

Συνήθως στη συνθήκη χρησιμοποιούνται σχεσιακοί τελεστές (relational operators), δηλαδή συγκρίσεις μεταξύ τιμών, των οποίων το αποτέλεσμα είναι είτε αληθές (Ναι), είτε ψευδές (Όχι):

Τελεστής	Σημαίνει	Παραδείγματα
>	Μεγαλύτερο	$X > 9$
>=	Μεγαλύτερο και ίσο	Ηλικία > = 18
<	Μικρότερο	Ηλικία < 18
<=	Μικρότερο και ίσο	Ακτίνα < = 0
=	Ίσον	Κωδικός = 1
<> ή !=	Άνισο	$Y <> 0$

## 6. Δημιουργία ενός προγράμματος

Η μεγάλη διάδοση των ηλεκτρονικών υπολογιστών οφείλεται σε μεγάλο βαθμό στην ευελιξία τους στο να επιτελούν διάφορες εργασίες, απλά με την επιλογή από διάφορα προγράμματα. Στην πραγματικότητα, κάθε **πρόγραμμα** αποτελείται από μια σειρά από εντολές που απευθύνονται στον υπολογιστή για την πραγματοποίηση κάποιας συγκεκριμένης εργασίας ή την επίλυση κάποιου προβλήματος.

Όπως θα δούμε στο επόμενο κεφάλαιο, η δημιουργία ενός προγράμματος αρχίζει από ένα πρόβλημα, το οποίο επιθυμεί κάποιος να επιλύσει με τη βοήθεια του υπολογιστή. Για να μπορέσει, όμως, κάποιος να μετατρέψει τη λύση του προβλήματος σε πρόγραμμα, θα πρέπει απαραίτητα η περιγραφή της λύσης να αποτελείται από απλά, συγκεκριμένα, βήματα με σαφήνεια και διατυπωμένα με τρόπο που να δίνουν τα επιθυμητά αποτελέσματα, δηλαδή να αποτελεί αλγόριθμο.



Συνήθως χρειάζεται κάποιος χρόνος για την ανάλυση ενός προβλήματος και τη διατύπωση της λύσης του σε αλγόριθμο. Ο αλγόριθμος, μπορεί να μελετηθεί προσεκτικά για να εξακριβωθεί ότι επιλύει ικανοποιητικά το πρόβλημα και στη συνέχεια «μεταφράζεται» από την ανθρώπινη γλώσσα σε **πρόγραμμα**, δηλαδή σε αντίστοιχη σειρά από εντολές στη γλώσσα του υπολογιστή, τις οποίες να μπορεί να εκτελέσει. Η μετάφραση πραγματοποιείται με την διατύπωση του αλγορίθμου χρησιμοποιώντας κάποια **γλώσσα προγραμματισμού**, η οποία είναι μια τεχνητή γλώσσα με συγκεκριμένο συντακτικό και γραμματική που εύκολα μεταφράζεται από ειδικά προγράμματα σε εντολές του υπολογιστή. Περιβάλλοντα όπως το Scratch και Alice περιλαμβάνουν τη δική τους γλώσσα προγραμματισμού, για να μας επιτρέπουν να εκφράσουμε τον αλγόριθμο που επιθυμούμε σε «πρόγραμμα».

Θα μπορούσε κάποιος να χαρακτηρίσει ένα πρόγραμμα ως αλγόριθμο διατυπωμένο με τρόπο που να εκτελείται από τον Ηλεκτρονικό Υπολογιστή.

### Βασικές Έννοιες

**Αλγόριθμος:** Μια συγκεκριμένη σειρά από βήματα ή εντολές τα οποία χρειάζονται για την εκτέλεση μιας εργασίας ή την επίλυση ενός προβλήματος.

#### Τρόποι περιγραφής Αλγόριθμων

**Λεκτική Περιγραφή:** Περιγραφή ενός αλγόριθμου σε βήματα χρησιμοποιώντας λέξεις από την καθημερινή μας γλώσσα.

**Ψευδοκώδικας (pseudocode):** Πιο αυστηρός τρόπος διατύπωσης ενός αλγορίθμου κατά την οποία χρησιμοποιούνται καθορισμένες λέξεις και συμβολισμοί.

**Λογικό Διάγραμμα:** Τρόπος διατύπωσης ενός αλγόριθμου με διάγραμμα, που αποτελείται από σύμβολα με συγκεκριμένη σημασία, που συνδέονται μεταξύ τους με βέλη.

#### Χαρακτηριστικά ενός Αλγόριθμου

**Σαφήνεια/ Καθοριστικότητα:** Κάθε εντολή (βήμα) του αλγόριθμου πρέπει να είναι απλή και ξεκάθαρα διατυπωμένη, ώστε να μην υπάρχει αμφιβολία για τον τρόπο εκτέλεσής της και να μπορεί να πραγματοποιηθεί με ακρίβεια και σε πεπερασμένο (όχι άπειρο) χρόνο.

**Περατότητα:** Ο αλγόριθμος πρέπει πάντα να τελειώνει ύστερα από την εκτέλεση ενός αριθμού βημάτων (να μην υπάρχει η πιθανότητα να συνεχίζει την εκτέλεση για πάντα).

**Αποτελεσματικότητα:** Να δίνει συγκεκριμένα αποτελέσματα, τα οποία να μπορούν να εξακριβωθούν από τον άνθρωπο εκτελώντας με «μολύβι και χαρτί» τις εντολές του αλγόριθμου, οι οποίες θα πρέπει να μπορούν να εκτελεστούν από αυτόν (υπολογιστή, άνθρωπο) στον οποίο απευθύνεται ο αλγόριθμος.

#### Δομές Αλγόριθμων

**Ακολουθιακή Δομή:** Σειρά από βήματα/εντολές τα οποία εκτελούνται το ένα μετά το άλλο.

**Δομή Διακλάδωσης:** Δομή που Περιλαμβάνει τον έλεγχο κάποιας συνθήκης η οποία μπορεί να έχει δύο τιμές (Αληθής/Ναι ή Ψευδής/Όχι) και ανάλογα με το αποτέλεσμα εκτελείται μία από δύο εναλλακτικές σειρές εντολών (ΕΑΝ..ΤΟΤΕ.., ΑΛΛΙΩΣ..).

**Γλώσσα προγραμματισμού:** Γλώσσα, που μοιάζει με την ανθρώπινη γλώσσα και έχει δική της γραμματική και συντακτικό, με στόχο τη διατύπωση ενός αλγορίθμου με τρόπο που να μεταφράζεται αυτόματα σε σειρά από εντολές τις οποίες να μπορεί να εκτελεί ο υπολογιστής.

**Πρόγραμμα:** Μια σειρά από εντολές που όταν εκτελούνται από τον υπολογιστή γίνεται συγκεκριμένη επεξεργασία δεδομένων και παράγονται συγκεκριμένα αποτελέσματα/πληροφορίες.

## **B7.2 Ο Κύκλος Ανάπτυξης Μιας Εφαρμογής (Προγράμματος)**

### **Τι θα μάθουμε σήμερα:**

Ποια βήματα πρέπει να ακολουθούμε για να δημιουργήσουμε ένα πρόγραμμα (Κύκλος Ανάπτυξης Εφαρμογής)

Πώς θα καθορίσουμε με ακρίβεια το πρόβλημα (δηλαδή τι πρέπει να γίνει)

Να αποφασίζουμε και περιγράφουμε τα βήματα/εντολές που χρειάζονται για την επίλυση του προβλήματος (δηλαδή πώς θα γίνει)

Πώς μετατρέπουμε τη σειρά από βήματα/εντολές σε πρόγραμμα που να επιλύει το πρόβλημα

Πώς ελέγχουμε εάν πράγματι το πρόγραμμα λειτουργεί ορθά και λύει το πρόβλημα που καθορίσαμε

Ποια διαδικασία θα πρέπει να ακολουθούμε για να κάνουμε αλλαγές σε ένα πρόγραμμα.

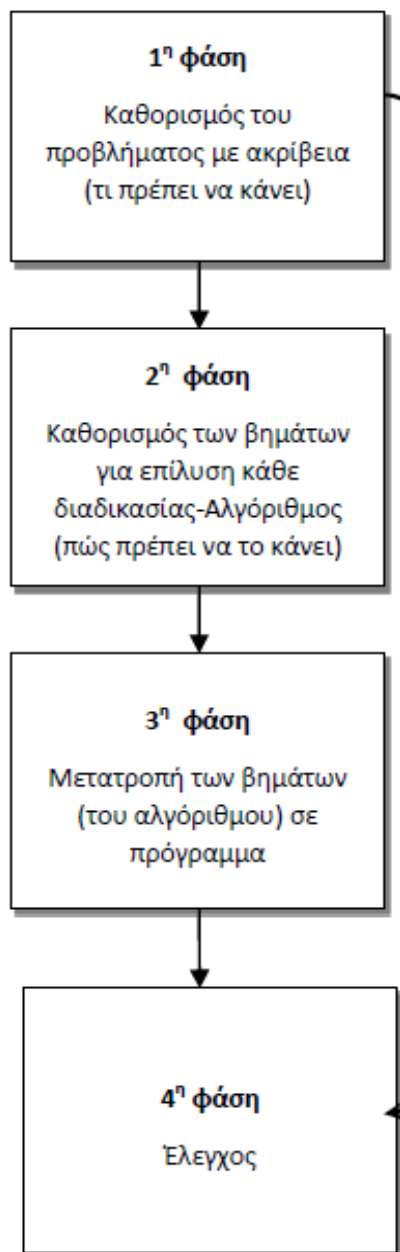
### **1. Εισαγωγή**

Κάθε πρόγραμμα (λογισμικό) που εκτελεί ο υπολογιστής, είτε είναι το λειτουργικό σύστημα, είτε εφαρμογές (π.χ. ο επεξεργαστής κειμένου, ο φυλλομετρητής ιστού, ή ακόμη και ένα παιχνίδι), δεν είναι τίποτα άλλο από μια συγκροτημένη σειρά από εντολές, προσεκτικά τοποθετημένες, που όταν εκτελούνται από τον υπολογιστή έχουν το συγκεκριμένο αποτέλεσμα που βλέπουμε.

### **2. Κύκλος ανάπτυξης εφαρμογής–Ποια βήματα/φάσεις πρέπει να ακολουθήσουμε για να δημιουργήσουμε ένα πρόγραμμα**

Τα προγράμματα τα δημιουργούν οι άνθρωποι και όπως όλες οι τεχνολογικές κατασκευές (π.χ. κτήρια, αυτοκίνητα, γέφυρες, ηλεκτρικές συσκευές), για να λειτουργούν σωστά θα πρέπει να σχεδιαστούν προσεκτικά, να κατασκευαστούν όπως έχουν σχεδιαστεί και να ελεγχθούν πριν τις χρησιμοποιήσουμε. Σκεφτείτε την κατασκευή ενός μεγάλου κτηρίου. Χωρίς διάφορες λεπτομερείς μελέτες, σχεδιαγράμματα από πολιτικούς μηχανικούς, αρχιτέκτονες, ηλεκτρολογικά και πολλά άλλα σχέδια, ο εργολάβος δεν μπορεί καν να ξεκινήσει δουλειά. Για να διορθωθεί μια κατασκευή που δεν έχει σχεδιαστεί σωστά, συνήθως απαιτεί την εξασφάλιση νέων υλικών και πολύ κόπο για την επιδιόρθωσή της, συχνά με ψηλό κόστος, πράγμα που αναγκάζει τον κατασκευαστή να τη σχεδιάσει πολύ προσεκτικά. Το λογισμικό, σε αντίθεση με άλλες κατασκευές, δεν απαιτεί την αγορά υλικών και συχνά δίνεται η εντύπωση ότι δεν απαιτείται σχεδιασμός. Στην πραγματικότητα όμως, το κόστος εντοπισμού και επίλυσης λαθών στο λογισμικό, ιδιαίτερα εάν δεν έχει σχεδιαστεί σωστά, είναι μια δύσκολη, μακροχρόνια και δαπανηρή διαδικασία.

Βλέπουμε λοιπόν ότι όλες οι σοβαρές κατασκευές γίνονται με καλή προεργασία και σχεδιασμό. Το ίδιο συμβαίνει και στην πληροφορική. Για τη δημιουργία ενός προγράμματος (λογισμικού) χρειάζεται να περάσουμε μέσα από μια διαδικασία που περιλαμβάνει διάφορες φάσεις / στάδια τα οποία ονομάζονται Κύκλος ανάπτυξη αλγοριθμικής εφαρμογής.



Εικόνα 80 Τα στάδια του κύκλου ανάπτυξης ενός προγράμματος

Ο λόγος που κατασκευάζουμε ένα πρόγραμμα είναι για να κάνει κάποια συγκεκριμένα πράγματα (διαδικασίες). Σε αυτή τη φάση θα μελετήσουμε καλά το πρόβλημά μας και θα εντοπίσουμε αυτές τις διαδικασίες/λειτουργίες του. Ακολούθως θα περιγράψουμε με σαφήνεια και ακρίβεια **τι θα κάνει** η καθεμιά από αυτές τις διαδικασίες.

Καθεμιά από τις διαδικασίες που εντοπίσαμε και περιγράψαμε στο προηγούμενο στάδιο θα αναλυθεί σε απλά βήματα. Τα βήματα αυτά θα τοποθετηθούν στη σωστή σειρά και θα δείχνουν ξεκάθαρα **πώς θα πραγματοποιείται** η κάθε διαδικασία.

Στην προηγούμενη φάση αναλύσαμε την κάθε διαδικασία σε απλά βήματα. Τώρα θα χρησιμοποιήσουμε μια γλώσσα προγραμματισμού για να μετατρέψουμε το κάθε βήμα των διαδικασιών της προηγούμενης φάσης στην αντίστοιχη εντολή της γλώσσας προγραμματισμού. Με αυτό τον τρόπο, θα έχουμε δημιουργήσει ένα πρόγραμμα που θα μπορεί να εκτελείται από τον υπολογιστή.

Αφού κατασκευάσουμε το πρόγραμμά μας, θα πρέπει να ελέγξουμε κατά πόσο κάνει αυτά που καταγράψαμε στη 2η φάση. Αν υπάρχουν λάθη ή παραλείψεις πρέπει να διορθωθούν ακολουθώντας τη διαδικασία που περιγράφεται πιο κάτω.

### 3. Διαδικασία διόρθωσης λαθών και παραλείψεων

Αν έχουμε εντοπίσει κάποια παράλειψη ή λάθος τότε θα πρέπει να πάμε ξανά στην αρχή του κύκλου, στη 1η φάση και να συμπληρώσουμε τα νέα στοιχεία ή να κάνουμε τις αλλαγές που χρειάζονται. Ακολούθως, θα περάσουμε από όλες τις άλλες φάσεις και θα συμπληρώσουμε ή θα κάνουμε αλλαγές αναλόγως. Στο τέλος θα ελέγξουμε ξανά το πρόγραμμά μας για να δούμε αν λειτουργεί σωστά. Όσες φορές χρειαστούν αλλαγές και διορθώσεις η διαδικασία (**κύκλος ανάπτυξης**) θα επαναλαμβάνεται ξανά και ξανά μέχρι να φτάσουμε σε επιθυμητό αποτέλεσμα.

### Βασικές Έννοιες

<b>Κύκλος Ανάπτυξης Αλγοριθμικής Εφαρμογής:</b>	Τα βήματα/φάσεις που πρέπει να ακολουθήσουμε για να δημιουργήσουμε ένα πρόγραμμα.
<b>1<sup>η</sup> Φάση:</b> <b>Καθορισμός του προβλήματος:</b>	Είναι η 1 <sup>η</sup> φάση του κύκλου ανάπτυξης κατά τη διάρκεια της οποίας θα μελετήσουμε καλά το πρόβλημα μας, θα εντοπίσουμε τις διαδικασίες /λειτουργίες που το αποτελούν και θα περιγράψουμε με σαφήνεια και ακρίβεια <b>τι θα κάνει</b> η καθεμιά από αυτές τις διαδικασίες.
<b>2<sup>η</sup> Φάση:</b> <b>Καθορισμός των βημάτων κάθε διαδικασίας:</b>	Είναι η 2 <sup>η</sup> φάση του κύκλου ανάπτυξης κατά τη διάρκεια της οποίας κάθε διαδικασία θα αναλυθεί σε απλά βήματα. Τα βήματα αυτά θα τοποθετηθούν στη σωστή σειρά και θα δείχνουν ξεκάθαρα <b>πώς θα πραγματοποιείται</b> η κάθε διαδικασία.
<b>3<sup>η</sup> Φάση:</b> <b>Μετατροπή των βημάτων σε πρόγραμμα:</b>	Είναι η 3 <sup>η</sup> φάση του κύκλου ανάπτυξης κατά τη διάρκεια της οποίας θα μετατρέψουμε τα βήματα των διαδικασιών της προηγούμενης φάσης σε εντολές μιας γλώσσας προγραμματισμού.
<b>4<sup>η</sup> Φάση:</b> <b>Έλεγχος:</b>	Είναι η τελευταία φάση του κύκλου ανάπτυξης κατά τη διάρκεια της οποίας θα ελέγξουμε κατά πόσο το πρόγραμμά μας κάνει αυτά που καταγράψαμε στη 2 <sup>η</sup> φάση.
<b>Διαδικασία διόρθωσης λαθών και παραλείψεων:</b>	Είναι η διαδικασία που πρέπει να ακολουθηθεί για διόρθωση τυχόν λαθών ή παραλείψεων που εντοπίστηκαν κατά τον έλεγχο που έγινε στην τελευταία φάση. Βασικά επαναλαμβάνονται όλες οι φάσεις του κύκλου και γίνονται οι απαραίτητες διορθώσεις στην κάθε φάση.



## B7.3 Ο Κύκλος Ανάπτυξης Μιας Εφαρμογής (Προγράμματος)

Τι θα μάθουμε σήμερα:

Να ενεργοποιούμε το λογισμικό Alice

Να δημιουργούμε έναν νέο τρισδιάστατο κόσμο

Να αναγνωρίζουμε τα κύρια μέρη του περιβάλλοντος του Alice

Να εισαγάγουμε, να αντιγράψουμε, να διαγράψουμε και να τροποποιούμε αντικείμενα

Να καθορίζουμε ιδιότητες για τα αντικείμενα

Να εισαγάγουμε κείμενο

Να εφαρμόσουμε (εισαγάγουμε/διαγράψουμε/τροποποιούμε) ενέργειες/εντολές στα αντικείμενα

Να χρησιμοποιούμε τις δομές «Do In Order»

Να εκτελούμε το πρόγραμμα

Να αποθηκεύουμε το πρόγραμμα

Να ανοίγουμε ένα πρόγραμμα που έχουμε αποθηκεύσει προηγουμένως.

### 1. Εισαγωγή

Το Alice είναι ένα τρισδιάστατο περιβάλλον προγραμματισμού, με το οποίο μπορούμε, χωρίς ιδιαίτερες γνώσεις προγραμματισμού, να δημιουργήσουμε κινούμενα τρισδιάστατα σχέδια για την αφήγηση μιας ιστορίας, σκηνές, διαδραστικά παιχνίδια, ή να δημιουργήσουμε ένα βίντεο και να το ανεβάσουμε στο διαδίκτυο. Το συγκεκριμένο λογισμικό διατίθεται δωρεάν μέσω της ιστοσελίδας <http://www.alice.org>.



### 2. Ενεργοποίηση του Alice

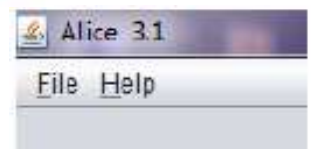
Διπλό κλικ στο εικονίδιο του Alice που βρίσκεται στην επιφάνεια εργασίας.



### 3. Δημιουργία Νέου Κόσμου (New World)

(α) Από το Κυρίως Μενού του προγράμματος κλικ στο **File**

(β) Κλικ στο **New**.



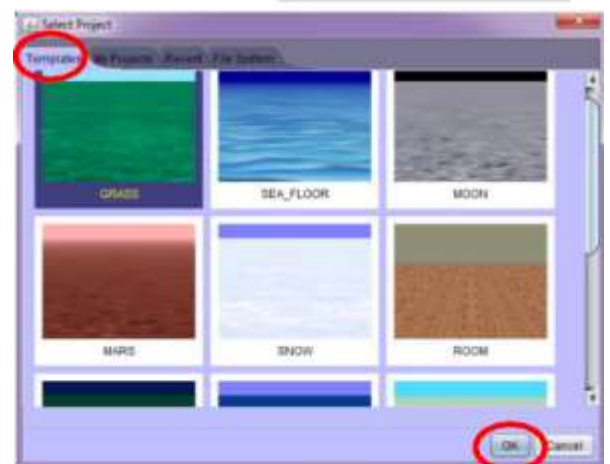
### 4. Επιλογή Αρχικής Σκηνής (Templates)

Ο πραγματικός κόσμος στο Alice ξεκινάει με μια φόρμα αρχικής σκηνής. Οι φόρμες επιλογής παρουσιάζονται στο αρχικό παράθυρο μόλις ξεκινήσει το Alice. Για να καθορίσετε την φόρμα αρχικής σκηνής θα πρέπει να κάνετε τα εξής:

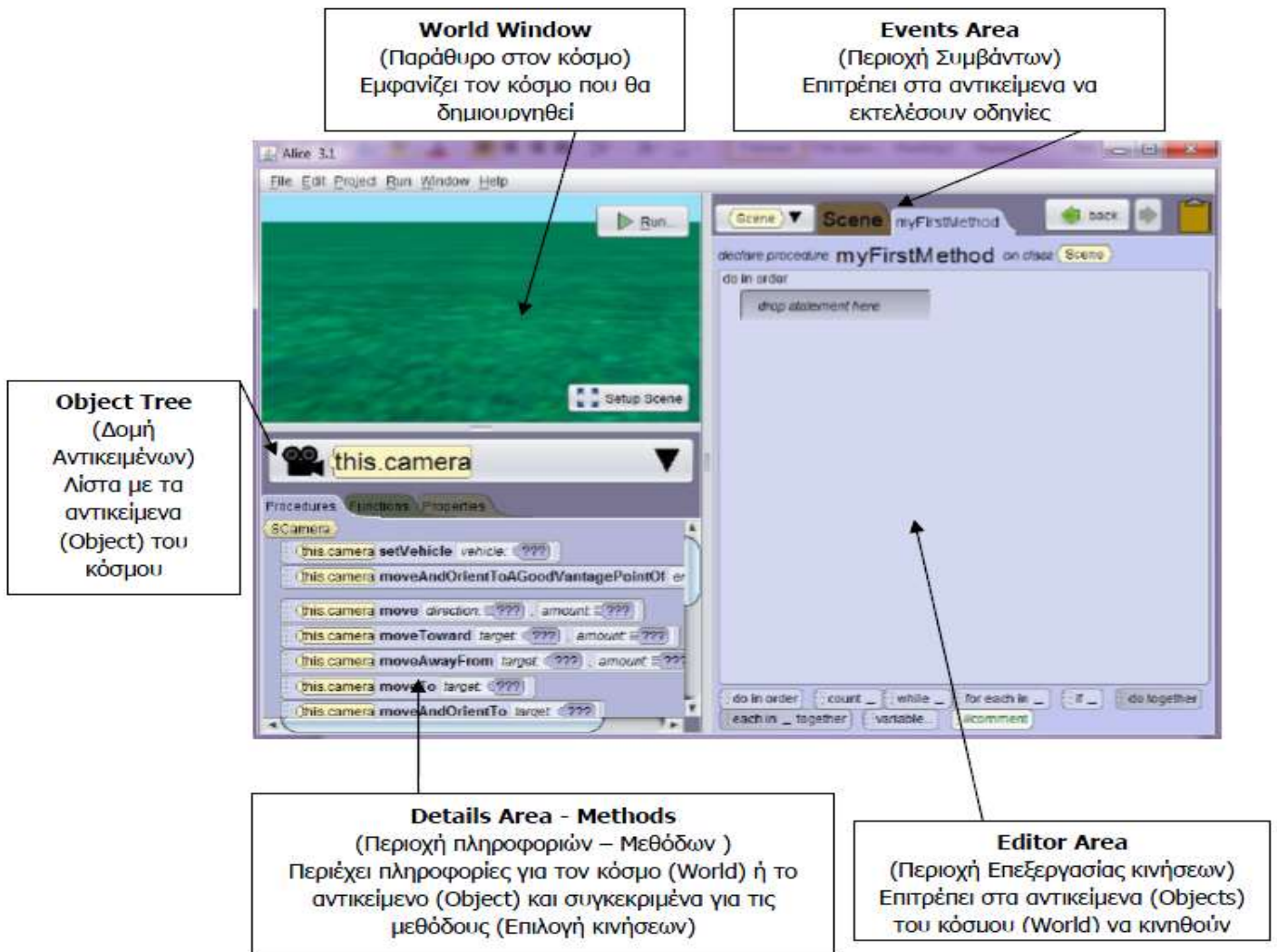
(α) Κλικ στον υποφάκελο **Πρότυπα (Templates)**.

(β) Κλικ σε ένα από τα **Πρότυπα (Templates)** για να το επιλέξετε.

(γ) Κλικ στο **Άνοιγμα (Open)**.



5. Παράθυρο Alice



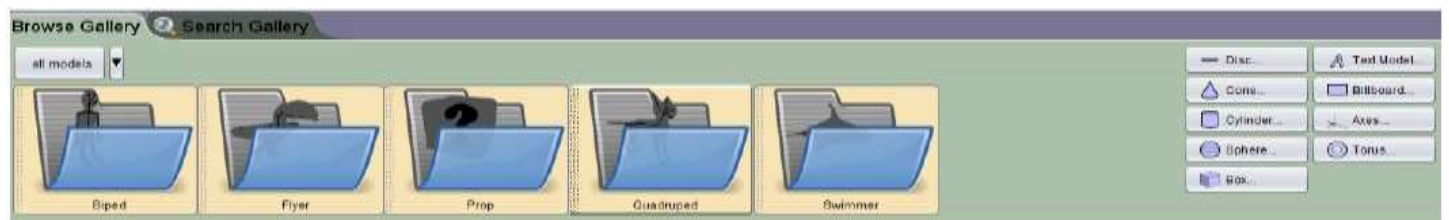
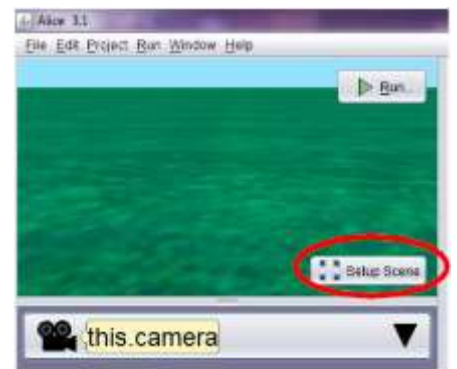
6. Παράθυρο Alice

Για να προσθέσουμε ένα αντικείμενο στη σκηνή μας θα πρέπει να κάνουμε τα εξής:

(α) Κλικ στο κουμπί **Setup Scene**.

(β) Επιλέγουμε το αντικείμενο που θέλουμε να προσθέσουμε από τη **Βιβλιοθήκη αντικειμένων (Gallery)** που εμφανίζεται στο κάτω μέρος της οθόνης:

(γ) Κλικ στην κατηγορία που επιθυμούμε.





(δ) Κλικ στο αντικείμενο.

(ε) Εάν το επιθυμούμε, τροποποιούμε το όνομα του αντικειμένου και κάνουμε κλικ στο **OK**. Μπορώ να προσθέσω όσα αντικείμενα επιθυμώ στον κόσμο που θα δημιουργήσω.



## 7. Τροποποίηση Θέσης και Μεγέθους του Αντικειμένου

Για να τροποποιήσουμε τη θέση και το μέγεθος ενός αντικειμένου πρέπει να κάνουμε τα εξής:

(α) Κλικ πάνω στο αντικείμενο ή κλικ στο αντίστοιχο αντικείμενο στη δομή αντικειμένων για να το επιλέξουμε.

(β) Χρησιμοποιούμε τα **κουμπιά ελέγχου** τα οποία βρίσκονται στη πάνω δεξιά μεριά του παραθύρου για να τοποθετήσουμε το

αντικείμενο στη σωστή θέση και να καθορίσουμε το επιθυμητό μέγεθος.



- Απλή μετακίνηση/περιστροφή αντικειμένου



- Μετακίνηση αντικειμένου



- Περιστροφή αντικειμένου



- Αλλαγή μεγέθους αντικειμένου



## 8. Διαγραφή αντικειμένων (Delete)

(α) Δεξί κλικ στο αντίστοιχο αντικείμενο στη δομή αντικειμένων.

(β) Κλικ στο **Διαγραφή (Delete)**.



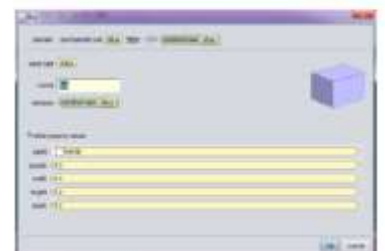
## 9. Προσθήκη σχημάτων στον κόσμο μας

Για να προσθέσουμε σχήματα στον κόσμο μας θα πρέπει να κάνουμε τα εξής:

(α) Κλικ στο σχήμα που θέλουμε το τοποθετήσουμε από το πλαίσιο που βρίσκεται στη κάτω δεξιά γωνιά της οθόνης μας.

(β) Καθορίζουμε τις ιδιότητες που θέλουμε να έχει το σχήμα χρησιμοποιώντας το πλαίσιο διαλόγου που θα εμφανιστεί. Σε περίπτωση που το σχήμα που επιλέξαμε είναι το **Text Model**, μπορούμε χρησιμοποιώντας της ιδιότητα **Value** να καθορίσουμε το κείμενο που επιθυμούμε να εμφανίζεται και με την επιλογή **Paint** να επιλέξουμε το χρώμα των γραμμάτων που επιθυμούμε.

(γ) Κλικ στο OK.



## 10. Καθορισμός Ιδιοτήτων στο αντικείμενο

Το πρόγραμμα μάς δίνει τη δυνατότητα να αλλάξουμε κάποιες από τις ιδιότητες των αντικειμένων που θα εισαγάγουμε, όπως είναι το χρώμα (Color). Για να αλλάξουμε μια ιδιότητα πρέπει να κάνουμε τα εξής:

(α) Ενώ βρισκόμαστε στη σκηνή μας (**Scene**), κλικ πάνω στο αντικείμενο για να το επιλέξουμε.



(β) Στη δεξιά μεριά της οθόνης εμφανίζονται οι ιδιότητες του συγκεκριμένου αντικειμένου τις οποίες μπορούμε εύκολα να τροποποιήσουμε.

## 11. Εφαρμογή ενεργειών στο Αντικείμενο (Methods)

Η πιο κάτω εικόνα παρουσιάζει τον χώρο στον οποίο μπορούμε να εισαγάγουμε τις **ενέργειες/διαδικασίες (Procedures)** τις οποίες θα κάνουν τα διάφορα αντικείμενα. Για να εφαρμόσουμε μία διαδικασία σε ένα αντικείμενο, αφού μεταβούμε στο παράθυρο του κώδικα κάνοντας κλικ στο **Edit Code** και αφού επιλέξουμε τις **διαδικασίες (Procedures)**, πρέπει να κάνουμε τα εξής:



(α) Επιλέγουμε το αντικείμενο

(β) Κλικ στη διαδικασία που επιθυμούμε και χρησιμοποιώντας τη μέθοδο του σύρε και άφησε (drag and drop), τη μεταφέρουμε στη δεξιά μεριά του παραθύρου μας, που είναι η περιοχή επεξεργασίας.

Μερικές από τις ενέργειες που μπορούμε να χρησιμοποιήσουμε είναι οι εξής:

- **Move:** Μετακίνηση αντικειμένου μπροστά, πίσω, πάνω, κάτω, δεξιά, αριστερά
- **Turn:** Περιστροφή αντικειμένου
- **Say:** Εμφάνιση κειμένου
- **Resize:** Αλλαγή μεγέθους αντικειμένου.

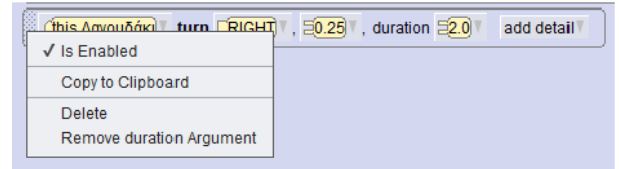


Οι ενέργειες θα εμφανίζονται με τη σειρά που τις τοποθετήσαμε στην περιοχή επεξεργασίας κινήσεων. Μπορούμε να τους αλλάξουμε τη σειρά εκτέλεσης των κινήσεων χρησιμοποιώντας την τεχνική σύρε και άφησε.

## 12. Διαγραφή ενέργειας

Για να διαγράψουμε μια ενέργεια που τοποθετήσαμε σε ένα αντικείμενο πρέπει να κάνουμε τα εξής:

(α) Στην περιοχή επεξεργασίας κινήσεων, κάνουμε δεξί κλικ στην ενέργεια που θέλουμε να διαγράψουμε.



(β) Κλικ στο Διαγραφή (Delete).

## 13. Η Δομή «do together»

Η δομή **do together** επιτρέπει να εκτελούνται ταυτόχρονα πολλές ενέργειες:



Για να χρησιμοποιήσουμε τη δομή **do together**, θα πρέπει να κάνουμε τα εξής:

(α) Χρησιμοποιώντας την τεχνική του Σύρε και Άφησε (Drag and Drop), μεταφέρουμε στην Περιοχή επεξεργασίας τη δομή **do Together**.



(β) Χρησιμοποιώντας την τεχνική του Σύρε και Άφησε (Drag and Drop), μεταφέρουμε μέσα στη δομή Do Together όλες τις ενέργειες που επιθυμούμε να εκτελούνται ταυτόχρονα.

## 14. Η Δομή «do in order»

Η Δομή **do in order** επιτρέπει να εκτελούνται ενέργειες σε σειρά, η μία μετά την άλλη (Ακολουθιακή Δομή).



Για να χρησιμοποιήσουμε τη δομή **do in order**, θα πρέπει να κάνουμε τα εξής:

(α) Χρησιμοποιώντας την τεχνική

του Σύρε και Άφησε (Drag and Drop), μεταφέρουμε στην Περιοχή επεξεργασίας τη δομή **do in order**.

(β) Χρησιμοποιώντας την τεχνική του Σύρε και Άφησε (Drag and Drop), μεταφέρουμε μέσα στη δομή **do in order** όλες τις ενέργειες που επιθυμούμε να εκτελούνται κατά σειρά.



## 15. Η Δομή «If/Else»

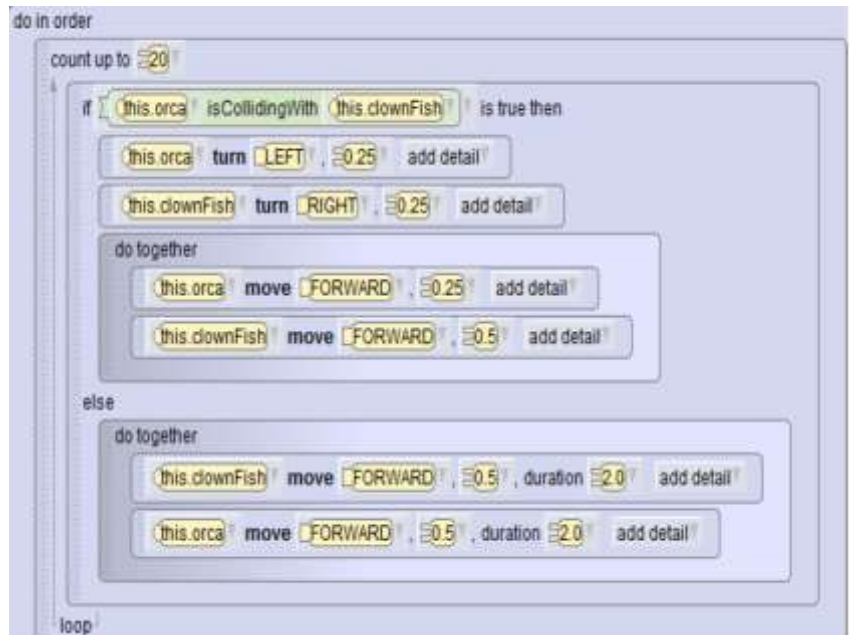
Η Δομή If/Else είναι μια ολοκληρωμένη πρόταση, η οποία περικλείει μια συνθήκη και δύο ομάδες εντολών, από τις οποίες θα εκτελεστούν, η μεν πρώτη ομάδα στην περίπτωση που ικανοποιείται η συνθήκη (Αληθής – True), η δε δεύτερη ομάδα στην περίπτωση που δεν ικανοποιείται η συνθήκη (Ψευδής – False) (Δομή Διακλάδωσης).



Για να χρησιμοποιήσουμε τη δομή **If/Else**, θα πρέπει να κάνουμε τα εξής:

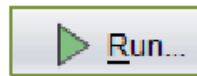
(α) Χρησιμοποιώντας την τεχνική του Σύρε και Άφησε (Drag and Drop), μεταφέρουμε στην Περιοχή επεξεργασίας τη δομή **If/Else**.

(β) Χρησιμοποιώντας την τεχνική του Σύρε και Άφησε (Drag and Drop), μεταφέρουμε μέσα στη δομή **If/Else** όλες τις ενέργειες που επιθυμούμε να εκτελούνται. Συγκεκριμένα στο **if** μεταφέρουμε τη συνθήκη από τη περιοχή **Functions**. Αν ισχύει η συνθήκη που ελέγχουμε (**True**), τότε εκτελείται η πρώτη ομάδα εντολών. Αν όχι (**False**) εκτελείται η δεύτερη ομάδα εντολών, που είναι ενσωματωμένες στο **Else**.



### 16. Εκτέλεση του προγράμματος (Play)

Για να εκτελεστούν όλες οι εντολές που έχουμε δούμε το αποτέλεσμα της εργασίας μας, θα πρέπει



τοποθετήσει σε έναν Κόσμο και να κάνουμε κλικ στο **Run**.

### 17. Αποθήκευση του κόσμου (Save World)

Για να αποθηκεύσουμε έναν κόσμο θα πρέπει να κάνουμε τα εξής:

(α) Από το Κυρίως Μενού κάνουμε κλικ στο **File**.

(β) Κλικ στο Save As.

(γ) Στο πλαίσιο διαλόγου επιλέγουμε τη μονάδα δίσκου και τον φάκελο μέσα στον οποίο θα αποθηκευτεί το αρχείο

(δ) Δακτυλογραφούμε το όνομα του αρχείου.

(ε) Κλικ στο Αποθήκευση (Save).

### 18. Άνοιγμα Αρχείου (Open World)

Για να ανοίξουμε ένα αρχείο θα πρέπει να κάνουμε τα εξής:

(α) Από το Κυρίως Μενού κάνουμε κλικ στο **File**.

(β) Κλικ στο Open.

(γ) Στο πλαίσιο διαλόγου επιλέγουμε τον υποφάκελο **File System**, κάνουμε κλικ στο **browse**, και επιλέγουμε τη μονάδα δίσκου και τον φάκελο μέσα στον οποίο είναι αποθηκευμένο το αρχείο.

**(δ) Κλικ στο όνομα του αρχείου**

(ε) Κλικ στο Άνοιγμα (Open).



## Υπόμνημα

Έναρξη προγράμματος Alice

διπλό κλικ στο



Επιλογή Αρχικής Σκηνής

κλικ σε έναν κόσμο



Προσθήκη αντικειμένων

κλικ στο



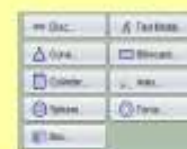
Επιλογή αντικειμένων

κλικ στην κατηγορία που θέλω



Εισαγωγή σχημάτων

κλικ στο σχήμα που θέλω



Επιλογή αντικειμένου για αλλαγές

κλικ στο αντικείμενο



Διαγραφή αντικειμένου

Δεξί κλικ ⇒ Delete

Αλλαγή ονόματος αντικειμένου

Δεξί κλικ ⇒ Rename



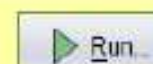
Προσθήκη εντολών  
(από την περιοχή Detail Area στην περιοχή Editor Area)

σύρε κι άφησε



Εκτέλεση ιστορίας:

κλικ στο



Καταγραφή βίντεο

Μενού ⇒ File

Export Video...

Αποθήκευση κόσμου

Μενού ⇒ File

Save World

Ανάκληση κόσμου

Μενού ⇒ File

Open World...

### Υπόμνημα Ερμηνείας Βασικών Εντολών

aliceLiddell	move forward	5 meters	more...
aliceLiddell	turn left	0,5 revolutions	more...
aliceLiddell	roll right	1 revolution	more...
aliceLiddell	resize	0,5	more...
aliceLiddell	say	Goodbye	more...
aliceLiddell	think	What is going on?	more...
aliceLiddell	play sound	world.hi (0:02,450)	more...
aliceLiddell	move to	bob	more...
aliceLiddell	move	amount = 0,5 meters	toward target = bob
aliceLiddell	move	amount = 1 meter	away from target = bob
aliceLiddell	move at speed	up	speed = 1 meter per second
aliceLiddell	roll at speed	left	speed = 0,25 revolutions per second
aliceLiddell	turn at speed	right	speed = 2 revolutions per second
aliceLiddell	turn to face	bob	more...
aliceLiddell	orient to	bob	more...
aliceLiddell	point at	bob	more...
aliceLiddell	constrain to face	bob	more...
aliceLiddell	constrain to point at	bob	more...

Κίνηση αντικειμένου προς συγκεκριμένη κατεύθυνση για ορισμένη απόσταση.

Περιστροφή αντικειμένου προς συγκεκριμένη κατεύθυνση για ορισμένο αριθμό περιστροφών.

Κύλιση αντικειμένου προς συγκεκριμένη κατεύθυνση για ορισμένο αριθμό περιστροφών.

Αλλαγή μεγέθους του αντικειμένου.

Το αντικείμενο εμφανίζει κάποιο μήνυμα.

Το αντικείμενο σκέφτεται κάτι.

Το αντικείμενο παίζει τον επιλεγμένο ήχο.

Κίνηση αντικειμένου προς ένα άλλο αντικείμενο.

Κίνηση αντικειμένου για καθορισμένη απόσταση προς ένα άλλο αντικείμενο.

Κίνηση αντικειμένου για καθορισμένη απόσταση μακριά από ένα άλλο αντικείμενο.

Κίνηση/Περιστροφή/Κύλιση αντικειμένου προς συγκεκριμένη κατεύθυνση με την καθορισμένη ταχύτητα.

Το αντικείμενο περιστρέφεται κατακόρυφα και προσανατολίζεται για να βλέπει το άλλο αντικείμενο.

Το αντικείμενο αλλάζει προσανατολισμό και βλέπει όπου βλέπει και το άλλο αντικείμενο.

Το αντικείμενο περιστρέφεται και προσανατολίζεται για να βλέπει το άλλο αντικείμενο.

Παρόμοιες με τις εντολές turn to face και point at, αλλά η κίνηση γίνεται άμεσα (δεν παίρνει χρόνο).

### Δομές προγραμματισμού

Do in order	aliceLiddell	roll right	1 revolution	more...
	aliceLiddell	say	Goodbye	more...
Do together	aliceLiddell	move forward	5 meters	more...
	aliceLiddell	turn left	0,5 revolutions	more...
If	aliceLiddell	distance to bob	< 2	
	aliceLiddell	think	What is going on?	more...
Else	aliceLiddell	move to	bob	more...
	aliceLiddell	resize	0,5	more...

#### Ακολουθιακή Δομή:

Οι εντολές που περιλαμβάνει εκτελούνται μία μία με τη σειρά.

Οι εντολές που περιλαμβάνει εκτελούνται όλες μαζί ταυτόχρονα.

#### Δομή διακλάδωσης IF/ELSE:

Εάν το αποτέλεσμα της συνθήκης είναι αληθές, τότε εκτελείται η πρώτη σειρά εντολών, διαφορετικά εκτελείται η δεύτερη σειρά εντολών. Στην κατηγορία functions του αντικειμένου World υπάρχουν συνθήκες για να επιλέξετε.