

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ «ΕΠΙΣΤΗΜΕΣ ΤΗΣ ΑΓΩΓΗΣ»

ΚΑΤΕΥΘΥΝΣΗ: «ΦΥΣΙΚΕΣ ΕΠΙΣΤΗΜΕΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ»

ΠΑΙΔΑΓΩΓΙΚΟ ΤΜΗΜΑ ΔΗΜΟΤΙΚΗΣ ΕΚΠΑΙΔΕΥΣΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ



**«ΕΚΠΑΙΔΕΥΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ ΣΤΗΝ ΠΡΩΤΟΒΑΘΜΙΑ  
ΕΚΠΑΙΔΕΥΣΗ. ΜΙΑ ΣΥΓΚΡΙΤΙΚΗ ΕΜΠΕΙΡΙΚΗ ΜΕΛΕΤΗ»**



Μεταπτυχιακός φοιτητής: Παπαδόπουλος Βασίλειος

Επιβλέπων Καθηγητής: Μικρόπουλος Αναστάσιος

ΙΩΑΝΝΙΝΑ 2017

## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Αναστάσιο Μικρόπουλο, καθηγητή του Παιδαγωγικού Τμήματος Δημοτικής Εκπαίδευσης του Πανεπιστημίου Ιωαννίνων για τη στήριξη, την εμπιστοσύνη, αλλά και την πολύτιμη καθοδήγηση και συμβολή του στην ολοκλήρωση αυτής της εργασίας.

Ένα μεγάλο ευχαριστώ στον κ. Αναστάσιο Εμβαλωτή, αναπληρωτή καθηγητή του Παιδαγωγικού Τμήματος Δημοτικής Εκπαίδευσης του Πανεπιστημίου Ιωαννίνων, για τις πολύτιμες συμβουλές του στη μεθοδολογία έρευνας, καθώς και για τη βοήθεια που μου παρείχε.

Ακόμα ένα ευχαριστώ στον κ. Κωνσταντίνο Τάτση, επίκουρο καθηγητή του Παιδαγωγικού Τμήματος Δημοτικής Εκπαίδευσης του Πανεπιστημίου Ιωαννίνων, για τη συνδρομή του, ως μέλους της τριμελούς επιτροπής να διαβάσει την εργασία μου.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους γονείς μου, στα αδέρφια μου Δημήτρη και Αλέξανδρο, στη σύζυγο μου Ελένη και περισσότερο απ' όλους στον γιο μου Παναγιώτη για την υπομονή, την κατανόηση που μου έδειξαν και αποτέλεσαν πηγή έμπνευσης καθ' όλη τη διάρκεια των σπουδών μου στο πλαίσιο του μεταπτυχιακού προγράμματος. Σας ευχαριστώ πολύ.

## ΠΕΡΙΛΗΨΗ

Η παρούσα μελέτη φιλοδοξεί να παρουσιάσει τα αποτελέσματα της συγκριτικής ανάλυσης των Προγραμματιστικών Περιβαλλόντων για Αρχαρίους (ΠΠΑ) βάσει των παιδαγωγικών και τεχνολογικών χαρακτηριστικών εδραζόμενη στις ομοιότητες και τις διαφορές τους και αφετέρου στο πώς μπορούν αυτά τα επιλεγμένα περιβάλλοντα να αξιοποιηθούν σε μια εμπειρική μελέτη καταγραφής της πορείας των μαθητών. Τρία προγραμματιστικά περιβάλλοντα αρχαρίων διαφορετικών τεχνολογικών και παιδαγωγικών χαρακτηριστικών, το Code.org το StarlogoTNG και το Scratch αξιοποιήθηκαν και χρησιμοποιήθηκαν στη διδασκαλία διάφορων προγραμματιστικών εννοιών στα δυο τμήματα της έκτης τάξης του δημοτικού σχολείου, όπως είναι η επανάληψη, η επιλογή, τα γεγονότα, οι μεταβλητές, οι διαδικασίες. Τα αποτελέσματα της έρευνας, όπως αυτά προκύπτουν από την Ιεραρχική Αξιολόγηση Προγραμματισμού καταδεικνύουν ότι τα ΠΠΑ συμβάλλουν στην εκπαιδευτική διαδικασία, στην κατανόηση προγραμματιστικών εννοιών και στη κατασκευή ολοκληρωμένων έργων. Επίσης έδειξαν ότι δεν υπάρχει διαφορά στα μαθησιακά αποτελέσματα κατά τη χρήση διαφορετικών ΠΠΑ ούτε κάποια διαφοροποίηση στην επίδοση των δύο φύλων. Η άποψη των μαθητών, χωρίς να υπάρχει σημαντική στατιστική διαφορά, αναδεικνύει το Code.org ως το πιο εύκολο περιβάλλον που τους βοηθά να εντοπίσουν και να διορθώσουν τα λάθη τους, το StarlogoTNG ως το πιο χρήσιμο περιβάλλον που συμβάλλει στη δημιουργία τρισδιάστατων έργων με έναν πιο διασκεδαστικό τρόπο και το Scratch ως το πιο ενδιαφέρον περιβάλλον που βοηθά τους μαθητές στην εξοικείωση και εμβάθυνση των προγραμματιστικών εννοιών. Τα ΠΠΑ μετατρέπουν τον προγραμματισμό και τη διδασκαλία του σε μια ελκυστική και ευχάριστη διαδικασία δημιουργώντας αισθήματα αρέσκειας και εξαλείφοντας ταυτόχρονα τα αρνητικά αισθήματα του φόβου και του άγχους.

### Λέξεις κλειδιά

Προγραμματιστικά Περιβάλλοντα για Αρχαρίους, Code.org, StarlogoTNG, Scratch, Επανάληψη, Επιλογή, Γεγονότα, Μεταβλητές, Διαδικασίες, Ιεραρχική Αξιολόγηση Προγραμματισμού.

## ABSTRACT

The study aims to present the results of a comparative study of Novice Programming Environments (NPE) based on pedagogical and technological characteristics, indicating similarities and differences and how these environments can be used in an empirical study of student's progress. Three novice programming environments of different technological and pedagogical features, Code.org StarlogoTNG and Scratch were utilized and used to teach various programming concepts in two groups of the sixth grade at an elementary school, such as loops, conditions, events, variables and procedures. The results of the study, that arise from the Hierarchical Assessment of Programming, have shown that NPEs contribute to the educational process, to the understanding of programming concepts, and to the constructing of integrated projects. They also showed that there is no difference in learning outcomes and in performance between the two sexes when used different NPEs. The students' opinion, without any significant statistical difference, highlighted Code.org as the easiest environment that helped students to identify and correct their mistakes, StarlogoTNG as the most useful environment that helped the creation of three-dimensional projects in a more entertaining way and Scratch as the most exciting environment that helped on familiarization and deeper understanding of programming concepts. The NPEs have transformed the programming into an attractive and enjoyable process, creating feelings of sympathy and at the same time they eliminated the negative feelings of fear and anxiety.

### Keywords

Novice Programming Environments, Code.org, StarlogoTNG, Scratch, Programming concepts, Loops, Conditions, Events, Variables , Procedures, Hierarchical Assessment of Programming.

## Πίνακας περιεχομένων

ΕΙΣΑΓΩΓΗ.....	9
1 Η ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	12
1.1 ΠΡΟΤΥΠΑ ΕΝΤΑΞΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΩΝ ΤΠΕ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ.....	12
1.2 Η ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΟΙ ΤΠΕ ΣΤΗΝ ΠΡΩΤΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ .....	13
1.3 ΤΑ ΠΡΟΓΡΑΜΜΑΤΑ ΣΠΟΥΔΩΝ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΣΤΗΝ ΠΡΩΤΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ.....	17
1.4 Η ΑΞΙΑ ΚΑΙ ΤΑ ΟΦΕΛΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ .....	21
1.5 ΔΥΣΚΟΛΙΕΣ ΚΑΙ ΠΑΡΑΝΟΗΣΕΙΣ ΑΡΧΑΡΙΩΝ ΜΑΘΗΤΩΝ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ .....	22
1.5.1 ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ.....	25
1.5.2 ΔΟΜΗ ΕΠΙΛΟΓΗΣ .....	26
1.5.3 ΜΕΤΑΒΛΗΤΕΣ .....	27
2 ΕΠΙΛΟΓΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ.....	29
2.1 ΤΑΞΙΝΟΜΙΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ .....	33
2.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ.....	37
2.3 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ .....	56
2.4 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ .....	59
2.4.1 CODE.ORG.....	59
2.4.2 SCRATCH .....	62
2.4.3 STARLOGO TNG .....	65
2.5 ΕΠΙΣΚΟΠΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΕΝΝΟΙΩΝ .....	67
3 ΜΕΘΟΔΟΛΟΓΙΑ.....	70
3.1 ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ.....	70
3.2 ΔΕΙΓΜΑ.....	70
3.3 ΕΡΕΥΝΗΤΙΚΗ ΜΕΘΟΔΟΣ .....	71
3.4 ΕΡΕΥΝΗΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ .....	71
3.4.1 ΟΙΚΟΔΟΜΗΣΗ ΕΝΝΟΙΩΝ .....	74
3.4.2 ΑΞΙΟΠΟΙΗΣΗ ΤΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ .....	75
3.4.3 ΙΕΡΑΡΧΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	76
3.4.4 ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ.....	78
3.4.5 ΑΠΟΝΟΜΗ ΠΙΣΤΟΠΟΙΗΤΙΚΟΥ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ.....	79
4 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	81
4.1 ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΕΝΝΟΙΩΝ ΣΤΟ CODE.ORG .....	81
4.1.1 ΕΠΑΝΑΛΗΨΗ .....	82

4.1.2 ΕΠΙΛΟΓΗ.....	85
4.1.3 ΓΕΓΟΝΟΤΑ.....	87
4.1.4 ΜΕΤΑΒΛΗΤΕΣ.....	90
4.1.5 ΔΙΑΔΙΚΑΣΙΕΣ.....	93
4.2. ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΕΝΝΟΙΩΝ ΣΤΟ STARLOGO TNG ΚΑΙ SCRATCH.....	95
4.2.1 ΕΠΑΝΑΛΗΨΗ.....	96
4.2.2 ΕΠΙΛΟΓΗ.....	97
4.2.3 ΓΕΓΟΝΟΤΑ.....	99
4.2.4 ΜΕΤΑΒΛΗΤΕΣ.....	100
4.2.5 ΔΙΑΔΙΚΑΣΙΕΣ.....	102
4.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ.....	104
4.3.1 ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΗΝ ΧΡΗΣΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ.....	104
4.3.2 ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ.....	106
4.3.3 ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ ΑΡΧΑΡΙΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΣΑΝ.....	110
5 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	117
5.1 ΠΕΡΙΟΡΙΣΜΟΙ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ.....	120
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	122
ΠΑΡΑΡΤΗΜΑΤΑ.....	131
Α ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	131
Β ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ ΜΕ ΙΕΡΑΡΧΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	160
Γ ΕΡΩΤΗΜΑΤΟΛΟΓΙΑ.....	164

## Ευρετήριο Πινάκων

Πίνακας 1.1 Τα Προγράμματα σπουδών για τον προγραμματισμό.....	19
Πίνακας 2.1 Τεχνολογικά χαρακτηριστικά Προγραμματιστικών Περιβαλλόντων.....	42
Πίνακας 2.2 Παιδαγωγικά Χαρακτηριστικά Προγραμματιστικών Περιβαλλόντων.....	54
Πίνακας 2.3 Ανάπτυξη Προγραμματιστικών Εννοιών από τα αντίστοιχα ΠΠΑ.....	67
Πίνακας 3.1 Ακολουθία διδακτικών ενοτήτων .....	75
Πίνακας 4.1. Παρανοήσεις μαθητών στην επανάληψη.....	84
Πίνακας 4.2. Παρανοήσεις μαθητών στην επιλογή.....	87
Πίνακας 4.3. Παρανοήσεις μαθητών στα γεγονότα.....	89
Πίνακας 4.4. Παρανοήσεις μαθητών στις μεταβλητές.....	92
Πίνακας 4.5 Παρανοήσεις μαθητών στις διαδικασίες.....	94

## Ευρετήριο Σχημάτων

Σχήμα 1.1 Πρότυπα ένταξης της Πληροφορικής και των ΤΠΕ στην εκπαίδευση .....	13
Σχήμα 2.1 Τεχνολογικά Χαρακτηριστικά ΠΠΑ.....	57
Σχήμα 2.2 Πλήθος Περιβαλλόντων ανά Ηλικία.....	58
Σχήμα 2.3 Δραστηριότητα από το περιβάλλον του Code.org.....	60
Σχήμα 2.4 Κατηγοριοποιημένες ενότητες δραστηριοτήτων ανά ηλικία στο Code.org.....	61
Σχήμα 2.5 Επισκόπηση δραστηριοτήτων μιας τάξης στο Code.org.....	62
Σχήμα 2.6 Το περιβάλλον Scratch και οι περιοχές του.....	64
Σχήμα 2.7 Το περιβάλλον προγραμματισμού StarLogo TNG.....	66
Σχήμα 3.1 Στάδια της έρευνας.....	73
Σχήμα 3.2 Πιστοποιητικό Παρακολούθησης.....	80
Σχήμα 4.1 Κατανομή μαθητών στην επανάληψη στο Code.org.....	83
Σχήμα 4.2 Κατανομή μαθητών στην επιλογή στο Code.org.....	85
Σχήμα 4.3 Κατανομή μαθητών στα γεγονότα στο Code.org.....	88
Σχήμα 4.4 Κατανομή μαθητών στις μεταβλητές στο Code.org.....	91
Σχήμα 4.5 Κατανομή μαθητών στις διαδικασίες στο Code.org.....	93
Σχήμα 4.6 Ολοκληρωμένα έργα στο Scratch και στο Starlogo TNG .....	96
Σχήμα 4.7 Κατανομή μαθητών στην επανάληψη για τα δυο περιβάλλοντα.....	97
Σχήμα 4.8 Κατανομή μαθητών στην επιλογή για τα δυο περιβάλλοντα.....	98
Σχήμα 4.9 Έλεγχος νίκης και από τα δυο περιβάλλοντα .....	99
Σχήμα 4.10 Κατανομή μαθητών στα γεγονότα για τα δυο περιβάλλοντα.....	100
Σχήμα 4.11 Κατανομή μαθητών στις μεταβλητές για τα δυο περιβάλλοντα .....	101
Σχήμα 4.12 Μείωση τιμής μιας μεταβλητής από τα δυο περιβάλλοντα .....	101
Σχήμα 4.13 Κατανομή μαθητών στις διαδικασίες για τα δυο περιβάλλοντα.....	102
Σχήμα 4.14 Δημιουργία και κλήση διαδικασίας για την κίνηση ενός χαρακτήρα .....	103
Σχήμα 4.15 Άνεση ως προς την χρήση ηλεκτρονικού υπολογιστή συνολικά και για το κάθε τμήμα.....	104
Σχήμα 4.16 Ώρες την εβδομάδα που χρησιμοποιούν τον υπολογιστή στο σπίτι.....	105
Σχήμα 4.17 Χρησιμοποιείς τον υπολογιστή για .....	106
Σχήμα 4.18 Μου αρέσει ο προγραμματισμός υπολογιστών.....	107
Σχήμα 4.19 Με ευχαριστεί να παρακολουθώ μαθήματα Προγραμματισμού.....	108
Σχήμα 4.20 Νιώθω άγχος και φόβο κατά τη διάρκεια του μαθήματος του Προγραμματισμού.....	109
Σχήμα 4.21 Είναι δύσκολος ο Προγραμματισμός συνολικά και για το κάθε τμήμα.....	109
Σχήμα 4.22 Πόσο εύκολο είναι το κάθε ΠΠΑ.....	111
Σχήμα 4.23 Πόσο βαρετό είναι το κάθε ΠΠΑ.....	112
Σχήμα 4.24 Πόσο ενδιαφέρον είναι το κάθε ΠΠΑ.....	113
Σχήμα 4.25 Πόσο χρήσιμο είναι το κάθε ΠΠΑ.....	114
Σχήμα 4.26 Ποιό ΠΠΑ χρησιμοποίησαν σπίτι τους.....	116



## ΕΙΣΑΓΩΓΗ

Η κοινωνία της πληροφορίας και της γνώσης, καθώς και η παγκοσμιοποίηση των εκφράσεων της κοινωνικοοικονομικής και της πολιτισμικής ζωής παράγουν νέες ανάγκες ως προς τις γνώσεις και τις δεξιότητες τις οποίες είναι ανάγκη να διαθέτουν οι άνθρωποι. Σε αυτό συνεισφέρουν οι υπολογιστές όπου στα πλαίσια των δυνατοτήτων τους θεωρούνται ως το ισχυρότερο γνωστικό εργαλείο που αποδεσμεύουν από μηχανιστικές διεργασίες και απομνημόνευση, μειώνουν τη γνωστική υπερφόρτωση, κάνουν υπολογισμούς, αναπαριστούν μεγέθη, φαινόμενα και καταστάσεις (Μικρόπουλος, 2006). Αντίθετα με την παραδοσιακή απομνημόνευση, σήμερα απαιτείται η αναζήτηση πληροφοριών, η επεξεργασία τους, η ανάλυση, η σύνθεση και η αξιολόγηση τους (Μικρόπουλος, 2006). Ο υπολογιστής λοιπόν ως γνωστικό εργαλείο συνεισφέρει και ανοίγει νέους δρόμους στη μαθησιακή διαδικασία (Μικρόπουλος, 1997).

Το παραδοσιακό σχολείο που βασιζόταν στον εκπαιδευτικό που κατείχε την πληροφορία και τη γνώση και τη «μετέδιδε» στο μαθητή, μετατρέπεται σε ένα νέου τύπου σχολείου, όπου ο ρόλος του εκπαιδευτικού είναι καθοδηγητικός και συμβουλευτικός και ο μαθητής αποκτά την πληροφορία και τη γνώση και με τη συνδρομή του υπολογιστή και των νέων τεχνολογιών. Με αυτόν τον τρόπο ο μαθητής λειτουργεί ως ερευνητής, καθοδηγούμενος από τον εκπαιδευτικό καλλιεργώντας έτσι τις δεξιότητες και τα ιδιαίτερα χαρακτηριστικά του (Sengupta, Kinnebrew, Basu, Biswas & Clark, 2012).

Άλλωστε, η ιδιότητα της αλληλεπιδραστικότητας, κύριο χαρακτηριστικό των Τεχνολογιών Πληροφορίας και Επικοινωνιών (ΤΠΕ), προσφέρει στο μαθητή τη δυνατότητα να συμμετέχει μαζί με τον δάσκαλό του στο σχεδιασμό των μαθησιακών δραστηριοτήτων και να εκφράζει ελεύθερα τις αντιλήψεις και τα συναισθήματά του (Sengupta, Kinnebrew, Basu, Biswas & Clark, 2012).

Η ενεργή συμμετοχή και η αλληλεπίδραση παρέχουν στο μαθητή τη δυνατότητα διαχείρισης του υλικού και του τρόπου διδασκαλίας αποφεύγοντας έτσι το πρόβλημα του διαφορετικού γνωστικού υπόβαθρου των μαθητών που παρατηρείται στις αίθουσες διδασκαλίας (Μικρόπουλος, 1997).

Η ταχύτατη ανάπτυξη της επιστήμης της Πληροφορικής και των ΤΠΕ επέφερε γρήγορα αλλαγές σε όλους τους τομείς της σύγχρονης ζωής. Ανεπηρέαστος δεν θα μπορούσε να μείνει ούτε ο χώρος της εκπαίδευσης όπου η νέα πραγματικότητα επέβαλε την ανάγκη προσαρμογής των υπαρχόντων εκπαιδευτικών συστημάτων, ώστε να μπορέσει να ανταπεξέλθει στις σύγχρονες απαιτήσεις

μόρφωσης και εκπαίδευσης και στις ραγδαίες εξελίξεις της αγοράς εργασίας (Κόμης & Μικρόπουλος, 2001).

Σε αυτό το εκπαιδευτικό τοπίο, έρχεται ο προγραμματισμός του ηλεκτρονικού υπολογιστή να αποτελέσει αναπόσπαστο κομμάτι της επιστήμης της Πληροφορικής και των ΤΠΕ, αλλά και ένα ελκυστικό αντικείμενο μελέτης και ενασχόλησης, αφού επιτρέπει την προσαρμογή του υλικού και του λογισμικού στις ανάγκες του εκάστοτε μαθητή (Μικρόπουλος & Μπέλλου, 2010). Ο προγραμματισμός συμπεριλαμβάνεται στο πρόγραμμα σπουδών της Πληροφορικής στη δευτεροβάθμια και ακολούθως στην πρωτοβάθμια εκπαίδευση, καθώς επίσης και στα προγράμματα σπουδών σχολών της τριτοβάθμιας εκπαίδευσης, ακόμη και αν δεν σχετίζονται άμεσα με τα αντικείμενα της Πληροφορικής ή της Τεχνολογίας, υποδηλώνοντας έτσι τη μεγάλη σημασία που έχει η εκμάθηση των αρχών του και η επίλυση προβλημάτων σε αυτό το πλαίσιο. Άλλωστε, αυτό υποστηρίζει σύσσωμη η ερευνητική και παιδαγωγική κοινότητα, με πρωτοπόρο τον Papert, σύμφωνα με τον οποίο ο προγραμματισμός μπορεί να αποτελέσει εκπαιδευτικό εργαλείο για την καλλιέργεια και ανάπτυξη νοητικών δεξιοτήτων σε όλους τους μαθητές σε όλα σχεδόν τα γνωστικά αντικείμενα (Papert, 1980).

Η διδασκαλία και η εκμάθηση του προγραμματισμού, όμως, συχνά δυσκολεύει τους μαθητές. Οι παραδοσιακές διδακτικές προσεγγίσεις δεν βοηθούν στην αντιμετώπιση των δυσκολιών με αποτέλεσμα την προβολή αρνητικής στάσης των μαθητών απέναντι στη διδασκαλία του προγραμματισμού. Πολλοί από αυτούς θεωρούν τον προγραμματισμό ως μια πολύπλοκη διαδικασία η οποία απαιτεί εξειδικευμένη τεχνική κατάρτιση. Οι ίδιοι περιγράφουν τα μαθήματα προγραμματισμού ως υπερβολικά τεχνικά, αποκομμένα από τον πραγματικό κόσμο και στερούμενα δημιουργικότητας (Gomes & Mendes, 2007).

Σημαντική λύση σε αυτή την προβληματική κατάσταση είναι η μετατροπή του εισαγωγικού προγραμματισμού σε μια εύκολη και διασκεδαστική διαδικασία για τους μαθητές. Ένας τρόπος για να επιτευχθεί αυτό είναι η εξάλειψη ή καλύτερα η μείωση των εμποδίων για την εισαγωγή στον προγραμματισμό, μέσω της χρήσης ειδικών Προγραμματιστικών Περιβαλλόντων για Αρχαρίους (ΠΠΑ). Η διδασκαλία του προγραμματισμού θα πρέπει να γίνεται με τέτοιο τρόπο, ώστε να ελαχιστοποιείται το γνωστικό φορτίο ενώ ταυτόχρονα να μεγιστοποιείται η παιδαγωγική αξία. Η εμπλοκή των μαθητών είναι πιο επιτυχής, όταν το πλαίσιο διδασκαλίας τροφοδοτείται από θέματα που έχουν άμεσο ενδιαφέρον για τους μαθητές (Gray, Abelson, Wolber, & Friend, 2012). Η επιλογή

ωστόσο του κατάλληλου προγραμματιστικού περιβάλλοντος για αρχαρίους αποτελεί μια σοβαρή πρόκληση (Μικρόπουλος, 1997; Webb, 2012).

Η παρούσα μελέτη πραγματεύεται αφενός μεν τη συγκριτική παρουσίαση προγραμματιστικών περιβαλλόντων για αρχαρίους με βάση τα παιδαγωγικά και τεχνολογικά χαρακτηριστικά εδραζόμενη στις ομοιότητες και τις διαφορές τους και αφετέρου στην αξιοποίηση επιλεγμένων περιβαλλόντων σε μια εμπειρική μελέτη της πορείας των μαθητών σε διάφορες προγραμματιστικές έννοιες.

Το πρώτο κεφάλαιο αφορά σε μια ανασκόπηση θεμάτων της Πληροφορικής, και εκτενέστερα το αντικείμενο του προγραμματισμού στην εκπαίδευση. Πραγματοποιείται μια παρουσίαση των προγραμμάτων σπουδών που έχουν κατά καιρούς προταθεί σε αυτήν τη θεματική περιοχή. Ιδιαίτερη μνεία γίνεται στην αξία και τα οφέλη που αποκομίζονται από την ενασχόληση με το αντικείμενο του προγραμματισμού, αλλά και τις δυσκολίες που συναντώνται και καταγράφονται κατά την εμπλοκή αρχάριων προγραμματιστών με αυτόν.

Το δεύτερο κεφάλαιο αναφέρεται σε διάφορα προγραμματιστικά περιβάλλοντα που έχουν κατά καιρούς χρησιμοποιηθεί στην εκπαίδευση αρχάριων προγραμματιστών. Σε αυτό περιγράφονται αναλυτικά διάφορες ταξινομίες περιβαλλόντων που έχουν αναπτυχθεί και ξεδιπλώνονται τα τεχνολογικά άλλα και τα παιδαγωγικά χαρακτηριστικά για τη μελέτη πενήντα πέντε (55) προγραμματιστικών περιβαλλόντων. Τέλος, διερευνώνται τα περιβάλλοντα που έχουν επιλεγεί για χρήση στη σχολική πρακτική.

Στο τρίτο κεφάλαιο αναπτύσσεται το μεθοδολογικό κομμάτι της έρευνας: παρουσιάζονται τα ερευνητικά ερωτήματα, το δείγμα στο οποίο πραγματοποιήθηκε η εφαρμογή των περιβαλλόντων και εν συνεχεία μια ολοκληρωμένη περιγραφή της ακολουθίας των σταδίων προκειμένου για την πραγματοποίηση της έρευνας και της ερευνητικής διαδικασίας.

Στο τέταρτο κεφάλαιο γίνεται λόγος για τα αποτελέσματα της έρευνας σχετικά με την επίδοση, αλλά και το πώς αξιολογούν οι ίδιοι οι μαθητές τα περιβάλλοντα πάνω στα οποία εργάστηκαν στα πλαίσια της συγκριτικής μελέτης των περιβαλλόντων.

Τέλος, το πέμπτο κεφάλαιο καταλαμβάνει η συζήτηση των αποτελεσμάτων. Αρχικά, καταγράφονται τα μαθησιακά αποτελέσματα των μαθητών- επίδοση και στη συνέχεια η αξιολόγηση- προτίμηση των μαθητών. Ακολουθούν τα συμπεράσματα της μελέτης, οι περιορισμοί και οι προτάσεις για μελλοντική έρευνα.

# 1 Η ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

## 1.1 ΠΡΟΤΥΠΑ ΕΝΤΑΞΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΩΝ ΤΠΕ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ

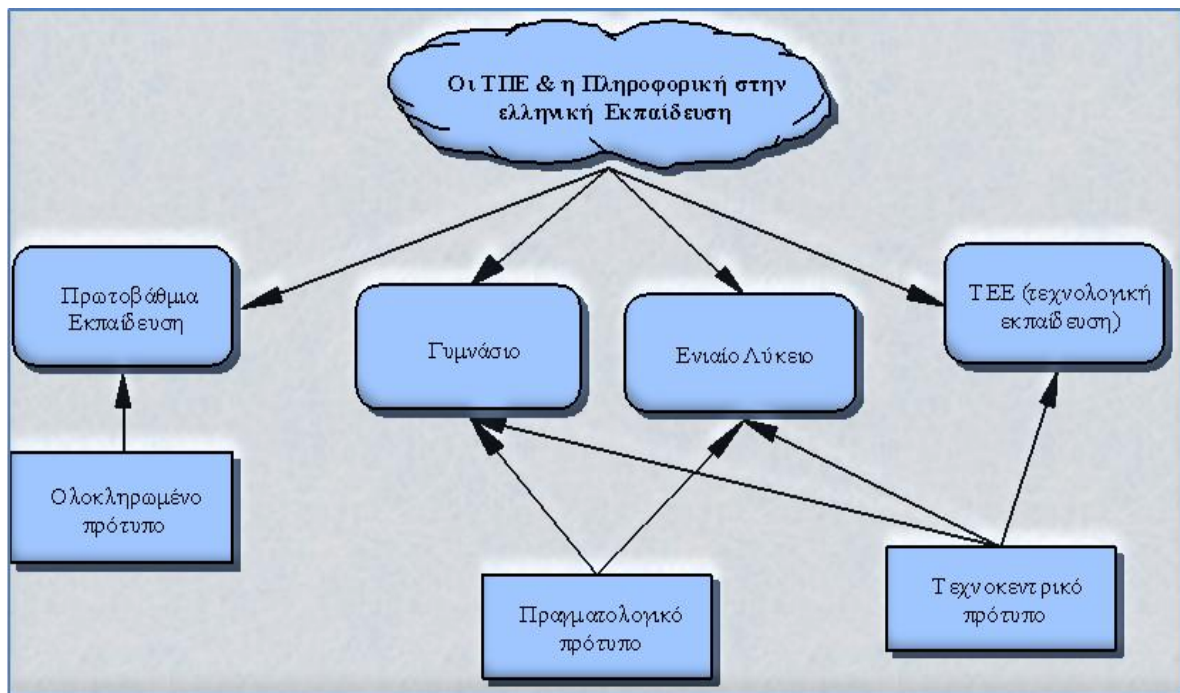
Σύμφωνα με τη βιβλιογραφία (Μακράκης, 2000; Κόμης & Μικρόπουλος, 2001; Κόμης, 2001; Κόμης, 2004; Κόμης, 2005; Κόμης, 2011) φαίνεται να επικρατούν τρία πρότυπα εισαγωγής και ένταξης των ΤΠΕ στην εκπαιδευτική διαδικασία: το τεχνοκεντρικό, το ολοκληρωμένο και το πραγματολογικό πρότυπο.

Το Τεχνοκεντρικό πρότυπο ορίζεται ως το μοντέλο εισαγωγής της Πληροφορικής και των ΤΠΕ στην εκπαιδευτική διαδικασία που χαρακτηρίζεται από τεχνολογικό ντετερμινισμό και περιγράφει προσεγγίσεις όπου η Πληροφορική αντιμετωπίζεται ως ανεξάρτητο γνωστικό αντικείμενο στο πρόγραμμα σπουδών με έμφαση στις τεχνικές γνώσεις και δεξιότητες.

Το ολοκληρωμένο πρότυπο εκφράζει την προσέγγιση κατά την οποία η Πληροφορική και οι ΤΠΕ αποτελούν εργαλείο μάθησης σε όλο το εύρος του αναλυτικού προγράμματος και συνιστούν μια διεπιστημονική μέθοδο εργασίας. Στα πλαίσια αυτού του μοντέλου, η χρήση της Πληροφορικής και των ΤΠΕ ενσωματώνεται στα επιμέρους γνωστικά αντικείμενα του αναλυτικού προγράμματος.

Ως πραγματολογικό πρότυπο ορίζεται το μοντέλο εισαγωγής της Πληροφορικής και των ΤΠΕ στην εκπαιδευτική διαδικασία το οποίο περιγράφει καταστάσεις όπου συμβαίνει διδασκαλία μαθημάτων Πληροφορικής με παράλληλη ένταξη της χρήσης των τεχνολογιών στη διδακτική και μαθησιακή διαδικασία.

Στο σχήμα 1.1 φαίνεται πως είναι διαμορφωμένη η κατάσταση από τα τρέχοντα προγράμματα σπουδών στο Δημοτικό, το Γυμνάσιο, το Λύκειο και στη Τεχνολογική Εκπαίδευση. Στη συνέχεια μελετώνται τα προγράμματα σπουδών που έχουν κατά καιρούς προταθεί στη Πρωτοβάθμια Εκπαίδευση για την Πληροφορική και τις ΤΠΕ.



Σχήμα 1.1 Πρότυπα ένταξης της Πληροφορικής και των ΤΠΕ στην εκπαίδευση (Κόμης, 2005)

## 1.2 Η ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΟΙ ΤΠΕ ΣΤΗΝ ΠΡΩΤΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ

Αντικείμενο μελέτης της παρούσας ενότητας αποτελούν τα διάφορα Προγράμματα σπουδών που εφαρμόστηκαν στην Πληροφορική και των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας (ΤΠΕ) στο χώρο της Πρωτοβάθμιας εκπαίδευσης. Η εκκίνηση γίνεται με το Ενιαίο Πλαίσιο Προγράμματος Σπουδών Πληροφορικής (ΕΠΠΣ) που αφορά στην εισαγωγή και ένταξη των ΤΠΕ σε όλες τις βαθμίδες της εκπαίδευσης, προφανώς και στο Δημοτικό, χωρίς όμως να προσδιορίζεται επακριβώς από ποια τάξη όσον αφορά τη χρήση τους στα διάφορα γνωστικά αντικείμενα (Κόμης, 2005). Παράλληλα στις δυο τελευταίες τάξεις καθιερώνεται κατ' επιλογήν εβδομαδιαία ώρα Πληροφορικής που αφορά το ένα τρίτο του ημερήσιου σχολικού χρόνου (Παιδαγωγικό Ινστιτούτο, 1997).

Το ΕΠΠΣ προτείνει να ενταχθεί η χρήση της πληροφορικής καταρχάς στα πλαίσια της διδασκαλίας των επιστημών και της τεχνολογίας στην πρωτοβάθμια εκπαίδευση και σταδιακά σε όλο το εύρος του αναλυτικού προγράμματος. Δεν πρόκειται για ένα μάθημα προαπαιτούμενο για τη χρήση του υπολογιστή αλλά για μια ευκαιρία επαρκούς και ορθολογικής συνάντησης του παιδιού με τις ΤΠΕ, συνάντηση που πρέπει να καλύπτει υπαρκτές ανάγκες της εκπαιδευτικής διαδικασίας. Η χρήση των υπολογιστών από τους μαθητές μπορεί επίσης να ειπωθεί και σε μια σειρά από

δραστηριότητες που δεν εμπίπτουν στα στενά πλαίσια του αναλυτικού προγράμματος (Κόμης & Μικρόπουλος, 2001). Τέτοιου τύπου δραστηριότητες εφαρμόζονται στα πλαίσια του ολοήμερου σχολείου (ευέλικτη ζώνη - πληροφορική), όχι όμως πάντα με τον ενδεδειγμένο τρόπο (Κόμης, 2004).

Ο γενικός σκοπός εισαγωγής των ΤΠΕ στο δημοτικό με το ΕΠΠΣ είναι: *«οι μαθητές να είναι σε θέση να χρησιμοποιούν με (ή χωρίς) τη βοήθεια του εκπαιδευτικού τον υπολογιστή ως γνωστικό-διερευνητικό εργαλείο, να αναζητούν πληροφορίες, να επικοινωνούν και να προσεγγίζουν βασικές αρχές που διέπουν τη χρήση της υπολογιστικής τεχνολογίας»* (Παιδαγωγικό Ινστιτούτο, 1997, σελ. 10).

Αργότερα το 2003 εγκρίνεται το Διαθεματικό Ενιαίο Πλαίσιο Προγραμμάτων Σπουδών (ΔΕΠΠΣ). Η εισαγωγή της πληροφορικής στο δημοτικό είχε σαν σκοπό να έρθουν οι μαθητές σε μια πρώτη επαφή με τον υπολογιστή και τις βασικές του λειτουργίες. Ο υπολογιστής στο δημοτικό, επιδιώκεται να χρησιμοποιηθεί σαν γνωστικό διερευνητικό εργαλείο με σκοπό την μάθηση, σαν εποπτικό μέσο διδασκαλίας σε βασικά γνωστικά αντικείμενα προάγοντας τη διαθεματικότητα, σαν εργαλείο επικοινωνίας και αναζήτησης πληροφοριών και σαν γνωστικό αντικείμενο ο πληροφορικός αλφαριθμητισμός, αποκτώντας δεξιότητες στην χρήση ενός υπολογιστικού συστήματος (Παιδαγωγικό Ινστιτούτο, 2003).

Ο ειδικός σκοπός της εισαγωγής της Πληροφορικής στο Δημοτικό Σχολείο είναι «να εξοικειωθούν οι μαθητές και οι μαθήτριες με τις βασικές λειτουργίες του υπολογιστή και να έλθουν σε μια πρώτη επαφή με διάφορες χρήσεις του ως εποπτικού μέσου διδασκαλίας, ως γνωστικού - διερευνητικού εργαλείου και ως εργαλείου επικοινωνίας και αναζήτησης πληροφοριών στο πλαίσιο των καθημερινών σχολικών τους δραστηριοτήτων με τη χρήση κατάλληλου λογισμικού και ιδιαίτερα ανοικτού λογισμικού διερευνητικής μάθησης. Σε καμία περίπτωση δεν νοείται η διδασκαλία της Πληροφορικής ως διδασκαλία γνωστικού αντικειμένου, λαμβάνοντας υπόψη ότι δεν διατίθεται χρόνος στο αντίστοιχο Ωρολόγιο Πρόγραμμα. Σκοπός είναι ο μαθητής να μαθαίνει με τη χρήση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας (ΤΠΕ) παρά για τη χρήση τους» (Παιδαγωγικό Ινστιτούτο, 2003).

Διαφαίνεται και στις δύο περιπτώσεις, ότι η εισαγωγή των ΤΠΕ στο Δημοτικό Σχολείο σκοπεύει σε μια σφαιρική προσέγγιση απ' όλους τους μαθητές, των διαφόρων χρήσεων, στα πλαίσια των καθημερινών σχολικών τους δραστηριοτήτων. Η έμφαση δίνεται στο να εκτελέσουν οι μαθητές δραστηριότητες με τον υπολογιστή και να κατανοήσουν βασικές αρχές που διέπουν τη χρήση της υπολογιστικής τεχνολογίας σε σημαντικές ανθρώπινες ασχολίες όπως είναι η πληροφορία και η

επεξεργασία της, η επικοινωνία, η ψυχαγωγία, οι νέες δυνατότητες προσέγγισης της γνώσης (Κόμης, 2004).

Το πλαίσιο αυτό επιδιώκει αφενός να οριοθετήσει έναν ενιαίο τρόπο θεώρησης της ένταξης των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών στο ελληνικό εκπαιδευτικό σύστημα και αφετέρου να δώσει απαντήσεις στα κύρια θέματα που αφορούν την ένταξη των ΤΠΕ σε όλο το φάσμα του ελληνικού σχολικού συστήματος (γενικό πλαίσιο, προγράμματα σπουδών, μεθοδολογία διδασκαλίας, προδιαγραφές σχολικών εργαστηρίων, κ.λπ.)(Κόμης, 2005).

Το 2010 εγκρίνεται το Ενιαίο Αναμορφωμένο Εκπαιδευτικό Πρόγραμμα (ΕΑΕΠ) όπου για πρώτη φορά διατίθεται χρόνος στο ωρολόγιο πρόγραμμα όλων των τάξεων του Δημοτικού για την διδασκαλία της Πληροφορικής ως γνωστικό αντικείμενο στα πλαίσια του μαθήματος «Τεχνολογίες της Πληροφορίας και Επικοινωνιών». Πιο συγκεκριμένα, για την διδασκαλία του μαθήματος διατίθενται μία ώρα εβδομαδιαίως για την Α' και την Β' τάξη και δύο ώρες για τις υπόλοιπες τάξεις. Ακολουθείται πλέον το πραγματολογικό πρότυπο από το ολοκληρωμένο που εφαρμόζονταν μέχρι τότε, σύμφωνα με το οποίο οι μαθητές και οι μαθήτριες διδάσκονται βασικές γνώσεις Πληροφορικής ενώ ταυτόχρονα οι ΤΠΕ αξιοποιούνται ως μέσο στήριξης της μαθησιακής διαδικασίας στα επιμέρους γνωστικά αντικείμενα. Οι γενικοί στόχοι στηρίζονται σε τρεις άξονες: α) Γνώση και μεθοδολογία, Συνεργασία, β) Επικοινωνία και κοινωνική αλληλεπίδραση και γ) Επιστήμη και ΤΠΕ στην καθημερινή ζωή (Παιδαγωγικό Ινστιτούτο, 2010).

Ο ειδικός σκοπός της εισαγωγής της Πληροφορικής στο δημοτικό σχολείο με το (ΕΑΕΠ) είναι «να εξοικειωθούν οι μαθητές και οι μαθήτριες με τις βασικές λειτουργίες του υπολογιστή και να έλθουν σε μια πρώτη επαφή με διάφορες χρήσεις του ως εποπτικού μέσου διδασκαλίας, ως γνωστικού – διερευνητικού εργαλείου και ως εργαλείου επικοινωνίας και αναζήτησης πληροφοριών. Να αποκτήσουν ικανότητες και να αναπτύξουν δεξιότητες χρήσης και αξιοποίησης των ΤΠΕ με ασφάλεια, αναστοχαστική συμπεριφορά έναντι των διαθέσιμων πληροφοριών, αυτοπεποίθηση και δημιουργικότητα, ώστε να προετοιμαστούν για την υπόλοιπη μαθητική ζωή τους και την πλήρη ένταξή τους στην κοινωνία της Γνώσης και της Πληροφορίας» (Παιδαγωγικό Ινστιτούτο, 2010, σελ. 4).

Το 2011 εγκρίνεται το Πιλοτικό Πρόγραμμα Σπουδών «ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα)» που είναι η εφαρμογή βασικών αρχών του Νέου Σχολείου στα υφιστάμενα Δημοτικά Σχολεία και ιδίως στα Δημοτικά Σχολεία με ενιαίο αναμορφωμένο εκπαιδευτικό πρόγραμμα. Κεντρικός στόχος της διδασκαλίας του μαθήματος είναι όλοι «οι μαθητές να αναπτύξουν τις γνώσεις και τις ικανότητες χρήσης των ΤΠΕ μέσα από δραστηριότητες που αφορούν στην αναζήτηση και διαχείριση

πληροφοριών, στην επίλυση προβλημάτων και στη λήψη αποφάσεων, στη δημιουργική έκφραση και στην επικοινωνία. Εξίσου σημαντικό είναι, μέσα από κατάλληλες χρήσεις και δραστηριότητες βασισμένες σε ΤΠΕ, οι μαθητές να κατανοήσουν τα όρια και την επίδραση που έχουν οι σύγχρονες τεχνολογίες στα άτομα και στις ομάδες, στις κοινότητες και στην κοινωνία ευρύτερα.» (ΥΠΑΔΒΜΘ, 2011, σελ. 8). Το ωρολόγιο πρόγραμμα είναι ίδιο με το ΕΑΕΠ και οι ΤΠΕ αντιμετωπίζονται ως: μαθησιακό-γνωστικό εργαλείο, μεθοδολογία επίλυσης προβλημάτων, τεχνολογικό εργαλείο, κοινωνικό φαινόμενο.

Τέλος εφαρμόζεται το 2016 ο Ενιαίος Τύπος Ολοήμερου Δημοτικού Σχολείου που διατηρεί τα θετικά στοιχεία του ΕΑΕΠ για 1337 ολόημερα 12/θέσια δημοτικά σχολεία και αποσκοπεί στον εμπλουτισμό και την επέκταση των προσφερόμενων γνωστικών αντικειμένων στους μαθητές όλων των τύπων σχολείων από 4/θέσια και πάνω. Η λειτουργία των μονοθέσιων, διθέσιων και τριθέσιων δημοτικών σχολείων παραμένει ως έχει χωρίς καμία διαφοροποίηση (ΥΠΕΠΘ, 2016). Αυτό που πρέπει να σημειωθεί για την Πληροφορική και τις ΤΠΕ είναι ότι μειώνονται οι ώρες σε μια ώρα την εβδομάδα από δυο σε Γ', Δ', Ε' και ΣΤ' τάξη του δημοτικού σχολείου ενώ για την πρώτη και δεύτερη τάξη παραμένουν ως έχουν.

Σε αυτό το σημείο αξίζει να αναφερθεί ότι η Πληροφορική και οι ΤΠΕ αναπτύσσονται γύρω από τα παρακάτω συμπληρωματικά αντικείμενα μελέτης (Κόμης, 2005):

- διδακτική εννοιών πληροφορικής,
- διδακτική τεχνολογίας υλικού,
- διδακτική λογισμικών γενικής χρήσης και
- διδακτική του προγραμματισμού.

Σύμφωνα με έρευνα που αφορούσε 308 εισηγήσεις πέντε συνεδρίων με θέμα «Διδακτική της Πληροφορικής» που διοργανώθηκαν την τελευταία δεκαετία από την Ελληνική Επιστημονική Ένωση Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση (ΕΤΠΕ) που πραγματοποιήθηκε από τους Μικρόπουλο, Μπέλλου και Θεοδώρου, (2014) προκύπτει ότι το επικρατέστερο και με το μεγαλύτερο ενδιαφέρον αντικείμενο μελέτης είναι ο προγραμματισμός. Πάνω σε αυτό θα γίνει μια επισκόπηση αναλύοντας χρονικά τα προγράμματα σπουδών που εφαρμόστηκαν στην πρωτοβάθμια εκπαίδευση στο πώς διαμορφώθηκε το αντικείμενο του προγραμματισμού.



### 1.3 ΤΑ ΠΡΟΓΡΑΜΜΑΤΑ ΣΠΟΥΔΩΝ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΣΤΗΝ ΠΡΩΤΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ

Στην προαναφερθείσα σχολική πραγματικότητα έρχεται ο προγραμματισμός ηλεκτρονικών υπολογιστών να αποτελέσει μια σημαντική δεξιότητα, καθώς μέσα από αυτόν προάγεται ένας διαφορετικός τρόπος σκέψης και επίλυσης προβλημάτων. Ο προγραμματισμός και η εκμάθηση μιας γλώσσας προγραμματισμού δεν αποσκοπεί στην προετοιμασία και εκπαίδευση μελλοντικών προγραμματιστών, αλλά στην ανάπτυξη μιας κουλτούρας τεχνολογικού αλφαριθμητισμού που θα ωθήσει τους μαθητές να αποκτήσουν μια πιο βαθιά γνώση του πως λειτουργεί ένα υπολογιστικό σύστημα, πως μπορούν να εφαρμόσουν τον προγραμματισμό στην καθημερινότητά τους και να αποτελέσουν ενεργά μέλη της νέας τεχνολογικής εποχής (Garneli, Giannakos, & Chorianoopoulos, 2015). Η Gans, (2010) αναφέρει ότι οι μαθητές μέσα από τον προγραμματισμό αποκτούν ικανότητες πληρέστερης και δημιουργικής έκφρασης των απόψεών τους, ανάπτυξη λογικού τρόπου σκέψης και κατανόηση του τρόπου λειτουργίας των Τεχνολογιών Πληροφορίας και Επικοινωνιών (ΤΠΕ), οι οποίες βρίσκονται διάσπαρτες στη καθημερινή τους ζωή.

Ο προγραμματισμός στο δημοτικό δεν εισήχθη το 1997 από το Ενιαίο Πλαίσιο Προγράμματος Σπουδών Πληροφορικής (ΕΠΠΣ) ενώ πρότεινε την εισαγωγή του για το γυμνάσιο και το λύκειο άρχισε να ενσωματώνεται το 2003 με το Διαθεματικό Ενιαίο Πλαίσιο Προγράμματος Σπουδών Πληροφορικής (ΔΕΠΠΣ) στις τάξεις Ε' και Στ' με το ίδιο ακριβώς τρόπο. Στο ΔΕΠΠΣ οι θεματικές ενότητες που αφορούν τον προγραμματισμό είναι «Ελέγχω και προγραμματίζω» και η «Χρήση μιας απλής γλώσσας προγραμματισμού (Logo like) για τον έλεγχο και το προγραμματισμό του υπολογιστή». Οι μαθητές καλούνται να κατανοήσουν ότι ο υπολογιστής εκτελεί οδηγίες που παίρνει από τον άνθρωπο σε μια κωδικοποιημένη μορφή καθώς και να χρησιμοποιήσουν απλές εντολές για δημιουργία σχημάτων ή απλών προβλημάτων (Παιδαγωγικό Ινστιτούτο, 2003).

Με το ΕΑΕΠ διαχωρίζονται οι τάξεις της Ε' και Στ' ως προς τις θεματικές ενότητες όπου στο ΔΕΠΠΣ ήταν ενιαίες. Πλέον ενώ οι θεματικές ενότητες ακολουθούν την ίδια ονομασία που είναι «Προγραμματίζω και ελέγχω» οι στόχοι για την κάθε τάξη είναι αισθητά διαφορετικοί, παρ' όλα αυτά πιο αναλυτικοί και οι ενδεικτικές δραστηριότητες διαφέρουν. Για τα περιβάλλοντα που προτείνονται είναι τύπου Logo like όπως και στην περίπτωση του ΔΕΠΣ χωρίς όμως να γίνεται κάποια συγκεκριμένη πρόταση για ποια περιβάλλοντα ενδείκνυνται να χρησιμοποιηθούν. Αυτό που

σημειώνεται είναι ότι «οι δραστηριότητες προσαρμόζονται στις δυνατότητες που προσφέρει το προγραμματιστικό περιβάλλον που χρησιμοποιεί ο εκπαιδευτικός»(Παιδαγωγικό Ινστιτούτο , 2010)

Τέλος στο Νέο Σχολείο παρατηρείται μια μεγάλη διαφοροποίηση-επέκταση και στις θεματικές ενότητες τις κάθε τάξης Ε' και Στ' και ως προς τους στόχους άλλα και ως προς τις ενδεικτικές δραστηριότητες που προτείνονται. Οι θεματικές ενότητες ονομάζονται πλέον «Προγραμματίζω τον υπολογιστή» όπου σε αυτές αναφέρονται τα Εκπαιδευτικά περιβάλλοντα οπτικού προγραμματισμού, ο Αλγόριθμος, το Πρόγραμμα, τα Γεγονότα (events), η Ανάπτυξη και εκτέλεση απλών εφαρμογών, η Διόρθωση σφαλμάτων και η Βελτιστοποίηση προγραμμάτων για την Ε' τάξη και αφορούν περισσότερο στην εξοικείωση των μαθητών με το γραφικό περιβάλλον και με την κίνηση ενός αντικειμένου, καθώς επίσης και το σχεδιασμό απλών γεωμετρικών σχημάτων. Οι στόχοι που προστίθενται στην Στ' τάξη είναι η Επαναληπτική δομή, η Δομή επανάληψης και η Διαδικασία, και οδηγούν το μαθητή να μάθει βασικές έννοιες προγραμματισμού αλλά και την χρησιμότητα αυτών. Αξιοσημείωτο είναι ότι πλέον προτείνονται και κάποια προγραμματιστικά περιβάλλοντα όπως είναι η EasyLogo, το Scratch, το BYOB, το Kodu, το Microworlds Pro, το gameMaker, το K-turtle, το Turtle Art, και η openStarlogo όπου ο κάθε εκπαιδευτικός μπορεί να χρησιμοποιήσει. Η προσέγγιση αυτή συμφωνεί με τη θεώρηση του προγραμματισμού ως βασική συνιστώσα για την ανάπτυξη πληροφοριακού γραμματισμού και την εξίσωση της σημαντικότητάς του με αυτή της γραφής, της ανάγνωσης και των μαθηματικών ως θεμελιώδεις λίθους για τη γνωστική ανάπτυξη του ατόμου (Kelleher, 2012).

Σε μια προσπάθεια μελέτης, σύγκρισης άλλα και καλύτερης κατανόησης στο πώς διαμορφώθηκε και εξελίχτηκε ο προγραμματισμός με τα διάφορα προγράμματα σπουδών που έχουν προταθεί κατά καιρούς στην πρωτοβάθμια εκπαίδευση γίνεται μια παράθεση όλων αυτών στον πίνακα 1.1

Πίνακας 1.1 Τα Προγράμματα σπουδών ΔΕΠΣ 2003, ΕΑΕΠ 2010, ΝΕΟ ΣΧΟΛΕΙΟ 2011 για τον προγραμματισμό

Πρόγραμμα Σπουδών	Τάξη	Θεματικές ενότητες	Στόχοι	Ενδεικτικές Δραστηριότητες
ΔΕΠΠΣ 2003	Ε- Στ	-Ελέγχω και προγραμματίζω -Χρήση μιας απλής γλώσσας προγραμματισμού (Logo like) για τον έλεγχο και προγραμματισμό του υπολογιστή	-Να κατανοήσουν ότι ο υπολογιστής εκτελεί οδηγίες που παίρνει από τον άνθρωπο σε μια κωδικοποιημένη μορφή. -Να χρησιμοποιούν απλές εντολές για τη δημιουργία σχημάτων ή τη λύση απλών προβλημάτων.	Καλούνται να δημιουργήσουν απλά γεωμετρικά σχήματα δίνοντας κατάλληλες εντολές μετακίνησης ή στροφής στη χελώνα. Μέσα από επιλεγμένα παραδείγματα, όπου υπάρχουν σκόπιμα «λάθη», τα παιδιά κατανοούν ότι ο υπολογιστής εκτελεί τις οδηγίες που ο άνθρωπος του δίνει.
ΕΑΕΠ 2010	Ε	<b>Προγραμματίζω και ελέγχω</b>	Να σχεδιάζουν και να επεξεργάζονται δικές τους ζωγραφιές σε ένα <b>Logo like προγραμματιστικό περιβάλλον</b> . Να εισάγουν και να επεξεργάζονται έτοιμα σχήματα. Να εισάγουν έτοιμες εικόνες και ζωγραφιές. Να εισάγουν χελώνα και να αλλάζουν τις διαστάσεις της. Να αλλάζουν τη μορφή της χελώνας χρησιμοποιώντας τα έτοιμα σχήματα. Να γνωρίζουν τις βασικές εντολές κίνησης της χελώνας (μπροστά, πίσω δεξιά, αριστερά, περίμενε, πάνω –pen up , κάτω – pen down κ.λπ.). Να σχεδιάζουν απλά γεωμετρικά σχήματα (τετράγωνο, παραλληλόγραμμο, κ.ά.) με τη γραφίδα της χελώνας.	Οι μαθητές ανακαλύπτουν τις δυνατότητες σχεδίασης ενός Logo like προγραμματιστικού περιβάλλοντος δημιουργώντας τις δικές τους ζωγραφιές. Εμπλουτίζουν τις εργασίες τους εισάγοντας χελώνες που τις κινούν με απλές εντολές. Αλλάζουν το σχήμα της χελώνας χρησιμοποιώντας έτοιμα γραφικά της εφαρμογής. Προγραμματίζουν τη χελώνα ώστε αυτή να κινείται όταν επιλέγεται με το ποντίκι. Αναλαμβάνουν τη δημιουργία ολοκληρωμένων σελίδων με κίνηση γραφικών όπως: η κίνηση στο δρόμο, τα ζώα στην εξοχή κ.ά. Μαθαίνουν για το στυλό της χελώνας. Πειραματίζονται με την κίνηση της χελώνας και την περιστροφή της. Θέτουν ερωτήματα, σχεδιάζουν, εφαρμόζουν και ανακαλύπτουν τους κανόνες σχεδίασης βασικών γεωμετρικών σχημάτων. Αναγνωρίζουν την αναγκαιότητα της εντολής επανάληψης. Μαθαίνουν την εντολή και τη χρησιμοποιούν στη σχεδίαση των σχημάτων τους. Πειραματίζονται με τη γωνία περιστροφής της χελώνας και ανακαλύπτουν πόσες μοίρες περιστρέφεται μια χελώνα για να ολοκληρώσει τη διαδρομή της γύρω από ένα κλειστό σχήμα. Ετοιμάζουν ερωτήσεις αντιστοιχίας, όπου οι συμμαθητές τους καλούνται να αντιστοιχίσουν ένα σύνολο εντολών με το αντίστοιχο παραγόμενο σχήμα και αντίστροφα.
	Στ	<b>Προγραμματίζω και ελέγχω</b>	Να συντάσσουν απλές διαδικασίες σε ένα <b>Logo like προγραμματιστικό περιβάλλον</b> . Να κατανοήσουν την έννοια της μεταβλητής. Να χρησιμοποιούν διαδικασίες με συνθήκη, παραμετρικές και αναδρομικές διαδικασίες.	Οι μαθητές συντάσσουν διαδικασίες για τη σχεδίαση γεωμετρικών σχημάτων στο προγραμματιστικό περιβάλλον της Logo. Συνδυάζουν τις διαδικασίες για το σχεδιασμό σύνθετων σχεδίων π.χ. κάστρο, χωριό, βάρκα, πλοίο κ.λπ. Εμπλουτίζουν τα σχέδια τους με χρώματα και με άλλα σχέδια από τα έτοιμα γραφικά. Μέσα από τα παραδείγματα, τα παιδιά κατανοούν ότι ο υπολογιστής εκτελεί τις οδηγίες που του δίνει ο άνθρωπος.
	Ε	<b>Προγραμματίζω τον υπολογιστή</b> Εκπαιδευτικά περιβάλλοντα οπτικού προγραμματισμού <i>Αλγόριθμος Πρόγραμμα</i>	Ο μαθητής/τρια πρέπει να είναι ικανός/ή : -Να αναγνωρίζει τις βασικές συνιστώσες ενός εκπαιδευτικού περιβάλλοντος οπτικού προγραμματισμού -Να περιγράφει με λεκτικό τρόπο τα βήματα απλών αλγορίθμων που καλείται να υλοποιήσει στο εκπαιδευτικό περιβάλλον οπτικού προγραμματισμού -Να διατυπώνει απλές εντολές στο περιβάλλον οπτικού	Βασικός στόχος της ενότητας αυτής είναι η σταδιακή εξοικείωση των μαθητών με τον προγραμματισμό μέσα από την αξιοποίηση διαθέσιμων εκπαιδευτικών περιβαλλόντων οπτικού προγραμματισμού. Οι μαθητές σε κατάλληλα εκπαιδευτικά περιβάλλοντα προγραμματισμού, χειρίζονται και διερευνούν έτοιμα προγράμματα και εισάγονται στην έννοια του αλγορίθμου, έχοντας ως γενικό προσανατολισμό τη μετάβαση από την ψηφιακή ζωγραφική στα προγραμματιζόμενα πολυμέσα. Ο εκπαιδευτικός, προκαλεί το ενδιαφέρον των μαθητών, ώστε να δημιουργήσουν μικρές εφαρμογές, στις οποίες θα προκαλείται η δράση αντικειμένων στη σκηνή ή στο χώρο επισκόπησης με

Πρόγραμμα Σπουδών «ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα)»	<p><i>Γεγονότα (events)</i> <i>Ανάπτυξη και εκτέλεση απλών εφαρμογών</i> <i>Διόρθωση σφαλμάτων</i> <i>Βελτιστοποίηση προγραμμάτων</i></p>	<p>προγραμματισμού</p> <ul style="list-style-type: none"> <li>-Να ορίζει ενέργειες και σενάρια που πρέπει να εκτελεστούν για να επιτευχθούν επιθυμητά γεγονότα</li> <li>-Να εξηγεί γιατί ένα αντικείμενο του προγραμματιστικού περιβάλλοντος συμπεριφέρεται με συγκεκριμένο τρόπο</li> <li>-Να κωδικοποιεί έναν αλγόριθμο σε προγραμματιστικό περιβάλλον και να αναπτύσσει μικρές εφαρμογές χρησιμοποιώντας ένα εκπαιδευτικό περιβάλλον οπτικού προγραμματισμού</li> <li>-Να αναλύει ένα πρόβλημα σε επιμέρους απλούστερα</li> <li>-Να συνθέτει ένα έργο από τα επιμέρους στοιχεία του (που έχουν προκύψει από την ανάλυση)</li> <li>-Να διακρίνει διάφορα γεγονότα (events) στο περιβάλλον οπτικού προγραμματισμού</li> <li>-Να εξοικειωθεί με τεχνικές διόρθωσης σφαλμάτων και βελτιστοποίησης των προγραμμάτων που αναπτύσσουν εκπαιδευτικό περιβάλλον προγραμματισμού</li> <li>-Να δημιουργεί σύνθετα έργα που βασίζονται στη σύνθεση</li> </ul>	<p>χρήση χειριστηρίων (πληκτρολόγιο, ποντίκι).</p> <p>Ενδεικτικά προτείνεται να χρησιμοποιηθεί παιχνίδι ρόλων με στόχο οι μαθητές να μυηθούν στον προγραμματισμό (π.χ. λογική της γεωμετρίας της χελώνας, αυστηρότητα διατύπωσης των εντολών κ.λπ.). Ένας μαθητής παίζει το ρόλο της χελώνας (του ρομπότ) και ένας άλλος είναι ο προγραμματιστής που τον κατευθύνει στο χώρο της τάξης (με τις εντολές Μπροστά, Αριστερά, Δεξιά). Ο εκπαιδευτικός παρουσιάζει το προγραμματιστικό περιβάλλον στους μαθητές αρχικά ως μια επέκταση του προγράμματος ζωγραφικής. Σχεδιάζουν εκτελώντας κατάλληλες εντολές απλά σχήματα.</p> <p>Στη συνέχεια, καλούνται να σχεδιάσουν γράμματα της αλφαβήτου όπως τα Ι, Γ, Π, Ξ, Τ, Ε, Η. Αναλύουν κάθε φορά και σχεδιάζουν στο χαρτί τα βήματα που χρειάζεται να πραγματοποιηθούν, μελετούν τα λάθη τις κωδικοποιήσεις τους και τα διορθώνουν. Στη συνέχεια, οι μαθητές τροποποιούν τα προγράμματά τους σχεδιάζοντας τα παραπάνω σχήματα με διαφορετικά χαρακτηριστικά (χρώμα, πάχος γραμμής κ.λπ.). Ακολουθεί συζήτηση σχετικά με κοινά σημεία που υπάρχουν στους διαφορετικούς αλγορίθμους που σχεδίασαν και στο πως ότι κάποια από αυτά θα μπορούσαν να επαναχρησιμοποιούνται. Στη συνέχεια, καλούνται να σχεδιάσουν σύνθετα σχήματα, αφού προηγηθεί η ανάλυσή τους σε απλούστερα γεωμετρικά σχήματα, όπως: ένα τετράγωνο ως η σύνθεση τεσσάρων ευθύγραμμων τμημάτων (ευθύγραμμη κίνησης και στροφής 90ο) ένα στίτι ως η σύνθεση ενός ορθογωνίου και ενός τριγώνου, ένα δένδρο ως σύνθεση ενός ορθογωνίου και ενός κύκλου (έλλειψης), μια σκάλα, ως η σύνθεση διαδοχικών Γ.</p>
	<p><b>Εκπαιδευτικό υλικό</b> Εκπαιδευτικά περιβάλλοντα οπτικού προγραμματισμού <b>EasyLogo, Scratch, BYOB, Kodu, Microworlds Pro gameMaker, K-turtle, Turtle Art openStarlogo</b> Εκπαιδευτική ρομποτική</p>	<p>Ο μαθητής/τρια πρέπει να είναι ικανός/ή:</p> <ul style="list-style-type: none"> <li>-Να εξηγεί γιατί ένα αντικείμενο του προγραμματιστικού περιβάλλοντος συμπεριφέρεται με συγκεκριμένο τρόπο</li> <li>-Να ορίζει ενέργειες και σενάρια που πρέπει να εκτελεστούν για να επιτευχθούν επιθυμητά γεγονότα</li> <li>-Να αντιλαμβάνεται την αναγκαιότητα και τη χρησιμότητα της δομής επανάληψης</li> <li>-Να χρησιμοποιεί εντολές επανάληψης στα προγράμματα που αναπτύσσει</li> <li>-Να αναλύει ένα πρόβλημα σε επιμέρους απλούστερα</li> <li>-Να συνθέτει ένα έργο προγραμματισμού από τα επιμέρους στοιχεία της ανάλυσης</li> <li>-Να αντιλαμβάνεται τη χρησιμότητα το ρόλο της διαδικασίας σε ένα πρόγραμμα</li> <li>-Να χρησιμοποιεί διαδικασίες στα έργα του</li> <li>-Να εφαρμόζει τεχνικές ελέγχου και διόρθωσης σφαλμάτων στα προγράμματα που δημιουργεί</li> <li>-Να κατανοεί τη λειτουργία έτοιμων προγραμμάτων που του δίνονται</li> </ul>	<p>Οι μαθητές υλοποιούν κατάλληλες δραστηριότητες ώστε να επεκτείνουν και να ενισχύσουν τις προγραμματιστικές τους δεξιότητες. Στόχος είναι, μέσα από την ενεργητική συμμετοχή τους, η διερευνητική προσέγγιση της γνώσης, η συνεργασία, η αυτενέργεια, η ανάπτυξη της δημιουργικότητας και της φαντασίας των μαθητών.</p> <p>Με τη χρήση κατάλληλων παραδειγμάτων αναδεικνύεται η ανάγκη της επαναχρησιμοποίησης τμήματος εντολών αλλά και της επαναληπτικής εκτέλεσης εντολών που είχαν τεθεί στην προηγούμενη τάξη. Παρουσιάζονται στους μαθητές οι έννοιες της διαδικασίας και της επανάληψης και καλούνται να επανασχεδιάσουν προγράμματα που έχουν ήδη υλοποιήσει. Οι μαθητές καθοδηγούνται να αναλύσουν τα σχήματα σε απλούστερα, να εντοπίσουν και να διορθώσουν λάθη στα προγράμματά τους.</p> <p>Ενδεικτικά παραδείγματα είναι: δημιουργία τρένου (σύνθεση βαγονιών, όπου κάθε βαγόνι είναι ένα ορθογώνιο με δύο κυκλικές ρόδες), δημιουργία πίνακα ζωγραφικής (με απλά γεωμετρικά σχήματα σε διαφορετικά μεγέθη και χρώματα), δημιουργία και κίνηση ανεμόμυλου (ως σύνθεση ορθογωνίου, τριγώνων και γραμμών). Ο προγραμματισμός της κίνησης ενός αντικειμένου (π.χ. μπάλα) στο χώρο ή σε λαβύρινθο. Προγραμματισμός διαδραστικών παιχνιδιών και ιστοριών</p> <p>Είναι δυνατή η χρήση προσομοιώσεων (java applets, flash animations), με στόχο οι μαθητές να προβληματιστούν για το πώς μπορούν να καταγράψουν τα βήματα επίλυσης ενός προβλήματος και πώς μπορούν να τα περιγράψουν στο προγραμματιστικό περιβάλλον.</p>
	<p><b>Εκπαιδευτικό υλικό</b> Περιβάλλοντα οπτικού προγραμματισμού <b>EasyLogo, Scratch BYOB, Kodu, Microworlds Pro, gameMaker, K-turtle Turtle Art, openStarlogo</b> Εκπαιδευτική ρομποτική</p>		

## 1.4 Η ΑΞΙΑ ΚΑΙ ΤΑ ΟΦΕΛΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η γνωστική αξία του Προγραμματισμού έχει επισημανθεί ήδη από τις αρχές της δεκαετίας του '80. Ο Προγραμματισμός υπολογιστών θεωρείται ένα ισχυρό μέσο για την ανάπτυξη δεξιοτήτων υψηλού επιπέδου, για τη διδασκαλία βασικών εννοιών που βρίσκουν εφαρμογή στα Μαθηματικά, τη Φυσική και τη Λογική, καθώς και τη μεταφορά δεξιοτήτων επίλυσης προβλημάτων σε άλλα γνωστικά αντικείμενα (Papert 1980; Howe et al. 1989). Ο Steve Jobs ανέφερε ότι όλοι πρέπει να μάθουν προγραμματισμό, διότι σου μαθαίνει πώς να σκέφτεσαι ενώ ο Hadi Partovi υποστηρίζει ότι κάθε μαθητής πρέπει να έχει την ευκαιρία να εμπλακεί με τον προγραμματισμό και πως όλα τα παιδιά πρέπει να μάθουν να προγραμματίζουν ως μέρος της βασικής τους εκπαίδευσης. «Κάθε μαθητής που θέλει να γίνει δικηγόρος, αρχιτέκτονας, γιατρός κ.α. πρέπει να μάθει να προγραμματίζει όπως ήδη μαθαίνει Μαθηματικά, Φυσική και Γεωμετρία» (Partovi, 2015).

Ο Μικρόπουλος αναφέρει τα οφέλη του προγραμματισμού ακόμα και για νεαρούς μαθητές υποστηρίζοντας ότι μέσα από τον προγραμματισμό μαθαίνουν το δομημένο και ιεραρχικό τρόπο σκέψης και την αντιμετώπιση προβλημάτων εκτός υπολογιστή. Δημιουργούν οι ίδιοι το περιβάλλον εργασίας της προτίμησης τους για διάφορα θέματα. Έτσι είναι σε θέση να κατασκευάζουν παιχνίδια εκπαιδευτικού περιεχομένου, να κτίζουν βάσεις δεδομένων, να προγραμματίζουν προσομοιώσεις, να δημιουργούν μικρόκοσμους σε διάφορα γνωστικά αντικείμενα. Μ' αυτόν τον τρόπο οι μαθητές αισθάνονται κυρίαρχοι της νέας τεχνολογίας και ταυτόχρονα έρχονται σε επαφή με τις ιδέες της επιστήμης και τη δημιουργία διανοητικών μοντέλων (Μικρόπουλος, 1997; Μικρόπουλος & Χαρίσης, 2008; Μικρόπουλος & Μπέλλου, 2010). Σε αυτό συμφωνούν και οι Resnick, Maloney, et al. (2009), που αναφέρουν ότι η χρήση δομημένου τρόπου σκέψης βοηθά στην επίλυση προβλημάτων διαφορετικού επιπέδου δυσκολίας, δίνει τη δυνατότητα στο άτομο να εντοπίσει τις αδυναμίες του και να τις βελτιώσει, επαυξάνοντας έτσι το γνωστικό του υπόβαθρο. Η μάθηση του προγραμματισμού μπορεί να οδηγήσει στην εγκαθίδρυση νέων τρόπων αντίληψης, μοντελοποίησης και επεξεργασίας προβλημάτων αναφέρει ο Κόμης (2005).

Οι Dufoyer (1988) και οι Hurtes, Allen, Stevens & Lee (2000) επισημαίνουν την ανάγκη της διδασκαλίας του προγραμματισμού στην εκπαίδευση και αποδέχονται ότι η μάθηση του προγραμματισμού μπορεί να οδηγήσει σε σημαντικές γνωστικές αλλαγές στους μαθητές που είναι

- α) Αυστηρότητα στη σκέψη, ακρίβεια στην έκφραση, συνειδητή ανάγκη για αποσαφήνιση των ενεργειών.
- β) Κατανόηση γενικών εννοιών, όπως διαδικασία, μεταβλητή, συνάρτηση, που

σχετίζονται άμεσα και με τη μαθηματική παιδεία. γ) Ευρετικές ικανότητες και μεθοδολογία όπως σχεδιασμός, αναζήτηση παρόμοιων περιπτώσεων, επίλυση με ανάλυση σε μέρη. δ) Μάθηση τεχνικών αναζήτησης λαθών που μπορούν να μεταφερθούν και σε άλλους χώρους εκτός προγραμματισμού. ε) Οικοδόμησης της λύσης με τη μορφή μικρών διαδικασιών ή στοιχειωδών τμημάτων, τα οποία μπορούν να συνδεθούν και να χρησιμοποιηθούν για την επίλυση πιο σύνθετων προβλημάτων. στ) Επέκταση της συνειδητοποίησης και της γνώσης πάνω σε τεχνικές επίλυσης προβλημάτων. ζ) Επέκταση και ανάπτυξη της χρήσης συγκριτικών μεθόδων που αφορούν στην πολλαπλότητα των τρόπων ώστε να επιτευχθεί ένας δεδομένος στόχος

Ο Τζιμογιάννης υποστηρίζει ότι κατά την επίλυση προβλημάτων σε προγραμματιστικά περιβάλλοντα χρησιμοποιούνται θεμελιώδεις έννοιες όπως η μεταβλητή, η δομή επιλογής, η δομή επανάληψης, η διαδικασία, που είναι δύσκολο να οικοδομηθούν από τους μαθητές με παραδοσιακά διδακτικά αντικείμενα και μέσα. Ο Προγραμματισμός μπορεί να βοηθήσει αποτελεσματικά τους μαθητές στην πρόσκτηση και λειτουργική εφαρμογή των εννοιών αυτών (Τζιμογιάννης, 2000; 2005).

Η διαδικασία της αποσφαλμάτωσης (debugging) αποτελεί σημαντικό χαρακτηριστικό της προγραμματιστικής δραστηριότητας. Ο Papert (1991) αναφέρει ότι πολλά παιδιά «μένουν πίσω» γνωστικά επειδή έχουν ένα μοντέλο μάθησης κατά το οποίο «είτε ξέρεις κάτι είτε δεν το ξέρεις σωστά». Όταν όμως μαθαίνεις να προγραμματίζεις ένα υπολογιστή, σχεδόν ποτέ δεν το κάνεις σωστά με την πρώτη φορά. «Το να μάθεις να είσαι έμπειρος προγραμματιστής σημαίνει ότι πρέπει να είσαι ικανός στην απομόνωση και διόρθωση σφαλμάτων, σημείων που αποτρέπουν την εκτέλεση του προγράμματος». Συνεχίζει λέγοντας ότι «η ερώτηση σχετικά με ένα πρόγραμμα δεν είναι αν είναι σωστό ή λανθασμένο, αλλά αν διορθώνεται» Papert (1991) . Η ανακάλυψη των λαθών από το μαθητή και συνεπώς η επίγνωση των νοητικών του λειτουργιών κατά τη λύση ενός προβλήματος αποτελεί μια μεταγνωστική δεξιότητα και η μεταγνώση συνιστά ώριμο αναπτυξιακό επίτευγμα του παιδιού που του επιτρέπει να ελέγχει τις νοητικές του λειτουργίες (Κόμης, 2005).

## **1.5 ΔΥΣΚΟΛΙΕΣ ΚΑΙ ΠΑΡΑΝΟΗΣΕΙΣ ΑΡΧΑΡΙΩΝ ΜΑΘΗΤΩΝ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ**

Παρά τα οφέλη και τις αξίες του προγραμματισμού, όπως αυτά αναφέρθηκαν παραπάνω, πολλοί επισημαίνουν ότι ο προγραμματισμός αποτελεί ένα δύσκολο αντικείμενο για τους αρχάριους μαθητές που εκδηλώνεται κατά την εκμάθηση και τη κατασκευή ενός αλγορίθμου ή ενός

προγράμματος στον υπολογιστή (Du Boulay, 1989; Soloway, 1983; Lahtinen, Ala-Mutka, & Järvinen, 2005; Jenkins, 2002; Milne, & Rowe, 2002; Μικρόπουλος, 1997; Μικρόπουλος & Χαρίσης, 2008; Μικρόπουλος & Μπέλλου, 2010). Όλοι οι παραπάνω συμφωνούν ότι ο προγραμματισμός είναι μια πολύ σύνθετη διαδικασία που απαιτεί προσπάθεια και ιδιαίτερη προσέγγιση στον τρόπο που διδάσκεται και προσεγγίζεται. Το πρόβλημα ξεκινάει για πολλούς μαθητές κατά την αρχική φάση της μάθησης, όταν πρέπει να κατανοήσουν και να εφαρμόσουν αφηρημένες προγραμματιστικές έννοιες, όπως είναι η δομή επανάληψης, η δομή επιλογής και οι μεταβλητές, ενώ οι ίδιοι δεν διαθέτουν ένα αποτελεσματικό νοητικό μοντέλο για την λειτουργία του υπολογιστή και τον τρόπο εκτέλεσης των προγραμματιστικών δομών.

Οι Gomes & Mendes, (2007) κατηγοριοποιούν αυτά τα προβλήματα σε πέντε βασικούς άξονες:

- 1) Στις διδακτικές τεχνικές (*The teaching methods*) που δεν είναι επαρκείς για τις ανάγκες των μαθητών για τους παρακάτω λόγους:
  - a) Η διδασκαλία δεν είναι εξατομικευμένη. Ο εκπαιδευτικός δεν είναι πάντα διαθέσιμος για να επιτρέψει περισσότερο εξατομικευμένη εποπτεία στους μαθητές με άμεση ανατροφοδότηση κατά την επίλυση προβλημάτων και αναλυτική επεξήγηση των λιγότερο κατανοητών πτυχών, που πιθανώς θα μπορούσε να τους βοηθήσει. Είναι σχεδόν αδύνατον να υπάρξει αυτή η υποστήριξη λόγω των χρονικών περιορισμών και του μεγέθους των μαθημάτων.
  - b) Οι τεχνικές των εκπαιδευτικών δεν υποστηρίζουν όλα τα στυλ των μαθητών οι οποίοι μαθαίνουν με διαφορετικούς τρόπους και έχουν διαφορετικές προτιμήσεις στη προσέγγιση ενός νέου αντικειμένου-θέματος. Στην παραδοσιακή εκπαίδευση όλοι οι μαθητές πρέπει να μάθουν με τον ίδιο ρυθμό και σύμφωνα με τις διδακτικές τεχνικές του εκπαιδευτικού. Είναι σημαντική ευθύνη για τον εκπαιδευτικό να διασφαλίσει ότι οι μαθητές υιοθετούν την πλέον ενδεδειγμένη προσέγγιση μάθησης (Jenkins, 2002).
  - c) Η διδασκαλία δυναμικών εννοιών μέσω στατικών υλικών. Ο προγραμματισμός περιλαμβάνει πολλές δυναμικές έννοιες που πολλές φορές διδάσκονται μέσω στατικών μέσων όπως παρουσιάσεις, προφορικές εξηγήσεις, διαγράμματα, μαυροπίνακες σχεδίων, κείμενα, κτλ. Για κάποιους μαθητές όμως αυτό είναι ένα πρόβλημα, δεδομένου ότι αδυνατούν να κατανοήσουν τη δυναμική του προγράμματος μέσω από υλικά αυτού του τύπου.
  - d) Οι εκπαιδευτικοί είναι πιο συγκεντρωμένοι στη διδασκαλία μια γλώσσας προγραμματισμού και στα συντακτικά της στοιχεία, από το να προωθήσουν την επίλυση προβλημάτων με τη χρήση μιας γλώσσας προγραμματισμού.

- 2) Στις μεθόδους μελέτης (*The study methods*) που ακολουθείται από πολλούς μαθητές, δεν είναι κατάλληλές για την εκμάθηση του προγραμματισμού.
- a) Οι μαθητές χρησιμοποιούν λανθασμένες μεθοδολογίες μελέτης. Πολλοί συνηθίζουν να λύνουν προβλήματα όπως σε άλλους επιστημονικούς κλάδους-πεδία, μέσω απομνημόνευσης τύπων ή διαδικασιών. Μερικές φορές οι μαθητές απομνημονεύουν τύπους, χωρίς την πλήρη κατανόηση της υποκείμενης έννοιας, απλά γνωρίζοντας ότι ένας συγκεκριμένος τύπος πρέπει να χρησιμοποιηθεί σε κάποιο πρόβλημα. Ο προγραμματισμός απαιτεί μια διαφορετική μέθοδο μελέτης. Θα πρέπει να είναι στην ουσία πρακτική, αλλά αρκετά διαφορετική από αυτή των άλλων πεδίων.
  - b) Οι μαθητές δεν εργάζονται αρκετά για να αποκτήσουν δεξιότητες προγραμματισμού.
- 3) Δεξιότητες και στάσεις των μαθητών (*The students' abilities and attitudes*) εννοώντας:
- a) Οι μαθητές δυσκολεύονται στην επίλυση προβλημάτων. Σημαντικός παράγοντας για την δυσκολία πολλών αρχάριων μαθητών στην εκμάθηση προγραμματισμού είναι η έλλειψη γενικών δεξιοτήτων επίλυσης προβλημάτων. Οι μαθητές δεν γνωρίζουν πώς να δημιουργήσουν αλγορίθμους κυρίως επειδή δεν γνωρίζουν τον τρόπο επίλυσης προβλημάτων (problem solving). Γι' αυτό απαιτείται η κατανόηση του προβλήματος αφού πολλοί μαθητές προσπαθούν να λύσουν ένα πρόβλημα χωρίς να το έχουν καταλάβει πλήρως. Η σχετιζόμενη γνώση αφού πολλοί μαθητές δεν καθιερώνουν σωστές αναλογίες με τα προβλήματα που έχουν λυθεί παλαιότερα και δεν μεταφέρουν την προηγούμενη γνώση στα νέα προβλήματα. Η συλλογιστική για το πρόβλημα και τη λύση, όπου οι μαθητές έχουν την τάση να γράφουν μια απάντηση, πριν την σκεφτούν προσεκτικά και χωρίς να έχουν κάνει περισσότερες εκτεταμένες δοκιμές. Τέλος, η έλλειψη επιμονής όπου συχνά οι μαθητές παραιτούνται από την επίλυση του προβλήματος αν δεν βρουν γρήγορα μια πιθανή λύση ενώ συχνά στην οποιαδήποτε δυσκολία προτιμούν να ζητήσουν λύση από έναν συμμαθητή τους αντί να συνεχίσουμε να προσπαθούμε να το επιλύσουν μόνοι τους.
  - b) Πολλοί μαθητές στερούνται από γνώσεις μαθηματικών και λογικής. Σε αυτό συμφωνούν και οι Gomes, Carmo, Bigotte & Mendes (2006) επισημαίνοντας ότι οι μαθητές αντιμετωπίζουν δυσκολίες απλών αριθμητικών υπολογισμών, σε έννοιες γεωμετρίας και τριγωνομετρίας, στην μεταφορά ενός προβλήματος από κείμενο σε μια μαθηματική φόρμουλα - εξίσωση για επίλυση. Η γνώση μαθηματικών είναι σημαντική για τον προγραμματισμό (Byrne & Lyons, 2001).



#### 4) Στην φύση του προγραμματισμού (The nature of programming)

a) Ο Προγραμματισμός απαιτεί υψηλό επίπεδο αφαίρεσης. Η εκμάθηση του προγραμματισμού απαιτεί δεξιότητες όπως η αφαίρεση, η γενίκευση, η μεταφορά και η κριτική σκέψη. Το πρόβλημα ξεκινά σε γενικές γραμμές κατά την αρχική φάση εκμάθησης, όταν οι μαθητές αναμένεται να κατανοήσουν και να εφαρμόσουν ορισμένες αφηρημένες προγραμματιστικές έννοιες, όπως οι δομές ελέγχου, επανάληψης κτλ, για την επίλυση προβλημάτων.

b) Αρκετές γλώσσες προγραμματισμού έχουν πολύ σύνθετους συντακτικούς κανόνες.

5) Ψυχολογικές Επιδράσεις (*Psychological Effects*). Οι μαθητές δεν έχουν κίνητρο για το προγραμματισμό γιατί υπάρχει ένας αρνητικός συνειρμός γι' αυτόν που μεταφέρεται από μαθητή σε μαθητή (Jenkins, 2002). Επιπροσθέτως, τα μαθήματα του προγραμματισμού έχουν αποκτήσει τη φήμη ότι είναι δύσκολα. Έτσι οι μαθητές προσεγγίζουν ένα μάθημα με την προσδοκία ότι θα είναι δύσκολο και με την έλλειψη εσωτερικών κινήτρων δύσκολα θα επιτύχουν.

Οι δυσκολίες και οι παρανοήσεις που έχουν οι μαθητές στην κατανόηση και εφαρμογή αυτών των προγραμματιστικών εννοιών διατυπώνονται παρακάτω.

Οι μαθητές αντιμετωπίζουν δυσκολίες στην εκμάθηση και σωστή χρήση των δομών. Συνήθως, τις μαθαίνουν προσωρινά και επιφανειακά με αποτέλεσμα να μην είναι σε θέση να τις ανακαλέσουν και να τις ενσωματώσουν στα καινούρια προγράμματα που φτιάχνουν (Du Boulay , 1989).

#### 1.5.1 ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

Η επαναληπτική δομή αποτελεί από τα πλέον διερευνημένα θέματα στον προγραμματισμό, είναι μία πολύ βασική προγραμματιστική έννοια που παρουσιάζει ιδιαίτερα διδακτικά προβλήματα για τους μαθητές που προσδιορίζονται στα εξής:

- Αδυναμία γενίκευσης: οι αρχάριοι προγραμματιστές παρουσιάζουν την τάση να χρησιμοποιούν στα προγράμματά τους μια λίστα επαναλαμβανόμενων εντολών αντί να χρησιμοποιούν ένα βρόχο (Hoc, 1989).
- Ανεπαρκή νοητά μοντέλα: οι αρχάριοι προγραμματιστές διαθέτουν ανεπαρκή μοντέλα για τις επαναληπτικές δομές (Kessler & Anderson, 1989). Μερικές φορές τα λάθη στα νοητά μοντέλα των αρχάριων προγραμματιστών είναι ανεπαίσθητα και δύσκολο να εντοπιστούν (Kahney, 1989).

- Αναλλοίωτες σχέσεις που διέπουν μια επαναληπτική δομή (Loop invariant): οι αρχάριοι προγραμματιστές δυσκολεύονται στον καθορισμό του τμήματος ενημέρωσης ενός βρόχου (loop invariant) που αποτελεί σημαντικό συστατικό του. Οι ίδιοι βασίζονται τα μοντέλα των βρόχων στην αναπαράσταση μιας αλληλουχίας ενεργειών (δυναμικό μοντέλο) και όχι στην αναπαράσταση των αναλλοίωτων σχέσεων μεταξύ των μεταβλητών (στατικό μοντέλο). Επίσης, χρησιμοποιούν διαφορετικά ονόματα σε κάθε βήμα της επανάληψης για τη σηματοδότηση μιας μεταβλητής με συγκεκριμένη λειτουργία (functional variable) και δεν έχουν την ικανότητα να προσδιορίσουν αυθόρμητα μια συνθήκη εξόδου (Rogalski & Samurcay, 1990).

Πέραν των δυσκολιών που αναφέρθηκαν παραπάνω οι μαθητές αντιμετωπίζουν δυσκολίες και στην επιλογή της καταλληλότερης δομής επανάληψης. Βιβλιογραφικά αναφέρονται ως προς την επιλογή των δομών τα εξής :

- Η επιλογή της καταλληλότερης επαναληπτικής δομής για ένα συγκεκριμένο πρόβλημα είναι αρκετά δύσκολη. Ακόμη και μαθητές με εμπειρία συναντούν δυσκολίες στην έκφραση σύνθετων επαναληπτικών δομών (Rogalski & Samurcay, 1990).
- Οι μαθητές συνήθως χρησιμοποιούν τη στρατηγική bottom-exit, δηλαδή τη δομή Αρχή\_επανάληψης ... μέχρις\_ότου (Pane & Myers, 1996).
- Ο μεγαλύτερος βαθμός δυσκολίας που παρουσιάζει η δομή Όσο ... επανάλαβε, σε σχέση τη δομή Αρχή\_επανάληψης ... μέχρις\_ότου, από τους αρχάριους προγραμματιστές (Soloway et al., 1983) ίσως να οφείλεται στη δυσκολία αναπαράστασης και εκφοράς μιας συνθήκης για ένα αντικείμενο το οποίο ακόμα δεν έχουν επεξεργαστεί.

### 1.5.2 ΔΟΜΗ ΕΠΙΛΟΓΗΣ

Η δομή αυτή επιτρέπει στον αλγόριθμο να επιλέγει τις εντολές που πρόκειται να εκτελεστούν ανάλογα με τα αποτελέσματα ελέγχου μιας δυαδικής συνθήκης. Χρησιμοποιείται για τη λήψη απόφασης μεταξύ δύο εναλλακτικών καταστάσεων, εκ των οποίων η μία είναι αληθής και η άλλη ψευδής. Ωστόσο, με τη δομή επιλογής, ο μαθητής εισέρχεται σε πιο σύνθετες έννοιες που απαιτούν έλεγχο για τη λήψη της τελικής απόφασης. Οι δυσκολίες που συναντούν οι μαθητές προκύπτουν από τη φύση των συνθηκών που ελέγχουν οι δομές. Σοβαρό πρόβλημα για τους μαθητές αποτελούν και οι εμφωλευμένες δομές επιλογής όπου ο βαθμός δυσκολίας τους αυξάνεται από τον αριθμό των

δομών που ενσωματώνεται μέσα σε κάποιες άλλες (Rogalski 1990; Green, 1980). Η δομή ελέγχου οικοδομείται με ιδιαίτερη δυσκολία από τους μαθητές, ενώ οι πρότερες μαθηματικές γνώσεις και οι γνώσεις λογικής παίζουν σημαντικό ρόλο. Σε γενικό επίπεδο οι έρευνες συγκλίνουν στο ότι οι μαθητές με ανεπτυγμένο γνωστικό υπόβαθρο στα μαθηματικά αντιμετωπίζουν μικρότερη δυσκολία (Κόμης, 2005). Μια ακόμη δυσκολία των μαθητών έγκειται κατά τη χρήση λογικών (Boolean) συναρτήσεων ή κατά τη δημιουργία εκφράσεων με τους λογικούς τελεστές ΚΑΙ (AND), Η΄ (OR) , ΟΧΙ (NOT) όπου οι μαθητές δυσκολεύονται στην κατανόηση της λειτουργίας μιας εντολής ελέγχου (Lahtinen, Ala-Mutka & Järvinen, 2005).

Σύμφωνα με τους Putnam (1989) και Sleeman (1988) τα πιο συχνά λάθη και παρανοήσεις των μαθητών που σχετίζονται με τις δομές επιλογής είναι:

- Στην περίπτωση που η δομή επιλογής δεν έχει τμήμα ELSE, αναμένουν τη διακοπή εκτέλεσης του προγράμματος και την εμφάνιση μηνύματος λάθους αν η συνθήκη της εντολής IF είναι ψευδής (false).
- Στην περίπτωση που η δομή επιλογής έχει και τμήμα ELSE, αναμένουν την εκτέλεση τόσο του τμήματος THEN όσο και του ELSE.
- Αναμένουν την εκτέλεση του τμήματος THEN ανεξάρτητα από το αν η συνθήκη είναι αληθής ή όχι.
- Στην περίπτωση που η δομή επιλογής δεν έχει τμήμα ELSE, διαχειρίζονται την αμέσως επόμενη εντολή (που δεν ανήκει στην IF..THEN) όπως η ELSE, πιστεύουν δηλαδή ότι η εντολή αυτή εκτελείται μόνο όταν η συνθήκη είναι ψευδής.

### 1.5.3 ΜΕΤΑΒΛΗΤΕΣ

Η έννοια της μεταβλητής αποτελεί δομικό λίθο της εισαγωγής στον προγραμματισμό, καθώς πρόκειται για ένα από τα βασικά στοιχεία ενός αλγορίθμου (Brookshear, 2005). Η μεταβλητή στον προγραμματισμό αντιπροσωπεύει έναν καταχωρητή της μνήμης του υπολογιστή, ενώ έχει πάντα εκχωρημένη μια τιμή, η οποία μπορεί να μεταβάλλεται δυναμικά καθώς εκτελείται το πρόγραμμα. Όταν οι μαθητές εισάγονται για πρώτη φορά στην έννοια της μεταβλητής, πολλές φορές συναντούν δυσκολίες στην αντίληψη του τρόπου λειτουργίας της μέσα στο πρόγραμμα (Kuittinen & Sajaniemi, 2004; Samurçay, 1989; Γλέζου & Γρηγοριάδου, 2007; Τζιμογιάννης & Κόμης, 2000). Όπως έχει φανεί από τις έρευνες, ορισμένες από τις πιο συχνές δυσκολίες που αντιμετωπίζουν οι μαθητές αφορούν:

- Στη διατήρηση τιμής της μεταβλητής: πολλές φορές οι μαθητές θεωρούν ότι η μεταβλητή σταματά να διατηρεί την τιμή της, όταν αυτή εκχωρείται σε μια άλλη μεταβλητή (π.χ.  $\alpha = \beta$ ). Σε άλλες περιπτώσεις έχει διαπιστωθεί ότι οι μαθητές πιστεύουν πως μια μεταβλητή μπορεί να διατηρεί ταυτόχρονα περισσότερες από μια τιμές ή ότι έχει την ικανότητα να «θυμάται» τις τιμές που της ανατέθηκαν.
- Στην εκχώρηση τιμής σε μεταβλητή: Δυσκολία παρουσιάζουν οι μαθητές και στην περίπτωση που σε μια μεταβλητή ανατίθεται μια τιμή με τη μορφή αριθμητικής έκφρασης (π.χ.  $x = y + 2$ ) όπου θεωρούν ότι η μεταβλητή αποθηκεύει την αριθμητική έκφραση και όχι την τιμή που προκύπτει από αυτή.
- Στη διαχείριση μεταβλητών διαφορετικών τύπων: Καθώς οι μαθητές είναι συνηθισμένοι από τα Μαθηματικά στη χρήση μεταβλητών με αριθμητικές τιμές, συχνά δυσκολεύονται να διαχειριστούν μεταβλητές που δέχονται τιμές άλλων τύπων, όπως π.χ. λογικές ή αλφαριθμητικές, και κυρίως όταν τις χρησιμοποιούν για τον έλεγχο λογικών συνθηκών.
- Στην αρχικοποίηση της τιμής της μεταβλητής: Οι μαθητές συχνά παραλείπουν να αναθέσουν αρχική τιμή όταν ορίζουν μια μεταβλητή με αποτέλεσμα να παραμένει κενή και να δημιουργούνται σφάλματα κατά την εκτέλεση του προγράμματός τους.

Έχει διαπιστωθεί πως οι παραπάνω δυσκολίες πολλές φορές προκύπτουν από παρανοήσεις των μαθητών που οφείλονται στην πρότερη γνώση που έχουν αποκτήσει για τη μεταβλητή από το μάθημα των Μαθηματικών (Τζιμογιάννης & Κόμης, 2000). Οι προγραμματιστικές μεταβλητές, ενώ μπορεί να έχουν κάποια κοινά χαρακτηριστικά με τις μαθηματικές, διαφέρουν σε αρκετά και σημαντικά σημεία. Παρ' όλα αυτά οι μαθητές τείνουν να συγχέουν τις ιδιότητες των δύο εννοιών, με αποτέλεσμα να επιχειρούν να μεταφέρουν ιδιότητες των μαθηματικών μεταβλητών στις προγραμματιστικές.

## 2 ΕΠΙΛΟΓΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ

Ο προγραμματισμός ως γνωστικό αντικείμενο απαιτεί πρακτική. Η θεωρία του προγραμματισμού είναι δύσκολο, αν όχι αδύνατο, να διδαχθεί χωρίς να συνδυαστεί με πρακτική εξάσκηση. Ο μαθητής κατανοεί πληρέστερα τον προγραμματισμό αν χρησιμοποιήσει τον υπολογιστή και γράψει τον δικό του κώδικα. Κατά συνέπεια ο εκπαιδευτικός που θα διδάξει το αντικείμενο πρέπει να αναζητήσει κάποιο Προγραμματιστικό Περιβάλλον για την πρακτική άσκηση των μαθητών του (Winslow, 1996).

Σύμφωνα με τον Μικρόπουλο (1997) ένα σημαντικό θέμα που προκύπτει είναι η επιλογή του κατάλληλου προγραμματιστικού περιβάλλοντος που ενδείκνυται ως εργαλείο για να ικανοποιήσει και να επηρεάζει άμεσα την πληροφορική παιδεία. Το πρόβλημα γίνεται δυσκολότερο όταν ο μαθητής δεν προέρχεται ή δεν ακολουθεί τις λεγόμενες θετικές επιστήμες και δεν έχει συναντήσει αντικείμενα παραπλήσιας λογικής. Τις περισσότερες φορές ο μαθητής μικρής ηλικίας, χρειάζεται απλά ένα ισχυρό εργαλείο για την αντιμετώπιση των προβλημάτων του και ένα περιβάλλον δράσης για συμβολή στην ανάπτυξη των νοητικών του μοντέλων.

Η Webb (2012) αναφέρει ότι για τον εκπαιδευτικό η εύρεση του κατάλληλου προγραμματιστικού περιβάλλοντος αποτελεί μια σοβαρή πρόκληση. Κάθε εφαρμογή μπορεί να αντιμετωπιστεί ως «τέλεια» ή «διασκεδαστική», γνωρίζοντας όμως τις έννοιες που υποστηρίζει και αν θα λειτουργήσει ή καλύτερα συνεργαστεί με τους τρέχοντες πόρους σε υλικό και λογισμικό αποτελούν κρίσιμες αποφάσεις για την υιοθέτηση τους στην εκπαιδευτική διαδικασία. Οι εκπαιδευτικοί πλέον δεν διερωτώνται για το αν θα χρησιμοποιήσουν την τεχνολογία για την ενίσχυση των μαθημάτων τους άλλα για το ποια τεχνολογία ή εφαρμογή θα χρησιμοποιηθεί. Γι' αυτούς η εύρεση του κατάλληλου εργαλείου ή προγραμματιστικού περιβάλλοντος για την εμπλοκή των μαθητών και την ικανοποίηση των διδακτικών τους στόχων κρίνεται αναγκαία.

Από το 1960 πολλές γλώσσες προγραμματισμού και περιβάλλοντα έχουν αναπτυχθεί με την πρόθεση να κάνουν τον προγραμματισμό προσιτό σε μεγαλύτερο αριθμό ανθρώπων (Kelleher & Pausch, 2005).

Ξεκινώντας την βιβλιογραφική επισκόπηση στα Προγραμματιστικά Περιβάλλοντα για Αρχάριους-ΠΠΑ (Novice Programming Environment) θα πρέπει να αναφερθεί ότι χωρίζονται σε δυο

μεγάλες κατηγορίες, στα περιβάλλοντα συγγραφής με κείμενο (Textual) με απλή σύνταξη και σημασιολογία και σε αυτά που αναπτύσσονται με μπλοκ – πλακίδια εντολών, *Block-Based Programming Environment* (BBPE) που επιτρέπουν στο χρήστη τη δημιουργία προγραμμάτων μέσα από το γραφικό χειρισμό προγραμματιστικών στοιχείων, αντί κειμένου. Πολλές έρευνες έχουν γίνει πάνω σε αυτές τις δυο κατηγορίες σε μια προσπάθεια σύγκρισης τους με αρχάριους μαθητές (Price, & Barnes, 2015; Price, Brown, Lipovac, Barnes, & Kölling, 2016; Weintrop & Wilensky, 2015; Weintrop, 2015; Bau, 2015; McCaffrey, 2006), καταλήγοντας στο συμπέρασμα ότι τα ΠΠΑ που αναπτύσσονται με κείμενο τείνουν να είναι πιο κατάλληλα για εξοικειωμένους προγραμματιστές ενώ τα ΠΠΑ με πλακίδια εντολών να προτιμώνται στη περίπτωση των αρχάριων προγραμματιστών γιατί είναι πιο εύκολα χρήση και στην ανάπτυξη προγραμμάτων. Αρκετοί βέβαια από τους παραπάνω αναφέρουν ότι παρά την αυξανόμενη δημοτικότητα και την ευρεία χρήση των ΠΠΑ με πλακίδια τα τελευταία χρόνια, λίγη δουλειά-έρευνα έχει γίνει για τον πληρέστερο προσδιορισμό της εννοιολογικής κατανόησης (*conceptual understanding*) των μαθητών από την χρήση αυτών των περιβαλλόντων (Weintrop, 2015; Weintrop, & Wilensky, 2015; Price, & Barnes, 2015).

Υπάρχουν αρκετές έρευνες που έχουν μελετήσει την επίδραση και εμπλοκή αρκετών ΠΠΑ σε μαθητές οι οποίοι μαθαίνουν προγραμματισμό αναφέροντας όλοι σε γενικές γραμμές μια θετική ανταπόκριση-απόκριση των μαθητών κατά την χρήση τους. Πιο συγκεκριμένα, στο Scratch (Meerbaum-Salant, Armoni, & Ben-Ari, 2011; Maloney et al. 2004; Wu et al. 2010; Malan & Leitner 2007; Gans, 2010) αναφέρουν ότι δεν ενθουσιάζει μόνο τους μαθητές άλλα τους εξοικειώνει με βασικές αρχές του προγραμματισμού χωρίς να αποσπάται η προσοχή τους από κανόνες σύνταξης, προσφέροντας τους ταυτόχρονα απλότητα και δύναμη, στοιχεία απαραίτητα για την εμπλοκή και την ενθάρρυνση των αρχάριων μαθητών. Στο StarLogo TNG (Klopfer & Yoon, 2004; McCaffrey, 2006) και στο Kodu (Stolee & Fristoe, 2011) αναφέρεται ότι μπορεί να χρησιμοποιηθεί για βασικές έννοιες προγραμματισμού και καταλήγουν ότι απαιτείται περαιτέρω ανάλυση για την επιβεβαίωση ότι οι χρήστες πράγματι αντιλαμβάνονται αυτές τις έννοιες. Στο GameSalade (Dekhane & Xu, 2012) υποστηρίζουν ότι είναι ένα περιβάλλον ικανό για την εμπλοκή και την αύξηση του ενδιαφέροντος των μαθητών στον προγραμματισμό. Στο Alice (Kelleher et al., 2007; Cooper, Dann & Pausch, 2000) καταλήγουν ότι είναι ένα περιβάλλον ικανό να προσφέρει ισχυρά πρώτα βήματα στον προγραμματισμό και μπορεί να δώσει στα κορίτσια μια θετική πρώτη εμπειρία με αυτό το αντικείμενο. Επίσης μπορεί να τα βοηθήσει εμπνέοντας περισσότερα στο να ξεκινήσουν και έτσι θα αρχίσει να διορθώνεται η υποεκπροσώπηση των γυναικών σε αυτό το πεδίο. Το Code.org (Kalelioğlu,

2015; Piech, Sahami, Huang, & Guibas, 2015) το χαρακτηρίζουν ως ψηφιακή παιδική χαρά για νεαρούς προγραμματιστές και υποστηρίζουν ότι είναι ένας νέος τρόπος εκπαίδευσης για τους μαθητές που τους εισάγει στην υπολογιστική σκέψη και στον προγραμματισμό μέσα από διασκεδαστικές δραστηριότητες. Στο Pencil Code (Bau, Dawson, Bau, & Pickens, 2015) που σχεδιάστηκε για να γεφυρώσει τους δυο τρόπος ανάπτυξης προγραμμάτων, πλακιδίων και κειμένου, είναι ένα ΠΠΑ με πλακίδια που βοηθά τους αρχάριους να μεταβούν ομαλά και στην ανάπτυξη με κείμενο.

Σε αυτό το σημείο δεν θα μπορούσαμε να παραλείψουμε τις βιβλιογραφικές μελέτες σύγκρισης προγραμματιστικών περιβαλλόντων για αρχάριους. Συγκεκριμένα αναφέρεται η σύγκριση του Alice, του Greenfoot και του Scratch, μια πολύ γενική δισέλιδη επισκόπηση που εξετάζονται χαρακτηριστικά βάσει χρονολογίας που εμφανίστηκαν, απεικόνισης του περιβάλλοντος δηλαδή αν είναι δυσδιάστατη όπως στην περίπτωση του Greenfoot και του Scratch και τρισδιάστατης όπως στην περίπτωση του Alice. Επιπλέον εξετάζουν με ποιο τρόπο γίνεται η ανάπτυξη του προγράμματος μέσα από κείμενο όπως στο Greenfoot ή μέσα από πλακίδια εντολών όπως στην περίπτωση των Alice και Scratch (Fincher, Cooper, Kölling, & Maloney. 2010). Συγκριτική μελέτη των ίδιων περιβαλλόντων παρουσιάζουν και οι Utting, Cooper, Kölling, Maloney & Resnick, (2010) πάνω στους στόχους, τους μηχανισμούς, και τα αποτελέσματα αυτών των περιβαλλόντων για την υποστήριξη, απόκτηση και ανάπτυξη προγραμματιστικών εννοιών σε αρχάριους μαθητές πριν την είσοδο τους στο πανεπιστήμιο (*pre-University and non-technical students*). Τα χαρακτηριστικά της συγκριτικής μελέτης βασίζονται στην ηλικία που ενδείκνυται ή καλύτερα που οι ίδιοι προτείνουν για να χρησιμοποιηθούν. Αναφέρουν ότι το Scratch μπορεί να χρησιμοποιηθεί για μαθητές ηλικίας από 8 έως 16 ετών, το Greenfoot από 14 και πάνω και το Alice από 15 και πάνω. Κάνουν λόγο για το φύλο των χρηστών εστιάζοντας περισσότερο στις γυναίκες και αν μπορεί κάποιο περιβάλλον να συμβάλλει σε αυτή τη «φυλετική ισορροπία», συμπεραίνοντας ότι και τα 3 περιβάλλοντα μπορούν βοηθήσουν σε αυτό αλλά με το Alice αναμένονται περισσότερο θετικά αποτελέσματα. Αναφέρουν ότι και τα 3 περιβάλλοντα «κρύβουν τις πολυπλοκότητες» επιτρέποντας στο χρήστη να αναπτύξει τον κώδικα και να εργαστεί με τις προγραμματιστικές έννοιες. Τέλος, οι σημαντικές διαφορές αυτών των τριών σχετικά με τον αντικειμενοστραφή προγραμματισμό για το Scratch επισημαίνουν ότι οι μορφές (sprites) που έχει θεωρούνται αντικείμενα που έχουν την δική τους στάση και συμπεριφορά, ικανά μόνο για να αντιγραφούν χωρίς όμως να προσφέρεται η δημιουργία κλάσεων και η κληρονομικότητα που είναι βασικά χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού. Το Greenfoot έχει το

αντικειμενοστραφές μοντέλο της Java ενώ το Alice βρίσκεται κάπου στην μέση αυτών των δυο, ως Java-like περιβάλλον, προσφέροντας μόνο ενός επιπέδου δημιουργία κλάσεων (Utting, Cooper, Kölling, Maloney & Resnick, 2010).

Άλλη σύγκριση που παρουσιάζεται είναι μεταξύ της Logo, που ανατάσσεται με κείμενο και του Scratch που αναπτύσσεται με πλακίδια. Η έκδοση της Logo που χρησιμοποιήθηκε ήταν από τον Guy Haas (<http://www.bfoit.org/itp/install.html>) για να μελετήσει πώς τα προγραμματιστικά περιβάλλοντα σχηματίζουν την αντίληψη, την μάθηση και τους στόχους σε φοιτητές του πανεπιστημίου του Berkeley. Κατέληξαν ότι ενώ το περιβάλλον της Logo εμφανίστηκε ικανό για να υποστηρίξει την ανάπτυξη της αυτοεκτίμησης των φοιτητών (*supporting student development of confidence*) και το ενδιαφέρον τους για τον προγραμματισμό στην κατανόηση της έννοιας της επανάληψης, με το Scratch οι φοιτητές είχαν καλύτερα μαθησιακά αποτελέσματα και ιδιαίτερα στην έννοια της επιλογής. Επίσης αναφέρουν ότι το Scratch προσφέρει μια σειρά από δυνατότητες που το κάνουν να είναι πιο εύκολο στη μάθηση και τη διερμηνεία (Lewis, 2010).

Μια ακόμη σύγκριση παρουσιάζεται μεταξύ του App Inventor και του GameSalad από τους Roy, Rouse, & DeMeritt (2012) σε παράγοντες α) υλικοτεχνικούς, εννοώντας το κόστος, δηλαδή αν είναι δωρεάν η χρήση και η εγκατάστασή τους και στην πολυπλοκότητα της εγκατάστασης, β) εκπαιδευτικούς που αναφέρονται στο αν υπάρχουν διαθέσιμοι εκπαιδευτικοί πόροι, στην ευκολία δημιουργίας αυτών των εκπαιδευτικών πόρων, στη δυνατότητα μετάβασης σε άλλα περιβάλλοντα και στο αν είναι κατάλληλα για την εισαγωγή βασικών προγραμματιστικών εννοιών, γ) χρηστικοί και λειτουργικοί παράγοντες αναφερόμενοι στην ποιότητα της διεπαφής του χρήστη, δηλαδή στο αν όλες οι λειτουργίες περιέχονται σε μια οθόνη όπως στο GameSalad ή αν χωρίζεται σε τρία βασικά μέρη όπως στο App Inventor. Στα ενσωματωμένα κοινωνικά χαρακτηριστικά δηλαδή αν επιτρέπεται η συλλογή έργων και κώδικα, επιτρέποντας τη λήψη ενός μεμονωμένου πλακιδίου εντολών από μια εφαρμογή και την ενσωμάτωσή του σε μια άλλη και τέλος αν λειτουργεί με όλα τα λειτουργικά συστήματα. Καταλήγουν στο ότι δεν υπάρχει καλύτερο από αυτήν τη σύγκριση αφού και τα δυο περιβάλλοντα έχουν τα ισχυρά και τα αδύνατα σημεία τους. Το GameSalad μπορεί να είναι πιο κατάλληλο σε σχολεία που έχουν υπολογιστές με το λειτουργικό σύστημα iOS και προτείνεται να χρησιμοποιείται για την εισαγωγή προγραμματιστικών εννοιών σε σχέση με το App Inventor που προτείνεται για πιο έμπειρους χρήστες (Roy, Rouse, & DeMeritt, 2012).

Τέλος, μια ακόμη σύγκριση μεταξύ του Scratch και App Inventor που έχει πάρα πολλά κοινά σημεία με την παραπάνω μελέτη αναφέρεται από τους Παπαδάκης, Ορφανάκης, Καλογιαννάκης, &



Ζαράνης (2014) συγκρίνοντας τα ως προς την εγκατάσταση, τη διεπαφή χρήστη, τη συλλογή εφαρμογών, στο αν μπορούν να δημιουργηθούν προγραμματιστικές έννοιες όπως οι διαδικασίες, στο αν ανήκουν στον οπτικό προγραμματισμό, αν χρησιμοποιούνται για εφαρμογές για έξυπνες συσκευές και για εφαρμογές φυσικής αλληλεπίδρασης. Καταλήγουν στο συμπέρασμα ότι και τα δυο ΠΠΑ έχουν τα δυνατά και τα αδύναμα σημεία τους. Καταλήγουν ότι το Scratch ενδεχομένως να είναι πιο κατάλληλο για διδασκαλία σε μικρούς μαθητές ή σε προγράμματα σπουδών των οποίων ο πρωταρχικός στόχος είναι η ευχάριστη ενασχόληση και όχι μια πιο σοβαρή γνωριμία με προγραμματισμό. Το App Inventor ενδεχόμενα να είναι πιο κατάλληλο για μια πιο επίσημη εισαγωγή στον προγραμματισμό στην οποία ο τελικός στόχος θα είναι η ανάπτυξη της ικανότητας του προγραμματισμού και η μετάβαση σε μια συμβατική γλώσσα (Roy et al., 2012). Για τη διδασκαλία του προγραμματισμού στο Δημοτικό Σχολείο ή στις πρώτες τάξεις του Γυμνασίου, αναφέρουν ότι είναι προτιμότερο κάποιος να ξεκινήσει με το Scratch και να συνεχίσει με το AI στις επόμενες τάξεις ή εκπαιδευτικές βαθμίδες.

Από τα παραπάνω προκύπτει το ερώτημα ποιο προγραμματιστικό περιβάλλον για αρχάριους ενδείκνυται να χρησιμοποιηθεί για την κάλυψη των διδακτικών αναγκών. Η ανακάλυψη και περισσότερο η επιλογή, λοιπόν, του κατάλληλου εργαλείου ή καλύτερα του πιο αποδοτικού προγραμματιστικού περιβάλλοντος για την εμπλοκή των μαθητών και την ικανοποίηση των διδακτικών τους στόχων, αποτελεί μια σοβαρή πρόκληση και ανάγκη.

Έτσι, στόχοι της παρούσας εργασίας είναι:

- i) Η συγκριτική παρουσίαση των διαθέσιμων προγραμματιστικών περιβαλλόντων για αρχάριους βάσει παιδαγωγικών και τεχνολογικών χαρακτηριστικών, ώστε να διευκολύνεται η διαδικασία επιλογής του καταλληλότερου από τον ενδιαφερόμενο εκπαιδευτικό για την ικανοποίηση των διδακτικών τους στόχων.
- ii) Η καταγραφή της επίδοσης των μαθητών κατά την εμπλοκή τους με αυτά.

## **2.1 ΤΑΞΙΝΟΜΙΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ**

Στη βιβλιογραφία αναφέρονται διάφορες ταξινομίες για την κατηγοριοποίηση προγραμματιστικών περιβαλλόντων. Μια από τις πρώτες είναι η ταξινόμια του Myers (1990), στην οποία τα συστήματα κατηγοριοποιούνται με βάση δύο συνιστώσες: α) τον τύπο της οπτικοποίησης,

δηλαδή τι οπτικοποιείται (κώδικας, δεδομένα, αλγόριθμος) και β) τη μορφή εμφάνισης, δηλαδή αν πρόκειται για στατική ή δυναμική οπτικοποίηση.

Συνεχίζοντας την έρευνα πάνω στις ταξινομίες, οι Stasko & Patterson (1992) χρησιμοποιούν τέσσερα κλιμακωτά κριτήρια που είναι η όψη (aspect) εννοώντας την οπτικοποίηση που δημιουργείται και τα τμήματα του προγράμματος που τονίζονται, το επίπεδο της αφάιρεσης (abstraction), δηλαδή αν η οθόνη είναι ισόμορφη με τα στοιχεία του προγράμματος, την κίνηση (animation), εννοώντας τη δυναμική ή τα κινούμενα σχέδια που φαίνονται κατά την εκτέλεση του προγράμματος και το επίπεδο αυτοματισμού (automation), εννοώντας τις όψεις που παράγονται κατά την εκτέλεση του προγράμματος με ή χωρίς την πυροδότηση κάποιων ενεργειών από τον χρήστη.

Οι Roman & Cox (1993) υποστηρίζουν ότι παρόλο που κάθε μία από τις παραπάνω ταξινομήσεις έχει τη δική της λογική και τη δική της ουσία και δεν βασίζονται σε κάποιο μοντέλο ή σε κάποια θεωρία. Οι ίδιοι χρησιμοποιούν πέντε κριτήρια κατηγοριοποίησης προγραμματιστικών περιβαλλόντων:

1. Ο σκοπός (scope): η πτυχή του προγράμματος που οπτικοποιείται.
2. Το επίπεδο αφάιρεσης (abstraction): το είδος της πληροφορίας που μεταφέρεται από την οπτικοποίηση. Η αφάιρεση κατηγοριοποιείται σε άμεση αναπαράσταση, σε δομική αναπαράσταση και σε συνθετική αναπαράσταση.
3. Η μέθοδος ανάπτυξης (specification method), δηλαδή πώς δημιουργείται η οπτικοποίηση.
4. Η διεπαφή (Interface): οι διευκολύνσεις που παρέχονται για την οπτική παρουσίαση των πληροφοριών. Αυτή χωρίζεται σε δυο βασικές κατηγορίες, στα γραφικά και στην αλληλεπίδραση. Η αλληλεπίδραση χωρίζεται σε αλληλεπίδραση μέσω ρυθμίσεων και σε αλληλεπίδραση μέσω εικόνων.
5. Την παρουσίαση (presentation) διευκρινίζοντας πώς το σύστημα χρησιμοποιείται για την μετάδοση της πληροφορίας.

Ο Price (1993) δημιουργεί τη δική του ιεραρχική ταξινομία για να αντιμετωπίσει δυο βασικά προβλήματα. Το πρώτο είναι η επιλογή χαρακτηριστικών με έναν πιο συστηματικό τρόπο και το δεύτερο είναι για να επιτραπεί η μελλοντική επέκταση και αναθεώρηση των περιβαλλόντων μιας και το πεδίο αυτό συνεχώς μεταβάλλεται και αναβαθμίζεται. Χρησιμοποιεί έξι κριτήρια που προκύπτουν από την ανάλυση 12 περιβαλλόντων προγραμματισμού:

1. Ο σκοπός (score) που περιγράφει το εύρος των προγραμμάτων ή αλγορίθμων που το σύστημα μπορεί να οπτικοποιήσει.
2. Το περιεχόμενο (content) που αναφέρεται στις πτυχές του προγράμματος ή του αλγορίθμου που θέλουμε να οπτικοποιήσουμε.
3. Η μορφή (form) που αφορά στα χαρακτηριστικά του αποτελέσματος της οπτικοποίησης, όπως το μέσο, τα γραφικά, οι αναπαραστάσεις και η κίνηση των αντικειμένων.
4. Η μέθοδος (method) που εξετάζει τον τρόπο με τον οποίο παράγεται η οπτικοποίηση.
5. Η αποτελεσματικότητα (effectiveness) εξετάζει πόσο αποτελεσματική είναι η οπτικοποίηση του αλγορίθμου τόσο σε τεχνολογικό όσο και σε μαθησιακό επίπεδο. Οι έρευνες που έχουν γίνει για αυτόν τον σκοπό είναι λίγες και δεν καλύπτουν όλα τα συστήματα, οπότε δεν μπορούν να βγουν ασφαλή συμπεράσματα.
6. Η αλληλεπίδραση (interaction) που υποδιαιρείται σε τρεις άξονες στο στυλ (style), που αναφέρεται σε ποια μέθοδο χρησιμοποιεί (μενού, κουμπιά, γραμμές εντολών, κτλ) ο χρήστης για να δώσει οδηγίες στο περιβάλλον, στην πλοήγηση (navigation), εννοώντας τον βαθμό της πλοήγησης που υποστηρίζεται από το περιβάλλον μέσω της οπτικοποίησης και στις κειμενικές ευκολίες (scripting facilities), δηλαδή αν το περιβάλλον παρέχει διευκολύνσεις για τη διαχείριση της καταγραφής και επανάληψης της συγκεκριμένης οπτικοποίησης.

Οι Naps et al., 2003 αναφέρονται επίσης στην αλληλεπίδραση (interaction) που αξιολογεί το βαθμό εμπλοκής του εκπαιδευόμενου με το λογισμικό σε παιδαγωγικό και γνωστικό επίπεδο. Χωρίζεται σε παρακολούθηση της οπτικοποίησης (viewing) όπου ο χρήστης απλά παρακολουθεί την οπτικοποίηση χωρίς να παρεμβαίνει. Σε ελεγχόμενη παρακολούθηση της οπτικοποίησης (controlled viewing) όπου μπορεί να ελέγχει την ταχύτητα εμφάνισης της οπτικοποίησης στην οθόνη ή να σταματάει εντελώς την εκτέλεση. Σε απόκριση σε ερωτήσεις (responding) που μπορεί να απαντάει σε ερωτήσεις κατά την εκτέλεση. Σε τροποποίηση (changing) που μπορεί να αλλάξει τα δεδομένα εισόδου και σε δημιουργία νέας οπτικοποίησης (constructing), δηλαδή σε δυνατότητες δημιουργίας δικής του οπτικοποίησης. Όσο υψηλότερο είναι το επίπεδο στην κλίμακα της ταξινόμησης τόσο μεγαλύτερα εκπαιδευτικά οφέλη αναμένονται για το μαθητή.

Οι Kelleher & Pausch το 2005 παρουσίασαν μια ενδιαφέρουσα ταξινόμηση σε γλώσσες και περιβάλλοντα προγραμματισμού για αρχάριους και έμπειρους χρήστες. Σύμφωνα με την έρευνα

τους υπάρχουν διάφορα προγραμματιστικά εργαλεία που υποστηρίζουν διάφορα προγραμματιστικά στυλ, προγραμματιστικές κατασκευές, εμφανίσεις του κώδικα και ενέργειες για την ανάπτυξή τους. Αναφέρουν ότι αν αυτά τα περιβάλλοντα εστιάσουν πάνω σε τρεις βασικούς άξονες α) στην απλοποίηση προγραμματιστικών τεχνικών, β) τη παροχή υποστήριξης στους εκπαιδευόμενους και γ) τη παροχή κινήτρων για τον προγραμματισμό από το να εστιάζουν μόνο στο πρώτο, που γίνεται στην πλειοψηφία τους, θα κατάφερναν να γίνουν περισσότερο προσβάσιμα.

Ο Εφόπουλος (2005) παρουσιάζει την δική του ταξινομία σε 22 προγραμματιστικά περιβάλλοντα για αρχάριους με βάση πέντε κριτήρια που είναι: α) αν είναι απλή εκπαιδευτική γλώσσα ή απλοποιημένη επαγγελματική, β) αν έχει δυναμική προσομοίωση εκτέλεσης, γ) αν εμπεριέχει εναλλακτικούς τρόπους σύνταξης των εντολών, δ) αν υπάρχει καθοδήγηση ενός ήρωα και τέλος ε) στο τί μπορεί να κατασκευαστεί και να δημιουργηθεί από το περιβάλλον.

Κλείνοντας το κομμάτι της ταξινομίας σε μια διαφορετική προσέγγιση η Webb (2012) προτείνει είκοσι προγραμματιστικά περιβάλλοντα που μπορούν επιτυχώς να υποστηρίξουν εκπαιδευτικές εφαρμογές. Κατηγοριοποιεί τα προγραμματιστικά περιβάλλοντα με βάση α) το λειτουργικό τους σύστημα, δηλαδή αν μπορούν να χρησιμοποιηθούν τοπικά και από ποιο λειτουργικό σύστημα ή αν μπορούν να χρησιμοποιηθούν διαδικτυακά β) Στις εννοιολογικές κατηγορίες (conceptual categorization), αναφερόμενη στις έννοιες που μια εφαρμογή δίνει έμφαση. Αυτή η κατηγορία υποδιαιρείται σε τέσσερις άξονες στην κοινωνική δικτύωση (social networking), στην οπτικοποίηση της πληροφορίας (information visualization), στην γνώση της πληροφορίας (information sense making) και στην υπολογιστική σκέψη (computational thinking).

Αξίζει να αναφερθεί ότι η λέξη αρχάριος σε όλους τους παραπάνω ερευνητές αναφέρεται σε χρήστες που μπορεί να είναι κάθε ηλικίας που εισάγονται στο προγραμματισμό και όχι απαραίτητα - αποκλειστικά σε μαθητές δημοτικού ή σε μαθητές νεαρής ηλικίας.

## 2.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ

Η παρούσα ενότητα αναφέρεται στα παιδαγωγικά και τεχνολογικά χαρακτηριστικά που χρησιμοποιήθηκαν για τη σύγκριση των 55 ΠΠΑ που παρουσιάζονται σε αυτήν την εργασία με έμφαση στους μαθητές πρωτοβάθμιας εκπαίδευσης. Θα πρέπει να διευκρινιστεί ότι τα προγραμματιστικά περιβάλλοντα που εξετάζονται δεν επιλέχθηκαν λόγω της δημοτικότητάς τους, αλλά επειδή είναι ενεργά και χρησιμοποιούνται αυτήν τη στιγμή.

Τα τεχνολογικά χαρακτηριστικά των ΠΠΑ υποδιαιρούνται σε 8 υποκατηγορίες οι οποίες κατανέμονται ως εξής: η απεικόνιση του περιβάλλοντος, η διεπαφή, η ελεύθερη διανομή, οι προγραμματιστικές έννοιες που υποστηρίζει, ο τρόπος λειτουργίας, οι λειτουργικές απαιτήσεις του υλικού και του λογισμικού, η υποστήριξη της ελληνικής γλώσσας και η δημιουργία.

Τα παιδαγωγικά χαρακτηριστικά κατηγοριοποιούνται στην ικανοποίηση των στόχων του προγράμματος σπουδών, στη δυνατότητα μετάβασης σε άλλα προγραμματιστικά περιβάλλοντα, στην υποστήριξη από την κοινότητα μάθησης, στην αλληλεπίδραση και στις τάξεις που προτείνεται.

Τα τεχνολογικά χαρακτηριστικά όπως αποτυπώνονται στον πίνακα 2 αναλύονται παρακάτω:

- Η απεικόνιση του περιβάλλοντος αναφέρεται στον τρόπο με τον οποίο παρουσιάζεται η οπτικοποίηση των ΠΠΑ και κατηγοριοποιείται σε δισδιάστατη ή τρισδιάστατη απεικόνιση, όπως αναφέρεται και από τους Utting, Cooper, Kölling, Maloney & Resnick, (2010); Fincher, Cooper, Kölling, & Maloney, (2010).
- Η διεπαφή αναφέρεται στον τρόπο ανάπτυξης των προγραμμάτων και κατηγοριοποιείται σε ανάπτυξη με κείμενο (text) ή με πλακίδια εντολών (block). Στην κατηγορία των πλακιδίων ενσωματώνεται και η ανάπτυξη προγραμμάτων με κωδικοποιημένες εικόνες (Utting, Cooper, Kölling, Maloney & Resnick, 2010; Fincher, Cooper, Kölling, & Maloney, 2010).
- Στην ελεύθερη διανομή εξετάζεται αν ο χρήστης μπορεί να χρησιμοποιήσει ή να μεταμορφώσει ελεύθερα το ΠΠΑ. Επίσης, επισημαίνεται σε ποια περιβάλλοντα απαιτείται η εγγραφή του χρήστη, καθώς και αν επιτρέπεται να χρησιμοποιηθεί η βασική του έκδοση ή η πλήρης έκδοση (Duncan, Bell, & Tanimoto, 2014).
- Στις προγραμματιστικές έννοιες που υποστηρίζει. Οι προγραμματιστικές έννοιες για το δημοτικό που προτείνονται από το Πρόγραμμα Σπουδών είναι τα γεγονότα (events), η

επαναληπτική δομή, η δομή επιλογής, η διαδικασία (υποπρόγραμμα), η κλήση διαδικασιών και η διόρθωση σφαλμάτων. Στη συγκεκριμένη μελέτη για λόγους πληρότητας κρίθηκε απαραίτητο να μελετηθούν όλες οι προγραμματιστικές έννοιες που θα μπορούσε ή καλύτερα υποστηρίζει ένα ΠΠΑ εξετάζοντας τη δομή ακολουθίας, τη δομή επιλογής, τη δομή επανάληψης, τις μεταβλητές, τους αριθμητικούς τελεστές, τους συγκριτικούς τελεστές, τους λογικούς τελεστές, τις διαδικασίες, την αναδρομή, την παράλληλη εκτέλεση, τον προγραμματισμό γεγονότων και τον αντικειμενοστραφή προγραμματισμό (Webb, 2012; Lewis, 2010). Παρακάτω γίνεται μια προσπάθεια ορισμού των προγραμματιστικών εννοιών που είναι η διαδικασία, η αναδρομή, η παράλληλη εκτέλεση, ο προγραμματισμός γεγονότων και ο αντικειμενοστραφής προγραμματισμός.

- Διαδικασία καλείται μία σειρά προγραμματιστικών εντολών με συγκεκριμένο *σημείο εισόδου εκτέλεσης* και ένα ή περισσότερα *σημεία εξόδου*, δηλαδή εντολές διακλάδωσης σε άλλο τμήμα του ευρύτερου εκτελούμενου προγράμματος. Οι διαδικασίες είναι αυτοτελή τμήματα ενός προγράμματος τα οποία επιτελούν μία συγκεκριμένη εργασία και στα οποία ανατίθενται διακριτές ονομασίες. Μέσω αυτών των ονομάτων μπορούν να καλούνται κατά βούληση, και πιθανώς επανειλημμένα, από το ευρύτερο εκτελούμενο πρόγραμμα.
- Αναδρομή (recursion) στα μαθηματικά και την επιστήμη υπολογιστών είναι μια μέθοδος για τον ορισμό συναρτήσεων κατά την οποία η οριζόμενη συνάρτηση εφαρμόζεται στον ίδιο της τον ορισμό. Επίσης, ο όρος χρησιμοποιείται γενικότερα για να περιγράψει τη διαδικασία όπου αντικείμενα επαναλαμβάνονται με αυτο-όμοιο τρόπο. Για παράδειγμα, όταν οι επιφάνειες δυο καθρεπτών είναι σχεδόν παράλληλες μεταξύ τους, τα φωλιασμένα είδωλα που προκύπτουν αποτελούν μια μορφή αναδρομής.
- Η παράλληλη εκτέλεση είναι μια τεχνική στην οποία τα προγράμματα ή τα αντικείμενα του προγράμματος έχουν σχεδιαστεί ως συλλογές αλληλεπιδρώντων διαδικασιών που μπορούν να εκτελεστούν παράλληλα εκτελώντας το καθένα το δικό του κομμάτι κώδικα.
- Ο προγραμματισμός γεγονότων (event-driven programming) είναι το παράδειγμα προγραμματισμού όπου η ροή του προγράμματος εξαρτάται από εισερχόμενα γεγονότα. Για παράδειγμα, σε μια γραφική διεπαφή χρήστη έχουμε ως γεγονότα τα

πατήματα πλήκτρων ή επιλογών με το ποντίκι ή μηνύματα από άλλα προγράμματα ή μέρη του ιδίου προγράμματος (Lee, 2011).

- Ο αντικειμενοστραφής προγραμματισμός ορίζεται ως η μεθοδολογία ανάπτυξης προγραμμάτων υποστηριζόμενη από κατάλληλες γλώσσες προγραμματισμού, όπου ο χειρισμός σχετιζόμενων δεδομένων και των διαδικασιών που επενεργούν σε αυτά γίνεται από κοινού, μέσω μίας δομής δεδομένων που τα περιβάλλει ως αυτόνομη οντότητα με ταυτότητα και δικά της χαρακτηριστικά. Αυτή η δομή δεδομένων καλείται αντικείμενο και αποτελεί πραγματικό στιγμιότυπο στη μνήμη ενός σύνθετου, και πιθανώς οριζόμενου από τον χρήστη, τύπου δεδομένων που ονομάζεται κλάση. Η κλάση προδιαγράφει τόσο τα δεδομένα όσο και τις διαδικασίες οι οποίες επιδρούν επάνω τους. Βασικά χαρακτηριστικά ενός αντικειμενοστραφούς περιβάλλοντος είναι η υποστήριξη αφηρημένων τύπων δεδομένων (data abstraction), η ενθυλάκωση (encapsulation), η κληρονομικότητα (inheritance) και ο πολυμορφισμός (polymorphism) (Rentsch, 1982; Pokkunuri, 1989 ; Smith, 2015).
- Στον τρόπο λειτουργίας αναφέρεται η δυνατότητα χρήσης του ΠΠΑ διαδικτυακά με την χρήση κάποιου φυλλομετρητή ή τοπικά με την εγκατάσταση του προγράμματος (Webb, 2012).
- Στις λειτουργικές απαιτήσεις του υλικού και του λογισμικού αναφέρονται οι ελάχιστες απαιτήσεις που πρέπει να έχει ο υπολογιστής από άποψη υλικού και λογισμικού για την εύρυθμη λειτουργία του περιβάλλοντος, όπως αναλυτικά αναφέρονται στην ιστοσελίδα του κάθε εξεταζόμενου ΠΠΑ.
- Στην υποστήριξη της ελληνικής γλώσσας εξετάζεται αυτή η δυνατότητα του περιβάλλοντος και αποτελεί σημαντικό χαρακτηριστικό όταν πρόκειται να αξιοποιηθεί - χρησιμοποιηθεί από μαθητές μικρής ηλικίας.
- Στη δημιουργία αναφέρεται στο τι μπορεί να δημιουργήσει και να κατασκευάσει ο μαθητής κατά την χρήση του ΠΠΑ (Εφόπουλος, 2005).

Τα παιδαγωγικά χαρακτηριστικά όπως αποτυπώνονται στον πίνακα 3 είναι:

- Η ικανοποίηση στόχων του προγράμματος σπουδών. Αναφέρεται στη χρήση του ΠΠΑ κατά την εκπαιδευτική διαδικασία, δηλαδή στη δυνατότητα ικανοποίησης και πραγματοποίησης

όλων των στόχων όπως ακριβώς αποτυπώνονται στο Πρόγραμμα Σπουδών για το Δημοτικό: «να εξηγεί γιατί ένα αντικείμενο του προγραμματιστικού περιβάλλοντος συμπεριφέρεται με συγκεκριμένο τρόπο, να ορίζει ενέργειες και σενάρια που πρέπει να εκτελεστούν για να επιτευχθούν επιθυμητά γεγονότα, αντιλαμβάνεται την αναγκαιότητα και τη χρησιμότητα της δομής επανάληψης, να χρησιμοποιεί εντολές επιλογής στα προγράμματα που αναπτύσσει, να αναλύει ένα πρόβλημα σε επιμέρους απλούστερα, να συνθέτει ένα έργο προγραμματισμού από τα επιμέρους στοιχεία της ανάλυσης, να αντιλαμβάνεται τη χρησιμότητα το ρόλο της διαδικασίας σε ένα πρόγραμμα, να χρησιμοποιεί διαδικασίες στα έργα του, να εφαρμόζει τεχνικές ελέγχου και διόρθωσης σφαλμάτων στα προγράμματα που δημιουργεί, να κατανοεί τη λειτουργία έτοιμων προγραμμάτων που του δίνονται» (ΥΠΔΒΜΘ, 2011).



- Η δυνατότητα μετάβασης σε άλλα προγραμματιστικά περιβάλλοντα. Αναφέρεται στην ικανότητα του μαθητή μετά τη χρήση του περιβάλλοντος να μπορεί να μεταβεί σε κάποιο άλλο ΠΠΑ και να το χρησιμοποιήσει (Bau, Dawson, Bau, & Pickens, 2015; Price & Barnes, 2015). Η μετάβαση τους βοηθάει να αναπτύξουν δεξιότητες για μελλοντικές μεταβάσεις σε άλλα προγραμματιστικά περιβάλλοντα που θα είναι απαραίτητες, όταν θα αναπτυχθούν νέα προγραμματιστικά περιβάλλοντα που σήμερα ίσως δεν μπορούμε ούτε να φανταστούμε τις δυνατότητές τους (Μικρόπουλος, 2007).
- Η υποστήριξη από κοινότητα μάθησης. Αναφέρεται στη δυνατότητα κοινής χρήσης των μαθησιακών δραστηριοτήτων ή συνδέσεων μεταξύ χρηστών για τη λήψη βοήθειας ή την ανταλλαγή πληροφοριών που σχετίζονται με προσωπικούς στόχους ή στόχους του έργου. Αρκετές από τις εφαρμογές που εμπίπτουν σε αυτήν την κατηγορία επιτρέπουν τη διαδικτυακή συνομιλία μέσω chat και e-mail ως αναπόσπαστο κομμάτι της εφαρμογής τους. Υποστηρίζουν επίσης τη δημιουργία blogs και ιστοσελίδων για χρήση ως χώρους συνεργασίας (Webb, 2012).
- Η αλληλεπίδραση αποτελεί σημαντική παράμετρο για την παιδαγωγική αποτελεσματικότητα ενός ΠΠΑ σύμφωνα με τους Naps et al. (2003) και Hundhausen et al. (2002). Για την ανάλυση του βαθμού αλληλεπίδρασης χρησιμοποιείται η ταξινόμια του Naps (2003) όπως έχει αναλύθει. Όσο υψηλότερο είναι το επίπεδο στην κλίμακα της ταξινόμιας τόσο μεγαλύτερα εκπαιδευτικά οφέλη αναμένονται για το μαθητή. Γι' αυτό το









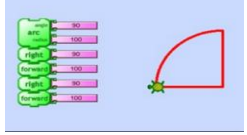



λόγο οι τρεις τιμές που δίνονται για το χαρακτηρισμό αυτής είναι η υψηλή, η μέτρια και η χαμηλή.






- Η ηλικία αναφέρεται στις ηλικίες των μαθητών που προτείνεται και μπορεί να χρησιμοποιηθεί το ΠΠΑ, έτσι όπως αυτές διαμορφώνονται και διαβαθμίζονται στα Ελληνικά σχολεία από τα προγράμματα σπουδών πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης. Για τον προσδιορισμό της ηλικίας χρησιμοποιήθηκαν τα προγράμματα σπουδών, οι δικτυακοί τόποι του κάθε ΠΠΑ και τέλος οι έρευνες των Duncan, Bell, & Tanimoto (2014) και του Good (2011), που κατηγοριοποιούν κάποια ΠΠΑ σύμφωνα με την ηλικία του μαθητή.


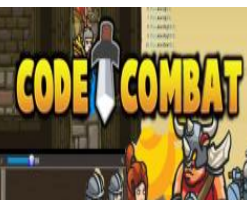
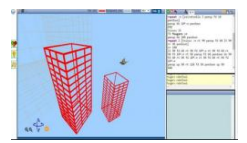


Πίνακας 2.1 Τεχνολογικά χαρακτηριστικά Προγραμματιστικών Περιβαλλόντων

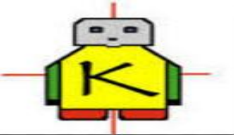




ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ	Απεικόνιση περιβάλλοντος		Διεπαφή		Ελεύθερη διανομή	Προγραμματιστικές έννοιες που υποστηρίζει													Τρόπος λειτουργίας	Λειτουργικές απαιτήσεις (υλικού-λογισμικού)	Υποστήριξη Ελληνικής Γλώσσας	Δημιουργία
	Δυσδιάστατη	Τρισδιάστατη	Πλακίδια εντολών	Εντολές κειμένου		Δομή Ακολουθίας	Δομή Επιλογής	Δομή Επανάληψης	Μεταβλητές	Αριθμητικοί Τελεστές	Συγκριτικοί Τελεστές	Λογικοί Τελεστές	Διαδικασίες	Αναδρομή	Παράλληλη Εκτέλεση	Προγραμματισμός Γεγονότων	Αντικειμενοστραφής Προγ/σμός					
	✓		✓		✓	✓	-	✓	-	-	-	✓	-	-	-	-	Τοπικά	Windows	NAI	-Σχέδια		
	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓ <sup>1</sup>	✓ <sup>1</sup>	✓	✓	-	Τοπικά Διαδικτυακά	-Windows -Ubuntu Linux -Mac	NAI	-Διαδραστικές ιστορίες -Παιχνίδια -Μουσική -Κινούμενες εικόνες -Σχέδια		






 <p>SCRATCH JR</p>	✓		✓		✓	✓	-	✓	✓	-	-	-	-	-	✓	✓	-	-Τοπικά με εγκατάσταση προσομοιωτή για κινητά	-Android -iPad	OXI	-Παιχνίδια -Κινούμενες εικόνες -Ηχογραφήσεις
 <p>KODU</p>		✓	✓		✓	✓	-	✓	-	-	-	-	-	-	✓	✓	-	Τοπικά	-Windows -Κάρτα Γραφικών με DirectX 9	NAI	-Παιχνίδια
 <p>GAMEMAKER</p>	✓		✓		✓ <sup>2</sup>	✓	✓	✓	✓	-	✓	-	-	-	✓	✓	✓	Τοπικά	-Windows	OXI	-Παιχνίδια -Κινούμενες εικόνες
 <p>MICROWORLDS PRO</p>	✓		✓		✓	✓	✓	✓	✓	✓	-	✓	-	-	-	-	-	Τοπικά	-Windows -Ubuntu Linux	NAI	-Σχέδια -Κινούμενες εικόνες
 <p>BYOB/SNAP</p>	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-Τοπικά -Διαδικτυακά μόνο με Chrome browser	-Windows -Ubuntu Linux -Mac	NAI	-Διαδραστικές ιστορίες -Παιχνίδια -Μουσική -Κινούμενες εικόνες -Σχέδια

 <p>K- TURTLE</p>	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	Τοπικά	-Windows <sup>3</sup> -Mac <sup>3</sup> -Ubuntu Linux	OXI  -Σχέδια -Κινούμενες εικόνες
 <p>TURTLE ART</p>	✓		✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	-	-	-	Τοπικά	-Windows -Mac -Ubuntu Linux	OXI  -Σχέδια -Κινούμενες εικόνες
 <p>OPENSTARLOGO</p>	✓		✓	✓	✓	✓	✓	✓	✓	-	✓	-	-	-	-	-	-	Τοπικά	-Windows -Mac -Ubuntu Linux	OXI  -Σχέδια -Κινούμενες εικόνες
 <p>STARLOGO TNG</p>		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓ <sup>4</sup>	Τοπικά Διαδικτυακά <sup>4</sup>	-Windows -Mac -Ubuntu Linux -CPU Intel Pentium 4 στα 1GHz -Κάρτα Γραφικών με DirectX 9 -Μνήμη 512Mb -Εγκατάσταση της Java.	NAI  -Διαδραστικές ιστορίες -Παιχνίδια -Κινούμενες εικόνες -Μοντέλα -Προσομοιώσεις
 <p>ALICE</p>		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Τοπικά	-Windows -Ubuntu Linux -Mac -Κάρτα Γραφικών με DirectX 9 -Μνήμη 1 Gb	OXI  -Διαδραστικές ιστορίες -Παιχνίδια -Κινούμενες εικόνες






 E-TOYS	✓		✓		✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-	Τοπικά	-Windows -Ubuntu Linux -Mac	ΟΧΙ	-Σχέδια -Κινούμενες εικόνες
 MIT App Inventor APP INVENTOR	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Τοπικά Διαδικτυακά	-Windows -Ubuntu Linux -Mac -Κινητά τηλέφωνα -Χρήση με Chrome ή Firefox -Απαιτείται εξομοιωτής για τηλέφωνα	ΟΧΙ	-Εφαρμογές για κινητά
 @codeorg #hourofcode CODE.ORG	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	Διαδικτυακά	-	ΝΑΙ	-Διαδραστικές ιστορίες -Παιχνίδια -Κινούμενες εικόνες -Σχέδια
 LIGHTBOT	✓		✓		✓	✓	✓	-	-	-	-	✓	-	-	-	-	-	Διαδικτυακά	-	ΝΑΙ	-Παιχνίδια
 KODABLE	✓		✓		✓	✓	✓	✓	-	-	-	✓	-	-	-	-	✓	Διαδικτυακά	-	ΟΧΙ	-Παιχνίδια





 BLOCKLY	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓	Διαδικτυακά	-	ΝΑΙ	-Παιχνίδια
 CODECOMBAT	✓		✓		✓ <sup>5</sup>	✓	✓ <sup>5</sup>	✓	✓	✓ <sup>5</sup>	✓ <sup>5</sup>	✓ <sup>5</sup>	✓ <sup>5</sup>	✓ <sup>5</sup>	✓ <sup>5</sup>	-	✓ <sup>5</sup>	✓ <sup>5</sup>	Διαδικτυακά	-	ΝΑΙ	-Παιχνίδια
 ΧΕΛΩΝΟΣΦΑΙΡΑ		✓		✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	-	-	-	Διαδικτυακά	-	ΝΑΙ	-Σχέδια -Δυναμικά γραφικά -Μοντέλα
 GAMESALAD	✓		✓		✓ <sup>2</sup>	✓	✓	-	✓	✓	✓	✓	-	-	✓	✓	-	-	Τοπικά	-Windows -Mac - Κάρτα Γραφικών με 256 RAM -Μνήμη 2 Gb	ΟΧΙ	Παιχνίδια
 ROBOMIND	✓		✓		✓ <sup>5</sup>	✓	✓	✓	-	-	-	✓	✓	✓	-	✓	-	-	Τοπικά Διαδικτυακά <sup>6</sup>	-Windows -Ubuntu Linux -Mac -Μνήμη 512 Mb -CPU Intel Pentium 4	ΝΑΙ	-Κίνηση Robot

 <p>KAREL THE ROBOT</p>	√		√	√	√	√	√	√	-	-	-	√	-	-	-	√	Τοπικά	-Windows -Ubuntu Linux -Mac	OXI	-Κίνηση Robot
 <p>NETLOGO</p>	√	√	√	√ <sup>5</sup>	√	√	√	√	√	√	√	√	-	√	√	√	Τοπικά	-Windows -Ubuntu Linux -Mac -Κάρτα Γραφικών με 512 RAM	OXI	Προσομοίωση φυσικών φαινομένων
 <p>PYTHON PYGAME</p>	√		√	√	√	√	√	√	√	√	√	√	√	√	√	√	Τοπικά	-Windows -Ubuntu Linux -Mac -Android	OXI	-Παιχνίδια -Πολυμεσικά προγράμματα
 <p>PYTHON TURTLE</p>	√		√	√	√	√	√	√	√	√	√	√	-	-	-	√	Τοπικά	-Windows -Mac	OXI	-Κίνηση -Σχέδια
 <p>GAMESTAR MECHANIC</p>	√		√	√ <sup>5</sup>	√	√	√	√	-	-	-	-	-	√	-	-	Διαδικτυακά	-	OXI	-Παιχνίδια






 <p>Code Monster CRUNCHZILLA CODE MONSTER</p>	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	Διαδικτυακά	-	ΟΧΙ	Σχέδια Κατασκευές
 <p>The WATERBEAR welcomes you!</p> <p>Waterbear is a toolkit for making programming more accessible and having a visual language means it's easier for focus on learning a syntax programming.</p> <p>Try it out</p> <p>WATERBEARlang</p>	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Διαδικτυακά	-	ΟΧΙ	-Διαδραστικές ιστορίες -Παιχνίδια -Μουσική -Κινούμενες εικόνες -Σχέδια
 <p>AgentSheets® computational thinking tools</p> <p>AGENTSHEETS</p>	✓	✓ <sup>7</sup>	✓	-	✓	✓	-	✓	✓	✓	-	-	-	✓	✓	-	Τοπικά Διαδικτυακά	-Windows 7 1 GB RAM CPU 1.0 GHz -Mac OS X 10.7 2GB RAM CPU 1.0 GHz	ΝΑΙ	-Παιχνίδια -Διαδραστικές Προσομοιώσεις	
 <p>yenka YENKA</p>		✓	✓	-	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-	Τοπικά	-Windows XP CPU 1 GHz 256MB RAM -Mac OS X 10.4 CPU 1.25 GHz 256MB RAM	ΟΧΙ	-Μοντέλα Προγραμματισμός μέσω Διαγραμμάτων Ροής	
 <p>Stagecast™ CREATOR STAGECAST CREATOR</p>	✓		✓	-	✓	✓	-	✓	✓	✓	-	-	-	✓	✓	-	Τοπικά Διαδικτυακά	-Windows -Ubuntu Linux -Mac	ΟΧΙ	-Προσομοίωση -Κινούμενα σχέδια -Παιχνίδια	






 <p><b>TYNKER</b> CODING FOR KIDS</p> <p>TYNKER</p>	✓		✓		-	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	-	✓ <sup>8</sup>	✓ <sup>8</sup>	-	Διαδικτυακά	-	ΟΧΙ	<ul style="list-style-type: none"> <li>-Διαδραστικές ιστορίες</li> <li>-Παιχνίδια</li> <li>-Κινούμενες εικόνες</li> <li>-Μουσική</li> <li>-Σχέδια</li> </ul>
 <p>Greenfoot</p> <p>GREENFOOT</p>	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Τοπικά	<ul style="list-style-type: none"> <li>-Windows 7,8,10</li> <li>-Ubuntu Linux</li> <li>-Mac OS X 10.8.3</li> </ul>	ΝΑΙ	<ul style="list-style-type: none"> <li>-Παιχνίδια</li> <li>-Προσομοιώσεις</li> <li>-Προγραμματισμός σε Java</li> </ul>
 <p>CODE AVENGERS</p> <p>CODE AVENGERS</p>	✓		✓	-	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	Διαδικτυακά	-	ΟΧΙ	<ul style="list-style-type: none"> <li>-Παιχνίδια,</li> <li>-Ιστοσελίδες,</li> <li>-Εφαρμογές για κινητά με JavaScript, HTML/ CSS, Python</li> </ul>
 <p>kids RUBY</p> <p>RUBY FOR KIDS</p>	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓	Τοπικά	<ul style="list-style-type: none"> <li>-Windows</li> <li>-Ubuntu Linux</li> <li>-Mac</li> </ul>	ΟΧΙ	Προγραμματισμός στη γλώσσα Ruby	
 <p>codecademy</p> <p>CODE ACADEMY</p>	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Διαδικτυακά	-	ΟΧΙ	<ul style="list-style-type: none"> <li>-Παιχνίδια,</li> <li>-Ιστοσελίδες,</li> <li>-Εφαρμογές για κινητά με Python , PHP , JavaScript, Ruby , HTML/CSS</li> </ul>

 <b>Code School</b> <small>a Pluralsight company</small> <b>CODE SCHOOL</b>	✓			✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Διαδικτυακά	-	ΟΧΙ	-Βάσεις Δεδομένων -Ιστοσελίδες, -Εφαρμογές για κινητά με JavaScript Ruby,HTML/CSS, Git, Objective-C
 <b>CodeHS</b> <b>CODEHS</b>	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Διαδικτυακά	-	ΟΧΙ	-Σχέδια -Κίνηση -Παιχνίδια -Ιστοσελίδες με Python , PHP , JavaScript, HTML/CSS, SQL
 <b>KHAN</b> <small>ACADEMY</small> <b>KHAN ACADEMY</b>	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Διαδικτυακά	-	ΟΧΙ	-Σχέδια -Κινούμενες εικόνες -Παιχνίδια Προσομιώσεις -Ιστοσελίδες με Python , PHP, JavaScript, HTML/CSS,
 <b>treehouse</b> <b>TREEHOUSE</b>	✓			✓	-	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	✓ <sup>8</sup>	Διαδικτυακά	-	ΟΧΙ	-Ανάπτυξη ιστοσελίδων, Android εφαρμογών. Προγραμματισμό σε Ruby, HTML, WordPress.



 A coding adventure around the world ALL CAN CODE	✓		✓		✓	✓	✓	✓	✓ <sup>9</sup>	✓ <sup>9</sup>	✓ <sup>9</sup>	✓ <sup>9</sup>	✓ <sup>9</sup>	-	-	-	-	Διαδικτυακά	-	NAI	-Παιχνίδια -Κίνηση
 CYBERIX3D		✓	✓		✓	✓	✓	✓	-	-	-	✓	-	-	✓	-		Διαδικτυακά	-	OXI	- Παιχνίδια
 GDEVELOP	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	Τοπικά	-Windows -Ubuntu-Linux 64bits μόνο -Mac OS X -iOS/Android	OXI	- Διαδικτυακά παιχνίδια με HTML5
 THE FOOS	✓		✓		✓	✓	✓	✓	-	-	-	✓	-	-	-	-		Διαδικτυακά	-	NAI	- Παιχνίδια
 MAMA		✓	✓		✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓		Τοπικά	-Windows XP / Vista / 7 / 2000 -Ελάχιστη μνήμη: 512MB	OXI	-Παιχνίδια -Κινούμενα σχέδια

 CODERBYTE	✓			✓	v <sup>5</sup>	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	Διαδικτυακά	-	OXI	-Διαδικτυακές εφαρμογές -Βάσεις Δεδομένων. -Προγράμματα σε Ruby, JavaScript, Python,
 GUIDO VAN ROBOT	✓			✓	✓	✓	✓	✓	-	-	-	✓	-	-	-	-	-	Τοπικά	-Windows -Ubuntu Linux -Mac	OXI	-Κίνηση ρομπότ
 PHASER	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓	Τοπικά	-CPU 1.5 GHz, -RAM 1.0 GB, - -64MB κάρτα γραφικών που υποστηρίζει ανάλυση 1028 x 768.	OXI	-Παιχνίδια σε HTML5 για τοπική ή διαδικτυακή χρήση

- 1 Στην έκδοση Scratch 2 ή στην διαδικτυακή έκδοση <https://scratch.mit.edu/> Διαδικασίες [https://wiki.scratch.mit.edu/wiki/Custom\\_Blocks](https://wiki.scratch.mit.edu/wiki/Custom_Blocks) Αναδρομή <https://wiki.scratch.mit.edu/wiki/Recursion>
- 2 Δωρεάν η Βασική έκδοση
- 3 Παρουσιάζει αστάθεια σε αυτά τα λειτουργικά συστήματα [https://edu.kde.org/kturtle/obtain.php?site\\_locale=el](https://edu.kde.org/kturtle/obtain.php?site_locale=el)
- 4 Στην έκδοση StarLogo Nova [http://education.mit.edu/portfolio\\_page/starlogonova/](http://education.mit.edu/portfolio_page/starlogonova/)
- 5 Απαιτείται εγγραφή του χρήστη
- 6 Στο RoboMind Academy <https://www.robomindacademy.com/go/course/robomind/HourOfCode/Getting%20started/0>
- 7 Στην έκδοση AgentCubes <http://www.agentsheets.com/agentcubes/index.html>
- 8 Απαιτείται συνδρομή
- 9 Αναμένεται σύντομα να κυκλοφορήσει <https://www.allcancode.com/web>

Πίνακας 2.2 Παιδαγωγικά Χαρακτηριστικά Προγραμματιστικών Περιβαλλόντων

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ	ΠΑΙΔΑΓΩΓΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ				Ηλικία που Προτείνεται
	Ικανοποίηση στόχων προγράμματος σπουδών	Δυνατότητα μετάβασης σε άλλα προγραμματιστικά περιβάλλοντα	Υποστήριξη από κοινότητα μάθησης	Αλληλεπίδραση	
EASYLOGO		✓		ΜΕΤΡΙΑ	9+
SCRATCH	✓	✓	✓	ΥΨΗΛΗ	8+
SCRATCH JR		✓		ΥΨΗΛΗ	5+
KODU		✓	✓	ΥΨΗΛΗ	10+
GAMEMAKER		✓	✓	ΜΕΤΡΙΑ	11+
MICROWORLDS PRO	✓	✓		ΜΕΤΡΙΑ	10+
BYOB/SNAP	✓	✓		ΥΨΗΛΗ	10+
K-TURTLE	✓	✓	✓	ΜΕΤΡΙΑ	10+
TURTLE ART	✓	✓		ΜΕΤΡΙΑ	11+
OPENSTARLOGO	✓	✓		ΜΕΤΡΙΑ	10+
STARLOGO TNG	✓	✓	✓	ΥΨΗΛΗ	12+
ALICE	✓	✓	✓	ΥΨΗΛΗ	13+
E-TOYS	✓	✓		ΜΕΤΡΙΑ	12+
APP INVENTOR	✓	✓	✓	ΥΨΗΛΗ	11+
CODE.ORG	✓	✓	✓	ΥΨΗΛΗ	4+
LIGHTBOT		✓		ΥΨΗΛΗ	4+
KODABLE	✓	✓	✓	ΥΨΗΛΗ	6+
BLOCKLY	✓	✓	✓	ΥΨΗΛΗ	10+
CODECOMBAT	✓	✓	✓	ΥΨΗΛΗ	10+
ΧΕΛΩΝΟΣΦΑΙΡΑ	✓	✓	✓	ΜΕΤΡΙΑ	10+
GAMESALAD		✓	✓	ΜΕΤΡΙΑ	10+
ROBOMIND		✓	✓	ΜΕΤΡΙΑ	6+
KAREL THE ROBOT	✓	✓		ΜΕΤΡΙΑ	10+
NETLOGO	✓	✓	✓	ΥΨΗΛΗ	12+
PYTHON PYGAME	✓	✓		ΜΕΤΡΙΑ	6+
PYTHON TURTLE	✓	✓		ΜΕΤΡΙΑ	12+

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ	Ικανοποίηση στόχων προγράμματος σπουδών	Δυνατότητα μετάβασης σε άλλα προγραμματιστικά περιβάλλοντα	Υποστήριξη από κοινότητα μάθησης	Αλληλεπίδραση	Ηλικία που Προτείνεται
GAMESTAR MECHANIC		✓	✓	ΥΨΗΛΗ	9+
CRUNCHZILLA CODE MONSTER	✓	✓		ΥΨΗΛΗ	6+
WATERBEAR	✓	✓	✓	ΥΨΗΛΗ	10+
AGENTSHEETS		✓		ΜΕΤΡΙΑ	10+
YENKA	✓	✓		ΜΕΤΡΙΑ	10+
STAGECAST CREATOR		✓		ΜΕΤΡΙΑ	8+
TYNKER	✓	✓	✓	ΥΨΗΛΗ	6+
GREENFOOT	✓	✓	✓	ΥΨΗΛΗ	13+
CODE AVENGERS	✓	✓	✓	ΥΨΗΛΗ	12+
RUBY FOR KIDS	✓	✓		ΥΨΗΛΗ	10+
CODE ACADEMY	✓	✓	✓	ΥΨΗΛΗ	13+
CODE SCHOOL	✓	✓	✓	ΥΨΗΛΗ	13+
CODEHS	✓	✓	✓	ΥΨΗΛΗ	6+
KHAN ACADEMY	✓	✓	✓	ΥΨΗΛΗ	7+
TREEHOUSE	✓	✓	✓	ΥΨΗΛΗ	12+
CODAKID	✓	✓		ΥΨΗΛΗ	7+
PENCIL CODE		✓		ΥΨΗΛΗ	7+
TOONTALK		✓		ΜΕΤΡΙΑ	5+
MERLIN PROGRAMMER FOR KIDS	✓	✓		ΜΕΤΡΙΑ	6+
CODEMONKEY		✓	✓	ΥΨΗΛΗ	9+
BLUEJ	✓	✓	✓	ΜΕΤΡΙΑ	13+
ALL CAN CODE		✓		ΥΨΗΛΗ	8+
CYBERIX3D	✓	✓	✓	ΥΨΗΛΗ	9+
GDEVELOP	✓	✓	✓	ΥΨΗΛΗ	10+
MAMA	✓	✓		ΥΨΗΛΗ	10+
THE FOOS	✓	✓		ΥΨΗΛΗ	4+
CODERBYTE	✓	✓	✓	ΜΕΤΡΙΑ	12+
GUIDO VAN ROBOT	✓	✓		ΜΕΤΡΙΑ	12+

PHASER	✓	✓	✓	ΜΕΤΡΙΑ	13+
--------	---	---	---	--------	-----

## 2.3 ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ

Στο σημείο αυτό κρίνεται απαραίτητη η ανάλυση των τεχνολογικών χαρακτηριστικών των ΠΠΑ που αφορούν την απεικόνιση, τη διεπαφή, τη δυνατότητα ελεύθερης χρήσης, τον τρόπο λειτουργίας και την υποστήριξη της ελληνικής γλώσσας.

Στα παιδαγωγικά χαρακτηριστικά εξετάζεται πόσα από αυτά παρέχουν δυνατότητες υποστήριξης από κοινότητα μάθησης, η αλληλεπίδραση και η ηλικία των μαθητών που προτείνεται. Από το παρακάτω σχήμα προκύπτει ότι από τα 45 (81,8%) από τα 55 ΠΠΑ, είναι δισδιάστατα (2D), τα 8 (14,5%) είναι τρισδιάστατα (3D) και 2 επιτρέπουν και τις δυο απεικονίσεις.

Η διεπαφή που αναφέρεται στον τρόπο ανάπτυξης των προγραμμάτων αφορά συνολικά 29 (52,7%) και με κείμενο 26 (47,3%). Ως προς την ελεύθερη χρήση των ΠΠΑ τα 37 (67,3%) είναι ελεύθερα. Ως προς την υποστήριξη της Ελληνικής γλώσσας τα 19 (34,5%) είναι εξελληνισμένα.

Τέλος, ως προς τον τρόπο λειτουργίας 24 από αυτά χρησιμοποιούνται τοπικά, 23 διαδικτυακά και 8 χρησιμοποιούνται και με τους δυο τρόπους.

Συγκεντρωτικά τα τεχνολογικά χαρακτηριστικά των εξεταζόμενων ΠΠΑ για την απεικόνιση, τη διεπαφή, τον τρόπο λειτουργίας και την υποστήριξη της Ελληνικής γλώσσας συνοψίζονται στο σχήμα 2.1.





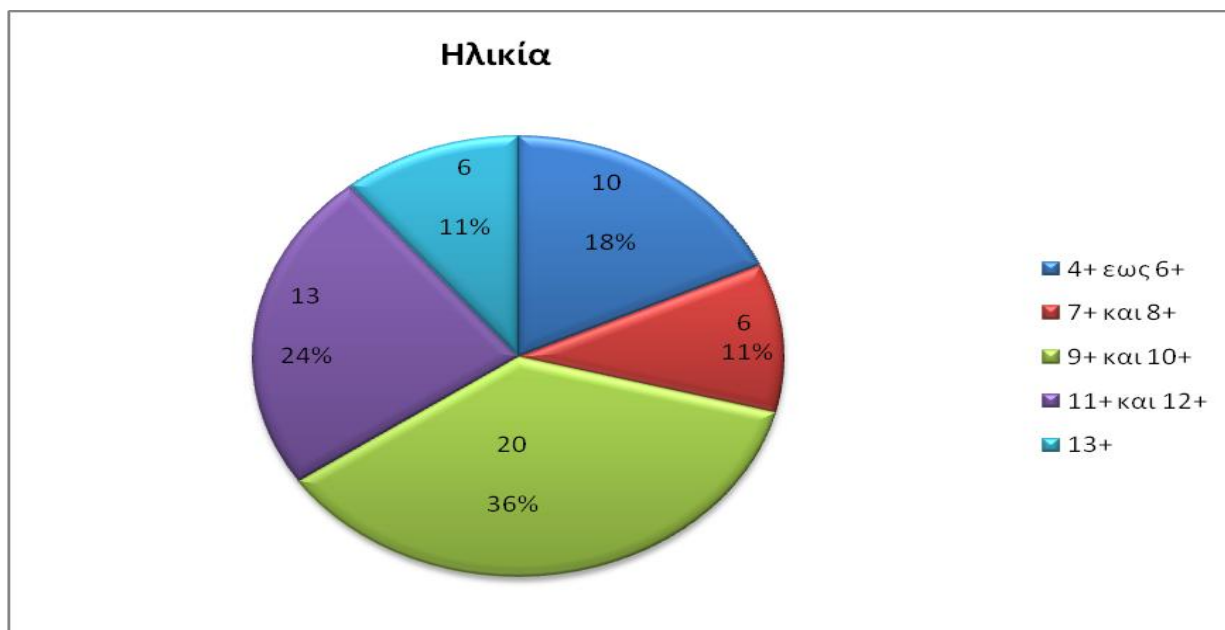
*Σχήμα 2.1 Τεχνολογικά Χαρακτηριστικά ΠΠΑ*

Ως προς τα παιδαγωγικά χαρακτηριστικά 19 (34,5%) παρέχουν δυνατότητες υποστήριξης από κοινότητα μάθησης. Ως προς την αλληλεπίδραση 23 εμφανίζονται να έχουν μέτρια αλληλεπίδραση και 22 υψηλή.

Όσον αφορά στην ηλικία των μαθητών χωρίστηκαν οι ακόλουθες ηλικιακές ομάδες

- ✓ 4+ έως 6+ που είναι οι πιο μικρές ηλικίες και αντιστοιχούν στην προσχολική ή καλύτερα προδημοτική, αλλά και στα παιδιά που ξεκινούν το δημοτικό σχολείο.
- ✓ 7+ και 8+ που είναι οι μικρές ηλικίες μαθητών δημοτικού.
- ✓ 9+ και 10+ που είναι οι ενδιάμεσες ηλικίες μαθητών στο δημοτικό.
- ✓ 11+ και 12+ που είναι οι μεγαλύτερες ηλικίες μαθητών δημοτικού.
- ✓ 13+ είναι οι μαθητές μετά το δημοτικό.

Στο σχήμα 2.2 αποτυπώνεται ο αριθμός των ΠΠΑ που μπορούν χρησιμοποιηθούν ως αφετηρία από τη συγκεκριμένη ηλικιακή ομάδα καθώς και το αντίστοιχο ποσοστό τους.



*Σχήμα 2.2 Πλήθος Περιβαλλόντων ανά Ηλικία*

Με βάση όλα τα παραπάνω στοιχεία επιλέχθηκαν 3 ΠΠΑ διαφορετικών χαρακτηριστικών για να χρησιμοποιηθούν σε μαθητές της έκτης τάξης δημοτικού σχολείου. Η επιλογή αυτή όφειλε να πληροί κάποια βασικά χαρακτηριστικά που τα παραπάνω ΠΠΑ έπρεπε να διαθέτουν, όπως είναι να υποστηρίζουν την Ελληνική γλώσσα, να υπάρχει υποστήριξη από κοινότητα μάθησης, η ανάπτυξη του κώδικα να γίνεται με χρήση πλακιδίων, να ανήκουν στην εξεταζόμενη ηλικιακή ομάδα των μαθητών, η αλληλεπίδραση να είναι υψηλή και να παρέχεται η δυνατότητα ελεύθερης χρήσης του προγραμματιστικού περιβάλλοντος.

Για την απεικόνιση του περιβάλλοντος επιλέχθηκαν δυο ΠΠΑ που υποστηρίζουν τη δισδιάστατη οπτικοποίηση αλλά και ένα που υποστηρίζει την τρισδιάστατη. Επίσης μελετήθηκαν και οι λειτουργικές απαιτήσεις, αφού όλα τα τρισδιάστατα περιβάλλοντα έχουν μεγάλες απαιτήσεις σε υπολογιστική ισχύ και πόρους όπου αρκετά εργαστήρια πληροφορικής στα δημοτικά σχολεία δεν μπορούν να ανταποκριθούν. Έτσι, το τρισδιάστατο περιβάλλον θα έπρεπε να έχει και τις λιγότερες λειτουργικές απαιτήσεις.

Σημαντικό παράγοντα επιλογής αποτέλεσε, επίσης, το να χρησιμοποιηθούν ΠΠΑ κάθε κατηγορίας ή καλύτερα όλης της γκάμας πέραν της απεικόνισης αλλά και του τρόπου λειτουργίας τους, δηλαδή και διαδικτυακά περιβάλλοντα αλλά και τοπικά περιβάλλοντα καθώς επίσης το να παρέχουν τη δυνατότητα χρήσης τοπικά άλλα και διαδικτυακά. Γι' αυτό επιλέχτηκε ένα καθαρά διαδικτυακό περιβάλλον, ένα καθαρά τοπικό περιβάλλον και ένα που μπορεί να αξιοποιηθεί και με

τους δυο τρόπους. Επιπρόσθετα, οι προγραμματιστικές έννοιες που εμπεριέχει το κάθε ΠΠΑ να συμβαδίζουν με αυτές που αποτυπώνονται στο Πρόγραμμα Σπουδών για τα δημοτικά και που έχουν αναφερθεί εκτενέστερα σε προηγούμενη ενότητα. Μια ακόμα σημαντική δυνατότητα είναι να μπορεί να χρησιμοποιηθεί το ΠΠΑ ελεύθερα. Έτσι λοιπόν χρησιμοποιήθηκε το δισδιάστατο διαδικτυακό περιβάλλον Code.org, το δισδιάστατο τοπικό και διαδικτυακό περιβάλλον Scratch και το τρισδιάστατο τοπικό περιβάλλον το Starlogo TNG.

## 2.4 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ

### 2.4.1 CODE.ORG

Το Code.org είναι ένας μη κερδοσκοπικός οργανισμός και ένας ευρέως διαδεδομένος διαδικτυακός τόπος εκμάθησης προγραμματισμού που δημιουργήθηκε και καθοδηγείται από τον Hadi Partovi (Partovi, 2015). Στοχεύει στην ενθάρρυνση των ανθρώπων και ιδιαίτερα των μαθητών πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης για την απόκτηση βασικών γνώσεων γύρω από την επιστήμη της πληροφορικής (Partovi, 2015).

Ο διαδικτυακός τόπος περιλαμβάνει δωρεάν μαθήματα προγραμματισμού και πρωτοβουλίες που στοχεύουν στα σχολεία σε μια προσπάθεια να συμπεριλάβουν περισσότερο την επιστήμη της πληροφορικής στα προγράμματα σπουδών τους.

Τον Δεκέμβριο του 2013 προώθησε την Ώρα του Κώδικα (Hour of Code), μια παγκόσμια δράση που διήρκησε μια εβδομάδα από 9 έως 15 Δεκεμβρίου 2013 (Partovi, 2015).

Τα τρία κύρια χαρακτηριστικά του Code.org είναι (Partovi, 2015):

- Προώθηση της πληροφορικής με την Ώρα του Κώδικα (Hour of Code) παγκοσμίως.
- Προσφορά προγραμμάτων σπουδών, εκπαιδευτικού υλικού και του εκπαιδευτικού προγραμματιστικού περιβάλλοντος.
- Επαγγελματική ανάπτυξη των εκπαιδευτικών και των σχολείων.

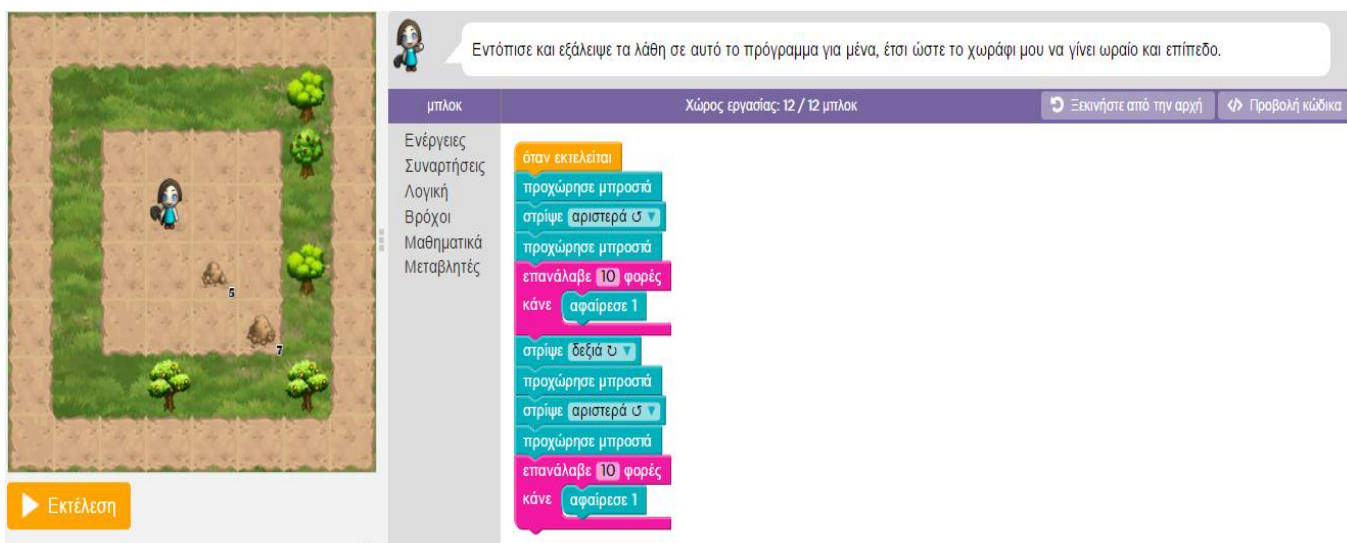
Τα αποτελέσματα της διδασκαλίας του προγραμματισμού με το code.org εξακολουθούν να παραμένουν ασαφή και λιγοστά στη βιβλιογραφία (Kalelioglu, 2015). Η «Ώρα του Κώδικα» παρέχει αυτοκαθοδηγούμενες βιωματικές δραστηριότητες διαθέσιμες σε 34 γλώσσες δομημένες ως παιχνίδια προσφέροντας βοήθεια μέσω βιντεοδιαλέξεων, πριν από την έναρξη όποιας προγραμματιστικής πρόκλησης που οι μαθητές καλούνται να αντιμετωπίσουν (Kalelioglu, 2015;

Briggs, Bau, Meeks, & Yongpradit, 2015). Σε αυτές τις βιντεοδιαλέξεις συμμετέχουν προσωπικότητες διάσημες στο χώρο της Πληροφορικής όπως ο Bill Gates, ο Jack Dorsey, ο Mark Zuckerberg, η Ruchi Sanghvi, το χώρο του αθλητισμού όπως ο διάσημος καλαθοσφαιριστής Chris Bosch, το χώρο της πολιτικής όπως ο Bill Clinton, ο πρόεδρος της Αμερικής Barack Obama και πολλοί άλλοι (Partoni, 2015; Wilson, 2013; Kalelioglu, 2015).

Όλοι οι παραπάνω ενθαρρύνουν, εμπνέουν και παροτρύνουν τους μαθητές και όποιον άλλο ενδιαφερόμενο να εξετάσουν την επιστήμη της πληροφορικής αλλά και την κάθε προγραμματιστική έννοια (Wilson, 2013).

Με τη χρήση του Code.org οι μαθητές μπορούν να μάθουν τη λογική των εντολών, του αλγορίθμου, τις δομές επανάληψης και επιλογής, τις μεταβλητές, τις διαδικασίες (και άλλες προγραμματιστικές έννοιες (όπως αναλυτικά περιγράφονται στον συγκριτικό πίνακα 2.1), χρησιμοποιώντας τη λογική των πλακιδίων (Block-based, Blocky) με σύρσιμο και εναπόθεση πλακιδίων εντολών (drag and drop) (Kalelioglu, 2015; Brown, Mönig, Bau, & Weintrop, 2016).

Το περιβάλλον προγραμματισμού του code.org μιας δραστηριότητας παρουσιάζεται στο σχήμα 2.3. Χωρίζεται από αριστερά προς τα δεξιά στην περιοχή όπου οπτικοποιείται ο κώδικας. Πατώντας το κουμπί Εκτέλεση αρχίζει το πρόγραμμα να εκτελείται. Στην κατηγορία «μπλοκ» παρουσιάζονται τα πλακίδια εντολών που μπορούν να χρησιμοποιηθούν και που είναι διαθέσιμα. Σε πολλές δραστηριότητες τα πλακίδια αυτά παρουσιάζονται και ομαδοποιημένα. Στο χώρο εργασίας τοποθετούνται τα πλακίδια των εντολών για την επίλυση όποιου προβλήματος – παιχνιδιού – κατάστασης.



Σχήμα 2.3 Δραστηριότητα από το περιβάλλον του Code.org

Οι δραστηριότητες είναι κατηγοριοποιημένες ανά ηλικία, όπως φαίνεται και στο σχήμα 2.4, όπου εμφανίζεται ένα στιγμιότυπο επιλογών για μαθητές πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης. Σημαντικό είναι το γεγονός ότι μπορούν να το χρησιμοποιήσουν νεαροί προγραμματιστές και να εμπλακούν με αυτόν χωρίς ιδιαίτερες ικανότητες ανάγνωσης.



Σχήμα 2.4 Κατηγοριοποιημένες ενότητες δραστηριοτήτων ανά ηλικία στο Code.org

Κατά την επιλογή κάποιου μαθήματος μια σειρά από ομαδοποιημένες ανά κατηγορία δραστηριότητες είναι διαθέσιμες. Το σύστημα καταγράφει την πορεία του κάθε μαθητή-ων σύμφωνα με τις:

- Δραστηριότητες που έχει καταφέρει να ολοκληρώσει πλήρως.
- Δραστηριότητες που έχει ολοκληρώσει αλλά με περισσότερες γραμμές κώδικα απ' ότι χρειαζόταν.
- Δραστηριότητες που είναι σε εξέλιξη.
- Δραστηριότητες που δεν έχουν ξεκινήσει.

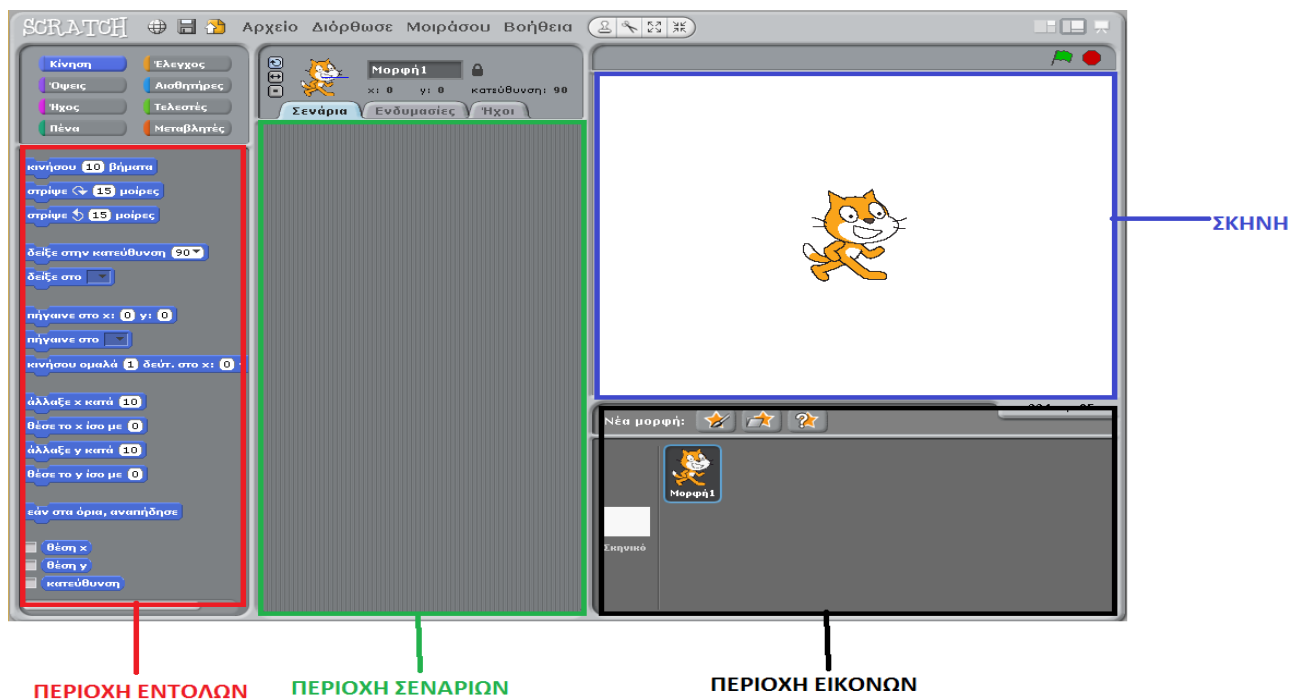
Ένα στιγμιότυπο αυτών των δραστηριοτήτων από την συμμετοχή μιας τάξης εμφανίζεται στο σχήμα 2.5.



είναι δισδιάστατης απεικόνισης και ο προγραμματισμός λαμβάνει χώρα με την τεχνική συρε και άφησε (drag and drop) πλακιδίων εντολών.

Μπορεί να χρησιμοποιηθεί τοπικά ή διαδικτυακά <https://scratch.mit.edu/>, είναι μεταφρασμένο σε περισσότερες από 40 γλώσσες συμπεριλαμβανομένου και των Ελληνικών και χρησιμοποιείται σε 150 χώρες. Το περιβάλλον υποδιαιρείται στις εξής περιοχές όπως εμφανίζονται και στο σχήμα 2.6 (Meerbaum-Salant, Armoni, & Ben-Ari, 2011):

- Η περιοχή των εντολών όπου είναι κατηγοριοποιημένες και ομαδοποιημένες ανά χρώμα σχετικά με την κίνηση, τις όψεις, τον ήχο, την πένα, τον έλεγχο, τους αισθητήρες, τους τελεστές και τις μεταβλητές. Πατώντας σε κάθε κατηγορία, εμφανίζονται από κάτω οι αντίστοιχες διαθέσιμες εντολές. Στην διαδικτυακή έκδοση υπάρχει και μια πρόσθετη κατηγορία η «Άλλες εντολές» όπου εκεί ομαδοποιούνται κομμάτια εντολών που μπορούν να εκτελεστούν όποτε κληθούν από το πρόγραμμα, όπως γίνεται και στις διαδικασίες και όπως υποστηρίζει και ο επίσημος δικτυακός τόπος του scratch [https://wiki.scratch.mit.edu/wiki/Custom Blocks](https://wiki.scratch.mit.edu/wiki/Custom_Blocks).
- Η περιοχή σεναρίων όπου γίνεται η τοποθέτηση των πλακιδίων εντολών για την ανάπτυξη του προγράμματος.
- Η περιοχή της σκηνής που είναι η κεντρική περιοχή του περιβάλλοντος. Εκεί πραγματοποιείται η οπτικοποίηση των εντολών της περιοχής των σεναρίων. Η Σκηνή έχει μήκος 480 μονάδες και ύψος 360 μονάδες. Το κέντρο της Σκηνής έχει τις συντεταγμένες x:0 και y:0.
- Η περιοχή των εικόνων όπου γίνεται ο ορισμός, η επιλογή έτοιμων εικόνων καθώς και η σχεδίαση νέων μορφών – αντικειμένων – χαρακτήρων.



Σχήμα 2.6 Το περιβάλλον Scratch και οι περιοχές του.

Οι μαθητές μπορούν, χωρίς να υπάρχει ο κίνδυνος συντακτικών λαθών και ανάγκη για απομνημόνευση εντολών, να δημιουργήσουν έργα μέσα από μια ακολουθία από πλακίδια τροποποιώντας την σειρά τους να φτάσουν στο επιθυμητό αποτέλεσμα. Με την χρήση του περιβάλλοντος μπορούν να κατασκευάσουν διαδραστικές ιστορίες, παιχνίδια, μουσική, κινούμενες εικόνες και σχέδια και να εμπλακούν με έννοιες του προγραμματισμού όπως η δομή ακολουθίας, επιλογής, επανάληψης, τη μεταβλητή, τις διαδικασίες, την παράλληλη εκτέλεση, τον προγραμματισμό γεγονότων, την χρήση τελεστών (αριθμητικών, λογικών και συγκριτικών) καθώς και την εισαγωγή σε έννοιες του αντικειμενοστραφή προγραμματισμού (Maloney et al. 2004; Wu et al. 2010; Malan & Leitner 2007).

Σημαντικό χαρακτηριστικό του περιβάλλοντος είναι και η υποστηρικτική κοινότητα μάθησης με δυνατότητες διαμοιρασμού των έργων διαδικτυακά. Το μότο του Scratch είναι «Φαντάσου - Φτιάξε – Μοιράσου». Ειδικά η τρίτη παράμετρος αποτελεί μεγάλη παιδαγωγική αξία για τους κατασκευαστές του. Ο κάθε χρήστης έχει δυνατότητα δημοσίευσης του έργου του αλλά και της μεταφόρτωσης, χρήσης, τροποποίησης και επέκτασης έργων άλλων χρηστών (Meerbaum-Salant, Armoni, & Ben-Ari, 2011). Οι εγγεγραμμένοι χρήστες ανέρχονται στα 15 εκατομμύρια και τα έργα που έχουν διαμοιραστεί ξεπερνούν τα 18 εκατομμύρια.



### 2.4.3 STARLOGO TNG

Η StarLogo TNG “The Next Generation” είναι ένα εργαλείο διδασκαλίας προγραμματισμού που αναπτύχθηκε για να καλύψει τις ανάγκες ενός προγράμματος εκπαίδευσης δασκάλων και καθηγητών από το MIT (Massachusetts Institute of Technology) κάτω από την εποπτεία των Eric Klopfer και Mitchel Resnick (Klopfer & Begel, 2005).

Πρόγονοι αυτού του περιβάλλοντος είναι τα περιβάλλοντα openStarLogo όπου η ανάπτυξη του προγράμματος γίνεται με κείμενο, το LogoBlocks που χρησιμοποιούνται πλακίδια εντολών για την ανάπτυξη του προγράμματος σε ένα δισδιάστατο περιβάλλον. Εξελίχτηκε στην υπό εξεταζόμενη έκδοση όπου η ανάπτυξη γίνεται με σύρσιμο και εναπόθεση πλακιδίων εντολών (drag and drop) σε ένα όμως τρισδιάστατο περιβάλλον με τη χρήση OpenGL – πρότυπο για τον καθορισμό γραφικών (*Open Graphics Library*). Είναι γραμμένο σε γλώσσα Java σε αντίθεση των υπολοίπων που ήταν κατασκευασμένα από την γλώσσα C (Klopfer & Yoon, 2005; Klopfer, Wang, McCaffrey & Wendel, 2006).

Οι βασικοί στόχοι για την δημιουργία αυτού του περιβάλλοντος ήταν (McCaffrey, 2006) :

- Να μειώσει τα εμπόδια για την εισαγωγή στον προγραμματισμό.
- Να προσελκύσει περισσότερους νέους στον προγραμματισμό μέσω εργαλείων που διευκολύνουν την κατασκευή παιχνιδιών .
- Να δημιουργήσουν συναρπαστικούς 3D κόσμους μέσω πλούσιων μοντέλων και προσομοιώσεων.

Το περιβάλλον παρουσιάζεται στο σχήμα 2.7. Η περιοχή των εντολών υποδιαιρείται σε τρεις υποομάδες τη Factory που είναι κατηγοριοποιημένες και ομαδοποιημένες ανά χρώμα σχετικά με την εγκατάσταση και εκτέλεση, την κίνηση, τη λογική, τους ελέγχους, το τερν, τα ίχνη, το κείμενο, τα μαθηματικά, τους άλλους πράκτορες, τις διαδικασίες, τις μεταβλητές, τη λίστα, τα χρώματα, τα σχήματα και τους ήχους. Οι άλλες δυο υποομάδες είναι η My Blocks και η Subsets οι οποίες ανανεώνονται και εμπλουτίζονται με πλακίδια εντολών κατά την διάρκεια δημιουργίας του προγράμματος από τον χρήστη κατά την προσθήκη πρακτόρων, την κατασκευή διαδικασιών και τη δημιουργία μεταβλητών.

Η περιοχή σεναρίων όπου γίνεται η τοποθέτηση και η συναρμολόγηση των διαφόρων πλακιδίων εντολών για την ανάπτυξη του προγράμματος. Η περιοχή αυτή χωρίζεται σε πέντε υπό-περιοχές. Α)

Στην περιοχή του κάθε πράκτορα-χαρακτήρα που υπάρχει ή έχει προστεθεί στον κόσμο και αφορούν τον προγραμματισμό της συμπεριφοράς του. Β) Στην περιοχή Όλοι που σχετίζεται με τον προγραμματισμό όλων των πρακτόρων του περιβάλλοντος, όταν υπάρχουν κοινές συμπεριφορές. Γ) Στην περιοχή Εγκατάσταση που σχετίζεται με την αρχικοποίηση και εγκατάσταση του κατασκευαζόμενου κόσμου. Δ) Στην περιοχή Συγκρούσεις προγραμματίζονται σενάρια κατά την επαφή-σύγκρουση των πρακτόρων μεταξύ τους και Ε) στην περιοχή Εκτέλεση που αφορά στον τρόπο που θα εκτελούνται οι αναφερόμενες περιοχές.

Στην περιοχή της Σκηνής που εκεί γίνεται η οπτικοποίηση της περιοχής σεναρίων σε μια τρισδιάστατη απεικόνιση που εκεί λαμβάνει χώρα η εκτέλεση και η παρατήρηση του κόσμου που έχει δημιουργηθεί.

Στην περιοχή εμφάνισης διαφόρων λειτουργιών σχετικών με την αρχικοποίησης την εκτέλεση και εγκατάσταση του προγράμματος που έχουν δημιουργηθεί κατά την ανάπτυξή του. Στην εικόνα εμφανίζονται αυτές οι λειτουργίες με την μορφή των κουμπιών με τα ονόματα εγκατάσταση, τρέξε μια φορά, για πάντα κτλ. Με το πάτημα κάποιου κουμπιού εκτελούνται οι αντίστοιχες εντολές που έχουν δοθεί στην περιοχή σεναρίων.




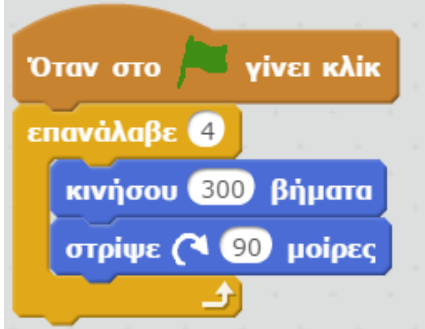

Σχήμα 2.7 Το περιβάλλον προγραμματισμού StarLogo TNG

Το περιβάλλον λειτουργεί τοπικά στον υπολογιστή και μπορεί να μεταφορτωθεί από την ιστοσελίδα [http://education.mit.edu/portfolio\\_page/starlogo-tng/](http://education.mit.edu/portfolio_page/starlogo-tng/). Οι γλώσσες που είναι διαθέσιμες είναι τα Αγγλικά, Ισπανικά, Πορτογαλικά και τα Ελληνικά. Η διαδικτυακή έκδοση του StarLogo TNG είναι η StarLogo Nova.

## 2.5 ΕΠΙΣΚΟΠΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΕΝΝΟΙΩΝ

Σε αυτό το σημείο κρίνεται απαραίτητο να παρουσιαστεί η ανάπτυξη κάποιων προγραμματιστικών εννοιών και από τα τρία ΠΠΑ που αφορούν την επανάληψη, την επιλογή, τις διαδικασίες, τις μεταβλητές και τα γεγονότα. Αυτό γίνεται, για να αποτυπωθούν περισσότερο οι ομοιότητες αλλά και κάποιες διαφορές αυτών των τριών κατά την εκτέλεση παρόμοιων-παρεμφερών προγραμματιστικών ενεργειών όπως παρουσιάζονται αναλυτικά στον πίνακα 2.3.

Πίνακας 2.3 Ανάπτυξη Προγραμματιστικών Εννοιών από τα αντίστοιχα ΠΠΑ

Επανάληψη		
Code.org	Scratch	Starlogo TNG
		
Επιλογή		
Code.org	Scratch	Starlogo TNG

```

όταν εκτελείται
  επανάλαβε μέχρις ότου
    κάνε
      εάν υπάρχει μονοπάτι αριστερά
      κάνε
        στρίψε αριστερά
      αλλιώς
        προχώρησε μπροστά
  
```

```

Όταν στο γίνε κλικ
  επανάλαβε ώσπου
    εάν αγγίζει το χρώμα
      τότε
        στρίψε 90 μοίρες
      αλλιώς
        κινήσου 10 βήματα
  
```

```

Turtles
για πάντα
  έλεγχος
    αν αλλιώς
      τότε
        αριστερά 90 μοίρες
      αλλιώς
        μπροστά βήματα 1
    = μοβ
  
```

### Γεγονότα

Code.org

Scratch

Starlogo TNG

```

όταν πατηθεί το αριστερό βέλος
  προχώρησε τον ηθοποιό 1 αριστερά

όταν πατηθεί το δεξί βέλος
  προχώρησε τον ηθοποιό 1 δεξιά

όταν πατηθεί το επάνω βέλος
  προχώρησε τον ηθοποιό 1 επάνω

όταν πατηθεί το κάτω βέλος
  προχώρησε τον ηθοποιό 1 κάτω
  
```

```

Όταν το πλήκτρο Πάνω βέλος πατηθεί
  στρίψε προς την κατεύθυνση των 0 μοιρών
  κινήσου 10 βήματα

Όταν το πλήκτρο κάτω βέλος πατηθεί
  στρίψε προς την κατεύθυνση των 180 μοιρών
  κινήσου 10 βήματα

Όταν το πλήκτρο δεξί βέλος πατηθεί
  στρίψε προς την κατεύθυνση των 90 μοιρών
  κινήσου 10 βήματα

Όταν το πλήκτρο αριστερό βέλος πατηθεί
  στρίψε προς την κατεύθυνση των 270 μοιρών
  κινήσου 10 βήματα
  
```

```

οχημα
για πάντα
  έλεγχος
    αν τότε
      πλήκτρο άνω βέλος?
      μπροστά βήματα 10
    έλεγχος
      αν τότε
        πλήκτρο κάτω βέλος?
        πίσω βήματα 10
    έλεγχος
      αν τότε
        πλήκτρο δεξί βέλος?
        δεξιά μοίρες 15
    έλεγχος
      αν τότε
        πλήκτρο αριστερό βέλος?
        αριστερά μοίρες 15
  
```

### Μεταβλητές

Code.org

Scratch

Starlogo TNG

```

όταν εκτελείται
  Γρίφος points σε 0
  όταν ο ηθοποιός 2 ακουμπάει τον ηθοποιό 1
    παίξε ήχο κτυπήματος
  Γρίφος points σε
    points + 1
  
```

```

Όταν στο γίνε κλικ
  άρισε το σκορ σε 0
  για πάντα
    εάν αγγίζει το Magic Wand
      τότε
        στρίψε προς την κατεύθυνση των
          180 - κατεύθυνση μοιρών
        άλλαξε το σκορ κατά 1
  
```

```

Εκτέλεση
SKOP
θέσε σκορ 0
Turtles
για πάντα
  έλεγχος
    αν τότε
      χρώμα μπαλιώματος μπροστά = πράσινο
      Εκτέλεση
        αυξ SKOP 1
      θέσε κατεύθυνση
        180 - κατεύθυνση
  
```

## Διαδικασίες

Code.org

Scratch

Starlogo TNG

```
όταν εκτελεστεί
επανάλαβε 3 φορές
  κάνε Διαδικασία Τετράγωνο επεξεργασίας
  κάνε άλμα προς τα εμπρός κατά 120 εικονοστοιχεία
```

```
Διαδικασία Τετράγωνο επεξεργασίας
επανάλαβε 4 φορές
  προχώρησε προς τα εμπρός κατά 100 εικονοστοιχεία
  στρίψε δεξιά κατά 90 μοίρες
```

```
Όταν στο σημείο γίνεται κλικ
επανάλαβε 3
  τετράγωνο
  πήγαινε στη θέση του random position

ορισμός του τετράγωνο
επανάλαβε 4
  κατέβασε την πένα
  κινήσου 100 βήματα
  στρίψε 90 μοίρες
  σήκωσε την πένα
```

```
Mario
τρέξε μία φορά
πήγαινε στην αρχή
Όλοι
  Διαδικασία TETRAGWNO
  μπροστά βήματα 10

Όλοι
  Διαδικασία TETRAGWNO
  φορές 4
  επανάλαβε
    στυλό κάτω
    μπροστά βήματα 5
    δεξιά μοίρες 90
```

### 3 ΜΕΘΟΔΟΛΟΓΙΑ

Στόχος της εργασίας είναι να καταδείξει το βαθμό κατανόησης των μαθητών για τις προγραμματιστικές έννοιες, καθώς και τα συχνά λάθη που εντοπίζονται σε κάθε στάδιο εκμάθησής τους. Επιπρόσθετα να διερευνήσει αν διαφοροποιούνται τα μαθησιακά αποτελέσματα, όπως και η επίδοση μεταξύ των δύο φύλων κατά τη χρήση διαφορετικών Προγραμματιστικών Περιβαλλόντων για Αρχαρίους. Τέλος, στοχεύει να αποτυπώσει την αξιολόγηση των ΠΠΑ από τους ίδιους τους μαθητές.

#### 3.1 ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ

Τα ερευνητικά ερωτήματα που καλείται να απαντήσει η παρούσα μελέτη είναι να διερευνηθεί:

1. Ποιος ο βαθμός κατανόησης των μαθητών για τις προγραμματιστικές έννοιες;
2. Ποιες οι εναλλακτικές ιδέες – συχνά λάθη που εντοπίζονται από τους μαθητές στο στάδιο της εκμάθησης της κάθε προγραμματιστικής έννοιας;
3. Διαφοροποιούνται τα μαθησιακά αποτελέσματα κατά τη χρήση διαφορετικών ΠΠΑ;
4. Υπάρχει διαφοροποίηση στην επίδοση μεταξύ των δυο φύλων κατά τη χρήση αυτών των ΠΠΑ;
5. Πώς αξιολογούν οι μαθητές τα ΠΠΑ;

#### 3.2 ΔΕΙΓΜΑ

Η μελέτη πραγματοποιήθηκε στο 11<sup>ο</sup> Δημοτικό Σχολείο της πόλης των Ιωαννίνων. Ο διευθυντής και οι μαθητές ενημερώθηκαν για την έρευνα από την εκπαιδευτικό της Πληροφορικής. Η έρευνα πραγματοποιήθηκε στο εργαστήριο Πληροφορικής του σχολείου το σχολικό έτος 2016-2017 (Σεπτέμβριος – Μάρτιος). Οι συμμετέχοντες ήταν 37 μαθητές δύο τμημάτων της ΣΤ' τάξης, ηλικίας 11 και 12 ετών. Αυτές οι δύο ομάδες των δύο τμημάτων ήταν περίπου ισοπληθείς και ως προς το σύνολο των μαθητών, άλλα και ως προς το φύλο των μαθητών. Ως μέθοδος επιλογής του δείγματος εφαρμόστηκε η βολική δειγματοληψία καθώς ο ερευνητής ήταν και ο εκπαιδευτικός της τάξης που έγινε η μελέτη.

Το τμήμα ΣΤ1 είχε 18 μαθητές στο σύνολο όπου 10 ήταν αγόρια και 8 κορίτσια ενώ το δεύτερο τμήμα ΣΤ2 είχε 19 μαθητές όπου 10 ήταν αγόρια και 9 ήταν κορίτσια.

Το δείγμα των 37 μαθητών της ΣΤ' τάξης που επιλέχθηκε δεν είχε πρότερες γνώσεις σχετικά με τον προγραμματισμό ούτε και κάποιος από αυτούς έχει ξαναέρθει σε επαφή με κάποιο ΠΠΑ. Η παρέμβαση ήταν ίδια και στα δυο τμήματα.

Στο πρώτο μέρος της έρευνας οι μαθητές και των δυο τμημάτων μετά την ολοκλήρωση του κάθε μαθήματος για την περιγραφή - εκμάθηση - οικοδόμηση της κάθε προγραμματιστικής έννοιας χρησιμοποιούσαν το ΠΠΑ Code.org, όπου καλούνταν να ακολουθήσουν συγκεκριμένες δραστηριότητες ανοιχτού ή κλειστού τύπου που αφορούσαν τη συγκεκριμένη έννοια. Στο δεύτερο μέρος, μετά την ολοκλήρωση αυτής της διαδικασίας, στο κάθε τμήμα έγινε μετάβαση σε διαφορετικό ΠΠΑ. Ένα τμήμα χρησιμοποίησε το Scratch και το άλλο τμήμα χρησιμοποίησε το Starlogo TNG.

### **3.3 ΕΡΕΥΝΗΤΙΚΗ ΜΕΘΟΔΟΣ**

Η μέθοδος που χρησιμοποιείται στο πλαίσιο της παρούσας έρευνας είναι η μελέτη περίπτωσης που αποτελεί μια εμπειρική έρευνα μικρής κλίμακας και πραγματοποιείται πριν την εφαρμογή μίας έρευνας μεγάλης κλίμακας. Μπορεί να αποτελέσει τη βάση για μία πλήρη και μεγάλης έκτασης έρευνα, αναδεικνύοντας σφάλματα, προβλήματα και αποτυχίες που μπορούν να αποφευχθούν στην κύρια έρευνα. Πραγματοποιείται με λιγότερους πόρους, χρόνο και κόστος και στοχεύει στο να μειώσει την αβεβαιότητα των παραμέτρων της κύριας έρευνας(Yin, 2003). Ο συγκεκριμένος τύπος έρευνας επιλέχθηκε διότι τόσο οι πόροι όσο και το μέγεθος του δείγματος ήταν περιορισμένα.

### **3.4 ΕΡΕΥΝΗΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ**

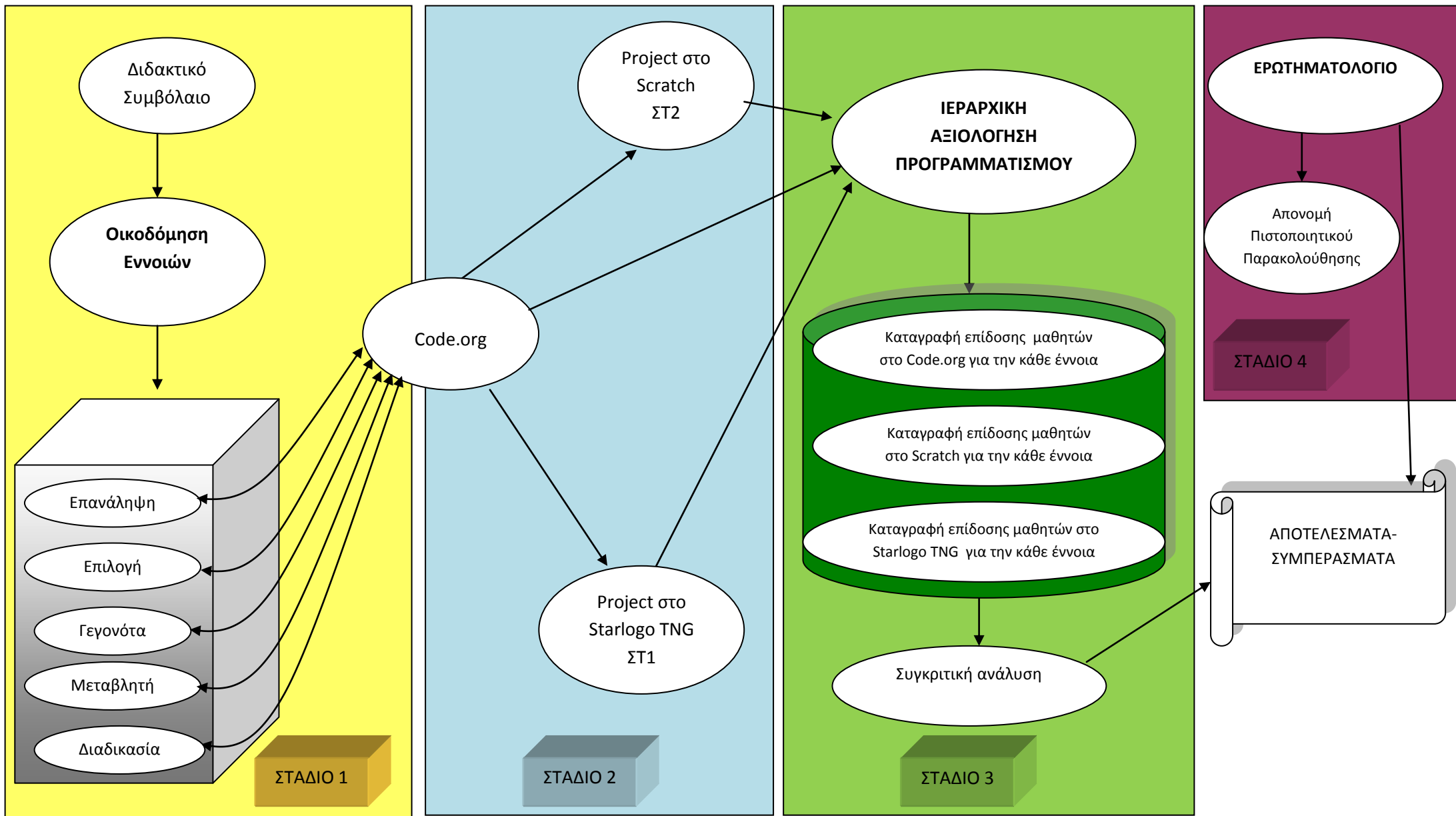
Τα κύρια στάδια της έρευνας είναι:

1. Η οικοδόμηση της γνώσης ποικίλων προγραμματιστικών εννοιών, όπως είναι η επανάληψη, η επιλογή, η διαδικασία, η μεταβλητή και τα γεγονότα με την υποστήριξη του ΠΠΑ Code.org.
2. Η αξιοποίηση των ΠΠΑ (Scratch και Starlogo TNG) για την κατασκευή ολοκληρωμένων έργων που βασίζονται στις παραπάνω έννοιες.

3. Η αξιολόγηση και η καταγραφή της επίδοσης των μαθητών σε αυτές τις έννοιες κατά τη χρήση διαφορετικών ΠΠΑ.
4. Η αξιολόγηση των ΠΠΑ, που επιλέχτηκαν και χρησιμοποιήθηκαν από τους μαθητές όπως προκύπτει από το ερωτηματολόγιο.

Αυτοί οι άξονες-στάδια εικονογραφούνται στο σχήμα 3.1.





Σχήμα 3.1 Στάδια της έρευνας

### 3.4.1 ΟΙΚΟΔΟΜΗΣΗ ΕΝΝΟΙΩΝ

Στο πρώτο στάδιο εφαρμόστηκε η εποικοδομητική προσέγγιση για τη διδασκαλία προγραμματιστικών εννοιών στην οποία κυρίαρχο ρόλο παίζουν οι ιδέες των μαθητών. Στην προσέγγιση αυτή δίνεται έμφαση στις εποικοδομητικές και συνεργατικές διαδικασίες, στη δημιουργικότητα και στην ανοικτή σκέψη.

Σύμφωνα με τους Μικρόπουλο & Μπέλλου (2005), οι Τεχνολογίες της Πληροφορίας και των Επικοινωνιών (ΤΠΕ) και τα ΠΠΑ ως γνωστικά εργαλεία προωθούν την εποικοδομητική προσέγγιση και συνεισφέρουν στην οικοδόμηση της γνώσης γιατί υποστηρίζουν:

- ο την κατασκευή γνώσης μέσω της αναπαράστασης των ιδεών, της οργάνωσης βάσεων γνώσης των μαθητών και της δημιουργίας ενός πλαισίου διατεταγμένων κατηγοριών ανάλυσης και κατανόησης δεδομένων
- ο την αναζήτηση και διερεύνηση για σύγκριση καταστάσεων, προσεγγίσεων, εκδοχών
- ο τη μάθηση μέσω ενεργειών παρέχοντας ένα ελεγχόμενο περιβάλλον εργασίας
- ο τη γνωστική σύγκρουση που προκύπτει από τη σύγκριση αιτίων – αποτελεσμάτων
- ο τη μάθηση ως απόρροια αναστοχασμού, που βοηθά το μαθητή να διατυπώσει με σαφήνεια και να αναπαραστήσει τις γνώσεις του.

Ο ρόλος του εκπαιδευτικού είναι καθοδηγητικός, διαμεσολαβητικός και διευκολυντικός. Αξιοποιεί τις δυνατότητες των ΤΠΕ και πιο συγκεκριμένα των ΠΠΑ για να σχεδιάσει, να οργανώσει, να επεξεργαστεί και να αναλύσει τα δεδομένα του θέματος που εξετάζει μαζί με τους μαθητές του, στο πλαίσιο μια ενεργητικής και ομαδοσυνεργατικής διδακτικής διαδικασίας. Στόχος του η κατανόηση της κάθε εξεταζόμενης κατάστασης-έννοιας. Η επικοινωνία εκπαιδευτικού-μαθητών δεν εξυπηρετεί τη «μεταφορά» γνώσεων μέσω της τεχνολογίας από τον εκπαιδευτικό στο μαθητή, αλλά τη γνωστική επεξεργασία μέσω των δυνατοτήτων που η ίδια η τεχνολογία παρέχει στην ανάλυση και την κατασκευή νέων εννοιών (Μικρόπουλος, 2011).

Για να επιτευχθούν όλα τα παραπάνω απαραίτητη προϋπόθεση ήταν να δημιουργηθεί το διδακτικό συμβόλαιο με τον τρόπο που προτείνεται από την Μπέλλου (2006). Το διδακτικό συμβόλαιο είναι το σύνολο των συμπεριφορών του διδάσκοντος που αναμένονται από το μαθητή και το σύνολο των συμπεριφορών που αναμένονται από τον διδάσκοντα. Είναι δηλαδή το σύνολο των κανόνων που προσδιορίζουν, εν μέρει ρητά αλλά κυρίως υπόρρητα, αυτά που ο συμμετέχων στη διδακτική σχέση θα διαχειρίζεται και για τα οποία θα είναι με τον έναν ή με τον άλλο τρόπο υπόλογος απέναντι στον άλλο (Μπέλλου, 2006).

Στους μαθητές και των δυο τμημάτων πραγματοποιήθηκαν εισαγωγικά μαθήματα αναφορικά με τον αλγόριθμο, το πρόγραμμα και τη δομή ακολουθίας. Στη συνέχεια πραγματοποιήθηκε μια σειρά μαθημάτων για την οικοδόμηση της κάθε εξεταζόμενης προγραμματιστικής έννοιας με την υποστήριξη του Code.org. Για την κάθε εξεταζόμενη έννοια χρησιμοποιήθηκαν αντίστοιχες δραστηριότητες από αυτό το περιβάλλον. Ο ρόλος των μαθητών ήταν ενεργητικός και εξερευνητικός καθ' όλη την διάρκεια της σειράς των ενοτήτων. Η σειρά και η χρονική διάρκεια αυτών των ενοτήτων παρουσιάζεται στον πίνακα 3.1.

*Πίνακας 3.1 Ακολουθία διδακτικών ενοτήτων*

Σειρά Μαθημάτων	Προγραμματιστικές Έννοιες	Χρονική Διάρκεια (διδασκτικές ώρες)
1	Επανάληψη	3
2	Επιλογή	3
3	Γεγονότα	3
4	Μεταβλητές	3
5	Διαδικασίες	3

### 3.4.2 ΑΞΙΟΠΟΙΗΣΗ ΤΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΓΙΑ ΑΡΧΑΡΙΟΥΣ

Τα Προγραμματιστικά Περιβάλλοντα για Αρχαρίους που επιλέχθηκαν και διαδραματίζουν πρωταγωνιστικό ρόλο για την υλοποίηση και των τεσσάρων σταδίων της έρευνας ήταν το Code.org, το Scratch και το Starlogo TNG.

Το Code.org χρησιμοποιήθηκε ως σημείο ελέγχου, για να διαπιστωθεί κατά πόσο κατανοήθηκαν οι αντίστοιχες έννοιες από τον κάθε μαθητή και για τα δυο τμήματα αλλά και για να ελεγχθεί το επίπεδο του κάθε τμήματος. Για το βαθμό κατανόησης και εξέτασης των μαθησιακών αποτελεσμάτων της κάθε έννοιας χρησιμοποιήθηκε η Ιεραρχική Αξιολόγηση Προγραμματισμού (Μικρόπουλος & Μπέλλου, 2008).

Στη συνέχεια για το κάθε τμήμα χρησιμοποιήθηκε διαφορετικό ΠΠΑ. Το τμήμα Στ1 χρησιμοποίησε το Starlogo TNG και το τμήμα Στ2 το Scratch που έγινε με τυχαίο τρόπο. Οι μαθητές και των δυο τμημάτων μετά από συζήτηση με τον εκπαιδευτικό αποφασίζουν και παροτρύνονται να δημιουργήσουν από το μηδέν ένα ολοκληρωμένο παιχνίδι – έργο το οποίο για την επιτυχή έκβαση του θα έπρεπε να χρησιμοποιήσουν όλες τις προγραμματιστικές έννοιες που είχαν ήδη διδαχτεί. Η

μέθοδος που χρησιμοποιήθηκε για την ολοκλήρωση αυτής της εργασίας ήταν η μέθοδος project όπως έχει προταθεί και από τους Μικρόπουλο & Μπέλλου (2005).

Ενδιαφέρον στοιχείο είναι το γεγονός ότι η διδασκαλία δεν έγινε με το παραδοσιακό δασκαλοκεντρικό μοντέλο αλλά μ' έναν τρόπο που ο καθηγητής, όσο περνούσε ο καιρός, εμπλεκόταν όλο και λιγότερο στη διαδικασία εκμάθησης. Σε αυτό το σημείο, κατά την δημιουργία του παιχνιδιού, ο ρόλος του εκπαιδευτικού ήταν περισσότερο ενθαρρυντικός και επενέβαινε όταν κάποια ομάδα δεν μπορούσε να συνεχίσει προσφέροντας κάθε φορά όμως την μικρότερη δυνατή βοήθεια. Αυτό καταγράφονταν κάθε φορά, στο φύλλο αξιολόγησης από τον εκπαιδευτικό.

### 3.4.3 ΙΕΡΑΡΧΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Στο τρίτο στάδιο χρησιμοποιήθηκε η Ιεραρχική Αξιολόγηση Προγραμματισμού (Hierarchical Assessment of Programming, HAP) των Μικρόπουλος & Μπέλλου, (2008) που είναι μια μέθοδος για την ποιοτική αξιολόγηση της επίλυσης προγραμματιστικών προβλημάτων. Οι βασικοί λόγοι για την επιλογή του στη παρούσα έρευνα είναι :

- Βασίζεται στα υπάρχοντα θεωρητικά μοντέλα για την ανάπτυξη της σκέψης στον προγραμματισμό.
- Υιοθετεί αρχές της γνωστικής ταξινόμιας μαθησιακών αποτελεσμάτων SOLO και λαμβάνει υπόψη τη διάκριση της γνώσης σε δηλωτική, δομική, διαδικαστική και στρατηγική.
- Ακολουθεί αναλυτική και συνθετική διαδικασία για τη συντακτική και εννοιολογική γνώση στον προγραμματισμό.
- Συγκεκριμενοποιεί και διατυπώνει τα κριτήρια αξιολόγησης της γνώσης προγραμματισμού, διαμορφώνοντας και παρουσιάζοντας πέντε ιεραρχικά επίπεδα, που ορίζονται με άξονες αφενός την ανάπτυξη αλγοριθμικής σκέψης για την επίλυση προβλήματος και αφετέρου τις δεξιότητες στη γλώσσα προγραμματισμού.

Η παραπάνω μέθοδος, όπως υποστηρίζουν και οι ίδιοι, μπορεί να εφαρμοστεί σε όλα τα προγραμματιστικά περιβάλλοντα για αρχαρίους και να υλοποιηθεί σε κάθε δραστηριότητα η οποία εμπλέκει προγραμματιστικές διεργασίες κάθε τύπου (Μικρόπουλος & Μπέλλου, 2008).

Άλλη μέθοδος αξιολόγησης είναι η ρουμπρίκα αξιολόγησης (Rubrics Assessment) που επιτρέπει την τυποποιημένη αξιολόγηση σύμφωνα με διευκρινισμένα κριτήρια (Andrade, 2000; Arter & McTighe, 2001). Τα βασικά μειονεκτήματα αυτής της μεθόδου είναι α) η δυσκολία εξαγωγής ενός γενικού συμπεράσματος από τα επιμέρους και β) μπορεί να οδηγήσει σε επουσιώδεις επισημάνσεις παραβλέποντας την ουσία (Nitko & Brookhart, 2007).

Η ποιοτική αξιολόγηση των μαθησιακών αποτελεσμάτων, με την ταξινόμια SOLO (*Structure of the Observed Learning Outcomes*) των Biggs & Collis (1982) αξιοποιείται σε ποικίλα γνωστικά αντικείμενα. Η ταξινόμια SOLO είναι γνωστική και αποτελεί μια εποικοδομητική προσέγγιση υποστηρίζει η Μπέλλου, (2003) στη μελέτη της «*ποιοτική αξιολόγηση μαθησιακών αποτελεσμάτων μαθητών μετά την αλληλεπίδρασή τους με εκπαιδευτικό λογισμικό*». Η συγκεκριμένη ταξινόμια δεν επιλέχτηκε, αφού η μέθοδος της Ιεραρχικής Αξιολόγησης βασίζεται σε αυτήν σε εξειδικευμένη και βελτιωμένη έκδοση για την περίπτωση του προγραμματισμού, κάνοντας διάκριση μεταξύ δεξιοτήτων επίλυσης προβλήματος (αλγοριθμική σκέψη) και δεξιοτήτων στη γλώσσα προγραμματισμού (γνώση στη γλώσσα προγραμματισμού).

Η Ιεραρχική Αξιολόγηση Προγραμματισμού διαδραματίζει βασικό ρόλο στην εξέλιξη της μελέτης. Χρησιμοποιείται σε δυο διαφορετικές χρονικές στιγμές στην πορεία της εργασίας. Στο πρώτο στάδιο κατά τη χρήση του Code.org για να χαρτογραφηθεί - αξιολογηθεί η επίδοση του κάθε μαθητή κατά την εμπλοκή του με την κάθε έννοια με σκοπό να καταγραφεί ο βαθμός της εννοιολογικής κατανόησης. Στο επόμενο στάδιο κατά τη μετάβαση σε διαφορετικά προγραμματιστικά περιβάλλοντα για αρχάριους, όπως το Scratch και το Starlogo TNG, όπου οι μαθητές καλούνται με την ελάχιστη βοήθεια να ολοκληρώσουν μόνοι τους ένα παιχνίδι με όλες τις έννοιες που έχουν ήδη διδαχτεί. Από αυτό θα εκμαιευτούν χρήσιμα στοιχεία-ευρήματα για την συγκριτική αξιολόγηση των ΠΠΑ αλλά και για την επίδοση των μαθητών κατά την χρήση διαφορετικών ΠΠΑ.

Βαθμολογία από το 1 έως το 5, από το μικρότερο στο μεγαλύτερο βαθμό, για την αξιολόγηση της επίδοσης του κάθε μαθητή για την κάθε έννοια, δόθηκε στους δυο άξονες στην γνώση της γλώσσα προγραμματισμού και στην αλγοριθμική σκέψη. Από αυτό προκύπτουν και τα πέντε ιεραρχικά επίπεδα μαθησιακών αποτελεσμάτων που συγκροτούν αυτό το μοντέλο. Στο Παράρτημα Α αναλυτικά παρουσιάζονται τα μαθησιακά αποτελέσματα κατά τη χρήση των ΠΠΑ που προκύπτουν από την Ιεραρχική Αξιολόγηση Προγραμματισμού.

Παραδείγματα αξιολόγησης για τις προγραμματιστικές έννοιες που μελετήθηκαν, με την Ιεραρχική Αξιολόγηση Προγραμματισμού αναπτύσσονται και επισυνάπτονται στο Παράρτημα Β. Σε αυτό παρουσιάζονται διάφορα προγραμματιστικά κομμάτια που χρησιμοποιούν αυτές τις έννοιες και διαβαθμίζονται σε πέντε ιεραρχικά επίπεδα μαθησιακών αποτελεσμάτων που συγκροτούν το μοντέλο της.

### 3.4.4 ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ

Στο τέταρτο στάδιο χρησιμοποιήθηκε το ερωτηματολόγιο ως μέθοδος αξιολόγησης των ΠΠΑ από τους μαθητές. Το ερωτηματολόγιο αποτελεί το μέσον επικοινωνίας μεταξύ του ερευνητή και των ερωτώμενων, με άμεσο ή έμμεσο τρόπο, ανάλογα με τη μέθοδο συλλογής των δεδομένων (Javeau, 2000). Είναι ένα έντυπο που απαρτίζεται μια ακολουθία ερωτήσεων, που παρουσιάζονται με μια συγκεκριμένη σειρά, που εξυπηρετεί τους σκοπούς του ερευνητή, και στις οποίες ο ερωτώμενος καλείται να απαντήσει γραπτά. Το ερωτηματολόγιο χρησιμοποιείται σε εφαρμοσμένες έρευνες για να ελεγχθούν οι υποθέσεις των ερευνητών και να επαληθευτούν θεωρίες, καθώς και για να προκύψουν περιγραφικές διαπιστώσεις (Javeau, 2000).

Επιπρόσθετα παρέχει δομημένα δεδομένα, είναι σχετικά εύκολο στην ανάλυσή του και μπορεί να διανεμηθεί χωρίς την παρουσία του ερευνητή (Wilson & McLean, 1994). Τα βασικά χαρακτηριστικά των ερωτηματολογίων είναι η ευκρίνεια στην διατύπωση των ερωτήσεων, η αμεσότητα, η πληρότητα, η απλότητα, η ισορροπία και το πλήθος των ερωτήσεων. Τα βασικότερα πλεονεκτήματα των ερωτηματολογίων είναι: α) Το χαμηλό κόστος υλοποίησης, β) η αποστολή σε μεγάλο πλήθος ανθρώπων, γ) η ελευθερία στην έκφραση των ερωτηθέντων, δ) η τυποποίηση των μεθόδων ανάλυσης του υλικού, ε) οι απαντήσεις των ερωτηθέντων δεν επηρεάζονται από τον ερευνητή και στ) η μικρότερη απαίτηση χρόνου από άλλα μέσα συλλογής δεδομένων.

Κατά το σχεδιασμό του ερωτηματολογίου καταβλήθηκε προσπάθεια να ληφθούν υπόψη ορισμένα χαρακτηριστικά όπως (Javeau, 2000):

- Πληρότητα
- Συνάφεια
- Συνοχή
- Κατάλληλη δομή

- Να διαθέτει ερωτήματα ελέγχου
- Να είναι όσο το δυνατόν πιο σύντομο
- Να έχει τελειότητα παρουσίασης από τεχνικής πλευράς
- Να περιλαμβάνει βασικές οδηγίες συμπλήρωσης και εννοιολογικές επεξηγήσεις
- Να επιδέχεται κωδικοποίηση και μηχανογραφική επεξεργασία

Μετά την ολοκλήρωση των προηγούμενων τριών σταδίων, όπως έχουν περιγράψει παραπάνω, δόθηκε στους μαθητές το ερωτηματολόγιο αξιολόγησης των ΠΠΑ που χρησιμοποιήθηκαν. Για την κατασκευή του ανώνυμου ερωτηματολογίου χρησιμοποιήθηκαν ερωτήσεις πεντάβαθμης κλίμακας τύπου Likert (καθόλου – πάρα πολύ). Αναλυτικά η μορφή του ερωτηματολογίου υπάρχει στο Παράρτημα Γ

Τα ερωτηματολόγια διαμορφώθηκαν για να καλύψουν ερωτήσεις των τριών παρακάτω ενοτήτων:

1. Τις απόψεις των μαθητών για τη χρήση του Υπολογιστή.
2. Τις απόψεις των μαθητών για τον Προγραμματισμό και
3. Τις Απόψεις των μαθητών για τα Προγραμματιστικά Περιβάλλοντα Αρχαρίων που χρησιμοποίησαν.

Για τις απόψεις των μαθητών για τη χρήση του Υπολογιστή και τον Προγραμματισμό οι ερωτήσεις ήταν ίδιες ενώ για τις απόψεις τους για τα Προγραμματιστικά Περιβάλλοντα Αρχαρίων που χρησιμοποίησαν υπήρχε διαφοροποίηση ως προς τις ερωτήσεις και γι' αυτό δημιουργήθηκαν δυο διαφορετικές παραλλαγές, αφού το ένα τμήμα (Στ1) ασχολήθηκε με τα Code.org και Starlogo TNG ενώ το άλλο τμήμα (Στ2) ασχολήθηκε με τα Code.org και Scratch.

### **3.4.5 ΑΠΟΝΟΜΗ ΠΙΣΤΟΠΟΙΗΤΙΚΟΥ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ**

Μετά την ολοκλήρωση των παραπάνω σταδίων πραγματοποιήθηκε η απονομή πιστοποιητικού παρακολούθησης σχήμα 3.2 σε όλους τους μαθητές και των δυο τμημάτων ως μια επιβράβευση για τη συμμετοχή αλλά και την ολοκλήρωση της εκπαιδευτικής διαδικασίας πάνω σε βασικές προγραμματιστικές έννοιες.

Η αναγνώριση της προσπάθειας άλλα και της βελτίωσης είναι σημαντική, καθώς τα παιδιά είναι πιο πιθανό να θέλουν να διατηρήσουν αυτή τη στάση τους απέναντι στον Προγραμματισμό και ίσως να προσπαθήσουν ακόμα περισσότερο. Η επιβράβευση αποτελεί σημαντικό κομμάτι στην εκπαιδευτική διαδικασία, αφού δημιουργεί συναισθήματα σιγουριάς, ικανοποίησης, χαράς, ικανότητες αρκετά χρήσιμες που έχουν καθοριστική σημασία στην προσπάθεια τους για να κατακτήσουν νέες δεξιότητες και εφόδια που θα χρειαστούν στο μέλλον (Benaroch, 2008). Σε αυτό συμφωνεί και ο Lewin Kurt, (1997) που αναφέρει ότι η ενίσχυση και η επιβράβευση ως τρόποι παροχής εξωτερικών κινήτρων, αυξάνουν τις πιθανότητες επανάληψης αυτής της ενέργειας.



Σχήμα 3.2 Πιστοποιητικό Παρακολούθησης



## 4 ΑΠΟΤΕΛΕΣΜΑΤΑ

Στην πρώτη ενότητα αυτού του κεφαλαίου παρουσιάζονται τα αποτελέσματα των μαθητών και των δυο τμημάτων σχετικά με την κάθε προγραμματιστική έννοια κατά τη χρήση του Code.org, με σκοπό να απαντηθεί το πρώτο ερευνητικό ερώτημα που αφορά στην επίδοση των μαθητών πάνω σε κάθε μια προγραμματιστική έννοια. Στη συνέχεια με τη χαρτογράφηση της επίδοσης του κάθε ενός ξεχωριστά, θα αναφερθούν οι εναλλακτικές ιδέες-συχνά λάθη που προέκυψαν από την ενασχόληση των μαθητών με αυτές τις έννοιες, προκειμένου να απαντηθεί και το δεύτερο ερευνητικό ερώτημα.

Σε επόμενη ενότητα παρουσιάζονται τα αποτελέσματα των μαθητών κατά τη μετάβαση και αξιοποίηση των ΠΠΑ, όπως το Scratch και το Starlogo TNG, για την κατασκευή ολοκληρωμένων έργων που βασίζονται στις έννοιες που διδαχτήκαν, με σκοπό να γίνει μια αποτίμηση και εκτίμηση του βαθμού κατανόησης και χρήσης της κάθε μιας. Έτσι, απαντάται το τρίτο ερευνητικό ερώτημα που αφορά στο αν η επίδοση των μαθητών κατά την ανάπτυξη και εμπλοκή τους με αυτές τις προγραμματιστικές έννοιες διαφοροποιείται κατά τη χρήση διαφορετικών ΠΠΑ. Στο τέλος της παρούσας ενότητας διερευνάται και στο τέταρτο ερευνητικό ερώτημα αναφορικά με το αν υπάρχει διαφοροποίηση στην επίδοση των δυο φύλων κατά τη χρήση αυτών των ΠΠΑ. Στο Παράρτημα Α αναλυτικά παρουσιάζονται τα μαθησιακά αποτελέσματα και των δυο τμημάτων κατά τη χρήση των ΠΠΑ Code.org, Starlogo TNG και Scratch.

Κλείνοντας την παρουσίαση των αποτελεσμάτων αναλύονται οι απαντήσεις των μαθητών όπως προέκυψαν από τα ερωτηματολόγια, διερευνώντας το τελευταίο ερευνητικό ερώτημα αναφορικά με τις απόψεις των μαθητών για τον Υπολογιστή, τον Προγραμματισμό, καθώς και για τα Προγραμματιστικά Περιβάλλοντα Αρχαρίων.

### 4.1 ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΕΝΝΟΙΩΝ ΣΤΟ CODE.ORG

Καταγράφονται τα αποτελέσματα για την κάθε μια Προγραμματιστική έννοια στο Code.org μέσω των δραστηριοτήτων με τις οποίες ασχολήθηκαν οι μαθητές.

#### 4.1.1 ΕΠΑΝΑΛΗΨΗ

Η πρώτη προγραμματιστική έννοια που θα εξεταστεί είναι η επανάληψη. Το σχήμα 4.1 περιγράφει την κατανομή των βαθμών για τα δυο τμήματα και πόσοι μαθητές κατάφεραν να φτάσουν στο αντίστοιχο επίπεδο, όπως καθορίζεται από την Ιεραρχική Αξιολόγηση Προγραμματισμού. Από αυτό προκύπτει η συνολική επίδοση των μαθητών για την επανάληψη στο Code.org. Τα ενδιάμεσα επίπεδα όπως το 2 προς 3 και αντίστοιχα 3 προς 4 προκύπτουν όταν οι μαθητές κατά την αξιολόγηση της γνώσης στη γλώσσα προγραμματισμού και στην αλγοριθμική σκέψη, δεν έχουν φτάσει στο ίδιο επίπεδο (Μικρόπουλος & Μπέλλου, 2008).

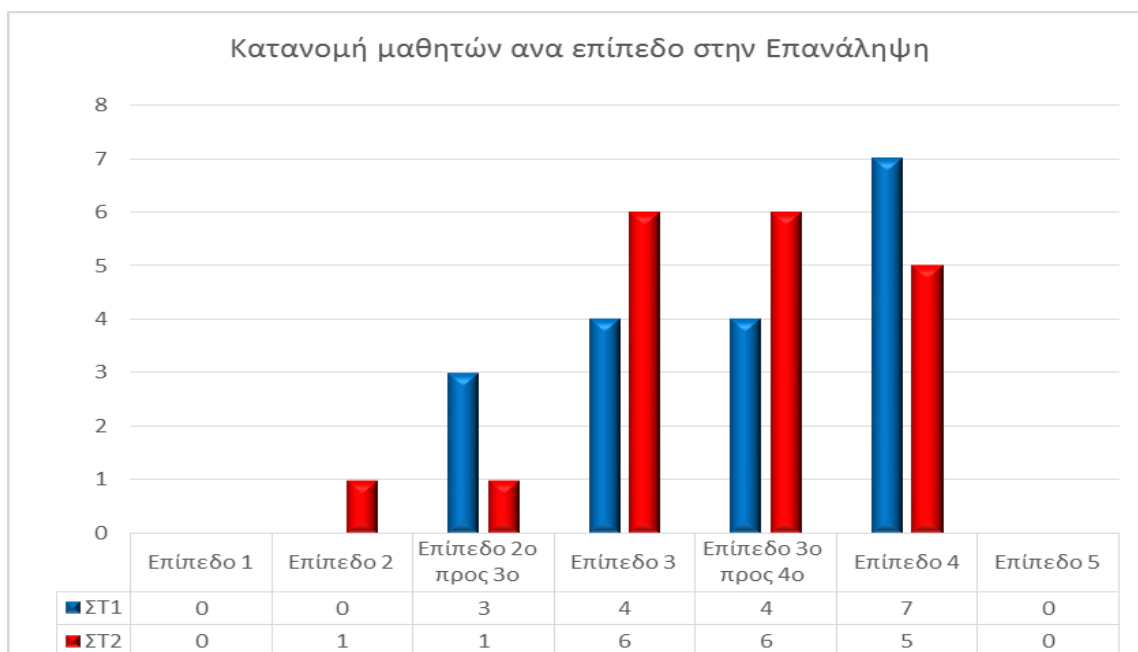
Κανένας μαθητής δεν αξιολογείται με το μικρότερο επίπεδο, δηλαδή το πρώτο, το προδρομικό. Ένας μαθητής φθάνει το αμέσως επόμενο επίπεδο, δηλαδή το δεύτερο που είναι της επιμέρους κατανόησης και τέσσερις μαθητές που αντιστοιχούν στο 11% βρίσκονται στο ενδιάμεσο επίπεδο από το δεύτερο και το τρίτο. Δέκα μαθητές, δηλαδή ένα ποσοστό 27% είναι στο τρίτο επίπεδο, αυτό της προσεγγιστικής κατανόησης. Δέκα μαθητές, 27%, βρίσκονται στο ενδιάμεσο επίπεδο από το τρίτο και το τέταρτο επίπεδο ενώ δώδεκα μαθητές, ποσοστό 32% κατάφεραν να φτάσουν το τέταρτο επίπεδο, το συνδυαστικό. Κανένας μαθητής δεν αξιολογείται στο ανώτερο επίπεδο το πέμπτο, που αφορά σε αυτό της εκτεταμένης θεώρησης.

Το γεγονός πως ούτε ένας μαθητής δεν βρίσκεται στο πρώτο επίπεδο μπορεί να οφείλεται στο ότι οι δραστηριότητες παρέχουν ανατροφοδότηση και καθοδήγηση. Από την άλλη, το ότι δεν υπάρχει κανείς από το δείγμα που να αξιολογείται στο πέμπτο επίπεδο μπορεί να οφείλεται στο ότι δεν υπάρχει μεγάλη δυνατότητα επιλογών ως προς τον συνδυασμό εντολών που παρέχει το ΠΠΑ Code.org, έτσι ώστε να δίνεται η δυνατότητα σε κάποιον χρήστη να τις χρησιμοποιήσει, για να δημιουργήσει μια ξεκάθαρη διαφορά από τους υπόλοιπους.

Τα δυο τμήματα δεν ικανοποιούν την κανονική κατανομή ( $p < 0.05$ ) σύμφωνα με τους ελέγχους κανονικότητας που πραγματοποιήθηκαν. Πιο συγκεκριμένα, το Στ1 έχει τιμή  $p=0.10$  και το Στ2 έχει  $p=0.45$ .

Για να εξεταστεί αν τα δυο τμήματα παρουσιάζουν κάποια διαφορά ως προς την επίδοσή τους κατά την χρήση της επανάληψης, θα χρησιμοποιηθεί το τεστ Mann-Whitney U που είναι και το καταλληλότερο, εφόσον δεν ισχύει η υπόθεση κανονικότητας. Το αποτέλεσμα του τεστ έδειξε ότι το Στ1 με (μέσο όρο - median = 3.41; μέση κατάταξη - mean rank =19.72) είχε καλύτερη επίδοση στην

επανάληψη από το Στ2 με (μέσο όρο - median = 3.342; μέση κατάταξη - mean rank =18.32). Η τιμή του Mann-Whitney U βρέθηκε να είναι μη στατιστικά σημαντική (U = 158.0, Z = -.411, p > 0.05).



Σχήμα 4.1 Κατανομή μαθητών στην επανάληψη στο Code.org

Παρακάτω αναφέρονται και επεξηγούνται κάποιες παρανοήσεις των μαθητών σχετικά με την επανάληψη, όπως και στον τρόπο που την αντιμετωπίζουν και την χρησιμοποιούν.

Οι μαθητές προτιμούν να προγραμματίζουν ακολουθιακά από το να χρησιμοποιούν την επαναληπτική δομή, όπως χαρακτηριστικά φαίνεται στην περίπτωση Δ του πίνακα 4.1. Αυτό βιβλιογραφικά αναφέρεται ως αδυναμία γενίκευσης όπου οι αρχάριοι προγραμματιστές παρουσιάζουν την τάση να χρησιμοποιούν στα προγράμματά τους μια λίστα επαναλαμβανόμενων εντολών αντί να χρησιμοποιούν ένα βρόχο επανάληψης (Hoc, 1989).

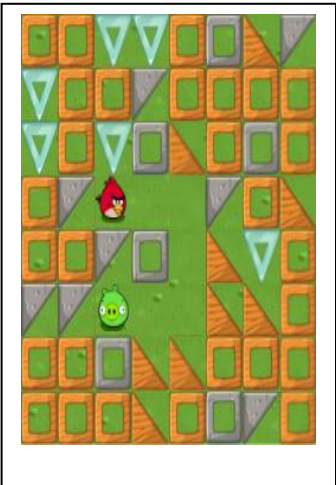
Αφού αποφασίσουν να την χρησιμοποιήσουν, το κάνουν αλλά με έναν ακολουθιακό τρόπο, όπως φαίνεται και στην περίπτωση Α. Αυτό δικαιολογείται εν μέρει, αφού η δομή επανάληψης είναι η πρώτη εξεταζόμενη προγραμματιστική δομή – έννοια που έρχονται σε επαφή αμέσως μετά από αυτήν της ακολουθίας.

Επίσης, γίνεται αντιληπτό πως οι μαθητές δεν ομαδοποιούν τις εντολές που είναι να χρησιμοποιηθούν στην επανάληψη, γεγονός που δικαιολογείται από το ότι δεν σκέφτονται το πρόβλημα συνολικά αλλά μονομερώς-βηματικά αφού δεν εξετάζουν τη συνολική λύση του

προβλήματος αλλά τη λύση του επόμενου βήματος, όπως αποδεικνύεται και από στις περιπτώσεις Β και Γ.

Μάλιστα φαίνεται να προτιμούν να επαναλαμβάνουν τα ίδια βήματα ξανά και ξανά από το να βρουν ένα μοτίβο επανάληψης για μια συνολική λύση, όπως άλλωστε αποτυπώνεται και στην εικόνα Γ. Αν κι αυτά τα αποτελέσματα αναφέρονται σε ένα μικρό ποσοστό μαθητών, μολαταύτα κρίθηκε απαραίτητο να αναφερθούν.

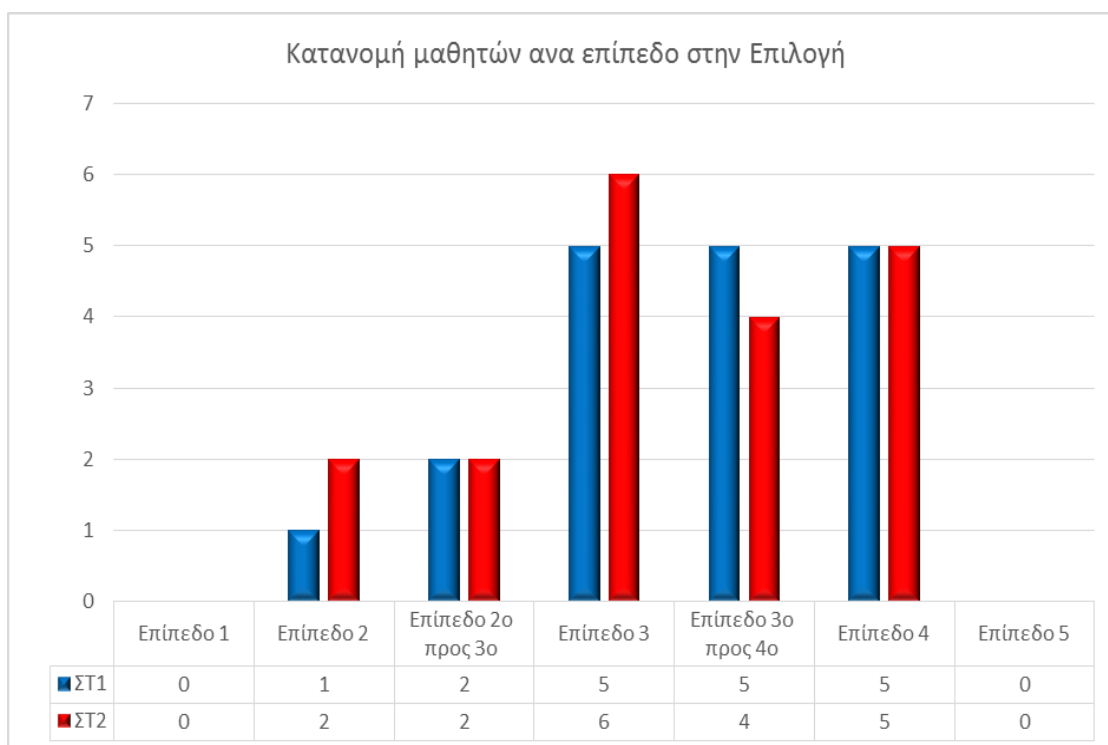
Πίνακας 4.1 Παρανοήσεις μαθητών στην επανάληψη

Επανάληψη			
Δραστηριότητα 1	Εναλλακτικές ιδέες – Συχνά λάθη	Ορθή αντιμετώπιση	
<p>Κίνηση του κόκκινου χαρακτήρα για να φτάσει στον πράσινο</p> 	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>όταν εκτελείται</p> <p>επανάλαβε 3 φορές</p> <p>κάνε προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> <p>στρίψε δεξιά 90°</p> <p>προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> <p>στρίψε δεξιά 90°</p> <p>προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> </div> <p style="text-align: center;">Α</p> <div style="border: 1px solid black; padding: 5px;"> <p>όταν εκτελείται</p> <p>επανάλαβε 2 φορές</p> <p>κάνε προχώρησε μπροστά</p> <p>επανάλαβε 1 φορές</p> <p>κάνε στρίψε δεξιά 90°</p> <p>επανάλαβε 2 φορές</p> <p>κάνε προχώρησε μπροστά</p> <p>επανάλαβε 1 φορές</p> <p>κάνε στρίψε δεξιά 90°</p> <p>επανάλαβε 3 φορές</p> <p>κάνε προχώρησε μπροστά</p> </div> <p style="text-align: center;">Γ</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>όταν εκτελείται</p> <p>προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> <p>στρίψε δεξιά 90°</p> <p>επανάλαβε 2 φορές</p> <p>κάνε προχώρησε μπροστά</p> <p>στρίψε δεξιά 90°</p> <p>επανάλαβε 3 φορές</p> <p>κάνε προχώρησε μπροστά</p> </div> <p style="text-align: center;">Β</p> <div style="border: 1px solid black; padding: 5px;"> <p>όταν εκτελείται</p> <p>προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> <p>στρίψε δεξιά 90°</p> <p>προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> <p>στρίψε δεξιά 90°</p> <p>προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> </div> <p style="text-align: center;">Δ</p>	<div style="border: 1px solid black; padding: 5px;"> <p>όταν εκτελείται</p> <p>επανάλαβε 3 φορές</p> <p>κάνε προχώρησε μπροστά</p> <p>προχώρησε μπροστά</p> <p>στρίψε δεξιά 90°</p> </div>

#### 4.1.2 ΕΠΙΛΟΓΗ

Η δεύτερη προγραμματιστική έννοια στην οποία θα αναφερθούμε και εξετάσαμε είναι η επιλογή. Το σχήμα 4.2 αφορά και στα δύο τμήματα και καταγράφει την κατανομή των βαθμών, αλλά και το πόσοι μαθητές καταφέρνουν να φτάσουν στο αντίστοιχο επίπεδο με βάση την Ιεραρχική Αξιολόγηση Προγραμματισμού.

Κανένας μαθητής δεν αξιολογείται με το μικρότερο επίπεδο το πρώτο, το προδρομικό. Τρεις μαθητές, δηλαδή το 8% φθάνουν το δεύτερο επίπεδο της επιμέρους κατανόησης. Τέσσερις μαθητές, δηλαδή το 11%, αξιολογούνται στο ενδιάμεσο από το δεύτερο και τρίτο επίπεδο. Έντεκα μαθητές, 30%, βρίσκονται στο τρίτο επίπεδο της προσεγγιστικής κατανόησης. Εννιά μαθητές, ποσοστό 24% από αυτούς αγγίζουν το ενδιάμεσο, από το τρίτο και το τέταρτο επίπεδο. Τέλος δέκα μαθητές, 27%, καταφέρνουν να φθάσουν το τέταρτο επίπεδο, το συνδυαστικό, ενώ αντίθετα κανένας μαθητής δεν αξιολογείται στο ανώτερο επίπεδο το πέμπτο που είναι αυτό της εκτεταμένης θεώρησης.



Σχήμα 4.2 Κατανομή μαθητών στην επιλογή στο Code.org

Τα δυο τμήματα ικανοποιούν την υπόθεση της κανονικής κατανομής, αφού το  $p > 0.05$  για το καθένα από αυτά, σύμφωνα με τους ελέγχους κανονικότητας που πραγματοποιήθηκαν. Πιο συγκεκριμένα, το Στ1 έχει τιμή  $p=0.55$  και το Στ2 έχει  $p=0.78$ .

Προκειμένου να εξεταστεί το αν τα δυο τμήματα παρουσιάζουν στατιστική διαφορά ως προς την επίδοσή τους κατά τη χρήση της επιλογής, θα χρησιμοποιηθεί το τεστ ανεξάρτητων δειγμάτων - independent samples t-test λόγω της ύπαρξης κανονικής κατανομής. Διαπιστώνεται ότι δεν υπάρχει στατιστικά σημαντική διαφορά μεταξύ του Στ1 ( $M = 3.3, SD = 0.59$ ) και του Στ2 ( $M = 3.19, SD = 0.65$ ) στην επιλογή ( $t(35)=0.461, p > 0.05$ ).

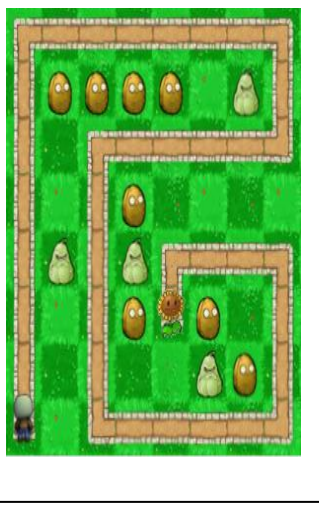
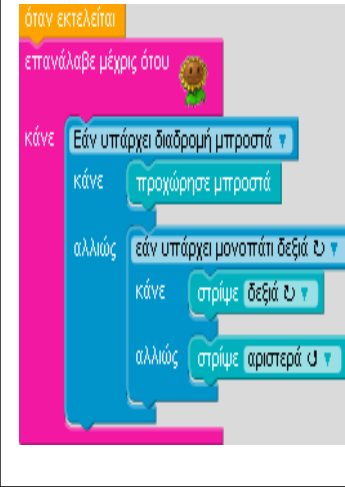
Παρακάτω καταγράφονται και περιγράφονται οι πιο συχνές παρανοήσεις των μαθητών σχετικά με την επιλογή και τον τρόπο που την αντιμετωπίζουν και την χρησιμοποιούν.

Επαναλαμβάνουν την εντολή «προχώρησε μπροστά» της αρχικής συνθήκης ελέγχου, και στις υπόλοιπες συνθήκες ελέγχου ακόμα και αν ο έλεγχος σε αυτές είναι διαφορετικός, όπως αυτό αποτυπώνεται στις περιπτώσεις Α, Β και Γ του πίνακα. Βιβλιογραφικά το συναντούμε στους Putnam (1989) και Sleeman, (1988) για την περίπτωση που η δομή επιλογής έχει και τμήμα ELSE όπου οι μαθητές αναμένουν την εκτέλεση τόσο του τμήματος THEN όσο και του ELSE. Επίσης αναμένουν την εκτέλεση του τμήματος «κάνε» - THEN ανεξάρτητα από το αν η συνθήκη είναι αληθής ή όχι, όπως ακριβώς αναφέρεται και στην περίπτωση του Putnam, (1989) και παρουσιάζεται στην περίπτωση Α.

Σημαντικό εμπόδιο για την οικοδόμηση της δομής ελέγχου αφορά η αναπαράσταση που διαθέτουν οι μαθητές για τη σειριακή εκτέλεση του προγράμματος. Μέσα σε ένα πρόγραμμα που διαθέτει μια αυστηρή αλληλουχία εντολών, για κάθε είσοδο δεδομένων, το σύνολο των περιγραφόμενων υπολογισμών πραγματοποιείται με τη σειρά εγγραφής των εντολών. Οι δομές ελέγχου διακόπτουν αυτό τον ισομορφισμό γραμμικής τάξης ανάμεσα στο κείμενο του προγράμματος και στην εκτέλεσή του (Rogalski & Vergnaud, 1987).

Στην περίπτωση των δομών ελέγχου μόνο μία από τις δυνατές διαδρομές θα ακολουθηθεί κατά την εκτέλεση. Μια δομή ελέγχου οφείλει να διαχειριστεί την τροποποίηση των σχέσεων ανάμεσα στη γραμμική γραφή του προγράμματος και τη σειριακότητα της εκτέλεσης.

Πίνακας 4.2 Παρανοήσεις μαθητών στην επιλογή

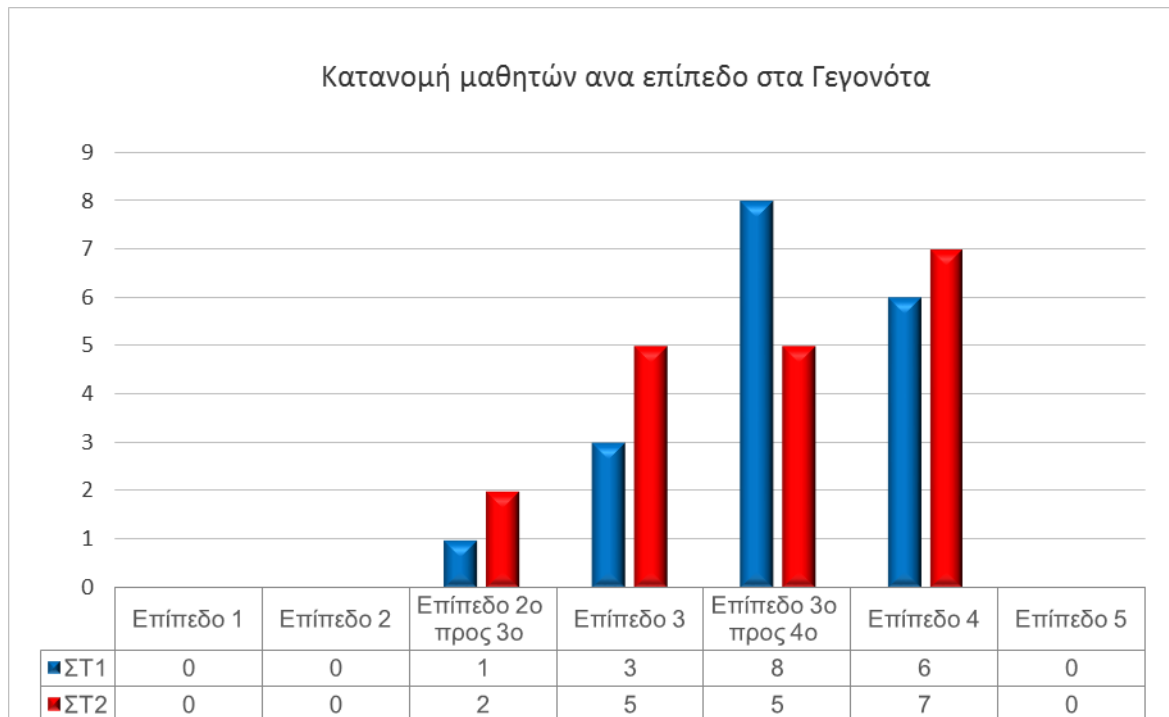
Επιλογή		
Δραστηριότητα 2	Εναλλακτικές ιδέες – Συχνά λάθη	Ορθή αντιμετώπιση
<p data-bbox="55 481 375 548">Κίνηση του ανθρώπου για να φτάσει στο λουλούδι</p> 	<div style="display: flex; justify-content: space-around;"> <div data-bbox="411 412 794 891"> <p><b>A</b></p> <pre> όταν εκτελείται   επανάλαβε μέχρις ότου     κάνει       Εάν υπάρχει διαδρομή μπροστά         κάνει           προχώρησε μπροστά         αλλιώς           εάν υπάρχει μονοπάτι δεξιά             κάνει               στρίψε δεξιά               προχώρησε μπροστά             αλλιώς               στρίψε αριστερά               προχώρησε μπροστά           </pre> </div> <div data-bbox="799 412 1182 891"> <p><b>B</b></p> <pre> όταν εκτελείται   επανάλαβε μέχρις ότου     κάνει       Εάν υπάρχει διαδρομή μπροστά         κάνει           προχώρησε μπροστά         αλλιώς           εάν υπάρχει μονοπάτι δεξιά             κάνει               στρίψε δεξιά             αλλιώς               στρίψε αριστερά               προχώρησε μπροστά           </pre> </div> </div> <div data-bbox="555 913 970 1397" style="text-align: center;"> <p><b>Γ</b></p> <pre> όταν εκτελείται   επανάλαβε μέχρις ότου     κάνει       Εάν υπάρχει διαδρομή μπροστά         κάνει           προχώρησε μπροστά         αλλιώς           εάν υπάρχει μονοπάτι δεξιά             κάνει               στρίψε δεξιά             αλλιώς               Εάν υπάρχει διαδρομή μπροστά                 κάνει                   προχώρησε μπροστά                 αλλιώς                   στρίψε αριστερά           </pre> </div>	 <pre> όταν εκτελείται   επανάλαβε μέχρις ότου     κάνει       Εάν υπάρχει διαδρομή μπροστά         κάνει           προχώρησε μπροστά         αλλιώς           εάν υπάρχει μονοπάτι δεξιά             κάνει               στρίψε δεξιά             αλλιώς               στρίψε αριστερά           </pre>

### 4.1.3 ΓΕΓΟΝΟΤΑ

Η τρίτη προγραμματιστική έννοια είναι τα γεγονότα. Το σχήμα 4.3 απεικονίζει την κατανομή των βαθμών και για τα δυο τμήματα, καθώς και το πόσοι μαθητές φθάνουν και σε ποιο επίπεδο κατά αντιστοιχία με την ΙΑΠ.

Κανένας μαθητής δεν αξιολογείται με το μικρότερο επίπεδο το πρώτο που είναι το προδρομικό και κανένας με το αμέσως επόμενο που είναι το δεύτερο της επιμέρους κατανόησης. Τέσσερις μαθητές (8%) βρίσκονται στο ενδιάμεσο από το δεύτερο και τρίτο επίπεδο. Οκτώ μαθητές, δηλαδή

ένα ποσοστό 22% φθάνουν το τρίτο επίπεδο της προσεγγιστικής κατανόησης και δεκατρείς, ένα ποσοστό 35%, αξιολογούνται στο ενδιάμεσο επίπεδο από το τρίτο και το τέταρτο επίπεδο. Τόσο είναι ακριβώς και οι μαθητές που αγγίζουν το τέταρτο επίπεδο το συνδυαστικό. Τέλος, κανένας από το δείγμα δεν αξιολογείται στο ανώτερο επίπεδο το πέμπτο που είναι της εκτεταμένης θεώρησης.



Σχήμα 4.3 Κατανομή μαθητών στα γεγονότα στο Code.org

Τα δυο τμήματα δεν ικανοποιούν την υπόθεση κανονικής κατανομής αφού το  $p < 0,05$ , σύμφωνα με τους ελέγχους κανονικότητας που πραγματοποιήθηκαν. Ειδικότερα, το Στ1 έχει τιμή  $p=0.7$  και το Στ2 έχει  $p=0.14$ . Για να εξεταστεί αν τα δυο τμήματα παρουσιάζουν στατιστική διαφορά ως προς την επίδοσή τους κατά την χρήση αυτής της προγραμματιστικής έννοιας, χρησιμοποιεί το τεστ Mann-Whitney U. Το αποτέλεσμα του τεστ έδειξε ότι το Στ1 με (μέσο όρο - median = 3.53; μέση κατάταξη - mean rank =19.99) είχε καλύτερη επίδοση στα γεγονότα από το Στ2 με (μέσο όρο - median = 3.44; μέση κατάταξη - mean rank =18.10). Η τιμή του Mann-Whitney U βρέθηκε να είναι μη στατιστικά σημαντική με  $U = 158.5$ ,  $Z = -.400$  και  $p > 0.05$ .


Σχετικά με τις παρανοήσεις-εναλλακτικές προσεγγίσεις των μαθητών στα γεγονότα, σύμφωνα με την παρακάτω δραστηριότητα, αναφέρεται ότι οι μαθητές συχνά παρανοούν τον προγραμματισμό των γεγονότων σε ενέργειες που αφορούν το στρίψιμο του αντικειμένου στα αριστερά και τα δεξιά,

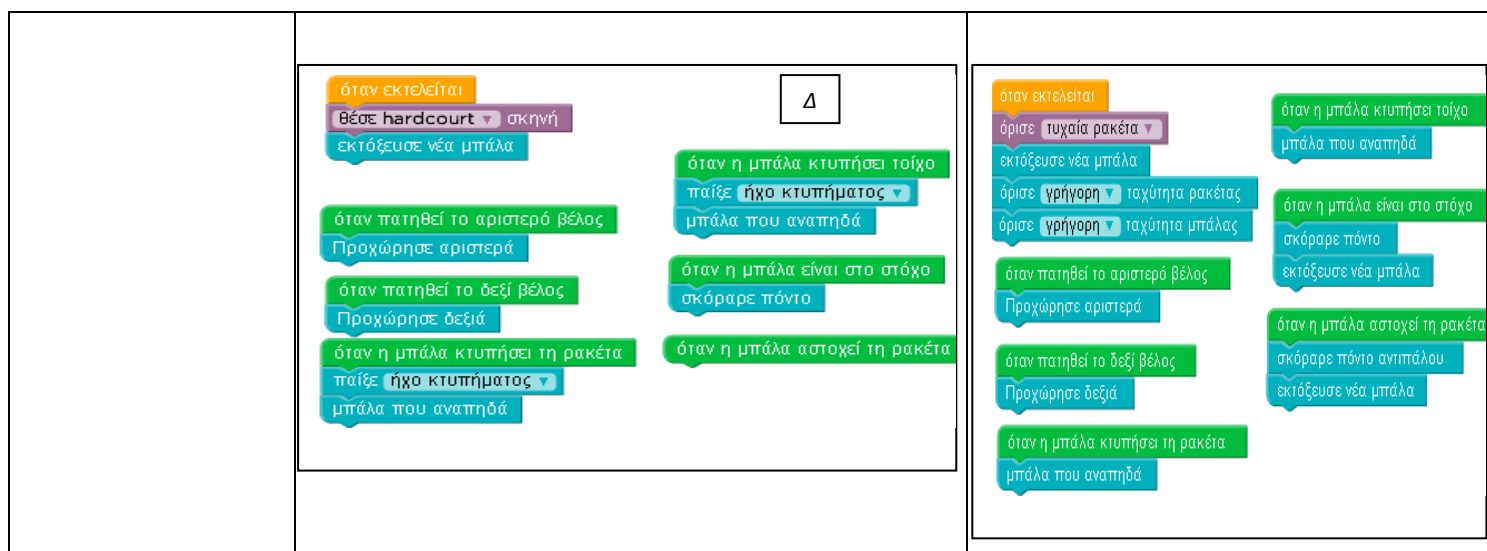


καθώς χρησιμοποιούν την εντολή κινήσου αριστερά, όταν πατηθεί το δεξί βέλος και την κίνηση στα δεξιά, όταν πατηθεί το αριστερό βέλος όπως φαίνεται στην περίπτωση A της δραστηριότητας 3. Αρκετές φορές μάλιστα δεν λαμβάνουν υπόψη τους τη διεκπεραίωση όλων των γεγονότων, όπως το τι θα γίνει αν η μπάλα βρει τον στόχο. Σε αυτήν την περίπτωση θα αυξηθεί το σκορ αλλά για να συνεχίσει το παιχνίδι θα πρέπει να εκτοξευτεί καινούργια μπάλα, πράγμα που παραλείπουν όπως απεικονίζεται και στις εικόνες Β και Γ του πίνακα 4.3.

Τέλος, υπάρχουν περιπτώσεις κατά τις οποίες δεν λαμβάνουν υπόψη τους περισσότερες καταστάσεις-γεγονότα όπως τι θα γίνει αν η ρακέτα αστοχήσει τη μπάλα, περίπτωση Δ.

Πίνακας 4.3 Παρανοήσεις μαθητών στα γεγονότα

Γεγονότα		
Δραστηριότητα 3	Εναλλακτικές ιδέες – Συχνά λάθη	Ορθή αντιμετώπιση
<div data-bbox="67 947 300 1115" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>Δημιουργία-Κατασκευή του γνωστού Παιχνιδιού Pong</p> </div> 	<div style="text-align: center; margin-bottom: 10px;"> <span style="border: 1px solid black; padding: 2px;">A</span> </div> <ul style="list-style-type: none"> <li>όταν εκτελείται εκτόξευσε νέα μπάλα</li> <li>όταν πατηθεί το αριστερό βέλος Προχώρησε δεξιά</li> <li>όταν πατηθεί το δεξί βέλος Προχώρησε αριστερά</li> <li>όταν η μπάλα κτυπήσει τη ρακέτα μπάλα που αναπηδή</li> <li>όταν η μπάλα κτυπήσει τοίχο μπάλα που αναπηδή</li> <li>όταν η μπάλα είναι στο στόχο σκόραρε πόντο</li> <li>σκόραρε πόντο αντιπάλου</li> <li>όταν η μπάλα αστοχεί τη ρακέτα σκόραρε πόντο αντιπάλου</li> <li>εκτόξευσε νέα μπάλα</li> </ul> <div style="text-align: center; margin-bottom: 10px;"> <span style="border: 1px solid black; padding: 2px;">B</span> </div> <ul style="list-style-type: none"> <li>όταν εκτελείται θέσε <b>hardcourt</b> σκηνή</li> <li>εκτόξευσε νέα μπάλα</li> <li>όρισε <b>κανονική</b> ταχύτητα μπάλας</li> <li>όταν πατηθεί το αριστερό βέλος Προχώρησε αριστερά</li> <li>όταν πατηθεί το δεξί βέλος Προχώρησε δεξιά</li> <li>όταν η μπάλα κτυπήσει τη ρακέτα μπάλα που αναπηδή</li> <li>όταν η μπάλα κτυπήσει τοίχο μπάλα που αναπηδή</li> <li>σκόραρε πόντο</li> <li>όταν η μπάλα είναι στο στόχο σκόραρε πόντο</li> <li>όταν η μπάλα αστοχεί τη ρακέτα σκόραρε πόντο αντιπάλου</li> </ul>	<div style="text-align: center; margin-bottom: 10px;"> <span style="border: 1px solid black; padding: 2px;">ή</span> </div> <ul style="list-style-type: none"> <li>όταν εκτελείται εκτόξευσε νέα μπάλα</li> <li>όταν πατηθεί το αριστερό βέλος Προχώρησε αριστερά</li> <li>όταν πατηθεί το δεξί βέλος Προχώρησε δεξιά</li> <li>όταν η μπάλα κτυπήσει τη ρακέτα παίξε ήχο κτυπήματος μπάλα που αναπηδή</li> <li>όταν η μπάλα κτυπήσει τοίχο παίξε ήχο κτυπήματος μπάλα που αναπηδή</li> <li>όταν η μπάλα είναι στο στόχο παίξε ήχο πόντου νίκης</li> <li>σκόραρε πόντο</li> <li>εκτόξευσε νέα μπάλα</li> <li>όταν η μπάλα αστοχεί τη ρακέτα παίξε ήχο απώλειας πόντου</li> <li>σκόραρε πόντο αντιπάλου</li> <li>εκτόξευσε νέα μπάλα</li> </ul>
	<div style="text-align: center; margin-bottom: 10px;"> <span style="border: 1px solid black; padding: 2px;">Γ</span> </div> <ul style="list-style-type: none"> <li>όταν εκτελείται εκτόξευσε νέα μπάλα</li> <li>όταν πατηθεί το αριστερό βέλος Προχώρησε αριστερά</li> <li>όταν πατηθεί το δεξί βέλος Προχώρησε δεξιά</li> <li>όταν η μπάλα κτυπήσει τη ρακέτα μπάλα που αναπηδή</li> <li>όταν η μπάλα κτυπήσει τοίχο μπάλα που αναπηδή</li> <li>όταν η μπάλα είναι στο στόχο σκόραρε πόντο</li> <li>όταν η μπάλα αστοχεί τη ρακέτα σκόραρε πόντο αντιπάλου</li> </ul>	

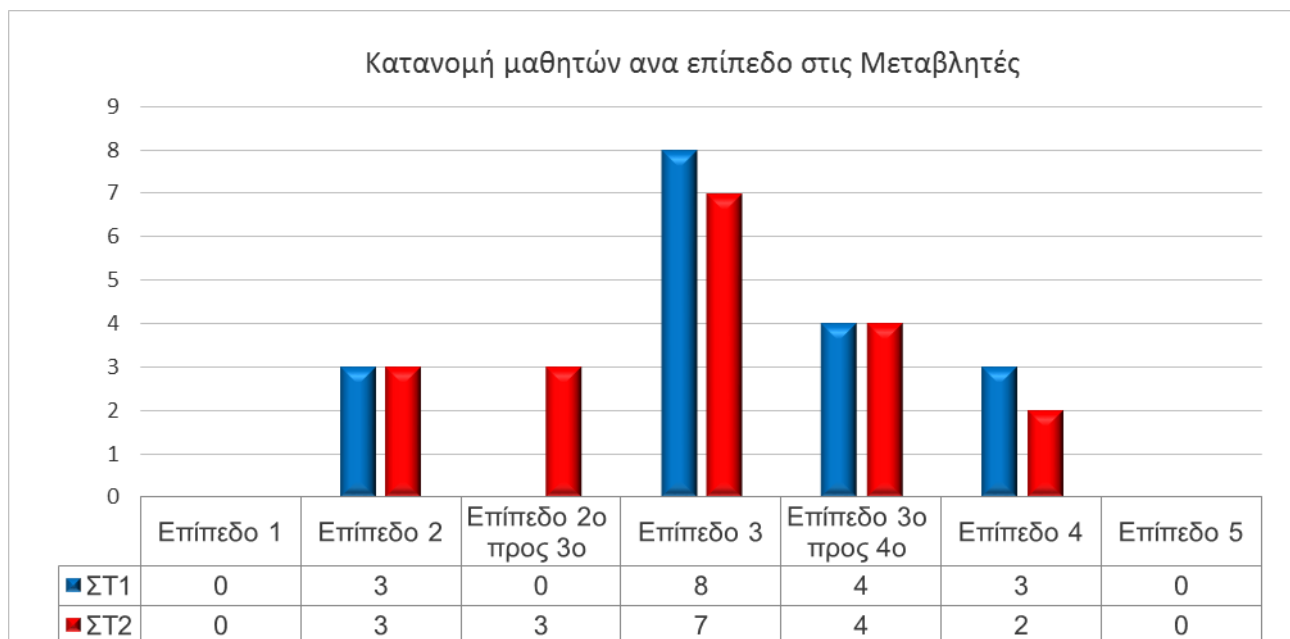


#### 4.1.4 ΜΕΤΑΒΛΗΤΕΣ

Η τέταρτη προγραμματιστική έννοια είναι οι μεταβλητές. Το σχήμα 4.4 απεικονίζει το πώς κατανεμήθηκαν οι βαθμοί των μαθητών, και για τα δύο τμήματα, και επιπρόσθετα το πόσοι μαθητές κατάφεραν να φτάσουν στο αντίστοιχο επίπεδο βάση της ΙΑΠ.

Κανένας μαθητής δεν αξιολογείται με το μικρότερο επίπεδο το πρώτο που είναι το προδρομικό. Έξι μαθητές (16%) φθάνουν στο αμέσως επόμενο επίπεδο, που αποτελεί και το δεύτερο, αυτό της επιμέρους κατανόησης. Τρεις μαθητές (8%) βρίσκονται στο ενδιάμεσο από το δεύτερο και το τρίτο επίπεδο. Δεκαπέντε μαθητές (41%) αξιολογούνται στο τρίτο επίπεδο, αυτό της προσεγγιστικής κατανόησης. Οκτώ από αυτούς (22%) βρίσκονται στο ενδιάμεσο από το τρίτο και το τέταρτο επίπεδο. Τέλος, πέντε μαθητές (14%) κατάφεραν να φτάσουν το τέταρτο επίπεδο, το συνδυαστικό, ενώ κανένας μαθητής δεν βρίσκεται στο ανώτερο επίπεδο, το πέμπτο, που αφορά σε αυτό της εκτεταμένης θεώρησης.

Σε αυτό το σημείο θα εξεταστεί αν τα δυο τμήματα παρουσιάζουν στατιστική διαφορά στις μεταβλητές. Αρχικά θα ελεγχθεί η υπόθεση κανονικής κατανομής η οποία όμως δεν ικανοποιείται, καθώς το  $p$  είναι μικρότερο από το 0.05 και γι' αυτό το λόγο θα χρησιμοποιηθεί το τεστ Mann-Whitney U.



Σχήμα 4.4 Κατανομή μαθητών στις μεταβλητές στο Code.org

Το περιγραφικά αποτέλεσμα του τεστ καταδεικνύουν ότι το Στ1 με (μέσο όρο - median = 3,1; μέση κατάταξη - mean rank = 20.5) παρουσιάζει καλύτερη επίδοση στις μεταβλητές από το Στ2 με (μέσο όρο - median = 2,9; μέση κατάταξη - mean rank = 17.7). Η τιμή του Mann-Whitney U βρέθηκε να είναι μη στατιστικά σημαντική ( $U = 147.5$ ,  $Z = -.756$ ,  $p > 0.05$  και ίση με 0.45).


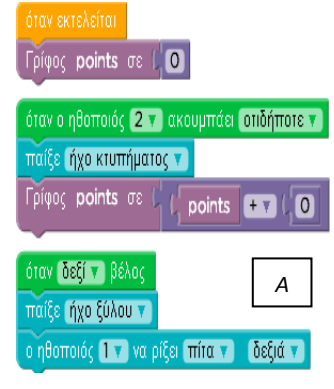
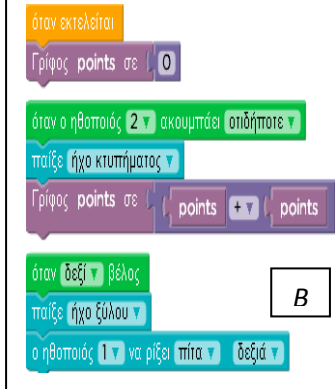
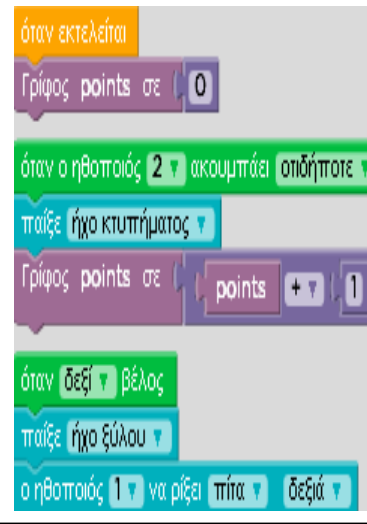
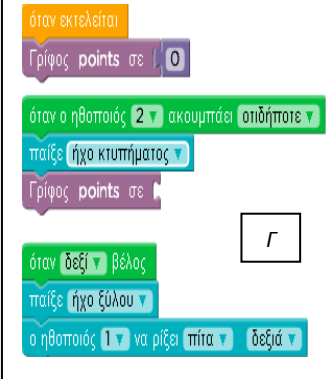
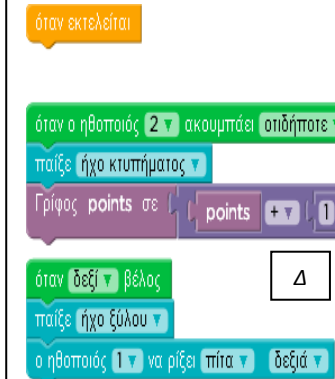

Σχετικά με τις παρανοήσεις και τις εναλλακτικές προσεγγίσεις των μαθητών στις μεταβλητές αναφέρεται ότι αρκετές από αυτές εστιάζονται στην αρχικοποίηση της τιμής της μεταβλητής. Αναλυτικότερα οι μαθητές συχνά παραλείπουν να αναθέσουν αρχική τιμή όταν ορίζουν μια μεταβλητή, με αποτέλεσμα να παραμένει κενή και να δημιουργούνται σφάλματα κατά την εκτέλεση του προγράμματός τους, όπως φαίνεται και στην περίπτωση Δ του πίνακα 4.4.

Δυσκολία παρουσιάζουν επίσης οι μαθητές και στην εκχώρηση τιμής σε μια μεταβλητή. Στην περίπτωση που σε μια μεταβλητή ανατίθεται μια τιμή με τη μορφή αριθμητικής έκφρασης (π.χ.  $x=y+2$ ) θεωρούν ότι η συγκεκριμένη μεταβλητή αποθηκεύει την αριθμητική έκφραση και όχι την τιμή που προκύπτει από αυτή. Δυσκολία φαίνεται να υπάρχει και στον τρόπο που γίνεται η αλλαγή μιας μεταβλητής, όταν μάλιστα η μεταβλητή χρησιμοποιείται ως ένας απλός μετρητής, όπως αποτυπώνεται και στις περιπτώσεις Α, Β και Ε.

Σε αυτό το σημείο, ωστόσο, πρέπει να αναφερθεί πως εντοπίζεται κι ένα μικρό ποσοστό μαθητών που δεν αλλάζει καθόλου την τιμή μιας μεταβλητής, όπως αυτό γίνεται αντιληπτό και στην περίπτωση Γ.

Οι μεταβλητές είναι από τις δύσκολες έννοιες που μπορεί να κατανοήσει αλλά και να εφαρμόσει κάποιος μαθητής και σε αυτό βιβλιογραφικά συμφωνούν πολλοί ερευνητές (Kuittinen & Sajaniemi, 2004; Samurçay, 1989).

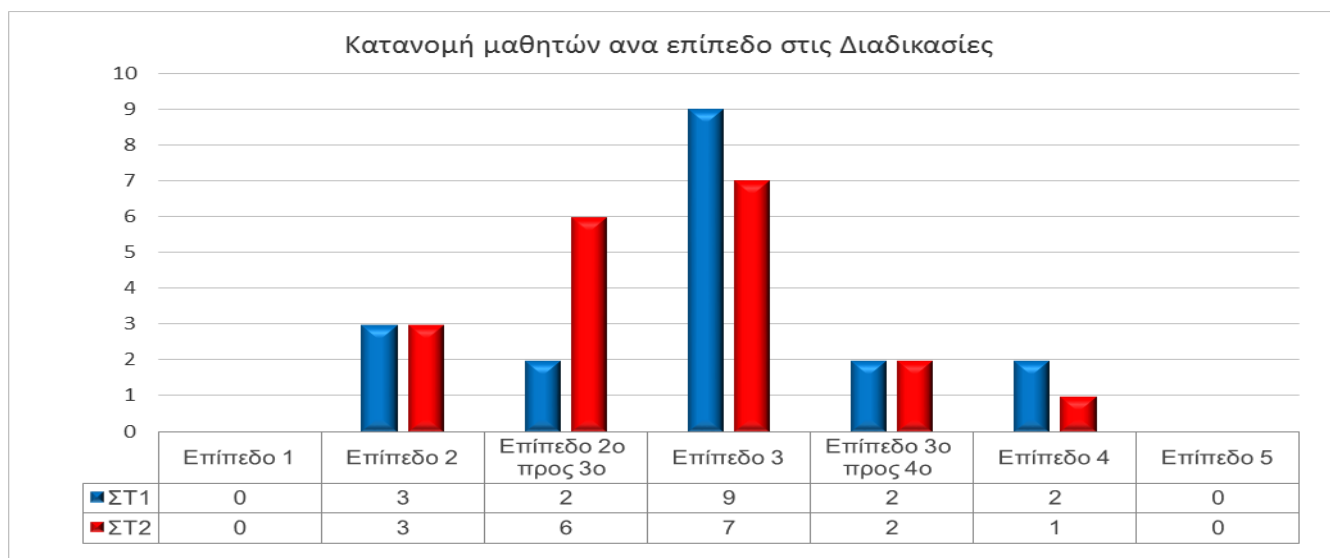
Πίνακας 4.4 Παρανοήσεις μαθητών στις μεταβλητές

Μεταβλητές	Εναλλακτικές ιδέες – Συχνά λάθη		Ορθή αντιμετώπιση
Δραστηριότητα	Εναλλακτικές ιδέες – Συχνά λάθη		Ορθή αντιμετώπιση
<p>Παιχνίδι με την χρήση μεταβλητών. Ο σκύλος(ηθοποιός 1) ρίχνει στην γάτα (ηθοποιός 2) κομμάτια από πίτα. Κάθε φορά που η γάτα ακουμπάει μια πίτα αυξάνεται το σκορ κατά 1.</p> 	 <p>A</p>	 <p>B</p>	
	 <p>Γ</p>	 <p>Δ</p>	
	 <p>E</p>		

#### 4.1.5 ΔΙΑΔΙΚΑΣΙΕΣ

Η τελευταία προγραμματιστική έννοια που εξετάζεται αναφορικά και με τα δυο τμήματα, κατά τη χρήση του περιβάλλοντος Code.org, είναι η διαδικασία. Στο σχήμα που παρατίθεται αποτυπώνεται η κατανομή των βαθμών, με σκοπό να εντοπιστεί και να χαρτογραφηθεί το πόσο είναι τελικά οι μαθητές αυτοί που καταφέρνουν να φτάσουν στο αντίστοιχο επίπεδο βάση της ΙΑΠ.

Σύμφωνα, λοιπόν, με το σχήμα 4.5 κανένας μαθητής δεν αξιολογείται με το μικρότερο επίπεδο, το πρώτο που είναι και το προδρομικό. Έξι μαθητές, δηλαδή το 16%, φθάνει στο αμέσως επόμενο επίπεδο, το δεύτερο, αυτό της επιμέρους κατανόησης. Οκτώ μαθητές, δηλαδή το 22%, βρίσκεται στο ενδιάμεσο από το δεύτερο και τρίτο επίπεδο. Δεκαέξι μαθητές, ποσοστό 43%, αξιολογούνται στο τρίτο επίπεδο που αναφέρεται στην προσεγγιστική κατανόηση. Τέσσερις μαθητές, ποσοστό που αντιστοιχεί στο 11%, φθάνει στο ενδιάμεσο από το τρίτο και το τέταρτο επίπεδο. Τέλος, τρεις μαθητές, δηλαδή το 8%, βρίσκονται στο τέταρτο επίπεδο, το συνδυαστικό, ενώ κανένας μαθητής δεν αξιολογείται στο ανώτερο επίπεδο, το πέμπτο, αυτό της εκτεταμένης θεώρησης.




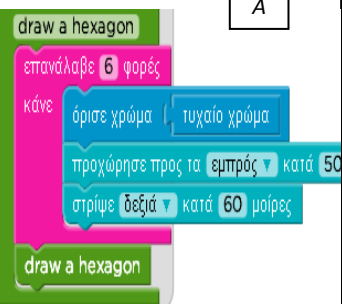
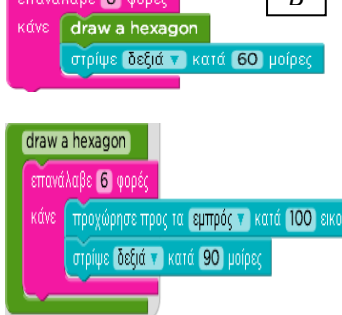

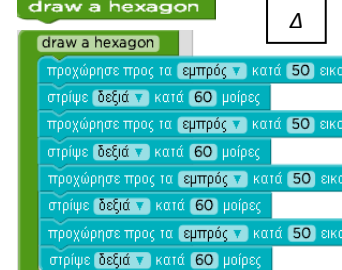
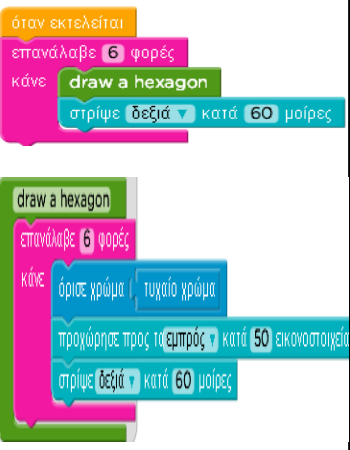
Σχήμα 4.5 Κατανομή μαθητών στις διαδικασίες στο Code.org

Ακολούθως θα εξεταστεί το αν τα δυο τμήματα παρουσιάζουν στατιστική διαφορά κατά την προγραμματιστική έννοια που είναι η διαδικασία. Αρχικά θα ελεγχθεί η υπόθεση κανονικής κατανομής η οποία και δεν ικανοποιείται, αφού το  $p$  είναι μικρότερο από το 0.05 και για τα δυο τμήματα και πιο συγκεκριμένα για το Στ1 το  $p=0.31$  ενώ για το Στ2 το  $p=0.49$ . Αυτός είναι και ο λόγος που θα χρησιμοποιηθεί το τεστ Mann-Whitney U ως η καλύτερη επιλογή. Τα περιγραφικά στοιχεία του τεστ φανερώνουν ότι το Στ1 με (μέσο όρο - median = 2.9; μέση κατάταξη - mean rank = 21) έχει καλύτερη

επίδοση στις διαδικασίες από το Στ2 με (μέσο όρο - median = 2.7; μέση κατάταξη - mean rank =17). Η τιμή του Mann-Whitney U βρέθηκε να είναι μη στατιστικά σημαντική με  $U = 142$ ,  $Z = -.927$ ,  $p > 0.05$  και ίση με 0.3.

Αναφορικά με τις εναλλακτικές ιδέες-παρανοήσεις των μαθητών στην κατασκευή και χρήση των διαδικασιών παρατηρείται ότι οι μαθητές θεωρούν ότι η κατασκευή της διαδικασίας αποτελεί μέρος του κυρίως προγράμματος και γι' αυτό δεν ασχολούνται με το να την καλέσουν – χρησιμοποιήσουν στο κυρίως πρόγραμμα, όπως αποτυπώνεται και στην περίπτωση Α του πίνακα 4.5. Επίσης καλούν μια φορά την διαδικασία και δεν επαναλαμβάνουν την κλήση της στο κυρίως πρόγραμμα, γιατί η διαδικασία για την κατασκευή της χρησιμοποιεί η ίδια την επανάληψη. Θεωρούν, δηλαδή, ότι η κλήση μιας διαδικασίας με την κατασκευή μιας διαδικασίας είναι το ίδιο πράγμα, αντίληψη που καταδεικνύεται και στις περιπτώσεις Γ και Δ. Άλλες εναλλακτικές ιδέες-παρανοήσεις εντοπίζονται και κατά την κατασκευή της διαδικασίας, αφού οι ίδιοι τους δυσκολεύονται να χωρίσουν το κυρίως πρόγραμμα σε επιμέρους επαναλαμβανόμενες διαδικασίες, οι οποίες θα χρησιμοποιηθούν εκ των υστέρων στο κυρίως πρόγραμμα, περιπτώσεις Β και Δ.

Πίνακας 4.5 Παρανοήσεις μαθητών στις διαδικασίες

Διαδικασίες	Εναλλακτικές ιδέες – Συχνά λάθη	Ορθή αντιμετώπιση
Δραστηριότητα	Εναλλακτικές ιδέες – Συχνά λάθη	Ορθή αντιμετώπιση
<p data-bbox="54 1249 376 1350">Κατασκευή έξι εξαγώνων 50 εικονοστοιχείων με γωνία 60 μοιρών</p> 	<div style="display: flex; flex-wrap: wrap;"> <div style="width: 50%;"> <p data-bbox="424 1205 767 1238">A</p>  </div> <div style="width: 50%;"> <p data-bbox="791 1205 1134 1238">B</p>  </div> <div style="width: 50%;"> <p data-bbox="424 1597 767 1630">Γ</p>  </div> <div style="width: 50%;"> <p data-bbox="791 1597 1134 1630">Δ</p>  </div> </div>	

## 4.2. ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΕΝΝΟΙΩΝ ΣΤΟ STARLOGO TNG ΚΑΙ SCRATCH

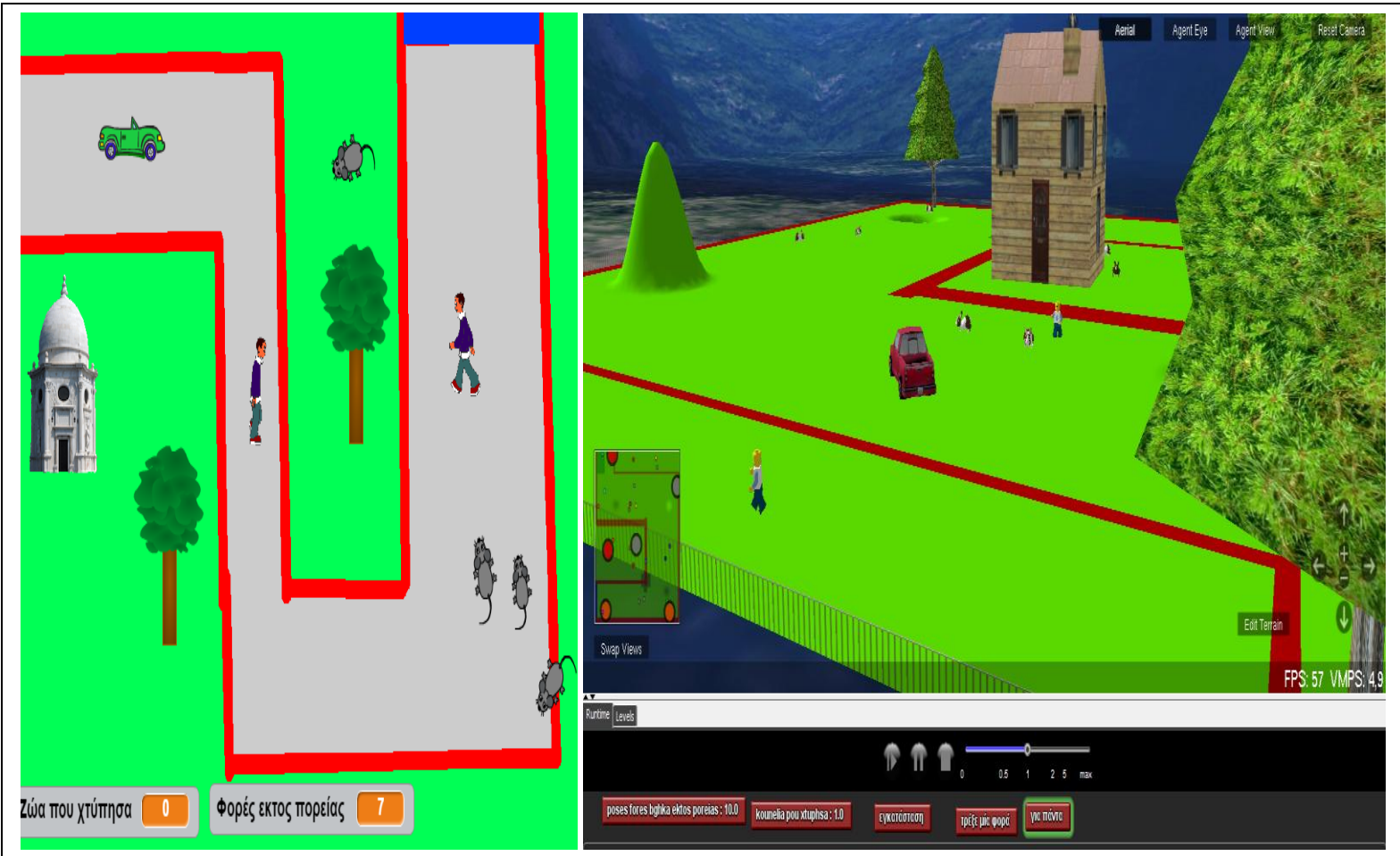
Αντικείμενο της παρούσας ενότητας αποτελεί η επίδοση των μαθητών κατά τη χρήση διαφορετικών ΠΠΑ στις εξεταζόμενες προγραμματιστικές έννοιες. Ένα τμήμα, το ΣΤ1, χρησιμοποίησε το Starlogo TNG και το άλλο τμήμα, το ΣΤ2, χρησιμοποίησε το Scratch, προκειμένου, για να υλοποιηθεί η τελική τους εργασία. Για την ολοκλήρωση της εργασίας χρησιμοποιήθηκαν όλες οι προγραμματιστικές έννοιες που οι μαθητές διδάχτηκαν στο Code.org.

Τα έργα είχαν ως κύριο θέμα την ολοκλήρωση ενός παιχνιδιού και ήταν παρόμοια και στα δυο ΠΠΑ, δηλαδή οι κανόνες του παιχνιδιού που ίσχυαν για το Scratch ίσχυαν εξίσου και για το Starlogo TNG.

Το σενάριο του παιχνιδιού που ξεδιπλωνόταν από τους μαθητές αναφέρεται, εν συντομία, στη μετακίνηση ενός αυτοκινήτου-οχήματος μέσα σε μια οριοθετημένη περιοχή-δρόμο-πίστα και το οποίο θα πρέπει να αποφύγει τόσο τους ανθρώπους όσο και τα ζώα που μετακινούνται μέσα σε αυτό το περιβάλλον. Η νίκη επιτυγχάνεται όταν το αυτοκίνητο καταφέρει να παραμείνει στον δρόμο του - πίστα χωρίς να συγκρουστεί με κάποιον άνθρωπο και με ένα σύνολο - αριθμό, ορισμένο από τον χρήστη, ζώων και εφόσον ακολουθήσει τη διαδρομή που έχει κατασκευαστεί από το χρήστη μέχρι τέλους.

Στη συνέχεια μελετάται και αναλύεται η επίδοση των μαθητών κατά τη χρήση των υπό εξέταση προγραμματιστικών εννοιών που είναι η επανάληψη, η επιλογή, τα γεγονότα, οι μεταβλητές και οι διαδικασίες, τις οποίες οι μαθητές χρησιμοποίησαν, συνδύασαν και ενσωμάτωσαν προκειμένου για την ολοκλήρωση των έργων τους. Ένα στιγμιότυπο από δυο ολοκληρωμένα έργα στα δυο περιβάλλοντα που δημιουργήθηκαν από μαθητές φαίνεται στο σχήμα 4.6.

Τα αποτελέσματα των δυο τμημάτων αναλυτικά παρουσιάζονται στο Παράρτημα Α. Παρακάτω ωστόσο παρουσιάζονται τα στοιχεία και στατιστική ανάλυση, έτσι ώστε να διερευνηθεί το αν η επίδοση των μαθητών στην κάθε προγραμματιστική έννοια διαφέρει κατά την χρήση διαφορετικών ΠΠΑ, όπως είναι το Starlogo TNG και το Scratch. Με το κλείσιμο αυτής της ενότητας θα εξεταστεί, επίσης, και το αν υπάρχει διαφορά στην επίδοση μεταξύ αγοριών και κοριτσιών, για να απαντηθεί έτσι και το τέταρτο ερευνητικό ερώτημα.



Σχήμα 4.6 Ολοκληρωμένα έργα στο Scratch (αριστερά) και στο Starlogo TNG (δεξιά)

#### 4.2.1 ΕΠΑΝΑΛΗΨΗ

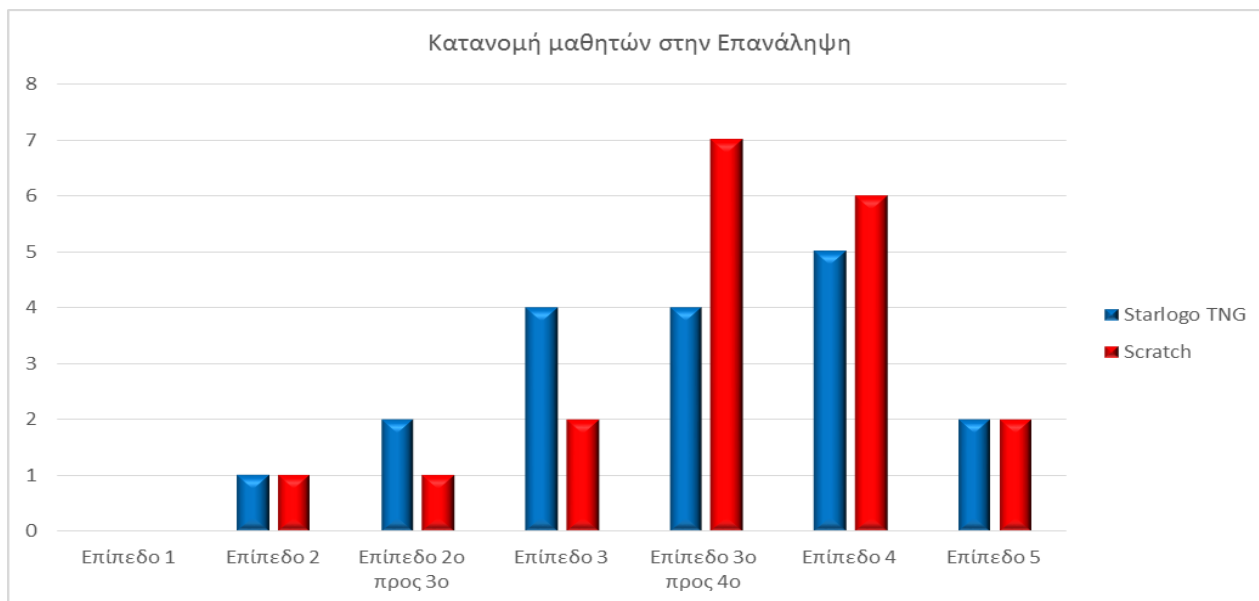
Ξεκινώντας τη μελέτη των προγραμματιστικών εννοιών εξετάζεται αρχικά η επανάληψη διερευνώντας την κατανομή των μαθητών ανά επίπεδο, όπως αυτή έχει προκύψει από την Ιεραρχική Αξιολόγηση Προγραμματισμού κατά τη μετάβασή τους σε διαφορετικά περιβάλλοντα. Ακολούθως θα εξεταστεί αν υπάρχει κάποια στατιστική διαφορά μεταξύ των περιβαλλόντων για αυτή την έννοια.

Η επανάληψη χρησιμοποιήθηκε και στα δυο περιβάλλοντα για την κίνηση του αυτοκινήτου, για την κίνηση των ανθρώπων και τέλος για την κίνηση των ζώων.

Αξίζει να σημειωθεί ότι στην επανάληψη δυο μαθητές από το κάθε περιβάλλον, όπως φαίνεται και στο σχήμα 4.7, φθάνουν στο πέμπτο επίπεδο που είναι αυτό της εκτεταμένης θεώρησης, καθώς και τα δυο περιβάλλοντα προσφέρουν πολλές δυνατότητες συνδυασμού εντολών, καθιστώντας δυνατό να δημιουργηθεί έτσι μια σαφής διαφορά από το τέταρτο επίπεδο, το συνδυαστικό. Από την άλλη πλευρά κανένας μαθητής δεν βρίσκεται στο πρώτο επίπεδο, το προδρομικό, γεγονός που



μπορεί να οφείλεται στο ότι οι μαθητές προέρχονται ήδη από ένα περιβάλλον που έχουν χρησιμοποιήσει την επαναληπτική δομή.



Σχήμα 4.7 Κατανομή μαθητών στην επανάληψη για τα δυο περιβάλλοντα

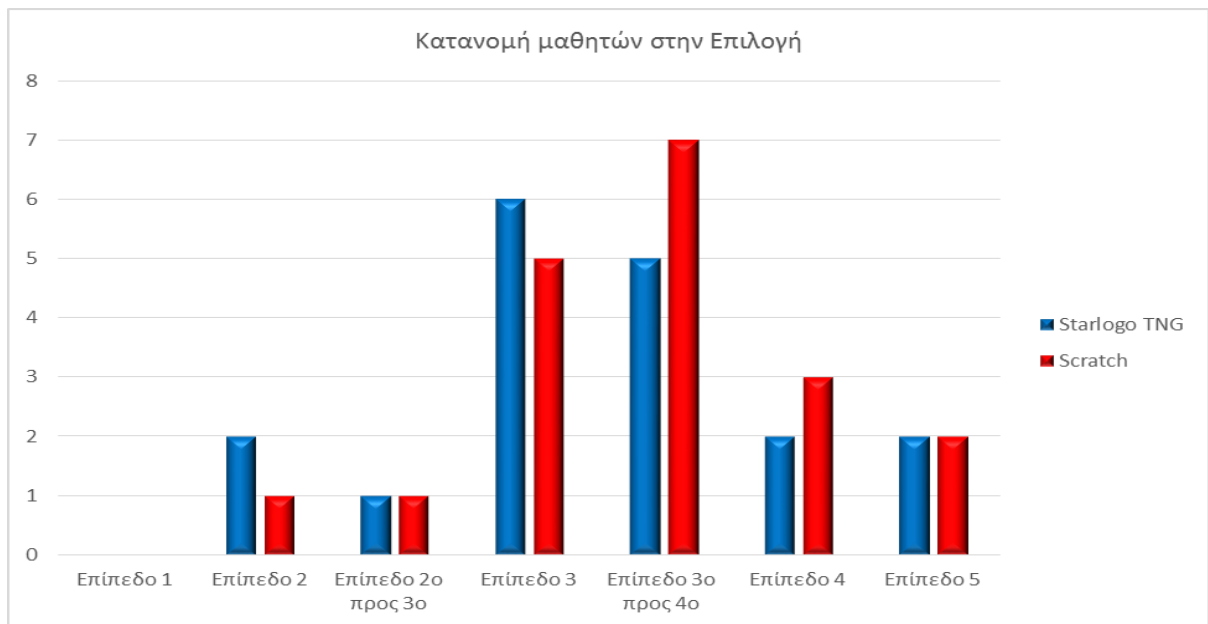
Τα δυο περιβάλλοντα ικανοποιούν την υπόθεση της κανονικής κατανομής αφού το  $p > 0.05$ , για το καθένα από αυτά, σύμφωνα με τους ελέγχους κανονικότητας που πραγματοποιήθηκαν. Πιο συγκεκριμένα το Starlogo TNG έχει τιμή  $p=0.3$  και το Scratch έχει  $p=0.06$ .

Για να εξεταστεί αν η επίδοση των μαθητών κατά την χρήση της επανάληψης εμφανίζει στατιστική διαφορά στα διαφορετικά περιβάλλοντα θα χρησιμοποιηθεί το τεστ ανεξάρτητων δειγμάτων - independent samples t-test λόγω της ύπαρξης κανονικής κατανομής.

Από την εκτέλεση του τεστ διαπιστώνουμε ότι οι μαθητές του Scratch ( $M = 3.63$ ,  $SD = 0.74$ ) παρουσιάζουν βελτιωμένη επίδοση στην επανάληψη από τους μαθητές του Starlogo TNG ( $M = 3.5$ ,  $SD = 0.85$ ) χωρίς όμως να υπάρχει στατιστικά σημαντική διαφορά μεταξύ των δυο περιβαλλόντων αφού το  $t(35) = 0.524$ , και το  $p > 0.05$ .

#### 4.2.2 ΕΠΙΛΟΓΗ

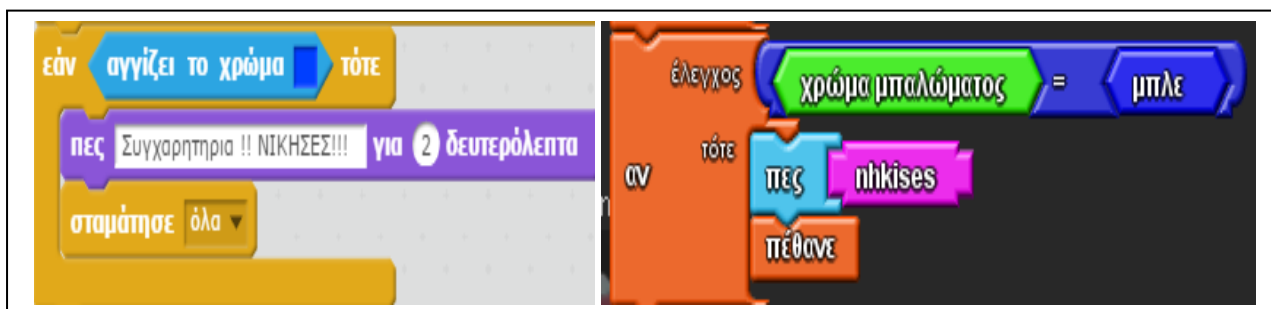
Στο σχήμα 4.8 απεικονίζεται η κατανομή των βαθμών στην επιλογή σύμφωνα με την ΙΑΠ, για το κάθε προγραμματιστικό περιβάλλον αρχαρίων που αξιοποιήθηκε.



Σχήμα 4.8 Κατανομή μαθητών στην επιλογή για τα δυο περιβάλλοντα

Οι μαθητές χρησιμοποίησαν την επιλογή στα έργα τους για να ελέγξουν τις διάφορες καταστάσεις που θα μπορούσε να βρεθεί το αυτοκίνητο τους, με σκοπό να τις αντιμετωπίσουν. Αυτές θέτουν τα θεμέλια για να διαμορφώσουν τους κανόνες του παιχνιδιού οι οποίοι και αναπτύχθηκαν μετά από συζήτηση και επιλογή των ίδιων. Κάποιοι από τους ελέγχους αναφέρονται παρακάτω:

- Αν το αυτοκίνητο βγει εκτός πορείας – εκτός δρόμου, τότε θα πρέπει να το επαναφέρουν.
- Αν το αυτοκίνητο ακουμπήσει – συγκρουστεί με κάποιον άνθρωπο. Τότε το πρόγραμμα να τερματίζει και ο παίχτης να χάνει.
- Αν το αυτοκίνητο έχει ακουμπήσει – συγκρουστεί με κάποιο ζώο, τότε ο μετρητής – ο καταγραφέας του αριθμού των ζώων αυξάνεται, αλλιώς θα παραμένει στάσιμος στον ίδιο αριθμό.
- Αν ο παραπάνω μετρητής ξεπεράσει κάποιον αριθμό που διαμορφώνεται από τον κάθε μαθητή, τότε ενημερωτικά μηνύματα διαβιβάζονται στον παίχτη ότι έχει χτυπήσει πολλά ζώα.
- Αν το αυτοκίνητο καταφέρει να φτάσει στο τερματισμό, θα σηματοδοτείται και η νίκη του παίκτη. Ένα σύντομο στιγμιότυπο αυτού του γεγονότος απεικονίζεται στο σχήμα 4.9 και για τα δυο περιβάλλοντα.

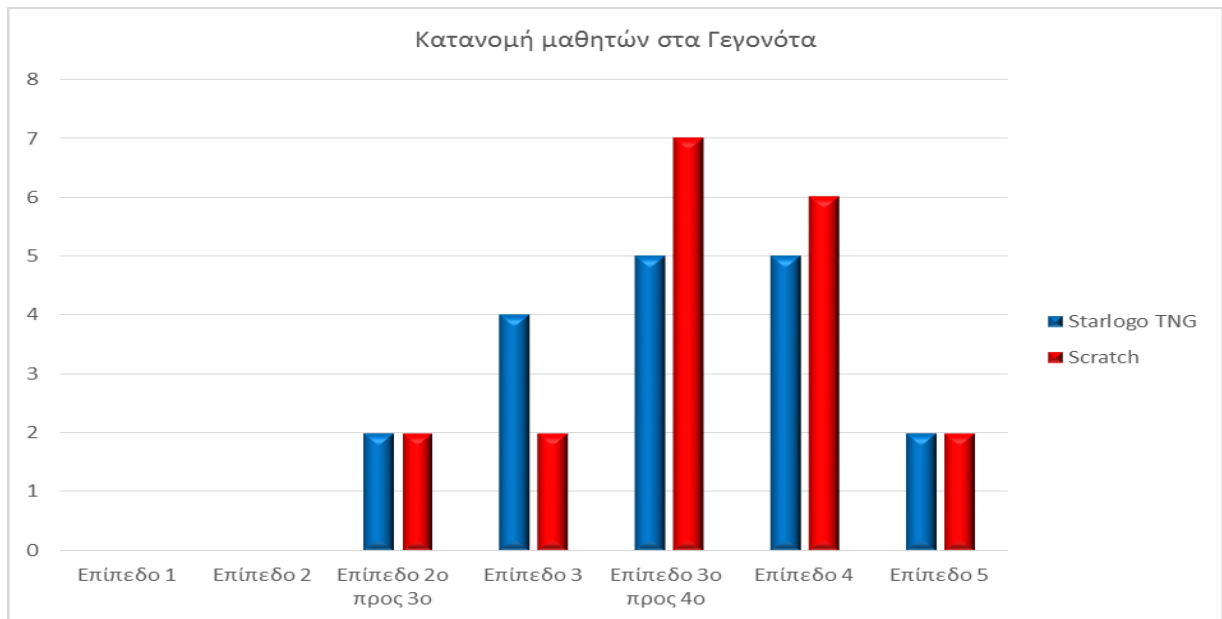


Σχήμα 4.9 Έλεγχος νίκης και από τα δυο περιβάλλοντα

Προκειμένου να εξεταστεί αν τα δυο τμήματα παρουσιάζουν στατιστική διαφορά στους μέσους όρους της επίδοσης τους, μεταξύ των δύο προγραμματιστικών περιβαλλόντων κατά την προγραμματική έννοια που είναι η επιλογή, αρχικά θα ελεγχθεί η υπόθεση κανονικής κατανομής, η οποία και δεν ικανοποιείται αφού το  $p$  είναι μικρότερο από το 0.05. Αυτός είναι και ο λόγος που θα χρησιμοποιηθεί το τεστ Mann-Whitney U ως η καλύτερη επιλογή. Τα περιγραφικά στοιχεία του τεστ δείχνουν ότι στο Scratch με (μέσο όρο - median = 3.47; μέση κατάταξη - mean rank = 20.16) οι μαθητές παρουσιάζουν καλύτερη επίδοση στην επιλογή έναντι του Starlogo TNG με (μέσο όρο - median = 3.3; μέση κατάταξη - mean rank = 17.7). Η δίπλευρη τιμή του Mann-Whitney U βρέθηκε να είναι μη στατιστικά σημαντική με  $U = 149$ ,  $Z = -0.690$ ,  $p > 0.05$  και ίση με 0.4.

#### 4.2.3 ΓΕΓΟΝΟΤΑ

Η επίδοση των μαθητών κατά την ανάπτυξη των γεγονότων στα δυο περιβάλλοντα απεικονίζεται στο σχήμα 4.10. Διαπιστώνεται ότι κανένας μαθητής δεν βρίσκεται στο δεύτερο επίπεδο αυτό της επιμέρους κατανόησης και πως το μεγαλύτερο μέρος των μαθητών βρίσκεται στο ενδιάμεσο επίπεδο από το τρίτο και το τέταρτο και στο αμέσως επόμενο επίπεδο που είναι το συνδυαστικό και για τα δύο περιβάλλοντα.



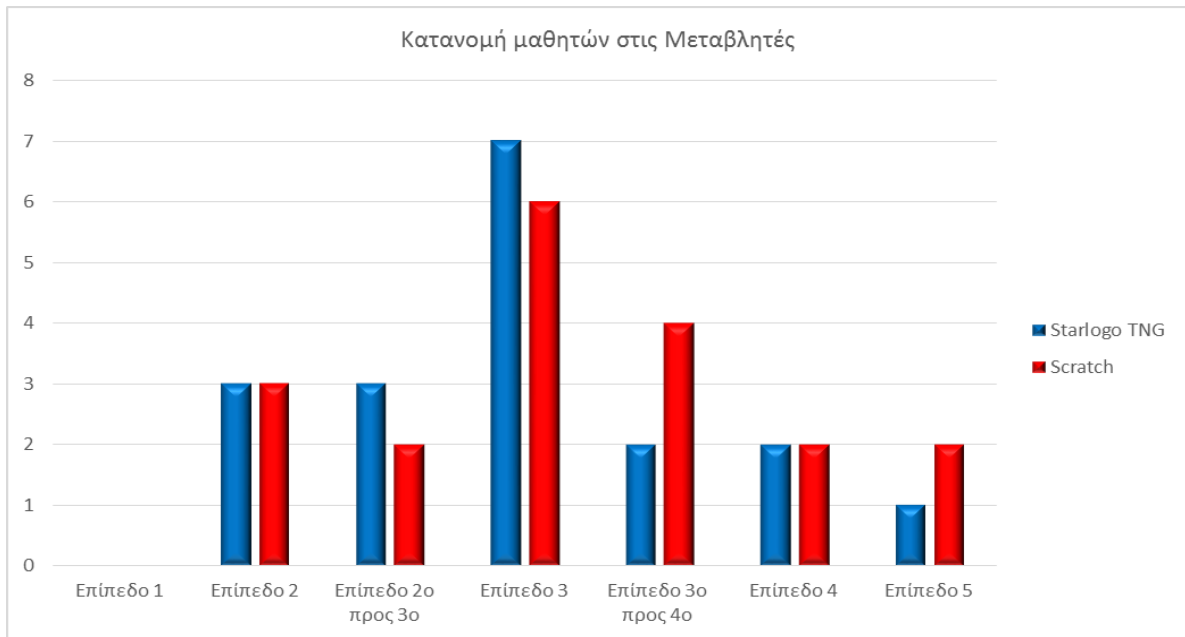
Σχήμα 4.10 Κατανομή μαθητών στα γεγονότα για τα δυο περιβάλλοντα

Οι μαθητές χρησιμοποίησαν τα γεγονότα για την κίνηση του αυτοκινήτου προς όλες τις κατευθύνσεις με το συνδυασμό των βελών του πληκτρολογίου είτε με όποιον άλλο συνδυασμό πλήκτρων που ο κάθε μαθητής επιθυμούσε.

Τα δυο περιβάλλοντα ικανοποιούν την υπόθεση της κανονικής κατανομής, αφού το  $p > 0.05$  για το καθένα από αυτά, σύμφωνα με τους ελέγχους κανονικότητας που πραγματοποιήθηκαν. Για να εξεταστεί αν η επίδοση των μαθητών στα γεγονότα εμφανίζει στατιστική διαφορά στα δυο περιβάλλοντα θα χρησιμοποιηθεί το τεστ ανεξάρτητων δειγμάτων - independent samples t-test ως την καλύτερη επιλογή αφού η κατανομή είναι κανονική. Από την εκτέλεση του τεστ διαπιστώνουμε ότι οι μαθητές του Scratch ( $M = 3.65$ ,  $SD = 0.66$ ) παρουσιάζουν βελτιωμένη επίδοση στα γεγονότα από τους μαθητές του Starlogo TNG ( $M = 3.5$ ,  $SD = 0.71$ ) χωρίς όμως να υπάρχει στατιστικά σημαντική διαφορά μεταξύ των δυο περιβαλλόντων αφού το  $t(35) = 0.329$ , και το  $p > 0.05$ .

#### 4.2.4 ΜΕΤΑΒΛΗΤΕΣ

Η κατανομή των βαθμών σε αυτά τα περιβάλλοντα παρουσιάζεται στο σχήμα 4.11. Σε αυτό διακρίνεται ότι το επίπεδο στο οποίο συγκεντρώνονται οι περισσότεροι μαθητές είναι το τρίτο, αυτό της προσεγγιστικής κατανόησης.



Σχήμα 4.11 Κατανομή μαθητών στις μεταβλητές για τα δυο περιβάλλοντα

Οι μαθητές αξιοποίησαν τις μεταβλητές ως δυο μετρητές και στα δυο περιβάλλοντα. Ο πρώτος μετρητής κατέγραφε τον αριθμό από τα ζώα που χτυπούσαν με το αυτοκίνητο και έχει ήδη αναφερθεί παραπάνω. Είχε μηδενική αρχική τιμή και αυξάνονταν κάθε φορά που ο παίχτης θα συγκρουόταν με κάποιο ζώο. Ο δεύτερος μετρητής κατέγραφε τις φορές που το αυτοκίνητο βγήκε εκτός πορείας. Η αρχική του τιμή ορίζονταν από τον μαθητή σε κάποιον αριθμό της επιλογής του, θα έπρεπε να ήταν διαφορετικός του μηδενός και θα μειωνόταν κατά μια μονάδα, κάθε φορά που συνέβαινε αυτό το γεγονός. Στην ουσία οι φορές που έβγαινε εκτός πορείας το αυτοκίνητο αντιμετωπίστηκαν ως οι προσπάθειες που παρέχονταν σε κάποιον παίχτη, για να περατώσει - τερματίσει το παιχνίδι. Αν ο μετρητής αυτός έφτανε στο μηδέν, θα έπρεπε επίσης να τερματίσει το παιχνίδι. Ένα στιγμιότυπο από την τελευταία μεταβλητή που περιγράφει την διαδικασία κατά την οποία μειώνεται η τιμή της για τα δυο περιβάλλοντα εμφανίζεται στο σχήμα 4.12.

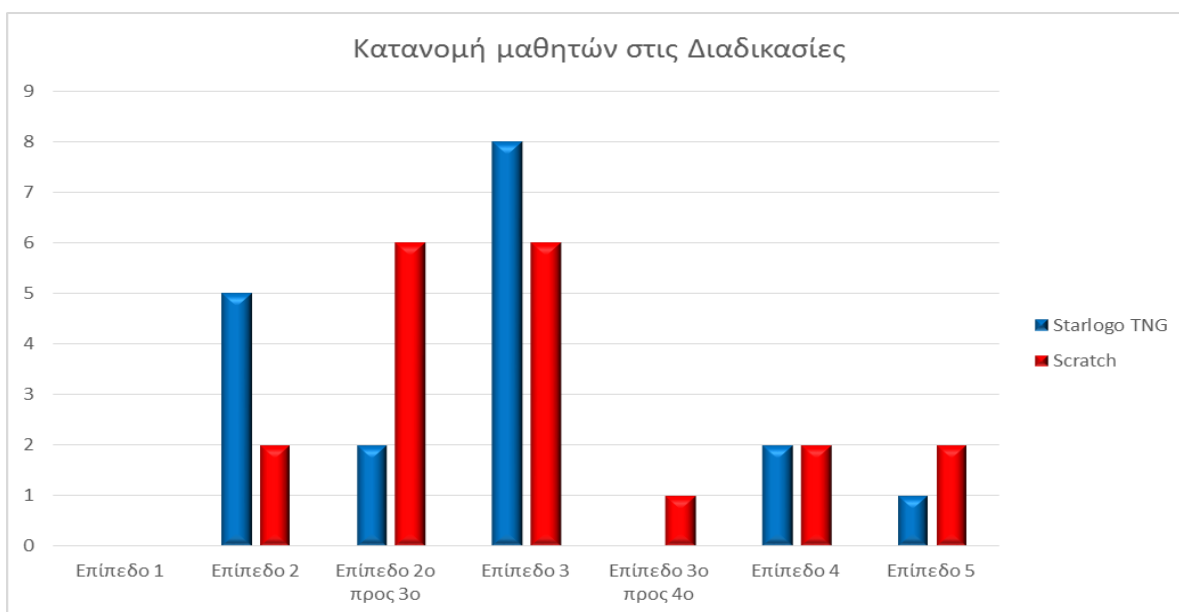


Σχήμα 4.12 Μείωση τιμής μιας μεταβλητής από τα δυο περιβάλλοντα

Για να εξεταστεί το αν τα δυο τμήματα παρουσιάζουν στατιστική διαφορά ως προς την επίδοσή τους στα δυο περιβάλλοντα, κατά την προγραμματική έννοια που είναι η μεταβλητή, αρχικά θα ελεγχθεί η υπόθεση κανονικής κατανομής η οποία και δεν ικανοποιείται, αφού το  $p$  είναι μικρότερο από το 0.05. Αυτός είναι και ο λόγος που θα χρησιμοποιηθεί το τεστ Mann-Whitney U ως η καλύτερη επιλογή. Τα περιγραφικά στοιχεία του τεστ καταδεικνύουν ότι στο Scratch με (μέσο όρο - median = 3.2; μέση κατάταξη - mean rank = 20), οι μαθητές εμφανίζουν καλύτερη επίδοση στις μεταβλητές έναντι του Starlogo TNG με (μέσο όρο - median = 3; μέση κατάταξη - mean rank = 16,5). Η δίπλευρη τιμή του Mann-Whitney U βρέθηκε να είναι μη στατιστικά σημαντική με ( $U = 150, Z = -0.730, p > 0.05$ ).

#### 4.2.5 ΔΙΑΔΙΚΑΣΙΕΣ

Παρατηρείται ότι από το δεύτερο, δηλαδή αυτό της επιμέρους κατανόησης, μέχρι το τρίτο επίπεδο, της προσεγγιστικής κατανόησης, συγκεντρώνεται το μεγαλύτερο μέρος - σύνολο μαθητών σε σχέση με τα υπόλοιπα επίπεδα (σχήμα 4.13).



Σχήμα 4.13 Κατανομή μαθητών στις διαδικασίες για τα δυο περιβάλλοντα

Οι μαθητές χρησιμοποίησαν τις διαδικασίες και στα δυο περιβάλλοντα για να κατασκευάσουν την κίνηση των ανθρώπων άλλα και των ζώων επίσης. Ένα παράδειγμα για την κίνηση του ανθρώπου εμφανίζεται στο σχήμα 4.14 και για τα δυο περιβάλλοντα. Πιο συγκεκριμένα, απεικονίζεται η κατασκευή άλλα και η κλήση μιας διαδικασίας.



Σχήμα 4.14 Δημιουργία και κλήση διαδικασίας για την κίνηση ενός χαρακτήρα και στα δυο περιβάλλοντα

Για το αν τα δυο τμήματα παρουσιάζουν διαφορά ως προς την επίδοσή τους στα δυο περιβάλλοντα στις διαδικασίες, ελέγχθηκε η υπόθεση κανονικής κατανομής, η οποία και δεν ικανοποιείται, αφού το  $p$  είναι μικρότερο από το 0.05. Για το λόγο αυτό και θα χρησιμοποιηθεί το τεστ Mann-Whitney U. Τα περιγραφικά στοιχεία του τεστ αποτυπώνουν ότι στο Scratch με (μέσο όρο - median = 3; μέση κατάταξη - mean rank = 19), οι μαθητές έχουν καλύτερη επίδοση στις διαδικασίες έναντι του Starlogo TNG με (μέσο όρο - median = 2.8; μέση κατάταξη - mean rank = 17,5). Η δίπλευρη τιμή του Mann-Whitney U βρέθηκε να είναι μη στατιστικά σημαντική με  $U = 155$ ,  $Z = -0.500$  και  $p > 0.05$ .

Συγκεκριρωατικώα, με όλουα τουα παραπάνω ελέγχουα απαντήθηκε και το τρίτο ερευνητικό ερώτημα που αφορά στο αν η επίδοση των μαθητών στην κάθε προγραμματιστική έννοια διαφέρει κατά τη χρήση διαφορετικών ΠΠΑ. Στη συνέχεια θα διερευνηθεί το τέταρτο ερευνητικό ερώτημα που αφορά στη διάσταση του φύλου σχετικά με την επίδοση μεταξύ αγοριών και κοριτσιών, για να ολοκληρωθεί έτσι και η συγκεκριμένη ενότητα.

Για την απάντησή του τέταρτου ερευνητικού ερωτήματος υπολογίστηκαν οι μέσοι όροι από όλες τις παραπάνω προγραμματιστικές έννοιες και στη συνέχεια διεξήχθη το τεστ, προκειμένου να εξετάσει το αν υπάρχει κάποια διαφορά στην επίδοση των δυο φύλων. Τα 20 αγόρια με μέσο όρο 3.4 και τυπική απόκλιση 0.78 εμφανίζουν ελάχιστη βελτίωση, σε σχέση με τα 17 κορίτσια με μέσο όρο 3.2 και τυπική απόκλιση 0.68 όπως φαίνεται από τα περιγραφικά στοιχεία του τεστ. Μολαταύτα δεν παρουσιάζουν καμία στατιστική διαφορά οι επιδόσεις τους αφού το  $t(35) = 139$  και  $p > 0.05$ .

### 4.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ

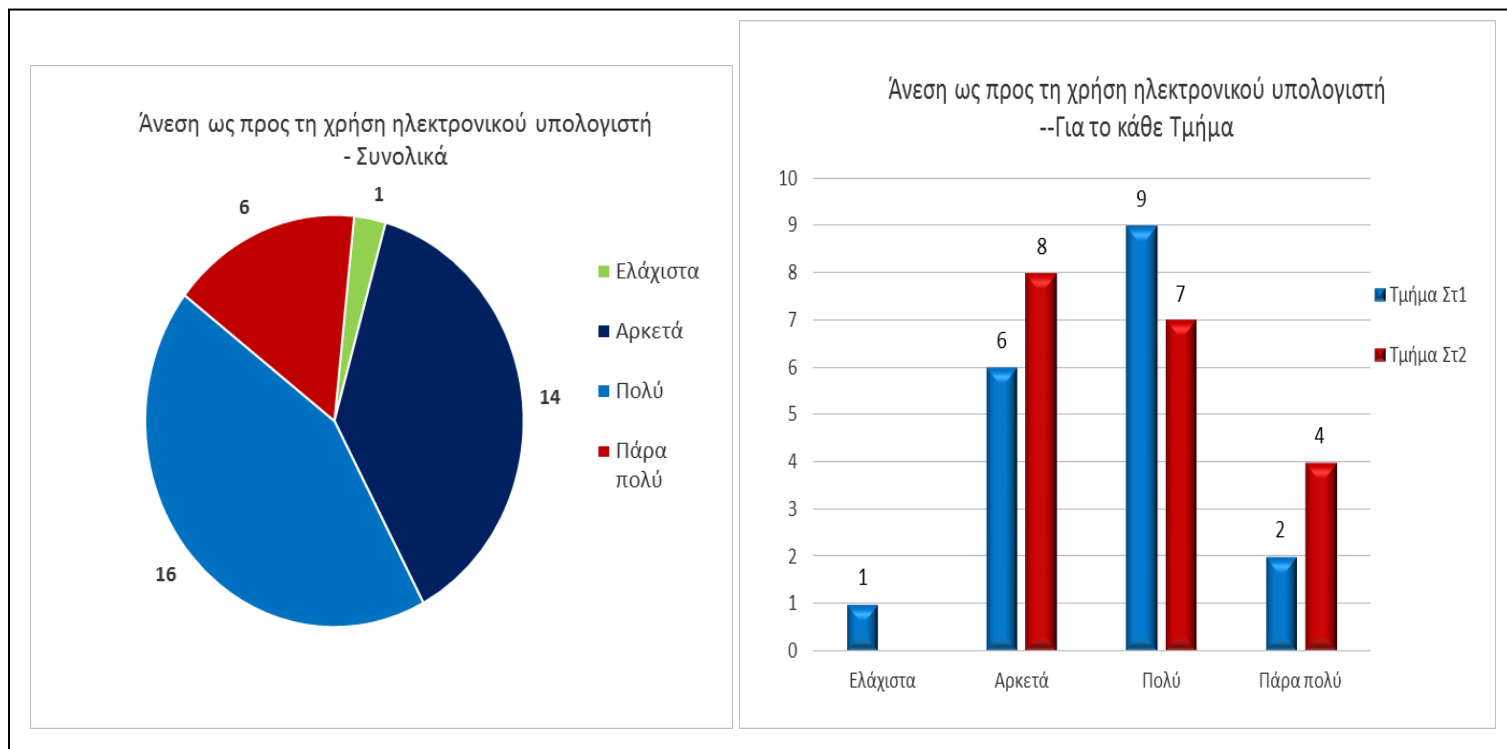
Σε αυτήν την ενότητα παρουσιάζονται τα αποτελέσματα από τις απαντήσεις των μαθητών που προέκυψαν από τα ερωτηματολόγια τα οποία και διαμορφώθηκαν για να καλύψουν συνολικά 14 ερωτήσεις, οι οποίες στη συνέχεια ομαδοποιήθηκαν σε τρεις κατηγορίες.

1. Τις απόψεις των μαθητών για την χρήση του Ηλεκτρονικού Υπολογιστή.
2. Τις απόψεις των μαθητών για τον Προγραμματισμό και
3. Τις απόψεις των μαθητών για τα Προγραμματιστικά Περιβάλλοντα Αρχαρίων που χρησιμοποίησαν.

Το ερωτηματολόγιο ελέγχθηκε ως προς την αξιοπιστία του, σχετικά με το πώς αξιολογούν οι μαθητές τα ΠΠΑ, με βάση τον δείκτη Cronbach's Alpha και διαπιστώθηκε ότι ήταν αξιόπιστο ( $\alpha = 0.86$ ).

#### 4.3.1 ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΗΝ ΧΡΗΣΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ

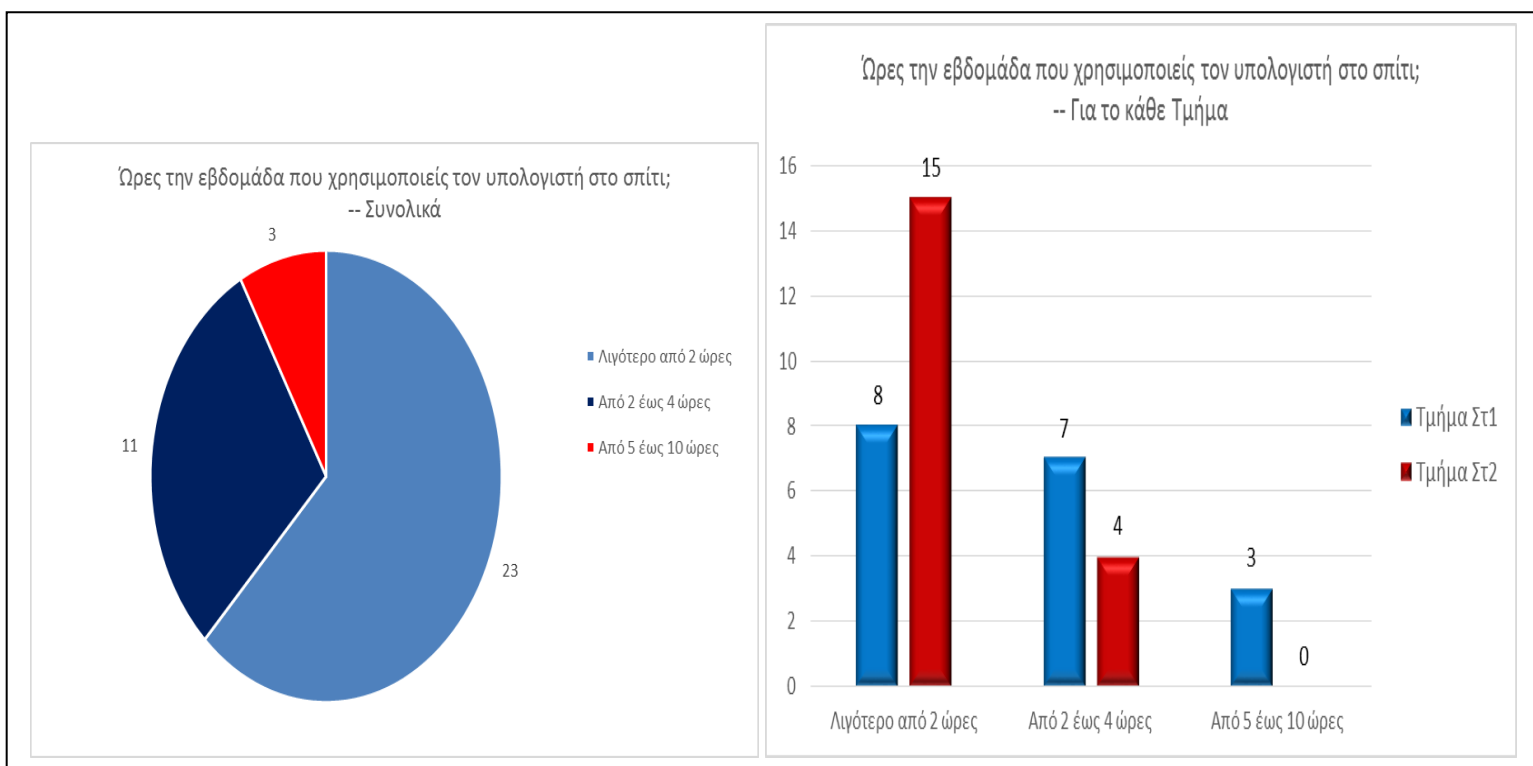
Οι απόψεις των μαθητών για την χρήση του Ηλεκτρονικού Υπολογιστή αναφορικά με την Ερώτηση 1 σχετικά με την εξοικείωση που έχουν στη χρήση του ηλεκτρονικού υπολογιστή για τα δυο τμήματα αποτυπώνονται στο σχήμα 4.15. Από την εξέτασή του διαπιστώνεται ότι οι περισσότεροι μαθητές δηλώνουν πως είναι αρκετά έως πολύ εξοικειωμένοι με τους ηλεκτρονικούς υπολογιστές.



Σχήμα 4.15 Άνεση ως προς την χρήση ηλεκτρονικού υπολογιστή συνολικά (δεξιά) και για το κάθε τμήμα (αριστερά)

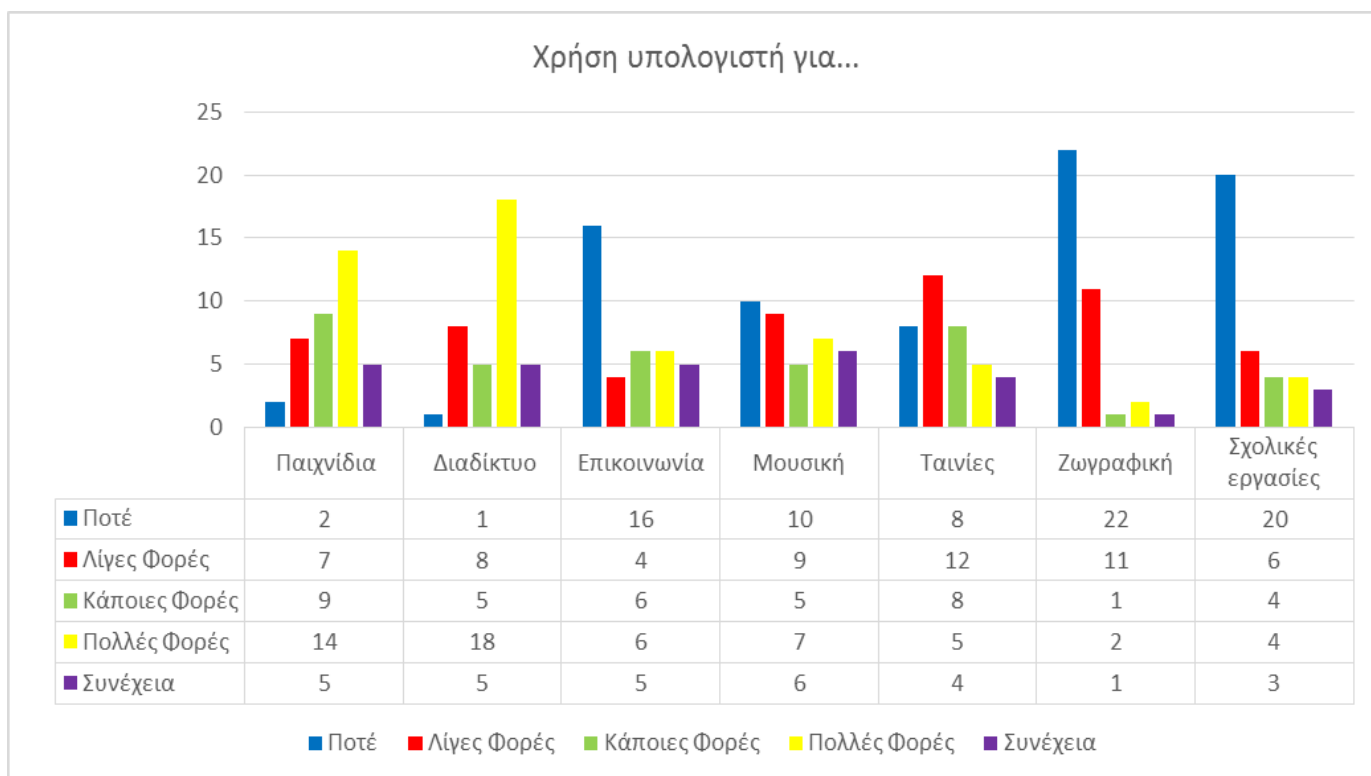


Οι απαντήσεις των μαθητών στην ερώτηση 2 ως προς τις ώρες που χρησιμοποιούν τον υπολογιστή στο σπίτι παρουσιάζεται στο σχήμα 4.16 . Διαπιστώνεται ότι η μεγαλύτερη μερίδα-ποσοστό, δηλαδή οι 23 συνολικά από τους 37 μαθητές, δηλώνει ότι χρησιμοποιεί τον ηλεκτρονικό υπολογιστή λιγότερο από δύο ώρες την εβδομάδα, ακολουθούν 11 μαθητές που υποστηρίζουν ότι τον χρησιμοποιούν από δυο έως τέσσερις ώρες και 3 μαθητές που τον χρησιμοποιούν από πέντε έως δέκα ώρες εβδομαδιαίως.



Σχήμα 4.16 Ωρες την εβδομάδα που χρησιμοποιούν τον υπολογιστή στο σπίτι συνολικά (αριστερά) και για το κάθε τμήμα (δεξιά)

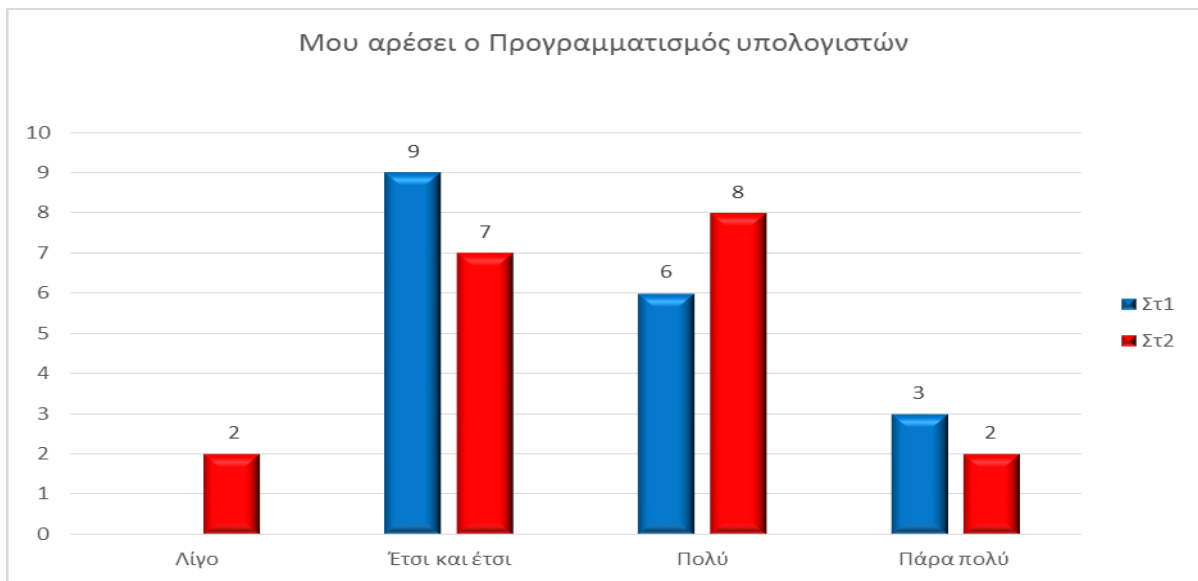
Η ερώτηση 3 που αποτελεί και την τελευταία της πρώτης κατηγορίας απεικονίζει για ποιο λόγο και πόσο συχνά οι μαθητές χρησιμοποιούν τον υπολογιστή. Από το σχήμα 4.17 συνάγεται ότι στο σύνολό τους οι μαθητές ασχολούνται και χρησιμοποιούν λιγότερο τον υπολογιστή για σχολικές εργασίες, για να ζωγραφίζουν και για να επικοινωνούν ενώ τις περισσότερες φορές τον χρησιμοποιούν για να ψάχνουν διάφορες πληροφορίες στο διαδίκτυο και για να παίζουν παιχνίδια.



*Σχήμα 4.17 Χρησιμοποιείς τον υπολογιστή για ...*

#### 4.3.2 ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

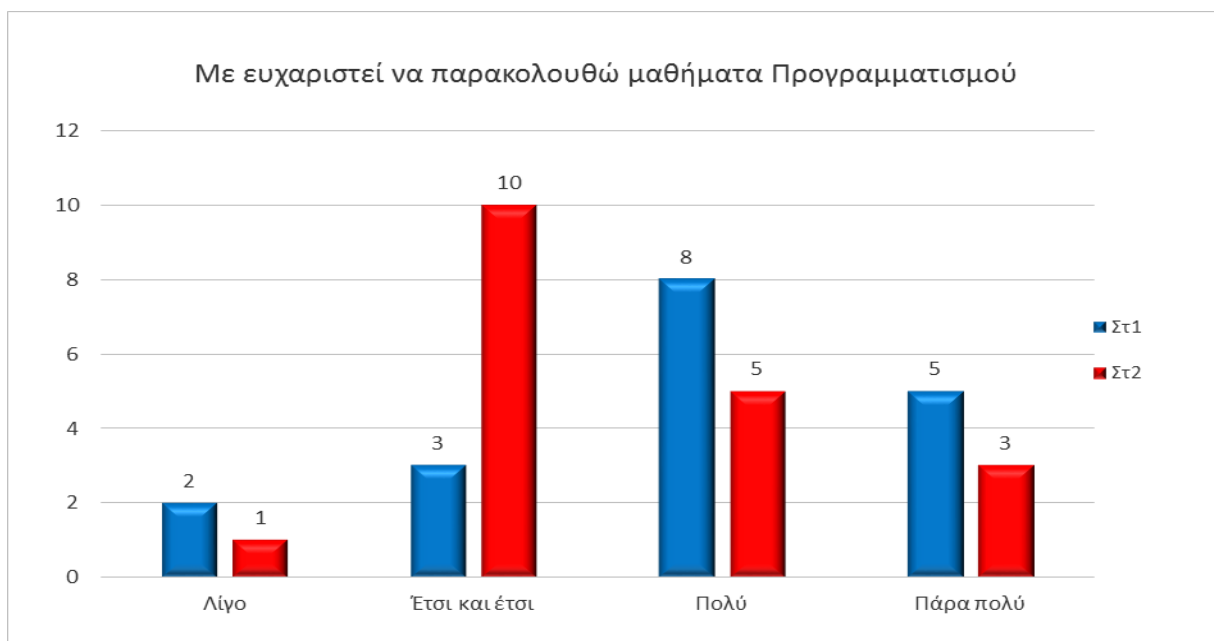
Ακολουθεί η δεύτερη κατηγορία ερωτήσεων που αναφέρεται στις απαντήσεις των μαθητών για τον προγραμματισμό και πιο συγκεκριμένα στα αποτελέσματα της ερώτησης 4 που είναι «Μου αρέσει ο προγραμματισμός υπολογιστών». Από τις απαντήσεις του προκύπτει λοιπόν το σχήμα 4.18 το οποίο καταδεικνύει πως κανένας μαθητής δεν διάλεξε την επιλογή «Καθόλου», και από τα δυο τμήματα, ενώ οι περισσότερες απαντήσεις διαπιστώνεται ότι κατανέμονται από την μέση και σε περισσότερο θετικές απαντήσεις παρά αρνητικές. Αυτό είναι ιδιαίτερα ενθαρρυντικό, αφού διαπιστώνεται ότι τα ΠΠΑ που χρησιμοποιήθηκαν είχαν μια θετική επίδραση στους μαθητές και των δύο τμημάτων αναφορικά με το αν τους αρέσει ο προγραμματισμός.



Σχήμα 4.18 Μου αρέσει ο προγραμματισμός υπολογιστών

Προκειμένου να εξεταστεί το αν υπάρχει στατιστική διαφορά στις απόψεις των δυο τμημάτων πάνω σε αυτήν ερώτηση, αρχικά ελέγχθηκε η υπόθεση κανονικής κατανομής η οποία δεν ικανοποιείται, αφού το  $p$  είναι μικρότερο από το 0.05 και για τα δυο τμήματα. Πιο συγκεκριμένα στο Στ1 το  $p=0.01$  και στο Στ2  $p=0.023$  γι' αυτό το λόγο που θα χρησιμοποιηθεί το τεστ Mann-Whitney U. Τα περιγραφικά στοιχεία του τεστ έδειξαν ότι στους μαθητές του τμήματος Στ1 αρέσει περισσότερο ο προγραμματισμός υπολογιστών (μέσο όρο - median = 3.67; μέση κατάταξη - mean rank = 19.58) σε σχέση με τους οι μαθητές του τμήματος Στ2 με (μέσο όρο - median = 3.53; μέση κατάταξη - mean rank = 18.4) χωρίς όμως να υπάρχει κάποια στατιστικά σημαντική διαφορά, αφού η δίπλευρη τιμή του Mann-Whitney U βρέθηκε να είναι με  $U = 160$ ,  $Z = -0.343$  και  $p > 0.05$ .

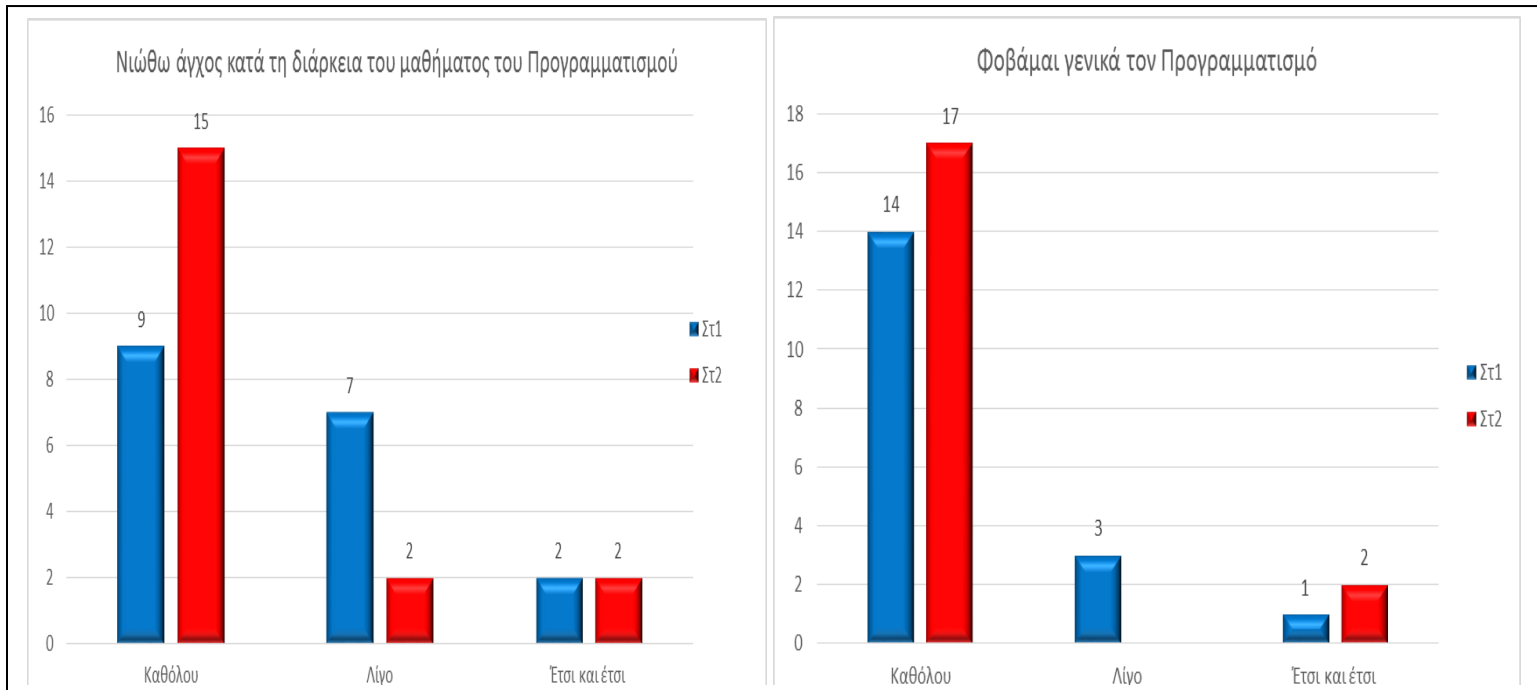
Στην ερώτηση 5 «Με ευχαριστεί να παρακολουθώ μαθήματα Προγραμματισμού», όπως αυτές παρουσιάζονται στο σχήμα 4.19, συμπεραίνεται ότι οι περισσότερες απαντήσεις των μαθητών είναι θετικές, γεγονός που καθιστά αντιληπτό το ότι τα ΠΠΑ που επιλέχτηκαν έκαναν το μάθημα του προγραμματισμού ευχάριστο και ελκυστικό. Επιπρόσθετα, κανένας μαθητής, και από τα δύο τμήματα, δεν διάλεξε την επιλογή Καθόλου.



*Σχήμα 4.19 Με ευχαριστεί να παρακολουθώ μαθήματα Προγραμματισμού*

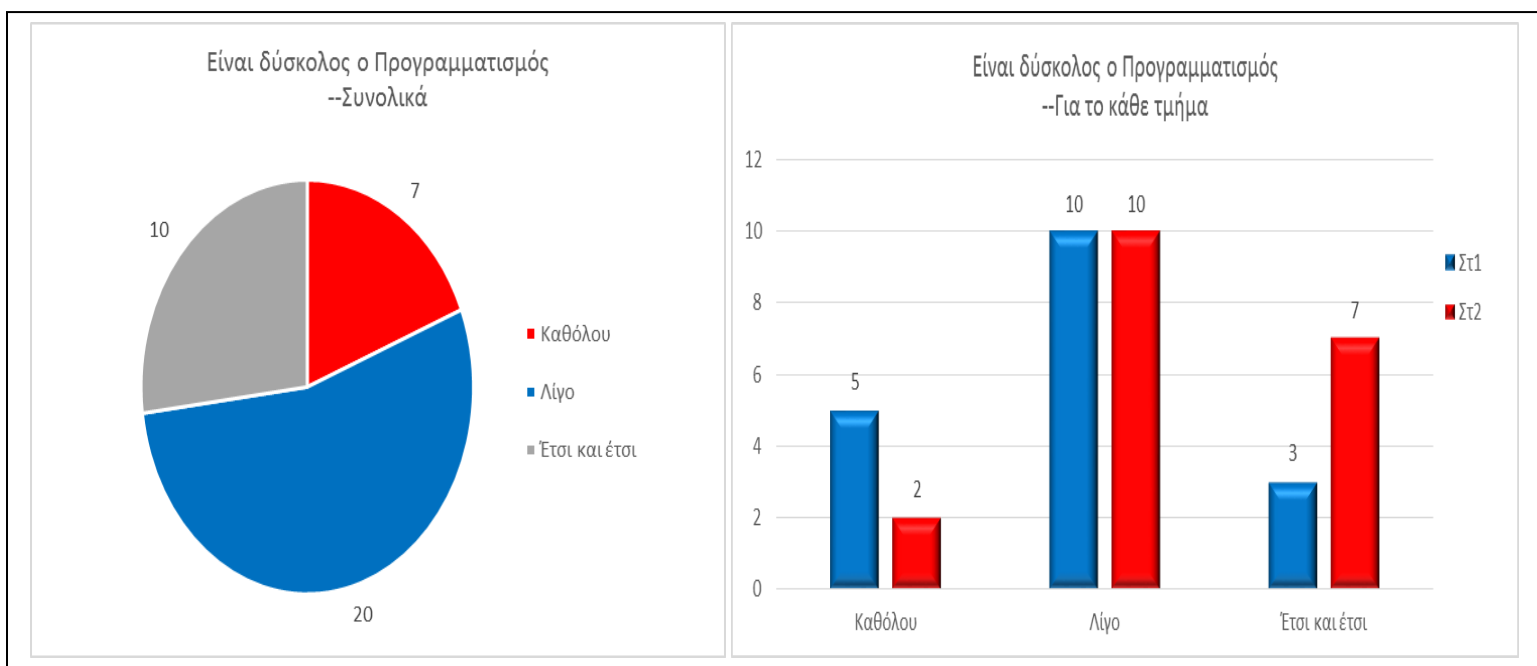
Για να ελέγξουμε αν υπάρχει στατιστική διαφορά στις απαντήσεις της παραπάνω ερώτησης θα εκτελεστεί το τεστ Mann-Whitney U, αφού δεν ικανοποιείται η υπόθεση κανονικής κατανομής, καθώς το  $p$  είναι μικρότερο από το 0.05. Ειδικότερα, στο Στ1 το  $p=0.011$  και στο Στ2  $p=0.003$ . Τα περιγραφικά στοιχεία του τεστ καταδεικνύουν ότι τους μαθητές του τμήματος Στ1 τους ευχαριστεί να παρακολουθούν περισσότερα μαθήματα προγραμματισμού (μέση κατάταξη - mean rank = 21.42), σε σχέση με τους οι μαθητές του τμήματος Στ2 με (μέση κατάταξη - mean rank =16.71) χωρίς όμως να υπάρχει κάποια στατιστικά σημαντική διαφορά, αφού η τιμή του Mann-Whitney U βρέθηκε να είναι με  $U = 127$ ,  $Z = -1.391$  και  $p = 0.16$ .

Τα αποτελέσματα των ερωτήσεων 6 «Νιώθω άγχος κατά τη διάρκεια του μαθήματος του Προγραμματισμού» και 7 «Φοβάμαι γενικά τον Προγραμματισμό» παρουσιάζονται στο σχήμα 4.20. Από τα γραφήματα που ακολουθούν συνάγεται ότι οι μαθητές και των δυο τμημάτων στη συντριπτική τους πλειοψηφία απάντησαν αρνητικά στο αν διακατέχονται από άγχος και φόβο που μπορεί να διακατέχονται, όταν προγραμματίζουν. Απαντήσεις οι οποίες αποδεικνύουν με τη σειρά τους ότι τα ΠΠΑ που χρησιμοποιήθηκαν δεν δημιούργησαν αίσθημα φόβου άλλα και άγχους κατά τη διάρκεια της εκπαιδευτικής διαδικασίας.



Σχήμα 4.20 Νιώθω άγχος κατά τη διάρκεια του μαθήματος του Προγραμματισμού(αριστερά) και φοβάμαι γενικά τον Προγραμματισμό (δεξιά)

Οι απαντήσεις στην τελευταία ερώτηση, την 8<sup>η</sup> της δεύτερης κατηγορίας ερωτήσεων, «Είναι δύσκολος ο Προγραμματισμός», αποτυπώνονται στο σχήμα 4.21. Η πλειοψηφία των μαθητών, 20 από τους 37, επέλεξε την απάντηση «Λίγο» για να περιγράψει το αν «Είναι δύσκολος ο Προγραμματισμός» ενώ ακολουθεί η απάντηση «Έτσι κι έτσι», 10 στους 37. Οι απαντήσεις αυτές είναι ενθαρρυντικές, καθώς οι μαθητές δεν είχαν ξανασχοληθεί με τον προγραμματισμό. Η ενασχόλησή τους με αυτόν μέσω των ΠΠΑ που επιλέχθηκαν έδειξε να μην τον αντιμετωπίζουν με δυσκολία.



Σχήμα 4.21 Είναι δύσκολος ο Προγραμματισμός συνολικά (αριστερά) και για το κάθε τμήμα (δεξιά)

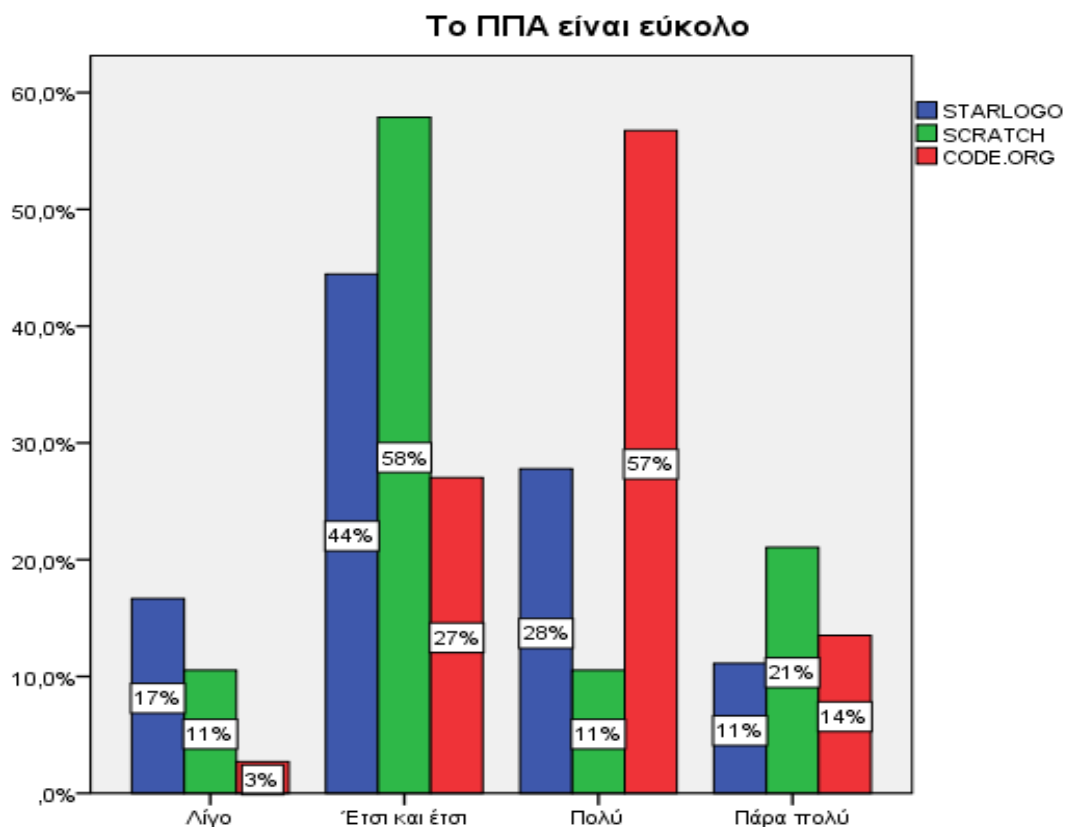
### 4.3.3 ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ ΑΡΧΑΡΙΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΣΑΝ

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι απαντήσεις των μαθητών σχετικά με την τρίτη κατηγορία του ερωτηματολογίου που αφορούν στις απόψεις τους για τα προγραμματιστικά περιβάλλοντα αρχαρίων που χρησιμοποίησαν.

Οι 37 απαντήσεις των μαθητών για το Code.org, οι 18 απαντήσεις για το Starlogo TNG και οι 19 για το Scratch δημιουργούν τις τρεις ομάδες των ΠΠΑ που θα εξεταστούν σχετικά με την ευκολία που παρουσιάζουν. Για να δημιουργηθούν αυτά τα στοιχεία χρησιμοποιήθηκαν οι ερωτήσεις εννιά και έντεκα και από τα δυο ερωτηματολόγια. Το σχήμα 4.22 απεικονίζει τα ποσοστά των απαντήσεων της κλίμακας Likert σχετικά με το πόσο εύκολο ήταν το κάθε ΠΠΑ.

Αξίζει να σημειωθεί ότι κανένας μαθητής για κανένα περιβάλλον δεν έδωσε την απάντηση «καθόλου», που είναι και η πιο αρνητική τιμή της κλίμακας Likert, για αυτήν την ερώτηση.

Τα αποτελέσματα από το τεστ Kruskal-Wallis κατέδειξαν ότι δεν υπήρχε στατιστική σημαντική διαφορά μεταξύ των τριών περιβαλλόντων αφού το  $p > 0.05$ . Η επιλογή αυτού του τεστ έγινε, αφού δεν ικανοποιείται η υπόθεση κανονικότητας για κανένα περιβάλλον και πιο συγκεκριμένα για το Starlogo TNG το  $p=0.3$ , για το Scratch  $p=0.001$  και τέλος για το Code.org το  $p= 0.000$ . Από τα αποτελέσματα του τεστ σχετικά με τη μέση κατάταξη-mean rank ως προς την ευκολία έχουμε 31.33 για το Starlogo TNG, 32.26 για το Scratch και τέλος 43.19 για το Code.org πράγμα που σημαίνει ότι στους μαθητές το Code.org τους φάνηκε το πιο εύκολο. Η σύγκριση μεταξύ Scratch και Starlogo TNG που ενδιαφέρει περισσότερο την εργασία δείχνει με πολύ μικρή διαφορά το Scratch να τους φάνηκε πιο εύκολο.

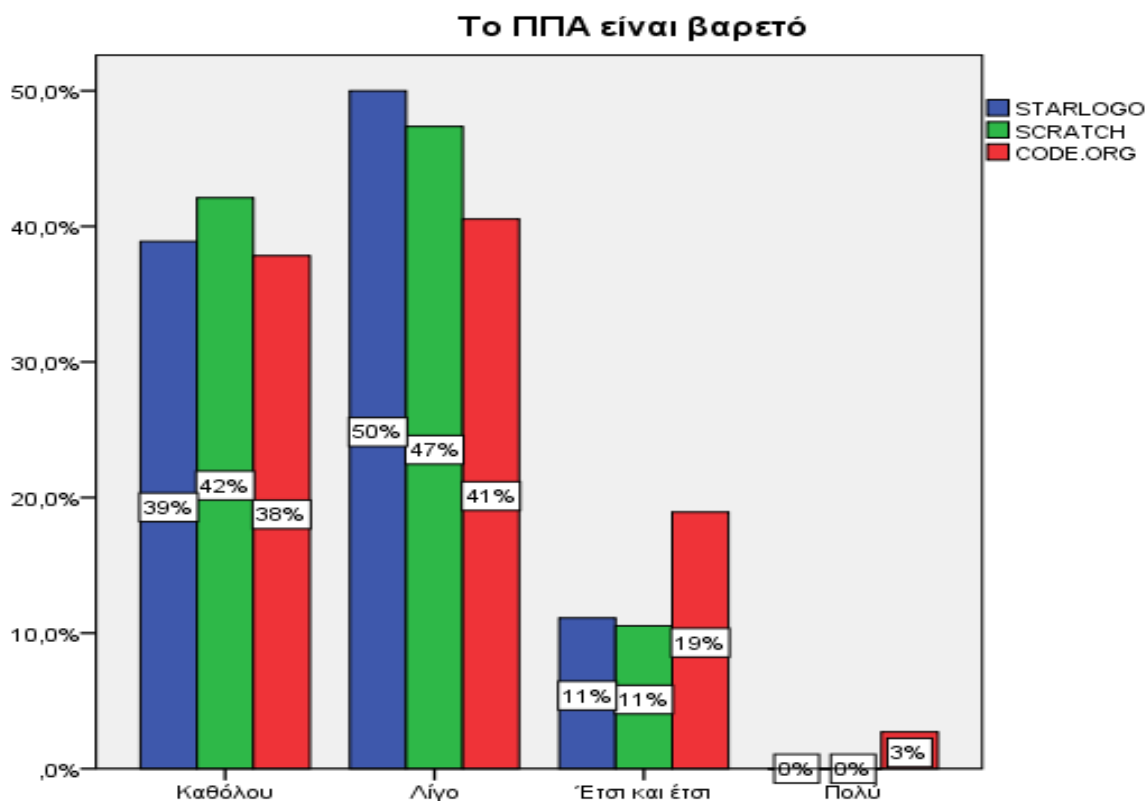


*Σχήμα 4.22 Πόσο εύκολο είναι το κάθε ΠΠΑ*

Παρακάτω εξετάζονται οι απαντήσεις που έδωσαν οι μαθητές για την περίπτωση όπου το περιβάλλον είναι βαρετό. Το Σχήμα 4.23 απεικονίζει τα ποσοστά των απαντήσεων της κλίμακας Likert για το πόσο βαρετό ήταν το κάθε ΠΠΑ. Από το σχήμα γίνεται αντιληπτό ότι οι περισσότεροι μαθητές έδωσαν την απάντηση «καθόλου» και «λίγο» για το σύνολο των ΠΠΑ που χρησιμοποίησαν στην ερώτηση αν το περιβάλλον τους φάνηκε βαρετό.

Παρακάτω θα εξεταστεί αν υπάρχει κάποια στατιστική διαφορά μεταξύ όλων των παραπάνω περιβαλλόντων σχετικά με το πόσο βαρετά ήταν. Η πραγματοποίηση του ελέγχου κανονικότητας κατέδειξε πως δεν ισχύει η υπόθεση αυτή για κανένα περιβάλλον και πιο συγκεκριμένα για το Starlogo TNG το  $p=0.001$ , για το Scratch το  $p=0.001$  και τέλος για το Code.org το  $p=0.000$ . Αφού η κατανομή δεν είναι κανονική θα αξιοποιηθεί το τεστ Kruskal-Wallis H, ως η καλύτερη επιλογή. Από τη μέση κατάταξη – mean rank συνάγεται ότι οι μαθητές που χρησιμοποίησαν το Code.org το θεώρησαν το πιο βαρετό. Η σύγκριση μεταξύ Scratch και Starlogo TNG που ενδιαφέρει περισσότερο την εργασία δείχνει ότι το Scratch τους φάνηκε το λιγότερο βαρετό περιβάλλον με ελάχιστη διαφορά σε σχέση με το Starlogo TNG. Πιο συγκεκριμένα ο μέσος όρος - median = 1.7 και η μέση κατάταξη -

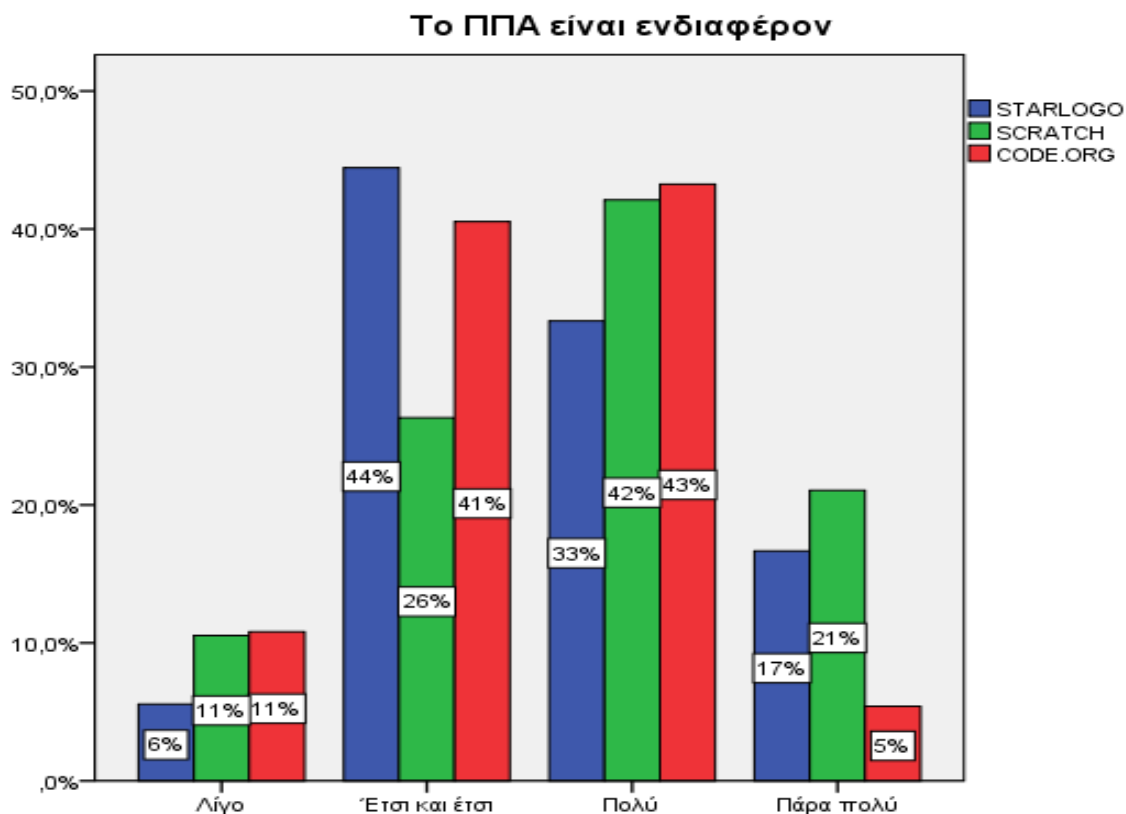
mean rank = 36.39 για το Starlogo TNG, ο μέσος όρος =1.68 και η μέση κατάταξη = 35.26 για το Scratch και τέλος ο μέσος όρος =1.86 και η μέση κατάταξη=39.19 για το Code.org. Μένει να αναφερθεί αν υπάρχει σημαντική στατιστική διαφορά μεταξύ αυτών των τριών περιβαλλόντων. Τα αποτελέσματα του τεστ αποδεικνύεται ότι δεν υπάρχει σημαντική στατιστική διαφορά μεταξύ αυτών των τριών ΠΠΑ που εξετάζονται, σύμφωνα με τις απαντήσεις που έδωσαν οι μαθητές, αφού το  $p > 0.05$ .



Σχήμα 4.23 Πόσο βαρετό είναι το κάθε ΠΠΑ

Αναφορικά με το ποιο περιβάλλον τους φάνηκε πιο ενδιαφέρον, αξίζει να σημειωθεί πως κανένας μαθητής δεν έδωσε την απάντηση «καθόλου» για κάποιο από τα περιβάλλοντα, εκδηλώνοντας έτσι το πόσο πολύ αυτά τους προκάλεσαν το ενδιαφέρον. Μένει να εξεταστεί αν υπάρχει κάποια στατιστική διαφορά μεταξύ όλων των παραπάνω περιβαλλόντων για το βαθμό που τους προσέλκυσε το ενδιαφέρον. Από την εξέταση του ελέγχου κανονικότητας συνάγεται ότι η κατανομή δεν είναι κανονική για κανένα περιβάλλον, καθώς το  $p$  είναι μικρότερο από 0.05 και για τα τρία με τιμές για το Starlogo TNG το  $p=0.014$ , για το Scratch το  $p=0.024$  και για το Code.org το  $p=0.00$ .

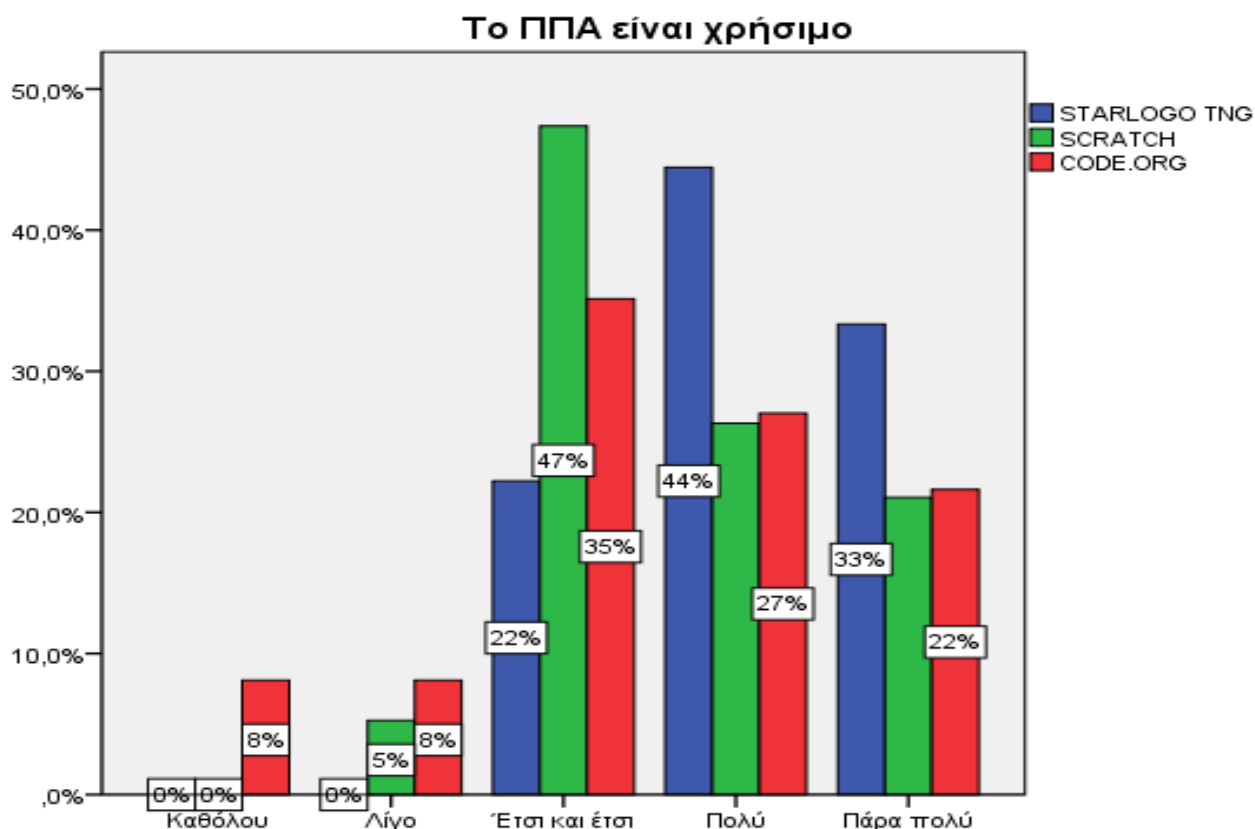




*Σχήμα 4.24 Πόσο ενδιαφέρον είναι το κάθε ΠΠΑ*

Για το λόγο αυτό θα αξιοποιηθεί το τεστ Kruskal-Wallis H, ως η καλύτερη λύση, αφού δεν ικανοποιείται η υπόθεση κανονικής κατανομής. Από τα στοιχεία του τεστ και τις συγκρίσεις των δύο περιβαλλόντων, εξάγεται το συμπέρασμα πως από τα δυο περιβάλλοντα που προέχουν περισσότερο την εργασία, δηλαδή το Starlogo TNG και το Scratch, οι μαθητές επέλεξαν το Scratch ως το πιο ενδιαφέρον. Ακολουθεί το Starlogo TNG και το Code.org. Ειδικότερα, η μέση κατάταξη-mean rank για το Starlogo TNG είναι 38.28, για το Scratch 42.08 και για το Code.org είναι 34.77. Τα αποτελέσματα του τεστ για την ύπαρξη σημαντικής στατιστικής διαφοράς στις απαντήσεις των μαθητών και για τα τρία περιβάλλοντα φανερώνουν ότι δεν υπάρχει αφού το  $p > 0.05$ .

Στη συνέχεια θα εξεταστεί και η τελευταία ερώτηση για το πόσο χρήσιμο φάνηκε στους μαθητές το κάθε ΠΠΑ. Στο σχήμα 4.25 αποτυπώνεται πως οι περισσότεροι από αυτούς έχουν μια θετική στάση ως προς τον βαθμό που θεωρούν τα ΠΠΑ χρήσιμα. Για να εξεταστεί αν υπάρχει σημαντική στατιστική διαφορά μεταξύ όλων των παραπάνω περιβαλλόντων ως προς την χρησιμότητα θα εφαρμοστεί το τεστ Kruskal-Wallis H, καθώς κανένα από αυτά δεν ικανοποιεί την υπόθεση της κανονικής κατανομής γιατί το  $p < 0.05$



Σχήμα 4.25 Πόσο χρήσιμο είναι το κάθε ΠΠΑ

Από τη μέση κατάταξη-mean rank αυτού του πίνακα συμπεραίνεται ότι το περιβάλλον που οι μαθητές θεωρούν πιο χρήσιμο είναι το Starlogo TNG με 46.64, ακολουθεί το Scratch με 35.87 και τελευταίο έρχεται το Code.org 34.38 με πάρα πολύ μικρή διαφορά και τιμή.

Τα αποτελέσματα του τεστ σχετικά με το αν υπάρχει σημαντική στατιστική διαφορά στις απαντήσεις των μαθητών και για τα τρία περιβάλλοντα καταδεικνύουν ότι δεν υπάρχει τέτοια, γιατί το  $p > 0.05$ .

Σε αυτό το σημείο θα διερευνηθούν από κοινού οι απαντήσεις των μαθητών για το ποιο περιβάλλον τους βοήθησε α) να κατανοήσουν καλύτερα τι είναι ο προγραμματισμός, β) να κατανοήσουν τις προγραμματιστικές έννοιες που διδάχθηκαν, γ) να εντοπίσουν και να διορθώσουν τα λάθη που μπορεί να έκαναν, δ) να δημιουργήσουν ενδιαφέροντα παιχνίδια, ε) να διασκεδάσουν και ζ) να πειραματιστούν.

Για να δημιουργηθούν αυτά τα στοιχεία χρησιμοποιήθηκαν οι ερωτήσεις 10 και 12 και από τα δυο ερωτηματολόγια. Όλες οι παραπάνω ερωτήσεις θα αντιμετωπιστούν ομαδικά – όλες μαζί ως μια

προσπάθεια για μια συγκεντρωτική διερεύνηση και μια προσέγγιση προς ένα ολοκληρωμένο συμπέρασμα.

Από την εξέταση του ελέγχου κανονικότητας συνάγεται ότι η κατανομή δεν είναι κανονική για κανένα, καθώς για όλες τις περιπτώσεις το  $p < 0.05$ . Ακολούθως για να ερευνηθεί αν υπάρχει σημαντική στατιστική διαφορά για την κάθε απάντηση των μαθητών για το κάθε περιβάλλον θα χρησιμοποιηθεί το μη παραμετρικό τεστ Kruskal-Wallis H.

Από τη μέση κατάταξη-mean rank μπορούν να αναφερθούν για την κάθε ερώτηση σε ότι αφορά την σύγκριση των περιβαλλόντων τα εξής. Οι μαθητές θεωρούν ότι το Scratch, με τιμή 41.61, τους βοήθησε να κατανοήσουν καλύτερα τι είναι ο προγραμματισμός σε σχέση με το Starlogo TNG με τιμή 38.00 και τέλος το Code.org με τιμή 35.15. Επίσης το Scratch τους βοήθησε σε μεγαλύτερο βαθμό να κατανοήσουν τις προγραμματιστικές έννοιες που διδάχθηκαν λαμβάνοντας τιμή ίση με 42.26, σε σχέση με το 39.08 του Starlogo TNG και το 34.28 του Code.org. Τέλος, αναφορικά με τον εντοπισμό και τη διόρθωση των λαθών οι μαθητές φαίνεται να προτιμούν σε μεγαλύτερο βαθμό το Scratch, με τιμή 37.68 από αυτούς που χρησιμοποίησαν το Starlogo TNG με τιμή 36.56.

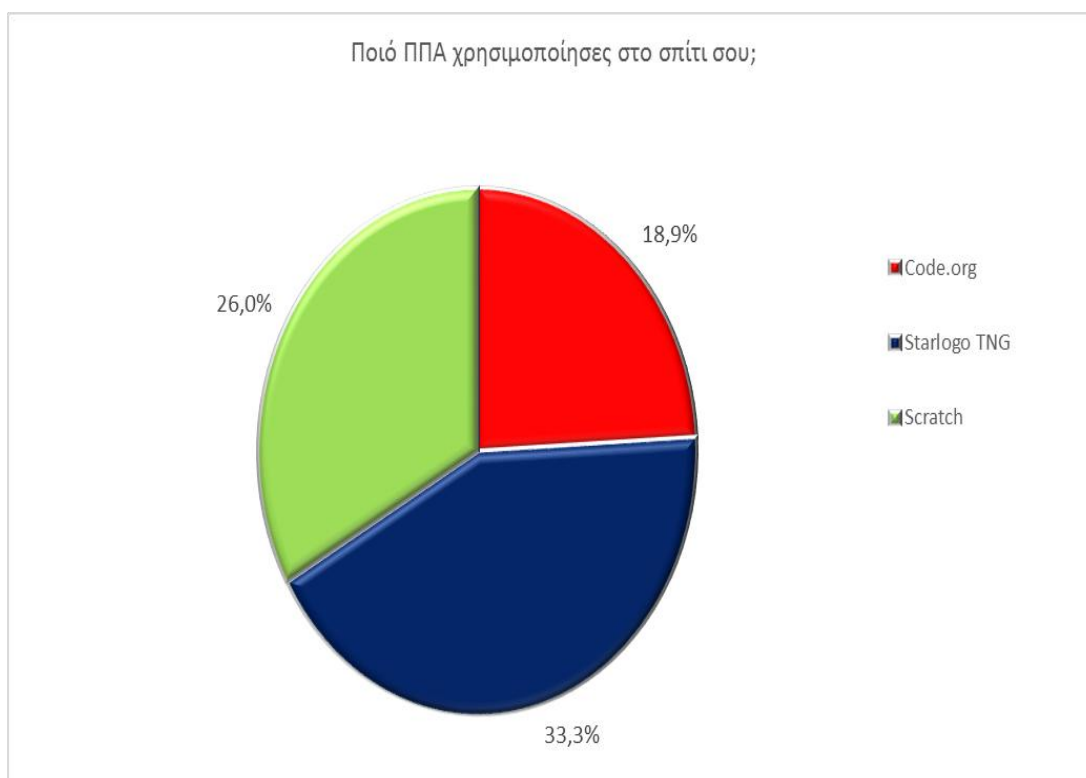
Αντίθετα οι μαθητές που χρησιμοποίησαν το Starlogo TNG φαίνεται να υποστηρίζουν ότι τους βοήθησε να δημιουργήσουν ενδιαφέροντα παιχνίδια, με τιμή που αντιστοιχεί στο 42.33, σε σχέση με το Scratch που λαμβάνει τιμή 38.47 και το Code.org με 34.65. Επίσης τους βοήθησε να διασκεδάσουν, αφού το Starlogo TNG αποτυπώνεται με τιμή 41.11 σε σχέση με το Scratch που έχει τιμή 37.63 και το Code.org με 35.68, αλλά και να πειραματιστούν περισσότερο από ότι με το Scratch που αντιστοιχεί σε τιμή 35.29 σε σχέση με το 37.94 του Starlogo TNG.

Τέλος, όλοι οι μαθητές που χρησιμοποίησαν το Code.org φαίνεται να ωφελήθηκαν από αυτό περισσότερο από τα άλλα δυο περιβάλλοντα στο να εντοπίσουν και να διορθώσουν τα λάθη που μπορεί να έκαναν, αλλά και στο να πειραματιστούν.

Από τα αποτελέσματα του τεστ, τα οποία εντοπίζουν αν υπάρχει σημαντική στατιστική διαφορά στις απαντήσεις των παραπάνω ερωτήσεων των μαθητών και για τα τρία περιβάλλοντα, καταδεικνύεται ότι δεν υπάρχει καμία στατιστική διαφορά για καμιά ερώτηση άλλα και για κανένα περιβάλλον, αφού το  $p > 0.05$ . Πιο συγκεκριμένα, στην πρώτη ερώτηση που αφορά ποιο ΠΠΑ τους βοήθησε να κατανοήσουν καλύτερα τι είναι ο προγραμματισμός το  $p=0.516$ . Στη δεύτερη ερώτηση που αναφέρεται στο ΠΠΑ που τους διευκόλυε να κατανοήσουν τις προγραμματιστικές έννοιες που διδάχθηκαν, η τιμή του  $p = 0.308$ . Στην τρίτη ερώτηση σχετικά με τον εντοπισμό και τη διόρθωση των λαθών το  $p=0.9$ . Στην τέταρτη αναφορικά με τη δημιουργία παιχνιδιών το  $p=0.405$ . Στις

τελευταίες δυο που αναφέρονται στην διασκέδαση και τον πειραματισμό το  $p=0.64$  και  $p=0.86$  αντίστοιχα.

Ολοκληρώνοντας με το ερωτηματολόγιο, απομένει η τελευταία ερώτηση με την οποία εξετάζεται το ποσοστό των μαθητών που χρησιμοποίησε το κάθε ΠΠΑ στο σπίτι του. Από το σχήμα 4.26 συνάγεται ότι ένα 18,9% των μαθητών χρησιμοποίησαν το Code.org σπίτι τους, ένα 26% χρησιμοποίησαν το Scratch και ένα 33,3% το Starlogo TNG. Το υπόλοιπο 21,8 δεν χρησιμοποίησε κανένα προγραμματιστικό περιβάλλον στο σπίτι του. Αξίζει να επισημανθεί ότι σε κανέναν μαθητή δεν έγινε υπόδειξη για να χρησιμοποιήσει συγκεκριμένο περιβάλλον και πως οι μαθητές από μόνοι τους - οικειοθελώς πήραν την απόφαση και την πρωτοβουλία να το πράξουν.



Σχήμα 4.26 Ποιό ΠΠΑ χρησιμοποίησαν σπίτι τους

## 5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι σημερινοί μαθητές, ως ενεργά μέλη μιας κοινωνίας της γνώσης, έχουν ανάγκη να υποστηριχθούν για να αναπτύξουν γνώσεις και δεξιότητες ανώτερου επιπέδου. Ο Προγραμματισμός υπολογιστών αποτελεί ένα ισχυρό μέσο για την ανάπτυξη τέτοιων δεξιοτήτων υψηλού επιπέδου. Η ανάγκη της διδασκαλίας του μπορεί να οδηγήσει σε σημαντικές γνωστικές αλλαγές μέσα από την εγκαθίδρυση νέων τρόπων αντίληψης, μοντελοποίησης, επεξεργασίας και επίλυσης προβλημάτων. Από την παρούσα εργασία έγινε αντιληπτό ότι ο προγραμματισμός είναι αναγκαίος, σίγουρα όμως δεν αποτελεί ένα εύκολο αντικείμενο για τους μαθητές. Η προσέγγιση βασικών προγραμματιστικών εννοιών και δομών φαίνεται να τους δυσκολεύει και να καθιστά το έργο του εκπαιδευτικού ακόμη πιο απαιτητικό αναφορικά με τη μεταφορά της γνώσης.

Ο εκπαιδευτικός είναι ανάγκη να εντοπίσει και να εφαρμόσει τα κατάλληλα εργαλεία, ώστε να κάνει αποδοτικότερο το έργο του, καθώς με αυτόν τον τρόπο θα καταστεί εφικτό από τη μια να μειώσει το γνωστικό φορτίο και από την άλλη να αυξήσει την παιδαγωγική αξία. Σε αυτό το πεδίο φιλοδοξεί να προσφέρει και η παρούσα εργασία ως ένα βοήθημα στα χέρια κάθε εκπαιδευτικού.

Σε αυτήν εντοπίζονται, παρουσιάζονται και συγκρίνονται βάσει τεχνολογικών και παιδαγωγικών χαρακτηριστικών πενήντα πέντε προγραμματιστικά περιβάλλοντα αρχάριων μαθητών που μπορούν να χρησιμοποιηθούν ως αφετηρία από την πρωτοβάθμια εκπαίδευση και μάλιστα από μαθητές νεαρής ηλικίας. Με αυτόν τον τρόπο διευκολύνεται η διαδικασία επιλογής του καταλληλότερου περιβάλλοντος από τον ενδιαφερόμενο εκπαιδευτικό για την ικανοποίηση των διδακτικών του στόχων, η οποία αποτελεί μια σοβαρή πρόκληση.

Τρία προγραμματιστικά περιβάλλοντα αρχάριων διαφορετικών τεχνολογικών και παιδαγωγικών χαρακτηριστικών, όπως είναι το Code.org το StarlogoTNG και το Scratch αξιοποιήθηκαν και χρησιμοποιήθηκαν στη διδασκαλία διάφορων προγραμματιστικών εννοιών στα δυο τμήματα της έκτης τάξης του δημοτικού σχολείου. Η μελέτη που πραγματοποιήθηκε σε πραγματικές συνθήκες σχολικής τάξης για αυτά τα τμήματα και διήρκεσε ένα αξιολογικό χρονικό διάστημα προσέφερε σημαντική ανατροφοδότηση, και ώθησε στον εντοπισμό των προβληματικών στη διδασκαλία του προγραμματισμού και παρανοήσεων από τους μαθητές.

Για το βαθμό κατανόησης και εξέτασης των μαθησιακών αποτελεσμάτων της κάθε έννοιας χρησιμοποιήθηκε η Ιεραρχική Αξιολόγηση Προγραμματισμού παρέχοντας την δυνατότητα για μια

ολοκληρωμένη αξιολόγηση τόσο της γνώσης στη γλώσσα προγραμματισμού όσο και της αλγοριθμικής σκέψης.

Η εφαρμογή του Code.org είχε ως στόχο την οικοδόμηση των προγραμματιστικών εννοιών της επανάληψης, της επιλογής γεγονότων, μεταβλητών και διαδικασιών άλλα και της χρήσης του ως σημείου ελέγχου, προκειμένου να διαπιστωθεί το επίπεδο του κάθε τμήματος. Η χρήση του περιβάλλοντος έδειξε να κεντρίζει το ενδιαφέρον των μαθητών, η αποδοχή εκ μέρους τους να είναι θετική και η εμπλοκή τους με τις έννοιες του προγραμματισμού να αποδίδουν χάρις των καθοδηγούμενων βιωματικών δραστηριοτήτων που προσφέρονταν από αυτό. Σε αυτήν την εκτίμηση συνηγορούν και τα αποτελέσματα της επίδοσης των μαθητών, αφού οι περισσότεροι από τους μισούς και των δυο τμημάτων φαίνεται να έχουν κατακτήσει το τρίτο και το τέταρτο επίπεδο για την κάθε έννοια αντίστοιχα. Αποτελεί ένα εύκολο εργαλείο, όπως αυτό συνάγεται από τα αποτελέσματα των ερωτηματολογίων των μαθητών, έτσι ώστε ένας αρχάριος προγραμματιστής να μπορεί να το χρησιμοποιήσει, για να κάνει τα πρώτα του βήματα με τον προγραμματισμό. Η αποδοχή των μαθητών για αυτό το περιβάλλον αποτυπώθηκε και στο γεγονός ότι τους βοήθησε να εντοπίσουν τα λάθη τους αλλά και να τα διορθώσουν, καθώς και στη δυνατότητα που τους προσέφερε για πειραματισμό με τις εν λόγω προγραμματιστικές έννοιες. Τα αποτελέσματα που εξήχθησαν φαίνεται να συμφωνούν και με τα ευρήματα των Piech, Sahami και Guibas (2015) άλλα και του Kalelioglu (2015).

Μέσω του Code.org τέθηκαν τα προγραμματιστικά θεμέλια των εξεταζόμενων εννοιών πάνω στις οποίες θα πραγματοποιηθεί το χτίσιμο του επόμενου έργου κατά τη μετάβαση σε νέα περιβάλλοντα.

Η μετάβαση αυτή συντελέστηκε στα περιβάλλοντα Starlogo TNG και Scratch, έτσι ώστε να δοθεί η δυνατότητα στους μαθητές να δημιουργήσουν εκ του μηδενός ένα ολοκληρωμένο παιχνίδι – έργο με τη μέθοδο project και το οποίο για την επιτυχή έκβαση ήταν απαραίτητο να χρησιμοποιηθούν όλες οι προγραμματιστικές έννοιες που είχαν ήδη διδαχτεί. Από τα αποτελέσματα της επίδοσης των μαθητών προκύπτει ότι το τμήμα Στ2 που χρησιμοποίησε το περιβάλλον Scratch παρουσίασε καλύτερα αποτελέσματα σε όλες τις προγραμματιστικές έννοιες κατά τις οποίες εξετάστηκε, σε σχέση με τους μαθητές του τμήματος Στ1 που ασχολήθηκαν με το Starlogo TNG, χωρίς ωστόσο να εντοπίζονται σημαντικές στατιστικές διαφορές και παρόλο που οι τελευταίοι παρουσίασαν καλύτερα αποτελέσματα από τους πρώτους στο εισαγωγικό περιβάλλον. Αυτό άλλωστε φαίνεται να

επιβεβαιώνεται και από τις απαντήσεις του ερωτηματολογίου, σύμφωνα με το οποίο το Starlogo TNG εμφανίζεται πιο δύσκολο σε σχέση με το Scratch.

Στο σημείο αυτό κρίνεται απαραίτητο να αναφερθεί ότι η συγκεκριμένη μετάβαση βοήθησε και τα δυο τμήματα ώστε να εμπλακούν καλύτερα με τον προγραμματισμό, την επίλυση προβλημάτων και την καλύτερη κατανόηση των εξεταζόμενων προγραμματιστικών εννοιών και σε αυτό συνηγορούν και τα μαθησιακά αποτελέσματα άλλα και τα ευρήματα από τα ερωτηματολόγια.

Από την ιεραρχική αξιολόγηση προγραμματισμού προέκυψε ότι για ένα ποσοστό οκτώ τοις εκατό του συνόλου των μαθητών επετεύχθη η κατάκτηση του πέμπτου επιπέδου που αντιστοιχεί σε αυτό της εκτεταμένης θεώρησης.

Επιπρόσθετα, από τα αποτελέσματα επίδοσης των μαθητών για τη σύγκριση των δυο φύλλων συμπεραίνεται ότι τα αγόρια εμφανίζονται ελάχιστα βελτιωμένα σε σχέση με τα κορίτσια, χωρίς όμως σημαντικές διαφορές, αποτύπωση που καταδεικνύει ότι και τα δυο περιβάλλοντα βοήθησαν την εκπροσώπηση των κοριτσιών και την ενασχόλησή τους με τις εξεταζόμενες προγραμματιστικές έννοιες γεγονός που συμφωνεί με τα ευρήματα της βιβλιογραφίας (Utting, Cooper, Kölling, Maloney & Resnick, 2010).

Οι μαθητές στην πλειονότητά τους θεωρούν ότι το Scratch αποτελεί ένα ενδιαφέρον περιβάλλον ικανό να τους βοηθήσει στην καλύτερη κατανόηση του προγραμματισμού, άλλα και των προγραμματιστικών εννοιών που διδάχθηκαν. Τα αποτελέσματα γι' αυτό το περιβάλλον συμφωνούν με τα ευρήματα των Meerbaum-Salant & Ben-Ari (2011) των Maloney et al. (2004, 2008) και των Wu et al. (2010) τα οποία αναφέρονται στο ότι το συγκεκριμένο προγραμματιστικό περιβάλλον αφενός μεν ενθουσιάζει τους μαθητές και αφετέρου εξοικειώνει αλλά και εμπραθύνει με βασικές έννοιες του προγραμματισμού.

Τα αποτελέσματα για το Starlogo TNG, το οποίο χαρακτηρίζεται ως πιο χρήσιμο, καθώς τους προσφέρει τη δυνατότητα να δημιουργήσουν ενδιαφέροντα τρισδιάστατα παιχνίδια με έναν πιο διασκεδαστικό τρόπο, συμφωνούν με τα συμπεράσματα των Klopfer και Yoon (2004) άλλα και του McCaffrey (2006).

Σε ότι αφορά στις δυσκολίες που προέκυψαν από την χρήση των Scratch και Starlogo TNG διαπιστώνεται ότι μετά την ολοκλήρωση των προγραμμάτων τους αφήνουν κομμάτια κώδικα ή αντικείμενα που δεν χρησιμοποιούνται πουθενά και αυτό εμφανίζεται περισσότερο στο πρώτο περιβάλλον. Επίσης, το γεγονός πως οι μαθητές δυσκολεύονταν αρκετές φορές να επιλέξουν την κατάλληλη περιοχή σεναρίων από αυτές που διαθέτει το περιβάλλον για την ανάπτυξη των

προγραμμάτων τους αφορά περισσότερο στο δεύτερο περιβάλλον, αφού οι διαθέσιμες περιοχές σεναρίων που το απαρτίζουν είναι πέντε.

Ο τρόπος που χρησιμοποιήθηκαν τα περιβάλλοντα έκαναν το μάθημα του προγραμματισμού ελκυστικό και ευχάριστο, σύμφωνα με τις απόψεις των μαθητών, δημιουργώντας ταυτόχρονα ένα αίσθημα αρέσκειας για τον προγραμματισμό υπολογιστών. Αναδεικνύεται ως εξίσου σημαντικό και το γεγονός πως εξάλειψαν τα αισθήματα του φόβου και του άγχους, τα οποία μπορεί να δημιουργούνταν κατά τη διάρκεια της εκπαιδευτικής διαδικασίας, αφού τα προγραμματιστικά περιβάλλοντα λειτούργησαν στο να φαίνεται η διαδικασία του προγραμματισμού λιγότερο δύσκολη.

Σημαντικό στοιχείο αποτελεί και η ενασχόληση των μαθητών με κάποιο από τα τρία περιβάλλοντα οικειοθελώς στο σπίτι τους σε ποσοστό 78,2%, γεγονός που συνάδει με τη διαπίστωση πως τα ΠΠΑ τους κίνησαν το ενδιαφέρον και ίσως τους ενεργοποίησαν ώστε να ασχοληθούν εκτενέστερα με το κάθε περιβάλλον. Το προαναφερθέν ποσοστό αντιστοιχεί στο Starlogo TNG με τιμή 33,3%, έπεται το Scratch με τιμή 26% και ακολουθεί το Code.org με 18.9%.

Τέλος, ο εκπαιδευτικός στον πολλαπλό του ρόλο κατά τη διάρκεια της μελέτης ως πρωτεργάτης, διευκολυντής, ρυθμιστής, εμπυχωτής, συμπαραστάτης και ερευνητής, προσπαθεί να διατηρήσει τις ισορροπίες μεταξύ της ενεργού συμμετοχής και της παρατήρησης, να μεταλαμπαδεύσει στους εκπαιδευόμενους τα εφόδια ώστε να αποκτήσουν γνώσεις και δεξιότητες, αλλά και ταυτοχρόνως να τους αφήσει ελεύθερους να διαμορφώσουν μόνοι τους στάσεις και απόψεις.

## 5.1 ΠΕΡΙΟΡΙΣΜΟΙ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Οι βασικοί περιορισμοί της έρευνας είναι το περιορισμένο δείγμα των μαθητών και στην υλικοτεχνική υποδομή του εργαστηρίου όσον αφορά στον αριθμό των διαθέσιμων υπολογιστών.

Με τη μελέτη αυτή επιχειρήθηκε αφενός μεν η συγκριτική παρουσίαση προγραμματιστικών περιβαλλόντων για αρχαίους βάσει των παιδαγωγικών και τεχνολογικών χαρακτηριστικών εδραζόμενη στις ομοιότητες και τις διαφορές τους και αφετέρου η αξιοποίηση επιλεγμένων περιβαλλόντων σε μια εμπειρική μελέτη της πορείας των μαθητών σε διάφορες προγραμματιστικές έννοιες.

Τα αποτελέσματα της έρευνας κατέδειξαν ότι τα ΠΠΑ συμβάλλουν στην εκπαιδευτική διαδικασία, στην κατανόηση προγραμματιστικών εννοιών, καθώς και στη δημιουργία υλικού μετατρέποντας τους ίδιους μαθητές από καταναλωτές σε παραγωγούς παιχνιδιών και έργων. Τα



ΠΠΑ μετέτρεψαν το προγραμματισμό και το μάθημα του σε μια ελκυστική και ευχάριστη διαδικασία δημιουργώντας αισθήματα αρέσκειας και συμπάθειας για αυτόν.

Αρχικά προτείνεται να διεξαχθεί η μελέτη με τη χρήση μεγαλύτερου μεγέθους δείγματος. Ακόμη, το δείγμα μπορεί να αποτελείται από μαθητές διαφόρων ηλικιών με στόχο τη διερεύνηση της επίδρασης των ΠΠΑ στα μαθησιακά αποτελέσματα μαθητών διαφορετικής ηλικίας.

Τέλος θα μπορούσε να προταθεί η αντικατάσταση ενός εκ των δυο περιβαλλόντων Scratch ή Starlogo TNG με ένα περιβάλλον συγγραφής με κείμενο για τη διεξαγωγή μιας μελλοντικής συγκριτικής μελέτης.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### Ελληνική Βιβλιογραφία

Γλέζου, Κ., & Γρηγοριάδου, Μ. (2004). Παίζω, διερευνώ και μαθαίνω προγραμματίζοντας τη χελώνα, Πρακτικά 2ης Διημερίδας με διεθνή συμμετοχή «Διδακτική της Πληροφορικής», 182-192, Βόλος.

Γλέζου, Κ., & Γρηγοριάδου, Μ. (2007). Ανάπτυξη προσομοίωσης της ελεύθερης πτώσης: μία εναλλακτική διαθεματική πρόταση διδασκαλίας, Στο Ν. Δαπόντες & Ν. Τζιμόπουλος (επιμ.), Πρακτικά 4ου Πανελληνίου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ «Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στη Διδακτική Πράξη», Τόμος Β΄, 210-219. Σύρος

Εμβαλωτής, Α., Σιδερίδης, Γ., & Κατσή, Α. (2006). Στατιστική Μεθοδολογία Εκπαιδευτικής Έρευνας, Α΄ Έκδοση, Ιωάννινα 2006

Εμβαλωτής, Α., Σιδερίδης, Γ., & Κατσή, Α. (2011). Στατιστικές μέθοδοι στις Κοινωνικές Επιστήμες, Εκδόσεις Τόπος

Δαγδιλέλης, Β. (1996). Διδακτική της Πληροφορικής. Η διδασκαλία του προγραμματισμού: αντιλήψεις των σπουδαστών για την κατασκευή κι επικύρωση προγραμμάτων και διδακτικές καταστάσεις για τη διαμόρφωσή τους, Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής, Πανεπιστήμιο Μακεδονίας.

Εφόπουλος, Β. (2005). Διαδικτυακό Εκπαιδευτικό Λογισμικό με υποστήριξη Βάσεων Δεδομένων για τη διδασκαλία της πληροφορικής σε αρχαρίους, Διδακτορική Διατριβή στο Πανεπιστήμιο Μακεδονίας, Τμήμα Εφαρμοσμένης Πληροφορικής. Θεσσαλονίκη.

Εφόπουλος, Β., Ευαγγελίδης, Γ., Δαγδιλέλης Β., & Κλεφτοδήμος, Αλ. (2005). Οι Δυσκολίες των Αρχάριων Προγραμματιστών, 3ο Συνέδριο Διδακτικής της Πληροφορικής, Κόρινθος.

Κόμης, Β., Μικρόπουλος, Τ.Α. (2001). Πληροφορική στην Εκπαίδευση, Ελληνικό Ανοικτό Πανεπιστήμιο.

Κόμης, Β. (2001). Διδακτική της Πληροφορικής, Ελληνικό Ανοικτό Πανεπιστήμιο.

Κόμης, Β. (2004). Εισαγωγή στις εκπαιδευτικές εφαρμογές των τεχνολογιών της πληροφορίας και των επικοινωνιών. Εκδόσεις Νέων Τεχνολογιών

Κόμης, Β. (2004). Επιμόρφωση των Εκπαιδευτικών στις Τεχνολογίες της Πληροφορίας και των Επικοινωνιών: από την τεχνολογική ευχέρεια στο εργαλείο επαγγελματικής ανάπτυξης. Στο Γ. Μπαγάκης (επιμέλεια) Επιμόρφωση και επαγγελματική ανάπτυξη του εκπαιδευτικού, 6ο ετήσιο συνέδριο με διεθνή συμμετοχή της Μονάδας Μεθοδολογίας και Προγραμμάτων Εκπαίδευσης του Παιδαγωγικού Τμήματος Νηπιαγωγών του Πανεπιστημίου Πατρών, Αθήνα: Μεταίχμιο.

Κόμης, Β. (2005). Εισαγωγή στη Διδακτική της Πληροφορικής, Εκδόσεις Κλειδάριθμος.

Κόμης, Β. (2011). Επιμορφωτικό υλικό για την εκπαίδευση εκπαιδευτικών. Πρόγραμμα Επιμόρφωσης Β' Επιπέδου, Πάτρα: ΕΑΙΤΥ

Μικρόπουλος, Τ.Α., & Λαδιάς, Τ. (1997), Η LOGO στην Εκπαιδευτική Διαδικασία, Ιωάννινα: Πανεπιστήμιο Ιωαννίνων

Μικρόπουλος, Τ.Α. (2000). Εκπαιδευτικό λογισμικό, θέματα σχεδίασης και αξιολόγησης λογισμικού πολυμέσων. Αθήνα: Κλειδάριθμος.

Μικρόπουλος, Τ. Α. (2004). Έχει θέση η Logo ως γνωστικό αντικείμενο και ολιστικό πρότυπο στην υποχρεωτική εκπαίδευση;, Πρακτικά 2ης Διημερίδας με διεθνή συμμετοχή «Διδακτική της Πληροφορικής», 65-72, Βόλος.

Μικρόπουλος, Τ. Α. (2006). Ο υπολογιστής ως γνωστικό εργαλείο. Αθήνα: Ελληνικά Γράμματα.

Μικρόπουλος, Τ. Α. (2007). Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα. Παρουσίαση βασισμένη σε ομιλία του Σχολικού Συμβούλου Πληροφορικής Α. Λαδιά.

Μικρόπουλος, Τ. Α. Μπέλλου, Ι., & Λαδιάς, Α.,(2009). Πλοήγηση και αλληλεπιδραστικότητα σε εκπαιδευτικά υπερμεσικά περιβάλλοντα. 1<sup>ο</sup> Εκπαιδευτικό Συνέδριο «Ένταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία»

Μικρόπουλος, Τ. Α., & Χαρίσης, Χ. (2008). Ρομποτική, Οπτικός Προγραμματισμός και Βασικές Προγραμματιστικές Δομές, στο Β. Κόμης (επ.) Πρακτικά 4ου Πανελληνίου Συνεδρίου Διδακτική της Πληροφορικής, 121-130, Πάτρα.

Μικρόπουλος, Τ. Α., & Μπέλλου, Ι. (2010). Σενάρια διδασκαλίας με υπολογιστή. Εκδόσεις Κλειδάριθμος.

Μικρόπουλος, Τ. Α., Μπέλλου, Ι., & Θεοδώρου, Α. (2014). Η Διδακτική της Πληροφορικής στην Ελλάδα: μία Βιβλιογραφική Επισκόπηση. Πρακτικά 7ου Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής». Πανεπιστήμιο Κρήτης, Ρέθυμνο, 3-5 Οκτωβρίου 2014.

Παπαδάκης, Στ., Ορφανάκης, Β., Καλογιαννάκης. Μ., & Ζαράνης, Ν. (2014). Περιβάλλοντα προγραμματισμού για αρχάριους. Scratch & App Inventor: μια πρώτη σύγκριση. Πρακτικά 7ου Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής». Πανεπιστήμιο Κρήτης, Ρέθυμνο, 3-5 Οκτωβρίου 2014.

Παιδαγωγικό Ινστιτούτο. (1997). Ενιαίο Πλαίσιο Προγράμματος Σπουδών Πληροφορικής. Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων.

Παιδαγωγικό Ινστιτούτο. (2003). Διαθεματικό Ενιαίο Πλαίσιο Προγραμμάτων Σπουδών. Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων.

Παιδαγωγικό Ινστιτούτο. (2010). Ενιαίο Αναμορφωμένο Εκπαιδευτικό Πρόγραμμα. Υπουργείο Διά Βίου Μάθησης και Θρησκευμάτων

ΥΠΔΒΜΘ. (2011). Πρόγραμμα Σπουδών για τις ΤΠΕ στην Πρωτοβάθμια Εκπαίδευση, ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα) – Νέο Πρόγραμμα Σπουδών, Οριζόντια Πράξη. MIS: 295450, ΕΣΠΑ 2007-13\Ε.Π. Ε&ΔΒΜ\Α.Π. 1-2-3, Παιδαγωγικό Ινστιτούτο, Αθήνα, 2011

ΥΠΕΠΘ. (2016) Ωρολόγιο Πρόγραμμα Ενιαίου Τύπου Ολοήμερου Δημοτικού Σχολείου. Αριθμ. Φ. 12/657/70691/Δ1

Τζιμογιάννης, Α., & Γεωργίου, Β. (1998). Η Αναγκαιότητα της Διδασκαλίας του Προγραμματισμού Η/Υ στη Δευτεροβάθμια Εκπαίδευση ως Μεθοδολογία Επίλυσης Προβλημάτων. Το παράδειγμα των πινάκων, Πρακτικά Διημερίδας Πληροφορικής «Η Πληροφορική στη Δευτεροβάθμια Εκπαίδευση», ΕΠΥ, 28-34

Τζιμογιάννης, Α., & Γεωργίου, Β. (1999). Οι δυσκολίες μαθητών δευτεροβάθμιας εκπαίδευσης στην εφαρμογή της δομής ελέγχου για την ανάπτυξη αλγορίθμων. Μία μελέτη περίπτωσης, Στο Α. Τζιμογιάννης (επιμ.). Πρακτικά Πανελληνίου Συνεδρίου «Πληροφορική και Εκπαίδευση», Σύλλογος Καθηγητών Πληροφορικής Ηπείρου, 183-192

Τζιμογιάννης, Α. (2005). Προς ένα Παιδαγωγικό Πλαίσιο Διδασκαλίας του Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση. Πρακτικά Εργασιών 3ου Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής», Κόρινθος.

## Ξενόγλωσση Βιβλιογραφία

Arter, J. & McTighe, J. (2001). *Scoring rubrics in the classroom: Using performance criteria for assessing and improving student performance*. Thousand Oaks, CA: Corwin Press, Inc

Andrade, H. (2000). Using rubrics to promote thinking and learning. *Educational Leadership*, 57(5), pp. 13-18.

Bau, D. (2015). Droplet, a blocks-based editor for text code. *J. Comput. Sci. Coll.* 30, 6 (June 2015), 138-144.

Bau, D., Bau, A., Dawson, M. and Pickens, S. (2015). Pencil code: block code for a text world. In *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*. ACM, New York, NY, USA, 445-448. DOI=<http://dx.doi.org/10.1145/2771839.2771875>

Biggs, J., & Collis, K. (1989). Towards a model of school-based curriculum development and assessment using the SOLO taxonomy. *Australian journal of education*, 33(2), 151-163.

Brown, M. (1998). Perspectives on Algorithm Animation, *CHI'88 Human Factors in Computing Systems*, Washington, DC, USA, pp. 33-38.

Byrne, P., & Lyons, G. (2001). The Effect of Student Attributes on Success in Programming, in *Proc. of 6th Annu. Conf. on Innovation and Technology in Computer Science Education - ITiCSE 2001*, United Kingdom, 2001, pp. 49-52

Cooper, S., Dann, W., & Pausch, R. (2000, April). Alice: a 3-D tool for introductory programming concepts. In *Journal of Computing Sciences in Colleges* (Vol. 15, No. 5, pp. 107-116). Consortium for Computing Sciences in Colleges.

Dekhane, S., & Xu, X. (2012). Engaging students in computing using GameSalad: a pilot study. *Journal of Computing Sciences in Colleges*, 28(2), 117-123

Dufoyer, J-P. (1988). *Informatique, éducation et psychologie de l'enfant*. Paris : PUF.

Duncan, C., Bell, T., & Tanimoto, S. (2014, November). Should your 8-year-old learn coding?. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 60-69). ACM.

Du Boulay, B. (1989). Some difficulties of learning to program, in E. Soloway & J. C. Spohrer (Eds.), *Studying the Novice Programmer*, 283-299, Hillsdale, NJ: Lawrence Erlbaum Associates

Eshet, Y. (2004). Digital Literacy: A Conceptual Framework for Survival Skills in the Digital era. *Journal of Educational Multimedia and Hypermedia*, 13(1), 93-106. Norfolk, VA: Association for the Advancement of Computing in Education (AACE).

Fincher, S., Cooper, S., Kölling, M., & Maloney, J. (2010, March). Comparing alice, greenfoot & scratch. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 192-193). ACM.

Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing education in k-12 schools: A review of the literature. In *2015 IEEE Global Engineering Education Conference (EDUCON)* (pp. 543-551). IEEE.

Gans, P. (2010). The benefits of using scratch to introduce basic programming concepts in the elementary classroom: poster session. *Journal of Computing Sciences in Colleges*, 25(6), 235-236.

Gomes, A., & Mendes, A. J. (2007, September). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE* (Vol. 2007).

Gomes, A., L. Carmo, E. Bigotte and A. J. Mendes,(2006). “Mathematics and programming problem solving”, in *Proc. of the 3rd E-Learning Conf. – Computer Science Education (CD-ROM)*, Coimbra, Portugal, September 2006.

Howe, J. A. M., Ross, P. M., Johnson, K. R., Plane, F. & Inglis, R. (1989). in E. Soloway & J. C. Spohrer (Eds.), *Studying the novice programmer*, 43-55, Hillsdale, NJ: Lawrence Erlbaum.

Hoc, J-M. (1989). Do we really have conditional statements in our brains?, in E.Soloway & J. C. Spohrer (Eds.), *Studying the Novice Programmer*, 179-90, Hillsdale,NJ: Lawrence Erlbaum Associates

Hurtes, K. P., Allen, L. R., Stevens, B. W., & Lee, C. (2000). Benefits-Based Programming: Making an Impact on Youth. *Journal of Park & Recreation Administration*, 18(1).

Jenkins, T. (2002, August). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, pp. 53-58).

Javeau, C. (2000). Η έρευνα με ερωτηματολόγιο : Το εγχειρίδιο του καλού ερευνητή. Αθήνα. Τυπωθήτω.

Kahney, H. (1989). What do novice programmers know about recursion, in E. Soloway & J. C. Spohrer (Eds.), *Studying the Novice Programmer*, 209-228, Hillsdale, NJ:Lawrence Erlbaum Associates

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. Baskent University, Faculty of Education, Ankara, Turkey. *Computers in Human Behavior* Volume 52, November 2015, Pages 200–210

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83-137

Kelleher, C., Pausch, R., & Kiesler, S. (2007, April). Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464). ACM.

Kelleher, C. (2012). Reading, Writing, Arithmetic and Programming? 6ο Πανελλήνιο Συνέδριο Διδακτική της Πληροφορικής, Φλώρινα.

Kessler C. M. & J. R. Anderson (1986), A model of novice debugging in LISP, in E. Soloway & S. Iyengar (Eds.), *Empirical Studies of Programmers*, 198-212, Washington, DC, Ablex Publishing Corporation

Klopfer, E., & Yoon, S.(2004). Developing games and simulations for today and tomorrow's tech savvy youth. *TechTrends*, 49(3), 33-41.

Kumar,D.(2014).Digital playgrounds for early computing education *ACM Inroads* 5(1):20-21

Lahtinen,E., Ala-Mutka,K., & Järvinen,H. (2005). A study of the difficulties of novice programmers. *SIGCSE Bull.* 37, 3 (June 2005), 14-18.

Lewis, C. M. (2010, March). How programming environment shapes perception, learning and goals: logo vs. scratch. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 346-350). ACM.

Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, 39(1), 223-227

Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004, January). Scratch: a sneak preview [education]. In *Creating, Connecting and Collaborating through Computing, 2004. Proceedings. Second International Conference on* (pp. 104-109). IEEE

McCaffrey, C. (2006). Starlogo tng: the convergence of graphical programming and text processing. Massachusetts Institute of Technology Master's Thesis.

Myers, B.A. (1990). Taxonomies of visual programming and program visualization. *J. Vis. Lang. Comput.* 1, 1 (March 1990), 97-123. DOI=[http://dx.doi.org/10.1016/S1045-926X\(05\)80036-9](http://dx.doi.org/10.1016/S1045-926X(05)80036-9)

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011, June). Habits of programming in scratch. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 168-172). ACM.

Milne, I. & Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors *Education and Information Technologies* 7: 55. doi:10.1023/A:1015362608943

Naps, T., Roessling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., & Velasquez Iturbide, J., A. (2003). Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, 35(2), 131-152.

Nitko, J.A., & Brookhart, S.M. (2007). *Assessment and Grading in Classrooms*. Brookhart. ISBN 9780132217446.

Roy, K., Rouse, W. C., & DeMeritt, D. B. (2012, October). Comparing the mobile novice programming environments: App Inventor for Android vs. GameSalad. In *2012 Frontiers in Education Conference Proceedings* (pp. 1-6). IEEE.

Roschelle, J., Pea, R., Hoadley, C., Gordin, D., & Means, B.(2000). "Changing how and what children learn in school with computer-based technologies".*The future of children*, pp. 76-101, 2000.

Roman, G. C., & Cox K. C. (1993). A taxonomy for program visualisation systems. *Computer*, 26(12), 11-24.

Rentsch, T. (1982). Object oriented programming. *ACM Sigplan Notices*, 17(9), 51-57.

Pane J. & Myers B. (1996), *Usability issues in the design of novice programming systems*, School of Computer Science Technical Report, Carnegie Mellon University, CMU-CS-96-132, Pittsburgh

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.

Papert, S. (1991). *The children's machine: rethinking school in the age of the computer*. New York: Basic Books.



Partovi, H. (2015). A comprehensive effort to expand access and diversity in computer science. Magazine ACM Inroads Volume 6 Issue 3, September 2015 Pages 67-72

Pratim Sengupta & John S. Kinnebrew & Satabdi Basu & Gautam Biswas & Douglas Clark ,(2012). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework Educ Inf Technol (2013) 18:351-380. DOI 10.1007/s10639-012-9240-x

Price, T. W., & Barnes, T. (2015, July). Comparing Textual and Block Interfaces in a Novice Programming Environment. In Proceedings of the eleventh annual International Conference on International Computing Education Research (pp. 91-99). ACM.

Pokkunuri, B. P. (1989). Object oriented programming. ACM Sigplan Notices, 24(11), 96-101.

Price, T. W., Brown, N. C., Lipovac, D., Barnes, T., & Kölling, M. (2016, September). Evaluation of a Frame-based Programming Editor. In Proceedings of the 2016 ACM Conference on International Computing Education Research (pp. 33-42). ACM.

Piech,C., Sahami,M., Huang,J., & Guibas. L.(2015). Autonomously Generating Hints by Inferring Problem Solving Policies. In Proceedings of the Second (2015) ACM Conference on Learning @ Scale (L@S '15). ACM, New York, NY, USA, 195-204.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. Communications of the ACM, vol. 52, no. 11, pp. 60-67.

Rogalski, J. & Samurçay, R. (1990). Acquisition of programming knowledge and skills.In J-M. Hoc, T.R.G. Green, D.J. Gilmore and R. Samurçay (Eds.) Psychology of Programming, pp 157-174. London: Academic Press.

Rogalski J. & Vergnaud G. (1987). Didactique de l'informatique et acquisitions cognitives en programmation, Psychologie Française, 32 (2), 267-273.

Rogalski J. (1987). Acquisition et didactique des structures conditionnelles en programmation informatique, Psychologie Française, 32 (2), 275-280.

Soloway E. & Spohrer J. C. (1989), Eds, Studying the Novice Programmer, (Hillsdale, NJ, Erlbaum).

Spohrer J. & Soloway E. (1986), Novice mistakes: are the folk wisdoms correct?, Communications of the ACM, 29(7), 624-632

Stasko, J. & Patterson, C. (1992). Understanding and characterizing software visualization systems. IEEE Workshop on Visual Languages (pp. 3-10). Seattle, USA: IEEE Computer Society

Stolee, K. T., & Fristoe, T. (2011, March). Expressing computer science concepts through Kodu game lab. In Proceedings of the 42nd ACM technical symposium on Computer science education (pp. 99-104). ACM

Smith, B. (2015). Object-Oriented Programming. In Advanced ActionScript 3 (pp. 1-23). Apress.

Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot, and scratch--a discussion. ACM Transactions on Computing Education (TOCE), 10(4), 17.

Yin R.K. (2003). Case Study Research: Design and Methods. 3rd edition. Sage Publications

Webb, H. (2012). Computer applications for the classroom: a review. J. Comput. Sci. Coll. 27, 3 (January 2012), 65-72.

Weintrop, D., & Wilensky, U. (2015, July). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In Proceedings of the eleventh annual International Conference on International Computing Education Research (pp. 101-110). ACM.

Weintrop, D. (2015, February). Minding the gap between blocks-based and text-based programming. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 720-720). ACM.

Weintrop, D., & Wilensky, U. (2015, June). To block or not to block, that is the question: students' perceptions of blocks-based programming. In Proceedings of the 14th International Conference on Interaction Design and Children (pp. 199-208). ACM.

Wu, W. Y., Chang, C. K., & He, Y. Y. (2010, May). Using Scratch as game- based learning tool to reduce learning anxiety in programming course. In Global Learn Asia Pacific (Vol. 2010, No. 1, pp. 1845-1852).

Winslow L.E., (1996). "Programming pedagogy - a psychological overview", In Newsletter ACM SIGCSE Bulletin, ACM, vol.28, iss.3, pages 17-22, Sept. 1996.

Wikipedia (2016), Τελευταία πρόσβαση 2/11/2016.

[https://el.wikipedia.org/wiki/%CE%9F%CF%80%CF%84%CE%B9%CE%BA%CE%AE\\_%CE%B3%CE%BB%CF%8E%CF%83%CF%83%CE%B1\\_%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D](https://el.wikipedia.org/wiki/%CE%9F%CF%80%CF%84%CE%B9%CE%BA%CE%AE_%CE%B3%CE%BB%CF%8E%CF%83%CF%83%CE%B1_%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D)

## ΠΑΡΑΡΤΗΜΑΤΑ

### Α ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

#### ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΠΟ ΤΗΝ ΧΡΗΣΗ ΤΟΥ CODE.ORG

##### Επανάληψη Γνώση στη γλώσσα προγραμματισμού

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως <b>επανάληψης</b>	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών <b>επανάληψης</b>	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
<b>ΣΤ1</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3		✓			
Μαθητής 4				✓	
Μαθητής 5				✓	
Μαθητής 6			✓		
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9				✓	
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15			✓		
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18				✓	
<b>ΣΤ2</b>					
Μαθητής 1			✓		
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6				✓	
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11		✓			
Μαθητής 12				✓	
Μαθητής 13				✓	
Μαθητής 14			✓		

Μαθητής 15				✓	
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19				✓	

**Επανάληψη  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>ΣΤ1</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5			✓		
Μαθητής 6		✓			
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14				✓	
Μαθητής 15		✓			
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18				✓	
<b>ΣΤ2</b>					
Μαθητής 1			✓		
Μαθητής 2			✓		
Μαθητής 3				✓	
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6				✓	
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9			✓		
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12			✓		
Μαθητής 13				✓	
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19				✓	

### Επανάληψη Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>ΣΤ1</b>							
Μαθητής 1						✓	
Μαθητής 2				✓			
Μαθητής 3			✓				
Μαθητής 4						✓	
Μαθητής 5					✓		
Μαθητής 6			✓				
Μαθητής 7				✓			
Μαθητής 8					✓		
Μαθητής 9						✓	
Μαθητής 10					✓		
Μαθητής 11				✓			
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14					✓		
Μαθητής 15			✓				
Μαθητής 16				✓			
Μαθητής 17						✓	
Μαθητής 18						✓	
<b>ΣΤ2</b>							
Μαθητής 1				✓			
Μαθητής 2					✓		
Μαθητής 3					✓		
Μαθητής 4						✓	
Μαθητής 5		✓					
Μαθητής 6						✓	
Μαθητής 7				✓			
Μαθητής 8				✓			
Μαθητής 9					✓		
Μαθητής 10					✓		
Μαθητής 11			✓				
Μαθητής 12					✓		
Μαθητής 13						✓	
Μαθητής 14				✓			
Μαθητής 15						✓	
Μαθητής 16					✓		
Μαθητής 17						✓	
Μαθητής 18				✓			
Μαθητής 19						✓	

### Επιλογή Γνώση στη γλώσσα προγραμματισμού

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης και	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός	Βαθιά γνώση και φιλοσοφία της γλώσσας
-----------------	--------------------------	--------------------------	--	---	---------------------------------------

			επιλογής.	κατάλληλων δομών επανάληψης, επιλογής	προγραμματισμού
ΣΤ1					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3	✓				
Μαθητής 4				✓	
Μαθητής 5				✓	
Μαθητής 6			✓		
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9				✓	
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15			✓		
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18				✓	
ΣΤ2					
Μαθητής 1			✓		
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5	✓				
Μαθητής 6				✓	
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11	✓				
Μαθητής 12				✓	
Μαθητής 13				✓	
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19				✓	

**Επιλογή  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
ΣΤ1					
Μαθητής 1				✓	

Μαθητής 2			✓		
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5			✓		
Μαθητής 6	✓				
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14				✓	
Μαθητής 15	✓				
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18				✓	
ΣΤ2					
Μαθητής 1			✓		
Μαθητής 2			✓		
Μαθητής 3				✓	
Μαθητής 4				✓	
Μαθητής 5	✓				
Μαθητής 6				✓	
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9			✓		
Μαθητής 10				✓	
Μαθητής 11	✓				
Μαθητής 12			✓		
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19			✓		

### Επιλογή Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
ΣΤ1							
Μαθητής 1						✓	
Μαθητής 2				✓			
Μαθητής 3			✓				
Μαθητής 4					✓		
Μαθητής 5					✓		
Μαθητής 6			✓				
Μαθητής 7				✓			
Μαθητής 8						✓	

Μαθητής 9					✓	
Μαθητής 10					✓	
Μαθητής 11			✓			
Μαθητής 12					✓	
Μαθητής 13			✓			
Μαθητής 14				✓		
Μαθητής 15	✓					
Μαθητής 16			✓			
Μαθητής 17					✓	
Μαθητής 18				✓		
ΣΤ2						
Μαθητής 1			✓			
Μαθητής 2				✓		
Μαθητής 3				✓		
Μαθητής 4					✓	
Μαθητής 5	✓					
Μαθητής 6					✓	
Μαθητής 7			✓			
Μαθητής 8			✓			
Μαθητής 9				✓		
Μαθητής 10				✓		
Μαθητής 11		✓				
Μαθητής 12				✓		
Μαθητής 13				✓		
Μαθητής 14		✓				
Μαθητής 15					✓	
Μαθητής 16				✓		
Μαθητής 17					✓	
Μαθητής 18			✓			
Μαθητής 19					✓	

**Γεγονότα**  
**Γνώση στη γλώσσα προγραμματισμού**

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης, επιλογής και γεγονότων	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης, επιλογής και γεγονότων	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
ΣΤ1					
Μαθητής 1			✓		
Μαθητής 2			✓		
Μαθητής 3				✓	
Μαθητής 4				✓	
Μαθητής 5			✓		
Μαθητής 6			✓		
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9				✓	
Μαθητής 10			✓		



Μαθητής 11				✓	
Μαθητής 12				✓	
Μαθητής 13				✓	
Μαθητής 14				✓	
Μαθητής 15			✓		
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18				✓	
<b>ΣΤ2</b>					
Μαθητής 1			✓		
Μαθητής 2				✓	
Μαθητής 3				✓	
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6				✓	
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19				✓	

**Γεγονότα  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>ΣΤ1</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3		✓			
Μαθητής 4			✓		
Μαθητής 5				✓	
Μαθητής 6			✓		
Μαθητής 7				✓	
Μαθητής 8				✓	
Μαθητής 9				✓	
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15		✓			

Μαθητής 16				✓	
Μαθητής 17				✓	
Μαθητής 18				✓	
ΣΤ2					
Μαθητής 1			✓		
Μαθητής 2				✓	
Μαθητής 3				✓	
Μαθητής 4				✓	
Μαθητής 5			✓		
Μαθητής 6				✓	
Μαθητής 7				✓	
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11	✓				
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19				✓	

#### Γεγονότα Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
ΣΤ1							
Μαθητής 1						✓	
Μαθητής 2				✓			
Μαθητής 3			✓				
Μαθητής 4					✓		
Μαθητής 5						✓	
Μαθητής 6				✓			
Μαθητής 7			✓				
Μαθητής 8					✓		
Μαθητής 9						✓	
Μαθητής 10					✓		
Μαθητής 11					✓		
Μαθητής 12						✓	
Μαθητής 13					✓		
Μαθητής 14					✓		
Μαθητής 15			✓				
Μαθητής 16					✓		
Μαθητής 17						✓	
Μαθητής 18						✓	
ΣΤ2							
Μαθητής 1				✓			
Μαθητής 2					✓		
Μαθητής 3					✓		

Μαθητής 4					✓		
Μαθητής 5			✓				
Μαθητής 6						✓	
Μαθητής 7					✓		
Μαθητής 8				✓			
Μαθητής 9						✓	
Μαθητής 10					✓		
Μαθητής 11			✓				
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14				✓			
Μαθητής 15						✓	
Μαθητής 16						✓	
Μαθητής 17						✓	
Μαθητής 18				✓			
Μαθητής 19						✓	

**Μεταβλητές  
Γνώση στη γλώσσα προγραμματισμού**

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης, επιλογής, γεγονότων και μεταβλητών.	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης, επιλογής, γεγονότων και μεταβλητών.	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
ΣΤ1					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3		✓			
Μαθητής 4			✓		
Μαθητής 5				✓	
Μαθητής 6		✓			
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15		✓			
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18			✓		
ΣΤ2					
Μαθητής 1			✓		
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4				✓	

Μαθητής 5		✓			
Μαθητής 6				✓	
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9			✓		
Μαθητής 10			✓		
Μαθητής 11		✓			
Μαθητής 12			✓		
Μαθητής 13				✓	
Μαθητής 14		✓			
Μαθητής 15			✓		
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19				✓	

**Μεταβλητές  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>ΣΤ1</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3		✓			
Μαθητής 4			✓		
Μαθητής 5			✓		
Μαθητής 6		✓			
Μαθητής 7				✓	
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15		✓			
Μαθητής 16			✓		
Μαθητής 17			✓		
Μαθητής 18				✓	
<b>ΣΤ2</b>					
Μαθητής 1			✓		
Μαθητής 2			✓		
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6			✓		
Μαθητής 7		✓			
Μαθητής 8			✓		
Μαθητής 9				✓	

Μαθητής 10			✓		
Μαθητής 11		✓			
Μαθητής 12			✓		
Μαθητής 13			✓		
Μαθητής 14		✓			
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17			✓		
Μαθητής 18			✓		
Μαθητής 19					✓

### Μεταβλητές Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>ΣΤ1</b>							
Μαθητής 1						✓	
Μαθητής 2				✓			
Μαθητής 3		✓					
Μαθητής 4					✓		
Μαθητής 5					✓		
Μαθητής 6		✓					
Μαθητής 7					✓		
Μαθητής 8					✓		
Μαθητής 9						✓	
Μαθητής 10				✓			
Μαθητής 11				✓			
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14				✓			
Μαθητής 15		✓					
Μαθητής 16				✓			
Μαθητής 17					✓		
Μαθητής 18					✓		
<b>ΣΤ2</b>							
Μαθητής 1				✓			
Μαθητής 2					✓		
Μαθητής 3				✓			
Μαθητής 4						✓	
Μαθητής 5		✓					
Μαθητής 6					✓		
Μαθητής 7			✓				
Μαθητής 8					✓		
Μαθητής 9					✓		
Μαθητής 10				✓			
Μαθητής 11		✓					
Μαθητής 12				✓			
Μαθητής 13					✓		
Μαθητής 14		✓					
Μαθητής 15					✓		
Μαθητής 16					✓		

Μαθητής 17					✓		
Μαθητής 18				✓			
Μαθητής 19						✓	

**Διαδικασίες  
Γνώση στη γλώσσα προγραμματισμού**

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης, επιλογής, γεγονότων, μεταβλητών και διαδικασιών	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης, επιλογής, γεγονότων, μεταβλητών και διαδικασιών.	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
<b>ΣΤ1</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3		✓			
Μαθητής 4			✓		
Μαθητής 5				✓	
Μαθητής 6		✓			
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15		✓			
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18			✓		
<b>ΣΤ2</b>					
Μαθητής 1			✓		
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6				✓	
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9			✓		
Μαθητής 10			✓		
Μαθητής 11		✓			
Μαθητής 12			✓		
Μαθητής 13				✓	
Μαθητής 14		✓			
Μαθητής 15			✓		
Μαθητής 16			✓		

Μαθητής 17				✓	
Μαθητής 18			✓		
Μαθητής 19				✓	

**Διαδικασίες  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>ΣΤ1</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3		✓			
Μαθητής 4			✓		
Μαθητής 5			✓		
Μαθητής 6		✓			
Μαθητής 7				✓	
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15		✓			
Μαθητής 16			✓		
Μαθητής 17			✓		
Μαθητής 18				✓	
<b>ΣΤ2</b>					
Μαθητής 1			✓		
Μαθητής 2			✓		
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6			✓		
Μαθητής 7		✓			
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11		✓			
Μαθητής 12			✓		
Μαθητής 13			✓		
Μαθητής 14		✓			
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17			✓		
Μαθητής 18			✓		
Μαθητής 19				✓	

### Διαδικασίες Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>ΣΤ1</b>							
Μαθητής 1						✓	
Μαθητής 2				✓			
Μαθητής 3		✓					
Μαθητής 4					✓		
Μαθητής 5					✓		
Μαθητής 6		✓					
Μαθητής 7					✓		
Μαθητής 8					✓		
Μαθητής 9						✓	
Μαθητής 10				✓			
Μαθητής 11				✓			
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14				✓			
Μαθητής 15		✓					
Μαθητής 16				✓			
Μαθητής 17					✓		
Μαθητής 18					✓		
<b>ΣΤ2</b>							
Μαθητής 1				✓			
Μαθητής 2					✓		
Μαθητής 3				✓			
Μαθητής 4						✓	
Μαθητής 5		✓					
Μαθητής 6					✓		
Μαθητής 7			✓				
Μαθητής 8					✓		
Μαθητής 9					✓		
Μαθητής 10				✓			
Μαθητής 11		✓					
Μαθητής 12				✓			
Μαθητής 13					✓		
Μαθητής 14		✓					
Μαθητής 15					✓		
Μαθητής 16					✓		
Μαθητής 17					✓		
Μαθητής 18				✓			
Μαθητής 19						✓	



**ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΠΟ ΤΗΝ ΧΡΗΣΗ ΤΟΥ SCRATCH - STARLOGO TNG****Επανάληψη****Γνώση στη γλώσσα προγραμματισμού**

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
<b>Starlogo TNG</b>					
Μαθητής 1					√
Μαθητής 2				√	
Μαθητής 3		√			
Μαθητής 4				√	
Μαθητής 5				√	
Μαθητής 6		√			
Μαθητής 7			√		
Μαθητής 8				√	
Μαθητής 9					√
Μαθητής 10				√	
Μαθητής 11			√		
Μαθητής 12				√	
Μαθητής 13			√		
Μαθητής 14			√		
Μαθητής 15		√			
Μαθητής 16			√		
Μαθητής 17				√	
Μαθητής 18				√	
<b>Scratch</b>					
Μαθητής 1				√	
Μαθητής 2			√		
Μαθητής 3			√		
Μαθητής 4				√	
Μαθητής 5		√			
Μαθητής 6					√
Μαθητής 7				√	
Μαθητής 8			√		
Μαθητής 9				√	
Μαθητής 10				√	
Μαθητής 11		√			
Μαθητής 12				√	
Μαθητής 13				√	
Μαθητής 14			√		
Μαθητής 15				√	
Μαθητής 16			√		
Μαθητής 17				√	
Μαθητής 18				√	
Μαθητής 19					√

**Επανάληψη  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>Starlogo TNG</b>					
Μαθητής 1					✓
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5			✓		
Μαθητής 6		✓			
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9					✓
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14				✓	
Μαθητής 15			✓		
Μαθητής 16			✓		
Μαθητής 17			✓		
Μαθητής 18				✓	
<b>Scratch</b>					
Μαθητής 1				✓	
Μαθητής 2				✓	
Μαθητής 3				✓	
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6					✓
Μαθητής 7			✓		
Μαθητής 8			✓		
Μαθητής 9			✓		
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12			✓		
Μαθητής 13				✓	
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17			✓		
Μαθητής 18				✓	
Μαθητής 19					✓

### Επανάληψη Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>Starlogo TNG</b>							
Μαθητής 1							√
Μαθητής 2						√	
Μαθητής 3			√				
Μαθητής 4						√	
Μαθητής 5					√		
Μαθητής 6		√					
Μαθητής 7				√			
Μαθητής 8					√		
Μαθητής 9							√
Μαθητής 10						√	
Μαθητής 11				√			
Μαθητής 12						√	
Μαθητής 13				√			
Μαθητής 14					√		
Μαθητής 15			√				
Μαθητής 16				√			
Μαθητής 17					√		
Μαθητής 18						√	
<b>Scratch</b>							
Μαθητής 1						√	
Μαθητής 2					√		
Μαθητής 3					√		
Μαθητής 4						√	
Μαθητής 5		√					
Μαθητής 6							√
Μαθητής 7					√		
Μαθητής 8				√			
Μαθητής 9					√		
Μαθητής 10						√	
Μαθητής 11			√				
Μαθητής 12					√		
Μαθητής 13						√	
Μαθητής 14				√			
Μαθητής 15						√	
Μαθητής 16					√		
Μαθητής 17					√		
Μαθητής 18						√	
Μαθητής 19							√

**Επιλογή  
Γνώση στη γλώσσα προγραμματισμού**

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης και επιλογής.	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης, επιλογής	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
<b>Starlogo TNG</b>					
Μαθητής 1					√
Μαθητής 2				√	
Μαθητής 3		√			
Μαθητής 4				√	
Μαθητής 5			√		
Μαθητής 6		√			
Μαθητής 7			√		
Μαθητής 8			√		
Μαθητής 9					√
Μαθητής 10			√		
Μαθητής 11			√		
Μαθητής 12				√	
Μαθητής 13			√		
Μαθητής 14			√		
Μαθητής 15		√			
Μαθητής 16			√		
Μαθητής 17				√	
Μαθητής 18				√	
<b>Scratch</b>					
Μαθητής 1			√		
Μαθητής 2			√		
Μαθητής 3			√		
Μαθητής 4					√
Μαθητής 5		√			
Μαθητής 6				√	
Μαθητής 7			√		
Μαθητής 8			√		
Μαθητής 9			√		
Μαθητής 10				√	
Μαθητής 11		√			
Μαθητής 12			√		
Μαθητής 13				√	
Μαθητής 14			√		
Μαθητής 15				√	
Μαθητής 16				√	
Μαθητής 17			√		
Μαθητής 18				√	
Μαθητής 19					√

**Επιλογή  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>Starlogo TNG</b>					
Μαθητής 1					✓
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4			✓		
Μαθητής 5				✓	
Μαθητής 6		✓			
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9					✓
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15		✓			
Μαθητής 16			✓		
Μαθητής 17			✓		
Μαθητής 18			✓		
<b>Scratch</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5			✓		
Μαθητής 6					✓
Μαθητής 7				✓	
Μαθητής 8			✓		
Μαθητής 9				✓	
Μαθητής 10			✓		
Μαθητής 11		✓			
Μαθητής 12			✓		
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16			✓		
Μαθητής 17			✓		
Μαθητής 18				✓	
Μαθητής 19					✓

### Επιλογή Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>Starlogo TNG</b>							
Μαθητής 1							✓
Μαθητής 2						✓	
Μαθητής 3			✓				
Μαθητής 4					✓		
Μαθητής 5					✓		
Μαθητής 6		✓					
Μαθητής 7				✓			
Μαθητής 8					✓		
Μαθητής 9							✓
Μαθητής 10				✓			
Μαθητής 11				✓			
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14				✓			
Μαθητής 15		✓					
Μαθητής 16				✓			
Μαθητής 17					✓		
Μαθητής 18					✓		
<b>Scratch</b>							
Μαθητής 1					✓		
Μαθητής 2				✓			
Μαθητής 3				✓			
Μαθητής 4						✓	
Μαθητής 5			✓				
Μαθητής 6							✓
Μαθητής 7					✓		
Μαθητής 8				✓			
Μαθητής 9					✓		
Μαθητής 10					✓		
Μαθητής 11		✓					
Μαθητής 12				✓			
Μαθητής 13					✓		
Μαθητής 14				✓			
Μαθητής 15						✓	
Μαθητής 16					✓		
Μαθητής 17					✓		
Μαθητής 18						✓	
Μαθητής 19							✓

**Γεγονότα**  
**Γνώση στη γλώσσα προγραμματισμού**

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης, επιλογής και γεγονότων	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης, επιλογής και γεγονότων	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
<b>Starlogo TNG</b>					
Μαθητής 1					✓
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5				✓	
Μαθητής 6			✓		
Μαθητής 7			✓		
Μαθητής 8				✓	
Μαθητής 9					✓
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14				✓	
Μαθητής 15			✓		
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18				✓	
<b>Scratch</b>					
Μαθητής 1			✓		
Μαθητής 2				✓	
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5		✓			
Μαθητής 6					✓
Μαθητής 7				✓	
Μαθητής 8				✓	
Μαθητής 9			✓		
Μαθητής 10				✓	
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13				✓	
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16				✓	
Μαθητής 17				✓	
Μαθητής 18				✓	
Μαθητής 19					✓

**Γεγονότα  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>Starlogo TNG</b>					
Μαθητής 1					✓
Μαθητής 2			✓		
Μαθητής 3			✓		
Μαθητής 4				✓	
Μαθητής 5				✓	
Μαθητής 6		✓			
Μαθητής 7				✓	
Μαθητής 8				✓	
Μαθητής 9					✓
Μαθητής 10			✓		
Μαθητής 11			✓		
Μαθητής 12				✓	
Μαθητής 13			✓		
Μαθητής 14			✓		
Μαθητής 15		✓			
Μαθητής 16			✓		
Μαθητής 17			✓		
Μαθητής 18				✓	
<b>Scratch</b>					
Μαθητής 1				✓	
Μαθητής 2			✓		
Μαθητής 3				✓	
Μαθητής 4				✓	
Μαθητής 5			✓		
Μαθητής 6					✓
Μαθητής 7				✓	
Μαθητής 8			✓		
Μαθητής 9			✓		
Μαθητής 10			✓		
Μαθητής 11		✓			
Μαθητής 12			✓		
Μαθητής 13				✓	
Μαθητής 14			✓		
Μαθητής 15				✓	
Μαθητής 16			✓		
Μαθητής 17				✓	
Μαθητής 18				✓	
Μαθητής 19					✓



### Γεγονότα Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>Starlogo TNG</b>							
Μαθητής 1							✓
Μαθητής 2					✓		
Μαθητής 3				✓			
Μαθητής 4						✓	
Μαθητής 5						✓	
Μαθητής 6			✓				
Μαθητής 7					✓		
Μαθητής 8						✓	
Μαθητής 9							✓
Μαθητής 10					✓		
Μαθητής 11				✓			
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14					✓		
Μαθητής 15			✓				
Μαθητής 16				✓			
Μαθητής 17					✓		
Μαθητής 18						✓	
<b>Scratch</b>							
Μαθητής 1					✓		
Μαθητής 2					✓		
Μαθητής 3					✓		
Μαθητής 4						✓	
Μαθητής 5			✓				
Μαθητής 6							✓
Μαθητής 7						✓	
Μαθητής 8					✓		
Μαθητής 9				✓			
Μαθητής 10					✓		
Μαθητής 11			✓				
Μαθητής 12					✓		
Μαθητής 13						✓	
Μαθητής 14				✓			
Μαθητής 15						✓	
Μαθητής 16					✓		
Μαθητής 17						✓	
Μαθητής 18						✓	
Μαθητής 19							✓

**Μεταβλητές  
Γνώση στη γλώσσα προγραμματισμού**

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης, επιλογής, γεγονότων και μεταβλητών.	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης, επιλογής, γεγονότων και μεταβλητών.	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
<b>Starlogo TNG</b>					
Μαθητής 1				√	
Μαθητής 2		√			
Μαθητής 3		√			
Μαθητής 4				√	
Μαθητής 5			√		
Μαθητής 6		√			
Μαθητής 7			√		
Μαθητής 8			√		
Μαθητής 9					√
Μαθητής 10			√		
Μαθητής 11			√		
Μαθητής 12				√	
Μαθητής 13			√		
Μαθητής 14			√		
Μαθητής 15		√			
Μαθητής 16		√			
Μαθητής 17			√		
Μαθητής 18			√		
<b>Scratch</b>					
Μαθητής 1			√		
Μαθητής 2			√		
Μαθητής 3			√		
Μαθητής 4				√	
Μαθητής 5		√			
Μαθητής 6					√
Μαθητής 7			√		
Μαθητής 8			√		
Μαθητής 9			√		
Μαθητής 10			√		
Μαθητής 11		√			
Μαθητής 12			√		
Μαθητής 13			√		
Μαθητής 14		√			
Μαθητής 15			√		
Μαθητής 16			√		
Μαθητής 17				√	
Μαθητής 18				√	
Μαθητής 19					√

**Μεταβλητές  
Αλγοριθμική σκέψη**

Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>Starlogo TNG</b>					
Μαθητής 1				√	
Μαθητής 2			√		
Μαθητής 3		√			
Μαθητής 4			√		
Μαθητής 5			√		
Μαθητής 6		√			
Μαθητής 7		√			
Μαθητής 8				√	
Μαθητής 9					√
Μαθητής 10			√		
Μαθητής 11			√		
Μαθητής 12				√	
Μαθητής 13			√		
Μαθητής 14			√		
Μαθητής 15		√			
Μαθητής 16			√		
Μαθητής 17			√		
Μαθητής 18			√		
<b>Scratch</b>					
Μαθητής 1			√		
Μαθητής 2		√			
Μαθητής 3			√		
Μαθητής 4				√	
Μαθητής 5		√			
Μαθητής 6					√
Μαθητής 7				√	
Μαθητής 8		√			
Μαθητής 9			√		
Μαθητής 10				√	
Μαθητής 11		√			
Μαθητής 12			√		
Μαθητής 13			√		
Μαθητής 14		√			
Μαθητής 15				√	
Μαθητής 16			√		
Μαθητής 17			√		
Μαθητής 18				√	
Μαθητής 19					√

### Μεταβλητές Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>Starlogo TNG</b>							
Μαθητής 1						✓	
Μαθητής 2			✓				
Μαθητής 3		✓					
Μαθητής 4					✓		
Μαθητής 5				✓			
Μαθητής 6		✓					
Μαθητής 7			✓				
Μαθητής 8					✓		
Μαθητής 9							✓
Μαθητής 10				✓			
Μαθητής 11				✓			
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14				✓			
Μαθητής 15		✓					
Μαθητής 16			✓				
Μαθητής 17				✓			
Μαθητής 18				✓			
<b>Scratch</b>							
Μαθητής 1				✓			
Μαθητής 2			✓				
Μαθητής 3					✓		
Μαθητής 4						✓	
Μαθητής 5		✓					
Μαθητής 6							✓
Μαθητής 7					✓		
Μαθητής 8			✓				
Μαθητής 9				✓			
Μαθητής 10					✓		
Μαθητής 11		✓					
Μαθητής 12				✓			
Μαθητής 13				✓			
Μαθητής 14		✓					
Μαθητής 15					✓		
Μαθητής 16				✓			
Μαθητής 17					✓		
Μαθητής 18						✓	
Μαθητής 19							✓

### Διαδικασίες Γνώση στη γλώσσα προγραμματισμού

Τμήμα / Μαθητής	Λανθασμένη χρήση εντολών	Ορθή χρήση απλών εντολών	Χρήση λογικών δομών, όπως επανάληψης, επιλογής, γεγονότων, μεταβλητών και διαδικασιών	Δεξιότητες προηγούμενου επιπέδου και επιπλέον ορθή χρήση και συνδυασμός κατάλληλων δομών επανάληψης, επιλογής, γεγονότων, μεταβλητών και διαδικασιών.	Βαθιά γνώση και φιλοσοφία της γλώσσας προγραμματισμού
<b>Starlogo TNG</b>					
Μαθητής 1				√	
Μαθητής 2		√			
Μαθητής 3		√			
Μαθητής 4			√		
Μαθητής 5			√		
Μαθητής 6		√			
Μαθητής 7			√		
Μαθητής 8			√		
Μαθητής 9					√
Μαθητής 10			√		
Μαθητής 11			√		
Μαθητής 12				√	
Μαθητής 13			√		
Μαθητής 14			√		
Μαθητής 15		√			
Μαθητής 16		√			
Μαθητής 17			√		
Μαθητής 18			√		
<b>Scratch</b>					
Μαθητής 1			√		
Μαθητής 2		√			
Μαθητής 3			√		
Μαθητής 4				√	
Μαθητής 5		√			
Μαθητής 6					√
Μαθητής 7				√	
Μαθητής 8		√			
Μαθητής 9			√		
Μαθητής 10			√		
Μαθητής 11			√		
Μαθητής 12			√		
Μαθητής 13			√		
Μαθητής 14		√			
Μαθητής 15			√		
Μαθητής 16			√		
Μαθητής 17			√		
Μαθητής 18				√	
Μαθητής 19					√

### Διαδικασίες Αλγοριθμική σκέψη

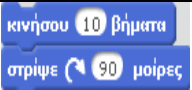
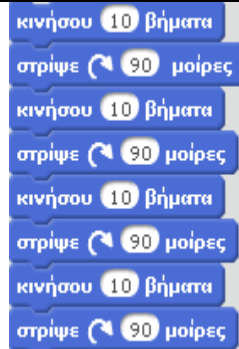
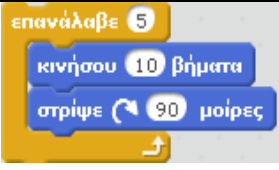
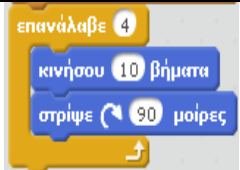
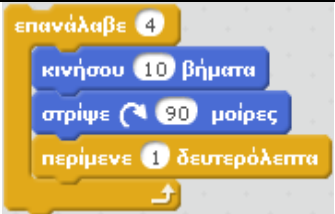
Τμήμα / Μαθητής	Μερική κατανόηση του προβλήματος	Κατανόηση του προβλήματος, ελλιπής αλγόριθμος	Σχεδιασμός της λύσης σε γενικό πλαίσιο χωρίς να λαμβάνονται υπόψη σημαντικές καταστάσεις	Οργάνωση και απόδοση της λύσης	Επιλογή και υλοποίηση του βέλτιστου αλγόριθμου
<b>Starlogo TNG</b>					
Μαθητής 1				√	
Μαθητής 2		√			
Μαθητής 3			√		
Μαθητής 4		√			
Μαθητής 5			√		
Μαθητής 6		√			
Μαθητής 7			√		
Μαθητής 8			√		
Μαθητής 9					√
Μαθητής 10			√		
Μαθητής 11		√			
Μαθητής 12				√	
Μαθητής 13			√		
Μαθητής 14			√		
Μαθητής 15		√			
Μαθητής 16		√			
Μαθητής 17		√			
Μαθητής 18			√		
<b>Scratch</b>					
Μαθητής 1		√			
Μαθητής 2			√		
Μαθητής 3		√			
Μαθητής 4			√		
Μαθητής 5		√			
Μαθητής 6					√
Μαθητής 7			√		
Μαθητής 8			√		
Μαθητής 9			√		
Μαθητής 10			√		
Μαθητής 11		√			
Μαθητής 12			√		
Μαθητής 13		√			
Μαθητής 14		√			
Μαθητής 15			√		
Μαθητής 16			√		
Μαθητής 17			√		
Μαθητής 18				√	
Μαθητής 19					√

### Διαδικασίες Επίπεδα

Τμήμα / Μαθητής	1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	Ενδιάμεσο 2ο - 3ο επίπεδο	3ο επίπεδο, προσεγγιστικής κατανόησης	Ενδιάμεσο 3ο - 4ο επίπεδο	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
<b>Starlogo TNG</b>							
Μαθητής 1						✓	
Μαθητής 2		✓					
Μαθητής 3			✓				
Μαθητής 4			✓				
Μαθητής 5				✓			
Μαθητής 6		✓					
Μαθητής 7				✓			
Μαθητής 8				✓			
Μαθητής 9							✓
Μαθητής 10				✓			
Μαθητής 11			✓				
Μαθητής 12						✓	
Μαθητής 13				✓			
Μαθητής 14				✓			
Μαθητής 15		✓					
Μαθητής 16		✓					
Μαθητής 17			✓				
Μαθητής 18				✓			
<b>Scratch</b>							
Μαθητής 1			✓				
Μαθητής 2			✓				
Μαθητής 3			✓				
Μαθητής 4						✓	
Μαθητής 5		✓					
Μαθητής 6							✓
Μαθητής 7					✓		
Μαθητής 8			✓				
Μαθητής 9				✓			
Μαθητής 10				✓			
Μαθητής 11			✓				
Μαθητής 12				✓			
Μαθητής 13			✓				
Μαθητής 14		✓					
Μαθητής 15				✓			
Μαθητής 16				✓			
Μαθητής 17				✓			
Μαθητής 18						✓	
Μαθητής 19							✓

## Β ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ ΜΕ ΙΕΡΑΡΧΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

### Παράδειγμα Επανάληψη Κίνηση τετράγωνου

1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	3ο επίπεδο, προσεγγιστικής κατανόησης	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
				
	Δεν χρησιμοποιεί καθόλου την επανάληψη	Χρειάζεται 4 επαναλήψεις	Είναι σωστή και αποδεκτή απάντηση (γίνεται όμως πολύ γρήγορα)	Ο μαθητής σε αυτήν την περίπτωση χωρίς να του αναφερθεί αποφάσισε να βάλει την τελευταία εντολή για να το βλέπει σε πιο αργή κίνηση

### Παράδειγμα Επιλογή Έλεγχος απάντησης σε ερώτηση

1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	3ο επίπεδο, προσεγγιστικής κατανόησης	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
				
	Δεν χρησιμοποιεί καθόλου την επιλογή	Δεν έλαβε υπόψη του τι θα γίνει αν απαντήσει λάθος	Είναι σωστή και αποδεκτή απάντηση (αλλά θα εκτελεστεί μια φορά και	Παίρνει την πρωτοβουλία να τερματίσει το πρόγραμμα μόνο αν δοθεί η σωστή



			δεν θα ξαναγίνει η ερώτηση σε περίπτωση λάθους. Θα πρέπει να ξανατρέξει το πρόγραμμα από την αρχή)	απάντηση.
--	--	--	--	-----------

**Παράδειγμα Γεγονότα**  
**Κίνηση χαρακτήρα-μορφής στη σκηνή**  
**με το πάτημα των πλήκτρων**

1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	3ο επίπεδο, προσεγγιστικής κατανόησης	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης
	Μόνο το X αυξάνεται	Το X και το Y αυξάνονται μόνο	Σωστή και αποδεκτή απάντηση το X και το Y αυξάνονται και μειώνονται με το πάτημα των πλήκτρων	Αποφασίζει χωρίς να του αναφερθεί να βελτιστοποιήσει το πρόγραμμα δίνοντας και κατεύθυνση στον χαρακτήρα

**Παράδειγμα Μεταβλητές**  
**Αρχικοποίηση και καταγραφή του σκορ με μεταβλητές**

1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	3ο επίπεδο, προσεγγιστικής κατανόησης	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης

	Δεν αλλάζει το σκορ	Αλλάζει το σκορ αλλά δεν το αρχικοποιεί	Σωστή και αποδεκτή απάντηση και ως προς την αρχικοποίηση και ως προς την αλλαγή του σκορ	Ο μαθητής από μόνος του σκέφτηκε να εισάγει και μια νέα μεταβλητή «ζωές» όπου θα καταγράφει και τις προσπάθειες

**Παράδειγμα Διαδικασίες  
Σχεδιασμός 3 τετραγώνων σε τυχαία θέση**

1ο επίπεδο, προδομικό	2ο επίπεδο, επιμέρους κατανόησης	3ο επίπεδο, προσεγγιστικής κατανόησης	4ο επίπεδο, συνδυαστικό	5ο επίπεδο, εκτεταμένης θεώρησης

	Φτιάχνει την διαδικασία αλλά δεν την καλεί στο κυρίως πρόγραμμα	Φτιάχνει την διαδικασία αλλά δεν την καλεί ορθά στο κυρίως πρόγραμμα	Σωστή και αποδεκτή λύση και στην κατασκευή και κλήση της διαδικασίας.	Ο μαθητής πήρε την πρωτοβουλία να ρωτήσει πόσα τετράγωνα θέλει ο χρήστης να κατασκευάσει. Αν η απάντηση είναι 3 τα κατασκευάζει.
--	---	--	---	--

## Γ ΕΡΩΤΗΜΑΤΟΛΟΓΙΑ

### ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΑΞΙΟΛΟΓΗΣΗΣ CODE.ORG – STARLOGO TNG

Α. ΠΡΟΣΩΠΙΚΑ ΣΤΟΙΧΕΙΑ	
Τμήμα	
Ηλικία	
Φύλλο	Αγόρι <input type="checkbox"/> Κορίτσι <input type="checkbox"/>

### Β. ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΗ ΧΡΗΣΗ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ

1. Πόσο άνετος/η θεωρείς ότι είσαι ως προς τη χρήση ηλεκτρονικού υπολογιστή;				
Καθόλου	Ελάχιστα	Αρκετά	Πολύ	Πάρα πολύ

2. Πόσες ώρες την εβδομάδα χρησιμοποιείς υπολογιστή στο σπίτι;				
Καθόλου	Λιγότερο από 2 ώρες	Από 2 έως 4 ώρες	Από 5 έως 10 ώρες	Περισσότερο από 11 ώρες

3. Χρησιμοποιείς τον υπολογιστή για					
	Ποτέ	Λίγες φορές	Κάποιες φορές	Πολλές φορές	Συνέχεια
A. Να παίζεις παιχνίδια					
B. Να ψάχνεις διάφορες πληροφορίες στο διαδίκτυο					

<b>C.</b> Να επικοινωνείς με φίλους / συγγενείς					
<b>D.</b> Να ακούς μουσική					
<b>E.</b> Να βλέπεις ταινίες					
<b>F.</b> Να ζωγραφίζεις					
<b>G.</b> Να γράφεις εργασίες για το σχολείο					
<b>H.</b> Να κάνεις κάτι άλλο; Τι; .....					
	.....				

### Γ. ΑΠΟΨΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

#### 4. Μου αρέσει ο Προγραμματισμός υπολογιστών

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

#### 5. Με ευχαριστεί να παρακολουθώ μαθήματα Προγραμματισμού

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

#### 6. Νιώθω άγχος κατά τη διάρκεια του μαθήματος του Προγραμματισμού

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

#### 7. Φοβάμαι γενικά τον Προγραμματισμό

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

<b>8. Είναι δύσκολος ο Προγραμματισμός</b>				
Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

**Δ. ΑΠΟΨΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ ΓΙΑ ΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ**

<b>9. Το Code.org είναι...</b>					
	Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ
<b>A.</b> Εύκολο					
<b>B.</b> Βαρετό					
<b>C.</b> Ενδιαφέρον					
<b>D.</b> Χρήσιμο					

<b>10. Το Code.org με βοήθησε</b>					
	Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
<b>A.</b> Να κατανοήσω καλύτερα τι είναι ο προγραμματισμός					
<b>B.</b> Να κατανοήσω τις προγραμματιστικές έννοιες που διδάχθηκα					
<b>C.</b> Να εντοπίσω και να διορθώσω τα λάθη που μπορεί να έκανα					
<b>D.</b> Να δημιουργήσω ενδιαφέροντα παιχνίδια					
<b>E.</b> Να διασκεδάσω					
<b>F.</b> Να πειραματιστώ					

<b>11. Το Starlogo TNG είναι ...</b>					
	Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ
<b>A.</b> Εύκολο					
<b>B.</b> Βαρετό					
<b>C.</b> Ενδιαφέρον					
<b>D.</b> Χρήσιμο					

<b>12. Το Starlogo TNG με βοήθησε</b>					
	Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
<b>A.</b> Να κατανοήσω καλύτερα τι είναι ο προγραμματισμός					
<b>B.</b> Να κατανοήσω τις προγραμματιστικές έννοιες που διδάχθηκα					
<b>C.</b> Να εντοπίσω και να διορθώσω τα λάθη που μπορεί να έκανα					
<b>D.</b> Να δημιουργήσω ενδιαφέροντα παιχνίδια					
<b>E.</b> Να διασκεδάσω					
<b>F.</b> Να πειραματιστώ					

<b>13. Έχεις χρησιμοποιήσει κάποιο από τα προγραμματιστικά περιβάλλοντα που γνώρισες στο σπίτι σου;</b>	
Ναι	Όχι

<b>14. Αν απάντησες με ΝΑΙ στην προηγούμενη ερώτηση ποιο περιβάλλον χρησιμοποίησες;</b>		
το Code.org	το Starlogo TNG	και τα δύο

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΑΞΙΟΛΟΓΗΣΗΣ CODE.ORG – SCRATCH

Α. ΠΡΟΣΩΠΙΚΑ ΣΤΟΙΧΕΙΑ	
Τάξη	
Ηλικία	
Φύλλο	Αγόρι <input type="checkbox"/> Κορίτσι <input type="checkbox"/>

**Β. ΑΠΟΨΕΙΣ ΜΑΘΗΤΩΝ ΓΙΑ ΤΗ ΧΡΗΣΗ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ**

1. Πόσο άνετος/η θεωρείς ότι είσαι ως προς τη χρήση ηλεκτρονικού υπολογιστή;				
Καθόλου	Ελάχιστα	Αρκετά	Πολύ	Πάρα πολύ

2. Πόσες ώρες την εβδομάδα χρησιμοποιείς υπολογιστή στο σπίτι;				
Καθόλου	Λιγότερο από 2 ώρες	Από 2 έως 4 ώρες	Από 5 έως 10 ώρες	Περισσότερο από 11 ώρες

3. Χρησιμοποιείς τον υπολογιστή για					
	Ποτέ	Λίγες φορές	Κάποιες φορές	Πολλές φορές	Συνέχεια
<b>A.</b> Να παίζεις παιχνίδια					
<b>B.</b> Να ψάχνεις διάφορες πληροφορίες στο διαδίκτυο					
<b>C.</b> Να επικοινωνείς με φίλους / συγγενείς					
<b>D.</b> Να ακούς μουσική					



<b>E.</b> Να βλέπεις ταινίες					
<b>F.</b> Να ζωγραφίζεις					
<b>G.</b> Να γράφεις εργασίες για το σχολείο					
<b>H.</b> Να κάνεις κάτι άλλο; Τι; .....					
	.....				

### Γ. ΑΠΟΨΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

#### 4. Μου αρέσει ο Προγραμματισμός υπολογιστών

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

#### 5. Με ευχαριστεί να παρακολουθώ μαθήματα Προγραμματισμού

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

#### 6. Νιώθω άγχος κατά τη διάρκεια του μαθήματος του Προγραμματισμού

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

#### 7. Φοβάμαι γενικά τον Προγραμματισμό

Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

<b>8. Είναι δύσκολος ο Προγραμματισμός</b>				
Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ

**Δ. ΑΠΟΨΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ ΓΙΑ ΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ**

<b>9. Το Code.org είναι...</b>					
	Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ
<b>A.</b> Εύκολο					
<b>B.</b> Βαρετό					
<b>C.</b> Ενδιαφέρον					
<b>D.</b> Χρήσιμο					

<b>10. Το Code.org με βοήθησε</b>					
	Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
<b>A.</b> Να κατανοήσω καλύτερα τι είναι ο προγραμματισμός					
<b>B.</b> Να κατανοήσω τις προγραμματιστικές έννοιες που διδάχθηκα					
<b>C.</b> Να εντοπίσω και να διορθώσω τα λάθη που μπορεί να έκανα					
<b>D.</b> Να δημιουργήσω ενδιαφέροντα παιχνίδια					
<b>E.</b> Να διασκεδάσω					
<b>F.</b> Να πειραματιστώ					

<b>11. Το Scratch είναι...</b>					
	Καθόλου	Λίγο	Έτσι και έτσι	Πολύ	Πάρα πολύ
<b>A.</b> Εύκολο					
<b>B.</b> Βαρετό					
<b>C.</b> Ενδιαφέρον					
<b>D.</b> Χρήσιμο					

<b>12. Το Scratch με βοήθησε</b>					
	Καθόλου	Λίγο	Μέτρια	Πολύ	Πάρα πολύ
<b>A.</b> Να κατανοήσω καλύτερα τι είναι ο προγραμματισμός					
<b>B.</b> Να κατανοήσω τις προγραμματιστικές έννοιες που διδάχθηκα					
<b>C.</b> Να εντοπίσω και να διορθώσω τα λάθη που μπορεί να έκανα.					
<b>D.</b> Να δημιουργήσω ενδιαφέροντα παιχνίδια					
<b>E.</b> Να διασκεδάσω					
<b>F.</b> Να πειραματιστώ					

<b>13. Έχεις χρησιμοποιήσει κάποιο από τα προγραμματιστικά περιβάλλοντα που γνώρισες στο σπίτι σου;</b>	
Ναι	Όχι

<b>14. Αν απάντησες με ΝΑΙ στην προηγούμενη ερώτηση ποιο περιβάλλον χρησιμοποίησες;</b>		
το Code.org	το Scratch	και τα δύο