

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων

ΑΞΙΟΠΟΙΗΣΗ ΤΟΥ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ ALICE ΓΙΑ ΤΗΝ  
ΕΚΜΑΘΗΣΗ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕΣΩ  
ΔΗΜΙΟΥΡΓΙΑΣ ΜΙΚΡΟΚΟΣΜΩΝ

Βατουσιανού Ελισάβετ

Μεταπτυχιακή Διπλωματική Εργασία



00161174

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ	
ΑΡ. ΕΙΣ.	61174 + CD
ΣΟΜΠ.	42486
ΤΑΣΗ	005.133 ΒΑΤ
ΒΙΒΛΙΟΘΗΚΗ	

Ιούνιος 2009

*Αφιερώνεται στην οικογένειά μου,  
στο Στάθη,  
και στους αγαπημένους μου φίλους  
που με βοηθούν πάντα να ανακαλύπτω καινούριους ορίζοντες...*

Πανεπιστήμιο Πειραιώς

## Περίληψη

Στην εκπαιδευτική διαδικασία ανακύπτει συνεχώς η ανάγκη για την ανάπτυξη εξειδικευμένων εκπαιδευτικών εργαλείων για την υποστήριξη της διδασκαλίας του Προγραμματισμού. Οι έρευνες αυτές έχουν στόχο να ανακαλύψουν τρόπους να κάνουν τον προγραμματισμό πιο προσιτό στους μαθητές. Η παρούσα εργασία αποσκοπεί στην παράθεση των πιο διαδεδομένων εργαλείων που έχουν αναπτυχθεί για την υποστήριξη της μαθησιακής διαδικασίας του Αντικειμενοστραφούς Προγραμματισμού. Τα εργαλεία και περιβάλλοντα αυτά συγκρίνονται. Στη συγκεκριμένη εργασία αναλύεται το εργαλείο ALICE και αναφέρονται οι λόγοι για τους οποίους το ALICE βοηθά περισσότερο στην εξοικείωση των μαθητών με τον προγραμματισμό.

Στόχος της διπλωματικής είναι να ερευνηθεί αν το Alice μπορεί να προσεγγίσει ευχάριστα τους μαθητές της έκτης δημοτικού, αν τους βοηθά να κατανοήσουν ευκολότερα τον προγραμματισμό και αν προσεγγίζει πιο ευχάριστα τους μαθητές

Πανεπιστήμιο Πελοποννήσου

## Ευχαριστίες

Θερμές ευχαριστίες εκφράζω στον Επίκουρο Καθηγητή κο Συμεών Ρετάλη για την επίβλεψη και τη βοήθεια που μου παρείχε για την ολοκλήρωση της διπλωματικής μου.

Τέλος εκφράζω την ευγνωμοσύνη μου στους γονείς μου για την υποστήριξη και βοήθειά τους σε όλη τη διάρκεια των μεταπτυχιακών σπουδών μου.

Πανεπιστήμιο Πειραιώς

## ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη .....	3
Ευχαριστίες .....	4
Συντομογραφίες .....	6
ΚΕΦΑΛΑΙΟ 1 .....	7
Εισαγωγή.....	7
1.1. Εισαγωγή.....	7
1.2 Η θετική επίδραση των εκπαιδευτικών εργαλείων .....	10
1.3 Η συμβολή των εργαλείων του υπολογιστή στην αναμόρφωση της μαθησιακής/διδασκτικής διαδικασίας .....	21
ΚΕΦΑΛΑΙΟ 2.....	27
Μικρόκοσμοι στην εκπαίδευση .....	27
2.1 Οι μικρόκοσμοι στην εκπαίδευση.....	27
2.2 Η προτιθέμενη αξία των μικρόκοσμων στη διδασκαλία του Αντικειμενοστραφή Προγραμματισμού.....	30
2.3 Περιβάλλοντα μικρόκοσμων στη διδασκαλία του Αντικειμενοστραφή Προγραμματισμού.....	33
2.3.1 Ο μικρόκοσμος Karel .....	37
2.3.2 Ο μικρόκοσμος Jeroo .....	49
2.3.3 Ο μικρόκοσμος της LOGO .....	50
2.3.4 Ο μικρόκοσμος Alice .....	57
ΚΕΦΑΛΑΙΟ 3.....	58
ΤΟ ΠΕΡΙΒΑΛΛΟΝ ALICE.....	58
3.1 Πλεονεκτήματα του Alice.....	58
3.2 Έρευνες και δημοσιεύσεις για το Alice .....	65
3.3 Πώς θα μάθετε να προγραμματίζετε με το Alice .....	80
Σχεδιασμός και υλοποίηση προγράμματος .....	90
ΚΕΦΑΛΑΙΟ 4.....	97
Ο μικρόκοσμος COSY-Java.....	97
4.1 Εκπαιδευτικό σενάριο.....	97
ΣΥΜΠΕΡΑΣΜΑΤΑ .....	105
Βιβλιογραφικές Αναφορές .....	115

## Συντομογραφίες

### Λατινικές

EA - Electronic Arts

IC - Ithaca College

SJU - Saint Joseph's University

### Ελληνικές

ΑΠ - Αντικειμενοστραφής Προγραμματισμός

Η/Υ - Ηλεκτρονικός Υπολογιστής

# ΚΕΦΑΛΑΙΟ 1

## Εισαγωγή

### 1.1. Εισαγωγή

Στην εκπαίδευση ο υπολογιστής μπορεί να χρησιμοποιηθεί ως πολυαισθητηριακό και δυναμικό μέσο διδασκαλίας σε όλα τα μαθήματα από τη γλώσσα και τα μαθηματικά μέχρι και τις τέχνες. Η εκπαιδευτική αυτή λειτουργία του υπολογιστή έχει συζητηθεί ευρέως στη σύγχρονη διεθνή βιβλιογραφία σε σημείο που να μη μιλάμε πλέον για διδακτικό μέσον, αλλά για τη δημιουργία πολλών ειδών διδακτικού περιβάλλοντος.

Εκτός από την προώθηση του αλφαριθμητισμού, οι υπολογιστές μπορούν να χρησιμοποιηθούν και για τη διευκόλυνση και προώθηση σημαντικών στόχων της μάθησης στους τομείς όλων των γνωστικών αντικειμένων. Αυτό μπορεί να γίνει με τη χρήση εκπαιδευτικού λογισμικού, που δεν θα αποτελεί μόνον μία εναλλακτική μορφή βιβλίου σε ηλεκτρονική μορφή ή άσκηση στην απομνημόνευση γνώσεων (που και αυτό είναι βέβαια αναγκαίο), αλλά και θα προωθεί μέσα από τις κατάλληλες μεθόδους νέες δυνατότητες μάθησης.

Φυσικά, ο τομέας της παραγωγής τέτοιου πρωτότυπου εκπαιδευτικού λογισμικού είναι ακόμα υπό ανάπτυξη και υπάρχει και το πρόβλημα του εφοδιασμού των σχολείων με την κατάλληλη τεχνολογική υποδομή, καθώς και της ενημέρωσης και πρόσβασης των εκπαιδευτικών στο υπάρχον εκπαιδευτικό λογισμικό. Ούτως ή άλλως όμως ένα μέλημα του σχολείου και κάθε εκπαιδευτικού είναι η βελτίωση της ποιότητας της μαθησιακής

διαδικασίας, η αξιοποίηση των ευεργετημάτων όλων των διαθέσιμων μέσων και μεθόδων της εκπαιδευτικής τεχνολογίας και ο προβληματισμός σχετικά με τα ζητήματα αυτά.

Υπάρχει όμως η ελπιδοφόρα προσδοκία ότι με την κατασκευή πρωτότυπων προγραμμάτων εκπαιδευτικού λογισμικού θα δημιουργηθούν νέες ευκαιρίες προώθησης και στήριξης διαδικασιών (με τις οποίες ο δάσκαλος δεν είναι αρκετά εξοικειωμένος) ή στόχων, τους οποίους οι εκπαιδευτικοί δυσκολεύονται να επιτύχουν υπό τις παρούσες συνθήκες της σχολικής πραγματικότητας. Τέτοιοι στόχοι είναι, παραδείγματος χάριν, η εξατομικευμένη διδασκαλία, η όσο γίνεται πιο πολυδιάστατη και ενεργός συμμετοχή των μαθητών στη διαδικασία της μάθησης, η αύξηση της παραγωγικότητας, το χτίσιμο της μάθησης κατά τρόπο, ώστε οι μαθητές να αφομοιώνουν και να δημιουργούν τη νέα γνώση, η διευκόλυνσή τους, ώστε να είναι σε θέση να λειτουργούν σε πιο προωθημένα επίπεδα συμβολικής, κριτικής και δημιουργικής σκέψης, να αναπτύσσουν ολόπλευρα την προσωπικότητά τους.

Όπως θα φανεί και σε επόμενα κεφάλαια, η υποβοηθούμενη από τον υπολογιστή διδασκαλία είναι σε θέση να στηρίξει την επιδίωξη τέτοιου είδους στόχων. Παρόλο που αρχικά υπήρχε ο φόβος ότι ο υπολογιστής θα μηχανοποιούσε τη γνώση και θα επέβαλε πρότυπα προγραμματισμένης μάθησης, ιδιαίτερα προσφιλή στους συμπεριφοριστές και τους ακραίους τεχνοκράτες, υπάρχουν βάσιμες προσδοκίες ότι η κατάλληλα σχεδιασμένη διδασκαλία, με τη βοήθεια προγραμμάτων του υπολογιστή, με την αξιοποίηση και άλλων μέσων, μπορεί να ενισχύσει την υιοθέτηση διδακτικών καινοτομιών και να φέρει μια μικρή επανάσταση στη μαθησιακή διαδικασία. Μία από τις «επαναστάσεις», που μπορεί να επιφέρει η χρήση των υπολογιστών, είναι και η αλλαγή του κοινωνικού κλίματος, των αλληλεπιδράσεων και των σχέσεων μέσα στη σχολική τάξη, όπου η επικοινωνία και η



συνεργασία ανάμεσα σε μαθητές θα είναι επιθυμητή και ο ρόλος του δασκάλου περισσότερο συντονιστικός και διευκολυντικός και λιγότερο εκείνος του προμηθευτή όλων των γνώσεων και των πληροφοριών.

Και σ' αυτή την περίπτωση βέβαια, ο υπολογιστής από μόνος του δεν μπορεί να κάνει πολλά πράγματα, αν αυτά δεν ταιριάζουν με τη φιλοσοφία των δασκάλων, των μαθητών και της κυρίαρχης ιδεολογίας γενικότερα, αν δεν υπάρχει η κατάλληλη βούληση και η ετοιμότητα για την υιοθέτηση των σχετικών καινοτομιών. Η εμπειρία έχει δείξει ότι η κουλτούρα του σχολείου στις μικρές ηλικίες (οπότε δίνεται περισσότερη έμφαση σε κριτήρια και στόχους παιδαγωγικής φύσεως) προσφέρεται πολύ περισσότερο για νεωτεριστικές και για τις λεγόμενες προοδευτικές εφαρμογές.

Εκεί επίσης όπου, κατά γενική ομολογία, έχουν διαπιστωθεί εφαρμογές με εντυπωσιακά αποτελέσματα, είναι η περιοχή της διδασκαλίας παιδιών με μαθησιακές δυσκολίες. Πολλές μάλιστα εφαρμογές έχουν αποδειχθεί χρήσιμες και για τα παιδιά χωρίς ιδιαίτερα μαθησιακά προβλήματα, αφού η γνώση που έχει αποκτηθεί από τις εφαρμογές αυτές μπορεί να βοηθήσει στην επίτευξη της μάθησής τους, παραδείγματος χάριν, της ανάγνωσης και της γραφής, της κατανόησης μαθηματικών εννοιών και σχέσεων, ώστε να είναι σε θέση να χρησιμοποιήσουν τις δεξιότητες αυτές με περισσότερη άνεση, να ανοίξουν δρόμους για παραπέρα μαθησιακές εμπειρίες και να έχουν στο ενεργητικό της μαθητικής τους ζωής περισσότερες εμπειρίες επιτυχίας.

## 1.2 Η θετική επίδραση των εκπαιδευτικών εργαλείων

### α. Η ιδιαιτερότητα του υπολογιστή

Ένα ερώτημα που τίθεται από πολλούς εκπαιδευτικούς και γονείς είναι: σε τι διαφέρει ο υπολογιστής από τα άλλα μέσα διδασκαλίας, όπως η τηλεόραση, το βίντεο, τα σλάιντς, το κασετόφωνο, το βιβλίο κτλ.; Είναι γεγονός ότι ο υπολογιστής, εφόσον διαθέτει το κατάλληλο λογισμικό, έχει ορισμένα επί πλέον πλεονεκτήματα σε σύγκριση ακόμη και με άλλα ηλεκτρονικά μέσα διδασκαλίας, παρέχοντας στους μαθητές δυνατότητες, όπως θα δούμε και πιο κάτω, στην περιγραφή της χρήσης του υπολογιστή ως γνωστικού εργαλείου.

Μία απορία, που συνήθως δημιουργείται, όταν τονίζουμε την ανάγκη εκτίμησης της παιδαγωγικής αξίας του υπολογιστή είναι: «μα τα ίδια δεν ισχύουν και για το βιβλίο;» Τα ίδια βέβαια ισχύουν και για τα βιβλία, υπάρχουν όμως και ορισμένες ιδιαιτερότητες του υπολογιστή, αλλά και προτερήματα του βιβλίου, που θα πρέπει να μας κάνουν πολύ προσεκτικούς κατά την αξιολόγηση και επιλογή ενός εκπαιδευτικού λογισμικού ή για μια γενικευμένη χρήση του στο σχολείο.

Κατ' αρχήν, υπάρχουν διαφορετικές διεργασίες που εμπλέκονται στην επικοινωνία μέσω του γραπτού και προφορικού λόγου, από τη μια μεριά, και του εκπαιδευτικού λογισμικού, από την άλλη, ως πηγών μάθησης. Μέσα από την ανάγνωση ενός βιβλίου ο αναγνώστης προσπαθεί να μετασχηματίσει τα νοήματα του κειμένου, ώστε να γίνουν κατανοητά, συσχετίζοντάς τα με τις ενεργές τη στιγμή εκείνη αναπαραστάσεις του χρησιμοποιώντας την αναλογική του σκέψη κτλ. Κατά την επικοινωνία συγγραφέα-αναγνώστη ο συγγραφέας κωδικοποιεί και εκπέμπει, ενώ ο αναγνώστης αποκωδικοποιεί και συλλογίζεται, έχοντας ως μοναδικό μέσο τη λεκτική αναπαράσταση (που εκ των

πραγμάτων είναι αρκετά αφηρημένη) χωρίς τη διαμεσολάβηση άλλου διευκολυντικού μέσου ή μιας ευκαιρίας για αλληλεπίδραση με το συγγραφέα. Η ανάγνωση μέσω του υπολογιστή, από την άλλη πλευρά, είναι δυνατόν να διευκολυνθεί με την *παρέμβαση* ορισμένων και άλλων διαμεσολαβητικών εργαλείων και τρόπων αναπαράστασης νοημάτων, πέραν της λεκτικής σε μορφή ενός γραμμικού, γραπτού λόγου. (Και στις δύο περιπτώσεις βέβαια η κατάσταση είναι πολύ διαφορετική, ως προς την αμοιβαιότητα, την αμεσότητα και τα άλλα χαρακτηριστικά της «συνομιλίας», σε σχέση με τη φυσική επικοινωνία μέσω του προφορικού λόγου). Επίσης, και οι δύο μορφές λόγου, ο έντυπος και ο προφορικός, είναι δυνατόν να ενσωματωθούν στον υπολογιστή παρέχοντας επί πλέον στο συγγραφέα των κειμένων ενός εκπαιδευτικού λογισμικού δυνατότητα να επικοινωνήσει με τον αναγνώστη μέσω του λόγου, της εικόνας, κινούμενης ή μη, των ήχων και διαφόρων άλλων συμβολικών τρόπων. (Είναι σε όλους μας γνωστή η δύναμη της οπτικοποίησης, της ακουστικής αντίληψης και της παρατήρησης της εξέλιξης ορισμένων εννοιών και φαινομένων).

Με τη μη γραμμική διάταξη του λόγου επίσης και την *ουσιαστική* (και όχι απλώς διακοσμητική) συμμετοχή της εικόνας και του ήχου, τα πολυαισθητηριακά κείμενα διευκολύνουν τη χρήση πολλαπλών τρόπων αναπαράστασης της πληροφορίας και της γνώσης και την οικοδόμηση προσωπικού νοήματος, σε αντίθεση με άλλες παραδοσιακές εκπαιδευτικές πρακτικές, που χαρακτηρίζονται από έμφαση στον αφηρημένο και λεκτικό κυρίως τρόπο αναπαράστασης της γνώσης. Παρόλο που συχνά υποστηρίζεται ότι η παρεμβολή της εικόνας δεσμεύει μάλλον, παρά απελευθερώνει την αναπαραστασιακή ικανότητα του μαθητευόμενου, όπως συμβαίνει με την εικονογράφηση ενός λογοτεχνικού έργου, υπάρχουν περιπτώσεις, που η οπτικοποίηση ορισμένων

αφηρημένων ή επιστημονικών εννοιών (π.χ. στην ιστορία, τη γεωγραφία, τις φυσικές επιστήμες κτλ), η πολυαισθητηριακή και δυναμική μοντελοποίηση ή προσομοίωση των σχέσεων ανάμεσα στα στοιχεία και δεδομένα ενός φαινομένου ή προβλήματος αποτελεί ένα από τα σημαντικότερα πλεονεκτήματα κατανόησης εννοιών και μάθησης μέσω του υπολογιστή, όπως θα δούμε και πιο κάτω στα επόμενα κεφάλαια.

Ένα άλλο χαρακτηριστικό γνώρισμα της ανάγνωσης στον ψηφιακό κόσμο είναι επίσης ότι «ο χώρος των πληροφοριών δεν περιορίζεται σε τρεις διαστάσεις και η απόδοση μιας ιδέας ή η διαδρομή της σκέψης μπορεί να περιλαμβάνει ένα πολυδιάστατο δίκτυο από σημεία, που οδηγούν σε περαιτέρω επεξεργασία και σύνθεση ή διαφωνία και απόρριψη από μέρους του χρήστη» (Κόμης, 1996:79), ο μαθητευόμενος μερικές φορές δεν είναι απλώς ένας αναγνώστης, αλλά γίνεται *μέτοχος στη διαδικασία οργάνωσης μηνυμάτων και διαδικασιών σκέψης*, κάτι που ευνοεί την ανάπτυξη της αυτόνομης μάθησης και ανάπτυξης του μαθητευόμενου (Spiro et al, 1992). Αυτή τη δυνατότητα βέβαια δεν την προσφέρουν όλων των ειδών τα λογισμικά, όμως με τα κατάλληλα σχεδιασμένα υπερκείμενα και υπερμέσα δίνονται πολλές ευκαιρίες για μια αυτόνομη και ανεξάρτητη επεξεργασία των μηνυμάτων, η οποία αναβαθμίζεται γνωστικά με τη διαμεσολάβηση των διδασκόντων και των συνομηλίκων.

Η πιο σημαντική ίσως διαφορά του υπολογιστή από άλλα μέσα διδασκαλίας, όπως είναι το βιβλίο, το χαρτί και το μολύβι, το φιλμ κ.ά., είναι το ότι μπορεί και αλληλεπιδρά με τον αναγνώστη, γι' αυτό και συγκαταλέγεται στα *δυναμικά εργαλεία* διερευνητικής και δοκιμαστικής μοντελοποίησης φαινομένων και προβλημάτων, ακόμη και μοντελοποίησης της σκέψης του μαθητή, με τη βοήθεια των επιτευγμάτων της τεχνητής νοημοσύνης.

Οι εκπαιδευτικοί, στην πλειοψηφία τους, δεν γνωρίζουν πώς ο υπολογιστής μπορεί να χρησιμοποιηθεί ως **γνωστικό εργαλείο** σε όλα τα σχολικά μαθήματα και να ευνοήσει την επίδιωξη προωθημένων μαθησιακών στόχων, την εφαρμογή σημαντικών παιδαγωγικών αρχών και την υιοθέτηση καινοτόμων μεθόδων διδασκαλίας. Σημειωτέον ότι οι εμπειρίες τους για τη συμβολή του υπολογιστή στην οργάνωση των σχολικών δραστηριοτήτων αναφέρονται κυρίως στην επεξεργασία ορισμένων δεδομένων, όπως είναι οι βαθμολογίες των μαθητών, η μισθοδοσία των εκπαιδευτικών, η δημιουργία και αλλαγή ωρολογίων προγραμμάτων, το χτίσιμο πηγών πληροφόρησης με πρόσβαση και στο διαδίκτυο κ.ά.

Παρόλο βεβαίως που υπάρχουν ακόμη μερικοί σκεπτικιστές, που υιοθετούν την ιδέα της αντίστασης στην εξάπλωση της χρήσης του υπολογιστή στη ζωή μας με το σκεπτικό ότι αυτή θα οδηγούσε στη μηχανοποίηση και αποπροσωποποίηση των ανθρώπινων σχέσεων και στη χειραγώγηση των κοινωνικά αδυνάτων μέσω ενός πανίσχυρου μέσου, υπερισχύει γενικά η άποψη ότι η κοινωνία και η εκπαίδευση, ιδιαίτερα στην εποχή της παγκοσμιοποίησης της οικονομίας, δεν θα μπορούσαν να επιβιώσουν αν μείνουν πίσω και δεν ακολουθήσουν το *ρεύμα των εξελίξεων*. Η πραγματικότητα όμως δεν είναι τόσο απλή για να απαντήσει κανείς άκριτα με ένα ναι ή ένα όχι σε αυτά τα ζητήματα ή να περιορίσει το πρόβλημα ανάγοντάς το σε ζήτημα απλού εκσυγχρονισμού.

Ο εκσυγχρονισμός λοιπόν που οραματιζόμαστε στην εκπαίδευση με την αξιοποίηση των Νέων Τεχνολογιών δεν αφορά απλώς τη χρήση ενός τεχνολογικά προηγμένου εργαλείου και ενός ακόμη συμβατικού μέσου διδασκαλίας, αλλά προϋποθέτει και προσδοκά ουσιαστικότερες αλλαγές στη φύση της παραδοσιακής διδασκαλίας και στη σχολική κουλτούρα της μάθησης, οι οποίες είναι δυνατόν να επιφέρουν θετικά και

επιθυμητά από εκπαιδευτική άποψη αποτελέσματα, όσον αφορά τη γνωστική ανάπτυξη και το μοντέλο του μαθητευόμενου και αυριανού πολίτη, που επιδιώκουμε, το ψυχοκοινωνικό κλίμα της μαθησιακής διαδικασίας, το ρόλο του εκπαιδευτικού και του σχολείου, γενικότερα κ.ά..

Δημιουργούνται τώρα νέα μαθησιακά περιβάλλοντα με πρωτόγνωρα εργαλεία και διαδικασίες, που αποδεικνύονται πολύτιμοι βοηθοί για ο δάσκαλο και δημιουργούν συνθήκες που αλλάζουν το ρόλο του.

Θα πρέπει όμως να επισημανθεί ότι οι απελευθερωτικές δυνατότητες του νέου αυτού και τόσο καθολικού εργαλείου, όπως ονομάζεται ο υπολογιστής, δεν είναι αυτονόητες και δεν πραγματώνονται μηχανικά και ανώδυνα, αλλά προϋποθέτουν αλλαγές νοοτροπιών και πρακτικών στο σύνολο των φορέων που εμπλέκονται στην εκπαιδευτική διαδικασία, επομένως και αλλαγές στις συνθήκες του ευρύτερου κοινωνικοπολιτικού πλαισίου του εκπαιδευτικού σχεδιασμού. Στο πλαίσιο αυτό ο ρόλος του δασκάλου θεωρείται καταλυτικός και η ανάπτυξη της παιδαγωγικής του ικανότητας, αλλά και της προσωπικότητάς του, αναγνωρίζεται πλέον ως ένας όρος εκ των ών ουκ άνευ για την πραγμάτωση των επιθυμητών αλλαγών. Για μία ακόμη φορά έχει γίνει φανερό ότι καμία τεχνολογική πρόοδος δεν μπορεί να αντισταθμίσει τυχόν αδυναμίες στο έμφυτο δυναμικό της εκπαίδευσης όλων των βαθμίδων. Η επαναφορά μάλιστα στο προσκήνιο του γενικότερου προβληματισμού, όσον αφορά τη σχέση των εργαλείων με τον άνθρωπο, οδηγεί σε μία μεγαλύτερη έμφαση στη σημασία της ικανότητας του εργαζόμενου (στην περίπτωση μας του εκπαιδευτικού), αλλά και των ευκαιριών που του δίνονται για έλεγχο και δημιουργική χρήση του επαναστατικού αυτού εργαλείου, που λέγεται υπολογιστής.

β. Πιθανές αλλαγές στο ψυχοκοινωνικό κλίμα της σχολικής τάξης

Όμως, το πρόβλημα της διδακτικής δεν είναι μόνον ζήτημα καθαρά νοητικό, αλλά συνολικό, ψυχοκοινωνικό. Όλες οι πλευρές της προσωπικότητας των μαθητών, αλλά και των διδασκόντων συμμετέχουν και αλληλεπιδρούν στη διαδικασία της μάθησης. Οι πολλοί και διάφοροι ψυχοκοινωνικοί παράγοντες που εμπλέκονται στη διδακτική επικοινωνία και τη μαθησιακή αλληλεπίδραση και την αγωγή συνθέτουν ένα άλλο τεράστιο και πολύ σημαντικό κεφάλαιο, στο οποίο αξίζει να στραφεί η προσοχή όλων μας, αν πραγματικά μας ενδιαφέρει όχι μόνον η «στεγνή» γνώση, αλλά και η ανάπτυξη ώριμων προσωπικοτήτων και κοινωνικών ατόμων μιας υγιούς κοινωνίας. Οι παράγοντες αυτοί που ασφαλώς αγγίζουν και το ζήτημα της διδακτικής με τη βοήθεια του υπολογιστή είναι πάρα πολλοί. Περιοριζόμαστε μόνον να τονίσουμε ότι ούτως ή άλλως η βελτίωση των μεθόδων διδασκαλίας εμπεριέχει στοιχεία, που επηρεάζουν θετικά την ολόπλευρη ανάπτυξη της προσωπικότητας των μαθητών και να αναφερθούμε γενικά σε μία επιθυμητή αλλαγή του κλίματος στη σχολική τάξη, μέσα από την αλλαγή του ρόλου του διδάσκοντος και των δομών επικοινωνίας στο πλαίσιο των εποικοδομιστικών δραστηριοτήτων και της εργασίας με το υπολογιστή.

Πολλές είναι οι αναφορές στη διεθνή βιβλιογραφία στην αλλαγή του ρόλου του δασκάλου και του μαθητευόμενου, που αναμένεται να αλλάξει προς το δημοκρατικότερο. Οι δημιουργικές δραστηριότητες με τον υπολογιστή και με την ενεργό συμμετοχή όλων των μαθητών ελπίζεται ότι θα «λύσουν τα χέρια» του δασκάλου, που, αντί να εξαντλείται στη διατήρηση της ησυχίας και την αντιμετώπιση των παρενεργειών της μαθησιακής αποξένωσης, θα μπορεί να βρίσκεται στο πλευρό των μαθητών και να παίζει έναν άλλο ρόλο, πιο προωθημένο από παιδαγωγική άποψη, πρωτόγνωρο και γι'

αυτό ίσως και πιο δύσκολο. Έναν διευκολυντικό ρόλο, για τον οποίο πολλά ακούγονται, αλλά λίγα μπορούν να γίνουν μέσα από την παραδοσιακή ρουτίνα και τη μαζικότητα του σχολείου, με τη στρατιωτική διάταξη των θρανίων και τη συγκεντρωτική δομή του ελέγχου της μαθησιακής διαδικασίας.

Αντίθετα, λόγω της εργαστηριακής φύσης του μαθήματος με την υποβοήθηση του υπολογιστή, έχει φανεί ότι μέσα στη σχολική ομάδα ευνοείται το δημοκρατικό και εποικοδομητικό κλίμα και η ψυχοκοινωνική ανάπτυξη της προσωπικότητας των μαθητών, αφού αυτοί, έχοντας κερδίσει μια σχετική αυτονομία, εργάζονται ανά ομάδες, όπου η συνεργασία, η αλληλοδιδασκτική και γενικά η πλούσια κοινωνική αλληλεπίδραση μεταξύ τους νομιμοποιείται, το λάθος απενοχοποιείται και ενθαρρύνονται να γίνουν οι ίδιοι παραγωγοί αυθεντικής γνώσης αξιοποιώντας τις ποικίλες λειτουργίες του υπολογιστή και των δικτύων. Έτσι, οι δομές της επικοινωνίας στη σχολική ομάδα αποκτούν μεγαλύτερο εύρος και ποιότητα από ό,τι συμβαίνει συνήθως στην παραδοσιακή τάξη, όπου οι ενεργοί πρωταγωνιστές είναι, εκ των πραγμάτων, λίγοι. Οι σύνθετες, συνεργατικές και ερευνητικές δραστηριότητες των μαθητών ευνοούνται όσο ποτέ άλλοτε με τη βοήθεια του υπολογιστή, αφού αυτός, εκτός από λογισμικό διεπιστημονικού (interdisciplinary) χαρακτήρα, μπορεί να λειτουργήσει ως πλούσια πηγή πληροφόρησης, στην οποία μπορούν να έχουν πρόσβαση όλοι οι μαθητές ανεξαρτήτως κοινωνικής προέλευσης, καθώς και πηγή εργαλείων έρευνας και επιστημονικής επεξεργασίας των δεδομένων, όπως είναι τα πακέτα μαθηματικής, στατιστικής και λογικής επεξεργασίας του υπολογιστή.

Η τεχνολογία που είχε παραδοσιακά στη διάθεσή της η διδασκαλία στα σχολεία πριν από τη σταδιακή ανάπτυξη της ψηφιακής τεχνολογίας για εκπαιδευτικούς σκοπούς



αυτό ίσως και πιο δύσκολο. Έναν διευκολυντικό ρόλο, για τον οποίο πολλά ακούγονται, αλλά λίγα μπορούν να γίνουν μέσα από την παραδοσιακή ρουτίνα και τη μαζικότητα του σχολείου, με τη στρατιωτική διάταξη των θρανίων και τη συγκεντρωτική δομή του ελέγχου της μαθησιακής διαδικασίας.

Αντίθετα, λόγω της εργαστηριακής φύσης του μαθήματος με την υποβοήθηση του υπολογιστή, έχει φανεί ότι μέσα στη σχολική ομάδα ευνοείται το δημοκρατικό και επικοινωνιακό κλίμα και η ψυχοκοινωνική ανάπτυξη της προσωπικότητας των μαθητών, αφού αυτοί, έχοντας κερδίσει μια σχετική αυτονομία, εργάζονται ανά ομάδες, όπου η συνεργασία, η αλληλοδιδασκτική και γενικά η πλούσια κοινωνική αλληλεπίδραση μεταξύ τους νομιμοποιείται, το λάθος απενοχοποιείται και ενθαρρύνονται να γίνουν οι ίδιοι παραγωγοί αυθεντικής γνώσης αξιοποιώντας τις ποικίλες λειτουργίες του υπολογιστή και των δικτύων. Έτσι, οι δομές της επικοινωνίας στη σχολική ομάδα αποκτούν μεγαλύτερο εύρος και ποιότητα από ό,τι συμβαίνει συνήθως στην παραδοσιακή τάξη, όπου οι ενεργοί πρωταγωνιστές είναι, εκ των πραγμάτων, λίγοι. Οι σύνθετες, συνεργατικές και ερευνητικές δραστηριότητες των μαθητών ευνοούνται όσο ποτέ άλλοτε με τη βοήθεια του υπολογιστή, αφού αυτός, εκτός από λογισμικό διεπιστημονικού (interdisciplinary) χαρακτήρα, μπορεί να λειτουργήσει ως πλούσια πηγή πληροφόρησης, στην οποία μπορούν να έχουν πρόσβαση όλοι οι μαθητές ανεξαρτήτως κοινωνικής προέλευσης, καθώς και πηγή εργαλείων έρευνας και επιστημονικής επεξεργασίας των δεδομένων, όπως είναι τα πακέτα μαθηματικής, στατιστικής και λογικής επεξεργασίας του υπολογιστή.

Η τεχνολογία που είχε παραδοσιακά στη διάθεσή της η διδασκαλία στα σχολεία πριν από τη σταδιακή ανάπτυξη της ψηφιακής τεχνολογίας για εκπαιδευτικούς σκοπούς

χρησιμοποιούνταν συνήθως για την παρουσίαση πληροφοριών και αφηγημάτων, μερικές φορές εμπλουτισμένων με εικόνα, ήχο και κίνηση. Οι μαθητές αναμενόταν να δημιουργήσουν νοήματα με την έκθεσή τους στα κείμενα αυτά και τις ιστορίες. Κατά τη διαδικασία αυτή οι μαθητές παρέμεναν παθητικοί αναγνώστες και δέκτες μηνυμάτων, ενώ όλο το έργο της καθοδηγούμενης αποκωδικοποίησης συγκεντρωνόταν στην ευθύνη του δασκάλου. Αλλά και με την εμφάνιση της προγραμματισμένης διδασκαλίας κατά τα πρώτα στάδια της λειτουργίας των υπολογιστών, ο προγραμματιστής του προγράμματος συμπεριφερόταν όπως ένας παραδοσιακός δάσκαλος, ο προμηθευτής της μιας και μοναδικής εκδοχής της «πραγματικότητας», ο οποίος ως καλός εκπαιδευτής θα πρέπει να προσδιορίζει εκ των προτέρων τόσο τη συμπεριφορά των μαθητών, όσο και τα αποτελέσματα της μαθησιακής διαδικασίας. (Jonassen, D., <http://it.coe.uga.edu/itforum/paper1/html>).

Είναι προφανές ότι η κατάσταση που περιγράφηκε παραπάνω δεν αφορά μόνο τα μέσα, αλλά και τις διδακτικές αντιλήψεις που τη συνόδευαν στο πλαίσιο του παραδοσιακού σχολείου, στο οποίο βέβαια υπήρχαν συνήθως και κάποιες φωτεινές εξαιρέσεις.

Στη σύγχρονη εκπαιδευτική τεχνολογία, όταν υπάρχει αναφορά στον υπολογιστή, αυτός δεν θεωρείται απλώς ένα διδακτικό μέσο, αλλά και ένα εν δυνάμει «νοητικά ή γνωστικό εργαλείο» ή καλύτερα, μια πηγή γνωστικών εργαλείων. Στον τομέα της Διδακτικής με τη βοήθεια του υπολογιστή γνωστικά εργαλεία ονομάζουμε εκείνα που λειτουργούν με ορισμένες γνωστικές δομές, οντότητες και διαδικασίες, με τη βοήθεια των οποίων οι μαθητές μπορούν να σκεφτούν και να μάθουν από τις σκέψεις τους, διευκολύνοντας έτσι τη γνωστική τους ανάπτυξη. Τα εργαλεία αυτά έχουν γίνει πολύ

δημοφιλή από διάφορες εφαρμογές και κυρίως με τη δουλειά του Papert, συνεργάτη του Piaget, που πρότεινε την ενεργό εποικοδομιστική μάθηση με τη χρήση γραφικών μέσω της γλώσσας προγραμματισμού για παιδιά, τη Logo, η οποία παρέχει ευκαιρίες ενεργοποίησης ποικίλων δεξιοτήτων μάθησης. Σύμφωνα όμως με το Vygotsky, ένα γνωστικό εργαλείο δεν μπορεί να είναι αποτελεσματικό, παρά όταν λειτουργεί σε σχέση με το πολυδιάστατο και συνεχώς εξελισσόμενο πολιτισμικό πλαίσιο του μαθησιακού περιβάλλοντος.

#### **α. Παράδειγμα γνωστικού εργαλείου**

Ένα βασικό ερώτημα που τίθεται είναι κατά πόσον η υποστηριζόμενη από τον υπολογιστή μάθηση μπορεί πράγματι να χρησιμοποιηθεί για τη γνωστική ανάπτυξη των παιδιών.

Οι δραστηριότητες που στοχεύουν στη νοητική ανάπτυξη των παιδιών, εκτός από εκείνες που αποσκοπούν στην απόκτηση μηχανιστικής γνώσης, μπορούν να χωριστούν, μεταξύ άλλων, και σε δύο ευρύτερες κατηγορίες: α) στις δραστηριότητες που χρησιμεύουν για να εμπεδώσουν τα παιδιά ένα σύνολο θεμελιωδών γνώσεων και δεξιοτήτων και β) στις δραστηριότητες, οι οποίες απαιτούν τη *μεταφορά* και εφαρμογή αυτών των δεξιοτήτων σε διαδικασίες επίλυσης γενικότερων προβλημάτων. Ένας γενικός στόχος της δεύτερης μαθησιακής προσέγγισης είναι η βοήθεια προς τα παιδιά να μάθουν να σκέπτονται, να μάθουν πώς τα ίδια μαθαίνουν και πώς να μαθαίνουν. Η προώθηση της ικανότητάς τους αυτής μπορεί να επιτευχθεί μέσω της παροχής ευκαιριών για τη διαχείριση ορισμένων γνώσεων με συνδυαστική χρήση ποικίλων δεξιοτήτων και για απόκτηση περισσότερο μετατρέψιμης γνώσης.

Παραδείγματα τέτοιων ευκαιριών είναι και οι σύνθετες δραστηριότητες που έχουν σχέση με τη συστηματική χρήση ή κατασκευή λογισμικού βάσεων δεδομένων, των επεξεργαστών κειμένου, καθώς και με τον προγραμματισμό μέσω μιας γλώσσας του υπολογιστή, όπως η γλώσσα Logo. Ένα κοινό χαρακτηριστικό αυτών των εργαλείων είναι το πλεονέκτημα της ενασχόλησης με τα λεγόμενα ανοιχτά προβλήματα και τη διερεύνηση ιδεών, περισσότερο, παρά με την ανταπόκρισή τους σε προκαθορισμένα βήματα διδακτικής καθοδήγησης.

Αντίθετα με ό,τι συμβαίνει με τις αυτοματοποιημένες γνωστικές δραστηριότητες, εκείνες δηλαδή που απαιτούν γνώσεις ή επί μέρους δεξιότητες που αποκτήθηκαν μέσα από επανειλημμένη πρακτική εξάσκηση (όπως είναι δεξιότητες της ανάγνωσης, η γνώση της προπαίδειας κ.ά.) η διαδικασία επίλυσης προβλημάτων προϋποθέτει τη δημιουργική παρέμβαση της σκέψης μας και μια πιο περίπλοκη νοητική λειτουργία, κατά την οποία χρειάζεται να αναπτύξουμε διάφορες στρατηγικές προσέγγισης των προβλημάτων.

Όταν τα παιδιά μαθαίνουν να προγραμματίζουν με τη γλώσσα Logo, παραδείγματος χάριν, για να πραγματοποιήσουν ένα στόχο τους, όπως το να ζωγραφίσουν ένα συγκεκριμένο σχήμα, σημασία δεν έχει τόσο το ίδιο το σχέδιο, που ίσως θα το έκαναν εξίσου καλά και με ένα μολύβι, ούτε ακόμη η εκμάθηση μιας γλώσσας προγραμματισμού, αλλά οι ευκαιρίες που παρέχονται στο παιδί να εμπλακεί στην ανάπτυξη των απαιτούμενων στρατηγικών επίλυσης προβλημάτων, η επινόηση ιδεών και η δοκιμή της ορθότητάς τους, η ενασχόληση με τα λάθη διάγνωσης του προβλήματος κατά τρόπο θετικό, η ανάπτυξη της εμπιστοσύνης του παιδιού στην ίδια του την κρίση (αφού την εμπιστεύεται και ο δάσκαλος) και γενικά η ενασχόληση του μαθητευόμενου με την ίδια τη διαδικασία της μάθησης. Η

κατασκευή, παραδείγματος χάριν, με τη γλώσσα Logo ενός γεωμετρικού σχήματος είναι γνωστικό και όχι τεχνικό πρόβλημα.

Η παιδαγωγική αξία των γνωστικών αυτών εξερευνητικών δραστηριοτήτων που απαιτούν σκέψη, είναι ότι βοηθούν το παιδί να μάθει να σκέφτεται και για άλλες περιπτώσεις επίλυσης προβλημάτων, γενικότερα, είτε με την υιοθέτηση πιο ευέλικτης στάσης και σκέψης, είτε με την επίτευξη της μετάβασης σε πιο προωθημένα νοητικά στάδια (με την έννοια που έδωσε σ' αυτά ο Piaget) ή συστήματα σκέψης (σύμφωνα με τον Bruner).

Η απόκτηση αυτοματοποιημένων δεξιοτήτων ή γνώσεων από στήθους, όπως της ανάγνωσης, της αριθμησης, της πρόσθεσης, των κανόνων κτλ., είναι βεβαίως απολύτως αναγκαία και πρωταρχικής σημασίας, αφού μας παρέχει βασικά προαπαιτούμενα εργαλεία για τη λεκτική επικοινωνία και το «ξεκλείδωμα των θυρών» της παραπέρα μάθησης. Εκτός όμως από τη συμβολή τους στην ανάπτυξη αυτών των δεξιοτήτων, πολλές δραστηριότητες με τη χρήση εκπαιδευτικού λογισμικού παρέχουν στο μαθητευόμενο και στο δάσκαλο τη δυνατότητα να αφιερώσουν περισσότερη ενέργεια για την ενασχόλησή τους με προωθημένα και με περισσότερο απρόβλεπτο χαρακτήρα προβλήματα. Υπάρχει ανάγκη επίσης να ασχοληθούμε - εκτός από τη γνώση του τι - και με τη μάθηση του πώς. Προς αυτή την κατεύθυνση υποστηρίζεται ότι βοηθάει και η εκμάθηση από τα μικρά παιδιά μιας γλώσσας προγραμματισμού υπολογιστή, όπως είναι η Logo, στην οποία αναφερόμαστε παρακάτω, για να δώσουμε ένα παράδειγμα γνωστικού εργαλείου που αξιοποιεί ορισμένα θετικά στοιχεία της υποβοηθούμενης από τον υπολογιστή μάθησης.

Κατ' αυτό τον τρόπο, όπως χαρακτηριστικά παρατηρεί ο Papert (1980), θεωρεί ότι η χρήση του υπολογιστή βάζει στα χέρια του παιδιού νέα γνωστικά εργαλεία και μ' αυτό

τον τρόπο μεταβάλλει τη μάθηση από αντικείμενο πειθαρχίας και ταλαιπωρίας σε διαδικασία ανακάλυψης και ενθουσιασμού. Με την έννοια αυτή, ο υπολογιστής είναι μια μηχανή με την οποία ο χρήστης αναπτύσσει διανοητική σχέση καθώς ο τελευταίος προσπαθεί να αναζητήσει τις πληροφορίες και να οργανώσει την προσωπική του γνώση με νέους τρόπους και ενεργοποιεί τη σκέψη του όχι μόνον εμπειρικά, αλλά και αναστοχαστικά. Η αναστοχαστική σκέψη είναι περισσότερο ενεργή, συνειδητή και βουλευτική.

### **1.3 Η συμβολή των εργαλείων του υπολογιστή στην αναμόρφωση της μαθησιακής/διδασκτικής διαδικασίας**

Είδαμε πώς ο υπολογιστής μπορεί να χρησιμοποιηθεί ως γνωστικό και αναπτυξιακό εργαλείο και πώς οι παιδαγωγικές του χρήσεις συνδέονται με τις σημαντικότερες θέσεις και αρχές των βασικών θεωριών της μάθησης. Μια αναμόρφωση λοιπόν των αναλυτικών προγραμμάτων θα έπρεπε να δώσει περισσότερη σημασία, καθώς και χώρο και χρόνο σε δασκάλους και μαθητές για ενασχολήσεις όπως αυτές που περιγράφηκαν παραπάνω. Προς την κατεύθυνση αυτή συμβάλλουν οι Νέες Τεχνολογίες με το κατάλληλο λογισμικό, ιδιαίτερα το ανοιχτό λογισμικό, όπου ο χρήστης έχει τη δυνατότητα σημαντικής παρέμβασης και αλληλεπίδρασης τόσο με τη διεπιφάνεια χρήσης, όσο και με τους διδάσκοντες, τους συνομηλίκους του και άλλους έμπειρους ειδικούς. Πολύ συνοπτικά, αναφέρονται χάριν παραδείγματος οι παρακάτω κατηγορίες λογισμικού (Πηγή:Κόμης, 1996):

- 1. Τα ανοιχτά εργαλεία γενικής χρήσης (κειμενογράφοι, εκδοτικά συστήματα,**

σχεδιαστικά, επεξεργαστές εικόνας, Βάσεις Δεδομένων, λογιστικό φύλλο κ.ά).

Με τη βοήθειά τους είναι δυνατόν να αναβαθμιστεί η εκπαιδευτική διαδικασία με καινοτόμες εφαρμογές, που επιδιώκουν ποικίλους μαθησιακούς στόχους μέσα από την παιδαγωγικά εμπνευσμένη και δημιουργική χρήση αυτών των ανοιχτών εργαλείων. Με τον επεξεργαστή κειμένου, για παράδειγμα, τα παιδιά μαθαίνουν να οικοδομούν, να αναθεωρούν ως γνήσιοι συγγραφείς και να εξελίσσουν το λόγο τους, να τον γονιμοποιούν στο πλαίσιο της ομαδοσυνεργατικής αλληλεπίδρασης και να τον δημοσιοποιούν κατά τρόπο επαγγελματικό και αισθητικά επιμελημένο, αναπτύσσοντας έτσι γενικές και διαχρονικές δεξιότητες, ενώ συγχρόνως δοκιμάζουν και ανακαλύπτουν τις χρήσεις της υπολογιστικής τεχνολογίας. Ανάλογα αποτελέσματα επιτυγχάνονται και με τη χρήση ή κατασκευή βάσεων δεδομένων από τους μαθητές, αλλά και τους εκπαιδευτικούς, για μια συστηματική οργάνωση και επεξεργασία της γνώσης, καθώς και με τη χρήση του λογιστικού φύλλου, για την αναζήτηση μαθηματικών ή και ποιοτικών σχέσεων ανάμεσα σε δεδομένα και έννοιες.

## **2. Οι προσομοιώσεις, τα παιδαγωγικά παιχνίδια και τα υπερ/πολυ/μέσα**

Με τα παιδαγωγικά παιχνίδια οι μαθητές εμπλέκονται σε διάφορες γνωστικές καταστάσεις, εξερευνούν άγνωστους κόσμους, επιλύουν προβλήματα και βιώνουν δύσκολους ρόλους αξιοποιώντας διάφορες πηγές πληροφοριών, πραγματοποιούν γνωστικές περιπέτειες και «ανακαλύψεις». Ο υπολογιστής γίνεται ένας χώρος παιχνιδιού και διαδραμάτισης εναλλακτικών σεναρίων στρατηγικής και δράσης, π.χ. παιχνίδια ιστορικής περιπέτειας, επίλυσης προβλημάτων διαθεματικού τύπου για την αντιμετώπιση υποθετικών καταστάσεων και εμποδίων κ.ά. Τα στοιχεία της δράσης, της

δραματοποίησης και της συνολικής συμμετοχής όλων των τομέων της προσωπικότητας, της αυτονομίας και συγχρόνως της πειθαρχίας σε κανόνες, της δημιουργικής έκφρασης και της άτυπης μάθησης, της αυθεντικότητας των μαθησιακών καταστάσεων, της αυτονομίας και της συλλογικότητας, είναι μερικά από τα προτερήματα της χρήσης των παιδαγωγικά σχεδιασμένων λογισμικών εκπαιδευτικού παιχνιδιού.

Αλλά και με τα καλά σχεδιασμένα υπερκείμενα και τα πολυμέσα, οι μαθητές επιχειρούν μια σχετικά ελεύθερη πλοήγηση σε πολυμεσικές βάσεις δεδομένων (που συνήθως συνοδεύονται και από δραστηριότητες, όπου είναι δυνατόν να ενσωματωθεί η χρήση άλλων λογισμικών) αναζητούν και αξιολογούν πληροφορίες, συσχετίζουν έννοιες, αναπτύσσουν προσωπικές και συλλογικές εμπειρίες, δοκιμάζουν στρατηγικές συσχέτισης εννοιών, οικοδομούν προσωπικούς γνωστικούς χάρτες, διαπραγματεύονται ιδέες, αναπτύσσουν συλλογικές εργασίες. Διευκολύνεται έτσι η διδασκαλία που αποβλέπει στην πραγματιστική ενασχόληση, στην αυθεντική και παιγνιώδη εμπειρικογνωστική δραστηριότητα, στη διαισθητική, ευρετική και ομαδοσυνεργατική μάθηση, όπου αξιοποιούνται πολλά στοιχεία της άτυπης και μη αυτόνομης εκπαίδευσης. Πρόκειται για ένα ακόμη παράδειγμα ολοκληρωμένου προτύπου χρήσης των υπολογιστών με ποικιλία στόχων.

### **3. Τα ανοιχτά προγραμματιστικά περιβάλλοντα διερευνητικής μάθησης (Logo, Smalltalk, Cabri), οι προσομοιώσεις πειράματος και οι μοντελοποιήσεις φαινομένων**

Ο υπολογιστής γίνεται χώρος πειραματισμού του προσωπικού τρόπου του σκέπτεσθαι και επιστημονικής διερεύνησης με τη διαχείριση ή κατασκευή γνωστικών μοντέλων, την εμπειρική αναπαράσταση και κατανόηση διαφόρων φαινομένων, καθώς και την



πειραματική και ανακαλυπτική διαδικασία κατανόησης εννοιών και επίλυσης προβλημάτων. Ο χρήστης ενεργεί ως «εξερευνητής» γνωστικών μικρόκοσμων, προγραμματιστής και κατασκευαστής γνωστικών μοντέλων, ως πειραματιστής και «εφευρέτης». Πειραματίζεται, αναπτύσσει προσωπικές στρατηγικές, επίλυσης προβλήματος σε προγραμματιστικό ανοικτό, ευέλικτο και φιλικό περιβάλλον. Ο προγραμματισμός με συναρτησιακές γλώσσες και με χρήση διαδικασιών εκσφαλμάτωσης, εξερεύνησης γεωμετρικών και εννοιολογικών μικρόκοσμων σε πλαίσιο μικρών ομάδων παρέχει ευκαιρίες ανάπτυξης πρωτοβουλιών στη σκέψη με ανοιχτά προβλήματα, έκφρασης-χρήσης συμβόλων, λογικής ανάλυσης, αξιοποίησης παραμετρικού περιβάλλοντος για γενίκευση-αφαίρεση, καθώς και καθοδηγούμενης ανακάλυψης γνωστικών μοντέλων, διερεύνησης ιδεών, οικοδόμησης λογικο-μαθηματικών εννοιών και μεταγνωστικών διεργασιών. Προάγει τη διαλογική σχέση δασκάλου-μαθητή και την ανάπτυξη δεξιοτήτων συνεργασίας. Γενικά διευκολύνεται η παιδαγωγική του εποικοδομισμού και της κοινωνικής γνωστικής μάθησης (Grusec, 1995: 504-521).

Οι λιγότερο ή περισσότερο ρεαλιστικές προσομοιώσεις ποικίλων φαινομένων των φυσικών επιστημών και άλλων προβλημάτων διαθεματικού χαρακτήρα (όπως αυτές που διερευνώνται με τη βοήθεια των λογισμικών Stella, Modellus, Δημιουργός Μοντέλων κ.ά), τα λογισμικά κατασκευής εννοιολογικών χαρτών (όπως είναι το λογισμικό Inspiration, Mindmap, Decision Explorer, και ΑΝΑΠΑΡΑΣΤΑΣΗ), καθώς και εκείνα που κάνουν χρήση των υπηρεσιών του διαδικτύου για την υποστήριξη συνεργατικών δραστηριοτήτων μάθησης (όπως το CSILE, το CoVis κ.ά) αποδεικνύονται πολύτιμα εργαλεία έκφρασης και διαπραγμάτευσης ιδεών, διερεύνησης, πειραματισμού, παραγωγής έργου και οικοδόμησης νέων νοητικών σχημάτων (Κόμης, 2000: 297-298).

Δεδομένου δε ότι σήμερα αναγνωρίζεται όλο και περισσότερο η συνθετότητα και περιπλοκότητα των μεγάλων κοινωνικοοικονομικών προβλημάτων, η μοντελοποίηση προβλημάτων τόσο ποσοτικού, όσο και ποιοτικού χαρακτήρα και η άσκηση στη δοκιμαστική επίλυσή τους μέσω ειδικού λογισμικού (όπου συνήθως τα παιδιά υποδύονται εκείνους που παίρνουν κρίσιμες αποφάσεις), συμβάλει στη σύνδεση της θεωρίας με την πράξη, στην ανάπτυξη σημαντικών κριτικών και ψυχοκοινωνικών δεξιοτήτων (κάτι που αποδεικνύεται εξαιρετικά διευκολυντικό όχι μόνον για τους μαθητές, αλλά και για τους σύγχρονους επιστήμονες), στην ενιαιοποίηση των διαφορετικών γνωστικών αντικειμένων, στη συνεργατική και συλλογική μάθηση.

#### **4. Τεχνητή Νοημοσύνη – Νοήμονα Διδακτικά Συστήματα - Αλληλεπιδραστικά περιβάλλοντα μάθησης**

Ο υπολογιστής ως μια «έξυπνη», επικοινωνιακή και εκπαιδευτική μηχανή, γίνεται «έμπειρος συνομιλητής», συνεργάτης» και «δάσκαλος». Παρέχονται όροι για μια όσο γίνεται πιο ανοιχτή συνεργασία απόμου-μηχανής για την επίλυση ορισμένων προβλημάτων, καθώς αυτή μοντελοποιεί την αλληλεπίδραση εκπαιδευτή-εκπαιδευόμενου.

Τα εν λόγω λογισμικά, καθώς στηρίζονται σε θεωρητικά, ερευνητικά και τεχνολογικά επιτεύγματα πολλών επιστημών (με βάση τα οποία δημιουργήθηκαν προγράμματα με τεχνητή νοημοσύνη, που ως ένα βαθμό κατανοούν φυσικές γλώσσες) έχουν τη δυνατότητα να βοηθήσουν το χρήστη να επιλύει συγκεκριμένα προβλήματα, να διαχειρίζεται δεδομένα αβέβαια και μη πλήρη ή ερωτήσεις μη προκατασκευασμένες και να πραγματοποιεί συλλογισμούς, κατά τρόπο προσαρμοστικό και εξελισσόμενο, πράγμα που αλλάζει εντελώς τη σχέση ανθρώπου-μηχανής.

## 5. Δυνητική Πραγματικότητα

Ο υπολογιστής γίνεται όχημα μεταφοράς του χρήστη σε μία άλλη, εικονική «πραγματικότητα» για άμεση συμμετοχή σε ζωντανές προσομοιώσεις υπαρκτών ή φανταστικών καταστάσεων με μικρό ή μεγάλο βαθμό γνωστικής πρόκλησης. Ο χρήστης εισέρχεται σε εικονικούς - υπαρκτούς ή φανταστικούς - κόσμους με μια χωρίς προηγούμενο μορφή τεχνητής βιωματικότητας. Ο χρήστης είναι δυνατόν να συμμετέχει στους τεχνητούς αυτούς κόσμους με όλες σχεδόν τις αισθήσεις του να γίνει θεατής, άμεσος παρατηρητής, μέτοχος ή και χειριστής ρεαλιστικών ή φανταστικών καταστάσεων, εξερευνητής πολυδιάστατων νοητικών χώρων και να αλληλεπιδρά σε άμεσο χρόνο και σε απόσταση με εικονικά αντικείμενα ή άτομα

Ο νέος αυτός κυβερνοχώρος προσφέρει ευκαιρίες ενεργού πραγματιστικής εκπαίδευσης με βιωματική εμπλοκή του χρήστη, όμως τα παραδείγματα ενός παιδαγωγικά σχεδιασμένου και αλληλεπιδραστικού περιβάλλοντος δυνητικής πραγματικότητας είναι ακόμη λίγα και προϋποθέτουν - πέραν του λογισμικού - και την ύπαρξη ειδικού τεχνολογικού υλικού. Γενικά, ο τομέας αυτός είναι ακόμη υπό εξέλιξη.

## 6. Δίκτυα

Ο υπολογιστής γίνεται διαμεσολαβητής πρόσβασης-ξενάγησης σε πηγές πληροφόρησης και επικοινωνίας μεταξύ χρηστών (τράπεζες πληροφοριών, ηλεκτρονικό ταχυδρομείο, τηλεσυζητήσεις, τηλεδιασκέψεις, τηλεεκπαίδευση κ.ά) και ο χρήστης πλοηγός και αναζητητής της πληροφορίας, της γνώσης, καθώς και της επικοινωνίας και δικτύωσης μεταξύ ανθρώπων και φορέων στην τοπική και οικουμενική κοινότητα.

## ΚΕΦΑΛΑΙΟ 2

### Μικρόκοσμοι στην εκπαίδευση

#### 2.1 Οι μικρόκοσμοι στην εκπαίδευση

Η στροφή που πραγματοποιείται την τελευταία δεκαετία στην ανάπτυξη λογισμικού με την χρήση αντικειμενοστραφών γλωσσών προγραμματισμού, έχει οδηγήσει πολλά ακαδημαϊκά ιδρύματα στην υιοθέτηση της προσέγγισης 'Objects-First' στην εισαγωγική διδασκαλία του προγραμματισμού (ACM 2001). Μάλιστα σε πολλές χώρες το αντικειμενοστραφές μοντέλο προγραμματισμού έχει υιοθετηθεί και στη Δευτεροβάθμια Εκπαίδευση (π.χ. Ελλάδα) σε κάποιο βαθμό. Ωστόσο, φαίνεται ότι η υιοθέτηση αυτού του μοντέλου και η χρήση αντικειμενοστραφών γλωσσών δημιουργεί στους αρχάριους προγραμματιστές σημαντικά προβλήματα κατανόησης και μάθησης όπως αδυναμία χρήσης των εννοιών πριν την υλοποίησή τους, η ταύτιση κλάσης και αντικειμένων, πέρασμα μηνυμάτων, σύγχυση ταυτότητας/ χαρακτηριστικών κ.α. (Holland et al. 1997). Η χρήση επαγγελματικών γλωσσών προγραμματισμού (Java, Visual Basic, C++, C#), αν και έχουν το πλεονέκτημα ότι είναι αυτές που θα χρησιμοποιήσουν οι φοιτητές στην αγορά εργασίας (Allen 1996), δημιουργούν στους αρχάριους προγραμματιστές, εκτός των κλασικών προβλημάτων ως επαγγελματικά περιβάλλοντα, όπως, η εκτεταμένη γλώσσα, ο ανύπαρκτος διδακτικός προσανατολισμός καθώς και φύση των εφαρμογών που μπορεί κανείς να υλοποιήσει στα πρώτα του βήματα (Brusilovsky et al. 1997) και μια σειρά από διδακτικά εμπόδια, που έχουν να κάνουν με την αντικειμενοστρέφεια, όταν αυτή αποτελεί το μοντέλο της πρώτης

προσέγγισης στον προγραμματισμό. «Μικρο-Γλώσσες» (Brusilovsky et al. 1997) όπως Logo (Papert 1980), Karel (Pattis 1981), Marta (Calabrese 1989), υποσύνολα γλωσσών όπως η Blue J (Kölling & Rosenberg 1996) καθώς και εκπαιδευτικά περιβάλλοντα όπως το MicroWorlds Pro® έχουν προταθεί από πολλούς ερευνητές στην αντιμετώπιση αυτών των προβλημάτων.

Η ανάγκη των εκπαιδευτικών για ανάπτυξη δραστηριοτήτων προσαρμοσμένων στην εκπαιδευτική διαδικασία, οδηγεί στη χρήση περιβαλλόντων ανάπτυξης απλών ή σύνθετων μικρόκοσμων.

Η μέχρι τώρα πρακτική στην ανάπτυξη διερευνητικών μικρόκοσμων βασιζόταν στη συνεργασία εκπαιδευτικών με κάποια ομάδα προγραμματιστών με δεξιότητες στα προγραμματιστικά εργαλεία, η οποία και αναλαμβάνει να εκπονήσει την εφαρμογή σύμφωνα με τις ανάγκες και τις υποδείξεις τους. Στη συνέχεια, η εφαρμογή χρησιμοποιείται για τον σκοπό που δημιουργήθηκε.

Η ανάπτυξη μικρόκοσμων από τον εκπαιδευτικό με τέτοιου είδους περιβάλλοντα (η οποία αναφέρεται και ως «δευτερογενής ανάπτυξη», Kynigos et al, 2001), εμπεριέχει ένα σημαντικό πλεονέκτημα. Η εστίαση παραμένει στο προς επίλυση πρόβλημα ή προς το διαπραγμάτευση θέμα. Δεν χρειάζεται να μελετήσει προγραμματιστικές δομές, συμπεριφορές των λειτουργιών κάποιου προγραμματιστικού περιβάλλοντος, πρωτογενείς δομές δεδομένων, αλγορίθμους και άλλες ιδιαιτερότητες, που τις περισσότερες φορές, αν όχι όλες, τον αποπροσανατολίζουν. Αντίθετα απαιτούνται:

- Αρκετά καλή γνώση του θέματος για το οποίο αναπτύσσεται ο μικρόκοσμος.
- Λεπτομερής σχεδίαση των λειτουργιών που θα εκτελεί ο μικρόκοσμος.

- Το απαραίτητο υπάρχουν ή προς δημιουργία υλικό πολυμέσων που θα εμπλουτίσει το περιβάλλον του μικρόκοσμου.

Οι προγραμματιστικοί μικρόκοσμοι (programming microworlds) αναπτύσσονται για καθαρά εκπαιδευτικούς σκοπούς και χρησιμοποιούνται τόσο για το συναρτησιακό όσο και για τον αντικειμενοστραφή προγραμματισμό. Επίσης, βασίζονται σε μια φυσική μεταφορά (metaphor) (χελώνα, ρομπότ, καγκουρό, τρισδιάστατα αντικείμενα σε εικονικούς κόσμους) δηλ. πρωταγωνιστές που «ζουν» μέσα στο περιβάλλον του μικρόκοσμου, είναι εύχρηστοι, ενσωματώνουν δυνατότητες οπτικοποίησης και κίνησης (animation), συνήθως χρησιμοποιούν μια εκπαιδευτική γλώσσα προγραμματισμού και παρέχουν τη δυνατότητα της δυναμικής προσομοίωσης της εκτέλεσης των προγραμμάτων, δηλ. της βήμα προς βήμα εκτέλεσης του και συγχρόνως απεικόνιση του αποτελέσματος της εκτέλεσης στην κατάσταση του μικρόκοσμου.

## 2.2 Η προστιθέμενη αξία των μικρόκοσμων στη διδασκαλία του

### Αντικειμενοστραφή Προγραμματισμού

Η βασική ιδέα των μικρόκοσμων προγραμματισμού στηρίζεται στην κατασκευή μιας μικρής και απλής γλώσσας. Σε πολλές περιπτώσεις ο μικρόκοσμος έχει επικρατήσει να δηλώνει το συνδυασμό ενός πρωταγωνιστή (*actor*), ο οποίος ενεργεί στον μικρόκοσμο, και μιας γλώσσας, που συνήθως αναφέρεται σε μικρό σύνολο εντολών, οι οποίες ελέγχουν τη λειτουργία του πρωταγωνιστή. Στις πιο πολλές υπάρχουσες μικρο-γλώσσες ο πρωταγωνιστής είναι συνήθως μια χελώνα (όπως οι παραλλαγές της Logo), ένα ρομπότ (Karel, StageCreator) ή οποιαδήποτε άλλη οντότητα που μπορεί να λειτουργήσει μέσα στον μικρόκοσμο. Οι περισσότερες μικρο-γλώσσες περιλαμβάνουν όλες τις βασικές δομές μιας κλασικής γλώσσας προγραμματισμού όπως οι δομές ελέγχου (δομές διακλάδωσης, επαναληπτικές δομές κτλ) και ένα μηχανισμό που επιτρέπει την κατασκευή νέων εντολών.

Η επιλογή μιας μικρο-γλώσσας, για μια εύκολη εισαγωγή στον προγραμματισμό, παρουσιάζει πολλά πλεονεκτήματα.

- Οι μικρο-γλώσσες προγραμματισμού έχουν περιορισμένο συντακτικό (*syntax*) και απλή σημασιολογία (*semantics*). Ο χρόνος για την εκμάθηση μιας μικρο-γλώσσας είναι μικρός και έτσι ο αρχάριος μπορεί να συγκεντρώσει τις προσπάθειές του στην ανάπτυξη του αλγόριθμου για τη λύση του προβλήματος και στην κατανόηση των βασικών αρχών σχεδιασμού ενός προγράμματος.

- Οι μικρο-γλώσσες προγραμματισμού στηρίζονται σε μεταφορικές έννοιες (*metaphors*), οι οποίες δίνουν τη δυνατότητα να επινοηθεί ένα μεγάλο σύνολο

προβλημάτων που καλύπτουν τις βασικές αρχές προγραμματισμού αλλά και προκαλούν το ενδιαφέρον των μαθητών, αφού συνήθως έχουν σχέση με θέματα που είναι εξοικειωμένοι.

- Οι λειτουργίες που εκτελεί ο πρωταγωνιστής οπτικοποιούνται στο μικρόκοσμο στον οποίο μπορεί να ενεργήσει, όπως οι κινήσεις μιας χελώνας ή ενός ρομπότ, και έτσι γίνεται φανερή η σημασιολογία των κατασκευών καθώς και η έννοια της εκτέλεσης ενός προγράμματος. Συνήθως, οι διδακτικοί μικρόκοσμοι συνοδεύονται και από ένα γραφικό ενδιάμεσο που προκαλεί το ενδιαφέρον του μαθητή.

Οι πρώτες προσπάθειες για μικρο-γλώσσες ξεκίνησαν με τη χελώνα της Logo [Papert 80], αν και η Logo δεν σχεδιάστηκε για εισαγωγικά μαθήματα προγραμματισμού. Ακολούθησε το Ρομπότ Karel [Pattis 81 & 95]. Από τότε έχουν παρουσιαστεί και άλλοι μικρόκοσμοι, όπως: Josef the robot [Tomek 82, 83], Turingal [Brusolovsky 91], Karel-3D [Hvorecky 92], Karel++ [Bergin 96].

Όλες οι προσπάθειες που αναφέραμε παραπάνω επικεντρώθηκαν στην κατασκευή μικρο-γλωσσών προγραμματισμού και στην επινόηση ενός συνόλου προβλημάτων που στόχο έχουν να διδάξουν την ανάπτυξη αλγορίθμων, το σχεδιασμό προγράμματος και την αποσφαλμάτωση των προγραμμάτων, προκαλώντας το ενδιαφέρον του μαθητή. Ωστόσο, θα πρέπει να σημειωθεί ότι σε ελάχιστες μόνο περιπτώσεις έγινε κάποια προσπάθεια εκμετάλλευσης της τεχνολογίας ώστε να ενσωματωθούν στα αναπτυσσόμενα εκπαιδευτικά περιβάλλοντα λειτουργίες που θα διευκόλυναν τη διδακτική χρήση τους. Για παράδειγμα, σε λίγους σχετικά μικρόκοσμους προγραμματισμού υπάρχουν τμήματα για την αξιολόγηση των μαθητών ή την



καταγραφή των ενεργειών τους, προκειμένου ο καθηγητής να διαπιστώσει τις φάσεις από τις οποίες πέρασε κάθε μαθητής στην πορεία λύσης ενός προβλήματος.

Πολλά από τα προαναφερθέντα προβλήματα κατανόησης και μάθησης που έχουν αναφερθεί, προέρχονται από κάποια γενικότερα εμπόδια που προκύπτουν, όταν ο αντικειμενοστραφής προγραμματισμός αποτελεί το μοντέλο της πρώτης προσέγγισης στον προγραμματισμό.

Στα πρώτα μαθήματα που είναι και αυτά που εισάγουν τους σπουδαστές στις βασικές έννοιες του προγραμματισμού, χρησιμοποιούνται παραδείγματα περιορισμένης πολυπλοκότητας που δεν αναδεικνύουν τα οφέλη της αντικειμενοστραφούς σχεδίασης και ανάπτυξης προγραμμάτων όπως επίσης δεν σχετίζονται άμεσα με τα γενικότερα ενδιαφέροντα τους.

Προϋποθέτουν την εκμάθηση βασικών προγραμματιστικών δομών και μια πληθώρα χαρακτηριστικών και δυνατοτήτων της γλώσσας πριν την επιτυχημένη ανάπτυξη αντικειμένων (Cooper, Dann & Pausch 2003).

Η Οπτική αναπαράσταση και παρακολούθηση των αντικειμένων είναι μια αρκετά σύνθετη διαδικασία που απαιτεί προηγούμενη γνώση προγραμματισμού και διαχείρισης του περιβάλλοντος επαφής (Proulx, Raab & Rasala 2002).

Η προσπάθεια λοιπόν κατανόησης και σωστής έκφρασης με μια επαγγελματική γλώσσα προγραμματισμού στην περίπτωση των αρχάριων προγραμματιστών, οδηγεί στο να εστιάζουν οι φοιτητές περισσότερο στην σύνταξη της γλώσσας και πολύ λιγότερο στην σχεδίαση και την ανάπτυξη δεξιοτήτων αντικειμενοστραφούς σχεδίασης και αλγοριθμικής σκέψης.

## 2.3 Περιβάλλοντα μικρόκοσμων στη διδασκαλία του

### Αντικειμενοστραφή Προγραμματισμού

Οι περισσότερες από τις μικρο-γλώσσες που έχουν κατασκευαστεί και έχουν εφαρμοστεί μέχρι σήμερα εστιάζουν στον Διαδικαστικό ή Συναρτησιακό προγραμματισμό. Ορισμένες από αυτές στις νέες τους εκδόσεις υποστηρίζουν χαρακτηριστικά και δομές του αντικειμενοστραφούς προγραμματισμού όπως η StarLogo (Resnick 1994), ο Karel++ (Bergin et al. 1997) και το MicroWorlds Pro®, χωρίς όμως να είναι αμιγώς αντικειμενοστραφή. Μια νέα γενιά, αμιγώς αντικειμενοστραφών περιβαλλόντων, έχουν αναπτυχθεί τα τελευταία χρόνια και βρίσκονται στο στάδιο της ερευνητικής τους εφαρμογής και χρήσης ως εργαλεία ανάπτυξης της προγραμματιστικής σκέψης και της εισαγωγής στον προγραμματισμό. Χαρακτηριστικά παραδείγματα τέτοιων περιβαλλόντων είναι ο Objectkarel (Xinogalos 2003), η ALICE (Pausch et al. 1995) που χρησιμοποιώντας μια πολύ πλούσια σε δομές και εντολές γλώσσα αντικειμενοστραφούς προγραμματισμού και ένα τρισδιάστατο εικονικό κόσμο που απευθύνεται κυρίως σε αρχάριους προγραμματιστές της τριτοβάθμιας εκπαίδευσης και το AgentSheets (Repenning 1995). Η δημιουργία προσομοιώσεων στο περιβάλλον AgentSheets έχει ως κύριο στόχο την ανάπτυξη της αντικειμενοστραφούς προγραμματιστικής σκέψης και λιγότερο την εξοικείωση με τις εμπορικές γλώσσες προγραμματισμού. Αν και οι γλώσσες προγραμματισμού που χρησιμοποιούν τα περιβάλλοντα αυτά, αποτελούν συνήθως υποσύνολο μιας εμπορικής γλώσσας προγραμματισμού και δεν υποστηρίζουν πάντα το σύνολο των προγραμματιστικών δομών, εντούτοις θα πρέπει να διαθέτουν χαρακτηριστικά τέτοια που να πλησιάζουν όσο

το δυνατόν περισσότερο τα θεμελιώδη χαρακτηριστικά του προγραμματιστικού μοντέλου που στοχεύουν. Τα θεμελιώδη χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού όπως αυτά διατυπώθηκαν από τον Kay (1993) είναι:

- Τα πάντα είναι αντικείμενα.
- Κάθε είδους υπολογισμοί διεκπεραιώνονται από τα αντικείμενα τα οποία επικοινωνούν μεταξύ τους διατυπώνοντας αιτήματα μέσω μηνυμάτων.
- Κάθε αντικείμενο έχει την δική του μνήμη, η οποία αποτελείται από άλλα αντικείμενα.
- Κάθε αντικείμενο είναι ένα παράδειγμα (instance) μίας κλάσης αντικειμένων. Η κλάση αυτή προσδιορίζει τα χαρακτηριστικά ενός συνόλου ομοειδών αντικειμένων.
- Στις κλάσεις περιγράφονται οι συμπεριφορές των αντικειμένων που ανήκουν σε αυτή την κλάση. Έτσι κάθε αντικείμενο που ανήκει σε μία κλάση μπορεί να εκτελέσει αυτή τη συμπεριφορά.
- Οι κλάσεις είναι οργανωμένες σε μια δενδροειδή ιεραρχική δομή που καθορίζει και την κληρονομικότητα των χαρακτηριστικών από τις γονικές κλάσεις.

Αυτά τα χαρακτηριστικά σκιαγραφούν τα χαρακτηριστικά μιας αντικειμενοστραφούς γλώσσας προγραμματισμού, που βέβαια, επεκτείνονται και αναλύονται, σε μια πληθώρα επιπλέον χαρακτηριστικών, τα οποία επίσης διαφοροποιούνται ανάλογα, με την γλώσσα προγραμματισμού που χρησιμοποιείται.

Μια μικρο-γλώσσα θα πρέπει να ενσωματώνει και να αναδεικνύει με όσο το δυνατόν ποιο ξεκάθαρο τρόπο τα παραπάνω χαρακτηριστικά για να μπορεί να λειτουργήσει ως εργαλείο ανάπτυξης της αντικειμενοστραφούς σκέψης και να επιτρέψει

την ομαλή και χωρίς παρανοήσεις μετάβαση σε μια εμπορική γλώσσα προγραμματισμού.

Τα εργαλεία λογισμικού που έχουν σχεδιασθεί με στόχο τη διδασκαλία του ΑΠ και γενικά το υποστηρικτικό μαθησιακό υλικό γι' αυτό το σκοπό, είναι αναμφίβολα λιγότερα και λιγότερο ώριμα από τα αντίστοιχα για το διαδικασιακό παράδειγμα προγραμματισμού (procedural programming paradigm) (Kölling and Rosenberg, 2001b), το οποίο έχει διδαχθεί και ερευνηθεί η διδασκαλία και η εκμάθησή του για πολύ περισσότερο χρόνο.

Στην ενότητα αυτή θα αναφερθούν οι κύριες κατηγορίες εργαλείων λογισμικού και υποστηρικτικού υλικού που έχουν αναπτυχθεί για τη διδασκαλία του ΑΠ και εντοπίστηκαν στη βιβλιογραφία. Αυτές είναι:

- οι *προγραμματιστικοί μικρόκοσμοι* (programming microworlds),
- *εκπαιδευτικά προγραμματιστικά περιβάλλοντα* (educational programming environments) για τη διδασκαλία του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού που χρησιμοποιούν τη γλώσσα Java ή υποστηρίζουν ένα τμήμα της γλώσσας,
- εργαλεία για την αναβάθμιση των δυνατοτήτων των προγραμματιστικών περιβαλλόντων,
- εργαλεία λογισμικού για την υποστήριξη της OF προσέγγισης, περιβάλλοντα παιχνιδιών,
- εκπαιδευτικές γλώσσες προγραμματισμού για την εκμάθηση του ΑΠ

και διδακτικοί πόροι ειδικά αναπτυγμένοι για την υποστήριξη της χρήσης της γλώσσας Java σε εισαγωγικά μαθήματα.

Οι μικρόκοσμοι που έχουν αναπτυχθεί για την εισαγωγή στον ΑΠ είναι οι Karel the Robot in Java, KarelJ.Robot, JKarelRobot, Jeroo, Alice και objectKarel. Σε μεγάλο βαθμό –όπως και από τα ονόματά τους δηλώνεται– οι περισσότεροι βασίζονται στα περιβάλλοντα των μικρόκοσμων Karel the Robot (Pattis, 1981) που αναπτύχθηκε για την εισαγωγή στο διαδικασιακό προγραμματισμό με μια γλώσσα τύπου Pascal (Pascal-like) και της μεταγενέστερης έκδοσης του Karel++ που ο Bergin έγραψε με τη βοήθεια των Stehlik, Roberts και Pattis για την εισαγωγή στον ΑΠ. Η δεύτερη αυτή έκδοση, ουσιαστικά εισήγαγε το μικρόκοσμο Karel στην αντικειμενοστρεφή εποχή με μια γλώσσα τύπου C++ (Bergin et al., 1997).

Ο *Karel the Robot in Java* αποτελεί τη *μετάφραση του Karel++ σε γλώσσα Java*, για την πλήρη υποστήριξη της γλώσσας αυτής χρησιμοποιώντας έτοιμες κλάσεις (Becker, 2001). Οι σπουδαστές σε αυτό το μικρόκοσμο δουλεύουν καθαρά σε περιβάλλον ανάπτυξης Java δηλ. με κώδικα Java. Αναπτύχθηκε στο Πανεπιστήμιο του Waterloo στο Ontario του Καναδά, στο τμήμα της Επιστήμης των Υπολογιστών.

*Χρήση μιας εκπαιδευτικής γλώσσας προγραμματισμού.* Εφόσον ο διδακτικός στόχος της διδασκαλίας του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση είναι η κατανόηση των βασικών αρχών του και όχι η εκμάθηση μιας γλώσσας προγραμματισμού, πιστεύεται ότι είναι προτιμότερο να χρησιμοποιηθεί μια εκπαιδευτική γλώσσα προγραμματισμού. Ιδανικός για την επίτευξη αυτού του στόχου είναι ένας *προγραμματιστικός μικρόκοσμος* (programming microworld), ο οποίος:

- ενσωματώνει μια γλώσσα προγραμματισμού με περιορισμένο ρεπερτόριο εντολών, απλή σύνταξη και σημασιολογία, έτσι ώστε να αποφευχθεί το πρόβλημα της κατανόησης των σχετικών κανόνων της γλώσσας προγραμματισμού

- βασίζεται σε ένα υπαρκτό μοντέλο που είναι ήδη γνωστό στο μαθητή και όχι στη μηχανή του von Neumann, η οποία διέπεται από ένα σύνολο δυσνόητων (για τον αρχάριο προγραμματιστή) κανόνων. Ένας μικρόκοσμος ενσωματώνει συνήθως ένα πρωταγωνιστή, τον οποίο ο μαθητής κατευθύνει με τα προγράμματα που αναπτύσσει. Χαρακτηριστικό παράδειγμα είναι η *Logo*

- ενσωματώνει τη δυνατότητα της δυναμικής προσομοίωσης εκτέλεσης των προγραμμάτων, η οποία παρέχει τα παιδαγωγικά οφέλη που ήδη αναφέραμε
- επιτρέπει την επίλυση προβλημάτων που προκαλούν το ενδιαφέρον του σπουδαστή.

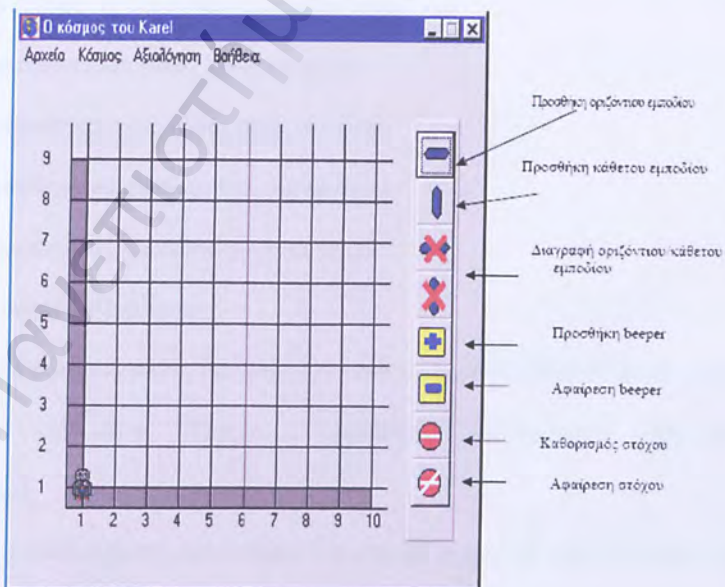
### 2.3.1 Ο μικρόκοσμος Karel

Ένας από τους πιο δημοφιλείς προγραμματιστικούς μικρόκοσμους είναι το *ρομπότ Karel* (Pattis et al., 1995) που χρησιμοποιείται για τη διδασκαλία του δομημένου προγραμματισμού. Ο Karel είναι ένα ρομπότ που ζει σε ένα απλό κόσμο στον οποίο ο χρήστης μπορεί να παρεμβαίνει με τη χρήση του ποντικιού. Οι εντολές που καθοδηγούν το ρομπότ για τις εργασίες που θα εκτελέσει συντάσσονται σε συγκεκριμένο συντάκτη.

Ο κόσμος του Karel αποτελείται από *οριζόντιους* και *κάθετους δρόμους* που απεικονίζονται στο περιβάλλον ως ένα δισδιάστατο πλέγμα οριζόντιων και κάθετων γραμμών. Ο Karel μπορεί να κινηθεί πάνω στους δρόμους οριζόντια και κάθετα και όχι διαγώνια και να σταθεί στις γωνίες και όχι πάνω στους δρόμους. Οι *εξωτερικοί τοίχοι* που περιβάλλουν το πλέγμα, περιορίζουν την κίνηση του ρομπότ μέσα σε συγκεκριμένα όρια.

Ο Karel, ο πρωταγωνιστής του μικρόκοσμου, εκτελεί διάφορες αποστολές (προγράμματα). Στον κόσμο του Karel υπάρχουν δυο είδη αντικείμενων: οι βομβητές (beepers) και τα εμπόδια. Οι βομβητές είναι μικρά αντικείμενα που τοποθετεί ο χρήστης ή και ο Karel στις διασταυρώσεις των δρόμων και τα οποία επιδιώκει να μαζέψει το ρομπότ. Τα εμπόδια μπορούν να τοποθετηθούν οριζόντια και κάθετα πάνω στους δρόμους για να εμποδίσουν την πορεία του. Ο Karel είναι ελεύθερος να μετακινηθεί οπουδήποτε πάνω στους δρόμους αρκεί να μην προσπαθήσει να διασχίσει περιοχές όπου υπάρχουν εμπόδια. Αν πέσει πάνω σε κάποιο εμπόδιο τότε η λειτουργία του τερματίζεται. Στον κόσμο του Karel πρέπει να τοποθετηθεί ένα σημείο τέλους (stop) που καθορίζει το σημείο όπου τελειώνει η διαδρομή του. Προκειμένου να διευκολυνθεί ο χρήστης κατά την τοποθέτηση εμπόδιου, βομβητή ή σημείου τέλους, ο δρομέας μετατρέπεται σε «χέρι» υποδεικνύοντας το σημείο του κόσμου που μπορεί να τοποθετηθεί

το



συγκεκριμένο αντικείμενο.

Ο Karel έχει τη δυνατότητα να κινείται κατά ένα μπλοκ (move), να στρίβει αριστερά κατά  $90^\circ$  (turnLeft), να σηκώνει beepers από τη διασταύρωση που βρίσκεται (pickBeeper) και να αφήνει beepers από την τσάντα του στη διασταύρωση (putBeeper).

Ο Karel έχει τη δυνατότητα να ελέγχει την κατάσταση του κόσμου. Η γλώσσα προγραμματισμού περιλαμβάνει τις βασικές δομές επιλογής και επανάληψης, αλλά δεν υποστηρίζει μεταβλητές. Εξέλιξή του αποτελεί ο μικρόκοσμος Karel++ (Bergin et al., 1997) για τη διδασκαλία του αντικειμενοστραφούς προγραμματισμού. Η βασική κλάση ρομπότ έχει τις δυνατότητες που έχει και το ρομπότ Karel. Αν οι δυνατότητες αυτές δεν επαρκούν για την εκτέλεση μιας αποστολής ο μαθητής δηλώνει νέες κλάσεις με αυξημένες δυνατότητες.

Ο Karel μπορεί να εκτελέσει ορισμένες *εργασίες* (tasks). Η εργασία που θα εκτελέσει το ρομπότ καθοδηγείται από ένα σύνολο εντολών. Οι *βασικές εντολές* (primitive commands) που μπορεί να εκτελέσει το ρομπότ είναι:

*go\_up*: Μετακίνηση μιας θέσης προς τα πάνω

*go\_down*: Μετακίνηση μιας θέσης προς τα κάτω

*go\_left*: Μετακίνηση μιας θέσης προς τα αριστερά

*go\_right*: Μετακίνηση μιας θέσης προς τα δεξιά

*pick\_beeper*: Μαζεύει ένα beeper

Ακόμη υποστηρίζεται η δομή ελέγχου *if/then/else*, η επαναληπτική δομή *repeat (x) times <εντολή>* και τέλος υπάρχει η δυνατότητα να οριστούν νέες εντολές (υποπρογράμματα).

Το περιβάλλον σύνταξης του προγράμματος που θα εκτελέσει στη συνέχεια ο Karel είναι ένα ξεχωριστό κομμάτι από τον κόσμο του. Περιλαμβάνει ένα συντάκτη με τη



βοήθεια του οποίου συντάσσεται το πρόγραμμα, επιλογές για την αποθήκευση ή την ανάκληση ενός ήδη υπάρχοντος προγράμματος και τέλος δυνατότητες για αντιγραφή/κόλληση/διαγραφή τμήματος του προγράμματος. Στην εικόνα 2 μπορούμε να δούμε το παράθυρο «Προγραμματίζοντας τον Karel», όπου φαίνεται το μενού μέσω του οποίου αποθηκεύουμε/ανακαλούμε προγράμματα στη γλώσσα του ρομπότ καθώς και τις δυνατότητες του συντάκτη. Στο αριστερό τμήμα του παραθύρου φαίνεται το τμήμα του συντάκτη προγράμματος. Στο δεξί τμήμα φαίνεται η εργαλειοθήκη που αναφέρεται στην εκτέλεση του προγράμματος. Με τη βοήθεια των εργαλείων αυτών επιτρέπεται η εκτέλεση του προγράμματος (run) ή η εκτέλεση του προγράμματος εντολή – εντολή (step). Τα άλλα τρία πλήκτρα διευκολύνουν στην εκτέλεση του προγράμματος και στη διαχείριση του ρομπότ. Για παράδειγμα το πλήκτρο «θησαυροφυλάκιο» εμφανίζει το σύνολο βομβητών που έχει συλλέξει το ρομπότ.

Ο μικρόκοσμος αυτός έγινε με βάση το μικρόκοσμο και τη μικρο-γλώσσα που όρισε ο Pattis (1981). Η μικρο-γλώσσα του Karel περιέχει όλες τις δομές μιας Pascal-like γλώσσας και διδάσκει τις βασικές έννοιες της σειριακής εκτέλεσης, της εκτέλεσης υπό συνθήκη, της επανάληψης και της διαδικαστικής αφαίρεσης. Η έλλειψη των μεταβλητών και των εκφράσεων έχει ως αποτέλεσμα τη μείωση της πολυπλοκότητας που έχει μια γλώσσα υψηλού επιπέδου. Οι μαθητές γράφοντας προγράμματα στο περιβάλλον του Karel εξοικειώνονται με τις βασικές έννοιες του προγραμματισμού ενώ εργάζονται σ' ένα ευχάριστο και πειστικά «μεταφορικό» περιβάλλον.

## Ενσωμάτωση ενός Συστήματος Αξιολόγησης & Καταγραφής Ενεργειών Μαθητών

Τα μέχρι σήμερα προγραμματιστικά περιβάλλοντα, είτε πρόκειται για επαγγελματικά είτε για εκπαιδευτικά, δεν ενσωματώνουν λειτουργίες οι οποίες θα διευκόλυναν τη διδακτική τους λειτουργία, όπως, για παράδειγμα, συστήματα τα οποία θα συνεισέφεραν στην αξιολόγηση των μαθητών ή ακόμη περισσότερο δυνατότητες που θα επέτρεπαν την καταγραφή των ενεργειών των μαθητών. Συνήθως ο δάσκαλος εκείνο που αξιολογεί είναι το τελικό προϊόν, δηλαδή το πρόγραμμα. Οι ενδιαμέσες προσπάθειες των μαθητών συνήθως δεν καταγράφονται από τα υπάρχοντα προγραμματιστικά περιβάλλοντα, με αποτέλεσμα να μην γνωρίζει ο δάσκαλος την πορεία που ακολουθεί ο μαθητής όταν επεξεργάζεται τη λύση ενός προβλήματος. Επιπλέον, όταν χρησιμοποιούνται επαγγελματικά προγραμματιστικά περιβάλλοντα, δεν δίνεται η δυνατότητα της βήμα προς βήμα εκτέλεσης τους προγράμματος με ταυτόχρονη οπτικοποίηση της εκτέλεσης αυτής. Η χρήση των ενσωματωμένων αποσφαλματιστών στα επαγγελματικά προγραμματιστικά περιβάλλοντα απαιτεί προφανώς ένα υψηλό επίπεδο κατανόησης και γενικότερα γνώσης προγραμματισμού, το οποίο δε διαθέτουν οι αρχάριοι μαθητές. Εξάλλου μια οπτικοποίηση της εκτέλεσης ενός προγράμματος θα έπρεπε να διαχειρίζεται με ιδιαίτερο τρόπο στοιχεία του προγράμματος τα οποία συχνά αναπαριστούνται με γραφικό τρόπο, όπως για παράδειγμα οι δείκτες, τα αλφαριθμητικά κτλ.

Σε μια προσπάθεια να υποστηριχθεί η διαδικασία της αξιολόγησης αλλά και να βοηθηθεί ο μαθητής στον έλεγχο του προγράμματος που παράγει, επεκτάθηκε ο διδακτικός μικρόκοσμος του Karel.

Το περιβάλλον του μικρόκοσμου που αναπτύχθηκε περιλαμβάνει τη δυνατότητα αποθήκευσης του κόσμου του Karel καθώς και του αντίστοιχου προγράμματος μετά από κάθε εκτέλεσή του. Σε περίπτωση που το πρόγραμμα του μαθητή περιλαμβάνει λογικά λάθη, η εκτέλεση διακόπτεται με την εμφάνιση σχετικού μηνύματος, παρέχοντας τη δυνατότητα να αποθηκευθεί ο κόσμος του Karel και το αντίστοιχο πρόγραμμα. Μέσω της επιλογής «Πορεία Karel» παρουσιάζεται με έντονη γραμμή η πορεία που ακολούθησε το ρομπότ όταν εκτέλεσε το αντίστοιχο πρόγραμμα. Ο δάσκαλος μπορεί να διατυπώσει κάποια σύντομα σχόλια σχετικά με το λανθασμένο πρόγραμμα, ώστε ο μαθητής να αντιληφθεί το λάθος που έκανε βοηθούμενος από τα 2 πλαίσια εκφοράς του προγράμματος, δηλαδή του κόσμου του Karel, όπου οπτικοποιούνται τα βήματα της εκτέλεσης και του συντάκτη του προγράμματος. Προφανώς ο δάσκαλος μπορεί να προσφέρει πολύ ουσιαστικότερη βοήθεια στους μαθητές αν τους καθοδηγήσει ώστε αυτοί μόνοι τους να βρουν τη σωστή λύση. Έτσι σχεδιάζεται η δημιουργία ενός συστήματος διαχείρισης δυσκολιών στο οποίο, για παράδειγμα, ο δάσκαλος υποδεικνύει την ορθή πορεία χρησιμοποιώντας έντονες γραμμές διαφόρων χρωμάτων και καλώντας το μαθητή να διορθώσει το πρόγραμμά του.

Θεωρείται ότι οι μικρο-γλώσσες προγραμματισμού αν και είναι ένας απλός και οπτικά διαισθητικός τρόπος για την εισαγωγή των μαθητών στον προγραμματισμό, αποτελούν ταυτόχρονα και μια ισχυρή μέθοδο διδασκαλίας του προγραμματισμού. Με τη βοήθεια των μικρο-γλωσσών προγραμματισμού επιχειρείται μια εισαγωγή στον προγραμματισμό αλλά και μια πολύ καλή εισαγωγή στην Επιστήμη των Η/Υ, αφού οι βασικές αρχές του προγραμματισμού αποτελούν τη βάση για τη θεμελίωση του λογικού και του

αφαιρετικού συλλογισμού που είναι θεμελιώδεις έννοιες στην επιστήμη της Πληροφορικής.

Πιστεύεται ότι οι διδακτικοί μικρόκοσμοι ανοίγουν μια νέα προοπτική στην εκπαιδευτική διαδικασία όχι μόνο ως μέσο διδασκαλίας εισαγωγικών εννοιών της Πληροφορικής αλλά και ως μέσο αξιολόγησης και μελέτης των αντιλήψεων των μαθητών.

Για το σκοπό αυτό αναπτύχθηκε το περιβάλλον προγραμματισμού βασισμένο στον κόσμο του Karel. Η προσπάθεια αυτή συνίσταται στην ανάπτυξη ενός ολοκληρωμένου περιβάλλοντος το οποίο εκτός από την υποστήριξη που θα παρέχει στη φάση της αξιολόγησης των εργασιών των μαθητών αποβλέπει και στην καταγραφή των αντιλήψεων των μαθητών. Μέχρι τώρα ελάχιστες εργασίες έχουν δημοσιευθεί σχετικά με τις αντιλήψεις των μαθητών όταν αυτοί αναπτύσσουν προγράμματα πολύ δε περισσότερο δεν έχουν αναπτυχθεί αυτόματα περιβάλλοντα καταγραφής των ενεργειών των μαθητών. Στόχος είναι να ενσωματωθούν λειτουργίες οι οποίες θα επιτρέπουν την καταγραφή των ενεργειών των μαθητών. Κάθε φορά που ο μαθητής θα ζητά την εκτέλεση του προγράμματος αυτόματα θα αποθηκεύεται ο κόσμος του Karel μαζί με το αντίστοιχο πρόγραμμα. Είναι σαφές ότι όταν ο μαθητής ζητά την εκτέλεση του προγράμματος αυτή η ενέργεια του υποδηλώνει, ότι πιστεύει πως έχει οδηγηθεί στη λύση του προβλήματος ή πως έχει ολοκληρώσει μια σημαντική φάση της λύσης.

Η δυνατότητα να ανακαλούμε και να βλέπουμε τις διαδοχικές φάσεις της ανάπτυξης του προγράμματος, θα βοηθήσει στην καταγραφή των λανθασμένων αντιλήψεων των μαθητών, στην καταγραφή των συστηματικών λαθών

και συνεπώς στην ανάπτυξη διδακτικών καταστάσεων που θα διευκολύνουν τη διδασκαλία του προγραμματισμού.

Το περιβάλλον αυτό μπορεί να χρησιμοποιηθεί για τη διδασκαλία του προγραμματισμού τόσο στη δευτεροβάθμια όσο και στην τριτοβάθμια εκπαίδευση. Το επίπεδο γνώσεων των χρηστών μπορεί να καθορίσει και το αντίστοιχο επίπεδο δυσκολίας των προτεινομένων προβλημάτων. Για παράδειγμα, μπορεί να ζητηθεί από τους σπουδαστές όταν είναι προχωρημένοι να προσδιορίσουν τη βέλτιστη διαδρομή του Karel, ή να επιλύσουν πιο πολύπλοκα προβλήματα. Ήδη σε πολλές χώρες έχουν από χρόνια χρησιμοποιήσει τη μικρο-γλώσσα του Karel, καθώς και άλλες μικρές-γλώσσες, για να διδάξουν προγραμματισμό στη δευτεροβάθμια και τριτοβάθμια εκπαίδευση [Brusilovsky et al. 97]. Σε πολλά πανεπιστήμια τα εισαγωγικά μαθήματα προγραμματισμού γίνονται με κάποια μικρο-γλώσσα ή με κάποια προγραμματιστικά περιβάλλοντα αναπτυγμένα ειδικά για εκπαιδευτικούς σκοπούς, όπως τα περιβάλλοντα Logo (π.χ. Πανεπιστήμιο J. Fourier, Grenoble), Pascal Genie 1.0 [Miller et. Al. 94], Thetis [Freund and Roberts 96] και άλλα.

Πρέπει να τονιστεί ότι τα μέχρι τώρα δημοσιευμένα αποτελέσματα σχετικά με την αποτελεσματικότητα των μικρο-γλωσσών προγραμματισμού δεν στηρίζονται σε μεγάλη εμπειρική αξιολόγηση [Brusilovsky et al. 97]. Θεωρείται ότι ο μικρόκοσμος αυτός θα συμβάλλει προς αυτή την κατεύθυνση με τη δυνατότητα που ενσωματώνει για καταγραφή των ενεργειών των μαθητών.

Ένας προσομοιωτής του ρομπότ Karel που διατίθεται ελεύθερα στο διαδίκτυο (<http://ocweb.otterbein.edu/csc/cs115/web/Karel>) είναι ο KarelTheRobot που παρέχει τη

δυνατότητα ανάπτυξης, αποθήκευσης, ανάκλησης και βηματικής εκτέλεσης - τόσο προς τα εμπρός όσο και προς τα πίσω - των προγραμμάτων.

Ένας προσομοιωτής που βασίζεται στον Karel++ είναι ο διδακτικός μικρόκοσμος objectKarel (Ξυνόγαλος, 2002; Xinogalos & Satratzemi 2002). Ο μικρόκοσμος αυτός ενσωματώνει μια σειρά μαθημάτων στα ελληνικά και δραστηριοτήτων για την εξοικείωση του μαθητή με τις βασικές έννοιες του αντικειμενοστραφούς προγραμματισμού και τις βασικές δομές. Για παράδειγμα, ο μαθητής πατώντας απλά κάποια κουμπιά μπορεί να παρακολουθήσει πως ανταποκρίνεται ένα ρομπότ στα τέσσερα μηνύματα της βασικής κλάσης ρομπότ, καθώς επίσης και τη σύνταξη της εντολής στη γλώσσα προγραμματισμού.

Επιπλέον, ο διδακτικός μικρόκοσμος:

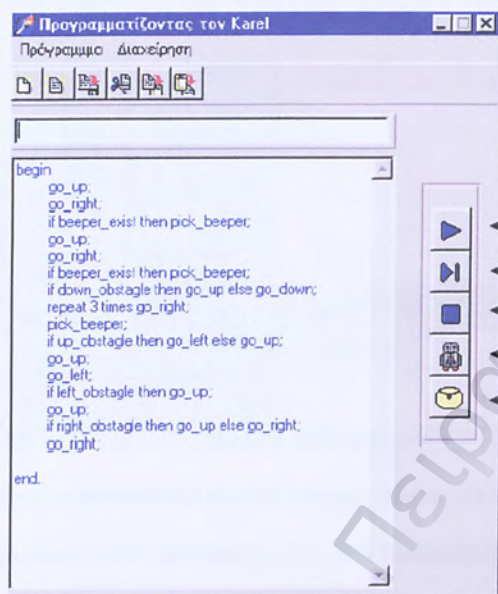
- ενσωματώνει έναν εκδότη δομής. Ο μαθητής αναπτύσσει τα προγράμματά του μέσω ενός μενού και πλαισίων διαλόγου. Έτσι, η διδασκαλία επικεντρώνεται στις έννοιες και όχι στις συντακτικές λεπτομέρειες της γλώσσας προγραμματισμού

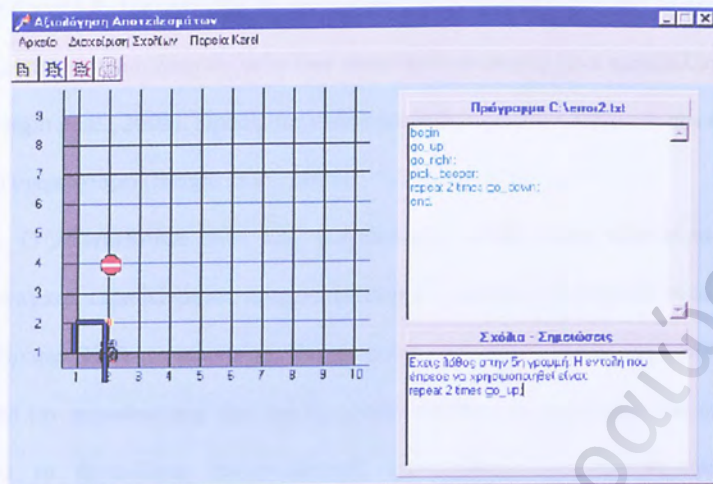
- ενσωματώνει τη δυνατότητα της βήμα προς βήμα εκτέλεσης των προγραμμάτων

- ενσωματώνει τη δυνατότητα της επεξηγηματικής οπτικοποίησης, την εμφάνιση δηλαδή επεξηγήσεων της τρέχουσας εντολής στα ελληνικά. Οι επεξηγήσεις αυτές εμφανίζονται στο κάτω μέρος του βασικού παραθύρου του προγραμματιστικού περιβάλλοντος. Η δυνατότητα αυτή βοηθάει το μαθητή στην πληρέστερη κατανόηση της σημασίας των διαφόρων δομών και της ροής εκτέλεσης ενός προγράμματος

- αναφέρει εξελληνισμένα μηνύματα λάθους που χρησιμοποιούν απλή γλώσσα και χαρακτηρίζονται από σαφήνεια και πληρότητα

- ενσωματώνει σύστημα βοήθειας, το οποίο έχει τη μορφή ενός υπερκειμένου.





- ενσωματώνει τη δυνατότητα της καταγραφής των ενεργειών των μαθητών. Η ιστορία των μεταγωγτίσεων παρουσιάζεται σε ένα διαφορετικό παράθυρο με τη μορφή ενός δένδρου δύο επιπέδων



Ο **KarelJ.Robot** αποτελεί τη *μετάφραση που ο Bergin έκανε στο μικρόκοσμο Karel++ σε αμιγώς γλώσσα Java* δηλαδή ένα αντικειμενοστρεφές Java περιβάλλον (Bergin, 2000b; Bergin et al., 2003). Πρόσφατα εκδόθηκε βιβλίο με παραδείγματα και ασκήσεις γι' αυτόν το μικρόκοσμο (Bergin et al., 2005).

Ο **JKarelRobot** είναι ένας μικρόκοσμος ανεξάρτητος πλατφόρμας (γραμμένος σε Java) και παραδείγματος προγραμματισμού, ο οποίος *υποστηρίζει τύπου Pascal, Java και Lisp περιβάλλοντα και επίσης διαγράμματα ροής (flow charts) (με αντίστροφη κατεύθυνση από την παραδοσιακή, δηλ. για τη μετατροπή έτοιμων προγραμμάτων σε διάγραμμα ροής) και τη δυνατότητα διαγραμματικής παρουσίασης του προγράμματος με τη χρήση διαγραμμάτων δομής (control structure diagrams (CSD))*. Τα τελευταία έχουν στόχο τη βελτίωση της καταληπτότητας του πηγαίου κώδικα και είναι διαγράμματα που απεικονίζουν γενικά τη συνολική δομή κάθε τμήματος του προγράμματος. Αναπτύχθηκε από τους Buck και Stucki και χρησιμοποιήθηκε στο Otterbein College (Buck and Stucki, 2001).

Ο προγραμματιστικός μικρόκοσμος **objectKarel**, αποτελεί ένα αξιόλογο εκπαιδευτικό περιβάλλον το οποίο *βασίζεται στον Karel++* (Ξυνόγαλος, 2002; Xinogalos et al., 2006, Xinogalos and Satratzemi, 2002) (<http://csis.pace.edu/~bergin/temp/findkarel.html>) και αναπτύχθηκε στο τμήμα *Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας*. Σημαντικές του πρωτοτυπίες αποτελούν, η ύπαρξη του *εκδότη δομής (structure editor)* για τη δημιουργία των προγραμμάτων, η δυνατότητα της *επεξηγηματικής οπτικοποίησης* (δηλ. της εμφάνισης επεξηγήσεων για την τρέχουσα εντολή), η *σειρά μαθημάτων (e-lessons)* που διαθέτει στα ελληνικά και στα αγγλικά και των δραστηριοτήτων για την εξοικείωση των

σπουδαστών με τις βασικές αντικειμενοστραφείς έννοιες και τις δομές ελέγχου και η δυνατότητα καταγραφής των ενεργειών των σπουδαστών καθώς δημιουργούν προγράμματα (καταγραφή των προγραμμάτων και όλων των μεταγλωττίσεων τους) για τον προσδιορισμό δυσκολιών και παρανοήσεων τους.

### 2.3.2 Ο μικρόκοσμος Jeroo

Ο προγραμματιστικός μικρόκοσμος *Jeroo* (<http://www.jeroo.org/>) (Sanders and Dorn 2003) χρησιμοποιεί σαν μεταφορέα ένα σπάνιο είδος θηλαστικού παρόμοιο με τα μικρόσωμα καγκουρό της Αυστραλίας. Από κατασκευής το περιβάλλον εστιάζει στις έννοιες «αντικείμενο» και «μέθοδος» και στις δομές ελέγχου (*control structures*). Υποστηρίζει δύο στυλ γλωσσών προγραμματισμού, ένα Java/C++/C# και ένα VB.NET στυλ. Το δεύτερο είναι ένα πραγματικό υποσύνολο της VB.NET. Εκτός από τη συνάφεια της γλώσσας του περιβάλλοντος με τη Java και την ευκολία σύνταξης της, πλεονέκτημά του αποτελεί και το ότι όλα είναι ορατά κάθε στιγμή σε ένα και μοναδικό παράθυρο. Ο κώδικας, το περιβάλλον του μεταφορέα, η περιοχή μηνυμάτων μεταγλώττισης και εκτέλεσης, η περιοχή ένδειξης της θέσης του αντικειμένου στο περιβάλλον, η περιοχή ένδειξης της κατάστασης του αντικειμένου κατά την εκτέλεση του προγράμματος. Επίσης διαθέτει οπτικοποίηση της κίνησης των αντικειμένων, οπτικοποίηση της εκτέλεσης του κώδικα (τονίζεται η γραμμή κώδικα η οποία εκτελείται με αλλαγή του χρώματος των χαρακτήρων της και του υποβάθρου της), παρέχει τη δυνατότητα εκτέλεσης βήμα προς βήμα και συνεχόμενα και επιλογής ταχύτητας εκτέλεσης.

### 2.3.3 Ο μικρόκοσμος της LOGO

Η γλώσσα **Logo** είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου (οι εντολές της συνεπώς μοιάζουν με τις λέξεις μιας φυσικής γλώσσας) και είναι απόγονος της LISP (γλώσσα Τεχνητής Νοημοσύνης (Artificial Intelligence)). Στηρίζεται στις αρχές του δομημένου προγραμματισμού (structured programming) και υπήρξε η πρώτη γλώσσα προγραμματισμού που προσέφερε στους χρήστες ένα γραφικό περιβάλλον εργασίας. Η γλώσσα Logo δημιουργήθηκε στα μέσα της δεκαετίας του 60 από τον W. Feurseg. Το όνομά της προέρχεται από την ελληνική λέξη λόγος - **συλλογισμός**, λόγος - **γλώσσα**, λόγος - **υπολογισμός**. Το 1968, η ομάδα των S. Papert (που υπήρξε συνεργάτης του Piaget) και M. Minsky στο M.I.T. χρησιμοποιεί τη Logo σαν πληροφορική βάση ώστε να αναπτύξει ένα καινοτόμο εκπαιδευτικό σχέδιο που ήθελε να απαντήσει στο ερώτημα: πώς να αλλάξουμε την Εκπαίδευση χρησιμοποιώντας την Τεχνολογία; Το γραφικό περιβάλλον επικοινωνίας (Graphics User Interface) της Logo που υλοποιείται με τη χελώνα και τα γραφικά της θεωρήθηκε ότι ανταποκρινόταν σε αυτές τις προσδοκίες.

Στο περιβάλλον αυτό διεργασίες κάθε είδους αναπαριστούνται με τα γραφικά και τα σχήματα που δημιουργεί η χελώνα με τις οδηγίες του μαθητή - προγραμματιστή. Οι οδηγίες για τις θέσεις και τον προσανατολισμό της αντιστοιχούν στο διαφορεικό τρόπο σκέψης και εισάγουν τις ιδέες προχωρημένων μαθηματικών όπως του διαφορικού λογισμού. Τα διάφορα γεωμετρικά αντικείμενα περιγράφονται με διαδικασίες χωρίς την απαίτηση για «προκαθορισμένες» εξισώσεις της αναλυτικής γεωμετρίας που δεν διδάσκεται στη γενική παιδεία. Στο πλαίσιο αυτό, ο μαθητής καθοδηγεί και προγραμματίζει τον υπολογιστή και όχι το αντίθετο.

## Πρόγραμμα και προγραμματισμός (στα πλαίσια της Logo)

- επιλογή προβλήματος (από το μαθητή),
- ανάλυση του επιλεγμένου προβλήματος (κατανόηση, κατάτμηση σε υποπροβλήματα, ορισμός σχέσεων ανάμεσα στα διάφορα μέρη κλπ),
- σχεδιασμός μιας λύσης (ανασύνθεση των μερών σε ένα ενιαίο όλο, καθορισμός της σειράς, ίσως ανάπτυξη στρατηγικής),
- μορφοποίηση της λύσης (σε μια φορμαλιστική γλώσσα),
- εκτέλεση του προγράμματος,
- ανάλυση των αποτελεσμάτων.

## Ο υλικός μικρόκοσμος της LOGO

Ο υλικός μικρόκοσμος της Logo συναπαρτίζεται από ένα σύνολο τεχνολογικών αντικειμένων. Θα σταθούμε σε αυτόν περισσότερο από τους άλλους εφόσον χρησιμοποιείται πλέον στα περισσότερα σχολεία. Στα αρχικά στάδια της Logo εκτός από τον υπολογιστή και το προγραμματιστικό περιβάλλον της γλώσσας υπήρχε και η «χελώνα» (turtle) Logo, ένα ρομπότ εδάφους, που ήταν συνδεδεμένο και μπορούσε να καθοδηγηθεί προγραμματιστικά μέσω της γλώσσας. Ομοίωμα της χελώνας εδάφους ήταν και η χελώνα οθόνης που εμφανίζεται στο γραφικό περιβάλλον εργασίας της Logo. Στη σύγχρονη εξέλιξη της Logo συναντάμε τα Logo. Lego που είναι κατασκευές τύπου Lego οι οποίες μπορούν να αποτελέσουν αντικείμενο χειρισμού μέσω του προγραμματιστικού περιβάλλοντος της Logo.

Η χελώνα. (οθόνης ή εδάφους) είναι ικανή να:

- αλλάζει θέση,
- αλλάζει διεύθυνση,
- αφήνει ή όχι ίχνη στο έδαφος (ή στην οθόνη) κατά τη μετακίνησή της,
- μεταδίδει ηχητικά ή φωτεινά σήματα,
- ελέγχει την επαφή της με ένα εμπόδιο.

### Το ψυχολογικό πλαίσιο της LOGO

Για τους δημιουργούς της Logo η χρήση της προσφέρει ένα νέο τύπο παιδαγωγικού περιβάλλοντος, μέσα στο οποίο το άτομο με την κυριαρχία του επί της μηχανής μπορεί να οδηγηθεί στην οικοδόμηση σκέψεων πάνω στις ίδιες του τις πράξεις. Το παιδί προγραμματίζει τον υπολογιστή, και προγραμματίζοντάς τον αποκτά μια αίσθηση κυριαρχίας πάνω σε ένα κομμάτι της πιο σύγχρονης και δυναμικής τεχνολογίας και διαμορφώνει μια στενότερη επαφή με μερικές από τις βαθύτερες ιδέες της επιστήμης, των μαθηματικών και της τέχνης της δημιουργίας διανοητικών μοντέλων [S. Papert, 1980]. Η παραπάνω προβληματική εντάσσεται σε ένα σύγχρονο πλαίσιο γνωστικής ψυχολογίας με έντονες αναφορές από την θεωρία των συστημάτων (βασικός ο ρόλος της ανατροφοδότησης μέσα στον ατομικό έλεγχο της δράσης) και του πιαζετικού οικοδομητισμού (η γνώση είναι μια προσωπική προοδευτική διαδικασία οικοδόμησης προερχόμενη από τη δράση).

Ο S. Papert πιστεύει ότι το παιδί διαθέτει δύο ισχυρά μέσα δράσης: τη σκέψη και το σώμα τα οποία το μικρό παιδί χρησιμοποιεί ταυτόχρονα. Η γλώσσα όμως, που είναι μετάφραση της σκέψης, του επιτρέπει να δρα για αυτό το λόγο ο Papert θέλησε να

δημιουργήσει μια παιδαγωγική γλώσσα με την οποία θα ήταν δυνατόν να κατανοήσουμε τις νοητικές διαδικασίες του παιδιού. Η γλώσσα Logo, κάτω από αυτό το πρίσμα, έχει ως στόχο να επιτρέψει στο παιδί να επιδράσει στον εξωτερικό κόσμο βασιζόμενο στα δικά του μοντέλα σκέψης. Τα υλικά τμήματα της Logo πρέπει να είναι «ουδέτερα» ώστε να μην εισαγάγουν στην παιδική σκέψη συλλογισμούς συνδεμένους με την τεχνολογία (να συνιστούν δηλαδή ένα υπάκουο υλικό «μικρόκοσμο»).

Η βασική ιδέα συνεπώς του ψυχολογικού πλαισίου μέσα στο οποίο εγγράφεται η Logo είναι ότι η γνωστική εξέλιξη δεν είναι αποτέλεσμα παθητικής μάθησης από το παιδί αλλά αντιθέτως το παιδί είναι δημιουργός, μέσω της προσωπικής του δράσης, της συγκρότησης και εξέλιξης των προσωπικών γνωστικών δομών του.

### Στόχοι της Logo

Στο ψυχολογικό επίπεδο, το περιβάλλον της Logo [C. Solomon 1986] είχε ως στόχο την ανάδειξη δεξιοτήτων όπως:

α) Εκσφαλμάτωση (debugging) του προγράμματος: κύρια δραστηριότητα μάθησης και αυτορύθμισης με βάση την παρατήρηση των διαφορών ανάμεσα στα αναμενόμενα και στα παρατηρούμενα αποτελέσματα. Η εκσφαλμάτωση είναι θεμελιώδης έννοια στα πλαίσια του προγραμματισμού από τα παιδιά. Η ανακάλυψη λαθών, και συνεπώς η επίγνωση των νοητικών τους λειτουργιών κατά τη λύση ενός προβλήματος αποτελεί μια μεταγνωστική δεξιότητα και η μεταγνώση συνιστά ένα ώριμο αναπτυξιακό επίτευγμα του παιδιού που του επιτρέπει να ελέγχει τις νοητικές του λειτουργίες.

β) Χρήση μιας «μεταγλώσσας» που αναπτύσσεται σταδιακά και αυθόρμητα στα πλαίσια της ομάδας εργασίας.

γ) Προσέγγιση των προβλημάτων και των καταστάσεων μέσω δύο τύπων λογικής ανάλυσης, την επαγωγική - φθίνουσα, δηλαδή από το ολικό στο μερικό (top - down) και την αναγωγική - αύξουσα, δηλαδή από το μερικό στο ολικό (bottom - up).

δ) Προσφυγή σε ανθρωπομορφικές αναφορές που βοηθά την αντίληψη του προγραμματισμού ως μια διαδικασία διαλόγου με τη μηχανή.

ε) Πρακτική της ονομασίας των διαδικασιών (procedures).

στ) Πρόσκτηση της ίδιας της έννοιας της διαδικασίας.

ζ) ανακάλυψη των διάφορων γνωστικών στυλ, ο τρόπος δηλαδή που αντιδρούν στα περιβαλλοντικά ερεθίσματα, ανάμεσα στα άτομα με τα αντίστοιχα πλεονεκτήματα και μειονεκτήματα.

η) Εξαγωγή βασικών αρχών και ιδεών με ισχύ τόσο στην καθημερινή ζωή όσο και στα μαθηματικά (η έννοια της διαδικασίας, της σταθεράς, της μεταβλητής, της αναδρομής, των εγκιβωτισμένων δομών, της επαλήθευσης των υποθέσεων κλπ.).

Η Logo, γίνεται στο πλαίσιο αυτό ένα εργαλείο με το οποίο μελετώνται οι τρόποι συλλογισμού και οι διαδικασίες μάθησης μέσα σε ένα πιαζετικό πλαίσιο. Αποτελεί λοιπόν προϊόν σύνθεσης των εργασιών του Piaget που ενδιαφέρεται πάνω από όλα για την οικοδόμηση των γνώσεων (constructivism) και παρεμπιπτόντως για τα χρησιμοποιούμενα εργαλεία, και των εργασιών πάνω στην τεχνητή νοημοσύνη.

Το πιο σημαντικό συνεπώς χαρακτηριστικό της Logo, το οποίο και την διαφοροποιεί από όλες τις άλλες γλώσσες προγραμματισμού, είναι ο προσανατολισμός της ως εργαλείο ανάλυσης της διαδικασίας της μάθησης. Η Logo, εκτός από μια γλώσσα προγραμματισμού, είναι ένας εκπαιδευτικός οραματισμός και ένα σύνολο από υλικά που θα αποτελέσουν εργαλεία της σκέψης με αφετηρία τις ψυχολογικές προσεγγίσεις του

Piaget και μπορεί να χρησιμοποιηθεί ως μέσο ανάλυσης της διαδικασίας της μάθησης (Thinking about Thinking).

Αν όμως ο Piaget προσπαθεί να κατανοήσει τη γνωστική εξέλιξη των παιδιών θέτοντάς τα σε κατάσταση ανακάλυψης και κάνοντάς τα να μιλάνε, ο Papert χρησιμοποιεί τον υπολογιστή ως ένα εργαλείο πολύ πιο λεπτής παρατήρησης. Έχει ανάγκη να δώσει στα παιδιά μια γλώσσα επικοινωνίας ώστε να συνομιλούν με τη μηχανή. Η γλώσσα αυτή πρέπει να είναι απλή σε πρόσβαση και ταυτόχρονα αρκετά ισχυρή ώστε να καθιστά δυνατή την πρόοδο στις έρευνες που σχετίζονται με την κατανόηση των γνωστικών διαδικασιών. Τις ιδιότητες αυτές ο Papert τις βρίσκει στη Logo, η οποία αποτελεί ένα πλούσιο προγραμματιστικό περιβάλλον μάθησης των μαθηματικών και της γεωμετρίας προσαρμοσμένο στις παιδικές γνωστικές λειτουργίες.

### **Η παιδαγωγική θεώρηση της LOGO**

Όπως τονίζει ο δημιουργός της Logo S. Papert [1980], το βασικό ερώτημα που τίθεται είναι το πώς επηρεάζουν οι υπολογιστές τον τρόπο με τον οποίο οι άνθρωποι σκέπτονται και μαθαίνουν ακόμα κι όταν δε βρίσκονται σε φυσική επαφή με τη μηχανή. Κάτω από αυτό το πρίσμα, κατευθυντήριοι άξονας της ένταξης των υπολογιστών στην εκπαίδευση γίνεται ο προβληματισμός πάνω στο πώς μπορούν να γίνουν φορείς σπερμάτων μορφωτικής αλλαγής και δυναμικών ιδεών, στο πώς μπορούν να βοηθήσουν τους μαθητές να διαμορφώσουν νέες σχέσεις με τη γνώση κόβοντας τις παραδοσιακές γραμμές που χωρίζουν τις θετικές από τις ανθρωπιστικές επιστήμες και τη γνώση του εαυτού τους.

Ο τύπος αυτός των παιδαγωγικών εφαρμογών σηματοδεύεται από δύο πρωτότυπους χαρακτήρες: α) Ο υπολογιστής βρίσκεται στη διάθεση του μαθητή για την



πραγματοποίηση σχεδίων που ο ίδιος συνέλαβε. β) Ο μαθητής οφείλει να μάθει να επεξηγεί την ιδέα του ώστε να είναι σε θέση να τη μεταφράσει στη συνέχεια σε μια γλώσσα η οποία μπορεί να αναγνωρισθεί από τη μηχανή. Η θεωρία της Logo βασίζεται σε δύο κύρια επιχειρήματα [E. De Corte, 1993]:

Πρώτον, η εμπειρία σε ένα προγραμματιστικό περιβάλλον Logo οδηγεί στην απόκτηση γενικών γνωστικών δεξιοτήτων πάνω στη λύση προβλημάτων, δεξιότητες που μπορούν να μεταφερθούν σε άλλους γνωστικούς χώρους.

Δεύτερον, το περιβάλλον συνιστά ένα ιδανικό χώρο για τη μάθηση βασικών μαθηματικών εννοιών όπως οι γωνίες, τα πολύγωνα, οι μεταβλητές, η αναδρομικότητα κλπ. Έτσι, το εκπαιδευτικό σχέδιο που αναπτύχθηκε γύρω από τη Logo, παρουσιάστηκε ως μια εναλλακτική λύση στην κλασσική Διδασκαλία με τη Βοήθεια Υπολογιστή - Δι.Β.Υ. (CAI - Computer Assisted Instruction), τα όρια της οποίας πολύ γρήγορα έγιναν εμφανή ενώ το συμπεριφοριστικό μοντέλο από το οποίο προερχόταν γνώρισε με τη σειρά του σταδιακή περιθωριοποίηση. Η Logo, εγκατέλειψε την κλασσική παραδοχή της Δι.Β.Υ., δηλαδή αυτή της υπό συνθήκη συμπεριφοράς - που άφηνε πολύ λίγα περιθώρια στο χρήστη - και εισήγαγε τα αλληλεπιδραστικά γραφικά και τη λύση προβλήματος ως πρωταρχικές της δραστηριότητες.

Στο πλαίσιο της Logo ενθαρρύνονται ο ενεργητικός πειραματισμός που οδηγεί στην ανακάλυψη και οι δραστηριότητες επίλυσης προβλημάτων (solving problems). Η οικοδόμηση όμως της λύσης ενός προβλήματος σημαίνει την ακριβή έκφραση της κατανόησής του και αυτό μπορεί να καταστεί εφικτό στο περιβάλλον του προγραμματισμού. Ο προγραμματισμός οδηγεί πάντα σε δύο δραστηριότητες: γράψιμο

μιας εντολής, ορισμός της επόμενης εντολής. Για να υλοποιηθούν συνεπώς οι αρχές της Logo απαιτείται ένα κατάλληλο περιβάλλον.

### 2.3.4 Ο μικρόκοσμος Alice

Το εργαλείο λογισμικού *Alice* (<http://www.alice.org>) είναι ένα τριδιάστατο διαδραστικό περιβάλλον απεικόνισης (*animation*) στο οποίο οπτικοποιούνται τα αντικείμενα και οι συμπεριφορές τους και το οποίο επιχειρεί να υλοποιήσει την OF προσέγγιση στη διδασκαλία εισαγωγικών μαθημάτων προγραμματισμού (Cooper et al., 2000; Cooper et al., 2003; Moskal et al., 2004). Η τριών διαστάσεων απεικόνιση «ενισχύει» την οπτικοποίηση των αντικειμένων παρέχοντας μια αίσθηση πραγματικότητας γι' αυτά και παρέχει ένα ευέλικτο και εποικοδομητικό πλαίσιο για την κατανόηση των αντικειμενοστρεφών εννοιών. Ουσιαστικά ακολουθεί την παράδοση των μικρόκοσμων Karel παρέχοντας μια επιπλέον διάσταση. Οι τριών διαστάσεων κόσμοι είναι περισσότερο ρεαλιστικοί από τους αντίστοιχους δισδιάστατους. Στο περιβάλλον αυτό οι σπουδαστές μπορούν να δημιουργούν τους δικούς του εικονικούς κόσμους. Το εργαλείο αναπτύχθηκε από το Randy Pausch και την ερευνητική ομάδα Stage 3 research στο πανεπιστήμιο *Carnegie Mellon* (*Carnegie Mellon University, CMU*).

## ΚΕΦΑΛΑΙΟ 3

### ΤΟ ΠΕΡΙΒΑΛΛΟΝ ALICE

#### 3.1 Πλεονεκτήματα του Alice

Στόχος αυτής της εργασίας είναι να παρουσιάσει την A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) και να διερευνήσει τις πιθανές εφαρμογές της στο χώρο της εκπαίδευσης.

Ενας παράγοντας που συμβάλει στο να χάσουν οι μαθητές το ενδιαφέρον τους για την πληροφορική είναι το γεγονός ότι η πρώτη τους εμπειρία με τους υπολογιστές δεν τους προσφέρει κάποιο κίνητρο.

Για να κάνουμε τη διαδικασία εκμάθησης του προγραμματισμού περισσότερο ενδιαφέρουσα για τους μαθητές είναι σημαντικό να εισάγουμε τον προγραμματισμό σαν μια διαδικασία με τελικό κίνητρο.

Το τι συνεπάγεται ένα τελικό κίνητρο εξαρτάται σε μεγάλο βαθμό από την ηλικία των μαθητών. Έχει παρατηρηθεί ότι υπάρχει μεγαλύτερη πιθανότητα να ασχοληθεί ένα παιδί με τον προγραμματισμό ή με παραπλήσια επιστήμη αν σε μικρή ηλικία έχει ασχοληθεί με χαρά με το αντικείμενο.

Το περιβάλλον ALICE δημιουργεί κινούμενες τρισδιάστατες ιστορίες. Θεωρείται καλή επιλογή για μικρούς σε ηλικία μαθητές για τους εξής λόγους:

- ✦ Μέσα σε λίγη ώρα μπορούν οι περισσότεροι μαθητές να επινοήσουν μια ιστορία την οποία θέλουν να διηγηθούν. Θα μπορούσε κανείς να πει ότι είναι και ένας τρόπος επικοινωνίας των μαθητών.
- ✦ Οι ιστορίες έχουν μια φυσική συνέχεια δίνοντας έτσι τη δυνατότητα στους χρήστες να προχωρήσουν ακολουθώντας τις οδηγίες του καθηγητή και έπειτα να προχωρήσουν και σε πιο δύσκολες δραστηριότητες εφόσον έχουν αποκτήσει εμπειρία και εξοικείωση με το περιβάλλον.
- ✦ Οι ιστορίες μπορούν να εκφράσουν προσωπικές προσδοκίες ή όνειρα του μαθητή. Του δίνουν έτσι τη δυνατότητα να πειραματίζεται σε διαφορετικούς ρόλους και να σκέφτεται πώς θα μπορούσε να αντιδράσει σε διάφορες καταστάσεις.
- ✦ Ακόμη και φίλοι τους που δεν μπορούν να διαβάσουν και να εκτιμήσουν ένα πρόγραμμα, μπορούν σίγουρα να καταλάβουν την αξία μιας κινούμενης ιστορίας.

Η ALICE είναι ένα προγραμματιστικό περιβάλλον που επιτρέπει στους χρήστες να δημιουργούν τρισδιάστατες ιστορίες, ταινίες και παιχνίδια. Η ALICE σχεδιάστηκε για να κάνει τη διαδικασία της εκμάθησης προγραμματισμού ευκολότερη για τους αρχάριους προγραμματιστές απαλορίζοντας δύο από τις μεγαλύτερες δυσκολίες που αντιμετώπιζαν: τα συντακτικά λάθη και το μη ορατό έλεγχο κατάστασης. Τα προγράμματα στην ALICE είναι κινούμενα σχέδια στα οποία ο μαθητής μπορεί να δει τα λάθη του. Στην ALICE οι μαθητές δημιουργούν προγράμματα σέρνοντας και τοποθετώντας στοιχεία κώδικα με το ποντίκι τους, κάτι που αποκλείει την πιθανότητα να κάνουν συντακτικά λάθη. Η ALICE

επιτρέπει στους μαθητές να εξοικειωθούν και να αποκτήσουν εμπειρία σε έννοιες προγραμματισμού που διδάσκονται σε μαθητές μεγάλης ηλικίας με πολύ απλό και κατανοητό τρόπο. Τους παρουσιάζει τις έννοιες αυτές μέσα από την κατασκευή των ιστοριών που επιλέγουν και έτσι παίρνουν μια πρώτη γεύση από έννοιες όπως μέθοδοι, μεταβλητές, παράμετροι και άλλες.

Επέλεξα να ασχοληθώ με το περιβάλλον ALICE γιατί πρόκειται για ένα περιβάλλον στο οποίο οι χρήστες μπορούν να δημιουργήσουν ένα πρόγραμμα σέρνοντας και τοποθετώντας με το ποντίκι τους τις κινήσεις που θέλουν να κάνουν οι πρωταγωνιστές τους.

Γιατί, όμως, να επιλέγουμε ALICE; Η ALICE προσφέρει υποστήριξη σε δύο από τα μεγαλύτερα προβλήματα που αντιμετωπίζουν οι μαθητές: 1. να πληκτρολογήσουν συντακτικά σωστά προγράμματα, και 2. να βρουν και φτιάξουν σωστά bugs (Cooper, Dann et al. 2003).

Ο αντικειμενοστραφής προγραμματισμός έγινε πολύ γνωστός στην επιστήμη των υπολογιστών. Ένα αντικειμενοστραφές πρόγραμμα θεωρείται ως μια συλλογή από αντικείμενα με συμπεριφορά, χαρακτηριστικά και ιδιότητες παρά μια λίστα εντολών. Έτσι η επαναχρησιμοποίηση ορισμένων τμημάτων κώδικα και οι φόρμες των αντικειμένων έχουν κάνει ευκολότερη την συντήρηση και την αναβάθμιση των προγραμμάτων.

Παρόλα αυτά παρουσιάστηκε ένα πρόβλημα. Οι μαθητές της μέσης εκπαίδευσης, ακόμη και οι πανεπιστημιακοί φοιτητές δεν επέλεγαν εύκολα ένα μάθημα προγραμματισμού. Οι περισσότεροι φοιτητές δεν προσελκύνονταν από ένα μάθημα προγραμματισμού και αυτοί που απλά παρακολουθούσαν, για δοκιμή, ένα τέτοιο

μάθημα δεν συγκινούνταν ώστε να το επιλέξουν. Οι κυριότεροι λόγοι ήταν απογοήτευση των μαθητών λόγω του επιπέδου δυσκολίας και η αμφιβολία για το αν θα καταφέρουν να αντεπεξέλθουν στις απαιτήσεις ενός τέτοιου μαθήματος. Το φαινόμενο αυτό παρατηρούταν περισσότερο σε φοιτήτριες παρά σε άνδρες φοιτητές. Με αφορμή το παραπάνω γεγονός έγινε μια έρευνα από το Πανεπιστήμιο Carnegie Mellon και τελικά το 1999 δημιουργήθηκε ένα εκπαιδευτικό προγραμματιστικό περιβάλλον με το όνομα "Alice". Η αρχική ιδέα του προγράμματος Alice εμφανίστηκε στις αρχές του 1991, όταν τα πρώτα συστήματα εικονικών κόσμων άρχισαν να δημιουργούνται στο Πανεπιστήμιο της Virginia. Ξεκινώντας με την επιθυμία δημιουργίας νέων τεχνικών αλληλεπίδρασης για την προσομοίωση προγραμμάτων, ανακαλύφθηκε ότι τα εργαλεία που ήταν διαθέσιμα για αυτό το σκοπό ήταν δύσκολα στη χρήση. Έτσι ξεκίνησε η ιδέα για δημιουργία νέων προγραμματιστικών εργαλείων που ήταν γρηγορότερα και ευκολότερα στη χρήση για οποιονδήποτε είχε την επιθυμία να μάθει να προγραμματίζει, ακόμη και για τους νέους μαθητές. Αυτή η προσπάθεια ξεκίνησε με πολλά διστακτικά προσχέδια με διάφορα ονόματα και τελικά κατέληξε στο σημερινό πρόγραμμα Alice.

Το Alice όπως αναφέρθηκε παραπάνω δημιουργήθηκε ουσιαστικά το 1999 από το Πανεπιστήμιο Carnegie Mellon με κύριους ερευνητές τους Wanda Dann, Stephen Cooper και Randy Pausch. Το Alice είναι ένα προγραμματιστικό περιβάλλον που χρησιμοποιείται ευρέως για εκπαιδευτικό σκοπό. Είναι ένα 3D περιβάλλον για δημιουργία εικονικών κόσμων που εμπεριέχει δυναμικές κινήσεις των χαρακτήρων και αλληλεπίδραση τους με το χρήστη. Αυτό είναι και το κύριο πλεονέκτημα του Alice. Χρησιμοποιεί 3D γραφικά ώστε να προσελκύσει τους μαθητές. Είναι πολύ πιο εύκολο και ευχάριστο για παράδειγμα να χρησιμοποιήσει κανείς μια κλάση Ανθρώπου – να της

μάθημα δεν συγκινούνταν ώστε να το επιλέξουν. Οι κυριότεροι λόγοι ήταν απογοήτευση των μαθητών λόγω του επιπέδου δυσκολίας και η αμφιβολία για το αν θα καταφέρουν να αντεπεξέλθουν στις απαιτήσεις ενός τέτοιου μαθήματος. Το φαινόμενο αυτό παρατηρούταν περισσότερο σε φοιτήτριες παρά σε άνδρες φοιτητές. Με αφορμή το παραπάνω γεγονός έγινε μια έρευνα από το Πανεπιστήμιο Carnegie Mellon και τελικά το 1999 δημιουργήθηκε ένα εκπαιδευτικό προγραμματιστικό περιβάλλον με το όνομα “Alice”. Η αρχική ιδέα του προγράμματος Alice εμφανίστηκε στις αρχές του 1991, όταν τα πρώτα συστήματα εικονικών κόσμων άρχισαν να δημιουργούνται στο Πανεπιστήμιο της Virginia. Ξεκινώντας με την επιθυμία δημιουργίας νέων τεχνικών αλληλεπίδρασης για την προσομοίωση προγραμμάτων, ανακαλύφθηκε ότι τα εργαλεία που ήταν διαθέσιμα για αυτό το σκοπό ήταν δύσκολα στη χρήση. Έτσι ξεκίνησε η ιδέα για δημιουργία νέων προγραμματιστικών εργαλείων που ήταν γρηγορότερα και ευκολότερα στη χρήση για οποιονδήποτε είχε την επιθυμία να μάθει να προγραμματίζει, ακόμη και για τους νέους μαθητές. Αυτή η προσπάθεια ξεκίνησε με πολλά διστακτικά προσχέδια με διάφορα ονόματα και τελικά κατέληξε στο σημερινό πρόγραμμα Alice.

Το Alice όπως αναφέρθηκε παραπάνω δημιουργήθηκε ουσιαστικά το 1999 από το Πανεπιστήμιο Carnegie Mellon με κύριους ερευνητές τους Wanda Dann, Stephen Cooper και Randy Pausch. Το Alice είναι ένα προγραμματιστικό περιβάλλον που χρησιμοποιείται ευρέως για εκπαιδευτικό σκοπό. Είναι ένα 3D περιβάλλον για δημιουργία εικονικών κόσμων που εμπεριέχει δυναμικές κινήσεις των χαρακτήρων και αλληλεπίδραση τους με το χρήστη. Αυτό είναι και το κύριο πλεονέκτημα του Alice. Χρησιμοποιεί 3D γραφικά ώστε να προσελκύσει τους μαθητές. Είναι πολύ πιο εύκολο και ευχάριστο για παράδειγμα να χρησιμοποιήσει κανείς μια κλάση Ανθρώπου – να της

δώσει χαρακτηριστικά όπως φύλλο, χρώμα δέρματος, ύψος, χρώμα μαλλιών και ματιών και άλλα- όταν βλέπει ένα χαρακτήρα σαν αντικείμενο στον κόσμο τους προγράμματος που δημιουργεί, παρά να γράψει αμέτρητες γραμμές κώδικα χωρίς να έχει άμεσο αποτέλεσμα στην οθόνη του υπολογιστή.

Για τον παραπάνω λόγο το πρόγραμμα Alice έγινε ιδιαίτερα γνωστό και αποδεκτό σε ευρύ προγραμματιστικό κοινό. Κυρίως όμως είχε απήχηση σε φοιτητές και παιδιά σχολείου που πλέον έβρισκαν ευχάριστο τον προγραμματισμό με αυτό το εργαλείο. Ακόμη και οι γυναίκες σπουδαστές ξεκίνησαν να επιλέγουν μαθήματα προγραμματισμού που γίνονταν με την βοήθεια του προγράμματος Alice γιατί αισθάνονται εξοικειωμένες με χαρακτήρες που υπάρχουν στην βιβλιοθήκη της Alice. Μερικοί ισχυρίζονται ότι το όνομα του προγράμματος Alice έχει δοθεί επίτηδες σε γένος θηλυκού έτσι ώστε να δημιουργηθεί ένα κλίμα «οικειότητας» και να προσεγγισθούν περισσότερες γυναίκες. Στην πραγματικότητα το σύστημα ονομάστηκε Alice προς τιμήν του Charles Lutwidge Dodson ο οποίος είχε ψευδώνυμο Lewis Carroll. Ο Lewis Carroll ήταν ο μαθηματικός που έγραψε το "Alice's Adventures in Wonderland" και το "Through the Looking Glass". Ο Carroll γνώριζε ότι το σημαντικότερο θέμα είναι να γίνουν τα πράγματα απλά και ενδιαφέροντα προς τον μαθητή.

Αναλυτικότερα στο προγραμματιστικό περιβάλλον Alice χρησιμοποιείται ένας συντάκτης μέσα στον οποίο ο χρήστης απλά επιλέγει, σύρει και αφήνει τα αντικείμενα και τις μεθόδους τους. Με αυτό τον τρόπο αποφεύγονται τα συντακτικά λάθη. Μπορεί να συμβούν λογικά λάθη αλλά εφόσον ο χρήστης μπορεί να εκτελέσει το πρόγραμμα, θα εντοπίσει οπτικά το λάθος και θα το διορθώσει εύκολα και κατόπιν δοκιμών. Έτσι προσαρμόζεται εύκολα ο κώδικας και εντοπίζονται τα αποτελέσματα των αλλαγών. Στο



Alice οι κλάσεις και τα αντικείμενα είναι κάτι που μπορεί κανείς να δει και να επιλέξει πριν τα προσθέσει στο πρόγραμμα του: Η ευκολία χρήσης του προγράμματος, κατέστησε το Alice ένα κοινώς αποδεκτό πρόγραμμα το οποίο πλέον χρησιμοποιείται σε πολλά πανεπιστήμια και σχολεία. Ενδεικτικά μερικά από αυτά είναι:

- Bucknell University
- California Lutheran University
- California State University at Humboldt
- Camden County College
- Carnegie Mellon University
- Clemson University
- Colorado School of Mines
- Community College of Philadelphia
- Cornell University
- Duke University
- Georgetown College
- Haverford College
- Ithaca College
- Manor College
- Mississippi Valley State University
- Plymouth State University
- Saint Edward's University
- Saint Joseph's University
- Saint Lawrence College

- San Diego State University
- Sierra Nevada College
- Southwestern University
- Tompkins Cortland Community College
- University of Colorado
- University of Illinois
- University of Mississippi
- Virginia Tech
- Και πολλά άλλα Γυμνάσια

Η χρήση του προγράμματος Alice δεν παρέμεινε μόνο στα σχολεία, προχώρησε και σε άλλες εφαρμογές. Ένα τέτοιο παράδειγμα είναι η συνεργασία του Carnegie Mellon με την EA (Electronic Arts), εταιρία δημιουργίας και παραγωγής videogames. Έτσι το Alice συνεργάζεται με χαρακτήρες του παιχνιδιού Sims 2 της EA. Το παιχνίδι Sims 2 είναι ένα videogame που είναι γνωστό παγκοσμίως. Στις 10 Μαρτίου, λοιπόν, του 2006 στο Pittsburg το Πανεπιστήμιο Carnegie Mellon συμφώνησε μια συνεργασία με την EA, που είχε σκοπό την αναζωογόνηση της επιστήμης των υπολογιστών στην Αμερική σε όλες τις βαθμίδες εκπαίδευσης. Η EA συμφώνησε να βοηθήσει στην ανάπτυξη του Alice 3.0 παρέχοντας χαρακτήρες του παιχνιδιού The Sims –του παιχνιδιού για ηλεκτρονικούς υπολογιστές που έχει πουλήσει περισσότερο από κάθε άλλο παιχνίδι- της εταιρίας. Το περιεχόμενο του Sims θα μετατρέψει το Alice από ένα άτεχνο και άκομφο, 3D προγραμματιστικό εργαλείο σε ένα όμορφο και φιλικό προς το

χρήστη προγραμματιστικό περιβάλλον. Η δημιουργία του Alice 3.0 ξεκίνησε κατευθείαν και θα διαρκέσει περίπου δύο χρόνια. Οι ειδικοί λένε πως όταν ολοκληρωθεί η μετατροπή, το νέο προγραμματιστικό περιβάλλον θα είναι σε θέση να γίνει το κύριο εργαλείο διδασκαλίας προγραμματισμού στην Αμερική. Με αυτόν τον τρόπο οι μαθητές που θα χρησιμοποιούν το Alice 3.0 θα δουλεύουν ουσιαστικά σε ένα περιβάλλον όμοιο με αυτό του Sims, εφόσον οι χαρακτήρες θα μοιάζουν και θα κινούνται όπως αυτοί του Sims και οι βιβλιοθήκη του Sims θα ενσωματωθεί στο πρόγραμμα. Με αυτό τον τρόπο ο προγραμματισμός θα γίνει ακόμη πιο διασκεδαστικός. Παρακάτω φαίνεται η διαφορά ανάμεσα στους χαρακτήρες των προγραμμάτων Alice v2.0 και Alice v3.0:

Με την χρήση του Alice v2.0 έχουν δημιουργηθεί κάποιες εφαρμογές και παιχνίδια αλλά με φτωχά γραφικά. Με το προγραμματιστικό εργαλείο Alice v3.0 αναμένεται να δημιουργηθούν παιχνίδια που τα γραφικά τους θα είναι άκρως ρεαλιστικά, ακόμη και παιχνίδια strategy ή shoot em up.

### **3.2 Έρευνες και δημοσιεύσεις για το Alice**

Το πρόγραμμα Alice έχει χρησιμοποιηθεί σε αρκετές έρευνες και δημοσιεύσεις που κατά κύριο λόγο είχαν ως θέμα την εκπαιδευτική χρήση του Alice και την ανταπόκριση των μαθητών σε αυτό. Βέβαια το Alice χρησιμοποιήθηκε και για έρευνες σχετικά με την νέα προσέγγιση του προγραμματισμού που γίνεται μέσω ενός 3D γραφικού περιβάλλοντος. Μερικές τέτοιες δημοσιεύσεις παρατίθενται στην συνέχεια:

1. Η πρώτη δημοσίευση που θα παρουσιαστεί είναι: Joel C. Adams, Department of Computer Science, Calvin College, Grand Rapids, MI, "Alice, Middle Schoolers and the

Στην έρευνα αυτή γίνεται γνωστό ότι πολλές μαθήτριες που επιλέγουν το μάθημα του προγραμματισμού αισθάνονται ότι έχουν μικρότερη σχέση με το αντικείμενο του μαθήματος απ’ ότι έχουν οι συμμαθητές τους του αντίθετου φύλλου. Επίσης την στιγμή που φτάνουν στο γυμνάσιο, πολλές νεαρές γυναίκες θεωρούν ότι το μάθημα του προγραμματισμού είναι για άτομα με γνωστικό υπόβαθρο και έτσι αποφεύγουν να το επιλέξουν. Αυτή η δημοσίευση παρουσιάζει ένα καλοκαιρινό πρόγραμμα με τίτλο “Imaginary World Camps”, στο οποίο χρησιμοποιείται το πρόγραμμα Alice ώστε να λυθούν αυτά τα προβλήματα πριν οι μαθητές φτάσουν στο γυμνάσιο. Τα προκαταρκτικά αποτελέσματα φάνηκαν πολύ ενθαρρυντικά. Με σχετικές έρευνες από το U.S. Bureau του Labor Statistics επιβεβαιώνεται ότι το 60% των αγοριών και 80% των κοριτσιών της Αμερικής απέφυγαν την περίοδο 2000 με 2004 την επιλογή του προγραμματισμού ως μάθημα παρακολούθησης. Αυτό μπορεί να έχει άμεσο αντίκτυπο στην οικονομία της Αμερικής καθώς οι δουλειές που σχετίζονται με υπολογιστές θα είναι από τις πλέον αναπτυσσόμενες δουλειές τις επόμενες δεκαετίες. Γι αυτό το λόγο αποφασίστηκε να παρουσιαστεί στις μαθήτριες ένα δελεαστικό και ευκολονόητο εκπαιδευτικό εργαλείο διδασκαλίας προγραμματισμού το οποίο θα έλυνε αρχικά το πρόβλημα της προσέγγισης του προγραμματισμού και μετέπειτα το πρόβλημα της “οικειότητας” των κοριτσιών σε σχέση με τα αγόρια. Το πρόγραμμα που επιλέχθηκε είναι το Alice. Έτσι το καλοκαίρι του 2003 ξεκίνησε ένα πιλοτικό πρόγραμμα για τους μαθητές του Λυκείου και το 2004, 2005 και 2006 ξεκίνησαν ολοκληρωμένα προγράμματα. Το πρώτο, πιλοτικό πρόγραμμα ξεκίνησε το καλοκαίρι του 2003 και συμμετείχαν 3 αγόρια και 1 κορίτσι. Το πρόγραμμα

διήρκεσε 1 εβδομάδα, δηλαδή 5 μέρες από τις 9.30 μέχρι τις 14.30. Η δομή του προγράμματος ήταν η εξής:

Ωρα	Δραστηριότητα
9.30	Παρουσίαση: Προγραμματισμός
10.00	Άσκηση: Προγραμματισμός
10.30	Ταινίες
11.30	Δραστηριότητα στην ύπαιθρο
11.45	Μεσημεριανό
12.20	Παρουσίαση: Δημιουργία ταινιών
12.45	Ταινίες
14.20	Σνακ
14.30	Αναχώρηση μαθητών

Το πρόγραμμα δούλεψε αρκετά καλά λόγω της διαφορετικότητας των δραστηριοτήτων της ημέρας και οι μαθητές ενθουσιάστηκαν με την δημιουργία δικών τους ταινιών. Έτσι αποφασίστηκε το 2004 να ξεκινήσουν ολοκληρωμένα προγράμματα. Τα ολοκληρωμένα προγράμματα ήταν των 8 ημερών, ξεχωριστά τμήματα για αγόρια και κορίτσια των 30 ατόμων το κάθε ένα. Μοιράστηκαν στα παιδιά εγχειρίδια του Alice αλλά και το πρόγραμμα, ώστε να μπορούν να διαβάζουν μόνα τους και να εξασκούνται. Το πρόγραμμα ονομάστηκε “The Imaginary Worlds Camps (IWC)” και τα δίδακτρα ήταν μόλις 260 δολάρια. Η δημογραφική σύσταση των τμημάτων δείχνει ότι τα κορίτσια συμμετείχαν σε αυτό το πρόγραμμα από μικρότερη ηλικία, κατά μέσο όρο το 63% των κοριτσιών ήταν 10 με 11 χρονών και το 69% των αγοριών ήταν 11 με 12.

διήρκεσε 1 εβδομάδα, δηλαδή 5 μέρες από τις 9.30 μέχρι τις 14.30. Η δομή του προγράμματος ήταν η εξής:

Ωρα	Δραστηριότητα
9.30	Παρουσίαση: Προγραμματισμός
10.00	Άσκηση: Προγραμματισμός
10.30	Ταινίες
11.30	Δραστηριότητα στην ύπαιθρο
11.45	Μεσημεριανό
12.20	Παρουσίαση: Δημιουργία ταινιών
12.45	Ταινίες
14.20	Σνακ
14.30	Αναχώρηση μαθητών

Το πρόγραμμα δούλεψε αρκετά καλά λόγω της διαφορετικότητας των δραστηριοτήτων της ημέρας και οι μαθητές ενθουσιάστηκαν με την δημιουργία δικών τους ταινιών. Έτσι αποφασίστηκε το 2004 να ξεκινήσουν ολοκληρωμένα προγράμματα. Τα ολοκληρωμένα προγράμματα ήταν των 8 ημερών, ξεχωριστά τμήματα για αγόρια και κορίτσια των 30 ατόμων το κάθε ένα. Μοιράστηκαν στα παιδιά εγχειρίδια του Alice αλλά και το πρόγραμμα, ώστε να μπορούν να διαβάζουν μόνα τους και να εξασκούνται. Το πρόγραμμα ονομάστηκε “The Imaginary Worlds Camps (IWC)” και τα δίδακτρα ήταν μόλις 260 δολάρια. Η δημογραφική σύσταση των τμημάτων δείχνει ότι τα κορίτσια συμμετείχαν σε αυτό το πρόγραμμα από μικρότερη ηλικία, κατά μέσο όρο το 63% των κοριτσιών ήταν 10 με 11 χρονών και το 69% των αγοριών ήταν 11 με 12.

Η δομή του νέου προγράμματος ήταν:

Ημέρα	Πρωινή Δραστηριότητα	Απογευματινή Δραστηριότητα
1	Αντικείμενα και μηνύματα	Δομές σεναρίου
2	Μέθοδοι	Κίνηση κάμερας, σκηνικό
3	Μεταβλητές και συναρτήσεις	Παράμετροι
4	Λίστες	Εφέ σκηνών
5	Επαναληπτικές δομές	Αποτίμηση, αξιολόγηση
6	Δομές ελέγχου	Ήχος
7	Συμβάντα	Τελικός έλεγχος
8	Έκθεση δημιουργημάτων για τους μαθητές και τις οικογένειες τους	

Πριν την παρακολούθηση του προγράμματος “The Imaginary Worlds Camps” και μετά την ολοκλήρωση της απαντήθηκαν ερωτηματολόγια από τους μαθητές. Αυτά είναι ενδεικτικά για την ανασκόπηση των αποτελεσμάτων του προγράμματος. Στην ερώτηση ποια είναι η άποψη σας για τους υπολογιστές το 56% των κοριτσιών πριν το πρόγραμμα και το 72% μετά απάντησαν ότι τους αρέσουν πολύ. Επίσης σχετικά με τις προγραμματιστικές τους ικανότητες το 36% πριν και το 52% μετά τα μαθήματα απάντησαν πολύ καλές, ενώ αυτός ο αριθμός έφτασε το 56% το 2006. Για την εμπειρία που έζησαν στο “The Imaginary Worlds Camps” το 2006 το 81% των κοριτσιών και το 62% των αγοριών απάντησαν πολύ καλή.

Τα αποτελέσματα της έρευνας ήταν:

- Για τους μαθητές του “The Imaginary Worlds Camps ” η δημιουργία των προσωπικών τους ταινιών ήταν ένα κίνητρο και μια άκρως ευχάριστη ενασχόληση
- Το πρόγραμμα Alice απαγορεύει την ύπαρξη συντακτικών λαθών, τα οποία είναι και ο κύριος παράγοντας για τον οποίο αποφεύγουν οι αρχάριοι τον προγραμματισμό.
- Επειδή τα λάθη δεν είναι συντακτικά, αλλά λογικά μπορούν εύκολα να διορθωθούν και να προκαλέσουν γέλιο αντί για απογοήτευση.

Το γενικό συμπέρασμα ήταν ότι τόσο τα αγόρια, όσο και τα κορίτσια απόλαυσαν την χρήση του Alice και παρακινούσαν το ένα το άλλο. Γι αυτό το λόγο γίνονται προσπάθειες να δημιουργηθούν σε διάφορα σχολεία καλοκαιρινά προγράμματα με βάση το πρόγραμμα Alice. Για την διευκόλυνση αυτών των προσπαθειών τα βιβλία του προγράμματος και οι ταινίες των μαθητών είναι διαθέσιμες στην ιστοσελίδα του “Imaginary Worlds Camps”.

2. Μια άλλη έρευνα σχετικά με το Alice είναι: Caitlin Kelleher, Randy Pausch, and Sara Kiesler, Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, “Storytelling Alice motivates middle school girls to learn computer programming”, SIGCHI conference on Human factors in computing systems, 1455-1464, 2007.

Στην έρευνα αυτή παρουσιάζεται το Storytelling Alice, ένα πρόγραμμα για την συγγραφή ιστοριών το οποίο χρησιμοποιείται από τους μαθητές του λυκείου και τους βοηθάει να δημιουργήσουν τα δικά τους 3D προγράμματα. Η έρευνα εστιάζει σε



μαθήτριες διότι αυτές συμμετέχουν σε μικρότερα ποσοστά σε μαθήματα προγραμματισμού. Το Alice υποστηρίζει την δημιουργία ιστοριών, σκηνών και παιχνιδιών παρέχοντας: 1) ένα σύνολο υψηλού επιπέδου κινήσεων, που υποστηρίζουν οι χαρακτήρες και με αυτό τον τρόπο μπορούν να αλληλεπιδράσουν μεταξύ τους, 2) μια συλλογή 3D χαρακτήρων και σκηνικών που ανταποκρίνονται σε κάθε είδους ιδέα του προγραμματιστή, 3) ένα εγχειρίδιο που εισάγει τους χρήστες στις έννοιες του προγραμματισμού και τους βοηθάει να δημιουργούν δικά τους παραδείγματα. Οι χρήστες του Alice και κυρίως τα κορίτσια, αποδείχτηκε ότι έμαθαν πολύ εύκολα και διασκεδαστικά τις βασικές δομές του προγραμματισμού με την χρήση του συγκεκριμένου προγράμματος. Παρακινούνταν περισσότερο από χρήστες που δούλευαν με άλλα προγράμματα η με απλές εκδόσεις της Alice και όχι της έκδοσης Storytelling, αφιέρωναν 42% περισσότερο χρόνο από κάθε άλλο χρήστη στις εφαρμογές τους, διαβεβαίωναν ότι θα χρησιμοποιούσαν και στο μέλλον νέες εκδόσεις του Alice και γενικότερα απέκτησαν ενδιαφέρον για τον προγραμματισμό.

Γενικότερα η αρχική έρευνα απέδειξε ότι τα κορίτσια δεν ασχολούνται με τον προγραμματισμό λόγω έλλειψης παρότρυνσης από τους γονείς, από τα άλλα κορίτσια και τους δασκάλους, έλλειψης ενδιαφέροντος και λιγότερες ευκαιρίες ενασχόλησης με τους υπολογιστές. Είναι δύσκολο να διορθωθούν οι πολιτιστικοί παράγοντες που επηρεάζουν τα κορίτσια, αλλά είναι πιο εύκολο να γίνει δελεαστικότερη η διαδικασία εκμάθησης προγραμματισμού, το οποίο αποτελεί βασικό κίνητρο. Γι αυτό το λόγο έπρεπε να δημιουργηθεί ένα περιβάλλον εκμάθησης προγραμματισμού το οποίο να είναι δελεαστικό ακόμη και για τις μαθήτριες του λυκείου. Έτσι επιλέχτηκε το πρόγραμμα Storytelling Alice για τους παρακάτω λόγους:

- Τα κορίτσια μπορούν να σκεφτούν σε λίγο χρόνο μια ιστορία που θα θέλουν να δημιουργήσουν.
- Οι ιστορίες έχουν φυσική ροή και είναι απίθανο να απαιτήσουν εξεζητημένες προγραμματιστικές τεχνικές, έτσι είναι κατάλληλο για τους αρχάριους μαθητές.
- Οι ιστορίες είναι ένα είδος προσωπικής έκφρασης και παρέχουν στα κορίτσια μια ευκαιρία να πειραματιστούν σε διαφορετικούς ρόλους, μια σημαντική δραστηριότητα για την εφηβεία.
- Οι φίλοι που δεν ασχολούνται με τον προγραμματισμό μπορούν άμεσα να καταλάβουν και να εκτιμήσουν ένα σχέδιο με κίνηση, το οποίο μπορεί να αποτελέσει θετική ανάδραση.

Τα τρία κίνητρα που παρέχει το Alice είναι:

- Αποτελεί εργαλείο που επιτρέπει στα παιδιά την διερεύνηση ιδεών
- Αποτελεί μέσο προσωπικής έκφρασης
- Αποτελεί κύριο λίθο για την μετέπειτα καριέρα ενός προγραμματιστή

Η ανάπτυξη του Storytelling Alice διήρκεσε πάνω από 2 χρόνια και έγινε με την βοήθεια 200 κοριτσιών. Ουσιαστικά έγινε ένα τεστ παραγωγικότητας που περιελάμβανε τετράωρη απογευματινή εργασία και εβδομαδιαία προγράμματα με ομάδες των 3 έως 20 κοριτσιών ηλικίας από 10 μέχρι 17. Κατά την διάρκεια του τεστ, τα κορίτσια δημιουργούσαν διατάξεις σκηνών ταινιών και μετέπειτα προσπαθούσαν να τις υλοποιήσουν με μια έκδοση του Storytelling Alice. Παράλληλα με το τεστ διεξάγονταν και τετράωρα μαθήματα στα οποία γινόταν μια σύγκριση συμπεριφοράς και επιδόσεων μεταξύ των κοριτσιών που χρησιμοποιούσαν το Storytelling Alice και κάποια άλλη

έκδοση του Alice. Σε αυτή την αξιολόγηση συμμετείχε ένα σύνολο από 88 κορίτσια, τα 43 έκαναν χρήση του Storytelling Alice και τα 45 του Alice. Ο μέσος όρος ηλικίας ήταν 12,6 χρονών. Οι 76 συμμετέχουσες δήλωσαν ότι θα παρακολουθήσουν δημόσιο σχολείο και οι 12 ιδιωτικό. Για παρότρυνση δόθηκε το χρηματικό ποσό των 10 δολαρίων σε κάθε κοπέλα. Οι συμμετέχουσες είχαν 2 ώρες και ένα τέταρτο να απαντήσουν κάποιες αρχικές ερωτήσεις και να δημιουργήσουν ένα πρόγραμμα με το αντίστοιχο πρόγραμμα Alice. Μετά είχαν 30 λεπτά να δοκιμάσουν την έκδοση της Alice που δεν είχαν επιλέξει. Στο τέλος οι συμμετέχουσες ζητήθηκαν να επιλέξουν ένα από τα δυο προγράμματα για να πάρουν σπίτι και μετά να επιλέξουν ένα από τα προγράμματα που είχε δημιουργηθεί ως το καλύτερο.

Τα αποτελέσματα ήταν πολύ θετικά αφού έδειξαν ότι αυξήθηκε το ενδιαφέρον των μαθητών χωρίς όμως να μειωθεί η εκπαιδευτική αξία του προγράμματος. Έτσι το Storytelling Alice δεν εμπόδισε τα κορίτσια να μάθουν βασικές αρχές και τεχνικές προγραμματισμού. Το ίδιο ενθαρρυντικές ήταν και οι απαντήσεις που έδωσαν τα διαγωνιζόμενα κορίτσια σε ερωτήσεις σχετικά με την ευκολία χρήσης του Alice, με το κατά πόσο είναι ψυχαγωγικό το πρόγραμμα, αν θα ενδιαφέρονταν να ξαναχρησιμοποιήσουν το Alice στο μέλλον και κατά πόσο ενδιαφέρονται πλέον για μαθήματα προγραμματισμού.

Συνοψίζοντας η έρευνα κατέληξε στο ότι τόσο οι συμμετέχουσες που χρησιμοποίησαν το Storytelling Alice όσο και το Alice έμαθαν τεχνικές προγραμματισμού. Παρόλα αυτά αυτές που χρησιμοποίησαν το Storytelling Alice ασχολήθηκαν και είχαν σκοπό να ασχοληθούν και μελλοντικά περαιτέρω με τον

προγραμματισμό. Βέβαια όλες οι συμμετέχουσες βρήκαν και τα δυο προγράμματα εξίσου ψυχαγωγικά και επιμορφωτικά.

3. Προχωρώντας θα εξεταστεί μια άλλη έρευνα: Barbara Moskal, Mathematics Department, Colorado School of Mines, Golden, Deborah Lurie, Mathematics Department, Saint Joseph's University, Philadelphia and Stephen Cooper, Computer Science Department, Saint Joseph's University, Philadelphia, "Evaluating the Effectiveness of a New Instructional Approach", 35th SIGCSE technical symposium on Computer science education, 75-79, 2004.

Η δημοσίευση κάνει μια αποτίμηση ενός εκπαιδευτικού ερευνητικού έργου που είχε ως χορηγό το ίδρυμα εθνικών επιστημών (NSF). Ο πρωταρχικός στόχος αυτού του ερευνητικού σχεδίου ήταν να αναπτυχθεί και να αξιολογηθεί μια σειρά μαθημάτων πανεπιστημίου, τα οποία θα σχεδιάζονταν έτσι ώστε να βελτιώσουν την απόδοση των φοιτητών σε εισαγωγικά θέματα προγραμματισμού. Τα αποτελέσματα αυτής της έρευνας αποδεικνύουν ότι η σειρά των μαθημάτων που αναπτύχθηκε, βελτίωσε την απόδοση των μαθητών και καλλιέργησε το ενδιαφέρον τους απέναντι στην επιστήμη των υπολογιστών.

Αυτό το άρθρο αναφέρει τα αποτελέσματα μιας μελέτης που αποτέλεσε μέρος μιας αρχικής σκέψης του ιδρύματος εθνικών επιστημών. Ο αρχικός σκοπός αυτής της έρευνας ήταν να καθοριστεί εάν οι θεμελιώδεις αρχές του αντικειμενοστραφή προγραμματισμού θα μπορούσαν να διδαχθούν σε μαθητές οι οποίοι θα είχαν ελάχιστο ή ανύπαρκτο προγραμματιστικό υπόβαθρο εισάγοντας μια καινοτομική προσέγγιση που αναπτύχθηκε από τους Wanda Dann και Stephen Cooper. Στα μαθήματά τους οι Wanda Dann και Stephen Cooper παρατήρησαν ότι πολλοί μαθητές που δεν είχαν καθόλου ή

είχαν μικρή επαφή με τον προγραμματισμό διέτρεχαν κίνδυνο αποτυχίας στο πρώτο μάθημα προγραμματισμού που μπορεί να ήταν δύσκολο και αυστηρό. Έτσι οι παραπάνω αποφάσισαν να αντιμετωπίσουν αυτό το πρόβλημα δημιουργώντας ένα νέο μάθημα στο οποίο θα χρησιμοποιούνταν ένα πρόγραμμα 3D, το Alice που αναπτύχθηκε στο Carnegie Mellon University. Οι Dann και Cooper επέλεξαν αυτό το εργαλείο γιατί θεώρησαν ότι το γραφικό περιβάλλον προγραμματισμού θα υποκινούσε τους φοιτητές. Μοιράστηκαν εγχειρίδια χρήσης και νέα βιβλία. Αντικείμενο του μαθήματος Alice και της έρευνας αυτής ήταν να εξεταστεί η αποτελεσματικότητα του Alice στην εκμάθηση βασικών αρχών προγραμματισμού στους μαθητές και κατά πόσο θα τους κινούσε το ενδιαφέρον. Η έρευνα διήρκεσε 2 χρόνια και στο Saint Joseph's University (SJU) και στο Ithaca College (IC). Το πρόγραμμα Alice επιλέχθηκε για τους εξής λόγους:

- Η εργασία με την βοήθεια ενός 3D περιβάλλοντος είναι ελκυστική και υποκινεί την σημερινή γενιά
- Η οπτική φύση και η άμεση ανάδραση του προγράμματος, κάνει εύκολο στους μαθητές να δουν την επίδραση μιας εντολής που έχουν δώσει. Επιπλέον κάνει το τεστ του προγράμματος ευκολότερο
- Ο συντάκτης σκηνών που λειτουργεί με σύρσιμο και επιλογή, εμποδίζει τους μαθητές από το να κάνουν συντακτικά λάθη στα οποία οι αρχάριοι είναι ευάλωτοι.
- Οι κλάσεις και τα 3D αντικείμενα στο Alice παρέχουν μια εμφανή ιδέα και περιγραφή του αντικειμένου.

Χρησιμοποιώντας το Alice, δεν παραλήφθηκε καμία από τις βασικές έννοιες του προγραμματισμού –συναρτήσεις, μέθοδοι, κλάσεις, κληρονομικότητα- που χρησιμοποιούνταν στην Java και την C++. Στο Saint Joseph's University (SJU) αλλά και

στο και στο Ithaca College (IC) τα άτομα που ήταν νέα στην επιστήμη των υπολογιστών ήταν 25 με 30. Στο τέλος του πρώτου χρόνου οι μαθητές που εκδήλωναν ενδιαφέρον για μαθήματα προγραμματισμού ήταν ελάχιστοι και ειδικά αυτοί που δεν είχαν καθόλου υπόβαθρο άγγιζαν το 0%. Οπότε οι συμμετέχοντες στην έρευνα αυτή ήταν οι πρωτοετείς φοιτητές και κατά κύριο λόγο αυτοί που δεν είχαν επαφή με προγραμματισμό και μαθηματικά. Στο IC το Alice γινόταν παράλληλα με το βασικό μάθημα προγραμματισμού, ενώ στο SJU το μάθημα του Alice έλαβε χώρα πριν το βασικό μάθημα.

Επίσης παρακάτω φαίνονται και τα ποσοστά ένδειξης ενδιαφέροντος των μαθητών:

Από αυτό φαίνεται ότι το 88% του συνόλου των μαθητών που παρακολούθησαν το Alice σε αντίθεση με το μόλις 47% αυτών που δεν παρακολούθησαν αισθάνονται σίγουροι και θα παρακολουθούσαν και επόμενα μαθήματα προγραμματισμού.

Μετά από όλα αυτά η έρευνα καταλήγει στο ότι οι μαθητές που χρησιμοποίησαν το Alice, όχι μόνο θα συμμετείχαν και σε άλλα παρόμοια μαθήματα, αλλά θα παρότρυναν και τους συμμαθητές τους. Επίσης αυτοί που συμμετείχαν στο μάθημα με το Alice τα πήγαν εξίσου καλά όσο οι μαθητές που είχαν εμπειρία στον προγραμματισμό με Java και C++. Το Alice δηλαδή ανέβασε το επίπεδο προγραμματιστικών γνώσεων των μαθητών.

4. Τέλος γίνεται η παρουσίαση της εργασίας: Stephen Cooper, Computer Science Department, Saint Joseph's University, Philadelphia, Wanda Dann, Computer Science Department, Ithaca College, NY and Randy Pausch, Computer Science Department, Carnegie Mellon University, Pittsburgh, "Teaching objects-first in Introductory

Computer Science”, 34th SIGCSE technical symposium on Computer science education, 191-195, 2003.

Η μελέτη σχετίζεται με την στρατηγική διδασκαλίας εισαγωγικών θεμάτων προγραμματισμού η οποία βασίζεται στα αντικείμενα. Ασχολείται με την απεικόνιση και οπτικοποίηση των αντικειμένων και την χρήση 3D περιβάλλοντος για την κίνηση τους. Η νέα μέθοδος διδασκαλίας απασχολεί πολλούς καθηγητές προγραμματισμού επειδή έχει αποδειχτεί στατιστικά αλλά και ανεπίσημα ότι η απόδοση των μαθητών εξαρτάται και επηρεάζεται άμεσα από αυτή τη νέα προσέγγιση διδασκαλίας. Επίσης γίνεται μια σύγκριση από παιδαγωγικής άποψης της νέας αυτής προσέγγισης με άλλη σχετική εργασία. Η ACM υποστηρίζει ότι η τεχνική με βάση τα αντικείμενα είναι η πιο ενδιαφέρουσα και πιο αποτελεσματική τεχνική εκμάθησης εισαγωγικών αρχών προγραμματισμού. Αυτή η τεχνική δίνει έμφαση στις αρχές του αντικειμενοστραφούς προγραμματισμού και σχεδιασμού από το πρώτο κίόλας μάθημα. Η τεχνική ξεκινάει με τις έννοιες των αντικειμένων και της κληρονομικότητας, συνεχίζει με τις πιο παραδοσιακές δομές ελέγχου αλλά πάντα μέσα στα πλαίσια του αντικειμενοστραφούς σχεδιασμού. Για να γίνει μια ενδιαφέρουσα προσέγγιση αυτής της τεχνικής δημιουργήθηκαν αρκετά προγραμματιστικά εργαλεία που να προωθούν την τεχνική που έχει ως βάση τα αντικείμενα. Σε αυτή τη μελέτη γίνεται λόγος για το προγραμματιστικό εργαλείο Alice που έχει 3D περιβάλλον εργασίας. Το 3D περιβάλλον εργασίας βοηθάει στην νοερή απεικόνιση των αντικειμένων και τους μαθητές να καταλάβουν και να “δουν” τις βασικές αντικειμενοστραφείς αρχές. Έτσι αν κάποιος μαθητής δημιουργήσει ένα αντικείμενο βάτραχο δεν χρειάζεται να γράψει κώδικα, παρά να προσθέσει το αντικείμενο στο πρόγραμμα που υλοποιεί τον κόσμο του.

Μετά από 3 χρόνια χρήσης του Alice παρατηρήθηκε ότι οι μαθητές:

- Ανέπτυξαν ικανότητα σχεδιασμού προσεγμένων και αξιόλογου περιεχομένου προγραμμάτων
- Ανέπτυξαν ικανότητα δημιουργίας αξιόλογων εικονικών κόσμων. Στους κόσμους που δημιουργούσαν οι μαθητές τα πάντα ήταν αντικείμενα που εκτελούσαν κινήσεις όπως σε ταινίες και σε videogames
- Έμαθαν να τεστάρουν και να επιδιορθώνουν κώδικα. Επειδή οι μαθητές γνώριζαν ποιο κομμάτι κώδικα αναφέρεται σε κάποιο αντικείμενο, με ξεχωριστές αλλαγές και αφού έβλεπαν τι επίδραση έχει στο αντικείμενο έκαναν τις κατάλληλες επιδιορθώσεις
- Έμαθαν να δημιουργούν το πρόγραμμα τους κομμάτι κομμάτι και να κάνουν παράλληλα τεστ σωστής λειτουργίας
- Ασχολήθηκαν με την κληρονομικότητα. Οι μαθητές έγραφαν κώδικα ώστε να δημιουργήσουν μια πιο ισχυρή κλάση
- Συνεργάστηκαν. Οι μαθητές δημιουργούσαν τους δικούς τους χαρακτήρες και μετά τους συνδύαζαν ώστε να υλοποιήσαν ένα κοινό κόσμο
- Απέφυγαν τα λάθη. Οι μαθητές δεν μπορούν να κάνουν λογικά λάθη αφού ο συντάκτης τους απαγορεύει να δώσουν λανθασμένη τιμή σε μια μεταβλητή η σε ένα επαναληπτικό βρόγχο.

Τα αποτελέσματα αυτής της έρευνας φαίνονται παρακάτω:

Οι μαθητές στο Ithaca College και στο Saint Joseph's University παρακολούθησαν ένα μάθημα Alice το 2001 με 2002. Επιλέχθηκαν οι 21 πιο αδύναμοι μαθητές στο αρχικό μάθημα του προγραμματισμού. Από αυτούς οι 11 παρακολούθησαν το Alice ενώ οι 10



όχι. Τα αποτελέσματα δείχνουν ότι οι 11 μαθητές που παρακολούθησαν τα πήγαν καλύτερα όχι μόνο από τους άλλους 10, αλλά και από το σύνολο των 49 μαθητών της τάξης οι οποίοι μπορεί να είχαν ένα προγραμματιστικό υπόβαθρο. Επίσης από το σύνολο των μαθητών που παρακολούθησαν το Alice το 91% παρακολούθησε και στο επόμενο εξάμηνο μάθημα προγραμματισμού. Παρόλο που αυτά τα στοιχεία είναι ενδεικτικά και άκρως πειστικά, ο αριθμός των μαθητών που συμμετείχε σε αυτή την έρευνα είναι μικρός και έτσι οι έρευνα συνεχίζεται λεπτομερέστερα και για ένα μεγαλύτερο πλήθος μαθητών έτσι ώστε να προκύψουν πιο ισχυρά στοιχεία.

Υπάρχουν πολλοί λόγοι για να θέλετε να μάθετε να προγραμματίζετε. Για πολλούς ο προγραμματισμός είναι τρόπος διασκέδασης. Αλλά για τους περισσότερους ο προγραμματισμός είναι εργασία. Έχουν κάτι σημαντικό να κάνουν, και ο υπολογιστής αποτελεί το κατάλληλο εργαλείο. Στην πραγματικότητα οι εφαρμογές των υπολογιστών είναι τόσο διεισδυτικές στην κοινωνία μας, ώστε αν για παράδειγμα γεννιόσασταν αύριο θα είχατε επαφή με αυτές από την ημέρα της γέννησης μέχρι την τελευταία σας ημέρα. Πολλά νοσοκομεία δένουν ένα μικροτσιπ στον αστράγαλο των νεογέννητων για να γνωρίζουν που βρίσκονται για όλη τους τη ζωή. Την τελευταία σας ημέρα, είναι πιθανόν να έχετε ένα μόνιτορ ενός υπολογιστή που θα δείχνει το καρδιακό σήμα όταν θα βρίσκεστε στο κρεβάτι. Στο μεταξύ μπορείτε να ζήσετε πιο πολύ και πιο υγιείς, εξαιτίας των προόδων που έγιναν με την βοήθεια υπολογιστών στην ιατρική έρευνα, στην τεχνολογία, στα ρυθμιζόμενα από υπολογιστές φρένα και αερόσακους στα αυτοκίνητα και στην μοντελοποίηση των φαρμάκων από υπολογιστές που βοηθούν στην αντιμετώπιση ασθενειών όπως το AIDS. Οι προγραμματιστές υπολογιστών βοηθούν στο να γίνουν αληθινές όλες αυτές οι τεχνολογικές προόδους. Οι υπολογιστές και τα

προγράμματα που γράφονται έχουν ανατρέψει την βιομηχανία της ψυχαγωγίας. Τα παιχνίδια στους υπολογιστές έχουν γίνει άκρως δημοφιλή. Οι Pew Internet και American Life Project ανέφεραν (το 2003) ότι περίπου 70% των φοιτητών παίζουν διαδικτυακά παιχνίδια τουλάχιστον μια φορά την εβδομάδα. Τα γραφικά των ταινιών Star Wars γίνονται μόνο μέσω υπολογιστών. Παρεμπιπτόντως ένας προπτυχιακός φοιτητής, συγγραφέας της Alice, αποφοίτησε και πήγε για δουλειά στην ILM (Βιομηχανικό Φως και Μαγεία), και εκεί έκανε τα ειδικά εφέ για το Star Wars. Οι υπολογιστές μας βοηθούν στην επικοινωνία, υποστηρίζοντας σύνθετα κινητά δίκτυα τηλεφωνίας, βοηθούν στην θαλάσσια έρευνα παρακολουθώντας τα ίχνη των αποδημητικών ζώων και μας επιτρέπουν να εξερευνούμε το διάστημα. Κανένα από αυτά δεν θα ήταν δυνατό χωρίς τους υπολογιστές. Φυσικά πολλοί από τους ανθρώπους που γράφουν προγράμματα είναι επαγγελματίες που πέρασαν πολλά χρόνια μελετώντας. Αλλά ακόμη και οι άνθρωποι που δεν σκοπεύουν να γίνουν επαγγελματίες μπορούν να επωφεληθούν ακόμη και από ένα απλό μάθημα προγραμματισμού. Οι σύγχρονες εφαρμογές, όπως τα προγράμματα λογιστικού φύλλου και ο επεξεργαστής κειμένου, δίνουν στον τελικό χρήστη την ευκαιρία να κερδίσουν χρόνο και προσπάθεια χρησιμοποιώντας μακροεντολές ή άλλα χαρακτηριστικά του προγραμματισμού τα οποία υποχρεώνουν τον υπολογιστή να κάνει κάτι χρονοβόρο και βαρετό, αντί να το κάνει ο χρήστης. Επίσης αν έχετε μια μικρή εμπειρία στον προγραμματισμό μπορείτε να γίνετε το “χρήσιμο άτομο” στο γραφείο που δουλεύετε. Κυριότερα ένα μάθημα προγραμματισμού μπορεί να καλλιεργήσει ένα νέο τρόπο σκέψης, όπως και ένα μάθημα ζωγραφικής μπορεί να βοηθήσει στο πως μπορεί κανείς να δει τον κόσμο με διαφορετικό μάτι. Το να μάθετε να σκέφτεστε με νέους τρόπους είναι πάντα πολύ χρήσιμο. Πολλοί από εμάς αναζητούμε τρόπους για την

βελτίωση μας στην λύση προβλημάτων. Με τον όρο λύση προβλημάτων εννοούμε να δίνει κανείς την σωστή απάντηση σε ένα πρόβλημα ή να εκτελεί μια αποστολή. Ο προγραμματισμός υπολογιστών είναι ένα είδος επίλυσης προβλημάτων. Έτσι μαθαίνοντας να προγραμματίζετε ένα υπολογιστή θα σας βοηθήσει να αναπτύξετε ένα νέο τρόπο σκέψης- καθιστώντας σας ικανό να βρίσκετε απαντήσεις σε ερωτήσεις και να βρίσκετε πώς να επιλύετε προβλήματα.

### 3.3 Πώς θα μάθετε να προγραμματίζετε με το Alice

Πολλά αρχικά μαθήματα προγραμματισμού ξεκινούν με υπολογισμούς και εκτυπώσεις μέσω ορών και αθροισμάτων. Πολλές φορές οι μαθητές αποθαρρύνονται γιατί πρέπει να μάθουν πολλές τεχνικές λεπτομέρειες μέχρι να λειτουργήσουν όλα σωστά.. Το σύστημα Alice το οποίο παρέχεται δωρεάν ως δημόσια παροχή από το πανεπιστήμιο Carnegie Mellon, παρέχει μια εντελώς νέα προσέγγιση στην εκμάθηση προγραμματισμού. Αρχικά αναπτύχθηκε στα πλαίσια ενός ερευνητικού project για την εικονική πραγματικότητα. Το πρόγραμμα Alice σας επιτρέπει να είστε ο σκηνοθέτης μιας ταινίας, ή ο δημιουργός ενός video game, όπου τα 3D αντικείμενα κινούνται τριγύρω σε ένα εικονικό κόσμο στην οθόνη ανάλογα με τις κατευθύνσεις που τους δίνετε. Δεν χρησιμοποιείται δύσκολες προγραμματιστικές εντολές παρά απλές καθημερινές Αγγλικές λέξεις όπως: “προχώρησε μπροστά” ή “στρίψε δεξιά”. Το καλύτερο από όλα είναι ότι δεν μπορείτε να κάνετε λάθη! Βέβαια, πάντα μπορείτε να κάνετε κάποια λάθη όπως για παράδειγμα να πείτε το αντικείμενο να μετακινηθεί μπροστά ενώ εσείς θέλατε πίσω. Δεν μπορείτε όμως να κάνετε λάθη που δεν κατανοεί ο

υπολογιστής, όταν πληκτρολογήσετε κάτι λάθος, τα οποία έχουν ως αποτέλεσμα να μην τρέχει το πρόγραμμά σας καθόλου. Εάν ο όρος προγραμματιστής υπολογιστών σας φέρνει στο μυαλό ένα “ταλαίπωρο” άνθρωπο, πάνω από ένα πληκτρολόγιο σε ένα σκοτεινό δωμάτιο-μην ανησυχείτε! Δεν θα ακουμπήσετε σχεδόν ποτέ το πληκτρολόγιο όσο χρησιμοποιείται το Alice. Θα δημιουργήσετε προγράμματα επιλέγοντας λέξεις και αντικείμενα και μεταφέροντας τα στην οθόνη με το ποντίκι. Τότε μόλις θα πατήσετε το κουμπί “Play” (τα αντικείμενα στον 3D κόσμο της οθόνης σας θα ζωντανέψουν και θα ακολουθήσουν το σενάριο που τους έχετε γράψει! Έτσι το να είστε προγραμματιστής υπολογιστών χρησιμοποιώντας το Alice είναι σαν να είστε σκηνοθέτης ταινίας ή χορογράφος-ο καθένας που δίνει οδηγίες στους ανθρώπους για το τι να κάνουν με ένα ακριβές και περιορισμένο λεξιλόγιο.

Αφού μάθατε πώς να χρησιμοποιείτε το Alice, θα καταλάβετε όλες τις θεμελιώδεις ιδέες του προγραμματισμού. Τότε θα είστε έτοιμοι να χρησιμοποιήσετε μια από τις συνηθισμένες γλώσσες προγραμματισμού στις οποίες πρέπει να πληκτρολογήσετε και να βάλετε όλα τα κόμματα και τις τελείες στην σωστή θέση. Θα γνωρίζετε πώς να προγραμματίζετε και το μόνο που θα πρέπει να μάθετε είναι οι συγκεκριμένοι κανόνες σύνταξης των γλωσσών όπως Java, C++, C# ή όποια άλλη. Τα βασικά του προγραμματισμού υπολογιστών Ένα πρόγραμμα στον υπολογιστή είναι τίποτα περισσότερο από μία σειρά οδηγιών οι οποίες καθοδηγούν τον υπολογιστή τι να κάνει. Φυσικά υπάρχουν πολλοί τρόποι με τους οποίους μπορείτε να πείτε στον υπολογιστή να κάνει κάτι, οπότε παίζει ρόλο το πώς θα το κάνετε. Οι προγραμματιστές υπολογιστών συχνά χρησιμοποιούν όρους όπως “κομψό” για να περιγράψουν καλογραμμένα προγράμματα. Συνιστούμε να βλέπετε ένα πρόγραμμα υπολογιστή όχι

μόνο ως τον τρόπο που θα πείτε στον υπολογιστή τι να κάνει άλλα και ως ένα τρόπο να πείτε σε έναν άλλο άνθρωπο το τι θέλετε να κάνει ο υπολογιστής. Ένα πρόγραμμα υπολογιστή δεν είναι μόνο ο τρόπος για να πείτε στον υπολογιστή τι να κάνει. Ένα πρόγραμμα υπολογιστή είναι και ο τρόπος να πείτε σε έναν άλλο άνθρωπο τι θέλετε να κάνει ο υπολογιστής. Αυτό κάνει πολύ ευκολότερο το να κριθεί κάτι ως κομψό. Ένα πρόγραμμα είναι κομψό εάν άλλοι άνθρωποι μπορούν εύκολα να καταλάβουν και να εκτιμήσουν τις προθέσεις του αρχικού προγραμματιστή. Γι αυτό το λόγο, ένα βασικό θέμα στην γραφή ενός προγράμματος είναι να εμπεριέχει τεκμηρίωση (σχόλια στο πρόγραμμα, μια ιστοσελίδα για παραπομπή ή ένα συνοδευτικό έγγραφο) η οποία βοηθάει κάποιο άλλο άτομο να καταλάβει τι προσπαθείτε να κάνετε. Το κλειδί στον προγραμματισμό υπολογιστών είναι να καταλάβετε τις βασικές θεμελιώδεις ιδέες. Ο προγραμματισμός είναι πολύ εύκολος. Όλα τα προγράμματα αποτελούνται από απλά σχέδια: Μία λίστα οδηγιών: Για παράδειγμα, “Χτυπήστε τα αυγά, αναμίξτε με την ζάχαρη, χύστε το μίγμα σε ένα ταψί και ψήστε στους 375 βαθμούς για 45 λεπτά.” Οι επιστήμονες υπολογιστών το ονομάζουν διαδοχική επεξεργασία. If (υποθέσεις): Για παράδειγμα, “Εάν βρέχει, πάρε μια ομπρέλα.” Οι επιστήμονες υπολογιστών το ονομάζουν εκτέλεση υπό συνθήκη (εκτέλεση υπό όρους). Επαναληπτική συμπεριφορά: Για παράδειγμα, “Χτύπα το πόδι πέντε φορές” ή “Όσο υπάρχουν κουλουράκια στο πιάτο, συνέχισε να τρως.” Οι επιστήμονες υπολογιστών το ονομάζουν επανάληψη. Να χωρίζετε τα πράγματα σε μικρότερα κομμάτια: Για παράδειγμα, “Ο τρόπος που θα καθαρίσεις το σπίτι είναι πρώτα να καθαρίσεις την κουζίνα, μετά το μπάνιο, μετά καθάρισε κάθε ένα από τα τρία δωμάτια.” Οι επιστήμονες υπολογιστών το ονομάζουν αποσύνθεση προβλήματος, ή βηματική διύλιση ή από πάνω προς τα κάτω σχεδίαση, αλλά είναι

πραγματικά μια παλιά φιλοσοφική προσέγγιση που καλείται υπεραπλούστευση. Όπως και να το ονομάσετε σημαίνει ότι μια πολύπλοκη διεργασία, χωρίζεται σε μικρότερες και απλούστερες διεργασίες. Το αποτέλεσμα της εκτέλεσης των απλών διεργασιών είναι η εκτέλεση και της σύνθετης διεργασίας. Υπολογισμός του αποτελέσματος: Εδώ εκτελούμε μια ακολουθία βημάτων ώστε να επιτύχουμε ένα αποτέλεσμα που είναι απάντηση σε μια ερώτηση. Για παράδειγμα: “Κοίταξε στον τηλεφωνικό κατάλογο και βρες τον αριθμό της Ρεβέκα Σμιθ, ” ή “Βάλε το μωρό στην ζυγαριά και πες μου πόσα κιλά είναι.” Στην πραγματικότητα κάθε μια από αυτές τις ενέργειες εμπεριέχει μια ερώτηση: “Ποιο είναι το τηλέφωνο της Ρεβέκας;” ή “Πόσο ζυγίζει το μωρό;” Μια ερώτηση είναι γνωστή ως συνάρτηση στον προγραμματισμό υπολογιστών. Κάνοντας μια ερώτηση για να υπολογιστεί το αποτέλεσμα (η απάντηση) είναι γνωστό ως κλήση συνάρτησης. Ο προγραμματισμός υπολογιστών είναι η χρήση όλων αυτών των ιδεών σε διάφορους συνδυασμούς. Αυτό που μπορεί να κάνει τα πράγματα δύσκολα είναι η πολυπλοκότητα. Η αλήθεια είναι ότι οι περισσότεροι υπολογιστές “αναγνωρίζουν” μόνο, περίπου 100 διαφορετικές οδηγίες. Τα εκατομμύρια προγράμματα που υπάρχουν στους υπολογιστές χρησιμοποιούν αυτές τις ίδιες 100 οδηγίες με διαφορετική σειρά και συνδυασμούς. Οπότε που υπάρχει η πολυπλοκότητα; Σκεφτείτε το ως εξής: Σε ένα παιχνίδι σκάκι, υπάρχουν μόνο έξι είδη πιονιών και κάθε κομμάτι κάνει μια απλή κίνηση. Το σκάκι είναι ένα σύνθετο παιχνίδι εξαιτίας όλων των πιθανών συνδυασμών κινήσεων. Διαφορετικά, γράφοντας ένα πρόγραμμα είναι σαν να βάζετε σε ένα θεατρικό έργο 200 ηθοποιούς, 500 κοστουμια και 5 καμήλες μαζί σε μια συγκεκριμένη σκηνή. Τα πράγματα μπορούν να δυσκολέψουν μόνο με την επίβλεψη όλων αυτών των αντικειμένων. Στην πραγματικότητα, το να μάθετε πώς να βάζετε σε σειρά μια

ακολουθία οδηγιών για να φέρετε σε πέρας μια διεργασία (πώς να σχεδιάσετε ένα πρόγραμμα) είναι το πολυτιμότερο μέρος στην εκμάθηση προγραμματισμού.

### **Γιατί καλείται Alice;**

Πρώτα απ' όλα, το Alice δεν είναι αρκτικόλεξο. Δεν είναι A.L.I.C.E. Η ομάδα ονόμασε το σύστημα Alice στην μνήμη του Charles Lutwidge Dodson, έναν άγγλο μαθηματικό που υπέγραφε με το όνομα Lewis Carroll. Ο Carroll έγραψε το έργο "Οι περιπέτειες της Alice στην χώρα των θαυμάτων". Όπως οι άνθρωποι που δημιούργησαν το Alice έτσι και ο Lewis Carroll ήταν ικανός να επιλύσει πολύπλοκα μαθηματικά, αλλά γνώριζε ότι το σημαντικότερο ήταν να κάνει τα πράγματα απλά και συναρπαστικά σε ένα μαθητή. Με τον ίδιο τρόπο με τον οποίο η Alice ήταν διστακτική όταν αρχικά πέρασε μέσα από το καθρέφτη, με τον ίδιο τρόπο μπορεί να έχετε κάποιες αμφιβολίες σχετικά με την εκμάθηση ενός προγράμματος.

Ο προγραμματισμός με το σύστημα Alice σημαίνει ότι θα δημιουργήσετε πραγματικούς κόσμους με διάφορα αντικείμενα στον υπολογιστή σας. Μετά θα γράψετε προγράμματα για να κατευθύνετε τα δικά σας κινούμενα σχέδια.

### **Ένας εικονικός κόσμος**

Τα video games και οι προσομοιώσεις μπορούν να είναι είτε δυο, είτε τριών διαστάσεων (2D, 3D). Μπορεί να έχετε χρησιμοποιήσει ένα 2D προσομοιωτή γραφικών σε κάποιο μάθημα οδήγησης. Ένα μέρος της εκπαίδευσης των πιλότων είναι η χρήση προσομοιωτών πτήσης. Το πλεονέκτημα των προσομοιώσεων είναι εμφανές-όταν ένας αρχάριος πιλότος καταστρέψει ένα πολεμικό αεροπλάνο, ούτε ο πιλότος άλλα ούτε το αεροπλάνο δεν βρίσκεται σε κίνδυνο. Ένα video game ή μια προσομοίωση υλοποιημένα

σε 3D καλείται εικονικός κόσμος. Χρησιμοποιώντας ένα εικονικό κόσμο, ο προσομοιωτής γίνεται ρεαλιστικότερος και αποτελεσματικότερος.

### Αντικείμενα και 3D μοντέλα

Ένα από τα ευχάριστα σημεία στην χρήση του Alice είναι η φαντασία που απαιτείται για την δημιουργία νέων κόσμων. Ξεκινάμε με μια απλή σκηνή και προσθέτουμε αντικείμενα. Στον κόσμο μας τα αντικείμενα είναι ένα δέντρο, ένας φράχτης, μια χελώνα και ένας λαγός. Μερικά αντικείμενα αποτελούν το σταθερό σκηνικό (δέντρα, σπίτια, ουρανός και άλλα). Άλλα αντικείμενα (άνθρωποι, ζώα, διαστημόπλοια) παίζουν τον ρόλο των ηθοποιών στο σκηνικό σας (κινούνται τριγύρω, μιλάνε). Για να διευκολυνθεί ο χρήστης στην δημιουργία ενός νέου κόσμου με διάφορα αντικείμενα το Alice παρέχει ένα μεγάλο αριθμό 3D μοντέλων. Ένα 3D μοντέλο είναι σαν ένα μηχανολογικό σχέδιο ενός σπιτιού. Το σχέδιο παρέχει ένα μοντέλο του σπιτιού το οποίο αναλύει την όψη του, την τοποθεσία και το μέγεθος των δωματίων και κάποιες πληροφορίες που θα ακολουθήσει ο εργολάβος για το χτίσιμο του σπιτιού. Παρόμοια ένα μοντέλο του Alice δίνει οδηγίες για το πώς θα δημιουργηθεί ένα νέο αντικείμενο στη σκηνή. Το 3D μοντέλο παρέχει πληροφορίες για το πώς θα σχεδιαστεί το αντικείμενο, τι χρώμα θα έχει, από τι κομμάτια θα αποτελείται, το μέγεθος του και πολλές άλλες λεπτομέρειες. Με την εγκατάσταση του λογισμικού στον υπολογιστή σας, εγκαθίσταται και μια τοπική βιβλιοθήκη που περιέχει διάφορα έτοιμα 3D μοντέλα. Πρόσθετα μοντέλα μπορεί να βρεθούν στην ιστοσελίδα (<http://www.alice.org>). Τα μοντέλα που παρέχονται στην βιβλιοθήκη είναι εύκολο να διαχειριστούν μέσω του διαχειριστή σκηνών. Η Alice δεν είναι ένα πρόγραμμα δημιουργίας 3D γραφικών. Γι αυτό το λόγο παρέχονται



βιβλιοθήκες με έτοιμα μοντέλα. Παρόλα αυτά είναι απίθανο να καλυφθούν οι απαιτήσεις όλων των χρηστών. Έτσι μπορείτε να φτιάξετε τον δικό σας χαρακτήρα με την βοήθεια των εργαλείων hebuilder και shebuilder που βρίσκονται στο φάκελο των ανθρώπων της τοπικής βιβλιοθήκης.

### Τρεις και έξι κατευθύνσεις

Τα αντικείμενα στο κόσμο του Alice είναι τριών διαστάσεων. Κάθε αντικείμενο έχει μήκος, πλάτος και βάθος. Το ύψος μετριέται κατά μήκος μιας νοητής γραμμής που είναι κατακόρυφη από πάνω προς τα κάτω, το πλάτος κατά μήκος μιας νοητής γραμμής που είναι οριζόντια από αριστερά προς τα δεξιά και το βάθος κατά μήκος μιας νοητής γραμμής από μπρος προς τα πίσω.

### Τρεις διαστάσεις

Όσο αναφορά αυτές τις τρεις διαστάσεις το αντικείμενο γνωρίζει ποιός δρόμος είναι πάνω ή κάτω σε σχέση με τον εαυτό του. Επίσης το αντικείμενο γνωρίζει το αριστερά και το δεξιά, το μπρος και το. Αυτό ισοδυναμεί με έξι πιθανές κατευθύνσεις στις οποίες μπορεί να κινηθεί ένα αντικείμενο. Αυτό είναι, το αντικείμενο έχει έξι βαθμούς ελευθερίας για να κινηθεί σε ένα κόσμο. Είναι σημαντικό να προσέξετε ότι οι κατευθύνσεις είναι δεξιά και αριστερά σε σχέση με τον πρωταγωνιστή, όχι με την θέση της κάμερας. Καλούμε έξι βαθμούς ελευθερίας (πιθανές κατευθύνσεις κίνησης) τον προσανατολισμό του αντικειμένου. Όταν κάνετε κλικ με το ποντίκι σε ένα αντικείμενο, εμφανίζεται ένα κίτρινο περίβλημα. Το κίτρινο περίβλημα αυτό τονίζει το αντικείμενο.

## Το κέντρο ενός αντικειμένου

Κάθε αντικείμενο στο πρόγραμμα Alice έχει ένα μοναδικό κέντρο. Το κεντρικό αυτό σημείο δεν υπολογίζεται. Είναι ένα χαρακτηριστικό κάθε αντικειμένου το οποίο ορίζεται από τον γραφίστα όταν δημιουργείται το 3D μοντέλο. Συνήθως το κεντρικό σημείο ενός μοντέλου βρίσκεται στο κέντρο του κίτρινου περιβλήματος του- ή κοντά στο κέντρο βάρους. Το κεντρικό σημείο παρέχει πληροφορία για τον άξονα περιστροφής. Έτσι μερικά αντικείμενα όπως ένας τροχός ή ένα πουλί θα περιστρέφεται γύρω από το κεντρικό σημείο του.

Δεν έχουν όλα τα αντικείμενα το κέντρο στο κέντρο της μάζας του. Αυτά τα οποία γενικότερα κάθονται ή στέκονται στο έδαφος ή σε ένα τραπέζι, έχουν το κέντρο τους στη βάση του κίτρινου περιβλήματός του. Για τα αντικείμενα- ανθρώπους, το κεντρικό σημείο βρίσκεται ανάμεσα στα πόδια τους. Αυτό γίνεται γιατί τα πόδια των ανθρώπων βρίσκονται στο έδαφος και η απόσταση του ανθρώπου από το έδαφος είναι μηδέν μέτρα. Άλλα είδη αντικειμένων που δεν έχουν το κέντρο τους στο κέντρο της μάζας τους είναι αυτά που “κρατούνται” όταν τα χρησιμοποιεί κανείς, για παράδειγμα ένα ρόπαλο του baseball. Το κέντρο ενός ροπάλου του baseball βρίσκεται εκεί που κρατείται. Το κέντρο βρίσκεται στην λαβή έτσι ώστε όταν περιστρέφεται, θα κινείται γύρω από το σημείο.

## Απόσταση

Η απόσταση ενός αντικειμένου από ένα άλλο μετριέται από το κέντρο του. Για παράδειγμα, η απόσταση του πουλιού από το έδαφος μετριέται από το κεντρικό σημείο του πουλιού.

## Θέση

Το κέντρο ενός αντικειμένου είναι το σημείο που δείχνει την θέση του στον κόσμο. Το σύστημα Alice ορίζει αυτόματα το κέντρο του εδάφους ως το κέντρο του κόσμου. Εστω ότι τρεις άξονες συντεταγμένων τοποθετούνται στο κέντρο του εδάφους. Στην λίστα ιδιοτήτων του εδάφους μπορείτε να δείτε ότι το κέντρο του εδάφους βρίσκεται στο σημείο (0,0,0) των αξόνων συντεταγμένων. Όπως το έδαφος έτσι και κάθε αντικείμενο τοποθετείται σε σχέση με το κέντρο του κόσμου. Το πουλί βρίσκεται στην θέση (-3,41, 1,59, 6,15). Αυτό σημαίνει ότι το κέντρο του πουλιού βρίσκεται 3,41 μέτρα αριστερά, 1,59 μέτρα πάνω και 6,15 μέτρα εμπρός από το κέντρο του κόσμου.

## Κίνηση

Στο Alice θα δημιουργήσετε εικονικούς κόσμους και θα δώσετε ζωή κινώντας τα αντικείμενα στον κόσμο με τον ίδιο τρόπο που κινούνται τα αντικείμενα σε ένα εξομοιωτή πτήσης ή σε ένα videogame. Θα χρησιμοποιήσετε πολλές τεχνικές κίνησης ίδιες με αυτές που χρησιμοποιούνται στα κινούμενα σχέδια στα φιλμ της Disney και της Pixar. Η ζωή των αντικειμένων είναι μια οφθαλμαπάτη, ένα φανταστικό θέαμα. Για να δημιουργηθεί αυτή η οφθαλμαπάτη, ο παραγωγός του φιλμ συνεργάζεται με τον ηθοποιό για να δημιουργηθεί μια ακολουθία καλλιτεχνικών σκηνών (ζωγραφίες και εικόνες), όπου η κάθε μια έχει μια μικρή διαφορά από την προηγούμενη. Η σκηνή σχεδιάζεται με αντικείμενα και μετά ξανασχεδιάζεται με τα αντικείμενα σε μια ελαφρά αλλαγμένη θέση. Η σκηνή σχεδιάζεται ξανά και τα αντικείμενα κινούνται λίγο ακόμη, λίγο ακόμη, λίγο ακόμη, λίγο ακόμη κτλ. Στην παραγωγή κινουμένων σχεδίων, τα πλαίσια σκηνών αποτυπώνονται σε μια ακολουθία πάνω σε ένα καρούλι φιλμ ή φωτογραφίζονται από μια

ψηφιακή κάμερα. Το φιλμ παίζεται σε ένα προτζέκτορα ή μια οθόνη, εμφανίζοντας γρήγορα πολλές εικόνες που υπάρχουν στην ακολουθία δημιουργώντας έτσι την ψευδαίσθηση της κίνησης. Το λογισμικό Alice δημιουργεί ένα παρόμοιο εφέ στην οθόνη του υπολογιστή. Δεν χρειάζεται να ανησυχείτε αν δεν είστε τέλειος καλλιτέχνης. Το Alice δημιουργεί από μόνο του την ακολουθία των σκηνών. Εσείς απλά είστε ο σκηνοθέτης και δίνετε οδηγίες για το τι να κάνει το κάθε αντικείμενο. Το Alice δημιουργεί (αποδίδει) την κίνηση.

### **Ξεκίνημα με το Alice**

Σας συνιστούμε να πειραματιστείτε με το Alice, όσο θα εξερευνούσατε το καινούριο σας κινητό τηλέφωνο. Το βγάξετε από το κουτί και δοκιμάζετε όλα τα χαρακτηριστικά του. Με τον ίδιο τρόπο μπορείτε να μάθετε πώς να χειρίζεστε το λογισμικό Alice.

### **3D Κείμενα**

Για να προσθέσετε ένα 3D κείμενο σε ένα κόσμο, κάντε κλικ στην επιλογή “Δημιουργία 3D κειμένου” στην τοπική βιβλιοθήκη. Ένα πλαίσιο κειμένου εμφανίζεται. Εκεί πληκτρολογείτε ένα κείμενο σε όποια μορφή θέλετε (για παράδειγμα έντονα, γράμματα, πλάγια γραφή). Όταν πατήσετε το OK, το λογισμικό προσθέτει ένα αντικείμενο κειμένου στον κόσμο και μια καταχώρηση του αντικειμένου στο δένδρο των αντικειμένων. Το όνομα του αντικειμένου είναι το ίδιο με το όνομα του κειμένου που πληκτρολογήσατε. Το αντικείμενο μπορεί να τοποθετηθεί χρησιμοποιώντας το ποντίκι με τον ίδιο τρόπο που γίνεται για κάθε άλλο αντικείμενο. Για να τροποποιήσετε το

κείμενο στο αντικείμενο, κάντε κλικ στο κείμενο στην λίστα ιδιοτήτων του καταλόγου λεπτομερειών. Τότε εισάγετε ένα νέο κείμενο στο πλαίσιο κειμένου. Προσέξτε ότι τροποποιώντας το κείμενο, δεν αλλάζει το όνομα του αντικειμένου. Το όνομα είναι ίδιο όπως ήταν αρχικά.

### **Γραφικές εικόνες (Billboards)**

Παρόλο που το Alice είναι ένα 3D σύστημα, είναι πιθανό να δημιουργηθούν επίπεδες 2D εικόνες σε ένα σκηνικό. Οι επίπεδες 2D εικόνες μπορούν να δημιουργηθούν σε ένα οποιοδήποτε εργαλείο ζωγραφικής και να σωθούν σε GIF, JPG ή TIF. Για να προσθέσετε μια 2D εικόνα (το Alice την ονομάζει billboard) στον κόσμο σας, επιλέξτε από το μενού File το Make Billboard. Στο πλαίσιο διαλόγου, επιλέξτε την αποθηκευμένη εικόνα και μετά κάντε κλικ στο κουμπί εισαγωγή.

Το Alice θα προσθέσει την επίπεδη εικόνα στον κόσμο. Το billboard υλοποιεί μια από τις χρήσεις των billboard – παρέχοντας πληροφορίες στον χρήστη για το πώς να παίξει ένα παιχνίδι.

### **Σχεδιασμός και υλοποίηση προγράμματος**

Πρόγραμμα είναι ένα σύνολο πληροφοριών που επεξηγεί στον υπολογιστή τι να κάνει. Κάθε πληροφορία είναι μια πράξη που πρέπει να εκτελεστεί. Το να γράψετε ένα πρόγραμμα για κίνηση 3D αντικειμένων σε ένα εικονικό κόσμο σχετίζεται με τα αντικείμενα και τις πράξεις που μπορούν να εκτελέσουν. Από μια πλευρά το γράψιμο ενός προγράμματος είναι σαν να λύνετε προβλήματα στα μαθηματικά. Πρώτα διαβάζουμε το πρόβλημα (μια περιγραφή της κατάστασης) και αποφασίζουμε πώς θα το λύσουμε (τι βήματα θα ακολουθήσουμε). Μετά επιλύουμε το πρόβλημα / γράφουμε μια

λύση και τεστάρουμε την απάντηση για να δούμε αν είναι σωστή. Παρόμοια, στο γράψιμο ενός προγράμματος που περιέχει κίνηση αντικειμένων, πρώτα διαβάζουμε το σενάριο (μια περιγραφή ιστορίας, παιχνιδιού-συχνά καλείται δήλωση προβλήματος) και αποφασίζουμε πως θα δώσουμε στα αντικείμενα ζωή (σχεδιασμός μιας σειράς σκηνών). Μετά γράφουμε τον κώδικα του προγράμματος (υλοποίηση) και το τεστάρουμε τρέχοντας το πρόγραμμα μας. Πρέπει να πείτε ακριβώς τι εννοείτε όταν γράφετε ένα πρόγραμμα. Ο καλύτερος τρόπος να γράψετε ένα πρόγραμμα είναι να ξεκινήσετε διαβάζοντας ένα σενάριο / περιγραφή ιστορίας, παιχνιδιού και μετά να σχεδιάσετε μια λίστα ενεργειών για το πρόγραμμα. Οι σειρές σκηνών που βασίζονται σε κείμενο, παρέχουν μια αλγοριθμική δομή (βήμα προς βήμα). Οι γραμμές του κειμένου σε μια σειρά σκηνών που βασίζονται σε κείμενο είναι παρόμοιες με τον ψευδοκώδικα-μια απλή έκδοση των οδηγιών που θα γίνουν τελικά κώδικας προγράμματος. Η ιδέα είναι να χρησιμοποιηθεί μια σειρά σκηνών σαν οδηγός για το γράψιμο ενός προγράμματος σε ένα συντάκτη του Alice που βασίζεται στο ποντίκι. Μπορούμε να εστιάσουμε σε μια βήμα προς βήμα λύση επειδή το Alice θα φροντίσει αυτόματα για όλες τις λεπτομέρειες του “συντακτικού” (δομή και στίξη των εντολών). Σε μια κίνηση, μερικές ενέργειες πρέπει να γίνουν διαδοχικά και μερικές ταυτοχρόνως. Αυτό σημαίνει ότι ο κώδικας του προγράμματος πρέπει να είναι δομημένος ώστε να λείει στο λογισμικό ποιες ενέργειες να γίνουν διαδοχικά και ποιες να γίνουν ταυτόχρονα.

### Σενάρια και σκηνές έργου

Η δημιουργία ενός προγράμματος που κινεί τα αντικείμενα σε ένα οπτικό κόσμο είναι μια διαδικασία τεσσάρων βημάτων: Διαβάστε το σενάριο (περιγραφή ενός

προβλήματος ή διεργασίας), σχεδιασμός (σχέδιο), υλοποίηση (γράψτε το πρόγραμμα) και τεστ (δείτε αν λειτουργεί). Αυτό το τμήμα παρουσιάζει τα πρώτα δύο βήματα. Το διάβασμα του σεναρίου και ο σχεδιασμός του πλάνου δράσης είναι σημαντικά βήματα στην κατασκευή ενός προγράμματος. Ένα σχέδιο είναι ένα “πλάνο” στρατηγικής. Διαβάστε το σενάριο. Πριν να συζητήσουμε πώς να δημιουργήσουμε ένα σχέδιο, πρέπει να γνωρίζουμε τι είδους πρόβλημα πρόκειται να λυθεί ή τι είδους διεργασία πρέπει να εκτελεστεί. Ένα σενάριο είναι μια δήλωση προβλήματος (ή διεργασίας) η οποία περιγράφει την συνολική κίνηση σε σχέση με το τι πρόβλημα πρέπει να λυθεί ή τι μάθημα πρέπει να διδαχθεί. (Πολλοί επιστήμονες πληροφορικής χρησιμοποιούν τον όρο προδιαγραφές απαιτήσεων. Στο Alice, ο όρος σενάριο είναι ευκολότερο να σχετισθεί με την σκηνή του κόσμου, τα αντικείμενα και τις πράξεις). Τα κινούμενα σχέδια και τα φιλμ ξεκινούν με ένα σενάριο που δημιουργείται από επαγγελματίες συγγραφείς, που καλείται “ιστορία”. Όπως χρησιμοποιείται εδώ, σε σχέση με το αρχικό νόημα, μια ιστορία μπορεί να είναι η διδασκαλία ενός μαθήματος, ένα παιχνίδι ή ένα κινούμενο σχέδιο. Σε ένα κόσμο του Alice, ένα σενάριο παρέχει όλες τις απαραίτητες λεπτομέρειες για το στήσιμο του αρχικού σκηνικού και μετά ο σχεδιασμός μιας ακολουθίας πληροφοριών για την επίτευξη της κίνησης. Αυτό είναι, ένα σενάριο παρέχει απαντήσεις στις παρακάτω ερωτήσεις:

1. Τι ιστορία έχει ειπωθεί;
2. Τι αντικείμενα χρειάζονται; Μερικά αντικείμενα θα παίξουν ηγετικό ρόλο στην ιστορία και μερικά άλλα θα χρησιμοποιηθούν απλά για το πίσω μέρος του σκηνικού.
3. Τι ενέργειες πρέπει να γίνουν; Οι ενέργειες στην ιστορία θα αποτελέσουν τελικά τις πληροφορίες του προγράμματος.

## Παράδειγμα σεναρίου

Ας θεωρήσουμε ένα σενάριο προκειμένου να καταλάβουμε πώς μπορούμε να καταλάβουμε τι ζητάει ένα πρόβλημα: Μετά από ένα ταξίδι στο διάστημα, ένα διαστημόπλοιο επανδρωμένο με ρομπότ έχει προσγειωθεί στο φεγγάρι. Το ρομπότ έχει βγει από το διαστημόπλοιο και έχει στήσει μια κάμερα έτσι ώστε οι επιστήμονες που βρίσκονται στο κέντρο της NASA στο Houston να δουν το ιστορικό γεγονός. Στο σκηνικό μας μπορούμε να δούμε το ρομπότ, το διαστημόπλοιο, το σεληνιακό έδαφος και μερικούς βράχους. Ξαφνικά ένας εξωγήινος κρυφοκοιτάζει πίσω από ένα βράχο το ρομπότ. Το ρομπότ εκπλήσσεται και γυρνάει το κεφάλι του ακριβώς προς τον εξωγήινο. Το ρομπότ περπατάει προς τον εξωγήινο για να ρίξει μια πιο κοντινή ματιά, με αποτέλεσμα ο εξωγήινος να κρυφτεί πίσω από τους βράχους. Τελικά το ρομπότ κοιτάζει στην κάμερα, κάνει σήμα κινδύνου, και λέει “Houston έχουμε πρόβλημα!”

Από αυτό το σενάριο έχουμε απαντήσεις στις ερωτήσεις:

- Τι ιστορία έχει ειπωθεί; Αυτό το σενάριο περιγράφει μια χιουμοριστική ιστορία για μια συνάντηση ενός ρομπότ και ενός εξωγήινου στο φεγγάρι.
- Τι αντικείμενα χρησιμοποιούνται; Τα αντικείμενα είναι το ρομπότ, το διαστημόπλοιο και ο εξωγήινος. Το σκηνικό είναι η επιφάνεια του φεγγαριού σε ένα διαστημικό κόσμο.
- Τι ενέργειες θα γίνουν; Οι ενέργειες είναι το κρυφοκοίταγμα πίσω από το βράχο, το γύρισμα του κεφαλιού του ρομπότ και η κίνηση του προς τον εξωγήινο, το κρύψιμο του εξωγήινου πίσω από το βράχο και η αποστολή μηνύματος από το ρομπότ στην γη.



## Σχεδιασμός

Μια διάταξη σκηνών είναι η σχεδιαστική προσέγγιση που θα χρησιμοποιήσουμε για να επιλύσουμε το πρόβλημα ή για να σχεδιάσουμε μια λίστα ενεργειών ώστε να εκτελεστεί η διεργασία που καθορίζεται στο σενάριο. Στην Disney και στην Pixar και σε άλλα μεγάλα στούντιο που κάνουν animation, οι εργαζόμενοι αποσυνθέτουν ένα μεγάλο και μακρύ σενάριο σε ακολουθίες πολλών μικρών σεναρίων. Για κάθε σενάριο, δημιουργείται μια διάταξη σκηνών ώστε να αναπαρασταθεί η ακολουθία των σκηνών. Η διάταξη των σκηνών μπορεί να αποτελείται από δεκάδες σκίτσα, που ζωγραφίζονται από καλλιτέχνες ή δημιουργούνται από ανιματέρ υπολογιστών που χρησιμοποιούν ειδικό λογισμικό. Παρουσιάζονται σκίτσα διαφόρων σκηνών από το “παιχνίδι του Geri”, ένα μικρό φιλμ από την Pixar που έχει γραφτεί από τον Jan Pinkava. Το φιλμ κέρδισε όσκαρ για την καλύτερη ταινία με animation. Σε αυτό το φιλμ, ο χαρακτήρας παίζει σκάκι με τον εαυτό του.

Η προσέγγιση της αποσύνθεσης ενός προβλήματος ή μιας διεργασίας σε υποπροβλήματα ή υποδιεργασίες δεν χρησιμοποιείται μόνο από τους προγραμματιστές υπολογιστών και τους ανιματέρ. Οι συγγραφείς θεατρικών έργων, διασπούν τα έργα τους σε ξεχωριστές πράξεις και τις πράξεις σε ξεχωριστές σκηνές! Οι μηχανικοί διαλύουν τα πολύπλοκα συστήματα (π.χ. αεροπλάνα) ή τις συσκευές (π.χ. μικροκυκλώματα) στα κομμάτια που τα συνθέτουν για να κάνουν το πρόβλημα πιο διαχειρίσιμο.

### Οπτικές διατάξεις σκηνών έργου

Μια οπτική διάταξη σκηνών “σπάει” ένα σενάριο σε μια ακολουθία σκηνών με μετάβαση μεταξύ των σκηνών. Κάθε σκίτσο αναπαριστά ένα στιγμιότυπο μιας σκηνής

(κατάσταση) κατά την διάρκεια της κίνησης. Κάθε στιγμιότυπο σχετίζεται με αντικείμενα σε συγκεκριμένες θέσεις, χρώμα, μέγεθος και πόζα. Όταν συμβαίνουν μια ή περισσότερες μεταβολές στο κινούμενο σχέδιο, οδηγούμαστε στην επόμενη σκηνή. Τα στιγμιότυπα αριθμίζονται σε μια ακολουθία και παρέχουν κατάλληλες πληροφορίες. Για μικρού μήκους κινούμενα σχέδια, η ταξινόμηση μπορεί να αναπαρασταθεί σε μια μεγάλη σελίδα φύλλου. Για πιο πολύπλοκα σχέδια, ένα ξεχωριστό φύλλο ζωγραφικής μπορεί να χρησιμοποιηθεί για κάθε σκηνή, επιτρέποντας στον σχεδιαστή να παραλείπει ή να αλλάξει την σειρά των σκηνών χωρίς να ξεκινήσει από την αρχή. Για να δημιουργήσουμε μια οπτική διάταξη σκηνών δανειζόμαστε μια τεχνική επαγγελματιών ανιματερ-μια ακολουθία από ζωγραφισμένες στο χέρι σκηνές. Κάθε στιγμιότυπο παρέχει πληροφορίες για τον αριθμό σκηνής και περιέχει ένα σκίτσο ή μια εικόνα που δείχνει που βρίσκονται τα αντικείμενα στη σκηνή. Η περιγραφή εξηγεί τι ενέργεια γίνεται. Εάν στα κινούμενα σχέδια είναι κατάλληλος ένας ήχος, η περιγραφή θα περιέχει μια λίστα ήχων που θα παίζει κατά τη διάρκεια της σκηνής. Επίσης μπορεί να χρησιμοποιηθεί και κείμενο. Ο ήχος και το κείμενο χρησιμοποιούνται μόνο εάν χρειάζεται.

Για εμάς εδώ δεν θα δώσουμε βάση σε σκίτσα άκρως καλλιτεχνικά. Απλοί κύκλοι, τετράγωνα και γραμμές μπορούν να χρησιμοποιηθούν για να αναπαραστήσουν αντικείμενα που εμφανίζονται στην σκηνή. Εάν είναι απαραίτητο, τα σχήματα μπορούν να ονομαστούν με το όνομα ή τον κωδικό του χρώματος του αντικειμένου.

### **Σειρές σκηνών που βασίζονται σε κείμενο**

Ενώ οι επαγγελματίες ανιματέρ χρησιμοποιούν διατάξεις σκηνών με εικόνες, δεν έχουν όλοι την υπομονή να φτιάξουν δεκάδες σκίτσα. Μια σκηνή που βασίζεται σε

κείμενο είναι μια εναλλακτική λύση. Είναι σαν μια “λίστα υποχρεώσεων ” και μας επιτρέπει να σχεδιάσουμε μια δομή για τον κώδικα του προγράμματος.

### **Τι είναι ένα πρόγραμμα;**

Όπως ξέρετε, ένα πρόγραμμα είναι μια λίστα εντολών (ενεργειών) για να επιτευχθεί μια διεργασία. Μπορείτε να φανταστείτε ένα πρόγραμμα στο Alice όπως ένα σενάριο σε ένα θεατρικό έργο. Ένα θεατρικό κείμενο διηγείται μια ιστορία περιγράφοντας τις πράξεις που θα γίνουν και τα λόγια που θα εκφωνήσουν οι ηθοποιοί στην σκηνή. Παρόμοια ένα πρόγραμμα στο Alice καθορίζει τις πράξεις που θα γίνουν, τους ήχους και τα κείμενα που θα χρησιμοποιηθούν από τα αντικείμενα σε ένα εικονικό κόσμο.

## ΚΕΦΑΛΑΙΟ 4

### Ο μικρόκοσμος COSY-Java

#### 4.1 Εκπαιδευτικό σενάριο

##### Γενικά

Ομάδες μαθητών καλούνται να κάνουν χρήση των δυνατοτήτων που παρέχουν οι νέες τεχνολογίες στην εκπαίδευση, προκειμένου να δημιουργήσουν ιστορίες με ποικίλη θεματολογία.

Οι μαθητές έχουν στη διάθεσή τους ένα εργαστήριο υπολογιστών με την κατάλληλη δικτυακή και διαδικτυακή υποδομή, ενώ αντιστοιχεί ένας υπολογιστής σε κάθε ομάδα μαθητών των δύο ατόμων. Οι μαθητές θα κάνουν χρήση του εκπαιδευτικού λογισμικού Alice.

##### Ένταξη του Σεναρίου στο Αναλυτικό Πρόγραμμα Σπουδών

Τάξη:	ΣΤ΄ Δημοτικού
Τομέας:	Όλα τα δημοτικά σχολεία
Μάθημα:	Πληροφορική
Διδακτικές ώρες:	2

##### Εκπαιδευτική στρατηγική

Το σενάριο βασίζεται σε εμπλουτισμένη παραλλαγή της εκπαιδευτικής στρατηγικής «Παιχνίδι Ρόλων» (Role Play) και χωρίζεται σε διάφορες δραστηριότητες.

Η εισαγωγική δραστηριότητα, που αφορά στη διδασκαλία στην τάξη, βασίζεται στις στρατηγικές «Καταιγισμός ιδεών» (Brainstorming).

### **Περιγραφή εκπαιδευτικού σεναρίου**

Ο καθηγητής έχει αναλάβει να δημιουργήσει ένα ηλεκτρονικό βιβλίο για παιδιά. Προκειμένου να το πραγματοποιήσει ζητά από τους μαθητές να δημιουργήσουν αυτοτελείς ιστορίες προκειμένου να απαρτήσουν το ηλεκτρονικό αυτό βιβλίο παραμυθιών.

### **Διδακτικοί Στόχοι**

Ως αποτέλεσμα της εκπαιδευτικής διαδικασίας του σεναρίου οι μαθητές αναμένεται να είναι σε θέση :

- Να δημιουργήσουν ένα ολοκληρωμένο πρόγραμμα
- Να κατανοήσουν τα πλεονεκτήματα του αντικειμενοστραφή προγραμματισμού
- Να συνεργάζονται με άλλους για την επίλυση προβλημάτων
- Να αντιληφθούν την ανάγκη επικοινωνίας και συνεργασίας με τους άλλους
- Να ενισχυθεί το ενδιαφέρον των μαθητών για θέματα που αφορούν τον προγραμματισμό

### **Περιγραφή Δραστηριοτήτων**

Το εκπαιδευτικό σενάριο αποτελείται από δραστηριότητες. Η Δραστηριότητα 0 επιτελείται όποτε το επιθυμούν οι ομάδες των μαθητών. Οι επόμενες δραστηριότητες

αφορούν στην προετοιμασία, στη συνεργατική επίλυση του προβλήματος που περιγράφεται στο σενάριο και στη συγγραφή της εργασίας των μαθητών. Εκτενέστερη αναφορά στην περιγραφή των δραστηριοτήτων γίνεται στο αναλυτικό πρόγραμμα της διδασκαλίας.

### Περιγραφή ρόλων συμμετεχόντων

Στην πραγματοποίηση των δραστηριοτήτων καθηγητής και μαθητές έχουν συγκεκριμένους ρόλους:

#### Ρόλος Καθηγητή

- Ο καθηγητής αντιπροσωπεύει τον υπεύθυνο συλλογής σεναρίων.
- Προετοιμάζει τους μαθητές για τις δραστηριότητες που πρόκειται να πραγματοποιήσουν
- Καθορίζει τη σύνθεση των ομάδων φροντίζοντας να είναι ισοδύναμες.
- Εγκαθιστά στους υπολογιστές του εργαστηρίου το απαιτούμενο λογισμικό (Alice)
- Κατά την πραγματοποίηση των δραστηριοτήτων παρέχει βοήθεια σχετικά με τη χρήση του απαραίτητου λογισμικού.
- Αξιολογεί κάθε ομάδα και παρέχει ανατροφοδότηση.

Ο Καθηγητής έχει στη διάθεσή του:

- Έναν φορητό υπολογιστή και ένα βιντοπροβολέα για χρήση στην σχολική τάξη.
- Στο εργαστήριο της πληροφορικής διαθέτει επίσης έναν υπολογιστή (server) με πρόσβαση στο τοπικό δίκτυο και το διαδίκτυο.

- Το λογισμικό Alice.

### **Ρόλος μαθητή**

Οι μαθητές χωρίζονται σε ομάδες των δύο ατόμων. Κάθε μαθητής αναλαμβάνει να φανταστεί μια ιστορία και να τη δημιουργήσει με το διπλανό του.

### **Αξιολόγηση**

Η αξιολόγηση των ομάδων βασίζεται στη συμμετοχή και στην ολοκλήρωση της ιδέας.

Κάθε ομάδα μαθητών έχει στη διάθεσή της τα εξής:

- Έναν υπολογιστή με πρόσβαση στο τοπικό δίκτυο και το διαδίκτυο.
- Λογισμικό: Alice

### **Απαραίτητοι Τεχνολογικοί Πόροι**

- Ένα εργαστήριο πληροφορικής με εννέα τουλάχιστον Η/Υ (ένας Η/Υ για κάθε ομάδα δύο μαθητών) και ένας Η/Υ που θα λειτουργεί σαν Server
- Το εργαστήριο πληροφορικής θα πρέπει να υποστηρίζεται από ένα τοπικό δίκτυο LAN, και να διαθέτει σύνδεση με το διαδίκτυο.
- Κάθε Η/Υ πρέπει να έχει εγκατεστημένο το λογισμικό Alice.

## Ανάλυση Δραστηριοτήτων

Δραστηριότητα 0:

### α. Γενικά

Η Δραστηριότητα αφορά την εισαγωγή εννοιών προγραμματισμού. Επεξήγηση βασικών εννοιών, και διαχωρισμός αντικειμενοστραφή προγραμματισμού με παραδείγματα.

### β. Αναλυτική Περιγραφή Δραστηριότητας

#### Ρόλος μελών κάθε ομάδας

Όλα τα μέλη των ομάδων, μπορούν να επικοινωνούν ελεύθερα θέτοντας ερωτήσεις και απαντώντας σε άλλες σε οποιαδήποτε χρονική στιγμή.

#### Στάδιο 1ο: Προετοιμασία

Ο καθηγητής κάνει κατά τη διάρκεια του μαθήματος και πριν την κοινοποίηση του σεναρίου στις ομάδες, μια εισαγωγή στη δραστηριότητα, αποσαφηνίζοντας τους διδακτικούς της στόχους.

#### Στάδιο 2ο: Παρουσίαση

Οι μαθητές βλέπουν για πρώτη φορά το λογισμικό Alice και πλοηγούνται σε αυτό. Κατανοούν τη διαφορά από τη Logo βλέποντας ότι στην Alice προγραμματίζουν με τη μέθοδο «σέρνω – αφήνω». Είναι ελεύθεροι να κάνουν ερωτήσεις.



### Στάδιο 3ο: Εφαρμογή

Κάθε μαθητής μπορεί να κάνει τα εξής:

- Να θέσει κάποιο ερώτημα που τον απασχολεί σαν μέλος της ομάδας του ή εκπροσωπώντας όλη την ομάδα.

### Στάδιο 4ο: Αξιολόγηση

Η δραστηριότητα δεν περιλαμβάνει αξιολόγηση. Όταν όμως μαθητές απαντούν σωστά σε ερωτήσεις συμμαθητών τους ή όταν θέτουν ενδιαφέρουσες ερωτήσεις, ανεβάζουν τον προφορικό τους βαθμό, όπως ακριβώς θα συνέβαινε αν απαντούσαν σωστά στην τάξη κατά τη διάρκεια του μαθήματος.

#### γ. Προσδοκώμενα αποτελέσματα

Δραστηριότητα	Ώρες / Χώρος	Προσδοκώμενα αποτελέσματα
Δραστηριότητα 0 Θεωρεία και παρουσίαση λογισμικού	1/ Εργαστήριο	Οι μαθητές κατανοούν βασικές έννοιες προγραμματισμού καθώς και τα πλεονεκτήματα του Alice.

## **Δραστηριότητα 1:**

Προετοιμασία στην τάξη.

α. Γενικά

Η Δραστηριότητα γίνεται στο εργαστήριο και αφορά στην παροχή όλων των πληροφοριών που θα χρειασθούν οι μαθητές, για να πραγματοποιήσουν την εργασία τους.

β. Αναλυτική Περιγραφή Δραστηριότητας

### **Στάδιο 1ο: Προετοιμασία**

Ο καθηγητής, κατά τη διάρκεια του μαθήματος και πριν την κοινοποίηση του σεναρίου στις ομάδες, κάνει μια εισαγωγή στη δραστηριότητα, αποσαφηνίζοντας τους διδακτικούς της στόχους.

### **Στάδιο 2ο: Παρουσίαση**

Οι μαθητές ενημερώνονται για την ύπαρξη και τον τρόπο χρήσης του λογισμικού Alice, για τις εργασίες που πρόκειται να πραγματοποιήσουν, χωρίζονται σε ομάδες, και αναλαμβάνουν ρόλους.

### **Στάδιο 3ο: Εφαρμογή**

Στο στάδιο αυτό ο καθηγητής μοιράζει στην κάθε ομάδα ένα φυλλάδιο με οδηγίες (όπως φαίνεται στο παράρτημα). Οι μαθητές ακολουθούν τις οδηγίες βήμα βήμα και ολοκληρώνουν την ιστορία.

## Δραστηριότητα 2:

Προετοιμασία στην τάξη.

α. Γενικά

Η Δραστηριότητα γίνεται στο εργαστήριο και αφορά στην παροχή όλων των πληροφοριών που θα χρειασθούν οι μαθητές, για να πραγματοποιήσουν την εργασία τους.

β. Αναλυτική Περιγραφή Δραστηριότητας

### Στάδιο 1ο: Προετοιμασία

Ο καθηγητής, κατά τη διάρκεια του μαθήματος και πριν την κοινοποίηση του σεναρίου στις ομάδες, κάνει μια εισαγωγή στη δραστηριότητα, αποσαφηνίζοντας τους διδακτικούς της στόχους.

### Στάδιο 2ο: Παρουσίαση

Ο καθηγητής ενημερώνει τους μαθητές για τις εργασίες που πρόκειται να πραγματοποιήσουν.

### Στάδιο 3ο: Εφαρμογή

Στο στάδιο αυτό ο καθηγητής ζητά από τους μαθητές να δημιουργήσουν τη δική τους ιστορία με τη βοήθεια του φυλλαδίου και του καθηγητή.

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην διπλωματική εργασία αυτή παρουσιάστηκε το εκπαιδευτικό προγραμματιστικό εργαλείο Alice. Η έκδοση που μελετήθηκε είναι το Alice v2.0, παρόλα αυτά το περιβάλλον είναι παρόμοιο και για τις προηγούμενες εκδόσεις. Αρχικά έγινε μια μελέτη και αξιολόγηση του Alice και στην συνέχεια αναπτύχθηκε διδακτικό υλικό για το πρόγραμμα Alice για οποιονδήποτε χρήστη.

Οι μαθητές της έκτης δημοτικού που χρησιμοποίησαν το Alice, έχουν διδαχθεί τη γλώσσα Logo από την Τετάρτη δημοτικού. Τα τμήματα είναι των 33 ατόμων σε ομάδες των 2. Οι παρατηρήσεις που έκανα είναι οι εξής:

- Οι μαθητές δούλεψαν με ενθουσιασμό και συνεργάστηκαν πολύ καλά.
- Δεν υπήρχαν διαφωνίες και καθυστερήσεις γιατί δεν πληκτρολόγούσαν εντολές αλλά μοίρασαν πιο εύκολα τις κινήσεις που θα έκανε ο καθένας.
- Τους άρεσε περισσότερο από τη Logo γιατί δε χρειαζόταν να γράφουν κώδικα, έτσι δεν έκαναν συχνά λάθη.
- Τους άρεσαν πολύ τα γραφικά και η ιδέα της δημιουργίας τρισδιάστατων κόσμων.
- Οι μαθητές της πρωτοβάθμιας εκπαίδευσης μπορούν εύκολα να δημιουργήσουν προγράμματα αρκεί αυτό να γίνει χρησιμοποιώντας το κατάλληλο εργαλείο.
- Ένας τρόπος να ενδιαφερθούν είναι τα τρισδιάστατα γραφικά.
- Με τη μέθοδο «σέρνω - αφήνω» δε χρειάζεται πληκτρολόγηση, απομνημόνευση ή χρόνος για να δημιουργήσουν μια ιστορία. Δεν κάνουν συντακτικά λάθη.
- Τους ενθουσίασε η τεράστια ποικιλία σε αντικείμενα και πρωταγωνιστές.

Για τους απλούς ενδιαφερόμενους οι οποίοι δεν έχουν προγραμματιστική εμπειρία το Alice είναι το κατάλληλο εργαλείο. Η δημιουργία ενός προγράμματος με το Alice είναι τόσο εύκολη και διασκεδαστική που κάθε απλός χρήστης θα μπορούσε να κάνει το δικό του πρόγραμμα. Γι αυτό το λόγο το Alice έχει εξελιχθεί σε ένα από τα κυριότερα εργαλεία εκμάθησης προγραμματισμού σε διάφορα σχολεία και γνωστά Πανεπιστήμια της Αμερικής. Οι μαθητές πλέον βρίσκουν ευχάριστο το μάθημα του προγραμματισμού αφού κάθε κόσμος που δημιουργούν μπορούν να τον δουν να εκτελείται άμεσα, να τον τροποποιήσουν και να πειραματιστούν πάνω σε αυτόν. Εξάλλου οι έρευνες που έγιναν σε σχολεία και πανεπιστήμια και τα στοιχεία έδειξαν ότι όχι μόνο αυξήθηκαν οι μαθητές που επιλέγουν το μάθημα του προγραμματισμού, όταν αυτό διδάσκεται με το Alice, αλλά και οι επιδόσεις τους στο μάθημα είναι πολύ καλύτερες κατά μέσο όρο από αυτές άλλων μαθητών που διδάσκονται προγραμματισμό με ένα άλλο εργαλείο. Η απήχηση του προγραμματιστικού εργαλείου Alice στο ευρύ κοινό έγινε αρωγός στο να θελήσουν οι ιδρυτές του να το αναβαθμίσουν. Για το λόγο αυτό προέκυψε η συνεργασία του Πανεπιστημίου Carnegie Mellon με την EA (Electronic Arts), εταιρία δημιουργίας και παραγωγής videogames. Με την συνεργασία αυτή το Alice θα αναβαθμίσει τις βιβλιοθήκες του με την προσθήκη των χαρακτήρων του παιχνιδιού Sims 2 της EA. Η αναβάθμιση αυτή των χαρακτήρων θα οδηγήσει στη δημιουργία του προγραμματιστικού εργαλείου Alice v3.0. Το προγραμματιστικό εργαλείο Alice v3.0 αναμένεται να έχει μεγαλύτερη απήχηση στους μαθητές διότι οι χαρακτήρες του Sims 2, λόγω του παιχνιδιού, είναι οικείοι στα περισσότερα άτομα νεαρής ηλικίας. Αν το Alice v3.0 έχει την απήχηση που αναμένεται τότε θα καθιερωθεί ως το βασικό εκπαιδευτικό εργαλείο εκμάθησης προγραμματισμού στα σχολεία της

Αμερικής. Θα ήταν χρήσιμο και ενδιαφέρον να ερευνηθούν στο μέλλον οι αλλαγές που θα γίνουν στο πρόγραμμα Alice 3.0 και κατά πόσο αυτό θα έχει την αναμενόμενη επιτυχία.

Πανεπιστήμιο Πειραιώς

## ΠΑΡΑΡΤΗΜΑ

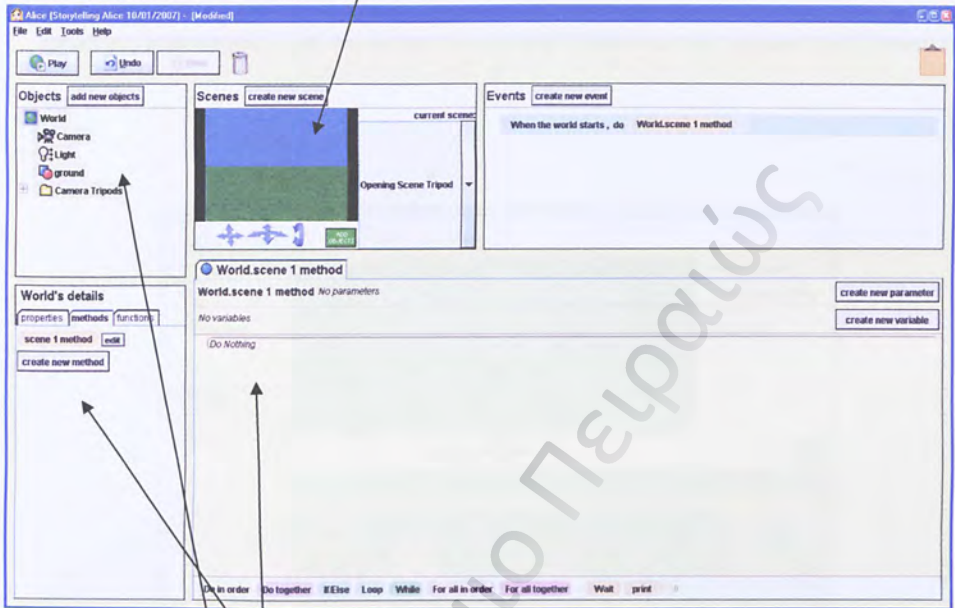
Στη δραστηριότητα αυτή θα μάθεις πώς να δημιουργείς ένα δικό σου κόσμο, να διαλέγεις δικούς σου χαρακτήρες, καθώς και πώς να κάνεις τους χαρακτήρες σου να κινούνται και να εκτελούν τις εντολές σου!

Ας ξεκινήσουμε ρίχνοντας μια ματιά στα βασικά σημεία του περιβάλλοντος Alice.

Ανοίγοντας το Alice, εμφανίζεται το παρακάτω παράθυρο. Μπορείς να επιλέξεις πού θες να διαδραματίζεται ο κόσμος σου, σε πράσινο, στην άμμο, στη θάλασσα και λοιπά. Για να γίνει αυτό επιλέγεις Templates και αφού διαλέξεις τι σου αρέσει πατάς Open.



Αυτό είναι το παράθυρο του κόσμου σου, εδώ θα βλέπεις ό,τι επιλέγεις να συμβεί στην ιστορία σου.



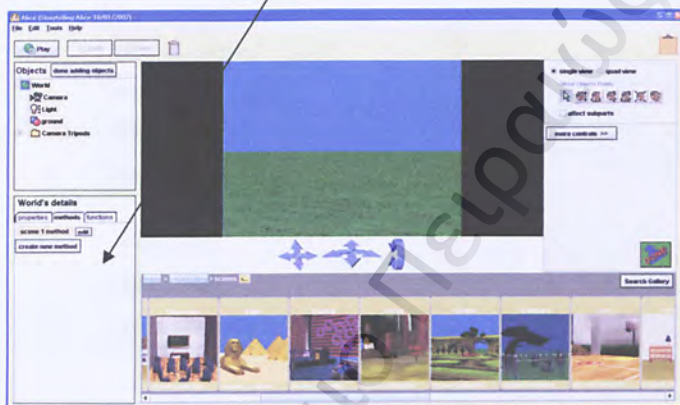
Εδώ είναι το Object Tree. Περιέχει μια λίστα με τους χαρακτήρες και τις σκηνές που έχεις συμπεριλάβει στον κόσμο σου.

Στο κομμάτι αυτό είναι η Details Area. Σου δίνει λεπτομέρειες για τον κόσμο σου, Εμφανίζει τις δυνατότητες των αντικειμένων και των χαρακτήρων σου.

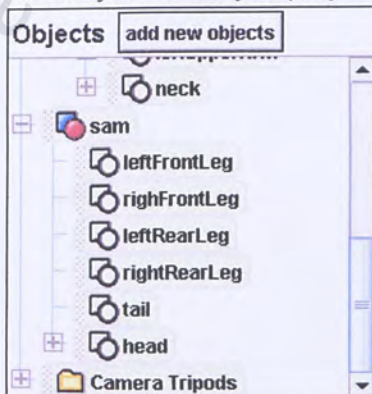
Αυτή είναι η Editor Area. Εδώ δίνεις τις οδηγίες για το τι ακριβώς θέλεις να γίνει και με ποιά σειρά.

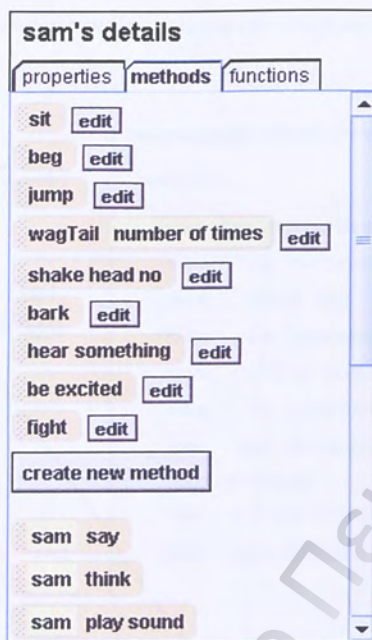


Για να επιλέξεις χαρακτήρες πάτησε το κουμπί Add new objects από το Object tree. Στη συνέχεια επέλεξε το φάκελο Scenes για να διαλέξεις σκηνικό και Characters για να επιλέξεις χαρακτήρες. Εδώ θα δούμε το σκηνικό Forest και τους χαρακτήρες Trevor και Sam.



Κάνοντας 2 φορές κλικ πάνω σε ένα χαρακτήρα ή αντικείμενο στο Object tree, μπορείς να δεις τι κινήσεις ή μεθόδους ξέρει να κάνει. Στη συνέχεια μπορείς να χρησιμοποιήσεις τις κινήσεις ή τις μεθόδους αυτές ή με βάση αυτές να μάθεις τους χαρακτήρες σου να κάνουν άλλες πιο σύνθετες κινήσεις.





Αυτές είναι όλες οι μέθοδοι που γνωρίζει ο Σαμ ο σκύλος.

Εχουμε πρωταγωνιστές στην ιστορία μας τον Τρέβορ και τον Σαμ τον σκύλο. Εχοντας δει τις κινήσεις που μπορούν να κάνουν οι πρωταγωνιστές μας αποφασίζουμε ποιές από αυτές θα χρησιμοποιήσουμε.

Για να δημιουργήσεις την ιστορία σου ακολούθησε τα παρακάτω βήματα:

The image shows the Scratch IDE interface. On the left, the 'trevor's details' panel is open to the 'methods' tab. It lists several methods for the character 'trevor', including 'hug who', 'shove who', 'cheer howManyTimes', and 'dance how many times', each with an 'edit' button. Below these is a 'create new method' button and a list of existing methods: 'trevor say', 'trevor think', 'trevor play sound', 'trevor walk to', and 'trevor walk offscreen'. On the right, the 'World.scene 1 method' panel shows a script with the following code: 'trevor.cheer howManyTimes = 2', 'trevor say It's time to teach you some tricks more...', 'trevor walk to sam more...', 'trevor say I will show you how to lie down! more...', 'trevor lie down ground more...', 'trevor say It's your turn now more...', 'sam think You are crazy more...', 'sam.shake head no', 'sam turn away from trevor more...', and 'sam move forward 5 meters more...'. Two arrows point from the 'cheer howManyTimes' method in the left panel to the corresponding line of code in the script area.

**Βήμα 1:** Για να κάνεις τον Τρέβρο να χοροπηδά χαρούμενος, κάνε κλικ και σύρε τη μέθοδο «cheer howManyTimes» και στη λίστα που θα εμφανιστεί επέλεξε πόσες φορές θέλεις να χοροπηδήσει, για παράδειγμα 2 φορές.

**Βήμα 2:** Τώρα θέλουμε να πει στο σκύλο του τον Σαμ ότι είναι ώρα να του μάθει κάποια κόλπα. Θα πάρεις λοιπόν τη μέθοδο Trevor say και θα τη σύρεις κάτω από την προηγούμενη. Στη λίστα που θα εμφανιστεί μπορείς να επιλέξεις “other” και να πληκτρολογήσεις τα λόγια που θες χωρίς εισαγωγικά, εδώ γράψε “It’s time to teach you some tricks”.

**Βήμα 3:** Επειτα, πρέπει ο Τρέβρο να περπατήσει προς το μέρος του Σαμ. Σύρε τη μέθοδο Trevor walk to κάτω από την προηγούμενη και στη λίστα που εμφανιστεί επέλεξε Sam The entire Sam.

**Βήμα 4:** Τώρα ο Τρέβορ λέει στο σκύλο «Θα σου δείξω πώς να ξαπλώνεις κάτω». Για να γίνει αυτό επέλεξε τη μέθοδο Trevor say και σύρε την στη δεξιά στήλη, από τη λίστα επέλεξε other και πληκτρολόγησε “I will show you how to lie down!” χωρίς εισαγωγικά.


**Βήμα 5:** Για να κάνεις τον Τρέβορ να ξαπλώσει κάνε κλικ στη μέθοδο Trevor lie down και σύρε τη στη δεξιά στήλη, ενώ από τη λίστα επέλεξε ground.

**Βήμα 6:** Στη συνέχεια ο Τρέβορ λέει στο σκύλο του ότι τώρα είναι η σειρά του. Για να γίνει αυτό επέλεξε τη μέθοδο Trevor say και σύρε την στη δεξιά στήλη, από τη λίστα επέλεξε other και πληκτρολόγησε “It’s your turn now” χωρίς εισαγωγικά.

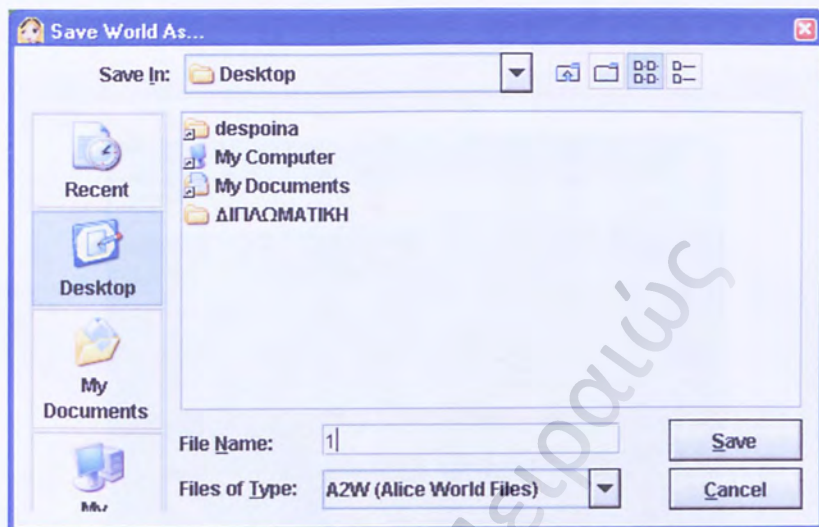
**Βήμα 7:** Κάνε τώρα δυο φορές κλικ πάνω στο χαρακτήρα του σκύλου Σαμ για να εμφανιστούν οι μέθοδοι που ξέρει να κάνει. Κάνε κλικ και σύρε τη μέθοδο Sam think στη δεξιά στήλη και στη λίστα που θα εμφανιστεί επέλεξε other πληκτρολογώντας “You are crazy” χωρίς εισαγωγικά.

**Βήμα 8:** Στη συνέχεια ο Σαμ κουνά το κεφάλι του αρνητικά, γιατί κάνει κλικ και σύρε τη μέθοδο Sam.shake head no. στη δεξιά στήλη κάτω από την προηγούμενη μέθοδο.

**Βήμα 9:** Τέλος, ο Σαμ φεύγει τρέχοντας μακριά από τον Τρέβορ. Για να το επιτύχει αυτό πρέπει να κάνεις κλικ και να σύρεις τη μέθοδο Sam move στη λίστα που θα εμφανιστεί επέλεξε forward και πάλι στη λίστα επέλεξε την απόσταση που θέλεις, εδώ 10 μέτρα.

**Βήμα 10:** Για να δεις την ιστορία που έφτιαξες πάτα το Play  πάνω αριστερά.

**Βήμα 11:** Για να αποθηκεύσεις την ιστορία σου κάνε κλικ επάνω αριστερά στο File και επέλεξε Save World as.. θα εμφανιστεί το παρακάτω παράθυρο:



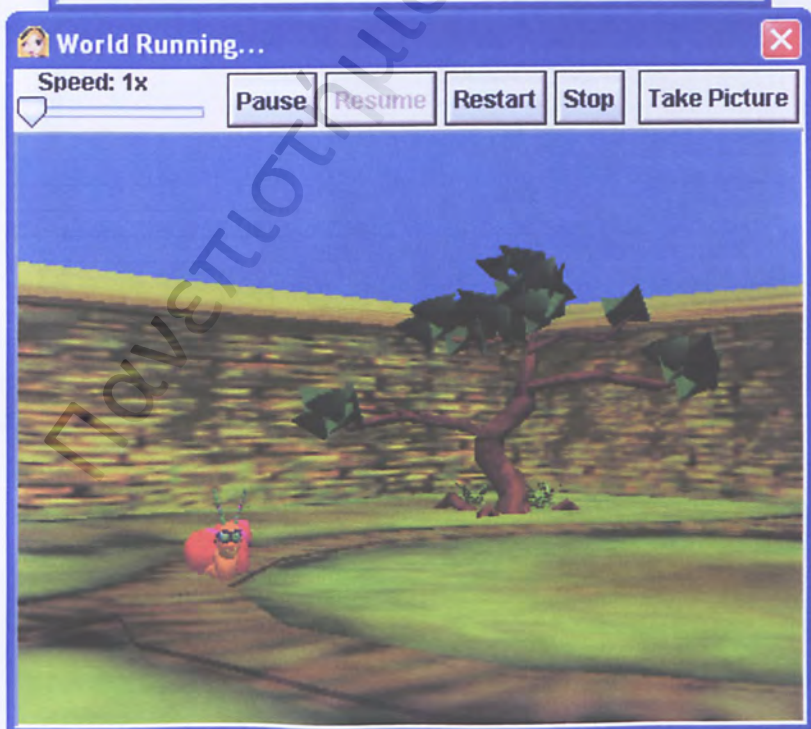
Επίλεξε σε ποιά φάκελο θέλεις να αποθηκεύσεις την εργασία σου και πληκτρολόγησε στο πεδίο File Name: τον αριθμό 1 (εφόσον είναι η πρώτη σου ιστορία) και πάτα Save.

Μόλις δημιούργησες και αποθήκευσες την πρώτη σου εργασία. Συγχαρητήρια!!!

## ΣΤΙΓΜΙΟΤΥΠΑ ΑΠΟ ΚΟΣΜΟΥΣ ΜΑΘΗΤΩΝ











## Βιβλιογραφικές Αναφορές

1. Adams J., *Alice in Action with Java*, Thomson Course Technology, 2007.
2. Adams J., “Alice, Middle Schoolers and the Imaginary World Camps”, *38th SIGCSE technical symposium on Computer science education*, 307-311, 2007.
3. Cooper S., Dann W., Pausch R., “Teaching objects-first in Introductory Computer Science”, *34th SIGCSE technical symposium on Computer science education*, 191-195, 2003.
4. Dann W., Cooper S., Pausch R., *Learning to Program with Alice*, Pearson Prentice Hall, 2006.
5. Herbert C., *An Introduction to Programming Using Alice*, Thomson Course Technology, 2006.
6. Kelleher C., Pausch R., Kiesler S., “Storytelling Alice motivates middle school girls to learn computer programming”, *SIGCHI conference on Human factors in computing systems*, 1455-1464, 2007.
7. Moskal B., Lurie D., Cooper S., “Evaluating the Effectiveness of a New Instructional Approach”, *35th SIGCSE technical symposium on Computer science education*, 75-79, 2004.
8. Shelly G., Cashman T., Herbert C., *Alice 2.0: Introductory Concepts and Techniques*, Thomson Course Technology, 2006.
9. <http://www.acm.org/>
10. <http://www.alice.org>
11. <http://www.alicetik2o.com/rpp/index.htm>

12. <http://www.cs.cmu.edu/~stage3/publications/95/journals/IEEEcomputer/CGandA/paper.html>

Πανεπιστήμιο Πειραιώς