

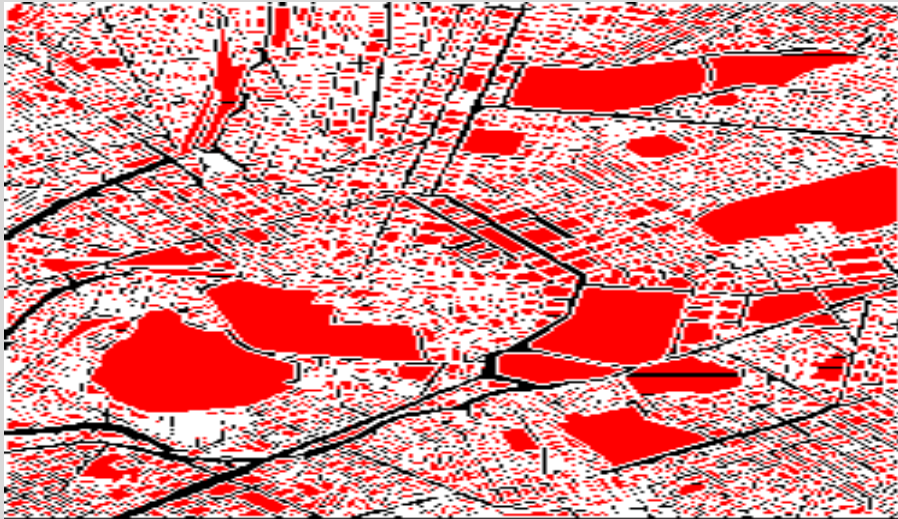
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σπουδάστρια:

Μάρθα ΜΗΤΡΟΠΟΥΛΟΥ

Θέμα:

Αναπαραγωγή **Vector Αρχείων** σε **GIF FORMAT**



Εισηγητής :

Καθηγητής
Δρ Σ.Κ Λεβέντης

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	2
ΕΙΣΑΓΩΓΗ.....	3
ΚΕΦΑΛΑΙΟ 1	10
1.1 Περιγραφή Vector αρχείων.....	10
1.2 Υλοποίηση Normalized.	12
1.3 Ανάπτυξη του εργαλείου Παραγωγής γεωγραφικών χαρτών.	13
1.4 ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ	14
1.5 ΕΠΕΞΗΓΗΣΗ ΒΑΣΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	17
1.6 Εισαγωγή του προγράμματος παραγωγής γεωγραφικών χαρτών.	19
1.6 ΚΡΙΤΙΚΗ ΕΡΓΑΛΕΙΟΥ	27
ΠΛΕΟΝΕΚΤΗΜΑΤΑ	27
ΜΕΙΟΝΕΚΤΗΜΑΤΑ.....	27
ΚΕΦΑΛΑΙΟ 2	28
Εισαγωγή	28
2.1 Ορισμός CGI SCRIPT	28
2.2 Τεχνική ISMAP.	29
2.3 ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ	30
2.4 ΕΠΕΞΗΓΗΣΗ ΒΑΣΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΤΟΥ CGI ΠΡΟΓΡΑΜΜΑΤΟΣ Zoom0.....	33
2.5 ΕΙΣΑΓΩΓΗ ΤΟΥ CGI ΠΡΟΓΡΑΜΜΑΤΟΣ Zoom0.....	35
2.6 ΕΠΕΚΤΑΣΙΜΟΤΗΤΑ.....	45
ΚΕΦΑΛΑΙΟ 3	46
3.1 ΕΙΚΟΝΕΣ ΚΑΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ	46
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	52

ΠΡΟΛΟΓΟΣ

Σκοπός της πτυχιακής εργασίας είναι η διερεύνηση τρόπου γραφικής απεικόνισης διανυσματικών αρχείων (Vector Files) με τυποποίηση GIF (Graphics Interchange Format).

Στόχος της εργασίας είναι η δημιουργία γεωγραφικών χαρτών, με την ανάγνωση οποιουδήποτε διανυσματικού αρχείου, προσδοκώντας στην δημοσιοποίηση των χαρτών στο διαδίκτυο, με σκοπό την ανάκτηση πολύτιμων πληροφοριών όσον αφορά τα γεωγραφικά διαμερίσματα της γης καθώς επίσης και περισσότερο εξειδικευμένες πληροφορίες, όπως για παράδειγμα σχολεία, φαρμακεία, νοσοκομεία, οδούς κ.λ.π. Το τελευταίο επιτυγχάνεται με την εξέλιξη του προγράμματος βάση της τεχνικής GIS (Geographic Information Systems).

Έχοντας ως υπόβαθρο προηγούμενες μελέτες σχετικά με την αναπαραγωγή χαρτών από διανυσματικά αρχεία οι οποίες δεν κατέληξαν στο επιθυμητό αποτέλεσμα και λαμβανομένου υπόψη ότι η γραφική απεικόνιση θα εκτελεστεί σε διαδικτυακό περιβάλλον (Internet) επιλέχτηκε η μορφή των χαρτών να είναι σε GIF format. Είναι γνωστό ότι τα αρχεία GIF είναι τα πλέον διαδεδομένα στην απεικόνιση εικόνων στο διαδίκτυο γιατί το μόνο που απαιτούν για την απεικόνισή τους είναι ένας οποιοσδήποτε Browser.

Η γραφική απεικόνιση χαρτών από διανυσματικά αρχεία πραγματοποιείται σε δύο φάσεις:

- Στην πρώτη φάση επιτυγχάνεται η δημιουργία εργαλείου, με το οποίο παράγουμε τον κεντρικό χάρτη της εφαρμογής μας.
- Στην δεύτερη φάση πραγματοποιείται η ενέργεια μεγέθυνσης συγκεκριμένου σημείου του χάρτη μετά από επιλογή του χρήστη. Η ενέργεια αυτή προγραμματίζεται βάση του μοντέλου Πελάτη / Εξυπηρετητή (Client - Server).

Τέλος θα θέλαμε να αναφέρουμε ότι το εργαλείο παραγωγής των γεωγραφικών χαρτών καθώς επίσης και η εφαρμογή μεταξύ Client -Server, πραγματοποιήθηκαν σε γλώσσα προγραμματισμού C. Η επιλογή της γλώσσας έγινε γνωρίζοντας ότι αυτή είναι συμβατή με τα συστήματα Unix του εργαστηρίου Περιφερειακών Μονάδων και Δικτύων (PeLAB).

ΕΙΣΑΓΩΓΗ

Η μελέτη που παρουσιάζεται αποτελεί τμήμα ενός project που σκοπό έχει την δημιουργία ενός γεωγραφικού οδηγού με απώτερο σκοπό την ανάκτηση χρήσιμων πληροφοριών (σχολείων, Εκπαιδευτικών Ιδρυμάτων, Νοσοκομείων, σταθμών ΗΣΑΠ, Μετρό κ.τ.λ) μέσω της τεχνικής G.I.S.

Κατά την ανάπτυξη του project, η οποία έχει ξεκινήσει εδώ και μερικά χρόνια βάση πτυχιακών μελετών από συναδέλφους του ΤΕΙ ΠΕΙΡΑΙΑ, έχουν επιτευχθεί τα εξής:

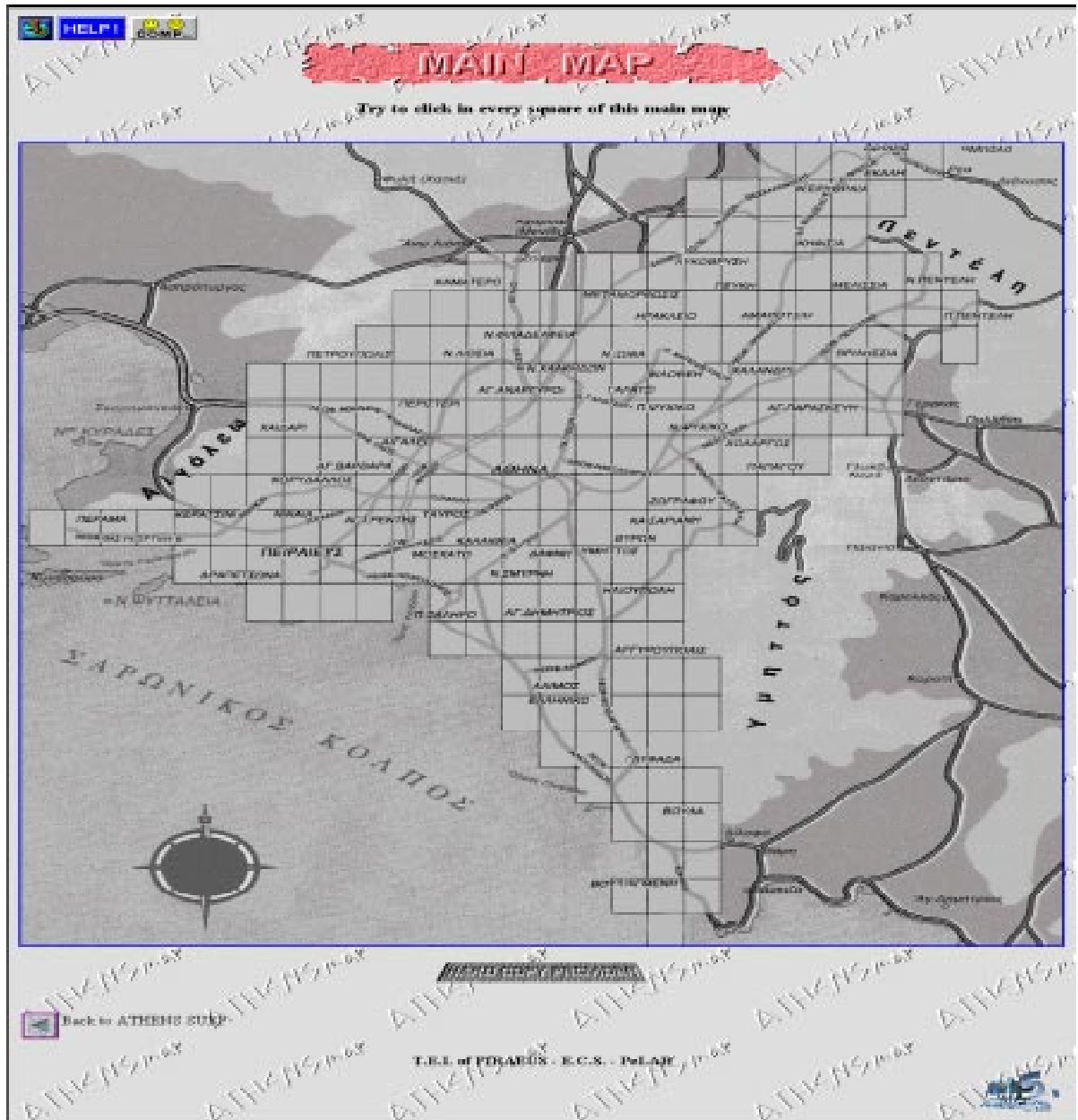
- Δημιουργία Vector (διανυσματικών) αρχείων απ' όπου αντλούνται πληροφορίες για τα οικοδομικά τετράγωνα ενός γεωγραφικού χάρτη.
- Απεικόνιση των διανυσματικών αρχείων στο διαδίκτυο. Η μελέτη αυτή κατάφερε να παρουσιάσει τους χάρτες στο δίκτυο, αλλά έκανε δύσκολη την προσπέλασή τους από τους χρήστες λόγω του ότι θα έπρεπε να κατεβάσουν ανάλογο plug-in.
- Μέσο της μελέτης G.I.S. SERVICES όπως φαίνεται στο σχήμα 1 έγινε προσπάθεια για την αναπαράσταση του γεωγραφικού χάρτη της Αττικής στο διαδίκτυο .



Σχήμα 1

Η μελέτη αυτή επιτυγχάνει τα εξής :

1. Πλοήγηση πάνω στον χάρτη. Από τον κεντρικό χάρτη (σχήμα 2), ο οποίος παρουσιάζει την ευρύτερη περιοχή των Αθηνών κατατημένη σε μικρότερες περιοχές, ο χρήστης μπορεί να επιλέξει την περιοχή που επιθυμεί με το απλό πάτημα του ποντικιού.



Σχήμα 2

2. Η μεγέθυνση της περιοχής που επέλεξε ο χρήστης εμφανίζεται στο σχήμα 3. Μέσα από τον χάρτη αυτό ο χρήστης μπορεί να κινηθεί προς οποιαδήποτε κατεύθυνση είτε χρησιμοποιώντας τα βελάκια που περιβάλλουν το χάρτη είτε απλά κάνοντας κλικ σε κάποιο σημείο της μικρογραφίας στην οποία μια κόκκινη κουκίδα αντιπροσωπεύει την τρέχουσα θέση.

Η παρούσα μελέτη έρχεται να διορθώσει τα προβλήματα που παρουσιάστηκαν κατά την ανάπτυξη του project, αναπαράγοντας τα Vector αρχεία σε Gif format. Με τον τρόπο αυτό καταφέρνουμε :

1. Όσο το δυνατόν πιο φιλικό περιβάλλον για τον χρήστη, αφού το μόνο που απαιτείται για την πλοήγηση αυτού πάνω στους γεωγραφικούς χάρτες είναι ένας οποιασδήποτε Browser.
2. Οι χάρτες οι οποίοι δημιουργούνται δυναμικά σε κάθε κλήση του χρήστη, μεγέθυνση - σμίκρυνση, πετυχαίνουν την μέγιστη δυνατή ανάλυση λόγω του ότι οι εικόνες ξαναδημιουργούνται.
3. Το πιο βασικό στοιχείο για το οποίο έγινε αυτή η μελέτη είναι το γεγονός του ότι το υλικό που θα παρουσιαστεί στο Internet θα αποτελεί πνευματική ιδιοκτησία του ΤΕΙ ΠΕΙΡΑΙΑ με αποτέλεσμα την άμεση προβολή των χαρτών στο διαδίκτυο.

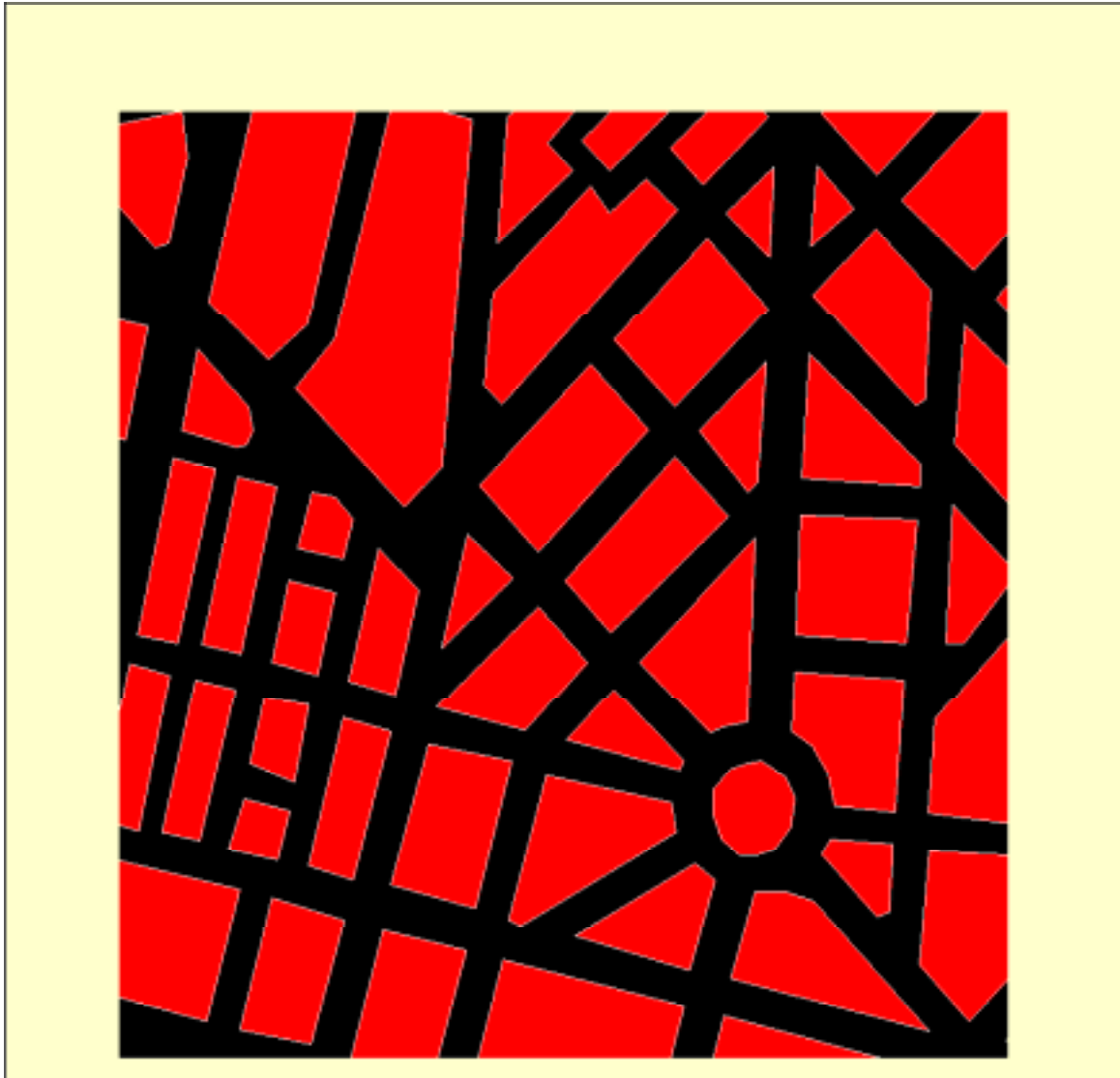
Μέσο της παρούσας μελέτης οι χάρτες της ευρύτερης περιοχής των Αθηνών παρουσιάζονται μέσα από τρία επίπεδα μεγέθυνσης . Στο σχήμα 4 είναι επιλεγμένο το πρώτο επίπεδο.



Σχήμα 4

Στο επίπεδο αυτό ο χρήστης επιλέγει την περιοχή που επιθυμεί και με το πάτημα του ποντικιού γίνεται zoom στην περιοχή που επέλεξε. Στο δεύτερο και τρίτο επίπεδο αντίστοιχα πραγματοποιείται μεγαλύτερη μεγέθυνση με αποτέλεσμα ο χρήστης να δει με μεγαλύτερη ευκρίνεια τις περιοχές που ο ίδιος επέλεξε.

Στις εικόνες που ακολουθούν, (σχήματα 5, 6 και 7), παρουσιάζεται η μεγέθυνση της περιοχής που επέλεξε ο χρήστης, ξεκινώντας από την μικρότερη δυνατή μεγέθυνση (επίπεδο 1) έως την μέγιστη δυνατή (επίπεδο 3).



Σχήμα 5



Σχήμα 6



Σχήμα 7

Στα κεφάλαια που ακολουθούν περιγράφονται αναλυτικά οι διαδικασίες με τις οποίες από διανυσματικά αρχεία καταφέρνουμε να παράγουμε γεωγραφικούς χάρτες σε gif format καθώς επίσης και οι διαδικασίες με τις οποίες επιτυγχάνουμε την πλοήγηση και μεγέθυνση των οικοδομικών τετραγώνων ενός γεωγραφικού χάρτη.

ΚΕΦΑΛΑΙΟ 1

1.1 Περιγραφή Vector αρχείων.

Το εργαλείο που δημιουργήσαμε στηρίζεται στην ανάγνωση ενός vector αρχείου και συγκεκριμένα του Center.az1. Το αρχείο AZ1 είναι ένα διανυσματικό αρχείο που περιέχει πληροφορίες για την απεικόνιση των οικοδομικών τετραγώνων του χάρτη. Τα οικοδομικά τετράγωνα του χάρτη δημιουργούνται σχεδιάζοντας πολύγωνα στην οθόνη. Στο τμήμα του αρχείου που ακολουθεί φαίνονται οι ταυτότητες των πολυγώνων, το πλήθος των γωνιών τους, οι συντεταγμένες της κάθε γωνίας, και άλλες πληροφορίες, των οποίων η δομή εξηγείται λεπτομερώς παρακάτω.

I2V v1.1

0, 0, 4260, 6270

0, 0

3064

0

50 50 50 50 50 50 50 50

0, 0, 0, 0, 0, 1, 158, 275, 14, 0, 210 0, 210 68, 275 158, 59 130, 48 99, 120 79, 110 109, 38

62, 18 32, 86 13, 77 41, 10 22, 1 0, 1

1, 0, 0, 0, 139, 0, 185, 37, 4, 148, 0 139, 22 171, 37 185, 1

2, 0, 0, 0, 190, 0, 249, 60, 4, 210, 0 190, 45 220, 60 249, 1

3, 0, 0, 0, 242, 0, 317, 74, 4, 275, 0 242, 71 295, 74 317, 23

4, 0, 0, 0, 0, 241, 87, 365, 4, 0, 241 87, 323 42, 365 0, 322

5, 0, 0, 0, 193, 123, 266, 184, 4, 216, 123 266, 123 237, 184 193, 178

6, 0, 0, 0, 123, 203, 225, 328, 4, 181, 203 225, 206 167, 328 123, 323

7, 0, 0, 0, 262, 124, 346, 191, 4, 291, 125 346, 124 315, 191 262, 186

8, 0, 0, 0, 315, 32, 382, 75, 4, 337, 32 382, 54 374, 75 315, 75

9, 0, 0, 0, 347, 0, 412, 36, 4, 351, 0 347, 13 394, 36 412, 0

10, 0, 0, 0, 416, 28, 432, 46, 3, 425, 28 416, 45 432, 46

11, 0, 0, 0, 227, 66, 443, 103, 7, 231, 93 371, 92 396, 90 407, 66 435, 66 443, 101 227, 103

12, 0, 0, 0, 346, 125, 425, 201, 5, 374, 126 425, 125 390, 201 354, 196 346, 183

13, 0, 0, 0, 230, 209, 301, 262, 5, 251, 209 293, 213 301, 223 281, 262 230, 256

14, 0, 0, 0, 191, 281, 269, 339, 4, 218, 281 269, 287 244, 339 191, 331

15, 0, 0, 0, 271, 290, 341, 347, 5, 292, 290 333, 294 341, 310 324, 347 271, 342

16, 0, 0, 0, 306, 218, 375, 272, 5, 326, 218 365, 220 375, 238 358, 272 306, 265

17, 0, 0, 0, 417, 145, 459, 208, 3, 446, 145 417, 203 459, 208

18, 0, 0, 0, 459, 58, 475, 85, 3, 459, 58 475, 61 465, 85

19, 0, 0, 0, 470, 99, 500, 212, 4, 470, 119 479, 99 500, 211 491, 212

20, 0, 0, 0, 496, 236, 537, 271, 4, 496, 236 537, 241 518, 271 505, 268

21, 0, 0, 0, 390, 227, 474, 287, 10, 408, 227 463, 232 474, 280 465, 287 446, 284 445, 269

427, 266 426, 282 398, 278 390, 264

22, 0, 0, 0, 356, 299, 423, 356, 5, 375, 299 423, 307 415, 356 361, 351 356, 338

23, 0, 0, 0, 435, 309, 484, 364, 5, 443, 309 480, 311 484, 328 462, 364 435, 361

24, 0, 0, 0, 447, 0, 503, 41, 4, 447, 0 454, 35 482, 41 503, 0

25, 0, 0, 0, 508,0,608,53, 7, 532,0 508,48 536,53 550,36 559,4 605,15 608,0
 26, 0, 0, 0, 626,0,691,34, 4, 632,0 626,20 686,34 691,0
 27, 0, 0, 0, 714,7,763,48, 4, 717,7 763,16 758,48 714,40
 28, 0, 0, 0, 704,59,759,104, 4, 709,59 759,70 752,104 704,101
 29, 0, 0, 0, 560,33,602,63, 5, 568,40 586,33 602,40 594,63 560,61
 30, 0, 0, 0, 500,82,541,217, 4, 500,82 524,83 541,217 522,217
 31, 0, 0, 0, 545,83,588,197, 3, 545,83 588,88 562,197
 32, 0, 0, 0, 598,42,683,173, 4, 623,42 683,56 669,173 598,163
 33, 0, 0, 0, 580,184,666,234, 5, 592,184 666,194 661,234 580,222 588,204
 34, 0, 0, 0, 535,245,593,300, 4, 535,292 563,245 593,249 585,300
 35, 0, 0, 0, 605,253,655,308, 4, 612,253 655,257 647,308 605,303
 36, 0, 0, 0, 693,126,747,168, 4, 701,126 747,132 738,168 693,163
 37, 0, 0, 0, 683,185,739,242, 4, 690,185 739,194 732,242 683,237
 38, 0, 0, 0, 671,261,731,320, 4, 682,261 731,268 724,320 671,312
 39, 0, 0, 0, 775,22,848,113, 4, 787,22 848,37 839,113 775,107
 40, 0, 0, 0, 753,134,833,255, 4, 771,134 833,139 819,255 753,245
 41, 0, 0, 0, 747,272,815,330, 4, 753,272 815,283 808,330 747,321
 42, 0, 0, 0, 860,39,896,119, 4, 870,39 896,46 885,119 860,114
 43, 0, 0, 0, 790,0,853,14, 3, 790,0 852,14 853,0
 44, 0, 0, 0, 873,0,940,29, 4, 874,0 873,14 935,29 940,0
 45, 0, 0, 0, 905,50,933,123, 4, 916,50 933,54 927,123 905,120
 46, 0, 0, 0, 840,141,883,262, 4, 857,141 883,143 864,262 840,259
 47, 0, 0, 0, 885,145,921,267, 4, 904,145 921,146 912,267 885,267
 48, 0, 0, 0, 829,282,863,339, 4, 838,282 863,284 857,339 829,334
 49, 0, 0, 0, 877,289,908,341, 4, 884,290 908,289 903,341 877,339
 50, 0, 0, 0, 958,0,979,37, 3, 964,0 958,34 979,37

Στην πρώτη σειρά υπάρχει η έκδοση του : π.χ. I2v v1.1. Στην επόμενη σειρά πληροφορούμαστε για την μέγιστη τετμημένη και τεταγμένη των γωνιών όλων των πολυγώνων του αρχείου. Ακολουθεί το πλήθος των πολυγώνων που περιέχονται στο συγκεκριμένο αρχείο, το οποίο συνεχίζεται με τις πληροφορίες των πολυγώνων που είναι ταξινομημένα κατά αύξουσα σειρά. Οι πληροφορίες κάθε γραμμής και κατ' επέκταση κάθε πολυγώνου έχουν φυσικά την ίδια δομή, οπότε ερμηνεύοντας τα στοιχεία μιας γραμμής καλύπτουμε όλο το αρχείο. Για την επεξήγηση των πληροφοριών που δίνονται για κάθε πολύγωνο παίρνουμε την εξής γραμμή :

1, 0, 0, 0, 139,0,185,37, 4, 148,0 139,22 171,37 185,1

Ο πρώτος αριθμός (1) αποτελεί την ταυτότητα του συγκεκριμένου πολυγώνου. Οι επόμενοι τρεις αριθμοί (0) έχουν τοποθετηθεί εκεί από το πρόγραμμα δημιουργίας του αρχείου AZ1 και δεν χρησιμοποιούνται. Στην συνέχεια υπάρχει μια τετράδα αριθμών (139), (0), (185), (37) η οποία ορίζει τη μικρότερη τετμημένη (139) και τεταγμένη (0), καθώς και την μέγιστη τετμημένη (185) και τεταγμένη (37) του εξεταζόμενου πολυγώνου. Ο επόμενος αριθμός (4) μας πληροφορεί για το πλήθος των γωνιών του πολυγώνου. Τέλος ακολουθούν οι συντεταγμένες των γωνιών (148,0),(139,22),(171,37),(185,1).

1.2 Υλοποίηση Normalized.

Κατά την ανάγνωση του vector αρχείου, αναφερθήκαμε σε δύο σημεία τα οποία αποτελούν τις ελάχιστες (0,0) και μέγιστες συντεταγμένες (4260,6270) μέσα στο αρχείο. Από τις συντεταγμένες αυτές καταλαβαίνουμε ότι για την υλοποίηση του κεντρικού μας χάρτη θα έπρεπε να δημιουργήσουμε μια εικόνα που να έχει αριθμό pixel 4260x6270, ώστε κάθε συντεταγμένη του αρχείου να αντιστοιχεί σε ένα pixel.

Το γεγονός αυτό θα οδηγούσε στην δημιουργία μεγάλων εικόνων οι οποίες θα ήταν δύσκολο να εμφανιστούν στην οθόνη, εκτυπωτή και το σημαντικότερο είναι ότι θα ήταν πολύ δύσκολο να δημοσιευτούν στο Internet, λόγω του μεγάλου μεγέθους τους.

Μέσα από το πρόγραμμά μας, για να αποφύγουμε τις μεγάλες εικόνες, πολλαπλασιάζουμε τις συντεταγμένες που διαβάζουμε από το Vector αρχείο με ένα συντελεστή. Ο συντελεστής αυτός προκύπτει από την διαίρεση του πλάτους του canva (Width) με την μέγιστη τιμή κατά X, και από την διαίρεση του ύψους (Height) με την μέγιστη τιμή κατά Y.

Μέσα στο πρόγραμμα εμφανίζεται σε δύο σημεία :

1. Κατά την διάρκεια του ορισμού του.

```
Sk=W;  
Sk1=H;  
  
Sin_x=Sk/max_x;  
Sin_y=Sk1/max_y;
```

2. Κατά τον πολλαπλασιασμό του με τις συντεταγμένες του αρχείου.

```
point[curpoint].x= (int) ((point[curpoint].x)*sin_x);  
point[curpoint].y= (int) ((point[curpoint].y)*sin_y);
```

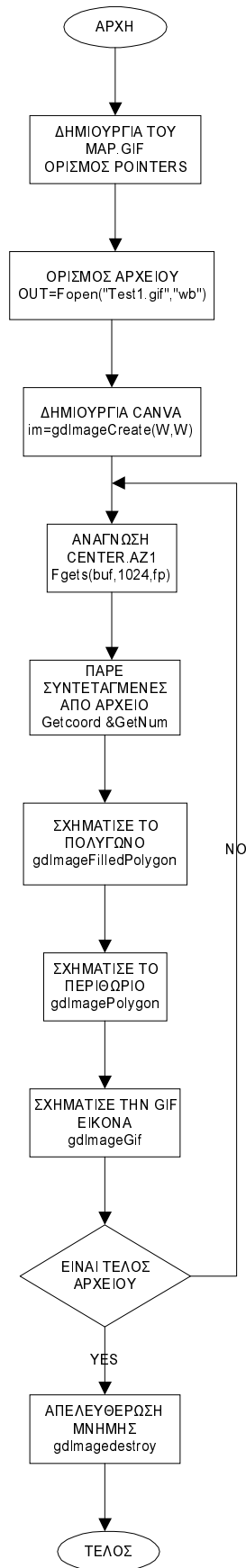
Με τον παραπάνω τρόπο καταφέρνουμε να απεικονίσουμε τις συντεταγμένες του Vector αρχείου σε οποιοδήποτε μέγεθος Canva, πετυχαίνοντας έτσι μικρές εικόνες και με μέγιστη δυνατή ανάλυση τόσο σε επίπεδο οθόνης όσο και σε επίπεδο εκτύπωσης.

1.3 Ανάπτυξη του εργαλείου Παραγωγής γεωγραφικών χαρτών.

Το εργαλείο που δημιουργήσαμε διαβάζει ένα οποιοδήποτε Vector αρχείο (AZ1) και παράγει μια GIF εικόνα. Η υλοποίηση του προγράμματος κατέστη εφικτή κάνοντας χρήση των βιβλιοθηκών γραφικών Gd.h, Gd.c, mtables.c. Οι βιβλιοθήκες αυτές είναι υπεύθυνες για τον σχηματισμό των πολυγώνων, για τον χρωματισμό αυτών και για την αποθήκευσή του σε GIF format, και είναι διαθέσιμες στο διαδίκτυο στο site <http://www.boutel.com>.

1.4 ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ

Για να γίνει κατανοητή η διαδικασία καλό θα ήταν να δούμε μέσα από το λογικό διάγραμμα που ακολουθεί, την πορεία του προγράμματος ως το τελικό αποτέλεσμα.



Το πρόγραμμα ξεκίνα δεσμεύοντας μια περιοχή μνήμης, στην οποία ορίζονται όλοι οι pointers που θα χρησιμοποιηθούν κατά την διάρκεια υλοποίησης του κεντρικού μας χάρτη. Στην συνέχεια ορίζουμε το μέγεθος της εικόνας (canvas) και θέτουμε ότι αυτό θα είναι 900 pixels κατά X και 1350 pixels κατά Y και ότι το χρώμα του θα είναι μαύρο. Μέσα από το πρόγραμμα υπάρχει η δυνατότητα αλλαγής του μέγεθος της εικόνας καθώς επίσης και του χρώματος αυτής. Σαν πρώτο βήμα έχουμε δημιουργήσει ένα μαύρο τετράγωνο με διαστάσεις 900X1350 pls. Το επόμενο βήμα είναι να απεικονίσουμε πάνω στο τετράγωνο που σχηματίσαμε τα πολύγωνα, (τρίγωνα, τετράγωνα, n-γωνα), που θα διαβάσουμε από το αρχείο Center .az1.

Το πρόγραμμα συνεχίζει με την ανάγνωση γραμμή-γραμμή του vector αρχείου center.az1. Διαβάζουμε μία - μία τις συντεταγμένες των κορυφών των πολυγώνων και τις αποθηκεύουμε σε πίνακα. Η αποθήκευση γίνεται αφού πρώτα εφαρμοστεί η τεχνική Normalized που αναφέραμε, ώστε να είναι δυνατή η απεικόνιση των συντεταγμένων των πολυγώνων σε οποιοδήποτε μέγεθος canva .

Το επόμενο βήμα είναι ο σχηματισμός των πολυγώνων από τις συντεταγμένες τους, οι οποίες είναι ήδη αποθηκευμένες. Χαράσσουμε ευθείες γραμμές από την μια κορυφή του πολυγώνου έως την άλλη, τόσες φορές όσες η μεταβλητή Typeofpolygon. (Όπου Typeofpolygon είναι η μεταβλητή που κρατά το πλήθος των κορυφών ενός πολυγώνου). Έτσι το πολύγωνο σχηματίζεται και ταυτόχρονα χρωματίζεται με κόκκινο χρώμα. Η επιλογή του χρώματος ήταν τυχαία και υπάρχει η δυνατότητα αλλαγής του.

Στη συνέχεια παίρνουμε στη διαδικασία δημιουργίας του περιθωρίου του πολυγώνου, η οποία γίνεται με τον ίδιο τρόπο κατά τον σχηματισμό του, μόνο που το χρώμα του επιλέγεται λευκό. Έτσι καταφέραμε να απεικονίσουμε πάνω στον Canva όλα τα πολύγωνα που διαβάσαμε από το αρχείο Center.az1.

Το τελικό στάδιο και το οποίο είναι αυτό που θα μας δώσει την εικόνα μας σε Gif format επιτυγχάνεται αποθηκεύοντας, όλα τα πολύγωνα που σχηματίσαμε πάνω στην οθόνη, στο αρχείο Map.gif. Όλες οι πληροφορίες που ήταν αποθηκευμένες στην περιοχή μνήμης που ορίσαμε αρχικά μεταφέρονται στο αρχείο και με ειδικούς αλγορίθμους συμπίεσης παράγεται το επιθυμητό αποτέλεσμα, ο κεντρικός χάρτης Map.gif.

1.5 ΕΠΕΞΗΓΗΣΗ ΒΑΣΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

GetCoord() && GetNum()

Αναλαμβάνουν την ανάγνωση, επεξεργασία, αποθήκευση των συντεταγμένων των κορυφών των πολυγώνων μέσα από το αρχείο Center.az1.

Κατά την διαδικασία ανάγνωσης είναι υπεύθυνες για τον διαχωρισμό των μη χρήσιμων χαρακτήρων που υπάρχουν μέσα στο αρχείο.

Κατά την διαδικασία της αποθήκευσης των συντεταγμένων των σημείων είναι υπεύθυνες για την υλοποίηση της τεχνικής Normalized.

GdImageSetPixel (gdImagePtr im, int x, int y, int color)

Η συνάρτηση αυτή θέτει το χρώμα ενός Pixel της εικόνας. Η θέση του pixel καθορίζεται από τις συντεταγμένες του x , y και το χρώμα του καθορίζεται από την μεταβλητή color. Η συνάρτηση αυτή θεωρείται ως η πιο βασική κατά την διάρκεια υλοποίησης της εικόνας γιατί όλες οι συναρτήσεις που θα αναφέρουμε πιο κάτω για να υλοποιηθούν κάνουν χρήση της συνάρτησης αυτής.

GdImageLine(gdImagePtr im, int x1, int y1, int x2, int y2, int color)

Η συνάρτηση αυτή χρησιμοποιείται από το πρόγραμμα προκειμένου να σχηματίσει γραμμή από το pixel με συντεταγμένη (x1, y1) έως το pixel με συντεταγμένη (x2, y2). Η συνάρτηση αυτή υλοποιείται κάνοντας χρήση του αλγορίθμου της εξίσωσης της ευθείας. Για να σχηματίσει την ευθεία από το pixel με συντεταγμένη (x1, y1) μέχρι το pixel με συντεταγμένη (x2, y2) υπολογίζει όλα τα ενδιάμεσα pixels και του θέτει χρώμα ίσο με color, μέσω της συνάρτησης gdImageSetPixel.

GdImagePolygon(gdImagePtr im, gdPointPtr p, int n, int c)

Μέσο της συνάρτησης αυτής σχηματίζονται τα πολύγωνα του γεωγραφικού χάρτη. Οι συντεταγμένες των γωνιών των πολυγώνων είναι αποθηκευμένες στον πίνακα gdPointPtr. Η συνάρτηση καλεί την gdImageLine προκειμένου να σχηματίσει γραμμή από τη μια κορυφή του πολυγώνου στην επόμενη κατά σειρά. Το πλήθος των κορυφών του πολυγώνου είναι απαραίτητο και δίνεται στην μεταβλητή n. Επίσης το χρώμα που θα έχει το πολύγωνο καθορίζεται από την μεταβλητή c (color).

GdImage(im,out)

Η συνάρτηση αυτή αποτελεί το τελικό στάδιο του προγράμματος και το πιο ουσιώδες γιατί είναι αυτό που θα παρουσιαστεί στον χρήστη. Μέσο της παραπάνω συνάρτησης γίνεται η παραγωγή της εικόνας, η δημιουργία του κεντρικού χάρτη. Σκοπός της είναι η μεταφορά όλων των πληροφοριών που αφορούν την εικόνα και που είναι αποθηκευμένες στην μνήμη, στο αρχείο Map.gif και το οποίο αποτελεί την παραγωγή του κεντρικού μας χάρτη από το διανυσματικό αρχείο Center.az1.

GdImageDestroy(im)

Σκοπός της παραπάνω συνάρτησης είναι να απελευθερώσει την δεσμευμένη μνήμη του συστήματος, που χρησιμοποιήθηκε καθ' όλη τη διάρκεια δημιουργίας της εικόνας μας.

1.6 Εισαγωγή του προγράμματος παραγωγής γεωγραφικών χαρτών.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>
#include "gd.h"
#include "gd.c"

#define MAXPOINT 50

struct point
{
    int x;
    int y;
};

struct point point[MAXPOINT];

int  typeofpolygon;
int  curpoint;
int  max_x ;
int  max_y ;

char  *GetNum ();

static char  *PTR;

gdPoint points[MAXPOINT];

char temp[8];
char name[8]= {'T','2','V',' ','v','1','!','1'};

float sk,sk1;
int w,h;
float sin_x,sin_y;
```

```
main (ac,av)

int ac;
char *av[];

{
    FILE *fp;
    char buf[1024];
    char filename[10];

    gdImagePtr im;

    FILE *out;

        int black;
        int white;
        int red;
        int counter=1;

if (ac!=2)
{
printf("usage mine :%s filename\n",av[0]);
exit(0);
}

printf("DOSE TO ONOMA THS EIKONAS SE GIF
FORMAT\n(example:map.gif).:");

gets(filename);

fp=fopen(av[1],"r");
out=fopen(filename,"wb");

    if (fp == NULL)
    {
        printf("TO ARXEIO(%s)DEN YPARXEI.\n",av[1]);
        abort();
    }

    w =0;
    h=0;
```

```
printf("DOSTE TO MEGETHOS TOY CANVA KATA X: ");
scanf("%d",&w);

printf("DOSTE TO MEGETHOS TOY CANVA KATA Y: ");
scanf("%d",&h);
```

```
im = gdImageCreate(w,h);
```

```
black = gdImageColorAllocate(im, 0, 0, 0);
```

```
white = gdImageColorAllocate(im, 255, 255, 255);
```

```
red = gdImageColorAllocate(im, 255, 0, 0);
```

```
while (fgets (buf, 1024, fp) != NULL)
{
```

```
    PTR = &buf[0];
```

```
    if(counter==1)
    {
```

```
        CheckTitle();
```

```
    }
```

```
    if(counter==2)
```

```
    {
```

```
        if (GetMaxValues() == 1);
```

```
    sk=w;
```

```
        sk1=h;
```

```
        sin_x=sk/max_x;
```

```
        sin_y=sk1/max_y;
```

```
    }
```

```
    if(counter>6)
    {
        if (strlen (buf) < 8)
            continue;

    if (GetCoord() == 1);

        gdImageFilledPolygon(im, points, typeofpolygon, red);
        gdImagePolygon(im, points, typeofpolygon, white);

    }

        counter++;

    }

fclose (fp);

gdImageGif(im, out);
fclose(out);

gdImageDestroy(im);
printf("TELOS PROGRAMATOS.\n ....");
}
```

GetCoord ()

```
{
    char *tmp;
    int i;
    for (i = 0; *PTR != '\0'; PTR++)
    {
        if (*PTR == ',')
            i++;
        if (i == 8 )
            break;
    }
    curpoint = 0;

    PTR++;

    tmp = GetNum ();
    if (tmp == NULL)
        return (0);

    typeofpolygon = atoi (tmp);

    for (;;)
    {

        if (*PTR == '\0' || *PTR == '\n')
            break;
        PTR++;
        tmp = GetNum ();
        if (tmp == NULL)
            break;
        point[curpoint].x = atoi (tmp);
        point[curpoint].x= (int) ((point[curpoint].x)*sin_x);
        points[curpoint].x=point[curpoint].x;

        if (*PTR == '\0' || *PTR == '\n')
            break;
        PTR++;

        tmp = GetNum ();
        if (tmp == NULL)
            break;

        point[curpoint].y = atoi (tmp);
        point[curpoint].y= (int) ((point[curpoint].y)*sin_y);
```

```

        points[curpoint].y=point[curpoint].y;

        curpoint ++;
    }
    return (1);
}

char *GetNum ()
{

    int idx;
    char ret[200];
    idx = 0;

    while (*PTR != ';' && *PTR != '\n' && *PTR != '\0')
    {
        if ((*PTR == ')')&&*(PTR-1)!=';')
        {
            PTR++;
            continue;
        }

        if ((*PTR == ')')&&*(PTR-1)!=';')
            break;

        ret[idx] = *PTR;

        idx++;
        PTR++;}

    ret[idx] = '\0';
    if (idx == 0)

    return (NULL);
    return (ret);}

```

```
CheckTitle()
{
    int i,flag;

    for(i=0; i<8;i++)
    {
        temp[i]=*PTR;
        PTR++;
    }

    flag=1;

    for(i=0;i<8;i++)
    {
        if(temp[i]!=name[i])
        {
            flag=0;
        }
    }

    if(flag==0)
    {
        printf("TO ARXEIO DEN EINAI AZI.\n TERMATISMOS PROGRAMATOS!\n ");

        abort();
    }

    else

    printf("TO ARXEIO EINAI AZI.\n PERIMENETE GIA THN DHMIOYRGIA TIS
            EIKONAS.\n ");
}
```

```
GetMaxValues()
{

    char *tmp;
    int i;
    for (i = 0; *PTR != '\0'; PTR++)
    {
        if (*PTR == ',')
            i++;
        if (i == 2 )
            break;
    }

    for (;;)
    {

        if (*PTR == '\0' || *PTR == '\n')
            break;
        PTR++;
        tmp = GetNum ();
        if (tmp == NULL)
            break;
        max_x = atoi(tmp);

        if (*PTR == '\0' || *PTR == '\n')
            break;
        PTR++;
        tmp = GetNum ();
        if (tmp == NULL)
            break;
        max_y = atoi(tmp);

    }

    return (1);
}
```

1.6 ΚΡΙΤΙΚΗ ΕΡΓΑΛΕΙΟΥ

ΠΛΕΟΝΕΚΤΗΜΑΤΑ

Παραγωγή εικόνας σε Gif format .

Η επιλογή του αρχείου Gif έγινε γνωρίζοντας ότι το συγκεκριμένο αρχείο είναι πολύ εύκολο να προσπελαστεί από οποιονδήποτε browser, έχοντας ως αποτέλεσμα την άμεση απεικόνισή του στο διαδίκτυο.

Λόγω των αλγορίθμων συμπίεσης που χρησιμοποιεί το αρχείο Gif, οι εικόνες που απεικονίζει έχουν μικρότερο μέγεθος σε σχέση με άλλο τύπο εικόνων (π.χ. bmp, jpg κ.ά.). Η ιδιότητα αυτή συμβάλλει στην γρήγορη προσπέλαση ιστοσελίδων, που περιέχουν εικόνες σε gif format, από το διαδίκτυο.

Μέγιστος χρόνος απόκρισης.

Το εργαλείο χρησιμοποιεί την μνήμη του συστήματος κατά την διάρκεια σχηματισμού της εικόνας. Η ενέργεια αυτή έχει ως αποτέλεσμα την δημιουργία των εικόνων, σε πολύ μικρό χρονικό διάστημα λόγω του ότι τα δεδομένα που απαιτούνται από το σύστημα υπάρχουν συνεχώς στη μνήμη και δεν είναι απαραίτητη η μεταφορά τους από το δίσκο του συστήματος.

Παραμετρικότητα

Υπάρχει η δυνατότητα ελέγχου της μονάδας μέτρησης των εικόνων κατά την εκτύπωσή τους. Από pixels, που έχει οριστεί στο πρόγραμμα, σε dots per inch για την καλύτερη και πιστότερη αναπαράστασή τους σε χαρτί.

Βέλτιστο περιβάλλον εργασίας

Κατά την εξομοίωση του προγράμματος σε περιβάλλον Unix επιτεύχθηκε η βέλτιστη απόδοση της εφαρμογής, επιτυγχάνοντας ανάλυση εικόνας 3000X3000 pixels.

ΜΕΙΟΝΕΚΤΗΜΑΤΑ

Χρήση σε προσωπικό υπολογιστή

Για την χρήση του προγράμματος σε προσωπικό υπολογιστή θα πρέπει ο υπολογιστής αυτός να διαθέτει το λιγότερο 128MB μνήμης για την επίτευξη ανάλυσης εικόνας 700X700 pixels. Σε αντίθετη περίπτωση, απαιτεί μεγάλο pagefile το οποίο καθιστά το πρόγραμμα αργό και με επικίνδυνες επιπτώσεις για το σκληρό δίσκο.

ΚΕΦΑΛΑΙΟ 2

Εισαγωγή

Σκοπός του κεφαλαίου είναι να παρουσιάσει τον τρόπο με τον οποίο οι χάρτες που δημιουργήθηκαν από τα διανυσματικά αρχεία εμφανίζονται στο διαδίκτυο με δυναμικό τρόπο.

Στόχος μας είναι μέσω της παρουσίασης των χαρτών στο διαδίκτυο, να δώσουμε την δυνατότητα στον χρήστη να επιλέξει μια οποιοδήποτε περιοχή πάνω στον χάρτη . Το αποτέλεσμα της ενέργειας αυτής θα είναι η παραγωγή μιας νέας εικόνας (GIF), η οποία θα αποτελεί την μεγέθυνση της περιοχής που επέλεξε ο χρήστης.

Ο χρήστης έχει την δυνατότητα να επιλέξει τρία διαφορετικά επίπεδα μεγέθυνσης . Η μεγέθυνση αυξάνεται όσο αυξάνεται και το επίπεδο. Δηλαδή στο τελευταίο επίπεδο(επίπεδο-τρίτο) πετυχαίνουμε την μέγιστη μεγέθυνση της περιοχής που επιλέξαμε.

Η ενέργεια αυτή πραγματοποιείται μέσα από τρία cgi προγράμματα Zoom0, Zoom1, Zoom2. Στο κεφάλαιο αυτό θα γίνει αναφορά μόνο για το πρώτο cgi Zoom0 μέσω του οποίου πετυχαίνουμε το πρώτο επίπεδο μεγέθυνσης.

2.1 Ορισμός CGI SCRIPT

Οι World Wide Web browsers απεικονίζουν έγγραφα γραμμένα σε γλώσσα HTML (HyperText Markup Language). Επίσης μπορούν να απεικονίζουν HTML forms οι οποίες επιτρέπουν την εισαγωγή δεδομένων από τους χρήστες. Οι browsers χρησιμοποιώντας αυτές τις forms μπορούν να συλλέγουν καθώς και να απεικονίζουν πληροφορίες.

Όταν η πληροφορία συλλέγεται από τον browser, στέλνεται σ'ένα HTTP server (Hyper Text Transfer Protocol) ο οποίος καθορίζεται από την HTML φόρμα. Αυτός ο server ξεκινάει ένα πρόγραμμα, το οποίο περιγράφεται επίσης στην HTML φόρμα και επεξεργάζεται την πληροφορία. Τέτοια προγράμματα είναι γνωστά σαν Common Gateway Interface προγράμματα ή CGI scripts.

2.2 Τεχνική ISMAP.

Η τεχνική αυτή του Image map δίνει την δυνατότητα σε μία εικόνα να είναι ένας ενεργός σύνδεσμος. Αυτό σημαίνει ότι πατώντας με το ποντίκι πάνω στην εικόνα μπορείς να επιλέξεις οποιοδήποτε σημείο επιθυμείς.

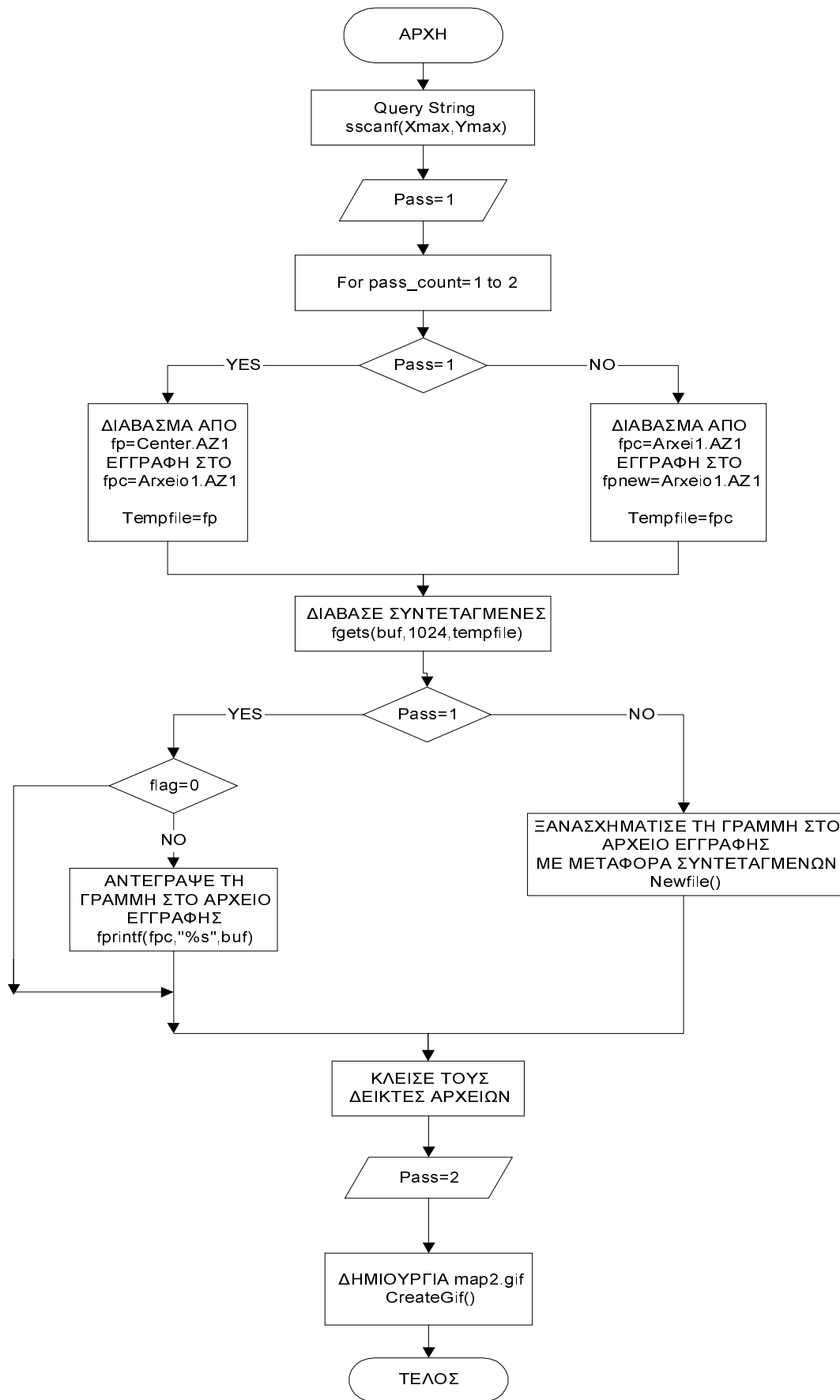
Ο κεντρικός χάρτης Map.gif γίνεται ενεργός σύνδεσμος μέσω της τεχνικής Ismap, όπως φαίνεται παρακάτω:

```
<IMG SRC="map.gif" ISMAP></A .
```

Πατώντας με το ποντίκι σε οποιοδήποτε σημείο του χάρτη οι συντεταγμένες του σημείου που πατήθηκε στέλνονται σε ένα πρόγραμμα στον server (cgi -bin), το οποίο επεξεργάζεται τις τιμές και αποφασίζει ποιά θα είναι η απάντηση που θα επιστρέψει στον χρήστη που έκανε την αίτηση προς τον server.

2.3 ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ

Για να γίνει κατανοητή η διαδικασία που ακολουθήθηκε ώστε από τον κεντρικό χάρτη map.gif να παραχθούν χάρτες που αφορούν συγκεκριμένη περιοχή που επιλέχθηκε ύστερα από αίτηση του χρήστη, καλό θα ήταν να δούμε μέσα από το λογικό διάγραμμα την πορεία του προγράμματος έως το τελικό αποτέλεσμα.



Το CGI πρόγραμμα (zoom0) ξεκινά από την στιγμή που ο χρήστης επιλέξει με το ποντίκι ένα σημείο πάνω στον χάρτη. Οι συντεταγμένες του σημείου που επιλέχθηκε μέσω της μεταβλητής Query String φθάνουν στον server.

Η πρώτη εργασία που γίνεται στον server, είναι η μετατροπή των τιμών που επιλέχθηκαν από τον χρήστη. Αυτό γίνεται λόγω του ότι ο κεντρικός χάρτης έχει δημιουργηθεί μέσω της τεχνικής normalized, που σημαίνει ότι κάθε συντεταγμένη πάνω στον χάρτη κυμαίνεται από 0 έως 1350 pixels. Επομένως ο server καλείται να μετατρέψει τις τιμές αυτές, ώστε αυτές να έχουν φυσική οντότητα μέσα στο διανυσματικό αρχείο .

Μετά τον καθορισμό του πρώτου σημείου θα πρέπει να γίνει από τον server αυτή τη φορά η επιλογή ενός δεύτερου σημείου, ώστε να δημιουργηθεί η περιοχή που αποτελεί και το βασικό συστατικό για την διεκπεραίωση του προγράμματος. Το σημείο ορίζεται προσθέτοντας στην τετμημένη και τεταγμένη του 600 pixels αντίστοιχα .

Το επόμενο βήμα είναι να γίνει το φιλτράρισμα του διανυσματικού αρχείου Center.Az1, ώστε να απομονωθούν οι συντεταγμένες των σημείων των οικοδομικών τετραγώνων που ανήκουν στην περιοχή που σχηματίστηκε. Οι συντεταγμένες των γωνιών των πολυγώνων αποθηκεύονται στο διανυσματικό αρχείο " arxeio1.az".

Το "arxeio2.az1" αποτελεί το διανυσματικό αρχείο στο οποίο θα βασιστεί η παραγωγή της εικόνας που θα επιστραφεί στον χρήστη. Μέσα στο αρχείο αυτό οι συντεταγμένες έχουν υποστεί μεταφορά αξόνων ως προς την αρχή .(Αρχή θεωρούμε το πάνω αριστερά σημείο της εικόνας 0,0).

Το τελικό στάδιο του προγράμματος είναι ο σχηματισμός της εικόνας map2.gif, η οποία αποτελεί την μεγέθυνση της περιοχής που επέλεξε ο χρήστης κατά την έναρξη του προγράμματος. Η διαδικασία σχηματισμού της εικόνας γίνεται με τον ίδιο τρόπο που αναλύθηκε στο προηγούμενο κεφάλαιο.

2.4 ΕΠΕΞΗΓΗΣΗ ΒΑΣΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΤΟΥ CGI ΠΡΟΓΡΑΜΜΑΤΟΣ Zoom0

```
CgiResponse( intx , inty )
```

Η συνάρτηση αυτή αποτελεί την απάντηση του Server στην αίτηση που δέχτηκε από τον Client. Επιστρέφει την ιστοσελίδα που περιέχει την μεγέθυνση της περιοχής που αιτήθηκε ο χρήστης.

```
Sscanf(argv[1], "%d,%d",&Xmax,&Ymax)
```

Μέσο της παραπάνω εντολής γίνεται η εισαγωγή στον Server των συντεταγμένων του σημείου που επέλεξε ο χρήστης.

```
X=(int)Xmax*(1419/900);  
Y=(int)Ymax*(2089/1350);
```

Όπως αναφέραμε, κατά την εισαγωγή των συντεταγμένων του σημείου που επέλεξε ο χρήστης ο Server καλείται να επεξεργαστεί τις τιμές αυτές. Πολλαπλασιάζει την τετμημένη (Xmax) και τεταγμένη (Ymax) του σημείου που πατήθηκε με ένα συντελεστή. Όπου 1419,2089 είναι η μέγιστη τετμημένη και τεταγμένη του αρχείου Center.Az1 και όπου 900,1350 είναι η μέγιστη τετμημένη και τεταγμένη του χάρτη Map.gif.

```
X1max=X + 600;  
Y1max=Y + 600;
```

Ο Server ορίζει την περιοχή που θα επιστέψει στον Client.

GetCoord()&&GetNum()

Οι συναρτήσεις αυτές σκοπό έχουν να επεξεργαστούν τα δεδομένα που θα διαβάσουν από το διανυσματικό αρχείο Center.Az1 και να τα αποθηκεύσουν. Κατά την επεξεργασία γίνεται ο έλεγχος για το ποιες είναι οι συντεταγμένες που ανήκουν στην περιοχή που σχηματίστηκε και ποιες όχι.

Newfile()

Η συνάρτηση αυτή ουσιαστικά δημιουργεί το διανυσματικό αρχείο "arxio2.Az1", μέσα από το οποίο θα σχηματιστεί η εικόνα που θα αποτελεί την μεγέθυνση της περιοχής που επέλεξε ο χρήστης.

CreateGif()

Η παραπάνω συνάρτηση είναι υπεύθυνη για τον σχηματισμό της εικόνας και η λειτουργία της γίνεται με ανάλογο τρόπο, όπως αυτόν του κεφαλαίου 1.

2.5 ΕΙΣΑΓΩΓΗ ΤΟΥ CGI ΠΡΟΓΡΑΜΜΑΤΟΣ Zoom0

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>
#include "gd.h"
#include "gd.c"

#define MAXPOINT 200

struct point
{
    int x;
    int y;
};

struct point point[MAXPOINT];
int xaraktr[8];

int  typeofpolygon;
int  curpoint;

char  *GetNum ();

static char  *PTR;
gdPoint points[MAXPOINT];

float norm1,norm2;
int w,h;
float sin_x,sin_y;

FILE *fpnew;
int Xmax,Ymax,X1max,Y1max,X,Y;
int counter=0;
int flag=0;
int pass=1;
```

```

void cgiResponse (int x, int y)
{
    printf ("Content-type: text/html\n\n");
    printf ("<HTML>\n");
    printf ("<HTML>\n");
    printf ("<BODY bgcolor=#FFFFFF>\n");
    printf ("<CENTER>\n");
    printf ("<p></p><br><br>\n");
    printf ("<p></p><br><br>\n");

    printf("<IMG SRC=\"http://iris.teipir.gr/~katzou/MARTHA/map2.gif\">");

    printf ("</BODY>\n");
    printf ("</HTML>\n");
}

main (int argc, char **argv)
{
    FILE *fp;
    FILE *fpc;
    FILE *tempfile;

    char buf[1024];
    int pass_count;

    X=0;
    Y=0;
    Xmax=0;
    Ymax=0;
    X1max=0;
    Y1max=0;

    if (sscanf (argv[1], "%d,%d", &Xmax, &Ymax) == 2)
    {
        X=(int)Xmax*(1419/900);
        Y=(int)Ymax*(2089/1350);

        X1max=X + 600;
        Y1max=Y + 600;

        cgiResponse (Xmax,Ymax);
    }
    else
        printf ("Query is invalid<BR>\n");
}

```

```

for (pass_count=1; pass_count<=2;pass_count++)
{

if(pass==1)
{
    fp = fopen ("/disk2/users/kotzou/MARTHA/center.az1", "r");

    if (fp == NULL)
    {
        printf ("Can't open the file center.az1");
        exit (0);
    }

    fpc = fopen ("arxeio1.az", "w");

    if (fpc == NULL)
    {
        printf ("Can't open the file arxeio1.az");
        exit (0);
    }

}
else
{
    fpc = fopen ("arxeio1.az", "r");

    if (fpc == NULL)
    {
        printf ("Can't read the file arxeio1.az");
        exit (0);
    }

    fpnew = fopen ("arxeio2.az", "w");
    if (fpnew == NULL)
    {
        printf ("Can't open the file arxeio2.az");
        exit (0);
    }

}

    typeofpolygon = 0;

if(pass==1)
{

```

```

    tempfile=fp;
}
else
{
    tempfile=fpc;
}
    while (fgets (buf, 1024, tempfile) != NULL)
    {
        PTR = &buf[0];

        if (strlen (buf) < 8)
            continue;
        if (GetCoord () == 1);
        if(pass==1)
        {
            if (flag==1)
            {
                fprintf(fpc,"%s",buf);
            }

            flag=0;
        }

        if(pass==2)
        {

            newfile();
        }

    }
    pass=2;
    fclose (fp);
    fclose (fpc);
    fclose(fpnew);
}
CreateGif();
}

```

```

GetCoord ()
{
    char *tmp;

    int i;
    int j;
    int k;

    if (pass==1)
    {
        for (i = 0; *PTR != '\0'; PTR++)
        {
            if (*PTR == ',')
                i++;
            if (i == 8 )
                break; } } else
    {
        for (k = 0; *PTR != '\0'; PTR++)
        {
            tmp=GetNum();
            xaraktr[k]=atoi(tmp);

            if (*PTR == ',')
                k++;
            if (k == 8 )
                break;
        }
    }

    curpoint = 0;

    PTR++;
    tmp = GetNum ();

    if (tmp == NULL)
        return (0);

    typeofpolygon = atoi (tmp);

```

```

for (;;)
{
    if (*PTR == '\0' || *PTR == '\n')
        break;
        PTR++;

    tmp = GetNum ();
    if (tmp == NULL)
        break;

    point[curpoint].x = atoi (tmp);

    if (*PTR == '\0' || *PTR == '\n')
        break;
        PTR++;

    tmp = GetNum ();
    if (tmp == NULL)
        break;

    point[curpoint].y = atoi (tmp);

    if( ((point[curpoint].x > X)&&(point[curpoint].x < X1max)) &&
((point[curpoint].y > Y)&& (point[curpoint].y < Y1max)))

    {
        flag=1;
    }

    curpoint ++;
}

return (1);
}

```

```

char *GetNum ()
{
    int idx;
    char ret[200];
    idx = 0;          while (*PTR != ',' && *PTR != '\n' && *PTR != '\0')
    {
        if ((*PTR == ' ')&&*(PTR-1)!=','))
        {
            PTR++;
            continue;
        }

        if ((*PTR == ' ')&&*(PTR-1)!=','))
            break;

        ret[idx] = *PTR;

        idx++;
        PTR++;
    }

    ret[idx] = '\0';

    if (idx == 0)
        return (NULL);

    return (ret);
}

```

```

newfile()
{

int i;

for(i=0;i<4;i++)
    {
        fprintf(fpnew,"%d",xaraktr[i]);
    }

for(i=4;i<8;i++)
    {
        fprintf(fpnew,"%d",xaraktr[i]);
    }

fprintf(fpnew," %d",typeofpolygon);
for(i=0;i<typeofpolygon;i++)
    {
        fprintf(fpnew," %d,%d",(point[i].x-X),(point[i].y-Y));
    }

fprintf(fpnew,"\n");

}

```

```

CreateGif ()
{

```

```

    FILE *fp;
    char buf[1024];
    gdImagePtr im;
    FILE *out;

```

```

        int black;
        int white;
        int red;

```

```

out = fopen("/disk2/users/kotzou/WWW/MARTHA/map2.gif", "wb");

```

```

fp = fopen ("arxeio2.az", "r");

```

```

if (fp == NULL)
    {
        printf ("Can't open the file arxeio2.az");

        exit (0);
    }

```

```
w =800;

im = gdImageCreate(w,w);

black = gdImageColorAllocate(im, 0, 0, 0);
white = gdImageColorAllocate(im, 255, 255, 255);
red = gdImageColorAllocate(im, 255, 0, 0);

    norm1=w;
norm2=w;
    sin_x=norm1/600;
    sin_y=norm2/600;
    typeofpolygon = 0;

while (fgets (buf, 1024, fp) != NULL)
{
    PTR = &buf[0];
    if (strlen (buf) < 8)
        continue;
    if (GetCoordGif() == 1);

        gdImageFilledPolygon(im, points, typeofpolygon, red);
        gdImagePolygon(im, points, typeofpolygon, white); }fclose (fp);

gdImageGif(im, out);
fclose(out);

gdImageDestroy(im);

}
```

```

GetCoordGif ()
{
    char *tmp;
    int i;
    for (i = 0; *PTR != '\0'; PTR++)
    {
        if (*PTR == ',')
            i++;
        if (i == 8 )
            break;
    }
    curpoint = 0;

    PTR++;
    tmp = GetNum ();
    if (tmp == NULL)
        return (0);
    typeofpolygon = atoi (tmp);
    for (;;)
    {
        if (*PTR == '\0' || *PTR == '\n')
            break;
        PTR++;
        tmp = GetNum ();
        if (tmp == NULL)
            break;
        point[curpoint].x = atoi (tmp);
        point[curpoint].x= (int) ((point[curpoint].x)*sin_x);
        points[curpoint].x=point[curpoint].x;

        if (*PTR == '\0' || *PTR == '\n')
            break;
        PTR++;
        tmp = GetNum ();
        if (tmp == NULL)
            break;
        point[curpoint].y = atoi (tmp);
        point[curpoint].y= (int) ((point[curpoint].y)*sin_y);
        points[curpoint].y=point[curpoint].y;
        curpoint ++;
    } return (1); }

```

2.6 ΕΠΕΚΤΑΣΙΜΟΤΗΤΑ

Η πτυχιακή αυτή εργασία εκμεταλλευόμενη τις πληροφορίες διανυσματικών αρχείων έκανε εφικτή την απεικόνιση χαρτών και έδωσε στο χρήστη εργαλεία για να εκμαιεύσει επιπλέον χρήσιμες πληροφορίες. Ολοκληρώνοντας αυτή την πτυχιακή εργασία έγινε φανερό ότι εξαντλώντας με λεπτομέρεια τα ζητούμενά της και φτιάχνοντας ένα περιβάλλον, όπου μπορούν με δυναμικό τρόπο να ενσωματωθούν νέες λειτουργίες, δημιουργήθηκε τελικά μια πλατφόρμα πάνω στην οποία μπορούν να χτιστούν πάμπολλες επιπρόσθετες εφαρμογές.

Κάποιες από αυτές τις εφαρμογές που θα μπορούσαν να προστεθούν, με τη βοήθεια των κατάλληλων αρχείων, είναι η απεικόνιση και αναζήτηση πάνω στον χάρτη: Φαρμακείων, Νοσοκομείων, Σχολείων, Εκπαιδευτικών Ιδρυμάτων, Βιβλιοθηκών, Γηπέδων, Αθλητικών Κέντρων, Υπουργείων, Πρεσβειών, Ξενοδοχείων, Πάρκων, Θεάτρων, δρομολογίων Λεωφορείων, ΟΣΕ, ΜΕΤΡΟ, ΚΤΕΛ, ΕΛΤΑ, ΔΟΥ.

Επίσης με την βοήθεια κατάλληλων αλγορίθμων θα μπορούσαν να ενσωματωθούν λειτουργίες, όπως η εύρεση μιας διασταύρωσης, η εύρεση των κατοίκων ενός οικοδομικού τετραγώνου, καθώς και η εύρεση ενός δρόμου. Θα μπορούσε ο χρήστης με το πάτημα ενός πλήκτρου να αναζητήσει τον δρόμο τον οποίο επιθυμεί, και μετά τον εντοπισμό του εν λόγω δρόμου, να εμφανίζεται με διακριτικό κόκκινο χρώμα όλη η περιοχή την οποία διασχίζει, ώστε ο χρήστης να μπορεί να προσανατολιστεί καλύτερα στο χάρτη.

Χρήσιμο θα ήταν να ενσωματωθούν λειτουργίες, μέσω των οποίων να μπορούμε να ανακτήσουμε πληροφορίες για τις αντικειμενικές αξίες όλων των οικοδομικών τετραγώνων, και πιο συγκεκριμένα την τιμή ζώνης ανά τετραγωνικό μέτρο, το συντελεστή εμπορικότητας και την τιμή του οικοπέδου ανά τετραγωνικό μέτρο.

Επιπλέον θα ήταν πολύ χρήσιμη η δημιουργία εργαλείων που θα επέτρεπε στο χρήστη να εντοπίσει ένα σπίτι, δίνοντας τη διεύθυνση ή το όνομα του ιδιοκτήτη, τους δρόμους που βρίσκονται στο δακτύλιο και τους πεζόδρομους .

Τέλος εξαιρετικά σημαντική θα ήταν η υλοποίηση εφαρμογής που θα επέτρεπε τη μέτρηση αποστάσεων και την εύρεση της βέλτιστης διαδρομής, η οποία για να είναι ρεαλιστική θα μπορούσε να γίνεται σε συνεργασία με βάσεις δεδομένων για την κίνηση στους δρόμους.

ΚΕΦΑΛΑΙΟ 3

3.1 ΕΙΚΟΝΕΣ ΚΑΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ

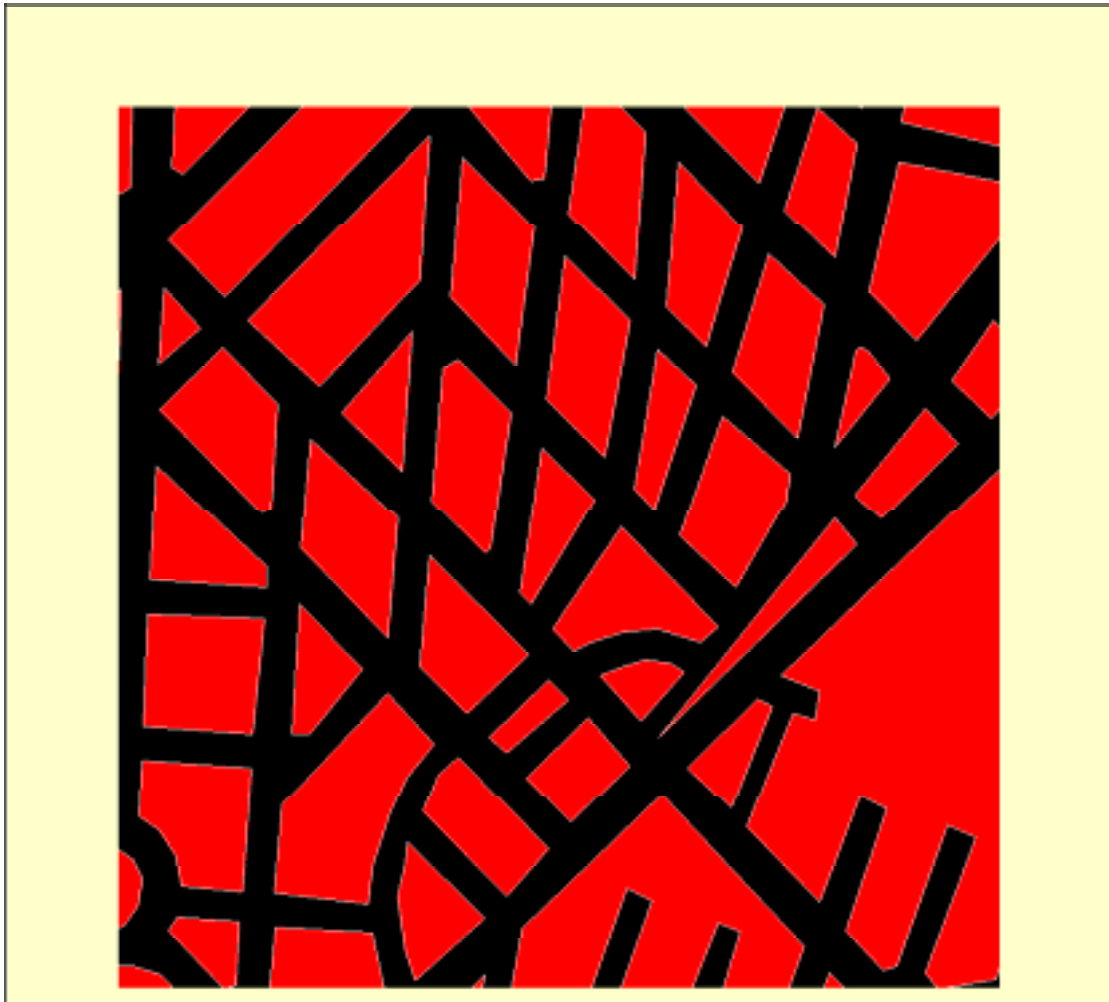
Η κεντρική σελίδα της εφαρμογής βρίσκεται στον Server Irida, και η διεύθυνσή της είναι : <http://iris.teipir.gr/~katzou/MARTHA/martha.html>.

Στην Εικόνα 3.1 εμφανίζεται ο κεντρικός χάρτης της εφαρμογής μας. Το κόκκινο κουμπί το οποίο είναι ενεργοποιημένο δηλώνει ότι βρισκόμαστε στο πρώτο επίπεδο μεγέθυνσης. Η πλοήγηση του χρήστη πάνω στον χάρτη ξεκινά και με το απλό πάτημα του ποντικιού πάνω στην περιοχή που ο ίδιος επιθυμεί.



Εικόνα 3.1

Η ενέργεια αυτή του χρήστη καταγράφεται και μέσω του cgi προγράμματος zoom0 οι συντεταγμένες του σημείου που πατήθηκε περνούν στον Server - Irida, ο οποίος επεξεργάζεται τις τιμές και επιστέφει στον Client την Εικόνα 3.2.



Εικόνα 3.2

Η εικόνα αυτή αποτελεί την μεγέθυνση της περιοχής του γεωγραφικού χάρτη που αιτήθηκε ο Client.

Η ανάγκη του χρήστη για μεγαλύτερη ευκρίνεια τον οδηγεί στην ενεργοποίηση του δεύτερου επιπέδου μεγέθυνσης, όπως φαίνεται στην εικόνα 3.3.



Εικόνα 3.3

Ο χρήστης ξεκινά και πάλι την πλοήγησή του πάνω στον κεντρικό χάρτη, και επιλέγει μια νέα περιοχή.

Οι συντεταγμένες του σημείου που πατήθηκε περνούν μέσω του Cgi προγράμματος – zoom1 στον server – Irida , ο οποίος τις επεξεργάζεται και επιστρέφει στον Client την περιοχή που επέλεξε, όπως φαίνεται στο σχήμα 3.4.



Εικόνα 3.4

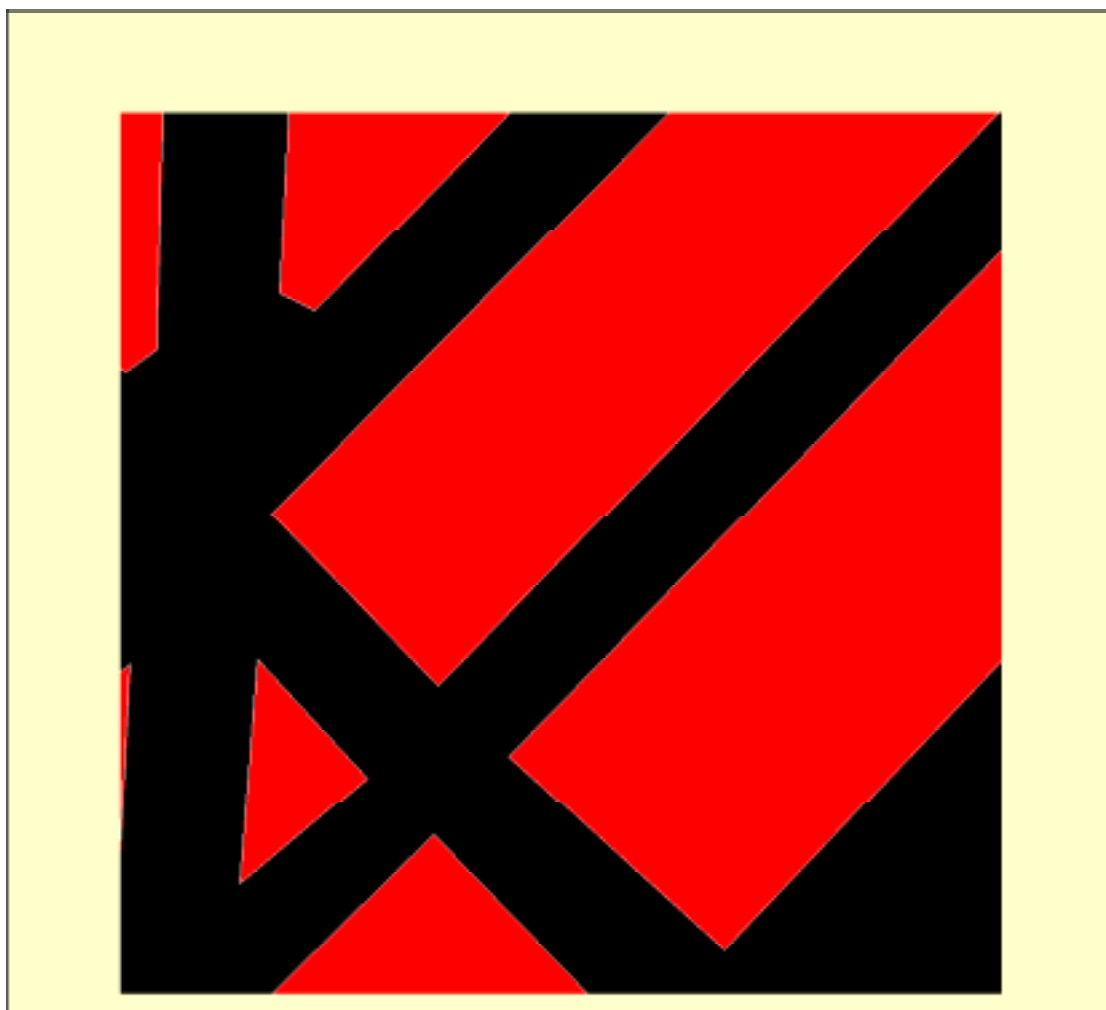
Η εικόνα 3.4 αποτελεί την μεγενθυμένη περιοχή που επέλεξε ο χρήστης και όπως φαίνεται έχει υποστεί μεγαλύτερη μεγέθυνση από αυτή που συντελέστηκε στο πρώτο επίπεδο.

Η ενεργοποίηση του τρίτου επιπέδου μεγέθυνσης, όπως φαίνεται στο σχήμα 3.5 έρχεται να ολοκληρώσει τις ανάγκες του χρήστη για καλύτερη ευκρίνεια.



Εικόνα 3.5

Ο χρήστης επιλέγει την περιοχή που επιθυμεί και περιμένει να δει όσο το δυνατό καλύτερα τα οικοδομικά τετράγωνα της περιοχής του χάρτη που επέλεξε. Οι συντεταγμένες του σημείου που επέλεξε περνούν μέσω του cgi-προγράμματος Zoom2 και η μεγεθυμένη περιοχή που επιστρέφεται εμφανίζεται στην εικόνα 3.6.



Εικόνα 3.6

Στο σημείο αυτό θα θέλαμε να τονίσουμε ότι η παραγωγή των εικόνων έγινε από διανυσματικά -Vector αρχεία και ότι η οι μεγεθύνσεις που συντελέστηκαν προσφέρουν την μέγιστη δυνατή ανάλυση τόσο σε επίπεδο εικόνας όσο και σε επίπεδο εκτύπωσης γιατί οι εικόνες δημιουργούνται κάθε φορά διαβάζοντας το αρχικό Vector αρχείο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Turbo C Robert Lafore
2. C για αρχάριους Σεφερίδης Βασίλειος
3. <http://www.boutel.com>
4. Gis servises Παπαδάτος-Τραυλού
5. Gis με τεχνικές Vector Παπαδόπουλος-Πουλής