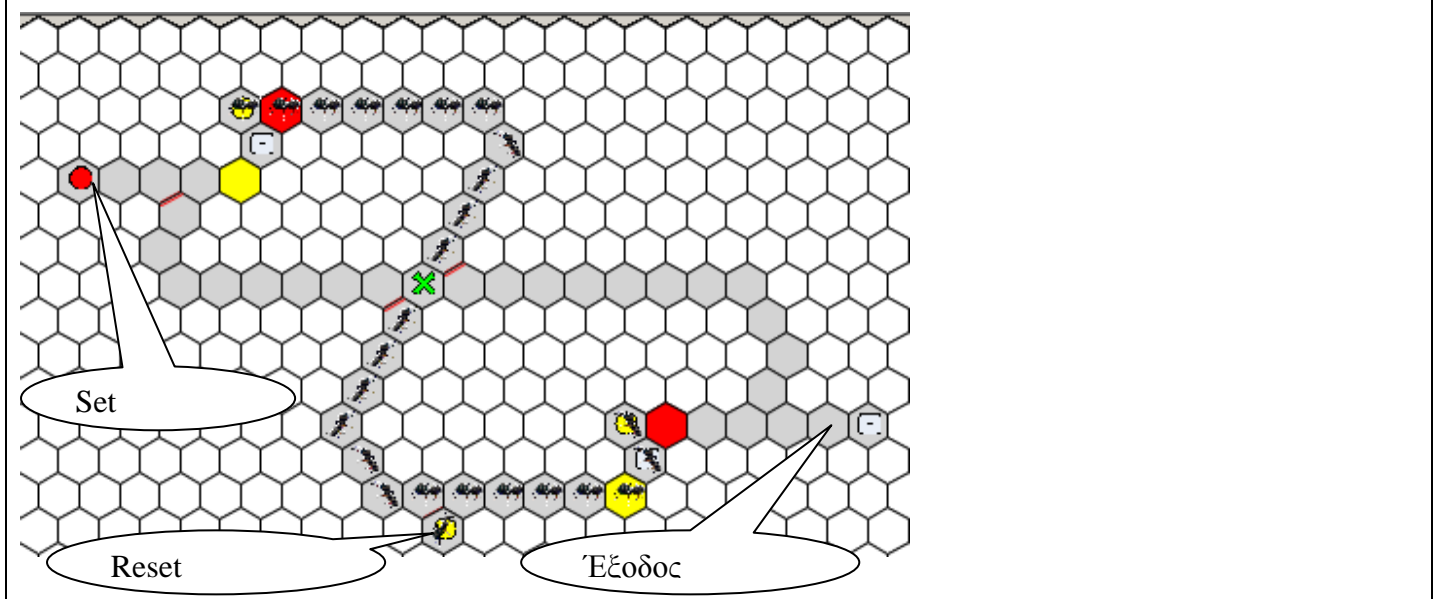
Για την δημιουργία μιας πύλης NAND αρκεί να τοποθετήσουμε ακόμα ένα αντιστροφέα στην έξοδο μια πύλης AND. Οι δύο αντιστροφείς που ακολουθούν αναιρεί ο ένας τον άλλο και τοποθετούνται για ενίσχυση του σήματος.



Ο Μανταλωτής SR (SR Latch) είναι ένα ακολουθιακό κύκλωμα (sequential circuit). Τα ακολουθιακά κυκλώματα αποθηκεύουν κατάσταση (state) σε μορφή δυαδικών ψηφίων (bits). Η έξοδος εξαρτάται από την είσοδο στο παρόν και την παρούσα κατάσταση. Η επόμενη κατάσταση εξαρτάται από την παρούσα κατάσταση και είσοδο. Η υλοποίηση ενός Μανταλωτή στο βιολογικό μας μοντέλο είναι πολύ σημαντική γιατί ένας μανταλωτής είναι ουσιαστικά ένα στοιχείο μνήμης το οποίο του δίνουμε τιμή στο set (S) και το μηδενίζουμε με το reset (R).



Σχήμα 2-20 SR LATCH flip flop – Set.



Σχήμα 2-21 SR LATCH flip flop – Reset.

Κεφάλαιο 3

Προσομοίωση

Προσομοίωση (simulation) είναι η μίμηση της λειτουργίας συστημάτων ή της εξέλιξης διαδικασιών μέσα στο χρόνο με τη βοήθεια υπολογιστή. Διαδικασία ή σύστημα ονομάζεται ένα σύνολο στοιχείων τα οποία εξελίσσονται και αλληλεπιδρούν σύμφωνα με κάποιους κανόνες. Οι κανόνες αυτοί εκφράζονται με μαθηματικές ή λογικές σχέσεις, και αποτελούν το μοντέλο του συστήματος. Κατάσταση είναι το σύνολο των μεταβλητών οι οποίες δίνουν την απαραίτητη πληροφορία για την περιγραφή του συστήματος. Σε αυτό το κεφάλαιο θα ασχοληθούμε με τις τεχνικές που χρησιμοποιούμε για την ανάπτυξη του προσομοιωτή για το βιολογικό μας μοντέλο και το περιβάλλον ανάπτυξης του

3.1 Περιβάλλον ανάπτυξης του προσομοιωτή

Στο υποκεφάλαιο αυτό περιγράφουμε την πλατφόρμα ανάπτυξης που επιλέχτηκε για την ανάπτυξη του προσομοιωτή μας, το τι υποστηρίζει, ποιες άλλες πλατφόρμες θα μπορούσαν να χρησιμοποιηθούν και γιατί επιλέχθηκε τελικά το Visual Studio.

Τι είναι το Visual Studio;

Το Microsoft Visual Studio είναι μια σουίτα που βοηθάει στην ανάπτυξη λογισμικού. Είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) το οποίο διασφαλίζει την ποιότητα του κώδικα σε ολόκληρο τον κύκλο ανάπτυξης λογισμικού, από την σχεδίαση μέχρι την παράδοση. Περιλαμβάνει επεξεργαστή πηγαίου κώδικα, μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή (debugger), συνδέτη (Linker), σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη.

Επιλογή γλώσσας ανάπτυξης

Το Microsoft Visual Studio υποστηρίζει πολλές γλώσσες ανάπτυξης. Μερικές από αυτές είναι η Visual Basic, η C++ , η Java και η C#. Για την δημιουργία του προσομοιωτή χρειαζόμασταν μια γλώσσα η οποία θα υποστήριζε αντικείμενα. Ένα αντικείμενο είναι ένα δομικό συστατικό λογισμικού. Κάθε αντικείμενο έχει

κατάσταση, συμπεριφορά και ταυτότητα. Η κατάσταση περιγράφει όλες τις στατικές ιδιότητες του αντικειμένου, όπως τιμές σε μεταβλητές μνήμης. Οι ιδιότητες αυτές είναι το αποτέλεσμα της συμπεριφοράς του αντικειμένου, δηλαδή του τρόπου με τον οποίο αυτό ανταποκρίνεται σε κλήσεις από το περιβάλλον του. Η ταυτότητα είναι η μοναδική διάκριση του αντικειμένου από τα ομοειδή του. Κατά την προγραμματιστική αναπαράσταση κάποιου προβλήματος του πραγματικού κόσμου ο αντικειμενοστραφής προγραμματισμός προσφέρεται για να συσχετίσουμε εύκολα, αντικείμενα του πραγματικού κόσμου με αντικείμενα στο πρόγραμμα μας. Η ευκολία στη χρήση, η υποστήριξη πληθώρας δομών δεδομένων (όπως λίστες, πίνακες που κάθε θέση με την σειρά της να μπορούσε να περιέχει πολύπλοκες δομές δεδομένων) αλλά και ο χειρισμός δισδιάστατων γραφικών ήταν τα υπόλοιπα χαρακτηριστικά που ζητούσαμε από μια γλώσσα προγραμματισμού.

Η Visual Basic υποστηρίζει όλες τις δομές που χρειαζόμαστε και παρέχει δυνατότητες εφάμιλλες με την C. Το ότι θα διαχειριζόμασταν γραφικά δεν μας ανησύχησε διότι αυτά θα δημιουργούνταν χρησιμοποιώντας τις βιβλιοθήκες γραφικών GDI+ που υποστηρίζονται γενικότερα από το Microsoft Visual Studio. Το GDI+ είναι μια ολοκληρωμένη και ευέλικτη βιβλιοθήκη γραφικών. Ο προσομοιωτής θα αναπαριστά δισδιάστατα γραφικά. Το GDI+ καλύπτει πλήρως την ανάγκη σχεδίασης δισδιάστατων γραφικών. Περισσότερα για την χρήση και δυνατότητες της GDI+ αναπτύσσονται στο Παράρτημα.

Άλλα Εργαλεία που εξετάστηκαν

SwinGame

Το SwinGame [3] είναι ένα SDK (Software Development Kit) φτιαγμένο ειδικά για τη δημιουργία δισδιάστατων παιχνιδιών. Βασίζεται σε βιβλιοθήκες που είναι φτιαγμένες σε Pascal. Μπορεί να χρησιμοποιηθεί μέσα από το περιβάλλον του Microsoft Visual Studio αν κάποιος το επιθυμεί. Η σχεδίαση ενός προσομοιωτή έχει πολλά κοινά στοιχεία με αυτά της σχεδίασης ενός παιχνιδιού. Μετά από δοκιμές που κάναμε που αφορούσαν δισδιάστατα γραφικά φάνηκε ότι τα χειρίζεται ιδιαίτερα καλά και ότι κρύβει μια πολύ καλή μηχανή γραφικών (graphics engine) στον πυρήνα του. Δεν βρέθηκαν όμως επαρκείς πηγές

καθώς και παραδείγματα χρήσης του σε πολύπλοκα σενάρια παράγοντας που ήταν αποτρεπτικός για τη χρήση του.

Blender

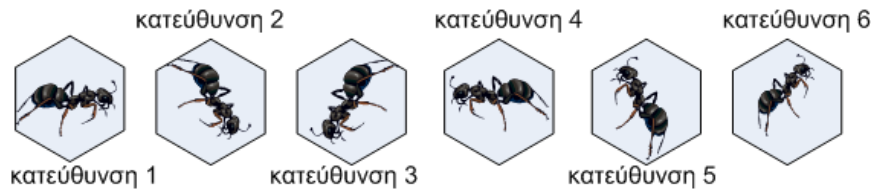
Το Blender [4] είναι μια πλατφόρμα σχεδίασης τρισδιάστατων γραφικών καθώς και simulation. Μπορεί κάποιος να σχεδιάσει ή να πάρει έτοιμα τρισδιάστατα μοντέλα, να τα τοποθετήσει στο χώρο, να τα φωτίσει με πηγές φωτός, να δώσει υφή, βάρος, να τα ορίσει να υπακούουν νόμους της φυσικής καθώς και χειρισμών συγκρούσεων. Ο προγραμματισμός του μπορεί να γίνει με διαδραστικό τρόπο ή μέσω της δικής του γλώσσας προγραμματισμού. Τα αποτελέσματα που μπορεί να έχει κάποιος χρησιμοποιώντας το είναι τουλάχιστον εντυπωσιακά [9]. Ανάμεσα στα σημεία που μας απέτρεψαν στην επιλογή του είναι η εξαιρετικά μεγάλη καμπύλη μάθησης (learning curve) που πρέπει να έχει κάποιος για να το χρησιμοποιήσει για το σκοπό που το προορίζαμε καθώς και η μεγάλη υπολογιστική ισχύς που χρειαζόταν για να τρέξει τέτοια εφαρμογή.

3.2 Δομές Δεδομένων

Στο κεφάλαιο 2 αναφερθήκαμε σε ένα μοντέλο κίνησης των μυρμηγκιών. Αναφερθήκαμε σε μυρμηγκία, τις θέσεις τους σε μια επιφάνεια εργασίας καθώς και στα διάφορα θεμελιώδη συστατικά. Σε αυτό το σημείο θα παρουσιάσουμε με ποιες δομές έχει γίνει η υλοποίηση όλων των στοιχείων που αφορούν τη προσομοίωση, δηλαδή πώς αυτά αναπαριστούνται στη μνήμη του ηλεκτρονικού υπολογιστή. Το πώς θα εμφανίζονται στην οθόνη θα μας απασχολήσει στο κεφάλαιο 4.

Η επιφάνεια σχεδιασμού αναπαρίσταται στη μνήμη ως ένας δισδιάστατος πίνακας. Έχουμε δύο τέτοιους πίνακες. Κάθε θέση αυτών των πινάκων περιέχει δεδομένα τύπου colony (βλέπε σχήμα 3-3). Ο ένας (prevmatrix) αναπαριστά την αμέσως προηγούμενη κατάσταση της επιφάνειας της οποίας μπορούν να κινούνται μυρμηγκία ενώ ο άλλος (matrix) την παρούσα κατάσταση. Οι διαστάσεις τους είναι μέρος των παραμέτρων του συστήματος και αναπαριστούν τον αριθμό των θέσεων που μπορούν να τοποθετηθούν θεμελιώδη συστατικά ή μυρμηγκία στην επιφάνεια σχεδίασης. Κάθε θέση του πίνακα μπορεί να

φιλοξενήσει ένα μυρμήγκι. Μια Boolean μεταβλητή (*ant*) καθορίζει αν ένα μυρμήγκι βρίσκεται σε συγκεκριμένη θέση. Το *direction* είναι μια μεταβλητή τύπου *coordinate* και έχει εύρος τιμών από 1-6 και καθορίζει μία από τις έξι κατευθύνσεις που μπορεί να έχει ένα μυρμήγκι.



Σχήμα 3-1 Οι πιθανές κατευθύνσεις ενός μυρμηγκιού.

Το **pheromon_amount** είναι μια μεταβλητή τύπου *double* (αριθμός κινητής υποδιαστολής) και καθορίζει την ποσότητα συγκέντρωσης φερομόνης σε μια συγκεκριμένη θέση της επιφάνειας εργασίας.

Τα *ant*, *direction*, *pheromon_amount* περιέχουν πληροφορίες που αλλάζουν ανά χρονική στιγμή.

Το **clear** είναι μια μεταβλητή τύπου *Boolean*. Αν η τιμή της είναι αληθής δηλώνει ότι η συγκεκριμένη θέση του πίνακα είναι άδεια και δεν περιέχει οτιδήποτε.

Το **path** δηλώνει τη δυνατότητα διέλευσης σε μια συγκεκριμένη θέση. Χαρακτηρίζει μια θέση προσπελάσιμη και απαιτείται να έχει τιμή *true* για τη παρουσία ή όχι ενός *path* στην εκάστοτε θέση του πίνακα.

Οι πιο κάτω μεταβλητές αφορούν στοιχεία που παραμένουν σταθερά και δεν επηρεάζονται από τη χρονική στιγμή.

Το **source** είναι μεταβλητή τύπου *Boolean*. Όταν παίρνει την τιμή *true* δηλώνει την ύπαρξη πηγής στη συγκεκριμένη θέση. Τροφοδοτεί το σύστημα με μυρμήγκια με ρυθμό που καθορίζεται από το *source_percentage*.

Το **source_percentage** είναι μια μεταβλητή ακεραίου. Ο λόγος $\text{source_percentage}/100$ καθορίζει την πιθανότητα εμφάνισης ενός μυρμηγκιού σε μια πηγή ανά μονάδα χρόνου.

Το **switch** είναι μεταβλητή τύπου Boolean. Το switch καθορίζει την κατάσταση λειτουργίας μια πηγής. Όταν πάρει την τιμή true διακόπτεται η ροή μυρμηγκιών στο σύστημα στη θέση στη οποία υπάρχει η αντίστοιχη πηγή. Η τιμή false απενεργοποιεί μια πηγή με αποτέλεσμα να μην παρουσιάζονται μυρμηγκία σε αυτή.

Το **sink** είναι μεταβλητή τύπου Boolean. Όταν παίρνει την τιμή true λειτουργεί ως έξοδος ενός μυρμηγκιού από το σύστημα από τη θέση που βρίσκεται το sink.

Το **brick** είναι μεταβλητή τύπου Boolean. Όταν παίρνει την τιμή true δηλώνει την παρουσία ενός brick στην εκάστοτε θέση του πίνακα. Το brick απαγορεύει την πρόσβαση μυρμηγκιού ή φερομόνης στην συγκεκριμένη θέση του πίνακα.

Το **pump** είναι μια μεταβλητή τύπου Boolean. Όταν παίρνει τιμή true δηλώνει την παρουσία αντλίας σε μια συγκεκριμένη θέση του πίνακα. Η αντλία τροφοδοτεί το σύστημα με καθορισμένη ποσότητα φερομόνης (η ποσότητα φερομόνης p που τροφοδοτεί ένας τύπος αντλίας το σύστημα καθορίζεται από τις παραμέτρους του προσομοιωτή) ανά χρονική στιγμή στην θέση που βρίσκεται.

Το **wall** μπορεί να περιέχει μία λίστα από coordinates. Αν δηλαδή σε μια θέση ενός πίνακα υπάρχει ένα τοίχος στη θέση **right** τότε θα υπάρχει μια εγγραφή με την τιμή right στη λίστα.

Το **note** είναι μια αλφαριθμητική μεταβλητή. Μπορεί να περιέχει μια περιγραφή για μια συγκεκριμένη θέση του πίνακα.

Αναλυτικές πληροφορίες για κάθε αντλία είναι αποθηκευμένες στον πίνακα pumps_matrix. Η κάθε θέση του πίνακα pumps_matrix παίρνει τιμές τύπου pump.

Οι θέσεις του πίνακα Pumps_Matrix αποτελούνται από δεδομένα τύπου pump.

Ο τύπος pump αποτελείται από τα

- GroupId – Κωδικός Group.
- Value - Τιμή, είναι η ποσότητα φερομόνης που θα αφήνει η κάθε αντλία του συγκεκριμένου Group ανά μονάδα χρόνου.
- Name – το όνομα του group στο οποίο ανήκει η αντλία.

Σχήμα 3-2 Η δομή pump.

3.3 Σχεδιαστικές αποφάσεις

Στο τμήμα αυτό θα παρουσιάσουμε σχεδιαστικές αποφάσεις που αφορούν το κομμάτι του simulation στη μνήμη. Θα διαπραγματευτούμε θέματα όπως το πώς αναπαριστάται η συμπεριφορά ενός πίνακα εξαγώνων χρησιμοποιώντας ένα δισδιάστατο πίνακα. Πως επιτυγχάνεται η κίνηση των μυρμηγκιών, τι προβλήματα προέκυψαν και με ποιο τρόπο έγινε η αντιμετώπιση τους.

Δημιουργία πίνακα χειρισμού εξαγώνων

Ορίσαμε ως επιφάνεια εργασίας (board) ένα επίπεδο στρωμένο με εξάγωνα τα οποία καλούμε κελιά ή cells. Κάθε κελί εκτός από μυρμήγκια μπορεί να φιλοξενήσει τα προαναφερθέντα θεμελιώδη συστατικά. Συμπεραίνουμε ότι για να αναπαραστήσουμε κάτι τέτοιο στη μνήμη του υπολογιστή χρειαζόμαστε μια δομή που να μπορεί να περιγράψει το μέγεθος της επιφάνειας, τη θέση ενός εξαγώνου σε αυτή καθώς και τις γειτονικές (adjacent) θέσεις της θέσης αυτής.

Ένας δισδιάστατος πίνακας (two dimensional array) ορίζεται από ένα σύνολο πεπερασμένου πλήθους στοιχείων του ίδιου τύπου, όπου κάθε στοιχείο προσδιορίζεται από ένα ζεύγος δεικτών.

Θα μπορούσε επομένως να χρησιμοποιηθεί για να περιγράψουμε θέσεις στο δισδιάστατο χώρο όπως στο πιο κάτω παράδειγμα.

Στο σχήμα 3-5 η θέση 2,2 έχει τις εξής γειτονικές διευθύνσεις:

(1,2 - 1,3 - 2,3 - 3,3 - 3,2 - 2,1)

ενώ στο σχήμα 3-6 η θέση 3,2 τις εξής γειτονικές διευθύνσεις:

(2,2 - 3,3 - 4,2 - 4,1 - 3,1 - 2,1)

να χρησιμοποιήσουμε τεχνικές ψευδό-παράλληλισμού για να αναπαράγουμε γεγονότα του πραγματικού κόσμου.

Στο συγκεκριμένο εγχείρημα έχουμε να αντιμετωπίσουμε τα εξής.

- Κάθε χρονική στιγμή όλα τα μυρμήγκια που έχουμε στην επιφάνεια εργασίας μας επιλέγουν να κινηθούν ταυτόχρονα προς μια θέση.
- Μέσα στην ίδια χρονική στιγμή οι αντλίες πρέπει να αφήσουν φερομόνη, η υπάρχουσα φερομόνη να διαχέεται και κάποιο ποσοστό της να εξατμίζεται στην ατμόσφαιρα.
- Οι πηγές να φέρνουν νέα μυρμήγκια στο σύστημα.
- Να λαμβάνονται υπόψη τυχόν εμπόδια που παρεμβάλλονται στην διαδρομή που ακολουθεί κάθε μυρμήγκι.

Η παρούσα χρονική στιγμή και η επόμενη χρονική στιγμή καθορίζεται μέσω ενός ρολογιού (timer), δηλαδή ένα ρολόι παράγει παλμούς κάθε στιγμή. Ο ρυθμός κάθε στιγμής είναι παράμετρος που καθορίζεται από το σύστημα και εκφράζεται σε milliseconds. Αυτό το ρολόι μπορεί να θεωρηθεί ως η καρδιά του προσομοιωτή.

Πως λοιπόν χειριζόμαστε τα μυρμήγκια σειριακά και αυτό να φαίνεται σαν να τα χειριζόμαστε παράλληλα; Θα μπορούσαμε να πούμε ότι παίρνουμε τυχαία ένα μυρμήγκι και το αναθέτουμε στην θέση που διεκδικεί. Αφού αναθέσουμε όλα τα μυρμήγκια τότε εμφανίζουμε τη νέα κατάσταση (επόμενη χρονική στιγμή). Τι γίνεται όμως αν η θέση που διεκδικείται είναι ήδη κατειλημμένη;

Κατά την ανάπτυξη του προσομοιωτή προβληματιστήκαμε έντονα με ποιο τρόπο θα κινούσαμε τα μυρμήγκια και με ποιο τρόπο θα αντιμετωπίζαμε τις συγκρούσεις. Στα πρώτα στάδια ανάπτυξης παίρναμε κάθε μυρμήγκι της προηγούμενης χρονικής στιγμής και το τοποθετούσαμε βάση της επιλογής του στη νέα κατάσταση. Η ιδέα φαινόταν να λειτουργεί για μυρμήγκια τα οποία δεν είχαν εμπόδια στη διαδρομή τους.

Όμως όταν βρίσκονταν εμπόδια μπροστά τους το ένα μυρμήγκι καταλάμβανε την θέση του άλλου και αυτό είχε ως αποτέλεσμα την εξαφάνιση των μυρμηγκιών από την επιφάνεια εργασίας μας.

Σε κατοπινό στάδιο αποφασίσαμε να βάλουμε ένα μηχανισμό στον οποίο θα θεωρούσε κάθε μυρμήγκι που δεν μπορεί να προχωρήσει ως εμπόδιο. Αυτό είχε ως αποτέλεσμα να λειτουργεί σωστά ο μηχανισμός χωρίς να εξαφανίζονται μυρμήγκια. Όλα φυσικά ανατράπηκαν όταν αποφασίσαμε να χρησιμοποιήσουμε το θεμελιώδες συστατικό συνένωση. Στο συγκεκριμένο συστατικό δύο μονοπάτια ενώνονται σε ένα με αποτέλεσμα τις αλληπάληλες συγκρούσεις μυρμηγκιών στην περίπτωση που είχαμε συνεχή ροή μυρμηγκιών και από τα δύο μονοπάτια.

Η λύση μας δεν αντιμετώπιζε προβλήματα συγκρούσεων και το πρόβλημα εξαφάνισης των μυρμηγκιών επανήλθε. Παρατηρήσαμε ότι σε κάποια σενάρια συγκεκριμένες θέσεις μπορούσαν να διεκδικούνται από πολλά μυρμήγκια ταυτόχρονα. Σε αυτό συνηδαιτοποιήσαμε ότι ένας μηχανισμός εντοπισμού συγκρούσεων ήταν αναγκαίος. Με ποιο μηχανισμό όμως θα μπορούσαμε να ανιχνεύσουμε ότι πρόκειται να ακολουθήσει σύγκρουση;

Η σύγκρουση ακολουθεί εάν μια θέση στην επιφάνεια εργασίας διεκδικείται την επόμενη χρονική στιγμή από περισσότερα του ενός μυρμήγκια. Τα μυρμήγκια της παρούσας κατάστασης έχουν επιλέξει που προτίθενται να κινηθούν στην επόμενη κατάσταση. Αυτό το σημείο είναι το κατάλληλότερο για να προσθέσουμε το μηχανισμό ανίχνευσης συγκρούσεων.

Κάθε πρόθεση μπαίνει σε μια θέση του πίνακα `Collisions_Matrix` όπως φαίνεται στο σχήμα 3-10. Με αυτό τον τρόπο εξασφαλίζουμε ένα ενδιάμεσο στάδιο το οποίο μπορούμε να χειριστούμε πριν τη ανάθεση της νέα κατάσταση.

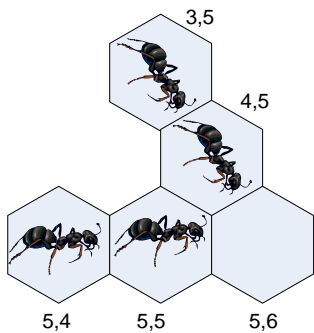
Ο συγκεκριμένος πίνακας χρησιμοποιείται ως μέσω ανίχνευσης συγκρούσεων. Στον εν λόγω πίνακα καταλήγουν όλα τα μυρμήγκια προτού ανατεθούν στην επόμενη κατάσταση. Με αυτό τον τρόπο μπορούμε να σαρώσουμε τον πίνακα προτού γίνει η ανάθεση και να βγάλουμε συμπεράσματα του τύπου.

- Υπάρχουν μυρμήγκια που διεκδικούν την παρούσα τους κατάσταση και την επόμενη χρονική στιγμή;

- Υπάρχουν θέσεις που διεκδικούνται από περισσότερα από ένα μυρμήγκια;

Είναι προφανές ότι στην πρώτη περίπτωση το μυρμήγκι που διεκδικεί την παρούσα του θέση την επόμενη χρονική στιγμή προηγείται να εξυπηρετηθεί σε σχέση με ένα άλλο που διεκδικεί εκείνη τη θέση.

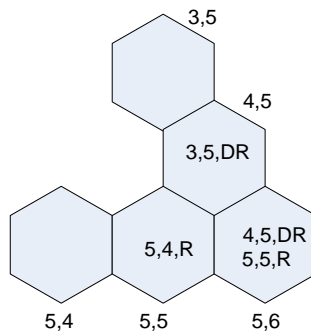
Επίσης αν δεν υπάρχει μυρμήγκι που να διεκδικεί την παρούσα του θέση και στην επόμενη κατάσταση, αλλά μια θέση διεκδικείται από πολλά μυρμήγκια τότε πρέπει να υπάρχει ένας μηχανισμός διαιτησίας ο οποίος να χειρίζεται με ποιο τρόπο θα αποφεύγονται οι συγκρούσεις.



Σχήμα 3-9

Παράδειγμα σύγκρουσης.

Στο συγκεκριμένο παράδειγμα η θέση (5,6) διεκδικείται ταυτόχρονα από δύο μυρμήγκια, αυτό που βρίσκεται στην θέση (4,5) και αυτό που βρίσκεται στην θέση (5,5).



Σχήμα 3-10

Παράδειγμα στον πίνακα

Collisions_Matrix.

Στο συγκεκριμένο παράδειγμα φαίνεται πως αναπαριστάται μια διεκδίκηση στον πίνακα Collisions_Matrix.

Αποφυγή συγκρούσεων

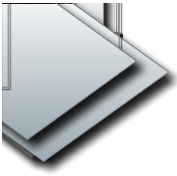
Αρχικά το χειρισμό των συγκρούσεων το χειριστήκαμε με τον εξής τρόπο. Παίρναμε τα μυρμήγκια που διεκδικούσαν μια συγκεκριμένη θέση επιλέγαμε τυχαία ένα από αυτά, το τοποθετούσαμε και τα υπόλοιπα τα βάζαμε πίσω στις αρχικές τους θέσεις. Αυτό το σενάριο αν και δούλευε εντούτοις το να επιστρέψουν τα μυρμήγκια στις αρχικές τους θέσεις αποδείχθηκε ότι οδηγούσε σε μεγάλες αναδρομικές διαδικασίες και σε μια λογική που ήταν εκτός της προσέγγισης ενός προσομοιωτή. Είναι πιο φυσικό τα μυρμήγκια να

παραμένουν ακίνητα παρά να κάνουν την επιλογή τους και μετά να την αναιρούμε. Έτσι καταλήξαμε στην προσέγγιση που αναπτύσσουμε πιο κάτω.

Ο μηχανισμός χειρισμού συγκρούσεων ενεργοποιείται πάντα πριν την τοποθέτηση των μυρμηγκιών στην νέα κατάσταση. Αν η θέση στην οποία πρόκειται να κινηθεί το μυρμήγκι δεν είναι γέφυρα τότε επιλέγεται τυχαία ένα από τα μυρμήγκια που διεκδικούν την εκάστοτε θέση προς αυτή ενώ τα υπόλοιπα (αν δηλαδή υπήρξε σύγκρουση) μένουν στάσιμα στην προηγούμενη τους θέση. Η τυχαιότητα επιβάλλεται για να αποφύγουμε την περίπτωση starvation. Αν τα μυρμήγκια εξάγονται πάντα με ίδια σειρά από τον πίνακα συγκρούσεων θα έχουμε ως αποτέλεσμα την εξυπηρέτηση μόνο ενός εκ των δύο μονοπατιών στην περίπτωση της συνένωσης με αποτέλεσμα ένα μονοπάτι να μην εξυπηρετείται ποτέ. Ακολουθεί διάγραμμα ροής του μηχανισμού χειρισμού συγκρούσεων.

Ο αλγόριθμος χειρισμού συγκρούσεων λειτουργεί με τον εξής τρόπο. Θεωρούμε ότι στην νέα κατάσταση τοποθετούνται μόνο τα μυρμήγκια τα οποία θεωρούνται ακίνητα. Βγάζουμε μυρμήγκια από τον πίνακα διεκδικήσεων, αν η θέση δεν είναι κατειλημμένη στην νέα κατάσταση τότε αυτά τοποθετούνται στην νέα κατάσταση διαφορετικά αυτά τα μυρμήγκια τοποθετούνται ξανά στον πίνακα χειρισμού συγκρούσεων και διεκδικούν την θέση που ήταν στην παρούσα κατάσταση, δηλαδή στην επόμενη επανάληψη ο αλγόριθμος θα τα θεωρήσει ακίνητα και θα τα τοποθετήσει στην αρχική τους θέση. Ο αλγόριθμος τερματίζει όταν δεν υπάρχουν πλέον μυρμήγκια στον πίνακα συγκρούσεων.

Ο αλγόριθμος εξασφαλίζει προτεραιότητα στην τοποθέτηση μυρμηγκιών που διεκδικούν την παρούσα τους θέση, είναι δηλαδή ακίνητα. Ο έλεγχος πριν την τοποθέτηση ενός εξαγόμενου μυρμηγκιού εξασφαλίζει ότι δεν θα τοποθετηθεί μυρμήγκι σε κατειλημμένη θέση.

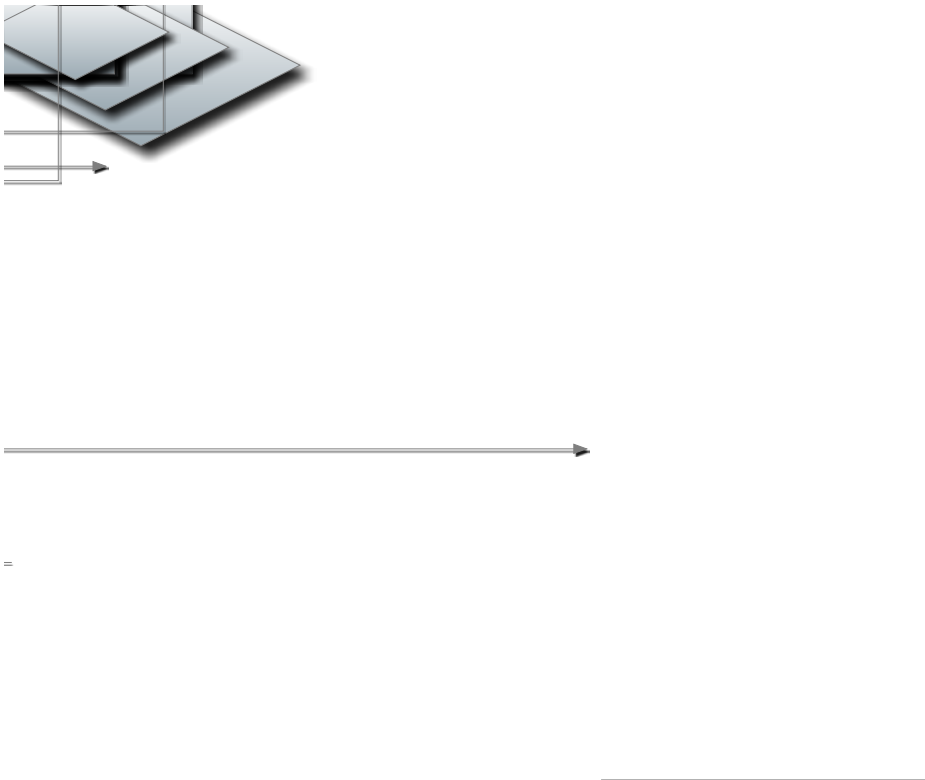


Σχήμα 3-11
Διάγραμμα ροής αλγόριθμου αποφυγής συγκρούσεων.

Αλγόριθμος κίνησης μυρμηγκιών

Η βάση του αλγόριθμου κίνησης των μυρμηγκιών είναι η ίδια με αυτού που παρουσιάζουμε στο κεφάλαιο 2. Κάποια σημεία που έχρηζαν διευκρίνησης ή επέκτασης έχουν γίνει. Σε αυτό το σημείο θα περιγράψουμε πως υλοποιούνται προγραμματιστικά τα κύρια και σημαντικά σημεία του αλγορίθμου, τυχόν παραδοχές ή προσθήκες που χρειαστήκαν να εισάγουμε.

Ακολουθεί το διάγραμμα ροής του τροποποιημένου αλγόριθμου κίνησης σε σχέση με τον αντίστοιχο του δοσμένου μοντέλου όπως αυτός τροποποιείται για την αναπαράσταση του με ηλεκτρονικό υπολογιστή.



**Σχήμα 3-12 Διάγραμμα ροής προσομοίωσης.
Πως επιλέγει ένα μυρμήγκι σε ποια θέση προτίθεται να κινηθεί.**

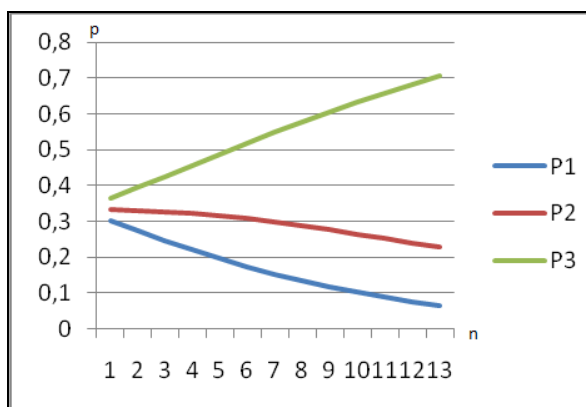
Ακολουθεί επεξήγηση για την υλοποίηση των σημαντικών τμημάτων του αλγόριθμου:

Έχουμε μια αρχική κατάσταση που αναπαριστάται με ένα πίνακα με τύπους δεδομένων Colony. Όποιες θέσεις έχουν την τιμή `ant=true` αυτές θεωρούνται θέσεις όπου υπάρχουν τοποθετημένα μυρμήγκια και το `direction` δείχνει την κατεύθυνση τους. Σαρώνοντας αυτό τον πίνακα δημιουργούμε μια λίστα όπου κάθε θέση της περιέχει πληροφορίες για τη θέση και τη κατεύθυνση κάθε μυρμηγκιού. Καλούμε αυτή τη λίστα, λίστα μυρμηγκιών.

Οι προσπελάσιμες θέσεις είναι αυτές που εφάπτονται των πλευρών ενός σημείου στην επιφάνεια σχεδίασης.

Ο τρόπος εύρεσης των γειτονικών θέσεων αναπτύσσεται στην αρχή του κεφαλαίου.

Η επιλογή μιας από τις γειτονικές προσπελάσιμες θέσεις γίνεται με τη χρήση του αλγόριθμου ChooseDirection. Ο αλγόριθμος ChooseDirection είναι υπεύθυνος για την επιλογή της θέσης που θα κινηθεί ένα μυρμήγκι. Η λειτουργία του είναι η εξής: Το μυρμήγκι πρέπει να διαλέξει μια από τις γειτονικές προσπελάσιμες από αυτό θέσεις. Ο μέγιστος αριθμός γειτονικών προσπελάσιμων θέσεων ενός μυρμηγκιού είναι το πολύ τρεις επειδή θεωρήθηκε πιο κοντινό στο βιολογικό μοντέλο όπου ένα μυρμήγκι δεν μπορεί να κάνει απότομες στροφές. Για να αποφασίσουμε σε ποια θέση πρέπει να κινηθεί ένα μυρμήγκι εφαρμόζουμε μια κατανομή πιθανοτήτων. Το n είναι μια σταθερά μη γραμμικότητας στη πιθανοτική συμπεριφορά των μυρμηγκιών. Με άλλα λόγια όσο πιο μεγάλο γίνεται το n τόσο πιο ευαίσθητα γίνονται τα μυρμήγκια στις διακυμάνσεις της φερομόνης και τόσο πιο σίγουρο είναι ότι το μυρμήγκι θα κινηθεί προς υψηλές συσσωρεύσεις φερομόνης.



Σχήμα 3-13

Γραφική παράσταση της σταθεράς μη γραμμικότητας συναρτήσει της πιθανότητας επιλογής. Η θέση 1 έχει την μικρότερη συγκέντρωση φερομόνης ενώ η θέση 3 την μεγαλύτερη. Τα 3 σημεία που αφορούν τα p_1 , p_2 , p_3 έχουν μικρή διαφορά συγκέντρωσης φερομόνης μεταξύ τους. Βλέπουμε ότι για υψηλές τιμές του n η πιθανότητα επιλογής του σημείου 3 είναι κατά πολύ μεγαλύτερη σε σχέση με τα υπόλοιπα σημεία.

```

ChooseDirection(list availLocations)
    a = 0
    b = 0
    result = null
    PLS = null

    Επανάλαβε για κάθε z που ανήκει στην λίστα availLocations
        a = a + z.pheromon_amount ^ n
        PLS.Add(z.pheromon_amount ^ n)
    Τέλος Επανάληψης

    Αν a = 0 Then
        result = availLocations(RandomNumber(0, availLocations.Count - 1))
    αλλιώς
        b = a * RandomNumberDouble()

        Αν (b >= 0 And b <= tPLs(0)) τότε
            result = availLocations(0)
        αλλιώς αν b >= tPLs(0) And b <= tPLs(0) + tPLs(1) τότε
            result = availLocations(1)
        αλλιώς αν b > tPLs(0) + tPLs(1) And b <= tPLs(0) + tPLs(1) + tPLs(2) τότε
            result = availLocations(2)
        Τέλος Αν

    Τέλος Αν

    Return result

```

Σχήμα 3-14 Αλγόριθμος ChooseDirection.

Ο αλγόριθμος είναι σχετικά απλός. Ο αλγόριθμος αθροίζει τις συγκεντρώσεις των φερομόνων σε όλα τα προσπελάσιμα σημεία. Επιλέγει τυχαία ένα δεκαδικό αριθμό και τον πολλαπλασιάζει με το άθροισμα.

Ανάλογα με το σε ποιο διάστημα τιμών ανήκει αυτός ο αριθμός επιλέγεται το συγκεκριμένο σημείο.

Υπολογισμός διάδοσης φερομόνης

Όπως είδαμε στο κεφάλαιο 2 η φερομόνη διαδίδεται στο σύστημα βάσει της πιο κάτω εξίσωσης.

$$P^t(L) = (1 - d) \cdot [(1 - f) \cdot P^{t-1}(L) + f \cdot W^{t-1}(L)] + s^{t-1}(L) + p(L)$$

Ακολουθεί ο αλγόριθμος που χρησιμοποιείται από τον προσομοιωτή για τον υπολογισμό της συγκέντρωσης της φερομόνης σε κάθε θέση μιας επιφάνεια εργασίας (board) την χρονική στιγμή t . Το $t-1$ εκφράζει την προηγούμενη χρονική στιγμή.

```

Επανάλαβε εφόσον η λίστα θέσεων φερομόνων δεν είναι κενή
  Φέρε θέση (i,j) από λίστα φερομόνων
  Βρες τις γειτονικές θέσεις της θέσης (i,j)
  Αθροισε στο  $W^{t-1}$  την ποσότητα φερομόνης που είχαν όλα τα γειτονικά σημεία στην
  προηγούμενη χρονική στιγμή
  Θέσε στο  $P^{t-1}$  την ποσότητα φερομόνης που είχε η θέση (i,j) την προηγούμενη χρονική
  στιγμή
  Αν υπάρχει αντιλία στη θέση (i,j) τότε
     $P1$  = ποσότητα φερομόνης που τροφοδοτείται στο σύστημα από αντιλία
  Αλλιώς
     $P1=0$ 
  Τέλος Αν
  Αν υπήρχε μυρμήγκι στη θέση (i,j) την προηγούμενη χρονική στιγμή τότε
    Αν αυτό αποφάσισε ότι πρέπει να εκκρίνει φερομόνη τότε
       $S^{t-1}$  = ποσότητα φερομόνης που εκκρίνει το μυρμήγκι
    Τέλος Αν
  Αλλιώς
     $S^{t-1} = 0$ 
  Τέλος Αν
   $Pt(i,j) = (1-d) * ((1-f) * P^{t-1} + f * W^{t-1}) + S^{t-1} + P1$ 
Τέλος επανάληψης

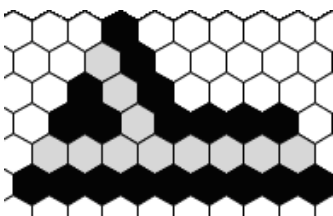
```

Σχήμα 3-15 Υπολογισμός διάδοσης φερομόνης.

3.5 Βελτιώσεις

Λειτουργία του θεμελιώδους συστατικού συνένωση

Σύμφωνα με το μοντέλο κίνησης των μυρμηγκιών η συνένωση αναπαριστάται όπως πιο κάτω.

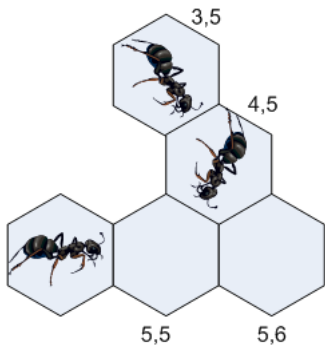


Σχήμα 3-16 Συστατικό συνένωση.

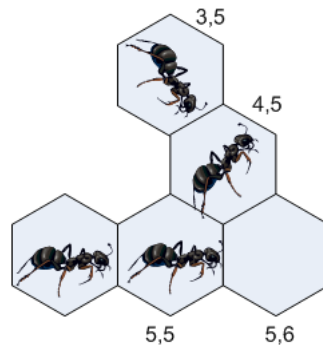
Κατά το τρέξιμο του προσομοιωτή διαπιστώσαμε ότι κάποια μυρμήγκια επέλεξαν να κατευθυνθούν σε άλλα σημεία παρά στο αναμενόμενο σημείο που ήταν η έξοδος.

Συγκεκριμένα στα πιο κάτω παραδείγματα τα μυρμήγκια θα έπρεπε να κατευθύνονται όλα προς την έξοδο που είναι η θέση (5,6). Αντί αυτού όμως στη μια περίπτωση (σχήμα 3-17) βλέπουμε το μυρμήγκι της θέσης (4,5) να κατευθύνεται προς την θέση (5,5) ενώ στην άλλη (σχήμα 3-18) το μυρμήγκι που βρισκόταν στην θέση (5,5) κατευθύνεται προς την θέση (4,5) αντί της θέσης (5,6) που αναμενόταν.

Αυτή η συμπεριφορά των μυρμηγκιών μας υποδεικνύει ότι μυρμήγκια που έχουν βρεθεί σε θέσεις όπως είναι η (4,5) και η (5,5) θα πρέπει να έχουν την επιλογή μόνο μιας κατεύθυνσης τέτοια ώστε να μπορέσουν να προχωρήσουν προς την έξοδο της συνένωσης. Αυτό μας έκανε να εισάγουμε ένα νέο στοιχείο που προστέθηκε στο σύνολο των θεμελιωδών συστατικών, τον τοίχο.



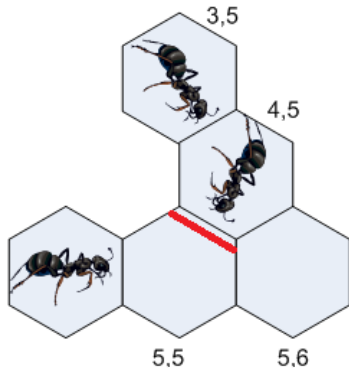
Σχήμα 3-17



Σχήμα 3-18

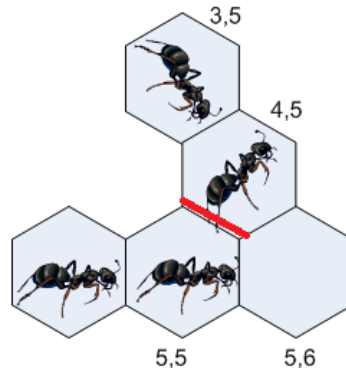
Το wall ή τοίχος είναι ουσιαστικά ένα εμπόδιο το οποίο παρεμβάλλεται μεταξύ 2 θέσεων μυρμηγκιών.

Ανάλογα με το πώς τοποθετείται υπάρχουν 2 είδη τειχών τα εξωτερικά και τα εσωτερικά. Ανάλογα σε ποια από τις πλευρές του εξαγώνου τοποθετούνται χαρακτηρίζονται όπως και οι κατευθύνσεις των μυρμηγκιών (1-6).



Σχήμα 3-18

Το σχήμα παρουσιάζει ένα εσωτερικό τοίχο στη θέση (5,5) (ή ένα εξωτερικό τοίχο στη θέση (4,5)) τοποθετημένο στην κατεύθυνση 1 της θέσης (5,5).

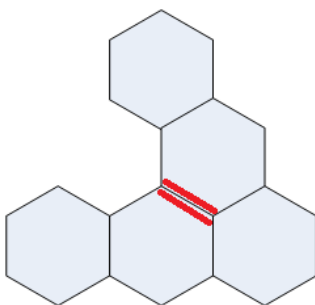


Σχήμα 3-19

Το σχήμα παρουσιάζει ένα εσωτερικό τοίχο στη θέση (4,5) (ή ένα εξωτερικό τοίχο στη θέση (5,5)) τοποθετημένο στην κατεύθυνση 4 της θέσης (4,5).

Στο σχήμα 3-18 ο τοίχος στην θέση (5,5) απαγορεύει την είσοδο του μυρμηγκιού που βρίσκεται στην θέση (4,5) ενώ στο σχήμα 3-19 ο τοίχος στη θέση (4,5) απαγορεύει την είσοδο του μυρμηγκιού που βρίσκεται στη θέση (5,5). Γενικά η συμπεριφορά του τοίχου είναι η εξής, κάθε εσωτερικός τοίχος που βρίσκεται σε μια από τις έξι πλευρές του εξαγώνου απαγορεύει την είσοδο σε μυρμηγκία που κατευθύνονται προς αυτή.

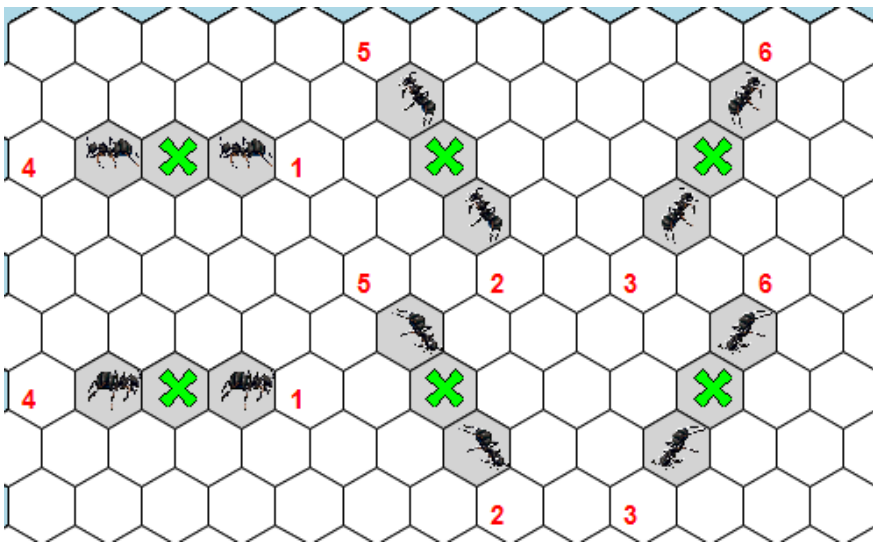
Επεκτείνοντας το μοντέλο κίνησης των μυρμηγκιών καταλήγουμε ότι η υλοποίηση θεμελιώδους συστατικού συνένωσης χρήζει χρήσης δύο τειχών που τοποθετούνται στο σημείο διασταύρωσης των μυρμηγκιών όπως και στο σχήμα.



Σχήμα 3-20 Τελική μορφή συστατικού Σύζευξης.

Γέφυρα

Όπως έχουμε αναφερθεί και προηγουμένως το στοιχείο γέφυρα φιλοξενεί μυρμήγκια τα οποία εισάχθηκαν από διαφορετικές κατευθύνσεις προς αυτό. Το μυρμήγκι που επιχειρεί είσοδο σε ένα κελί που περιέχει γέφυρα θα βγει από αυτό με την κατεύθυνση που μπήκε όπως στο πιο κάτω παράδειγμα.



Σχήμα 3-21 Πιθανοί είσοδοι και έξοδοι ενός μυρμηγκιού σε μια γέφυρα.

Η υλοποίηση της γέφυρας γίνεται με τον εξής τρόπο: Εάν μια θέση περιέχει γέφυρα (δηλαδή η τιμή cross είναι αληθής) τότε όσα μυρμήγκια διεκδικούν την συγκεκριμένη θέση καταγράφονται σαν διεκδικητές της στον πίνακα `collisions_matrix`. Ο μηχανισμός χειρισμού συγκρούσεων δεν ενεργοποιείται για αυτές τις θέσεις. Τα μυρμήγκια εξάγονται από τον πίνακα `collisions` στην επόμενη χρονική στιγμή (σαν να τοποθετήθηκαν δηλαδή όλα στην θέση που ήταν η γέφυρα) και λαμβάνουν μέρος στην όλη διαδικασία επιλογής επόμενης θέσης.

Κεφάλαιο 4

Γραφικό Περιβάλλον

Γραφικό περιβάλλον χρήστη είναι το σύνολο γραφικών στοιχείων, τα οποία εμφανίζονται στην οθόνη κάποιας ψηφιακής συσκευής και χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με τη συσκευή αυτή. Παρέχουν στον τελευταίο, μέσω γραφικών, ενδείξεις και εργαλεία προκειμένου αυτός να φέρει εις πέρας κάποιες επιθυμητές λειτουργίες. Στο κεφάλαιο αυτό παρουσιάζουμε το γραφικό περιβάλλον του προσομοιωτή μας και γίνεται επεξήγηση των βασικών λειτουργιών του.

4.1 Περιβάλλον σχεδίασης

Το περιβάλλον σχεδίασης δίνει την δυνατότητα στο χρήστη να αλληλεπιδρά με ότι περιγράψαμε μέχρι τώρα ως συστατικά ενός ant based computer. Ο χρήστης μπορεί να αλληλεπιδρά με την επιφάνεια εργασίας, να τοποθετεί θεμελιώδη συστατικά σε αυτή και να «τρέχει» τον προσομοιωτή.

Εκτός αυτών ο χρήστης μπορεί να σώσει την εργασία του σε αρχείο με σκοπό να την ανακτήσει ή να την τροποποιήσει στο μέλλον. Επίσης παρέχεται η δυνατότητα ομαδοποίησης των θεμελιωδών συστατικών και η φύλαξη τους με συγκεκριμένο όνομα για την επαναχρησιμοποίηση τους σε μελλοντικές σχεδιάσεις.

Για παράδειγμα ένας χρήστης μπορεί να αφιερώσει αρκετό χρόνο για την σχεδίαση ενός πλήρους αθροιστή. Αντί η εργασία να σωθεί απλά σαν ένα αρχείο, μπορεί να σωθεί ως συστατικό (component) και να είναι διαθέσιμο στην λίστα των components για την μελλοντική του χρήση. Για παράδειγμα θα μπορούσε ο χρήστης να σώσει το συστατικό ημιαθροιστής με σκοπό την επαναχρησιμοποίηση του σε ένα μεγαλύτερο κύκλωμα όπως είναι ένας αθροιστής 2 bit.

4.2 Δομή αρχείων

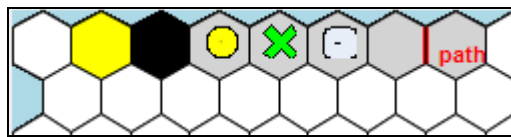
Τα αρχεία που δημιουργούνται για τη φύλαξη της επιφάνειας εργασίας έχουν την ίδια δομή με αυτά που χρησιμοποιούνται για τη δημιουργία components. Τα αρχεία φύλαξης έχουν επέκταση spr (Simulator Project) ενώ τα αρχεία για την δημιουργία components έχουν επέκταση scp (Simulator Component).

Η δομή του αρχείου είναι τέτοια ώστε να επιτρέπει την αναπαραγωγή ολόκληρης της επιφάνειας εργασίας.

Ειδικότερα ακολουθεί την οργάνωση της δομής Structure Colony.

Στο πιο κάτω σχήμα εμφανίζεται ένα μικρό κομμάτι της επιφάνειας σχεδίασης του προσομοιωτή.

Συγκεκριμένα τοποθετήθηκαν όλα τα θεμελιώδη συστατικά που έχουν έκταση ένα σημείο, στον πίνακα εργασίας ξεκινώντας από την θέση 1,2.



Σχήμα 4-1

Θεμελιώδη συστατικά τοποθετημένα σε κομμάτι της επιφάνειας σχεδιασμού

Η δομή του αρχείου που αναπαριστά το πιο πάνω φαίνεται στον πιο κάτω πίνακα Ακολουθήθηκε η δομή του structure colony αναπαριστώντας την σε αρχείο προσθέτοντας την εγγραφή κεφαλίδας .

sample.spr	1	1	199	199												
1	1	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1
1	2	0	0	1	1	0	0	0	0	0	-1	-1	-1	-1	-1	-1
1	3	0	0	0	0	1	0	0	0	0	-1	-1	-1	-1	-1	-1
1	4	0	100	1	0	0	1	0	0	0	-1	-1	-1	-1	-1	-1
1	5	0	0	1	0	0	0	0	1	0	-1	-1	-1	-1	-1	-1
1	6	0	0	1	0	0	0	1	0	0	-1	-1	-1	-1	-1	-1
1	7	0	0	1	0	0	0	0	0	0	1	-1	-1	-1	-1	-1
1	8	0	0	1	0	0	0	0	0	path	4	-1	-1	-1	-1	-1
1	9	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1
1	10	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1
1	11	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1
ΓΡΑΜΜΗ	ΣΤΗΛΗ Group Id	Path	Pump	Brick	Sourc	Sink	Cross	Note	wall1	wall2	wall3	wall4	wall5	wall6		
	Source	Timer	%													

Σχήμα 4-2

Η πρώτη γραμμή του αρχείου είναι η εγγραφή κεφαλίδας και περιέχει τις εξής πληροφορίες:

- Όνομα αρχείου
- Μετατόπιση (i,j). Η μετατόπιση χρησιμοποιείται στην περίπτωση που η δομή χρησιμοποιείται για αναπαράσταση spr αρχείου. Δηλώνει από ποια θέση ξεκινά να περιγράφει τον πίνακα εργασίας το εν λόγω αρχείο. Στην περίπτωση spr αρχείου η μετατόπιση έχει τιμή 1,1.
- Άνω όριο πίνακα εργασίας. Δηλώνει τις συντεταγμένες της τελευταίας θέσης στον πίνακα εργασίας.

Στην περίπτωση spr αρχείου το συγκεκριμένο πεδίο έχει τις διαστάσεις του πίνακα εργασίας.

Οι υπόλοιπες γραμμές του αρχείου περιγράφουν και μια θέση του πίνακα εργασίας.

Τα πεδία που χαρακτηρίζουν την κάθε γραμμή είναι τα εξής

Γραμμή	Στήλη	Group Id	Source Timer %	Path	Pump
Brick	Source	Sink	Cross	Note	
wall1	wall2	wall3	wall4	wall5	wall6

Σχήμα 4-3

Όλα τα πεδία παίρνουν τιμές από τα πεδία τιμών που παίρνουν οι αντίστοιχες μεταβλητές στο Structure Colony.

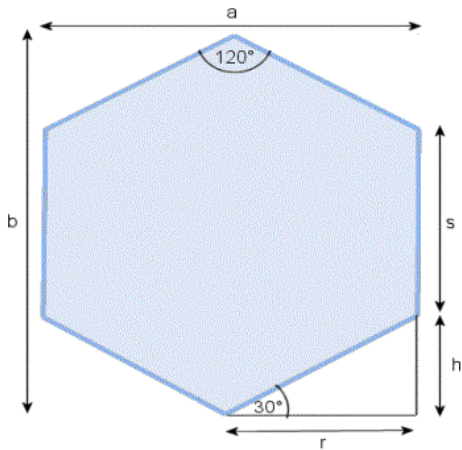
4.3 Σχεδιαστικές αποφάσεις

Στο τμήμα αυτό θα παρουσιάσουμε σχεδιαστικές αποφάσεις που αφορούν το κομμάτι του interface. Θα διαπραγματευτούμε θέματα όπως το στρώσιμο μιας επιφάνειας με εξάγωνα, μετατροπή των pixel συντεταγμένων σε συντεταγμένες εξαγώνων, καθώς και η αναπαράσταση μιας μεγάλης επιφάνειας που είναι κατά πολύ μεγαλύτερη από το μέγεθος μιας οθόνης ηλεκτρονικού υπολογιστή.

Στρώσιμο της επιφάνεια εργασίας με εξάγωνα.

Σε ένα εξάγωνο όλες οι πλευρές έχουν το ίδιο μήκος s οι οποίες συναντώνται σε γωνία 120 μοιρών.

Όλες οι υπόλοιπες τιμές μπορούν εύκολα να υπολογιστούν αφού βασίζονται στο s .



Σχήμα 4-4 Η γεωμετρία ενός εξαγώνου.

Από το πιο πάνω σχήμα προκύπτουν ότι:

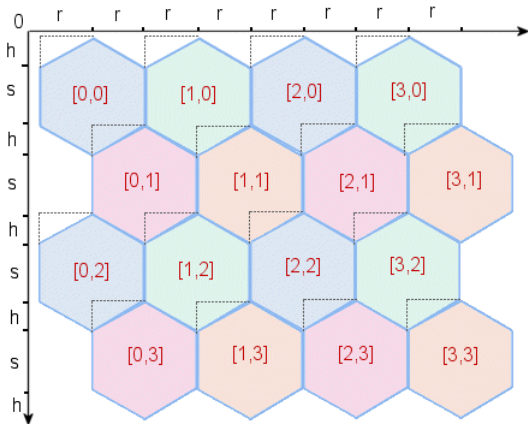
$$h = \sin(30^\circ) * s$$

$$r = \cos(30^\circ) * s$$

$$b = s + 2 * h$$

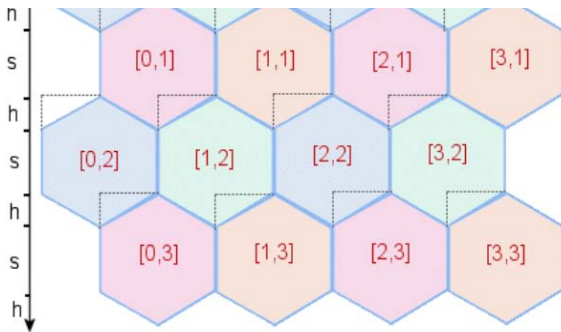
$$a = 2 * r$$

Παρατηρώντας το σχήμα 4-5 είναι εύκολο να συμπεράνουμε ότι για να πλακοστρώσουμε ένα επίπεδο με εξάγωνα αρκεί να τοποθετούμε ένα εξάγωνο και να μετακινούμαστε κατά $2r$ στο άξονα x για να δημιουργήσουμε μια γραμμή από εξάγωνα. Μετά κινούμαστε στον άξονα y κατά $h+s$ και δημιουργούμε ακόμα μια γραμμή και ούτω κάθε εξής.



Σχήμα 4-5 Επιφάνεια στρωμένη με εξάγωνα.

Φέροντας νοητές γραμμές ανά $2r$ στον κατακόρυφο και ανά $h+s$ στον οριζόντιο άξονα σε μια επιφάνεια στρωμένη με εξάγωνα δημιουργούνται τετράγωνα που ανάλογα το που σχηματίζονται τα χαρακτηρίζουμε ως τύπου «Α» και ως τύπου «Β» (σχήμα 4-6).



Σχήμα 4-6 Τετράγωνα τύπου Α και τετράγωνα τύπου Β.

Τα τετράγωνα που σχηματίζονται στις περιττές γραμμές εξαγώνων τα ονομάζουμε τύπου «Α» και τα τετράγωνα που σχηματίζονται στις άρτιες γραμμές εξαγώνων τύπου “Β” (η πρώτη γραμμή εξαγώνων έχει τιμή μηδέν).

Μετατροπή των συντεταγμένων εικονοστοιχείων σε συντεταγμένες τετραγώνων

Κάθε τετράγωνο έχει μήκος $2*r$ και πλάτος $h +s$. Διαιρώντας τις pixel συντεταγμένες στον άξονα x με το $2*r$ βρίσκουμε τις συντεταγμένες στον οριζόντιο άξονα και διαιρώντας τις συντεταγμένες y με $h+s$ βρίσκουμε τις συντεταγμένες στον κατακόρυφο άξονα.

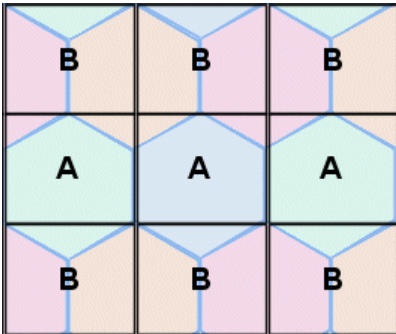
Επομένως αν θεωρήσουμε X, Y οι συντεταγμένες σε pixels στην επιφάνεια εργασίας τότε οι συντεταγμένες των τετραγώνων ισούνται με

$$\text{SectX} = X / (2 * r)$$

$$\text{SectY} = Y / (h + s)$$

Μετατροπή των συντεταγμένων εικονοστοιχείων συντεταγμένες εξαγώνων

Αφού εντοπίσουμε σε τι τύπο τετραγώνου είμαστε μένει να βρούμε σε ποιο σημείο του είμαστε για να αποφασίσουμε τις συντεταγμένες του εξαγώνου.



Σχήμα 4-7 Τμήματα τετραγώνων τύπου A και τύπου B.

Τετράγωνα τύπου «A»

Τα τετράγωνα τύπου «A» αποτελούνται από τρεις περιοχές:

- Εάν το pixel ανήκει στην μεγάλη περιοχή, τότε οι συντεταγμένες του εξαγώνου είναι οι ίδιες με τις συντεταγμένες του τετραγώνου.
- Εάν το pixel ανήκει στην αριστερή άνω γωνία, τότε αφαιρούμε ένα από τον οριζόντιο και κατακόρυφο άξονα.
- Εάν το pixel ανήκει στην δεξιά άνω γωνία, τότε αφαιρούμε ένα από τον κατακόρυφο άξονα.

```
// μεγάλη περιοχή
Arrayy = SectY
Arrayx = SectX
// αριστερή άνω γωνία
Αν SectPx1Y < (h - SectPx1X * m) τότε
    Arrayy = SectY - 1
    Arrayx = SectX - 1
Τέλος Αν
// δεξιά άνω γωνία
Αν SectPx1Y < (-h + SectPx1X * m) τότε
    Arrayy = SectY - 1
    Arrayx = SectX
Τέλος Αν
```

Σχήμα 4-8 Κώδικας επιλογής εξαγώνου σε τετράγωνα τύπου A.

Τετράγωνα τύπου «B»

Τα τετράγωνα τύπου «B» αποτελούνται από τρεις περιοχές αλλά διαφορετικές από τα τετράγωνα τύπου «A»:

- Εάν το pixel ανήκει δεξιά περιοχή, τότε οι συντεταγμένες του εξαγώνου είναι οι ίδιες με τις συντεταγμένες του τετραγώνου.
- Εάν το pixel ανήκει στην αριστερή περιοχή, τότε αφαιρούμε ένα από τον οριζόντιο άξονα.
- Εάν το pixel ανήκει στην πάνω περιοχή, τότε αφαιρούμε ένα από τον κατακόρυφο άξονα.

```

Av SectPx1X >= r τότε
    Av SectPx1Y < (2 * h - SectPx1X * m) τότε
// αριστερή περιοχή
    Arrayy = SectY - 1
    Arrayx = SectX
    αλλιώς
// δεξιά περιοχή
    Arrayy = SectY
    Arrayx = SectX
    Τέλος Av
Τέλος Av
//αριστερή περιοχή
Av SectPx1X < r τότε
    Av SectPx1Y < (SectPx1X * m) τότε
    Arrayy = SectY - 1
    Arrayx = SectX
    αλλιώς
// πάνω περιοχή
    Arrayy = SectY
    Arrayx = SectX - 1
    Τέλος Av
Τέλος Av

```

Σχήμα 4-9 Κώδικας επιλογής εξαγώνου σε τετράγωνα τύπου B

Αναπαράσταση μιας μεγάλης επιφάνειας

Κατά την φάση ανάπτυξης του simulator πολύ γρήγορα αντιληφθήκαμε ότι για να τρέξουμε μεγάλα κυκλώματα όπως ένα πλήρη αθροιστή χρειάζεται μια επιφάνεια πολύ μεγαλύτερη από μια οθόνη του υπολογιστή. Πρώτα χρειάστηκε να δούμε πόσο απαιτητικός είναι ο αλγόριθμος κίνησης σε υπολογιστική ισχύ. Καταλήξαμε ότι μια επιφάνεια εργασίας 200 x 200 είναι αρκετή για την αναπαράσταση αρκετά

πολύπλοκων κυκλωμάτων όπως ένας αθροιστής 2 bit. Θέσαμε ως όριο κάθε κίνηση των μυρμηγκιών να ολοκληρώνεται σε ένα δευτερόλεπτο. Αυτό επιτεύχθηκε όσο κάναμε υπολογισμούς στην μνήμη.

Για τη σχεδίαση της επιφάνειας στην οθόνη προσπαθήσαμε αρχικά να προσεγγίσουμε το πρόβλημα κάνοντας τα πιο κάτω βήματα.

1. Υπολογισμός μεγέθους δομής εικόνας (bitmap) ώστε να χωρέσει 200 x 200 εξάγωνα συγκεκριμένου μεγέθους (ορισμένα από το χρήστη).
2. Δημιουργία μιας δομής εικόνας στην μνήμη.
3. Σάρωση του πίνακα Matrix κάνοντας έλεγχο για κάθε θέση του αν περιέχει κάποιο θεμελιώδες συστατικό καθώς και αν βρίσκεται μυρμήγκι στην συγκεκριμένη θέση.
4. Σχεδίαση κάθε εξαγώνου με τα περιεχόμενα του στην εικόνα.

Στην μνήμη υπάρχει ολόκληρη η κατάσταση της επιφάνειας εργασίας ζωγραφισμένη και με ένα μηχανισμό κύλισης ο χρήστης θα μπορεί να βλέπει ολόκληρη της επιφάνεια εργασίας.

- Αυτός ο τρόπος έδειξε να έχει δύο σοβαρά μειονεκτήματα:
- Η δημιουργία της εικόνας στην μνήμη χρειάζεται πολύ περισσότερο χρόνο από το αρχικό όριο του ενός δευτερολέπτου.
- Η θέση ενός εξαγώνου είναι δύσκολο να καθοριστεί γιατί η εικόνα αποτελείται από εξάγωνα και όχι τετράγωνα.

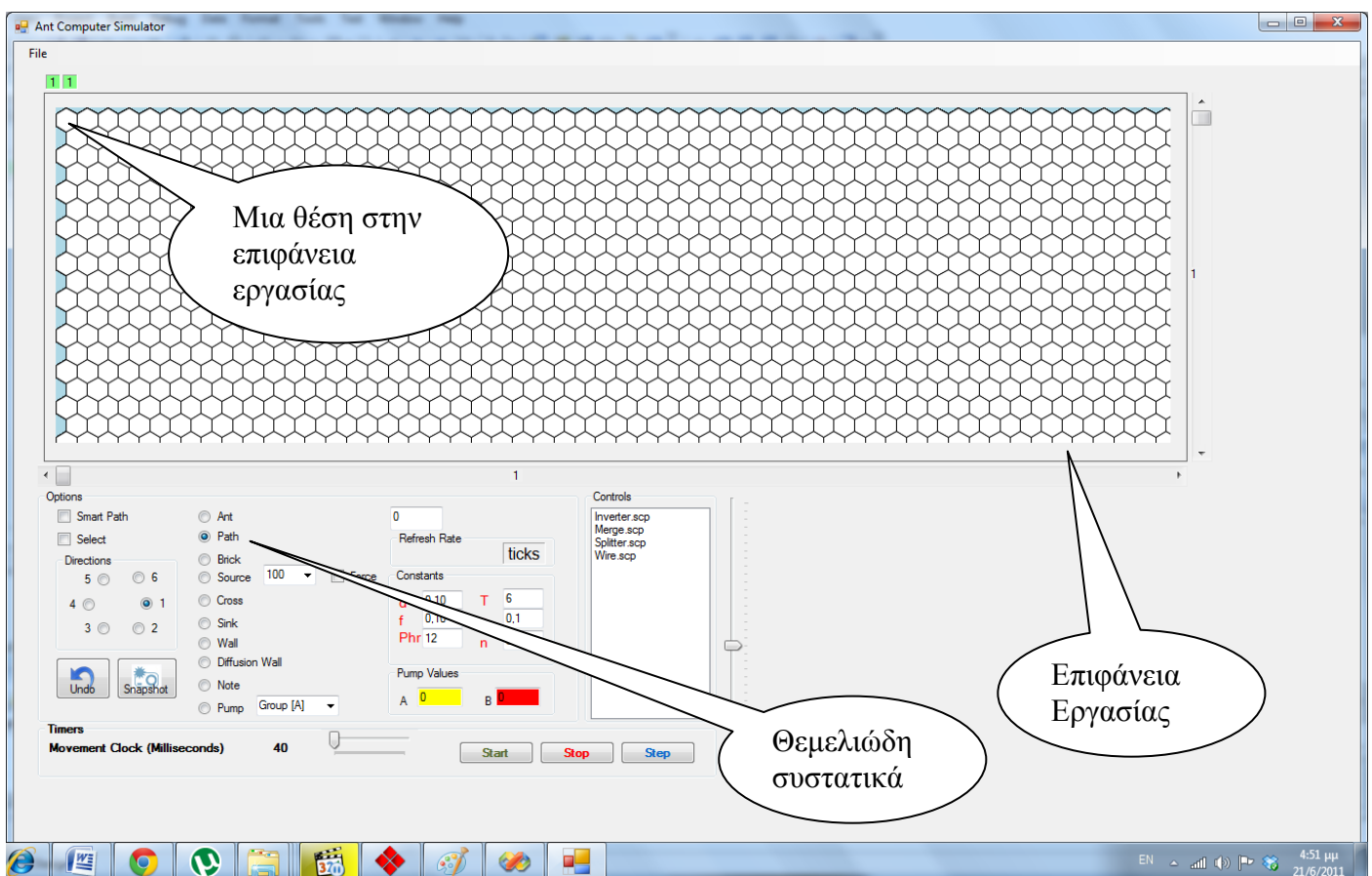
Για να κερδίσουμε χρόνο στην δημιουργία της επιφάνειας εργασίας αποφασίσαμε να σχεδιάσουμε στην μνήμη ότι θα βλέπει ο χρήστης την δεδομένη στιγμή. Προσθέσαμε μπάρες κύλισης τέτοιες ούτως ώστε η πιο μικρή μεταβολή που μπορεί να κάνει ο χρήστης να είναι κατά ένα εξάγωνο. Έτσι ανά πάσα στιγμή βρίσκεται στην μνήμη η κατάσταση της επιφάνειας σχεδίασης και επιλεκτικά προβάλλεται η κατάσταση αυτή στην επιφάνεια εργασίας αναλόγως των τιμών της οριζόντιας και κατακόρυφης μπάρας κύλισης.

4.4 Διεπαφή με το χρήστη

Η διεπαφή με το χρήστη αφορά το γραφικό μέρος του προσομοιωτή. Σε αυτό ο χρήστης μπορεί να τοποθετήσει primitive components στην επιφάνεια εργασίας, να «τρέξει» το simulation και να παρακολουθήσει την κίνηση των μυρμηγκιών. Έγινε προσπάθεια η επιφάνεια της διεπαφής με το χρήστη να είναι απλή και όσο πιο πιστή γίνεται με την περιγραφή του δοσμένου μοντέλου που αναφερθήκαμε στο κεφάλαιο 2.

Περιγραφή

Ο χρήστης μόλις ξεκινήσει την εφαρμογή έχει μπροστά του μια επιφάνεια η οποία είναι στρωμένη με εξάγωνα. Τα εξάγωνα αυτά είναι θέσεις στις οποίες ο χρήστης μπορεί αφού επιλέξει ένα θεμελιώδες συστατικό να το τοποθετήσει σε μια θέση της επιφάνειας εργασίας κάνοντας κλικ σε αυτή. Οι συντεταγμένες της θέσης που βρισκόμαστε εμφανίζονται άνω αριστερά της επιφάνειας εργασίας.



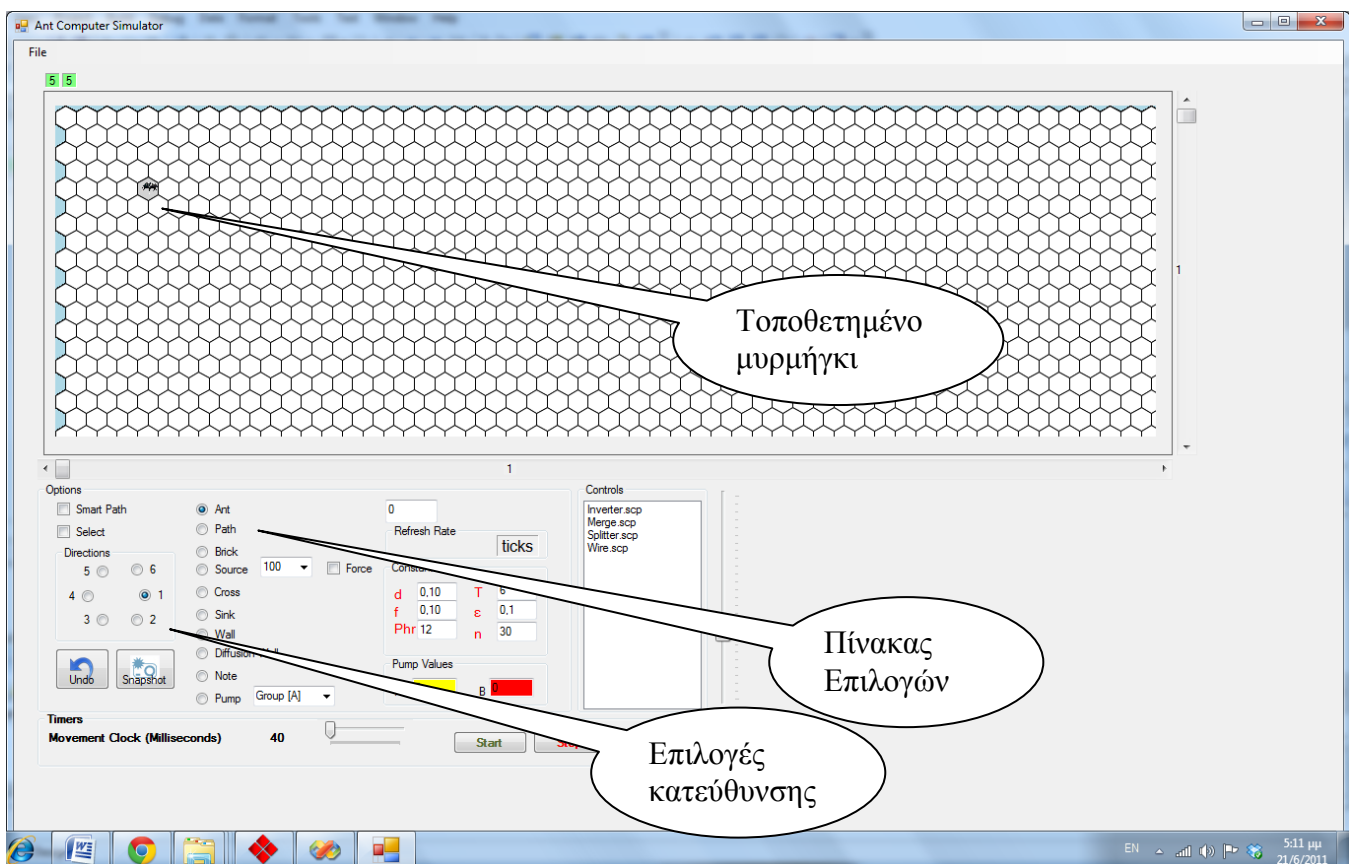
Σχήμα 4-10. Η επιφάνεια εργασίας του προσομοιωτή.

Χάραξη ενός μονοπατιού

Τα μονοπάτια είναι οι θέσεις που χαρακτηρίζονται ως προσπελάσιμες από ένα μυρμήγκι. Κάθε θέση για να είναι προσπελάσιμη πρέπει να διαθέτει αυτό το χαρακτηριστικό. Ο χρήστης μπορεί να δημιουργήσει μονοπάτια κάνοντας κλικ σε κάθε θέση της επιφάνειας εργασίας ή να χρησιμοποιήσει τη λειτουργία έξυπνης σχεδίασης.

Τοποθέτηση μυρμηγκιού

Όπως έχουμε αναφέρει σε πολλά σημεία σε αυτή την εργασία ένα μυρμήγκι χαρακτηρίζεται από την θέση του και την κατεύθυνση του. Για να τοποθετήσουμε ένα μυρμήγκι στην επιφάνεια εργασίας θα πρέπει πρώτα να επιλέξουμε Ant από τον πίνακα επιλογών και κατόπιν την κατεύθυνση του από τις επιλογές κατεύθυνσης. Κατόπιν κάνουμε κλικ στη θέση της επιφάνειας εργασίας που επιθυμούμε να τοποθετήσουμε το μυρμήγκι. Το μυρμήγκι εμφανίζεται στην επιφάνεια εργασίας με την κατεύθυνση που έχουμε επιλέξει.



Σχήμα 4-11. Τοποθέτηση ενός μυρμηγκιού πάνω στην επιφάνεια εργασίας στη θέση 5,5 με κατεύθυνση δεξιά.

Τοποθέτηση τοίχου

Η τοποθέτηση τοίχου γίνεται με παρόμοιο τρόπο με την τοποθέτηση μυρμηγκιού. Αν ο χρήστης επιλέξει να τοποθετήσει τοίχο τότε θα πρέπει να επιλέξει και μια από τις έξι κατευθύνσεις όπως θα επέλεγε αν τοποθετούσε μυρμήγκι.

Καταστάσεις Λειτουργίας

Ο προσομοιωτής λειτουργεί σε δύο καταστάσεις. Σε κατάσταση σχεδίασης ο χρήστης μπορεί να κάνει τα πιο κάτω:

- Τοποθετεί συστατικά στην επιφάνεια εργασίας.
- Θέτει τιμές σε σταθερές που αφορούν τη διάδοση της φερομόνης (d , f), την ποσότητα φερομόνης που εκκρίνεται από τα μυρμήγκια (phr), την ευαισθησία των μυρμηγκιών στη συγκέντρωση φερομόνης (n) καθώς και την ποσότητα φερομόνης που τροφοδοτούν οι αντλίες το σύστημα.
- Επιλέγει κομμάτι της επιφάνειας εργασίας με σκοπό είτε να το αντιγράψει σε άλλο σημείο της επιφάνειας είτε να το διαγράψει είτε να το σώσει.

Όταν ο προσομοιωτής είναι σε κατάσταση “simulation” τότε ξεκινάει η προσομοίωση. Σχεδιάζει όλα τα θεμελιώδη συστατικά και τα μυρμήγκια της κάθε χρονικής στιγμής τρέχοντας τον αλγόριθμο κίνησης των μυρμηγκιών. Στη λειτουργία αυτή απαλείφονται οι θέσεις της επιφάνειας σχεδίασης που δεν παίρνουν μέρος στην προσομοίωση ώστε να είναι πιο ευδιάκριτη η προσομοίωση.

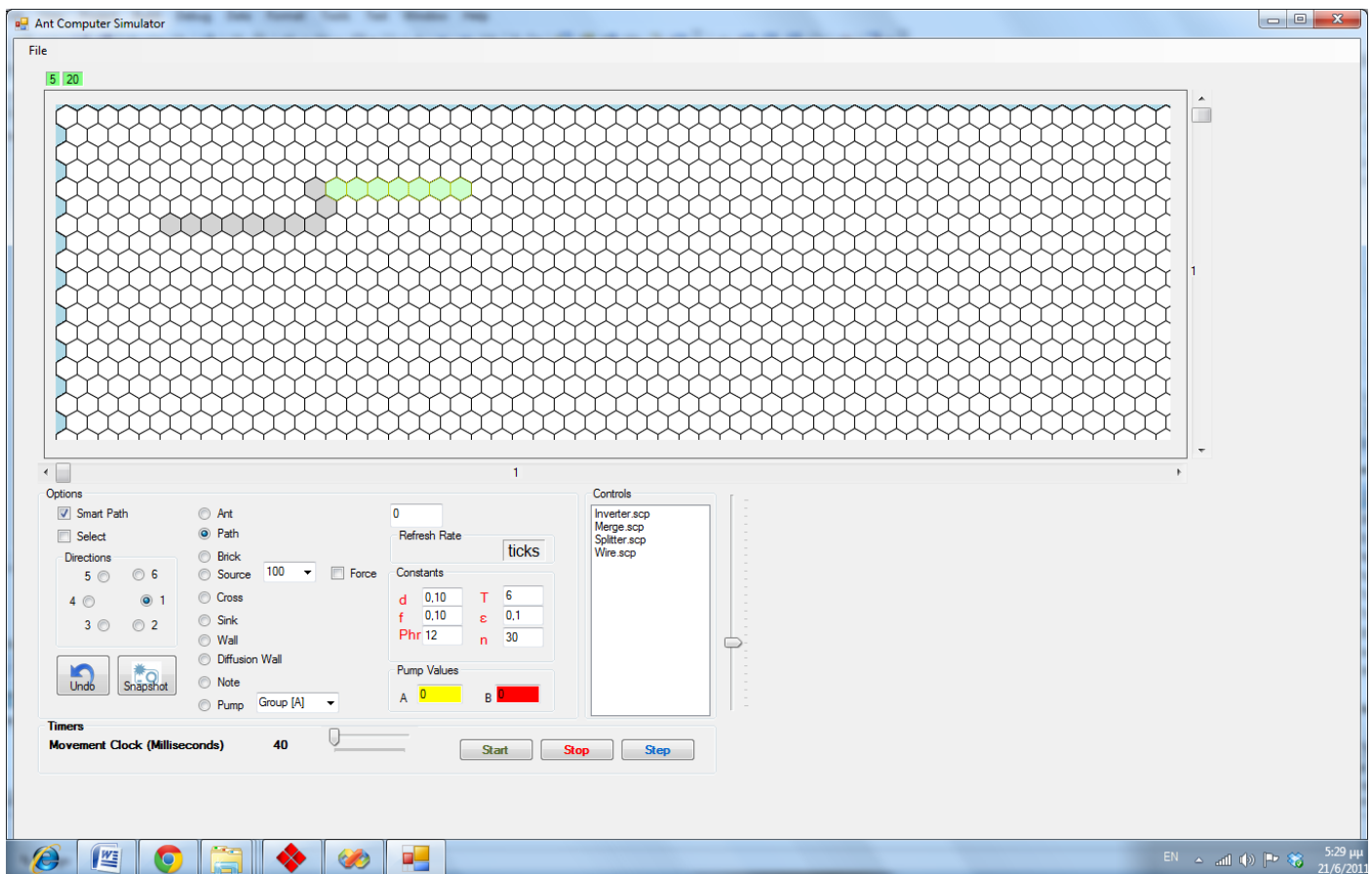
Λειτουργία Έξυπνης Σχεδίασης

Κατά την προσπάθεια σχεδιασμού κυκλωμάτων παρατηρήσαμε ότι ήταν εξαιρετικά άβολο το να σχεδιάζουμε μονοπάτια επιλέγοντας ένα ένα τα εξάγωνα στην επιφάνεια σχεδιασμού. Κρίναμε ότι ένας μηχανισμός που θα επέτρεπε το σχεδιασμό γραμμών από συστατικά μονοπατιών είναι αναγκαίος. Παρατηρώντας τη δομή της επιφάνειας σχεδίασης βλέπουμε ότι μπορούν να χαραχτούν γραμμές από μονοπάτια με δύο τρόπους:

- Χάραξη στον οριζόντιο άξονα.
- Χάραξη στις διαγώνιους.

Ο χρήστης επομένως χρειάζεται ένα σημείο για αρχή της γραμμής από μονοπάτια και ένα σημείο τέλους που θα καθορίζει το τέλος μιας γραμμής. Θέλοντας να κάνουμε τη λειτουργία έξυπνης σχεδίασης περισσότερο φιλική προς το χρήστη προσθέσαμε τη δυνατότητα προεπισκόπησης της γραμμής που πρόκειται να σχεδιάσει, σχεδιάζοντας την στην επιφάνεια σχεδιασμού με διαφορετικό χρώμα και πριν αυτός οριστικοποιήσει τα όρια της νέας γραμμής.

Στη λειτουργία αυτή, ο χρήστης επιλέγει ένα σημείο της επιφάνειας σχεδίασης στην οθόνη πατώντας αριστερό κλικ σε μια θέση. Ο χρήστης μετακινεί το ποντίκι σε ένα σημείο στο board και αν μπορεί να γίνει αυτή η σύνδεση χρωματίζεται όλη η διαδρομή μέχρι το προαναφερθέν σημείο. Αν ο χρήστης θέλει να καταχωρώσει αυτή την διαδρομή ως path τότε κάνει αριστερό κλικ στο σημείο αυτό.



Σχήμα 4-12 Λειτουργία Smart Path

`smart path (startMark point, endMarkPoint)`

Δημιούργησε λίστα με τις θέσεις που περιλαμβάνονται μεταξύ StartMark και endMark και σώσε την κατάσταση τους.

Αν η λίστα αυτή περιέχει στοιχεία τότε κάνε τα highlight.

Οριστικοποίησε αυτά τα στοιχεία δημιουργώντας Μονοπάτι.

Σχήμα 4-10 Ψευδοκώδικας λειτουργίας Smart Path.

Μηχανισμός αναίρεσης

Η ύπαρξη ενός μηχανισμού αναίρεσης κατά την σχεδίαση κρίθηκε επίσης αναγκαία. Έτσι κατά τη σχεδίαση, ο χρήστης μπορεί να κάνει όσες αναιρέσεις προς τα πίσω θέλει. Ο μηχανισμός αποθηκεύει όλες τις ενέργειες του χρήστη που αφορούν εξάγωνα ακόμα και αυτές που έγιναν με τη λειτουργία έξυπνης σχεδίασης. Για την υλοποίηση αυτού του μηχανισμού είναι αναγκαίο να φυλάσσεται η κατάσταση κάθε σημείου ακόμα και αν αυτό επικαλύπτεται αρκετές φορές.

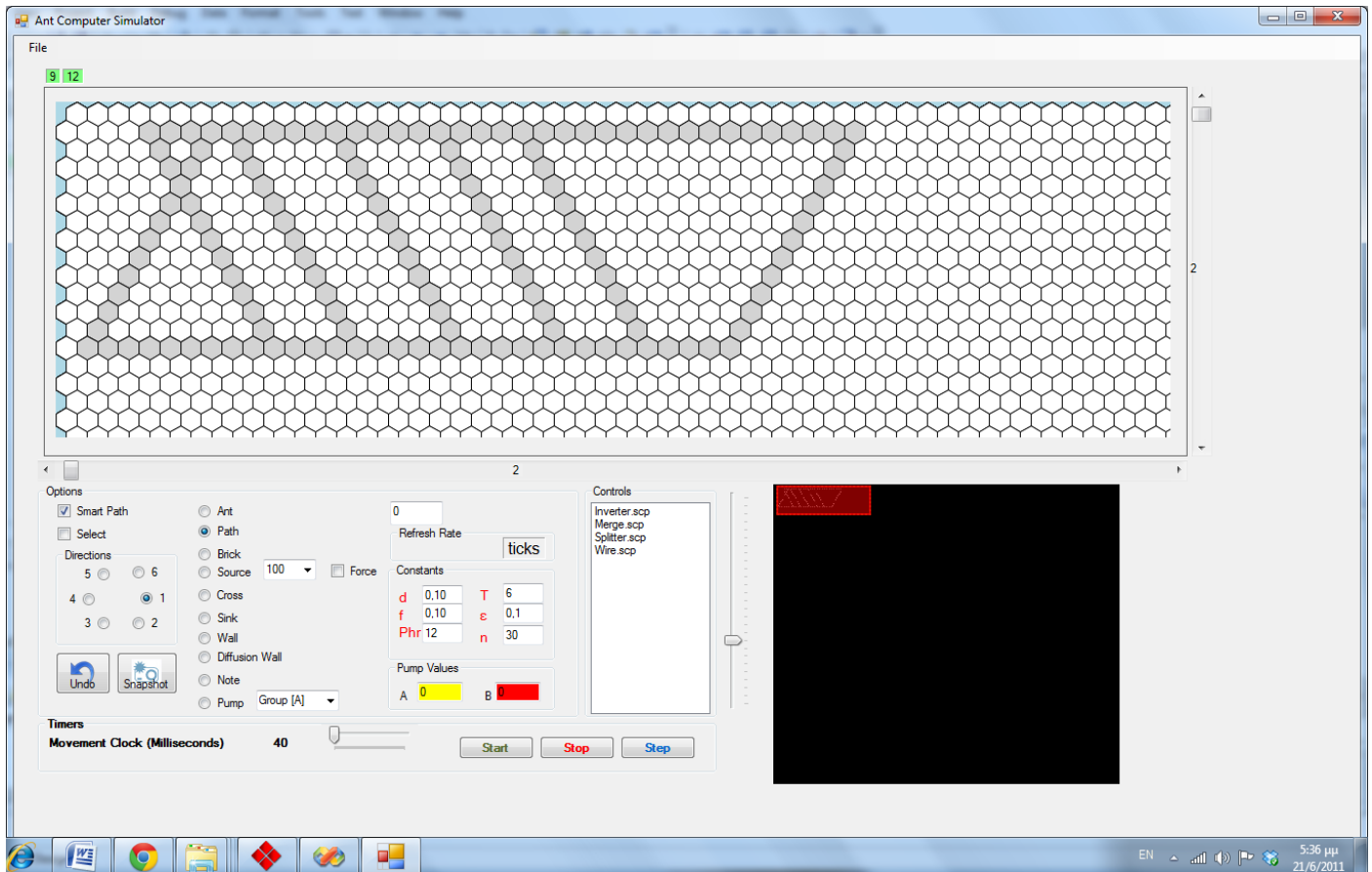
Λειτουργία Selection

Η λειτουργία επιλογής μας δίνει την δυνατότητα select, copy, paste μιας περιοχής του πίνακα μας. Ο χρήστης, αφού έχει επιλέξει το κουμπί «Select», έχοντας πατημένο το αριστερό κουμπί του ποντικιού μπορεί να σύρει το ποντίκι πάνω στην επιφάνεια του πίνακα. Αυτό έχει ως αποτέλεσμα την δημιουργία ενός πλαισίου το οποίο περικλείει την επιφάνεια με την οποία τα components έχουν επιλεγεί. Ακολούθως πατώντας δεξί κλικ εμφανίζονται οι επιλογές cut, copy, paste. Η συμπεριφορά αυτών των επιλογών είναι όπως σε όλες τις windows εφαρμογές.

Λειτουργία χάρτη επιφάνειας σχεδίασης

Επειδή η επιφάνεια σχεδίασης στην οθόνη ενός υπολογιστή είναι πολύ μικρή, κρίναμε αναγκαίο να εισάγουμε ένα χάρτη επιφάνειας σχεδίασης. Ο χάρτης αυτός είναι μια μικρογραφία του πίνακα εξαγώνων που υπάρχουν στην μνήμη και κάθε εξάγωνο αναπαριστάται με δύο εικονοστοιχεία. Ο χρήστης μπορεί να σείρει το πλαίσιο επιλογής με το ποντίκι στην επιφάνεια του χάρτη και να επιλέξει μετάβαση στο σημείο αυτό πάνω στην επιφάνεια εργασίας. Ας σημειωθεί ότι ένα πλαίσιο επιλογής στο χάρτη σχεδίασης

επιφάνειας αντιστοιχεί με όλη την σχεδιαστική επιφάνεια που είναι ορατή προς το χρήστη. Ο χρήστης μπορεί να επιλέξει μεγέθυνση και το μέγεθος του πλαισίου επιλογής να μικραίνει ή να μεγαλώνει διατηρώντας πάντα την αναλογία της σχεδιαστικής επιφάνειας με το πλαίσιο επιλογής.



Σχήμα 4-11 Η σχεδιαστική επιφάνεια και ο χάρτης επιφάνειας στο κάτω δεξιά μέρος της οθόνης

Κεφάλαιο 5

Συμπεράσματα - Επεκτάσεις

Στο κεφάλαιο αυτό παρουσιάζουμε κάποιους προσομοιωτές της βιβλιογραφίας που αφορούν μυρμήγκια. Γίνεται μια μικρή σύγκριση τους με το προσομοιωτή που παρουσιάζουμε και αναπτύσσονται θέματα όπως οι μελλοντικές επεκτάσεις του προσομοιωτή καθώς και τα γενικά συμπεράσματα μας .

5.1 Προσομοιωτές με μυρμήγκια

3D Ant Farm Simulation [10]

Το 3D Ant Farm Simulation είναι ένα πρόγραμμα τρισδιάστατης προσομοίωσης. Ο συγκεκριμένος προσομοιωτής παρουσιάζει δύο αποικίες μυρμηγκιών οι οποίες ανταγωνίζονται η μια την άλλη στην εξεύρεση τροφής. Ο χρήστης μπορεί να αλληλεπιδρά με το σύστημα μετακινώντας εμπόδια ή πηγές τροφής και να παρατηρεί τη διάδοση της φερομόνης καθώς και τις διαδρομές που επιλέγουν τελικά τα μυρμήγκια για τη μεταφορά τροφής στην αποικία τους.



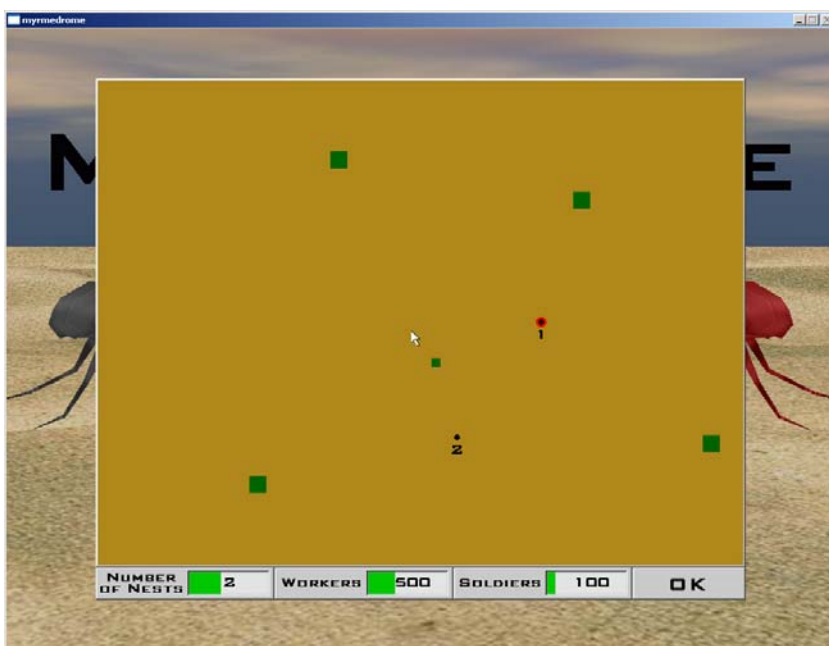
Σχήμα 5-1 Οι δύο αποικίες μυρμηγκιών.



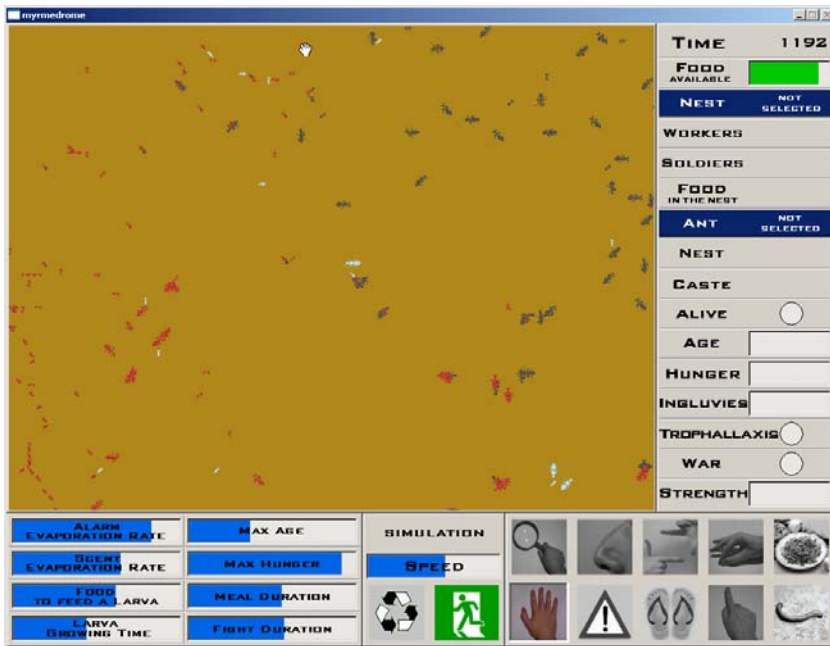
Σχήμα 5-2 Μονοπάτια φερομόνων.

Myrmedrome [11]

Το Myrmedrome είναι μια προσομοίωση που προσπαθεί να μιμηθεί τη ζωή των μυρμηγκιών όσο πιο πιστά γίνεται χωρίς την προσθήκη τεχνητών (artificial) ιδιοτήτων. Δέχεται παραμέτρους για τον αριθμό των αποικιών που θα λάβουν μέρος στην προσομοίωση, τον αριθμό για τα μυρμήγκια-εργάτες και τον αριθμό για τα μυρμήγκια-στρατιώτες. Μπορεί να παρουσιάσει μάχες μυρμηγκιών, νεκρά μυρμήγκια και τη συγκομιδή τροφής από μια πηγή τροφής προς στη φωλιά των μυρμηγκιών.



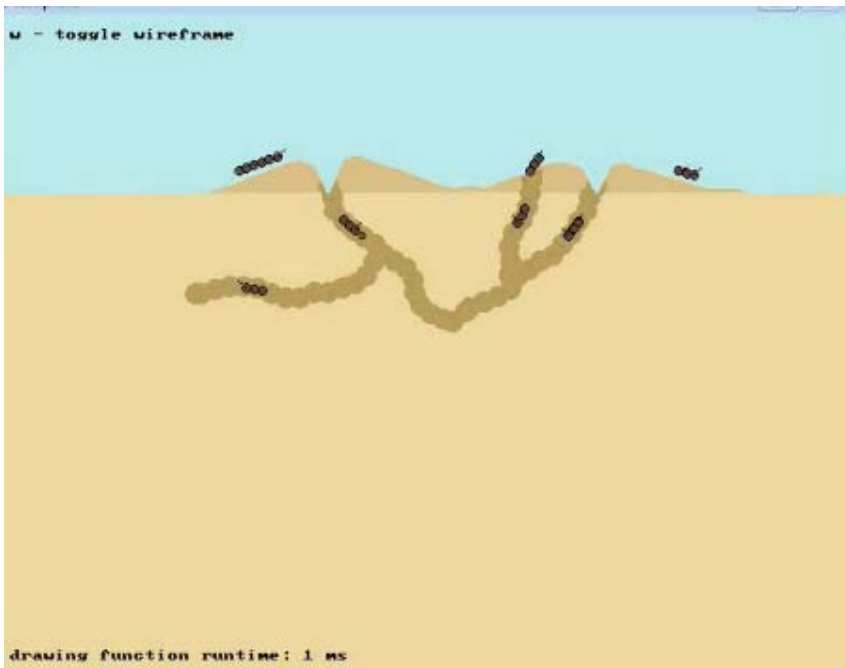
Σχήμα 5-3 Επιλογή Παραμέτρων.



Σχήμα 5-4 Περιβάλλον προσομοίωσης

Ant Farm Simulation [12]

Είναι ένα προσομοιωτής που προσομοιώνει τη χάραξη μονοπατιών από τα μυρμηγκία καθώς και τον καθαρισμό της φωλιάς τους από ακαθαρσίες.



Σχήμα 5-5 Περιβάλλον προσομοίωσης

5.2 Συμπεράσματα

Οι προσομοιωτές μυρμηγκιών που προτείνονται από τη βιβλιογραφία αφορούν προσομοίωση της βιολογικής συμπεριφοράς των μυρμηγκιών δηλαδή παρουσιάζουν το πώς αντιδρούν με το περιβάλλον τους, πώς γίνεται η εξεύρεση τροφής ή άλλα χαρακτηριστικά τους. Κανένα από αυτά δεν περιγράφει διάταξη μυρμηγκιών στα πλαίσια ενός βιολογικού μοντέλου με τέτοιο τρόπο ώστε να δημιουργηθούν συνθήκες για την εκτέλεση μαθηματικών πράξεων.

Τοποθετώντας μυρμήγκια σε ένα ελεγχόμενο περιβάλλον βάση του μοντέλου που περιγράψαμε είναι εφικτό να αναπαραστήσουμε όποιο υπολογισμό θέλουμε. Χρησιμοποιήσαμε τον προσομοιωτή για αρκετά μεγάλα κυκλώματα όπως είναι ο αθροιστής 2 bit (σχήμα 5-5). Φυσικά ένας αθροιστής είναι πολύ μικρός σε σχέση με τα εκατομμύρια τρανζίστορ που χρειάζονται για να δημιουργηθεί ένας μοντέρνος επεξεργαστής. Θεωρούμε όμως ότι λύσαμε το πρόβλημα σε μικρή κλίμακα. Αναμένουμε την ίδια συμπεριφορά και σε πολύ μεγαλύτερα κυκλώματα. Σίγουρα όσο πιο πολλά συστατικά παίρνουν μέρος στην προσομοίωση τόσο πιο αργά θα εξελίσσεται.

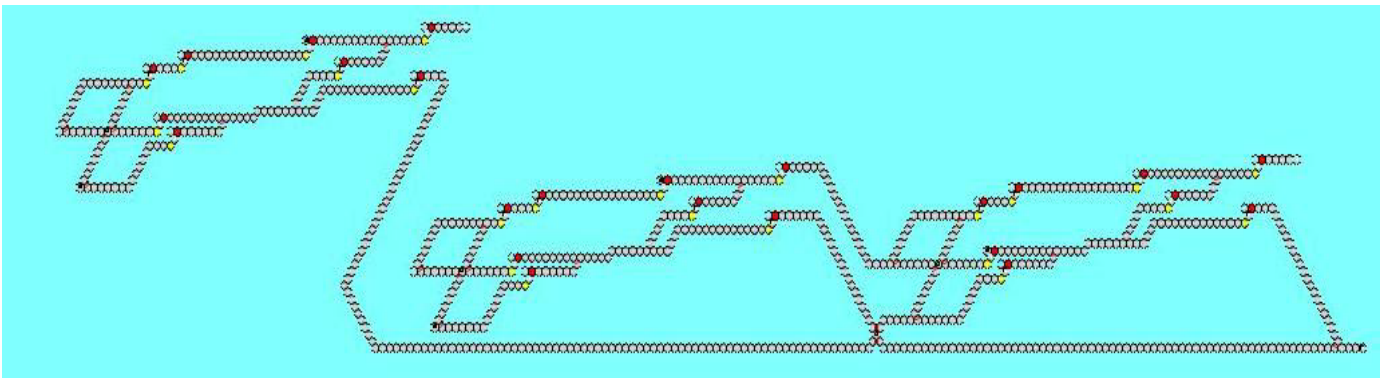
5.3 Μελλοντικές Επεκτάσεις

Ο αλγόριθμος κίνησης που παρουσιάσαμε είναι σειριακός. Η επανασχεδίαση του με τη σκέψη να λειτουργεί παράλληλα, θα εκμεταλλευτεί τις δυνατότητες παράλληλης επεξεργασίας των μοντέρνων επεξεργαστών και τη προσομοίωση ακόμη μεγαλύτερων κυκλωμάτων.

Η προσθήκη ενός επιπέδου αφαίρεσης στη σχεδίαση κυκλωμάτων θα δώσει μεγάλες δυνατότητες στη σχεδίαση. Συγκεκριμένα ο χρήστης θα μπορεί να συνδέει μεταξύ τους components όπως ακριβώς θα συνέδεε λογικά κυκλώματα υποστηρίζοντας ταυτόχρονα την προσομοίωση σε επίπεδο επιφάνειας σχεδίασης.

Η χρήση του ως εκπαιδευτικό εργαλείο για κάποιο νεοφερμένο στην ψηφιακή λογική. Οι εκπαιδευόμενοι θα αντιλαμβάνονται ευκολότερα την λογική των λογικών κυκλωμάτων και θα μπορούν παρατηρώντας την συμπεριφορά των μυρμηγκιών να την παραλληλίσουν με τη συμπεριφορά του ρεύματος σε μια λογική πύλη.

Τέλος θα ήταν μια πρόκληση να γίνουν όλες οι πιο πάνω προεκτάσεις προσθέτοντας ακόμα και τη δυνατότητα προσομοίωσης με γραφικά 3D (τρισδιάστατα). Ο χρήστης θα παρατηρεί την προσομοίωση από όποια γωνία θέλει, να μεγεθύνει, να μπορεί να βλέπει πως σκιάζονται τα αντικείμενα όταν αυτά επηρεάζονται από πηγές φωτός και γενικότερα να νιώθει ότι παίρνει μέρος σε αυτή.



Σχήμα 5-6 Αθροιστής 2 bit

Βιβλιογραφία

1. Loizos Michael Ant-Based Computing. Artificial Life, Summer 2009, Vol. 15, No. 3, Pages 337-349.
2. Eric Bonabeau (Author), Guy Theraulaz (Author), Marco D. Swarm Intelligence: From Natural to Artificial Systems.
3. <http://www.swingame.com/> [22-06-2011].
4. <http://www.blender.org/> [22-06-2011].
5. [http://msdn.microsoft.com/en-us/library/ms533798\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533798(v=VS.85).aspx) [22-06-2011].
6. J.L. Deneubourg, S. Goss, N. Franks, J.M. Pasteels, The Blind Leading the Blind: Modeling Chemically Mediated Army Ant Raid Patterns, Journal of Insect Behavior, Vol.2, No. 5, 1989, 719-725.
7. J.L. Deneubourg, S. Aron, S. Goss, J.M. Pasteels, The Self-Organizing Exploratory Pattern of the Argentine Ant, Journal of Insect Behavior, Vol. 3, No. 2, 1990, 159-169.
8. S. Goss, S. Aron, J.L. Deneubourg, J.M. Pasteels, Self-organized shortcuts in the Argentine ant, Naturwissenschaften 76 (1989) 579–581.
9. <http://www.youtube.com/watch?v=-AiLyQWXjIg> [06-07-2011]
10. <http://www.forgefx.com/casestudies/prenticehall/ph/ants/ants.htm> [06-07-2011]
11. http://www.not-equal.eu/myrmedrome/main_en.html [06-07-2011]
12. <http://www.youtube.com/watch?v=h5HHUBQ5bUw&feature=related> [06-07-2011]
13. http://en.wikipedia.org/wiki/HSL_and_HSV [06-07-2011]

Παράρτημα

Το παράρτημα αυτό δεν είναι απλά ένα εγχειρίδιο χρήσης μιας βιβλιοθήκης γραφικών. Εξάλλου αναλυτική παρουσίαση της χρήσης του GDI+ μπορεί κάποιος να βρει στην ιστοσελίδα [11] της εταιρίας ανάπτυξης του. Ας θεωρηθεί ένα σημείο εκκίνησης για κάποιο που ξεκινάει προγραμματισμό με γραφικά. Σκοπός του είναι η γνωριμία με όρους, ορισμούς που αφορούν θέματα γραφικών παρουσιάζοντας ταυτόχρονα κάποια βασικά χαρακτηριστικά της βιβλιοθήκης GDI+. Αναλύονται θέματα που έχουν ισχύ σε οποιαδήποτε περιβάλλοντα σχεδιασμού με ηλεκτρονικό υπολογιστή. Εξηγεί όρους λειτουργίες και τεχνικές που κάθε προγραμματιστής θα συναντήσει αν θελήσει να ασχοληθεί με θέματα γραφικών.

Σχεδιάζοντας Επιφάνειες

Όταν γράφουμε κώδικα για GDI+, μπορούμε να εξάγουμε τα γραφικά μας σε τρία μέσα: σε ένα παράθυρο στην οθόνη, σε μια σελίδα που στάλθηκε σε ένα εκτυπωτή ή σε μια εικόνα (Bitmap) στην μνήμη. Κάθε μέσο απαιτεί μια επιφάνεια σχεδιασμού δηλαδή μια επιφάνεια που αποτελείται από εικονοστοιχεία. Τα χαρακτηριστικά για την κάθε επιφάνεια είναι τα ίδια (μέγεθος, ανάλυση εικονοστοιχείων, βάθος χρώματος) αλλά ο τρόπος χειρισμού διαφοροποιείται ανάλογα του τύπου της επιφάνειας. Στο παράρτημα θα ασχοληθούμε περισσότερο με το πώς σχεδιάζουμε στην οθόνη και στην μνήμη.

Ένα σύστημα συντεταγμένων χρησιμοποιείται για να περιγράψει θέσεις στην επιφάνεια σχεδιασμού. Μέσω του συστήματος συντεταγμένων καθορίζουμε το μέγεθος, το σχήμα και τη θέση κάθε στοιχείου ενός γραφήματος. Είναι πολύ σημαντική η σωστή χρήση του συστήματος συντεταγμένων γιατί μόνο ένα εικονοστοιχείο να τοποθετηθεί εκ παραδρομής σε λάθος σημείο μπορεί να αλλοιώσει κατά πολύ το αναμενόμενο οπτικό αποτέλεσμα. Έρευνες έχουν δείξει ότι οι άνθρωποι μπορούν να δουν αντικείμενα που έχουν μέγεθος το 1/3000 της ίντσας όταν το αντικείμενο αυτό είναι πηγή φωτός.

Raster-Based και Vector-Based συστήματα

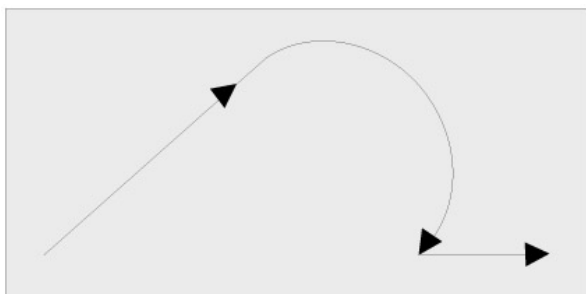
Μια επιφάνεια σχεδιασμού, όπως αυτή ορίζεται στο GDI+ είναι μια αφηρημένη (abstract) αναπαράσταση ενός παραθύρου στην οθόνη, μιας σελίδας στον εκτυπωτή ή μια εικόνα στην μνήμη. Ο τρόπος που

αναπαριστούνται οι εικόνες στο GDI+ είναι Raster-based. Raster-Based είναι η αναπαράσταση μιας εικόνας που αποτελείται από εικονοστοιχεία σε ένα δισδιάστατο πλέγμα. Το εικονοστοιχείο είναι το πιο μικρό στοιχείο μιας επιφάνειας σχεδιασμού στο οποίο μπορεί να γίνει αναφορά. Η αναφορά σε ένα στοιχείο μπορεί να γίνει με ζεύγος συντεταγμένων της μορφής (x,y), ακολουθούν δηλαδή το καρτεσιανό σύστημα συντεταγμένων.

Για να σχεδιάσουμε μια γραμμή σε ένα Raster-Based σύστημα χρωματίζουμε μια ακολουθία από εικονοστοιχεία στο πλέγμα μας. Η γραμμή ορίζεται δηλώνοντας τις συντεταγμένες του κάθε εικονοστοιχείου που χρησιμοποιείται για τον σχεδιασμό.

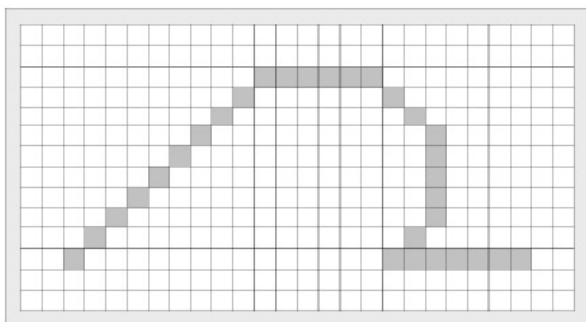
Σε ένα Vector Based σύστημα η διαδικασία είναι αρκετά διαφορετική. Στο Vector Based σύστημα δεν αναφερόμαστε σε εικονοστοιχεία αλλά σε διανύσματα τα οποία έχουν ένα καλά ορισμένο σημείο εκκίνησης, διεύθυνση (που ορίζεται με γωνία ή διαδρομή) και μήκος.

Το πιο κάτω σχήμα παρουσιάζει την αναπαράσταση ενός Vector-Based Γραφήματος.



Σχήμα 1

Vector-Based γράφημα που παρουσιάζει τρία διανύσματα.



Σχήμα 2

Raster –Based αναπαράσταση του ίδιου γραφήματος.

Η επιτυχής μετατροπή ενός διανύσματος σε εικονοστοιχεία εξαρτάται από την ανάλυση (resolution) των εικονοστοιχείων της Raster-Based επιφάνειας. Η ανάλυση εξαρτάται από το μέγεθος των εικονοστοιχείων. Στην περίπτωση μας το γράφημα του σχήματος 2 είναι σε πολύ χαμηλή ανάλυση.

Στις μέρες μας σχεδόν όλα τα μέσα αναπαραγωγής γραφικών είναι Raster-Based. Οι Raster-Based συσκευές είναι πολύ πιο εύκολο να σχεδιαστούν και να παραχθούν. Συχνά, για προγραμματιστικούς λόγους θεωρούμε ότι είναι Vector-Based διότι είναι πιο εύκολο να περιγράψουμε μια εικόνα χρησιμοποιώντας Vector-Based τεχνικές και να εξαρτόμαστε από ένα μηχανισμό που επιτρέπει την μετατροπή τους σε Raster-Based την στιγμή που θα προβληθεί σε μια επιφάνεια σχεδιασμού.

Όταν γράφουμε κώδικα για γραφικά θέλουμε να σκεφτόμαστε με όρους του τύπου « σχεδιάσε μια γραμμή από το σημείο A στο σημείο B με συγκεκριμένο πάχος γραμμής». Δεν χρειάζεται να γνωρίζουμε για μεμονωμένα εικονοστοιχεία. Οι Vector-Based σχεδιασμοί περιγράφουν τα σχήματα με αυτό ακριβώς τον τρόπο.

Σε μερικά χρόνια οι Vector-Based οθόνες θα είναι πολύ πιο διαδεδομένες από ότι είναι τώρα. Πολύπλοκα ηλεκτρονικά κυκλώματα θα μπορούν να καθοδηγούν τους καθοδικούς σωλήνες των οθονών με τέτοιο τρόπο για να σχεδιάσουν διανύσματα.

Η πιο σημαντική τεχνική που χρησιμοποιείται στις μέρες μας είναι η συγγραφή κώδικα χρησιμοποιώντας συντεταγμένες διανυσμάτων και η απεικόνιση τους σε συσκευές που είναι Raster-Based. Πραγματικά, πολλές από τις σχεδιαστικές λειτουργίες που παρέχει το GDI+ εκφράζονται με διανύσματα. Το GDI+ παίρνει ότι έχει εκφραστεί σε διανύσματα και τα μετατρέπει σε raster-based για να μπορέσουν να παρουσιαστούν σε μια συσκευή εξόδου.

Χαρακτηριστικά μιας επιφάνειας σχεδιασμού (Drawing Surface)

Κάθε σχέδιο έχει τρία χαρακτηριστικά: μέγεθος (πλάτος και ύψος), ανάλυση εικονοστοιχείων και βάθος χρώματος. Ας εξετάσουμε αυτά τα χαρακτηριστικά αναλυτικότερα

Το μέγεθος μιας επιφάνειας σχεδιασμού

Το μέγεθος μιας επιφάνειας σχεδιασμού καθορίζεται από τον αριθμό των εικονοστοιχείων που είναι τοποθετημένα οριζόντια και κάθετα σε αυτή. Για παράδειγμα η raster-based εικόνα που φαίνεται στο σχήμα 2 είναι σχεδιασμένη σε ένα πλέγμα μεγέθους 26 εικονοστοιχείων οριζόντια και 14 εικονοστοιχείων κατακόρυφα.

Ανάλυση επιφάνειας σχεδιασμού

Στο GDI+ η ανάλυση σχεδιασμού εκφράζεται πάντα σε εικονοστοιχεία ανά ίντσα ή (**PPI, pixels per inch**) ή στιγμές ανά ίντσα (**DPI, dots per inch**). Εάν μια επιφάνεια έχει ανάλυση 72 Dpi αυτό σημαίνει ότι αν πάρουμε μια τετραγωνική ίντσα στην επιφάνεια αυτή θα μετρήσουμε 72 εικονοστοιχεία στον οριζόντιο άξονα και 72 στον κατακόρυφο.

Μερικές φορές χρειάζεται να βρούμε την ανάλυση μιας σχεδιασμένης επιφάνειας. Αν σχεδιάζουμε ένα πολύπλοκο διάγραμμα θα βρούμε πολύ βολικό να το εκτυπώσουμε χρησιμοποιώντας υψηλή ανάλυση. Αν σχεδιάζουμε όμως σε ένα μέσω χαμηλής ανάλυσης όπως είναι η οθόνη ίσως χρειαστεί να τροποποιήσουμε τον αλγόριθμο σχεδίασης για να πάρουμε καλύτερο αποτέλεσμα.

Ανάλυση οθόνης (Monitor)

Η οθόνη θεωρούμε ότι έχει ανάλυση 96 DPI άσχετα της ανάλυσης της οθόνης. Αυτό γίνεται γιατί ιστορικά δεν υπήρχε τρόπος να ξέρει ο οδηγός της κάρτας γραφικών ποια ήταν η ανάλυση της οθόνης. Αυτή η κατάσταση μπορεί να αλλάξει καθώς οι plug and play οθόνες ενημερώνουν τον οδηγό της κάρτας γραφικών για το μέγεθος της ανάλυσης τους. Γνωρίζοντας το μέγεθος της οθόνης και την ανάλυση της παρέχεται αρκετή πληροφορία για να μπορέσουμε να υπολογίσουμε την ανάλυση.

Βάθος χρώματος επιφάνειας σχεδιασμού

Πολλές λειτουργίες στο GDI+ προαπαιτούν χρώμα. Όταν σχεδιάζουμε μια γραμμή ή ένα ορθογώνιο ή κείμενο το GDI+ χρειάζεται να ξέρει ποιο χρώμα θα χρησιμοποιηθεί για τη σχεδίαση. Το βάθος χρώματος

μιας επιφάνειας σχεδίασης έχει να κάνει με το πόση πληροφορία πρέπει να φυλάγεται (αριθμός χρωμάτων) για κάθε εικονοστοιχείο της επιφάνειας αυτής.

Υποστήριξη χρωμάτων

Στο GDI+ σε κάθε εικονοστοιχείο φυλάγεται η πληροφορία για το χρώμα. Ένα χρώμα στο GDI+ εκφράζεται με το συνδυασμό των τριών βασικών χρωμάτων, το κόκκινο το πράσινο και το μπλε (R,G,B). Υπάρχουν 256 διαβαθμίσεις για το κάθε χρώμα με πεδίο τιμών το 0 μέχρι 255, οπότε κάθε εικονοστοιχείο μπορεί να έχει οποιοδήποτε χρώμα από μια παλέτα 16.777.216 χρωμάτων (256 X 256 X 256). Για να μπορούν να αναπαρασταθούν όλα αυτά τα χρώματα η επιφάνεια θα πρέπει να υποστηρίζει 24 bit χρώμα.

Η δομή Color

Στο GDI+ τα χρώματα μπορούν να αναπαρασταθούν χρησιμοποιώντας τιμές RGB ή HSB. Τα χρώματα στο GDI+ έχουν επίσης Alfa Blending τιμές οι οποίες μπορούν να καθορίσουν το βαθμό διαφάνειας ενός χρώματος.

RGB Τιμές

Μπορούμε να δημιουργήσουμε ένα νέο Instance μιας δομής color με το να περάσουμε παραμέτρους στην τύπου static συνάρτηση της color όπως πιο κάτω.

```
Color c = Color.FromArgb(100, 100, 255);
```

```
Brush b = new SolidBrush(c);
```

```
g.FillRectangle(b, ClientRectangle);
```

Προτυποποιημένα χρώματα (Colors)

Ένας άλλος τρόπος να επιλέγουμε χρώματα είναι χρησιμοποιώντας το όνομα τους αντί να πρέπει να θυμόμαστε το συνδυασμό των RGB τιμών τους. Υπάρχουν 141 χρώματα στα οποία μπορεί να γίνει αναφορά με το όνομα όπως πιο κάτω.

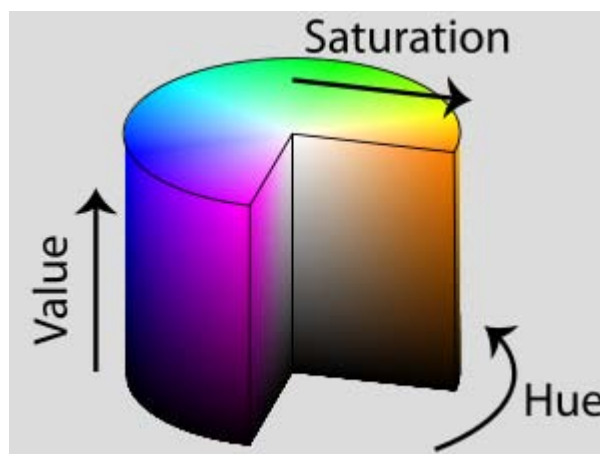
```
Color c1 = Color.LavenderBlush;
```

```
Color c2 = Color.Red;
```

HSB Τιμές

Ένας άλλος τρόπος για να προσδιορίσουμε κάποιο χρώμα είναι να το διαχωρίσουμε στα πιο κάτω συστατικά, γνωστό σαν HSB μοντέλο.

- **Hue:** Είναι το χρώμα βασισμένο στο μήκος κύματος τους. Το Hue έχει εύρος τιμών κινητής υποδιαστολής από 0 μέχρι 360. Μια τιμή Hue αναπαριστά μια γωνία σε μοίρες που με τη σειρά της αναπαριστά ένα χρώμα που βρίσκεται σε ένα δίσκο χρώματος.
- **Saturation:** Καθορίζει πόσο έντονο είναι ένα χρώμα. Ένα χρώμα με υψηλό saturation εμφανίζεται ως πολύ ζωντανό – αυθεντικό. Ένα χρώμα με χαμηλό saturation εμφανίζεται ως φθαρμένο. Το saturation έχει εύρος τιμών αριθμών κινητής υποδιαστολής από 0 μέχρι 1.
- **Brightness ή Value:** Καθορίζει τη σχετική φωτεινότητα ενός χρώματος. Έχει εύρος τιμών αριθμών κινητής υποδιαστολής από το 0 (μαύρο) μέχρι 1 (άσπρο).



Σχήμα 3 Το HSB [13] μοντέλο.

Η δομή Color περιέχει στατικές μεθόδους οι οποίες επιτρέπουν την ανάκτηση των HSB τιμών. Ακολουθεί παράδειγμα για το πώς χρησιμοποιούνται.

```
Color c = Color.Blue;
```

```
float h = c.GetHue();
```

```
float s = c.GetSaturation();
```

```
float b = c.GetBrightness();
```

Συνήθως το HSB μοντέλο δεν χρησιμοποιείται στον προγραμματισμό. Είναι περισσότερο διαδεδομένο στο χώρο του Graphic Design. Ας σημειωθεί ότι υπάρχουν μαθηματικές συναρτήσεις οι οποίες συσχετίζουν το RGB με το HSB μοντέλο και μπορεί εύκολα να γίνει μετατροπή των τιμών από το ένα μοντέλο στο άλλο.

Το χαρακτηριστικό Alpha

Τα χρώματα στο GDI+ μπορούν να έχουν ένα επιπρόσθετο χαρακτηριστικό. Το χαρακτηριστικό Alpha το οποίο μπορεί να χρησιμοποιηθεί για να καθορίσει το βαθμό διαφάνειας ενός χρώματος (transparency). Είναι ιδιαίτερα χρήσιμο γιατί μπορεί να γίνει σχεδίαση πάνω από άλλα γραφικά αλλά και στην δημιουργία οπτικών εφέ.

Το alpha καθορίζεται από ένα ακέραιο αριθμό με εύρος τιμών από το 0 μέχρι το 255 όπου το 0 αναπαριστά το εντελώς διάφανο χρώμα. Εάν δεν χρησιμοποιηθεί το alpha το GDI+ θεωρεί ότι χρησιμοποιείται η τιμή 255 (μη διαφανές).

Σχεδιάζοντας σε εικόνες (Images)

Το μέγεθος μιας εικόνας στο GDI+ καθορίζεται την στιγμή που το αντικείμενο Image δημιουργείται. Εάν διαβάσουμε μια εικόνα από αρχείο (όπως BMP, GIF, JPG) τότε το μέγεθος της καθορίζεται από το αρχείο.

Μπορεί να οριστεί η ανάλυση σε όποια DPI επιθυμούμε. Τα αντικείμενα τύπου Image μπορούν να έχουν μεγάλη ποικιλία σε βάθος χρώματος, από 1bit μονοχρωμία, 16 bit, 24-bit, 32-bit, 48-bit, ή 64-bit βάθος.

Όσο πιο μεγάλο είναι το βάθος χρώματος τόσο πιο καλή η ποιότητα της εικόνας.

Η κλάση Graphics

Κάθε φορά που επιθυμούμε να σχεδιάσουμε σε μια επιφάνεια σχεδίασης πρέπει να έχουμε ένα Instance της κλάσης Graphics.

Δημιουργία ενός αντικειμένου Graphics σε μια φόρμα

Στο πιο κάτω παράδειγμα βλέπουμε πως μπορούμε να δηλώσουμε ένα instance της κλάσης Graphics για να σχεδιάσουμε σε μια φόρμα των Windows, όταν ενεργοποιείται το γεγονός Mouse Move δηλαδή με την μετακίνηση του ποντικιού.

```
private void Form1_MouseMove(object sender,
    System.Windows.Forms.MouseEventArgs e)
{
    Graphics g = this.CreateGraphics();
    ...}
```

Σχεδιασμός σε μια επιφάνεια σχεδιασμού (Drawing Surface)

Ο σχεδιασμός σε μια επιφάνεια γίνεται καλώντας μεθόδους του αντικειμένου Graphics. Στον πίνακα που ακολουθεί εμφανίζονται επιλεγμένες μέθοδοι για σχεδίαση και χρωμάτισμα που παρέχονται από την κλάση Graphics. Υπάρχουν αρκετές μέθοδοι στην κλάση Graphics. Έχουν επιλεγθεί οι πιο ενδεικτικές ώστε να δώσουν μια εικόνα για τις δυνατότητες του GDI+

Ενδεικτικές μέθοδοι της κλάσης Graphics

Μέθοδος	Περιγραφή
Clear	Χρωματίζει την σχεδιαστική επιφάνεια με συγκεκριμένο χρώμα.
DrawArc	Σχεδιάζει μια ελλειπτική καμπύλη. Η καμπύλη είναι κομμάτι από μια έλειψη.
DrawBezier	Σχεδιάζει καμπύλη Bezier.
DrawCurve	Σχεδιάζει μία cardinal spline.
DrawEllipse	Σχεδιάζει μία έλλειψη.
DrawIcon	Σχεδιάζει ένα εικονίδιο.

Ενδεικτικές μέθοδοι της κλάσης Graphics

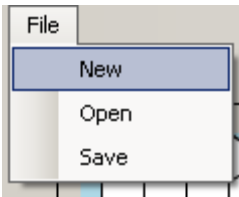
Μέθοδος	Περιγραφή
DrawImage	Σχεδιάζει μέρος ή ολόκληρο κομμάτι μιας εικόνας στην σχεδιαστική επιφάνεια.
DrawLine	Σχεδιάζει μια ευθεία γραμμή.
DrawPath	Σχεδιάζει ένα μονοπάτι από σημεία που πρέπει να δημιουργηθούν στο αντικείμενο Graphics Path.
DrawPie	Σχεδιάζει ένα σχήμα πίτας (pie shape).
DrawPolygon	Σχεδιάζει ένα πολύγωνο το οποίο εκφράζεται ως ένας πίνακας από σημεία.
DrawRectangle	Σχεδιάζει ένα ορθωγόνιο.
DrawString	Σχεδιάζει κείμενο ενός καθορισμένου φόντου στην επιφάνεια εργασίας.
FillClosedCurve	Χρωματίζει μια κλειστή καμπύλη.
FillEllipse	Χρωματίζει μια έλλειψη.
FillPath	Χρωματίζει το μονοπάτι που καθορίζει το αντικείμενο GraphicsPath.
FillPie	Χρωματίζει το περιεχόμενο μιας πίτας χρησιμοποιώντας το αντικείμενο Brush (βούρτσα).
FillPolygon	Χρωματίζει το περιεχόμενο ενός πολυγώνου χρησιμοποιώντας το αντικείμενο Brush.
FillRectangle	Χρωματίζει το περιεχόμενο ενός ορθογώνιου χρησιμοποιώντας το αντικείμενο Brush.
FillRegion	Χρωματίζει το περιεχόμενο μια περιοχής που καθορίζεται από το αντικείμενο Region

Οι περισσότερες συναρτήσεις κάνουν χρήση του αντικειμένου Pen ή Brush. Το Pen χρησιμοποιείται για να σχεδιασμό του περιγράμματος ενός σχήματος ενώ το Brush για το χρωματισμό του.

Εγχειρίδιο χρήσης

Στο σημείο αυτό παρουσιάζονται οι λειτουργίες του προσομοιωτή. Δίνονται οδηγίες για το πως μπορεί κάποιος να εκτελέσει λειτουργίες όπως άνοιγμα αρχείου, αποθήκευση αρχείου, τοποθέτηση θεμελιωδών συστατικών καθώς και πως γίνεται η δημιουργία συστατικών καθορισμένων από το χρήστη.

Νέο Αρχείο

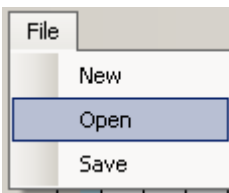


Σχήμα 1 Επιλογή Νέου Αρχείου.

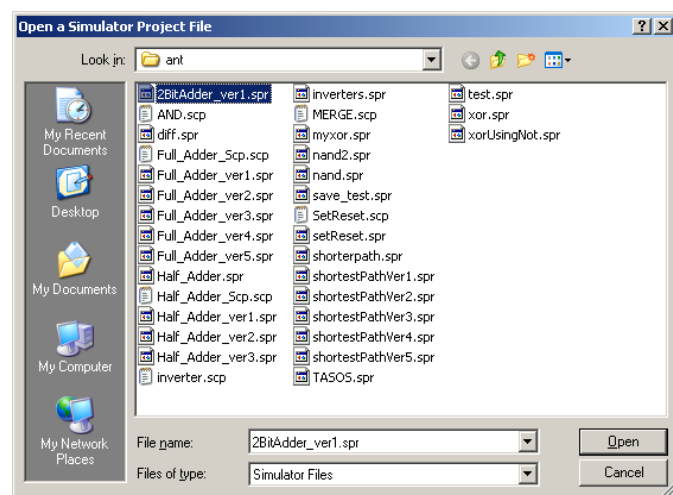
Η επιλογή της λειτουργίας «Νέο Αρχείο» (New) έχει ως αποτέλεσμα το καθαρισμό της επιφάνειας σχεδίασης.

Άνοιγμα αρχείου

Η επιλογή της λειτουργίας «Άνοιγμα Αρχείου» (Open) έχει ως αποτέλεσμα την εμφάνιση του πλαισίου διάλογου επιλογής ανοίγματος αρχείου (σχήμα 3). Ο χρήστης επιλέγει ένα αρχείο με επέκταση srr και αφού πατήσει Open στο διάλογο ο προσομοιωτής εμφανίζει το σωσμένο κύκλωμα στην επιφάνεια εργασίας.



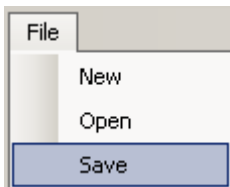
Σχήμα 2 Επιλογή άνοιγμα αρχείου.



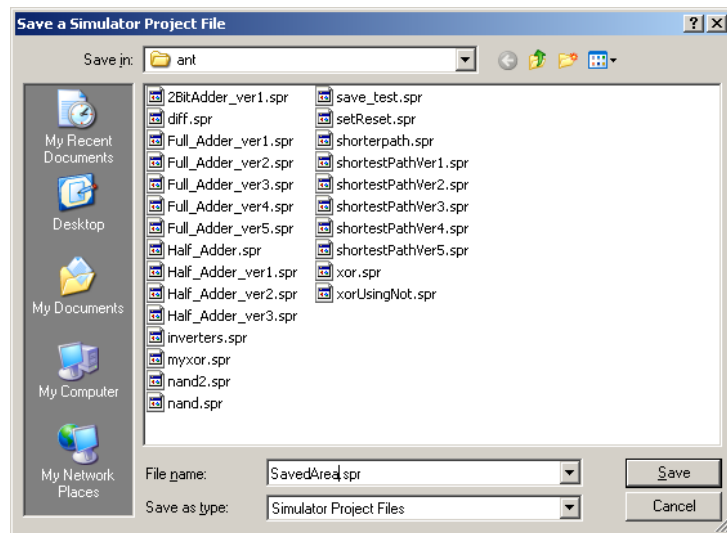
Σχήμα 3 Πλαίσιο διαλόγου επιλογής ανοίγματος αρχείου.

Αποθήκευση αρχείου

Η επιλογή της λειτουργίας «Αποθήκευση αρχείου» (Save) έχει ως αποτέλεσμα την εμφάνιση του πλαισίου διάλογου επιλογής σωσίματος αρχείου (σχήμα 4). Ο χρήστης πληκτρολογεί το όνομα του αρχείου που επιθυμεί και αυτό σώζεται στο δίσκο με επέκταση spr.



Σχήμα 4 Επιλογή Σώσιμο Αρχείου.

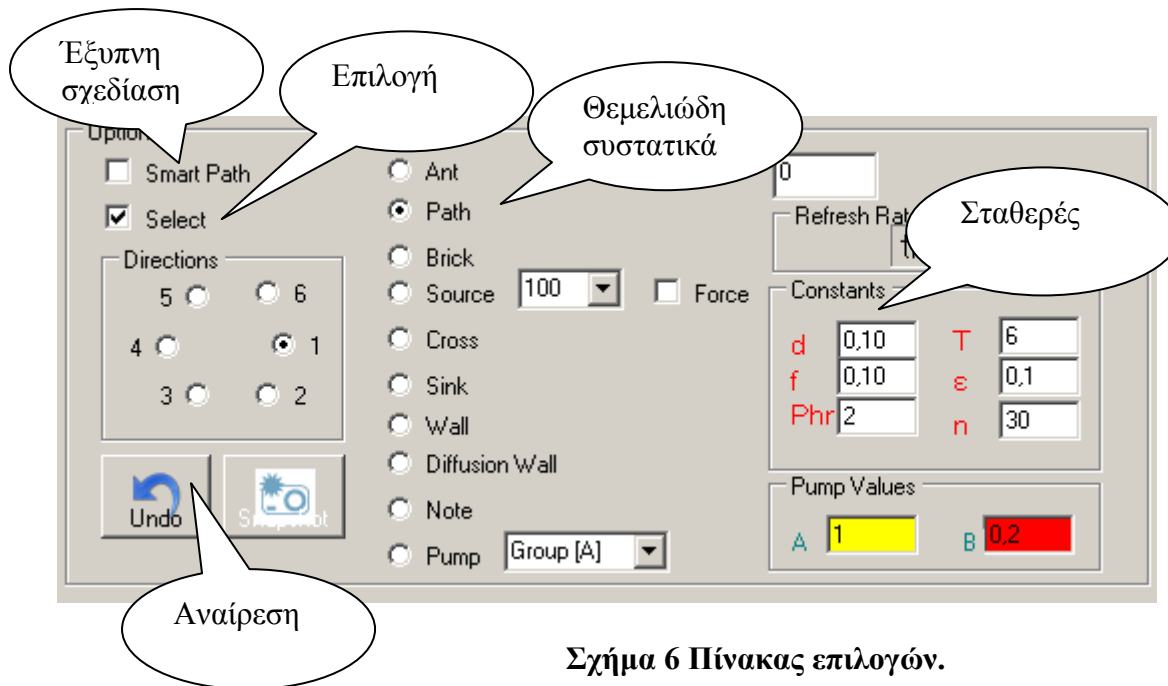


Σχήμα 5 Πλαίσιο διαλόγου αποθήκευσης αρχείου.

Πίνακας Επιλογών

Ο πίνακας επιλογών παρέχει τις εξής δυνατότητες:

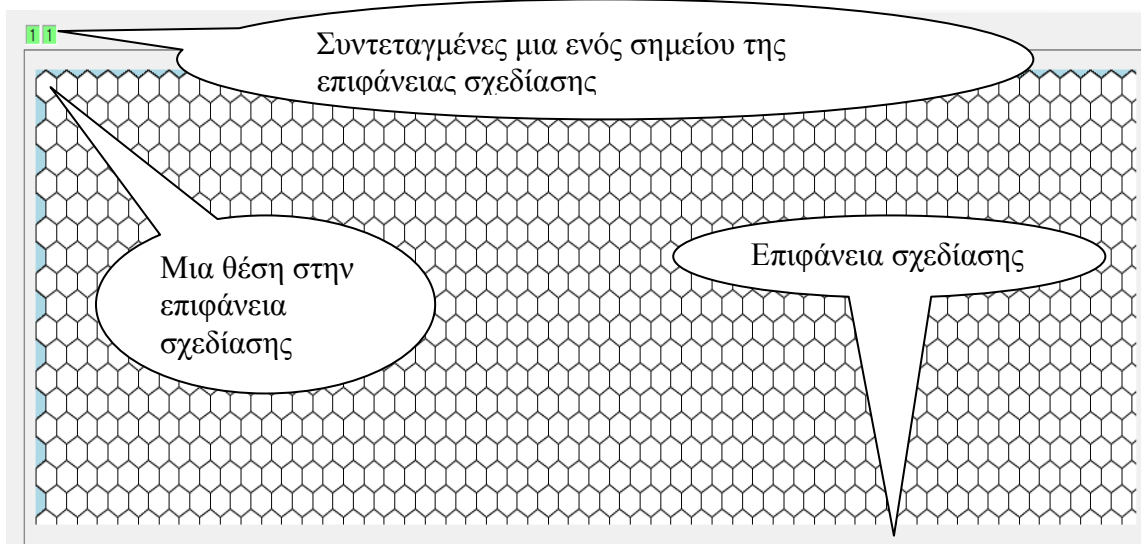
- Τοποθέτηση θεμελιωδών συστατικών στην επιφάνεια σχεδίασης.
- Ενεργοποίηση της λειτουργίας έξυπνης σχεδίασης μονοπατιών.
- Ενεργοποίηση της λειτουργίας επιλογής.
- Αναίρεσης.
- Αντίγραφο της επιφάνειας εργασίας.
- Ανάθεση τιμών σε σταθερές.



Σχήμα 6 Πίνακας επιλογών.

Τοποθέτηση θεμελιωδών συστατικών στην επιφάνεια σχεδίασης

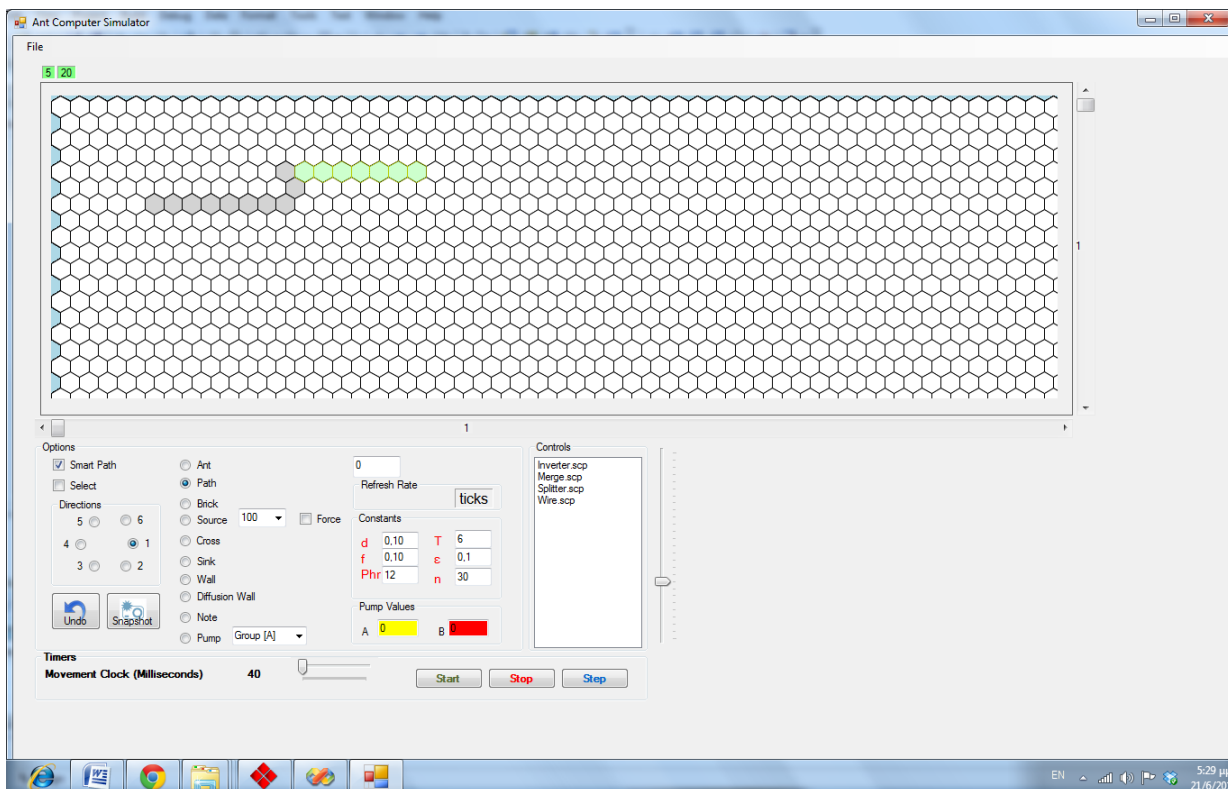
Ο χρήστης επιλέγει ένα από τα θεμελιώδη συστατικά κάνοντας κλικ σε ένα από αυτά και κατόπιν επιλέγει τη θέση που θέλει να το τοποθετήσει πάνω στην επιφάνεια εργασίας κάνοντας κλικ σε αυτή.



Σχήμα 7 Η επιφάνεια σχεδίασης.

Ενεργοποίηση της λειτουργίας έξυπνης σχεδίασης μονοπατιών.

Στη λειτουργία αυτή, ο χρήστης επιλέγει ένα σημείο της επιφάνειας σχεδίασης στην οθόνη πατώντας αριστερό κλικ σε μια θέση. Ο χρήστης μετακινεί το ποντίκι σε ένα σημείο στην επιφάνεια σχεδίασης και αν μπορεί να γίνει αυτή η σύνδεση χρωματίζεται όλη η διαδρομή μέχρι το προαναφερθέν σημείο. Αν ο χρήστης θέλει να κατοχωρήσει αυτή την διαδρομή ως path τότε κάνει αριστερό κλικ στο σημείο αυτό.



Σχήμα 7 Λειτουργία έξυπνης σχεδίασης.

Λειτουργία αναίρεσης

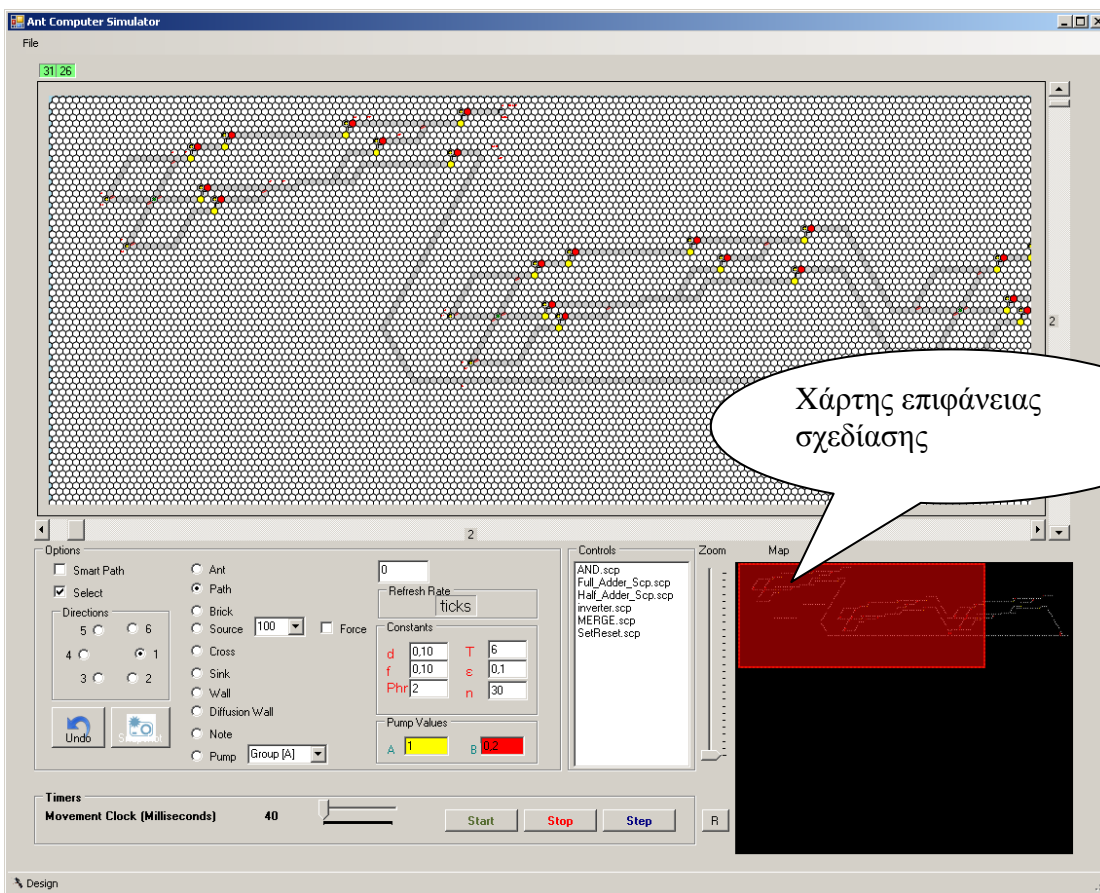
Η αναίρεση ενεργοποιείται με το πάτημα του κουμπιού αναίρεσης και έχει ως αποτέλεσμα την ακύρωση της τελευταίας τοποθέτησης ενός θεμελιώδους συστατικού.

Λειτουργία αντιγράφου της επιφάνειας εργασίας

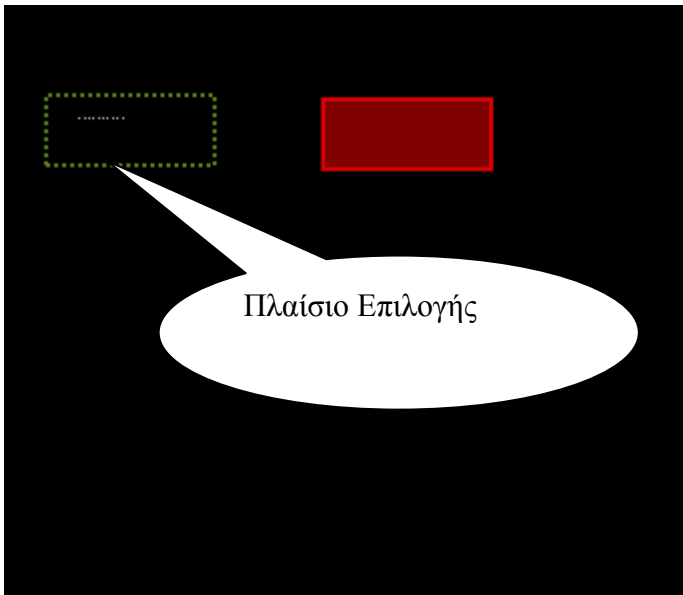
Στην λειτουργία αυτή ο χρήστης πατώντας το κουμπί snapshot δημιουργεί ένα αντίγραφο της παρούσας κατάστασης της επιφάνειας εργασίας, σε αρχείο εικόνας στο δίσκο.

Χάρτης Επιφάνειας σχεδίασης

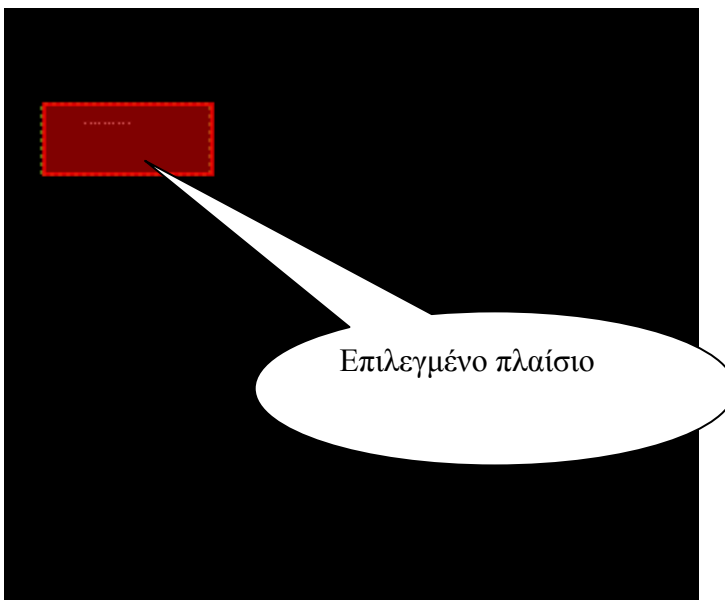
Ο χάρτης είναι μια μικρογραφία του πίνακα εξαγώνων που υπάρχουν στην μνήμη. Ο χρήστης μπορεί να σείρει το πλαίσιο επιλογής (άδειο πλαίσιο με διακεκομμένη γραμμή) με το ποντίκι στην επιφάνεια του χάρτη και να επιλέξει μετάβαση κάνοντας κλικ στο σημείο αυτό πάνω στην επιφάνεια εργασίας. Τότε ένα κόκκινο πλαίσιο εμφανίζεται το οποίο οριοθετεί την επιφάνεια σχεδίασης πάνω στο χάρτη.



Σχήμα 8 Χάρτης επιφάνειας σχεδίασης.



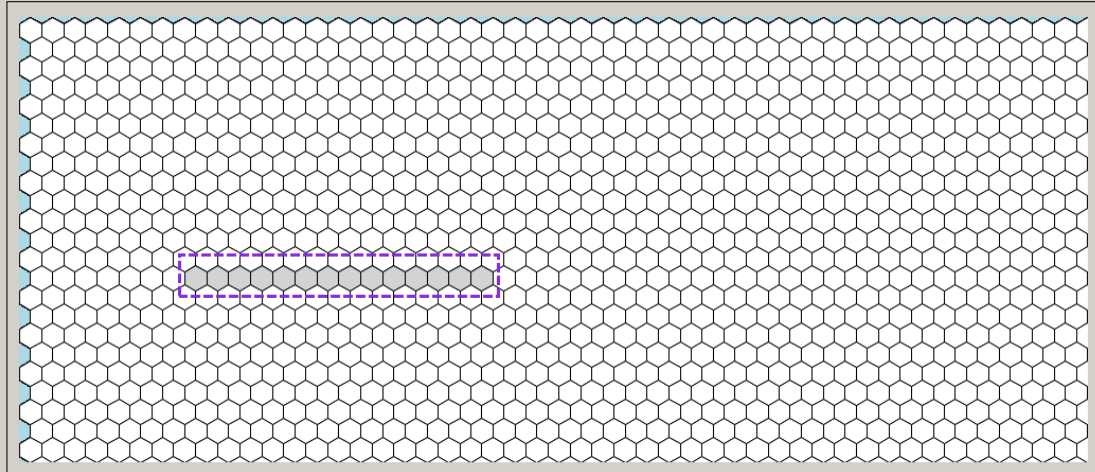
Σχήμα 9 Πλαίσιο επιλογής στο χάρτη επιφάνειας σχεδίασης.



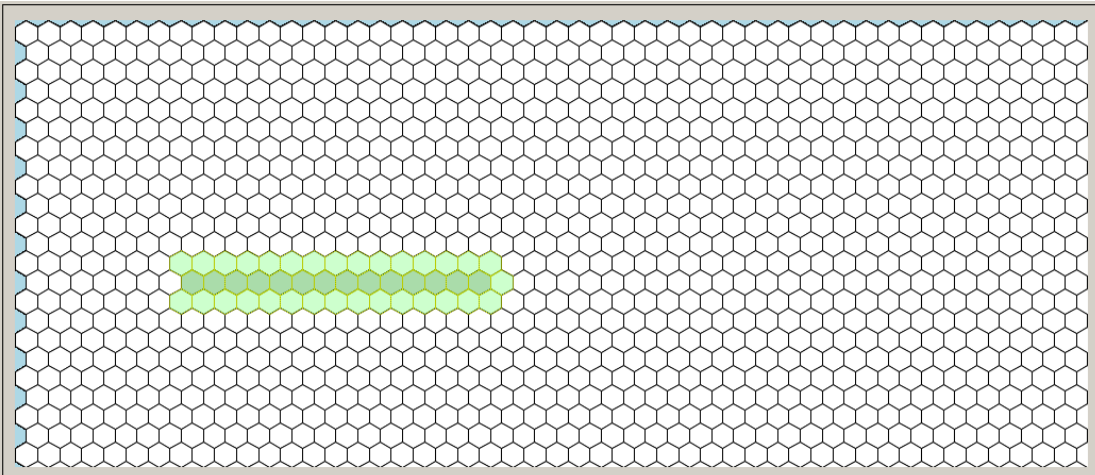
Σχήμα 10 Επιλεγμένο πλαίσιο στο χάρτη επιφάνειας σχεδίασης.

Λειτουργίες επικόλλησης, αποκοπής, διαγραφής

Οι λειτουργίες αυτές προϋποθέτουν την επιλογή της λειτουργίας «επιλογή». Έχοντας ο χρήστης επιλέξει select μπορεί να σύρει το ποντίκι έχοντας πατημένο το αριστερό κουμπί του πάνω στην επιφάνεια σχεδίασης επιλέγοντας έτσι μια επιφάνεια της επιφάνειας σχεδιασμού όπως φαίνεται στο σχήμα 10.



Σχήμα 10 Επιλογή ενός μέρους της επιφάνειας σχεδίασης.

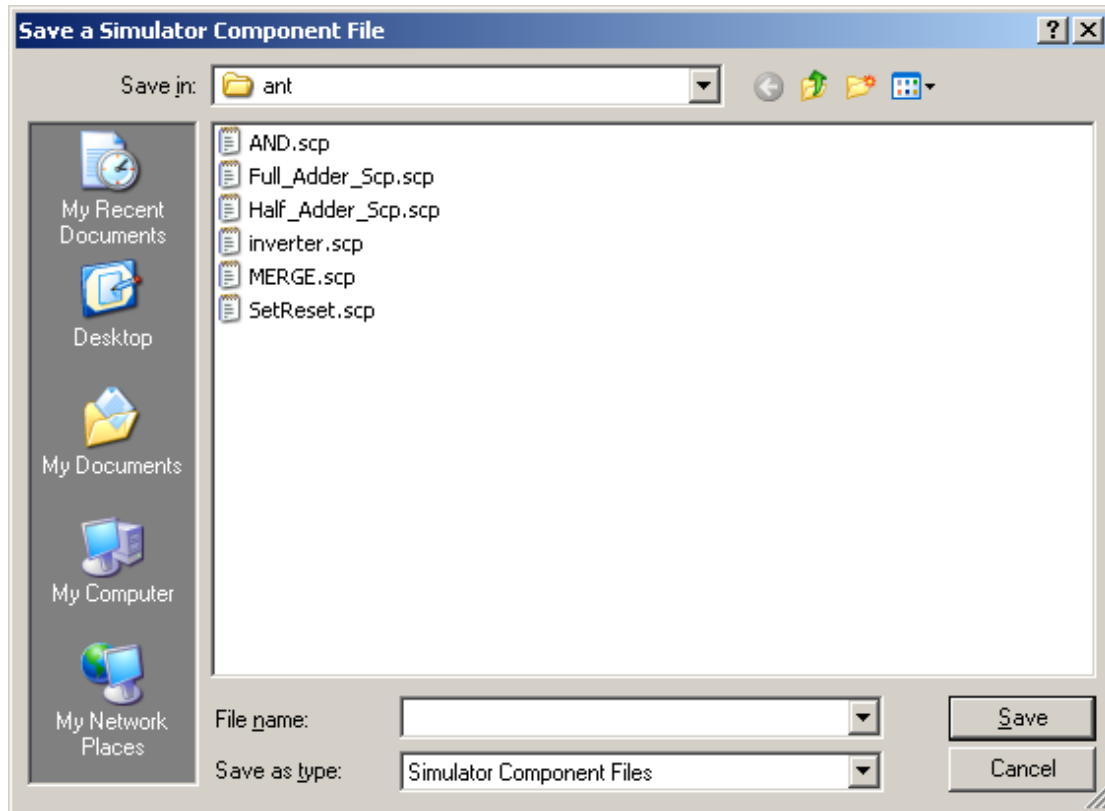


Σχήμα 11 Επιλεγμένο μέρος της επιφάνειας σχεδίασης.

Ο χρήσης αφού έχει κάνει επιλογή επιφάνειας μπορεί πατώντας δεξί κλικ σε ένα σημείο με συντεταγμένες (x,y) πάνω στην επιφάνεια σχεδίασης να:

- Επιλέξει Delete και να διαγράψει τα περιεχόμενα της επιλεγμένης επιφάνειας σχεδίασης.
- Επιλέξει Paste και να αντιγράψει την επιλεγμένη επιφάνεια στο σημείο με συντεταγμένες (x,y) .
- Επιλέξει Cut και να αντιγράψει την επιλεγμένη επιφάνεια στο σημείο με συντεταγμένες (x,y) και η επιλεγμένη επιφάνεια θα διαγραφεί.

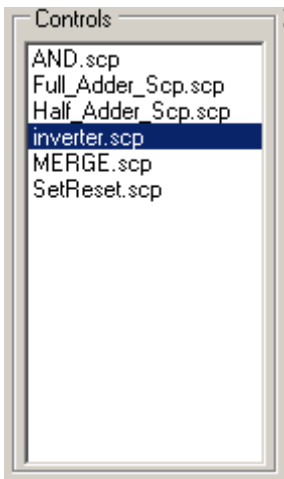
- Επιλέξει Save component και το σύστημα να του εμφανίζει το πλαίσιο διαλόγου του σχήματος 12 όπου ο χρήστης μπορεί να αποθηκεύσει την επιλεγμένη επιφάνεια ως συστατικό.



Σχήμα 12 Αποθήκευση συστατικού.

Τοποθέτηση συστατικού στην επιφάνεια σχεδίασης

Η τοποθέτηση συστατικού στην επιφάνεια σχεδίασης γίνεται με τη λειτουργία “drag & drop” από τη λίστα των συστατικών καθορισμένων από το χρήστη (control) (σχήμα 13). Ο χρήστης κάνει κλικ στο συστατικό που θέλει να τοποθετήσει στην επιφάνεια σχεδιασμού και σύρει το ποντίκι στο σημείο όπου επιθυμεί να γίνει η τοποθέτηση. Ελευθερώνοντας το αριστερό κουμπί του ποντικιού το συστατικό τοποθετείται στο σημείο που ο χρήστης ελευθέρωσε το αριστερό κουμπί του ποντικιού πάνω στην επιφάνεια σχεδίασης.



Σχήμα 13 Συστατικά καθορισμένα από το χρήστη.

Λειτουργία προσομοίωσης

Επιλέγοντας το start, το πρόγραμμα λειτουργεί σε κατάσταση προσομοίωσης. Ο χρήστης μπορεί να διακόψει την προσομοίωση πατώντας stop ή να τη συνεχίσει ανά βήμα πατώντας το step κάθε φορά. Ο ρυθμός της χρονικής στιγμής κάθε βήματος καθορίζεται από την ένδειξη του slider και εκφράζεται σε milliseconds. Κατά τη διάρκεια της προσομοίωσης ο χρήστης μπορεί να κάνει κλικ στις πηγές και να διακόψει την ροή μυρμηγκιών στο σύστημα κάνοντας την αντλία να δουλεύει σαν διακόπτης.



Σχήμα 14 Πλαίσιο προσομοίωσης.