



**ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ**

**Π Τ Υ Χ Ι Α Κ Η   Ε Ρ Γ Α Σ Ι Α**

*Θέμα:*

**DYNAMIC MAP GENERATOR**

*Σπουδαστής:*

**Παπασυμεού Παναγιώτης**

**ΑΜ 569**

*Εισηγητής:*

**Καθηγητής  
PhD Βιδάκης Νικόλαος**  
Ηράκλειο, Μάιος 2007

*Ευχαριστίες  
στον καθηγητή μου κ. Βιδάκη Νικόλα  
για την βοήθεια και συμπαράσταση  
που μου προσέφερε κατά την διάρκεια  
της πτυχιακής μου εργασίας .*

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΚΕΦ.1</b>	<b>5</b>
<u>Εισαγωγή</u>	
1.1 Γενικά για τους ηλεκτρονικούς Υπολογιστές	5
1.2 Γενικά για τους Αλγόριθμους	6
1.3 Μοντέλα Διαδικασίας Λογισμικού	6
1.4 Ιδιότητες καλού λογισμικού	7
1.5 Διαδικασία Λογισμικού	8
1.6 Ανάλυση Λογισμικού	8
1.7 Ανάπτυξη Προγράμματος	8
<b>ΚΕΦ.2</b>	<b>10</b>
2 1. Visual Basic	
2.1.1 Γενικά για την Visual Basic	10
2.1.2 Είδη Μεταβλητών	11
2.1.3 Από τι αποτελείτε ένα project	11
2.1.4 Λογικοί τελεστές	12
2.1.5 Λογικές Συναρτήσεις	12
2.1.6 Λόγοι Επιλογής VISUAL BASIC	12
2.2 Database Access	
2.2.1 Τι είναι η Βάση Δεδομένων;	13
2.2.2 Πίνακες και Πεδία	13
2.2.3 Λόγοι Επιλογής MS-Access	14
<b>ΚΕΦ.3</b>	<b>15</b>
3 Ανάλυση Προγράμματος	
3.1 Δυναμικοί Χάρτες	15
3.2 Απαιτήσεις	15
3.3 Λειτουργία Προγράμματος	16
3.3.1 Επιλογή Άνοιγμα –Open	16
3.3.2 Εισαγωγή δεδομένων στο χάρτη	17
3.3.3 Πρόσβαση στη Βάση Δεδομένων	18
3.3.4 Τροποποιήσεις στα δεδομένα του χάρτη	18
3.3.5 Πρόσθεση Επιλογών από το χρήστη	19
3.3.6 Η επιλογή Εκτύπωσης	20
3.3.7 Η επιλογή Φόρτωσης	21
3.3.8 Η επιλογή Εξόδου	22
3.3.9 Επιλογή Edit – Delete All	22
3.3.10 Επιλογή About – Calendar	22

3.4 Τεχνικά Προγράμματος	23
3.4.1 Κίνηση Mouse	24
3.4.2 Εμφάνιση συντεταγμένων	25
3.4.3 Εάν πατηθεί το Mouse	25
3.4.4 Βάση Δεδομένων	25
3.4.5 Η σύνδεση με τη βάση	27
3.4.6 Εμφάνιση δεδομένων στην φόρμα	28
3.4.7 Δημιουργία Δυναμικών Text Boxes	28
3.4.8 Συντεταγμένες χ ,ψ	29
3.4.9 OLE DB	30
3.4.10 Τροποποίηση δεδομένων	31
3.4.11 Scroll bars	33
3.4.12 Menu bar	34
3.4.13 Η επιλογή Αποθήκευσης	36
3.4.14 Η επιλογή Φόρτωσης	36
3.4.15 Έξοδος Από το πρόγραμμα	37
3.4.16 Διαγραφή όλων των στοιχείων στη Φόρμα	38
3.4.17 About – Ημερολόγιο	38
3.4.18 About – Help	38
3.4.19 Χρονιστές	39
3.5 Access	40
<b><u>Παράρτημα 1</u></b>	<b>43</b>
Error Report	43
<b><u>Παράρτημα 2</u></b>	<b>46</b>
Κώδικας Προγράμματος Dynamic Map	46
<b><u>Παράρτημα 3</u></b>	<b>73</b>
Εγχειρίδιο Χρήστη	73
<b><u>Βιβλιογραφία</u></b>	<b>78</b>

## ΚΕΦ.1

### ΕΙΣΑΓΩΓΗ

#### 1.1 Γενικά για τους ηλεκτρονικούς Υπολογιστές

Είναι γνωστό στις μέρες μας πως σκοπός των Ηλεκτρονικών Υπολογιστών είναι να διευκολύνουν και να βοηθήσουν τη ζωή μας καθώς και τη δουλειά μας Ένας Ηλεκτρονικός υπολογιστής αποτελείται από : Hardware και Software .

Ας έχουμε υπόψη μας πως με τον όρο Hardware εννοούμε το σύνολο των ηλεκτρικών , ηλεκτρονικών , ηλεκτρομαγνητικών , οπτικών και μηχανικών εξαρτημάτων .

Με τον όρο Software εννοούμε : α) το Λειτουργικό Σύστημα (OS) που σκοπό έχει τη σωστή λειτουργία του υπολογιστή καθώς και την ανάπτυξη νέου λογισμικού και

β) το Λογισμικό Εφαρμογών που είναι τα διάφορα προγράμματα για τα οποία αγοράσαμε τον Ηλεκτρονικό Υπολογιστή.

Πως γίνεται τώρα η ανάπτυξη των προγραμμάτων. Οι προγραμματιστές με την βοήθεια των διαφόρων ‘Γλωσσών Προγραμματισμού ‘ όπως ονομάζονται είναι σε θέση να δημιουργήσουν είτε μέσα από ομάδες είτε αυτόνομα προγράμματα τα οποία θα εξυπηρετούν τον πελάτη . Τα προγράμματα αυτά που είναι μέρος του Λογισμικού μπορεί να είναι : α) Γενικής Χρήσεως β) Ειδικής κατασκευής , που είναι για κάποιο συγκεκριμένο πελάτη .

Μερικές Γλώσσες Προγραμματισμού είναι : C++ , Java , PHP , VB κ.τ.λ

Εμείς εδώ θα ασχοληθούμε εδώ με ένα πρόγραμμα ειδικής κατασκευής που θα δίνει τη δυνατότητα στο χρήστη να επεξεργαστεί χάρτες . Θα είναι σε θέση να προσθέσει ή και να αφαιρέσει δεδομένα σε αυτόν . Θα είναι δηλαδή ένας δυναμικός χάρτης.

Τα Δεδομένα αυτά πρέπει να αποθηκεύονται σε κάποιο χώρο . Για αυτό υπάρχουν οι βάσεις δεδομένων . Στο πρόγραμμα αυτό ( Dynamic Map Generator ) χρησιμοποιούμε την Access για το λόγο ότι είναι απλή , εύχρηστη και λειτουργική .

Σε τέτοια πρόγραμμα ένας προγραμματιστής συχνά χρησιμοποιεί αλγόριθμους για δική του διευκόλυνση .

## 1.2 Αλγόριθμοι

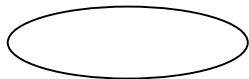
Ο Αλγόριθμος δεν είναι τίποτα άλλο από μία μέθοδο που χρησιμοποιεί ο προγραμματιστής για να υπολογίσει το πρόγραμμα και θα χρησιμοποιηθεί και εκτελεστεί κατά τη διάρκεια του .

Ένας αλγόριθμος θα πρέπει :

- Να είναι ακριβής , με απλά βήματα
- Να είναι αποτελεσματικός

Επειδή είναι ευκολότερο να μεταφέρουμε αυτό που θέλουμε σε εικόνες παρά με λόγια συχνά εκφράζουμε τον αλγόριθμο σε εικονική μορφή ή αλλιώς διαγράμματα ροής . Αυτά αποτελούνται από κλειστά σχήματα τα οποία περιέχουν εντολές και συνδέονται μεταξύ τους με γραμμές . Για να διαβάσουμε ένα διάγραμμα ροής θα πρέπει να ακολουθούμε γραμμές από σχήμα σε σχήμα .

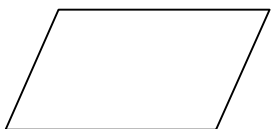
Διαγράμματα ροής :



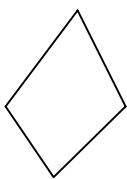
Αρχή / Τέλος



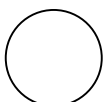
Επεξεργασία



Είσοδος / Έξοδος

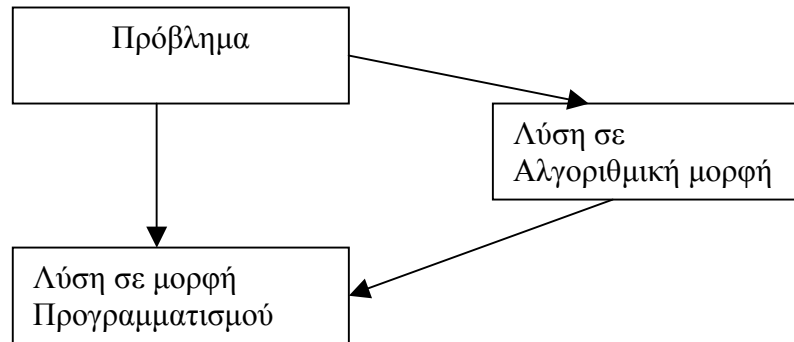


Επιλογή



Διασύνδεση

Η αντιμετώπιση που πρέπει να έχει ένα πρόγραμμα είναι η ακόλουθη :



### Μοντέλα Διαδικασίας λογισμικού :

Για την καλύτερη αντιμετώπιση ενός προγράμματος χρησιμοποιούμε τα μοντέλα διαδικασίας λογισμικού .

Μερικά γενικά μοντέλα διαδικασίας είναι :

- Waterfall
- Evolutionary development
- Formal transformation

Στα προγράμματα χρησιμοποιούνται συχνά συναρτήσεις από τους προγραμματιστές . Οι συναρτήσεις ( Functions) είναι ειδικές ρουτίνες τις οποίες παρέχει μία γλώσσα προγραμματισμού για να βοηθήσει τους προγραμματιστές στην υλοποίηση πολύπλοκων υπολογισμών .

### 1.3 Ιδιότητες Καλού Λογισμικού

- Να προσφέρει αυτά που έχουν ζητηθεί από τον πελάτη
- Αξιοπιστία
- Ευχρηστία

Στα επόμενα κεφάλαια θα αναλύσουμε όλες τις διαδικασίες , αλγορίθμους καθώς και τον κώδικα που συναντάτε στο Dynamic Map v1.1

## **1.4 Διαδικασία λογισμικού**

Κατά τη διάρκεια της διαδικασίας καταγράφονται και αναλύονται οι απαιτήσεις του λογισμικού που υλοποιούμε .

## **1.5 Ανάλυση Λογισμικού**

### Διαδικασία Επίλυση του προβλήματος

- α) Προσδιορισμός εξόδου
- β) Αναγνώριση Εισόδου
- γ) Καθορισμός Επεξεργασίας

### Σχεδιασμός Προγράμματος

Η έξοδος καθορίζει την είσοδο και την επεξεργασία

## **Απαιτήσεις Λογισμικού**

Η Απαίτηση λογισμικού είναι ένας περιορισμός του συστήματος ή ο τεχνικός προσδιορισμός ενός προβλήματος.

Οι τύποι των απαιτήσεων είναι τρεις :

- Οι απαιτήσεις του χρήστη
- του συστήματος
- και ο προσδιορισμός του λογισμικού

## **1.6 Ανάπτυξη Προγράμματος**

1. Ανάλυση : Προσδιορισμός Προβλήματος

2. Σχεδιασμός : σχεδιασμοί επίλυσης του προβλήματος

- π.χ ανάπτυξη του αλγόριθμου
- Χρησιμοποίηση προγραμματιστικών εργαλείων όπως τα : flowcharts, pseudocode

3. Επιλογή της διεπαφής : Επιλογή εργαλείων και αντικειμένων που θα χρησιμοποιηθούν (text boxes, command buttons, κ.τ.λ).

4. Κώδικας: Μετάφραση και μετατροπή του αλγόριθμου σε γλώσσα προγραμματισμού.

- Κωδικοποίηση: Γράψιμο του κώδικα



5. Τεστ: Εντοπισμός πιθανόν ‘λαθών’ στο πρόγραμμα .

- Είδη λαθών

α) Συντακτικά λάθη: Εντοπίζονται από την Visual Basic κατά τη διάρκεια της εκτέλεσης του προγράμματος

β) Λογικά λάθη : Εντοπίζονται από το χρήστη κατά τη διάρκεια του debugging

γ) Τυπικά λάθη : εντοπίζονται από το χρήστη κατά την εκτέλεση του προγράμματος

## **ΚΕΦ 2**

### **Τεχνολογίες και Εργαλεία Υλοποίησης**

#### **2.1 VISUAL BASIC 6**

Η Visual Basic είναι η αγαπημένη γλώσσα πολλών προγραμματιστών . Όταν πρωτοεμφανίστηκε δημιούργησε μία επανάσταση στον γραφικό προγραμματισμό .Ποτέ πριν ο προγραμματισμός δεν ήταν τόσο εύκολος να γίνει κάτω από ένα γραφικό περιβάλλον και να υλοποιηθεί – τρέξει στην συνέχεια .

Έχοντας τα γραφικά , για τον προγραμματιστή είναι πιο ευχάριστος ο προγραμματισμός και σταματά να είναι αγγαρεία . Με την πάροδο του χρόνου η Visual Basic έγινε πιο πολύπλοκη και ταυτόχρονα πιο ισχυρή και αποτελεσματική .

Ας δούμε τώρα εν συντομία μερικές λειτουργίες της γλώσσας αυτής γνωστής στο κοινό ως Visual Basic 6 .

Δημιουργία καινούργιου project στην Visual Basic . Καταρχάς αρχίζοντας ένα καινούργιο project έχουμε πλειάδα επιλογών σχετικά για το πρόγραμμα που θα δημιουργήσουμε όπως :

- Standard Windows EXE programs
- ActiveX EXE files
- ActiveX DLLs
- ActiveX controls
- ActiveX document DLLs
- ActiveX document EXE files
- DHTML applications κ.τ.λ

Με την πάροδο του χρόνου αυτή η λίστα ολοένα και αυξάνεται δίνοντας στον χρήστη και άλλες επιλογές .

Ας δούμε τώρα τα διάφορα είδη των μεταβλητών που υπάρχουν σε αυτή τη γλώσσα καθώς μερικές από αυτές θα χρησιμοποιηθούν και στο πρόγραμμα Dynamic Map .

### **2.1.2 Είδη μεταβλητών :**

Boolean : 2 True or False

Byte : 1 0 to 255

Decimal : 12 -79,228,162,514,264,337,593,543,950,335 μέχρι  
79,228,162,514,264,337,593,543,950,335

Double : 8 -1.79769313486232E308 μέχρι 4.94065645841247E-324

Integer : 2 -32,768 μέχρι 32,767

Long : 4 -2,147,483,648 έως 2,147,483,647

Single : 4 -3.402823E38 έως -1.401298E

String : Μπορεί να είναι ίσο μέχρι και δύο δισεκατομμύρια

### **2.1.3 Από τι αποτελείτε ένα project**

Ένα project είναι δυνατό να αποτελείται από :

#### **Φόρμες (Forms)**

Δεν είναι τίποτα άλλο από περιγράμματα – φόρμες(Windows) όπου φαίνεται και η εργασία του προγραμματιστή .

#### **Υπομονάδα (Modules)**

Οι υπομονάδες είναι μία συλλογή από κώδικα και δεδομένα που λειτουργούν με τα διάφορα αντικείμενα που υπάρχουν στη φόρμα .

#### **Γενικά Αντικείμενα (Global Items)**

Τα αντικείμενα αυτά είναι προσβάσιμα από όλες τις υπομονάδες και όλες τις φόρμες του προγράμματος . Τις δηλώνουμε στην αρχή του κώδικα χρησιμοποιώντας την λέξη κλειδί - Public

## Αντικείμενα Μεταβλητών (Variable Scope)

Οι μεταβλητές δηλώνονται με διάφορους τρόπους . Συχνότερα χρησιμοποιείται η δήλωση Dim . Παρακάτω βλέπουμε τα διάφορα είδη μεταβλητών που χρησιμοποιούνται στην Visual Basic .

### 2.1.4 Λογικοί Τελεστές

Σε μία συνθήκη υπάρχουν διάφορες εκφράσεις όπως : =, <>, <, >, <=, >= , καθώς και τα λογικά : And, Or, Not .

### 2.1.5 Λογικές Συναρτήσεις :

Έστω ότι έχουμε δύο συνθήκες : συνθ1 και συνθ2 τότε :

Οι συνθήκες ( συνθ1 And συνθ2) είναι αληθείς εάν και οι δύο συνθήκες είναι αληθείς .

Οι συνθήκες ( συνθ1 Or συνθ2) είναι αληθείς εάν μία από τις δύο συνθήκες είναι αληθής .

Οι συνθήκη (Not συνθ1) είναι αληθής εάν η συνθήκη συνθ1 είναι ψευδής .

### 2.1.6 Λόγοι Επιλογής Visual Basic

- υποστηρίζει αντικειμενοστραφή προγραμματισμό
- αναλυτική και ευανάγνωστη
- δυνατότητα υποστήριξης βάσεων δεδομένων
- γρήγορος μεταγλωττιστής,

## **2. MS-Access Database**

Οι περισσότεροι προγραμματιστές που χρησιμοποιούν την Visual Basic κάποια στιγμή χρειάζονται μία βάση δεδομένων για να υποθηκεύσουν τα δεδομένα τους . Γιατί Access : η Access είναι απλή ,εύχρηστη και με απλές εντολές μπορεί να γίνει η σύνδεση μέσω της Visual Basic .

### **2.2.1 Τι είναι η Βάση Δεδομένων;**

Μία βάση δεδομένων είναι μία αποθήκη πληροφοριών . Υπάρχουν διάφορων ειδών βάσεων δεδομένων . Με την Access μπορούμε :

- Να αποθηκεύουμε δεδομένα σε πίνακες ( tables) οι οποίοι αποτελούνται από διάφορες στήλες και γραμμές
- Να ανακτούμε δεδομένα μέσα από τους πίνακες της βάσης
- Να συνδέουμε τους πίνακες μεταξύ τους για την ανάκτηση δεδομένων που υπάρχουν σε άλλους πίνακες .

### **2.2.2 Πίνακες και Πεδία**

Μία βάση δεδομένων αποτελείται από πίνακες οι οποίοι με τη σειρά τους αποτελούνται από εγγραφές οι οποίες αυτές εγγραφές με την σειρά τους αντιπροσωπεύουν τα πεδία .

Με τη βοήθεια της Visual Basic και τον κώδικα πίσω από αυτήν μπορούμε να διαχειριστούμε τους πίνακες τις εγγραφές και τα πεδία που υπάρχουν στη βάση δεδομένων .

Ένας πίνακας αποτελεί ένα τρόπο με τον οποίο τα δεδομένα συγκροτούν και οργανώνουν την πληροφορία μέσα στη βάση .Οι πίνακες αυτοί έχουν μία προκαθορισμένη δομή , περιέχουν δεδομένα τα οποία ταιριάζουν με την δομή του . Οι πίνακες οργανώνουν την πληροφορία σε γραμμές και στήλες . Στον πίνακα μία γραμμή δεδομένων αποκαλείται 'εγγραφή' , και μία στήλη δεδομένων αποκαλείται 'πεδίο' .

Μία εγγραφή αντιπροσωπεύει ένα συγκεκριμένο στοιχείο ( δεδομένο) όπως για παράδειγμα την διεύθυνση κάποιου ατόμου .

Ένα πεδίο αποτελεί μία υποκατηγορία των δεδομένων σε μία εγγραφή .Μία εγγραφή η οποία αντιπροσωπεύει την διεύθυνση κάποιου ατόμου αποτελείται από τα πεδία :

- Επίθετο
- Όνομα
- Διεύθυνση
- Ταχυδρομικός κώδικας
- Τηλέφωνο κ.τ.λ

### **2.2.3 Λόγοι Επιλογής MS-Access**

- Απλός ο σχεδιασμός της βάσης
- Εύκολη στη χρήση
- Απλή η σύνδεση με το πρόγραμμα

## ΚΕΦ.3

### Ανάλυση Προγράμματος

Η πτυχιακή εργασία έχει σκοπό να χρησιμοποιεί στατικούς χάρτες και να τους μετατρέπει σε δυναμικούς . Όλα τα δεδομένα τα οποία προσθέτονται – αποθηκεύονται πάνω στο χάρτη είναι δυναμικά .

#### 3.1 Δυναμικοί χάρτες

Με τον όρο αυτό αναφερόμαστε σε χάρτες οι οποίοι είναι σε διανυσματική μορφή και στους οποίους παρέχεται η δυνατότητα μεγέθυνσης, σμίκρυνσης. Ο χάρτης ως φωτογραφία που φορτώνεται στο πρόγραμμα δεν αλλοιώνεται . Ο χρήστης δημιουργεί ενεργές περιοχές πάνω στο χάρτη .Όλα τα δεδομένα αποθηκεύονται σε μία βάση δεδομένων αυτόματα .Οι δυναμικοί χάρτες δημιουργούνται σε πραγματικό χρόνο, παράγοντας κάθε φορά διαφορετικό χάρτη ανάλογα με τις προτιμήσεις του χρήστη.

#### 3.2 Απαιτήσεις

1. Ο χρήστης θα πρέπει να έχει τη δυνατότητα να φορτώνει ένα χάρτη οποιασδήποτε μορφής στο πρόγραμμα.
2. Θα παρέχεται στο χρήστη επιλογή γεωμετρικών σχημάτων για ορισμό ενεργών περιοχών .
3. Στην διάθεση του ο χρήστης θα πρέπει να έχει μία ευρεία επιλογή αντικειμένων (κτιρίων) καθώς και η δυνατότητα πρόσθεσης αντικειμένων από το χρήστη .
4. Η δυνατότητα τροποποίησης των ενεργών περιοχών .
5. Δυνατότητα σμίκρυνσης και μεγέθυνσης του χάρτη .
6. Αναλυτική εμφάνιση όλων των δεδομένων στο πρόγραμμα.
7. Πρόσβαση του χρήστη στη βάση δεδομένων .

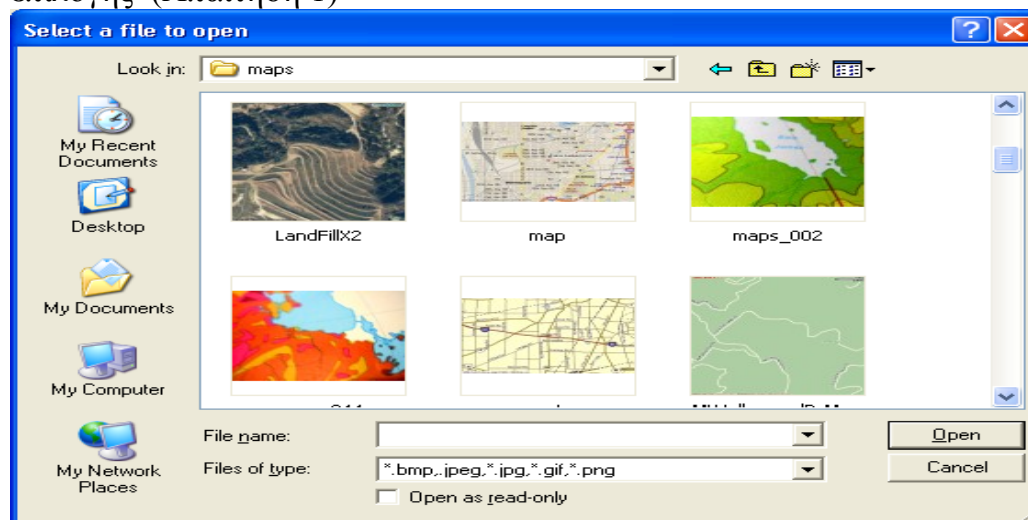
### 3.3 Λειτουργίες Προγράμματος

Στην παρακάτω εικόνα ο χρήστης φορτώνει ένα χάρτη με το όνομα city3 του τύπου jpg . Στην συνέχεια επιλέγει για σχήμα σχεδίασης το ορθογώνιο . Στα combo boxes επιλέγει στο πρώτο 'hotel' και στο δεύτερο 1000 τετραγωνικά μέτρα . Στην συνέχεια με χρήση του mouse επιλέγει την περιοχή που θέλει και τοποθετεί το ξενοδοχείο . Αυτό γίνεται και για τα υπόλοιπα



#### 3.3.1 Επιλογή Άνοιγμα –Open

Πρώτη επιλογή κάτω από τον τίτλο 'File' βρίσκεται η 'Open File' . Κάνοντας κλικ σε αυτή την επιλογή ο χρήστης εμφανίζεται το παράθυρο επιλογής (Απαίτηση 1)





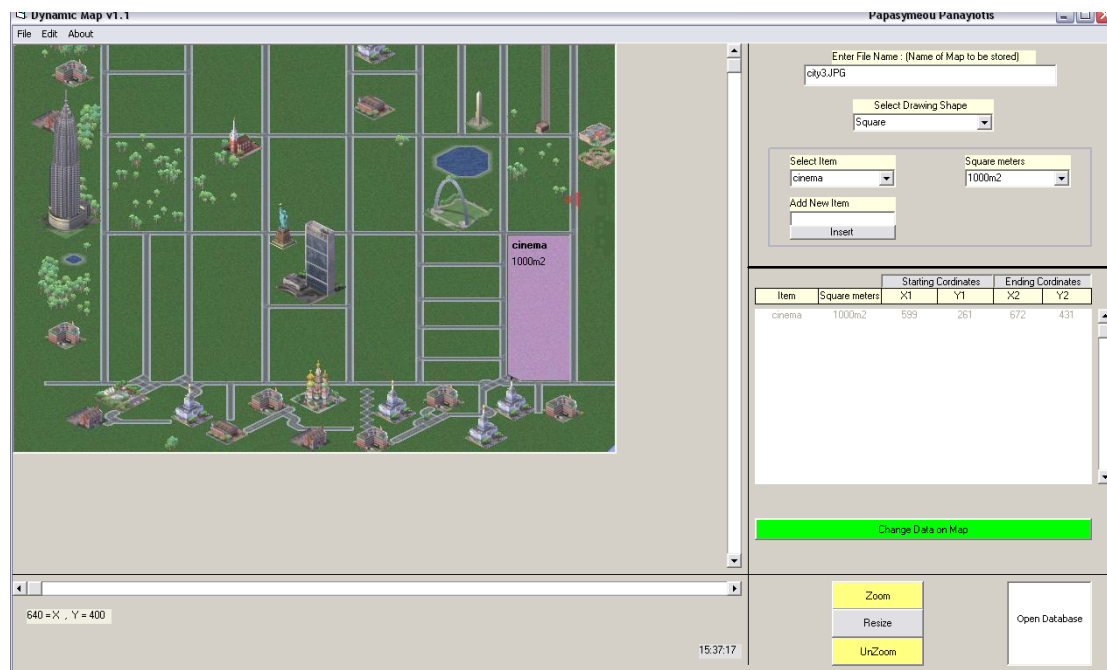
Στο πρόγραμμα υπάρχει ένα πεδίο με το όνομα 'Map Name', εκεί ο χρήστης εισάγει το όνομα του χάρτη όπου και καταχωρείται στη βάση δεδομένων. Τα δεδομένα που εισάγουμε στο χάρτη εμφανίζονται στα δεξιά της φόρμας. Από κάτω υπάρχει το κουμπί αλλαγής δεδομένων το οποίο και θα αναλύσουμε αργότερα.

### 3.3.2 Εισαγωγή δεδομένων στο χάρτη

Πρώτο βήμα είναι Επόμενο βήμα είναι να ορίσουμε το όνομα του χάρτη που θα αποθηκευτεί στη βάση. Στα δεξιά της φόρμα πάνω ψηλά υπάρχει μία περιοχή κειμένου όπου ο χρήστης εισάγει το όνομα του χάρτη. Κάνοντας κλικ ο χρήστης σε αυτή την περιοχή εμφανίζεται το όνομα του χάρτη που ανοίξαμε – αποτελεί προεπιλογή για σκοπούς διευκόλυνσης του χρήστη.

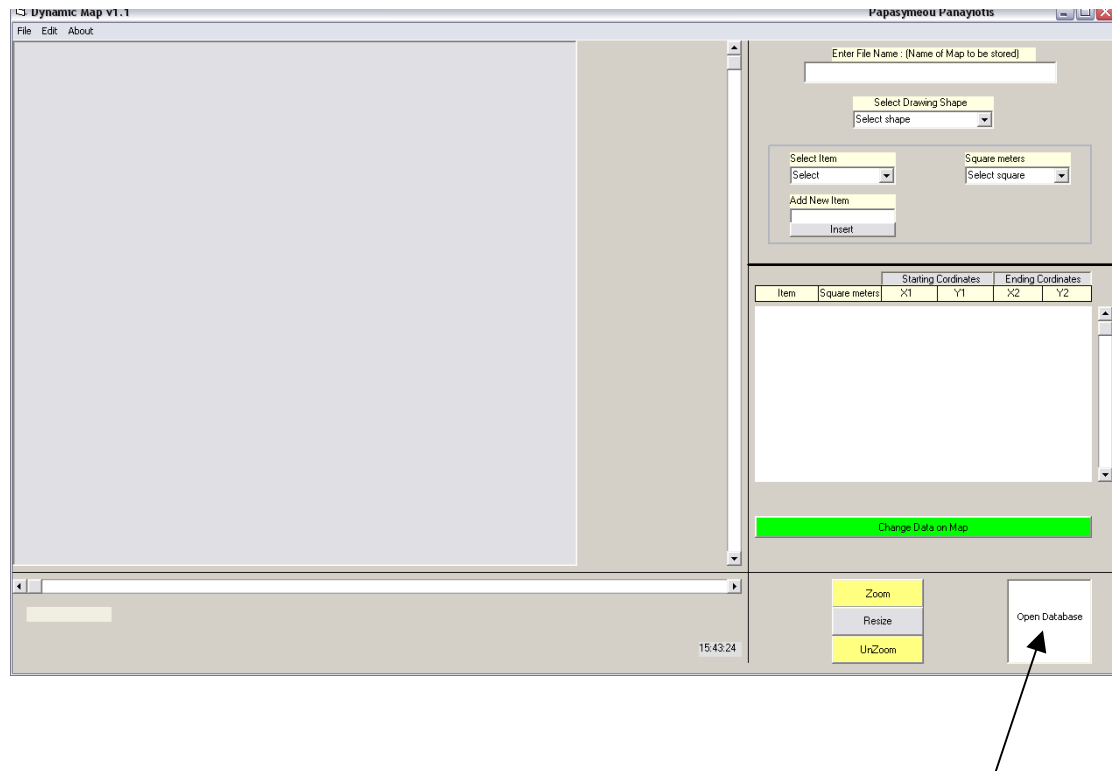
Στην συνέχεια επιλέγουμε σχεδιαστικό σχήμα από τις επιλογές 'Select Drawing Shape'. Υπάρχει η επιλογή του 'κύκλου' και του 'παράλληλογράμμου'. Ακολούθως επιλέγουμε το όνομα του κτιρίου από την επιλογή 'Select Item' και τα τετραγωνικά μέτρα από την επιλογή 'Select Meters' (Απαίτηση 4). Στην συνέχεια επιλέγουμε μία περιοχή στο χάρτη και εισάγουμε τα δεδομένα. (Απαίτηση 2)

Στην παρακάτω εικόνα υπάρχει το παραπάνω παράδειγμα.



### 3.3.3 Πρόσβαση στη Βάση Δεδομένων

Η πρόσβαση στη βάση γίνεται με τη χρήση του κουμπιού OLE .  
(Απαίτηση 7)



Κουμπί OLE

### 3.3.4 Τροποποιήσεις στα δεδομένα του χάρτη (Modifying Data)

(Απαίτηση 4)

Στα δεξιά της φόρμας μας έχει τοποθετηθεί ένα κουμπί με το όνομα ‘Change Data on Map’ . Πατώντας το ο χρήστης έχει την δυνατότητα να διαγράψει μία περιοχή στο χάρτη ή να την τροποποιήσει .

Αναλυτικά: επιλέγοντας αυτό το κουμπί θα εμφανιστεί στον χρήστη η παρακάτω φόρμα

Item	Square meters	x1	y1	x2	y2
Airport	10000m2	226	263	305	433

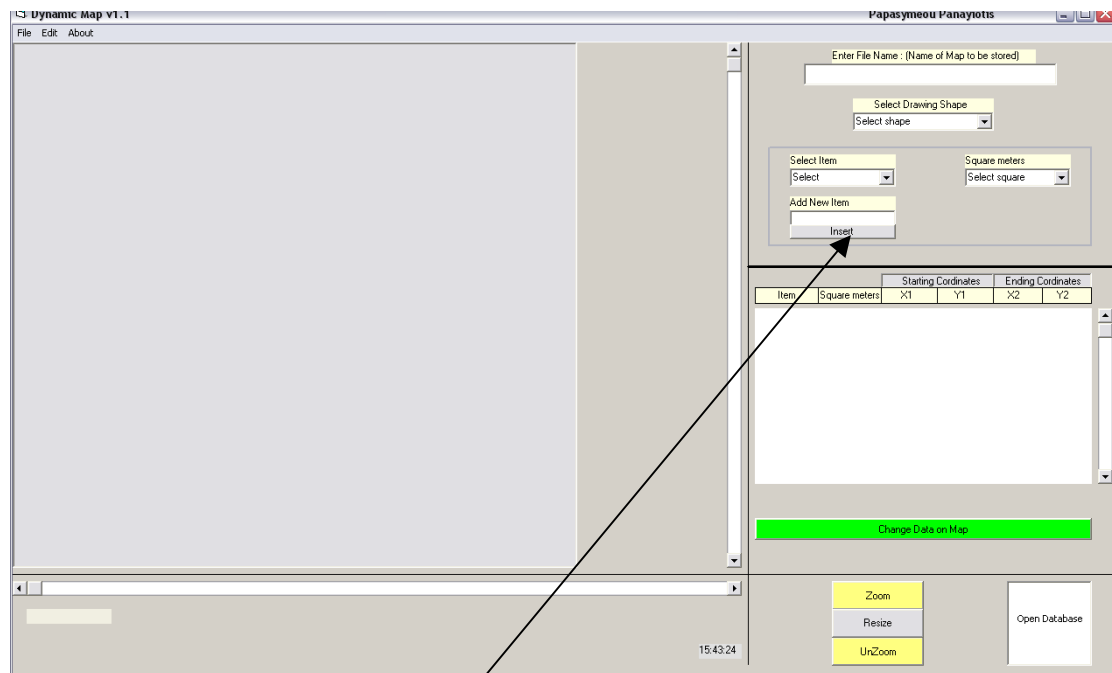
Εδώ ο χρήστης με την βοήθεια των δύο κουμπιών previous και next βλέπει τις εγγραφές που υπάρχουν στο χάρτη μία προς μία . Με αυτό τον τρόπο του παρέχεται η δυνατότητα να αντικαταστήσει το κτίριο ή τα τετραγωνικά μέτρα ή και τα δύο καθώς και να διαγράψει όλη την περιοχή που υπάρχει στο χάρτη .

Στην συνέχεια αφού ολοκληρώσει τις αλλαγές που χρειάζεται πατώντας το κουμπί 'Refresh Map' η παραπάνω φόρμα 'φεύγει' και στον χάρτη μας γίνονται όλες οι αλλαγές τις οποίες κάναμε . Φυσικό είναι αυτές οι αλλαγές να έχουν γίνει και στην βάση δεδομένων μας .

### **3.3.5 Πρόσθεση Επιλογών από το χρήστη**

(Απαίτηση 3)

Στο πρόγραμμα μας ο χρήστης μπορεί να προσθέσει τις δικές του επιλογές στα Combo boxes με δύο τρόπους . Ο ένας είναι να γράψει κατευθείαν στο κουτάκι αυτό που χρειάζεται και να κάνει ανανέωση της βάσης ούτως ώστε να καταχωρηθεί αυτό που θέλει .



‘ Insert ‘(Command7)

Ο δεύτερος τρόπος είναι μέσω του κουμπιού ‘ Insert ‘ που υπάρχει στις επιλογές στα δεξιά της φόρμας μας. Εκεί ο χρήστης εισάγει το αντικείμενο και στην συνέχεια πατάει το κουμπί ‘ Insert ‘ . Αυτό έχει σαν αποτέλεσμα στις επιλογές μας να δημιουργηθεί και το αντικείμενο μας . Αυτό γίνεται με τον παρακάτω κώδικα .

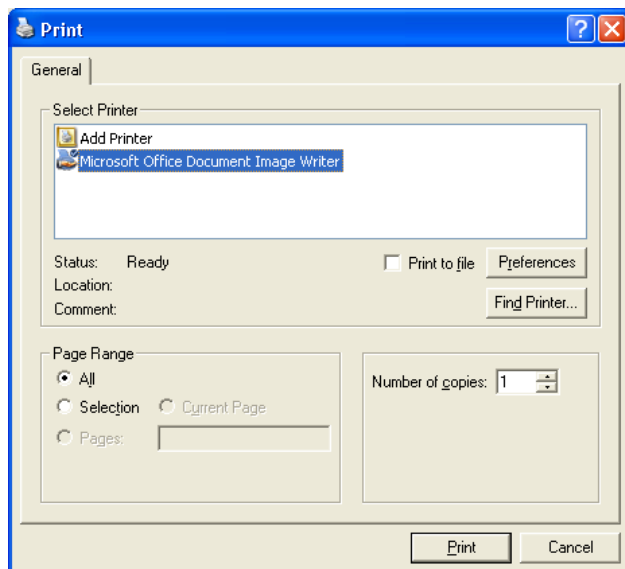
### 3.3.6 Η επιλογή Εκτύπωσης (Print)

Ωραία ,έχουμε φτιάξει το χάρτη μας όπως τον θέλουμε ,τον έχουμε υποθηκεύσει αλλά τώρα θέλουμε να τον εκτυπώσουμε .Πως γίνεται αυτό; Χρησιμοποιώντας των παρακάτω κώδικα μπορούμε να εκτυπώσουμε στον Default εκτυπωτή μας .

```
Private Sub Print_Click()
```

```
    CommonDialog2.ShowPrinter
```

```
End Sub
```

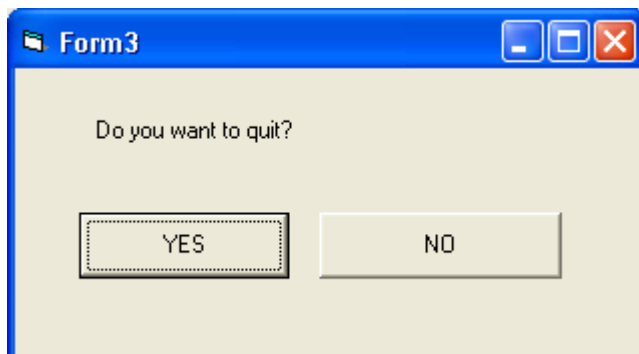


### **3.3.7 Η επιλογή Φόρτωσης (Load Picture)**

Έχουμε υποθηκεύσει το χάρτη και τώρα θέλουμε να τον φορτώσουμε και να τον επεξεργαστούμε και πάλι. Το θέμα εδώ είναι ότι τα εργαλεία πρέπει να φορτώνονται από τη βάση δεδομένων μας. Χρειάζεται να φορτώνουμε τις σωστές συντεταγμένες, τα σωστά Text boxes, και τις σωστές επιλογές που έχουν αποθηκευτεί από το χρήστη. Άρα χρειάζεται πρώτα να ανοίξουμε τη βάση και να κάνουμε έλεγχο. Ο έλεγχος γίνεται βάση το όνομα του χάρτη (Map Name), που υπάρχει στη φόρμα και χρειάζεται να εισαχθεί από το χρήστη. Το πρόγραμμα όταν πατηθεί η επιλογή Φόρτωσης κάνει έλεγχο εάν ο χάρτης ο οποίος ανοίγουμε είναι αυτός που υπάρχει πρώτος στη βάση μας αποθηκευμένος. Ας μην ξεχνάμε ότι δεν υπάρχουν τόσος πίνακες όσοι είναι και οι χάρτες οι οποίοι αποθηκεύονται. Εάν δεν είναι ο πρώτος χάρτης τότε ελέγχει τον επόμενο και ούτω καθεξής ώστε το όνομα του χάρτη που φορτώνω να υπάρχει και στη βάση. Όταν γίνει αυτή η αντιστοιχία τότε φορτώνει και τις συντεταγμένες καθώς και τις επιλογές που υπάρχουν κάτω από το όνομα του χάρτη που φορτώνουμε. Οι επιλογές μας φορτώνονται δυναμικά, δηλαδή έχοντας τις συντεταγμένες ξανασχεδιάζω τις επιλογές μου καθώς όταν έκανα αποθήκευση δεν έχουν αποθηκευτεί ως φωτογραφία στην εικόνα-χάρτη.

### 3.3.8 Η επιλογή Εξόδου (Exit)

Έχουμε σχεδιάσει μία τρίτη φόρμα (Form3) όπως φαίνεται παρακάτω .



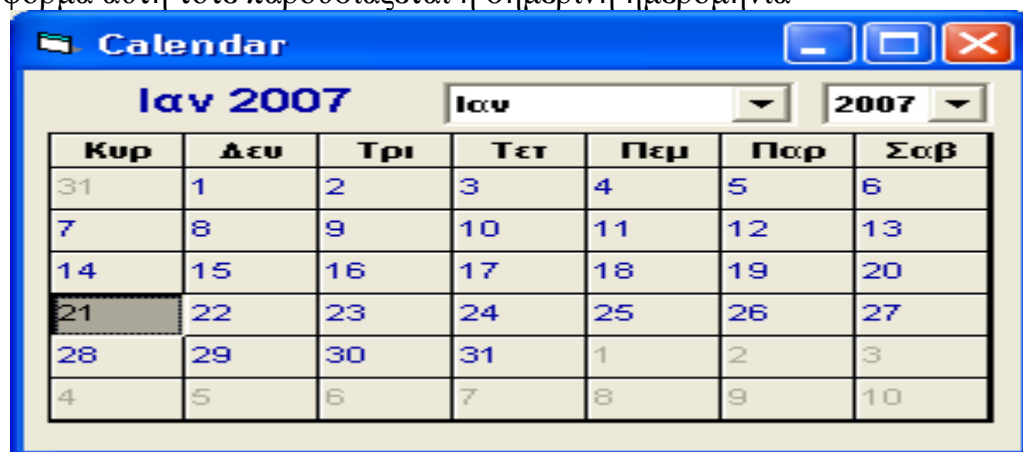
Όταν ο χρήστης επιλέξει στο μενού Exit τότε εμφανίζεται η παραπάνω φόρμα .

### 3.3.9 Επιλογή Edit – Delete All

Η επιλογή Delete All διαγράφει τα περιεχόμενα όλα της εικόνας 1 καθώς και την ίδια την εικόνα . Είναι χρήσιμη η επιλογή κατά τη διάρκεια του φορτώματος μιας εικόνας καθώς ένα δουλεύουμε σε ένα χάρτη , τελειώσουμε και στην συνέχεια θελήσουμε να φορτώσουμε ένα άλλο χάρτη τότε αναγκαστικά πρέπει να χρησιμοποιήσουμε αυτή την επιλογή και αυτό γιατί το φόρτωμα της εικόνας γίνεται δυναμικά . Εάν δεν γίνει αυτό τότε υπάρχει περίπτωση να εμφανιστούν και οι επιλογές εργαλεία που υπήρχαν και στην εικόνα την οποία επεξεργαζόμασταν πριν γίνει η φόρτωση καινούργιου χάρτη . Ο κώδικας που το κάνει αυτό είναι ο ακόλουθος .

### 3.3.10 Επιλογή About – Calendar

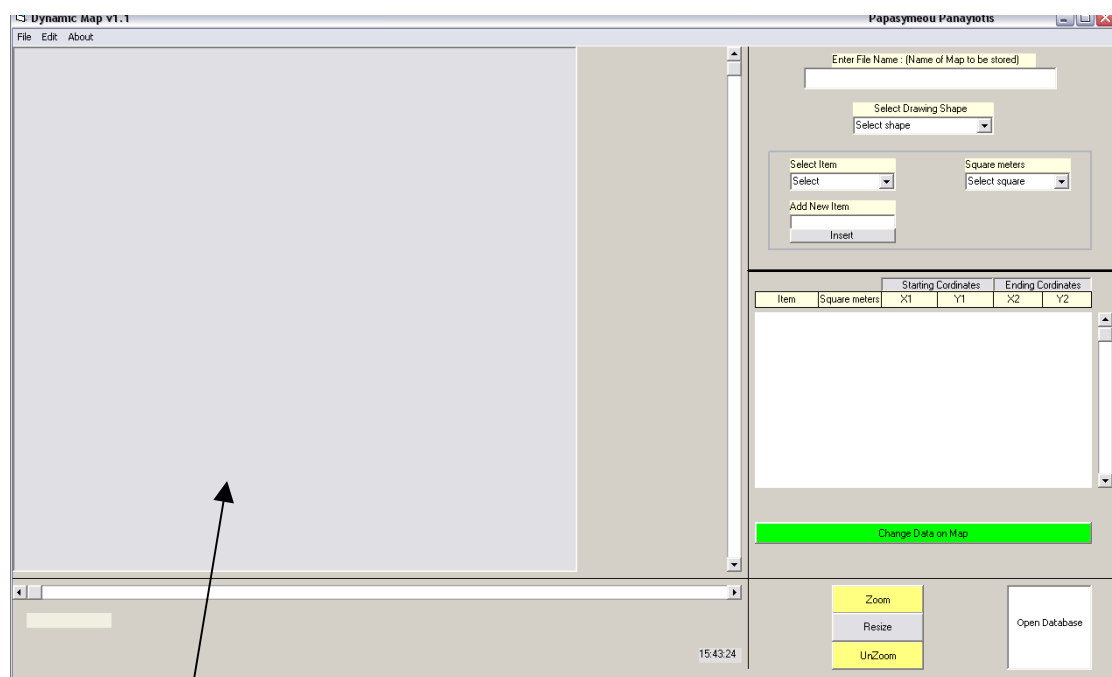
Πρόκειται για ένα ημερολόγιο που έχει τοποθετηθεί στο πρόγραμμα και συγκεκριμένα στη φόρμα2 . (Form2) . Όταν ο χρήστης κάνει κλικ στην φόρμα αυτή τότε παρουσιάζεται η σημερινή ημερομηνία



### 3.4 Τεχνική Ανάλυση Λειτουργιών

Ας δούμε τώρα τι κρύβεται πίσω από το πρόγραμμα .

Το πρόγραμμα μας περιέχει μία φόρμα (Form1) στην οποία περιέχονται όλα τα εργαλεία που χρειαζόμαστε. Με την βοήθεια του Toolbox ζωγραφίζουμε στην αρχή ένα κουτί φωτογραφιών όπως ονομάζεται ( Picture Box) . Σε αυτό φτιάχνουμε τις συναρτήσεις που χρειαζόμαστε . Εδώ χρησιμοποιούμε τις 'Private Sub Picture1\_MouseMove', 'Private Static Sub Picture1\_MouseDown' και 'Private Sub Picture1\_MouseUp' .



Picture Box

Ας δούμε αναλυτικά τον κώδικα , και τις λειτουργίες του προγράμματος

### 3.4.1 Κίνηση Mouse

Μας ενδιαφέρει μόνο η κίνηση στο Picture Box , εκεί όπου ουσιαστικά εργαζόμαστε . Αναλυτικά ο κώδικας είναι :

```
Private Sub Picture1_MouseMove(Button As Integer, Shift
As Integer, x As Single, y As Single)

    If Combo1.Text = "Square" Then
        If Button = vbLeftButton Then
            blnDrawFlag = True
            Set Picture1.Picture = Picture18.Picture
            Combo1.Refresh
            Picture1.Line (sX, sY)-(x, y), vbBlack, B
            If blnDrawFlag Then Picture1.Line -(x, y)
        Else
            sX = x
            sY = y
        End If
    ..... (βλ. Παράρτημα ΙΙ)
End If
```

Στην φόρμα μας που ονομάζουμε Form1 έχουμε τοποθετήσει – σχεδιάσει ένα combo box επιλογής σχεδίου στο χάρτη . Έχει τις επιλογές : ‘Κύκλος’, ‘Ορθογώνιο’ . Αυτά αποτελούν και τους μόνους τρόπους με τους οποίους ο χρήστης μπορεί να τοποθετήσει ένα αντικείμενο στην οθόνη .

Η εντολή : If Button = vbLeftButton Then

Εδώ λέω εάν πατηθεί το αριστερό κουμπί του ‘ ποντικιού’ τότε το περιεχόμενο της φωτογραφίας 1 να γίνει ίσο με το περιεχόμενο της φωτογραφίας 2(ή 18 στην συγκεκριμένη περίπτωση) . Αυτό γίνεται για σχεδιαστικούς λόγους . Για να είμαι σε θέση να ζωγραφίσω στην εικόνα 1 πρέπει να υπάρχει και μία δεύτερη που θα αντιγράφει συνεχώς το περιεχόμενο της αλλιώς δεν θα αποθηκεύεται το προηγούμενο σχέδιο . Η παρακάτω εντολή Picture1.Line (sX, sY)-(X, y), vbBlack, B είναι αυτή με την οποία ζωγραφίζουμε ελεύθερα (free hand) στην εικόνα .



### **3.4.2 Εμφάνιση συντεταγμένων**

Στον παραπάνω κώδικα συναντάμε την εντολή: `Label1.Caption = GetCursor`. Στο πρόγραμμά μας έχουμε ζωγραφίσει ένα Label το `label1`. Με το `Caption` εννοούμε την τιμή ή καλύτερα τον τίτλο που θα περιέχει αυτό το Label. Η Εντολή `GetCursor`, είναι μία συνάρτηση που καλούμε εδώ για να μας δώσει τις συντεταγμένες `x`, `y` και θα την αναλύσουμε παρακάτω.

### **3.4.3 Εάν πατηθεί το Mouse**

Η συνάρτηση που είδαμε λίγο πριν είναι η συνάρτηση `Mouse Move`. Παρά το γεγονός ότι χρησιμοποιούμε την συνάρτηση αυτή για να σχεδιάζουμε, και ο λόγος είναι ότι στην συνάρτηση αυτή έχουμε συντεταγμένες τις οποίες χρησιμοποιεί κατά την εκτέλεση του προγράμματος ο χρήστης, χρειαζόμαστε μία δεύτερη συνάρτηση που λέγεται `Mouse Down`. Σε αυτή την συνάρτηση θα δούμε πως δημιουργούμε δυναμικά `text boxes` πως ανοίγουμε τη βάση δεδομένων μας και πως εισάγουμε δεδομένα σε αυτήν.

### **3.4.4 Βάση Δεδομένων**

Στην συνέχεια αυτής της συνάρτησης χρησιμοποιούμε τις εντολές για την σύνδεση του προγράμματος με τη βάση. Γιατί χρειάζεται βάση;

Ο λόγος για ύπαρξη βάσης είναι εξόφθαλμος καθώς είναι ο μόνος τρόπος για να υποθηκεύσεις δεδομένα του προγράμματος. Εάν θέλαμε ένα πρόγραμμα το οποίο θα το χρησιμοποιούσαμε μόνο μία φορά ή σχεδιάζαμε κάτι προσωρινό που δεν θα το χρησιμοποιούσαμε ξανά τότε δεν θα ήταν αναγκαία η ύπαρξη της βάσης. Για το λόγο όμως ότι θα θέλαμε στο μέλλον να ασχοληθούμε ξανά με το χάρτη που έχουμε φτιάξει η ύπαρξη της βάσης κρίνεται ως αναγκαία. Ας δούμε αναλυτικά της εντολές του προγράμματος για την σύνδεση με τη βάση

```
Set cn = New ADODB.Connection
    cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source= C:\vb6\DB1.mdb"
cn.Open
```

```
Set RS = New ADODB.Recordset 'as we did with the
connection
    RS.Open "tbl_master", cn, adOpenKeyset,
adLockPessimistic, adCmdTable
```

```
Set cn = New ADODB.Connection
```

```

        cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source= C:\vb6\DB1.mdb"

```

```
cn.Open
```

```

Set RS = New ADODB.Recordset 'as we did with the connection
    RS.Open "tbl_master", cn, adOpenKeyset, adLockPessimistic, adCmdTable
Set ps = New ADODB.Recordset 'as we did with the connection
    ps.Open "Cord", cn, adOpenKeyset, adLockPessimistic, adCmdTable

```

Η σύνδεση με τη βάση γίνεται πάντα με τις ακόλουθες εντολές :

```

Set cn = New ADODB.Connection
    cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source= C:\vb6\DB1.mdb"

```

Και στην συνέχεια με την εντολή `cn.Open` την ανοίγουμε . Το `"Data Source= C:\vb6\DB1.mdb"` δεν είναι τίποτα άλλο από το μονοπάτι (path) όπου υπάρχει η βάση την οποία δημιουργήσαμε .

Αναλυτικά ο κώδικας :

```

Private Static Sub Picture1_MouseDown(Button As Integer, Shift As Integer, x
As Single, y As Single)

```

```

    Set RS = New ADODB.Recordset
    RS.Open "tbl_master", cn, adOpenKeyset, adLockPessimistic, adCmdTable

```

```

    If Button = vbLeftButton Then

```

```

        .....

```

```

        Set Txts(j) = Me.Controls.Add("Vb.TextBox", "cmdButton" &
Me.Controls.Count, Picture1)

```

```

        With Txts(j)

```

```

            ....

```

```

        End With

```

```

    If Not ps.EOF Then

```

```

        Text3.Text = Text3.Text & ps.Fields("X Coordinates").Value & vbCrLf

```

```

        Text4.Text = Text4.Text & ps.Fields("Y Coordinates").Value & vbCrLf

```

```

        If h = CommonDialog1.FileTitle Then

```

```

            ps.MoveNext

```

```

        End If

```

```
End If
End Sub
(βλ. Παράρτημα II)
```

### 3.4.5 Η σύνδεση με τη βάση

Το άνοιγμα της βάσης γίνεται με τις εντολές :

```
Set cn = New ADODB.Connection
    cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\vb6\DB1.mdb"
cn.Open
```

Τις παραπάνω εντολές θα τις χρησιμοποιούμε συχνά στο πρόγραμμα αφού πρόκειται για δυναμικά δεδομένα , που σημαίνει ότι χρειαζόμαστε μία βάση για να αποθηκεύουμε τις πληροφορίες μας συνεχώς .

Για να ανοίξουμε κάποιο συγκεκριμένο πίνακα στην συνέχεια έχουμε τις παρακάτω εντολές :

```
Set RS = New ADODB.Recordset 'as we did with the connection
    RS.Open "tbl_master", cn, adOpenKeyset, adLockPessimistic, adCmdTable
```

Πως γίνεται τώρα η αποθήκευση των δεδομένων στη βάση .

Χρησιμοποιώντας τις εντολές :

With

```
.AddNew
.Fields("Quantity") = Txts(i).Text
.Fields("Item") = Txt(i).Text
.Fields("Map Name") = Text9.Text 'CommonDialog1.FileName
.Fields("shape") = Combo1.Text
.Update
End With
```

Ανοίγουμε τον πίνακα RS όπως τον ονομάζουμε στο πρόγραμμα μας που αντιστοιχεί με τον πίνακα tbl\_master στη βάση δεδομένων μας . Με την εντολή : .AddNew Προσθέτουμε μία νέα εγγραφή στον πίνακα tbl\_master . Με τις εντολές : .Fields .Προσθέτουμε στο κατάλληλο πεδίο της εγγραφής τα δεδομένα που θέλουμε .

### 3.4.6 Εμφάνιση δεδομένων στην φόρμα

Όλα τα δεδομένα που εισάγουμε στο πρόγραμμα και συγκεκριμένα σε ένα χάρτη εμφανίζονται στα δεξιά του προγράμματος . Αυτό γίνεται με τις εντολές (Απαίτηση 6)

```
If Not RS.EOF Then
    Text1.Text = Text1.Text & RS.Fields("Item").Value & vbCrLf
    Text2.Text = Text2.Text & RS.Fields("Quantity").Value & vbCrLf
```

.....

```
End If
```

(βλ.Παράρτημα ΙΙ)

### 3.4.7 Δημιουργία Δυναμικών Text Boxes

Κατά την εκτέλεση του προγράμματος καθώς ο χρήστης πατήσει το αριστερό κουμπί του mouse παράλληλα με το σχέδιο που κατασκευάζει δημιουργούνται δύο Text Boxes . Αυτό γίνεται με τις παρακάτω εντολές :

```
Set Txt(j) = Me.Controls.Add("VB.Textbox", "cmdButton" &
Me.Controls.Count, Picture1)
```

```
With Txt(j)
    .Left = x
    .Top = y
    .Width = 60
    .Height = 5
    .Text = Combo7.Text
    .Visible = True
    .BorderStyle = 0
    .BackColor = &HFF8080
    .Alignment = 2
    .SetFocus
```

```
End With
```

.....

```
End With
```

(βλ. Παράρτημα ΙΙ)

### 3.4.8 Συντεταγμένες χ,ψ

Public Function GetCursor()

```

    GetCursorPos mousexy
    GetCursor = (mousexy.X - wndrect.Left) & " = X , Y = " & (mousexy.y -
wndrect.Top)

```

```

    xi = mousexy.X
    yi = mousexy.y

```

End Function

Με αυτή τη συνάρτηση είμαστε σε θέση να εμφανίσουμε στην οθόνη τις συντεταγμένες χ, ψ .Η εντολή που το κάνει αυτό ουσιαστικά είναι η :

```

GetCursor = (mousexy.X - wndrect.Left) & " = X , Y = " & (mousexy.y -
wndrect.Top)

```

Παραπάνω συναντήσαμε την εντολή Label1.Caption= GetCursor . Το περιεχόμενο δηλαδή του Label1 ή καλύτερα ο τίτλος γίνεται ίσως με το GetCursor όπου η συνάρτηση επιστρέφει δύο τιμές .Την mousexy.X - wndrect.Left για την τιμή του χ και την mousexy.y - wndrect.Top για την τιμή του ψ .

Οι επόμενες δύο συναρτήσεις είναι πανομοιότυπες και χρησιμοποιούνται για τις επιστροφές των τιμών χ , ψ αντίστοιχα .

Για την τιμή του χ :

Public Function GetCursorX()

```

.....
    GetCursorPos mousexy
    GetCursorX = mousexy.X - wndrect.Left

```

```

    xi = mousexy.X
    yi = mousexy.y

```

End Function

Για την τιμή του ψ :

Public Function GetCursorY()

.....

```
GetCursorPos mousexy  
GetCursorY = mousexy.y - wndrect.Top
```

```
xi = mousexy.X  
yi = mousexy.y
```

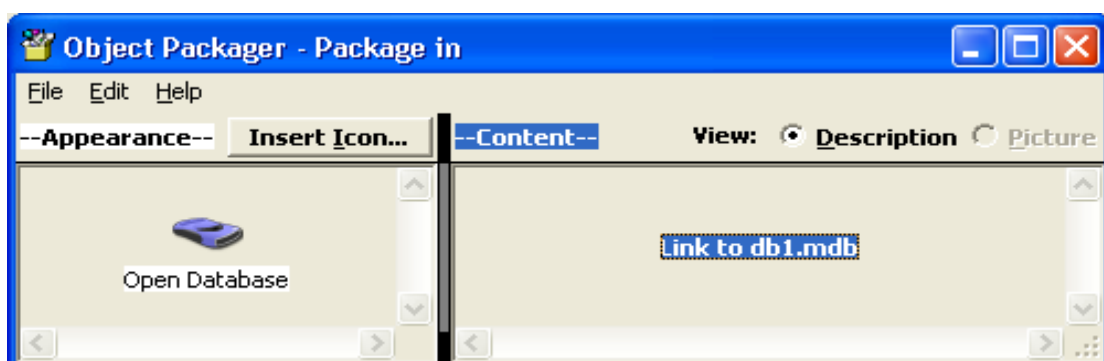
End Function

(βλ. Παράρτημα II)

### 3.4.9 OLE DB

Τα αρχικά OLE σημαίνουν : Object Linking and Embedding . Και αποτελεί ουσιαστικά τον τρόπο ή την μέθοδο με την οποία μπορούμε να ανοίξουμε τη βάση κατευθείαν από το πρόγραμμα μας . Εκτός από τη βάση βέβαια μπορούμε να ανοίξουμε ,με τις κατάλληλες ρυθμίσεις και : Microsoft Word και Excel.

Κατά τη διάρκεια των ρυθμίσεων εμφανίζεται το παρακάτω σχεδιάγραμμα .



### **3.4.10 Τροποποίηση δεδομένων**

#### Το κουμπί 'Next'

```
Private Sub Command1_Click()
```

```
Set Text1.DataSource = RS  
Text1.DataField = "Item"
```

```
.....  
Set Text6.DataSource = ts  
Text6.DataField = "y2"
```

```
End Sub
```

(βλ. Παράρτημα II)

#### Το κουμπί 'Previous'

```
Private Sub Command2_Click()
```

```
.....
```

```
Set Text1.DataSource = RS  
Text1.DataField = "Item"
```

```
Set Text2.DataSource = RS  
Text2.DataField = "Quantity"
```

```
.....  
End Sub
```

(βλ. Παράρτημα II)

Το κουμπί 'Delete'

```
Private Sub Command3_Click()
```

```
.....  
ps.MoveNext
```

```
.....  
ts.MoveNext
```

```
.....  
End Sub
```

(βλ. Παράρτημα II)

Το κουμπί 'Refresh'

```
Private Sub Command4_Click()
```

```
a = Form1.Text9.Text  
Form4.Visible = True  
Unload Form1  
Load Form1  
Form1.Visible = True
```

```
cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _  
"Data Source= C:\vb6\DB1.mdb" 'this is the connection string explained in  
the notes section.
```

```
cn.Open
```

```
x = ps.Fields("X Coordinates").Value
```

```
.....
```

```
If p = "Square" Then
```

```
.....
```

```
End If
```

```
.....  
End Sub
```

(βλ. Παράρτημα II)



Τα κουμπιά 'Replace With'

```
Private Sub Command5_Click()
```

```
    RS.Fields("Item").Value = Combo1.Text  
    RS.Update
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
    RS.Fields("Quantity").Value = Combo2.Text  
    RS.Update
```

```
End Sub
```

**3.4.11 Scroll bars**

Στο πρόγραμμα χρησιμοποιούμε δύο Scroll bars που σκοπό έχουν την μετακίνηση του χάρτη στο σημείο που θέλουμε. Έχουμε ένα οριζόντιο (HScroll) και ένα κάθετο (VScroll). Ο κώδικας για αυτά είναι ο ακόλουθος :

```
Private Sub VScroll1_Change()
```

```
    Picture1.Top = -VScroll1.Value
```

```
    VScroll1.Min = 1
```

```
    VScroll1.Max = 100
```

```
End Sub
```

```
Private Sub HScroll1_Change()
```

```
    Picture1.Left = -HScroll1.Value
```

```
    HScroll1.Min = 1
```

```
    HScroll1.Max = 100
```

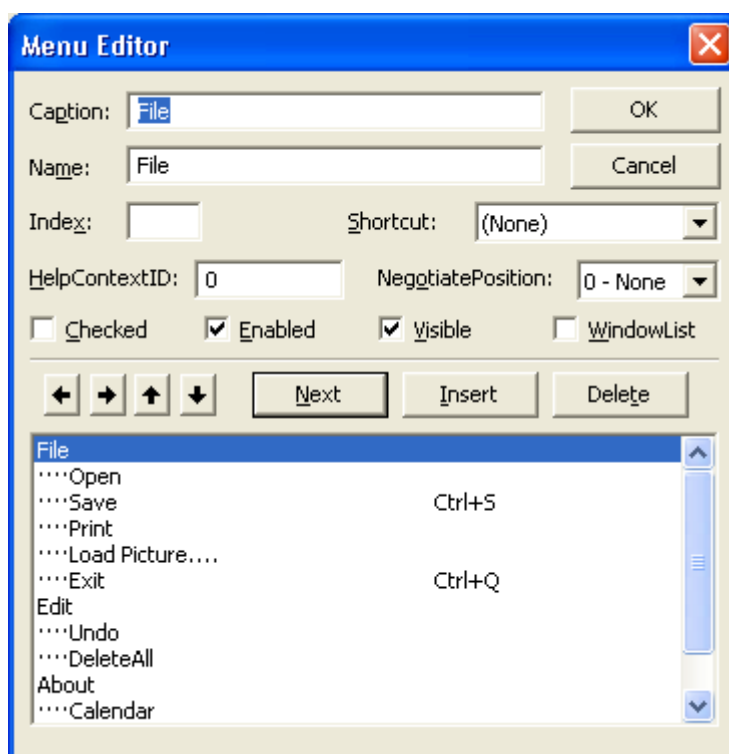
```
    HScroll1.LargeChange = 10
```

End Sub

### 3.4.12 Menu bar

Στο πρόγραμμα μας έχουμε την παρουσία του μενού (menu bar) . Για να δημιουργήσουμε ένα μενού πατάμε δεξί κλικ στην φόρμα μας και επιλέγουμε Menu editor.

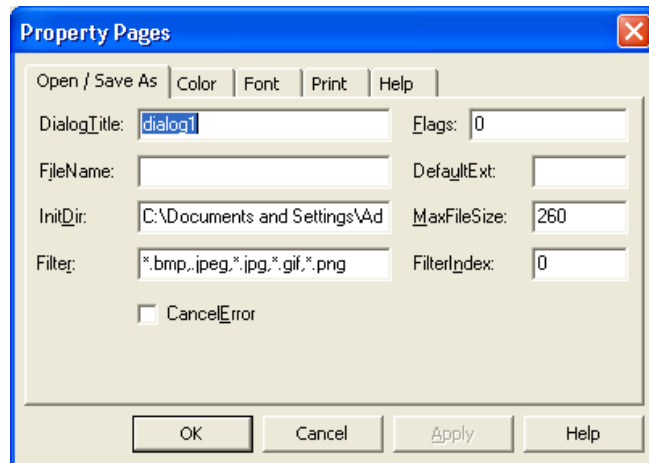
Τότε εμφανίζεται το ακόλουθο σχεδιάγραμμα όπου καθορίζουμε τον τρόπο και τι θέλουμε να εμφανίζεται σε αυτό το μενού .



#### Επιλογές Μενού – Ανάλυση

#### Εμφάνιση Μενού – Άνοιγμα Αρχείου

Στο πρόγραμμα μας και συγκεκριμένα στη φόρμα μας έχουμε τοποθετήσει ένα κουμπί το οποίο δεν είναι εμφανίσιμο κατά τη διάρκεια που τρέχει το πρόγραμμα και λέγεται 'CommonDialog', όπως φαίνεται και στο παρακάτω διάγραμμα :



Στον κώδικα μας υπάρχουν δύο τρόποι οι οποίοι μπορεί να εμφανιστεί το μενού επιλογών ( άνοιγμα χάρτη ) .Ας δούμε τον κώδικα .

Private Sub Open\_Click()

‘ Πρώτος τρόπος

With CommonDialog1

```
.DialogTitle = "Open"
.CancelError = False
.Filter = "All Files (*.*)|*.*"
.ShowOpen
```

'End With

‘ Δεύτερος τρόπος

```
CommonDialog1.DialogTitle = "Select a file to open"
CommonDialog1.ShowOpen
Picture1.Picture = LoadPicture(CommonDialog1.FileName)
```

End Sub

Στο διάλογο(CommonDialog) αυτό μπορούμε να τοποθετήσουμε φίλτρα τα οποία αποτελούν τις κωδικοποιήσεις χαρτών των οποίων το πρόγραμμα μας θα ανοίγει .

### 3.4.13 Η επιλογή Αποθήκευσης (Save)

Αφού έχουμε σχεδιάσει και τοποθετήσει τις επιλογές μας στο χάρτη , έφτασε η ώρα όπου θέλουμε να υποθηκεύσουμε όλα όσα έχουμε κάνει . Και πάλι κάνουμε χρήση του 'CommonDialog' .Ας έχουμε υπόψη ότι στο πρόγραμμα υπάρχουν δύο κουτιά φωτογραφιών (Picture boxes) για σχεδιαστικούς λόγους και ότι μόνο το ένα είναι ορατό από το χρήστη. Κατά τη διάρκεια που τρέχει το πρόγραμμα τα σχεδιαγράμματα αντιγράφονται συνεχώς από το ένα κουτί στο άλλο επομένως για να υποθηκεύσουμε και το τελευταίο σχεδιάγραμμα χρειάζεται και στον κώδικα της αποθήκευσης να αντιγράψουμε τα περιεχόμενο του ενός κουτιού στο άλλο . Ας δούμε τώρα τον κώδικα .

```
Private Sub Save_Click()
```

```
    CommonDialog1.ShowSave
```

```
    Picture18.Picture = Picture1.Image
```

```
    Picture1.Picture = Picture18.Picture
```

```
    MsgBox "File to save: " & CommonDialog1.FileName
```

```
    SavePicture Picture1.Picture, CommonDialog1.FileName
```

```
    On Error GoTo Cancel
```

```
    Cancel:
```

```
    MsgBox "File Error!"
```

```
End Sub
```

### 3.4.14 Η επιλογή Φόρτωσης (Load)

```
Private Sub Loads_Click()
```

```
.....
    CommonDialog1.DialogTitle = "Select a file to open"
```

```
        ts.MoveNext
```

```
    End If
```

```
    If Not h = CommonDialog1.FileTitle Then
```

```
        GoTo x:
```

```
    End If
```

```
    If RS.EOF Then
```

```
End If
.....
End If

Next i
x:
End Sub

(βλ.Παράρτημα ΙΙ)
```

### **3.4.15 Έξοδος Από το πρόγραμμα**

```
Private Sub Exit_Click()

Form3.Visible = True

End Sub
```

Εάν επιλεγθεί το YES τότε το πρόγραμμα τερματίζεται εάν επιλέξει NO τότε συνεχίζει το πρόγραμμα να τρέχει κανονικά και η φόρμα3 ,φόρμα εξόδου, τερματίζεται . Ας δούμε τώρα τον κώδικα των δύο επιλογών.

#### Εάν επιλεγθεί το YES

```
Private Sub Command1_Click()

Unload Form1
Unload Form2

End Sub
```

#### Εάν επιλεγθεί το NO

```
Private Sub Command2_Click()

Form1.Visible = True
Form3.Visible = False

End Sub
```

### **3.4.16 Διαγραφή όλων των στοιχείων στη Φόρμα**

```
Private Sub DeleteAll_Click()

    Picture1.Picture = Picture
    Set Picture1.Picture = Nothing

    Form4.Visible = True
    Unload Form1

    Clipboard.SetData Picture18.Picture
    Clipboard.SetData Picture1.Picture
    Cls

    Load Form1
    Form1.Visible = True
    Unload Form4

    .....
End Sub
(βλ. Παράρτημα ΙΙ)
```

### **3.4.17 Επιλογή About – Ημερολόγιο**

```
Private Sub Calendar1_Click()

    Calendar1.Today

End Sub
```

### **3.4.18 Επιλογή About – Help**

Κάνοντας κλικ ο χρήστης σε αυτή την επιλογή του δίνονται γενικές πληροφορίες για το πρόγραμμα το οποίο 'τρέχει'. Ο κώδικας είναι ο ακόλουθος :

```
Private Sub Help_Click()

    CommonDialog1.HelpCommand = &HB
    CommonDialog1.HelpFile = "c:\windows\help\windows.hlp"
    CommonDialog1.ShowHelp

End Sub
```

### 3.4.19 Χρονιστές

Στο πρόγραμμα μας υπάρχουν δύο χρονιστές (Timers) οι οποίοι δεν φαίνονται οι ίδιοι καθαυτοί αλλά το αποτέλεσμα τους . Τον πρώτο τον χρειαζόμαστε για να ανανεώνονται οι συντεταγμένες στο πρόγραμμα μας και είναι ο ακόλουθός :

```
Private Sub Timer1_Timer()
```

```
Label1.Caption = GetCursor
```

```
End Sub
```

Τον δεύτερο τον χρειαζόμαστε για την παρουσία της ώρας στο πρόγραμμα και είναι ο ακόλουθος :

```
Private Sub Timer2_Timer()
```

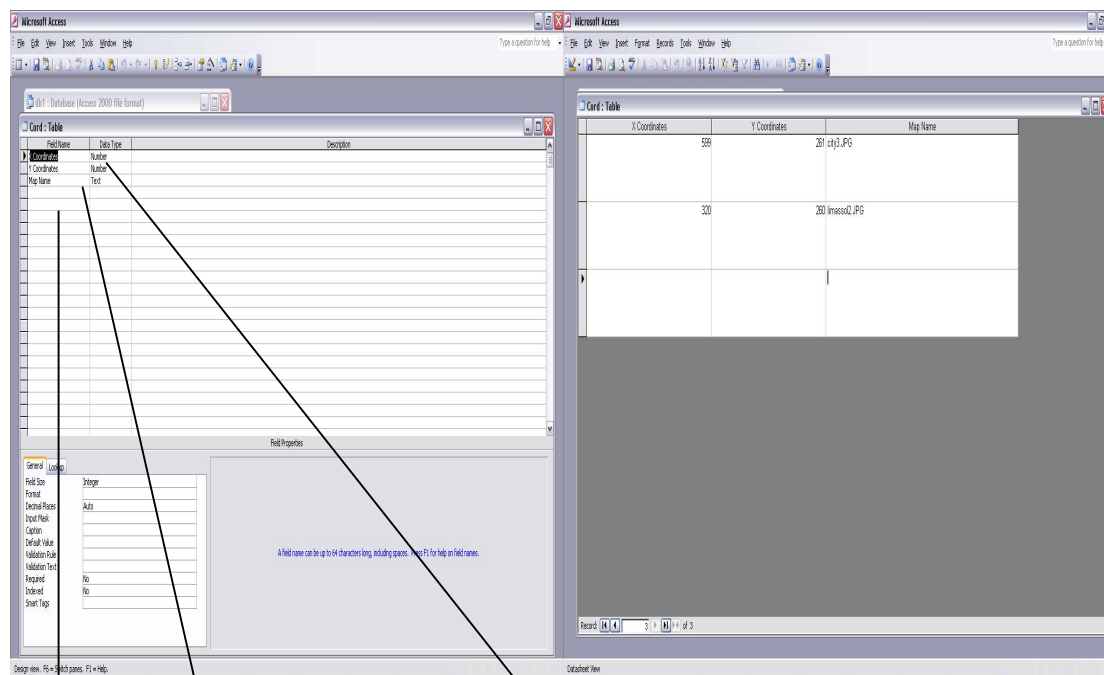
```
Label4.Caption = Time$
```

```
End Sub
```

Το Label4 είναι εκεί όπου παρουσιάζεται η ώρα .

### 3.5 Ανάλυση & Σχεδιασμός Βάσης Δεδομένων

Προτού προχωρήσουμε ας δούμε τώρα τι περιέχει η βάση μας .Καταρχάς ονομάζουμε την βάση μας ως : DB1.mdb . Μέσα σε αυτή δημιουργούμε3 tables όπως ονομάζονται . Ένα με το όνομα ' Cord' , ένα με το όνομα 'tbl\_master' και ένα με το όνομα 'third' . Στον πρώτο περιέχουμε τρία πεδία . Το όνομα του χάρτη και οι συντεταγμένες χ , ψ . Αναλυτικά στο παρακάτω σχήμα.



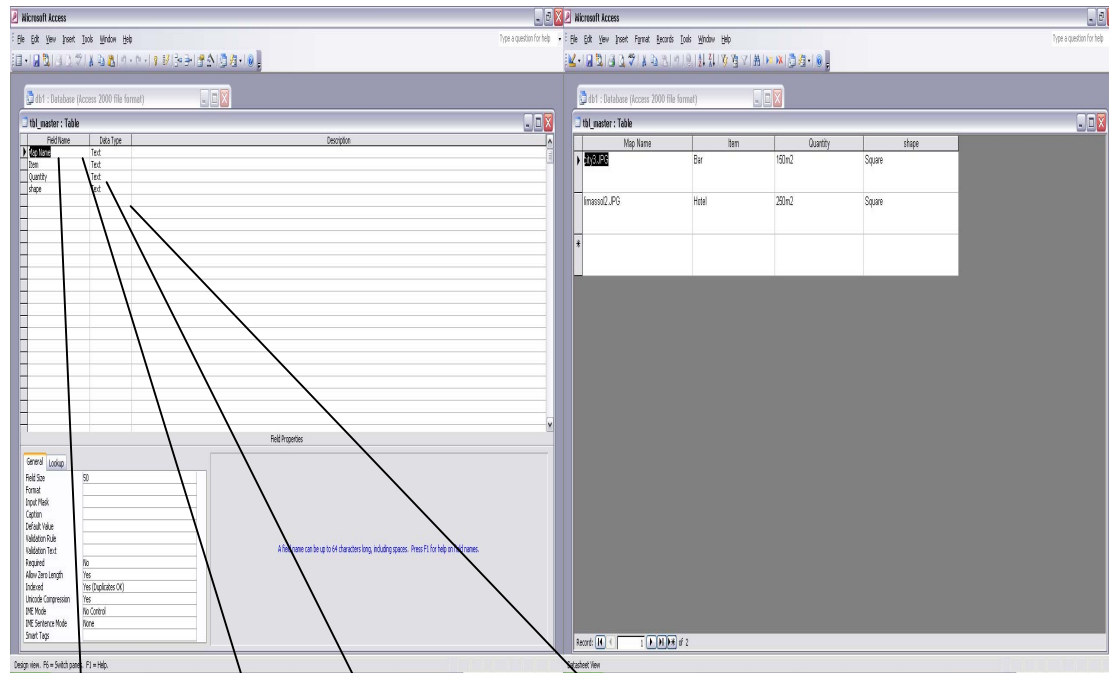
Υ Συντεταγμένες

Όνομα Χάρτη

X Συντεταγμένες

Στον δεύτερο έχουμε το όνομα του χάρτη , αντικείμενο , ποσότητα και σχεδιαστικό σχήμα . Όπως φαίνεται και στο παρακάτω σχήμα :





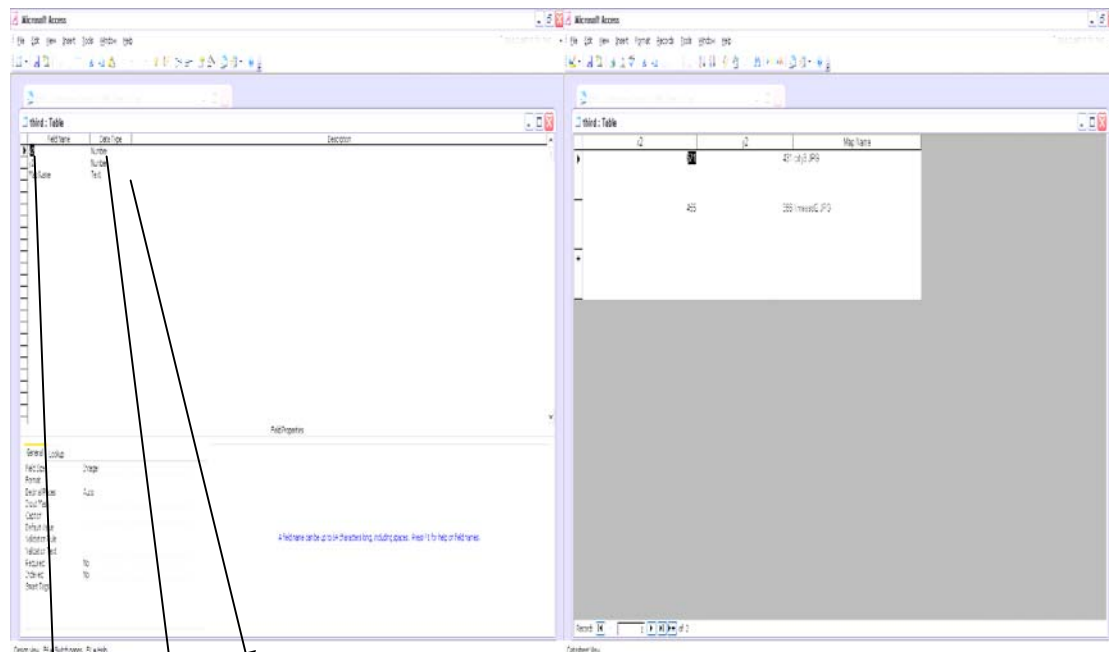
Όνομα Χάρτη

Αντικείμενο

Τετραγωνικά Μέτρα

Σχεδιαστικό σχήμα

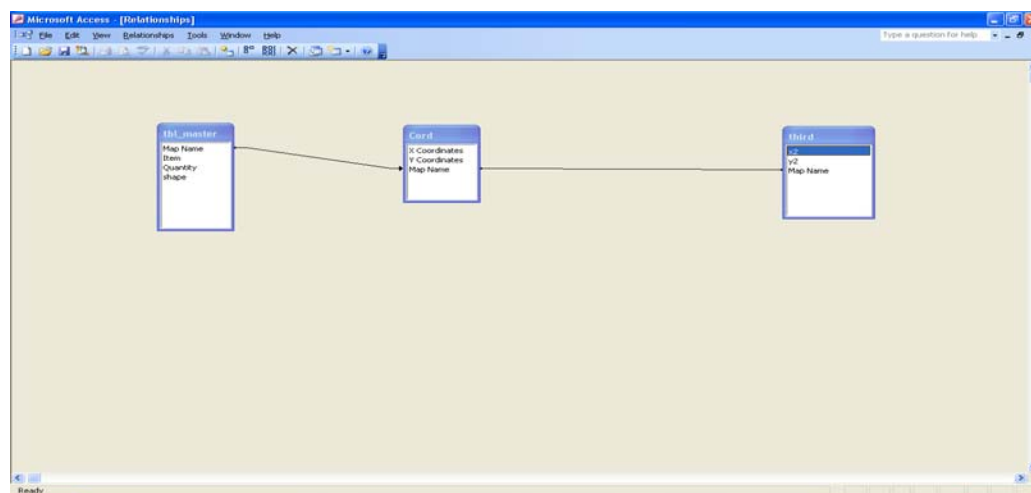
Στον τρίτο πίνακα έχουμε τις συντεταγμένες ‘τέλους’, εκεί που τελειώνει δηλαδή το σχήμα μας και το όνομα του χάρτη .



Όνομα Χάρτη

Y2 Συντεταγμένες  
X2 Συντεταγμένες

Ας έχουμε υπόψη ότι τα δεδομένα ενός σχήματος και μίας πληροφορίας στο χάρτη αποθηκεύονται και στους τρεις πίνακες . Αυτό σημαίνει ότι πρέπει η ανανέωση και η εισαγωγή να γίνεται ταυτόχρονα και στους τρεις πίνακες . Και στους τρεις πίνακες έχουμε ένα κοινό πεδίο που ονομάζεται 'Map Name' . Στην συνέχεια χρησιμοποιούμε αυτό το πεδίο για να συνδέσουμε όλους τους πίνακες μεταξύ τους όπως φαίνεται και στην παρακάτω εικόνα



Την σύνδεση του προγράμματος με τον πίνακα 'Cord' την ονομάζω 'ps' . Στο πρόγραμμα μας χρησιμοποιούμε τις εντολές :

With ps

```
.AddNew
.Fields("X Coordinates") = GetCursorX
.Fields("Y Coordinates") = GetCursorY
.Fields("Map Name") = Text9.Text
.Update
```

End With

Για να εισάγουμε δεδομένα που χρειαζόμαστε σε αυτόν τον πίνακα . Με την εντολή ps.AddNew δημιουργούμε ένα καινούργιο πεδίο ούτως ώστε να είμαστε σε θέση να εισάγουμε όσα δεδομένα χρειαζόμαστε και θέλουμε . Στην συνέχεια λέμε στη βάση στο πεδίο που ονομάζεται "X Coordinates" να τοποθετήσεις την τιμή GetCursorX . Η εντολή GetCursorX είναι μία συνάρτηση η οποία μας επιστρέφει την τιμή του κέρσορα χ . Αντίστοιχα και για την GetCursorY . Στη βάση αποθηκεύεται το όνομα του χάρτη που ορίζω εγώ μέσα από το πρόγραμμα και αυτό γίνεται μέσω της εντολής ps.Fields("Map Name") = Text9.Text . Στην συνέχεια κάνουμε Update τη βάση μας .

## **ΠΑΡΑΡΤΗΜΑ 1**

### **Error Message Report**

<b>Code</b>	<b>Message</b>
3	Return without GoSub
5	Invalid procedure call
6	Overflow
7	Out of memory
9	Subscript out of range
10	This array is fixed or temporarily locked
11	Division by zero
13	Type mismatch
14	Out of string space
16	Expression too complex
17	Can't perform requested operation
18	User interrupt occurred
20	Resume without error
28	Out of stack space
35	Sub, Function, or Property not defined
47	Too many code resource
48	Error in loading code resource or DLL
49	Bad code resource
51	Internal error
52	Bad file name or number
53	File not found
54	Bad file mode
55	File already open
57	Device I/O error
58	File already exists
59	Bad record length
61	Disk full
62	Input past end of file
63	Bad record number
67	Too many files
68	Device unavailable
70	Permission denied
71	Disk not ready
74	Can't rename with different drive
75	Path/File access error
76	Path not found
91	Object variable not set
92	For loop not initialized

93	Invalid pattern string
94	Invalid use of Null
97	Can't call Friend procedure on an object
98	A property call cannot include a reference
298	System resource or DLL could not be loaded
320	Can't use character device names
321	Invalid file format
322	Can't create necessary temporary file
325	Invalid format in resource file
327	Data value named not found
328	Illegal parameter; can't write arrays
335	Could not access system registry
336	ActiveX Component not correctly registered
337	ActiveX Component not found
338	ActiveX Component did not run correctly
360	Object already loaded
361	Can't load or unload this object
363	ActiveX Control specified not found
364	Object was unloaded
365	Unable to unload within this context
368	The specified file is out of date
371	The specified object can't be
380	Invalid property value
381	Invalid property-array index
382	Property Set can't be executed at runtime
383	Property Set can't be used
385	Need property-array index
387	Property Set not permitted
393	Property Get can't be executed at runtime
394	Property Get can't be executed
400	Form already displayed; can't show modally
402	Code must close topmost modal form first
419	Permission to use object denied
422	Property not found
423	Property or method not found
424	Object required
425	Invalid object use
429	ActiveX Component can't create object
430	Class doesn't support Automation
432	File name or class name not found
438	Object doesn't support this property
440	Automation error
442	Connection to type library has been lost
445	Object doesn't support this action
446	Object doesn't support named arguments

447	Object doesn't support current locale setting
448	Named argument not found
449	Argument not
450	Wrong number of arguments
451	Object not a collection
452	Invalid ordinal
454	Code resource not found
455	Code resource lock error
457	This key is already associated
458	Variable uses a type not supported
460	Invalid Clipboard format
461	Specified format doesn't match
480	Can't create AutoRedraw image
481	Invalid picture
482	Printer error
484	Problem getting printer
485	Invalid picture type
486	Can't print form image to this type of printer
520	Can't empty Clipboard
521	Can't open Clipboard
735	Can't save file to TEMP directory
744	Search text not found
746	Replacements too long
31001	Out of memory
31004	No object
31018	Class is not set
31027	Unable to activate object
31032	Unable to create embedded object
31036	Error saving to file
31037	Error loading from file

## ΠΑΡΑΡΤΗΜΑ 2

### Κώδικας Προγράμματος Dynamic Map

```
Option Explicit

Private Const DT_EDITCONTROL = &H2000&

Private Declare Function BitBlt Lib "gdi32" _
    (ByVal hDCDest As Long, ByVal XDest As Long, ByVal
YDest As Long, _
    ByVal nWidth As Long, ByVal nHeight As Long, ByVal
hDCSrc As Long, _
    ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop
As Long) As Long

Private Declare Function GetDC Lib "user32" (ByVal hWnd
As Long) As Long

Private Declare Function DrawText Lib "user32" Alias
"DrawTextA" _
    (ByVal hdc As Long, ByVal lpStr As String, ByVal
nCount As Long, _
    lpRect As RECT, ByVal wFormat As Long) As Long

Private cn As ADODB.Connection
Private RS As ADODB.Recordset
Private ps As ADODB.Recordset
Private ts As ADODB.Recordset

Private Type POINTAPI
    x As Long
    y As Long
End Type

Dim Combos(10) As ComboBox
Dim Combo(10) As ComboBox

Dim Label(10) As Label
Dim Labl(10) As Label
Dim Txt(20) As TextBox
Dim Txts(20) As TextBox

Private WithEvents cboCombo As ComboBox

Dim blnDrawFlag As Boolean

Private Declare Function GetCursorPos Lib "user32"
(lpPoint As POINTAPI) As Long
```

```
Public px As Single
Public py As Single

Private Declare Function SendMessage Lib "user32" Alias
"SendMessageA" (ByVal hWnd As Long, ByVal wParam As Long,
ByVal lParam As Integer, ByVal lParam As Long) As Long
Const WM_VSCROLL = &H115
Const SB_LINEDOWN = 1
Const SB_LINEUP = 0

Dim sX As Single, sY As Single

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Public Function GetCursor()

    Dim wndrect As RECT
    Dim mousexy As POINTAPI
    Dim xi As Boolean
    Dim yi As Boolean

    GetCursorPos mousexy
    GetCursor = (mousexy.x - wndrect.Left) & " = X , Y =
" & (mousexy.y - wndrect.Top)

    xi = mousexy.x
    yi = mousexy.y

End Function

Public Function GetCursorX()

    Dim wndrect As RECT
    Dim mousexy As POINTAPI
    Dim xi As Boolean
    Dim yi As Boolean

    GetCursorPos mousexy
    GetCursorX = mousexy.x - wndrect.Left

    xi = mousexy.x
    yi = mousexy.y

End Function
```

```
Public Function GetCursorY()  
  
    Dim wndrect As RECT  
    Dim mousexy As POINTAPI  
    Dim xi As Boolean  
    Dim yi As Boolean  
  
    GetCursorPos mousexy  
    GetCursorY = mousexy.y - wndrect.Top  
  
    xi = mousexy.x  
    yi = mousexy.y  
  
End Function  
  
Private Sub Calendar_Click()  
  
    Form2.Visible = True  
  
End Sub  
  
Private Sub Command11_Click()  
  
    RS.Delete  
    RS.MoveNext  
    ps.Delete  
    ps.MoveNext  
  
End Sub  
  
Private Sub Command12_Click()  
  
    Dim i As Integer  
  
    Picture1.Width = 767  
    Picture1.Height = 608  
  
    For i = 100 To 300  
  
        Picture1.PaintPicture Picture1.Picture, 0, 0, 797,  
638  
        i = i + 100  
  
    Next i  
  
End Sub  
  
Private Sub Command2_Click()  
  
    Picture1.Width = 700
```



```
Picture1.Height = 500

Picture1.PaintPicture Picture1.Picture, 0, 0, 730, 520

End Sub

Private Sub Command3_Click()

    Form5.Visible = True
    On Error GoTo x
    SavePicture Picture1.Picture, "c:\vb6\temp.jpg"
    On Error GoTo x

x:
End Sub

Private Sub Command4_Click()

    Dim i As Integer
    Dim x As String

    For i = 10 To 10

        Set cn = New ADODB.Connection 'we've declared it as a
ADODB connection lets set it.
        cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" &
        "Data Source= C:\vb6\DB1.mdb" 'this is the
connection string explained in the notes section.
        cn.Open

        Set RS = New ADODB.Recordset 'as we did with the
connection
        RS.Open "tbl_master", cn, adOpenKeyset,
adLockPessimistic, adCmdTable 'opening the recordset
explained in the notes

        Txt(i).Text = Text13.Text

        Set Txt(i) = Me.Controls.Add("Vb.TextBox",
"cmdButton" & Me.Controls.Count, Picture1)

        With Txt(i)

            .Text = Txt(i).Text

        End With

    End With
```

```
With RS

    .AddNew
    .Fields("Quantity") = Text13.Text
    .Fields("Item") = Text13.Text
    .Fields("Map Name") = Text9.Text
    .Update

End With

Next i

End Sub

Private Sub Command5_Click()

    Text12.Text = "0"

End Sub

Private Sub Command7_Click()

    Dim i As Integer

    With Combo7
        .AddItem Text10.Text
    End With

End Sub

Private Sub DeleteAll_Click()

    Dim i As Integer
    Dim kkk As Integer

    Clipboard.Clear
    Picture1.Cls
    Picture18.Cls
    Picture18.Refresh
    Picture1.Refresh
    Picture1.Picture = Picture
    Set Picture1.Picture = Nothing
```

```
Form4.Visible = True
Unload Form1

Clipboard.SetData Picture18.Picture
Clipboard.SetData Picture1.Picture
Cls

Load Form1
Form1.Visible = True
Unload Form4
Y:

End Sub

Private Sub Form_Load()

    Picture1.ToolTipText = "Open a Map to work with (Use
File- Open)"

End Sub

Private Sub Help_Click()

    CommonDialog1.HelpCommand = &HB
    CommonDialog1.HelpFile = "c:\windows\help\windows.hlp"

    CommonDialog1.ShowHelp

End Sub

Private Sub HScroll1_Change()

    Picture1.Left = -HScroll1.Value

    HScroll1.Min = 1
    HScroll1.Max = 100
    HScroll1.LargeChange = 10

    VScroll1.Min = 1
    VScroll1.Max = 100
    VScroll1.LargeChange = 10

End Sub

Private Sub Loads_Click()

    Dim h As String
    Dim x, y As Long
    Dim i As Integer
```

```
Dim k, j As String
Dim x2 As Integer
Dim y2 As Integer
Dim h2 As String
Dim p As String
Dim r As Integer
CommonDialog1.DialogTitle = "Select a file to open"
CommonDialog1.ShowOpen
Picture1.Picture = LoadPicture(CommonDialog1.FileName)

Set cn = New ADODB.Connection 'we've declared it as a
ADODB connection lets set it.
cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=C:\vb6\DB1.mdb" 'this is the
connection string explained in the notes section.
cn.Open

Set RS = New ADODB.Recordset 'as we did with the
connection
RS.Open "tbl_master", cn, adOpenKeyset,
adLockPessimistic, adCmdTable

Set cn = New ADODB.Connection 'we've declared it as a
ADODB connection lets set it.
cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source= C:\vb6\DB1.mdb" 'this is the
connection string explained in the notes section.
cn.Open

Set ps = New ADODB.Recordset 'as we did with the
connection
ps.Open "Cord", cn, adOpenKeyset, adLockPessimistic,
adCmdTable 'opening the recordset explained in the notes

Set cn = New ADODB.Connection 'we've declared it
as a ADODB connection lets set it.
cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=C:\vb6\DB1.mdb" 'this is the
connection string explained in the notes section.
cn.Open

Set ts = New ADODB.Recordset 'as we did with the
connection
ts.Open "third", cn, adOpenKeyset,
adLockPessimistic, adCmdTable
```

```
        blnDrawFlag = False
For i = 0 To 9
    blnDrawFlag = False
    h = ps.Fields("Map Name").Value
    h2 = ts.Fields("Map Name").Value

    Text11.Visible = True

    If Not h = CommonDialog1.FileTitle Then
On Error GoTo x
        ps.MoveNext
        RS.MoveNext
        ts.MoveNext
        If RS.EOF Then
            GoTo x:
        End If

        If ps.EOF Then
            GoTo x:
        End If

    End If

    If h = CommonDialog1.FileTitle Then

        Text11.Text = h
        Text11.Visible = True

        x = ps.Fields("X Coordinates").Value
        y = ps.Fields("Y Coordinates").Value
        k = RS.Fields("Item").Value
        j = RS.Fields("Quantity").Value
        p = RS.Fields("shape").Value

        x2 = ts.Fields("x2").Value
        y2 = ts.Fields("y2").Value

    If p = "Square" Then

        Picture1.Line (x, y)-(x2, y2), vbBlack, B

        Set Txt(i) = Me.Controls.Add("VB.Textbox",
"cmdButton" & Me.Controls.Count, Picture1)

        With Txt(i)
```

```
.Left = x '+ 9
.Top = y
.Width = 60
.Height = 5
.Text = k
.Visible = True
.BorderStyle = 0
.BackColor = &HFF8080
.Alignment = 2
.SetFocus
End With
```

```
Set Txts(i) = Me.Controls.Add("Vb.TextBox",
"cmdButton" & Me.Controls.Count, Picture1)
```

```
With Txts(i)
.Left = x '+ 9
.Top = y + 20
.Width = 60
.Height = 5
.Text = j
.BackColor = &H80FF80
.Alignment = 2
.Visible = True
End With
```

```
If Not RS.EOF Then
```

```
Text1.Text = Text1.Text & RS.Fields("Item").Value
& vbCrLf
```

```
Text2.Text = Text2.Text &
RS.Fields("Quantity").Value & vbCrLf
```

```
End If
```

```
If Not ps.EOF Then
```

```
Text3.Text = Text3.Text & ps.Fields("X
Coordinates").Value & vbCrLf
```

```
Text4.Text = Text4.Text & ps.Fields("Y
Coordinates").Value & vbCrLf
```

```
End If
```

```
If Not ts.EOF Then
```

```
Text14.Text = Text14.Text & ts.Fields("x2").Value  
& vbCrLf  
Text15.Text = Text15.Text & ts.Fields("y2").Value  
& vbCrLf
```

```
End If
```

```
If Not RS.EOF Then  
    RS.MoveNext  
End If
```

```
If Not ps.EOF Then  
    ps.MoveNext  
End If
```

```
If Not ts.EOF Then  
    ts.MoveNext  
End If
```

```
If Not h = CommonDialog1.FileTitle Then  
    GoTo x:  
End If
```

```
If RS.EOF Then  
    GoTo x:  
End If
```

```
If ps.EOF Then  
    GoTo x:  
End If
```

```
End If
```

```
End If
```

```
If p = "Circle" Then
```

```
    r = x - x2  
    Picture1.Circle (x, y), r
```

```
    Set Txt(i) = Me.Controls.Add("VB.Textbox",  
"cmdButton" & Me.Controls.Count, Picture1)
```

```
    With Txt(i)  
        .Left = x ' + 9  
        .Top = y  
        .Width = 60
```

```
.Height = 5
.Text = k
.Visible = True
.BorderStyle = 0
.BackColor = &HFF8080
.Alignment = 2
.SetFocus
End With

Set Txts(i) = Me.Controls.Add("Vb.TextBox",
"cmdButton" & Me.Controls.Count, Picture1)

With Txts(i)
.Left = x ' + 9
.Top = y + 20
.Width = 60
.Height = 5
.Text = j
.BackColor = &H80FF80
.Alignment = 2
.Visible = True
End With

If Not RS.EOF Then

    Text1.Text = Text1.Text &
RS.Fields("Item").Value & vbCrLf
    Text2.Text = Text2.Text &
RS.Fields("Quantity").Value & vbCrLf

End If

If Not ps.EOF Then

    Text3.Text = Text3.Text & ps.Fields("X
Coordinates").Value & vbCrLf
    Text4.Text = Text4.Text & ps.Fields("Y
Coordinates").Value & vbCrLf

End If

If Not ts.EOF Then

    Text14.Text = Text14.Text &
ts.Fields("x2").Value & vbCrLf
```



```
        Text15.Text = Text15.Text &
ts.Fields("y2").Value & vbCrLf

        End If

        If Not RS.EOF Then

            RS.MoveNext

        End If

        If Not ps.EOF Then

            ps.MoveNext

        End If

        If Not ts.EOF Then

            ts.MoveNext

        End If

        If Not h = CommonDialog1.FileTitle Then

            GoTo x:

        End If

        If RS.EOF Then

            GoTo x:

        End If

        If ps.EOF Then

            GoTo x:

        End If

    End If

Next i
```

x:

End Sub

```
Private Sub Picture1_MouseMove(Button As Integer, Shift  
As Integer, x As Single, y As Single)
```

```
    Label1.Caption = GetCursor  
    Dim l As Integer  
    Dim k As Integer  
    Dim blnDrawFlag As Boolean  
    Dim x1 As Double  
    Dim x2 As Double  
    Dim y1 As Double  
    Dim y2 As Double  
    Dim square As Shape
```

```
    Me.ScaleMode = vbPixels
```

```
    blnDrawFlag = False
```

```
    If Combo1.Text = "Square" Then
```

```
        blnDrawFlag = False  
        If Button = vbLeftButton Then
```

```
            y2 = x - sX
```

```
            blnDrawFlag = True
```

```
            Set Picture1.Picture = Picture18.Picture
```

```
            Combo1.Refresh
```

```
            Picture1.Line (sX, sY)-(x, y), vbBlack, B  
            If blnDrawFlag Then Picture1.Line -(x, y)
```

```
Else
    sX = x
    sY = y

End If

End If

If Comb1.Text = "Circle" Then

    blnDrawFlag = False
    If Button = vbLeftButton Then

        blnDrawFlag = True
        y2 = sX - x + 10
        Set Picture1.Picture = Picture18.Picture

        Comb1.Refresh

        On Error GoTo x:

        Picture1.Circle (sX, sY), y2

x:

Else
    sX = x
    sY = y
End If

End If

l = y
k = x
```

```
blnDrawFlag = False

End Sub
Private Sub Command1_Click()

    Picture1.Cls
    Picture18.Cls

    Picture1.Width = 400
    Picture1.Height = 430

    Picture1.PaintPicture Picture1.Picture, 0, 0, 400, 430

End Sub

Private Static Sub Picture1_MouseDown(Button As Integer,
Shift As Integer, x As Single, y As Single)

    Dim i As Integer
    Dim k As Integer
    Dim li As Integer
    Dim b As Integer
    Dim h As String
    Dim j As Integer
    Dim l As Integer
    Dim bytData() As Byte
    Dim p As Integer
    Dim ii As Integer

    Picture1.CurrentX = CurrentX
    Picture1.CurrentY = CurrentY

    Dim ke As String
    Dim je As String

    Set cn = New ADODB.Connection 'we've declared it as a
ADODB connection lets set it.
    cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" &
    "Data Source=C:\vb6\DB1.mdb" 'this is the
connection string explained in the notes section.
    cn.Open

    Set RS = New ADODB.Recordset 'as we did with the
connection
```

```
RS.Open "tbl_master", cn, adOpenKeyset,  
adLockPessimistic, adCmdTable
```

```
Set cn = New ADODB.Connection 'we've declared it as a  
ADODB connection lets set it.
```

```
cn.ConnectionString =  
"Provider=Microsoft.Jet.OLEDB.4.0;" &  
"Data Source= C:\vb6\DB1.mdb" 'this is the  
connection string explained in the notes section.  
cn.Open
```

```
Set ps = New ADODB.Recordset 'as we did with the  
connection
```

```
ps.Open "Cord", cn, adOpenKeyset, adLockPessimistic,  
adCmdTable 'opening the recordset explained in the notes
```

```
If Button = vbLeftButton Then
```

```
Set Picture18.Picture = Picture1.Image
```

```
CurrentX = li
```

```
If k = 0 & l = 0 Then
```

```
k = CurrentX
```

```
l = CurrentY
```

```
End If
```

```
If Not k = 0 Then
```

```
h = CurrentX
```

```
j = CurrentY
```

```
End If
```

```
CurrentX = x
```

```
CurrentY = y
```

```
For k = 0 To 9
```

```
Next k
```

```
For p = 0 To 9
```

```
p = p + 1
```

```
For j = 10 To 19

    Set Txt(j) = Me.Controls.Add("VB.TextBox",
    "cmdButton" & Me.Controls.Count, Picture1)

    With Txt(j)
        .Left = x
        .Top = y
        .Width = 60
        .Height = 5
        .Text = Combo7.Text
        .Visible = True
        .BorderStyle = 0
        .BackColor = &HFF8080
        .Alignment = 2
        .SetFocus
    End With

    Set Txts(j) = Me.Controls.Add("Vb.TextBox",
    "cmdButton" & Me.Controls.Count, Picture1)

    With Txts(j)
        .Left = x
        .Top = y + 20
        .Width = 60
        .Height = 5
        .Text = Combo8.Text
        .BackColor = &H80FF80
        .Alignment = 2
        .Visible = True
    End With

Next j

Set cn = New ADODB.Connection 'we've declared it
as a ADODB connection lets set it.
cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source= C:\vb6\DB1.mdb" 'this is the
connection string explained in the notes section.
cn.Open
```

```
        Set RS = New ADODB.Recordset 'as we did with the
connection
        RS.Open "tbl_master", cn, adOpenKeyset,
adLockPessimistic, adCmdTable 'opening the recordset
explained in the notes

    Next p

        If Not RS.EOF Then

            RS.MoveNext
        End If

        If Not ps.EOF Then

            ps.MoveNext

        End If

    End If

End If

For i = 10 To 10

    On Error GoTo x

    With RS

        .AddNew
        .Fields("Quantity") = Txts(i).Text
        .Fields("Item") = Txt(i).Text
        .Fields("Map Name") = Text9.Text
        .Fields("shape") = Combo1.Text
        .Update

    On Error GoTo x
    End With

Next i

Set cn = New ADODB.Connection
cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source= C:\vb6\DB1.mdb"
```

```
cn.Open

Set RS = New ADODB.Recordset
RS.Open "tbl_master", cn, adOpenKeyset,
adLockPessimistic, adCmdTable

Set ps = New ADODB.Recordset
ps.Open "Cord", cn, adOpenKeyset, adLockPessimistic,
adCmdTable

Set cn = New ADODB.Connection
cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
  "Data Source= C:\vb6\DB1.mdb"
cn.Open

Set ts = New ADODB.Recordset
ts.Open "third", cn, adOpenKeyset, adLockPessimistic,
adCmdTable

If Not RS.EOF Then

    RS.MoveLast

End If

If Not ps.EOF Then

    ps.MoveLast

End If

With ps
    .AddNew
    .Fields("X Coordinates") = x
    .Fields("Y Coordinates") = y
    .Fields("Map Name") = Text9.Text
    .Update
End With

ps.Update
```



```
h = RS.Fields("Map Name").Value

If Not RS.EOF Then

    Text1.Text = Text1.Text & RS.Fields("Item").Value
    & vbCrLf
    Text2.Text = Text2.Text &
    RS.Fields("Quantity").Value & vbCrLf

    If h = CommonDialog1.FileTitle Then

        RS.MoveNext

    End If

End If

If Not ps.EOF Then

    Text3.Text = Text3.Text & ps.Fields("X
Coordinates").Value & vbCrLf
    Text4.Text = Text4.Text & ps.Fields("Y
Coordinates").Value & vbCrLf
    If h = CommonDialog1.FileTitle Then

        ps.MoveNext

    End If

End If

x:
End Sub
Private Sub Picture1_MouseUp(Button As Integer, Shift As
Integer, x As Single, y As Single)

    Dim h As String

    Set cn = New ADODB.Connection
    cn.ConnectionString =
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source= C:\vb6\DB1.mdb"
```

```
cn.Open

Set ts = New ADODB.Recordset 'as we did with the
connection
ts.Open "third", cn, adOpenKeyset, adLockPessimistic,
adCmdTable 'opening the recordset explained in the notes

On Error GoTo x

If Not ps.EOF Then

    ts.MoveLast

End If

With ts
    .AddNew
    .Fields("x2") = x
    .Fields("y2") = y
    .Fields("Map Name") = Text9.Text
End With

h = ts.Fields("Map Name").Value

ts.Update

If Not ps.EOF Then

    ts.MoveNext

End If

If Not ts.EOF Then

    Text14.Text = Text14.Text & ts.Fields("x2").Value &
vbCrLf
    Text15.Text = Text15.Text & ts.Fields("y2").Value &
vbCrLf
    If h = CommonDialog1.FileTitle Then

        ts.MoveNext

    End If
End If

x:
End Sub
```

```
Private Sub Picture3_Click()

    Picture1.Line (Picture1.CurrentX, Picture1.CurrentY)-
    (CurrentX, CurrentY), vbBlack

End Sub
Private Sub Picture3_OLEDragDrop(Data As DataObject,
Effect As Long, Button As Integer, Shift As Integer, x As
Single, y As Single)

    Picture1.Line (Picture1.CurrentX, Picture1.CurrentY)-
    (CurrentX, CurrentY), vbBlack

End Sub
Private Sub Print_Click()

    CommonDialog2.ShowPrinter

End Sub

Private Sub StructuredGraphicsControll1_onmousemove(ByVal
Button As Long, ByVal Shift As Long, ByVal x As Long,
ByVal y As Long)

    Dim square As Shape
    Dim x1 As Double
    Dim x2 As Double
    Dim y1 As Double
    Dim y2 As Double

    Me.AutoRedraw = True
    Me.ScaleMode = vbPixels
    Set square = Me.Controls.Add("VB.Shape", "square")

        With square
            .Move 10, 10, 100, 200 'for example
            x1 = square.Left
            y1 = square.Top
            x2 = square.Left + square.Width
            y2 = square.Top + square.Height
            .Visible = True
        End With

    Debug.Print x1
    Debug.Print y1
    Debug.Print x2
    Debug.Print y2

End Sub
```

```
Private Sub Text9_Click()  
    Text9.Text = CommonDialog1.FileTitle  
  
End Sub  
  
Private Sub Timer1_Timer()  
    Label1.Caption = GetCursor  
  
End Sub  
Private Sub Exit_Click()  
    Form3.Visible = True  
  
End Sub  
  
Private Sub Open_Click()  
    Dim i As Integer  
    Dim x As Single, y As Single  
  
    Picture1.Cls  
    Picture1.Picture = Nothing  
    Picture1.Refresh  
  
    Picture18.Cls  
    Picture18.Picture = Nothing  
  
    Picture1.ScaleHeight = 613  
    Picture1.ScaleWidth = 667  
    Picture1.Width = 667  
    Picture1.Height = 600  
  
    CommonDialog1.DialogTitle = "Select a file to open"  
    CommonDialog1.ShowOpen  
  
    Picture1.Picture = LoadPicture(CommonDialog1.FileName)  
    Picture1.ToolTipText = ""  
    On Error GoTo x  
    SavePicture Picture1.Picture, "c:\vb6\temp.jpg"  
    On Error GoTo x  
x:  
  
End Sub  
  
Private Sub Save_Click()
```

```
If Text9.Text = "" Then GoTo x
    SavePicture Picture1.Picture, "c:\vb6\temp.jpg"

    CommonDialog1.ShowSave

    Picture18.Picture = Picture1.Image
    Picture1.Picture = Picture18.Picture
    MsgBox "File to save: " & CommonDialog1.FileName
    SavePicture Picture1.Picture, CommonDialog1.FileName

x:
    MsgBox "Enter a Map Name in the Fields unter 'Enter
File Name : (Name of Map to be stored)' "
End Sub

Private Sub Command9_Click()

    Dim wndrect As RECT
    Dim mousexy As POINTAPI
    Dim xi As Boolean
    Dim yi As Boolean
    Dim li As Long
    Dim i As Integer
    Dim bytData() As Byte

    Open CommonDialog1.FileName For Binary As #1
        ReDim bytData(FileLen(CommonDialog1.FileName))
    Get #1, , bytData
    Close #1

    Set cn = New ADODB.Connection
        cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source= C:\vb6\DB1.mdb"
    cn.Open

    Set RS = New ADODB.Recordset
        RS.Open "tbl_master", cn, adOpenKeyset,
adLockPessimistic, adCmdTable

        With RS

            .AddNew
            .Fields("Quantity") = Combo8.Text
            .Fields("Item") = Combo7.Text
            .Fields("Map Name") = Text9.Text
```

```
        .Update
        On Error GoTo x
    End With
x:
End Sub

Private Sub Timer2_Timer()

    Label4.Caption = Time$

End Sub

Private Sub Undo_Click()

    Picture1.Cls
    Picture18.Cls

    Combo3.Visible = False
    Combo6.Visible = False

End Sub

Private Sub VScroll11_Change()

    Picture1.Top = -VScroll11.Value
    VScroll11.Min = 1
    VScroll11.Max = 100

End Sub

Private Sub VScroll12_Change()

    SendMessage Text1.hWnd, WM_VSCROLL, SB_LINEDOWN, ByVal
0
    SendMessage Text2.hWnd, WM_VSCROLL, SB_LINEDOWN, ByVal
0
    SendMessage Text3.hWnd, WM_VSCROLL, SB_LINEDOWN, ByVal
0
    SendMessage Text4.hWnd, WM_VSCROLL, SB_LINEDOWN, ByVal
0

    'SendMessage Text1.hwnd, WM_VSCROLL, SB_LINEUP, ByVal 0
    'SendMessage Text2.hwnd, WM_VSCROLL, SB_LINEUP, ByVal 0
    'SendMessage Text3.hwnd, WM_VSCROLL, SB_LINEUP, ByVal 0
    'SendMessage Text4.hwnd, WM_VSCROLL, SB_LINEUP, ByVal 0

End Sub

Sub TextTrans(MyTB As Object)
```

```
Dim TempDC As Long
Dim Temp As String
Dim MyLoc As RECT

On Error Resume Next

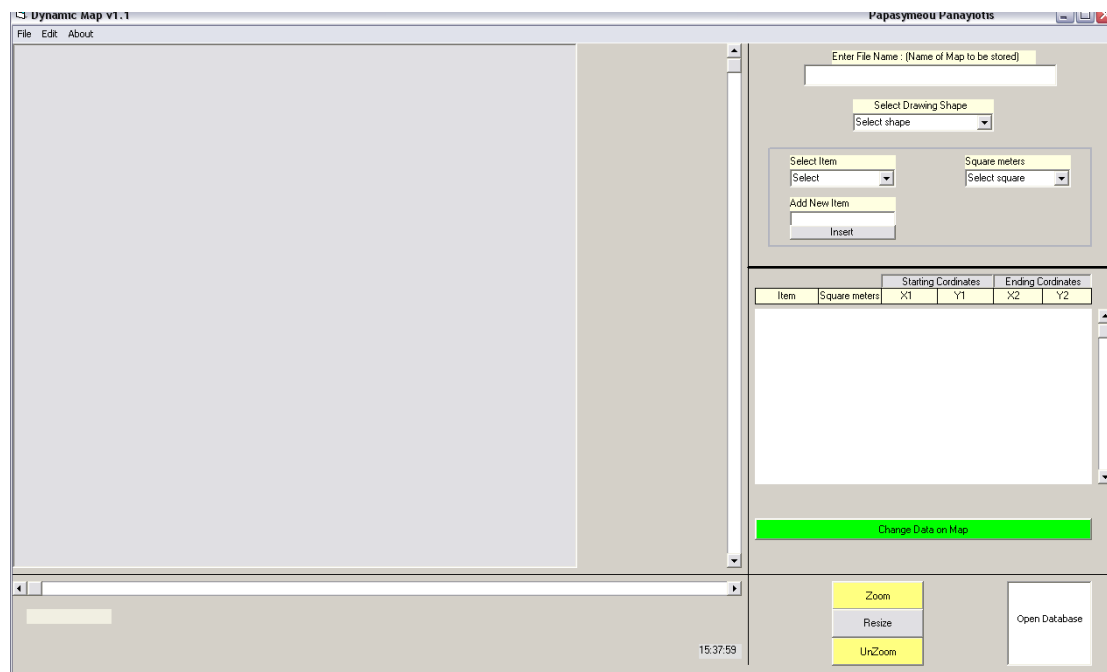
    Temp = MyTB.Text
    MyLoc.Left = MyTB.Left
    MyLoc.Top = MyTB.Top
    MyLoc.Right = MyLoc.Left + MyTB.Width
    MyLoc.Bottom = MyLoc.Top + MyTB.Height
    MyTB.Parent.Cls
    MyTB.Parent.ForeColor = MyTB.ForeColor
    Set MyTB.Parent.Font = MyTB.Font
    DrawText MyTB.Parent.hdc, Temp, Len(Temp), MyLoc,
DT_EDITCONTROL
    TempDC = GetDC(MyTB.hWnd)
    BitBlt TempDC, 0, 0, MyTB.Width, MyTB.Height,
MyTB.Parent.hdc, MyTB.Left, MyTB.Top, vbSrcCopy

End Sub
```

## ΠΑΡΑΡΤΗΜΑ 3

### ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ

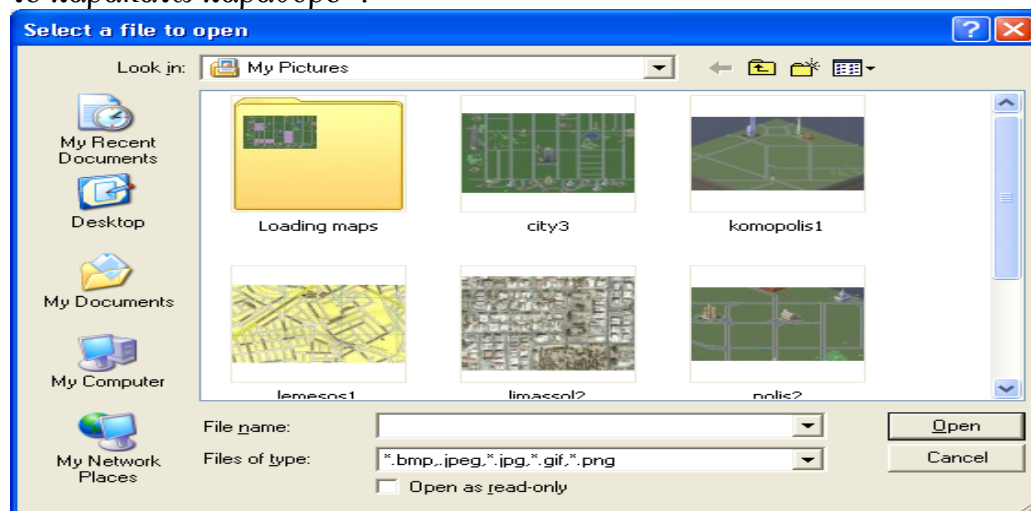
Κάνοντας διπλό κλικ ο χρήστης στο εικονίδιο με το όνομα '*Dynamic Map*' εμφανίζεται η παρακάτω φόρμα :



Αυτή είναι η φόρμα επεξεργασίας του χάρτη .Όλα τα εργαλεία υπάρχουν σε αυτή την φόρμα . Υπάρχει ένα μενού με τις ακόλουθες επιλογές : *File* , *Edit* , *About* .

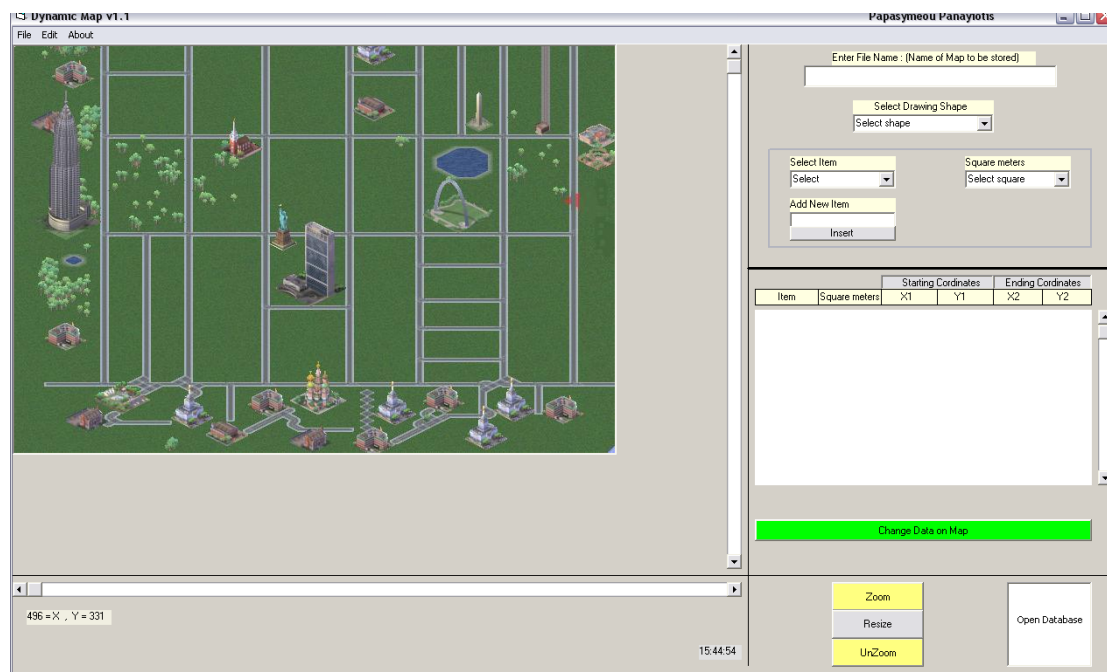
### Getting Started

Από το μενού και την επιλογή '*File*' επιλέγουμε '*Open Image*' . Εμφανίζεται το παρακάτω παράθυρο :





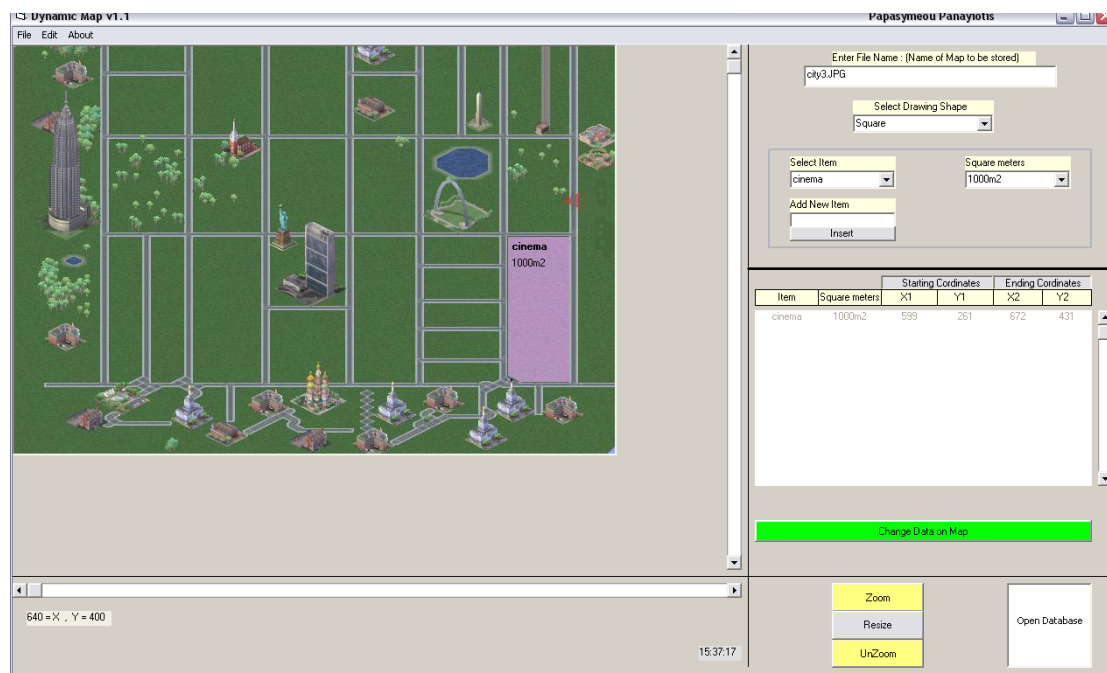
Στην συνέχεια επιλέγουμε το χάρτη τον οποίο θα επεξεργαστούμε και πατάμε ‘Open’ .Ο χάρτης που επιλέξαμε εμφανίζεται στην φόρμα μας όπως φαίνεται παρακάτω :



Επόμενο βήμα είναι να ορίσουμε το όνομα του χάρτη που θα αποθηκευτεί στη βάση . Στα δεξιά της φόρμα πάνω ψηλά υπάρχει μία περιοχή κειμένου όπου ο χρήστης εισάγει το όνομα του χάρτη . Κάνοντας κλικ ο χρήστης σε αυτή την περιοχή εμφανίζεται το όνομα του χάρτη που ανοίξαμε – αποτελεί προεπιλογή για σκοπούς διευκόλυνσης του χρήστη .

Στην συνέχεια επιλέγουμε σχεδιαστικό σχήμα από τις επιλογές ‘Select Drawing Shape’. Υπάρχει η επιλογή του ‘κύκλου’ και του ‘παραλληλογράμμου’ . Ακολούθως επιλέγουμε το όνομα του κτιρίου από την επιλογή ‘Select Item’ και τα τετραγωνικά μέτρα από την επιλογή ‘Select Meters’ . Στην συνέχεια επιλέγουμε μία περιοχή στο χάρτη και εισάγουμε τα δεδομένα .

Στην παρακάτω εικόνα υπάρχει το παραπάνω παράδειγμα.



Εδώ ο χρήστης επέλεξε για σχήμα ‘παράλληλόγραμμο’ , για κτίριο ‘cinema’ και χίλια τετραγωνικά μέτρα .

Τα δεδομένα που εισάγουμε εμφανίζονται στα δεξιά της φόρμας . Όπου X1,Y1,X2,Y2 είναι οι συντεταγμένες όπου υπάρχει το κτίριο .

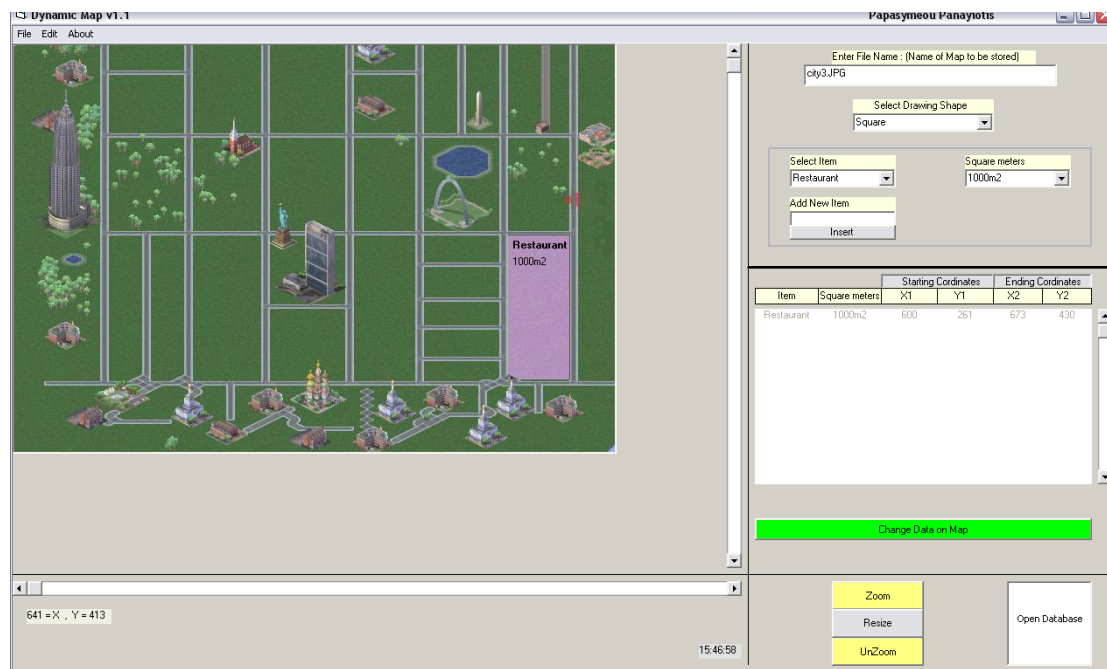
Στην συνέχεια ο χρήστης μπορεί να εισάγει όσα δεδομένα χρειάζεται με τον ίδιο τρόπο .

Να σημειωθεί ότι ο χρήστης δεν χρειάζεται να αποθηκεύσει το χάρτη . Τα δεδομένα αποθηκεύονται αυτόματα κατά την διάρκεια λειτουργίας του προγράμματος .Ο χάρτης δεν αλλοιώνεται καθώς όλα τα δεδομένα εισάγονται δυναμικά .

Σε περίπτωση που ο χρήστης χρειαστεί να κάνει μετατροπή στο χάρτη χρειάζεται να πατηθεί το κουμπί ‘Change Data on Map ‘ .Εμφανίζεται το παρακάτω παράθυρο .

Item	Square meters	x1	y1	x2	y2
cinema	1000m2	599	262	671	432

Με τα βελάκια ο χρήστης βλέπει την προηγούμενη ή την επόμενη εγγραφή στο χάρτη. Τα δεδομένα εμφανίζονται ένα προς ένα. Έστω ότι θέλουμε να κάνουμε μία τροποποίηση στο κτίριο 'cinema'. Από τις επιλογές 'Select Item' που υπάρχει από κάτω επιλέγουμε το κτίριο που θέλουμε. Έστω ότι είναι 'Restaurant'. Στην συνέχεια πατάμε το κουμπί 'Replce With' και στην συνέχεια το κουμπί 'Refresh Map'. Τα δεδομένα έχουν τροποποιηθεί και εμφανίζεται η φόρμα μας με το χάρτη με τροποποιημένα τα δεδομένα. Παράδειγμα στην παρακάτω εικόνα.



Αν ο χρήστης έχει τελειώσει μπορεί να τερματίσει το πρόγραμμα είτε πατώντας το 'x' πάνω δεξιά της φόρμας είτε από το μενού στην επιλογή 'Exit' και στην συνέχεια 'Yes' .

Αν ο χρήστης θέλει να επεξεργαστεί άλλο χάρτη τότε από το μενού επιλέγει 'Edit' και στην συνέχεια 'Delete All' . Στην συνέχεια επαναλαμβάνει τα παραπάνω βήματα που έχουν αναφερθεί .

Αν ο χρήστης θέλει να επεξεργαστεί ένα χάρτη που έχει επεξεργαστεί στο παρελθόν τότε από το μενού επιλέγει 'Load Image' .Επιλέγει την εικόνα που χρειάζεται και κάνει 'Open' . Όλα τα δεδομένα του χάρτη φορτώνονται από τη βάση δεδομένων και ο χρήστης επεξεργάζεται το χάρτη όπως αυτός θέλει .

Αν ο χρήστης επιθυμεί να εισάγει κάποιο κτίριο που δεν είναι στις επιλογές τότε αυτό μπορεί να γίνει με τη χρήση του κουμπιού 'Insert' αφού πρώτα αναγράψει στον απαραίτητο χώρο κάτω από το όνομα 'Add New Object in Choices' .

Αν ο χρήστης επιθυμεί να χρησιμοποιήσει τις επιλογές 'zoom' και 'unzoom' τότε αυτό θα πρέπει να γίνει πριν την εισαγωγή δεδομένων. Η αλλαγή στο χάρτη θα πρέπει να αποθηκευτεί . Από το menu και την επιλογή 'File' επιλέγουμε 'Save' .

Δίνεται η δυνατότητα στο χρήστη να επεμβαίνει απευθείας στη βάση δεδομένων για αλλαγή στοιχείων ή και διαγραφή . Αυτό γίνεται με τη χρήση του κουμπιού OLE που βρίσκεται κάτω δεξιά της φόρμας με το όνομα 'Open Database' .

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] **Visual Basic 6 Black Book**, Steven Holzner
- [2] **The Visual Basic Bible**, Bill Evjen, Jason Beres
- [3] **Database Access with Visual Basic**, Jeffrey Mcmanus

### **Links στο Internet:**

- [4] <http://www.vbforums.com>
- [5] <http://www.vbcode.com>
- [6] <http://msdn2.microsoft.com>