

**ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ**

**Τμήμα Ηλεκτρονικής & Μηχανικών Ηλεκτρονικών
Υπολογιστών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ
ΤΗ ΔΙΔΑΣΚΑΛΙΑ ΓΛΩΣΣΑΣ
ΣΥΜΒΟΛΩΝ (SOFTWARE)**

Παναγόπουλος Παναγιώτης

Εξεταστική επιτροπή :

Ζερβάκης Μιχάλης , Αν. Καθηγητής
Καλαϊτάζκης Κώστας , Αν. Καθηγητής
Σταυρακάκης Γιώργος , Καθηγητής

ΧΑΝΙΑ, ΣΕΠΤΕΜΒΡΙΟΣ 1998

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη

Εισαγωγή

1 Σύμβολα BLISS

1.1 Γενικά για τα σύμβολα (εισαγωγή)

1.2 Περιγραφή Συμβόλων - Κανόνες

1.3 Κατηγοριοποίηση Συμβόλων

1.4 Απαιτήσεις των συμβόλων από το λογισμικό

2 Γλώσσα προγραμματισμού - VISUAL BASIC 5.0

2.1 Γενικά για τη γλώσσα (εισαγωγή)

2.2 Το περιβάλλον της Visual Basic (Αντικείμενα - Φόρμες - Controls)

2.3 Σύνδεση με το περιβάλλον WINDOWS (API)

2.4 Δυνατότητες της γλώσσας - Κάλυψη των απαιτήσεων

3 Λογισμικό για την διδασκαλία της γλώσσας

3.1 Μέρη προγράμματος

i Letter Editor

ii Word Editor

iii Page Editor

iv Sentence Player

3.2 Λειτουργίες - εργαλεία (εργαλειοθήκη - παλέτες)

3.3 Συναρτήσεις - Υπορουτίνες

3.4 Εξομοιωτής Πληκτρολογίου

4 Συμπεράσματα

Παράρτημα Α : Βασικά “γράμματα” BLISS

Παράρτημα Β : Εγχειρίδιο χρήσης προγράμματος

Ευρετήριο Όρων

Βιβλιογραφία - Παραπομπές - Σελίδες στο διαδίκτυο

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία έχει σαν θέμα της την ανάπτυξη του λογισμικού μέρους συσκευής διδασκαλίας γλώσσας συμβόλων. Οι γλώσσες συμβόλων χρησιμοποιούνται ως εναλλακτικό μέσο επικοινωνίας από άτομα με ειδικές ανάγκες. Η αδυναμία των ατόμων αυτών να επικοινωνούν με το συμβατικό αλφάβητο δεν συνοδεύεται απαραίτητα από νοητική καθυστέρηση· έτσι με την χρήση εναλλακτικών μέσων επικοινωνίας, καταφέρνουν να “σπάσουν” τη σιωπή τους. Η γλώσσα συμβόλων Bliss, που αποτελεί αντικείμενο της εργασίας, παρέχει ικανοποιητικό αριθμό τρόπων έκφρασης και προσεγγίζει αρκετά την συμβατική.

Το λογισμικό που παρουσιάζουμε, ασχολείται κυρίως με την κατασκευή των συμβόλων και λέξεων Bliss καθώς και με την ομαδοποίηση τους. Μέρος του λογισμικού επίσης είναι και η σύνθεση προτάσεων από σύμβολα - λέξεις που επιλέγει ο χρήστης μέσω του ειδικού πληκτρολογίου που έχει αναπτυχθεί για αυτό το σκοπό. Τα σύμβολα που εμφανίζονται κάθε φορά στο πληκτρολόγιο, είναι το αποτέλεσμα της ομαδοποίησης των λέξεων Bliss σε σελίδες από το αντίστοιχο τμήμα του λογισμικού.

Εισαγωγή

ΕΙΣΑΓΩΓΗ ΣΤΟ ΘΕΜΑ ΤΗΣ ΕΡΓΑΣΙΑΣ

Αφετηρία της παρούσας εργασίας αποτέλεσαν τα προβλήματα επικοινωνίας που παρουσιάζουν τα άτομα με ειδικές ανάγκες. Μερικά από αυτά δεν έχουν την δυνατότητα να χρησιμοποιήσουν το συμβατικό αλφάβητο, να καταλάβουν και σχηματίσουν λέξεις ή εννοιολογικά σχήματα με αυτό. Μπορούν όμως να επικοινωνήσουν χρησιμοποιώντας άλλους κώδικες, που έχουν αναπτυχθεί για αυτό τον σκοπό και που βασίζονται κυρίως σε εικόνες και σύμβολα.

Σε όλο τον κόσμο υπάρχουν φορείς που ασχολούνται με την περίθαλψη και φροντίδα των ατόμων με ειδικές ανάγκες. Στην Ελλάδα συγκεκριμένα, ο σημαντικότερος είναι η Ελληνική Εταιρία Προστασίας και Αποκατάστασης Αναπήρων Παιδιών (ΕΛ.Ε.Π.Α.Π.), που ιδρύθηκε το 1937 και έχει προσφέρει από τότε βοήθεια και υπηρεσίες σε περισσότερα από 80.000 παιδιά με διάφορες μορφές αναπηριών. Κύριος στόχος της ΕΛ.Ε.Π.Α.Π. είναι η θεραπεία, αποκατάσταση και κοινωνική ένταξη των παιδιών με ειδικές ανάγκες.

Υπεύθυνο τμήμα της ΕΛ.Ε.Π.Α.Π. για την ανάπτυξη και αποκατάσταση της επικοινωνίας των αναπήρων παιδιών είναι το τμήμα λογοθεραπείας. Εδώ γίνεται η χρήση των διάφορων κωδίκων-συστημάτων επικοινωνίας πέραν του κλασσικού αλφαβήτου. Τέτοια συστήματα για παράδειγμα είναι το Mager-Jonshon, το σύστημα που βασίζεται στα σύμβολα Bliss και το Cregon για άτομα με πολύ περιορισμένη αντίληψη. Ανάλογα με τις ανάγκες κάθε ατόμου γίνεται και η επιλογή του αντίστοιχου συστήματος.

Στα Χανιά πιο συγκεκριμένα, στο αντίστοιχο τμήμα λογοθεραπείας της ΕΛ.Ε.Π.Α.Π. Χανίων, που λειτουργεί από το 1991, το δεύτερο σύστημα (Bliss) χρησιμοποιήθηκε με απόλυτη επιτυχία σε ένα παιδί, ενώ ξεκίνησε να διδάσκεται σε άλλα δυο που όμως αργότερα διέκοψαν την διαδικασία. Το πρώτο σύστημα (Mager-Johnson), χρησιμοποιεί απλούστερα σύμβολα, που είναι λιγότερο αφαιρετικά από αυτά του δεύτερου αλλά με πολύ πιο περιορισμένες δυνατότητες και χρησιμοποιείται σε παιδιά με βαρύτερες πνευματικές αναπηρίες.

Ύστερα λοιπόν από σειρά επαφών μεταξύ του παραρτήματος της ΕΛ.Ε.Π.Α.Π. στα Χανιά και του Πολυτεχνείου Κρήτης γεννήθηκε η ιδέα της ανάπτυξης μίας συσκευής με σκοπό την αυτοματοποίηση της διδασκαλίας των γλωσσών συμβόλων. Η συσκευή αυτή θα βοηθήσει τόσο κατά την διάρκεια εκπαίδευσης των ατόμων, όσο και στην μετέπειτα πορεία τους προς την κοινωνική ένταξη αφού θα λειτουργεί σαν ενδιάμεσος σταθμός επικοινωνίας, μετατρέποντας σύμβολα σε λέξεις ή προτάσεις και το αντίστροφο.

Στην τελική της μορφή η συσκευή αυτή θα αποτελεί έναν αυτόνομο, φορητό σταθμό επικοινωνίας. Έτσι θα πολλαπλασιάσει τις δυνατότητες επικοινωνίας των ατόμων αυτών και θα ξεφύγει από τον περιορισμένο χώρο της διδασκαλίας. Εξ' αιτίας των ιδιοτήτων των ατόμων, δημιουργούνται πρόσθετες προδιαγραφές και απαιτήσεις από τη συσκευή. Η λειτουργία της οφείλει λοιπόν να είναι απρόσκοπτη και πλήρως προβλέψιμη. Δεν επιτρέπονται ασάφειες και ασταθής συμπεριφορά από αυτήν, γιατί στα άτομα με ειδικές ανάγκες, ακόμα και η παραμικρή απόκλιση από τη «συνήθη» λειτουργία μπορεί να οδηγήσει σε σύγχυση.

Λόγω της πολυπλοκότητας που παρουσιάζει η συνύπαρξη ήχου, γραφικών και κειμένου κρίνεται απαραίτητη η ύπαρξη επεξεργαστή και εντός της συσκευής, ώστε να συντονίζει την λειτουργία της. Εκτός από το

ειδικό πληκτρολόγιο, που θα παρουσιάζει τα σύμβολα, θα υπάρχει και οθόνη χαρακτήρων, όπου θα γίνεται η σύνθεση των προτάσεων. Επίσης επιβάλλεται η ύπαρξη ηχείου, ώστε να καθίσταται δυνατή η ηχητική αναπαραγωγή των λέξεων. Τέλος, απαραίτητη κρίνεται και η ύπαρξη της δυνατότητας επικοινωνίας με υπολογιστή ώστε να μπορούμε να τροφοδοτούμε την συσκευή με νέα σύμβολα και επιπλέον στοιχεία.

Η μελέτη και ανάπτυξη της εν λόγω συσκευής έχει χωριστεί σε δύο κύρια τμήματα : (α) το υλικό και (β) το λογισμικό μέρος. Το υλικό θα είναι υπεύθυνο για την εμφάνιση των συμβόλων της εκάστοτε γλώσσας και για την δυνατότητα επιλογής κάποιων από αυτά. Αντίθετα το λογισμικό ασχολείται με την δημιουργία, ομαδοποίηση και αντιστοίχιση των συμβόλων με λέξεις .

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο το λογισμικό μέρος της προαναφερθείσας συσκευής. Σαν βασική γλώσσα της συσκευής, σε συνεργασία πάντα με την ΕΛ.Ε.Π.Α.Π. Χανίων, επιλέχθηκε αυτή που χρησιμοποιεί τα σύμβολα Bliss. Η επιλογή αυτή δημιουργεί τις ανάλογες απαιτήσεις από το λογισμικό, αφού αυτό θα πρέπει να είναι προσαρμοσμένο πάνω της. Αυτό βέβαια δεν περιορίζει την συσκευή μια και τα σύμβολα Bliss αποτελούν κατά κάποιον τρόπο ένα υπερσύνολο των άλλων συμβόλων.

Αυτό μας οδηγεί σε μία βαθύτερη ανάλυση των συμβόλων Bliss αλλά και της μεθόδου διδασκαλίας που βασίζεται σε αυτά, ώστε να προκύψουν οι αντίστοιχες απαιτήσεις από το λογισμικό.

ΣΥΜΒΟΛΑ BLISS

1.1 Γενικά για τα σύμβολα (εισαγωγή)

Το σύστημα επικοινωνίας με χρήση των γραφικών συμβόλων Bliss επινοήθηκε από τον Charles K. Bliss το 1949, με στόχο να αποτελέσει μία οικουμενική γλώσσα και έτσι να συμβάλλει στην επίτευξη της παγκόσμιας ειρήνης μεταξύ των λαών. Δυστυχώς όμως η πρόταση αυτή αγνοήθηκε και έτσι το σύστημα εγκαταλείφθηκε για αρκετά χρόνια. Στα τέλη της δεκαετίας του 60 επαναχρησιμοποιήθηκε, απευθυνόμενο όμως αυτή τη φορά σε άτομα με δυσκολίες στην επικοινωνία τους λόγω ανεπαρκειών στην ομιλία, την ακοή ή ακόμα λόγω ψυχικών αναπηριών. Το βασικότερο πρόβλημα των ατόμων αυτών είναι η δυσκολία χρήσης του συμβατικού αλφαβήτου και γραμματικής στη σύνθεση λέξεων και προτάσεων. Εδώ το σύστημα του C.K.Bliss αποτέλεσε μια εναλλακτική πρόταση αλφαβήτου και γραμματικών - συντακτικών κανόνων.

Η “συμβολική” γλώσσα επικοινωνίας Bliss έχει σαν βασικό δομικό της στοιχείο τη χρήση των σχημάτων για την σύνθεση λέξεων και προτάσεων. Τα σχήματα δεν απαιτούν ιδιαίτερη πνευματική προσπάθεια, αφού εμπεριέχουν πληροφορία άμεσα συσχετιζόμενη με την έννοια που πραγματεύονται, σε αντίθεση με την χρήση γραμμάτων - λέξεων που απαιτούν μνημονική σύνδεση με την αντίστοιχη έννοια. Επίσης τα σύμβολα είναι ευκολότερα αναγνωρίσιμα, διακριτά και αποτυπώσιμα από τα άτομα. Έτσι αυξάνεται η ικανότητα επικοινωνίας των ατόμων με το περιβάλλον τους αφού τους δίνεται ένα πιο πρόσφορο μέσο με μικρότερες απαιτήσεις από μέρος τους. Δίνεται λοιπόν η δυνατότητα στα άτομα αυτά, χρησιμοποιώντας “λεξικά” Bliss, να μεταφέρουν μαζί τους μεγάλη ποσότητα πληροφορίας και να την χρησιμοποιούν για να επικοινωνήσουν με τους υπόλοιπους που δεν γνωρίζουν την “γλώσσα” Bliss.

Οι δυνατότητες της γλώσσας, εκτός του πλουσίου λεξιλογίου, είναι πάρα πολλές αφού :

- Η γραμματική της επιτρέπει την χρήση χρόνων στο παρελθόν - παρόν - μέλλον
- Υπάρχουν δείκτες που μπορούν να δηλώσουν κατοχή, ενικό - πληθυντικό αριθμό καθώς και ερωτηματικό - υποθετικό λόγο και προστακτική
- Περιγράφονται αφηρημένες έννοιες όπως η αγάπη, ο φόβος, η ευτυχία, η δυστυχία ...

- Το λεξιλόγιο είναι ανοιχτό στην δημιουργία νέων συμβόλων
- Είναι συνολικά αναγεννήσιμη γλώσσα αφού από την ανάλυση των τμημάτων κάθε “λέξης - συμβόλου” ο “αναγνώστης” μπορεί να την κατανοήσει χωρίς να την έχει διδαχτεί.
- Αναπτύσσεται και προσαρμόζεται στην ικανότητα του κάθε ατόμου αφού τα νέα σύμβολα αποτελούν συνδυασμούς προηγούμενων .^{[1][2][3]}

Η χρησιμοποίηση των συμβόλων Bliss ως μέσο επικοινωνίας ατόμων με ειδικές ανάγκες ξεκίνησε από το “Κέντρο Αναπήρων Παιδιών” στο Τορόντο του Καναδά (Ontario Crippled Children’s Center in Toronto, Ontario Canada). Εκεί ξεκίνησε και η προσαρμογή των αρχικών συμβόλων του C. Bliss για χρήση τους από άτομα με ειδικές ανάγκες και δυσκολίες επικοινωνίας. Αργότερα με την συνδρομή και υποστήριξη του κέντρου αυτού, ιδρύθηκε το Ινστιτούτο Επικοινωνίας με σύμβολα Bliss (Blissymbolics Communication Institute- **BCI**), το οποίο έλαβε και τα αποκλειστικά δικαιώματα για τα σύμβολα Bliss από τον ίδιο τον C. Bliss. Βασικός σκοπός του ινστιτούτου είναι η ανάπτυξη και διάδοση του συστήματος επικοινωνίας, ο καθορισμός και τυποποίηση των συμβόλων σε διεθνές επίπεδο. Τέλος το ινστιτούτο έχει αναλάβει την έκδοση ειδικών εντύπων γύρω από τα σύμβολα Bliss, την καθιέρωση νέων συμβόλων, την ανάπτυξη μέσων και βοηθημάτων επικοινωνίας και τέλος πραγματοποιεί μελέτες και παρακολουθεί το πλήθος των ατόμων που χρησιμοποιεί το σύστημα και των οποίων ο αριθμός αυξάνεται συνεχώς τα τελευταία χρόνια.^{[2][3]}

1.2 Περιγραφή Συμβόλων - Κανόνες

Καθώς είπαμε τα σύμβολα Bliss ακολουθούν ένα διεθνές πρότυπο και έχουν τυποποιημένους κανόνες οργάνωσης και τοποθέτησης.

Μπορούμε να έχουμε σύμβολα που αποτελούνται μόνο από μία “μονάδα” (βασικό σύμβολο) :

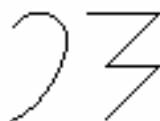


Αυτί (ear)

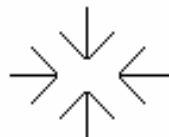


προς τα εμπρός (forwad)

ή ακόμα περισσότερων “μονάδων” :



Κεραυνός(thunder)

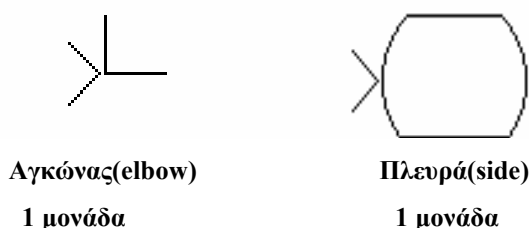


Συγκέντρωση(gathering)

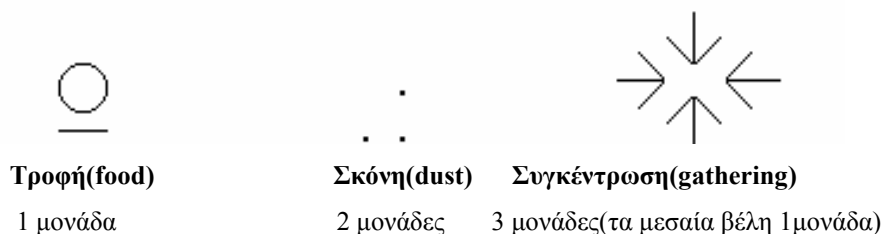
Για να μπορέσουμε να δούμε από πόσες μονάδες αποτελείται το κάθε σύμβολο φανταζόμαστε κατακόρυφες γραμμές μεταξύ των τμημάτων του:



Εξαίρεση αποτελούν όλοι οι δείκτες :

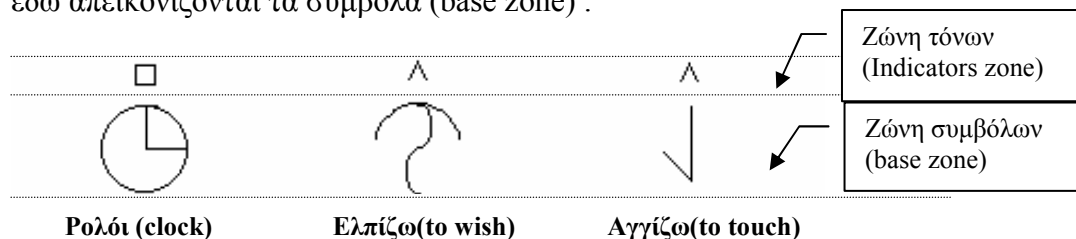


Όταν δύο τμήματα που δεν ακουμπούν μεταξύ τους, αλλά βρίσκονται το ένα πάνω από το άλλο, θεωρούμε ότι ανήκουν στην ίδια μονάδα, έστω και αν μπορούμε να τα ξεχωρίσουμε με μία οριζόντια φανταστική γραμμή :



[1]

Έκτός από τα σύμβολα και ο χώρος απεικόνισής τους διέπεται από κανόνες. Έτσι χωρίζεται σε δύο οριζόντιες ζώνες, η πρώτη καταλαμβάνει το 1/3 του συνολικού χώρου και χρησιμοποιείται για τους τόνους - δείκτες (indicators zone), η δεύτερη καταλαμβάνει τα υπόλοιπα 2/3 και εδώ απεικονίζονται τα σύμβολα (base zone) :



[1]

Τέλος οι τόνοι - δείκτες (indicators) χρησιμοποιούνται για να δηλώσουν κάποιο γενικό χαρακτηριστικό του εκάστοτε συμβόλου, ή να το εντάξουν σε κάποιο μέρος του λόγου (ρήμα, επίθετο, επίρρημα ή και χρόνο, αριθμό ...). Παραδείγματος χάριν :

- □ Δείκτης αντικειμένου (thing) : το σύμβολο είναι πράγμα - αντικείμενο.
- ^ Δείκτης ενέργειας (action) : απαρέμφατο ρήματος ή χρόνος στο παρών (Ενεστώτας)
- v Δείκτης περιγραφής (description) : το σύμβολο είναι επίθετο - επίρρημα



Αγγιξα (touched)



Αγγίζω (to touch)









Θα αγγίξω (will touch)

Γίνεται λοιπόν προφανές πόση λειτουργικότητα και ελευθερία προσδίδουν οι τόνοι - δείκτες στη γλώσσα Bliss.^{[2][3]}

1.3 Κατηγοριοποίηση Συμβόλων

Λόγω του μεγάλου αριθμού των συμβόλων Bliss ($\approx 1400^{[1]}$), γίνεται προφανής η ανάγκη μίας κοινά αποδεκτής κατάταξης τους, σύμφωνα πάντα και με την τυποποίηση που έγινε από το ινστιτούτο BCI. Έτσι αφού τα σύμβολα διαιρέθηκαν σε 26 τελείως βασικά σχήματα, τα σχήματα αυτά αντιστοιχίστηκαν στα 26 γράμματα της Αγγλικής αλφαβήτου. Έχουμε λοιπόν τα “γράμματα Bliss” :

a :  , **b** :  , **c** :  , **d** :  , ... , **x** :  , **y** : 

z : ? στο z αντιστοιχίζονται όλα τα σημεία στίξης, γράμματα και αριθμοί

[1]

Εκτός όμως από την κατηγοριοποίηση για όσους δεν γνωρίζουν την γλώσσα (με τα γράμματα), μπορούμε να χωρίσουμε τα σύμβολα και σύμφωνα με τις έννοιες ή και τα σχήματα αφού παραπλήσια σημασία συνεπάγεται επανάληψη βασικών σχημάτων μέσα στην “λέξη Bliss”.

1.4 Απαιτήσεις των συμβόλων από το λογισμικό

Έχοντας γνωρίσει από κοντά, στην προηγούμενη παράγραφο, τα σύμβολα Bliss, τους κανόνες που τα διέπουν και τις βασικές μορφές ομαδοποίησης τους μπορούμε να προχωρήσουμε στον ορισμό απαιτήσεων που πρέπει να εκπληρώσει το υπ' ανάπτυξη λογισμικό. Στην ανάλυση αυτή θα πρέπει να συμπεριλάβουμε και μερικές εξωτερικές απαιτήσεις (hardware) καθώς και τις επιθυμίες - προτάσεις του φορέα που θα χρησιμοποιήσει τελικά το σύστημα, δηλαδή της ΕΛ.Ε.Π.Α.Π. .

Από το υλικό (hardware) έχουμε τους παρακάτω περιορισμούς :

- Λόγω των οθονών υγρών κρυστάλλων που χρησιμοποιούνται για την απεικόνιση των συμβόλων - λέξεων έχουμε μέγιστη δυνατή ανάλυση για τα σύμβολα 64 * 128 κουκίδες (pixels).
- Απαιτήση χρήσης σειριακής θύρας (RS-232) του υπολογιστή για την επικοινωνία υλικού - λογισμικού .
- Χρήση προσωπικού υπολογιστή (PC) ώστε να έχουμε όσο το δυνατόν ευρύτερη και οικονομικά συμφέρουσα λειτουργία της συσκευής.
- Λόγω του προσωπικού υπολογιστή, θεωρούμε σαν λειτουργικό σύστημα και κατ' επέκταση πλατφόρμα λειτουργίας και ανάπτυξης τα Windows.

Κατόπιν επαφών με την ΕΛ.Ε.Π.Α.Π. και μελετώντας τις ήδη υπάρχουσες συσκευές κρίνεται απαραίτητο να υπάρχουν :

- Σύνδεση συμβόλου Bliss με αντίστοιχη έννοια και ήχο
- Ανάδραση με κείμενο, ηχητικό και γραφικό τρόπο στην εκάστοτε επιλογή συμβόλου από το υλικό.
- Ύπαρξη μετατροπέα - δημιουργού (editor) συμβόλων Bliss, σε γραφικό περιβάλλον, όπου θα παράγονται και αποθηκεύονται τα σύμβολα Bliss.
- Ύπαρξη “σελιδοποιητή”, υπεύθυνου για την κατηγοριοποίηση των λέξεων σε σελίδες και αποθήκευσή τους

Τέλος γνωρίζοντας τα ίδια τα σύμβολα Bliss κρίνεται απαραίτητη και η :

- Ύπαρξη όλου του “αλφαβήτου” Bliss μέσα στο λογισμικό, ακολουθώντας τα ισχύοντα πρότυπα.
- Έχοντας μέγιστη ανάλυση για τις “λέξεις” 128*64, άμεσα καθορίζουμε την ανάλυση για τα γράμματα ως τετράγωνο με πλευρά ίση με τα 2/3 του ύψους της λέξης, δηλαδή 42*42 κουκίδες (pixels).

Έχοντας λοιπόν παρακολουθήσει την ανάλυση του προβλήματος και των απαιτήσεων είμαστε σε θέση να προχωρήσουμε στην επιλογή του μέσου ανάπτυξης. Τα Windows, η συνύπαρξη ήχου, εικόνας, κειμένου και σειριακής επικοινωνίας, μας οδηγούν σε δύο προοπτικές για την πλατφόρμα ανάπτυξης, α) Την γλώσσα προγραμματισμού Visual

C++ και β) την γλώσσα Visual Basic. Όμως η ανάγκη γρήγορης ανάπτυξης και προσαρμογής πάνω στους χρήστες, η έλλειψη απαιτήσεων μέγιστης ταχύτητας, αφού δεν υπάρχουν χρονικά κρίσιμες λειτουργίες, και η απαίτηση φιλικής και εύκαμπτης επικοινωνίας χρήστη – υπολογιστή, μας οδηγούν στη δεύτερη λύση, που όπως θα δούμε στο επόμενο κεφάλαιο ανταποκρίνεται πλήρως στις απαιτήσεις μας.

Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ VISUAL BASIC

2.1 Γενικά για την γλώσσα (εισαγωγή)

Η γλώσσα προγραμματισμού Basic (Beginners All-purpose Symbolic Instruction Code, που αναπτύχθηκε από τους J.Kemeny, T.Kurtz) βρίσκεται σε συνεχή εξέλιξη για 30 και πλέον χρόνια. Σήμερα, κώδικας που ήταν γραμμένος σε προηγούμενες εκδόσεις της γλώσσας (όπως MS-Basic, Qbasic, AppleSoft, QuickBasic ...), συνεχίζει να είναι εκτελέσιμος με ελάχιστες μετατροπές. Παρόλα αυτά το νεότερο στυλ προγραμματισμού έχει εκτοπίσει πολλά από τα αρχικά στοιχεία της γλώσσας. Η Visual Basic περιλαμβάνει γύρω στις 250 βασικές εντολές και συναρτήσεις. Υποστηρίζει αντικειμενοστρεφή προγραμματισμό μέσω του τελεστή (·) που επιτρέπει την “καθοδήγηση” μέσα στις ιεραρχίες των αντικειμένων. Μη επιτρέποντας πολλαπλή κληρονομικότητα (όπως η C++) κάνει τα προγράμματά της πολύ πιο εύκολα στην κατανόηση και αποσφαλμάτωση. Στο προσεχές μέλλον η καθεαυτού γλώσσα δεν πρόκειται να αλλάξει πολύ, οι σημαντικότερες αλλαγές θα έρθουν περισσότερο με την μορφή επεκτάσεων της λειτουργικότητας μέσω “δυνατότερων” νέων αντικειμένων. Μάλιστα κάποιες συναρτήσεις που ήταν μέρος της γλώσσας έχουν ήδη μετακινηθεί σε συγκεκριμένα αντικείμενα (π.χ. Eng object).^[4]

Η Visual Basic είναι κάτι το τελείως διαφορετικό από τις πρώτες εκδόσεις της γλώσσας. Διατηρώντας την ευκολία της Basic, εκμεταλλεύεται σχεδόν το σύνολο των δυνατοτήτων των Windows. Αποτελεί ένα πάντρεμα μεταξύ του κώδικα και των γραφικών αντικειμένων. Η βασική διαφορά που συναντούμε, σε σχέση με τον μονολιθικό προγραμματισμό του DOS, είναι η αρχιτεκτονική των μηνυμάτων (Message Driven) που χαρακτηρίζει το περιβάλλον (Windows) και κατ' επέκταση την γλώσσα.^[5]

2.2 Το περιβάλλον της Visual Basic (Αντικείμενα - Φόρμες - controls)

Βασικές οντότητες της γλώσσας είναι οι φόρμες, που αντιστοιχίζονται στα παράθυρα των Windows. Κάθε πρόγραμμα ξεκινά με τον γραφικό σχεδιασμό των φορμών και τον καθορισμό των ιδιοτήτων τους. Έτσι για παράδειγμα καθορίζουμε το μέγεθος, τον χρωματισμό του φόντου, τον τύπο των γραμματοσειρών καθώς και πλήθος άλλων πιθανών ρυθμίσεων.

Έχοντας λοιπόν σχεδιάσει το βασικό παράθυρο (φόρμα) του προγράμματος μπορούμε να προχωρήσουμε στην εισαγωγή και σχεδιασμό μίας ιδιαίτερης κατηγορίας παραθύρων, τα επονομαζόμενα controls τα οποία εισάγονται μόνο πάνω στις φόρμες. Αυτά είναι παράθυρα που έχουν ενσωματωμένο μέσα τους κώδικα, ώστε να μπορούν να εκτελούν αυτόματα κάποιες συγκεκριμένες λειτουργίες. Τελειώνοντας και με αυτό το βήμα έχουμε ήδη ολοκληρώσει τα δύο πρώτα σκέλη δημιουργίας ενός “παραθυρικού” προγράμματος (δηλ. που τρέχει στα Windows). Σκέλη τα οποία είναι σημαντικό να υπενθυμίσουμε ότι λαμβάνουν χώρα σε καθαρά γραφικό περιβάλλον σχεδίασης, χωρίς απαίτηση εισαγωγής γραμμής κώδικα από πλευράς του προγραμματιστή.

Επόμενο και τελικό στάδιο της δημιουργίας ενός παραθυρικού προγράμματος είναι η εισαγωγή του κώδικα. Μέχρι αυτό το στάδιο έχουμε γνωρίσει μόνο τα γραφικά αντικείμενα της γλώσσας. Εδώ θα συναντήσουμε και τα κλασσικά αντικείμενα. Εκτός λοιπόν από φόρμες και controls η εφαρμογή μας μπορεί να περιλαμβάνει και : α) αρχεία που περιέχουν ρουτίνες και συναρτήσεις γενικής φύσεως και που χρησιμοποιούνται από οποιοδήποτε άλλο τμήμα της εφαρμογής - προγράμματος (Modules), καθώς και β) αρχεία που περιέχουν ολόκληρα αντικείμενα, μαζί δηλαδή με τον κώδικα που περιγράφει τις ιδιότητες και εκτελείται όταν συμβεί κάποιο γεγονός που αναφέρεται στο αντικείμενο, τις κλάσεις δηλαδή (Classes).

Ο κώδικας όμως εκτός από τα συνοδευτικά αρχεία ενσωματώνεται και μέσα στα γραφικά αντικείμενα. Αυτό γίνεται δυνατόν όταν επιλέξουμε κάποιο γραφικό αντικείμενο και κατόπιν το ενεργοποιήσουμε διπλατώντας πάνω του. Αυτόματα μεταφερόμαστε στην κατάλληλη θέση μέσα στον κώδικα του αντικειμένου όπου και μπορούμε να εισάγουμε τον δικό μας κώδικα. Κώδικα που θα εκτελεστεί όταν ενεργοποιηθεί το αντίστοιχο γραφικό αντικείμενο κατά την διάρκεια της εκτέλεσης της εφαρμογής - προγράμματος. Αυτή η εισαγωγή κώδικα είναι ο επονομαζόμενος “προγραμματισμός των γεγονότων” (Event programming), που αποτελεί και την κεντρική ιδέα των ίδιων των Windows και συνοψίζεται στο απόφθεγμα : “μην με καλέσεις αν δεν σε καλέσω εγώ πρώτα”.^{[5][6]}

2.3 Σύνδεση με το περιβάλλον Windows (A.P.I.)

Μία από τις επαναστάσεις που έφεραν τα Windows ήταν εισαγωγή της δυναμικής σύνδεσης βιβλιοθηκών (Dynamic Link Libraries - DLLs). Πριν από την εμφάνισή τους (DOS), η σύνδεση ενός προγράμματος με μία βιβλιοθήκη συναρτήσεων γινόταν στατικά. Αυτό στην ουσία σημαίνει σπατάλη πόρων μια και για κάθε σύνδεση όλη η βιβλιοθήκη (και τα άχρηστα προς το πρόγραμμα μέρη της) αντιγράφεται στην εκτελέσιμη μορφή του προγράμματος και εκτελείται ξανά και ξανά από

τα διάφορα προγράμματα, που δεν έχουν τρόπο να γνωρίζουν ότι η συνάρτηση που καλούν βρίσκεται ήδη στην μνήμη του υπολογιστή. Την σπατάλη αυτή καλύπτει ο δυναμικός τρόπος σύνδεσης. Με τις δυναμικές βιβλιοθήκες υπάρχει μόνο ένα αντίγραφο της βιβλιοθήκης και τα προγράμματα καλούν μόνο αυτό και μοιράζονται στην ουσία τις βιβλιοθήκες. Η Visual Basic μας παρέχει δύο τρόπους κλήσεις μιας συνάρτησης που βρίσκεται σε μία βιβλιοθήκη : α) με χρήση δηλώσεων (Declare statement), όπου λέμε στα Windows πια βιβλιοθήκη περιέχει τις συναρτήσεις που θέλουμε και β) με αναφορά σε βιβλιοθήκη “τύπων” Windows A.P.I. (Application Programmers Interface) που περιέχει τις ίδιες πληροφορίες.

Το API είναι ένα σύνολο συναρτήσεων που είναι διαθέσιμες σε κάθε προγραμματιστή εφαρμογών. Αν έπρεπε να αντιστοιχίσουμε το API με κάποιο τμήμα του DOS αυτό θα ήταν οι συναρτήσεις “διακοπών” του (Interrupt Functions). Αντίστοιχα αν θέλαμε να χρησιμοποιήσουμε συναρτήσεις του EXCEL θα λέγαμε ότι χρησιμοποιούμε το Excel API. Λαμβάνοντας υπ’ όψιν μας ότι το Windows API περιέχει στην κυριολεξία χιλιάδες συναρτήσεις, γίνεται προφανές το μέγεθος των παρεχόμενων δυνατοτήτων.^[7]

2.4 Δυνατότητες της γλώσσας - Κάλυψη των απαιτήσεων

Η γλώσσα προγραμματισμού Visual Basic δεν είναι απλά μία γλώσσα που τα προγράμματά της εκτελούνται στα Windows. Είναι πολύ στενά δεμένη μαζί τους, έτσι τα χαρακτηριστικά της σχεδόν πάντα έχουν άμεση αντιστοίχιση με αυτά των Windows. Αν και δεν είναι η γλώσσα στην οποία είναι γραμμένος ο κώδικας του ίδιου του λειτουργικού συστήματος, αποτελεί κατά κάποιον τρόπο ένα απλοποιημένο υποσύνολο της, μία “εκλαΐκευση” που δεν στερείται όμως καθόλου σε δυνατότητες. Εξάλλου ο σκοπός μας δεν είναι η συγγραφή λειτουργικού συστήματος, παρά εφαρμογών που εκμεταλλεύονται όμως πλήρως τις παρεχόμενες δυνατότητες.

Οι εικόνες, το κείμενο και η επικοινωνία καλύπτονται άμεσα από την ίδια την γλώσσα. Ο ήχος όμως, που αποτελεί και αυτός μέρος των απαιτήσεων της εφαρμογής μας, χρίζει λίγο μεγαλύτερης ευελιξίας στον χειρισμό από την άμεσα προσφερόμενη στην γλώσσα. Εδώ λοιπόν, που αισθανόμαστε κάπως περιορισμένοι από την γλώσσα, την παρακάμπτουμε και απευθυνόμαστε πλέον άμεσα στα ίδια τα Windows μέσω των γνωστών συναρτήσεων API. Με αυτές τις συναρτήσεις μπορούμε να επικοινωνήσουμε άμεσα με τις συσκευές που μας ενδιαφέρουν και να αποφύγουμε την διαμέσω της γλώσσας επικοινωνία, επονομαζόμενη και ως “κέλυφος” της γλώσσας. Μπορούμε λοιπόν τώρα να αντιστοιχίσουμε τις απαιτήσεις της εφαρμογής με τα controls της γλώσσας και τις συναρτήσεις που θα μας “δανείσουν” τα Windows. Έτσι

θα τα γνωρίσουμε από κοντά, πριν τα δούμε σε λειτουργία μέσα στην εφαρμογή που ακολουθεί στο επόμενο κεφάλαιο :

- 1) Για την εικόνα έχουμε δύο controls της γλώσσας που ικανοποιούν τις διάφορες ανάγκες μας :
 - Picture Box και
 - Image Box
- 2) Για το κείμενο χρησιμοποιούμε επίσης δυο :
 - Text Box και
 - Rich Text Box
- 3) Για την επικοινωνία το απαραίτητο control είναι :
 - MSComm
- 4) Τέλος για τον ήχο έχουμε την API συνάρτηση :
 - mciSendStringA της βιβλιοθήκης WinMM

Ξεκινώντας λοιπόν από τα απλούστερα έχουμε :

Text Box : Αποτελεί τον στοιχειώδη μηχανισμό παρουσίασης και εισαγωγής κειμένου και είναι το πιο συνηθισμένο στοιχείο του interface των Windows. Λειτουργεί σαν ένας μικρός κειμενογράφος και ενσωματώνει τους βασικούς χειρισμούς κειμένου (όπως εισαγωγή, επιλογή και κύλιση). Μερικές από τις βασικές του ιδιότητες είναι :

multiline : Επιλέγουμε αν το Text Box θα συγκρατεί μία ή περισσότερες γραμμές κειμένου.(True/False)

MaxLength : Καθορίζουμε το μέγιστο μήκος κειμένου(αριθμός)

Text : Μεταβλητή τύπου string όπου κρατείται το περιεχόμενο κείμενο του TextBox και όπου μπορούμε να εφαρμόσουμε όλες τις συναρτήσεις της γλώσσας για τα strings καθώς και να τροποποιήσουμε κατά την εκτέλεση του προγράμματος.

RichTextBox : Πολύ πιο “δυνατό” από το απλό. Αποτελεί έναν κειμενογράφο απο μόνο του καθώς παρέχει πλήρη έλεγχο πάνω στο κείμενο(στοίχιση, εισαγωγή εικόνων, συνδυασμός πολλαπλών fonts, υποστήριξη RTF κειμένου ...).Εκτός από τις ιδιότητες του απλού TextBox που παραμένουν έχουμε επίσης μερικές από τις πιο συνήθεις του :

SaveFile : Αποθηκεύει αυτόματα τα περιεχόμενα του σε αρχείο στον δίσκο.

LoadFile : “Φορτώνει” αυτόματα το control από αρχείο του δίσκου.

Span : Επιλέγει τμήμα του κειμένου του control βασισμένο σε προκαθορισμένους χαρακτήρες (π.χ.: αν καθορίζαμε το κενό σαν έναν τέτοιο χαρακτήρα η λειτουργία span θα μας επέλεγε μία λέξη κάθε φορά, αφού περικλείεται από κενά (αντίστοιχο με τον γνωστό συνδυασμό στους κειμενογράφους “Ctrl+Shift+arrow”))

UpTo : Μετακινεί τον δείκτη μέσα στο κείμενο έως τους προκαθορισμένους χαρακτήρες (αντίστοιχο με τον γνωστό συνδυασμό στους κειμενογράφους "Ctrl +arrow")

Image Box : Αποτελεί το μικρότερο σε απαιτήσεις - δυνατότητες control, από τα 3, που έχει την δυνατότητα απεικόνισης εικόνων (Bmp,Dib,Wmf ...). Καλύπτει όμως πολλές από τις συνήθεις ανάγκες και σπαταλά τους λιγότερους πόρους. Μερικές από τις ιδιότητες του είναι :

Stretch : Ιδιότητα που υπάρχει μόνο στο Image Box. Αναπροσαρμόζει το μέγεθος της εικόνας στο μέγεθος του.

Picture : Με αυτή την ιδιότητα μπορούμε να προσδιορίσουμε - αναφερθούμε στην εικόνα του Image Box.

LoadPicture - SavePicture : Βασικές μέθοδοι, κοινές σε όλα τα controls που ασχολούνται με εικόνες, με τις οποίες φορτώνουμε - αποθηκεύουμε εικόνες από - προς τον δίσκο.

Picture Box : Το βασικό control χειρισμού εικόνων. Εκτός από την ιδιότητα stretch, έχει τις ίδιες με το Image Box καθώς και τις :

Autosize : Αλλάζει τις διαστάσεις του Picture Box ώστε να χωρέσει την εκάστοτε εικόνα.

AutoRedraw : Ξαναζωγραφίζει αυτόματα όσες περιοχές του control καλύφθηκαν από κάποιο άλλο παράθυρο ή σβήστηκαν καθ' οιονδήποτε τρόπο.

Scale : Πολύ σημαντική ιδιότητα που καθορίζει το σύστημα αναφοράς μέσα στο control. Ρυθμίζει την μονάδα μέτρησης (points, pixels, twips, centimetes, inch ...), αλλά δέχεται και δικές μας τιμές (user), οπότε εκτελεί αυτόματα την μετατροπή από τις πραγματικές διαστάσεις του control στο δικό μας σύστημα συντεταγμένων.

MSComm : Αποτελεί τον χειριστή της σειριακής θύρας. Μπορούμε μέσω αυτού να ελέγξουμε πλήρως κάθε είδους μεταφορά δεδομένων, μέσω πάντα του πρωτοκόλλου RS-232. Έχει σαν ιδιότητες όλες τις ρυθμίσεις του πρωτοκόλλου καθώς και :

CommPort : Επιλέγουμε ποια θύρα θέλουμε να ενεργοποιήσουμε.(Com1/Com2 etc.)

Handshaking : Επιλογή πρωτοκόλλου για τον συγχρονισμό της επικοινωνίας (XonXoff,RTS,none)

InBufferSize-OutBufferSize : Ρύθμιση μεγέθους των καταχωρητών εισόδου - εξόδου.

InputLen : Ρυθμίζει πόσοι χαρακτήρες θα διαβαστούν από τον καταχωρητή εισόδου (input buffer).

InputMode : Επιλέγει την κατάσταση εισόδου της θύρας. Οι δύο περιπτώσεις είναι α) text, όπου διαβάζει χαρακτήρες και β) Binary όπου στην είσοδο έχουμε δυαδικούς αριθμούς άρα δεν γίνεται μετατροπή των αριθμών σε χαρακτήρες.

RThreshold-SThreshold : Κανονίζει μετά από πόσους χαρακτήρες στην είσοδο - έξοδο θα προκληθεί το αντίστοιχο γεγονός της αποστολής - λήψης δεδομένων.

MCI Player (Media Control Interface) : Αποτελεί μέρος των προεκτάσεων πολυμέσων για τα Windows. Στην ουσία είναι σχεδιασμένο να επικοινωνεί με πλήθος συσκευών όπως τα waveform audio, MIDI, CD audio, MPEG video κ.τ.λ. Η λειτουργία του βασίζεται στην επικοινωνία με τους εκάστοτε “οδηγούς” (device drivers) των υπό έλεγχο συσκευών και την έκδοση εντολών προς αυτούς. Οι εντολές μπορεί να σταλούν με δύο τρόπους : α) Απευθείας στους οδηγούς και β) με αποστολή “γραπτής” εντολής (text command) μέσω του MCI, το οποίο είναι πολύ ευκολότερο από πλευράς χρήστη.

mciSendStringA : Είναι συνάρτηση ελέγχου του MciPlayer με συμβολοσειρές (strings) σαν εντολές, της βιβλιοθήκης WinMM. Ο ορισμός της είναι :

```
Declare Function mciSendStringA Lib "WinMM"  
(ByVal MCIcommand As String, ByVal returnStr As  
String, ByVal returnLength As Integer, ByVal callBack  
As Integer) As Long
```

Οι μεταβλητές returnStr και returnLength χρησιμοποιούνται για τον έλεγχο αφού κληθεί η συνάρτηση και μας επιτρέπουν να καταλάβουμε αν εκτελέστηκε σωστά η κάθε εντολή προς το mciplayer ή πιο ακριβώς λάθος συνέβη. Η μεταβλητή callBack δεν υποστηρίζεται από την γλώσσα και δεν θα ασχοληθούμε μαζί της. Τέλος η μεταβλητή MCIcommand χρησιμοποιείται για να μεταφέρει στο mciplayer την αντίστοιχη εντολή προς εκτέλεση, η οποία είναι πάντα σε μορφή κειμένου (text) άρα ορθά φυλάσσεται σε μεταβλητή τύπου χαρακτήρων (string). Οι πιο συνηθισμένες εντολές που δέχεται το mciplayer είναι :

open - close: Ανοίγει - κλείνει την αντίστοιχη συσκευή, στην περιπτώσή μας το waveaudio

play : Αρχίζει την διαδικασία αναπαραγωγής (ήχου, video, cd audio κ.τ.λ.).

stop : Τερματίζει την αναπαραγωγή.

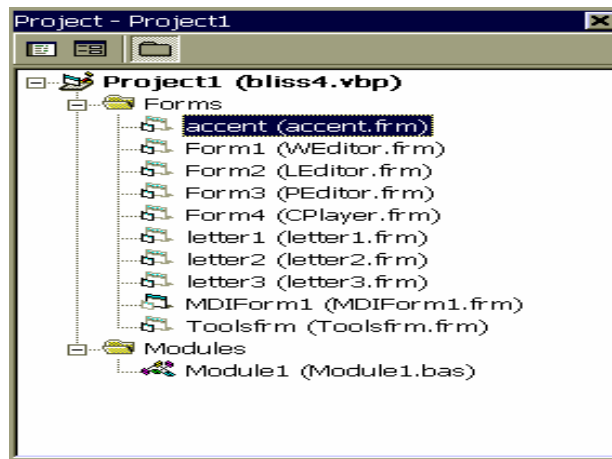
record : Αρχίζει την διαδικασία ηχογράφησης κ.τ.λ.

wait : Λέει στη συσκευή να μην επιστρέψει τον έλεγχο στο πρόγραμμα πριν τελειώσει η αναπαραγωγή.^[6]

ΛΟΓΙΣΜΙΚΟ ΓΙΑ ΤΗΝ ΔΙΔΑΣΚΑΛΙΑ ΤΗΣ ΓΛΩΣΣΑΣ

3.1 Μέρη του προγράμματος

Το πρόγραμμα έχει σαν βασικά του τμήματα τις αντίστοιχες φόρμες της Visual Basic, οι οποίες υλοποιούν τις λειτουργίες του και την επικοινωνία με τον χρήστη. Το σύνολο των φορμών (Forms) και των βοηθητικών αρχείων (Modules), εμφανίζονται στον “εξερευνητή προγράμματος” (Project Explorer) της γλώσσας που ακολουθεί :



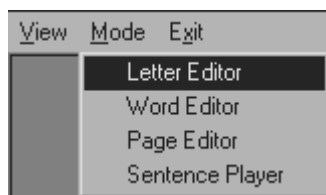
Η βασική φόρμα, η οποία και εμφανίζεται πρώτη κατά την εκκίνηση του προγράμματος, είναι η MDIForm1. Λόγω της ιδιότητας MDI (Multi Document Interface) είναι και η υπεύθυνη για το φόρτωμα και την απεικόνιση όλων των υπολοίπων. Στο γραφικό της αντίστοιχο μπορούμε να δούμε και το menu της, το οποίο κληρονομείται από όλες τις υποφόρμες (“παιδιά”) της (MDIChild forms), που θα το συναντούμε συνεχώς κατά την διάρκεια της περιήγησής μας στα διάφορα μέρη του προγράμματος.

Το 1ο τμήμα του είναι το View. Εδώ μπορούμε να ενεργοποιήσουμε κάποια από τις βοηθητικές φόρμες, όπως την εργαλειοθήκη ή τις παλέτες των γραμμάτων. :



Στο δεύτερο μισό τμήμα του menu βλέπουμε όσα παράθυρα είναι ήδη ανοιχτά και μπορούμε να τα επαναφέρουμε επιλέγοντάς τα. Στο σχήμα είναι όλα ανοιχτά αφού κατά την εκκίνηση φορτώνονται όλα. Με τις βοθητικές παλέτες θα ασχοληθούμε αργότερα.

Το επόμενο τμήμα του menu είναι το Mode. :



Εδώ υπάρχουν και μπορούν να ενεργοποιηθούν όλες οι βασικές λειτουργίες του προγράμματος, αυτές είναι οι : 1) Σχεδιαστής γραμμάτων (Letter Editor), 2) Σχεδιαστής λέξεων (Word Editor), 3) Σχεδιαστής σελίδων (Page Editor) και 4) Παρουσιαστής προτάσεων (Sentence Player), τις οποίες και θα εξετάσουμε εις βάθος στις επόμενες παραγράφους.

Τελευταία επιλογή στο menu είναι η Exit, η οποία και τερματίζει την λειτουργία του προγράμματος. Ακολουθεί και ο αντίστοιχος κώδικας που εκτελείται αφού επιλέξουμε το Exit :

```
Private Sub Exit_Click()
    Unload Form1
    Unload Form2
    Unload Form3
    Unload Form4
    Unload letter1
    Unload letter2
    Unload letter3
    Unload accent
    Unload Toolsfrm
```

```

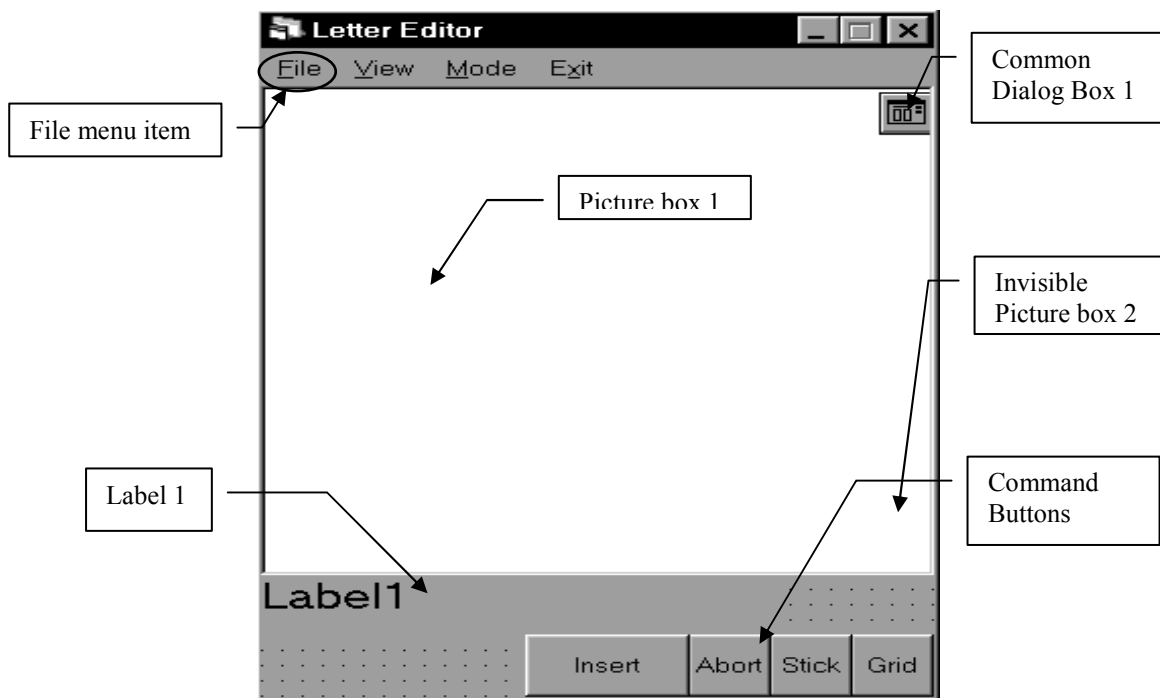
Unload MDIForm1
End
End Sub

```

Βλέπουμε ότι απλά “ξεφορτώνει” τις φόρμες του προγράμματος από την μνήμη και ύστερα με την εντολή End τερματίζει το πρόγραμμα. Με αυτό τον τρόπο αποφεύγουμε την αλόγιστη κατανάλωση πόρων του συστήματος.

i) Letter Editor

Ο Letter Editor, είναι το τμήμα που ασχολείται με τον σχεδιασμό των “γραμμάτων” Bliss. Βασικό του μέρος είναι το Picture box 1 πάνω στο οποίο λαμβάνει χώρα ο σχεδιασμός των συμβόλων. Στο σχήμα που ακολουθεί βλέπουμε το γραφικό κομμάτι των αντικειμένων του “σχεδιαστή γραμμάτων” :



Παρατηρούμε την προσθήκη στο menu της επιλογής για τα αρχεία (File). Οι εικόνες εδώ έχουν σταθερό μέγεθος 42*42 pixels αφού αυτό υπαγορεύεται από τα σύμβολα Bliss και το προκαθορισμένο μέγεθος των LCD's. Έτσι το picture box 1 έχει υπό κλίμακα τις διαστάσεις αυτές (ώστε να τις βλέπουμε καθαρά στην οθόνη). Αντίθετα στο picture box 2, όπου γίνεται και η αποθήκευση – φόρτωση των εικόνων, οι διαστάσεις απαιτείται να είναι ακριβώς οι ίδιες. Τα “κουμπιά εντολών” (command buttons) εκτελούν τις αντίστοιχες λειτουργίες :

- 1) Insert : Δηλώνει ότι επιθυμούμε να εισάγουμε ένα ήδη έτοιμο σύμβολο στο υπό κατασκευή σύμβολο. Αυτό γίνεται δυνατόν επιλέγοντας κάποιο σύμβολο από τις παλέτες συμβόλων, των

οποίων η περιγραφή ακολουθεί στις επόμενες παραγράφους. Η δυνατότητα σχεδιασμού σύνθετων συμβόλων αυξάνει την δυναμικότητα του σχεδιαστή και αποδεικνύεται πολύ χρήσιμη όπως προκύπτει και από τους ανθρώπους της ΕΛ.Ε.Π.Α.Π. .

```
Private Sub Command8_Click()  
    lettertoeditor = True  
    xbase = 0  
    ybase = 0  
End Sub
```

Καθώς βλέπουμε από τον κώδικα που εκτελείται όταν πατηθεί το κουμπί, απλά η μεταβλητή *lettertoeditor* παίρνει την τιμή *TRUE* και δίνονται οι συντεταγμένες όπου θα ξεκινήσει η εισαγωγή του νέου γράμματος.

2) Abort : Ακυρώνει το επιλεγμένο (κοκκινο) σύμβολο και το διαγράφει από την εικόνα.

3) Stick : “Κολλάει” το επιλεγμένο σύμβολο στην εικόνα και το ζωγραφίζει πλέον μαύρο. Ο αντίστοιχος κώδικας είναι :

```
Private Sub Command6_Click()  
    Dim i%, j%  
    For i = 0 To 41  
        For j = 0 To 41  
            lback(xbase + i, ybase + j) = lback(xbase + i,  
                ybase + j) Or letter(i, j)  
        Next j  
    Next i  
    lbacktop2  
    lbacktop1  
    clearletter  
End Sub
```

Βλέπουμε πως ο πίνακας *lback* (περιέχει την πληροφορία της εικόνας 1) γίνεται “ OR ” με τον πίνακα *letter* (περιέχει την πληροφορία του επιλεγμένου “γράμματος”) και κατόπιν αντιγράφεται στις εικόνες 1 και 2 από τις συναρτήσεις *lbacktop1* και *lbacktop2*.

4) Grid : Εμφανίζει – εξαφανίζει το βοηθητικό πλέγμα σχεδίασης :

```
Private Sub Command1_Click()  
    lgrprs = True  
    lgrid (grey)  
End Sub
```

```
Private Sub Command1_MouseUp(Button As Integer, Shift  
As Integer, X As Single, Y As Single)
```



```

    If Button <> 2 Then Exit Sub
    lgrprs = False
    lgrid (white)
End Sub

```

Οι δύο συναρτήσεις που καλούν την συνάρτηση *lgrid* με διαφορετικό χρώμα, γκρι για να εμφανιστεί και άσπρο για να σβηστεί και δίνουν τιμή στην μεταβλητή *lgrprs* ώστε να “βλέπουν” και οι υπόλοιπες συναρτήσεις την κατάσταση του “πλέγματος”(ON / OFF).

Το Label 1, που είναι επικεφαλίδα και χρησιμοποιείται για την εμφάνιση των συντεταγμένων του κέρσορα κατά την κίνησή του μέσα στην εικόνα 1(*picture1*), παίρνει τιμές από την συνάρτηση :

```

Private Sub Picture1_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
    Label1.Caption = " X , Y : " & CStr(CInt(X)) & " , "
& CStr(CInt(Y))

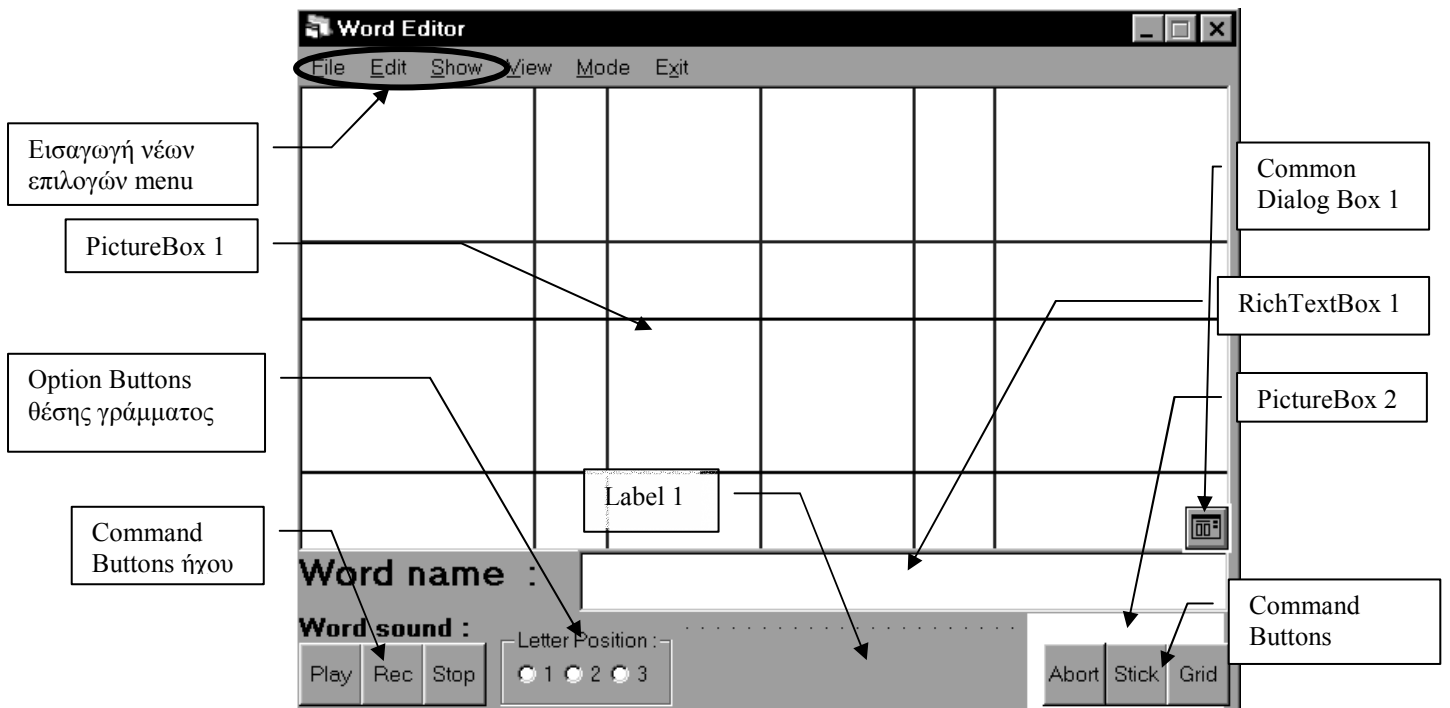
```

Αυτή η συνάρτηση εκτελείται κάθε φορά που μετακινείται ο κέρσορας μέσα στην εικόνα 1(*picture1*) και καθώς βλέπουμε οι μεταβλητές X,Y μετατρέπονται ώστε να γίνουν ακέραιοι αριθμοί και να εμφανιστούν στο Label 1. Τέλος το *CommonDialogBox1* είναι υπεύθυνο για την εμφάνιση των “διαλόγων” εισόδου – εξόδου. Καθώς είναι κοινό σημείο πολλών προγραμμάτων έχει τυποποιηθεί σε μία κοινά αποδεκτή μορφή. Έτσι επιλέγοντας το άνοιγμα ή την αποθήκευση ενός αρχείου βλέπουμε τα συνηθισμένα παράθυρα των Windows.

ii) Word Editor

Στον Σχεδιαστή Λέξεων (Word Editor) γίνεται ο κύριος όγκος της εργασίας όλου του προγράμματος. Εδώ σχεδιάζονται οι λέξεις Bliss και γίνεται η αντιστοίχιση συμβόλων – γραπτού κειμένου (έννοιας) – ήχου. Περιλαμβάνει τα περισσότερα controls και έχει τις περισσότερες απαιτήσεις από οποιοδήποτε άλλο τμήμα. Οι εικόνες εδώ έχουν διαστάσεις 64*128 pixels, που είναι και οι ανάλυση των LCD's. Παρομοίως με τον σχεδιαστή γραμμάτων, στο Picture Box 1 βλέπουμε την ανάλυση αυτή υπό κλίμακα, ενώ στο Picture Box 2 απαιτείται η ακριβής αντιστοίχιση των pixels.

Στο σχήμα που ακολουθεί βλέπουμε όλα τα αντικείμενα της φόρμας, στην γραφική τους πάντα μορφή :



Τα **Command Buttons** της **εικόνας** λειτουργούν με τον ίδιο τρόπο που είδαμε και στον Letter Editor στην προηγούμενη παράγραφο.

Επίσης το **Label 1** και το **Common Dialog** έχουν την ίδια συμπεριφορά με τα αντίστοιχα του Letter Editor.

Τα **Option Buttons** χρησιμοποιούνται για να επιλέξουμε την αρχή της τοποθέτησης των γραμμάτων πάνω στον Editor. Έχουμε λοιπόν τρεις δυνατές περιπτώσεις τοποθέτησης, στα αντίστοιχα τρίτα της εικόνας 1 (Picture Box 1). Στον κώδικα που εκτελείται όταν επιλεγεί κάποιο από αυτά :

```
Private Sub Option1_Click()
    xbase = 0
    opt(1) = True
    ...
End Sub

Private Sub Option2_Click()
    xbase = 42
    opt(2) = True
    ...
End Sub

Private Sub Option3_Click()
    xbase = 84
    opt(3) = True
    ...
```

```
End Sub
```

Βλέπουμε λοιπόν ότι η μεταβλητή *xbase* παίρνει τις τιμές 0, 42, 84 ανάλογα, ώστε να ξεκινά η απεικόνιση του γράμματος – τόνου από την σωστή θέση μέσα στην εικόνα 1. Επίσης η μεταβλητή *opt* γίνεται *TRUE* ώστε να φαίνεται και στις άλλες συναρτήσεις ποιο Option Button έχει επιλεγθεί.

Το **RichTextBox** χρησιμοποιείται για να εισάγουμε την αντίστοιχη ερμηνεία, του σχεδιασμένου συμβόλου. Λόγω της ικανότητας του να αποθηκεύει κείμενο, την στιγμή που αποθηκεύουμε την εικόνα, χρησιμοποιώντας το ίδιο όνομα αρχείου αλλά με διαφορετική κατάληξη, αποθηκεύουμε και το κείμενο. Αυτό κάνει και το τμήμα του κώδικα που ακολουθεί :

```
Private Sub Save_Click()  
    Dim txtfname As String, tmp As String, wavtmp As  
String  
    Dim errorCode As Long  
    Dim returnStr As String * 255  
    If CommonDialog1.filename <> "" Then  
        tmp = CommonDialog1.filename  
        txtfname = Left(tmp, Len(tmp) - 3) & ".txt"  
        SavePicture Picture2.Picture,  
CommonDialog1.filename  
        RichTextBox1.SaveFile txtfname, 1  
        saved = True  
    Else  
        MsgBox ("Must supply File name ... ")  
        SaveAs_Click  
    End If  
End Sub
```

Η μεταβλητή *txtfname* έχει το ίδιο όνομα με αυτό της εικόνας αλλά με διαφορετική κατάληξη και με την ιδιότητα *SaveFile* του control αποθηκεύουμε το “κείμενο” της “λέξης” *Bliss*.

Τα **Command Buttons** του ήχου είναι υπεύθυνα για την αναπαραγωγή και εγγραφή του ήχου που θα αντιστοιχίσουμε στην λέξη *Bliss*. Η κύρια συνάρτηση που καλούν είναι η *pciSendStringA*, την οποία παρουσιάσαμε στο προηγούμενο κεφάλαιο.

Ακολουθεί η συνάρτηση που εκτελείται όταν επιλεγεί το πλήκτρο **Play** :

```
Private Sub Command3_Click()  
    Dim wavfname As String, tmp As String  
    If recprsed Then Exit Sub  
    tmp = CommonDialog1.filename
```

```

    If tmp <> "" Then wavfname = Left(tmp, Len(tmp) -
3) & ".wav"
    PlayWave (wavfname)
    playprsed = True
End Sub

```

Παρατηρούμε ότι, η μεταβλητή *wavfname* περιέχει το όνομα του αρχείου ήχου που θα εκτελέσουμε.

Ακολουθεί η συνάρτηση *PlayWave* :

```

Sub PlayWave(wavfile As String)
    Dim errorCode As Long
    Dim returnStr As String * 255
    Dim returnCode As Integer
    Dim errorStr As String * 255
    Dim tmp As String
    tmp = "open " & wavfile & " alias tmp"
    errorCode = mciSendStringA(tmp, returnStr, 255, 0)
    errorCode = mciSendStringA("seek tmp to 0",
returnStr, 255, 0)
    errorCode = mciSendStringA("play tmp", returnStr,
255, 0)
End Sub

```

Στη μεταβλητή *tmp* σχηματίζεται, σε μορφή string, η εντολή που θα σταλεί για εκτέλεση στο *MCIPlayer*.

Όταν επιλέξουμε αντίστοιχα το πλήκτρο **REC** εκτελείται η ακόλουθη συνάρτηση :

```

Private Sub Command4_Click()
    Dim errorCode As Long
    Dim returnStr As String * 255
    Dim returnCode As Integer
    Dim tmp As String
    If playprsed Then Exit Sub
    tmp = "open " & App.Path & "\words\x.wav alias x"
    errorCode = mciSendStringA(tmp, returnStr, 255, 0)
    errorCode = mciSendStringA("record x overwrite
from 0 to 5000", returnStr, 255, 0)
    recprsed = True
    saved = False
End Sub

```

Παρατηρούμε ότι πρώτα ανοίγουμε το προσωρινό αρχείο “\words\x.wav” όπου και ηχογραφείται το νέο αρχείο. Όταν αργότερα (στο Save) αποφασίσουμε το όνομα της εικόνας το αρχείο θα αποθηκευτεί με το νέο του όνομα.

Τέλος αν επιλέξουμε **Stop** εκτελείται η συνάρτηση :

```
Private Sub Command5_Click()
    Dim errorCode As Long
    Dim returnStr As String * 255
    Dim returnCode As Integer

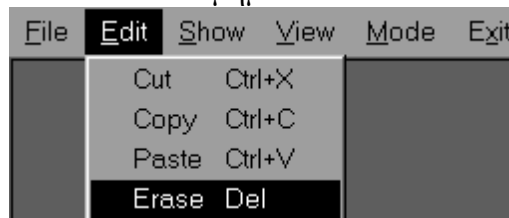
    If recprsed Then
        errorCode = mciSendStringA("save x ", returnStr,
255, 0)
        errorCode = mciSendStringA("close x", returnStr,
255, 0)
        recprsed = False
    End If
    If playprsed Then
        errorCode = mciSendStringA("close tmp",
returnStr, 255, 0)
        playprsed = False
    End If
End Sub
```

Εδώ ανάλογα με το αν το stop ακολουθεί rec ή play, είτε αποθηκεύουμε στο προσωρινό αρχείο, ή απλά απελευθερώνουμε τους δεσμευμένους πόρους από το σύστημα.

Τέλος, οι νέες επιλογές που εμφανίζονται στο menu της βασικής φόρμας είναι οι :

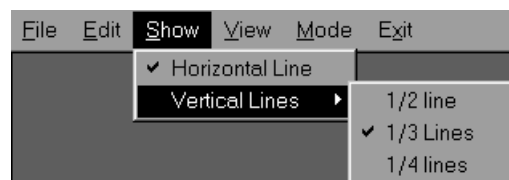
File : Εδώ συναντούμε τις κλασσικές εντολές για την διαχείριση των αρχείων. (New, Open, Save, Save As)

Edit : Σε αυτό το τμήμα του menu καθώς φαίνεται στο σχήμα



υποστηρίζονται οι βασικές λειτουργίες Cut, Copy, Paste, Erase, οι οποίες βοηθούν στην επεξεργασία τμημάτων της εικόνας.

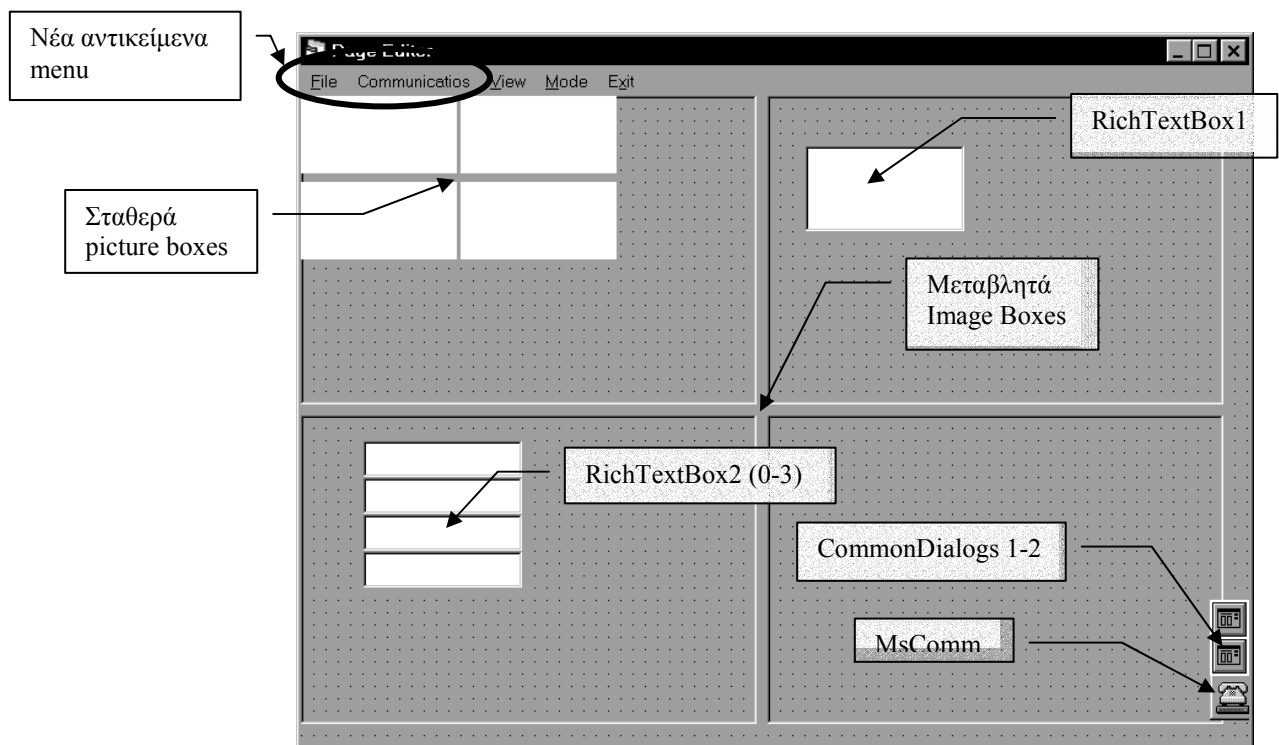
Show : Αυτό το τμήμα του menu είναι υπεύθυνο για την εμφάνιση γραμμών που βοηθούν την σχεδίαση των λέξεων Bliss.



Καθώς βλέπουμε στο ακόλουθο σχήμα μπορούμε να εμφανίσουμε γραμμές στις θέσεις 1 / 2 , 1 / 3 και 1 / 4 :

iii) Page Editor

Ο Page Editor είναι το μέρος του προγράμματος που είναι υπεύθυνο για την σελιδοποίηση των λέξεων Bliss. Στο σχήμα που ακολουθεί μπορούμε να δούμε όλα τα γραφικά αντικείμενα του:



Ακολουθεί η περιγραφή της λειτουργίας των γραφικών αντικειμένων.

Τα δύο **CommonDialogBoxes** χρησιμοποιούνται, το μεν πρώτο για αποθήκευση – ανάκτηση από τον κατάλογο των σελίδων (Pages), ενώ το δεύτερο για αναφορές στον κατάλογο των προσχεδιασμένων λέξεων (Words).

Στα σταθερού μεγέθους αόρατα **PictureBoxes** φορτώνονται οι εικόνες που αντιστοιχούν στις τέσσερις λέξεις τις σελίδας. Κατόπιν οι εικόνες μεταφέρονται στα ορατά από τον χρήστη **ImageBoxes**, τα οποία έχουν την ιδιότητα stretch που “τεντώνει” την εικόνα στις διαστάσεις

τους. Αυτό κάνει δυνατή και την μεταβολή των διατάσεων της φόρμας. Μεταβολή που πραγματοποιεί η συνάρτηση *Form_Resize* που ακολουθεί :

```
Private Sub Form_Resize()  
    If Me.WindowState = vbMinimized Then Exit Sub  
    Image1.Width = Form3.ScaleWidth / 2 - 4  
    Image1.Height = Form3.ScaleHeight / 2 - 4  
    Image2.Width = Form3.ScaleWidth / 2 - 4  
    Image2.Height = Form3.ScaleHeight / 2 - 4  
    Image2.Left = Image1.Width + 2  
    Image3.Width = Form3.ScaleWidth / 2 - 4  
    Image3.Height = Form3.ScaleHeight / 2 - 4  
    Image3.Top = Image1.Height + 2  
    Image4.Width = Form3.ScaleWidth / 2 - 4  
    Image4.Height = Form3.ScaleHeight / 2 - 4  
    Image4.Top = Image1.Height + 2  
    Image4.Left = Image1.Width + 2  
End Sub
```

Εδώ παρατηρούμε ότι κατά τη μετατροπή του μεγέθους της φόρμας, απλώς επαναορίζονται οι διαστάσεις των ImageBoxes σε σχέση με το μέγεθός της που είναι και τα μόνα ορατά αντικείμενά της.

Το RichTextBox 1 χρησιμοποιείται για να αποθηκεύσει – επαναφέρει κάποια σελίδα, αφού αυτή αποτελείται απλά από τα πλήρη ονόματα των αρχείων των εικόνων της, χωριζόμενα από ειδικούς χαρακτήρες. Στο τμήμα του κώδικα που ακολουθεί βλέπουμε την διαδικασία αποθήκευσης μίας σελίδας, καθώς και την χρήση των ιδιοτήτων του RichTextBox :

```
Private Sub SPAGE_Click()  
    Dim txtfname As String  
    If CommonDialog1.filename = "" Then  
        MsgBox ("Must supply File name ... ")  
        SPAs_Click  
    End If  
    RichTextBox1.Text = Pfiles(1) & "!" & Pfiles(2) & "@" &  
Pfiles(3) & "#" & Pfiles(4)  
    txtfname = CommonDialog1.filename  
    RichTextBox1.SaveFile txtfname, 1  
    saved = True  
End Sub
```

Ο πίνακας *Pfiles* συγκρατεί στις θέσεις του τα ονόματα των αρχείων των εικόνων. Στο κείμενο του RichTextBox προσθέτουμε τα ονόματα, παρεμβάλλοντας τους ειδικούς χαρακτήρες : “!, @, #”. Έχοντας σχηματίσει αυτό το σύνθετο όνομα, το αποθηκεύουμε με την ιδιότητα SaveFile του αντικειμένου.

Κατά την διαδικασία της ανάκτησης ακολουθούμε αντίστροφα την σειρά των γεγονότων, καθώς φαίνεται και από την συνάρτηση που ακολουθεί :

```
Private Sub OPage_Click()
    Dim tst As String, txtfname As String
    Dim pos1 As Long, pos2 As Long, pos3 As Long, i As
Long
    CommonDialog1.Flags = cdIOFNReadOnly Or
cdIOFNHelpButton Or cdIOFNHideReadOnly
    CommonDialog1.Filter = "Pages|*.txt"
    CommonDialog1.DefaultExt = ".txt"
    CommonDialog1.ShowOpen
    saved = False
    RichTextBox1.LoadFile CommonDialog1.filename, 1
    tst = RichTextBox1.Text
    pos1 = InStr(tst, "!")
    pos2 = InStr(tst, "@")
    pos3 = InStr(tst, "#")
    bmpfiles(0) = Left(tst, pos1 - 1)
    bmpfiles(1) = Mid(tst, pos1 + 1, pos2 - pos1 - 1)
    bmpfiles(2) = Mid(tst, pos2 + 1, pos3 - pos2 - 1)
    bmpfiles(3) = Right(tst, Len(tst) - pos3)
    ...
End Sub
```

Οι μεταβλητές *pos1*, *pos2*, *pos3* συγκρατούν τις θέσεις στις οποίες βρίσκονται οι ειδικοί χαρακτήρες μέσα στην αρχική συμβολοσειρά. Κατόπιν διαχωρίζουμε στις θέσεις του πίνακα *bmpfiles* την συμβολοσειρά και πλέον έχουμε τα πλήρη ονόματα των αρχείων των εικόνων.

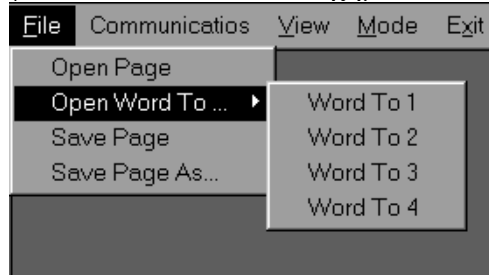
Το RichTextBox 2 είναι πίνακας από τέσσερα RTBoxes στον οποίο μεταφέρεται το κείμενο των τεσσάρων λέξεων της σελίδας. Το κείμενο αυτό εμφανίζεται και στον τίτλο του Page Editor όταν ο κέρσορας κινηθεί πάνω από την αντίστοιχη εικόνα. Την διαδικασία αυτή αναλαμβάνει η ακόλουθη συνάρτηση :

```
Private Sub Image4_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
    Form3.Caption = " Word4 txt : " & RichTextBox2(3).Text
```


End Sub

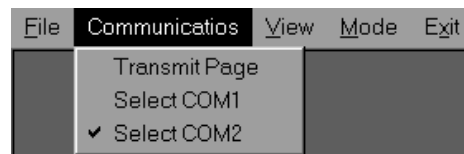
Οι προσθήκες στο menu της βασικής φόρμας είναι δύο :

- Το γνωστό File, που όμως έχει κάποιες τροποποιήσεις καθώς φαίνεται στο ακόλουθο σχήμα :



Παρατηρούμε την προσθήκη της επιλογής Open Word To που μας παροτρύνει να επιλέξουμε μία έτοιμη λέξη Bliss σε αντίστοιχη θέση στην σελίδα.

- Το δεύτερο νέο στοιχείο είναι αυτό των επικοινωνιών. Εδώ έχουμε τις ακόλουθες επιλογές :



Μπορούμε είτε να επιλέξουμε μία από τις δύο σειριακές θύρες του υπολογοστή μας (αυτή στην οποία συνδέσαμε την συσκευή) ή να δώσουμε την εκκίνηση της διαδικασίας αποστολής των δεδομένων.

Η αποστολή των δεδομένων γίνεται μέσω του **MsComm1**, που είναι και το τελευταίο αντικείμενο της περιγραφής μας. Τις περισσότερες ιδιότητες του τις εξετάσαμε αναλυτικά στο προηγούμενο κεφάλαιο. Εδώ θα δούμε μόνο τις αντίστοιχες ρυθμίσεις που κάνουμε ώστε να ελέγξουμε την επικοινωνία μέσω της σειριακής θύρας. Η συνάρτηση που ακολουθεί δείχνει αυτό ακριβώς :

```
Private Sub TPage_Click()  
    Dim i%, j%  
  
    pic2arr  
    arr2byte  
    MSComm1.Settings = "115200,n,8,1"  
    MSComm1.InputMode = comInputModeBinary '???'  
    MSComm1.SThreshold = 4096 '????? all the buffer
```

```
If Com1.Checked Then MSComm1.CommPort = 1
If Com2.Checked Then MSComm1.CommPort = 2
Form3.Caption = "Downloading ... "
MSComm1.PortOpen = True

MSComm1.Output = bit1
MSComm1.Output = bit2
MSComm1.Output = bit3
MSComm1.Output = bit4
MSComm1.PortOpen = False
Form3.Caption = "Download completed .Turning to
Centence Player ..."
Me.WindowState = vbMinimized

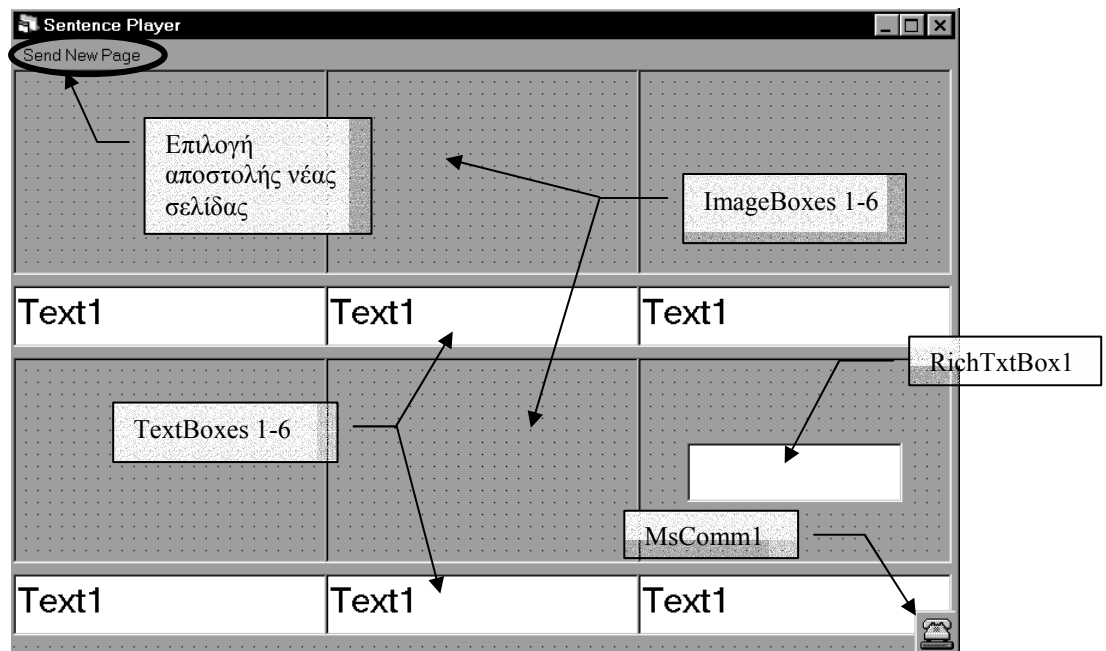
Form4.WindowState = vbNormal
If Com1.Checked Then Form4.MSComm1.CommPort = 1
If Com2.Checked Then Form4.MSComm1.CommPort = 2
Form4.MSComm1.PortOpen = True

End Sub
```

Οι συναρτήσεις *pic2arr* και *arr2byte* μετατρέπουν διαδοχικά την εικόνα σε πίνακα και τον πίνακα σε bytes. Τα bytes με τη σειρά τους, αντιγράφονται στον καταχωρητή εξόδου της θύρας. Τέλος, καλείται ο Sentence Player και η σειριακή ρυθμίζεται, ώστε να δεχτεί την ανάδραση από την συσκευή.

iv) Sentence Player

Ο Sentence Player υλοποιεί την διαδικασία παρουσίασης των επιλογών του χρήστη της συσκευής. Στο σχήμα που ακολουθεί βλέπουμε την γραφική αναπαράσταση των αντικειμένων του :



Ο Page Editor πριν καλέσει τον Sentence Player αποθηκεύει τα ονόματα των αρχείων των εικόνων της σελίδας σε μεταβλητές όπου έχει πρόσβαση ο δεύτερος. Κατά την λήψη λοιπόν από την σειριακή της ανάδρασης της συσκευής, γνωρίζουμε ποιά λέξη έχει επιλεγθεί.

Στον καταχωρητή εισόδου λαμβάνουμε δυο χαρακτήρες, οι οποίοι περιέχουν κωδικοποιημένα και τον χρόνο, το πρόσωπο και τον αριθμό της επιλεγμένης λέξης. Στο τμήμα του κώδικα που ακολουθεί βλέπουμε αυτή την αποκωδικοποίηση :

```
inp = MSComm1.Input
inp = inp & MSComm1.Input
i = CInt(inp)
```

```

inp = CStr(i Mod 4)
If inp = "0" Then inp = "4"
prefix = ""

i = i - (1 Mod 4)
If i >= 48 Then
    plural = True
    prefix = prefix & "(plur.)"
    i = i - 48
End If
If i >= 32 Then
    third = True
    prefix = prefix & "(3rd.)"
    i = i - 32
End If ...

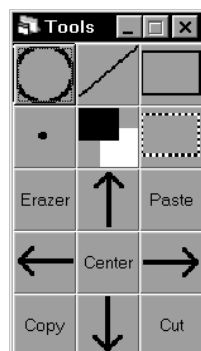
```

Η μεταβλητή *inp* συγκρατεί τα δύο ψηφία του κωδικού αριθμού. Το υπόλοιπο της διαίρεσης με το τέσσερα μας δείχνει ποία από τις λέξεις έχει επιλεγθεί. Τέλος, με διαδοχικές αφαιρέσεις και συγκρίσεις σχηματίζουμε στην μεταβλητή *prefix* το επιθυμητό πρόθεμα, το οποίο διορθώνει την σύνταξη της πρότασης.

3.2 Λειτουργίες - εργαλεία (εργαλειοθήκη - παλέτες)

Εκτός από τους σχεδιαστές, στο πρόγραμμα καθώς είδαμε υπάρχουν και βοηθητικές φόρμες, με εργαλεία και έτοιμα γράμματα Bliss.

A) Η εργαλειοθήκη αποτελείτε από μία συλλογή πλήκτρων με διάφορες λειτουργίες. Στο σχήμα που ακολουθεί βλέπουμε την γενική της μορφή :



Τα τέσσερα πρώτα πλήκτρα, επιλέγουν διαφορετικό σχήμα σχεδίασης κάθε φορά. Το πέμπτο είναι η αλλαγή χρώματος σχεδιασμού, στην περίπτωση μας άσπρο - μαύρο. Το έκτο πλήκτρο επιλέγει την σχεδίαση παραλληλογράμμου επιλογής που χρησιμοποιείται σε

συνδυασμό με τα πλήκτρα Paste, Copy, Cut. Το πλήκτρο Eraser, όταν επιλεγεί “σβήνει” από την εικόνα μας, ζωγραφίζοντας στο αντίθετο χρώμα, ένα τετράγωνο προεπιλεγμένης πλευράς. Οι συναρτήσεις που καλούνται όταν επιλεγεί το πλήκτρο είναι :

```
Private Sub Command7_Click()  
    Static pressed&  
    pressed = Not pressed  
    Call SendMessage(Command7.hWnd, BM_SETSTATE,  
pressed, 0)  
    cmmnd(7) = True  
    If cmmnd(2) Then Call SendMessage(Command2.hWnd,  
BM_SETSTATE, Not (pressed), 0)  
    If cmmnd(1) Then Call SendMessage(Command1.hWnd,  
BM_SETSTATE, Not (pressed), 0)  
    If cmmnd(3) Then Call SendMessage(Command3.hWnd,  
BM_SETSTATE, Not (pressed), 0)  
    If cmmnd(4) Then Call SendMessage(Command4.hWnd,  
BM_SETSTATE, Not (pressed), 0)  
    If cmmnd(6) Then Call SendMessage(Command6.hWnd,  
BM_SETSTATE, Not (pressed), 0)  
    cmmnd(1) = False  
    cmmnd(2) = False  
    cmmnd(3) = False  
    cmmnd(4) = False  
    cmmnd(6) = False  
End Sub
```

```
Private Sub Command7_MouseDown(Button As Integer,  
Shift As Integer, X As Single, Y As Single)  
    Dim width$  
    If Button <> 2 Then Exit Sub  
    width = InputBox("Enter erasers width ...(e.g.  
5)", "Eraser size", "3")  
  
    eraserwidth = CLng(width)  
    eraserheight = eraserwidth  
End Sub
```

Στην πρώτη συνάρτηση βλέπουμε ότι απλά παίρνουν τιμές οι μεταβλητές *cmmnd*, που είναι “σηματοδότες” προς τις υπόλοιπες φόρμες για το ποιο πλήκτρο έχει επιλεγεί. Αντίθετα στη δεύτερη καλείτε ένα *InputBox* όπου ο χρήστης επιλέγει το μέγεθος του τετραγώνου διαγραφής.

Τέλος, τα πλήκτρα κατεύθυνσης μαζί με το πλήκτρο center χρησιμοποιούνται για την μετατόπιση των γραμμάτων μέσα στους σχεδιαστές. Στον κώδικα που ακολουθεί βλέπουμε τι συμβαίνει όταν πατηθεί ένα από αυτά :

```

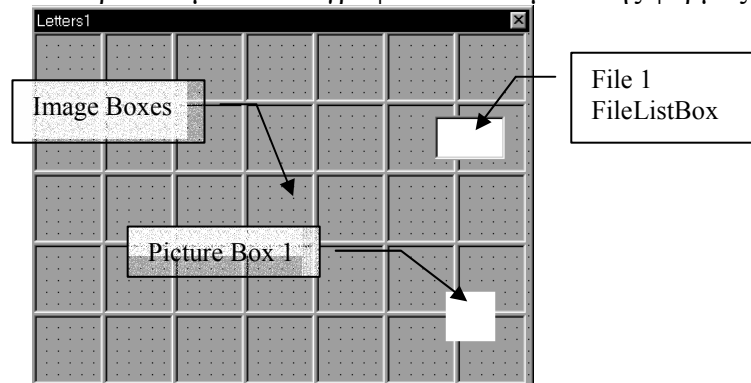
Private Sub Command12_Click() 'right
    xbase = xbase + xstep
    If lettertoeditor Then
        lettertoeditor
    Else
        If acc Then
            showaccent
        Else
            showletter
        End If
    End If
End Sub

Private Sub Command12_MouseDown(Button As Integer,
Shift As Integer, X As Single, Y As Single)
    Dim xstr$
    If Button <> 2 Then Exit Sub
13    xstr = InputBox("Enter Pixel step on X axis
... (e.g. 3)", "X Pixel Step", "1")
    If IsNumeric(xstr) Then
        xstep = CLng(xstr)
    Else
        MsgBox ("Integer values only .Please reeneter
...")
        GoTo 13
    End If
End Sub

```

Η μεταβλητή *xbase* αυξάνεται κατά *xstep*, που είναι η προεπιλεγμένη μετατόπιση κατά τον X άξονα. Κατόπιν ανάλογα με το ποίος σχεδιαστής είναι ενεργός και το αν έχει επιλεγθεί γράμμα ή τόνος, καλείτε η αντίστοιχη συνάρτηση που θα απεικονίσει το γράμμα ή τον τόνο. Η δεύτερη συνάρτηση είναι αυτή που θέτει τιμές στην μεταβλητή *xstep*.

B) Οι παλέτες γραμμάτων είναι μία συλλογή από Image Boxes τα οποία γεμίζουν εικόνες (γράμματα) από την λίστα File 1. Στο σχήμα που ακολουθεί βλέπουμε όλα τα γραφικά αντικείμενα της φόρμας :

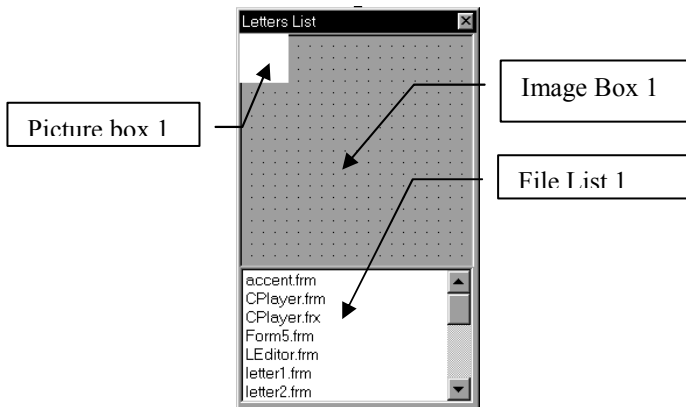


Η λίστα FileListBox είναι συνδεδεμένη με τον υποκατάλογο των γραμμμάτων και τα Image Boxes φορτώνουν από εκεί με την σειρά τις εικόνες τους. Όταν επιλεγθεί κάποιο από αυτά ενεργοποιείται ο ακόλουθος κωδικός :

```
Private Sub Image1_Click(Index As Integer)
    Dim i&, j&
    If lettertoeditor Then
        ybase = 0
    Else
        ybase = 21
    End If
    acc = False
    Picture1.Picture = Image1(Index).Picture
    For i = 0 To 41
        For j = 0 To 41
            If Picture1.Point(i, j) = black Then
                letter(i, j) = 1
            Else
                letter(i, j) = 0
            End If
        Next j
    Next i
    If lettertoeditor Then
        lettertoeditor
    Else
        showletter
    End If
End Sub
```

Παρατηρούμε ότι αναθέτουμε στο άορατο Picture box 1 την εικότα του επιλεγμένου Image Box. Κατόπιν, αφού το Picture Box έχει σταθερές διαστάσεις, δίνουμε τιμές στον πίνακα *letter* αντίστοιχες με το χρώμα των pixels της εικόνας.

Γ) Η παλέτα με την λίστα των γραμμμάτων λειτουργεί με το ίδιο ακριβώς σκεπτικό, μόνο που εδώ ο χρήστης επιλέγει το γράμμα από την λίστα, σύμφωνα με το όνομά του και όχι απευθείας. Στην γραφική αναπαράσταση έχουμε :



Όταν επιλεγθεί κάποιο γράμμα από την λίστα εμφανίζεται στο Image box και αν επιλεγθεί αυτό με την σειρά του, μεταφέρει την εικόνα στο Picture Box. Από εκεί μετατρέπονται τα pixels σε πίνακα, που τελικά θα ζωγραφιστεί στην αντίστοιχη θέση.

3.3 Συναρτήσεις - Υπορουτίνες (Module1.bas)

Σε αυτό το τμήμα θα εξετάσουμε μερικές συναρτήσεις που περιέχονται στο τμήμα Module1.bas του προγράμματος. Θα εξετάσουμε λοιπόν με την σειρά τις ακόλουθες συναρτήσεις 1) *showletter* 2) *borwcolor* 3) *grid* 4) *backtop1* 5) *backtop2*:

1) *showletter* :

```

Sub showletter()
    Dim X, Y As Integer

    backtop1
    For X = 0 To 41
        For Y = 0 To 41
            If letter(X, Y) = 1 Then
                Form1.Picture1.Line (X + xbase, Y +
ybase)-(X + xbase + 1, Y + ybase + 1), red, BF
            End If
        Next Y
    Next X

End Sub

```

Βλέπουμε ότι, ανάλογα με τις τιμές του πίνακα *letter*, στην εικόνα 1 (Picture Box 1) ζωγραφίζουμε με την μέθοδο *Line* ένα τετράγωνο διάστασης 1, το οποίο κατόπιν γεμίζουμε με κόκκινο χρώμα. Έτσι επιτυγχάνουμε να ζωγραφίσουμε τα "pixels" που μας ενδιαφέρουν αλλά στις δικές μας διαστάσεις, ώστε να είναι εύκολα ορατά. Δηλαδή

ένα pixel της αρχικής εικόνας αντιστοιχίζεται σε ένα τετράγωνο με διάσταση 1, όπου το μέτρο το έχουμε θέσει εμείς όμως.

2) *borwcolor* :

```
Function borwcolor() As Long
    If cmmnd(5) Then
        borwcolor = white
    Else
        borwcolor = black
    End If
End Function
```

Ανάλογα με το αν έχει πατηθεί το πλήκτρο που επιλέγει το χρώμα σχεδίασης στη εργαλειοθήκη, η συνάρτηση δίνει στον εαυτό της την τιμή *black* ή *white* .

3) *grid* :

```
Sub grid(color As Long)
    Dim i, j As Integer
    For i = 0 To 127
        If ((i Mod 10) = 0) Then
            Form1.Picture1.DrawWidth = 2
        Else
            Form1.Picture1.DrawWidth = 1
        End If
        Form1.Picture1.Line (i, 0)-(i, 63), color
    Next i
    For j = 0 To 63
        If ((j Mod 10) = 0) Then
            Form1.Picture1.DrawWidth = 2
        Else
            Form1.Picture1.DrawWidth = 1
        End If
        Form1.Picture1.Line (0, j)-(127, j), color
    Next j

    Form1.Line1.Refresh
    Form1.Line2.Refresh
    Form1.Line3.Refresh
    Form1.Line4.Refresh
    Form1.Line5.Refresh
    Form1.Line6.Refresh
    Form1.Line7.Refresh
    Form1.Line8.Refresh
End Sub
```

Παρατηρούμε ότι η συνάρτηση ζωγραφίζει διαδοχικά κάθετες και οριζόντιες γραμμές στην εικόνα 1. Αν το υπόλοιπο της διαίρεσης του τρέχοντος αριθμού γραμμής με το δέκα είναι μηδέν, τότε ζωγραφίζει με

διπλό πάχος την επόμενη γραμμή. Η μεταβλητή *color* καθορίζεται σαν παράμετρος της συνάρτησης και προσδιορίζει το χρώμα σχεδιασμού των γραμμών. Έτσι στην περίπτωση που θέλουμε να σβήσουμε το πλέγμα καλούμε την συνάρτηση με παράμετρο το άσπρο χρώμα. Τέλος, πριν επιστρέψουμε τον έλεγχο στο πρόγραμμα φροντίζουμε να ανανεώσουμε τις βοηθητικές γραμμές σχεδίασης από τυχών επεμβάσεις.

4) *backtop1* :

```
Sub backtop1()  
  
    Dim i, j As Integer  
    Form1.Picture1.Refresh  
    For i = 0 To 127  
    For j = 0 To 63  
        If back(i, j) = 1 Then  
            Form1.Picture1.Line (i, j)-(i + 1, j +  
1), black, BF  
        End If  
    Next j  
    Next i  
    If (gridprsed) Then grid (grey)  
End Sub
```

Αυτή η συνάρτηση λειτουργεί προς αντικατάσταση της μεθόδου *Refresh*. Λόγω του γεγονότος ότι έχουμε θέσει την ιδιότητα *autoredraw Off* στα *Picture Boxes*, ώστε να μη παρεμποδίζεται η σχεδίαση, χρειαζόμαστε έναν τρόπο ανανέωσης της εικόνας. Η βασική της λειτουργία είναι να μεταφέρει τα περιεχόμενα του πίνακα *back* στην εικόνα 1. Τέλος, είναι υπεύθυνη και για την επανεμφάνιση του πλέγματος όποτε αυτό κρίνεται απαραίτητο.

5) *backtop2*:

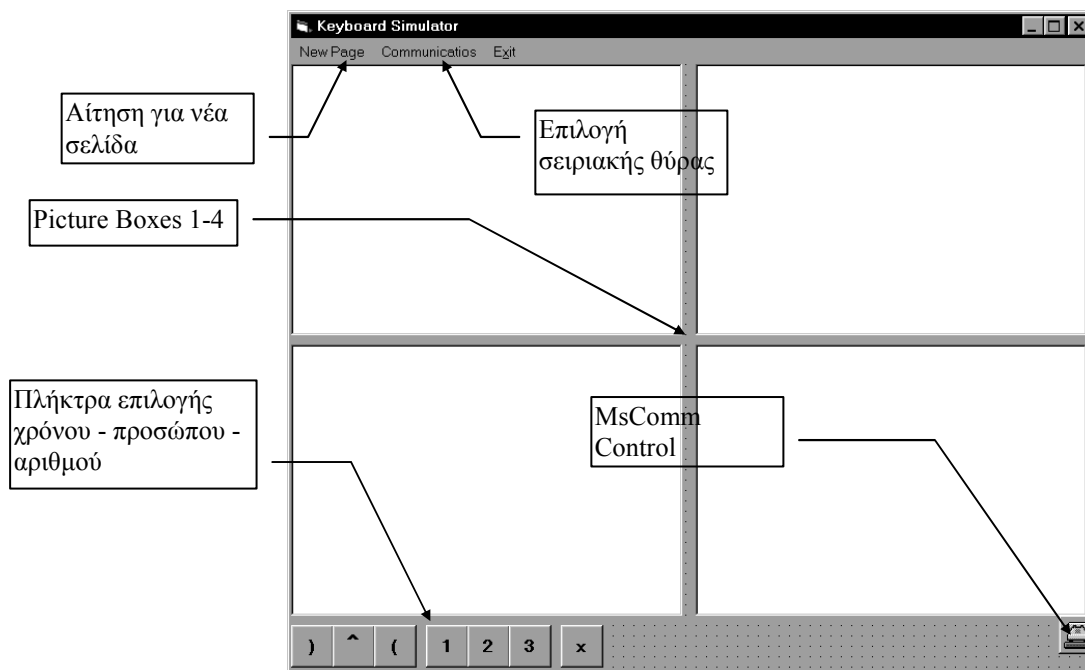
```
Sub backtop2()  
  
    Dim i, j As Integer  
  
    Form1.Picture2.Cls  
    For i = 0 To 127  
    For j = 0 To 63  
        If (back(i, j) = 1) Then  
            Form1.Picture2.PSet (i, j), black  
        Else  
            Form1.Picture2.PSet (i, j), white  
        End If  
    Next j  
    Next i
```

End Sub

Η τελευταία συνάρτηση καλείτε όταν είναι απαραίτητο να ενημερωθεί το σταθερο και αόρατο Picture Box 2 από το πρόγραμμα. Αυτό συμβαίνει κυρίως κατά την επιλογή αποθήκευσης μίας εικόνας, αφού η εικόνα που θα αποθηκευτεί τελικά στο αρχείο είναι αυτή το δεύτερου Picture Box. Η λειτουργία της συνάρτησης περιορίζεται στο να δώσει τιμές στα pixels της εικόνας, ανάλογα με τις τιμές του πίνακα *back*. Έτσι και αλλιώς η εικόνα έχει σταθερές διαστάσεις, ίδιες με αυτές του πίνακα.

3.4 Εξομοιωτής Πληκτρολογίου

Η ανάγκη επίδειξης του προγράμματος μας οδήγησε στην ανάπτυξη αυτού του μικρού εξομοιωτή, που η βασική του λειτουργία είναι να συμπεριφέρεται όπως θα έκανε η αντίστοιχη συσκευή. Στο ακόλουθο σχήμα βλέπουμε την γραφική αναπαράσταση των αντικειμένων που τον απαρτίζουν :



Από την στιγμή εκκίνησής του ο εξομοιωτής "ακούει" την σειριακή θύρα για δεδομένα. Μόλις αρχίσουν να φτάνουν, αυτός τα μετατρέπει πρώτα σε πίνακα και κατόπιν τα παρουσιάζει στα Picture Boxes. Αφού τελειώσει και με αυτό το στάδιο, μας

πληροφορεί για την πρόοδο του στον τίτλο του παραθύρου, ετοιμάζει την σειριακή θύρα για αποστολή και περιμένει την επιλογή μας. Όταν πατηθεί κάποιο Picture Box ενεργοποιείται ο ακόλουθος κώδικας :

```
Private Sub Picture1_Click()  
    Dim tmp As String, i As Integer  
    i = 1  
    If Tence(1) Then i = i + 4  
    If Tence(3) Then i = i + 8  
    If person(2) Then i = i + 16  
    If person(3) Then i = i + 32  
    If plural Then i = i + 48  
    If i < 10 Then  
        tmp = "0" & CStr(i)  
    Else  
        tmp = CStr(i)  
    End If  
    If MSComm1.PortOpen Then MSComm1.PortOpen  
= False  
    If Com1.Checked Then MSComm1.CommPort = 1  
    If Com2.Checked Then MSComm1.CommPort = 2  
    Me.Caption = "Responding word 1..."  
    MSComm1.PortOpen = True  
    MSComm1.Output = tmp  
    MSComm1.PortOpen = False  
End Sub
```

Παρατηρούμε ότι με διαδοχικές συγκρίσεις και προσθέσεις κωδικοποιεί τον συνδυσμό των επιλεγμένων πλήκτρων. Κατόπιν ανοίγει την επιλεγμένη σειριακή θύρα, στέλνει τον κωδικοποιημένο αριθμό και κλείνει ξανά την θύρα, ώστε να επιτραπεί και σε άλλα προγράμματα η πρόσβαση σε αυτή. Η επιλογή στο menu για αίτηση νέας σελίδας, καθαρίζει τα Picture Boxes από τις υπάρχουσες εικόνες και θέτει την σειριακή θύρα σε κατάσταση αναμονής δεδομένων. Η διαδικασία αυτή φαίνεται και στον ακόλουθο κώδικα :

```
Private Sub NPage_Click()  
    Picture1.Cls  
    Picture2.Cls  
    Picture3.Cls  
    Picture4.Cls  
    MSComm1.Settings = "115200,n,8,1"  
    MSComm1.InputMode = comInputModeBinary  
'???'  
    MSComm1.RThreshold = 4095 '???? all the  
buffer  
    MSComm1.InputLen = 4096  
    If MSComm1.PortOpen Then MSComm1.PortOpen  
= False
```

```
If Com1.Checked Then MSComm1.CommPort = 1  
If Com2.Checked Then MSComm1.CommPort = 2  
MSComm1.PortOpen = True
```

End Sub

Κλείνοντας πρέπει να αναφέρουμε ότι το σταθερό μέγεθος των δεδομένων κάνει τον χειρισμό των επικοινωνιών πολύ πιο απλό, αφού δεν απαιτεί ειδικά πρωτόκολλα συντονισμού και αποσφαλμάτωσης.

ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΠΟ ΤΗ ΧΡΗΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ - ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

4.1 Συμπεράσματα από τη χρήση

Ύστερα από επαφές με την ΕΛ.Ε.Π.Α.Π καταλήξαμε στο συμπέρασμα ότι η ανάπτυξη του προγράμματος και κατ' επέκταση της συσκευής κρίνεται επαρκής για την αντικατάσταση του έως τώρα τρόπου διδασκαλίας. Για να επιτευχθεί αυτή η αντικατάσταση όμως, απαιτείται η ύπαρξη περισσότερων πλήκτρων στο πληκτρολόγιο και η προσαρμογή στη νέα διάταξη από το λογισμικό. Στα πλαίσια της παρούσας διπλωματικής εργασίας και του πρώτου σταδίου ανάπτυξης, μπορούμε να πούμε ότι η λειτουργία του συστήματος καλύπτει τις απαιτήσεις μας. Βέβαια, η ανάπτυξη μιας πλήρως αυτόνομης και λειτουργικής συσκευής απαιτεί περισσότερα στάδια, καθώς και πιο “συμπαγή” κατασκευή.

4.2 Μελλοντικές επεκτάσεις

Λαμβάνοντας υπ' όψιν, την αδυναμία που παρουσιάζει το λογισμικό στην γραμματικά σωστή σύνταξη προτάσεων, μπορούμε να εντείνουμε τις προσπάθειες μας σε αυτόν τον τομέα. Η ελληνική γλώσσα όμως, με το πλήθος των διαφορετικών καταλήξεων, πτώσεων, χρόνων, ανωμάτων ρημάτων κ.λ.π είναι πολύ δύσκολο να συστηματοποιηθεί. Άλλο ένα βασικό πρόβλημα είναι η σωστή εκφορά όλων αυτών των περιπτώσεων για κάθε λέξη. Αυτό δημιουργεί απαιτήσεις σύνθεσης φωνής, που γίνεται μόνο από ειδικά κυκλώματα, δύσκολα προσαρμόσιμα στην ελληνική γλώσσα. Μια εναλλακτική πρόταση προς αυτή την κατεύθυνση είναι η χρήση βάσεως δεδομένων. Σε αυτή τη βάση θα αποθηκευθούν όλες οι εικόνες, ο ήχος και τα αντίστοιχα κείμενα και με την χρήση των κατάλληλων ερωτήσεων προς αυτή, να παίρνουμε την γραμματικά και ηχητικά επιθυμητή απάντηση. Εξάλλου η απαίτηση χρήσεως βάσεων δεδομένων, προκύπτει και από την φύση και πλήθος των συμβόλων. Έτσι για την καλύτερη κατηγοριοποίηση των συμβόλων, η χρήση των βάσεων δεδομένων είναι μια αναμενόμενη εξέλιξη.

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΑΡΑΠΟΜΠΕΣ - ΣΕΛΙΔΕΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

- [1] **Barbara Hehner - Blissymbols for Use**, published by the Blissymbolics Communication Institute (BCI) (1986).
- [2] **Shirley McNaughton - Communicating with Blissymbolics**, published by the Blissymbolics Communication Institute (BCI) (1985).
- [3] **Eugene T. McDonald - Teaching and Using Blissymbolics**, published by the Blissymbolics Communication Institute (BCI) (1982).
- [4] **Dan Rahmel - Visual Basic Programmer's Reference**, published by Osborne McGraw - Hill
- [5] **Δημοσθένης Ποσειδών - Poseidon's Visual Basic magazine**
- [6] **Evangelos Petroustos - Mastering Visual Basic 5**, published by SYBEX
- [7] **Dan Appleman - VB 5.0 programmer's guide to the WIN32 API**, published by ZD press
- [8] **William Newman and Michael Lamming – Interactive System Design**, published by Addison Welsey

Bliss on Internet :

- 1 : <http://www.indigo.ie/egt/standards/by/bliss.html>
- 2 : <http://home.istar.ca/~bci/>
- 3 : http://www.disserv.stu.umn.edu/disability/Communication_Disorders/Bliss_Symbols/
- 4 : <http://call-centre.cogsci.ed.ac.uk/OtherLinks>
- 5 : <http://zoom1.telepath.com/gail/bliss/symbols.html>

ΕΥΡΕΤΗΡΙΟ

A
Αριστοτέλης,3