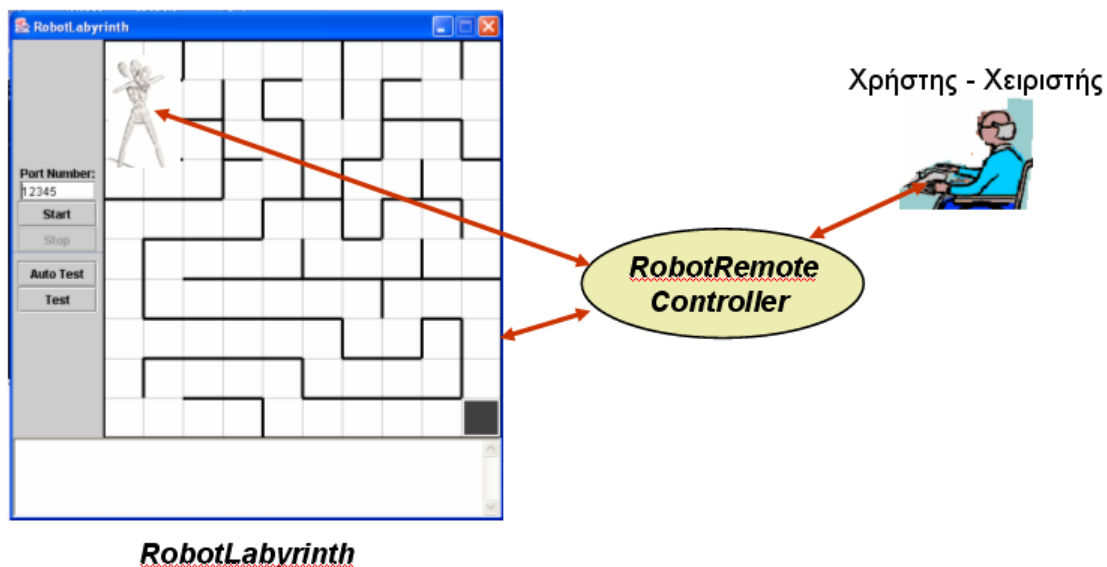


# Άσκηση RobotLabyrinth

## 1. Περιγραφή

Στα πλαίσια της άσκησης αυτής θα αναπτύξετε ένα πρόγραμμα που θα επιτρέπει στον χρήστη του να καθοδηγεί, χρησιμοποιώντας ορισμένα πλήκτρα του πληκτρολογίου, ένα ρομπότ μέσα σε έναν λαβύρινθο με στόχο να βρει το συντομότερο δυνατό την έξοδο. Σας δίνεται το πρόγραμμα [RobotLabyrinth](#) (δες σχήμα 2.1 παρακάτω) και σας ζητείται να αναπτύξετε το πρόγραμμα *RobotRemoteController*. Αναλυτική περιγραφή θα βρείτε στη συνέχεια.



Σχήμα 2.1. Ο χρήστης ελέγχει το ρομπότ χρησιμοποιώντας συγκεκριμένα πλήκτρα του πληκτρολογίου.

## 2. Στόχος

Στόχος της άσκησης είναι η εξοικείωση με:

1. την περιγραφή βασικών διεργασιών χρησιμοποιώντας λεκτική περιγραφή,
2. τη δόμηση προγράμματος ως συνάθροιση έτοιμων συναρτήσεων,
3. τη δήλωση μεταβλητών και εισαγωγή δεδομένων από την κύρια είσοδο,
4. τη χρήση συναρτήσεων της βασικής βιβλιοθήκης και συναρτήσεων τρίτου κατασκευαστή,
5. τις προτάσεις ελέγχου ροής προγράμματος,
6. την τεχνική της αυξητικής ανάπτυξης προγράμματος.

## 3. Γενικές Οδηγίες

Για την αποτελεσματική προετοιμασία και εκτέλεση της άσκησης σας δίνονται τα παρακάτω:

A) Βιβλιοθήκες που περιέχουν χρήσιμες για την ανάπτυξη του προγράμματος συναρτήσεις ([librobot.dll](#), [robot.h](#)).

B) [Τεκμηρίωση των συναρτήσεων](#) που θα πρέπει να χρησιμοποιήσετε για την ανάπτυξη του προγράμματος RobotRemoteController(RRC).

## 4. Περισσότερα για Λαβυρίθους

[Robot in a wooden maze](#): (from Wikipedia)

[The MazeGen program](#): The program generates mazes using three standard algorithms: Depth-first search, Prim's algorithm, and Kruskal's algorithm. (<http://www.mazeworks.com/mazegen/>)

### Maze generation

[WikiPedia Link](#): [http://en.wikipedia.org/wiki/Maze\\_generation\\_algorithm](http://en.wikipedia.org/wiki/Maze_generation_algorithm)  
A modified version of [Prim's algorithm](#) to generate a perfect maze (in java):  
<http://en.wikipedia.org/wiki/Image:Maze.png>

### Maze Solving

[Real-time maze solving](#): <http://www.swallow.co.uk/dash/dash10.htm>  
[Wikipedia link](#): [http://en.wikipedia.org/wiki/Maze#Solving\\_mazes](http://en.wikipedia.org/wiki/Maze#Solving_mazes)

Ένα ενδιαφέρον site για λαβυρίθους: [Think Labyrinth](#)

For [more links](#) on Mazes: <http://www.astrolog.org/labyrnth/maze.htm>

## 5. Μεθοδολογία εκτέλεσης

### 5.1 Η Τεχνική της Αυξητικής Ανάπτυξης

Για την ανάπτυξη του προγράμματος σας στα πλαίσια των εργαστηριακών ασκήσεων ακολουθείστε την τεχνική της αυξητικής ανάπτυξης (incremental development).

Σύμφωνα με την τεχνική αυτή

1. γράφετε ένα τμήμα του προγράμματος που θα εκτελεί ένα μέρος της λειτουργικότητας του,
2. το μεταγλωττίζετε,
3. διορθώνετε τα συντακτικά λάθη,
4. ελέγχετε το εκτελέσιμο
5. διορθώνετε την λειτουργικότητα του αν απαιτείται
6. επαναλαμβάνεται από το βήμα 2 μέχρι την επιτυχή ανάπτυξη της επιλεγείσας λειτουργικότητας.

Τα παραπάνω αποτελούν τα βήματα μιας επανάληψης (iteration). Επαναλαμβάνετε τα βήματα αυτά προσθέτοντας κάθε φορά επιπλέον λειτουργικότητα στο πρόγραμμα.

Εντοπίστε την λειτουργικότητα που θα προσθέσετε σε κάθε επανάληψη και ακολουθείστε αυστηρά την αυξητική τεχνική.

Για ένα παράδειγμα εφαρμογής της τεχνικής αυτής μπορείτε να ανατρέξετε στην άσκηση 4 του κεφαλαίου 11. Μελετήστε την ενότητα 11.4.3 χωρίς να ασχοληθείτε με τις λεπτομέρειες της συγκεκριμένης άσκησης.

Στη συνέχεια και μόνο στα πλαίσια της Δραστηριότητας 5 που ορίζεται στη συνέχεια μελετήστε προσεκτικά την εφαρμογή της τεχνικής από το βήμα 1 της ενότητας 11.4.4 όπου θα βρείτε και σχόλια για την πρόταση

```
while((choice =get_choice()) != EXIT);
```

Όπως μπορείτε να παρατηρήσετε οι δραστηριότητες που σας δίνονται στα πλαίσια της άσκησης αυτής είναι στο πνεύμα της τεχνικής της αυξητικής ανάπτυξης προγράμματος. Προφανώς βέβαια η τεχνική μπορεί να εφαρμοστεί και κατά την διάρκεια της ανάπτυξης της κάθε επιμέρους έκδοσης.

## 5.2 Γενικές Οδηγίες

A) Θα χρησιμοποιήσετε τις συναρτήσεις γενικού σκοπού για επικοινωνία με τον server RobotLabyrinth των οποίων η τεκμηρίωση σας δίνεται στο κείμενο [Τεκμηρίωση συναρτήσεων](#). Στο κείμενο αυτό θα βρείτε και οδηγίες για την χρήση της βιβλιοθήκης librobot.dll που τις περιέχει.

B) Αξιοποιήστε την τεχνική της αυξητικής ανάπτυξης σύμφωνα με την οποία αναπτύσσετε το πρόγραμμα σας σε διαδοχικά βήματα, όπου το κάθε βήμα προσθέτει επιπλέον λειτουργικότητα στο πρόγραμμα σας.

Γ) Στο τέλος της άσκησης σας δίνεται ένα ενδεικτικό πλαίσιο κώδικα (*skeleton code*). Χρησιμοποιήστε το μόνο μετά από την δική σας προσπάθεια.

## 6. Δραστηριότητες Άσκησης RobotLabyrinth

Η άσκηση δίνεται σαν ένα σύνολο από 7 δραστηριότητες που πρέπει να εκτελέσετε μόνοι σας.

[Δραστηριότητα 1](#) – using RobotLabyrinth

[Δραστηριότητα 2](#) – using RRController

[Δραστηριότητα 3](#) – MyRRController – Set in Auto, Incremental development

[Δραστηριότητα 4](#) – MyRRController – Set in Manual, Hard coded navigation

[Δραστηριότητα 5](#) – MyRRController – Set in Manual, user-driven navigation

[Δραστηριότητα 6](#) – MyRRController – Set in Manual, algorithm-driven navigation

[Δραστηριότητα 7](#) – MyRRController – Final menu-driven version

Σε κάθε δραστηριότητα σας δίνονται οδηγίες και βοήθεια ώστε να ξεπεράσετε τα δύσκολα σημεία. Στο τέλος της άσκησης σας δίνονται ορισμένες ενδεικτικές απαντήσεις.



### 6.1 Δραστηριότητα 1

Στόχος της δραστηριότητας είναι η εξοικείωση με το πρόγραμμα **RobotLabyrinth**.

Θα χρησιμοποιήσετε το αρχείο [RobotLabyrinth.zip](#) το οποίο περιέχει:

- α) το εκτελέσιμο αρχείο του προγράμματος RobotLabyrinth που χρησιμοποιήσατε ήδη στα πλαίσια του εργαστηρίου υπ' αριθμόν 1, και
- β) ορισμένα αρχεία λαβυρίνθων.



## 6.2 Δραστηριότητα 2

Στόχος της δραστηριότητας είναι η εξοικείωση με ένα πρωτότυπο του προγράμματος **RobotRemoteController** ([RRController](#))<sup>1</sup>. Μια έκδοση του προγράμματος **RobotRemoteController** θα αναπτύξετε μόνοι σας στις επόμενες δραστηριότητες.



## 6.3 Δραστηριότητα 3

Στόχος είναι η ανάπτυξης μιας πρώτης έκδοσης του προγράμματος **RobotRemoteController** (θα αναφέρεται στη συνέχεια ως **RRController**) η οποία θα έχει περιορισμένη λειτουργικότητα. Απλά θα επιτρέπει στον χρήστη να φορτώνει τον λαβύρινθο της επιλογής του, να εισάγει ένα ρομπότ και να το βάζει σε mode AUTO.

### Οδηγίες

Για την Δραστηριότητα αυτή θα πρέπει να χρησιμοποιήσετε τις συναρτήσεις της βιβλιοθήκης **robot** των οποίων η τεκμηρίωση σας δίνεται στο κείμενο [Τεκμηρίωση συναρτήσεων](#).

Παρακάτω σας δίνονται οι δηλώσεις των συναρτήσεων (function prototypes) της βιβλιοθήκης **robot**.

```
SOCKET connect2Labyrinth(const char *ip, int port);
int insertRobot(SOCKET socket, char *name);
int robotSetMode(SOCKET socket, char mode);
int robotGoAhead(SOCKET socket);
int robotTurnLeft(SOCKET socket);
int robotTurnRight(SOCKET socket);
int robotGetState(SOCKET socket);
int robotGetSensors(SOCKET socket, struct sensors *sens);
int robotStep(SOCKET sock, int n);
int robotGoHome(SOCKET sock);
int robotStop(SOCKET sock);
int destroyRobot(SOCKET sock);
int loadLabyrinth(SOCKET sock, char *fname);
int disconnect(SOCKET sock);
int robotExitFound(SOCKET sock);
```

<sup>1</sup> Το πρόγραμμα **RRController** αναπτύχθηκε από τον Μεταπτυχιακό Φοιτητή Γιώργο Δούκα στα πλαίσια του ΕΠΕΑΕΚ «Αναβάθμιση Προγράμματος Σπουδών»

### Ενέργεια 3.1

Δώστε τη λεκτική περιγραφή του παραπάνω προγράμματος.

Στο τέλος της άσκησης σας δίνεται μια ενδεικτική λεκτική περιγραφή. Χρησιμοποιήστε την μόνο μετά από την δική σας προσπάθεια.

### Ενέργεια 3.2

Αναπτύξτε τον αντίστοιχο C κώδικα και ελέγξτε την λειτουργικότητά του. Αναλυτικές οδηγίες για την εκτέλεση της ενέργειας αυτής σας δίνονται παρακάτω.

#### Αναλυτικές Οδηγίες για την ενέργεια 3.2

Αξιοποιήστε την τεχνική της αυξητικής ανάπτυξης. Για παράδειγμα μπορείτε να ακολουθήσετε την παρακάτω διαδικασία.

#### 1<sup>ο</sup> Βήμα

Γράψτε πρώτα ένα πρόγραμμα που θα παίρνει από τον χρήστη IP address και port number στην μορφή που τα θέλει η συνάρτηση `connect2Labyrinth` που θα χρησιμοποιήσετε στη συνέχεια για την αποκατάσταση της σύνδεσης με τον server `RobotLabyrinth`. **Ελέγξτε αν** οι αντίστοιχες μεταβλητές σας έχουν τις επιθυμητές τιμές.

Στη φάση αυτή χρησιμοποιείτε μόνο συναρτήσεις της Βασικής Βιβλιοθήκης της C. Δεν απαιτείται η χρήση συναρτήσεων τρίτου κατασκευαστή. Άρα δεν χρησιμοποιείτε την βιβλιοθήκη `Robot`.

#### **Ερωτήσεις που θα πρέπει να γνωρίζετε να απαντήσετε**

*Πως χρησιμοποιώ μια συνάρτηση της βασικής βιβλιοθήκης;*

*Πού είναι το `function prototype` μιας συνάρτησης της βασικής βιβλιοθήκης;*

*Γιατί πρέπει να συμπεριλάβω στον κώδικα μου τα συγκεκριμένα αρχεία επικεφαλίδας (`header files`);*

*Σε ποια φάση και από ποιο πρόγραμμα αξιοποιείται το `function prototype` μιας συνάρτησης της βασικής βιβλιοθήκης;*

*Που είναι το σώμα μιας συνάρτησης της βασικής βιβλιοθήκης;*

*Σε ποια φάση και από ποιο πρόγραμμα αξιοποιείται το σώμα της συνάρτησης της βασικής βιβλιοθήκης;*

#### **Συχνά Προβλήματα**

Το σύστημα διακόπτει την διαδικασία εκτέλεσης του προγράμματος (`abnormal termination`) με σχετικό μήνυμα.

Ελέγξτε τον τρόπο χρήσης της `scanf`. Η `scanf` θέλει ως όρισμα την διεύθυνση της μεταβλητής στην οποία θέλετε να αποδώσετε τιμή.

#### 2ο Βήμα

Προχωρήστε στο βήμα αυτό μόνο μετά την επιτυχή εκτέλεση του 1ου Βήματος.

Χρησιμοποιήστε την κατάλληλη συνάρτηση της βιβλιοθήκης `Robot` για να αποκαταστήσετε σύνδεση με τον λαβύρινθο.

Ενημερώστε τον χρήστη για την περίπτωση που η συνάρτηση αποκατάστασης σύνδεσης με τον λαβύρινθο δεν αποκατέστησε την σύνδεση. Εναλλακτικά

ενημερώστε τον χρήστη αν αποκαταστάθηκε η επικοινωνία με το πρόγραμμα RobotLabyrinth.

### **Ερωτήσεις που θα πρέπει να είστε σε θέση να απαντήσετε**

*Πως χρησιμοποιώ μια συνάρτηση βιβλιοθήκης τρίτου κατασκευαστή (third party library);*

*Πού είναι το function prototype μιας συνάρτησης βιβλιοθήκης τρίτου κατασκευαστή;*

*Που είναι το σώμα μιας συνάρτησης βιβλιοθήκης τρίτου κατασκευαστή;*

*Πως θα βρει ο προ-επεξεργαστής (preprocessor) το κατάλληλο header file; Τι θα το κάνει; Και ποιος θα αξιοποιήσει τα περιεχόμενα του;*

*Πως θα βρει ο linker το σώμα της συνάρτησης; Τι θα το κάνει;*

*Ποιος είναι ο σκοπός της αξιοποίησης συναρτήσεων τρίτου κατασκευαστή στα πλαίσια αυτής της άσκησης;*

*Ποιες είναι οι προτάσεις διακλάδωσης υπό συνθήκη της γλώσσας C;*

*Πως χρησιμοποιώ τις προτάσεις διακλάδωσης υπό συνθήκη;*

### **Συχνά Προβλήματα**

*Το Dev-C++ δεν βρίσκει το header file της βιβλιοθήκης που χρησιμοποιώ;*

*Βεβαιωθείτε πως έχετε κατεβάσει το συγκεκριμένο header file και έχετε ενημερώσει σωστά το IDE σας για το που θα το βρει.*

*Το Dev-C++ δεν βρίσκει τη βιβλιοθήκη τρίτου κατασκευαστή (π.χ. Robot) συναρτήσεων της οποίας χρησιμοποιώ;*

*Βεβαιωθείτε πως έχετε κατεβάσει το συγκεκριμένο αρχείο της βιβλιοθήκης και έχετε ενημερώσει σωστά το IDE σας για το που θα το βρει.*

*Το πρόγραμμα RobotLabyrinth δεν αποκρίνεται στις προσταγές του προγράμματος μου.*

*Βεβαιωθείτε πως έχετε ενεργοποιήσει το πρόγραμμα RobotLabyrinth και το έχετε βάλει σε mode remote. Ελέγξτε IP address και port number πριν την κλήση της labConnect.*

## **3<sup>ο</sup> Βήμα**

Μόνο μετά την επιτυχή εκτέλεση του 2ου Βήματος.

Φορτώστε το λαβύρινθο της επιλογής σας και βάλτε το RobotLabyrinth σε mode AUTO. Τερματίζοντας το πρόγραμμα σας διακόψτε την σύνδεση με το πρόγραμμα RobotLabyrinth.

### **Συχνά Προβλήματα**

*Το πρόγραμμα RobotLabyrinth δεν φορτώνει τον συγκεκριμένο λαβύρινθο;*

*Ελέγξτε αν το αρχείο του λαβυρίνθου βρίσκεται στο κατάλληλο ευρετήριο.*



## **6.4 Δραστηριότητα 4**

Στόχος είναι η ανάπτυξης μιας δεύτερη έκδοση του προγράμματος RobotRemoteController η οποία :

1. να φορτώνει τον λαβύρινθο labyrinth6,
2. να εισάγει ένα ρομπότ,
3. να το βάζει σε mode MANUAL και
4. να του στέλνει εντολές κίνησης για να φτάσει μέχρι την έξοδο.

Η έκδοση αυτή δεν θα δέχεται από τον χειριστή εντολές κίνησης αλλά θα πρέπει ο προγραμματιστής να έχει ενσωματώσει στον κώδικα του (hard coded) τις εντολές κίνησης.

## Οδηγίες

### Ενέργεια 4.1

Δώστε τη λεκτική περιγραφή του παραπάνω προγράμματος.

*Στο τέλος της άσκησης σας δίνεται μια ενδεικτική λεκτική περιγραφή. Χρησιμοποιήστε την μόνο μετά από την δική σας προσπάθεια.*

### Ενέργεια 4.2

Αναπτύξτε τον αντίστοιχο C κώδικα και ελέγξτε την λειτουργικότητα του.

Για να προσδιορίσετε την σειρά των εντολών που πρέπει να αποστείλετε στο ρομπότ θα χρειασθεί να κάνετε μερικές δοκιμές.

### Ενέργεια 4.3

Παρατηρήστε την επανάληψη της κλήσης της συνάρτησης robotStep. Προτείνετε ένα τρόπο για την αποφυγή επανειλημμένων κλήσεων της. Αξιοποιήστε σταδιακά τις προτάσεις επανάληψης και την κατασκευή της συνάρτησης προς την κατεύθυνση αυτή.

## Οδηγίες

A) Ανατρέξτε στην ενότητα 6.2.5 του κεφαλαίου 6 «Προτάσεις ελέγχου ροής». Δώστε ιδιαίτερη έμφαση στις προτάσεις επανάληψης στις ενότητες 6.2.3 και 6.2.5.

B) Ανατρέξτε στην ενότητα 5.3 του κεφαλαίου 5. «Αφαιρετικότητα στις διεργασίες» για να δείτε πως μπορείτε να δηλώσετε και να ορίσετε μια δική σας συνάρτηση.

## Αναλυτικές Οδηγίες για την ενέργεια 4.3

### 1<sup>ο</sup> Βήμα

Χρησιμοποιήστε την κατάλληλη πρόταση επανάληψης για να αποφύγετε την επανειλημμένη κλήση της συνάρτησης robotStep.

### **Ερωτήσεις που θα πρέπει να είστε σε θέση να απαντήσετε**

*Ποιες είναι οι προτάσεις επανάληψης της C;*

*Πως χρησιμοποιώ την κάθε πρόταση επανάληψης;*

*Πως επιλέγω την κατάλληλη για την κάθε περίπτωση πρόταση που πρέπει να χρησιμοποιήσω;*

### 2<sup>ο</sup> Βήμα

Βελτιώστε την επαναχρησιμοποίηση κώδικα ορίζοντας την κατάλληλη συνάρτηση.

Χρησιμοποιείστε την έννοια της τυπικής παραμέτρου για να κάνετε τη συνάρτησή σας πιο γενική.

**Ερωτήσεις που θα πρέπει να είστε σε θέση να απαντήσετε**

*Τι είναι συνάρτηση;*

*Γιατί δημιουργώ δικές μου συναρτήσεις στην διαδικασία ανάπτυξης προγράμματος;*

*Πως δημιουργώ μια δική μου συνάρτηση;*

*Τι είναι δήλωση, τι ορισμός και τι κλήση συνάρτησης;*

*Πού βάζω το function prototype της συνάρτησης μου και τι ρόλο αυτό παίζει;*

*Που βάζω το σώμα της συνάρτησης μου και ποιος το αξιοποιεί;*

*Τι είναι τυπική παράμετρος;*

*Τι είναι πραγματική παράμετρος;*

*Τι σημαίνει έλεγχος τύπων για μια συνάρτηση και ποιος την κάνει;*

**Αναφέρατε τα τυχόν προβλήματα σας** στην περιοχή συζητήσεων στην ιστοσελίδα του εργαστηρίου.