

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ**



**ΤΙΤΛΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

**ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΚΕΤΟΥ ΑΣΚΗΣΕΩΝ ΣΤΟ ΜΑΤΛΑΒ ΓΙΑ  
ΤΟΝ ΠΡΟΣΔΙΟΡΙΣΜΟ ΘΕΣΗΣ ΚΙΝΟΥΜΕΝΟΥ ΣΩΜΑΤΟΣ ΣΤΟΝ ΧΩΡΟ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΑΠΟΣΤΟΛΟΣ ΚΟΥΙΡΟΥΚΙΔΗΣ**

**ΣΠΟΥΔΑΣΤΗΣ: ΚΑΤΣΙΟΥΛΑΣ ΕΥΑΓΓΕΛΟΣ**

**ΣΕΡΡΕΣ 2017**

**Υπεύθυνη Δήλωση**: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών και Πληροφορικής.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία με θέμα “Ανάπτυξη εκπαιδευτικού πακέτου ασκήσεων στο Matlab για τον Προσδιορισμό θέσης κινούμενο σώματος στον χώρο.”, πραγματοποιήθηκε, στο πλαίσιο του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής ΤΕ του Τ.Ε.Ι. Κεντρικής Μακεδονίας.

Στο σημείο αυτό αισθάνομαι την ανάγκη να εκφράσω τις ειλικρινείς και θερμές ευχαριστίες μου στον επιβλέποντα καθηγητή κύριο Κουιρουκίδη Απόστολο, επιστημονικό συνεργάτη του τμήματος Μηχανικών Πληροφορικής του ΤΕΙ Κεντρικής Μακεδονίας, τόσο για την προσφορά του θέματος, όσο και για την εμπιστοσύνη του. Τον ευχαριστώ επίσης για τις πολύτιμες γνώσεις και συμβουλές που μου παρείχε καθ’ όλη τη διάρκεια εκπόνησης της εργασίας, καθώς και για την καθοδήγηση και πολύτιμη βοήθεια του όποτε και όπου υπήρξε απαραίτητη.

Τέλος, θα ήθελα να ευχαριστήσω και όλους τους ανθρώπους εντός και εκτός του ακαδημαϊκού περιβάλλοντος, τους συμφοιτητές και φίλους που συνάντησα στα όμορφα αυτά φοιτητικά μου χρόνια, και πάνω απ’ όλα την οικογένεια μου για την στήριξη, υπομονή και συμπαράσταση τους.

## **ΠΕΡΙΛΗΨΗ**

Ως γλώσσα προγραμματισμού και εργαλείο γραφικής αναπαράστασης δεδομένων, το Matlab διαθέτει ένα πλούσιο σύνολο δυνατοτήτων και λειτουργιών για την επίλυση προβλημάτων σε τεχνικούς, επιστημονικούς, υπολογιστικούς και μαθηματικούς τομείς. Σπουδαία δυνατότητα του προγράμματος μεταξύ άλλων είναι και αυτή του προγραμματισμού καθώς επιτρέπει ανάπτυξη εφαρμογών παρόμοιων με εκείνες άλλων γλωσσών προγραμματισμού.

Ανάμεσα στα εργαλεία του Matlab ξεχωρίζουμε το εργαλείο διασύνδεσης με το χρήστη μέσω γραφικών (graphical user interface – GUI) το οποίο μας επιτρέπει να το χρησιμοποιήσουμε ως εφαρμογή εργαλείων ανάπτυξης. Έτσι ο συνδυασμός δομών δεδομένων πίνακα, δυνατοτήτων προγραμματισμού, και εργαλείων GUI κάνει το Matlab ένα εξαιρετικά ισχυρό εργαλείο για την επίλυση προβλημάτων σε διάφορους επιστημονικούς τομείς. Έτσι ως περιβάλλον ανάπτυξης της παρούσας εργασίας, όπου αντικείμενο αποτελεί η κατασκευή και η ανάπτυξη ενός εκπαιδευτικού πακέτου ασκήσεων το οποίο προσομοιώνει το προσδιορισμό θέσης κινούμενου σώματος στον χώρο, μέσω του Matlab.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ.....	6
1ο ΚΕΦΑΛΑΙΟ.....	7
1.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΚΙΝΗΣΗΣ.....	7
1.2 Ο ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ ΘΕΣΗΣ ΕΝΟΣ ΣΩΜΑΤΟΣ.....	7
1.3 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ ΘΕΣΗΣ ΣΩΜΑΤΙΟΥ ΣΕ ΕΥΘΕΙΑ ΓΡΑΜΜΗ.....	8
1.4 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ ΘΕΣΗΣ ΣΤΟ ΕΠΙΠΕΔΟ.....	9
1.5 Η ΜΕΤΑΤΟΠΙΣΗ ΣΩΜΑΤΙΟΥ ΣΕ ΑΞΟΝΑ.....	12
ΚΕΦΑΛΑΙΟ 2 <sup>ο</sup> .....	14
2.1 ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB.....	14
2.2 ΠΕΡΙΒΑΛΛΟΝ MATLAB.....	14
2.3 Τι είναι το GUIDE.....	16
2.4 Δημιουργία ενός παραθύρου GUI.....	18
2.5 ΠΑΡΟΥΣΙΑΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ MATLAB.....	21
2.5.1 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ 1η ΑΣΚΗΣΗΣ.....	22
2.5.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ 2η ΑΣΚΗΣΗΣ.....	24
2.5.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ 3η ΑΣΚΗΣΗΣ.....	26
ΚΕΦΑΛΑΙΟ 3ο.....	28
3.1 ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	28
3.1.1 Το m-filegcentral.....	28
3.1.2 Το m-filegthesn1d – Άσκηση 1 (Κίνηση σε μία Διάσταση).....	32
3.1.3 Το m-filegthesn2d – Άσκηση 2 (Κίνηση σε δύο Διαστάσεις).....	44
3.1.4 Το m-fileg8esn1 – Άσκηση 3 (Κίνηση σε Τρεις Διαστάσεις).....	55
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	66

## **ΕΙΣΑΓΩΓΗ**

Κεντρική ιδέα της παρούσας πτυχιακής εργασίας ήταν να αναπτυχθεί ένα περιβάλλον προσομοίωσης για πειράματα φυσικής τα οποία αναφέρονται στους νόμους του προσδιορισμού θέσης κινούμενου σώματος στον χώρο.

Η δημιουργία της παραθυρικής αυτής εφαρμογής αναπτύχθηκε με την βοήθεια του Matlab σε περιβάλλον GUI . Η εργαλειοθήκη GUI ή αλλιώς GUIDE (Graphical User Interface Design Environment) , το οποίο αποτελεί μία τεχνική προγραμματισμού υψηλού επιπέδου, περιέχει μία πληθώρα χρήσιμων εργαλείων ελέγχου όπως κουμπιά , πλαίσια κ.α.

Σε πρώτη προσέγγιση, στο παρόν εγχειρίδιο περιγράφεται αναλυτικά το γνωστικό αντικείμενο που πραγματεύεται η προσομοίωση σε θεωρητικό υπόβαθρο, δηλαδή τους νόμους και τους ορισμούς του προσδιορισμού θέσης κινούμενου σώματος στο χώρο.

Εν συνεχεία περιγράφεται πως η θεωρητική αυτή προσέγγιση μεταφέρεται και αναπτύσσεται σε ένα διαδραστικό περιβάλλον, το οποίο θα μπορούσε να χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς σε ένα εργαστήριο φυσικής.

# 1<sup>ο</sup> ΚΕΦΑΛΑΙΟ

Παρακάτω παρατίθενται κάποιοι ορισμοί και κανόνες για το θεωρητικό υπόβαθρο της παρούσας εργασίας, του προσδιορισμού θέσης κινούμενου σώματος στο χώρο.

## **1.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΚΙΝΗΣΗΣ**

Η κίνηση είναι γενική και χαρακτηριστική ιδιότητα της ύλης. Εκδηλώνεται σε κάθε σώμα από τα σωματίδια του μικροκόσμου μέχρι τους μακρινούς γαλαξίες. Κατά τη μελέτη της κίνησης, θεωρούμε τα αντικείμενα ως υλικά σημεία, δηλαδή σώματα που δεν έχουν διαστάσεις αλλά καταλαμβάνουν ένα σημείο του χώρου.

Ως κίνηση ορίζεται στη φυσική η μεταβολή της θέσης ή του προσανατολισμού ενός σώματος ως προς ένα δεδομένο σύστημα αναφοράς. Το σύστημα αναφοράς θεωρείται προσδεμένο σε ένα σώμα (ή σύστημα σωμάτων) το οποίο θεωρούμε κατά σύμβαση ακίνητο. Συνεπώς, μπορούμε καταχρηστικά να μιλάμε για την κίνηση ενός σώματος ως προς άλλο σώμα.

Οι κινήσεις μπορούν να χαρακτηριστούν ως εξής:

- Ευθύγραμμη κίνηση, ονομάζεται η κίνηση που γίνεται σε ευθεία γραμμή.
- Καμπυλόγραμμη κίνηση, ονομάζεται η κίνηση σε καμπύλη γραμμή.
- Κυκλική κίνηση, ονομάζεται η κίνηση στην περιφέρεια του κύκλου.

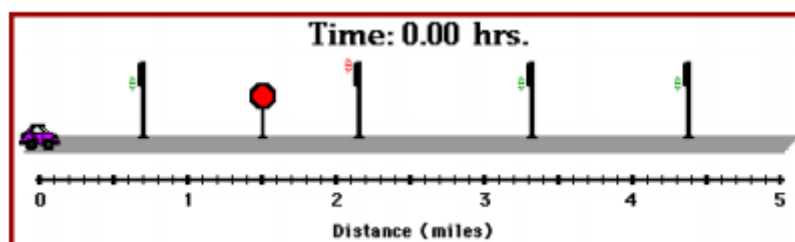
Προκειμένου να σχεδιάσουμε την κίνηση-τροχιά ενός κινητού, θα πρέπει να γνωρίζουμε τη θέση του κάθε χρονική στιγμή.

Διάστημα  $s$  ονομάζεται το μήκος της κίνησης-τροχιάς που διανύει το κινητό σε ορισμένο χρόνο  $t$ .

Για να μετρήσουμε το διάστημα και τον αντίστοιχο χρόνο της κινήσεως παίρνουμε αυθαίρετα κάποια θέση του κινητού, ως αρχή ( $s=0, t=0$ ) και τη λέμε αφετηρία ή σημείο αναφοράς.

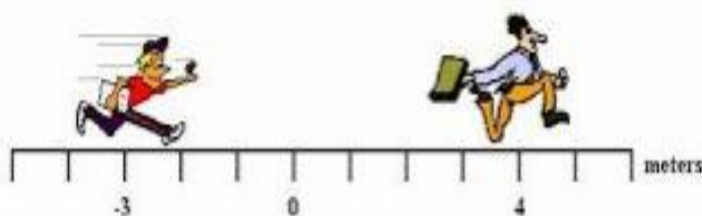
## **1.2Ο ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ ΘΕΣΗΣ ΕΝΟΣ ΣΩΜΑΤΟΣ**

Κλίμακα ονομάζεται μια ευθεία ή ένας άξονας αριθμημένος με θετικές και αρνητικές τιμές. Οι θετικές τιμές είναι προς τα δεξιά με αύξουσα σειρά και οι αρνητικές τιμές είναι προς τα αριστερά πάλι με αύξουσα σειρά (κατά απόλυτη τιμή). Το σημείο με την τιμή μηδέν είναι το σημείο αναφοράς.



**Εικόνα 1: Μια κλίμακα θέσεως.**

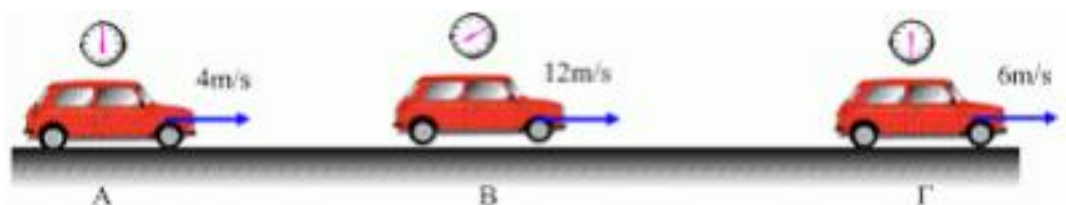
Θέση ονομάζεται ένα σημείο στο χώρο στο οποίο μπορεί να βρίσκεται ένα σώμα μια δεδομένη χρονική στιγμή  $t$ . Το σημείο αυτό μπορεί να ανήκει σε μια ευθεία, για παράδειγμα ο άξονας  $x$  ή ο άξονας  $y$ . Για να προσδιορίσουμε τη θέση ενός σώματος χρειαζόμαστε ένα σημείο αναφοράς, δηλαδή μια γνωστή αρχική θέση. Στο σημείο αναφοράς αντιστοιχούμε την ένδειξη μηδέν και γράφουμε  $x=0$ .



**Εικόνα 2: Απόσταση είναι το μήκος της συνολικής διαδρομής που διανύει ένα κινούμενο σώμα.**

### 1.3 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ ΘΕΣΗΣ ΣΩΜΑΤΙΟΥ ΣΕ ΕΥΘΕΙΑ ΓΡΑΜΜΗ

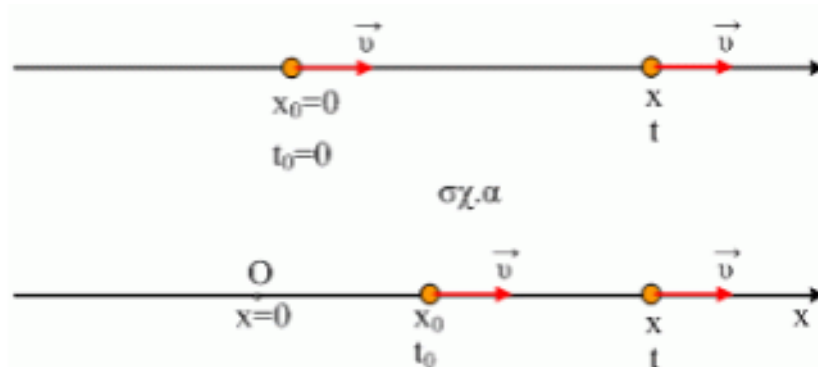
Στη φυσική, για να προσδιορίσουμε τη θέση ενός σωματίου, πρέπει να αναφερθούμε σε κάποιο σημείο, που το θεωρούμε ως σημείο αναφοράς. Απαιτείται ο ακριβής ποσοτικός προσδιορισμός της, που προκύπτει από μετρήσεις.



**Εικόνα 3: Ένα σωματίο κινείται σε ευθεία γραμμή.**

Έστω ότι ένα σωματίο βρίσκεται ή κινείται σε ευθεία γραμμή. Για να προσδιορίσουμε τη θέση του σωματίου πρέπει να ορίσουμε ένα σημείο αναφοράς ή αρχή, για τις μετρήσεις μας.

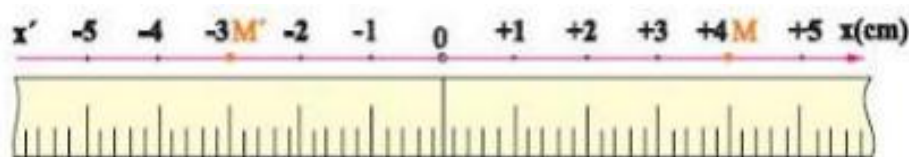




**Εικόνα 4: Η θέση ισορροπίας και πως κινείται το σώμα.**

Σημαντικό είναι ότι πρέπει να προσδιορίσουμε, αν το σωματίο κινείται δεξιά ή αριστερά, σε σχέση με την αρχή. Μπορούμε κατά σύμβαση να συμβολίσουμε το δεξιά με (+) και το αριστερά με (-).

Στην παρακάτω εικόνα φαίνεται η ευθεία πάνω στην οποία μπορεί να κινείται ένα σωματίο, όπου η κίνηση μπορεί να γίνεται δεξιά ή αριστερά του σημείου Ο. Τοποθετούμε πάνω στην ευθεία δυο μετροταινίες με την αρχή τους στο Ο, μια δεξιά του και μια αριστερά του. Οι δύο μετροταινίες μαζί με το σημείο Ο (αρχή), αποτελούν το σύστημα αναφοράς.



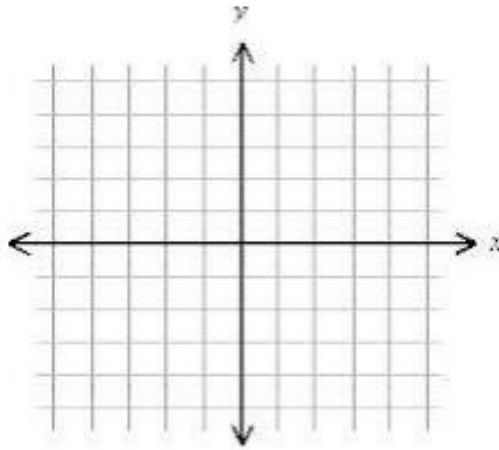
**Εικόνα 5: Ένα σύστημα αναφοράς σε ευθεία γραμμή.**

Για παράδειγμα, αν το σωματίο βρίσκεται στο σημείο Μ η θέση του θα είναι  $x=+4\text{cm}$ . Αντίστοιχα, αν το σωματίο βρίσκεται στο σημείο Μ' η θέση του θα είναι  $x'=-3\text{cm}$ .

Η θέση του σωματίου στο συγκεκριμένο σύστημα αναφοράς, προσδιορίζεται με έναν αριθμό, ο οποίος συμβολίζεται με το γράμμα  $x$  και ο οποίος μπορεί να πάρει θετικές ή αρνητικές τιμές.

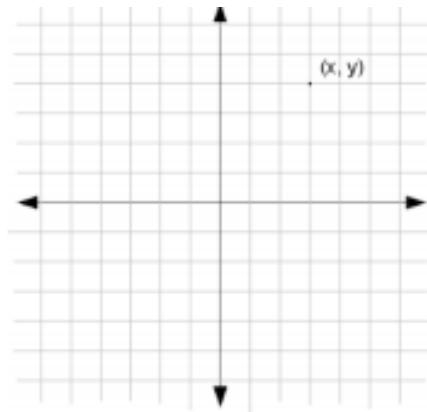
## 1.4 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΗΣ ΘΕΣΗΣ ΣΤΟ ΕΠΙΠΕΔΟ

Θεωρούμε ένα σωματίο, που βρίσκεται στο επίπεδο. Για να προσδιορίσουμε τη θέση του σωματίου χρειαζόμαστε δύο άξονες. Το σύστημα αναφοράς μας είναι ένα ορθογώνιο σύστημα συντεταγμένων. Στα Μαθηματικά αυτό λέγεται το σύστημα αυτό Καρτεσιανό.

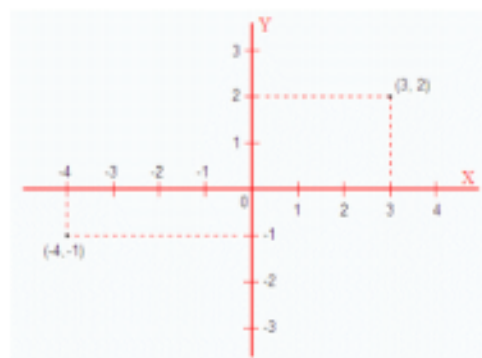


**Εικόνα 6: Ορθογώνιο σύστημα αξόνων.**

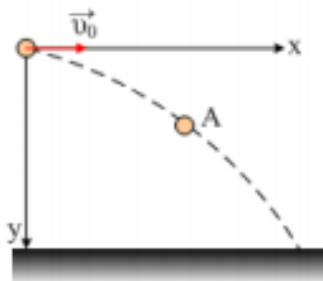
Η θέση του σωματίου προσδιορίζεται με δύο αριθμούς ( $x, y$ ) που ονομάζονται συντεταγμένες του σωματίου. Για να βρούμε τη θέση του σωματίου φέρνουμε από αυτό κάθετες πάνω στους άξονες  $x, y$ .



Για να βρούμε παραδείγματος χάρη, τη θέση των σημείων, φέρνουμε από αυτό κάθετες πάνω στους άξονες  $X, Y$ . Για το πρώτο σημείο τα ίχνη των καθέτων αυτών πάνω στους άξονες  $x, y$ , αντιστοιχούν στους αριθμούς 3 και 2. Το διατεταγμένο ζεύγος αριθμών  $(3, 2)$  αποτελεί τις συντεταγμένες του σημείου, και προσδιορίζει τη θέση του στο επίπεδο.

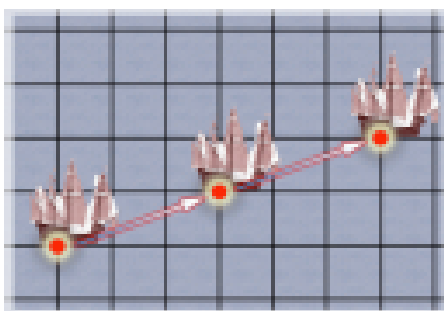


Για το δεύτερο σημείο τα ίχνη των καθέτων αυτών πάνω στους άξονες  $\chi$ ,  $\psi$ , αντιστοιχούν στους αριθμούς  $-4$  και  $-1$ . Το διατεταγμένο ζεύγος αριθμών  $(-4, -1)$  αποτελεί τις συντεταγμένες του σημείου, και προσδιορίζει τη θέση του στο επίπεδο.



**Εικόνα 7:** Ένα κινητό κινείται στο επίπεδο των ορθογωνίων αξόνων  $O\chi$  και  $O\psi$ .

Παρακάτω παρουσιάζεται η περίπτωση ενός κινητού. Ας υποθέσουμε ότι ένα κινητό κινείται στο επίπεδο των ορθογωνίων αξόνων  $O\chi$  και  $O\psi$ , όπως φαίνεται στην παραπάνω εικόνα. Η θέση του κινητού σε κάθε χρονική στιγμή καθορίζεται, όταν γνωρίζουμε τις αποστάσεις  $\chi$  και  $\psi$  από τους άξονες  $O\psi$  και  $O\chi$  αντίστοιχα. Οι αποστάσεις αυτές  $\chi$  και  $\psi$  λέγονται συντεταγμένες του κινητού και κατά την διάρκεια της κίνησης μεταβάλλονται με το χρόνο. Όταν ένα σώμα ηρεμεί οι συντεταγμένες του παραμένουν αμετάβλητες.



**Εικόνα 8:** Η θέση ενός πλοίου καθορίζονται με δυο συντεταγμένες.



**Εικόνα 9:** Η θέση ενός αεροπλάνου που καθορίζεται από τρεις συντεταγμένες.

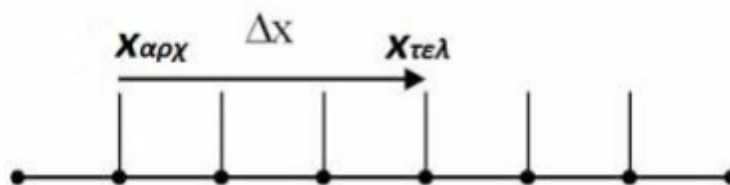
## 1.5Η ΜΕΤΑΤΟΠΙΣΗ ΣΩΜΑΤΙΟΥ ΣΕ ΑΞΟΝΑ

Θεωρούμε ένα σωματίο που κινείται στην ευθεία  $\chi'\chi$ . Υποθέτουμε ότι το σωματίο μετακινείται από ένα αρχικό σημείο  $M_1$ , το οποίο βρίσκεται στη θέση  $\chi_1$ , σε ένα άλλο σημείο  $M_2$ , το οποίο βρίσκεται στη θέση  $\chi_2$ .

$$\Delta\chi = \chi_2 - \chi_1$$

Την διαφορά  $\chi_2 - \chi_1$  την ορίζουμε ως μετατόπιση  $\Delta\chi$  του σωματίου πάνω στην ευθεία κίνησης.

Μετατόπιση  $\Delta\chi$  του σωματίου πάνω στην ευθεία κίνησης του, από ένα αρχικό σημείο το οποίο βρίσκεται στην αρχική θέση  $\chi_{\text{αρχ}}$ , σε ένα άλλο σημείο το οποίο βρίσκεται στη τελική θέση  $\chi_{\text{τελ}}$ , ονομάζεται η διαφορά  $\chi_{\text{τελ}} - \chi_{\text{αρχ}}$ .



Εικόνα 10: Πως ορίζεται η μετατόπιση.

Παρακάτω παρατίθενται μερικά παραδείγματα για την καλύτερη κατανόηση των παραπάνω ορισμών.

### Παράδειγμα 1:

Έστω ότι το σωματίο μετακινείται από ένα αρχικό σημείο  $M_1$ , το οποίο βρίσκεται στη θέση  $\chi_1 = 4\text{cm}$ , σε ένα άλλο σημείο  $M_2$ , το οποίο βρίσκεται στη θέση  $\chi_2 = 12\text{cm}$ . Η μετατόπιση  $\Delta\chi$  του σωματίου πάνω στην ευθεία κίνησής του είναι:

$$\Delta\chi = \chi_2 - \chi_1 = +12\text{cm} - 4\text{cm} = +8\text{cm}$$

### Παράδειγμα 2:

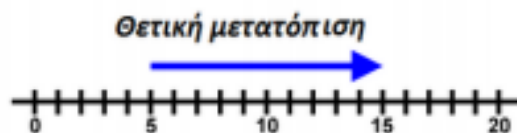
Αν υποθέσουμε ότι το σωματίο μετακινήθηκε από το σημείο  $M_1$  έως το σημείο  $M_3$ , του οποίου η θέση είναι  $\chi_3 = +1\text{cm}$ , τότε η μετατόπιση του θα είναι:

$$\Delta\chi' = \chi_3 - \chi_1 = +1\text{cm} - 4\text{cm} = -3\text{cm}$$

## ΣΥΜΠΕΡΑΣΜΑΤΑ

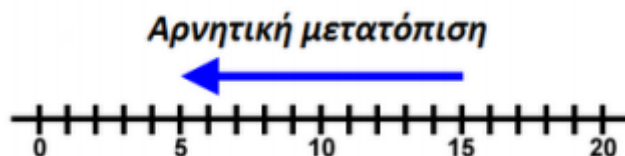
Το πρόσημο (+) στην πρώτη μετατόπιση σημαίνει ότι το σωματίο μετακινήθηκε προς τα δεξιά, ενώ το πρόσημο (-) στη δεύτερη μετατόπιση σημαίνει ότι το σωματίο κινήθηκε προς τα αριστερά. Άρα:

- 1) Το διάνυσμα της μετατόπισης είναι θετικό, όταν η τελική θέση του σώματος είναι δεξιότερα της αρχικής του θέσης.



Δηλαδή:  $x_{\text{τελ.}} > x_{\text{αρχ.}}$

- 2) Το διάνυσμα της μετατόπισης είναι αρνητικό, όταν η τελική θέση του σώματος είναι αριστερότερα της αρχικής του θέσης.



Δηλαδή:  $x_{\text{τελ.}} < x_{\text{αρχ.}}$

Η μετατόπιση είναι διάνυσμα που έχει αρχή την αρχική θέση του κινητού και τέλος την τελική του θέση.



Έτσι στην πρώτη περίπτωση η μετατόπιση  $\Delta x$  είναι το διάνυσμα με αρχή  $M_1$ , τέλος το σημείο  $M_2$  και αλγεβρική τιμή  $\Delta x = +8\text{cm}$ . Στην δεύτερη περίπτωση η μετατόπιση  $\Delta x'$  είναι το διάνυσμα που έχει αρχή το σημείο  $M_1$ , τέλος το σημείο  $M_3$  και αλγεβρική τιμή  $\Delta x' = -3\text{cm}$ .

Μπορούμε να καθορίσουμε τη θέση ενός κινητού με ένα διάνυσμα  $x$ , που έχει αρχή το σημείο αναφοράς ( $O$ ) και τέλος το σημείο  $M$  στο οποίο βρίσκεται το κινητό. Στην περίπτωση αυτή η μετατόπιση  $\Delta x$  του κινητού από μια θέση  $x_1$  μέχρι μια άλλη θέση  $x_2$  ορίζεται ως:

$$\Delta x = x_2 - x_1$$

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>

### 2.1 ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB

Το λογισμικό MATLAB, που παίρνει το όνομά του από τις λέξεις MATrix LABoratory, είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό πακέτο που χρησιμοποιείται εκτενώς στα πανεπιστήμια και στη βιομηχανία. Είναι ένα διαδραστικό (interactive) πρόγραμμα για αριθμητικούς υπολογισμούς και για κατασκευή γραφημάτων, αλλά παρέχει επίσης και τη δυνατότητα προγραμματισμού, κάτι που το καθιστά ένα χρησιμότερο εργαλείο για όλους όσους ασχολούνται με τις θετικές επιστήμες (και όχι μόνο). Σε αντίθεση με τα λογισμικά MAPLE και MATHEMATICA, το MATLAB στις αρχικές του εκδοχές δεν έκανε συμβολικούς υπολογισμούς. Στις νεότερες εκδοχές του, το πακέτο περιλαμβάνει εργαλειαθήκες που επιτρέπουν συμβολικούς υπολογισμούς.

Όπως υποδηλώνεται και από το όνομα του, το MATLAB είναι ειδικά σχεδιασμένο για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικού πίνακα κλπ. Επιπλέον το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη δική του γλώσσα προγραμματισμού για την επίλυση άλλων προβλημάτων όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων, η επίλυση προβλημάτων αρχικών τιμών με συνήθεις διαφορικές εξισώσεις κ.α. Η γλώσσα προγραμματισμού του MATLAB δίνει την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα. Συχνά θα λέμε η MATLAB (εννοώντας τη γλώσσα προγραμματισμού) και όχι το (πακέτο) MATLAB.

Το MATLAB είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε αριθμητική πεπερασμένης ακρίβειας (finite-precision arithmetic). Με άλλα λόγια, δεν βρίσκει την ακριβή λύση αλλά μια προσεγγιστική λύση ενός προβλήματος. Αυτή είναι και η βασική του διαφορά από τα συστήματα συμβολικών υπολογισμών όπως το Maple και το Mathematica.

### 2.2 ΠΕΡΙΒΑΛΛΟΝ MATLAB

Όπως αναφέραμε και παραπάνω, η κύρια χρήση του matlab είναι για την επίλυση των μαθηματικών προβλημάτων, ωστόσο είναι πολύ "ισχυρό" και μπορεί να χρησιμοποιηθεί και για προγραμματισμό καθώς περιέχει εντολές από την C++ όπως την while, την switch και την if.

Στον τομέα των γραφικών όσον αφορά τον μαθηματικό κλάδο μπορεί να υλοποιήσει συναρτήσεις πραγματικές, μιγαδικές, πεπλεγμένες συναρτήσεις δύο μεταβλητών και άλλες.

Όσον αφορά τον στατιστικό κλάδο μπορεί να υλοποιήσει ιστογράμματα, τομεογράμματα, ραβδοδιαγράμματα, εμβαδογράμματα και άλλα.

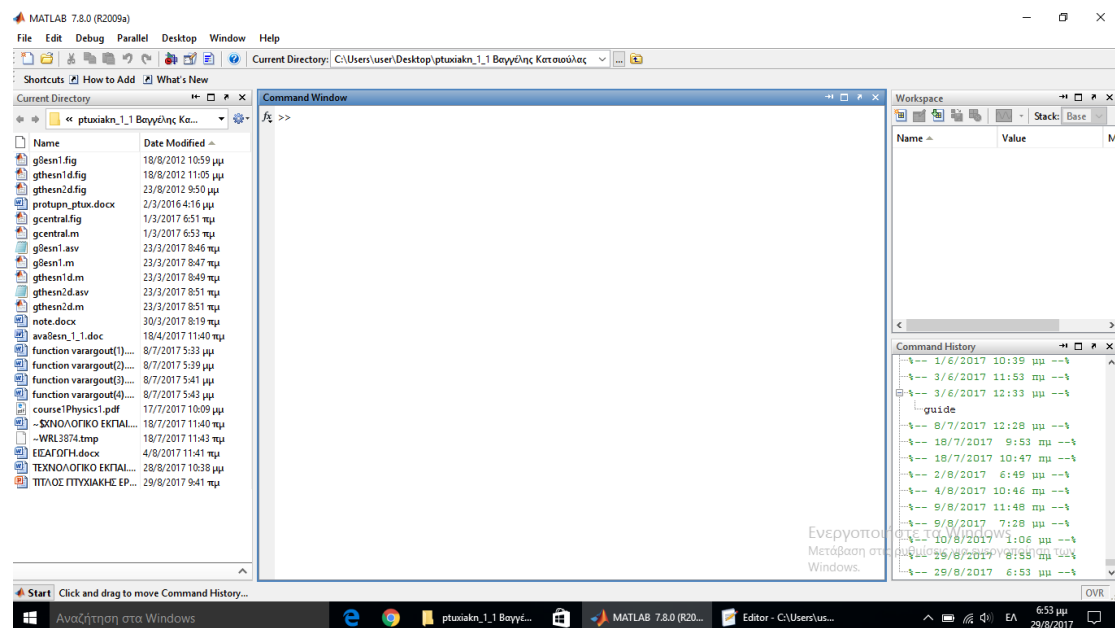
Οι εντολές δίνονται μέσω του παραθύρου εντολών (MATLAB command window). Με την εκκίνηση του προγράμματος, εμφανίζεται το παράθυρο διαχείρισης (Command Window) του MATLAB. Μέσα στο παράθυρο αυτό, γράφονται οι εντολές, δίνεται η εντολή εκτέλεσης τους, και εμφανίζονται τα αποτελέσματα των πράξεων, δηλαδή γίνονται όλες οι αλληλεπιδράσεις του χρήστη με το πρόγραμμα.

Οι εντολές αυτές μπορεί να είναι:

1. Ορισμοί μεταβλητών και πράξεις.
2. Κλήση ενσωματωμένων συναρτήσεων της MATLAB και των εγκατεστημένων εργαλειοθηκών της (toolboxes).
3. Κλήση συναρτήσεων (functions) ή αρχείων εντολών MATLAB (scripts) που κατασκευάζονται από τους χρήστες με τη μορφή m-file.

Επίσης, υπάρχει και ο βασικός χώρος λειτουργίας (Workspace) του MATLAB. Ο χώρος αυτός είναι για την διεκπεραίωση των πράξεων και είναι ο βασικός χώρος λειτουργίας του προγράμματος (Workspace). Ο χώρος αυτός βρίσκεται στη μνήμη του υπολογιστή, και μέσα σ' αυτόν εκτελούνται οι πράξεις και ταυτόχρονα διατηρούνται όλα τα αποτελέσματα τους, με σκοπό να μπορούν να χρησιμοποιηθούν σε επόμενες πράξεις.

Παρακάτω παρουσιάζεται το περιβάλλον του matlab που περιγράφηκε παραπάνω και κάποια άλλα στοιχεία περιγράφονται παρακάτω.



**Εικόνα 2.1: Περιβάλλον του MATLAB.**

Στη εικόνα 2.1 παρατηρούμε το παράθυρο του Matlab. Το παράθυρο αυτό είναι χωρισμένο σε τρεις στήλες. Στην αριστερή στήλη βρίσκεται ένα toolbox με ονομασία Current Folder. Στην περιοχή αυτή βρίσκουμε τα αρχεία που θέλουμε να τρέξουμε.

Ακριβώς από κάτω βρίσκεται μία περιοχή που μας εμφανίζει τις συναρτήσεις που υπάρχουν σε κάθε αρχείο που έχουμε επιλέξει στο Current Folder (αν υπάρχουν στο αρχείο αυτό).

Στην δεξιά στήλη βρίσκονται δύο περιοχές. Το Workspace και το Command History. Στην πρώτη αποθηκεύονται οι μεταβλητές που έχουμε δημιουργήσει μέσω των εντολών καθώς και οι τιμές τους. Στην δεύτερη περιοχή μπορούμε να δούμε τις εντολές που έχουν τρέξει μέχρι στιγμής.

Τέλος στην μεσαία στήλη είναι το Command Window στο οποίο μπορούμε να δώσουμε τις εντολές μας. Τις εντολές τις δίνουμε είτε απευθείας γράφοντάς τις είτε φτιάχνοντας ένα script αρχείο και τρέχοντάς το. Το αρχείο αυτό το αποθηκεύουμε με την μορφή .m .

## 2.3 Τι είναι το GUIDE

Το MATLAB προσφέρει στον χρήστη τη δυνατότητα να κατασκευάσει δικές του γραφικές διεπιφάνειες, (Graphical User Interfaces). Η χρησιμότητα αυτής της λειτουργίας είναι μεγάλη, επειδή τα προγράμματα – εφαρμογές που περιέχουν γραφική διεπιφάνεια γίνονται πιο φιλικές στον τελικό χρήστη.



Το MATLAB προσφέρει μια ικανοποιητική εργαλειοθήκη , η οποία διευκολύνει πολύ τη δημιουργία μιας γραφικής διεπιφάνειας χρήστη . Αυτή η εργαλειοθήκη ή αλλιώς GUIDE, περιέχει μια πληθώρα χρήσιμων εργαλείων ελέγχου όπως κουμπιά, πλαίσια κ.α.

Σημαντικό όμως ρόλο παίζει και η αρχική σωστή σχεδίαση της διεπιφάνειας χρήστη, έτσι ώστε να δημιουργηθεί ένα απλό και λειτουργικό GUI, χωρίς περιττά στοιχεία που θα περιπλέκουν τον απλό χρήστη.

Η ανάπτυξη μιας εφαρμογής σε GUI χωρίζεται σε τέσσερα στάδια:

- 1. Θεωρητική σχεδίαση της εφαρμογής πριν την υλοποίηση της σε MATLAB.**
- 2. Σχεδίαση της εφαρμογής στο κατάλληλο περιβάλλον (GUIDE Layout Editor) και δημιουργία των απαιτούμενων αντικειμένων.**
- 3. Καθορισμός των ιδιοτήτων του GUI και του κάθε αντικειμένου.**
- 4. Προγραμματισμός του κάθε αντικειμένου, εάν χρειάζεται.**

Μία Γραφική Διεπιφάνεια Χρήστη αποτελείται συνήθως από διάφορα παράθυρα τα οποία περιέχουν ποικίλα στοιχεία ελέγχου όπως πεδία κειμένου, γραμμές κύλισης κ.α. Τα παράθυρα αυτά είναι δυνατόν να καλούν το ένα το άλλο, να δέχονται δεδομένα από τον χρήστη, να μεταβιβάζουν πιθανώς τα δεδομένα από το ένα παράθυρο στο άλλο, και γενικά να επιτελούν διάφορες λειτουργίες. Το GUIDE για να το πετύχει αυτό δημιουργεί για κάθε νέο παράθυρο, δύο αρχεία. Τα αρχεία αυτά είναι το FIG- αρχείο και το m- αρχείο .

- Το FIG-αρχείο , ουσιαστικά είναι το παράθυρο – figure, όπου το MATLAB αποθηκεύει τα στοιχεία ελέγχου και την ακριβή τους θέση. Εδώ ο προγραμματιστής σχεδιάζει την εμφάνισή του παραθύρου.
- Το m-αρχείο , όπου ο προγραμματιστής πρέπει να γράψει τον κώδικα που θα ενσωματωθεί στα στοιχεία ελέγχου (π.χ. κουμπιά ή φόρμες εισαγωγής δεδομένων ), ώστε αυτά να επιτελέσουν τις επιθυμητές λειτουργίες .

Σημειώνεται ότι για κάθε αρχείο \*.fig πρέπει να συνοδεύεται από το αντίστοιχο αρχείο \*.m, με το ίδιο όνομα. Αν για κάποιο λόγο το αρχείο \*.m χαθεί ή καταστραφεί τότε το παράθυρο \*.fig δεν θα είναι λειτουργικό. Κάθε φορά που κάποιος χρήστης δημιουργεί ένα νέο παράθυρο figure, το GUIDE δημιουργεί αυτομάτως και τους δύο τύπους αρχείων \*.fig και \*.m .

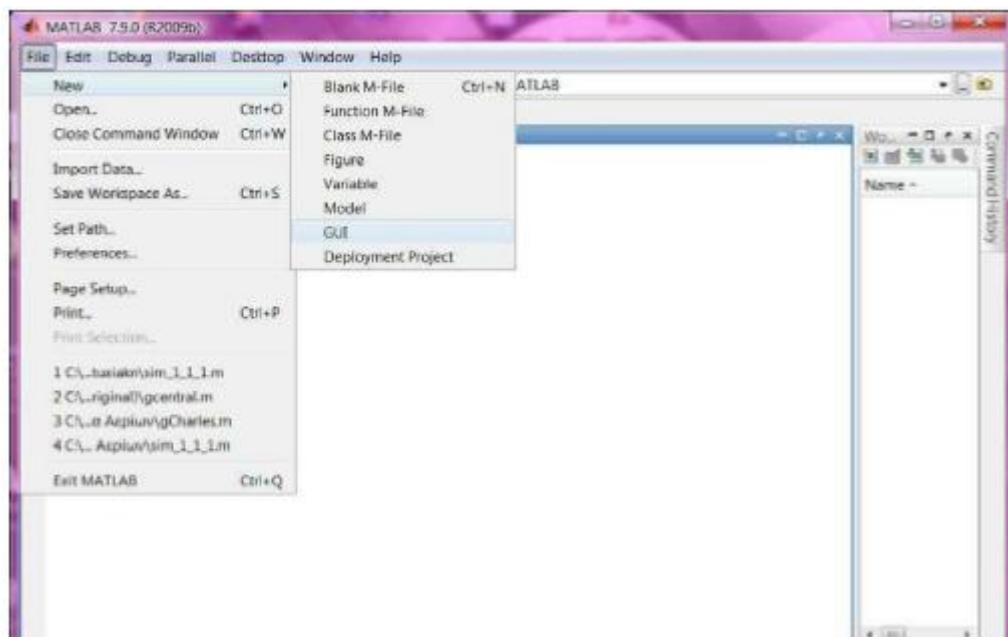
## 2.4 Δημιουργία ενός παραθύρου GUI

Η εκκίνηση του GUIDE γίνεται εύκολα με δύο τρόπους.

Πρώτον, με την κλίση της ομώνυμης συνάρτησης από την γραμμή εντολών του MATLAB .

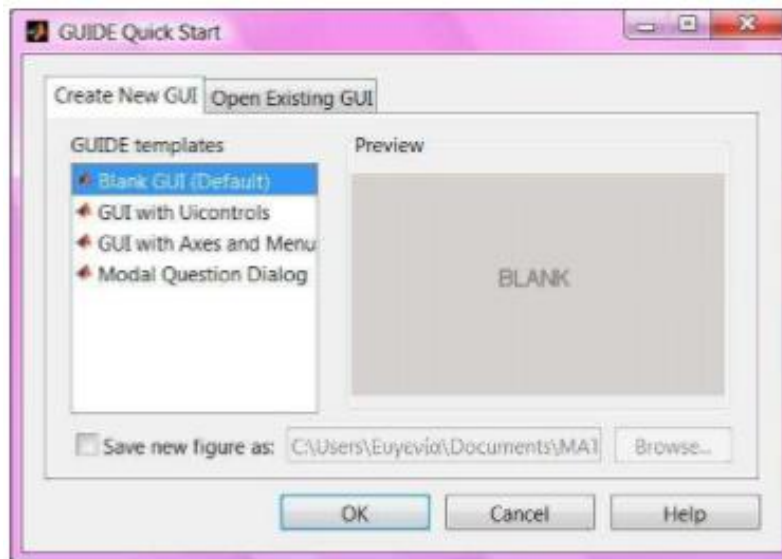
**>> guide**

Δεύτερον, επιλέγοντας από τη γραμμή μενού File→New→GUI (εικόνα 2.2).



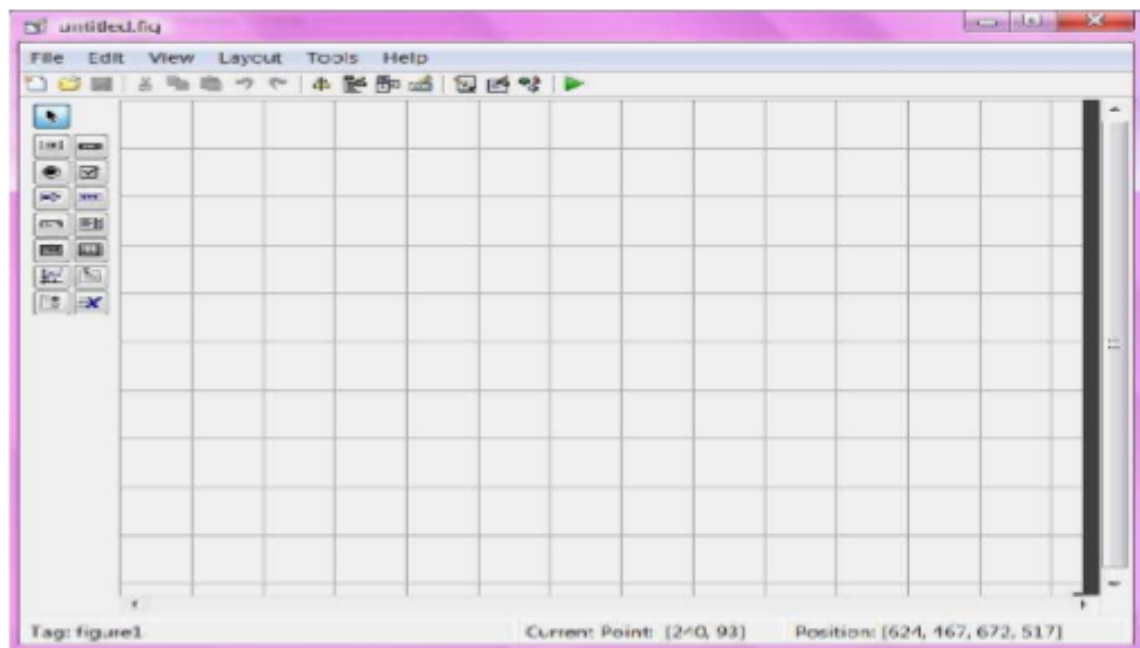
**Εικόνα 2.2: Η εκκίνηση του guide.**

Όποιον τρόπο και να επιλέξει ο χρήστης θα εμφανιστεί ένας οδηγός που τον καθοδηγεί στη δημιουργία ενός παραθύρου (εικόνα 2.3). Ο οδηγός ρωτάει τον χρήστη εάν θέλει να δημιουργήσει ένα νέο κενό παράθυρο (Blank GUI), αν θέλει να δημιουργήσει ένα κενό παράθυρο βασισμένο σε κάποια πρότυπα (π.χ. με άξονες και μενού ) ή αν θέλει να ανοίξει ένα έτοιμο παράθυρο.



**Εικόνα 2.3**

Αν ο χρήστης αφήσει την προεπιλεγμένη επιλογή και πατήσει OK , τότε θα δημιουργηθεί ένα νέο άδειο παράθυρο, όπως στην εικόνα 2.4.



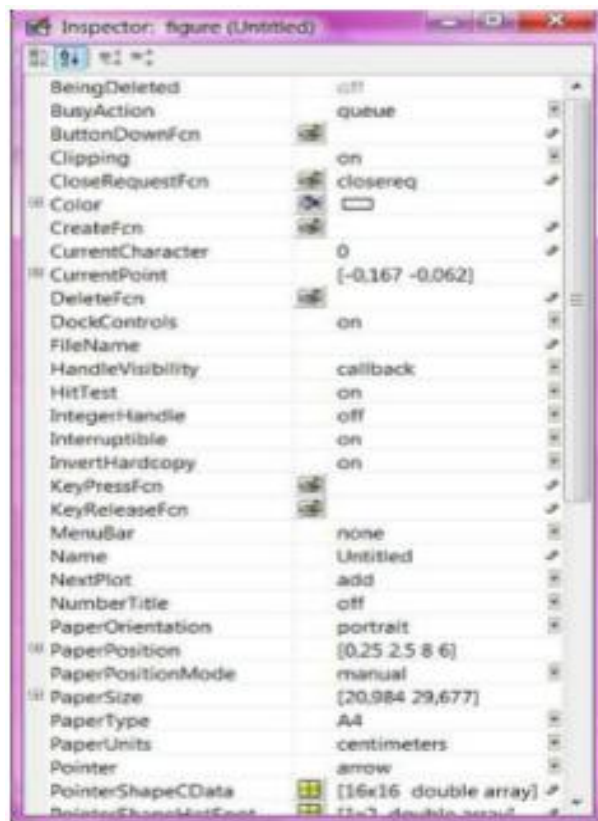
**Εικόνα 2.4**

Το περιβάλλον δημιουργίας του παραθύρου αποτελείται από μια κεντρική γραμμή επιλογών, μία γραμμή εργαλείων, καθώς και μια κάθετη εργαλειοθήκη στα αριστερά. Η εργαλειοθήκη περιέχει όλα τα στοιχεία ελέγχου , τα οποία είναι διαθέσιμα στον χρήστη. Η εισαγωγή τους στο παράθυρο είναι απλή και γίνεται με απλό drag and drop. Η γκριζα περιοχή με το πλέγμα το οποίο καλύπτει το παράθυρο, είναι το φόντο, η ταπετσαρία του παραθύρου. Ιδιαίτερης σημασίας είναι τα τέσσερα τελευταία κουμπιά της γραμμής εργαλείων, που φαίνονται στην εικόνα 2.5.



**Εικόνα 2.5**

Το πρώτο εικονίδιο από αριστερά εκκινεί τον γνωστό M-File Editor, ανοίγοντας ταυτόχρονα το αντίστοιχο m-αρχείο (m-file) του παραθύρου μας. Το δεύτερο κουμπί, εκκινεί τον Property Inspector (εικόνα 2.6), από όπου αλλάζουμε όποιες ιδιότητες επιθυμούμε είτε έχοντας επιλέξει πρώτα κάποιο στοιχείο ελέγχου, ή και το ίδιο το παράθυρο της εφαρμογής μας. Το τρίτο κουμπί της εικόνας ανοίγει ένα παράθυρο που ονομάζεται Object Browser και μας δείχνει πόσα και ποια αντικείμενα υπάρχουν στο συγκεκριμένο παράθυρο της εφαρμογής μας. Τέλος το τέταρτο κατά σειρά Run Figure κουμπί εκτελεί την εφαρμογή μας αλλά ταυτόχρονα μας ενημερώνει ότι πρέπει να αποθηκευτεί προηγουμένως.

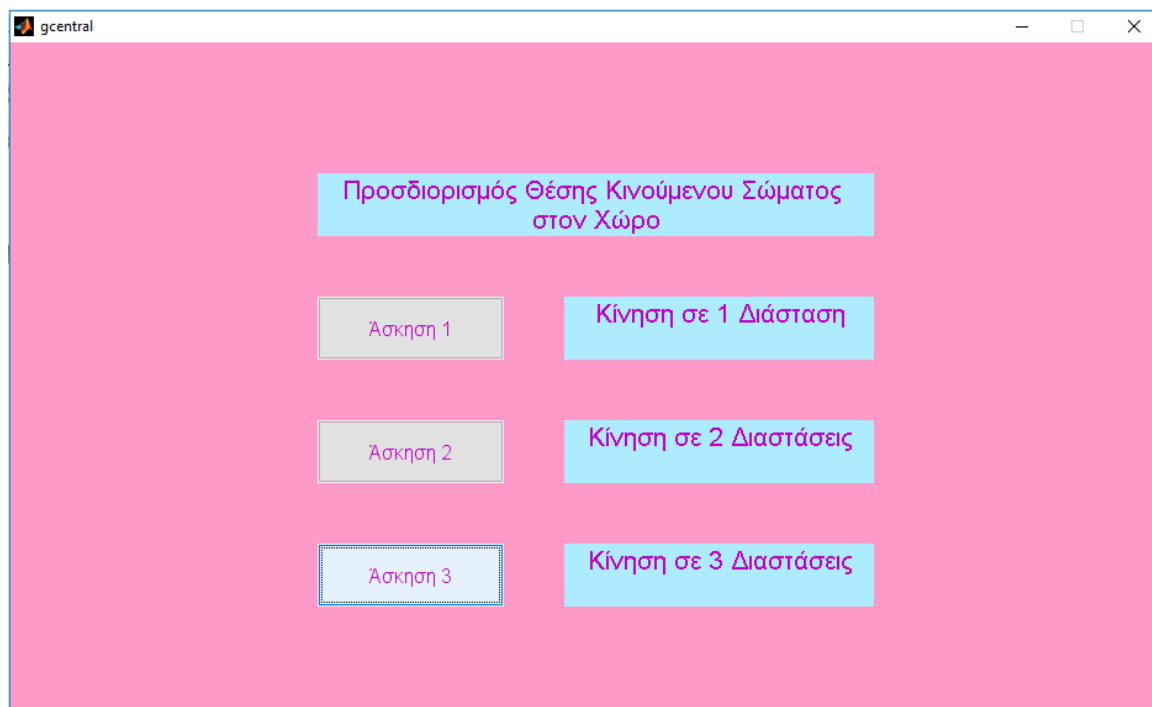


**Εικόνα 2.6**

## 2.5 ΠΑΡΟΥΣΙΑΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ MATLAB

Σε αυτήν την ενότητα θα παρουσιαστεί το πώς λειτουργεί ο κώδικας και ποιες είναι οι ασκήσεις που μπορούμε να πάρουμε ως αποτελέσματα για τον προσδιορισμό θέσης σώματος σε μια ευθεία γραμμή, προσδιορισμός θέσεις σε δύο διαστάσεις και προσδιορισμός θέσης στο χώρο.

Ανοίγοντας το matlab τρέχουμε το κώδικα matlabM-Filegcentral. Μας εμφανίζεται το παρακάτω παράθυρο.



**Εικόνα 2.7: Το παράθυρο με τις ασκήσεις για τον προσδιορισμό θέσεων.**

Σε αυτό το παράθυρο ανάλογα με ποια άσκηση θέλουμε ασχοληθούμε είτε για την κίνηση σε 1 διάσταση είτε για την κίνηση σε 2 διαστάσεις ακόμη και για κίνηση 3 διαστάσεων.

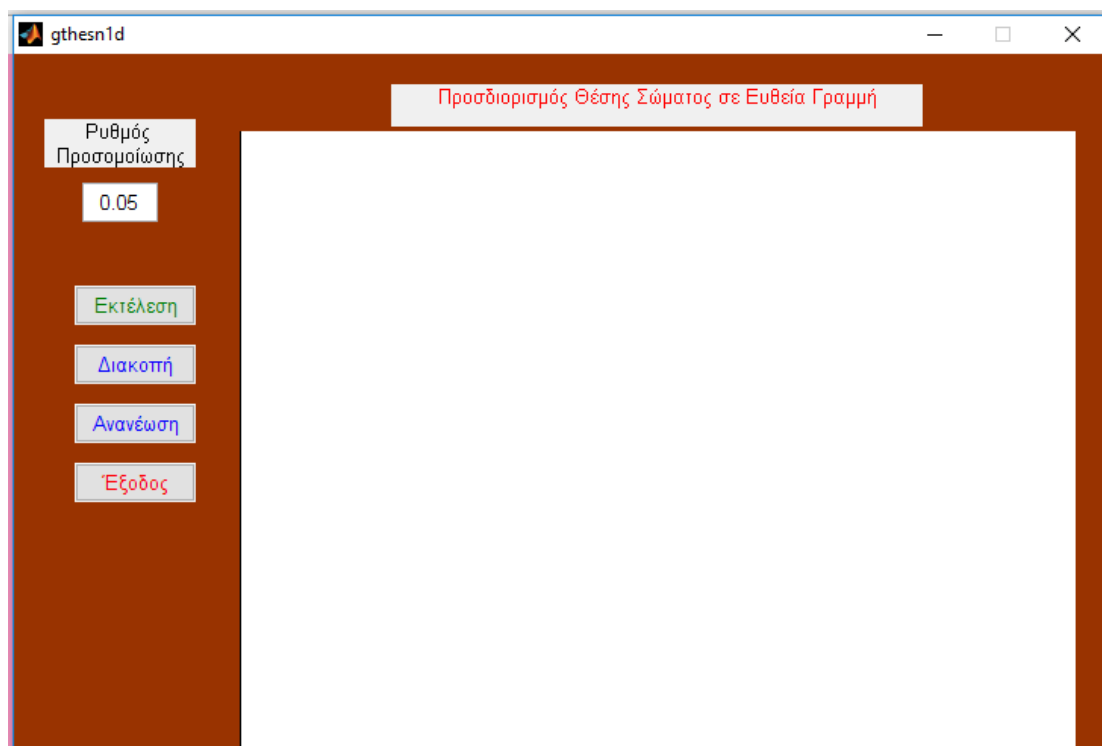
Το παράθυρο αυτό είναι ένα GUI παράθυρο (.fig αρχείο) το οποίο αποτελείται από τέσσερα text boxes και από τρία κουμπιά. Αφού κατασκευάστηκε το παράθυρο, αποθηκεύτηκε με όνομα gcentral.fig.

Στην συνέχεια όπως υπάρχει σε επόμενο κεφάλαιο, θα παρατηρηθεί ότι το όνομα του αρχείου αυτού καλείται από ένα Matlab (.m) αρχείο για να αξιοποιηθούν τα περιεχόμενά του.

Κλειδιά σημεία του αρχείου αυτού είναι τα ονόματα που έχουν το κάθε κουμπί έτσι ώστε να κλιθούν από το προηγούμενο αρχείο.

### 2.5.1 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ 1<sup>η</sup> ΑΣΚΗΣΗΣ

Αφού έχει τρέξει το αρχείο που αναφέραμε παραπάνω, αν ο χρήστης πατήσει στο κουμπί Άσκηση 1, τότε φορτώνεται το δεύτερο GUI παράθυρο το οποίο φαίνεται στην εικόνα 2.8.

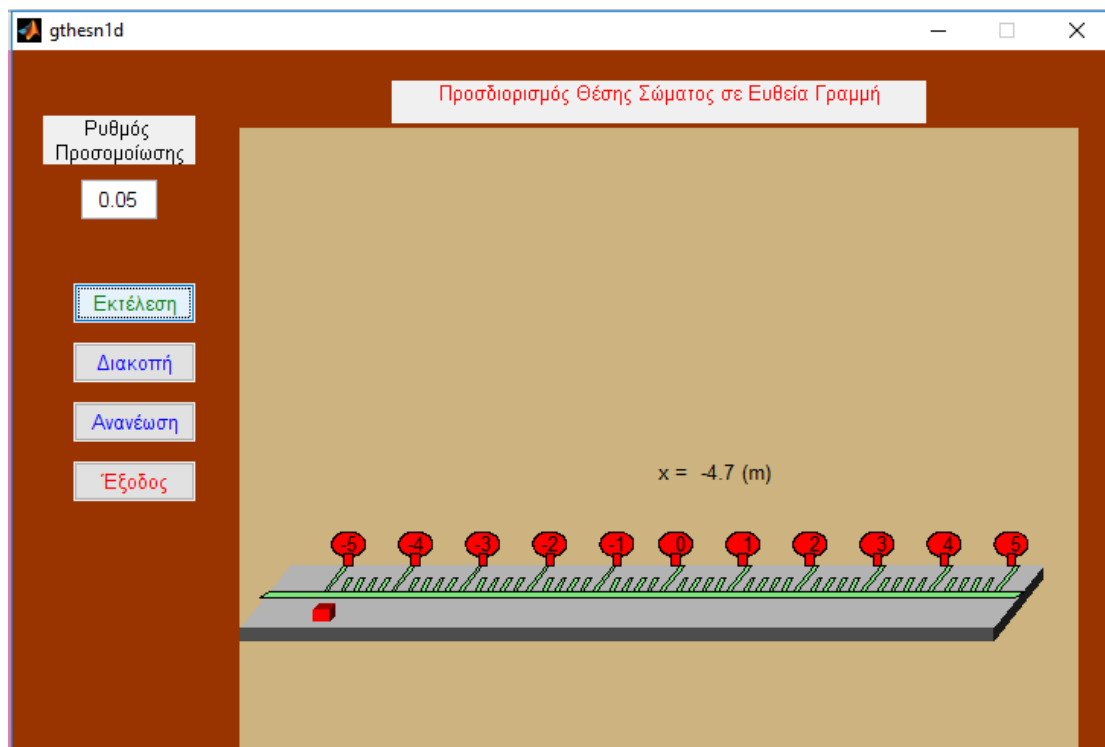


**Εικόνα 2.8:** Το περιβάλλον της 1<sup>η</sup> άσκησης.

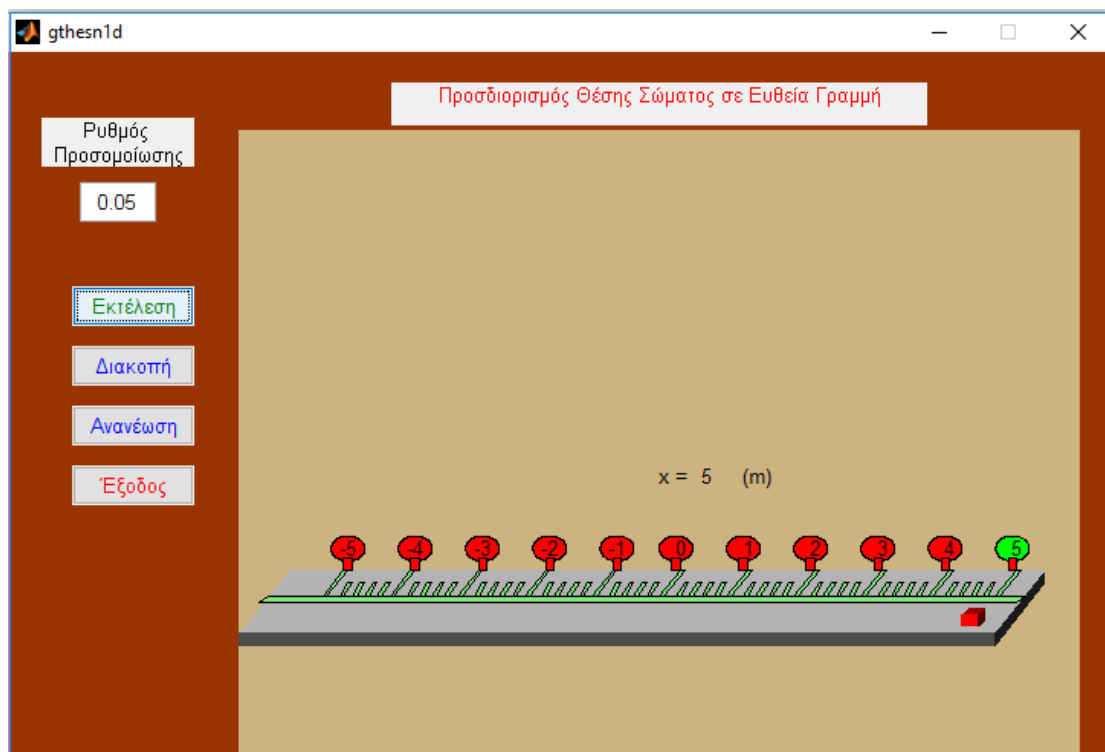
Το παράθυρο αυτό αποτελείται από ένα κουτάκι στο οποίο γράφεται ο ρυθμός προσομοίωσης με τον οποίο τρέχει το παράδειγμα και τέσσερα κουμπιά τα οποία κάνουν τις εξής διεργασίες:

- Εκτέλεση του παραδείγματος (Εκτέλεση)
- Διακοπή του παραδείγματος (Διακοπή)
- Επαναφορά του παραδείγματος στην αρχική κατάσταση (Ανανέωση)
- Έξοδος από το παράδειγμα (Έξοδος)

Μόλις ο χρήστης πατήσει το κουμπί έναρξη τότε το παράδειγμα αρχίζει και εμφανίζεται στο σημείο των αξόνων (το λευκό κομμάτι στο σχήμα 2.8) το παράθυρο της εικόνας 2.9.



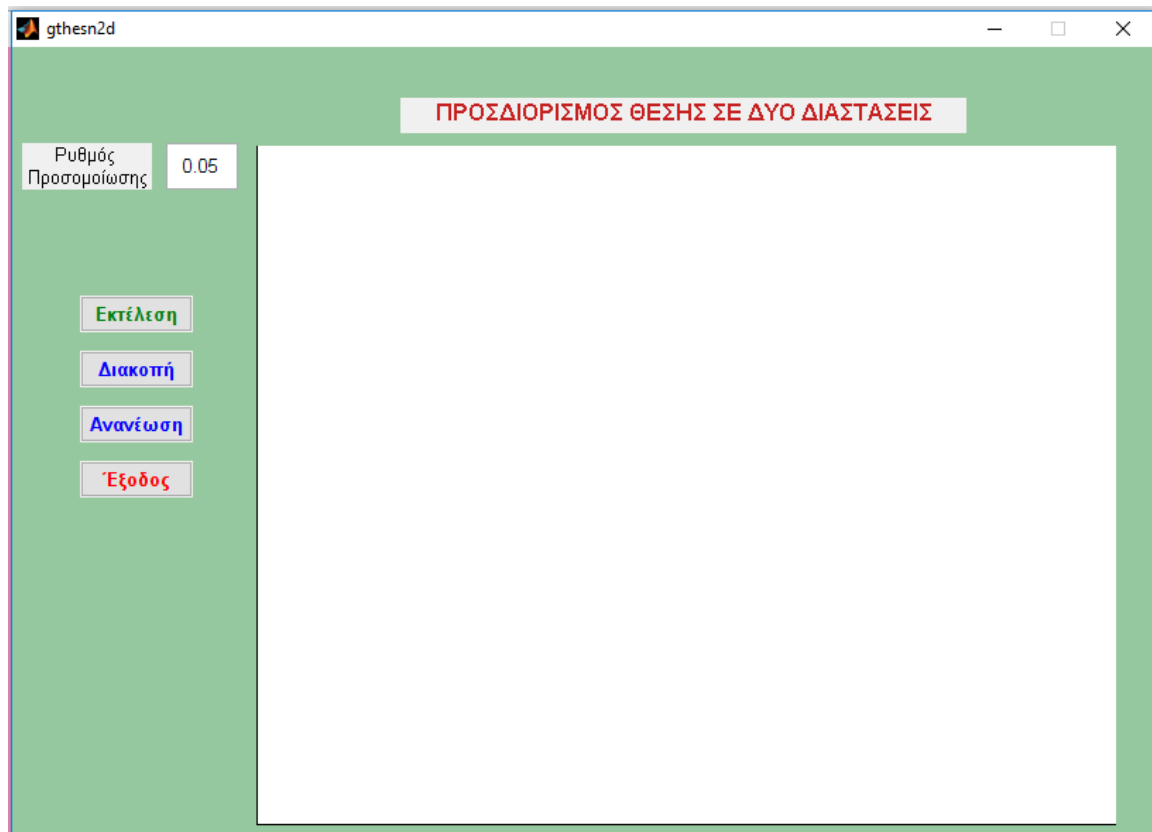
**Εικόνα 2.9: Η έναρξη της άσκησης.**



**Εικόνα 2.10: Η λήξη της άσκησης.**

## 2.5.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ 2<sup>η</sup>ΑΣΚΗΣΗΣ

Αφού έχει τρέξει το αρχείο που αναφέραμε παραπάνω, αν ο χρήστης πατήσει στο κουμπί Άσκηση 2, τότε φορτώνεται το δεύτερο GUI παράθυρο το οποίο φαίνεται στην εικόνα 2.11.



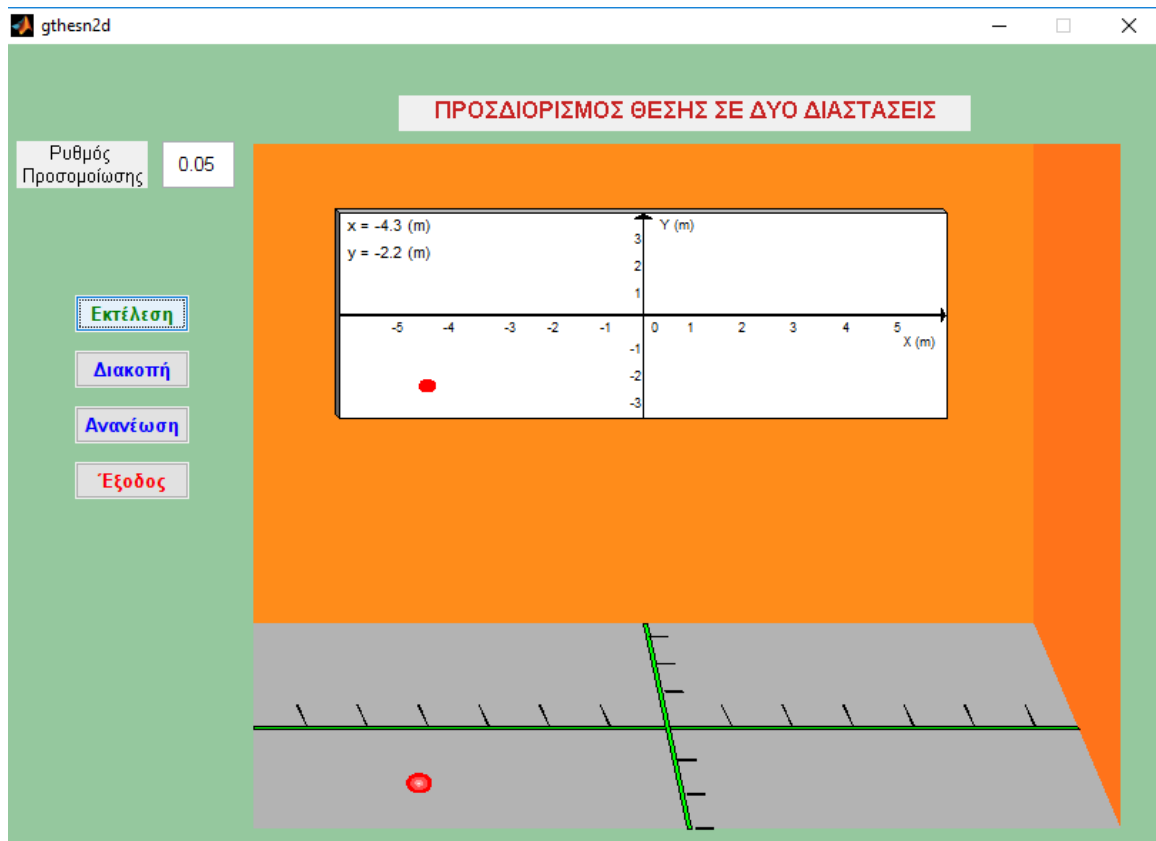
**Εικόνα 2.11: Το περιβάλλον της 2<sup>η</sup> άσκησης.**

Το παράθυρο αυτό αποτελείται από ένα κουτάκι στο οποίο γράφεται ο ρυθμός προσομοίωσης με τον οποίο τρέχει το παράδειγμα και τέσσερα κουμπιά τα οποία κάνουν τις εξής διεργασίες:

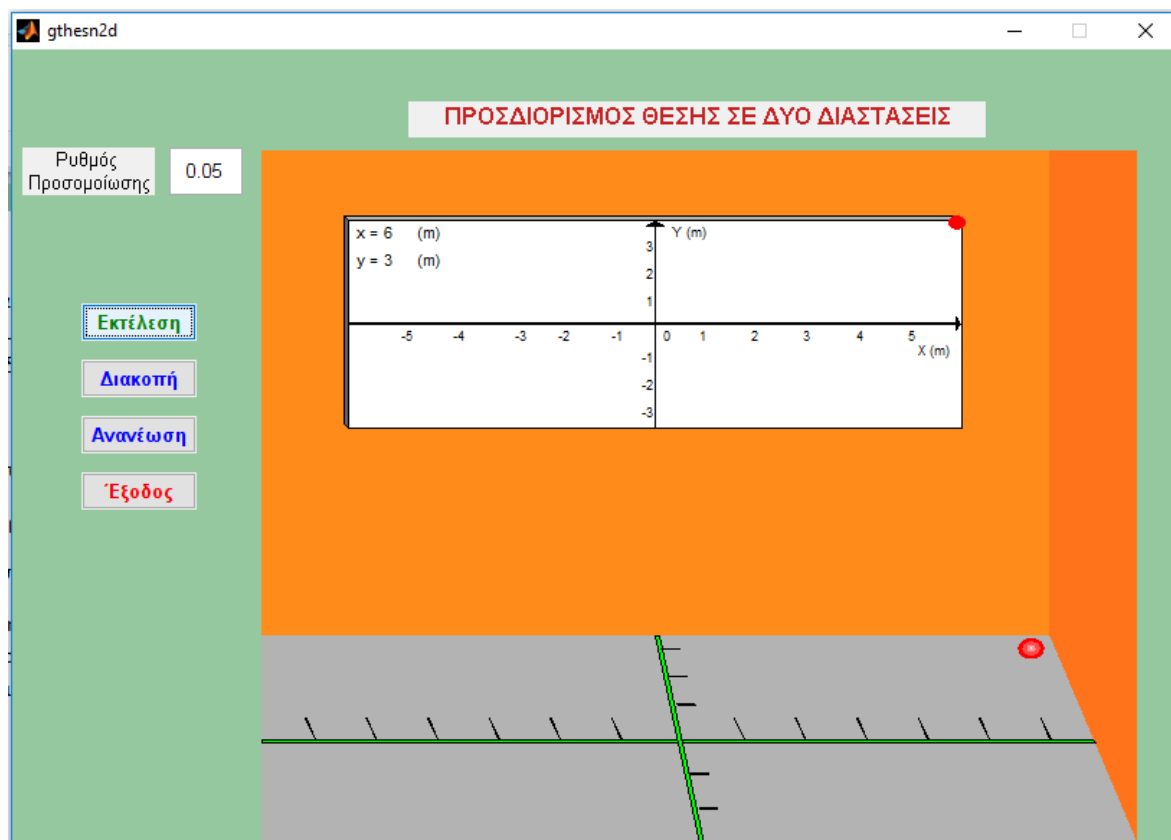
- Εκτέλεση του παραδείγματος (Εκτέλεση)
- Διακοπή του παραδείγματος (Διακοπή)
- Επαναφορά του παραδείγματος στην αρχική κατάσταση (Ανανέωση)
- Έξοδος από το παράδειγμα (Έξοδος)

Μόλις ο χρήστης πατήσει το κουμπί έναρξη τότε το παράδειγμα αρχίζει και εμφανίζεται στο σημείο των αξόνων (το λευκό κομμάτι στο σχήμα 2.11) το παράθυρο της εικόνας 2.12.





Εικόνα 2.12: Η έναρξη της άσκησης.



Εικόνα 2.13: Η λήξη της άσκησης.

### 2.5.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ 3<sup>η</sup>ΑΣΚΗΣΗΣ

Αφού έχει τρέξει το αρχείο που αναφέραμε παραπάνω, αν ο χρήστης πατήσει στο κουμπί Άσκηση 3, τότε φορτώνεται το δεύτερο GUI παράθυρο το οποίο φαίνεται στην εικόνα 2.14.

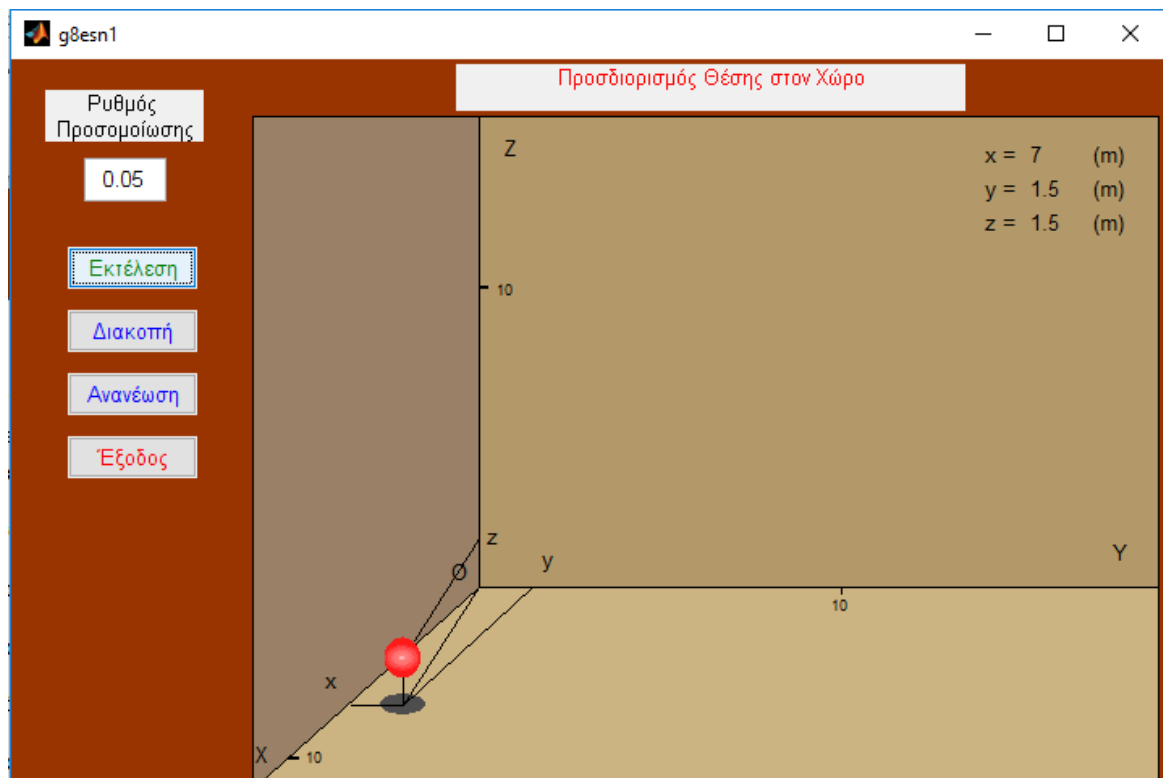


**Εικόνα 2.14:** Το περιβάλλον της 3<sup>η</sup> άσκησης.

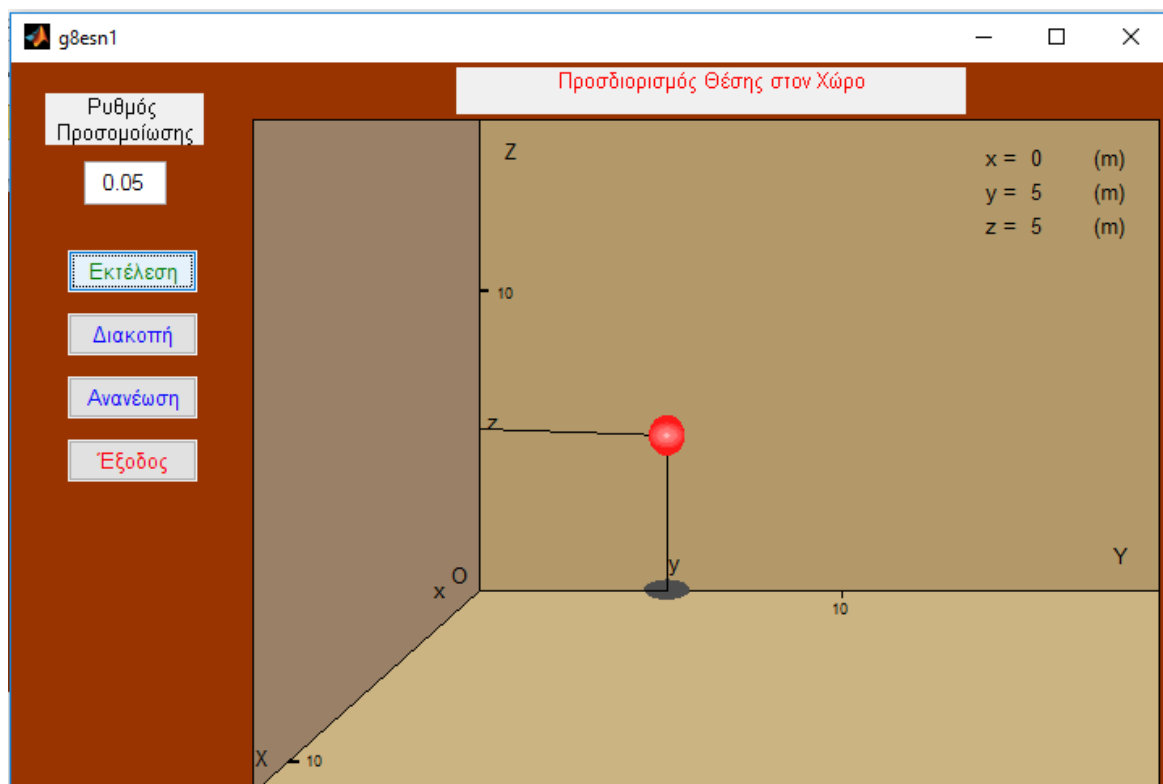
Το παράθυρο αυτό αποτελείται από ένα κουτάκι στο οποίο γράφεται ο ρυθμός προσομοίωσης με τον οποίο τρέχει το παράδειγμα και τέσσερα κουμπιά τα οποία κάνουν τις εξής διεργασίες:

- Εκτέλεση του παραδείγματος (Εκτέλεση)
- Διακοπή του παραδείγματος (Διακοπή)
- Επαναφορά του παραδείγματος στην αρχική κατάσταση (Ανανέωση)
- Έξοδος από το παράδειγμα (Έξοδος)

Μόλις ο χρήστης πατήσει το κουμπί έναρξη τότε το παράδειγμα αρχίζει και εμφανίζεται στο σημείο των αξόνων (το λευκό κομμάτι στο σχήμα 2.11) το παράθυρο της εικόνας 2.15.



Εικόνα 2.15: Η έναρξη της άσκησης.



Εικόνα 2.16: Η λήξη της άσκησης.

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>

### 3.1 ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

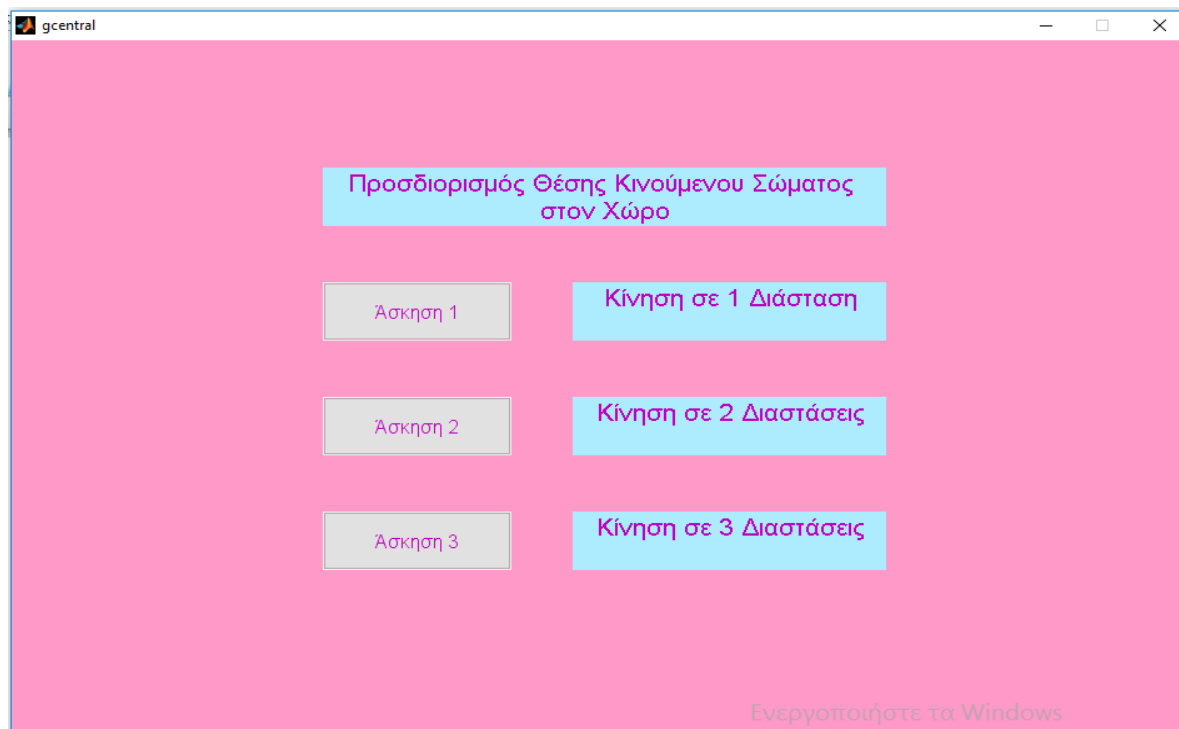
Η προσομοίωση αποτελείται από τέσσερα κυρίως m-files τα οποία είναι αλληλένδετα μεταξύ τους. Ως κεντρικό m-file αποτελεί αυτό με το όνομα gcentral, μέσω του οποίου όπως θα δείξουμε στη συνέχεια τρέχουν και τα υπόλοιπα m-files της προσομοίωσης, το κάθε ένα από τα οποία περιγράφει γραφικά τον προσδιορισμό θέσης κινούμενου σώματος στον χώρο, κίνηση σε μια διάσταση, κίνηση σε δύο διαστάσεις και κίνηση σε τρεις διαστάσεις.

Να αναφέρουμε εδώ, πως κάθε φορά που εκκινούμε το GUI και δημιουργούμε ένα παράθυρο figure, αυτόματα το MATLAB εκκινεί τον m-file editor και δημιουργεί το αντίστοιχο m-file , όπου δημιουργούνται οι εξής συναρτήσεις:

- **function varargout = name(varargin)** η οποία είναι απαραίτητη ώστε να μπορεί ο χρήστης να καλεί το παράθυρο από τη γραμμή εντολών πληκτρολογώντας το όνομα του (>>name)
- **function name\_OpeningFcn(hObject, eventdata, handles, varargin)** ο κώδικας της εκτελείται ακριβώς πριν εκκινήσει το παράθυρο
- **function varargout = name\_OutputFcn(hObject, eventdata, handles)** οι εντολές της εξάγουν τα αποτελέσματα τους στην γραμμή εντολών του MATLAB.

#### 3.1.1 Το m-file gcentral

Για να γίνει εκκίνηση της προσομοίωσης, τρέχουμε το m-file με όνομα gcentral. Έτσι εμφανίζεται το περιβάλλον της εικόνας 3.1 (gcentral-figure), που αποτελεί το κεντρικό περιβάλλον – μενού της προσομοίωσης.



**Εικόνα 3.1 (gcentral figure)**

Τα στοιχεία του gcentral.fig αποτελούν, ένα static text με περιεχόμενο (string) “Προσδιορισμός Θέσης Κινούμενου Σώματος στον Χώρο”, τρία PushButton, δηλαδή κουμπιά, με περιεχόμενο “Άσκηση 1 –Κίνηση σε 1 Διάσταση”, “Άσκηση 2- Κίνηση σε 2 Διαστάσεις” και “Άσκηση 3-Κίνηση σε 3 Διαστάσεις”.

Ο κώδικας που συμπεριλαμβάνεται στο m-file gcentral περιγράφεται και εξηγείται παρακάτω:

```
function varargout = gcentral(varargin)
% GCENTRAL M-file for gcentral.fig
%     GCENTRAL, by itself, creates a new GCENTRAL or raises the
existing
%     singleton*.
%
%     H = GCENTRAL returns the handle to a new GCENTRAL or the
handle to
%     the existing singleton*.
%
%     GCENTRAL('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in GCENTRAL.M with the given input
arguments.
%
%     GCENTRAL('Property','Value',...) creates a new GCENTRAL or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before gcentral_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
```

```
%      stop. All inputs are passed to gcentral_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gcentral

% Last Modified by GUIDE v2.5 01-Mar-2017 05:49:36
```

**«Οι συναρτήσεις `functionvarargout = gcentral(varargin), functiongcentral_OpeningFcn`, και `functionvarargout = gcentral_OutputFcn` δημιουργούνται αυτόματα από το MATLAB με το που δημιουργήσουμε ένα νέο m-file.»**

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @gcentral_OpeningFcn, ...
'gui_OutputFcn',  @gcentral_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gcentral is made visible.
function gcentral_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to gcentral (see VARARGIN)

% Choose default command line output for gcentral
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gcentral wait for user response (see UIRESUME)
```

```

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gcentral_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
gthesn1d;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
gthesn2d;

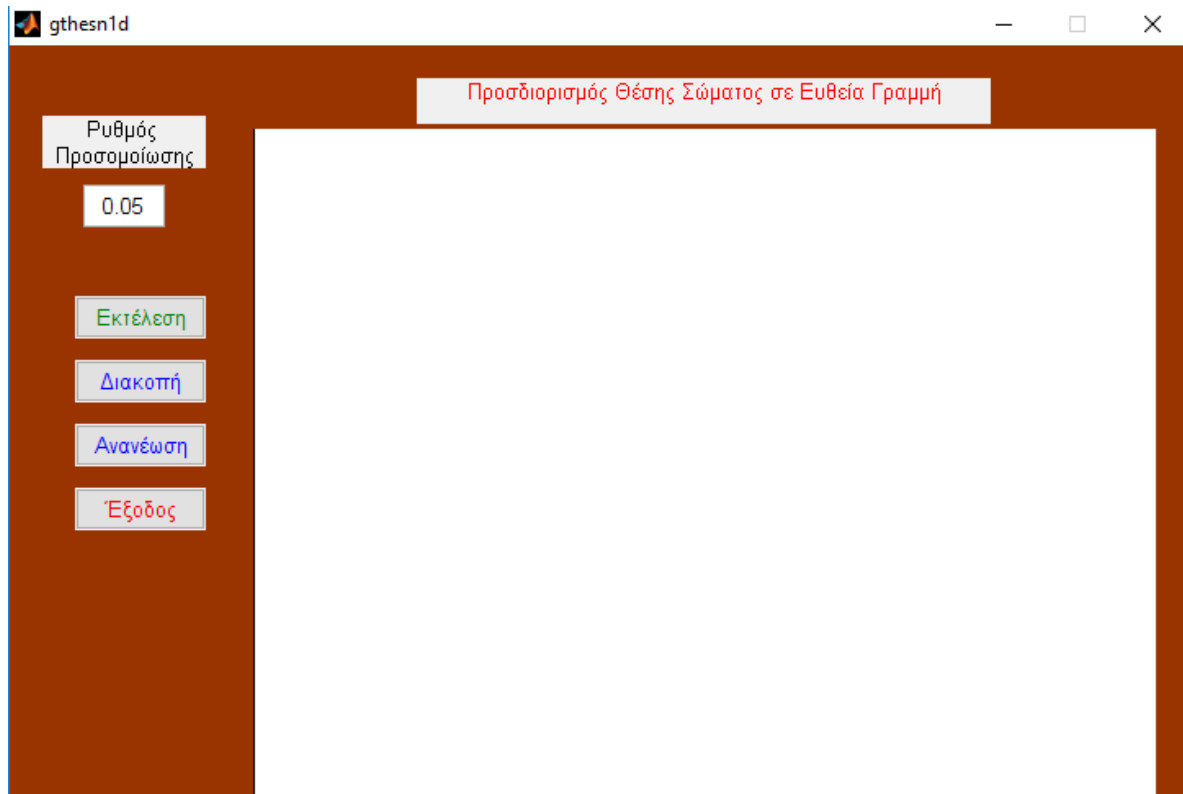
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
g8esn1;

```

Η συνάρτηση `functionpushbutton1_Callback(hObject, eventdata, handles)` αναφέρεται στο πρώτο κουμπί, δηλαδή στην επιλογή “ Άσκηση 1”, και μέσω αυτής οδηγούμαστε στο αντίστοιχο περιβάλλον προσομοίωσης. Αντίστοιχα η συνάρτηση `functionpushbutton2_Callback(hObject, eventdata, handles)` αναφέρεται στο δεύτερο κουμπί και παραπέμπει στο “Άσκηση 2” Και τέλος η πρόσβαση στην τρίτη επιλογή η οποία είναι ο “ Άσκηση 3 ” επιτυγχάνεται μέσω της συνάρτησης `functionpushbutton3_Callback(hObject, eventdata, handles)`

### 3.1.2 Το m-file gthesn1d – Άσκηση 1 (Κίνηση σε μία Διάσταση)

Επιλέγοντας την πρώτη επιλογή του μενού, “Άσκηση 1” , θα εμφανιστεί το περιβάλλον της εικόνας 3.2.

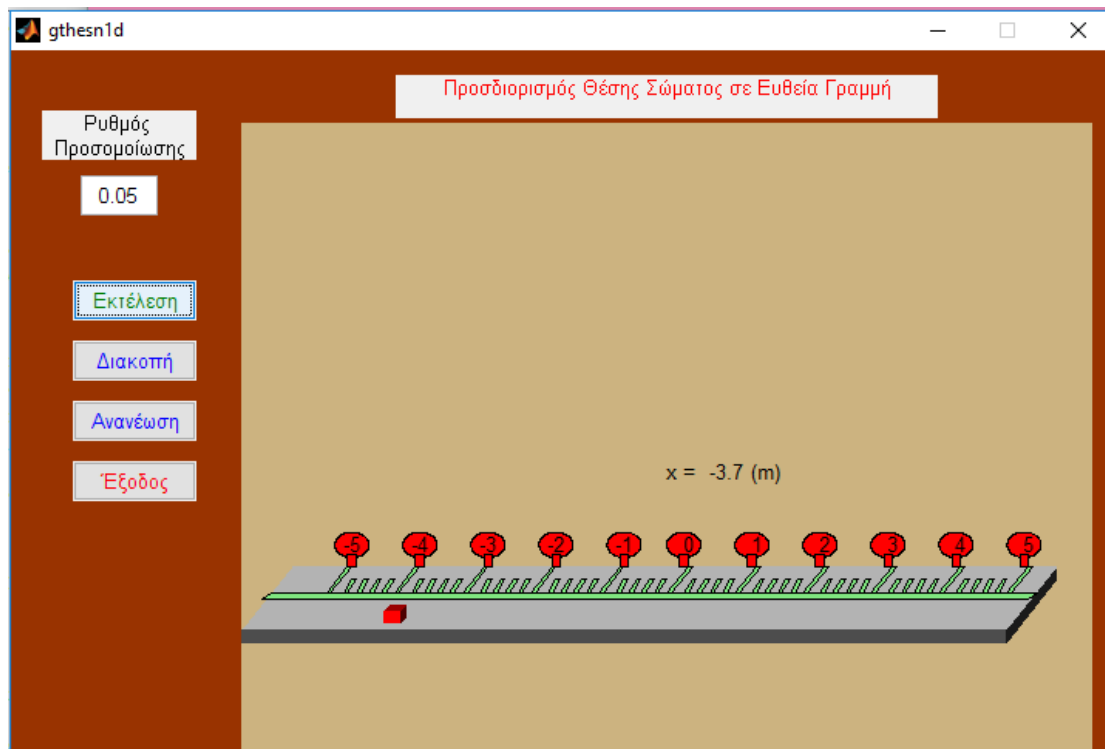


**Εικόνα 3.2: m-file gthesn1d**

Το περιβάλλον της πρώτης προσομοίωσης περιγράφει γραφικά την κίνηση σε μία διάσταση. Αποτελείται από δύο static text , “Προσδιορισμός Θέσης Σώματος σε Ευθεία Γραμμή”, “Ρυθμός Προσομοίωσης”, ένα edit text δίπλα από το τελευταίο static text που αναφέρθηκε παραπάνω όπου και μας δίνεται η δυνατότητα να επιλέξουμε μέσα από τις επιτρεπόμενες τιμές. Τέσσερα Push Button , “Εκτέλεση” , “Διακοπή”, “Ανανέωση” και “Έξοδος” και ένα axes.

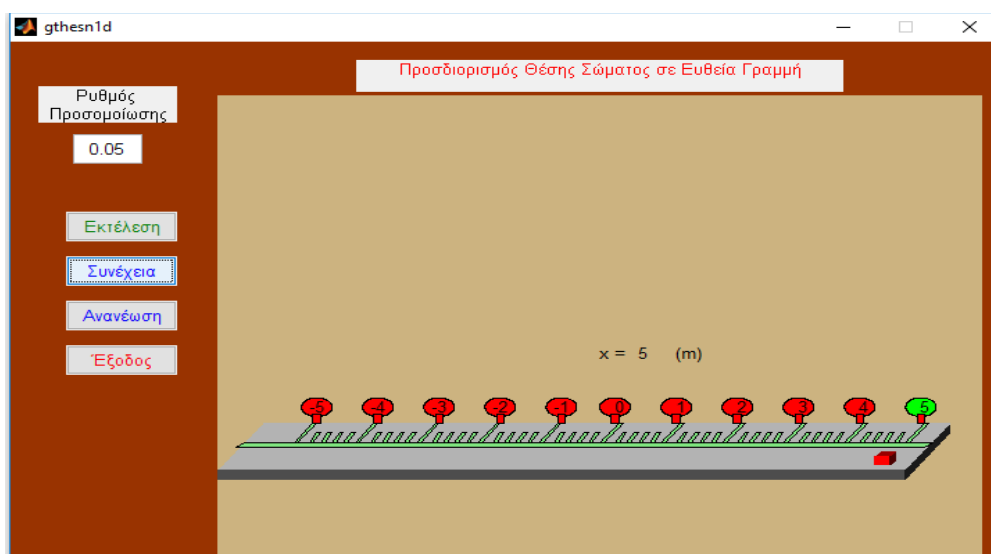
Εάν επιλέξουμε να εκτελεστεί η προσομοίωση το περιβάλλον παίρνει την μορφή της παρακάτω εικόνας 3.3.



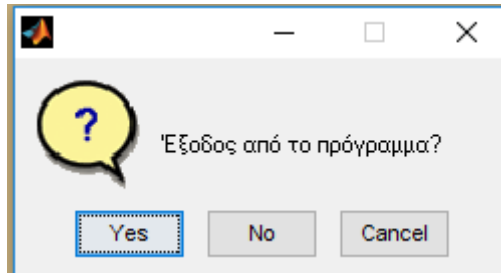


**Εικόνα 3.3: m-file gthesn1d**

Εδώ καθ' όλη την εξέλιξη της προσομοίωσης, μας δίνεται η δυνατότητα ανά πάσα στιγμή να κάνουμε διακοπή της και παρατηρούμε ότι ενεργοποιείται μία νέα επιλογή με όνομα “Συνέχεια”, ώστε να συνεχίσουμε από το σημείο που σταματήσαμε την εξέλιξη (εικόνα 3.4), να ανανεώσουμε την προσομοίωση με νέες τιμές στο πεδίο “Ρυθμός Προσομοίωσης”, όπως και να βγούμε από αυτό το περιβάλλον και να επιστρέψουμε στο αρχικό μενού με την επιλογή “Έξοδος”. Πατώντας το κουμπί “Έξοδος” θα εμφανιστεί ένα παράθυρο που θα μας ζητήσει να επιβεβαιώσουμε την επιλογή μας (εικόνα 3.5)



**Εικόνα 3.4: m-file gthesn1d**



**Εικόνα 3.5: m-file gthesn1d**

Ο κώδικας που συμπεριλαμβάνεται στο m-file gthesn1d περιγράφεται και εξηγείται παρακάτω:

```
function varargout = gthesn1d(varargin)
% GTHESN1D M-file for gthesn1d.fig
%     GTHESN1D, by itself, creates a new GTHESN1D or raises the
existing
%     singleton*.
%
%     H = GTHESN1D returns the handle to a new GTHESN1D or the
handle to
%     the existing singleton*.
%
%     GTHESN1D('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in GTHESN1D.M with the given input
arguments.
%
%     GTHESN1D('Property','Value',...) creates a new GTHESN1D or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before gthesn1d_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to gthesn1d_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gthesn1d

% Last Modified by GUIDE v2.5 16-Aug-2012 06:51:33

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @gthesn1d_OpeningFcn, ...
'gui_OutputFcn',  @gthesn1d_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
```

```

    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gthesn1d is made visible.
function gthesn1d_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gthesn1d (see VARARGIN)

% Choose default command line output for gthesn1d
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gthesn1d wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gthesn1d_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

**Οι συναρτήσεις function gthesn1d\_OpeningFcn(hObject, eventdata, handles, varargin) και function varargout = gthesn1d\_OutputFcn(hObject, eventdata, handles) δημιουργούνται από μόνες αυτόματα από το matlab.**

```

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1
as a double
```

Οι συναρτήσεις `function edit1_Callback` και `function edit1_CreateFcn` δημιουργούνται με το που προσθέτουμε το στοιχείο `edit1` στο σχεδιαστικό περιβάλλον του GUI, στο οποίο βάζουμε την τιμή του “Ρυθμού Προσομοίωσης”, εδώ θέτω το χρώμα του `background` να είναι λευκό.

```
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Η συνάρτηση `function pushbutton1_Callback` αναφέρεται στο πρώτο κουμπί της προσομοίωσης “Εκτέλεση”, το οποίο με το που πατηθεί γίνονται οι εξής λειτουργίες : γίνεται έλεγχος της τιμής που έχει δοθεί στο πεδίο “Ρυθμό Προσομοίωσης”, `ryt`.

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global ryt;
ryt=str2double(get(handles.edit1,'String'));
```

Εάν οι παρακάτω περιορισμοί δεν έχουν τις προαπαιτούμενες τιμές, εμφανίζεται ένα προειδοποιητικό παράθυρο διαλόγου όπου μας καλεί να βάλουμε τις σωστές τιμές στα προκαθορισμένα όρια. Τέλος όταν όλα έχουν γίνει σωστά, απενεργοποιούνται τα `edit1`, `pushbutton1` ώστε να προχωρήσει η προσομοίωση.

```

if (ryt>0.05)
    h=warndlg('Βάλτε τιμή στο ρυθμό μικρότερη ή ίση του 0.05');
return
end

if (ryt<=0.0)
    h=warndlg('Βάλτε τιμή στο ρυθμό θετική');
return
end

```

**Η εντολή axisoff; αφαιρεί από τους άξονες την αρίθμηση.**

```

global stam;
global status;
status=0;
axes(handles.axes1)
axis off;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
stam=0;
set(handles.pushbutton2,'string','Διακοπή')

```

**Ο σχεδιασμός ο οποίος είναι και σημαντικός για να κατανοήσουμε και το πώς δουλεύει η προσομοίωση, γίνεται σχεδιάζοντας ξεχωριστά το κάθε μέρος της γραφικής προσέγγισης, θεωρώντας ότι αποτελείται από διάφορα σχήματα. Ο τρόπος που σχεδιάζουμε ένα σχήμα είναι ο εξής:**

**Έστω ότι θέλουμε να σχεδιάσουμε τον τοίχο, όπως υπάρχει και στην προσομοίωση. Το σχήμα του τοίχου είναι ένα ορθογώνιο οπότε θα σχεδιάσουμε τέσσερα σημεία στους άξονες x και y δίνοντας συντεταγμένες που επιθυμούμε. Για τον άξονα x έχουμε xtoixos=[-5,5,5,-5]; και για τον άξονα y ytoixos=[-5,-5,5,5];**

```

xtoixos=[-5,5,5,-5];
ytoixos=[-5,-5,5,5];

xtable1=[-5.0,4.0,4.6,-4.4];
ytable1=[-3.0,-3.0,-2.0,-2.0];

xtable2=[-5.0,4.0,4.0,-5.0];
ytable2=[-3.0,-3.0,-3.2,-3.2];

xtable3=[4.0,4.6,4.6,4.0];

```

```

ytable3=[-3.2,-2.2,-2.0,-3.0];

xxarakas=[-4.75,4.25,4.35,-4.65];
yxarakas=[-2.5,-2.5,-2.4,-2.4];

xsignlarge=[0.0,0.05,0.3,0.25];
ysignlarge=[-2.4,-2.4,-2.0,-2.0];

xsignsmall=[0.0,0.05,0.15,0.10]+0.2;
ysignsmall=[-2.4,-2.4,-2.2,-2.2];

R=0.2;
th=0:0.01:2*pi;

xcycle=R*cos(th)-3.7;
ycycle=R*sin(th)-1.65;

xsupp=[-3.75,-3.65,-3.65,-3.75];
ysupp=[-2.0,-2.0,-1.8,-1.8];

for i=1:ryt:17

if (stam==1)
    cc=stam;
while (cc==1)
    cc=stam;
    pause(0.01);
if (status==1)
return
end
end
end

xmove=(i-1)*0.5;

x1=-5.0+(i-1)*10/16;
x=0.1*round(10*x1);
xx=num2str(x);

xcube1=[-4.4,-4.2,-4.2,-4.4]+xmove;
ycube1=[-2.9,-2.9,-2.7,-2.7];

xcube2=[-4.2,-4.12,-4.12,-4.2]+xmove;
ycube2=[-2.9,-2.8,-2.6,-2.7];

xcube3=[-4.4,-4.2,-4.12,-4.32]+xmove;
ycube3=[-2.7,-2.7,-2.6,-2.6];

if (x==-5)
    color1=[0.0,1.0,0.0];
else
    color1=[1.0,0.0,0.0];
end

if (x==-4)
    color2=[0.0,1.0,0.0];
else
    color2=[1.0,0.0,0.0];
end

```

```

end

if (x==-3)
    color3=[0.0,1.0,0.0];
else
    color3=[1.0,0.0,0.0];
end

if (x==-2)
    color4=[0.0,1.0,0.0];
else
    color4=[1.0,0.0,0.0];
end

if (x==-1)
    color5=[0.0,1.0,0.0];
else
    color5=[1.0,0.0,0.0];
end

if (x==0)
    color6=[0.0,1.0,0.0];
else
    color6=[1.0,0.0,0.0];
end

if (x==1)
    color7=[0.0,1.0,0.0];
else
    color7=[1.0,0.0,0.0];
end

if (x==2)
    color8=[0.0,1.0,0.0];
else
    color8=[1.0,0.0,0.0];
end

if (x==3)
    color9=[0.0,1.0,0.0];
else
    color9=[1.0,0.0,0.0];
end

if (x==4)
    color10=[0.0,1.0,0.0];
else
    color10=[1.0,0.0,0.0];
end

if (x==5)
    color11=[0.0,1.0,0.0];
else
    color11=[1.0,0.0,0.0];
end

```

Στη συνέχεια θα γεμίσουμε αυτό το χώρο που δημιουργήσαμε με την εντολή `fill` ως εξής: `fill(xtoixos,ytoixos,[0.8,0.7,0.5],... 'Linestyle',' None')` όπου δίνουμε τα ονόματα των μεταβλητών που θέλουμε το χρώμα και τον τύπο της περιμετρικής γραμμής. Ο χρωματισμός γίνεται βάζοντας μέσα στις αγκύλες τις τιμές έντασης των RedGreenBlue δημιουργώντας την επιθυμητή απόχρωση.

```
fill(xtoixos,ytoixos,[0.8,0.7,0.5],...
    xtable1,ytable1,[0.7,0.7,0.7],...
    xtable2,ytable2,[0.3,0.3,0.3],...
    xtable3,ytable3,[0.1,0.1,0.1],'Linestyle','None')
hold on;
fill(xxarakas,yarakas,[0.5,0.9,0.5],...
    xsignlarge,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.45,ysignsmall,[0.5,0.9,0.5],...
    xsignlarge+0.8,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall+0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15+0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30+0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.45+0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignlarge+0.8+0.8,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall+2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15+2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30+2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.45+2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignlarge+0.8+2*0.8,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall+3*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15+3*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30+3*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.45+3*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignlarge+0.8+3*0.8,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall+4*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15+4*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30+4*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.45+4*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignlarge+0.8+4*0.8,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall-0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15-0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30-0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.45-0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignlarge+0.8-0.8,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall-2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15-2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30-2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.45-2*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignlarge+0.8-2*0.8,ysignlarge,[0.5,0.9,0.5],...
    xsignsmall-3*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.15-3*0.8,ysignsmall,[0.5,0.9,0.5],...
    xsignsmall+0.30-3*0.8,ysignsmall,[0.5,0.9,0.5],...
```



```

xsignsmall+0.45-3*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignlarge+0.8-3*0.8,ysignlarge,[0.5,0.9,0.5],...
xsignsmall-4*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignsmall+0.15-4*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignsmall+0.30-4*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignsmall+0.45-4*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignlarge+0.8-4*0.8,ysignlarge,[0.5,0.9,0.5],...
xsignsmall-5*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignsmall+0.15-5*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignsmall+0.30-5*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignsmall+0.45-5*0.8,ysignsmall,[0.5,0.9,0.5],...
xsignlarge+0.8-5*0.8,ysignlarge,[0.5,0.9,0.5],...
xsignlarge+0.8-6*0.8,ysignlarge,[0.5,0.9,0.5],...
xcycle,ycycle,color1,xsupp,ysupp,'r',...
xcycle+0.8,ycycle,color2,xsupp+0.8,ysupp,'r',...
xcycle+1.6,ycycle,color3,xsupp+1.6,ysupp,'r',...
xcycle+2.4,ycycle,color4,xsupp+2.4,ysupp,'r',...
xcycle+3.2,ycycle,color5,xsupp+3.2,ysupp,'r',...
xcycle+4.0-0.1,ycycle,color6,xsupp+4.0-0.1,ysupp,'r',...
xcycle+4.8-0.1,ycycle,color7,xsupp+4.8-0.1,ysupp,'r',...
xcycle+5.6-0.1,ycycle,color8,xsupp+5.6-0.1,ysupp,'r',...
xcycle+6.4-0.1,ycycle,color9,xsupp+6.4-0.1,ysupp,'r',...
xcycle+7.2-0.1,ycycle,color10,xsupp+7.2-0.1,ysupp,'r',...
xcycle+8.0-0.1,ycycle,color11,xsupp+8.0-0.1,ysupp,'r')
hold on
fill(xcubel,ycubel,[1.0,0.0,0.0],...
xcube2,ycube2,[0.3,0.0,0.0],...
xcube3,ycube3,[0.6,0.0,0.0],'LineStyle','None')
hold off;
axis([-5,5,-5,5])
axis off;

```

**Με την εντολή text εισάγουμε σε οποιοδήποτε σημείο των αξόνων θέλουμε το κείμενο που επιθυμούμε.**

```

text(0.2,-1.65,'0')
text(1.0,-1.65,'1')
text(1.8,-1.65,'2')
text(2.6,-1.65,'3')
text(3.4,-1.65,'4')
text(4.2,-1.65,'5')

text(-0.6,-1.65,'-1')
text(-1.4,-1.65,'-2')
text(-2.2,-1.65,'-3')
text(-3.0,-1.65,'-4')
text(-3.8,-1.65,'-5')

text(0.0,-0.5,'x =')
text(0.5,-0.5,xx)
text(1.0,-0.5,'(m)')

if (i==1)
    pause(1.0);
else

```

```

        pause(0.01);
end
end

```

**Ενεργοποίηση του κουμπιού “Διακοπή”, το οποίο όταν το πατήσουμε σταματάει τη προσομοίωση και μετατρέπεται σε “Συνέχεια”, και αντίστοιχα εάν ξαναπατηθεί συνεχίζει η εκτέλεση της προσομοίωσης απ’ το σημείο που σταμάτησε και ξαναενεργοποιείται η “Διακοπή”.**

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global stam;

if (stam==0)
    set(handles.pushbutton2,'string','Συνέχεια')
    stam=1;
elseif (stam==1)
    set(handles.pushbutton2,'string','Διακοπή')
    stam=0;
else
end
guidata(hObject, handles);

```

**Σχεδιασμός του κουμπιού “Ανανέωση” , το οποίο όταν το πατήσουμε, θέτει ξανά τις αρχικές τιμές στις μεταβλητές ryt.**

```

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global ryt;
global status;
status=0;
axes(handles.axes1)
cla
clear ryt;
set(handles.edit1,'enable','on','string','0.05');
guidata(hObject, handles);

```

Σχεδιασμός του κουμπιού “Εξοδος” , το οποίο όταν πατηθεί εμφανίζεται ένα παράθυρο διαλόγου και μας ρωτάει για επιβεβαίωση εξόδου από το πρόγραμμα. Εν συνεχεία γίνεται έλεγχος επιλογής και τερματίζει την εφαρμογή όταν η επιλογή είναι 'Yes'.

```
% --- Executes on button press in pushbutton4.  
function pushbutton4_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton4 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
global status;  
hfin=questdlg('Εξοδος από το πρόγραμμα?');  
switch hfin  
case 'Yes'  
    status=1;  
closereq;  
end
```

### 3.1.3 Το m-file gthesn2d – Άσκηση 2 (Κίνηση σε δύο Διαστάσεις)

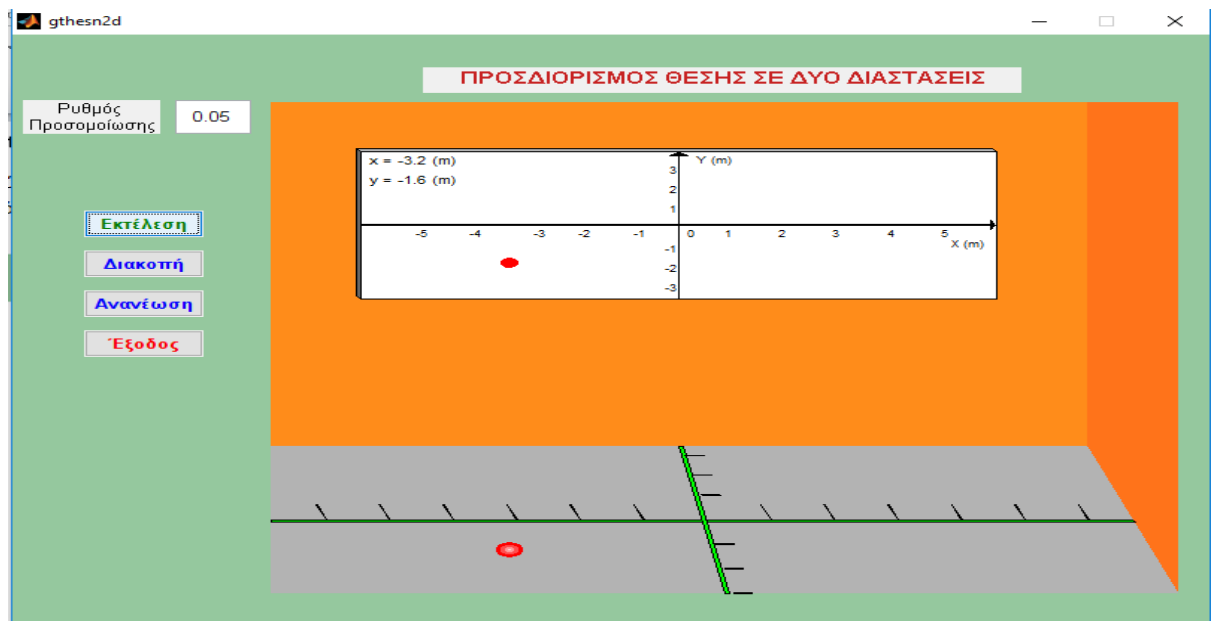
Επιλέγοντας την δεύτερη επιλογή του μενού, “Άσκηση 2” , θα εμφανιστεί το περιβάλλον της εικόνας 3.6.



**Εικόνα 3.6: m-file gthesn2d**

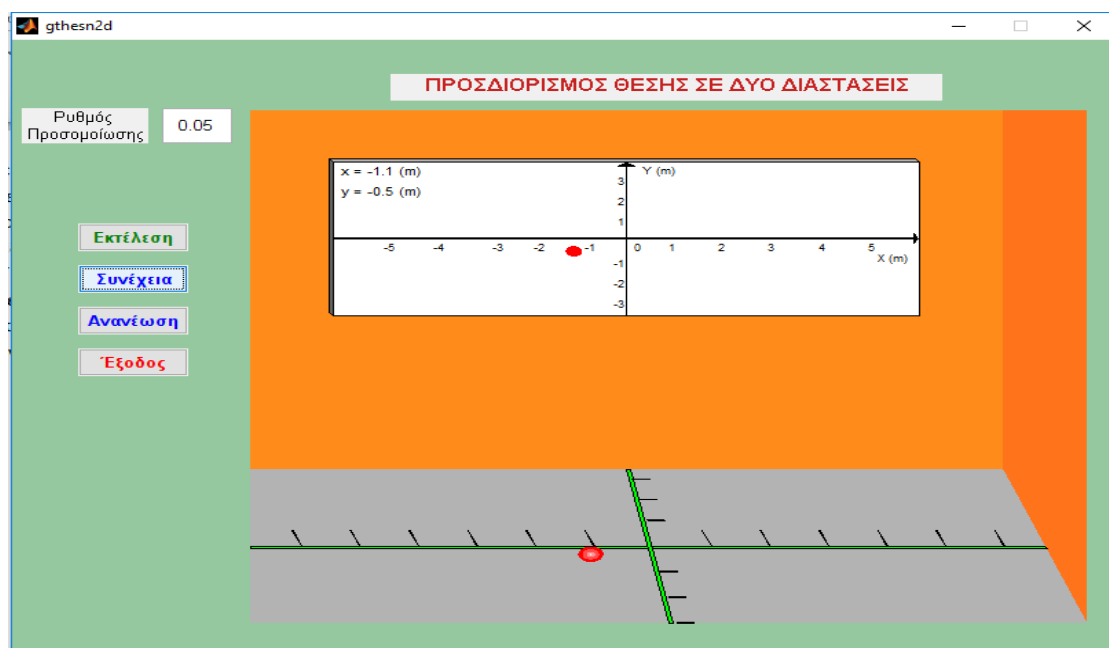
Το περιβάλλον της δεύτερης προσομοίωσης περιγράφει γραφικά την κίνηση σε δύο διάστας. Αποτελείται από δύο static text , “Προσδιορισμός Θέσης σε Δύο Διαστάσεις”, “Ρυθμός Προσομοίωσης”, ένα edit text δίπλα από το τελευταίο static text που αναφέρθηκε παραπάνω όπου και μας δίνεται η δυνατότητα να επιλέξουμε μέσα από τις επιτρεπόμενες τιμές. Τέσσερα Push Button , “Εκτέλεση” , “Διακοπή”, “Ανανέωση” και “Έξοδος” και ένα axes.

Εάν επιλέξουμε να εκτελεστεί η προσομοίωση το περιβάλλον παίρνει την μορφή της παρακάτω εικόνας 3.7.

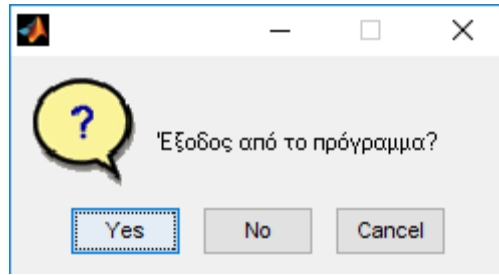


Εικόνα 3.7: m-file gthesn2d

Εδώ καθ' όλη την εξέλιξη της προσομοίωσης, μας δίνεται η δυνατότητα ανά πάσα στιγμή να κάνουμε διακοπή της και παρατηρούμε ότι ενεργοποιείται μία νέα επιλογή με όνομα “Συνέχεια”, ώστε να συνεχίσουμε από το σημείο που σταματήσαμε την εξέλιξη (εικόνα 3.8), να ανανεώσουμε την προσομοίωση με νέες τιμές στο πεδίο “Ρυθμός Προσομοίωσης”, όπως και να βγούμε από αυτό το περιβάλλον και να επιστρέψουμε στο αρχικό μενού με την επιλογή “Έξοδος”. Πατώντας το κουμπί “Έξοδος” θα εμφανιστεί ένα παράθυρο που θα μας ζητήσει να επιβεβαιώσουμε την επιλογή μας (εικόνα 3.9).



Εικόνα 3.8: m-file gthesn2d



**Εικόνα 3.9: m-file gthesn2d**

Ο κώδικας που συμπεριλαμβάνεται στο m-file gthesn2d περιγράφεται και εξηγείται παρακάτω:

```
function varargout = gthesn2d(varargin)
% GTHESN2D M-file for gthesn2d.fig
%     GTHESN2D, by itself, creates a new GTHESN2D or raises the
existing
%     singleton*.
%
%     H = GTHESN2D returns the handle to a new GTHESN2D or the
handle to
%     the existing singleton*.
%
%     GTHESN2D('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in GTHESN2D.M with the given input
arguments.
%
%     GTHESN2D('Property','Value',...) creates a new GTHESN2D or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before gthesn2d_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to gthesn2d_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gthesn2d

% Last Modified by GUIDE v2.5 24-Aug-2012 04:49:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @gthesn2d_OpeningFcn, ...
'gui_OutputFcn',  @gthesn2d_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
```

```

    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gthesn2d is made visible.
function gthesn2d_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to gthesn2d (see VARARGIN)

% Choose default command line output for gthesn2d
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gthesn2d wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gthesn2d_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

```



```

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
%        as a double

```

```
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

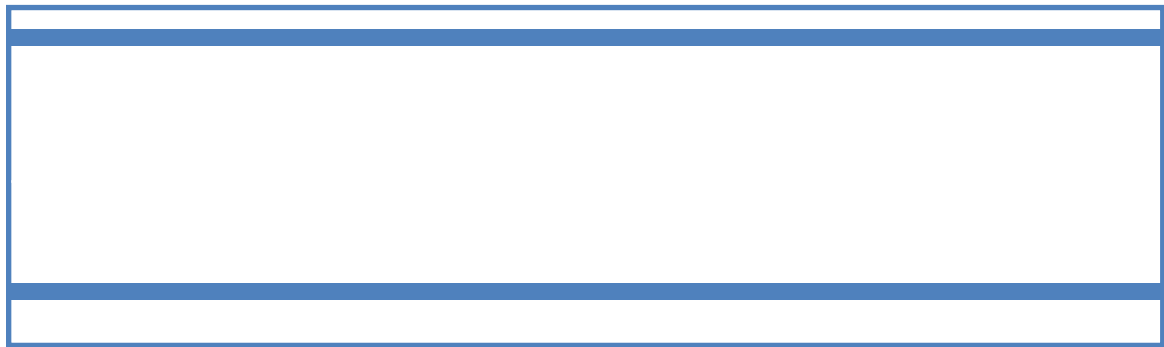
```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global ryt;
ryt=str2double(get(handles.edit1,'String'));

if (ryt>1)
    h=warndlg('Βάλτε τιμή στο ρυθμό μικρότερη ή ίση του 1');
return
end

if (ryt<=0.0)
    h=warndlg('Βάλτε τιμή στο ρυθμό θετική');
return
end
```

```
global stam;
global status;
status=0;
axes(handles.axes1)
axis off;
```





```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stam=0;
set(handles.pushbutton2,'string','Διακοπή')
```



```
xtoixos1=[-5.0,4.0,4.0,-5.0];
ytoixos1=[-2.0,-2.0,5.0,5.0];

xtoixos2=[-5.0,5.0,4.0,-5.0];
ytoixos2=[-5.0,-5.0,-2.0,-2.0];

xtoixos3=[4.0,5.0,5.0,4.0];
ytoixos3=[-2.0,-5.0,5.0,5.0];

xaxisx=[-5.0,4.53,4.5,-5.0];
yaxisx=[-3.55,-3.55,-3.5,-3.5];
```

```

xaxisy=[0.0,0.05,-0.45,-0.50];
yaxisy=[-5.0,-5.0,-2.0,-2.0];

xo8ovn1=[-4.0,3.0,3.0,-4.0];
yo8ovn1=[1.0,1.0,4.0,4.0];

xo8ovn2=[-4.05,-4.0,3.0,2.95];
yo8ovn2=[4.05,4.0,4.0,4.05];

xo8ovn3=[-4.05,-4.0,-4.0,-4.05];
yo8ovn3=[1.05,1.0,4.0,4.05];

xaxis1x=[-4.0,2.93,2.93,2.99,2.93,2.93,-4.0];
yaxis1x=[2.49,2.49,2.40,2.5,2.6,2.51,2.51];

xaxis1y=[-0.005,0.005,0.005,0.1,0.0,-0.1,-0.005]-0.5;
yaxis1y=[1.0,1.0,3.90,3.90,3.96,3.90,3.90];

xtickx=[0.52,0.51,0.41,0.40];
ytickx=[-3.5,-3.5,-3.2,-3.2];

% xtick1x=[0.5,0.51,0.47,0.46]-0.6;
% ytick1x=[-3.5,-3.5,-3.3,-3.3];

xticky=[-0.09,0.11,0.1,-0.1]-0.15;
yticky=[-3.0,-3.0,-2.99,-2.99];
%
% xtick1y=[-0.02,0.04,0.02,-0.04]+0.15;
% ytick1y=[-4.90,-4.90,-4.89,-4.89];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
R=0.15;
R1=0.1;
R2=0.05;
R3=0.015;
th=0:0.01:2*pi;
% ryt=0.05;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

for i=1:0.1:11

if (stam==1)
    cc=stam;
while (cc==1)
    cc=stam;
    pause(0.01);
if (status==1)
return
end
end
end

    x=-6+(i-1)*12/10;

```

```

y=-3.0+(i-1)*6.0/10;

xx=num2str(0.1*round(10*x));
yy=num2str(0.1*round(10*y));

xmove=-4.2+(i-1)*8/10;
ymove=-4.7+(i-1)*2.5/10;

xball=xmove+R*cos(th);
yball=ymove+R*sin(th);

xmove1=-3.95+(i-1)*6.9/10;
ymove1=1.05+(i-1)*2.9/10;

xball1=xmove1+R1*cos(th);
yball1=ymove1+R1*sin(th);

xball2=xmove+R1*cos(th);
yball2=ymove+R1*sin(th);

xball3=xmove+R2*cos(th);
yball3=ymove+R2*sin(th);

xball4=xmove+R3*cos(th);
yball4=ymove+R3*sin(th);

```



```

fill (xtoixos1,ytoixos1,[1.0, 0.55,0.1],...
      xtoixos2,ytoixos2,[0.7,0.7,0.7],...
      xtoixos3,ytoixos3,[1.0, 0.45, 0.1],...
      'LineStyle','None')
hold on

fill(xaxisx,yaxisx,[0.0,1.0,0.0],...
     xaxisy,yaxisy,[0.0,1.0,0.0],...
     xo8ovn1,yo8ovn1,[1.0,1.0,1.0],...
     xo8ovn2,yo8ovn2,[0.7,0.7,0.7],...
     xo8ovn3,yo8ovn3,[0.4,0.4,0.4],...
     xaxis1x,yaxis1x,'k',xaxis1y,yaxis1y,'k',...
     xtickx,ytickx,'k',...
     xtickx+0.7,ytickx,'k',...
     xtickx+1.4,ytickx,'k',...
     xtickx+2.1,ytickx,'k',...)

```

```

xtickx+2.8,ytickx,'k',...
xtickx+3.5,ytickx,'k',...
xtickx-1.4,ytickx,'k',...
xtickx-2.1,ytickx,'k',...
xtickx-2.8,ytickx,'k',...
xtickx-3.5,ytickx,'k',...
xtickx-4.2,ytickx,'k',...
xtickx-4.9,ytickx,'k',...
xticky,ytickx,'k',...
xticky-0.1,ytickx+0.40,'k',...
xticky-0.18,ytickx+0.80,'k',...
xticky+0.15,ytickx-1.0,'k',...
xticky+0.25,ytickx-1.5,'k',...
xticky+0.35,ytickx-2.0,'k');
hold on
fill(xball,yball,[1.0,0.0,0.0],...
xball1,yball1,[1.0,0.0,0.0],...
xball2,yball2,[1.0,0.3,0.3],...
xball3,yball3,[1.0,0.6,0.6],...
xball4,yball4,[1.0,0.9,0.9],'LineStyle','None')
hold off

```



```

text(-3.9,3.8,'x = (m)', 'FontSize',8)
text(-3.6,3.8,xx, 'FontSize',8)
text(-3.9,3.4,'y = (m)', 'FontSize',8)
text(-3.6,3.4,yy, 'FontSize',8)

text(-0.4,2.3,'0', 'FontSize',7)
text(0.0,2.3,'1', 'FontSize',7)
text(0.6,2.3,'2', 'FontSize',7)
text(1.2,2.3,'3', 'FontSize',7)
text(1.8,2.3,'4', 'FontSize',7)
text(2.4,2.3,'5', 'FontSize',7)
% text(2.9,2.3,'6', 'FontSize',6)

text(-1.0,2.3,'-1', 'FontSize',7)
text(-1.6,2.3,'-2', 'FontSize',7)
text(-2.1,2.3,'-3', 'FontSize',7)
text(-2.8,2.3,'-4', 'FontSize',7)
text(-3.4,2.3,'-5', 'FontSize',7)
% text(-4.0,2.3,'-6', 'FontSize',6)

text(-0.6,2.8,'1', 'FontSize',7)
text(-0.6,3.2,'2', 'FontSize',7)
text(-0.6,3.6,'3', 'FontSize',7)

text(-0.65,2.0,'-1', 'FontSize',7)
text(-0.65,1.6,'-2', 'FontSize',7)
text(-0.65,1.2,'-3', 'FontSize',7)

```

```
text(2.5,2.1,'X (m)','FontSize',7)
text(-0.3,3.8,'Y (m)','FontSize',7)
```

```
axis([-5.0,5.0,-5.0,5.0]);
```

```
axis off;
```

```
if (i==1)
    pause(1.0)
```

```
else
    pause(ryt)
```

```
end
```

```
end
```



```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global stam;
```

```
if (stam==0)
    set(handles.pushbutton2,'string','Συνέχεια')
    stam=1;
```

```
elseif (stam==1)
    set(handles.pushbutton2,'string','Διακοπή')
    stam=0;
```

```
else
```

```
end
```

```
guidata(hObject, handles);
```



```
% --- Executes on button press in pushbutton3.
```

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

```

% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global ryt;
global status;
status=0;
axes(handles.axes1)
cla
clear ryt;
set(handles.edit1,'enable','on','string','0.05');
guidata(hObject, handles);

```

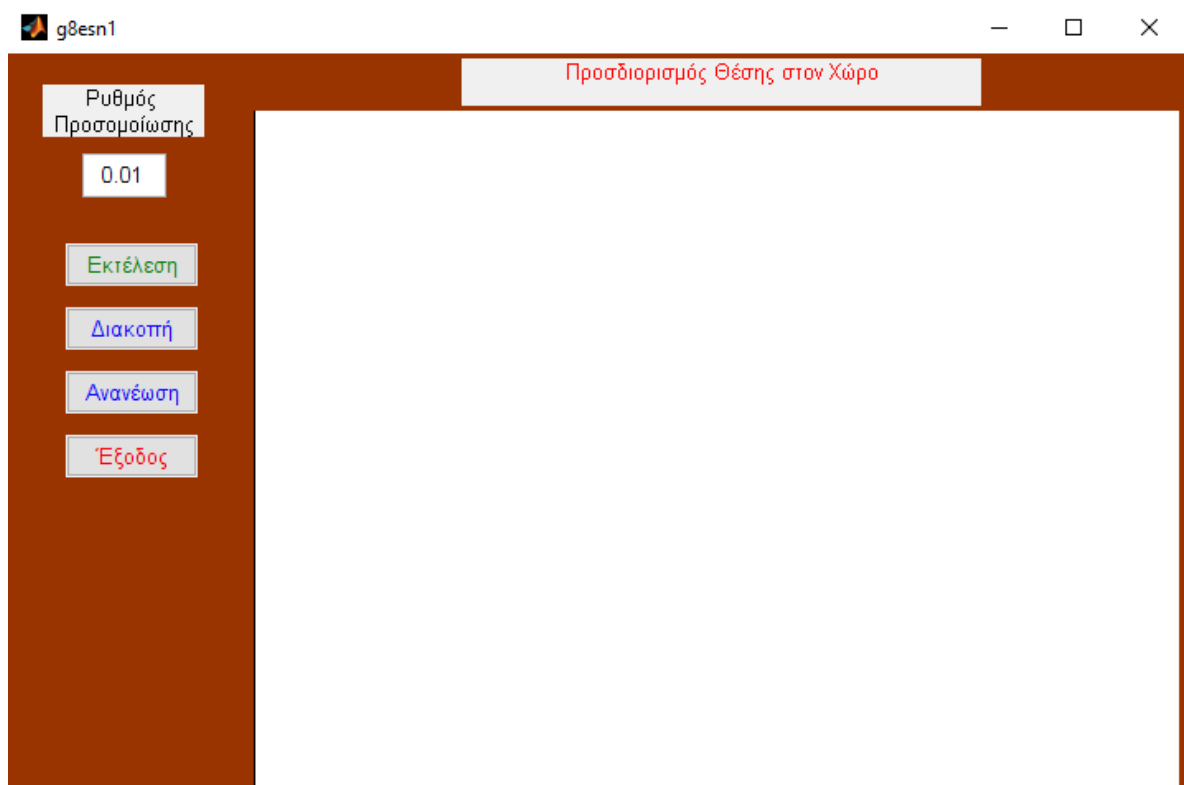
```

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global status;
hfin=questdlg('Εξοδος από το πρόγραμμα?');
switch hfin
case 'Yes'
    status=1;
closereq;
end

```

### 3.1.4 Το m-file g8esn1 – Άσκηση 3 (Κίνηση σε Τρεις Διαστάσεις)

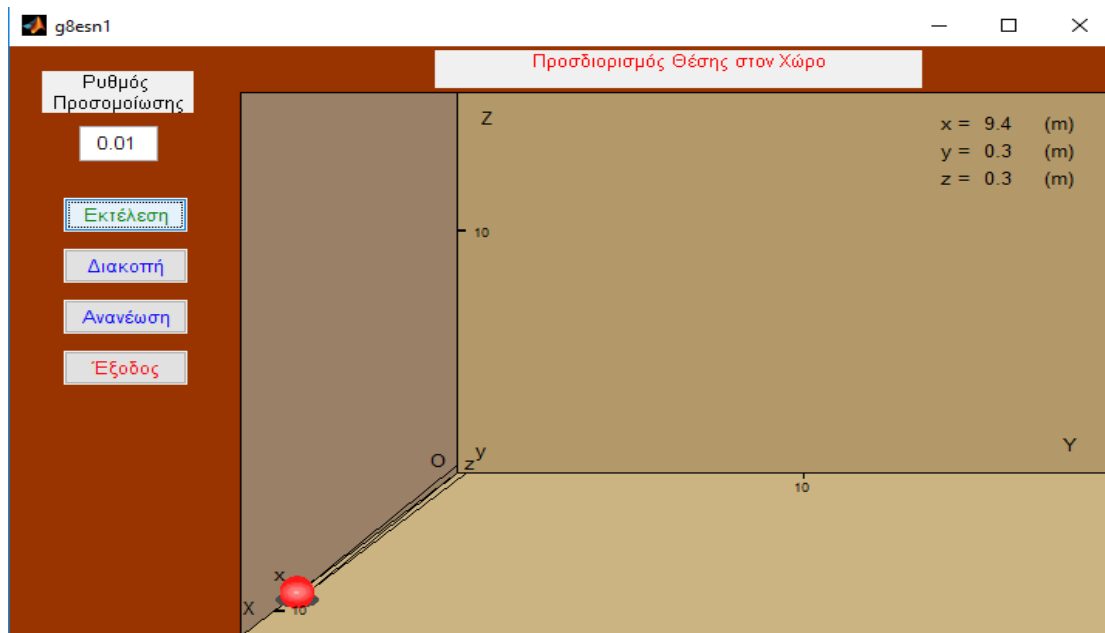
Επιλέγοντας την δεύτερη επιλογή του μενού, “Άσκηση 3” , θα εμφανιστεί το περιβάλλον της εικόνας 3.10.



**Εικόνα 3.10: m-file g8esn1.**

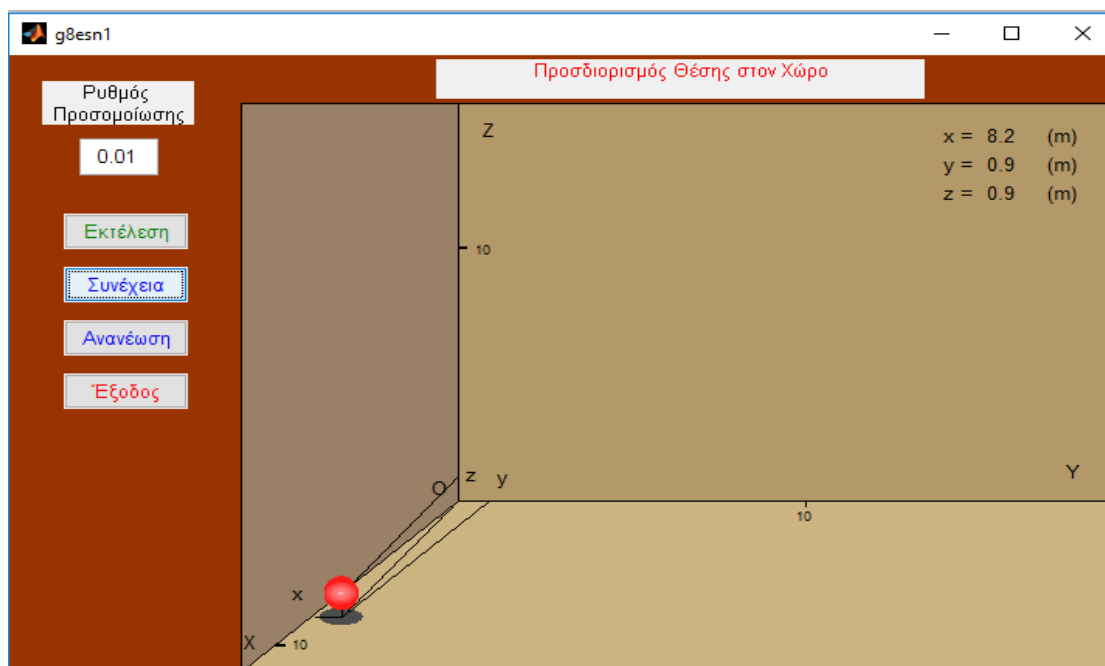
Το περιβάλλον της τρίτης προσομοίωσης περιγράφει γραφικά την κίνηση σε χώρο. Αποτελείται από δύο static text , “Προσδιορισμός Θέσης στο Χώρο”, “Ρυθμός Προσομοίωσης”, ένα edit text δίπλα από το τελευταίο static text που αναφέρθηκε παραπάνω όπου και μας δίνεται η δυνατότητα να επιλέξουμε μέσα από τις επιτρεπόμενες τιμές. Τέσσερα Push Button , “Εκτέλεση” , “Διακοπή”, “Ανανέωση” και “Έξοδος” και ένα axes.

Εάν επιλέξουμε να εκτελεστεί η προσομοίωση το περιβάλλον παίρνει την μορφή της παρακάτω εικόνας 3.11.



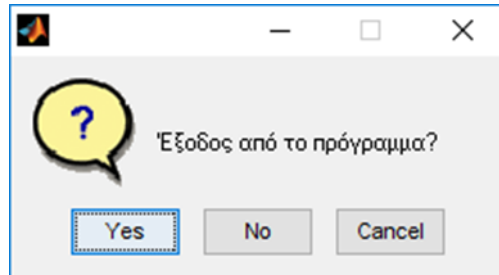
**Εικόνα 3.11: m-file g8esn1.**

Εδώ καθ' όλη την εξέλιξη της προσομοίωσης, μας δίνεται η δυνατότητα ανά πάσα στιγμή να κάνουμε διακοπή της και παρατηρούμε ότι ενεργοποιείται μία νέα επιλογή με όνομα "Συνέχεια", ώστε να συνεχίσουμε από το σημείο που σταματήσαμε την εξέλιξη (εικόνα 3.12), να ανανεώσουμε την προσομοίωση με νέες τιμές στο πεδίο "Ρυθμός Προσομοίωσης", όπως και να βγούμε από αυτό το περιβάλλον και να επιστρέψουμε στο αρχικό μενού με την επιλογή "Έξοδος". Πατώντας το κουμπί "Έξοδος" θα εμφανιστεί ένα παράθυρο που θα μας ζητήσει να επιβεβαιώσουμε την επιλογή μας (εικόνα 3.13).



**Εικόνα 3.12: m-file g8esn1.**





**Εικόνα 3.13: m-file g8esn1.**

Ο κώδικας που συμπεριλαμβάνεται στο m-file gthesn2d περιγράφεται και εξηγείται παρακάτω:

```
function varargout = g8esn1(varargin)
% G8ESN1 M-file for g8esn1.fig
%     G8ESN1, by itself, creates a new G8ESN1 or raises the existing
%     singleton*.
%
%     H = G8ESN1 returns the handle to a new G8ESN1 or the handle to
%     the existing singleton*.
%
%     G8ESN1('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in G8ESN1.M with the given input
arguments.
%
%     G8ESN1('Property','Value',...) creates a new G8ESN1 or raises
the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before g8esn1_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to g8esn1_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help g8esn1

% Last Modified by GUIDE v2.5 26-Jul-2012 00:02:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @g8esn1_OpeningFcn, ...
'gui_OutputFcn',  @g8esn1_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before g8esn1 is made visible.
function g8esn1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to g8esn1 (see VARARGIN)

% Choose default command line output for g8esn1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes g8esn1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = g8esn1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

```

```

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
%        as a double

```

```
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global ryt;
ryt=str2double(get(handles.edit1,'String'));

if (ryt>0.05)
    h=warndlg('Βάλτε τιμή στο ρυθμό μικρότερη ή ίση του 0.05');
return
end

if (ryt<=0.0)
    h=warndlg('Βάλτε τιμή στο ρυθμό θετική');
return
end
```

```
axes(handles.axes1)
axis off;
```

[illegible]


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
stam=0;
set(handles.pushbutton2,'string','Διακοπή')
```

[illegible]

```

% ΕΞΙΣΩΣΕΙΣ ΔΙΑΜΑΤΙΟΥ
xtoixos1 = [0.0,0.25,0.25,0.0];
ytoixos1 = [0.0,0.3,1.0,1.0];

xtoixos2 = [0.0,0.25,1.0,1.0];
ytoixos2 = [0.0,0.3,0.3,0.0];

xtoixos3 = [0.25,1.0,1.0,0.25];
ytoixos3 = [0.3,0.3,1,1];

%Swma
% 0<=xbody<=10
% 0<=ybody<=10
% 0<=zbody<=10
xtick1 = [0.02,0.03,0.03,0.02]+0.02;
ytick1 = [0.01,0.01,0.011,0.011]+0.035;

xtick2 = [0.65,0.651,0.651,0.65];
ytick2 = [0.3,0.3,0.29,0.29];

for t=0:ryt:1;

if (stam==1)
    cc=stam;
while (cc==1)
    cc=stam;
    pause(0.01);
if (status==1)
return
end
end
end

xbody=10.0-10.0*t;
xbodymax=12;
ybody=5.0*t;
ybodymax=12;
zbody=5.0*t;
zbodymax=12;
theta=atan(0.25/0.30);
xmax=(0.25*0.25+0.3*0.3)^0.5;

x1=num2str(xbody);
y1=num2str(ybody);
z1=num2str(zbody);

zdip=(xbody/xbodymax)*xmax*cos(theta);
xdisp=(0.3-zdip*tan(theta))+0.5*(ybody/ybodymax);
zpos=0.3-zdip+(zbody/zbodymax)*0.55;

th=0:0.01:2*pi;
R=0.02;
R1=0.025;

```

```

R2=0.015;
R3=0.03;

xball1=xdisp+R*cos(th)-0.05;
yball1=zpos+R3*sin(th);

xball11=xdisp+0.95*R*cos(th)-0.05;
yball11=zpos+0.95*R*sin(th);

xball12=xdisp+0.8*R*cos(th)-0.05;
yball12=zpos+0.8*R*sin(th);

xball13=xdisp+0.6*R*cos(th)-0.05;
yball13=zpos+0.6*R*sin(th);

xball14=xdisp+0.5*R*cos(th)-0.05;
yball14=zpos+0.5*R*sin(th);

xball15=xdisp+0.3*R*cos(th)-0.05;
yball15=zpos+0.3*R*sin(th);

xball16=xdisp+0.15*R*cos(th)-0.05;
yball16=zpos+0.15*R*sin(th);

% ProBoles

xvert1=R1*cos(th)+xdisp-0.05;
yvert1=R2*sin(th)+0.28-zdip+0.02;

xpol=[0.25,xdisp-0.05,xdisp-0.05,0.25];
ypol=[0.30,0.3-zdip,0.30-zdip,0.30];

xpolh=[0.25,xdisp-0.05,xdisp-0.05,0.25];
ypolh=[0.3+t*0.24,zpos,zpos,0.3+t*0.24];

xlinex=[0.25+t*0.20,xdisp-0.05,xdisp-0.05,0.25+t*0.20];
ylinex=[0.30,0.3-zdip,0.30-zdip,0.30];

xliney=[0.05+t*0.20,xdisp-0.05,xdisp-0.05,0.05+t*0.20];
yliney=[0.30-zdip,0.3-zdip,0.30-zdip,0.30-zdip];

xlinez=[xdisp-0.05,xdisp-0.05,xdisp-0.05,xdisp-0.05];
ylinez=[zpos,0.3-zdip,0.3-zdip,zpos];

```



```

fill(xtoixos1, ytoixos1, [0.6, 0.5, 0.4],...
     xtoixos2, ytoixos2, [0.8 0.7 0.51],...
     xtoixos3, ytoixos3, [0.7 0.6 0.41])
hold on;
fill(xvert1,yvert1,[0.3,0.3,0.3], 'LineStyle','None')
hold on;
fill(xpol,ypol,'k',...
     xpolh,ypolh,'k',...
     xlinex,ylinex,'k',...
     xliney,yliney,'k',...
     xlinez,ylinez,'k',...
     xtack1,ytick1,'k',...
     xtack2,ytick2,'k',...
     xtack1+0.21,ytick1+0.70,'k')
hold on;
fill(...
     xball,yball,[1.0,0.1,0.1],...
     xball1,yball1,[1.0,0.2,0.2],...
     xball2,yball2,[1.0,0.3,0.3],...
     xball3,yball3,[1.0,0.4,0.4],...
     xball4,yball4,[1.0,0.45,0.45],...
     xball5,yball5,[1.0,0.5,0.5],...
     xball6,yball6,[1.0,0.7,0.7],...
     'LineStyle','None')
hold off;

```



```

text(0.005,0.05,'X')
text(0.95,0.35,'Y')
text(0.28,0.95,'Z')
text(0.22,0.32,'O')

text(0.81,0.94,'x =')
text(0.86,0.94,x1)
text(0.93,0.94,'(m)')
text(0.81,0.89,'y =')
text(0.86,0.89,y1)
text(0.93,0.89,'(m)')
text(0.81,0.84,'z =')
text(0.86,0.84,z1)
text(0.93,0.84,'(m)')

xx1=0.03+t*(0.2-0.03);
yy1=0.1+t*(0.2);

text(xx1,yy1,'x')

xx2=0.26+t*(0.2);
yy2=0.34;

```

```

text(xx2,yy2,'y')

xx3=0.26;
yy3=0.3+t*(0.25);

text(xx3,yy3,'z')

text(0.06,0.045,'10','FontSize',7)
text(0.64,0.27,'10','FontSize',7)
text(0.27,0.74,'10','FontSize',7)

%axis equal;
axis off;

pause(0.25)

end

```



```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global stam;

if (stam==0)
    set(handles.pushbutton2,'string','Συνέχεια')
    stam=1;
elseif (stam==1)
    set(handles.pushbutton2,'string','Διακοπή')
    stam=0;
else
end
guidata(hObject, handles);

```





```

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global ryt;
global status;
status=0;
axes(handles.axes1)
cla
clear ryt;
set(handles.edit1,'enable','on','string','0.01');
guidata(hObject, handles);

```

```

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global status;
hfin=questdlg('Εξοδος από το πρόγραμμα?');
switch hfin
case 'Yes'
    status=1;
closereq;
end

```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Βοήθεια του λογισμικού Matlab
2. [http://gym-peir-irakl.ira.sch.gr/stud\\_toump/notes/physB\\_kefalaio2.pdf](http://gym-peir-irakl.ira.sch.gr/stud_toump/notes/physB_kefalaio2.pdf)
3. [http://physiclessons.blogspot.gr/2012/02/blog-post\\_418.html](http://physiclessons.blogspot.gr/2012/02/blog-post_418.html)
4. <http://ebooks.edu.gr/modules/ebook/show.php/DSGL-A103/529/3518,14438/>
5. [https://el.wikipedia.org/wiki/%CE%9A%CF%85%CE%BA%CE%BB%CE%B9%CE%BA%CE%AE\\_%CE%BA%CE%AF%CE%BD%CE%B7%CF%83%CE%B7](https://el.wikipedia.org/wiki/%CE%9A%CF%85%CE%BA%CE%BB%CE%B9%CE%BA%CE%AE_%CE%BA%CE%AF%CE%BD%CE%B7%CF%83%CE%B7)
6. [https://el.wikipedia.org/wiki/%CE%95%CF%85%CE%B8%CF%8D%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B7\\_%CE%BF%CE%BC%CE%B1%CE%BB%CE%AE\\_%CE%BA%CE%AF%CE%BD%CE%B7%CF%83%CE%B7](https://el.wikipedia.org/wiki/%CE%95%CF%85%CE%B8%CF%8D%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B7_%CE%BF%CE%BC%CE%B1%CE%BB%CE%AE_%CE%BA%CE%AF%CE%BD%CE%B7%CF%83%CE%B7)