

Η χρήση της δομής επανάληψης ΟΣΟ στο ApplInventor

Πίνακας περιεχομένων

1. Τίτλος διδακτικού σεναρίου.....	2
2. Εκτιμώμενη διάρκεια διδακτικού σεναρίου	2
3. Ένταξη του διδακτικού σεναρίου στο πρόγραμμα σπουδών	2
4. Σκοποί και στόχοι του διδακτικού σεναρίου	3
5. Περιγραφή του διδακτικού σεναρίου	4
6. Επιστημολογική προσέγγιση και εννοιολογική ανάλυση - Θέματα θεωρίας του διδακτικού σεναρίου	10
7. Χρήση Η.Υ. και γενικά ψηφιακών μέσων για το διδακτικό σενάριο («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)	11
8. Αναπαραστάσεις των μαθητών/πρόβλεψη δυσκολιών στο διδακτικό σενάριο	12
9. Διδακτικό συμβόλαιο –Διδακτική μετατόπιση –Θεωρητικά θέματα –Διδακτικός θόρυβος	13
10. Χρήση εξωτερικών πηγών	13
11. Υποκείμενη θεωρία μάθησης	15
12. Επισήμανση μικρομεταβολών.....	16
13. Οργάνωση της τάξης – Εφικτότητα σχεδίασης.....	16
14. Επεκτάσεις / διασυνδέσεις των εννοιών ή των δραστηριοτήτων	16
15. Περιγραφή και ανάλυση φύλλων εργασίας	17
16. Αξιολόγηση	17
17. Το επιμορφωτικό σενάριο	17
18. Φύλλο εργασίας 1 –.....	17
Δραστηριότητα 1	17
Δραστηριότητα 2	19
Δραστηριότητα 3	20
Δραστηριότητα 4	21
Δραστηριότητα 5	23
19. Φύλλο εργασίας 2.....	25
Δραστηριότητα 1	25
Δραστηριότητα 2	28
20. Φύλλο εργασίας 3 - Επανάληψη - Εμπέδωση	30
21. Προτάσεις για περαιτέρω δραστηριότητες - προτεινόμενες εργασίες.....	31

Η χρήση της δομής επανάληψης ΟΣΟ στο ApplInventor

1. Τίτλος διδακτικού σεναρίου

Η χρήση της δομής επανάληψης ΟΣΟ στο ApplInventor

2. Εκτιμώμενη διάρκεια διδακτικού σεναρίου

Η προβλεπόμενη διάρκεια του σεναρίου είναι 2 διδακτικές ώρες

3. Ένταξη του διδακτικού σεναρίου στο πρόγραμμα σπουδών

Το διδακτικό σενάριο, εντάσσεται στο πρόγραμμα σπουδών της δευτεροβάθμιας βαθμίδας εκπαίδευσης (Γυμνάσια και τη μεταϋποχρεωτική Δευτεροβάθμια Εκπαίδευση που προσφέρεται από το Γενικό και το Τεχνολογικό Λύκειο).

Πιο αναλυτικά στα Νέα Προγράμματα Σπουδών Υποχρεωτικής εκπαίδευσης (Γυμνάσιο & Δημοτικό) ο πληροφορικός γραμματισμός (ICT Literacy) θεωρείται γνωστικό-μαθησιακό αντικείμενο αντίστοιχης σπουδαιότητας με τον γλωσσικό γραμματισμό (literacy), τα μαθηματικά και τον επιστημονικό γραμματισμό (scientific literacy). Ως εκ τούτου, η διδασκαλία των αντικειμένων της Πληροφορικής στο Γυμνάσιο και στο Δημοτικό δεν έχει ως στόχο την κατάρτιση των μαθητών σε εφήμερες τεχνολογικές δεξιότητες. Περιλαμβάνει δηλαδή πολλά περισσότερα από την απλή εξοικείωση με τους υπολογιστές και τις λειτουργίες συγκεκριμένων λογισμικών, και στοχεύει στην ανάπτυξη ικανοτήτων αυτόνομης αξιοποίησης των υπολογιστικών και δικτυακών εργαλείων για την επίλυση προβλημάτων, την ενημέρωση, την ψυχαγωγία και, γενικά, τη συμμετοχή τους στη σύγχρονη κοινωνία. Το νέο Π.Σ. παρέχει την ελευθερία στον καθηγητή να σχεδιάσει τη διδασκαλία του, ανάλογα με τις γνώσεις και τις ιδιαιτερότητες των μαθητών του και την υλικοτεχνική υποδομή που έχει στη διάθεσή του. Καλείται δηλαδή να μετατραπεί σε παραγωγό εκπαιδευτικού περιεχομένου που θα ικανοποιεί τις εκάστοτε ιδιαίτερες συνθήκες εφαρμογής του περιεχομένου αυτού.

Στο Γυμνάσιο και στις 3 τάξεις υπάρχει η ενότητα 'Προγραμματίζω τον υπολογιστή μου'. Βασικός στόχος της ενότητας αυτής είναι η σταδιακή εξοικείωση των μαθητών με τον προγραμματισμό μέσα από την αξιοποίηση διαθέσιμων εκπαιδευτικών περιβαλλόντων οπτικού προγραμματισμού. Ειδικότερα αναφέρεται ότι με την ολοκλήρωση της ενότητας ο μαθητής/τρια εκτός των άλλων θα πρέπει να είναι ικανός/ή:

να αντιλαμβάνεται την αναγκαιότητα και τη χρησιμότητα των δομών επανάληψης και επιλογής να χρησιμοποιεί εντολές επανάληψης και επιλογής στα προγράμματα που αναπτύσσει.

Για την υλοποίηση των δραστηριοτήτων γίνεται αναφορά σε ποικίλα προγραμματιστικά

περιβάλλοντα όπως Scratch, BYOB, Microworld pro, Starlogo TNG, Turtle Art, Kodu, Storytelling Alice, gameMaker, GreenFoot, e-toy.

Αναφορικά με τα προγράμματα σπουδών του Γενικού και του Τεχνολογικού Λυκείου για τη διδασκαλία της πληροφορικής ένας από τους στόχους που τίθεται είναι να επεκτείνουν οι μαθητές τη γενική πληροφορική παιδεία τους με έμφαση στην ανάπτυξη ικανοτήτων και δεξιοτήτων στη χρήση και αξιοποίηση των υπολογιστικών και δικτυακών τεχνολογιών ως εργαλείων μάθησης και σκέψης. Άλλωστε στα πλαίσια αυτά προβλέπονται συνθετικές εργασίες με λογισμικό εφαρμογών γενικής χρήσης, εκπαιδευτικό λογισμικό και προγραμματιστικά περιβάλλοντα.

Το λογισμικό AppInventor παρότι δεν αναφέρεται ούτε στα πλέον πρόσφατα προγράμματα σπουδών (Δημοτικού & Γυμνασίου) κρίνουμε ότι μπορεί να χρησιμοποιηθεί καθώς πρόκειται για ένα ιδιαίτερα εύχρηστο αλλά και 'ισχυρό' προγραμματιστικό περιβάλλον που μπορεί να ανταποκριθεί και στις πιο απαιτητικές μαθησιακές δραστηριότητες σε όλες τις εκπαιδευτικές βαθμίδες καθώς κατασκευάστηκε για να διευκολύνει τη δημιουργία αλληλεπιδραστικών εφαρμογών από παιδιά και νέους αφού ο προγραμματισμός γίνεται με οπτικό τρόπο και η γλώσσα προγραμματισμού περιλαμβάνει διαισθητικό χειρισμό πολυμέσων.

4. Σκοποί και στόχοι του διδακτικού σεναρίου

Σκοπός του σεναρίου είναι οι μαθητές να κατανοήσουν οι μαθητές τα χαρακτηριστικά και τη λειτουργία της επαναληπτικής δομής (Όσο.....επανάλαβε).

Στόχοι του σεναρίου είναι να:

- αναγνωρίζουν την εντολή αρχικοποίησης της τιμής της μεταβλητής ελέγχου
- αναγνωρίζουν την εντολή ανανέωσης της τιμής της μεταβλητής ελέγχου
- ελέγχουν αν μια δομή επανάληψης «Όσο.....επανάλαβε» εκτελείται ή όχι
- ελέγχουν αν μια δομή επανάληψης «Όσο.....επανάλαβε» τερματίζει την εκτέλεσή της ή όχι
- εξηγούν πότε τερματίζεται η εκτέλεση της επαναληπτικής δομής «Όσο.....επανάλαβε»
- εφαρμόζουν την επαναληπτική δομή «Όσο.....επανάλαβε» στην επίλυση ενός προβλήματος με την χρήση του AppInventor

Προηγούμενες γνώσεις απαραίτητες για την χρήση/ολοκλήρωση της εισαγωγής στις βασικές προγραμματιστικές έννοιες των δομών επανάληψης:

- Οι μαθητές γνωρίζουν το προγραμματιστικό περιβάλλον του AppInventor. Είναι εξοικειωμένοι με τις έννοιες των αντικειμένων, των ιδιοτήτων τους και τον χειρισμό συμβάντων.
- Οι μαθητές, γνωρίζουν ήδη την δομή ακολουθίας καθώς και την έννοια της

μεταβλητής.

- Οι μαθητές, γνωρίζουν τον τρόπο ορισμού και χρήσης των μεταβλητών στο AppInventor.
- Οι μαθητές, γνωρίζουν την λειτουργία των αριθμητικών και λογικών τελεστών και τον τρόπο σύνταξης τους στο AppInventor.
- Οι μαθητές, γνωρίζουν την λειτουργία της δομής επιλογής και τον τρόπο σύνταξης της στο AppInventor.
- Οι μαθητές, κατανοούν την σκοπιμότητα χρήσης εμφωλευμένων δομών επιλογής και τον τρόπο σύνταξης τους στο AppInventor.
- Οι μαθητές είναι σε θέση να ανακτούν, αποθηκεύουν, πακετάρουν και να δημιουργούν απλά προγράμματα με τη χρήση του AppInventor.

5. Περιγραφή του διδακτικού σεναρίου

Η βελτιστοποίηση της διδασκαλίας εννοιών προγραμματισμού είναι ένα από τα διαχρονικά ζητήματα για τη διδακτική της Πληροφορικής (Γόγουλου, Α., Γουλή, Ε., & Γρηγοριάδου, Μ. 2009). Η δομή επανάληψης παρουσιάζει αρκετές δυσκολίες στην οικοδόμησή της (Κόμης, 2012) και ως εκ τούτου οι αρχάριοι προγραμματιστές δεν χρησιμοποιούν αυθόρμητα την επαναληπτική διαδικασία. Τα νοητικά μοντέλα των μαθητών για την επανάληψη έχουν συνήθως την ακόλουθη δομή:

*Περιγραφή δράσης → μετρητής επαναλήψεων → προσδιορισμός επανάληψης
→ συνθήκη ελέγχου*

Συνεπώς, η δομή ΕΠΑΝΑΛΑΒΕ ... ΜΕΧΡΙΣ_ΟΤΟΥ ταιριάζει περισσότερο στις αρχικές ιδέες των μαθητών ενώ η δομή ΟΣΟ συνθήκη ΕΠΑΝΑΛΑΒΕ προκαλεί περισσότερα προβλήματα κατανόησης καθώς στη δομή αυτή είναι δυνατόν να μην υπάρξει επανάληψη (γεγονός που έρχεται σε αντίθεση με την αναπαράσταση της επαναληπτικής διαδικασίας που διαθέτουν οι μαθητές).

Η έρευνα για τη διδακτική του προγραμματισμού έχει αποκαλύψει ποικίλα προβλήματα, μεταξύ των οποίων και η εγκυρότητα του μαθήματος στα μάτια των μαθητών (Kolikant, 2004) ενώ συχνά, ούτε οι χρησιμοποιούμενες στη διδασκαλία (εκπαιδευτικές) γλώσσες προγραμματισμού νομιμοποιούνται στα μάτια των μαθητών. Δηλαδή, πολύ συχνά οι μαθητές αμφισβητούν τη χρησιμότητα μιας εκπαιδευτικής γλώσσας όπως η Logo, σε αντιδιαστολή με εμπορικές γλώσσες (π.χ. C, C++, Java). Το μάθημα του προγραμματισμού μέσα από το σχεδιασμό ενός ηλεκτρονικού παιχνιδιού (Papert, 1998) διδάσκεται πιο αποδοτικά, ελκυστικά, διασκεδαστικά γιατί η μάθηση η βασιζόμενη στην τεχνολογία, μπορεί να είναι διασκεδαστική (Rickel et al, 2001). Προκαλεί την άμεση ικανοποίηση του μαθητή η οποία επιταχύνει τον κύκλο μάθησης (Ο'Kelly et al., 2006). Προσελκύει το ενδιαφέρον των μαθητών οι οποίοι αφιερώνουν πολλές ώρες σε ηλεκτρονικά

παιχνίδια (Malone, 1981). Παρουσιάζει ενδιαφέρον γιατί συνδυάζει θέματα τεχνητής νοημοσύνης, προσομοίωσης και άλλων προσεγγίσεων όπως εφαρμογών κινητών τηλεφώνων κ.α. (Shin, et al., 2006). Μπορεί να έχει εκπαιδευτικό χαρακτήρα και αποτέλεσμα (Papert, 1998). Βοηθά τους μαθητές να προσεγγίζουν την τεχνολογία με ένα πιο ευφάνταστο στυλ που ενθαρρύνει την ανάπτυξη δεξιοτήτων (Martocchio et al., 1992).

Το εκπαιδευτικό σενάριο που θα περιγράψουμε μπορεί να εφαρμοστεί τόσο στο Γυμνάσιο όσο και στο Γενικό και Τεχνολογικό Λύκειο, εφόσον η δομή επανάληψης ΟΣΟ αποτελεί διδακτικό στα μαθήματα και των 2 σχολικών βαθμίδων. Σκοπός του είναι η διδασκαλία στους μαθητές των εννοιών που σχετίζονται με την ορθή χρήση της δομής επανάληψης ΟΣΟ. Οι μαθητές εργάζονται ατομικά ή σε ομάδες 2-3 μαθητών (ανάλογα την διαθεσιμότητα του εργαστηρίου Πληροφορικής). Ο εκπαιδευτικός μοιράζει πολλαπλά φύλλα εργασίας. Οι μαθητές μέσω των φύλλων εργασίας και των δραστηριοτήτων τους προσπαθούν να γνωρίσουν και να εμβαθύνουν στην έννοια της δομής επανάληψης ΟΣΟ.

Ο εκπαιδευτικός δεν περιορίζεται στο μοίρασμα δραστηριοτήτων και στην παθητική παρακολούθηση αλλά συμμετέχει ενεργά στη μαθησιακή διαδικασία, παρακολουθώντας και ελέγχοντας τις απαντήσεις των μαθητών, συμβουλευοντας τους κριτικά κατά τη φάση επίλυσης των ασκήσεων ή ανάπτυξης των αλγορίθμων ενθαρρύνοντας τους να συνεχίσουν την διερευνητική τους προσπάθεια μέσω στοχευόμενων ερωτήσεων και προβληματισμών. Συνοψίζοντας η βοήθεια που προσφέρει ο εκπαιδευτικός στους μαθητές, άλλοτε ρητά και άλλοτε άρρητα μπορεί να είναι είτε υποστηρικτική ή και συνερευνητική ή και καθοδηγητική έχοντας υπόψη του ότι οι μαθητές μαθαίνουν αλληλεπιδρώντας με το περιβάλλον τους και ότι τα προβλήματα αποτελούν το κριτήριο και την πηγή της γνώσης.

Αναγκαίο είναι ο εκπαιδευτικός να έχει ετοιμάσει τα προγράμματα σε διάφορες ανεξάρτητες εκδοχές τους (π.χ. το σχεδιαστικό κομμάτι καθώς και το ολοκληρωμένο πρόγραμμα) και να έχει φροντίσει ιδίως τα προγράμματα με το σχεδιαστικό τμήμα να τα έχει ήδη τοποθετήσει σε ένα δημόσιο αποθετήριο προσβάσιμο από τους μαθητές είτε τοπικά στους ΗΥ του εργαστηρίου Πληροφορικής. Αυτονόητο είναι οι μαθητές να μην αφιερώνουν χρόνο στη σχεδίαση των εφαρμογών αλλά η προσπάθεια τους να επικεντρωθεί στην υλοποίηση του προγραμματικού τμήματος των δραστηριοτήτων.

Μια ενδεικτική πορεία διδασκαλίας είναι η ακόλουθη:

Ο εκπαιδευτικός προτού καν πραγματοποιήσει οποιαδήποτε αναφορά στο προγραμματιστικό περιβάλλον, προσπαθώντας να κερδίσει το ενδιαφέρον των μαθητών, μπορεί να αναφέρει παραδείγματα που σχετίζονται με τη καθημερινή ζωή των μαθητών όπως π.χ. αναφέροντας τους ότι επειδή ήταν αδιάβαστοι στο σημερινό μάθημα ο καθηγητής πληροφορικής τους βάζει τιμωρία να γράψουν στο χαρτί 100 φορές την πρόταση 'δεν θα είμαι αδιάβαστος'. Επειδή οι μαθητές δεν έχουν εναλλακτική λύση, θα αφιερώνουν αρκετό χρόνο να γράψουν 100 φορές την

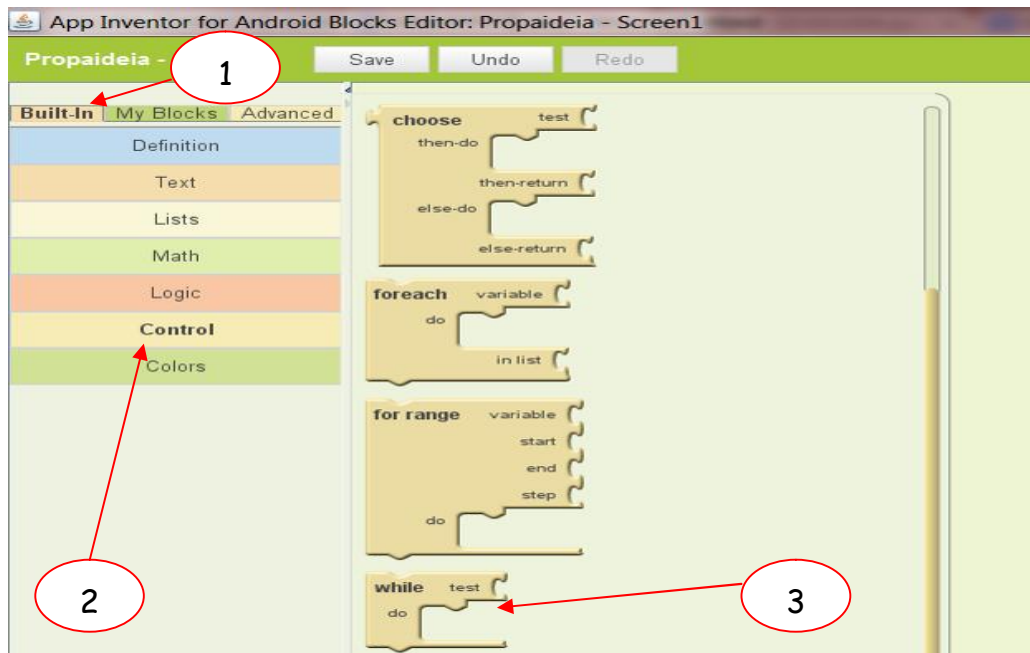
ζητούμενη πρόταση στο χαρτί. Αν ο εκπαιδευτικός τους έλεγε αντί να γράψουν την τιμωρία τους στο χαρτί να την γράψουν σε μορφή προγράμματος στον υπολογιστή εξίσου 100 φορές θεωρούν ότι θα ήταν λιγότερο κουραστικό και χρονοβόρο για αυτούς; Ο εκπαιδευτικός αφού ακούσει τις σκέψεις των μαθητών, πρέπει να τους επισημάνει ότι ο κάματος γι' αυτούς θα ήταν σχεδόν ο ίδιος αφού με τις υπάρχουσες γνώσεις τους (δομή ακολουθίας και επιλογής) δεν θα είχαν άλλη εναλλακτική από το να γράψουν 100 φορές μια εντολή της μορφής Γράψε 'δεν θα είμαι αδιάβαστος'. Στο σημείο αυτό πρέπει να τους εισάγει τον προβληματισμό αν θα ήταν πιο εύκολο και χρήσιμο γι' αυτούς να υπήρχε η δυνατότητα σχεδόν με μια μόνο εντολή να 'λέγανε' στον υπολογιστή 100 φορές να εμφανίσει το ζητούμενο μήνυμα; Σκοπός είναι να προβληματίσει τους μαθητές στο γεγονός ότι πολύ συχνά για την επίλυση καθημερινών προβλημάτων, χρειάζεται κάποιες ενέργειες να τις πράξουμε περισσότερες από μια φορές (π.χ. μια τράπεζα υπολογίζει τόκους ανά εξάμηνο ή προγραμματιστικά κάποια τμήματα του αλγόριθμου πρέπει να εκτελεστούν περισσότερες από μια φορές) και ως εκ τούτου 'κάποιοι' θα έχουν φροντίσει ώστε με εύκολο τρόπο οι γλώσσες προγραμματισμού να εκτελούν τις διάφορες επαναληπτικές διαδικασίες.

Ο εκπαιδευτικός στη συνέχεια εξηγεί την γενική μορφή της δομής επανάληψης ΟΣΟ, τονίζοντας ότι ανάλογα την τιμή της συνθήκης υπάρχει περίπτωση οι εντολές που περικλείονται στο σώμα της δομής να μην εκτελεστούν καμία φορά, μια, ή περισσότερες ή και για πάντα. Επίσης ο εκπαιδευτικός τονίζει στους μαθητές ότι θα πρέπει να είναι ιδιαίτεροι προσεκτικοί όταν χρησιμοποιούν την δομή επανάληψης ΟΣΟ, διότι υπάρχει περίπτωση να δημιουργήσουν έναν ατέρμων βρόγχο (ή άπειρη ανακύκλωση) με αποτέλεσμα ο αλγόριθμος τους να μην ολοκληρώνεται ποτέ. Ιδιαίτερης βαρύτητας είναι η επισήμανση εκ μέρους του εκπαιδευτικού ότι οι μαθητές θα πρέπει να δίνουν κατάλληλες τιμές στις μεταβλητές που χρησιμοποιούν για τον έλεγχο της δομής (π.χ. αρχικοποίηση μετρητή, ορθή χρήση λογικών τελεστών και ανισότητας για τον έλεγχο της συνθήκης, ορθή αύξηση του μετρητή). Χρήσιμη είναι η χρήση εκ μέρους του εκπαιδευτικού παραδειγμάτων με ακραίες τιμές όπως π.χ. αρχικοποίηση μετρητή με 0 ή 1 και έλεγχος συνθήκης με τους τελεστές $\leq$ & $<$ ή $\geq$ & $>$ και μέτρηση του πλήθους των επαναλήψεων κ.α. στοχεύοντας όπως έχουμε ήδη στο να προκαλέσει το ενδιαφέρον των μαθητών ώστε να ενταχθούν ενεργητικά στη διαδικασία της μάθησης.

Για την διδασκαλία του σεναρίου θα χρησιμοποιήσουμε το AppInventor (AI) (εφευρέτης εφαρμογών). Το AI αποτελεί ένα νέο δωρεάν οπτικό περιβάλλον προγραμματισμού με πλακίδια (blocks), για τη δημιουργία εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα Android. Οι συγκεκριμένες εφαρμογές τρέχουν και σε προσομοιωτή (emulator). Πρόκειται για ένα δικτυακό περιβάλλον ανάπτυξης εφαρμογών το μεγαλύτερο πλεονέκτημα του οποίου είναι ότι δεν απαιτεί ιδιαίτερες γνώσεις στον προγραμματισμό ή εναλλακτικά η εκμάθηση του λόγω της παιγνιώδους φύσης του είναι αρκετά εύκολη συγκρινόμενη μ' άλλα προγραμματιστικά

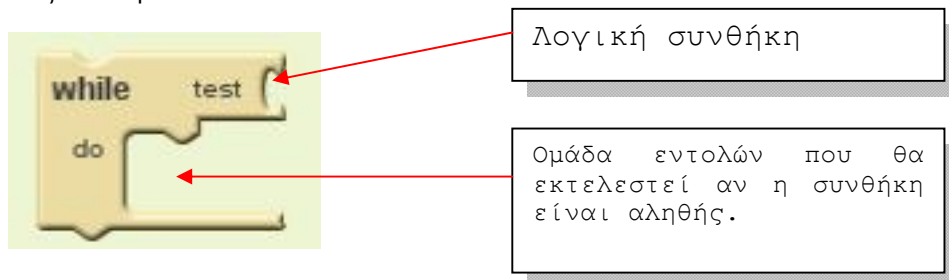
περιβάλλοντα (Lohr, 2010).

Ας δούμε πως το AppInventor χειρίζεται την δομή επανάληψης ΟΣΟ. Καταρχήν να υπενθυμίσουμε ότι ο χειρισμός του προγραμματιστικού τομέα της εφαρμογής γίνεται από τον Block editor. Εφόσον λοιπόν έχουμε μεταβεί σε αυτόν, μπορούμε να δούμε τα εργαλεία που μας παρέχει το AppInventor για τον χειρισμό της δομής επανάληψης ΟΣΟ. Πηγαίνοντας στην ενότητα Built-in (βήμα 1) και επιλέγοντας την κατηγορία Control (βήμα 2), μπορούμε να δούμε το πλακίδιο που έχουμε στην διάθεση μας για τον χειρισμό της δομής επανάληψης ΟΣΟ (βήμα 3), όπως φαίνεται στην εικόνα 1.



Εικόνα 1. Χειρισμός δομής επανάληψης ΟΣΟ στο AppInventor

Η δομή επανάληψης **ΟΣΟ** (συνθήκη=αληθινή) **επανάλαβε** Ομάδα εντολών **Τέλος_επανάληψης**, παρουσιάζεται στην εικόνα 2.



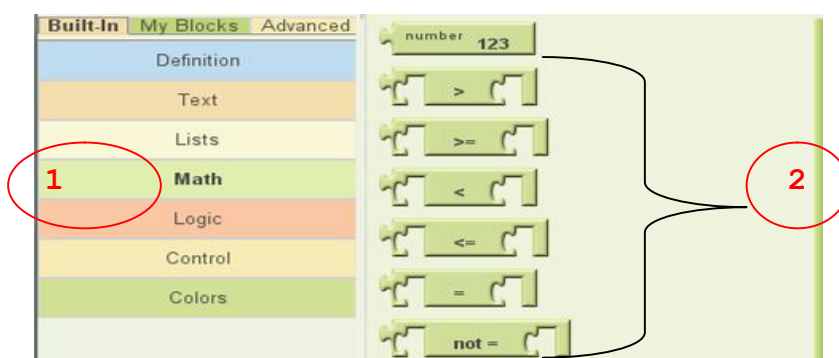
Εικόνα 2. Η δομή επανάληψης ΟΣΟ στο AppInventor

Υπενθυμίζουμε ότι για τον έλεγχο της συνθήκης μπορούμε να μεταβούμε στην κατηγορία Logic (βήμα 1) και να χρησιμοποιήσουμε είτε τους λογικούς τελεστές **ΚΑΙ**, **Η**, **ΟΧΙ** για τον έλεγχο σύνθετων λογικών συνθηκών (βήμα 2) ενώ αν μια συνθήκη είναι αληθής ή ψευδής την συγκρίνουμε με τα πλακίδια true και false (βήμα 3).



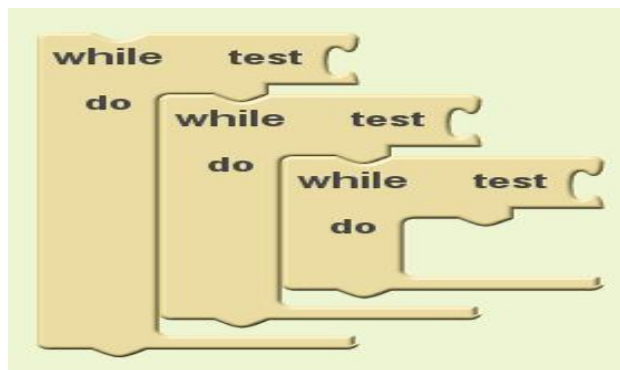
Εικόνα 3. Χειρισμός λογικών τελεστών στο AppInventor

Επίσης αναφέρουμε ξανά ότι στο AppInventor υπάρχουν και οι τελεστές ανισότητας (εικόνα 4).



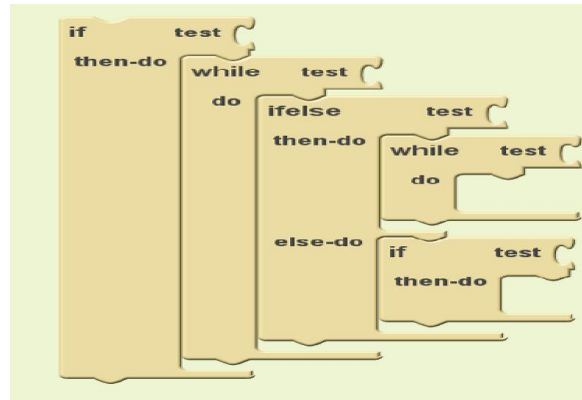
Εικόνα 4. Χειρισμός τελεστών ανισότητας στο AppInventor

Τέλος με την ορθή τοποθέτηση των δομών επανάληψης, μπορούμε να υλοποιήσουμε εμφωλευμένες δομές επανάληψης, όπως φαίνεται στο παράδειγμα της εικόνας 5.



Εικόνα 5. Δημιουργία εμφωλευμένων δομών επανάληψης στο AppInventor

Τέλος όπως είναι λογικό μπορούμε να έχουμε συνδυασμό και των 3 αλγοριθμικών δομών (ακολουθίας, επιλογής, επανάληψης), όπως φαίνεται στο παράδειγμα της εικόνας 6.



Εικόνα 6. Δημιουργία εμφωλευμένων αλγοριθμικών δομών στο AppInventor

6. Επιστημολογική προσέγγιση και εννοιολογική ανάλυση - Θέματα θεωρίας του διδακτικού σεναρίου

Το App Inventor (AI) ανακοινώθηκε για πρώτη φορά ως ένα μικρό έργο των εργαστηρίων της Google (Google Lab) στα τέλη του 2010. Αναπτύχθηκε στα εργαστήρια της Google από μια ομάδα με επικεφαλής τον καθηγητή του MIT Harold Abelson (Abelson, 2009) και βασίστηκε σε προηγούμενες μελέτες βασισμένες στη χρήση γραφικών περιβαλλόντων προγραμματισμού όπως το StarLogo TNG (Starlogo TNG, 2012) και τη βιβλιοθήκη Open Blocks (Open Blocks, 2012), ένα σύστημα προγραμματισμού που αναπτύχθηκε για εκπαιδευτικούς λόγους στο MIT. Στη συνέχεια, μεταφέρθηκε στο κέντρο για τη εκμάθηση της φορητής μάθησης του MIT (Mobile Learning Center, 2011) για δημόσια χρήση ως λογισμικό ανοικτού κώδικα. Το AI αποτελεί ένα νέο δωρεάν οπτικό περιβάλλον προγραμματισμού με πλακίδια (blocks), για τη δημιουργία εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα Android. Οι συγκεκριμένες εφαρμογές τρέχουν και σε προσομοιωτή (emulator).

Αναφορικά με την εκμάθηση προγραμματισμού μια μετανάλυση των Dehnadi et al., (2009) για την διερεύνηση των παραγόντων που επηρεάζουν την επιτυχία στον προγραμματισμό διαπίστωσε ότι ένας ισχυρός παράγοντας είναι το γνωστικό φορτίο το οποίο απαιτεί το κάθε προγραμματιστικό περιβάλλον από τον χρήστη. Ο Sweller (2010) αναφέρει ότι προκειμένου να μειώσουμε το ενδογενές νοητικό φορτίο για αρχάριους στο προγραμματισμό χρήστες αρκεί να μειώσουμε την ποσότητα της πληροφορίας που απαιτείται να χρησιμοποιήσουν προκειμένου να επιλύσουν το πρόβλημα. Αυτό μπορεί να επιτευχθεί με την απομόνωση προγραμματιστικών εννοιών ώστε οι χρήστες να μην είναι υποχρεωμένοι να θυμούνται αρκετές προγραμματιστικές γνώσεις ταυτόχρονα. Προγραμματιστικά περιβάλλοντα τύπου drag & drop όπως το AI και το Scratch, αντικαθιστούν τον προς συγγραφή κώδικα με οπτικά αντικείμενα τα οποία επιλέγονται μέσω ενός μενού επιλογών μειώνοντας το νοητικό φορτίο που απαιτείται για την συγγραφή κώδικα και ταυτόχρονα βοηθώντας τους χρήστες να επικεντρωθούν στην επίλυση ενός προβλήματος (Brennan, 2009). Επίσης οι Resnick et al., (2009) αναφέρουν ότι προγραμματιστικά περιβάλλοντα αυτού του τύπου θεωρούνται εύκολα στην εκμάθηση τους για όλες τις ηλικίες, διαφορετικά εκπαιδευτικά υπόβαθρα και ενδιαφέροντα καθώς επιτρέπουν στους χρήστες να πειραματίζονται με προγραμματιστικές δομές απλά ενώνοντας κομμάτια κώδικα με παρόμοιο τρόπο που συνδέουν τουβλάκια τύπου Lego. Σύμφωνα με τους ίδιους ερευνητές η παραπάνω προσέγγιση είναι ιδανική για αρχάριους στον προγραμματισμό χρήστες καθώς τους προσφέρεται η δυνατότητα να επικεντρωθούν στη δομή των λύσεων παρά στη σύνταξη προγραμματιστικών εντολών.

Το AppInventor αποτελεί ένα θαυμάσιο προγραμματιστικό περιβάλλον που δίνει την ευκαιρία στους χρήστες να "οικοδομούν πράγματα με νόημα" με σχετική ευκολία και λειτουργεί ως

καταλύτης για πλούσιες μαθησιακές ευκαιρίες. Το σενάριο είναι βασισμένο στην εποικοδομητική θεωρία του Papert (1980) σύμφωνα με την οποία σημασία έχει να δίνεις στα παιδιά καλά πράγματα να κάνουν έτσι ώστε να μπορούν να μάθουν κάνοντας πολύ καλύτερα απ' ότι μπορούσαν πριν. Κατά συνέπεια το διδακτικό σενάριο είναι βασισμένο στη θεωρία μάθησης του εποικοδομητισμού (Constructivism).

7. Χρήση Η.Υ. και γενικά ψηφιακών μέσων για το διδακτικό σενάριο («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Για την διδασκαλία του σεναρίου θα χρειαστεί ένα εργαστήριο πληροφορικής που να έχει πρόσβαση στο διαδίκτυο για να μπορούν οι μαθητές να προσπελάζουν το διαδικτυακό περιβάλλον δημιουργίας του AppInventor. Επίσης θα πρέπει τοπικά στους σταθμούς εργασίας να έχει εγκατασταθεί μια οποιαδήποτε σχετικά σύγχρονη έκδοση της java (www.java.com). Επιθυμητό είναι να υπάρχει ένας video projector (ή εφόσον πρόκειται για Γυμνάσιο και ένας διαδραστικός πίνακας) για να μπορεί ο εκπαιδευτικός να παρουσιάζει το περιβάλλον του AppInventor και ότι άλλο αυτός κρίνει απαραίτητο. Επίσης θα μπορούσε ο εκπαιδευτικός προκειμένου να κεντρίσει και να κερδίσει το ενδιαφέρον των μαθητών, να τους πει να κρατούν τα κινητά τους τηλέφωνα ή τις ταμπλέτες τους (αν διαθέτουν) μαζί με τα USB καλώδια που τα συνοδεύουν εντός του εργαστηρίου εφόσον όμως είναι τύπου smartphone και τρέχουν λειτουργικό σύστημα Android. Εναλλακτικά εφόσον το εργαστήριο έχει πρόσβαση σε ασύρματο δίκτυο, οι μαθητές απλά θα κρατούν μόνο τις συσκευές τους καθώς η σύνδεση θα γίνεται ασύρματα.

Όπως ήδη αναφέραμε το περιβάλλον AppInventor, δημιουργία του MIT υπό την αιγίδα του καθηγητή Hal Abelson, είναι το πλέον σύγχρονο και εύκολο περιβάλλον για μια εισαγωγή στον προγραμματισμό. Όπως χαρακτηριστικά αναφέρει ο Hal Abelson σε συνέντευξη του στους New York Times τον Ιούλιο του 2010, ο γενικότερος σκοπός ανάπτυξης του App Inventor είναι να επιτρέψει στους χρήστες συσκευών με εγκατεστημένο λογισμικό Android να μετατραπούν από καταναλωτές σε δημιουργούς. Το συγκεκριμένο πρόγραμμα της Google και του MIT, ανέφερε ο Abelson (2010), έχει ως βασικό στόχο να παράσχει στους χρήστες, ιδίως τους νέους, ένα απλό εργαλείο το οποίο θα τους επιτρέψει να πειραματιστούν με το λογισμικό της έξυπνης κινητής συσκευής τους, με παρόμοιο τρόπο που οι άνθρωποι τόσες δεκαετίες κάνουν με τους προσωπικούς τους υπολογιστές. Επίσης, θεωρεί ότι με την πάροδο των χρόνων τα απλοποιημένα εργαλεία προγραμματισμού όπως η Basic, η Logo και το Scratch έχουν ανοίξει την πόρτα σε καινοτομίες όλων των ειδών.

Όπως επισημαίνει ο Kirkpatrick (2010), στην ενθουσιώδη παρατήρηση αν το App Inventor θα συμβάλει στην ανάπτυξη εφαρμογών για έξυπνες κινητές συσκευές όσο συνέβαλε το λογισμικό

Quark στις εκδόσεις ή το Blogger στην ανάπτυξη των ιστολογίων η απάντηση από ειδικούς των Media είναι σαφέστατα θετική.

8. Αναπαραστάσεις των μαθητών/πρόβλεψη δυσκολιών στο διδακτικό σενάριο

Οι έρευνες που αφορούν στη διδακτική προγραμματιστικών δομών μέσω της χρήσης υπολογιστικών περιβαλλόντων εστιάζουν κυρίως: στο είδος των δομών που διδάσκονται, το υπολογιστικό περιβάλλον και τη γλώσσα προγραμματισμού που χρησιμοποιείται και την διδακτική προσέγγιση που ακολουθείται (Κόμης, 2001; DuBoulay, 1989, Brusilovsky et al., 1997, Γρηγοριάδου, κ.α., 2002). Η δομή επανάληψης παρουσιάζει αρκετές δυσκολίες στην οικοδόμησή της (Κόμης, 2005). Οι μαθησιακές δυσκολίες των μαθητών βασίζονται κυρίως στην κατανόηση και λειτουργική εφαρμογή της εκτέλεσης της εντολής καθώς και στον προσδιορισμό της τιμής της έκφρασης (αν και τότε η συνθήκη είναι αληθής ή ψευδής) (Τζιμογιάννης και Κόμης, 1999; DuBoulay, 1989). Επιπλέον, οι μαθητές αντιμετωπίζουν δυσκολίες με τη σύνταξη των δομών ελέγχου ή την κατανόηση πολύπλοκων εκφράσεων μέσα στη συνθήκη, όταν για παράδειγμα περιέχουν πολλούς λογικούς τελεστές (Κόμης, 2005).

Επιγραμματικά τα προβλήματα συνοψίζονται σε:

- Αναπαράσταση ακολουθιακής εκτέλεσης προγράμματος
- Άρση του ισομορφισμού μεταξύ του προγράμματος (ως κειμένου) και της εκτέλεσής του
- Λογικό περιεχόμενο και προσδιορισμός στοιχειωδών δράσεων που πρέπει να επαναληφθούν
- Φύση & προσδιορισμός συνθήκης τερματισμού επανάληψης
- Προσδιορισμός αρχικής κατάστασης μεταβλητών που εμπλέκονται στην επανάληψη
- Πρότερες λογικομαθηματικές γνώσεις (μαθηματική επαγωγή)
- Ο καθορισμός της συνθήκης τερματισμού

Σε κάθε περίπτωση ο εκπαιδευτικός οφείλει να λάβει υπόψη του τις πρότερες «γνώσεις» (ιδέες, αντιλήψεις, αναπαραστάσεις) των μαθητών έχοντας υπόψη του ότι οι "προεπιστημονικές" πρότερες γνώσεις των μαθητών δεν εξαλείφονται εύκολα αλλά συνιστούν σημαντικά **γνωστικά εμπόδια** στην οικοδόμηση νέων γνώσεων. Ωστόσο, φαίνεται πως με τη χρήση διερευνητικών τρόπων διδασκαλίας και προγραμματιστικών περιβαλλόντων που επιτρέπουν την αυτενέργεια των μαθητών οι δυσκολίες αυτές ξεπερνιούνται (Αλεξοπούλου & Κυνηγός, 2008; Γλέζου κ.α., 2005; Ελευθεριώτη, κ.α., 2010).

9. Διδακτικό συμβόλαιο –Διδακτική μετατόπιση –Θεωρητικά θέματα –Διδακτικός θόρυβος

Εφόσον πριν την εκτέλεση του σεναρίου υπάρχουν οι ελάχιστες τεχνικές προδιαγραφές (σύνδεση στο διαδίκτυο και εγκατεστημένη java) και ο εκπαιδευτικός έχει φροντίσει να έχει τοποθετήσει τοπικά στους ΗΥ των μαθητών ή σε ένα δημόσιο αποθετήριο τις διάφορες εκδόσεις των αλγορίθμων - προγραμμάτων αναμένεται ότι δεν θα υπάρξουν ιδιαίτερα προβλήματα κατά την εκτέλεση του σεναρίου. Επομένως δεν θεωρούμε ότι δεν θα εμφανιστεί διδακτικός θόρυβος. Επίσης λόγω του γεγονότος ότι τα φύλλα εργασίας έχουν δημιουργηθεί με τέτοιο τρόπο ώστε να είναι απλά, ρεαλιστικά και να οδηγούν το μαθητή βήμα - βήμα στην ομαλή εξοικείωση του με την εφαρμογή θεωρούμε ότι το διδακτικό συμβόλαιο δεν θα ανατραπεί.

10. Χρήση εξωτερικών πηγών

Πηγές εκμάθησης

- [Online εκτέλεση του AppInventor](#)
- [Οδηγοί \(tutorials\) για το AppInventor](#)
- [Δωρεάν βιβλίο για το AppInventor](#)

Ιστολόγια και αναφορές στο Διαδίκτυο

Abelson, H. (2009). App Inventor for Android. Retrieved 10 January 2013 from <http://googleresearch.blogspot.gr/2009/07/app-inventor-for-android.html>

App Inventor Learning Portal (2012). Retrieved 10 January 2013 from <http://appinventor.mit.edu/explore/content/what-app-inventor.html>

Bloomberg Businessweek (2012). Can Android Market Catch Up to the App Store? Retrieved 10 January 2013 from <http://www.businessweek.com/stories/2009-10-26/can-android-market-catch-up-to-the-app-store>

King, R. (2012). Turning Girls into Tech Entrepreneurs with a Single App. Bloomberg Business Week 2012. Retrieved 28 February 2013 from <http://www.businessweek.com/technology/turning-girls-into-tech-entrepreneurs-with-a-single-app-02062012.html>

Kirkpatrick, M. (2010). 5 Big Questions About Google's New App Inventor. Retrieved 10 January 2013 from http://www.readwriteweb.com/archives/5_big_questions_about_googles_app_inventor.php

Lohr, S. (2010). Google's Do-It-Yourself App Creation Software. Retrieved 10 January 2013 from http://www.nytimes.com/2010/07/12/technology/12google.html?_r=2&partner=rss&emc=rss

MIT Center for Mobile Learning, (2011). Retrieved 10 December 2012 from <http://mitmobilelearning.org/welcome>

StarLogo TNG (2012). MIT Scheller Teacher Education Program. Retrieved 28 February 2013 from <http://education.mit.edu/projects/starlogo-tng>.

Open Blocks (2012). MIT Scheller Teacher Education Program. Retrieved 28 February 2013 from
Page, L. (2012). Update from the CEO. Retrieved 10 December 2012 from
<http://investor.google.com/corporate/2012/ceo-letter.html>

Αρθρογραφία για το AppInventor και τον προγραμματισμό

- Brennan, K. (2009). Scratch-Ed: an online community for scratch educators. In A. Dimitracopoulou, C. O'Malley, D. Suthers & P. Reimann (Eds.). *Proceedings of the 9th International Conference on Computer supported collaborative learning (CSCL'09), (Vol. 2) International Society of the Learning Sciences*, (pp. 76-78).
- Dehnadi, S., Bornat, R., & Adams, R. (2009). Meta-analysis of the effect of consistency on success in early learning of programming. 21st Annual Workshop of the Psychology of Programming Interest Group (p. 10pp).
- Hsu, Y.-C., Rice, K., & Dawley, L. (2012). Empowering educators with Google's Android App Inventor: An online workshop in mobile app design. *British Journal of Educational Technology*, 43(1) E1-E5.
- Morreli, R., de Lanerolle, T., Lake, P., Limardo, N., Tamotsu, E., & Uche, C. (2011). Can Android App Inventor bring Computational Thinking to K-12? *Proceedings of the 34th SIGCSE Technical Symposium on Computer science education*, March 9-12, 2011, USA: Dallas-Texas.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Commun. ACM* 52, 11 November 2009, 60-67.
- Γρηγοριάδου, Μ. και συνεργάτες, Διδακτικές Προσεγγίσεις και Εργαλεία για τη διδασκαλία της Πληροφορικής, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 2009.
- Δαγδιλέλης, Β. Παυλοπούλου, Κ., Τρίγγα, Π., Διδακτική, Μέθοδοι και Εφαρμογές, Εκδόσεις Μπένου 1998.
- Κόμης, Β., Εισαγωγή στη Διδακτική της Πληροφορικής, Εκδόσεις Κλειδάριθμος, Αθήνα, 2005.
- Κόμης, Β., Εισαγωγή στις Εκπαιδευτικές Εφαρμογές των ΤΠΕ, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 2004.
- Μάργαρης, Α., & Παπαστεργίου, Μ. (2008). Εισάγοντας αρχάριους στον προγραμματισμό με τα περιβάλλοντα Kara: Μια προσέγγιση βασισμένη στη θεωρία υπολογισμού. Στο Β. Κόμης (επιμ) *Πρακτικά 4^{ου} Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Διδακτική της Πληροφορικής»*, (σσ. 81-90), Πάτρα, Μάρτιος 2008 Αθήνα: Εκδόσεις Νέων Τεχνολογιών.
- Ουασίτσα, Ε., & Σάμψων, Δ. (2012). Αξιοποίηση Εργαλείων για τη δημιουργία Ηλεκτρονικών Μαθημάτων κατάλληλων για Κινητές Συσκευές. Στο Χ. Καραγιαννίδης, Π. Πολίτης & Η. Καρασαββίδης (επιμ.) *Πρακτικά του 8ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Τεχνολογίες Πληροφορίας & Επικοινωνίας στην Εκπαίδευση»*, Πανεπιστήμιο Θεσσαλίας, Βόλος, 28-30 Σεπτεμβρίου 2012.

Ελληνική ακαδημαϊκή βιβλιογραφία

Στην Ελλάδα δεν υπάρχει σχεδόν καμία ακαδημαϊκή αναφορά για το AppInventor και την εκπαιδευτική αξιοποίηση του, εκτός της εργασίας των Παπαδάκη Σταμάτιου, Καλογιαννάκη Μιχαήλ και Ζαράνη Νικόλαου η οποία παρουσιάστηκε στο 7ο Πανελλήνιο Συνέδριο των Καθηγητών Πληροφορικής στην Θεσσαλονίκη τον Απρίλιο του 2013. Τίτλος εργασίας 'Δημιουργώντας εφαρμογές για έξυπνες φορητές συσκευές με το AppInventor'

11. Υποκείμενη θεωρία μάθησης

Το περιβάλλον App Inventor όπως το SCRATCH, το BYOB κ.α. έχει χαρακτήρα παιγνιώδη. Επίσης αναφέρεται σε ένα τομέα της καθημερινής ζωής (κινητή τηλεφώνια, έξυπνες φορητές συσκευές) ο οποίος σύμφωνα με έρευνες έχει κυριαρχήσει στη ζωή των μαθητών ανεξαρτήτως ηλικίας. Όπως χαρακτηριστικά αναφέρουν οι Johnson et al., (2010, 2011) οι φορητές συσκευές με τη μορφή των έξυπνων κινητών τηλεφώνων και των ταμπλετών αποκτούν όλο και μεγαλύτερη δημοτικότητα καθώς λόγω της ισχυρής επεξεργαστικής ισχύς τους σε συνδυασμό με τη δυνατότητα συνδεσιμότητά τους με το διαδίκτυο και τη διαθεσιμότητα διαφόρων τύπων, πολυπληθών και εύκολων στη χρήση εφαρμογών για φορητές συσκευές ("mobile apps") έχουν διεισδύσει στις περισσότερες καθημερινές ανθρώπινες δραστηριότητες. Τα έξυπνα κινητά τηλέφωνα και οι ταμπλέτες θεωρούνται ως μία από τις έξι νέες τεχνολογίες που μπορούν να έχουν σημαντικό αντίκτυπο στη διδασκαλία, τη μάθηση και την έρευνα στην εκπαίδευση (Johnson et al., 2011).

Με βάση όσα προαναφέραμε αναμένεται λοιπόν να υπάρξει ζωντανή συμμετοχή των μαθητών σε όλες τις δραστηριότητες. Επιπλέον, είναι ενδεχόμενο, πολύ γρήγορα οι μαθητές να θελήσουν να δοκιμάσουν τροποποιήσεις του περιβάλλοντος και των λοιπών στοιχείων των (μικρών αρχικά) προγραμμάτων, επιθυμώντας να τροποποιήσουν τα χαρακτηριστικά του σύμφωνα με τις επιθυμίες τους. Στο παρόν διδακτικό σενάριο λοιπόν, θα πραγματοποιηθεί πειραματισμός με τα στοιχεία του προγραμματιστικού περιβάλλοντος με ενεργητική συμμετοχή. Άλλωστε όπως αναφέρει η Γλέζου (2002) *"οι ΤΠΕ και ιδιαίτερα τα Logo-like περιβάλλοντα μπορούν να αξιοποιηθούν για τη σχεδίαση και την ανάπτυξη υπολογιστικών εργαλείων, τα οποία προσφέρουν στους μαθητές τη δυνατότητα έκφρασης κι αξιοποίησης των σκέψεων, ιδεών και διαισθησέων τους και υποστηρίζουν τη διαδικασία οικοδόμησης της γνώσης διαμορφώνοντας πλούσια σε ευκαιρίες προβληματισμού και πειραματισμού περιβάλλοντα μάθησης"*.

Το στοιχείο αυτό, σε συνδυασμό με την οργάνωση της τάξης σε μικρές ομάδες, προσφέρει ένα πολύ καλό περιβάλλον για την ανάπτυξη μιας ισχυρής αλληλεπίδρασης που μπορεί να ευνοήσει τη μάθηση. Ευνοείται λοιπόν ιδιαίτερα η δημιουργία ενός τυπικού περιβάλλοντος κοινωνιο-κονστрукτιβιστικού καθώς θεωρούμε ότι μέσω των κοινωνικών αλληλεπιδράσεων επηρεάζεται η διαδικασία με την οποία οικοδομείται η γνώση. Το σενάριο είναι θεμελιωμένο στην θεωρία μάθησης του εποικοδομητισμού διότι ο μαθητής χτίζει την γνώση του

ανιχνεύοντας, διερευνώντας και αλληλεπιδρώντας οπτικά με τις βασικές εντολές της γλώσσας προγραμματισμού του AppInventor δημιουργώντας απλά προγράμματα.

Όπως χαρακτηριστικά αναφέρουν οι Kafai & Resnick (1996), *"Ο Κονστρουκτιονισμός προτείνει ότι οι μαθητευόμενοι κατασκευάζουν νέες ιδέες όταν ενεργά ασχολούνται με τη δημιουργία εξωτερικής κατασκευής -μπορεί ένα ρομπότ, ένα ποίημα, ένα κάστρο στην άμμο, ένα πρόγραμμα στον υπολογιστή- πάνω στις οποίες αναστοχάζονται και μοιράζονται με άλλους. Έτσι ο κονστρουκτιονισμός εμπλέκει δυο διαπλεκόμενους τύπους κατασκευής: την οικοδόμηση της γνώσης στο πλαίσιο οικοδόμησης κατασκευών με προσωπικό νόημα"*.

12. Επισήμανση μικρομεταβολών

Μοναδικό πρόβλημα που ενδεχόμενα μπορεί να παρουσιαστεί είναι εφόσον τα προγράμματα που θα κατασκευαστούν είναι αρκετά μεγάλα σε μέγεθος ή εκτέλεση τους στον προσομοιωτή είναι αρκετά αργή (εφόσον χρησιμοποιηθεί προσομοιωτής και όχι οι κινητές συσκευές των μαθητών). Ωστόσο επειδή οι δραστηριότητες που θα εκτελέσουν οι μαθητές θα είναι μικρές σε μέγεθος και κατάλληλα στοχευμένες δε αναμένεται να υπάρξουν ιδιαίτερα προβλήματα. Εξάλλου το παιγνιώδες ύφος της εφαρμογής και η δυνατότητα προγραμματισμού με πλακίδια (blocks), μειώνει το νοητικό φορτίο που απαιτείται για την συγγραφή κώδικα και ταυτόχρονα βοηθάει ιδίως τους αρχάριους στον προγραμματισμό χρήστες να επικεντρωθούν στην επίλυση ενός προβλήματος δίχως να είναι αναγκασμένοι να δώσουν βαρύτητα στο αυστηρό τυπικό και συντακτικό μιας κανονικής γλώσσας προγραμματισμού.

13. Οργάνωση της τάξης – Εφικτότητα σχεδίασης

Η διδασκαλία θα λάβει χώρα εντός του σχολικού εργαστηρίου πληροφορικής. Οι μαθητές θα εργαστούν ατομικά είτε σε ομάδες των δύο ή τριών ατόμων ανά ηλεκτρονικό υπολογιστή. Ενδεχόμενα ο εκπαιδευτικός μπορεί να επιτρέψει σε όσους μαθητές το επιθυμούν και διαθέτουν, να φέρουν και να συνδέσουν με τον υπολογιστή τους το κινητό τους τηλέφωνο με λειτουργικό Android, για μεγαλύτερη ταχύτητα και αληθοφάνεια στην εκτέλεση των προγραμμάτων σε σχέση με τη χρήση του προσομοιωτή.

14. Επεκτάσεις / διασυνδέσεις των εννοιών ή των δραστηριοτήτων

Μετά την ολοκλήρωση του παρόντος σεναρίου οι μαθητές θα είναι προετοιμασμένοι για να εργαστούν με επόμενα σενάρια τα οποία θα περιλαμβάνουν τη χρήση πιο σύνθετων αλγοριθμικών δομών όπως εμφωλευμένων δομών με χρήση ακολουθίας επιλογής και επανάληψης αλλά και χειρισμό στοιχείων αντικειμενοστραφούς προγραμματισμού προκειμένου οι εφαρμογές που θα δημιουργήσουν οι μαθητές να είναι πιο 'απαιτητικές' αλλά

και πιο διαδραστικές.

15. Περιγραφή και ανάλυση φύλλων εργασίας

Τα φύλλα εργασίας αφορούν την εκμάθηση την εκμάθηση της δομής επανάληψης ΟΣΟ μέσα από το περιβάλλον του AppInventor..

16. Αξιολόγηση

Ο εκπαιδευτικός μπορεί να χρησιμοποιήσει τα συνηθισμένα και αξιόπιστα open source λογισμικά όπως π.χ. το κλασσικό hot potatoes, το ελληνικό και δωρεάν hot pepper ή τον αξιολογητή προκειμένου να δημιουργήσει τεστ αυτοαξιολόγησης (κλειστού τύπου), σταυρόλεξα, κρυπτόλεξα κλπ. αλλά και κουίζ, flashcards κ.α.

17. Το επιμορφωτικό σενάριο

Θα πραγματοποιηθεί γνωριμία των μαθητών με την δομή επανάληψης ΟΣΟ στο περιβάλλον του AppInventor και θα γίνει προσπάθεια δημιουργίας εφαρμογών για έξυπνες φορητές συσκευές με λειτουργικό σύστημα Android από τους μαθητές. Θα αντιμετωπιστούν ενδεχόμενα προβλήματα που οφείλονται σε λανθασμένες αντιλήψεις και προϋπάρχουσες γνώσεις των μαθητών όπως παραλληλισμός, ανθρωπομορφισμός κ.α. Τα φύλλα εργασίας που θα δοθούν στους μαθητές παρουσιάζονται στις επόμενες ενότητες.

18. Φύλλο εργασίας 1 –

Δραστηριότητα 1

Άσκηση

Γνωρίζετε ήδη ότι αρκετές φορές στην καθημερινή σας ζωή προκειμένου για να λύσετε ένα πρόβλημα πρέπει να επαναλάβετε κάποιες ενέργειες επαναληπτικά. Για παράδειγμα στα μαθηματικά προκειμένου να υπολογίσουμε τον μέσο όρο της βαθμολογίας 30 μαθητών χρειάζεται να διαβάσουμε την βαθμολογία 30 μαθητών ή σε επίπεδο αλγορίθμου να χρησιμοποιήσουμε ακολουθιακά 30 φορές μια εντολή της μορφής 'διάβασε βαθμολογία'.

Συμπληρώστε και εσείς 3 παραδείγματα από την καθημερινή ή την σχολική σας ζωή, στα οποία χρησιμοποιείτε δομή επανάληψης.

1.
2.
3.

Μπορείτε να σκεφτείτε 3 παραδείγματα και να γράψετε τις εντολές όπως θα τις γράφατε στον υπολογιστή στα υποθετικά προγράμματα σας; Ας θυμηθούμε ότι οι τελεστές ανισότητας που χρησιμοποιούμε είναι οι $>$, $<$, $>=$, $<=$, $=$, $<>$ ενώ οι λογικοί τελεστές είναι οι: ΚΑΙ, Η, ΟΧΙ.

Σημειώστε στα παραδείγματα σας, ποια είναι η συνθήκη που εφόσον είναι αληθής θα εκτελεστεί η εντολή ή η ομάδα εντολών και ποια/ποιες είναι αυτές.

Παράδειγμα 1:

Συνθήκη:.....

Εντολή /Ομάδα εντολών:

Παράδειγμα 2:

Συνθήκη:.....

Εντολή /Ομάδα εντολών:

Παράδειγμα 3:

Συνθήκη:.....

Εντολή /Ομάδα εντολών:

Καταγράψτε τυχόν απορίες και προβληματισμούς σας και συζητείστε τους εντός του σχολικού εργαστηρίου με τους συμμαθητές σας και τον καθηγητή σας.

.....
.....
.....

Δραστηριότητα 2

Οδηγίες υλοποίησης

1. Στην πρόταση που ακολουθεί συμπληρώστε τα κενά που λείπουν. Μπορείτε να χρησιμοποιήσετε τις παρακάτω λέξεις: **επανάληψη, λογική, αρχή, συνθήκη, οποιοδήποτε**

Με την δομή επανάληψης ΟΣΟ, η ελέγχεται από μία έκφραση στην και εκτελείται όσο η αυτή είναι αληθής. Αντιμετωπίζει πρόβλημα απαιτεί επαναληπτική δομή για την αντιμετώπισή του.

2. Στην πρόταση που ακολουθεί συμπληρώστε τα κενά που λείπουν. Μπορείτε να χρησιμοποιήσετε τις παρακάτω λέξεις: **ψευδής, συνθήκη, ποτέ, αρχή, επανάληψη**

Το γεγονός ότι η ελέγχεται στην σημαίνει ότι οι εντολές μέσα στην μπορεί να μην εκτελεστούν (αν η αρχική τιμή της συνθήκης είναι).

3. Στην πρόταση που ακολουθεί συμπληρώστε τα κενά που λείπουν. Μπορείτε να χρησιμοποιήσετε τις παρακάτω λέξεις: **εκτελούνται, συνέχεια συνθήκη**

Σε περίπτωση που η είναι αληθής, οι εντολές επ' άπειρο.

4. Παρακάτω σας δίδεται η γενική μορφή σύνταξης της δομής επανάληψης ΟΣΟ. Μπορείτε να συμπληρώσετε τα κενά που λείπουν;

ΟΣΟ ΕΠΑΝΑΛΑΒΕ

.....

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Καταγράψτε τυχόν απορίες και προβληματισμούς σας και συζητείστε τους εντός του σχολικού εργαστηρίου με τους συμμαθητές σας και τον καθηγητή σας.

.....
.....
.....

Δραστηριότητα 3

Οδηγίες υλοποίησης

Ας προσπαθήσουμε να υλοποιήσουμε ένα αλγόριθμο. Μιας και είσαστε μαθητές ας προσπαθήσουμε να υλοποιήσουμε τον αλγόριθμο της προπαίδειας ενός αριθμού π.χ. το 5. Αφού θυμηθείτε την προπαίδεια του 5 όπως παρουσιάζεται παρακάτω απαντήστε στις παρακάτω ερωτήσεις:

$$1 \times 5 = 5$$

$$2 \times 5 = 10$$

$$3 \times 5 = 15$$

.....

$$9 \times 5 = 45$$

$$10 \times 5 = 50$$

Η προπαίδεια του 5 μπορεί να υλοποιηθεί με χρήση της δομής επανάληψης ΟΣΟ. Αν κάποιος λόγο, θεωρείτε το αντίθετο συζητείστε το εντός της τάξης με τους συμμαθητές και τον καθηγητή σας.

Στην συνέχεια απαντήστε στις ακόλουθες ερωτήσεις:

1. αριθμός επαναλήψεων;
2. Ποιός είναι ο μετρητής και ποιά η αρχική του τιμή;
3. Ποιά είναι η συνθήκη ;
4. Ποιό είναι το βήμα άθροισης του μετρητή;.....
5. Ποιά είναι τα σταθερά μέρη που εμφανίζονται σε κάθε επανάληψη;
6. Ποιά τιμή επηρεάζεται από την μεταβολή της τιμής του μετρητή;

Συναντήσατε κάποια δυσκολία/δυσκολίες στην απάντηση των ερωτήσεων; Αν ναι, συζητείστε το εντός της τάξης με τους συμμαθητές και τον καθηγητή σας.

Στην συνέχεια ας προσπαθήσουμε να συμπληρώσουμε το απόσπασμα του αλγορίθμου προκειμένου να υπολογίζει την προπαίδεια του 5.

I <-

ΟΣΟ (..... <=) ΕΠΑΝΑΛΑΒΕ

ΓΡΑΨΕ, 'X 5 = ', I*.....

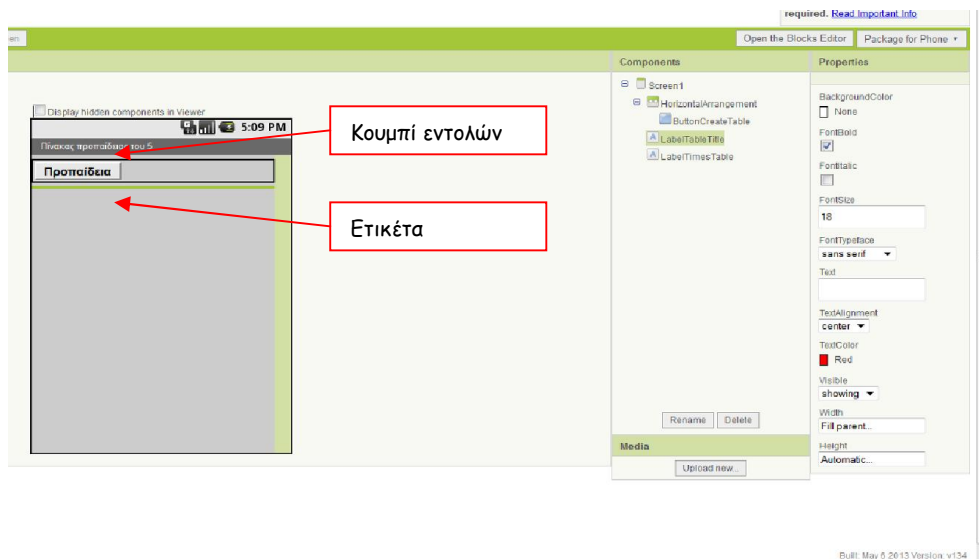
I <- I+.....

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Δραστηριότητα 4

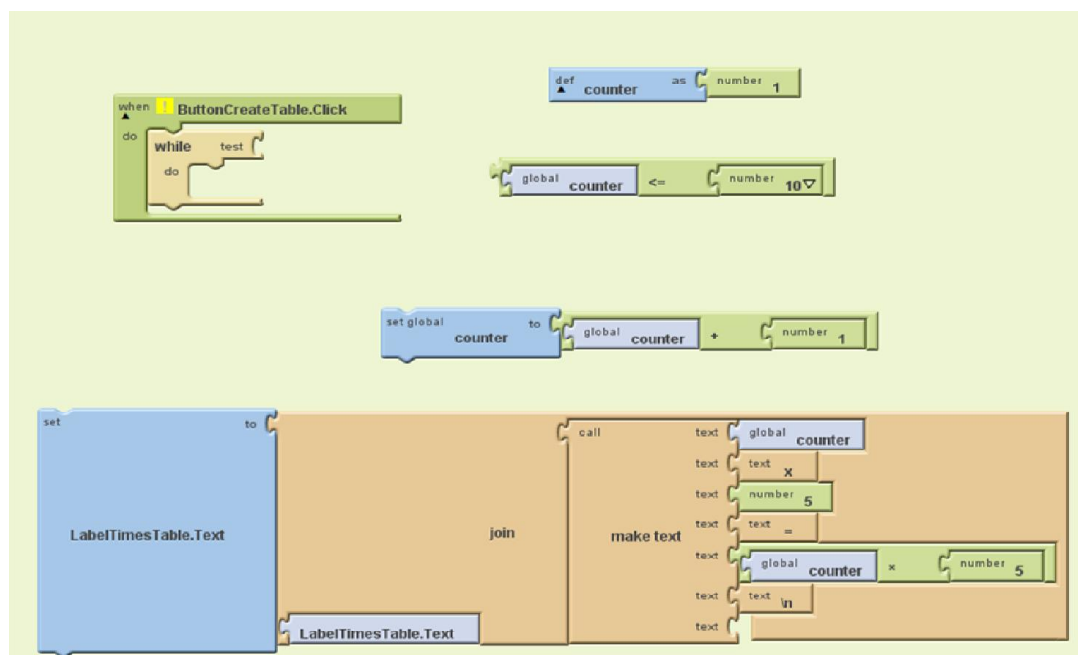
Οδηγίες υλοποίησης

Στη συνέχεια θα προσπαθήσουμε να υλοποιήσουμε τον παραπάνω αλγόριθμο στο περιβάλλον του AppInventor. Γνωρίζετε ήδη τα βήματα για την δημιουργία μιας εφαρμογής στο AppInventor, οπότε προσπαθήστε να δημιουργήσετε σχεδιαστικά μια εφαρμογή η οποία να μοιάζει με την ακόλουθη:



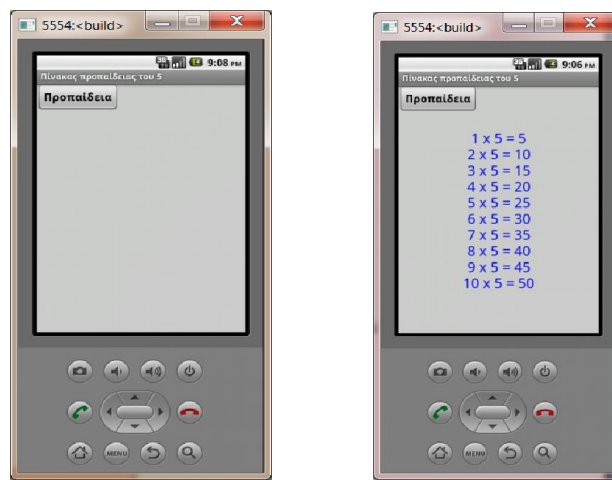
Αν σε περίπτωση που δυσκολεύεστε να υλοποιήσετε το οπτικό κομμάτι της εφαρμογής, ζητήστε από τον καθηγητή σας, να σας δώσει το project (έργο) προκειμένου να ασχοληθείτε μόνο με την υλοποίηση της δομής επανάληψης στον Block Editor.

Τα πλακίδια που θα χρειαστείτε να συνδέσετε είναι τα :



Ιδίως το τελευταίο (μεγάλο) πλακίδιο που αποτελεί την ένωση αρκετών μικρών πλακιδίων δεν χρειάζεται να σας προβληματίσει. Χρησιμοποιείται απλά για να δημιουργηθεί ένα μεγάλο μήνυμα της μορφής **νούμερο X 5 = νούμερο*5**, τοποθετεί το αποτέλεσμα σε μια ετικέτα και στο τέλος εισάγει μια κενή γραμμή (σαν να πατάτε δηλαδή το enter) απλά για να μην 'κολλήσουν' τα αποτελέσματα μεταξύ σε μια μόνο γραμμή.

Στις επόμενες 2 εικόνες, βλέπετε την εφαρμογή που θα εμφανιστεί στον προσομοιωτή όταν εκκινεί και το αποτέλεσμα που θα εμφανιστεί όταν πατήσετε το κουμπί με τίτλο Προπαίδεια.



Δραστηριότητα 5

Οδηγίες υλοποίησης

Χρησιμοποιώντας ως βάση την προηγούμενη εφαρμογή, προβείτε στις αλλαγές που σας ζητούνται κάθε φορά:

1. Τροποποιήστε τα 2 αρχικά πλακίδια στα ακόλουθα:



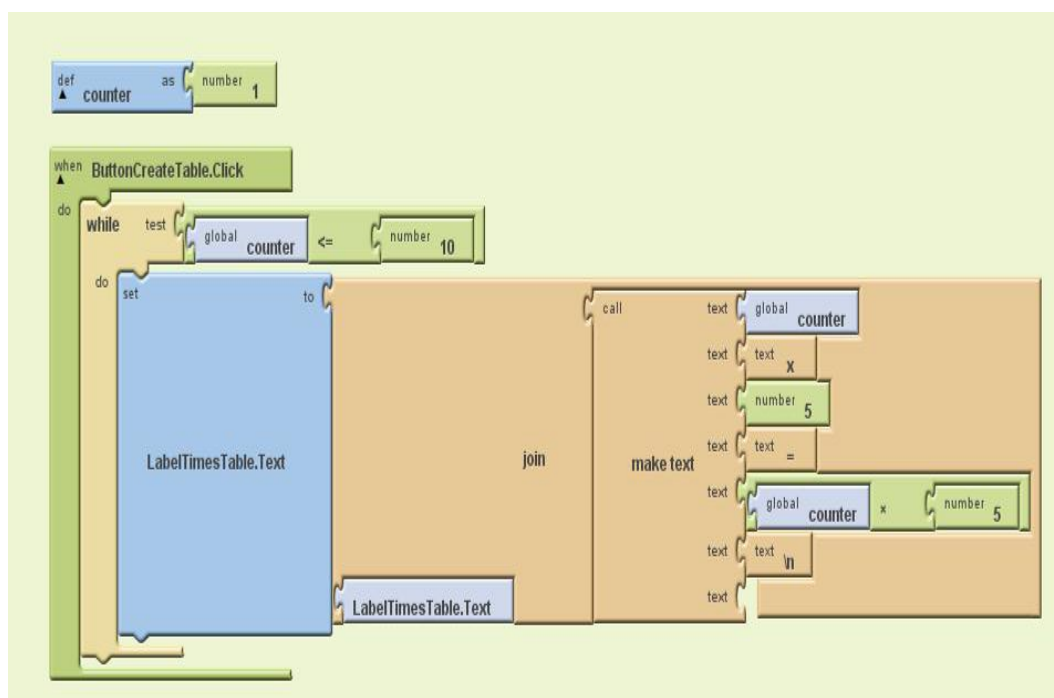
Πόσες φορές θα τρέξει ο αλγόριθμος; Παρουσιάζει την προπαίδεια του 5; Γιατί συμβαίνει αυτό;

.....

.....

.....

2. Εκτελέστε τον παρακάτω τροποποιημένο αλγόριθμο:

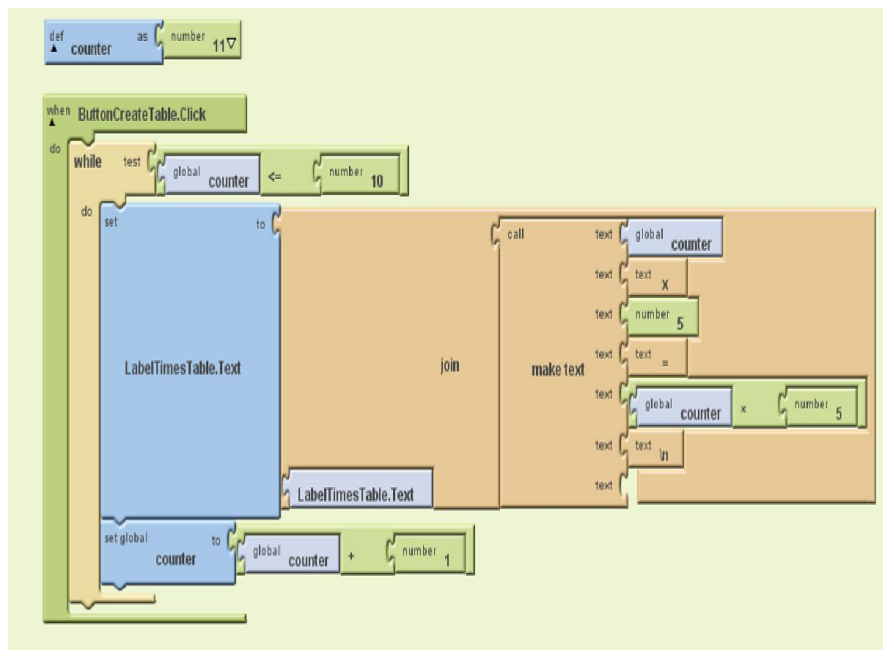


Τι παρατηρείτε; Για πιο λόγο συμβαίνει αυτό;

.....

.....

3. Εκτελέστε τον παρακάτω τροποποιημένο αλγόριθμο:

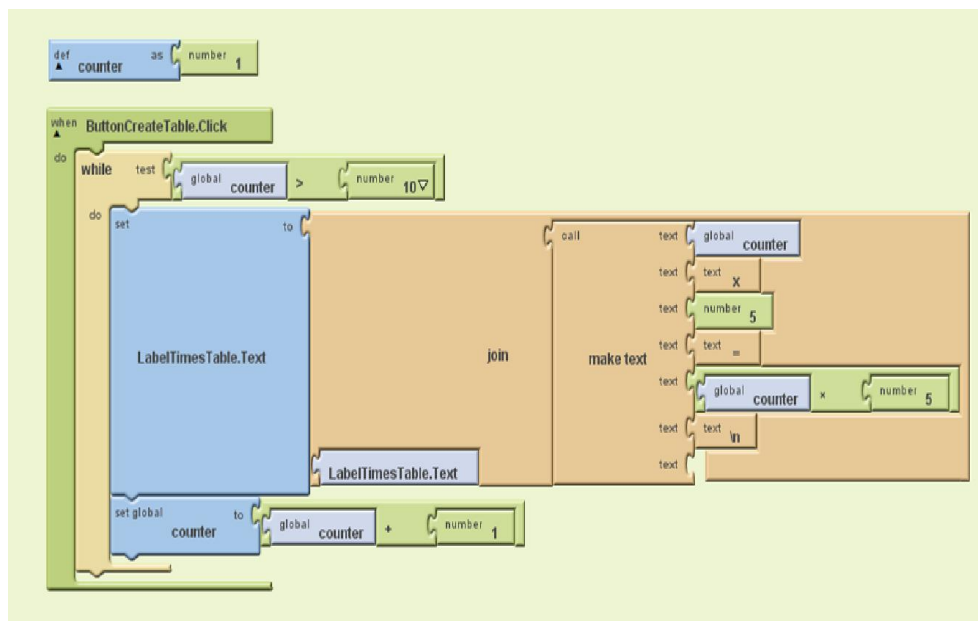


Τι παρατηρείτε; Για πιο λόγο συμβαίνει αυτό;

.....

.....

4. Εκτελέστε τον παρακάτω τροποποιημένο αλγόριθμο:



Τι παρατηρείτε; Για πιο λόγο συμβαίνει αυτό;

.....

.....

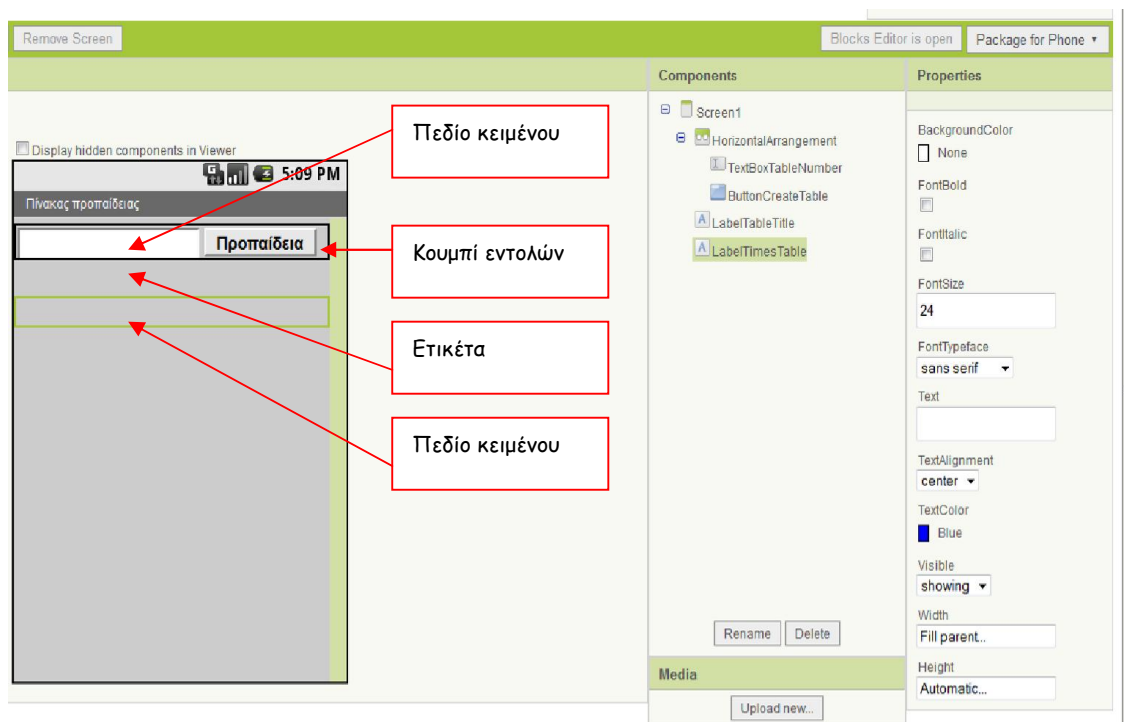
19. Φύλλο εργασίας 2

Δραστηριότητα 1

Στη συνέχεια θα προσπαθήσουμε να υλοποιήσουμε μια εφαρμογή η οποία θα εμφανίζει και πάλι την προπαίδεια ενός αριθμού, αλλά με την διαφορά ότι ο αριθμός δεν θα είναι σταθερός, αλλά κάθε φορά ο χρήστης θα πληκτρολογεί τον αριθμό που επιθυμεί.

Οδηγίες υλοποίησης:

Το οπτικό κομμάτι της εφαρμογής σας θα πρέπει να μοιάζει με την παρακάτω εικόνα:



Αν σε περίπτωση που δυσκολεύεστε να υλοποιήσετε το οπτικό κομμάτι της εφαρμογής, ζητήστε από τον καθηγητή σας, να σας δώσει το project (έργο) προκειμένου να ασχοληθείτε μόνο με την υλοποίηση της δομής επανάληψης στον Block Editor.

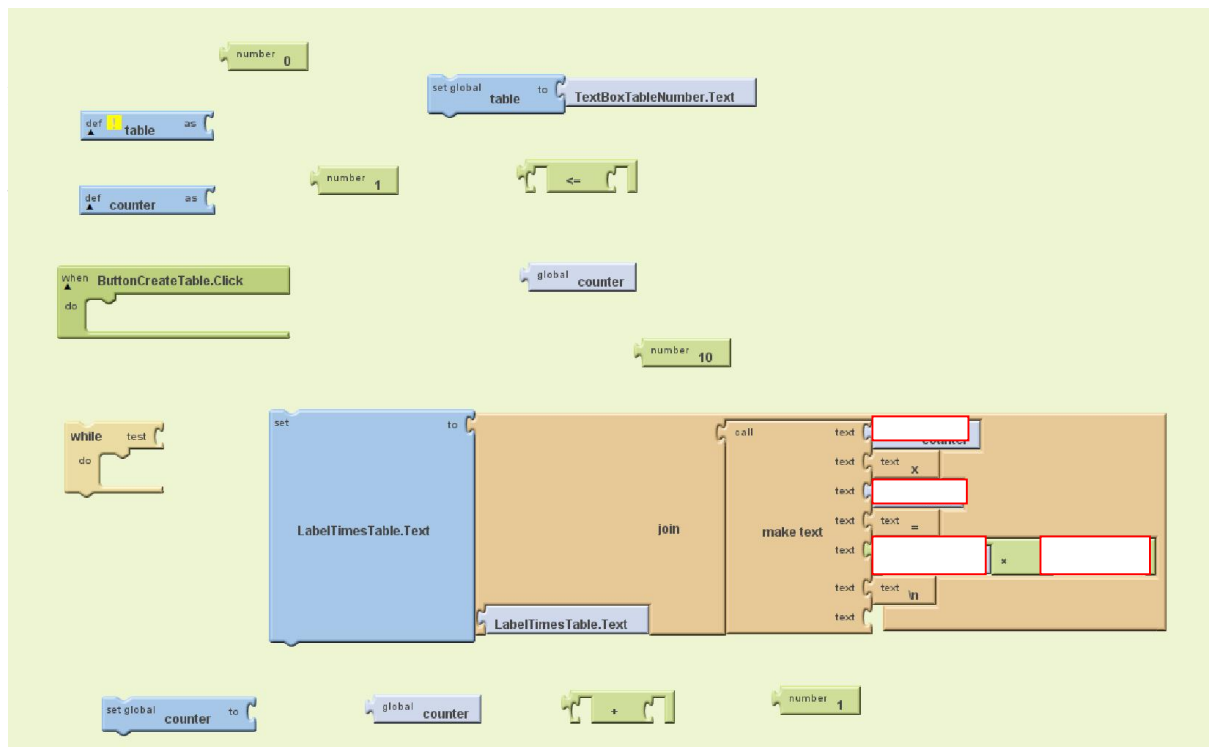
Πριν προχωρήσετε στην υλοποίηση της εφαρμογής στον block Editor, ας δούμε πως θα υλοποιήσετε την εφαρμογή. Επειδή ο αριθμός δεν είναι προκαθορισμένος (π.χ. το 5) θα χρειάζεται ο χρήστης να καταχωρεί κάθε φορά τον αριθμό που επιθυμεί. Επομένως η εφαρμογή σας θα διαβάζει έναν αριθμό [1..10] και με βάση αυτή την τιμή του αριθμού, ο οποίος στην πραγματικότητα θα είναι μια μεταβλητή (π.χ. table) θα δημιουργήσετε τον αλγόριθμο σας.

Ας απαντήσουμε επομένως στις παρακάτω ερωτήσεις:

- Ποιά είναι η αρχική τιμή του μετρητή σας:.....

- Ποιά είναι η σύνταξη της συνθήκης σας:.....
- Ποιό είναι το βήμα αύξησης του μετρητή σας:.....
- Ποιες μεταβλητές θα χρησιμοποιήσετε προκειμένου να επιτύχετε την εμφάνιση της προπαίδειας, δηλ. π.χ. το $2 \times 10 = 20$

Στην εικόνα που ακολουθεί παρουσιάζονται τα πλακίδια μιας υποθετικής λύσης της παραπάνω άσκησης.



Στις υπόλοιπες 3 εικόνες παρουσιάζονται τα αποτελέσματα της εφαρμογής σας στον προσομοιωτή.

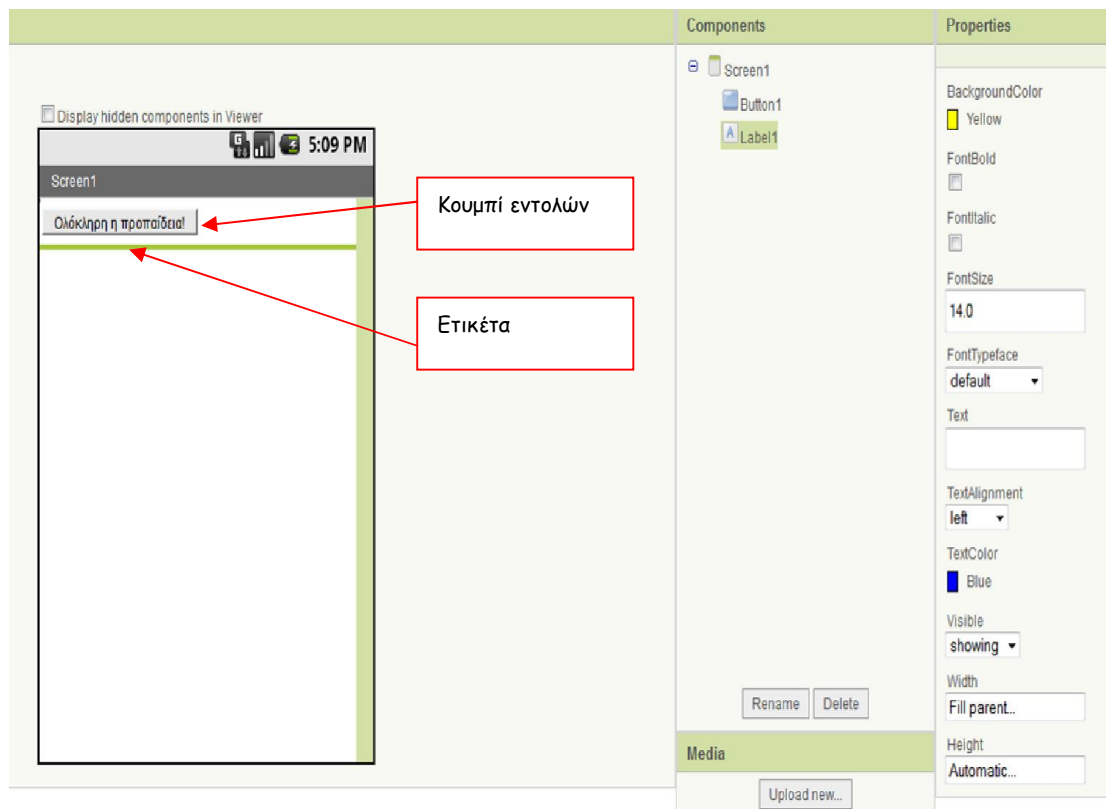


Συναντήσατε κάποια δυσκολία/δυσκολίες στην απάντηση των ερωτήσεων; Αν ναι, συζητείστε το εντός της τάξης με τους συμμαθητές και τον καθηγητή σας.

Δραστηριότητα 2

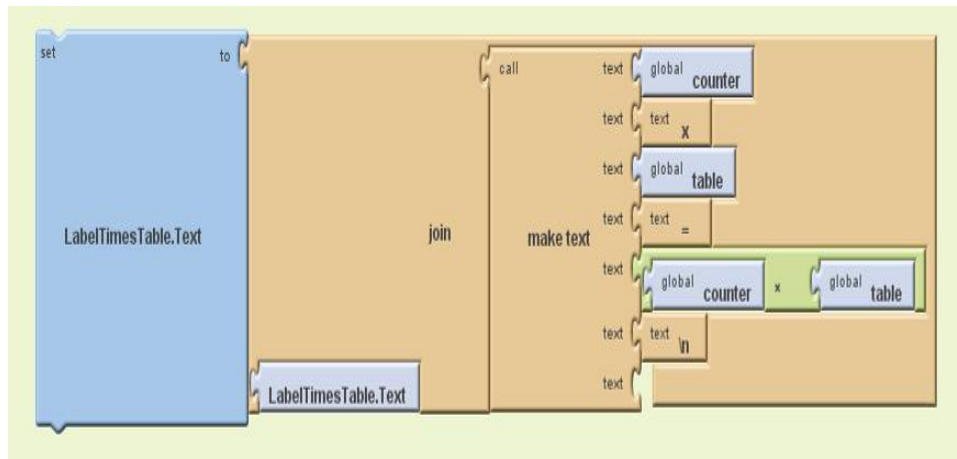
Άσκηση για περαιτέρω προβληματισμό:

Οδηγίες υλοποίησης



Υπόδειξη: για την υλοποίηση του προγραμματιστικού κομματιού της εφαρμογής σας θα χρειαστεί να χρησιμοποιήσετε 2 εμφωλευμένες δομές επανάληψης. Η εσωτερική σας θα είναι η γνωστή από τις προηγούμενες δραστηριότητες σας, η οποία θα υπολογίζει την προπαίδεια ενός αριθμού και η εξωτερική δομή θα μας δίνει κάθε φορά των αριθμό [1..10], την προπαίδεια του οποίου θέλουμε να υπολογίζουμε και να εμφανίζουμε.

Για την εμφάνιση του αποτελέσματος θα χρησιμοποιήσετε το γνωστό σας 'μεγάλο πλακίδιο'. Θα πρέπει να είσαστε ιδιαίτεροι προσεκτικοί στην απόδοση αρχικών τιμών στους μετρητές σας, στην σύνταξη των συνθηκών σας καθώς και στην μεταβολή των τιμών των μετρητών σας.



20. Φύλλο εργασίας 3 - Επανάληψη - Εμπέδωση

Ερωτήσεις κλειστού τύπου

1. Η δομή επανάληψης ΟΣΟ χρησιμοποιείτε όταν θέλουμε μια ομάδα εντολών να εκτελεστεί περισσότερες από μια φορά.

Σωστό

Λάθος

2. Οι εντολές που βρίσκονται στο σώμα της δομής επανάληψης είναι δυνατόν να μην εκτελεστούν καμία φορά

Σωστό

Λάθος

3. Δεν μπορούμε να έχουμε μια δομή επανάληψης μέσα σε μια άλλη δομή επανάληψης

Σωστό

Λάθος

4. Ατέρμων βρόγχος ή άπειρη ανακύκλωση έχουμε στην περίπτωση κατά την οποία η δομή επανάληψης δεν τερματίζει ποτέ.

Σωστό

Λάθος

5. Στην δομή επανάληψης ο μετρητής πρέπει να αρχικοποιείται πάντα με το νούμερο 1

Σωστό

Λάθος

21. Προτάσεις για περαιτέρω δραστηριότητες - προτεινόμενες εργασίες

Ολοκληρώστε τη δημιουργία των φύλλων εργασίας που προορίζονται για τους μαθητές, παραλλάσσοντας στοιχεία των προτεινόμενων δραστηριοτήτων ή προσθέτοντας νέες δραστηριότητες όπως π.χ. τον υπολογισμό του αριθμού των πιθανών λύσεων μιας εξίσωσης της μορφής π.χ. $7X + 25 = 100$ στο διάστημα $[-100...+100]$ ή τον υπολογισμό του αριθμού των πιθανών λύσεων μιας εξίσωσης της μορφής π.χ. $7X + 25Y = 100$ στο διάστημα $[-100...+100]$ κ.α.