

ΤΙΤΛΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ
Ανάπτυξη γραφικού περιβάλλοντος
για την Ψηφιακή Επεξεργασία Εικόνας
σε MATLAB



Γεράσιμος Δήμας
Νίκος Ράγκος

ΠΡΟΛΟΓΟΣ

Στο πλαίσιο της πτυχιακής εργασίας ασχοληθήκαμε με την ανάπτυξη ενός γραφικού περιβάλλοντος που ασχολείται με την Ψηφιακή Επεξεργασία Εικόνας σε περιβάλλον MATLAB. Για το λόγο αυτό συνεργαστήκαμε με τον επιστημονικό συνεργάτη του Τμήματος Τηλεπληροφορικής και Διοίκησης του ΤΕΙ Άρτας κ. Αντώνιο Μαϊργιώτη, ο οποίος μας υποδύκνει τη χρήση των συναρτήσεων που σχετίζονται με το πεδίο της επιστημονικής περιοχής «Ψηφιακή επεξεργασία εικόνας».

Οι φοιτητές του τμήματος Τηλεπληροφορικής και Διοίκησης του ΤΕΙ Άρτας
Δήμας Γεράσιμος
Ράγκος Νικόλαος



ΚΕΦΑΛΑΙΟ 0. ΕΙΣΑΓΩΓΗ

0.1 Εικόνες και φωτογραφίες

Τα ανθρώπινα όντα είναι κυρίως οπτικά πλάσματα: στηριζόμαστε σε μεγάλο βαθμό στα οπτικά ερεθίσματα για να κατανοήσουμε τον κόσμο γύρω μας. Όχι μόνο εξετάζουμε τα πράγματα και τα προσδιορίζουμε ώστε να μπορέσουμε να τα ταξινομήσουμε, αλλά έχουμε την δυνατότητα να αναζητήσουμε τυχόν διαφορές, και να λάβουμε μια γενική εικόνα για ένα συμβάν με μια γρήγορη ματιά.

Οι άνθρωποι έχουν εξελιγμένες πολύ ακριβείς οπτικές δεξιότητες: έτσι μπορούμε να προσδιορίσουμε ένα πρόσωπο σε μια χρονική στιγμή, μπορούμε να αντιληφθούμε την διαφορετικότητα των χρωμάτων, καθώς γενικά έχουμε την δυνατότητα να επεξεργαστούμε ένα μεγάλο όγκο οπτικών πληροφοριών σε μικρό χρονικό διάστημα πολύ γρήγορα.

Εντούτοις, ο κόσμος κινείται σταθερά: εστιάστε την προσοχή σας σε κάτι για αρκετό καιρό και θα δείτε κάποια αλλαγή. Ακόμη και μια μεγάλη στερεά δομή, όπως ένα κτίριο ή ένα βουνό, θα αλλάξει την εμφάνισή του ανάλογα με το χρόνο της ημέρας (ημέρα ή νύχτα), με την ένταση των ακτίνων του ηλίου πάνω του (ξεκάθαρο ή νεφελώδες), ή διάφορες σκιές που το καλύπτουν.

Ενδιαφερόμαστε για κάθε εικόνα ξεχωριστά: στιγμιότυπα, εάν προτιμάτε από μια οπτική σκηνή. Αν και η επεξεργασία μιας εικόνας μπορεί να ασχοληθεί με τις μεταβαλλόμενες σκηνές, στο κείμενο που ακολουθεί δεν θα αναφερθούμε στις λεπτομέρειες .

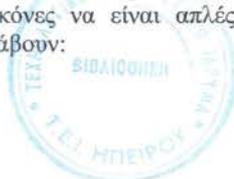
Για δικούς μας λόγους μια εικόνα (image) είναι ένα είδος ζωγραφιάς ή εικόνας που αντιπροσωπεύει ένα γεγονός. Μπορεί να είναι η εικόνα ενός προσώπου, μιας ομάδας ανθρώπων, ζώων, μιας υπαίθριας σκηνής, μιας μικροφωτογραφίας ενός ηλεκτρονικού στοιχείου, ή το αποτέλεσμα της ιατρικής εικόνας. Ακόμα κι αν η εικόνα δεν είναι άμεσα αναγνωρίσιμη, δεν θα είναι μια τυχαία θαμπάδα αλλά κάτι θα συμβολίζει .

0.2 Τι είναι επεξεργασία εικόνας;

Η επεξεργασία εικόνας ασχολείται με την αλλαγή της αρχικής φύσης μιας εικόνας σε μια άλλη

1. βελτιώστε τις εικονογραφικές πληροφορίες για την ανθρώπινη ερμηνεία,
2. μετατρέψτε αυτές σε μορφή κατάλληλη για την αυτόνομη αντίληψη μηχανών.

Θα ενδιαφερθούμε για την **ψηφιακή επεξεργασία εικόνας**, η οποία περιλαμβάνει τη χρησιμοποίηση ενός υπολογιστή για να αλλάξει τη φύση της σε μια ψηφιακή εικόνα . Είναι απαραίτητο να κατανοηθεί ότι αυτές οι δύο πτυχές αντιπροσωπεύουν δύο ξεχωριστές αλλά εξίσου σημαντικές πτυχές της επεξεργασίας εικόνας. Μια διαδικασία που ικανοποιεί τη συνθήκη (1) και κάνει την εικόνα "να φαίνεται καλύτερη" - μπορεί να είναι μία κακή διαδικασία για τη συνθήκη (2) . Ο άνθρωπος προτιμά οι εικόνες να είναι αιχμηρές, σαφείς και λεπτομερείς ενώ οι μηχανές προτιμούν οι εικόνες να είναι απλές και ξεκάθαρες. Παραδείγματα της πρώτης συνθήκης μπορεί να περιλάβουν:



Ενίσχυση των ακμών για να δημιουργηθεί μία αιχμηρότερη εικόνα. Ένα παράδειγμα παρουσιάζεται στην **εικόνα 0.1**. Σημειώστε πώς η δεύτερη εικόνα εμφανίζεται καθαρότερη και για αυτό τον λόγο είναι πιο ευχάριστη εικόνα. Η πραγματοποίηση αιχμηρότερων ακμών είναι ζωτικής σημασίας συστατικό για την εκτύπωση της: για να εμφανίζεται μια εικόνα με το καλύτερο δυνατό αποτέλεσμα κατά την εκτύπωσή της καλό είναι να εκτελείται συνήθως η μέθοδος της όξυνσης-ενίσχυσης (sharpening).



Εικόνα 0.1 Η δεύτερη εικόνα είναι το αποτέλεσμα του Sharpening που εφαρμόστηκε στην αρχική εικόνα

Αφαίρεση του θορύβου από μια εικόνα: ο θόρυβος είναι τα τυχαία λάθη που εμφανίζονται στην εικόνα. Ένα παράδειγμα δίνεται στην **εικόνα 0.2**. Ο θόρυβος είναι ένα κοινό πρόβλημα στη μετάδοση δεδομένων: σε όλα τα είδη των ηλεκτρονικών συστατικών που διέρχονται δεδομένα μπορούν να έχουν επιπτώσεις από τον θόρυβο, και τα αποτελέσματα τις περισσότερες φορές είναι ανεπιθύμητα. Ο θόρυβος έχει διαφορετικές μορφές οπού κάθε μορφή του θορύβου απαιτεί μια διαφορετική μέθοδο για την αφαίρεση του.

Αφαίρεση της θόλωσης (θαμπάδας) κίνησης από μια εικόνα. Ένα παράδειγμα δίνεται στην **εικόνα 0.3**, όπου στην εικόνα (β) είναι εύκολο να διαβαστεί η πινακίδα κυκλοφορίας, να φανούν οι ακτίνες στις ρόδες από το αυτοκίνητο, καθώς επίσης είναι ορατές και άλλες λεπτομέρειες χωρίς να είναι ξεκάθαρες σε ολόκληρη την αρχική εικόνα (α). Η θόλωση κίνησης μπορεί να εμφανιστεί όταν η ταχύτητα του διαφράγματος της φωτογραφικής μηχανής είναι πάρα πολύ μεγάλη για την ταχύτητα του αντικειμένου. Φωτογραφίες με γρήγορη κίνηση των αντικειμένων είναι οι αθλητές καθώς και τα οχήματα όπου το πρόβλημα της θαμπάδας μπορεί να είναι ιδιαίτερα σημαντικό.

Τα παραδείγματα της συνθήκης (2) μπορεί να περιλάβουν:

Λήψη των ακμών μιας εικόνας. Αυτό μπορεί να είναι απαραίτητο για τη μέτρηση των αντικειμένων μέσα σε μια εικόνα. Ένα παράδειγμα παρουσιάζεται στις **εικόνες 0.4**. Έχοντας τις ακμές μπορούμε να μετρήσουμε την έκτασή τους καθώς και την περιοχή η οποία τα περιείχε. Μπορούμε επίσης να χρησιμοποιήσουμε τους αλγορίθμους ανίχνευσης ακμών ως πρώτο βήμα στην ενίσχυση ακμών, όπως είδαμε πιο πριν.



Εικόνα 0.2 Η δεύτερη εικόνα είναι το αποτέλεσμα μετά την απομάκρυνση του θορύβου από την αρχική εικόνα



Εικόνα 0.3 Ξε-θόλωμα εικόνας (Image deblurring). Η δεύτερη εικόνα είναι το αποτέλεσμα της απομάκρυνσης του θολώματος από την πρώτη εικόνα



Εικόνα 0.4 Εύρεση ακμών σε μία εικόνα. Η δεύτερη εικόνα είναι το αποτέλεσμα της εύρεσης ακμών στην πρώτη εικόνα

Αφαίρεση των λεπτομερειών από μια εικόνα. Για λόγους μέτρησης ή αρίθμησης, μπορεί να μην ενδιαφερόμαστε για όλες τις λεπτομέρειες μιας εικόνας. Παραδείγματος χάριν, σε μια μηχανή που χρησιμοποιείται σε μία γραμμή παραγωγής και ελέγχου προϊόντων, τα στοιχεία που μας ενδιαφέρουν μπορεί να είναι το σχήμα, το μέγεθος ή το χρώμα. Σε τέτοιες περιπτώσεις μπορεί να θέλουμε να απλοποιήσουμε την εικόνα. Στην **εικόνα 0.5** παρουσιάζεται ένα παράδειγμα: στην εικόνα (α) είναι μια εικόνα ενός αφρικανικού βούβαλου, και η εικόνα (β) παρουσιάζει την εικόνα του βουβαλιού θολωμένη στην οποία η ξένη λεπτομέρεια (όπως τα κούτσουρα του ξύλου στο υπόβαθρο) έχει αφαιρεθεί. Παρατηρήστε ότι στην εικόνα (β) όλα τα περιττά στοιχεία και οι λεπτομέρειες έχουν αφαιρεθεί και αυτό που παραμένει είναι η χονδροειδής δομή της εικόνας. Θα μπορούσαμε παραδείγματος χάριν, να μετρήσουμε το μέγεθος και το σχήμα του ζώου χωρίς να αποσπαστούμε από την περιττή λεπτομέρεια.

0.3 Εικόνες και ψηφιακές εικόνες

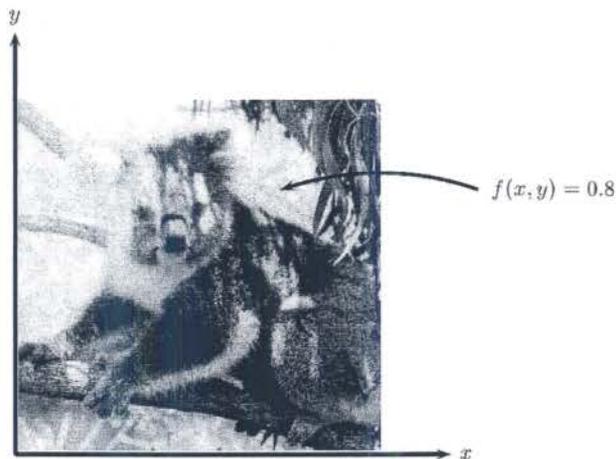
Υποθέστε ότι παίρνουμε μια εικόνα, μια φωτογραφία, για παράδειγμα. Προς το παρόν, αφήνουμε τα πράγματα απλά και υποθέτουμε ότι η φωτογραφία αποτελείται από μαύρο και άσπρο (δηλαδή μέρη των σκιών του γκρι), έτσι ώστε να μην υπάρχει κανένα άλλο χρώμα. Μπορούμε να θεωρήσουμε αυτήν την εικόνα ως μια δισδιάστατη συνάρτηση, όπου οι τιμές της συνάρτησης δίνουν τη φωτεινότητα σε κάθε σημείο, όπως φαίνεται στην **εικόνα 0.6**. Μπορούμε να υποθέσουμε ότι σε μια τέτοια εικόνα οι τιμές που προσδιορίζουν την φωτεινότητα μπορεί να αποτελούνται από οποιοδήποτε πραγματικό αριθμό στο εύρος 0.0 (όπου είναι το μαύρο) έως το 1.0 (όπου είναι το λευκό). Οι σειρές x και y θα εξαρτηθούν σαφώς από την εικόνα, αλλά μπορούν να πάρουν όλες τις πραγματικές τιμές μεταξύ των ελάχιστων και των μεγίστων τους.

Μια ψηφιακή εικόνα διαφέρει από μια φωτογραφία δεδομένου στο ότι τα x , y και $f(x,y)$ οι τιμές τους είναι όλες διακριτές. Συνήθως παίρνουν μόνο τιμές ακέραιων αριθμών, έτσι η εικόνα που παρουσιάζεται στην **εικόνα 0.6** θα έχει x και y όπου θα κυμαίνονται από 1 έως 256 το καθένα τους, και οι τιμές φωτεινότητας θα κυμαίνονται επίσης από 0 (μαύρο) ως 255 (λευκό). Μια ψηφιακή εικόνα μπορεί να θεωρηθεί ως ένα μεγάλο διάνυσμα (πίνακας) διακριτών σημείων, κάθε ένα από τα οποία έχει μία φωτεινότητα που συνδέεται με αυτό.



Εικόνα 0.5. Θόλωση εικόνας (Blurring an image). Η δεύτερη εικόνα είναι η θολωμένη έκδοση της πρώτης, που σαν στόχο έχει την απομάκρυνση των λεπτομερειών

Αυτά τα σημεία ονομάζονται εικονοστοιχεία, ή απλούστερα pixels (picture elements). Τα pixels που περιβάλλουν ένα συγκεκριμένο pixel αποτελούν την γειτονιά του. Μια γειτονιά μπορεί να χαρακτηριστεί από τη μορφή της η οποία είναι ουσιαστικά ένας πίνακας: μπορούμε να μιλήσουμε για μια γειτονιά 3x3, ή για μια γειτονιά 5x7 .



Εικόνα 0.6. Η εικόνα σαν συνάρτηση

Εκτός από πολύ ειδικές περιπτώσεις, οι γειτονιές έχουν περιττούς αριθμούς γραμμών και στηλών. Κάτι τέτοιο εξασφαλίζει ότι το τρέχον pixel είναι στο κέντρο της γειτονιάς. Ένα παράδειγμα μιας γειτονιάς δίνεται στην **εικόνα 0.7**. Εάν μια γειτονιά έχει έναν περιττό αριθμό γραμμών ή στηλών (ή και των δύο), μπορεί να είναι απαραίτητο να διευκρινιστεί ποιο pixel στη γειτονιά είναι το «τρέχον pixel» έτσι ώστε να προσδιοριστεί η γειτονιά του.



Εικόνα 0.7 pixels με μία γειτονιά

0.4 Μερικές εφαρμογές

Η επεξεργασία εικόνας έχει μια τεράστια γκάμα εφαρμογών, όπου σχεδόν κάθε τομέας της επιστήμης και της τεχνολογίας μπορεί να χρησιμοποιήσει τις μεθόδους επεξεργασίας εικόνας. Εδώ παρουσιάζεται μία μικρή λίστα η οποία μας δίνει κάποια ένδειξη του εύρους των εφαρμογών της επεξεργασίας εικόνας.

1. Ιατρική

- Επιθεώρηση και ερμηνεία των εικόνων που λαμβάνονται από τις ακτίνες X, τις ανιχνεύσεις MRI ή CAT,
- ανάλυση των εικόνων κυττάρων, του χρωμοσώματος karyotypes.

2. Γεωργία

- Δορυφορικές / εναέριες απόψεις του εδάφους, π.χ. για να καθορίσουν την έκταση του εδάφους που χρησιμοποιείται για διαφορετικούς σκοπούς ή για να ερευνήσει την καταλληλότητα διάφορων περιοχών για τις διαφορετικές συγκομιδές,
- Επιθεώρηση των φρούτων και των λαχανικών όπου διακρίνονται τα καλά και φρέσκα προϊόντα από τα παλαιά

3. Βιομηχανία

- Αυτόματη επιθεώρηση των στοιχείων σε μια γραμμή παραγωγής;
- Επιθεώρηση των δειγμάτων διάφορων εγγράφων.

4. Επιβολή νόμου

- Ανάλυση των δακτυλικών αποτυπωμάτων,
- Ενίσχυση (sharpening) ή ξεθόλωμα (de-blurring) των εικόνων που παράγονται από φωτογραφικές μηχανές υψηλών ταχυτήτων

0.5 Πτυχές της επεξεργασίας εικόνας

Είναι βολικό να υποδιαιρεθούν οι διαφορετικοί αλγόριθμοι επεξεργασίας εικόνας μέσα σε ευρείες υποκατηγορίες. Υπάρχουν διαφορετικοί αλγόριθμοι για διαφορετικές εργασίες καθώς και προβλήματα, και συχνά θα επιθυμούσαμε να διακρίνουμε τη φύση του συγκεκριμένου στόχου:

Ενίσχυση εικόνας (Image enhancement): Αναφέρεται στην επεξεργασία μιας εικόνας έτσι ώστε το αποτέλεσμα να είναι καταλληλότερο για μια συγκεκριμένη εφαρμογή. Τα παραδείγματα περιλαμβάνουν:

- Ενίσχυση ή ξεθόλωμα μίας εκτός σφαιρικής εστίασης εικόνας (out of focus image)
- Έμφαση των ακμών
- Βελτίωση της αντίθεσης της εικόνας, ή αύξηση της φωτεινότητας μιας εικόνας,

- Αφαίρεση του θορύβου

Ανάκτηση εικόνας (Image restoration): Μπορεί να θεωρηθεί ως αντιστροφή της ζημιάς που γίνεται σε μια εικόνα από μια γνωστή αιτία, παραδείγματος χάριν:

- αφαίρεση της θόλωσης που προκαλείται από τη γραμμική κίνηση,
- αφαίρεση των οπτικών διαστρεβλώσεων,
- αφαίρεση των περιοδικών παρεμβάσεων (periodical interferences)

Κατάτμηση εικόνας (Image segmentation): Αυτό περιλαμβάνει την υποδιαίρεση μιας εικόνας στα μέρη από τα οποία αποτελείται, ή την απομόνωση ορισμένων πτυχών μιας εικόνας:

- Εύρεση των γραμμών, των κύκλων, ή των ιδιαίτερων μορφών σε μια εικόνα,
- Σε μια αεροφωτογραφία, που προσδιορίζει τα αυτοκίνητα, τα δέντρα, τα κτήρια, ή τους δρόμους.

Αυτές οι κατηγορίες δεν είναι ξένες μεταξύ τους. Ένας δεδομένος αλγόριθμος μπορεί να χρησιμοποιηθεί μόνο για την ενίσχυση της εικόνας ή για την αποκατάσταση της εικόνας. Εντούτοις, πρέπει να είμαστε σε θέση να αποφασίσουμε το τι θέλουμε να κάνουμε με την εικόνα μας: απλά να την κάνουμε να φαίνεται καλύτερη με μια ματιά (ενίσχυση), ή να αφαιρέσουμε την ζημιά (αποκατάσταση).

0.6 Μια εργασία επεξεργασίας εικόνας

Θα εξετάσουμε με κάποιες λεπτομέρειες έναν ιδιαίτερο πραγματικό στόχο, και θα δούμε πώς οι ανωτέρω κατηγορίες μπορούν να χρησιμοποιηθούν για να περιγράψουν τα διάφορα στάδια εκτέλεσης αυτού του στόχου. Η εργασία είναι να ληφθούν, με μια αυτόματη διαδικασία, οι ταχυδρομικοί κώδικες από τους φακέλους. Στα επόμενα παρατηρούμε πώς αυτό μπορεί να ολοκληρωθεί:

Απόκτηση της εικόνας. Πρώτα πρέπει να παραγουμε μια ψηφιακή εικόνα από έναν φάκελο εγγράφου. Αυτό γίνεται χρησιμοποιώντας είτε μια CCD φωτογραφική μηχανή, είτε έναν σαρωτή (scanner).

Προεπεξεργασία. Αυτό είναι το βήμα που γίνεται πριν από τη «βασική» εργασία επεξεργασίας εικόνας. Το πρόβλημα εδώ είναι να εκτελεστούν μερικές βασικές εργασίες προκειμένου η εικόνα που θα προκύψει να είναι πιο βολική για την εργασία που ακολουθεί. Σε αυτή την περίπτωση μπορεί να περιληφθεί η ενίσχυση της αντίθεσης, η αφαίρεση του θορύβου, ή ο προσδιορισμός των πιθανών περιοχών οι οποίες περιέχουν τον ταχυδρομικό κώδικα.

Κατάτμηση. Εδώ είναι όπου «παίρνουμε» πραγματικά τον ταχυδρομικό κώδικα ή με άλλα λόγια εξάγουμε από την εικόνα εκείνο το ακριβές τμήμα που περιέχει τον ταχυδρομικό κώδικα

Αντιπροσώπωση και περιγραφή. Αυτοί οι όροι αναφέρονται στην εξαγωγή των ιδιαίτερων χαρακτηριστικών γνωρισμάτων που μας επιτρέπουν να διαφοροποιούμε μεταξύ τους τα αντικείμενα. Εδώ θα ψάχνουμε τις καμπύλες, τις τρύπες και τις γωνίες που μας επιτρέπουν να διακρίνουμε μεταξύ των διαφορετικών ψηφίων που αποτελούν ένα ταχυδρομικό κώδικα.

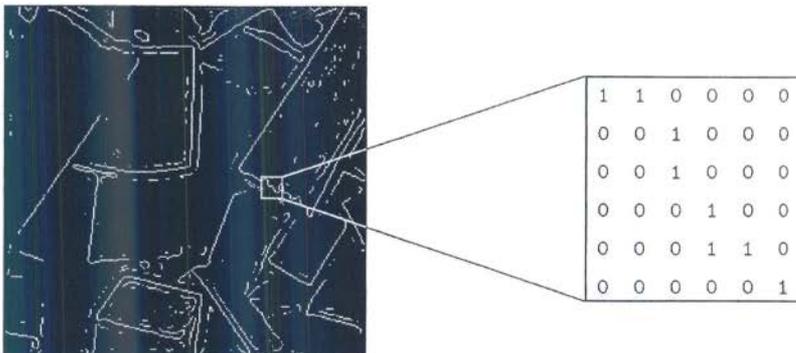
Αναγνώριση και ερμηνεία. Αυτό σημαίνει ότι αναθέτουμε ετικέτες στα αντικείμενα που βασίζονται στους περιγραφείς τους (από το προηγούμενο βήμα) και αναθέτουμε έννοιες σε αυτές τις ετικέτες. Έτσι προσδιορίζουμε συγκεκριμένα ψηφία, και ερμηνεύουμε μια σειρά τεσσάρων ψηφίων στο τέλος της διεύθυνσης ως ταχυδρομικό κώδικα.

0.7 Τύποι ψηφιακών εικόνων

Θα εξετάσουμε τέσσερις βασικούς τύπους εικόνων:

Δυαδικός (Binary). Κάθε pixel είναι είτε μαύρο είτε άσπρο. Δεδομένου ότι υπάρχουν μόνο δύο πιθανές τιμές για κάθε pixel, χρειαζόμαστε μόνο ένα bit ανά pixel. Οι δυαδικές εικόνες δεν έχουν ιδιαίτερες απαιτήσεις όσο αφορά την αποθήκευση τους και για τον λόγο ότι καταλαμβάνουν μικρό αποθηκευτικό χώρο (άρα είναι πολύ αποδοτικές στη διαχείρισή τους). Οι εικόνες οι οποίες είναι προτιμότερο να είναι δυαδικές είναι το κείμενο (τυπωμένο ή απλά γραμμένο με το χέρι), τα δακτυλικά αποτυπώματα ή τα αρχιτεκτονικά σχέδια όπου δεν μας ενδιαφέρουν να απεικονίζονται με τις διάφορες αποχρώσεις του γκρι ή με ένα συνδυασμό χρωμάτων.

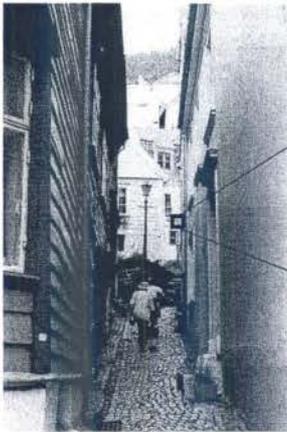
Ένα παράδειγμα ήταν η εικόνα που παρουσιάστηκε στην **Εικόνα 1.4 (β)** πιο πριν. Σε αυτήν την εικόνα έχουμε μόνο τα δύο χρώματα: λευκό για τις άκρες, και μαύρο για το υπόβαθρο. Δείτε την εικόνα 1.8 πιο κάτω.



Εικόνα 0.8. Δυαδική εικόνα

Επιπέδων του γκρι (Greyscale-αποχρώσεις του γκρι). Κάθε pixel είναι μια σκιά του γκρι, κανονικά από 0-(μαύρο) έως 255-(λευκό). Αυτό το εύρος σημαίνει ότι κάθε pixel μπορεί να αντιπροσωπευθεί από οκτώ bit, ή ακριβώς από ένα byte (ψηφιολέξη). Αυτό είναι ένα πολύ

φυσικό εύρος για χειρισμό εικόνας. Αν και χρησιμοποιούνται και άλλα greyscale εύρη , γενικά είναι όλα δυνάμεις του 2 . Τέτοιες εικόνες προκύπτουν στην ιατρική (ακτίνες-Χ), είναι εικόνες τυπωμένων εργασιών, και όντως 256 διαφορετικά επίπεδα του γκρι είναι επαρκή για την αναγνώριση των περισσότερων φυσικών αντικειμένων.



230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Εικόνα 0.9. Παράδειγμα εικόνας επιπέδων του γκρι (greyscale image)

Ένα παράδειγμα είναι η σκηνή οδών που παρουσιάζεται στην **Εικόνα 0.1** πιο πάνω, και στην **Εικόνα 0.9**.

Αληθινό χρώμα (true colour), ή RGB. Εδώ κάθε pixel έχει ένα συγκεκριμένο χρώμα όπου αυτό το χρώμα περιγράφεται από το ποσό του κόκκινου, του πράσινου και του μπλε μέσα σε αυτό. Εάν κάθε ένα από αυτά τα συστατικά εύρος 0 – 255, αυτό μας δίνει συνολικά $255^3 = 16,777,216$ διαφορετικά πιθανά χρώματα στην εικόνα. Αυτά είναι αρκετά χρώματα για οποιαδήποτε εικόνα. Δεδομένου ότι ο συνολικός αριθμός κομματιών που απαιτούνται για κάθε pixel είναι 24, αυτές οι εικόνες ονομάζονται επίσης *24-bit colour images*.

Μια τέτοια εικόνα μπορεί να θεωρηθεί ότι αποτελείται από ένα «σωρό» τριών πινάκων οι οποίοι αναπαριστούν τις κόκκινες, τις πράσινες και τις μπλε τιμές για κάθε pixel. Αυτό σημαίνει ότι για κάθε pixel αντιστοιχούν τρεις τιμές. Παρουσιάζουμε ένα παράδειγμα στην **εικόνα 0.10**.



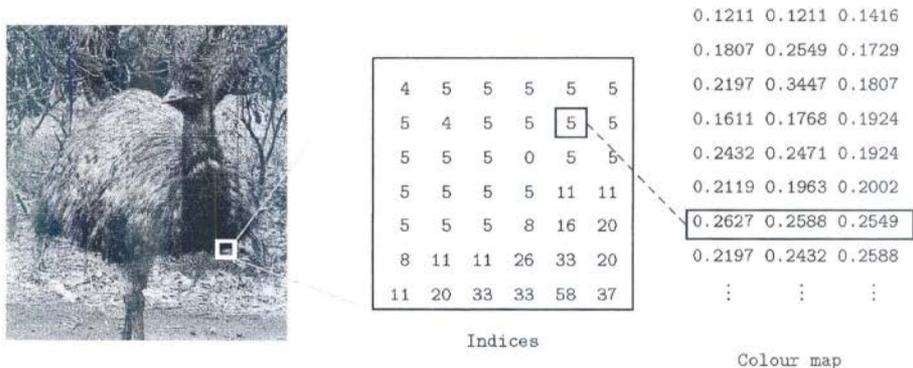
49	55	56	57	52	53	64	76	82	79	78	78	66	80	77	80	87	77
58	60	60	58	55	57	93	93	91	91	86	86	81	93	96	99	86	85
58	58	54	53	55	56	88	82	88	90	88	89	83	83	91	94	92	88
83	78	72	69	68	69	125	119	113	108	111	110	135	128	126	112	107	106
88	91	91	84	83	82	137	136	132	128	126	120	141	129	129	117	115	101
69	76	83	78	76	75	105	108	114	114	118	113	95	99	109	108	112	109
61	69	73	78	76	76	96	103	112	108	111	107	84	93	107	101	105	102
Red						Green						Blue					

Εικόνα 0.10. Μία εικόνα πραγματικών χρωμάτων

Δεικτών (Indexed). Οι περισσότερες εικόνες χρώματος έχουν μόνο ένα μικρό υποσύνολο από τα δέκα έξι εκατομμύρια πιθανά χρώματα. Για την ευκολία της αποθήκευσης και του χειρισμού του αρχείου, η εικόνα είναι συνδεδεμένη με ένα χάρτη χρωμάτων, ή παλέτα χρωμάτων όπως αποκαλείται, ο οποίος είναι απλά ένας κατάλογος όλων των χρωμάτων που χρησιμοποιούνται σε μια συγκεκριμένη εικόνα. Κάθε pixel έχει μια τιμή η οποία δεν δίνει το χρώμα της (όπως για μια RGB εικόνα), αλλά είναι ένας δείκτης (index) ο οποίος δείχνει ένα συγκεκριμένο χρώμα στο χάρτη.

Είναι βολικό εάν μια εικόνα έχει 256 χρώματα ή λιγότερα, για την τιμή του δείκτη να χρησιμοποιηθεί ένα byte για την αποθήκευση. Μερικές επεκτάσεις αρχείων εικόνας (π.χ. GIF Compruserve), επιτρέπουν μόνο 256 χρώματα ή λιγότερα σε κάθε εικόνα για αυτόν ακριβώς τον λόγο.

Η εικόνα 1.11 παρουσιάζει ένα παράδειγμα. Σε αυτήν την εικόνα οι δείκτες, δεν είναι οι τιμές του γκρι των εικόνων, είναι απλά δείκτες στο χάρτη χρώματος. Χωρίς το χάρτη χρώματος, η εικόνα θα ήταν πολύ σκοτεινή και άχρωμη. Στην **εικόνα 0.11**, π.χ. τα εικονοστοιχεία επονομαζόμενα 5 αντιστοιχούν σε 0,2627 0,2588 0,2549 το οποίο είναι ένα σκοτεινό γκρι χρώμα.



Εικόνα 0.11. Μία indexed έγχρωμη εικόνα

0.8 Μορφοποιήσεις (format) εικόνων

1. JPEG

Το JPEG (προφέρεται «τζέϊ-πεγκ») είναι ένας τυποποιημένος μηχανισμός συμπίεσης εικόνας, ο οποίος έχει πάρει το όνομα του από τα αρχικά της επιτροπής (Joint Photographic Experts Group) που έγραψε το πρότυπο.

Το JPEG είναι κατασκευασμένο για συμπίεση είτε εγχρωμών είτε ασπρόμαυρων εικόνων και δουλεύει πολύ καλά σε φωτογραφίες, εικόνες φυσικής τέχνης και παρόμοια είδη αλλά όχι τόσο καλά σε γραφή και απλά ή γραμμικά σχέδια.

Το JPEG είναι μια μέθοδος που έχει απώλειες κατά την συμπίεση της εικόνας, δηλαδή η αποσυμπιεσμένη εικόνα δεν είναι ακριβώς η ίδια με αυτή που είχαμε πριν τη συμπίεση. Παρόλα αυτά λόγω των περιορισμών της ανθρώπινης όρασης, κυρίως του ότι μια μικρή αλλαγή στα χρώματα γίνεται λιγότερο αντιληπτή από μια αντίστοιχη αλλαγή στην φωτεινότητα, οι διαφορές δεν γίνονται αντιληπτές με γυμνό μάτι. όμως αν μας ενδιαφέρει η

λεπτομέρεια στα χρώματα, κυρίως στην περίπτωση που η αποσυμπιεσμένη εικόνα θα περάσει από μηχανήματα ανάλυσης εικόνας τότε η μέθοδος συμπίεσης JPEG δεν είναι συνιστώμενη. Υπάρχουν βέβαια άλλες μέθοδοι που έχουν λιγότερες απώλειες αλλά δεν έχουν τόσο υψηλό ποσοστό συμπίεσης (π.χ. TIFF, PPM, PNG κλπ.).

Το γεγονός ότι έχουμε απώλειες δεν πρέπει να μας κάνει να νομίζουμε ότι η μέθοδος συμπίεσης JPEG μειονεκτεί σε σχέση με άλλες, αφού δεν υπάρχει πρότυπο ψηφιακής εικόνας που να διατηρεί όλες τις ορατές πληροφορίες, ενώ σε σχέση με την GIF συμπίεση, η JPEG χάνει πολύ λιγότερες πληροφορίες.

Μια ενδιαφέρουσα ιδιότητα της μεθόδου JPEG είναι ότι το ποσοστό απωλειών μπορεί να ρυθμιστεί παραμετρικά κατά την συμπίεση. Αυτό σημαίνει ότι ανάλογα με τις απαιτήσεις μας μπορούμε να «ανταλλάξουμε» την ποιότητα της εικόνας με το μέγεθος του αρχείου που την περιέχει. Έτσι, μπορούμε να δημιουργήσουμε πολύ μικρά αρχεία με τις εικόνες που θα χρησιμοποιούν ως δείκτες για το τι περιέχουν τα αντίστοιχα αρχεία μεγάλου μεγέθους (και υψηλής ποιότητας) των εικόνων αυτών. Η χρήση της παραπάνω μεθόδου βρίσκει εφαρμογή σε WEB PAGES, όπου μια περίληψη ενός link σε εικόνα δίνεται για να αποφευχθεί άσκοπο downloading και χάσιμο χρόνου μετακίνησης μεταξύ διαφόρων PAGES.

Μια άλλη ενδιαφέρουσα ιδιότητα του JPEG είναι ότι οι αποκωδικοποιητές μπορούν να μειώσουν την ποιότητα της εικόνας σε αντάλλαγμα με την ταχύτητα αποκωδικοποίησης. Με αυτό το τρόπο μερικοί viewers πετυχαίνουν σημαντικές ταχύτητες προβολής εικόνων JPEG.

2. TIFF

Το TIFF είναι ακρωνύμιο του **Tag Image File Format**. Ο προσδιορισμός TIFF αρχικά κυκλοφόρησε το 1986 από την Albus Corporation ως μια στάνταρ μέθοδος αποθήκευσης ασπρόμαυρων εικόνων, που έχουν δημιουργηθεί από σαρωτές και από εφαρμογές desktop publishing. Αυτή η πρώτη δημόσια κυκλοφορία ήταν στην πραγματικότητα η τρίτη βελτιωμένη έκδοση του TIFF, έτσι αυτή η πρώτη κυκλοφορία μπορεί να θεωρηθεί ως η TIFF 3.0 έκδοση. Η πρώτη ευρέως χρησιμοποιούμενη έκδοση η TIFF 4.00, κυκλοφόρησε τον Απρίλιο 1987. Στο TIFF 4.0 προστέθηκε η υποστήριξη για ασυμπίεστες RGB εικόνες με χρώμα και γρήγορα την ακολούθησε η κυκλοφορία του TIFF 5.0, τον Αύγουστο 1988. Το TIFF 5.0 ήταν η πρώτη βελτιωμένη έκδοση που πρόσθεσε την ικανότητα να αποθηκεύει εικόνες με χρώματα παλέτας και υποστήριξε τον LZW αλγόριθμο συμπίεσης. Το TIFF 6.0 κυκλοφόρησε τον Ιούνιο 1992 και πρόσθεσε υποστήριξη για CMYK και YCbCr έγχρωμες εικόνες και την JPEG μέθοδο συμπίεσης. Σήμερα το TIFF είναι ένα στάνταρ format αρχείων που βρίσκεται στα περισσότερα paint, imaging και desktop publishing προγράμματα και είναι ένα format που προέρχεται από το Microsoft Windows GUI (graphics User Interface). Η επεκτάσιμη φύση του TIFF επιτρέποντας αποθήκευση πολλαπλών raster εικόνων, οποιουδήποτε βάθους pixel, το κάνει ιδανικό για τις περισσότερες ανάγκες αποθήκευσης εικόνων.

Το TIFF έχει κερδίσει μία φήμη για δύναμη και ευκαμψία, αλλά θεωρείται το ίδιο περίπλοκο και μυστηριώδες. Στο σχεδιασμό του το TIFF προσπαθεί να είναι πολύ επεκτάσιμο και να εξασφαλίζει πολλά χαρακτηριστικά που ο προγραμματιστής μπορεί να χρειάζεται σε ένα format αρχείου. Επειδή το TIFF είναι πολύ επεκτάσιμο και έχει αρκετές ικανότητες πέρα από αυτές των άλλων format αρχείων εικόνας, αυτό το format είναι περίπλοκο format για να το κατανοήσεις και να το χρησιμοποιήσεις.

3. GIF

Αναπτύχθηκε από την CompuServe με στόχο τη διευκόλυνση της ανταλλαγής εικόνων μέσω δικτύου. Υποστηρίζει συμπίεση εικόνας που βασίζεται στο αλγόριθμο LZW (Lempel -Zin - Welch). Ο αλγόριθμος αυτός ανήκει στην κατηγορία των διανυσματικών τεχνικών συμπίεσης

και λειτουργεί χωρίς απώλειες. Οι λόγοι συμπίεσης που επιτυγχάνει είναι 4:1 αλλά περιορίζεται σε εικόνες βάθους χρώματος των 8 bits (256 χρώματα).

Το GIF είναι ένα format που έχει οριστεί και τεκμηριωθεί σωστά σε ευρεία χρήση το οποίο διαβάζεται εύκολα και γρήγορα και αποσυμπιέζεται εύκολα. Παρόλα αυτά έχουν παρατηρηθεί και κάποια προβλήματα.

4. BMP

Πρόκειται για το format που υποστηρίζουν εγγενώς τα MS Windows γι' αυτό και συναντάται πολύ συχνά. Πρόκειται ίσως για το πιο απλό format, το οποίο δεν προσφέρει καμιά δυνατότητα συμπίεσης. Απλώς περιγράφει το τρόπο με τον οποίο θα αποθηκευτούν τα bits της εικόνας σε ένα αρχείο ο οποίος είναι μάλιστα ανεξάρτητος της μονάδας εξόδου. Μπορούν να αποθηκεύσουν εικόνες με 1-bit (μονόχρωμες), 4-bit (16 χρώματα), 8-bit (256 χρώματα) και 24-bit (16 εκατομμύρια χρώματα) ανά pixel.

Ένα αρχείο BMP περιέχει μια επικεφαλίδα που περιέχει πληροφορίες για το αρχείο (file header), στην συνέχεια πληροφορίες για την αποθηκευμένη εικόνα (bitmap-info header), μετά μια παλέτα με τα χρώματα που χρησιμοποιούνται στην εικόνα μας (η παλέτα μπορεί να μην υπάρχει) και τέλος βρίσκεται αποθηκευμένη η εικόνα.

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ ΣΤΟ ΓΡΑΦΙΚΟ ΕΝΔΙΑΜΕΣΟ ΜΕ ΤΟΝ ΧΡΗΣΤΗ (GUI) MATLAB 6.5

Εισαγωγή

Ένα γραφικό ενδιάμεσο με τον χρήστη (GUI) είναι μια εικονογραφική διεπαφή σε ένα πρόγραμμα. Μία καλή γραφική διεπαφή με το χρήστη μπορεί να καταστήσει τα προγράμματα ευκολότερα να χρησιμοποιηθούν παρέχοντας τους μια συνεπή εμφάνιση και διαισθητικούς ελέγχους όπως, τα κουμπιά, τα παράθυρα καταλόγων, οι ολισθαίνοντες ρυθμιστές, επιλογές κ.τ.λ. Το GUI πρέπει να συμπεριφερθεί κατά τρόπο κατανοητό και προβλέψιμο, έτσι ώστε ο χρήστης να ξέρει τι να περιμένει ή πότε εκτελεί μια δράση. Παραδείγματος χάριν, όταν κανουμε κλικ με το ποντίκι πάνω σε ένα κουμπί, τότε το GUI πρέπει να αρχίσει τη δράση που περιγράφεται στην ετικέτα του κουμπού. Στα επόμενα περιγράφονται τα βασικά στοιχεία των GUIs του MATLAB. Χρησιμοποιώντας τα πιο κάτω έχουμε όλα εκείνα τα απαραίτητα συστατικά που απαιτούνται για να δημιουργήσουν λειτουργικό γραφικό περιβάλλον (GUI) για τα προγράμματά σας.

1.1 Πώς λειτουργεί ένα γραφικό ενδιάμεσο με τον χρήστη

Ένα γραφικό ενδιάμεσο με τον χρήστη παρέχει στο χρήστη ένα γνωστό περιβάλλον στο οποίο μπορεί να εργαστεί. Αυτό το περιβάλλον περιέχει τα κουμπιά, κουμπιά τραβερσών (toggle), καταλόγους, επιλογές, παράθυρα κειμένου, και ούτω καθ'εξής, τα οποία είναι ήδη εύκολα στην χρήση, έτσι ώστε ο χρήστης να μπορεί να επικεντρωθεί στη χρησιμοποίηση της εφαρμογής παρά στους μηχανισμούς που υπάρχουν για να κάνει δουλεύει τα διάφορα στοιχεία. Εντούτοις, τα GUIs είναι σκληρότερα για τον προγραμματιστή επειδή ένα βασικό πρόγραμμα GUI πρέπει να είναι προετοιμασμένο για διάφορα κλικ του ποντικιού (ή ενδεχομένως όταν εισάγουμε κάτι από το πληκτρολόγιο) για οποιοδήποτε στοιχείο GUI σε οποιαδήποτε στιγμή. Τέτοιες εισαγωγές είναι γνωστές ως γεγονότα, και ένα πρόγραμμα που αποκρίνεται στα γεγονότα λέγεται ότι είναι *καθοδηγούμενο από τα γεγονότα (event-driven)*. Τα τρία κύρια στοιχεία που απαιτούνται για να δημιουργήσουν ένα MATLAB γραφικό ενδιάμεσο με τον χρήστη είναι:

1. Συστατικά: Κάθε στοιχείο σε ένα MATLAB GUI (τα κουμπιά, τα παράθυρα, κ.λπ.) είναι ένα γραφικό συστατικό. Οι τύποι συστατικών περιλαμβάνουν τους γραφικούς ελέγχους (τα κουμπιά, τα παράθυρα, οι κατάλογοι, οι ολισθαίνοντες ρυθμιστές κ.λπ.), τα στατικά στοιχεία (πλαίσια και σειρές κειμένων), οι επιλογείς, και οι άξονες. Οι γραφικοί έλεγχοι και τα στατικά στοιχεία δημιουργούνται από τη λειτουργία `uicontrol`, και οι επιλογές δημιουργούνται από τις λειτουργίες `uimenu` και `uicontextmenu`. Οι άξονες, που χρησιμοποιούνται στα γραφικά στοιχεία επίδειξης, δημιουργούνται από τους άξονες λειτουργίας.

2. Αριθμοί (figure). Τα συστατικά ενός GUI πρέπει να τακτοποιηθούν μέσα σε έναν αριθμό, ο οποίος είναι ένα παράθυρο στην οθόνη του υπολογιστή. Στο παρελθόν, οι μορφές (figure) δημιουργούνταν αυτόματα κάθε φορά που σχεδιάζαμε τα στοιχεία. Εντούτοις, οι κενοί αριθμοί μπορούν να δημιουργηθούν με τη λειτουργία `figure` και μπορεί να χρησιμοποιηθεί για να κρατήσει οποιοδήποτε συνδυασμό συστατικών.

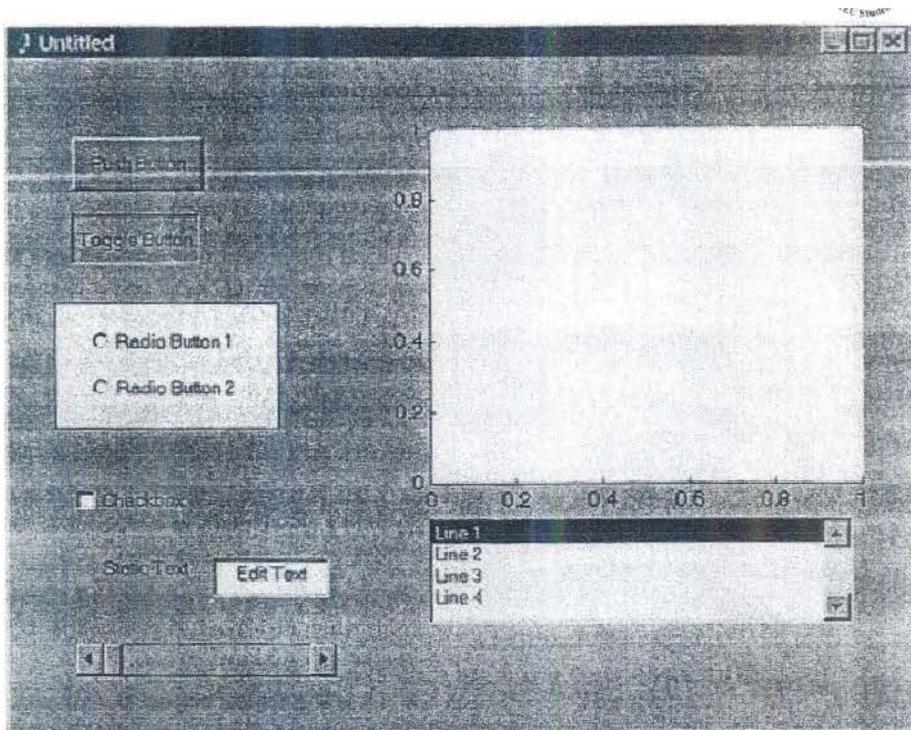
3. Επανακλήσεις (callbacks). Τέλος, πρέπει να υπάρξει κάποιος τρόπος να εκτελεστεί μια δράση εάν ένας χρήστης κάνει κλικ με το ποντίκι του σε ένα κουμπί ή δακτυλογραφεί τις πληροφορίες για ένα πληκτρολόγιο. Ένα κλικ του ποντικιού ή ένα πάτημα κουμπού είναι ένα γεγονός, και το πρόγραμμα MATLAB πρέπει να αποκρίθει σε κάθε γεγονός εάν το

πρόγραμμα πρόκειται να εκτελέσει τη λειτουργία του. Παραδείγματος χάριν, εάν ένας χρήστης κάνει κλικ σε ένα κουμπί, εκείνο το γεγονός πρέπει να προκαλέσει τον κώδικα MATLAB που εφαρμόζει τη λειτουργία του κουμπιού που εκτελείται. Ο κώδικας που εκτελείται σε απάντηση σε ένα γεγονός είναι γνωστός ως call back (επανάκληση). Πρέπει να υπάρξει μια επανάκληση για να εφαρμόσει τη λειτουργία κάθε γραφικού συστατικού στο GUI. Τα βασικά στοιχεία GUI συνοψίζονται στον πίνακα 1.1, και τα στοιχεία δειγμάτων παρουσιάζονται στο σχήμα 1.1. Θα μελετάμε τα παραδείγματα αυτών των στοιχείων για να χτίσουμε έπειτα την εργασία με το γραφικό περιβάλλον(GUI).

Table 10.1 Some Basic GUI Components

Element	Created By	Description
Graphical Controls		
Pushbutton	uicontrol	A graphical component that implements a pushbutton. It triggers a callback when clicked with a mouse.
Toggle button	uicontrol	A graphical component that implements a toggle button. A toggle button is either "on" or "off," and it changes state each time that it is clicked. Each mouse button click also triggers a callback.
Radio button	uicontrol	A radio button is a type of toggle button that appears as a small circle with a dot in the middle when it is "on." Groups of radio buttons are used to implement mutually exclusive choices. Each mouse click on a radio button triggers a callback.
Check box	uicontrol	A check box is a type of toggle button that appears as a small square with a check mark in it when it is "on." Each mouse click on a check box triggers a callback.
Edit box	uicontrol	An edit box displays a text string and allows the user to modify the information displayed. A callback is triggered when the user presses the Enter key.
List box	uicontrol	A list box is a graphical control that displays a series of text strings. A user can select one of the text strings by single- or double-clicking on it. A callback is triggered when the user selects a string.
Popup menu	uicontrol	A popup menu is a graphical control that displays a series of text strings in response to a mouse click. When the popup menu is not clicked on, only the currently selected string is visible.
Slider	uicontrol	A slider is a graphical control to adjust a value in a smooth, continuous fashion by dragging the control with a mouse. Each slider change triggers a callback.
Static Elements		
Frame	uicontrol	Creates a frame, which is a rectangular box within a figure. Frames are used to group sets of controls together. Frames never trigger callbacks.
Text field	uicontrol	Creates a label, which is a text string located at a point on the figure. Text fields never trigger callbacks.
Menus and Axes		
Menu items	uimenu	Creates a menu item. Menu items trigger a callback when a mouse button is released over them.
Context menus	uicontextmenu	Creates a context menu, which is a menu that appears over a graphical object when a user right-clicks the mouse on that object.
Axes	axes	Creates a new set of axes to display data on. Axes never trigger callbacks.

Πίνακας 1.1. Κάποια βασικά 'συστατικά' του GUI.



Σχήμα 1.1 ένα παράθυρο αριθμού που παρουσιάζει παραδείγματα των στοιχείων GUI του MATLAB. Από την κορυφή προς το τέλος και αριστερά προς τα δεξιά τα στοιχεία είναι: (1) ένα κουμπί (2) ένα κουμπί τραβερσών (toggle), (3) δύο ραδιο-κουμπιά που περιβάλλονται από ένα πλαίσιο, (4) ένα παράθυρο ελέγχου, (5) ένας τομέας κειμένων και το παράθυρο, (6) ένας ολισθαίνων ρυθμιστής (κείμενο με μπάρα), (7) ένα σύνολο αξόνων και (8) ένα παράθυρο καταλόγων.

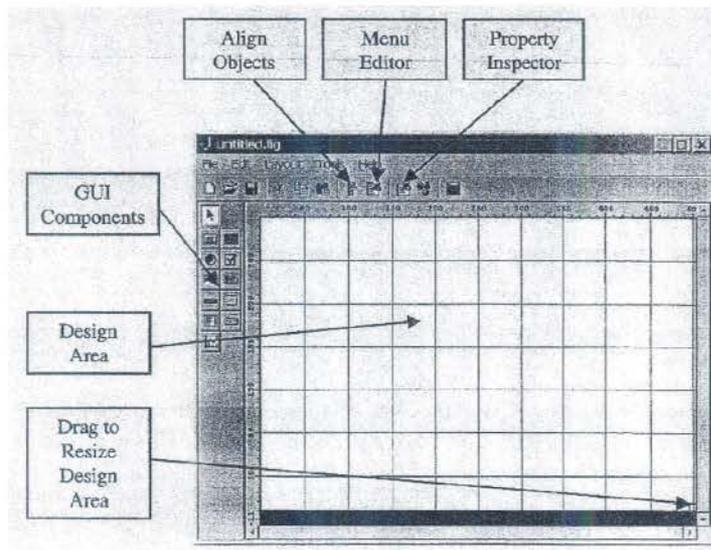
1.2 Δημιουργία και επίδειξη ενός γραφικού ενδιαμέσου με τον χρήστη

Το MATLAB GUI δημιουργείται χρησιμοποιώντας ένα εργαλείο αποκαλούμενο οδηγό (guide), το περιβάλλον ανάπτυξης GUI. Αυτό το εργαλείο επιτρέπει σε έναν προγραμματιστή στο σχεδιάγραμμα του GUI, να επιλέγει και να ευθυγραμμίζει τα συστατικά του GUI που τοποθετούνται σε αυτό. Μόλις είναι τα συστατικά σε ισχύ, ο προγραμματιστής μπορεί να εκδώσει τις ιδιότητές τους: όνομα, χρώμα, μέγεθος, πηγή, το κείμενο που να εμφανίζεται, και ούτω καθ'εξής. Όταν ο οδηγός σώζει το GUI, δημιουργεί το πρόγραμμα εργασίας συμπεριλαμβανομένων των λειτουργιών σκελετών που ο προγραμματιστής μπορεί να τροποποιήσει για να εφαρμόσει τη συμπεριφορά του GUI.

Όταν ο οδηγός εκτελείται, δημιουργεί το συντάκτη (Layout Editor) σχεδιαγράμματος, που παρουσιάζεται στο σχήμα 1.2. Η μεγάλη άσπρη περιοχή με τις γραμμές πλέγματος είναι η περιοχή σχεδιαγράμματος, όπου ένας προγραμματιστής μπορεί να σχεδιάσει το GUI.

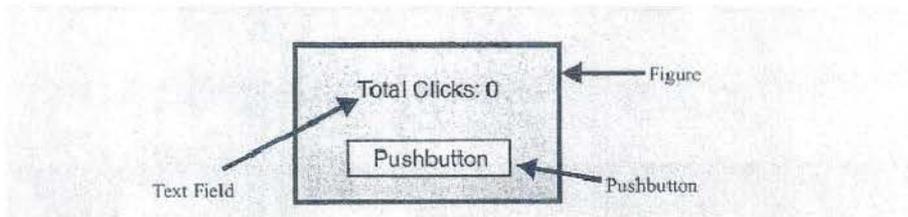
Το παράθυρο συντακτών σχεδιαγράμματος έχει μια παλέτα των συστατικών GUI κατά μήκος της αριστερής πλευράς της περιοχής σχεδιαγράμματος. Ένας χρήστης μπορεί να δημιουργήσει οποιοδήποτε αριθμό συστατικών GUI επιλέγοντας πρώτα το επιθυμητό συστατικό κάνοντας ένα κλικ επάνω του με το ποντίκι, και έπειτα να το σύρει (μετακινήσει) στην περιοχή σχεδιαγράμματος. Στη κορυφή του παραθύρου έχει μια μπάρα εργαλείων με μια σειρά χρήσιμων εργαλείων που επιτρέπουν στο χρήστη να διανείμουν και να ευθυγραμμίσουν τα συστατικά του GUI, ακόμα μπορούν να τροποποιήσουν τις ιδιότητες των συστατικών του GUI, να προσθέσουν διάφορες επιλογές στο GUI, και τα λοιπά. Τα βασικά βήματα που απαιτούνται για να δημιουργήσουν ένα MATLAB GUI είναι:

1. Λήψη απόφασης για το ποια στοιχεία απαιτούνται για το GUI και ποια θα είναι η λειτουργία του κάθε στοιχείου. Κατ' αυτόν τον τρόπο δημιουργούμε ένα πρώτο σχεδιάγραμμα των συστατικών με το χέρι σε ένα κομμάτι χαρτιού.
2. Χρησιμοποιούμε ένα εργαλείο MATLAB που καλείται οδηγός (περιβάλλον ανάπτυξης GUI)
3. Χρησιμοποιούμε ένα εργαλείο MATLAB αποκαλούμενο 'property inspector' για να δώσουμε σε κάθε συστατικό ένα όνομα (μια "ετικέτα") και για να θέσουμε τα χαρακτηριστικά κάθε συστατικού, όπως το χρώμα του, το κείμενο που επιδεικνύει, και τα λοιπά.
4. Σώζουμε τον αριθμό (figure) σε ένα αρχείο. Όταν ο αριθμός σώζεται, δύο αρχεία θα δημιουργηθούν στο δίσκο με το ίδιο όνομα αλλά με διαφορετικές επεκτάσεις. Το αρχείο .fig περιέχει το πραγματικό GUI που έχει δημιουργηθεί, και το .m αρχείο περιέχει τον κώδικα για να φορτώσει τους αριθμούς (figure) κλήσης και τους σκελετούς για κάθε στοιχείο GUI.
5. Γράφουμε τον κώδικα για να εφαρμόσουμε τη συμπεριφορά που συνδέεται με κάθε λειτουργία επανάκλησης



Σχήμα 1.2 το παράθυρο εργαλείων

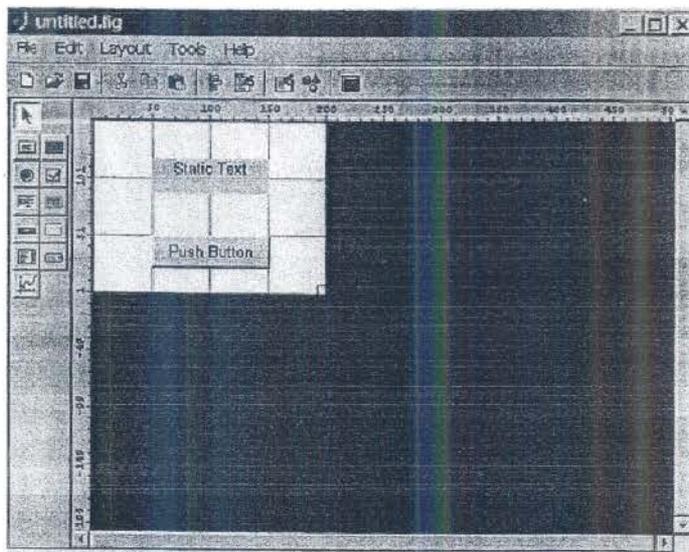
Σαν παράδειγμα αυτών των βημάτων, εξετάστε ένα απλό GUI που περιέχει ένα ενιαίο κουμπί και μια ενιαία σειρά κειμένων. Κάθε φορά που πατάμε αυτο το κουμπί, η σειρά κειμένων θα ενημερωθεί για να παρουσιάσει συνολικό αριθμό 'χτυπημάτων' δεδομένου ότι το GUI άρχισε.



Σχήμα 1.3 Ένα αφαιρετικό σχεδιάγραμμα για ένα GUI που περιέχει ένα ενιαίο κουμπί και έναν ενιαίο τομέα ετικετών.

Βήμα 1: Το σχέδιο αυτής της GM είναι πολύ απλό. Περιέχει ένα ενιαίο κουμπι και έναν ενιαίο τομέα κειμένων. Η επανάκληση από το κουμπι θα αναγκάσει τον αριθμό που επιδεικνύεται στον τομέα κειμένων να αυξάνεται κατα ένα κάθε φορά που το κουμπί πρίζεται. Ένα τραχύ σκίτσο του GUI παρουσιάζεται στο σχήμα 1.3.

Βήμα 2: Στο σχεδιάγραμμα τα συστατικά στο GUI, τρέχουν τον οδηγό λειτουργίας του MATLAB. Όταν ο οδηγός εκτελείται, δημιουργεί το παράθυρο που παρουσιάζεται στο σχήμα 1.2.



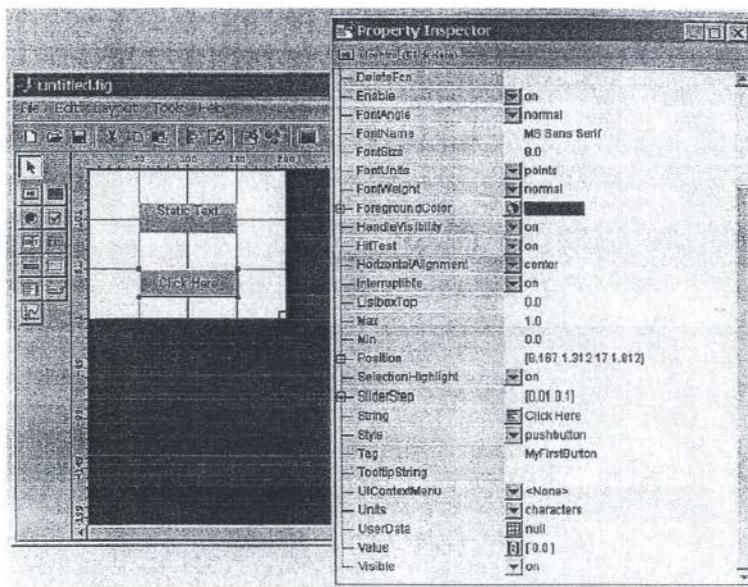
Σχήμα 1.4 Το ολοκληρωμένο σχεδιάγραμμα GUI μέσα στο παράθυρο οδηγών

Κατ' αρχάς, πρέπει να θέσουμε το μέγεθος της περιοχής σχεδιαγράμματος, η οποία θα γίνει το μέγεθος στο τελικό GUI. Κάνουμε αυτό με το σύρσιμο του μικρού τετραγώνου στη χαμηλότερη δεξιά γωνία της περιοχής του σχεδιαγράμματος έως ότου έχει το επιθυμητό μέγεθος και τη μορφή. Κατόπιν, κάντε κλικ στο κουμπί "Pushbutton" στον κατάλογο του GUI και τα συστατικά, δημιουργούν τη μορφή του κουμπιού στην περιοχή σχεδιαγράμματος. Τέλος, κάντε κλικ στο κουμπί "κειμένων (text)" στα τμήματα καταλόγων GUI, και δημιουργήστε τη μορφή του τομέα κειμένων στην περιοχή σχεδιαγράμματος. Η προκύπτουσα μορφή μετά από αυτά τα βήματα παρουσιάζεται στο σχήμα 1.4. Θα μπορούσαμε τώρα να

ρυθμίσουμε την ευθυγράμμιση αυτών των δύο στοιχείων χρησιμοποιώντας το εργαλείο ευθυγράμμισης, εάν το επιθυμούμε.

Βήμα 3: Για να θέσετε τις ιδιότητες του κουμπιού, κάντε κλικ με το ποντίκι στο κουμπί στην περιοχή σχεδιαγράμματος και επιλέξτε έπειτα τον Property Inspector από τη μπάρα εργαλείων.

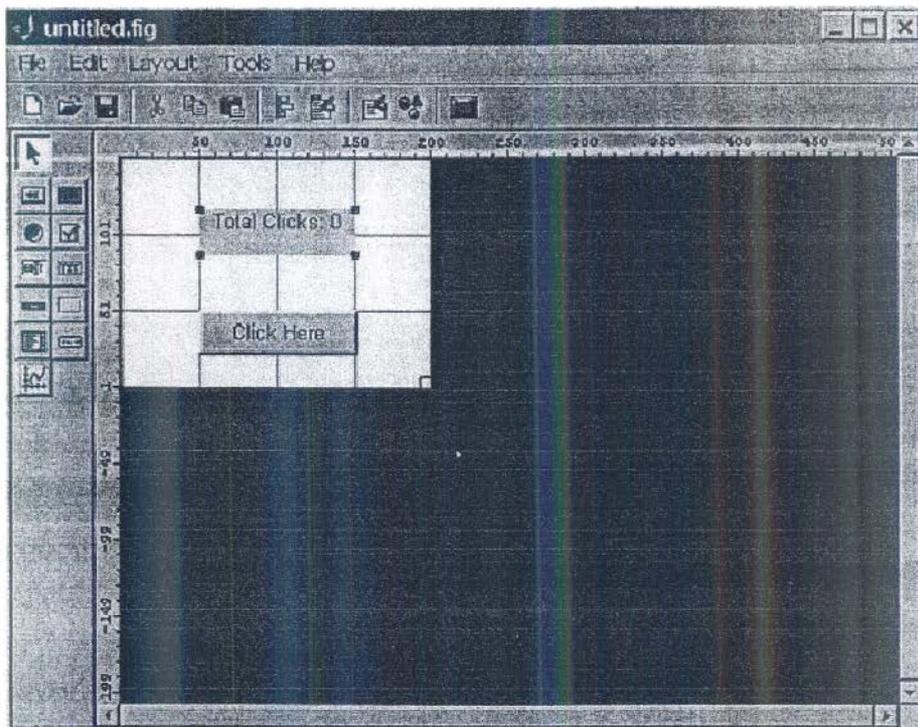
Εναλλακτικά, κάνουμε δεξί κλικ στο κουμπί και επιλέγουμε τον Property Inspector από τις υπερεμφανιζόμενες επιλογές. Το παράθυρο Property Inspector που παρουσιάζεται στο σχήμα 1.5 θα εμφανιστεί. Σημειώστε ότι αυτό το παράθυρο απαριθμεί κάθε ιδιότητα διαθέσιμη για το κουμπί και σας επιτρέπει να θέσετε κάθε τιμή χρησιμοποιώντας μια διαπαφή GUI. Ο επιθεωρητής ιδιότητας εκτελεί την ίδια λειτουργία όπως με τις get και set λειτουργίες, αλλά σε μια καταλληλότερη μορφή.



Σχήμα 1.5 Ο Property Inspector που παρουσιάζει τις ιδιότητες του κουμπιού. Σημειώστε ότι η σειρά τίθεται ως στόχος "click here", και η ετικέτα τίθεται "MyFirstButton".

Για το κουμπί, μπορούμε να θέσουμε πολλές ιδιότητες όπως το χρώμα, μέγεθος, πηγή, ευθυγράμμιση κειμένων, και τα λοιπά. Εντούτοις, *πρέπει* να θέσουμε δύο ιδιότητες: την ιδιότητα σειράς, που περιέχουν το κείμενο που επιδεικνύεται, και την ιδιότητα ετικετών, η οποία είναι το όνομα του κουμπιού. Σε αυτήν την περίπτωση, η ιδιότητα της σειράς θα τεθεί ως στόχος "click here", και η ιδιότητα ετικετών θα τεθεί MyFirstButton. Για τον τομέα κειμένων, *πρέπει* να θέσουμε δύο ιδιότητες: την ιδιότητα σειράς, που περιέχει το κείμενο που επιδεικνύεται, και την ιδιότητα ετικετών, η οποία είναι το όνομα του τομέα κειμένων. Αυτό το όνομα θα απαιτηθεί από τη λειτουργία επανάκλησης για να εντοπίσει και να ενημερώσει τον τομέα κειμένων. Σε αυτήν την περίπτωση, η ιδιότητα σειράς θα τεθεί ως στόχος "Total clicks: 0", και η ιδιότητα ετικετών σε MyFirstText". Η περιοχή σχεδιαγράμματος μετά από αυτά τα βήματα παρουσιάζεται στο σχήμα 1.6. Είναι πιθανό να θέσουμε τις επιλογές της μορφής κλικάρωντας πάνω σε ένα κενό σημείο στο layout editor και μετα χρησιμοποιώντας τον Property inspector εξετάζουμε και θέτουμε τα στοιχεία της μορφής. Αν και δεν

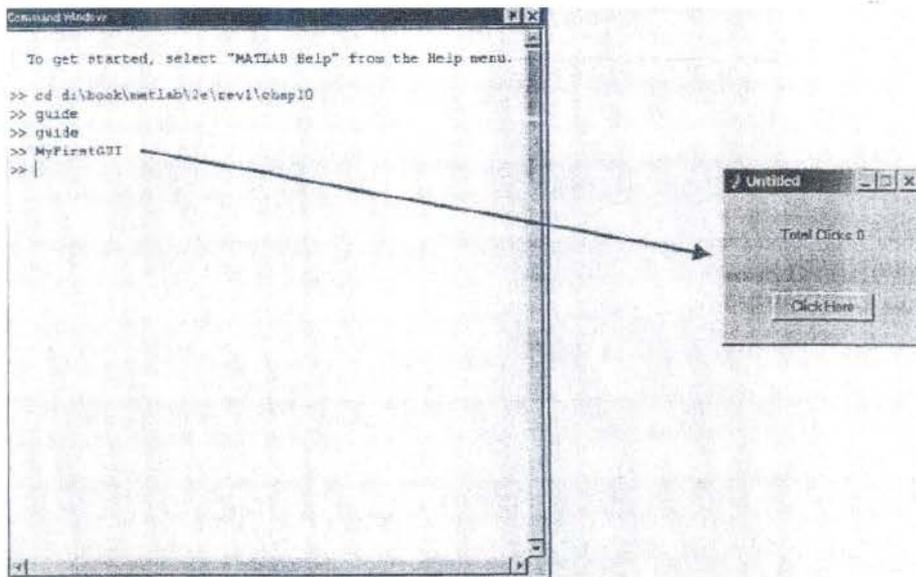
απαιτείται, είναι μια καλή ιδέα να τεθεί η ιδιότητα ονόματος του αριθμού. Η σειρά στην ιδιότητα ονόματος θα επιδειχθεί στο φραγμό τίτλου της κατάληξης GUI όταν εκτελείται.



Σχήμα 1.6 Η περιοχή σχεδίου αφότου έχουν τροποποιηθεί και οι ιδιότητες του κουμπιού και του τομέα κειμένων.

Βήμα 4: Θα σώσουμε τώρα την περιοχή σχεδιαγράμματος με το όνομα MyFirstGUI.

Επιλέξτε το στοιχείο "file/SaveAs" δακτυλογραφήστε το όνομα MyFirstGUI ως όνομα αρχείων, και χτυπήστε "Save". Αυτή η δράση θα δημιουργήσει αυτόματα δύο αρχεία, MyFirstGUI.fig και MyFirstGUI.m. Το αρχείο αριθμού περιέχει το πραγματικό GUI που έχουμε δημιουργήσει. Το M αρχείο περιέχει τον κώδικα που φορτώνει το αρχείο αριθμού και δημιουργεί το GUI, συν μια λειτουργία επανάκλησης σκελετών για κάθε ενεργό συστατικό GUI. Σε αυτό το σημείο, έχουμε μια πλήρη GM, αλλά κάποια που δεν κάνει ακόμα την εργασία που αυτό είχε ως σκοπό να κάνει. Μπορείτε να αρχίσετε αυτήν την GM με τη δακτυλογράφηση MyFirstGUI στο παράθυρο εντολής, όπως φαίνεται στο σχήμα 1.7. Εάν κλικάρουμε το κουμπί σε αυτό το GUI, το ακόλουθο μήνυμα εμφανίζεται στο παράθυρο εντολής: MyFirstButton επανάκληση που δεν εφαρμόζεται ακόμα (MyFirstButton Callback not implemented yet). Μια μερίδα του m αρχείου που δημιουργείται αυτόματα από τον οδηγό παρουσιάζεται στο σχήμα 1.8. Αυτό το αρχείο περιέχει τη λειτουργία MyFirstGUI, συν τις πλαστές υπο-λειτουργίες εφαρμόζοντας τις επανακλήσεις για κάθε ενεργό συστατικό GUI. Εάν η λειτουργία MyFirstGUI καλείται χωρίς ορίσματα, κατόπιν η λειτουργία επιδεικνύει τη GM που περιλαμβάνεται στο αρχείο MyFirstGUI.fig



Σχήμα 1.7 Η δακτυλογράφηση MyFirstGUI το παράθυρο εντολής αρχίζει το GUI.

Εάν η λειτουργία MyFirstGUI καλείται με τα ορίσματα, έπειτα η λειτουργία υποθέτει ότι τα πρώτα επιχειρήματα είναι το όνομα μιας υπο-λειτουργίας, και αυτά καλούν εκείνη την χρησιμοποίηση λειτουργίας feval, περνώντας τα άλλα ορίσματα προς εκείνη την λειτουργία. Κάθε λειτουργία επανάκλησης χειρίζεται τα γεγονότα από ένα ενιαίο συστατικό GUI. Εάν κάνεις κλικ με το ποντίκι (ή κάνεις εισαγωγή από το πληκτρολόγιο για Edit τομείς) εμφανίζεται στο συστατικό GUI, κατόπιν η λειτουργία συστατικής επανάκλησης θα κληθεί αυτόματα από MATLAB. Το όνομα της λειτουργίας επανάκλησης θα είναι η τιμή στην ιδιότητα ετικετών του συστατικού GUI συν των χαρακτήρων "επανάκληση". Κατά συνέπεια, η λειτουργία επανάκλησης για MyFirstButton θα ονομαστεί MyFirstButton_Callback. Τα m αρχεία που δημιουργούνται από τον οδηγό περιέχουν τις επανακλήσεις για κάθε ενεργό συστατικό GUI, αλλά αυτές οι επανακλήσεις επιδεικνύουν απλά ένα μήνυμα λέγοντας ότι η λειτουργία της επανάκλησης δεν έχει εφαρμοστεί ακόμα.

Βήμα 5: Τώρα, πρέπει να εφαρμόσουμε την υπο-λειτουργία επανάκλησης για το κουμπι. Αυτή η λειτουργία θα περιλάβει μια επίμονη μεταβλητή που μπορεί να χρησιμοποιηθεί για να μετρήσει τον αριθμό των κλικαρισμάτων που έχουν εμφανιστεί. Όταν κανουμε ένα κλικαρισμα στο κουμπι, το MATLAB θα καλέσει τη λειτουργία MyFirstGUI με MyFirstButton_callback ως πρώτο επιχειρήμα. Κατόπιν η λειτουργία MyFirstGUI θα καλέσει την υπο-λειτουργία MyFirstButton_callback, όπως φαίνεται στο σχήμα 1.9. Αυτή η λειτουργία πρέπει να αυξήσει την αρίθμηση των πατημάτων κατά ένα, να δημιουργήσει μια νέα σειρά κειμένων που περιέχει την αρίθμηση, και να αποθηκεύσει τη νέα σειρά στην ιδιότητα σειράς του τομέα MyFirstText κειμένων.

```

D:\book\matlab\2e\rev1\chap10\MyFirstGUI.m
File Edit View Text Debug Breakpoints Web Window Help
function varargout = MyFirstGUI(varargin)
% MYFIRSTGUI Application M-file for MyFirstGUI.fig
% FIG = MYFIRSTGUI launch MyFirstGUI GUI.
% MYFIRSTGUI('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.0 22-Jun-2001 21:14:46
if nargin == 0 % LAUNCH GUI
    fig = openfig(mfilename, 'reuse');
    % Use system color scheme for figure:
    set(fig, 'Color', get(0, 'defaultUicontrolBackgroundColor'));
    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);
    if nargin > 0
        varargout{1} = fig;
    end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
    try
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
    catch
        disp(lasterr);
    end
end
% -----
function varargout = MyFirstButton_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.MyFirstButton.
disp('MyFirstButton Callback not implemented yet.')
end

```

Annotations in the image:

- Arrow pointing to line 10: "If called without an argument, open the GUI."
- Arrow pointing to line 26: "If called with an argument, execute the argument as a function."
- Text "Callbacks are subfunctions." with an arrow pointing to line 34.

Σχήμα 1.8 Το m αρχείο forMyFirstGUI, που δημιουργείται αυτόματα από τον οδηγό.

Μια λειτουργία για να εκτελέσει αυτό το βήμα παρουσιάζεται παρακάτω:

```

function varargout = MyFirstButton_Callback(h, eventdata, ...
    handles, varargin)

% Declare and initialize variable to store the count
persistent count
if isempty(count)
    count = 0;
end

```

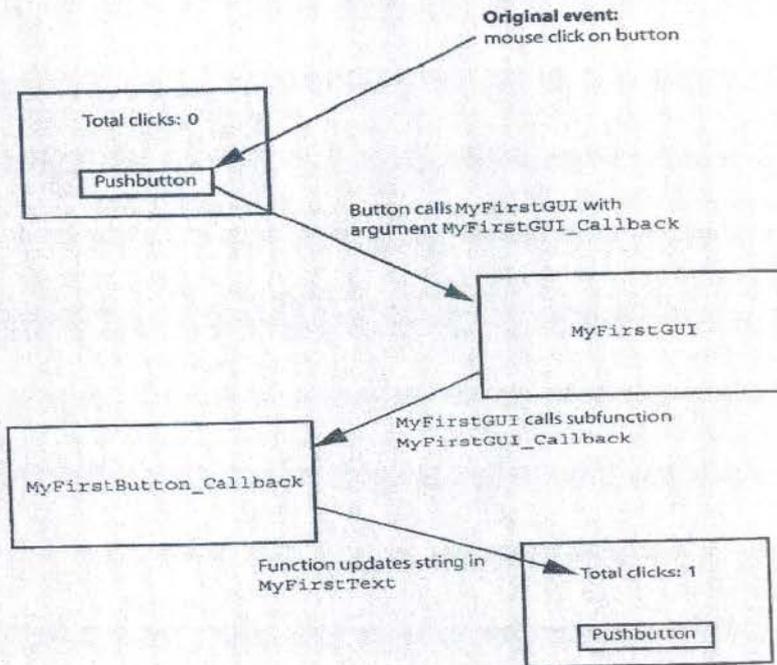
```

% Update count
count = count + 1;

% Create new string
str = sprintf('Total Clicks: %d',count);

% Update the text field
set(handles.MyFirstText,'String',str);

```



Σχήμα 1.9 Χειρισμός γεγονότος στο πρόγραμμα MyFirstGUI. Όταν ένας χρήστης πατά το κουμπί με το ποντίκι, η λειτουργία MyFirstGUI καλείται αυτόματα με το επιχειρήμα MyFirstButton_callback. Η λειτουργία MyFirstGUI καλεί στη συνέχεια την υπο-λειτουργία MyFirstButton_Callback. Οι αυξήσεις αυτής της λειτουργίας μετρούν, και σώζουν έπειτα τη νέα αρίθμηση στον τομέα κειμένων στο GUI.



Σχήμα 1.10 Το προκύπτων πρόγραμμα μετά από τρία πατήματα κουμπιών.

Σημειώστε ότι αυτή η λειτουργία δηλώνει μια επίμονη μεταβλητή αρίθμηση και την μονογράφει σε μηδέν. Κάθε φορά που καλείται αυτή η λειτουργία, μετρά αυξήσεις από 1 και δημιουργεί μια νέα σειρά που περιέχει την αρίθμηση. Κατόπιν, η λειτουργία ενημερώνει τη σειρά που επιδεικνύεται στον τομέα MyFirstText κειμένων. Το προκύπτον πρόγραμμα εκτελείται με τη δακτυλογράφηση του MyFirstGUI στο παράθυρο εντολής. Όταν ο χρήστης πατά το κουμπί, το MATLAB καλεί αυτόματα τη λειτουργία MyFirstGUI με την MYFirstButton_Callback ως πρώτο επιχειρήμα, και η λειτουργία MyFirstGUI καλεί την υπο-λειτουργία MyFirstButton_Callback. Αυτή η μεταβλητή αρίθμηση αυξήσεων λειτουργίας από τη μια και οι αναπροσαρμογές της τιμής από την άλλη επιδεικνύονται στον τομέα κειμένων. Η κατάληξη του GUI μετά από τρία πατήματα κουμπιών παρουσιάζεται στο σχήμα 1.10.

Ορθή πρακτική προγραμματισμού

Ο οδηγός χρήσης για το σχεδιάγραμμα ενός νέου GUI, χρησιμοποιεί τον Property Inspector για να θέσει τις αρχικές ιδιότητες κάθε συστατικού όπως το κείμενο που επιδεικνύεται στο συστατικό, το χρώμα του συστατικού, και το όνομα της λειτουργίας επανάκλησης, αν είναι απαραίτητο. Μετά την δημιουργία ενός GUI με τον οδηγό, εκδώστε με το χέρι την προκύπτουσα λειτουργία για να προσθέσετε τα σχόλια περιγράφοντας το σκοπό και τα συστατικά του και για να εφαρμόσετε τη λειτουργία των επανακλήσεων.

1.2.1 Ένα βλέμμα κάτω από την κουκούλα

Το σχήμα 1.8 παρουσιάζει το m αρχείο που παρήχθη αυτόματα από τον οδηγό για MyFirstGUI. Θα εξετάσουμε τώρα αυτό το m αρχείο για να καταλάβουμε περισσότερο πώς λειτουργεί. Κατ' αρχάς, εξετάστε την ίδια τη δήλωση λειτουργίας. Σημειώστε ότι αυτή η λειτουργία χρησιμοποιεί varargin για να αντιπροσωπεύσει τα ορίσματα εισαγωγής της και varargout για να αντιπροσωπεύσει τα αποτελέσματα παραγωγής της.

Η λειτουργία varargin μπορεί να αντιπροσωπεύσει έναν αυθαίρετο αριθμό επιχειρημάτων εισαγωγής, και η λειτουργία varargout μπορεί να αντιπροσωπεύσει έναν ποικίλο αριθμό επιχειρημάτων παραγωγής. Επομένως, ένας χρήστης μπορεί να καλέσει τη λειτουργία MyFirstGUI με οποιοδήποτε αριθμό επιχειρημάτων.

Κλήση του m αρχείου χωρίς ορίσματα

Εάν ο χρήστης καλεί MyFirstGUI χωρίς ορίσματα, η τιμή που επιστρέφεται από το περιθώριο θα είναι μηδέν. Σε αυτήν την περίπτωση, το πρόγραμμα φορτώνει τη GM από το αρχείο MyFirstGUI.fig αριθμού χρησιμοποιώντας τη λειτουργία openfig. Η μορφή αυτής της λειτουργίας είναι:

```
fig = openfig(mfilename,'reuse');
```

όπου mfilename είναι το όνομα του αρχείου αριθμού που φορτώνει. Το δεύτερο όρισμα στη λειτουργία διευκρινίζει εάν μπορεί να υπάρξει μόνο ένα αντίγραφο του αριθμού που τρέχει σε μία δεδομένη στιγμή, ή εάν τα πολλαπλάσια αντίγραφα μπορούν να οργανωθούν. Εάν το επιχειρήμα είναι "επαναχρησιμοποίηση", κατόπιν μόνο ένα αντίγραφο του αριθμού μπορεί να οργανωθεί. Εάν καλείται το openfig με την επιλογή "επαναχρησιμοποίησης" και ο

διευκρινισμένος αριθμός υπάρχει ήδη, ο προϋπάρχων αριθμός θα παρουσιαστεί στην κορυφή της οθόνης και θα επαναχρησιμοποιηθεί.

Αντίθετα, εάν το επιχείρημα είναι "νέο", τα πολλαπλάσια αντίγραφα του αριθμού μπορούν να οργανωθούν. Εάν καλείται το `openfig` με τη "νέα" επιλογή, ένα νέο αντίγραφο του αριθμού θα δημιουργηθεί κάθε φορά. Εξ ορισμού, ένα GUI που δημιουργείται από τον οδηγό έχει την επιλογή "επαναχρησιμοποίησης", τόσο μόνο ένα αντίγραφο της μορφής που μπορεί να υπάρξει οποιαδήποτε στιγμή. Εάν θέλετε να έχετε τα πολλαπλάσια αντίγραφα του GUI, πρέπει με το χέρι να εκδώσετε αυτήν την κλήση λειτουργίας. Αφότου φορτώνεται η μορφή, η λειτουργία εκτελεί τη δήλωση

```
set(fig,'Color','get(0,'defaultUIControlBackgroundColor')');
```

Αυτή η λειτουργία θέτει το χρώμα υποβάθρου του αριθμού για να ταιριάζει με το χρώμα υποβάθρου προεπιλογής που χρησιμοποιείται από τον υπολογιστή στον οποίο η MATLAB λειτουργεί. Αυτή η λειτουργία κάνει το χρώμα της αντιστοιχίας GUI ίδιο με το χρώμα των εγγενών παραθύρων στον υπολογιστή. Επομένως, ένα GUI μπορεί να γραφτεί σε ένα βασισμένο στα WINDOWS PC και να χρησιμοποιηθεί σε έναν βασισμένο σε Unix υπολογιστή, και αντίστροφα. Θα φανεί φυσικό σε καθένα περιβάλλον. Οι επόμενες δύο δηλώσεις δημιουργούν μια δομή που περιέχει τους χειριστές όλων των αντικειμένων στον τρέχοντες αριθμούς και αυτό που κτίζουν ως στοιχεία εφαρμογής στον αριθμό. Η λειτουργία `guihandles` δημιουργεί μια δομή που περιέχει τους χειριστές σε όλα τα αντικείμενα μέσα στη διευκρινισμένη μορφή.

```
handles = guihandles(fig);  
guidata(fig, handles);
```

Τα ονόματα στοιχείων στη δομή αντιστοιχούν στις ιδιότητες ετικετών κάθε συστατικό GUI, και οι τιμές είναι οι χειριστές κάθε συστατικού. Παραδείγματος χάριν, η δομή χειριστών που επιστρέφεται στο `MyFirstGUI.m` είναι

```
>> handles = guihandles(fig)  
handles =  
          figure: 99.0005
```

```
          MyFirstText: 3.0021  
          MyFirstButton: 100.0007
```

Υπάρχουν τρία συστατικά GUI σε αυτόν τον αριθμό: ο ίδιος ο αριθμός, συν έναν τομέα κειμένων και ένα κουμπι. Το `guidata` λειτουργίας σώζει τη δομή χειριστών ως στοιχεία εφαρμογής στον αριθμό, χρησιμοποιώντας τη λειτουργία `setappdata`.

Το τελικό σύνολο δηλώσεων επιστρέφει εδώ το χειριστή αριθμού

```
if nargin > 0  
    vararginout(1) = fig;  
end
```

Κλήση του m αρχείου με τα ορίσματα

Εάν ο χρήστης καλεί το MyFirstGUI με τα επιχειρήματα, η αξία που επιστρέφεται από το nargin θα είναι μεγαλύτερη από μηδέν. Σε αυτήν την περίπτωση, το πρόγραμμα μεταχειρίζεται το πρώτο καλώντας όρισμα ως όνομα λειτουργίας επανάκλησης και εκτελεί τη χρησιμοποίηση λειτουργίας feval.

Αυτή η λειτουργία εκτελεί τη λειτουργία που ονομάζεται σε varargin1 και περνάει όλα τα υπόλοιπα ορίσματα (varargin2 , varargin3 , κλπ....) στη λειτουργία. Αυτός ο μηχανισμός επιτρέπει στις λειτουργίες επανάκλησης για να είναι υπο-λειτουργίες που δεν μπορούν να κληθούν τυχαία από κάποιο άλλο μέρος του προγράμματος.

1.2.2 Η δομή μιας callback υπο-λειτουργίας κλήσης

Κάθε υπο-λειτουργία επανάκλησης έχει την τυποποιημένη μορφή όπου ComponentTag είναι το όνομα του συστατικού που παράγει την επανάκληση (η σειρά στην ιδιότητα ετικετών του). Τα επιχειρήματα της υπο-λειτουργίας αυτής είναι

- h – ο χειριστής του αριθμού γονέων
- eventdata – πρόσφατη, χρησιμοποιητη σε πίνακα (σε MATLAB Version6)
- handles - η δομή χειριστών περιέχει τις λαβές όλων των συστατικών GUI στην μορφή
- varargin - ένα συμπληρωματικό επιχειρήμα που περνά οποιαδήποτε πρόσθετα καλώντας επιχειρήματα στο callback λειτουργεί. Ένας προγραμματιστής μπορεί να χρησιμοποιήσει αυτό το χαρακτηριστικό γνώρισμα για να παρέχει τις πρόσθετες πληροφορίες στη λειτουργία επανάκλησης εάν είναι απαραίτητο

■ Σημειώστε ότι κάθε λειτουργία επανάκλησης έχει πλήρη πρόσβαση στη δομή χειριστών, έτσι κάθε λειτουργία επανάκλησης μπορεί να τροποποιήσει οποιοδήποτε συστατικό GUI στην μορφή. Εκμεταλλευθήκαμε αυτήν την δομή στη λειτουργία επανάκλησης για το μπουτόν σε MyFirstGUI, όπου η λειτουργία επανάκλησης για το κουμπι τροποποίησε το κείμενο που επιδείχθηκε στον τομέα κειμένων

```
%Update the text field  
set(handles.MyFirstText,'String' str);
```

1.2.3 Προσθήκη των στοιχείων εφαρμογής σε μια μορφή

Είναι δυνατό να αποθηκευτούν οι οριζόμενες από εφαρμογή πληροφορίες που απαιτούνται από ένα πρόγραμμα GUI στη δομή χειριστών αντί της χρησιμοποίησης της ολικής μνήμης για εκείνο το στοιχείο. Το προκύπτον σχέδιο GUI είναι πιά εύρωστο επειδή άλλα προγράμματα MATLAB δεν μπορούν τυχαία να τροποποιήσουν τα ολικά στοιχεία GUI και επειδή τα πολλαπλάσια αντίγραφα του ίδιου GUI δεν μπορούν να παρεμποδιστούν μεταξύ τους. Για να προσθέσουμε τα τοπικά στοιχεία στη δομή χειριστών, πρέπει με το χέρι να τροποποιήσουμε το Mfile αφότου δημιουργείται από τον οδηγό. Ένας προγραμματιστής προσθέτει τα στοιχεία εφαρμογής στη δομή χειριστών μετά από την κλήση στα guihandles και πριν από την κλήση στο guidata.

Παραδείγματος χάριν, να προσθέσει τον αριθμό όταν κλικάρουμε με το ποντικί στη δομή των χειριστών, θα τροποποιούσαμε το πρόγραμμα ως εξής:

```
%Generate a structure of handles to pass to callbacks  
handles=guihandles(fig);
```

```
%Add count to the structure  
handles.count=0;
```

```
%Store the structure  
guidata(fig,handles);
```

Από το στοιχείο εφαρμογής θα περάσουμε τώρα με τη δομή χειριστών σε κάθε λειτουργία επανάκλησης, όπου μπορεί να χρησιμοποιηθεί. Μια έκδοση της λειτουργίας επανάκλησης κουμπιών που χρησιμοποιεί την τιμή αρίθμησης στη δομή δεδομένων χειριστών παρουσιάζεται παρακάτω. Σημειώστε ότι πρέπει να σώσουμε τη δομή χειριστών με μια κλήση στο guidata εάν οποιαδήποτε από τις πληροφορίες σε αυτήν έχει τροποποιηθεί.

```
function varargout =MyFirstButton_Callback(h, eventdata, handles,varargin)  
%Update count  
handles.count=handles.count+1;
```

```
%Save the updated handle structure  
guidata(h, handles)
```

```
%Create new string  
str = sprintf('Total Clicks: %d', handles, count);
```

```
%Update the text field  
set(handles.MyFirstText, 'string', str);
```

Ορθή πρακτική προγραμματισμού

Τα στοιχεία εφαρμογής GUI στους χειριστές κτίζονται, έτσι ώστε να είναι αυτόματα διαθέσιμα σε οποιαδήποτε λειτουργία επανάκλησης.

Εάν τροποποιείτε οποιοδήποτε από τα στοιχεία εφαρμογής GUI στη δομή χειριστών, να είστε βέβαιοι να σώσετε τη δομή με μια κλήση στο guidata πριν βγει η λειτουργία όπου οι τροποποιήσεις εμφανίστηκαν.

1.2.4 Μερικές Χρήσιμες λειτουργίες

Τρεις ειδικές λειτουργίες χρησιμοποιούνται περιστασιακά στο σχέδιο των λειτουργιών επανάκλησης: gcbo, gcbf, και findobj. Αν και αυτές οι λειτουργίες απαιτούνται λιγότερο με MATLAB 6,5 GUI απ'ό,τι με τις προηγούμενες εκδόσεις, είναι ακόμα πολύ χρήσιμες, και ένας προγραμματιστής είναι βέβαιος να τις αντιμετωπίσει.

Το gcbo λειτουργίας (*get callbackobject*) επιστρέφει το χειριστή του αντικείμενου που παρήγαγε την επανάκληση, και η λειτουργία gcbf (*get callbackfigure*) επιστρέφει το χειριστή του αριθμού που περιέχει εκείνο το αντικείμενο. Αυτές οι λειτουργίες μπορούν να χρησιμοποιηθούν από τη λειτουργία επανάκλησης που καθορίζει το αντικείμενο και ο αριθμός που παράγουν την επανάκληση, έτσι ώστε μπορεί να τροποποιήσει τα αντικείμενα σε εκείνη την μορφή.

Η μεταβλητή findobj ψάχνει μεταξύ όλων των "παιδιών" αντικειμένων μέσα από τους "γονείς" αντικείμενα, ψάχνει για ένα το οποίο έχει συγκεκριμένη τιμή μιας συγκεκριμένης ιδιότητας. Επιστρέφει σε εμάς οποιοδήποτε αντικείμενο με τα χαρακτηριστικά που ταιριάζουν. Η πιο κοινή μορφή findobj είναι:

```
Hndl = findobj(parent, 'Property', Value);
```

όπου ο γονέας είναι ο χειριστής ενός αντικειμένου γονέων όπως ένας αριθμός, "η ιδιότητα" είναι η ιδιότητα που εξετάζει, και "η τιμή" είναι η τιμή που ψάχνει.

Παραδείγματος χάριν, υποθέστε ότι ένας προγραμματιστής θα επιθυμούσε να αλλάξει το κείμενο σε ένα κουμπι με την ετικέτα "Button1" όταν εκτελεί μια λειτουργία επανάκλησης. Ο προγραμματιστής θα μπορούσε να βρεί το κουμπι και να αντικαταστήσει το κείμενο με τις ακόλουθες δηλώσεις :

```
Hndl=findobj(gcf,'Tag', 'Button1');  
set (Hndl, 'String', 'New text');
```

1.3 Ιδιότητες αντικειμένου

Κάθε αντικείμενο GUI περιλαμβάνει έναν εκτενή κατάλογο ιδιοτήτων που μπορεί να χρησιμοποιηθεί για να προσαρμόσει το αντικείμενο. Αυτές οι ιδιότητες είναι ελαφρώς διαφορετικές για κάθε τύπο αντικειμένου (αριθμοί, άξονες, uicontrols, κ.λπ.).

Όλες οι ιδιότητες για όλους τους τύπους αντικειμένων είναι τεκμηριωμένες σε απευθείας σύνδεση με την βοήθεια, αλλά μερικές από τις σημαντικότερες ιδιότητες για τα αντικείμενα αριθμού και uicontrol συνοψίζεται στους πίνακες 1.2 και 1.3.

Οι ιδιότητες αντικειμένου μπορούν να τροποποιηθούν χρησιμοποιώντας είτε τον Property Inspector, είτε να παίρνουν και θέτουν τις λειτουργίες. Αν και ο Property Inspector είναι ένας κατάλληλος τρόπος να ρυθμιστούν οι ιδιότητες κατά τη διάρκεια του σχεδίου GUI, πρέπει να χρησιμοποιήσουμε ότι τις παίρνουμε (get) και τις θέτουμε (set), για να ρυθμίσουμε δυναμικά από μέσα από ένα πρόγραμμα, όπως σε μια λειτουργία επανάκλησης.

Property	Description
Color	Specifies the color of the figure. The value is either a predefined color such as 'r', 'g', or 'b', or else a 3-element vector specifying the red, green, and blue components of the color on a 0-1 scale. For example, the color magenta would be specified by [1 0 1].
MenuBar	Specifies whether or not the default set of menus appears on the figure. Possible values are 'figure' to display the default menus, or 'none' to delete them.
Name	A string containing the name that appears in the title bar of a figure.
NumberTitle	Specifies whether or not the figure number appears in the title bar. Possible values are 'on' or 'off'.
Position	Specifies the position of a figure on the screen, in the units specified by the 'units' property. This value accepts a 4-element vector in which the first two elements are the x and y positions of the lower left-hand corner of the figure, and the next two elements are the width and height of the figure.
SelectionType	Specifies the type of selection for the last mouse click on this figure. A single click returns type 'normal'. A double click returns type 'open'. There are additional options; see the MATLAB on-line documentation.
Tag	The 'name' of the figure, which can be used to locate it.
Units	The units used to describe the position of the figure. Possible choices are 'inches', 'centimeters', 'normalized', 'pixels', 'points', 'pixels', or 'screenormal'. The default units are 'pixels'.
Visible	Specifies whether or not this figure is visible. Possible values are 'on' or 'off'.
WindowStyle	Specifies whether this figure is normal or modal (see discussion of Dialog Boxes). Possible values are 'normal' or 'modal'.

Πίνακας 1.2 Σημαντικές ιδιότητες μορφής

1.4 Βασικά χαρακτηριστικά ενός GUI

Αυτό το τμήμα συνοψίζει τα βασικά χαρακτηριστικά συστατικά του κοινού γραφικού ενδιάμεσου τμήματος με τον χρήστη. Περιγράφει πώς να δημιουργεί και να χρησιμοποιήσει κάθε συστατικό, καθώς επίσης και τους τύπους και τα συμβάντα του κάθε συστατικού μέρους που μπορεί να παραγάγει. Τα συστατικά που συζητούνται σε αυτό το μέρος είναι

- Πεδία κειμένων (Text Fields)
- Κουτιά κειμένων (Edit Boxes)
- Πλαίσια (Frames)
- Κουμπιά (Pushbuttons)
- Κουμπιά toggle (Toggle Buttons)
- Κουμπιά ελέγχου (Checkboxes)
- Ράδιο κουμπιά (Radio buttons)
- Αναδυόμενες επιλογές (Popup Menus)
- Κουτιά λιστών (List Boxes)
- Φωτογραφικές διαφάνειες (Slide)

1.4.1 Πεδία κειμένων (Text Fields)

Ένα πεδίο-κειμένου είναι ένα γραφικό αντικείμενο που εμφανίζει μια σειρά (κειμένων). Μπορείτε να επιλέξετε πώς να στοιχίζετε το κείμενο στην περιοχή εμφάνισης με τον καθορισμό της οριζόντιας στοίχισης. Εξ ορισμού, τα πεδία κειμένων έχουν στοίχιση στο κέντρο. Ένα πεδίο κειμένου δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα του είναι η "σύνταξη κειμένων". Ένα πεδίο κειμένου μπορεί να προστεθεί σε ένα GUI με τη χρησιμοποίηση του εργαλείου κειμένων στο σχεδιάγραμμα Σύνταξης. Τα πεδία κειμένων δεν δημιουργούν τις επανακλήσεις, αλλά η τιμή που εμφανίζετε στο πεδίο κειμένου μπορεί να είναι ενημερωμένη σε μια λειτουργία επανάκλησης με την αλλαγή των ιδιοτήτων στους χαρακτήρες των πεδίων κειμένου.

1.4.2 Συντακτικά Κουτιά (Edit Boxes)

Το συντακτικό πλαίσιο είναι ένα γραφικό αντικείμενο που επιτρέπει σε έναν χρήστη να εισαγάγει μια σειρά. Το συντακτικό πλαίσιο παράγει μια επανάκληση όταν ο χρήστης πατάει το πλήκτρο Enter και ύστερα μπορεί να πληκτρολογήσει μια σειρά στο συντακτικό πλαίσιο. Το συντακτικό πλαίσιο δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα είναι η "σύνταξη κειμένων". Το συντακτικό πλαίσιο μπορείτε να το προστεθεί σε ένα GUI με το να χρησιμοποιήσετε το εργαλείο του συντακτικού πλαισίου το οποίο βρίσκεται στο συντάκτη σχεδιαγράμματος.

Στο σχήμα 1.11a παρουσιάζεται ένα απλό GUI το οποίο περιέχει ένα συντακτικό πλαίσιο το οποίο το ονομάζει «συντακτικό πλαίσιο»(Edit Box) και ένα πεδίο κειμένου το οποίο το ονομάζει «πλαίσιο κειμένου»(textbox). Όταν ο τύπος δεδομένων είναι χαρακτήρες στο συντακτικό πλαίσιο τότε καλεί αυτόματα την λειτουργία EditBox_Callback, η οποία παρουσιάζεται στο σχήμα 1.11b. Αυτή η λειτουργία εντοπίζει το συντακτικό πλαίσιο χρησιμοποιώντας τη δομή και τη σειρά που πληκτρολογούνται από το χρήστη. Κατόπιν, αυτό εντοπίζει το πεδίο κειμένων και τους εμφανίζει σε σειρά στο πεδίο κειμένου.

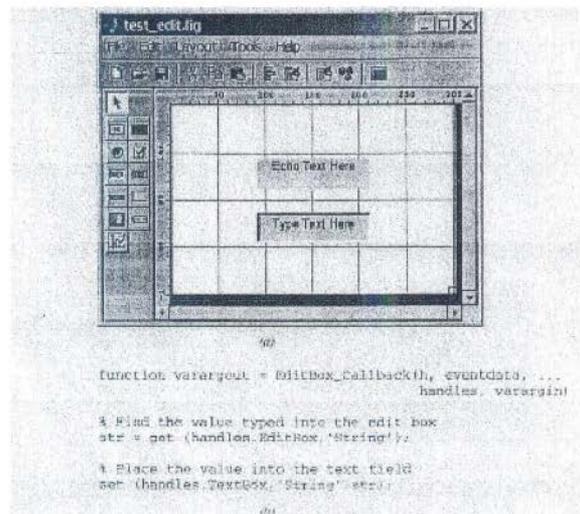
1.4.3 Πλαίσια (Frames)

Ένα πλαίσιο είναι ένα γραφικό αντικείμενο που εμφανίζει ένα ορθογώνιο στο GUI. Μπορείτε να χρησιμοποιήσετε τα πλαίσια(frames) για να «σύρετε» τα κιβώτια γύρω από τις ομάδες

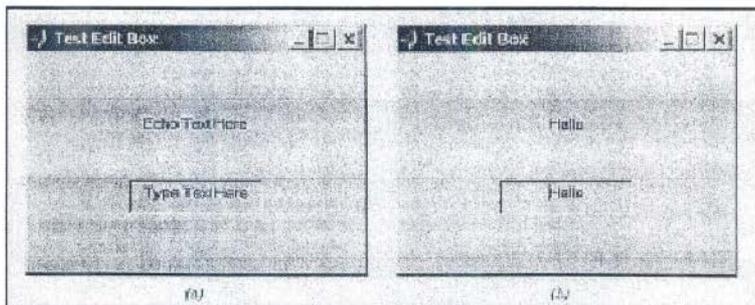
λογικών σχετικά αντικειμένων. Παραδείγματος χάριν, ένα πλαίσιο χρησιμοποιείται για να συγκεντρώσει τα ράδιο κουμπιά σχήμα 1.1. Ένα πλαίσιο δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα του στυλ είναι "πλαίσιο"(Frame). Ένα πλαίσιο προστίθεται σε ένα GUI χρησιμοποιώντας το εργαλείο πλαισίων το οποίο βρίσκεται στο LayoutEditor. Τα πλαίσια δεν παράγουν επανακλήσεις.

1.4.4 Κουμπιά (Pushbuttons)

Τα κουμπιά είναι ένα συστατικό που ένας χρήστης μπορεί να τα πατήσει έτσι ώστε να συμβεί ένα συγκεκριμένο γεγονός (δραστηριότητα). Το κουμπί παράγει μια επανάκληση όταν πατά ο χρήστης το ποντίκι πάνω σε ένα από αυτά. Το κουμπί δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα του στυλ είναι «Κουμπί» (Pushbutton). Το κουμπί μπορεί να προστεθεί σε ένα GUI χρησιμοποιώντας το εργαλείο «Κουμπί» το οποίο βρίσκεται στο Layout Editor. Η λειτουργία του MyFirstGUI στο σχήμα 1.10 αποτελεί επεξήγηση της χρήσης των κουμπιών.



Σχήμα 1.11 (α) Layout ενός απλού GUI με το συντακτικό πλαίσιο και ένας πεδίο κειμένου. (β) Η επανάκληση των λειτουργιών για το GUI.



Σχήμα 1.12 (α) το GUI το οποίο παράγεται από το πρόγραμμα test_edit. (β) Το GUI μετά από το τύπο δεδομένων Hello στο συντακτικό πλαίσιο

1.4.5 Κουμπιά toggle (Toggle Buttons)

Το κουμπιά toggle είναι ένας τύπος κουμπιού το οποίο έχει δύο καταστάσεις : on (όταν είναι πατημένο) και off (όταν δεν είναι πατημένο). Τα κουμπιά toggle έχουν διαφορετικές ιδιότητες μεταξύ δύο καταστάσεων όταν το πατάμε με το ποντίκι το οποίο παράγει μια επανάκληση κάθε φορά. Η ιδιότητα της «τιμής» του κουμπιού toggle είναι συνήθως 1 όταν δεν πατάμε το κουμπιά και 0 όταν πατάμε το κουμπιά. Ένα κουμπιά toggle δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα του στυλ είναι «Κουμπιά toggle» (Button Toggle). Το κουμπιά toggle μπορεί να προστεθεί σε ένα GUI με τη χρησιμοποίηση του εργαλείου κουμπιά toggle το οποίο βρίσκεται στο Layout Editor. στο συντάκτη σχεδιαγράμματος. Στο σχήμα 1.1a παρουσιάζεται απλό GUI περιέχοντας ένα κουμπιά toggle το οποίο ονομάζεται "ToggleButton" και ένα πεδίο κειμένου το οποίο ονομάζεται "TextBox". Όταν ένας χρήστης πατά σε ένα κουμπιά toggle τότε καλείται αυτόματα η λειτουργία ToggleButton_Callback, η οποία παρουσιάζεται στο σχήμα 1.13b. Αυτή η λειτουργία εντοπίζει το κουμπιά toggle χρησιμοποιώντας τη δομή και ανακτά την κατάσταση από την ιδιότητα της «τιμής». Κατόπιν, η λειτουργία εντοπίζει το πεδίο κειμένου και εμφανίζει την κατάσταση του πεδίου κειμένου. Στο σχήμα 1.14 παρουσιάζεται το GUI αμέσως μετά αφότου έχει αρχίσει, και μετά από το πάτημα πάνω στο κουμπιά toggle το οποίο έχει συμβεί για πρώτη φορά.

1.4.6 Κουτιά ελέγχου και ράδιο κουμπιά

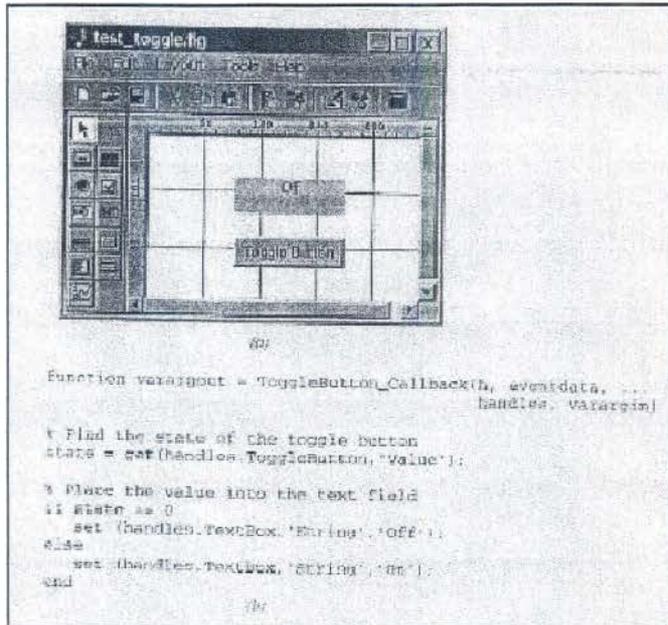
Τα κουτιά ελέγχου και τα ράδιο κουμπιά είναι ουσιαστικά ίδια με τα κουμπιά toggle εκτός από το ότι αυτά έχουν διαφορετικές μορφές. Όπως τα κουμπιά toggle, τα κουτιά ελέγχου και τα ράδιο κουμπιά έχουν δύο καταστάσεις οι οποίες είναι on «ανοικτό» και off «κλειστό». Αυτά εναλλάσσονται μεταξύ δύο καταστάσεων όταν το ποντίκι πατά πάνω σε αυτά, παράγοντας μια επανάκληση κάθε φορά που γίνεται αυτή η διαδικασία. Η ιδιότητα της «τιμής» του κουτιού ελέγχου ή του ράδιο κουμπιού είναι συνήθως 1 όταν δεν πατάμε το κουμπιά και συνήθως 0 όταν το πατάμε. Και τα δύο, τα κουτιά ελέγχου και τα ράδιο κουμπιά διευκρινίζονται στο σχήμα 1.1.

Ένα κουτιά ελέγχου δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα του στυλ είναι "κουτιά ελέγχου" (CheckBoxes), και το ράδιο κουμπιά δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα του στυλ είναι "ράδιο κουμπιά" (Radiobutton). Το κουτιά ελέγχου μπορεί να προστεθεί σε ένα GUI με τη χρησιμοποίηση του εργαλείου κουτιά ελέγχου το οποίο βρίσκεται στο Layout Editor, και το ράδιο κουμπιά μπορεί να προστεθεί σε ένα GUI με τη χρησιμοποίηση του ράδιο κουμπιού το οποίο βρίσκεται στο Layout Editor.

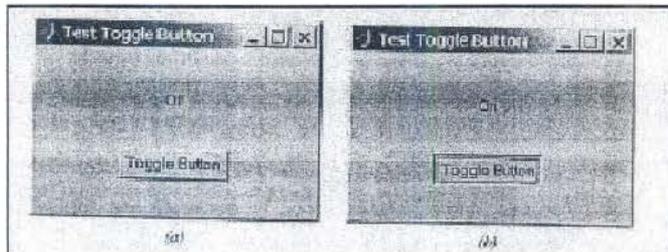
Τα κουμπιά ελέγχου χρησιμοποιούνται παραδοσιακά για την εμφάνιση on /off επιλογών και οι ομάδες ράδιο κουμπιών χρησιμοποιούνται παραδοσιακά για την επιλογή μεταξύ μιας επιλογής από τις πολλές που έχουν τα ραδιο κουμπια. Στο σχήμα 1.1δ παρουσιάζεται παράδειγμα για το πώς δημιουργείται μια ομάδα αμοιβαία αποκλειστικών επιλογών με τα ράδιο κουμπιά.

Το GUI σε αυτό το σχήμα δημιουργεί τρία ράδιο κουμπιά, τα οποία ονομάζονται "επιλογή 1," "επιλογή 2," και "επιλογή 3." Κάθε ράδιο κουμπιά χρησιμοποιεί την ίδια λειτουργία επανάκλησης, αλλά το καθένα με μια ξεχωριστή παράμετρο. Οι αντίστοιχες λειτουργίες επανάκλησης παρουσιάζονται στο σχήμα 1.1 (5b). Όταν ο χρήστης κάνει κλικ σε ένα ράδιο κουμπιά, εκτελείται η αντίστοιχη λειτουργία επανάκλησης. Αυτή η λειτουργία τίθεται στο παράθυρο κειμένου για να εμφανίσει την τρέχουσα επιλογή, και επιλέγει το συγκεκριμένο ράδιο κουμπιά, και δεν επιλέγει τα υπόλοιπα ράδιο κουμπιά.

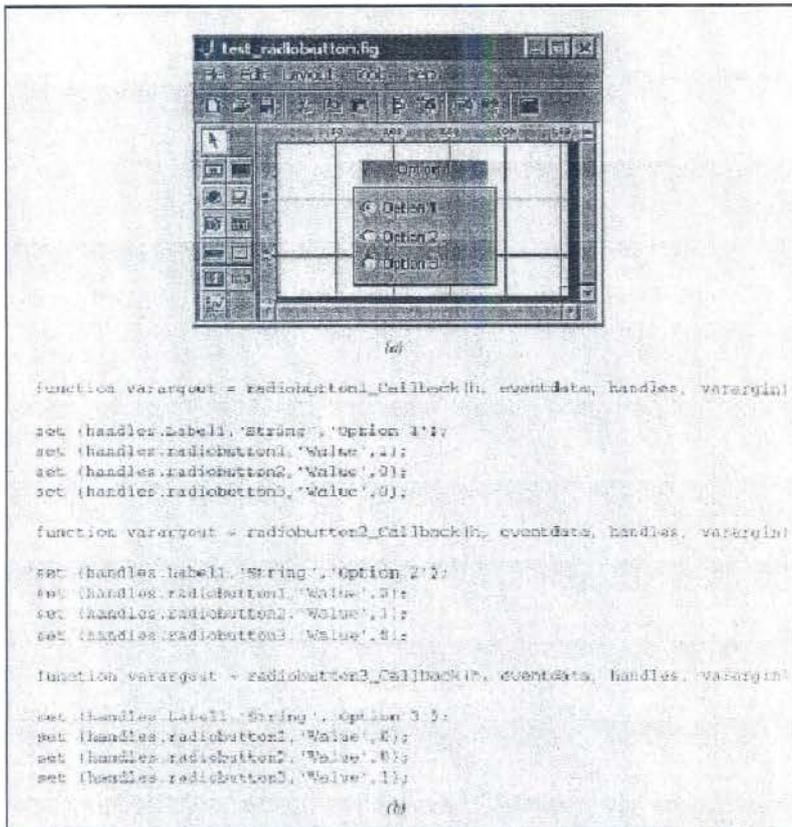
Σημειώστε ότι το GUI χρησιμοποιεί ένα πλαίσιο για να ομαδοποιήσει τα ράδιο κουμπιά μαζί, καθιστώντας το προφανές ότι είναι ένα σύνολο. Στο σχήμα 1.16 παρουσιάζεται το GUI εφόσον έχει επιλεγεί η «επιλογή 2».



Σχήμα 1.13 (α) Layout ενός απλού GUI με ένα κουμπί toggle και ένα πεδίο κειμένου. (β) Η επανάκληση της λειτουργίας για το GUI.



Σχήμα 1.14 (α) το GUI το οποίο παράγεται από το πρόγραμμα test_togglebutton όταν το κουμπί toggle δεν είναι πατημένο. (β) Το GUI όταν είναι πατημένο το κουμπί toggle.



Σχήμα 1.15 (α) Layout ενός απλού GUI με τρία ράδιο κουμπιά σε ένα πλαίσιο, συν ένα πεδίο κειμένου για να εμφανίζει την τρέχουσα επιλογή. (β) Και οι λειτουργίες επανάκλησης για αυτό το GUI.



Σχήμα 1.16 το GUI το οποίο παράγεται από το πρόγραμμα test_radiobutton

1.4.7 Αναδυόμενες επιλογές (Popurp Menus)

Οι αναδυόμενες επιλογές είναι γραφικά αντικείμενα τα οποία επιτρέπουν σε έναν χρήστη να επιλέξει μια επιλογή από μια λίστα επιλογών. Η λίστα επιλογών είναι μια λίστα από την οποία ο χρήστης μπορεί να επιλέξει από ένα σύνολο επιλογών μια συγκεκριμένη επιλογή και η ιδιότητα της «τιμής» προσδιορίζει την επιλογή την οποία έκανε ο χρήστης. Οι αναδυόμενες

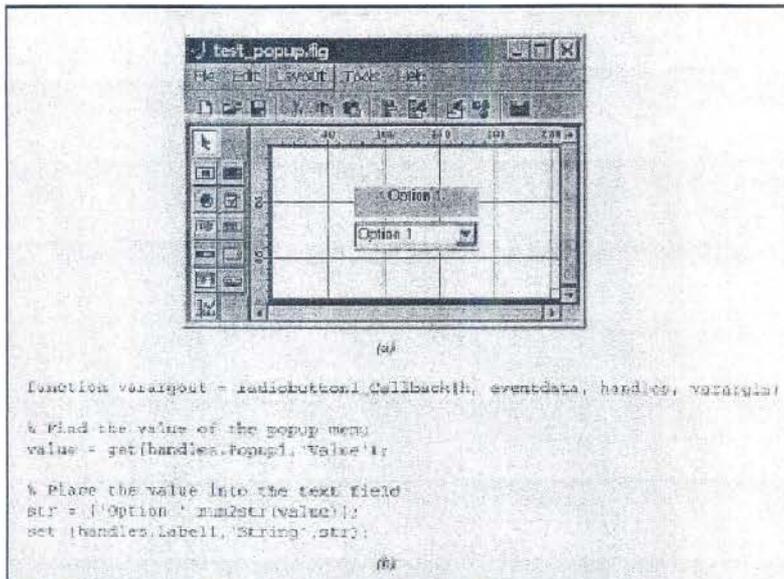
επιλογές μπορούν να προστεθούν σε ένα GUI με τη χρησιμοποίηση του εργαλείου των αναδυόμενων επιλογών το οποίο βρίσκεται στο Layout Editor.

Στο σχήμα 1.17a παρουσιάζεται ένα παράδειγμα αναδυόμενων επιλογών. Το GUI σε αυτό το σχήμα δημιουργεί έναν μενού αναδυόμενων επιλογών με πέντε επιλογές, με τα εξής ονόματα "επιλογή 1," "επιλογή 2," και ούτω καθ' εξής. Η αντιστοιχία των λειτουργιών των επανακλήσεων παρουσιάζονται στο σχήμα 1.17b. 1.17b. Η επανάκληση της λειτουργίας ανακτά την επιλεγμένη επιλογή με τον έλεγχο της «τιμής» της παραμέτρου των αναδυόμενων επιλογών, η οποία δημιουργεί και τα εμφανίζει στην σειρά που περιέχει εκείνη η τιμή στο πεδίο του κειμένου. Το σχήμα 1.18 παρουσιάζει αυτήν την GM αφότου έχει επιλεγθεί η επιλογή «επιλογή 4».

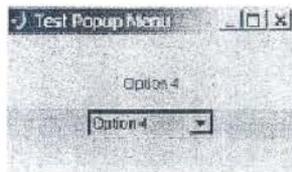
1.4.8 Κουτιά λιστών (List Boxes)

Τα κουτιά λιστών είναι γραφικά αντικείμενα που επιδεικνύουν πολλές γραμμές κειμένου και επιτρέπουν σε έναν χρήστη να επιλέξει μια ή περισσότερες από αυτές εκείνες τις γραμμές. Εάν υπάρχουν περισσότερες γραμμές κειμένου από την επιφάνεια του κουτιού λίστας, τότε θα δημιουργηθεί μια μπάρα για να επιτρέψει στο χρήστη να δει και το «κρυφό» κείμενο το οποίο βρίσκεται στο κουτί λίστας(list box). Οι γραμμές κειμένου μεταξύ των οποίων ο χρήστης μπορεί να τις επιλέξει από μια σειρά γραμμών με βάση την ιδιότητα της «τιμής» η οποία προσδιορίζει την σειρά των γραμμών που επιλέχθηκαν. Ένα κουτί λίστας δημιουργείται με τη δημιουργία ενός widget του οποίου η ιδιότητα του στιλ είναι "πλαίσιο"(κουτί λίστας) (listbox). Ένα κουτί λίστας μπορεί να προστεθεί σε ένα GUI με τη χρησιμοποίηση του εργαλείου listbox το οποίο βρίσκεται στο Layout Editor.

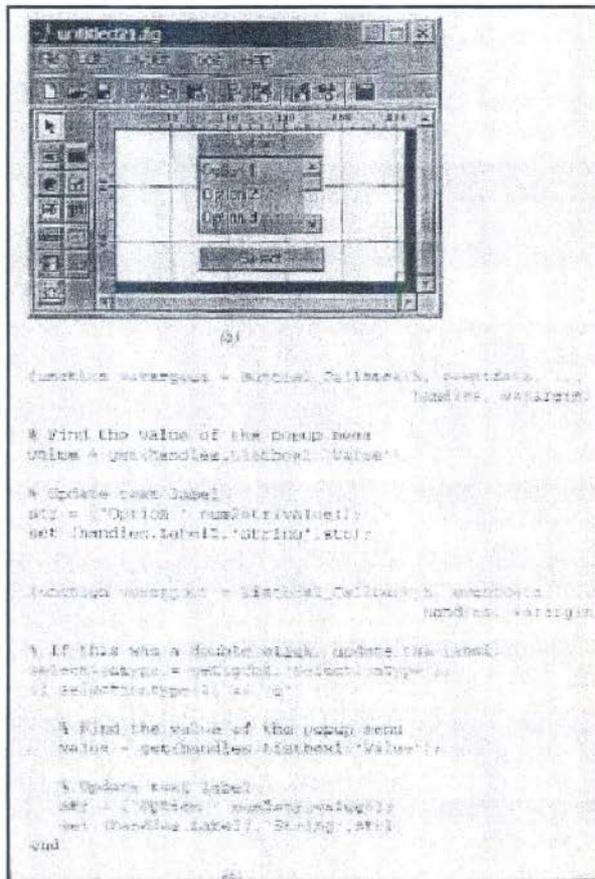
Τα κουτιά λιστών μπορούν να χρησιμοποιηθούν για να επιλέξουν ένα στοιχείο από ένα σύνολο πιθανών επιλογών. Σε μια συνηθισμένη χρήση του περιβάλλοντος GUI ένα ποντίκι κάνει κλικ σε ένα στοιχείο μιας λίστας επιλογής στοιχείων και επιλέγει ένα στοιχείο αλλά το επιλεγμένο στοιχείο δεν προκαλεί κάποιο γεγονός του οποίου το αποτέλεσμα εμφανίζεται. Αντ' αυτού, το γεγονός περιμένει κάποια εξωτερική δραστηριότητα, όπως το πάτημα ενός κουμπιού. Εντούτοις, ένα ποντίκι κάνει διπλό κλικ ώστε να συμβεί ένα γεγονός αμέσως. Το μονό κλικ και το διπλό κλικ σαν γεγονότα μπορούν να διακριθούν χρησιμοποιώντας την ιδιότητα του SelectionType με το οποίο μπορούμε να επιλέξουμε ποίο γεγονός θα συμβεί αν γίνει μονό ή διπλό κλικ. Σε ένα μονό κλικ θα τοποθετήσει στην σειρά "κανονικά" στην ιδιότητα SelectionType, και ένα διπλό κλικ θα τοποθετήσει στην σειρά "ανοικτά" στην ιδιότητα του SelectionType.



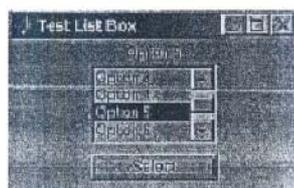
Σχήμα 1.17 (α) Layout ενός απλού GUI με αναδυόμενες επιλογές και έναν πεδίο κειμένου για να εμφανίσει την τρέχουσα επιλογή. (β) Οι λειτουργίες επανάκλησης για το GUI.



Σχήμα 1.18 το GUI το οποίο παράγεται από το πρόγραμμα test_popup.



Σχήμα 1.19 (α) Σχεδιάγραμμα μιας απλής GM με ένα listbox, ένα κουμπι, και έναν τομέα κειμένων. (β) Η επανάκληση λειτουργεί για αυτήν την σημείωση GUI ότι η επιλογή μπορεί να εμφανιστεί είτε με το κλικαρισμα του κουμπιού είτε όταν κανουμε διπλο κλικαρισμα σε ένα στοιχείο στο listbox.



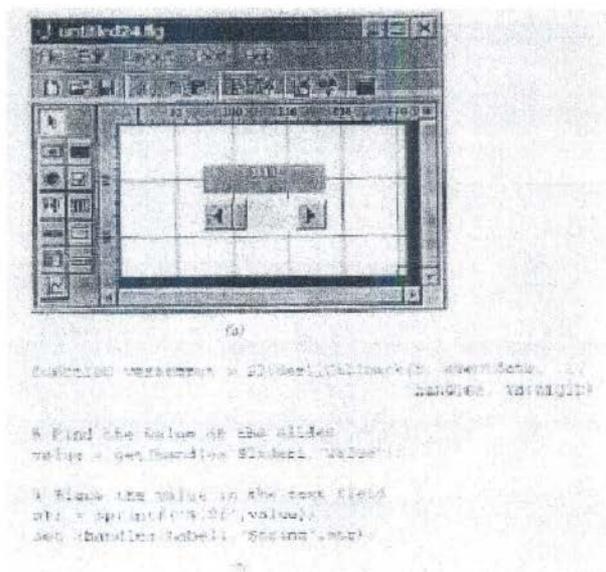
Σχήμα 1.20 το GUI που παράγεται από το πρόγραμμα test_listbox.

Αυτή η λειτουργία θα ελέγξει τον αριθμό παράγοντας την επανάκληση (που χρησιμοποιεί τη λειτουργία `geb1`) για να δει εάν η δράση επιλογής ήταν ένα και μοναδικό κτύπημα του

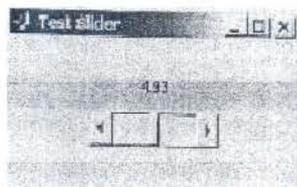
ποντικιού ή έχουμε χτυπήσει δύο φορές. Εάν ήταν ένα κτύπημα του ποντικιού, η λειτουργία δεν κάνει τίποτα. Εάν ήταν δύο κτυπήματα του ποντικιού, κατόπιν η λειτουργία παίρνει την επιλεγμένη αξία από το listbox, και γράφει μια κατάλληλη σειρά στον τομέα κειμένων. Εάν το κουμπι επιλέγεται, κατόπιν η `functionButton1_Callbackwill` εκτελείται. Αυτή η λειτουργία παίρνει την επιλεγμένη αξία από το listbox, και γράφει μια κατάλληλη σειρά στον τομέα κειμένων. Το GUI που παράγεται από το πρόγραμμα `test_listbox` παρουσιάζεται στο σχήμα 1.20. Στο τέλος του κεφαλαίου σε μια άσκηση, θα κληθείτε να τροποποιήσετε αυτό το παράδειγμα για να επιτρέψετε τις πολλαπλάσιες επιλογές στο παράθυρο καταλόγων.

1.4.9 Ολισθαίνοντες ρυθμιστές(Sliders)

Οι ολισθαίνοντες ρυθμιστές είναι γραφικά αντικείμενα που επιτρέπουν σε έναν χρήστη να επιλέξει τις τιμές από μια συνεχή σειρά μεταξύ μιας διευκρινισμένης ελάχιστης αξίας και μιας διευκρινισμένης μέγιστης αξίας με την κίνηση του ποντικιού. Η ιδιότητα "value" του ολισθαίνοντος ρυθμιστή τίθεται ενδιάμεσα σε μια αξία μεταξύ ελάχιστου και ανώτατου ανάλογα με τη θέση του ολισθαίνοντος ρυθμιστή. Ένας ολισθαίνων ρυθμιστής δημιουργείται με τη δημιουργία ενός `icontrol` του οποίου ιδιότητα ύφους είναι "ολισθαίνων ρυθμιστής". Ένας ολισθαίνων ρυθμιστής μπορεί να προστεθεί σε ένα GUI με τη χρησιμοποίηση του εργαλείου ολισθαίνοντων ρυθμιστών στο συντάκτη σχεδιαγράμματος. Το σχήμα 1.21a παρουσιάζει το σχεδιάγραμμα για ένα απλό GUI που περιέχει έναν ολισθαίνοντα ρυθμιστή και έναν τομέα κειμένων. Η "ελάχιστη" ιδιότητα για αυτόν τον ολισθαίνοντα ρυθμιστή τίθεται μηδέν, και η "ανώτατη" ιδιότητα τίθεται δέκα. Όταν ένας χρήστης σέρνει τον ολισθαίνοντα ρυθμιστή, καλεί αυτόματα τη λειτουργία `Slider1_Callback`, η οποία παρουσιάζεται στο σχήμα 1.21b. Αυτή η λειτουργία παίρνει την τιμή του ολισθαίνοντος ρυθμιστή από την ιδιότητα "τιμής" και επιδεικνύει την τιμή στον τομέα κειμένων. Το σχήμα 1.22 παρουσιάζει αυτήν την GM με τον ολισθαίνοντα ρυθμιστή σε κάποια ενδιάμεση θέση στη σειρά του.



Σχήμα 1.21 (α) Σχεδιάγραμμα ενός απλού GUI με έναν ολισθαίνοντα ρυθμιστή και έναν τομέα κειμένων. (β) Η λειτουργία επανάκλησης για αυτό το GUI.



Σχήμα 1.22 το GUI που παράγεται από το πρόγραμμα test_slider.

1.5 Πλαίσια διαλόγου(Dialog Boxes)

Ένα πλαίσιο διαλόγου είναι ένας ειδικός τύπος αριθμού που χρησιμοποιείται στις πληροφορίες επίδειξης ή για να πάρει την εισαγωγή από έναν χρήστη. Τα πλαίσια διαλόγου χρησιμοποιούνται στα λάθη επίδειξης, παρέχουν τις προειδοποιήσεις, υποβάλλουν τις ερωτήσεις, ή παίρνουν την εισαγωγή χρηστών. Χρησιμοποιούνται επίσης για να επιλέξουν τα αρχεία ή τις ιδιότητες εκτυπωτών. Τα πλαίσια διαλόγου μπορούν να είναι τοπικά ή μη-τοπικά. Ένα τοπικό πλαίσιο διαλόγου δεν επιτρέπει σε οποιοδήποτε άλλο παράθυρο στην εφαρμογή για να προσεγγιστεί έως ότου απομακρύνεται, ενώ ένα μη-τοπικό πλαίσιο διαλόγου δεν εμποδίζει την πρόσβαση σε άλλα παράθυρα. Τα τοπικά πλαίσια διαλόγου χρησιμοποιούνται χαρακτηριστικά για τα μηνύματα προειδοποίησης και λάθους, ότι υπάρχει επείγουσα προσοχή ανάγκης και δεν μπορούν να αγνοηθούν. Εξ ορισμού, τα περισσότερα πλαίσια διαλόγου είναι μη-τοπικά, και το MATLAB περιλαμβάνει πολλούς τύπους πλαισίων διαλόγου, ενώ ο σημαντικότερος συνοψίζεται στον πίνακα 1.4. Θα εξετάσουμε μόνο μερικούς από τους τύπους πλαισίων διαλόγου εδώ, αλλά μπορείτε να συμβουλευθείτε την σε απευθείας σύνδεση τεκμηρίωση MATLAB για τις άλλες λεπτομέρειες.

Property	Description
dialog	Creates a generic dialog box.
errordlg	Displays an error message in a dialog box. The user must click the OK button to continue.
helpdlg	Displays a help message in a dialog box. The user must click the OK button to continue.
inputdlg	Displays a request for input data and accepts the user's input values.
listdlg	Allows a user to make one or more selections from a list.
prindlg	Displays a printer selection dialog box.
questdlg	Asks a question. This dialog box can contain either two or three buttons, which by default are labeled Yes, No, and Cancel.
uigetfile	Displays a file open dialog box. This box allows a user to select a file to open, but does not actually open the file.
uiputfile	Displays a file save dialog box. This box allows a user to select a file to save, but does not actually save the file.
uicontrol	Displays a color selection dialog box.
uisetfont	Displays a font selection dialog box.
warndlg	Displays a warning message in a dialog box. The user must click the OK button to continue.

Πινάκας 1.4 Επιλεγμένα πλαίσια διαλόγου



Σχήμα 1.25 Ένα πλαίσιο διαλόγου λάθους

1.5.1 Πλαίσια διαλόγου λάθους και προειδοποίησης

Τα πλαίσια διαλόγου λάθους και προειδοποίησης έχουν παρόμοιες παραμέτρους και συμπεριφορά. Στην πραγματικότητα, η μόνη διαφορά μεταξύ τους είναι η εικόνα που επιδεικνύεται στο πλαίσιο διαλόγου. Η πιο κοινή ακολουθία που καλούν αυτά τα πλαίσια διαλόγου είναι:

```
errordlg(error_string,box_title,create_mode);  
warnrdlg(warning_string,box_title,create_mode);
```

Error_string είναι το μήνυμα που επιδεικνύει στο χρήστη, και το box_title είναι ο τίτλος του πλαισίου διαλόγου. Τέλος, το create_mode είναι μια σειρά που μπορεί να είναι "τοπική" ή "nonmodal", ανάλογα με τον τύπο πλαισίου διαλόγου που θέλετε να δημιουργήσετε. Παραδείγματος χάριν, η ακόλουθη δήλωση δημιουργεί ένα τοπικό μήνυμα λάθους που δεν μπορεί να αγνοηθεί από το χρήστη. Το πλαίσιο διαλόγου που παράγεται από αυτήν την δήλωση παρουσιάζεται στο σχήμα 1.25.

```
errordlg("Invalid τιμές εισαγωγής!", "Πλαίσιο διαλόγου λάθους", "modal")
```

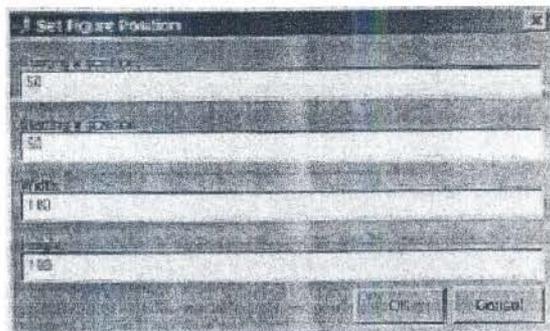
1.5.2 Εισηγμένα πλαίσια διαλόγου

Τα πλαίσια διαλόγου εισαγωγής προτρέπουν έναν χρήστη να εισαγάγει μια ή περισσότερες τιμές που μπορούν να χρησιμοποιηθούν από ένα πρόγραμμα. Τα πλαίσια διαλόγου εισαγωγής μπορούν να δημιουργηθούν καλώντας μια από τις ακόλουθες ακολουθίες.

```
answer = inputdlg(prompt)  
answer = inputdlg(prompt,title)  
answer = inputdlg(prompt,title,line_no)  
answer = inputdlg(prompt,title,line_no,default_answer)
```

Εδώ, η υπαγόρευση είναι μια σειρά απο στοιχεία, με κάθε στοιχείο της σειράς που αντιστοιχεί σε μια τιμή ότι ο χρήστης θα κληθεί να εισαγάγει. Ο τίτλος παραμέτρου διευκρινίζει τον τίτλο του πλαισίου διαλόγου, και το line_no διευκρινίζει τον αριθμό γραμμών που επιτρέπονται για κάθε απάντηση.

Τέλος, `default_answer` είναι μια σειρά πινάκων που περιέχει τις απαντήσεις προεπιλογής που θα χρησιμοποιηθούν εάν ο χρήστης αποτύχει να εισαγάγει τα στοιχεία για ένα ιδιαίτερο στοιχείο. Σημειώστε ότι πρέπει να υπάρξουν τόσες απαντήσεις προεπιλογής όσες είναι οι υπαγορεύσεις.



Σχήμα 1.26 ένα πλαίσιο διαλόγου εισαγωγής

Όταν ο χρήστης χτυπά το πληκτρο OK στο πλαίσιο διαλόγου, οι απαντήσεις του θα επιστραφούν ως μια σειρά στη μεταβλητή `answer`. Σαν παράδειγμα ενός πλαισίου διαλόγου εισαγωγής, υποθέστε ότι θελήσαμε να επιτρέψουμε σε έναν χρήστη για να διευκρινίσουμε τη θέση ενός αριθμού χρησιμοποιώντας έναν διάλογο εισαγωγής. Ο κώδικας για να εκτελέσει αυτήν την λειτουργία θα ήταν:

```
prompt{1} = 'Starting x position:';
prompt{2} = 'Starting y position: ';
prompt{3} = 'Width: ';
prompt{4} = 'Height: ';
title = 'Set Figure Position';
default_ans = {'50','50','180','100'};
answer = inputdlg(prompt,title,1,default_ans);
```

Το προκύπτον πλαίσιο διαλόγου παρουσιάζεται στο σχήμα 1.26.

1.5.3 Τα `uigetfile` και `uisetfile` πλαίσια διαλόγου

Τα `uigetfile` και `uisetfile` πλαίσια διαλόγου σχεδιάζονται για να επιτρέψουν σε έναν χρήστη και να επιλέξουν αμφίδρομα τα αρχεία που ανοίγουν ή που σώζουν.

Αυτά τα πλαίσια διαλόγου επιστρέφουν το όνομα και την πορεία του αρχείου αλλά όχι πραγματικά για να διαβάσει ή να σώσει. Ο προγραμματιστής είναι αρμόδιος για το γράψιμο του συμπληρωματικού κώδικα για εκείνο τον σκοπό. Η μορφή αυτών των δύο πλαισίων διαλόγου είναι

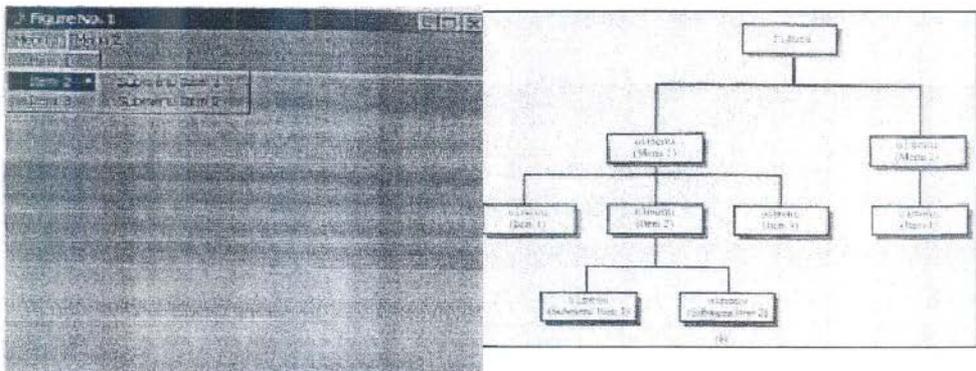
```
[filename, pathname] = uigetfile(filter_spec,title);
[filename, pathname] = uisetfile(filter_spec,title);
```


Property	Description
Accelerator	A single character specifying the keyboard equivalent for the menu item. The keyboard combination CTRL + key allows a user to activate the menu item from the keyboard.
Callback	Specifies the name and parameters of the function to be called when the menu item is activated. If the menu item has a submenu, the callback executes before the submenu is displayed. If the menu item does not have submenus, then the callback executes when the mouse button is released.
Checked	When this property is 'on', a checkmark is placed to the left of the menu item. This property can be used to indicate the status of menu items that toggle between two states. Possible values are 'on' or 'off'.
Enable	Specifies whether or not this menu item is selectable. If it is not enabled, the menu item will not respond to mouse clicks or accelerator keys. Possible values are 'on' or 'off'.
Label	Specifies the text to be displayed on the menu. The ampersand character (&) can be used to specify a keyboard mnemonic for this menu item. If not appear on the label. For example, the string '&file' will create a menu item displaying the text 'file' and responding to the F key.
Parent	The handle of the parent object for this menu item. The parent object could be a figure or another menu item.
Position	Specifies the position of a menu item on the menu bar or within a menu. Position 1 is the left-most menu position for a top-level menu, and the highest position within a submenu.
Separator	When this property is 'on', a separating line is drawn above this menu item. Possible values are 'on' or 'off'.
Tag	The 'name' of the menu item, which can be used to locate it.
Visible	Specifies whether or not this menu item is visible. Possible values are 'on' or 'off'.

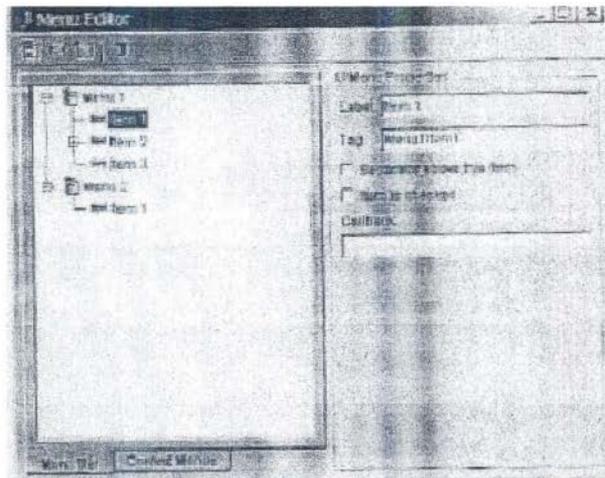
Πίνακας 1.5 Σημαντικές ιδιότητες του uimenu

Τα μενού είναι χρήσιμα για τις λιγότερο χρησιμοποιημένες επιλογές χωρίς να επικρατεί «αναρχία» στο GUI με πολλά πρόσθετα κουμπιά. Υπάρχουν δύο τύποι μενού στην MATLAB: τα τυποποιημένα μενού, τα οποία βρίσκονται από την μπάρα του μενού που βρίσκεται στην κορυφή του σχήματος (πλαίσιο), και τα μενού πλαίσιο, τα όποια αναδύονται πάνω από το σχήμα (πλαίσιο) όταν ο χρήστης κάνει δεξί κλικ με το ποντίκι πάνω στο γραφικό αντικείμενο. Θα μάθουμε πώς μπορούμε να δημιουργήσουμε και να χρησιμοποιήσουμε και τους δύο τύπους μενού σε αυτό το σημείο. Τυποποιημένα μενού δημιουργούνται με την βοήθεια των αντικείμενων uimenu. Κάθε στοιχείο σε ένα μενού είναι ένα ξεχωριστό αντικείμενο uimenu, συμπεριλαμβανομένων και των στοιχείων στα δευτερεύοντα μενού. Τα αντικείμενα uimenu είναι παρόμοια με τα αντικείμενα uicontrol, και έχουν πολλές ομοιότητες όσο αφορά τις ιδιότητες τους όπως η κληρονομικότητα, την επανάκληση, και ούτω καθ' εξής. Ο κατάλογος των σημαντικότερων ιδιοτήτων των uimenu παρατίθεται στον πίνακα 1.5.

Κάθε στοιχείο των μενού είναι συνδεδεμένο με ένα αντικείμενο πηγής, όπου είναι στην κορυφή του επιπέδου των μενού, ή ένα άλλο στοιχείο μενού για τα δευτερεύοντα υπομενού. Όλα τα uimenus συνδέονται με την ίδια πηγή τα οποία εμφανίζονται στο ίδιο μενού, και ο καταρράκτης των στοιχείων διαμορφώνει ένα δέντρο



Σχήμα 1.28 (α) μία χαρακτηριστική δομή μενού . (β) Οι σχέσεις μεταξύ των στοιχείων uimenu που δημιουργούν τα μενού.



Σχήμα 1.28 (συνέχεια). (γ) Η δομή του συντάκτη μενού (Menu Editor) που παράγει αυτά τα μενού.

από τα υπομενού. Στο σχήμα 1.2ε παρουσιάζονται στην MATLAB χαρακτηριστικά μενού σε λειτουργία, ενώ στο σχήμα 1.28b παρουσιάζεται η σχέση μεταξύ των αντικειμένων που αποτελούν τα μενού. Στην MATLAB τα μενού δημιουργούνται χρησιμοποιώντας το συντάκτη μενού (Menus Editor). Στο σχήμα 1.28c παρουσιάζεται ο συντάκτης μενού (Menu Editor) με τα στοιχεία του μενού που παράγουν αυτήν την δομή του μενού. Οι πρόσθετες ιδιότητες στον πίνακα 1.5 δεν παρουσιάζονται στο συντάκτη μενού (Menu Editor) αλλά μπορούν να τεθούν με το συντάκτη ιδιοτήτων (propedit). Τα υψηλότερα (κορυφαία) μενού πλαίσιων δημιουργούνται από τα αντικείμενα uicontextmenu, και τα χαμηλότερα στοιχεία μέσα στα μενού στοιχείων δημιουργούνται από τα αντικείμενα uimenu. Τα μενού πλαισίου είναι βασικά τα ίδια με τα τυποποιημένα (Standard) μενού, εκτός από το ότι μπορούν να συνδεθούν με οποιοδήποτε αντικείμενο GUI (άξονες, γραμμές, κείμενα, σχήματα, κ.λπ.). Ένας κατάλογος των σημαντικότερων ιδιοτήτων uicontextmenu παρατίθεται στον πίνακα 1.6.

1.6.1 Καταστολή των επιλογών προεπιλογής

Κάθε MATLAB σχήμα έρχεται με ένα σύνολο προεπιλεγμένων τυποποιημένων μενού. Εάν θέλετε να διαγράψετε αυτά τα μενού από ένα σχήμα και να δημιουργήσετε τα δικά σας μενού, θα πρέπει πρώτα να απενεργοποιήσετε την προεπιλογή του μενού. Η εμφάνιση του μενού προεπιλογής ελέγχονται από την ιδιότητα MenuBar. Οι πιθανές τιμές αυτής της ιδιότητας είναι "Figure" και "none".

Property	Description
callback	Specifies the name and parameters of the function to be called when the context menu is activated. The callback executes before the context menu is displayed.
parent	The handle of the parent object for this context menu.
tag	The 'name' of the context menu, which can be used to locate it.
visible	Specifies whether or not this context menu is visible. This property is set automatically and should normally not be modified.

Πίνακας 1.6 οι σημαντικότερες ιδιότητες uicontextmenu

Εάν η ιδιότητα τίθεται ως στόχος "figure", κατόπιν τα μενού προεπιλογής εμφανίζονται. Εάν η ιδιότητα τίθεται "none", έπειτα τα μενού προεπιλογής καταστέλλονται. Μπορείτε να χρησιμοποιήσετε την Property Inspector για να θέσετε την ιδιότητα MenuBar για το GUIs όταν τους δημιουργείτε.

1.6.2 Δημιουργία των δικό σας μενού

Η δημιουργία των δικό σας τυποποιημένων μενού για μια GM είναι βασικά μια διαδικασία τριών βημάτων.

1. *Πρώτα*, δημιουργήστε μια νέα δομή μενού με το συντάκτη μενού (Menu Editor). Χρησιμοποιήστε το συντάκτη επιλογών (Menu Editor) για να καθορίσετε τη δομή, που δίνει σε κάθε στοιχείο του μενού μια ετικέτα κατά την εμφάνιση και μια μοναδική τιμή ετικέτας. Επίσης, πρέπει χειρονακτικά να δημιουργήσετε τη σειρά επανάκλησης για τα στοιχεία του μενού. Αυτή η δήλωση είναι αληθής για την MATLAB 6,5. Εντούτοις, στη MATLAB 6,5 τροποποιήθηκε για να παραγάγει αυτόματα τις σειρές επανάκλησης του μενού. Αυτό το χειρονακτικό βήμα δεν είναι απαραίτητο στην MATLAB 6,5 και στις επόμενες εκδόσεις της. Αυτός ο τρόπος μπορεί να είναι δυσνόητος- αν και ένας καλύτερος τρόπος για να δημιουργηθεί μια επανάκληση μενού είναι να εξετάσει την επανάκληση και να δημιουργείται αυτόματα από ένα uicontrol, και το χρησιμοποιήστε το για παράδειγμα. Η κατάλληλη μορφή μιας σειράς επανάκλησης ενός uimenu είναι

```
MyGui('MenuitemTag_Callback',gcbo, [J ,guidata(gcbo))
```

όπου πρέπει να αντικαταστήσετε το όνομα GUI ως MyGUI, και την τιμή ετικετών των στοιχείων του μενού στο MenuitemTag.

2. *Δεύτερον*, συντάξτε τις ιδιότητες κάθε στοιχείου του μενού χρησιμοποιώντας το συντάκτη ιδιοτήτων (propedit), εάν αυτό είναι απαραίτητο. Οι σημαντικότερες ιδιότητες που θέτονται είναι η επανάκληση, και η ετικέτα, οι οποίες μπορούν να τεθούν και από το συντάκτη του μενού (Menu Editor), έτσι ο συντάκτης ιδιοτήτων συνήθως δεν είναι χρήσιμος. Εντούτοις, εάν πρέπει να θέσετε οποιοδήποτε από τις άλλες ιδιότητες που απαριθμούνται στον πίνακα 1.5, θα πρέπει να χρησιμοποιήσετε τον συντάκτη ιδιοτήτων. Ο κατάλογος ιδιοτήτων που μπορεί να τεθεί από το συντάκτη των μενού (Menu Editor) είναι επεκταμένος στην MATLAB 6,5, έτσι τώρα είναι περιττό να χρησιμοποιηθεί ο συντάκτης ιδιοτήτων. Όλες οι ιδιότητες μπορούν τώρα να τεθούν άμεσα στο συντάκτη μενού (Menu editor) στην MATLAB 6,5 και σε μετέπειτα εκδόσεις.

3. Ο τρίτος, εφαρμόζει μια λειτουργία επανάκλησης για να εκτελέσει τις ενέργειες που απαιτούνται από στοιχεία του μενού. Θα πρέπει χειρονακτικά να δημιουργήσετε τη λειτουργία επανάκλησης για τα στοιχεία του μενού. Η διαδικασία της οικοδόμησης του μενού θα διευκρινιστεί σε ένα παράδειγμα στο τέλος αυτού του τμήματος.

Ορθή πρακτική προγραμματισμού

Αντίθετα από τα αντικείμενα GUI, στην MATLAB δεν δημιουργούνται αυτόματα οι σειρές των επανακλήσεων και ο κορμός των λειτουργιών για τα στοιχεία του μενού. Πρέπει να εκτελέσετε αυτήν την λειτουργία χειρονακτικά. Μόνο η ετικέτα, η επανάκληση, ο έλεγχος, και ο διαχωρισμός των ιδιοτήτων ενός στοιχείου του μενού μπορούν να τεθούν από το

συντάκτη μενού (Menu Editor). Εάν πρέπει να θέσετε οποιοσδήποτε από τις άλλες ιδιότητες, θα πρέπει να χρησιμοποιήσετε την ιδιότητα συντάκτη (propedit) στο σχήμα (πλαίσιο), και επιλέξετε το κατάλληλο στοιχείο του μενού για σύνταξη.

1.6.3 Πλήκτρα επιταχυντές και πληκτρολόγιο "μνήμης" (Keyboard Mnemonics)

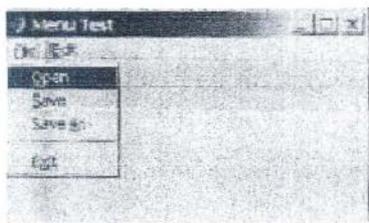
Στην MATLAB τα μενού υποστηρίζουν και μνημονικό πληκτρολόγιο. Πλήκτρα επιταχυντές είναι «CTRL+πλήκτρο» συνδυασμοί που αναγκάζουν ένα στοιχείο ενός μενού να εκτελεσθεί χωρίς πρώτα να έχει προηγηθεί άνοιγμα του συγκεκριμένου μενού. Παραδείγματος χάριν, το πλήκτρο επιταχυντή "O" έχει οριστεί το άνοιγμα/αρχείο του μενού. Σε αυτή την περίπτωση, ο συνδυασμός CTRL+ o του πληκτρολογίου θα προκαλέσει το άνοιγμα/αρχείο και εκτελείται η λειτουργία επανάκληση.

Μερικοί Συνδυασμοί CTRL+πλήκτρο είναι δεσμευμένοι για τη χρήση του λειτουργικού συστήματος του "φιλοξενούμενου". Αυτοί οι συνδυασμοί διαφέρουν μεταξύ των συστημάτων PC και Unix γι' αυτό συμβουλευθείτε την MATLAB σε απευθείας σύνδεση για να καθορίσετε και να τεκμηριώσετε ποιοι συνδυασμοί είναι δυνατοί για τον τύπο υπολογιστή σας. Τα πλήκτρα των επιταχυντών καθορίζονται με τον καθορισμό της ιδιότητας επιταχυντών στο αντικείμενο uimenu. Το μνημονικό πληκτρολογίων είναι ενιαίες εντολές που μπορούν να πιεστούν για να προκαλέσουν ένα στοιχείο του μενού να εκτελείται μόλις τα μενού είναι ανοικτά. Η μνημονική εντολή πληκτρολογίων για ένα δεδομένο στοιχείο του μενού είναι υπογραμμισμένη. Για το μενού τα οποία βρίσκονται στην κορυφή, το πληκτρολόγιο μνημονικό εκτελείται με τη συμπίεση του ALT συν το μνημονικό πλήκτρο συγχρόνως. Μόλις το μενού το οποίο βρίσκεται στην κορυφή είναι ανοικτό, απλά πιάστε το μνημονικό πλήκτρο και θα αναγκάσει ένα στοιχείο του μενού να εκτελεστεί. Στο σχήμα 1.29 εξηγείτε η χρήση του μνημονικού πληκτρολογίων. Το μενού αρχείου ανοίγει με τα πλήκτρα ALT+f, και μόλις αυτό ανοιχθεί, το στοιχείο μενού Exit μπορεί να εκτελεσθεί απλά με την πληκτρολόγηση του "X".

Το μνημονικό πληκτρολογίων καθορίζεται με την τοποθέτηση του χαρακτήρα ampersand (&) πριν από την επιθυμητή μνημονική επιστολή στην ιδιότητα της ετικέτας. Το ampersand δεν θα εμφανισθεί αλλά η ακόλουθη επιστολή θα υπογραμμιστεί, και θα ενεργήσει ως μνημονικό κλειδί. Παραδείγματος χάριν, η ιδιότητα της ετικέτας του στοιχείου του μενού Exit στο σχήμα 1.29 είναι "E&xit".

1.6.4 Δημιουργία των μενού πλαίσιου

Τα μενού πλαίσιου δημιουργούνται με τον ίδιο τρόπο με τα συνηθισμένα μενού, εκτός από το ότι στοιχείο του μενού που βρίσκετε στην κορυφή είναι ένα uicontextmenu. Η πηγή ενός uicontextmenu



Σχήμα 1.29 το μενού που παρουσιάζετε από τη πληκτρολόγηση των πλήκτρων ALT+f, και η Exit επιλογή θα μπορούσε να εκτελεσθεί με απλά με την πληκτρολόγηση "X".

πρέπει να είναι ένα σχήμα (πλαίσιο), αλλά το μενού πλαισίου μπορεί να συνδεθεί και να αποκριθεί στο πάτημα του δεξιού πλήκτρου του ποντικιού σε οποιοδήποτε γραφικό αντικείμενο. Τα μενού πλαισίου δημιουργούνται χρησιμοποιώντας την «Context Menu» επιλογή στο συντάκτη μενού (Menu Editor). Μόλις δημιουργηθούν το μενού πλαισίου, οποιοσδήποτε αριθμός μενού μπορεί να δημιουργηθεί κάτω από αυτό. Για να συνδεθεί το μενού πλαισίου με ένα συγκεκριμένο αντικείμενο, θα πρέπει να θέτει την ιδιότητα `UIContextMenu` του αντικειμένου στο `uicontextmenu`. Αυτό κανονικά γίνεται χρησιμοποιώντας το Property Inspector, αλλά μπορεί να γίνει και με μια καθορισμένη εντολή όπως παρουσιάζεται στην συνέχεια. Εάν `Hcm` είναι ο χειριστής σε μενού πλαισίου, τότε ακολουθήστε τις παρακάτω δηλώσεις και συνδέστε το μενού πλαισίου με μια γραμμή που δημιουργείται από μια γραμμή εντολών.

```
H1 = plot (x, y);  
set (H1,'UicontextMenu',Hcm);
```

ΚΕΦΑΛΑΙΟ 2.

ΠΕΡΙΒΑΛΛΟΝ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ

Για να δημιουργήσουμε το περιβάλλον που θα χρησιμοποιηθεί για την Ψηφιακή Επεξεργασία εικόνας έπρεπε χρησιμοποιήσουμε το ολοκληρωμένο προγραμματιστικό περιβάλλον της MATLAB (στην έκδοση 6.5.1), τις συναρτήσεις που παρέχει η εργαλειοθήκη Image Processing Toolbox καθώς και το περιβάλλον δημιουργίας γραφικού ενδιάμεσου με το χρήστη guide (το παρέχει η Matlab).

Αρχικά έπρεπε να καθορίσουμε ποιο θα είναι το βασικό μενού τη εφαρμογής, οπότε και αποφασίστηκε να είναι το εξής:

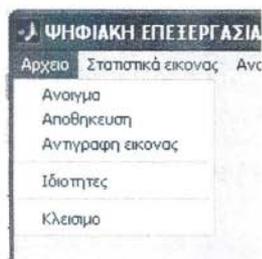
Βασικό μενού εφαρμογής «ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ»	
Αρχείο	Παρέχει επιλογές που σχετίζονται με το άνοιγμα αρχείων εικόνων, την αποθήκευσή τους, το κλείσιμο της εφαρμογής κ.τ.λ.
Στατιστικά εικόνας	Παρέχει πληροφορίες για τα στατιστικά της εικόνας, όπως το ιστόγραμμα, η μέγιστη και ελάχιστη τιμή φωτεινότητας κ.τ.λ.
Ανάλυση εικόνας	Μας δίνει τη δυνατότητα να επιλέξουμε μεταξύ διαφόρων κατηγοριών, όπως ενίσχυση εικόνας, πρόσθεση θορύβου, φιλτράρισμα εικόνας κ.τ.λ.
Μετασχηματισμοί	Παρέχεται η δυνατότητα να δούμε το διδιάστατο μετασχηματισμό Fourier
Βοήθεια	Πληροφορίες για τους συντελεστές της εφαρμογής

Αναλυτικότερα, για κάθε επιλογή από το βασικό μενού της εφαρμογής είχαμε τα εξής:

Η επιλογή Αρχείο

Στην επιλογή Αρχείο, έχουμε την εμφάνιση των εξής κατηγοριών:

Επιλογή Αρχείο	
Άνοιγμα	Δίνεται η δυνατότητα εμφάνισης ενός παραθύρου διαλόγου, με σκοπό να ανακτήσουμε ένα αρχείο εικόνας. Η όλη διαδικασία γίνεται με αλληλεπιδραστικό τρόπο
Αποθήκευση	Δίνεται η δυνατότητα εμφάνισης ενός παραθύρου διαλόγου, με σκοπό να αποθηκεύσουμε ένα αρχείο εικόνας. Η όλη διαδικασία γίνεται με αλληλεπιδραστικό τρόπο
Αντιγραφή εικόνας	Δίνεται η δυνατότητα αντιγραφής της αρχικής εικόνας
Ιδιότητες	Στη παρούσα μορφή της εφαρμογής οι ιδιότητες που εμφανίζονται αφορούν μόνο το μέγεθος της εικόνας
Κλείσιμο	Κλείνει η εφαρμογή, αλλά και το περιβάλλον της Matlab.



Στην διπλανή εικόνα βλέπουμε τις επιλογές που προκύπτουν αφού ο χρήστης επιλέξει το μενού Αρχείο.

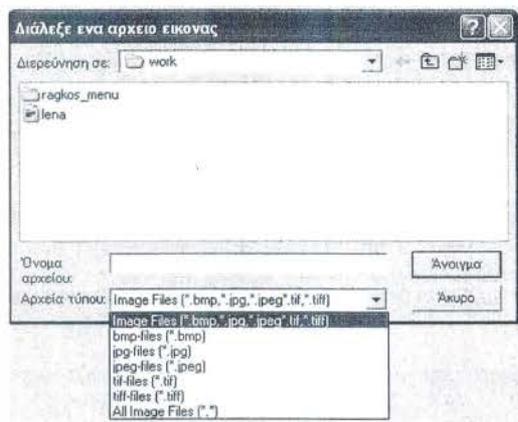
Ο αλληλεπιδραστικός τρόπος ανάκτησης ενός αρχείου μέσα στη δομή φακέλων των Windows καθώς και η αποθήκευση μέσα στην ίδια δομή επιτεύχθηκε με τη χρήση της συνάρτησης uigetfile για

την αναζήτηση και ανάκτηση και της `uidputfile` για την διαδικασία της αποθήκευσης.

Οι μορφοποιήσεις αρχείων που μπορούμε να ανοίξουμε αλλά και να αποθηκεύσουμε στην προκείμενη εφαρμογή είναι:

`bmp`, `jpg`, `jpeg`, `tif`, `tiff`.

όπως εξάλλου φαίνεται και στην εικόνα που ακολουθεί:



Για την αντιγραφή της εικόνας απλά αντιγράφουμε τη μεταβλητή που αναπαριστά την εικόνα και στη συνέχεια την εμφανίζουμε σε ένα νέο παράθυρο (`figure`).

Όσον αφορά τις Ιδιότητες στην παρούσα μορφή της εφαρμογής έχουμε πληροφορίες μόνο για το μέγεθος της εικόνας. Περαιτέρω έκδοση της εφαρμογής θα μπορούσε να εμφανίσει πληροφορίες για το μέγεθος της εικόνας, για το βάθος χρώματος κ.τ.λ. με τη χρήση της συνάρτησης της Matlab `imfinfo`.

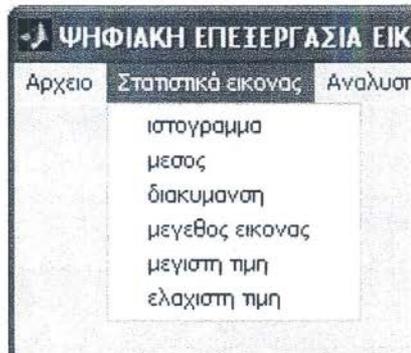
Η επιλογή Στατιστικά εικόνας

Στην επιλογή Στατιστικά εικόνας, έχουμε την εμφάνιση των εξής κατηγοριών:

Επιλογή Στατιστικά εικόνας	
Ιστόγραμμα	Μας δίνεται η δυνατότητα να δούμε το ιστόγραμμα της εικόνας. Πριν την εμφάνιση του ιστογράμματος έχουμε τη δυνατότητα να καθορίσουμε το πλήθος των bins (δηλαδή το πλήθος των διαστημάτων τιμών φωτεινότητας) που θα εμφανιστούν στο ιστόγραμμα
Μέσος	Μας δίνεται η δυνατότητα εμφάνισης της μέσης τιμής της φωτεινότητας της εικόνας με τη συνάρτηση της Matlab <code>mean2</code>
Διακύμανση	Αντίστοιχα με προηγουμένως μας δίνεται η δυνατότητα υπολογισμού της διακύμανσης των τιμών της φωτεινότητας της εικόνας
Μέγεθος εικόνας	Εμφανίζεται το μέγεθος της εικόνας σε Pixels
Μέγιστη τιμή	Εμφανίζεται σε ένα παράθυρο μηνύματος η μέγιστη τιμή φωτεινότητας

Ελάχιστη τιμή	Εμφανίζεται σε ένα παράθυρο μηνύματος η ελάχιστη τιμή φωτεινότητας
----------------------	--

Στην εικόνα που ακολουθεί βλέπουμε τις επιλογές που μας παρέχει το μενού Στατιστικά εικόνας.

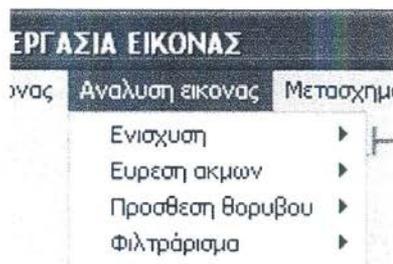


Η επιλογή Ανάλυση εικόνας

Στην επιλογή Ανάλυση εικόνας, έχουμε την εμφάνιση των εξής κατηγοριών:

Επιλογή Ανάλυση εικόνας	
Ενίσχυση	Μας παρέχει τη δυνατότητα, με την εμφάνιση των υπο-μενού που εμφανίζονται να ενισχύσουμε την εικόνα με τη βοήθεια των αλγορίθμων ισοστάθμισης ιστογράμματος και αύξησης της αντίθεσης
Εύρεση ακμών	Μπορούμε με τη χρήση κατάλληλων τελεστών (Sobel, Prewitt, Roberts) και του ανιχνευτή ακμών Canny να ανιχνεύσουμε τις ακμές που περιέχονται στην εικόνα μας
Πρόσθεση θορύβου	Με τη χρήση της συνάρτησης imnoise μπορούμε να προσθέσουμε στην εικόνα μας διάφορες ζμορφές θορύβου όπως Gaussian, salt & pepper, poisson, speckle.
Φιλτράρισμα	Στην παρούσα μορφή της εφαρμογής, έχουμε τη δυνατότητα να διώξουμε το θόρυβο που είναι της μορφής salt & pepper

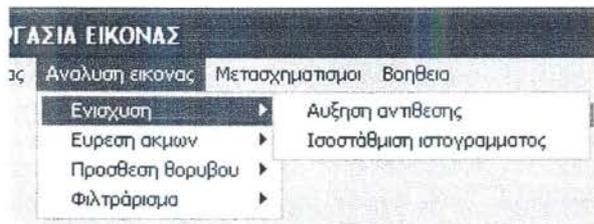
Η εικόνα που ακολουθεί δείχνει τις προαναφερόμενες επιλογές όπως αυτές εμφανίζονται στην εφαρμογή.



Η επιλογή Ενίσχυση μας δίνει ένα υπο-μενού με τις επιλογές που φαίνονται στον πίνακα και στην εικόνα που ακολουθούν:

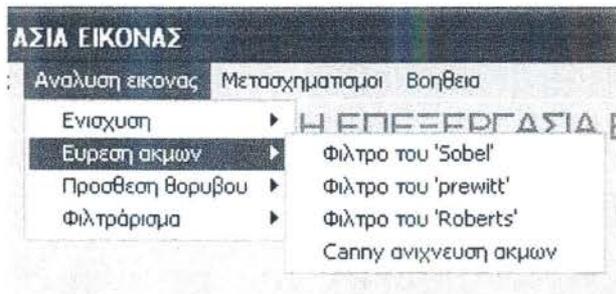
Επιλογές Ενίσχυση	
Αύξηση αντίθεσης	Χρησιμοποιώντας τη συνάρτηση της Matlab <code>imadjust</code> έχουμε τη δυνατότητα να αυξήσουμε την αντίθεση των φωτεινότητων των εικόνων
Ισοστάθμιση ιστογράμματος	Αντίστοιχα με τη χρήση της συνάρτησης <code>histeq</code> μπορούμε να επιτύχουμε ισοστάθμιση ιστογράμματος, άρα και αύξηση της αντίθεσης, συνεπώς ενίσχυση των λαπτομερειών της εικόνας

Η εικόνα που ακολουθεί μας δείχνει τις επιλογές του υπο-μενού Ενίσχυση.



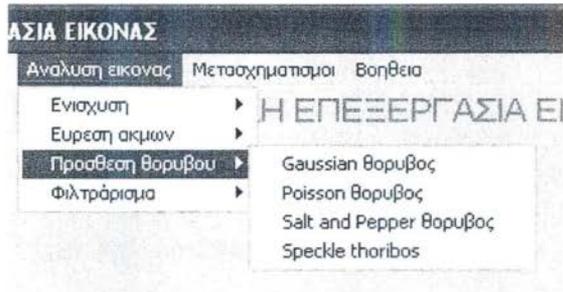
Η επιλογή Εύρεση ακμών μας δίνει τη δυνατότητα να χρησιμοποιήσουμε συναρτήσεις της Matlab για να επιτελέσουμε μία σημαντική εργασία στη ψηφιακή επεξεργασία εικόνας, η οποία είναι η ανίχνευση ακμών.

Στην εικόνα που ακολουθεί βλέπουμε το υπομενού Εύρεση ακμών:



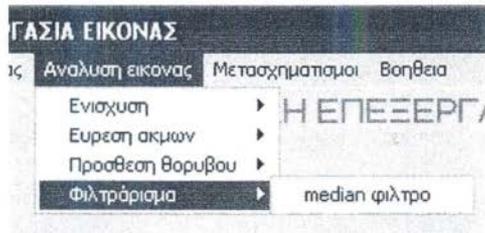
Η επιλογή Πρόσθεση θορύβου μας δίνει τη δυνατότητα να προσθέσουμε θόρυβο στις εικόνες μας, προσφέροντάς μας έτσι βοήθεια στην προσπάθειά μας να κατανοήσουμε τις διάφορες μορφές θορύβου επάνω στις εικόνες, καθώς εν συνεχεία με τη χρήση του φιλτραρίσματος στην εικόνα που έχει θόρυβο μορφής «salt & pepper», να παρατηρήσουμε την επανάκτησή της.

Στην εικόνα που ακολουθεί βλέπουμε τις διάφορες επιλογές του υπο-μενού Πρόσθεση θορύβου.



Η επιλογή φιλτράρισμα μας δίνει τη δυνατότητα να χρησιμοποιήσουμε το median φίλτρο για να διώξουμε τη μορφή θορύβου «salt & pepper».

Στην εικόνα που ακολουθεί βλέπουμε την επιλογή Φιλτράρισμα\

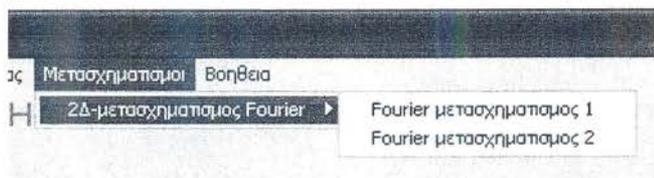


Οι επόμενες δύο εικόνες μας δείχνουν την εφαρμογή θορύβου «salt & pepper» και εν συνεχεία την εικόνα που παίρνουμε εάν εφαρμόσουμε την επιλογή του median φίλτρου στην εικόνα με θόρυβο.

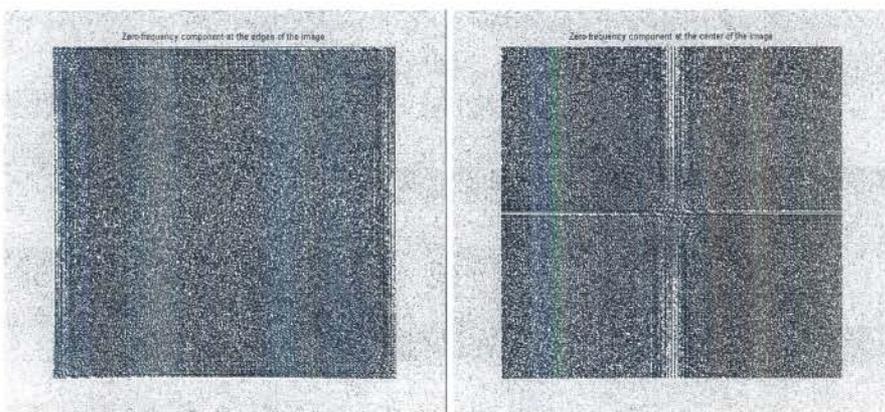


Εικόνα με θόρυβο μορφής "salt & pepper" και η εικόνα μετά την εφαρμογή του median φίλτρου

Στη συνέχεια έχουμε την επιλογή Μετασχηματισμοί όπου μας δίνεται η δυνατότητα να δούμε την εικόνα στο πεδίο των συχνοτήτων εφαρμόζοντας το διδιάστατο μετασχηματισμό Fourier στην εικόνα μας με τη χρήση των συναρτήσεων $\text{fft2}(\text{εικόνα})$ και $\text{fftshift}(\text{fft}(\text{εικόνα}))$. Η πρώτη επιλογή αφορά την εφαρμογή του διδιάστατου μετασχηματισμού Fourier χωρίς τη μετατόπιση των συστατικών μηδενικής συχνότητας στο κέντρο της εικόνας. Η δεύτερη επιλογή μετατοπίζει τα συστατικά μηδενικής συχνότητας στο κέντρο της εικόνας. Η εικόνα που ακολουθεί μας δείχνει το μενού των δύο επιλογών της εφαρμογής:



Οι επόμενες δύο εικόνες μας δείχνουν το πραγματικό μέρος του διδιάστατου μετασχηματισμού στην αρχική εικόνα της γυναικείας μορφής που είδαμε πιο πάνω.



Η τελευταία επιλογή αφορά τη Βοήθεια και μία πο τις επιλογές αφορά την εμφάνιση των δημιουργών της συγκεκριμένης εφαρμογής.

