



**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Διδακτική του προγραμματισμού  
Μια μελέτη περίπτωσης με χρήση του App Inventor**

**Ευστράτιος Α. Παπουτσόγλου**

**Εισηγητής: Δρ Ιωάννης Ψαρομήλιγκος**

**ΑΘΗΝΑ  
2017**

Διδακτική του προγραμματισμού – Μια μελέτη περίπτωσης με χρήση του App Inventor

Διδακτική του προγραμματισμού – Μια μελέτη περίπτωσης με χρήση του App Inventor

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Διδακτική του προγραμματισμού  
Μια μελέτη περίπτωσης με χρήση του App Inventor**

**Ευστράτιος Α. Παπουτσόγλου  
Α.Μ. 38100**

**Εισηγητής:**

**Δρ Ιωάννης Ψαρομήλιγκος, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης**



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, το App Inventor. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, Ιωάννης Ψαρομήλιγκος, τον οποίο θα ήθελα να ευχαριστήσω.



## ΠΕΡΙΛΗΨΗ

Στην εποχή μας η παρουσία της τεχνολογίας, και ιδιαίτερα με τη μορφή των έξυπνων συσκευών, σε κάθε πτυχή των ανθρωπίνων δραστηριοτήτων θεωρείται πλέον κάτι περισσότερο ίσως από δεδομένη. Ο άνθρωπος, το πιο προσαρμοστικό ον του πλανήτη μας, και πιο ειδικά οι μαθητές κάθε ηλικίας για να μπορέσουν να ανταπεξέλθουν σε αυτές τις ανάγκες της εποχής θα πρέπει να έχουν στη διάθεση τους εκπαίδευση ικανή να τους οπλίσει με κάθε απαραίτητη, προς αυτή την κατεύθυνση, δεξιότητα. Η ανάγκη αυτή του εκσυγχρονισμού της εκπαίδευσης έχει επισημανθεί αρκετά χρονιά τώρα και μέσα από πολλές έρευνες και συνέδρια έχει σημειωθεί πρόοδος της «διδακτικής του προγραμματισμού». Τα περιθώρια βελτίωσης ωστόσο είναι ανάλογα και της ανάπτυξης της τεχνολογίας, με αποτέλεσμα να προβάλλει επιτακτική η ανάγκη να διευρυνθούν ακόμη περισσότερο οι προσεγγίσεις από τους εκπαιδευτές της πληροφορικής, έτσι ώστε η διδακτική του προγραμματισμού να ενταχθεί σε όλες τις βαθμίδες της εκπαίδευσης και κατάρτισης, αλλά και να καλύψει το ευρύ φάσμα των αναγκών των εκπαιδευομένων όπως αυτό προκύπτει από την διαφορετικότητα αυτών ανάλογα με την ηλικία, το αντικείμενο και το επίπεδο των.

Έχοντας στο στόχαστρο, λοιπόν, όχι μόνο την διδασκαλία του προγραμματισμού αυτού καθ' αυτού, αλλά την συνολική κατάρτιση του ατόμου με την δυνατότητα να μπορεί να επιλύει, να σχεδιάζει αλλά και να ανταπεξέρχεται σε κάθε μορφή προβλήματος [Seymour Papert and Alan Kay], έχουν αναπτυχθεί, από μεγάλα εκπαιδευτικά ιδρύματα (σ.σ. Μ.Ι.Τ.) και μεγαθήρια εταιρειών του χώρου (σ.σ. Google), πολλά εργαλεία έτσι ώστε η διδακτική του προγραμματισμού να γίνει πιο ελκυστική, πιο εύκολη αλλά κυρίως πιο αποτελεσματική. Παράδειγμα των τελευταίων αποτελεί και το App Inventor, ένα ελεύθερο και ανοικτό περιβάλλον οπτικού προγραμματισμού, με το οποίο χρήστες όλων των δυνατοτήτων μπορούν να δημιουργήσουν τις δικές τους εφαρμογές. Με τον τρόπο αυτό η διδακτική του προγραμματισμού τείνει να γίνεται πιο ελκυστική, και γεμάτη κίνητρα για όλους, αφού όλοι μπορούν να δημιουργήσουν εφαρμογές για τις έξυπνες συσκευές που μπορούν να χρησιμοποιούν οι ίδιοι τους αλλά και όχι μόνο. Στη δική μας μελέτη περίπτωσης στοχεύουμε, χρησιμοποιώντας το App Inventor, να καλύψουμε και το πρώτο σκέλος του προβλήματος της διδακτικής του προγραμματισμού, και να απλοποιηθούν οι βασικές έννοιες του προγραμματισμού.

## ABSTRACT

In our times, the presence of technology, and in particular in the form of smart devices, in every aspect of human activity is now considered indisputably a fact. The man, as the most adaptive being of our planet, and more specifically students of all ages, in order to be able to cope with these needs of the time, they should be able to be trained with all the necessary skills in this direction. This need for the modernization of education has been highlighted several years now, through many studies and conferences, there has been progress in its 'programming teaching'. The room for improvement is, however, analogous to the development of technology, leading to the urgent need to further expand the approaches by IT instructors so that programming curricula can be integrated into all levels of education and training, to cover the wide range of learner needs as this arises from their diversity according to their age, subject and level.

Having therefore not only the teaching of programming itself, but also the overall training of the individual with the ability to resolve, plan, and cope with any form of problem [Seymour Papert and Alan Kay] Large-scale educational institutions (MIT) and large-scale companies in the field (Google) have developed many tools so that programming teaching becomes more attractive, easier but most effective. An example of the latter is App Inventor, a free and open visual programming environment, with which users of all capabilities can create their own applications. In this way, programming teaching tends to become more attractive, and motivated for all, since everyone can create applications for smart devices that can be used by themselves but also not only. In our case study, we aim, using App Inventor, to cover the first part of the problem of its syllabus, and to simplify the basic concepts of programming.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Διδακτική Προγραμματισμού

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: εκπαιδευτικές γλώσσες, App Inventor

SCIENTIFIC AREA: Programming Didactics

KEY WORDS: Learning languages, App Inventor





## ΠΕΡΙΕΧΟΜΕΝΑ

1	Διδακτική του προγραμματισμού .....	13
1.1	Εισαγωγή .....	13
1.2	Ένταξη των Τ.Π.Ε. στην εκπαίδευση.....	15
1.2.1	Από την παραδοσιακή στη σύγχρονη διδασκαλία.....	15
1.2.2	Φάσεις και μοντέλα ένταξης των Τ.Π.Ε. στην εκπαίδευση .....	18
1.2.2.1	Χρονολογική εξέλιξη εισαγωγής των Τ.Π.Ε. στην εκπαίδευση.....	19
1.2.2.2	Μοντέλα ένταξης των ΤΠΕ στην εκπαίδευση.....	19
1.3	Θεωρίες μάθησης και Τ.Π.Ε.....	21
1.4	Ο μετασχηματισμός της διδασκαλίας με τη χρήση του Η/Υ.....	25
1.4.1	Οι εκπαιδευτικές λειτουργίες του Η/Υ .....	25
1.4.2	Πλεονεκτήματα από τη χρήση του Η/Υ.....	26
1.4.3	Προϋποθέσεις αποτελεσματικής χρήσης των νέων μέσων .....	27
1.5	Προγραμματισμένη διδασκαλία: η θεωρητική θεμελίωση των νέων μέσων.....	29
1.6	Ο προγραμματισμός ως δραστηριότητα μάθησης .....	32
1.6.1	Τα οφέλη από τη διδασκαλία του προγραμματισμού.....	32
1.6.2	Δυσκολίες των αρχάριων προγραμματιστών.....	33
1.6.3	Διδακτικές προσεγγίσεις του προγραμματισμού .....	34
2	Εκπαιδευτικά περιβάλλοντα .....	38
2.1	Εισαγωγή .....	38
2.2	Scratch .....	38
2.3	λsnaρ(παλιότερα γνωστό ως byob).....	40
2.4	Alice 3.....	40
2.5	Kodu .....	41
2.6	Stencyl.....	42
2.7	Pocket code .....	43
3.	Μελέτη περίπτωσης της διδακτικής του προγραμματισμού με χρήση του app inventor .....	44
3.1	Εισαγωγή .....	44
3.2	Τι είναι το app inventor.....	44
3.3	Γιατί χρησιμοποιούμε το app inventor .....	45
3.4	Τι εφαρμογές μπορούμε να δημιουργήσουμε με το app inventor .....	48
3.5	Σε ποιον απευθύνεται το app inventor .....	51

## Διδακτική του προγραμματισμού – Μια μελέτη περίπτωσης με χρήση του App Inventor

3.5.1	Ξενάγηση στο περιβάλλον εργασίας του app inventor.....	51
3.5.2	Αντικείμενα συμπεριφορές και γεγονότα .....	54
3.5.3	Σχεδιάζοντας τα αντικείμενα .....	55
3.5.4	Δίνοντας συμπεριφορές στα αντικείμενα .....	56
3.5.5	Οι εντολές στο app inventor.....	57
3.6	Παράδειγμα χρήσης: Παρουσίαση βασικών εννοιών του προγραμματισμού, με τη βοήθεια μιας εφαρμογής/παιχνιδιού σχεδιασμένη με το App Inventor, σε μη ειδικευμένους στον προγραμματισμό φοιτητές. ....	61
3.6.1	Περιγραφή της Εφαρμογής.....	62
3.6.2	Δομή της εφαρμογής.....	62
3.6.3	Παραδείγματα χρήσης και εμφάνισης των προγραμματιστικών εννοιών στην εφαρμογή. 65	
	BIBΛΙΟΓΡΑΦΙΑ.....	69

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<b>Εικόνα 1.1:</b> Ο «κώνος» του Dale.....	<b>16</b>
<b>Εικόνα 2.1:</b> Το περιβάλλον του Scratch.....	<b>38</b>
<b>Εικόνα 2.2:</b> Το περιβάλλον του Alice.....	<b>40</b>
<b>Εικόνα 2.3:</b> Το περιβάλλον του Kodu.....	<b>40</b>
<b>Εικόνα 2.4:</b> Μορφή εντολών του Kodu.....	<b>41</b>
<b>Εικόνα 2.5:</b> Παιχνίδια φτιαγμένα με το Stencyl.....	<b>41</b>
<b>Εικόνα 2.6:</b> Το περιβάλλον του Stencyl.....	<b>42</b>
<b>Εικόνα 2.7:</b> Προγραμματισμός με το Pocket Code .....	<b>42</b>
<b>Εικόνα 3.1:</b> Παράδειγμα κώδικα App Inventor .....	<b>44</b>
<b>Εικόνα 3.2:</b> Το περιβάλλον του App Inventor .....	<b>52</b>
<b>Εικόνα 3.3:</b> Το μενού του App Inventor .....	<b>52</b>
<b>Εικόνα 3.4:</b> Το πεδίο block του App Inventor.....	<b>55</b>
<b>Εικόνα 3.5:</b> Τα σεντ εντολών και τα αντικείμενα .....	<b>56</b>
<b>Εικόνα 3.6:</b> Δομή επανάληψης και δομή επιλογής στο App Inventor .....	<b>56</b>
<b>Εικόνα 3.7:</b> Δομή επιλογής με λογικές πράξεις .....	<b>57</b>
<b>Εικόνα 3.8:</b> Δημιουργία χρώματος RGB με χρήση λίστας.....	<b>58</b>
<b>Εικόνα 3.9:</b> Δημιουργία και αρχικοποίηση μεταβλητής.....	<b>58</b>
<b>Εικόνα 3.10:</b> Μπλοκ χειρισμού των διαδικασιών.....	<b>59</b>
<b>Εικόνα 3.11:</b> Η πρώτη οθόνη με συμπληρωμένο το πρώτο επίπεδο .....	<b>62</b>
<b>Εικόνα 3.12:</b> Η δεύτερη οθόνη όπως εμφανίζεται όταν επιλεγεί ο χρήστης επίπεδο.....	<b>62</b>
<b>Εικόνα 3.13:</b> Η δεύτερη οθόνη όταν ο χρήστης «ζυγίζει» δυο από τα κλάσματα .....	<b>63</b>
<b>Εικόνα 3.14:</b> Διάγραμμα διάρθρωσης δομής επιλογής .....	<b>64</b>
<b>Εικόνα 3.15:</b> Μπλοκ εντολής δομής επιλογής στο App Inventor .....	<b>65</b>
<b>Εικόνα 3.16:</b> Μπλοκ σχετικών και λογικών τελεστών στο App Inventor .....	<b>65</b>
<b>Εικόνα 3.17:</b> Μπλοκ εντολών στην εφαρμογή μας στο App Inventor .....	<b>66</b>
<b>Εικόνα 3.18:</b> Παράδειγμα εντολών χωρίς το μπλοκ εντολών επανάληψης .....	<b>66</b>
<b>Εικόνα 3.19:</b> Μπλοκ εντολών δομής επανάληψης στην εφαρμογή μας στο App Inventor .....	<b>67</b>



## ΚΕΦΑΛΑΙΟ 1

# 1 ΔΙΔΑΚΤΙΚΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

## 1.1 Εισαγωγή

Η εποχή μας, εποχή της ύστερης νεωτερικότητας ή μετανεωτερικότητας, σηματοδοτεί μια νέα ιστορική περίοδο, καθώς αποτελεί το τέλος της νεωτερικότητας, ως αποτέλεσμα των διαφορετικών αναγκών και των αλλαγών που παρατηρούνται την περίοδο αυτή. Η ταυτότητα της μετανεωτερικότητας δεν έχει ακόμη αποσαφηνιστεί πλήρως. Εντούτοις, είναι δυνατόν να διακριθούν κάποια χαρακτηριστικά της γνωρίσματα που καταδεικνύουν τόσο τη συνέχεια, όσο και την απόσταση από το νεωτερικό κληρονομή. Ο πλουραλισμός, η πολυπολιτισμικότητα, η μεγάλη έκρηξη της τεχνολογίας, η ανάδυση της κοινωνίας της γνώσης, η παγκοσμιοποίηση της οικονομίας και του πολιτιστικού κεφαλαίου σηματοδοτούν την έλευση σε μια νέα εποχή που τη χαρακτηρίζει η μετάβαση από τη βιομηχανική κοινωνία στην κοινωνία της πληροφορίας και της γνώσης, από την εθνικιστική στην πολυπολιτισμική κοινωνία. (Ματθαίου, 1999)

Αδιαμφισβήτητο ορόσημο της νέας αυτής εποχής αποτελεί η εκρηκτική ανάπτυξη της τεχνο-επιστήμης. Η ραγδαία εξέλιξη της τεχνολογίας δεν αποτελεί, ασφαλώς, φαινόμενο μόνο της εποχής μας. Οι επιστημονικές και τεχνολογικές εξελίξεις, όμως, των τελευταίων δεκαετιών έχουν σαφείς ποιοτικές διαφορές από τις προηγούμενες. Η επιτάχυνση της τεχνολογικής εξέλιξης με ξέφρενους ρυθμούς δοκιμάζει τα όρια της ικανότητας ανταπόκρισης του ατόμου και της κοινωνίας στις νέες συνθήκες. Οι νέες τεχνολογίες πληροφορίας και επικοινωνίας επέτρεψαν την άμεση και ανοικτή πρόσβαση όλων στην πληροφορία και εκμηδένισαν τις αποστάσεις, γεωγραφικές, ηλικιακές και ταξικές, προσφέροντας σε όλους ευκαιρίες για γνώση και ένα ανοικτό βήμα επικοινωνίας και έκφρασης. Υπό αυτό το πρίσμα και κάτω από το βάρος των ριζικών αυτών μεταβολών, το άτομο καλείται να διαφοροποιήσει άλλοτε παγιωμένες στάσεις και θεσμούς. Τα θεμελιώδη χαρακτηριστικά της γνώσης αμφισβητούνται, καθώς αυτή

τεμαχίζεται σε μικρές αυθύπαρκτες ψηφίδες και η εκπαίδευση προσανατολίζεται στην ανάπτυξη δεξιοτήτων που συνδέονται με τη διαχείριση και όχι με την κατάκτησή της. Ο δάσκαλος στην Κοινωνία της Πληροφορίας αναλαμβάνει σε νέα πλαίσια καινούρια καθήκοντα και επωμίζεται νέους ρόλους, διατηρώντας, ωστόσο, τον διαχρονικό ρόλο του βασικού στυλοβάτη και οραματιστή της εκπαιδευτικής προσπάθειας (Ντούσκας, 2007).

Σε αυτή την Κοινωνία της Πληροφορίας και των Νέων Τεχνολογιών, η εισαγωγή της σύγχρονης τεχνολογίας στην εκπαίδευση είναι πραγματικότητα κι η ουσιαστική συνεισφορά της αρχίζει από τη στιγμή που ο μαθητής γίνεται από παθητικός ακροατής –έστω μιας ελκυστικής παρουσίασης– χρήστης με επιλογές. Μέσα στους «κόσμους» του υπολογιστή, ο μαθητής αποκτά κίνητρο και μπορεί να βρει τη χαρά της ανακάλυψης και της αυτενέργειας, την ικανοποίηση της προσαρμογής του υλικού στις δικές του ανάγκες, την ευκαιρία να κυνηγήσει ο ίδιος την πληροφορία, τη δυνατότητα να διερευνήσει και να πειραματιστεί ελέγχοντας τις υποθέσεις του, την αυτοπεποίθηση για την ανάληψη της προσωπικής ευθύνης για το μονοπάτι που θα ακολουθήσει η προσωπική του εμπειρία. Ο δάσκαλος από παντογνώστης-μεταδότης ενός συνόλου πληροφοριών και γνώσεων γίνεται σχεδιαστής εκπαιδευτικών σεναρίων βασισμένων στα διαθέσιμα τεχνολογικά εργαλεία ή και συγγραφέας εφαρμογών προσαρμοσμένων στις απαιτήσεις των μαθητών του, συντονιστής, βοηθός, διευκολυντικός παράγοντας στη διαδικασία παροχής πλούσιων μορφωτικών εμπειριών, αλλά και κριτής της παιδαγωγικής αξίας των δραστηριοτήτων και των τεχνολογικών εργαλείων.

## 1.2 Ένταξη των Τ.Π.Ε. στην εκπαίδευση

Μια μεγάλη ποικιλία νέων μέσων και τεχνολογιών εμφανίστηκαν και καθιερώθηκαν τα τελευταία χρόνια (Tzanavaris, Nikiforos, 2012). Οι νέες τεχνολογίες συνιστούν ισχυρότατα εργαλεία και για αυτό επιδρούν καθοριστικά στη σύγχρονη κοινωνία με πολλούς τρόπους και σε πολλά πεδία και δραστηριότητες του ανθρώπου. Μετασχηματίζουν τις παραγωγικές διαδικασίες και τις μεθόδους επικοινωνίας. Αλλάζουν τις διαδικασίες πρόσβασης στις πληροφορίες και πρόσκτησης γνώσης. Με λίγα λόγια, κάθε στοιχείο της καθημερινότητας του ανθρώπου μεταβάλλεται ριζικά ακολουθώντας το κύμα των τεχνολογικών εξελίξεων. Πέρα, όμως, από αυτήν τη διάσταση των νέων τεχνολογιών, η εξάπλωσή τους στον χώρο της εκπαίδευσης φαίνεται να καλύπτει και μια αναγκαιότητα. Υπό αυτό το πρίσμα, τα προϊόντα της Πληροφορικής (προγράμματα, υπολογιστές, αυτοματισμοί, ρομποτικές διατάξεις, διαδίκτυο) συγκροτούν ένα σύνθετο πλέγμα εργαλείων που συμβάλλουν στην απόκτηση γνώσης και δεξιοτήτων, βοηθούν τη σκέψη και τις αποφάσεις των ανθρώπων, ατομικά και συλλογικά (Μαϊστρος, 2011). Καθώς τα εργαλεία αυτά, τα οποία συνήθως αναφέρονται και ως *Τεχνολογίες Πληροφορικής και Επικοινωνιών: ΤΠΕ*, τίθενται στην υπηρεσία της Εκπαίδευσης, επιτελούν έναν ουσιώδη ρόλο στον μετασχηματισμό της, έτσι ώστε να ανταποκρίνεται στις απαιτήσεις της σύγχρονης εποχής.

### 1.2.1 Από την παραδοσιακή στη σύγχρονη διδασκαλία

Κατά τον Αριστοτέλη τα όντα θεωρούνται ως πραγματικά και οι αισθήσεις ως αξιόπιστοι πληροφοριοδότες της πραγματικότητας. Χωρίς την αίσθηση των πραγμάτων είναι αδύνατη η οικοδόμηση της διάνοιας καθώς από την επαφή και μόνο με τα αντικείμενα θα σχηματιστούν οι πνευματικές εικόνες, παραστάσεις και έννοιες, με τις οποίες εργάζεται το πνεύμα. Η γνωστή ρήση «ουδέν εν τω νω, ο μη πρότερον εν τη αισθήσει» αποδίδεται στον Αριστοτέλη καθώς υποστηρίζει ότι χωρίς την κατ' αίσθηση αντίληψη ούτε να μάθει κανείς ούτε να καταλάβει κάτι μπορεί, και όταν ακόμα εργάζεται θεωρητικά, πρέπει να έχει κάποια πνευματική εικόνα να σκεφτεί. Γιατί οι πνευματικές εικόνες είναι όμοιες με τα αντικείμενα που βλέπει, αλλά άυλες. (Γιαννικοπούλου, 1989) Για πολλούς αιώνες, παρά την αριστοτελική ρήση, το κύριο μέσο διδασκαλίας ήταν το βιβλίο και ο λόγος του δασκάλου καθιστώντας τη διδακτική και μαθησιακή διαδικασία



λογοκοπική διδασκαλία. Ο πρώτος παιδαγωγός που αντέδρασε σε αυτού του τύπου μεσαιωνική διδασκαλία ήταν ο J.A. Comenius (1592-1670) εξαίροντας την αξία της εποπτείας και τονίζοντας την αναγκαιότητά της. Στο έργο του *Didactica Magna* (Μεγάλη Διδακτική) τονίζει ότι για τον διδάσκοντα όλα πρέπει, όπου είναι δυνατόν, να παρουσιάζονται στις αισθήσεις, ό,τι ορατό στην όραση, ό,τι ακουστό στην ακοή, ό,τι οσφραντό στην όσφρηση, ό,τι γευστό στη γεύση, ό,τι απτό στην αφή. Κι αν μπορεί κάτι να γίνει αντιληπτό με διαφορετικές αισθήσεις, πρέπει να προβάλλεται ταυτόχρονα στις διαφορετικές αυτές αισθήσεις. Το 1658 ο Comenius εξέδωσε το πρώτο εικονογραφημένο βιβλίο «*Orbis sensualium pictus*» (Ο κόσμος σε εικόνες), που χρησιμοποιήθηκε για πολλά χρόνια στη διδασκαλία της λατινικής γλώσσας καθιστώντας δίκαια τον Comenius πατέρα της εποπτικής διδασκαλίας.

Η «αρχή της εποπτείας» τονίστηκε ιδιαίτερα κι από τον J.J. Rousseau (1712-1778): «Τα πράγματα! Τα πράγματα! Δε θα είναι ποτέ αρκετή αυτή η επανάληψή μου, αφού δίνουμε τόσο υπερβολική δύναμη στις λέξεις».

Λίγο αργότερα ο J.H. Pestalozzi (1746-1827), θα αναγνωρίσει την εποπτεία ως τη σημαντικότερη βασική αρχή της διδασκαλίας, το μόνο θεμέλιο της πραγματικής γνώσης. «Κατάρρα στις γνώσεις που προσφέρονται μόνο με λόγια. Όπου απουσιάζει η εξωτερική εποπτεία, η γλώσσα γίνεται άψυχη και άδεια, η δύναμη της γνώσης χάνεται και γίνεται σχεδόν αδύνατο να φτάσουμε σε σαφείς έννοιες και στην αλήθεια».

Από τις αρχές του 21<sup>ου</sup> αιώνα, τόσο η 'αρχή της εποπτείας' όσο και η 'αρχή της αυτενέργειας' υποστηρίζονται ένθερμα σε χώρες της Ευρώπης και στην Αμερική στοχεύοντας στη δραστηριοποίηση και την αυτενέργεια των μαθητών με την αξιοποίηση κατάλληλων εποπτικών μέσων ώστε με την εποπτικοποίηση των περιεχομένων της διδασκαλίας αυτή να γίνεται αποτελεσματικότερη και η μάθηση ελκυστικότερη. Σπουδαίο παράδειγμα αποτελεί η ιταλίδα γιατρός και παιδαγωγός M. Montessori (1870-1952) που επινοεί και χρησιμοποιεί διδακτικά υλικά και μέσα για τη δραστηριοποίηση, την αισθητηριακή άσκηση και την αυτενέργεια των μαθητών ως βασικές προϋποθέσεις της πνευματικής τους ανάπτυξης.

**Ο «κώνος της εμπειρίας» του Ed. Dale** αποτελεί την πιο διαδεδομένη ταξινόμηση των μέσων διδασκαλίας και μάθησης με κριτήριο το βαθμό ή το επίπεδο αφαίρεσης. Οι διαφορετικές μορφές εμπειρίας (άμεση εμπειρία, εμπειρία εικόνων, εμπειρία

συμβόλων) και οι αντίστοιχες μορφές διδασκαλίας-μάθησης χαρακτηρίζουν τη θεωρητική του βάση. (Κανάκης, 1989)



ΕΙΚΟΝΑ 1.1.2.11 Ο «ΚΩΝΟΣ» ΤΟΥ DALE

συμμετοχής του μαθητή. Κατ' αυτόν τον τρόπο, ανεβαίνοντας τα επίπεδα του κώνου, ο βαθμός της αμεσότητας των μέσων με την πραγματικότητα μειώνεται σταδιακά, τα δε γλωσσικά σύμβολα στην κορυφή του, αντιπροσωπεύουν μόνο συμβατικά την πραγματικότητα και επιτρέπουν στο μαθητή να δρα εσωτερικά, με τη σκέψη του. Αυτή η διάκριση δεν είναι απόλυτη καθώς οι ζώνες ταξινόμησης εκλαμβάνονται ως συμβατικές διαιρέσεις που επιτρέπουν στον εκπαιδευτικό να κατανοήσει ότι η μάθηση είναι δυνατόν να επιτυγχάνεται με την αξιοποίηση συγκεκριμένης κατηγορίας μέσων και να συνειδητοποιήσει ότι όσο απομακρύνεται από την άμεση πραγματικότητα ο κίνδυνος για ασάφεια, σύγχυση και παρανόηση από το μαθητή αυξάνει. Εξάλλου, η ταξινόμηση γίνεται ακόμη πιο σχετική αφού ο βαθμός αφαίρεσης ή εποπτείας ενός μέσου εξαρτάται από την προηγούμενη εμπειρία του ατόμου που το χρησιμοποιεί (Τριλιανός, 2003).

Βασικό σημείο της ταξινόμησης του Dale είναι ότι ο «κώνος της εμπειρίας», εκτός από τη γενική αντιστοίχιση των διαφορετικών μορφών εμπειρίας με τις μορφές διδασκαλίας-μάθησης (μάθηση με δράση, μάθηση με παρατήρηση, μάθηση με εξήγηση), δεν κατευθύνει το δάσκαλο να επιλέγει κάθε φορά το αποτελεσματικότερο για τους καθορισμένους μαθησιακούς στόχους και τα περιεχόμενα της διδασκαλίας μέσο (Κανάκης, 1989).

Έρευνες μαρτυρούν ότι ο βαθμός συμμετοχής των αισθήσεων ποικίλει σε σχέση με την αντίληψη των ερεθισμάτων του περιβάλλοντος: 83% όραση, 11% ακοή, 3,5% όσφρηση,

Ο «κώνος» του Dale (εικόνα 1.1) δίνει τη δυνατότητα στο δάσκαλο να προσδιορίσει την αμεσότητα της εμπειρίας και το επίπεδο αφαίρεσης των διαφορετικών μέσων διδασκαλίας. Η διάκριση σε έντεκα κατηγορίες-ζώνες δηλώνουν το βαθμό της άμεσης αισθητηριακής

1,5% αφή, 1% γεύση. Επισημαίνεται ότι η παρατήρηση είναι μεν πολύ σημαντική διεργασία στη συγκράτηση και διατήρηση του μηνύματος, αλλά καλύτερα αποτελέσματα προκύπτουν από το συνδυασμό της παρατήρησης με την ακοή, την αλληλεπίδραση και την πρακτική εφαρμογή. (Κανάκης, 1989, Σολωμονίδου, 1999)

Με τη διάδοση των μέσων μαζικής επικοινωνίας, χάρη στην πρόοδο της τεχνολογίας, η Παιδαγωγική επιστήμη αρχικά αποστασιοποιήθηκε από την εξέλιξη αυτή και τήρησε αρνητική στάση. Αργότερα, εμφανίζεται μια κριτική στάση απέναντί τους, ιδιαίτερα απέναντι στη τηλεόραση και σήμερα στο χώρο του διαδικτύου, και τίθενται ερωτήματα σχετικά με τη διαπαιδαγώγηση των μαθητών για τη σωστή χρήση τους.

Στις μέρες μας επικρατεί παιδαγωγική πολυφωνία για τον τρόπο αντιμετώπισης των μέσων διδασκαλίας και μάθησης. Η αξιοποίηση των μέσων επικοινωνίας και η εκπαιδευτική τεχνολογία θεωρούνται απαραίτητα στον εμπλουτισμό του διδακτικού και μαθησιακού έργου καθώς αυξάνουν την αποτελεσματικότητά του, στο βαθμό που δεν καταδικάζουν το μαθητή σε παθητικό και άκριτο δέκτη, αλλά συνδυάζουν την άμεση διαπροσωπική επικοινωνία με την αυτενεργό δράση των μαθητών. (Κανάκης, 1989)

Όπως διαπιστώνει ο Τζιμογιάννης (2002), παρά το γεγονός ότι οι εκπαιδευτικοί, στην πλειονότητά τους, αντιμετωπίζουν θετικά την ένταξη των ΤΠΕ στο σχολείο, οι προσεγγίσεις τους φαίνεται να έχουν περισσότερο καταναλωτικά χαρακτηριστικά, παρά να είναι συγκροτημένες στη βάση ενός παιδαγωγικού πλαισίου που αξιοποιεί τις ΤΠΕ.

### **1.2.2 Φάσεις και μοντέλα ένταξης των Τ.Π.Ε. στην εκπαίδευση**

Το σύγχρονο εκπαιδευτικό περιβάλλον θέλει τις ΤΠΕ να αποτελούν πλέον αναπόσπαστο εργαλείο της διδακτικής και μαθησιακής διαδικασίας (Μήτκας, Τσουλής, Πόθος, 2011). Η επισκόπηση των επιμέρους μοντέλων ή προσεγγίσεων που σχετίζονται με την εισαγωγή, την ένταξη και την ενσωμάτωση των ΤΠΕ στα διάφορα εκπαιδευτικά συστήματα μπορεί να επιτευχθεί γύρω από δύο διαφορετικούς άξονες: τον άξονα της χρονολογικής εξέλιξης με τις συνεπακόλουθες αλλαγές προβληματικής και τον άξονα των διαφορετικών παιδαγωγικών μοντέλων που αφορούν στη μεθοδολογία ένταξης και τις τροποποιήσεις που επιφέρουν στο πρόγραμμα σπουδών.

### **1.2.2.1 Χρονολογική εξέλιξη εισαγωγής των Τ.Π.Ε. στην εκπαίδευση**

Σύμφωνα με τη διαθέσιμη βιβλιογραφία, η διαδικασία εισαγωγής των ΤΠΕ στην Εκπαίδευση πραγματοποιήθηκε σε τέσσερα διακριτά στάδια:

- Κατά την πρώτη ή εισαγωγική φάση, η οποία διήρκησε μέχρι το 1970, παρατηρήθηκαν οι πρώτες προσπάθειες ένταξης της τεχνολογίας στην εκπαίδευση. Χαρακτηριστικό ορόσημο στην προβληματική αυτής της περιόδου αποτελεί η χρήση των διδακτικών μηχανών.
- Κατά τη δεύτερη φάση (1970-1980), πραγματοποιούνται πιλοτικές εφαρμογές εισαγωγής υπολογιστών στα σχολεία (δευτεροβάθμιας εκπαίδευσης), με βασική επιδίωξη τη μάθηση για τους υπολογιστές και όχι με τους υπολογιστές.
- Κατά την τρίτη φάση (1980-1990), πραγματοποιείται η ένταξη της πληροφορικής και των τεχνολογιών στα διάφορα εκπαιδευτικά συστήματα και κυρίως αυτά των ανεπτυγμένων χωρών. Παρατηρούνται στο στάδιο αυτό ολοκληρωμένα προγράμματα, η χρήση των οποίων εξαπλώνεται συστηματικά στην πάροδο της δεκαετίας.
- Η τέταρτη φάση ξεκινά από το 1990 και αφορά στις προσπάθειες που συνεχίζονται μέχρι και σήμερα με σκοπό την επίτευξη πλήρους ενσωμάτωσης των ΤΠΕ στην εκπαίδευση. Στο στάδιο αυτό, οι τεχνολογίες αξιοποιούνται πρωτευόντως ως μέσο και υποχωρεί η έμφαση στη διδασκαλία της χρήσης τους.

### **1.2.2.2 Μοντέλα ένταξης των ΤΠΕ στην εκπαίδευση**

Παράλληλα με την παραπάνω προσέγγιση, η καθημερινή εκπαιδευτική πρακτική έχει αναδείξει τρεις βασικές τάσεις χρήσης των νέων τεχνολογιών της πληροφορίας και της επικοινωνίας στην εκπαιδευτική διαδικασία:

- Η θεώρηση της τεχνολογίας ως αυτοτελούς γνωστικού αντικειμένου το οποίο ενσωματώνεται στη διδακτέα ύλη των διαφόρων βαθμίδων εκπαίδευσης αποτελεί τη λεγόμενη τεχνοκρατική προσέγγιση. Πρόκειται για ένα ντετερμινιστικό μοντέλο (Μακράκης, 2000), το οποίο προβάλλει ως βασική επιδίωξη την απόκτηση γνώσεων πάνω στη λειτουργία των υπολογιστών και την εισαγωγή τον προγραμματισμό τους.

- Η θεώρηση της τεχνολογίας αποκλειστικά ως μέσου διδασκαλίας, ως εργαλείου και όχι αντικειμένου μάθησης αποτελεί το ολοκληρωμένο ή ολιστικό μοντέλο ένταξης. Σύμφωνα με αυτήν τη θεωρία, η χρήση των ΤΠΕ εντάσσεται στη διδασκαλία όλων των μαθημάτων, ως έκφραση μιας διαθεματικής προσέγγισης της μάθησης. Ωστόσο, η εφαρμογή της προσέγγισης αυτής θεωρείται βραχυπρόθεσμα μη εφαρμόσιμη (Τζιμογιάννης, 2002), καθώς προϋποθέτει τη θεμελίωση γνώσεων και πρακτικών, οι οποίες βρίσκονται σε πρώιμο ακόμη στάδιο.
- Σε μια προσπάθεια συγκερασμού των δύο προηγούμενων προσεγγίσεων, διαμορφώνεται ένα άμεσα εφικτό, μικτό μοντέλο. Σύμφωνα με αυτό, η διδασκαλία ενός αμιγούς μαθήματος γενικών γνώσεων πληροφορικής θα πρέπει να συνδυαστεί με την προοδευτική ένταξη της χρήσης των ΤΠΕ ως μέσου στήριξης της μαθησιακής διαδικασίας σε όλα τα γνωστικά αντικείμενα του προγράμματος σπουδών. Με αυτόν τον τρόπο, καλύπτονται οι ανάγκες τεχνολογικού αλφαριθμητισμού με σαφή προσανατολισμό στην επίτευξη του ολιστικού μοντέλου, το οποίο θα μεγιστοποιήσει και τα παιδαγωγικά οφέλη.

### 1.3 Θεωρίες μάθησης και Τ.Π.Ε.

Το σχολείο και οι εκπαιδευτικοί για να επιτύχουν τους εκπαιδευτικούς στόχους της γνώσης και της μάθησης γενικότερα, λειτουργούν αξιοποιώντας πρακτικές και αρχές που έχουν διατυπωθεί από παιδαγωγούς και ερευνητές. Οι διάφορες θεωρητικές προσεγγίσεις που έχουν προταθεί πηγάζουν από άλλη φιλοσοφία και εφαρμόζουν διαφορετικές αρχές και μεθοδολογίες. Η ενασχόληση με τη μελέτη των κυριότερων θεωριών μάθησης θεωρείται απαραίτητη για την εφαρμογή των γονιμότερων στοιχείων τους στην εκπαιδευτική πράξη, αλλά και για τη διαμόρφωση ικανοποιητικών κριτηρίων αξιολόγησης της παιδαγωγικής σημασίας και της αποτελεσματικότητας των εκάστοτε προσεγγίσεων. Στη βιβλιογραφία, προτείνονται διάφορες κατηγοριοποιήσεις των θεωριών μάθησης. Στην παρούσα έρευνα, θα επιχειρηθεί η σύντομη παρουσίαση ορισμένων από τις πιο αντιπροσωπευτικές θεωρητικές προσεγγίσεις της μάθησης: του συμπεριφορισμού, του εποικοδομισμού και της κονστρουκτιβιστικής μάθησης. Στην πρώτη κατηγορία του Συμπεριφορισμού εντάσσονται οι: I.Pavlov, J.B. Watson, E.L.Thorndike, B.F.Skinner (Γραμμική Οργάνωση), N. Crowder (Διακλαδισμένη Οργάνωση), R. Gagne (Διδακτικός σχεδιασμός). Στη δεύτερη κατηγορία του εποικοδομισμού κατατάσσουμε τους: J. Piaget, S. Papert (παιδαγωγική θεωρία της LOGO), R. Gagne και A. Newell και H. Simon (θεωρία της επεξεργασίας της πληροφορίας), Boyle (Μαθησιακά περιβάλλοντα με υπολογιστές). Στις Κοινωνικοπολιτικές θεωρίες μάθησης (θεωρία της δραστηριότητας) έχουν συμβάλει ο J.Bruner με την ανακαλυπτική μάθηση, ο L.Vygotsky με την επικοινωνιακή και πολιτισμική διάσταση της μάθησης και τέλος ο Vygotsky με τον Leontiev, τον Luria και τον Nardi, πρόσωπα που ασχολήθηκαν με τη Θεωρία της δραστηριότητας.

Οι εκπρόσωποι του Συμπεριφορισμού υποστήριξαν πως η μάθηση είναι τροποποίηση της συμπεριφοράς. Δίνεται στον οργανισμό ένα ερέθισμα και αυτός αντιδράει. Όλα αυτά γίνονται μέσω μιας προγραμματισμένης διδασκαλίας. Ο δάσκαλος συλλέγει και δίνει στον μαθητή τα κατάλληλα ερεθίσματα. Όταν ο μαθητής επιδείξει τη ζητούμενη συμπεριφορά, ο δάσκαλος την ενισχύει θετικά. Σε αντίθετη περίπτωση, τον επανατροφοδοτεί με ερεθίσματα μέχρι να επιτύχει την επιδιωκόμενη αντίδραση-συμπεριφορά. Ο μαθητής θεωρείται παθητικός αποδέκτης κάθε πληροφορίας που δίνεται από τον δάσκαλο, ο οποίος έχει καταμερίσει το μάθημα σε διάφορες ενότητες.

Βασική αρχή αυτής της θεωρίας μάθησης είναι πως η μάθηση πρέπει να επικεντρώνεται στην επίτευξη της επιθυμητής συμπεριφοράς και όχι σε πνευματικές επεκτάσεις ή γλωσσικές αναπαραστάσεις. Επιπροσθέτως, η διαδικασία της μάθησης οδηγεί σε μεταβολή της συμπεριφοράς, η μελέτη της οποίας πρέπει να εκκινεί από τα απλά της στοιχεία που είναι τα ειδικά ερεθίσματα που θα οδηγήσουν και σε ειδικές αντιδράσεις. Ο B.F.Skinner υποστηρίζει ότι για να συντελεστεί μάθηση απαιτείται η ενεργός εμπλοκή του μαθητή και πως η διδακτέα ύλη θα πρέπει να διδάσκεται σε ενότητες και βαθμιαία, ανάλογα με την πρόοδο του μαθητή, να γίνεται άμεσα η επαλήθευση στην απόκριση του μαθητή και να ενισχύεται θετικά η σωστή απάντηση στην ερώτηση που τίθεται κάθε φορά (γραμμική οργάνωση: ερώτηση-απάντηση, ερώτηση- απάντηση, κ.ο.κ.). Ο N. Crowder ακολουθεί πορεία παρόμοια με του Skinner, με τη διαφορά ότι, ενώ βαθμιαία και προοδευτικά προχωρεί ο μαθητής προς την κατάκτηση της νέας γνώσης μέσα από τις ερωταποκρίσεις, η πορεία που θα ακολουθηθεί εξαρτάται κάθε φορά από τη συγκεκριμένη απάντηση που θα δώσει. Κάθε απάντηση, δηλαδή, οδηγεί σε διαφορετικό «μονοπάτι». Στον Skinner επιδιώκεται να αποφεύγονται τα λάθη από τον μαθητή, ενώ στον Crowder επιδιώκεται κάθε φορά που ο μαθητής θα κάνει λάθος, να του παρέχονται περαιτέρω εξηγήσεις. Το μοντέλο του Διδακτικού σχεδιασμού βασίζεται στην εμφάνιση του υπολογιστή στη δεκαετία του '70. Η χρήση του περιοριζόταν μονάχα στην υλοποίηση του βιβλίου από τον υπολογιστή μέσα από πολλαπλές ερωτήσεις. Στη σύγχρονη εκδοχή του, πρόκειται για μοντέλα και προγράμματα διδασκαλίας τα οποία έχουν διαμορφωθεί με τη βοήθεια του «μοντέλου» του διδακτικού σχεδιασμού. Η θεωρίας μάθησης του Συμπεριφορισμού θεωρείται από τις σύγχρονες παιδαγωγικές τάσεις πως απομακρύνει τον μαθητή από τη σκέψη και την νόηση και ότι σκοπό έχει μόνο την απόκτηση συγκεκριμένων δεξιοτήτων χαμηλού επιπέδου, ενώ ο μαθητής είναι παθητικός και αδρανής σε όλη τη διάρκεια της διδασκαλίας. Ο διαχωρισμός επίσης της διδακτέας ύλης σε μέρη και ενότητες περιπλέκει τη διαδικασία και ο εκπαιδευτικός δύσκολα βρίσκει λύσεις και μπορεί να τη χειριστεί. Η αποκτηθείσα γνώση είναι και αυτή αδρανής και δεν μπορεί να εφαρμοσθεί για την επίλυση πραγματικών, ίσως και περισσότερο πολύπλοκων προβλημάτων. Προωθείται ο ανταγωνισμός και η ατομική εργασία, ενώ η μορφή αυτή της μάθησης και της διδασκαλίας δεν μπορεί να εφαρμοσθεί σε όλους τους τομείς της γνώσης.

Οικοδομισμός είναι μια διαδικασία που στηρίζεται στην οικοδόμηση νοημάτων που γίνεται μέσα από την οργάνωση και προσαρμογή της νέας πληροφορίας στην ήδη υπάρχουσα. Ο μαθητής έχει ενεργό ρόλο στην ανοικοδόμηση αυτή της γνώσης του, ενώ ο ρόλος του δασκάλου είναι υποστηρικτικός και συμβουλευτικός. Η παιδαγωγική θεωρία της LOGO, της οποίας το όνομα οφείλεται στην ελληνική λέξη «λογικό», βασίζεται πρώτον στο γεγονός ότι η εμπειρία σε αυτήν οδηγεί στην απόκτηση γνωστικών δεξιοτήτων και επίλυσης προβλημάτων για όλα τα μαθήματα, ενώ συμβάλλει στη μάθηση μαθηματικών εννοιών. Σχετικά με τη Θεωρία της επεξεργασίας της πληροφορίας, η γνωστική διεργασία συνίσταται από επεξεργασίες αναπαραστάσεων (βραχυπρόθεσμη μνήμη) και γνώσεων (μακροπρόθεσμη μνήμη). Οι αναπαραστάσεις είναι δομές περιστασιακές, ενώ οι γνώσεις είναι σταθεροποιημένες. Ο οικοδομισμός μέσα στα σύγχρονα μοντέλα σχεδιασμού εκπαιδευτικού υλικού, συμβάλλει ώστε να συνδέονται οι μαθησιακές δραστηριότητες που σκοπό έχουν την επίλυση προβλημάτων, με αντίστοιχες καταστάσεις εκτός σχολείου. Υπάρχουν, μάλιστα, κάποιες αρχές που πρέπει να τηρούνται κατά τον σχεδιασμό των δραστηριοτήτων. Πιο συγκεκριμένα, οι εμπειρίες που παρέχονται θα πρέπει να οικοδομούν τη γνώση και να έχουν πολλές προοπτικές. Δεύτερον, θα πρέπει να χρησιμοποιούνται πολλαπλές μορφές αναπαραστάσεων. Τρίτον, η μάθηση θα πρέπει να γίνεται σε ρεαλιστικές προεκτάσεις, έτσι ώστε οι προβληματικές που ανακύπτουν να συνδέονται με την πραγματικότητα και να ενθαρρύνεται το αυτοσυναίσθημα κατά την οικοδόμηση της γνώσης.

Η τελευταία κατηγορία από τις Θεωρίες μάθησης, είναι οι Κοινωνικοπολιτισμικές θεωρίες. Ο Bruner, έκανε λόγο για ανακαλυπτική μάθηση. Ο μαθητής εδώ πειραματίζεται και συμμετέχει σε διάφορες και διαφορετικές δραστηριότητες με αποτέλεσμα να αποκτά δεξιότητες ή και να ανακαλύπτει αρχές. Πρόκειται για τις έμπρακτες αναπαραστάσεις. Λαμβάνουν μέρος όμως και οι εικονικές αναπαραστάσεις που συνίστανται από εσωτερικές νοητικές εικόνες και αφορούν σε δομές χώρου ενώ εμφανίζονται και οι συμβολικές αναπαραστάσεις (αναπαραστάσεις σχέσεων με σύμβολα). Ο ρόλος του εκπαιδευτικού κατά τον Bruner, είναι ρόλος εμπνευστή και συντονιστή στη διαδικασία της μάθησης και το αναλυτικό πρόγραμμα πρέπει να οργανώνεται σε σπειροειδή μορφή. Σημαντική στη τελευταία αυτή κατηγορία των



θεωριών μάθησης είναι και η συμβολή του Vygotsky. Το παιδί είναι δρών υποκείμενο στη διαδικασία της μάθησης και αναπτύσσει τη νόησή του μέσα από την κοινωνική αλληλεπίδραση στην οποία σημαντικό ρόλο παίζει η γλώσσα. Η θεωρία του βασίζεται στη *ζώνη της επικείμενης ανάπτυξης*. Πρόκειται για μια «ανεξερεύνητη περιοχή στο εσωτερικό του κάθε μαθητή, ο οποίος βρίσκεται σε μια εν δυνάμει λανθάνουσα κατάσταση εξέλιξης». Η εμπλοκή του δασκάλου, του γονέα κι γενικότερα του ενήλικα και ο ρόλος του κοινωνικού περιβάλλοντος διακρίνονται για την εξαιρετη σημασία τους στη γνωστική αυτή ανάπτυξη του παιδιού. Κάθε ένα δραστηριοποιείται με αφετηρία τις γνώσεις που έχει. Η ΖΕΑ μπορεί να πραγματοποιήσει δραστηριότητες γύρω από τον πυρήνα αυτόν, των γνώσεων του παιδιού, μόνο όταν υπάρχουν κι άλλοι εμπλεκόμενοι. Τέλος, η θεωρία της δραστηριότητας βασίζεται στην αρχή ότι «η ανθρώπινη δράση διαμεσολαβείται από πολιτισμικά σύμβολα, λέξεις και εργαλεία, τα οποία επιδρούν στην δραστηριότητα του ατόμου και συνεπώς στις νοητικές του διεργασίες». Η δραστηριότητα είναι η βασική μονάδα ανάλυσης και αποτελείται από το υποκείμενο, το αντικείμενο, τους κανόνες και τις λειτουργίες. Συντελείται μέσα από όργανα, σήματα και γλώσσες που δημιουργούν τα άτομα προκειμένου να ελέγξουν τη συμπεριφορά τους. Η θεωρία της δραστηριότητας βρίσκει εφαρμογή σε δραστηριότητες ανθρώπου-μηχανής και σε μαθησιακά περιβάλλοντα με χρήση υπολογιστή. Αυτό το τελευταίο αναφέρεται στη συνεργατική μάθηση, την αλληλεπίδραση δηλαδή υποκειμένου (μαθητή) - αντικειμένου (στόχου μάθησης) και διαθέσιμων εργαλείων.

#### **1.4 Ο μετασχηματισμός της διδασκαλίας με τη χρήση του Η/Υ**

Η σύγχρονη βιβλιογραφία ασχολήθηκε εκτενώς με την ένταξη του ηλεκτρονικού υπολογιστή (Η/Υ) ανάμεσα στα μέσα επικοινωνίας και μάθησης. Πιο συγκεκριμένα, αυτό που απασχόλησε τους θεωρητικούς της παιδαγωγικής επιστήμης είναι οι λειτουργίες του υπολογιστή, η συμβολή του στη μάθηση, οι διάφορες χρήσεις και οι συνέπειές τους, οι αλλαγές που αυτές θα επιφέρουν στη μαθησιακή διαδικασία αλλά και στις συνθήκες εργασίας του ίδιου του εκπαιδευτικού, καθώς και οι τυχόν παρενέργειες από την εκτεταμένη και άκριτη χρήση του.

##### **1.4.1 Οι εκπαιδευτικές λειτουργίες του Η/Υ**

Ανάμεσα στις εκπαιδευτικές λειτουργίες του Η/Υ, περιλαμβάνεται ο ρόλος του ως γνωστικού αντικειμένου, ως πηγής πληροφόρησης, ως γνωστικού εργαλείου και ως διδακτικού μέσου (Ράπτης-Ράπτη,2007). Πιο αναλυτικά:

- Γνωστικό αντικείμενο: αφορά στη διδασκαλία της πληροφορικής ως αυτόνομου γνωστικού πεδίου με στόχο την προώθηση του λεγόμενου «τεχνολογικού αλφαριθμητισμού». Ο όρος αυτός προέκυψε από τα νέα δεδομένα, τα οποία καθιστούν την εκπαίδευση αυτή προϋπόθεση για την απόκτηση των απαραίτητων προεπαγγελματικών γνώσεων και δεξιοτήτων.
- Πηγή πληροφόρησης: ο Η/Υ προσφέρει τεράστιες δυνατότητες, όσον αφορά στη ροή πληροφοριών. Ο χρήστης δεν αποκτά απλώς πρόσβαση στις πληροφορίες μέσω των δικτύων, αλλά είναι σε θέση να τις αποθηκεύει, να τις οργανώνει, να τις συσχετίζει και να τις ανακαλεί, όποτε το επιθυμεί. Ωστόσο, είναι σημαντικό να αναφέρουμε ότι σε αυτήν τη λειτουργία ενέχεται ο κίνδυνος μονόπλευρης πληροφόρησης από μη έγκυρες πηγές. Για το λόγο αυτό, είναι απαραίτητη η εκπαίδευση του χρήστη, αφενός, στις διαδικασίες απόκτησης της πληροφορίας και, αφετέρου, στις δεξιότητες που απαιτούνται για την «ενεργή αναζήτηση και την κριτική ανάγνωση του πληροφοριακού κειμένου» (Ράπτης –Ράπτη,2007).
- Διδακτικό μέσο: η λειτουργία αυτή προϋποθέτει τη μεσολάβηση του κατάλληλου προγράμματος -εκπαιδευτικού λογισμικού. Η χρήση του Η/Υ ως ενός πολυαισθητηριακού μέσου διδασκαλίας προσφέρει νέες δυνατότητες μάθησης σε οποιοδήποτε γνωστικό αντικείμενο και διευκολύνει τον εκπαιδευτικό στην επίτευξη

ορισμένων στόχων και προϋποθέσεων επιτυχούς διδασκαλίας, όπως η εξατομίκευση, η ενεργή συμμετοχή των μαθητών, η αύξηση της παραγωγικότητας κ.ά.

- Γνωστικό εργαλείο: αφορά στη διαμόρφωση ενός εργαλείου, που δε στοχεύει μόνο στη διδασκαλία και κατανόηση ενός μόνο διδακτικού αντικειμένου, αλλά και στην εκπαίδευση του μαθητή στις νοητικές δομές και γνωστικές διαδικασίες, που χρειάστηκαν για να το κατακτήσει. Η λειτουργία αυτή απομακρύνει τον Η/Υ από τον στόχο της «μηχανιστικής γνώσης» της παραδοσιακής εκπαίδευσης και βοηθά το μαθητή, αφενός, να εμπεδώσει ένα σύστημα θεμελιωδών γνώσεων και δεξιοτήτων και, αφετέρου, να εξασκηθεί στις διαδικασίες που θα επιτρέψουν τη μεταφορά και αξιοποίηση αυτών των δεξιοτήτων στην επίλυση γενικότερων προβλημάτων.

#### **1.4.2 Πλεονεκτήματα από τη χρήση του Η/Υ**

Ανάμεσα στα πλεονεκτήματα του Η/Υ, συγκαταλέγονται η προγραμματισιμότητά του, η αλληλεπιδραστικότητά του με το μαθητευόμενο καθώς και η προσαρμοστικότητά του, που προάγει και εφαρμόζει την αρχή της εξατομικευμένης διδασκαλίας, πολλαπλασιάζοντας τα αποτελέσματά της, τόσο σε ποσοτικό, όσο και σε ποιοτικό επίπεδο. Για το λόγο αυτό, η εισαγωγή του στη μαθησιακή διαδικασία θεωρείται ιδιαίτερα επικερδής τόσο για τον εκπαιδευτικό, όσο και για τον εκπαιδευόμενο.

Εφόσον στόχος της διδασκαλίας είναι η αφομοίωση και όχι απλώς η επαφή με τις παρεχόμενες γνώσεις, είναι σημαντικό για τον εκπαιδευτικό να εξασφαλίσει την προσοχή των μαθητών. Σε αυτήν την κατεύθυνση, το εποπτικό υλικό παίζει πρωτεύοντα ρόλο, προκαλώντας ισχυρές εντυπώσεις μέσα από τη ρεαλιστικότητα των παραστάσεων. Συμβάλλουν στην κατανόηση του μαθήματος, συμπληρώνοντας την απλή προφορική διδασκαλία και βοηθώντας το μαθητή που έχει αμφιβολίες και προβληματισμούς να «δει» την αλήθεια. Εξοικονομούν πολύτιμο χρόνο, αφενός χάρη στην ταχύτητα με την οποία εντυπώνονται οι πληροφορίες στο μυαλό των μαθητών και αφετέρου λόγω της συγκέντρωσης της προσοχής τους στη διδασκαλία. Καταπολεμούν τη λογοκοπία-βερμπαλισμό, τη μεγάλη -δηλαδή- χρήση του προφορικού λόγου, η οποία ευθύνεται για τον παθητικό χαρακτήρα της εκπαίδευσης. Προάγουν την αυτενέργεια, μια διδακτική αρχή που θεωρείται προϋπόθεση για την επιτυχημένη αγωγή και προάγεται

σε μεγάλο βαθμό από τα εποπτικά μέσα, χάρη στην αφύπνιση των δυνατοτήτων των μαθητών. Εξυπηρετούν τις ατομικές διαφορές, καθώς ανταποκρίνονται στις διαφορετικές ανάγκες του κάθε μαθητή, εφαρμόζοντας την αρχή της εξατομικευμένης διδασκαλίας. Κάνουν τη διδασκαλία επίκαιρη, αναδεικνύοντας τη σχέση του σχολείου και των παρεχόμενων σε αυτό γνώσεων με τη σύγχρονη ζωή και τις απαιτήσεις του μέλλοντος -εργασιακού και κοινωνικού. Αισθητοποιούν τη διδασκαλία, εξασφαλίζοντας, μέσα από τον ταυτόχρονο ερεθισμό πολλών αισθήσεων, καλύτερη αφομοίωση και πιο αποτελεσματική απομνημόνευση των μεταδιδόμενων πληροφοριών. Διοχετεύουν γνώσεις, διευκολύνοντας τη μεταβίβασή τους και εξασφαλίζοντας την ποιότητα και ποσότητά τους. Εκσυγχρονίζουν την εκπαίδευση, ικανοποιώντας, έτσι, ένα από τα πιο επίκαιρα αιτήματα των εκπαιδευτών και των εκπαιδευόμενων, οι οποίοι προσδοκούν από αυτή τη διαδικασία ενσωμάτωσης των νέων μέσων να οδηγήσει σε μία εκπαίδευση πιο γρήγορη, «πιο ρεαλιστική και παραστατική, πιο ελκυστική και αποδοτική». Καλλιεργούν το καλαισθητικό συναίσθημα, αλλά και συνδυάζουν θεωρία και πράξη, συγκεκριμενοποιώντας την αφηρημένη γνώση και εξυπηρετώντας το αίτημα για εμπειρική μάθηση. Τέλος, διευκολύνουν τη διδασκαλία, εξασφαλίζοντας τη δημιουργία εκείνης της ατμόσφαιρας, που θα διεγείρει τα κατάλληλα ερωτήματα και θα δώσει τις πιο πειστικές απαντήσεις.

#### **1.4.3 Προϋποθέσεις αποτελεσματικής χρήσης των νέων μέσων**

Το εκπαιδευτικό σύστημα, σε μια προσπάθεια να εκμεταλλευτεί αυτά τα χαρακτηριστικά, για τον ουσιαστικό εκσυγχρονισμό της διδασκαλίας, συμπεριλαμβάνει τη χρήση τους στο σχεδιασμό των Διαθεματικών Ενιαίων Προγραμμάτων (ΔΕΠΠΣ) και των Αναλυτικών Προγραμμάτων Σπουδών (ΑΠΣ). Μεγάλη σημασία κατά τη διαδικασία εκπόνησης του νέου εκπαιδευτικού υλικού κατέχει ο καθορισμός κάποιων ενιαίων κριτηρίων αξιολόγησης σχετικά με την παιδαγωγική καταλληλότητα, την αισθητική του υλικού, την αποτύπωση της διαθεματικότητας, τη διαδραστικότητα και τη δημιουργικότητα των λογισμικών. Στο σημείο αυτό, κρίνουμε σκόπιμο να αναφερθούμε σε κάποια χαρακτηριστικά κριτήρια αξιολόγησης των προτεινόμενων μέσων, τα οποία καλούνται να ελέγχουν τόσο οι διαμορφωτές του επίσημου αναλυτικού προγράμματος, όσο και οι εκπαιδευτικοί. Τα κριτήρια αυτά προσεγγίζονται και ταξινομούνται με

διαφορετικό τρόπο σε κάθε βιβλιογραφική αναφορά. Εμείς θα συνοψίσουμε ορισμένα από αυτά, υπό τη μορφή ερωτήσεων (Ράπτης-Ράπτη 2007:245):

- Καλύπτεται η διδακτέα ύλη επαρκώς;
- Ο χρόνος που απαιτείται είναι διαθέσιμος;
- Ανταποκρίνεται το διδακτικό υλικό στο επίπεδο των μαθητών;
- Είναι οι δραστηριότητες ενδιαφέρουσες και η παρουσίαση ελκυστική;
- Είναι προσεγγμένο από παιδαγωγικής άποψης, χωρίς ανεπιθύμητες παρενέργειες (π.χ. ενίσχυση προκαταλήψεων ή στερεοτύπων, φανατισμού και επιθετικότητας, κοινωνικά μη αποδεκτών συμπεριφορών);
- Προσφέρεται για συνεργατικές/σύνθετες δραστηριότητες;
- Τι είδους δεξιότητες, στάσεις και αξίες καλλιεργεί;
- Συνοδεύεται από επαρκείς οδηγίες και υλικό στήριξης του δασκάλου;
- Αξιοποιούνται σε ικανοποιητικό βαθμό οι αλληλεπιδραστικές δυνατότητες του υπολογιστή;
- Αφήνει περιθώρια για έλεγχο της μαθησιακής διαδικασίας από μέρος του μαθητή και του εκπαιδευτικού;
- Προσφέρονται ορισμένα χαρακτηριστικά του (γραφικά, ήχος, κίνηση κτλ.) για εποικοδομητική μάθηση ή λειτουργούν απλώς για εντυπωσιασμό;
- Η διόρθωση των λαθών γίνεται με παιδαγωγικά προσεγγμένο τρόπο ή βασίζεται σε παραδοσιακές νοοτροπίες;
- Η ενίσχυση των αντιδράσεων και της απόδοσης του μαθητή είναι αποτελεσματική και γίνεται με κατάλληλο τρόπο;
- Είναι φιλικό και εύκολο στη χρήση του;
- Αξίζει τις απαιτούμενες δαπάνες;

### **1.5 Προγραμματισμένη διδασκαλία: η θεωρητική θεμελίωση των νέων μέσων**

Η προγραμματισμένη διδασκαλία είναι μία μορφή διδασκαλίας που θεωρήθηκε ως η σημαντικότερη διδακτική επανάσταση της δεκαετίας του 1960 (Πηγιάκη,1998-Χαραλαμπόπουλος,2001). Η τεχνική της βασίζεται σε πρότυπα του συμπεριφορισμού και προωθεί το εξατομικευμένο μοντέλο διδασκαλίας. Κατά την προγραμματισμένη διδασκαλία, ο μαθητής «αυτοδιδάσκεται» επαγωγικά με μικρά βήματα. Τον ρόλο του εκπαιδευτικού αναλαμβάνουν να διεκπεραιώσουν τα ειδικά διδακτικά βιβλία (το πρόγραμμα της διδακτέας ύλης τεμαχισμένο σε ενότητες-πλαίσια) ή οι διδακτικές μηχανές (ηλεκτρονικές συσκευές, μεταξύ των οποίων και ο υπολογιστής, οι οποίες αναλαμβάνουν να παρουσιάσουν κάθε τμήμα της ύλης και στη συνέχεια να ελέγξουν το επίπεδο κατανόησης των μαθητών με ερωτήσεις και άμεση αξιολόγηση). Σε κάθε περίπτωση και ανεξάρτητα από τα χρησιμοποιούμενα τεχνικά μέσα, η προγραμματισμένη διδασκαλία υπόκειται πάντα σε κάποια κριτήρια ( Minder,1999):

Το παραδοσιακό μοντέλο προγραμματισμένων διδακτικών δομών ακολουθούσε ένα γραμμικό πρότυπο, που συμβάδιζε με τη βασική έκφραση της συντελεστικής παιδαγωγικής (Skinner, Persey). Σε αυτήν τη μορφή, ακολουθείται από το μαθητή μια ευθύγραμμη πορεία, στην οποία οι πληροφορίες παρουσιάζονται με τη μορφή μιας «προσεκτικά διατεταγμένης αλληλουχίας στοιχειωδών εντολών» (Clausse, 1972). Η άμεση αξιολόγηση του μαθητή και η επιβράβευση που δέχεται είναι τα στοιχεία εκείνα που κινητοποιούν την προσπάθειά του, όπως περιγράφεται από τη σχολή του συμπεριφορισμού.

Ωστόσο, σήμερα υπάρχουν πιο εξελιγμένα προγράμματα, τα οποία συνήθως υποστηρίζονται από υπολογιστή και επιτρέπουν την αυτοδιόρθωση των λαθών. Τα προγράμματα αυτά ονομάζονται δένδροειδή ή πολυακολουθιακά (Crowder). Σε αυτόν τον τύπο προγραμματισμένης δομής, αποκαθίσταται το δικαίωμα του λάθους, καθώς, σε αντίθεση με τα γραμμικά προγράμματα που έχουν στόχο την αποφυγή του λάθους, εδώ τα σφάλματα είναι μάλλον το ζητούμενο. Πράγματι, μέσα από μια αρκετά πολύπλοκη διακλαδιζόμενη πορεία, ο μαθητής μέσα από ένα ενδεχόμενο λάθος μπορεί να μάθει περισσότερα και να αποτρέψει «κάθε άλλη πιθανότητα επανάληψης της ίδιας

εσφαλμένης απάντησης» (Χρυσαφίδης, Χ.Χ.). Τα δένδροειδή προγράμματα καταρρίπτουν την παραδοσιακή αντίληψη ότι ο μόνος τρόπος για να διορθώσουμε το λάθος είναι να αποσυνθέσουμε τη μη αφομοιωμένη πληροφορία και προτείνουν την επεξεργασία του λάθους και μία «συμπληρωματική διορθωτική αλληλουχία» (Minder, 1999).

Οι υποστηρικτές της προγραμματισμένης διδασκαλίας τονίζουν τη σημασία της ενίσχυσης και της παρώθησης που προσφέρει στο μαθητή, καθώς και στη διευκόλυνση που παρέχεται κατά τη διδασκαλία ορισμένων μαθημάτων που σχετίζονται με γεγονότα και κανόνες. Υπάρχουν, ωστόσο, και αρκετοί επικριτές, οι οποίοι διαπιστώνουν ότι απουσιάζει από τη μαθησιακή διαδικασία η διαπροσωπική σχέση μαθητή – εκπαιδευτικού. Επίσης, παρατηρούν ότι η χρήση των μηχανών διεγείρει αρχικά το ενδιαφέρον των μαθητών, αλλά, σταδιακά, με τη συνεχή εφαρμογή τους, επέρχεται κορεσμός. Τέλος, δίνουν ιδιαίτερο βάρος στο γεγονός ότι οι μαθητές δεν εξαντλούν τις ικανότητές τους, διότι «ο καθένας εργάζεται χωριστά με τον δικό του ρυθμό» (Χαραλαμπίδης, 2001).

Μέσα από αυτόν τον μετασχηματισμό, η διδακτική πράξη αναδεικνύεται ως μια κυκλική διαδικασία, ένας κρίκος που συνδέει το ρόλο του δασκάλου και αυτόν του μαθητή: ο μὲν ρόλος του εκπαιδευτικού ως «διαχρονικού και αναντικατάστατου καταλύτη της μάθησης» (Ράπτης-Ράπτη, 2007), ο δε ρόλος του μαθητή κατά τη μετατροπή του από αντικείμενο της διδασκαλίας σε υποκείμενο της μάθησης. Στην πραγματικότητα, καμία θεμελιακή αντίφαση δεν ορίζεται ανάμεσα σε αυτές τις δύο ιδέες και πρακτικές. Αντίθετα, ο συγκερασμός τους σαν απάντηση σε ένα από τα πάγια και σημαντικά διλήμματα της εκπαίδευσης μπορεί να συντελεστεί μέσα από την επιλογή και αξιοποίηση των κατάλληλων επικοινωνιακών μέσων.

Μέσα από μια σειρά ανάλογων οργανωμένων ενεργειών, το προφίλ του εκπαιδευτικού συστήματος μπορεί να αλλάξει ριζικά, προς μια κατεύθυνση αποκέντρωσης και εκσυγχρονισμού. Στα πλαίσια ενός τέτοιου περιβάλλοντος διδασκαλίας, ο εκπαιδευτικός θα είναι σε θέση να αξιοποιήσει στο μέγιστο βαθμό τα οφέλη των νέων τεχνολογιών., προτάσσοντας παράλληλα ένα κριτικό λόγο απέναντι στα μειονεκτήματα και τους κινδύνους που μπορεί να προκύψουν από τη χρήση τους. Με αυτόν τον τρόπο, ο

υπολογιστής θα χρησιμοποιείται ως «μια προγραμματιζόμενη και όχι προγραμματισμένη μηχανή» (Μαλέτσκος, 2001)

Στο πλαίσιο μιας τέτοιας διδακτικής-μεθοδολογίας θα ήταν ιδανική η παρώθηση και η καθοδήγηση των μαθητών με τρόπο, ώστε μέσα από την άμεση επαφή τους με τα πράγματα και την κριτική αντιπαράθεσή τους με τα οπτικοακουστικά μηνύματα να εξερευνούν και να κατανοούν το περιβάλλον τους. Η αξιοποίηση των εν λόγω μέσων επιτελεί διττό ρόλο. Αφενός λειτουργούν ως φορείς πληροφοριών και μεταδότες μηνυμάτων και αφετέρου γίνονται η αφετηρία και τα εργαλεία για την επέκταση της εμπειρίας του μαθητή αποτελώντας μέσα για αυτοβελτίωση και αυτομόρφωση. Χωρίς να απαιτείται διαρκής καθοδήγηση, θα μπορεί μόνος του ή σε συνεργασία, να μαθαίνει να ανακαλύπτει τη γνώση ασκούμενος στη σωστή χρήση υλικών και συσκευών για να επινοήσει ή να κατασκευάσει κάτι δοκιμάζοντας τη χαρά της δημιουργίας, της προσωπικής επιτυχίας, της ομαδοσυνεργατικότητας αλλά και της ανάληψης πρωτοβουλιών, της αιτιολόγησης αποφάσεων και της αποδοχής ευθυνών για τις πράξεις του. Η κριτική αντιπαράθεση απέναντι στο πλήθος των εικόνων, των ήχων και των συμβόλων θα συντελεί στην ανάπτυξη αυτόνομης σκέψης, η ανάγκη για επιδέξιο χειρισμό απλών υλικών, οργάνων και συσκευών θα τον ωθεί στην τελειοποίηση των χειροτεχνικών του δεξιοτήτων, ενώ η διαπροσωπική επικοινωνία και ομαδική συνεργασία θα τον μαθαίνει να αποδέχεται το συνάνθρωπό του, να ακούει απροκατάληπτα τη διαφορετική άποψη και να συζητά εποικοδομητικά.



## **1.6 Ο προγραμματισμός ως δραστηριότητα μάθησης**

Στην επιστήμη των υπολογιστών, με τον όρο πρόγραμμα αναφερόμαστε σε μια συγκεκριμένη ακολουθία εντολών, τις οποίες πρέπει να εκτελέσει ένας υπολογιστής για να παραγάγει το επιθυμητό για τον χρήστη αποτέλεσμα. Η διαδικασία με την οποία προσπαθούμε να δώσουμε στον υπολογιστή τις κατάλληλες οδηγίες για την επίτευξη της επιθυμητής λύσης ονομάζεται προγραμματισμός. Ο προγραμματισμός αποτελεί μια ταχύτατα εξελισσόμενη επιστήμη, ενώ οι επαγγελματίες του κλάδου, μέσα από μακρόχρονη και εξειδικευμένη κατάρτιση έχουν συμβάλει κατά πολύ στη διαμόρφωση του τεχνολογικού κόσμου, όπως τον γνωρίζουμε σήμερα. Αν και η λειτουργία καθενός από τα υπολογιστικά περιβάλλοντα που είναι πλέον ευρέως διαδεδομένα θεωρείται ίσως αυτονόητη, η συνειδητοποίηση ότι κρύβουν πίσω τους ατελείωτες ακολουθίες εντολών και αλγορίθμων προκαλεί τον θαυμασμό.

### **1.6.1 Τα οφέλη από τη διδασκαλία του προγραμματισμού**

Τα προηγούμενα χρόνια, στα πλαίσια της εφαρμογής του τεχνοκρατικού μοντέλου ένταξης της τεχνολογίας στην εκπαίδευση, εισήχθη η διδασκαλία βασικών αρχών προγραμματισμού στο μάθημα της Πληροφορικής που διδάσκονται οι μαθητές του Ενιαίου Γενικού Λυκείου. Αυτή η μετατροπή του προγραμματισμού από εξειδικευμένο πεδίο επαγγελματικής κατάρτισης σε δραστηριότητα μάθησης εξυπηρετεί ποικίλους σκοπούς. Πιο συγκεκριμένα, οι μαθητές αναμένεται, μέσα από τη διδασκαλία του προγραμματισμού, να αναπτύξουν αναλυτική και συνθετική σκέψη, να αποκτήσουν ικανότητες μεθοδολογικού χαρακτήρα, να μπορούν να επιλύουν απλά προβλήματα σε προγραμματιστικό περιβάλλον, να αναπτύξουν δεξιότητες αλγοριθμικής προσέγγισης (ανάλυση προβλήματος, σχεδίαση αλγορίθμου, δομημένη σκέψη, αυστηρότητα έκφρασης) και να αναπτύξουν τη δημιουργικότητα και τη φαντασίας τους.

Ο προγραμματισμός ως γνωστική δραστηριότητα περιλαμβάνει, σύμφωνα με τον Τζιμογιάννη (2003), την πρόσκτηση και την εφαρμογή τριών αλληλοεξαρτώμενων μορφών γνώσης:

- τη συντακτική, που αποτελεί τη γνώση των ειδών χαρακτηριστικών μιας γλώσσας προγραμματισμού και των κανόνων χρήσης της,

- την εννοιολογική, που αφορά στην πλήρη κατανόηση των προγραμματιστικών δομών και αρχών και διακρίνεται περαιτέρω σε σημασιολογική και σχηματική και
- τη στρατηγική γνώση, που εξετάζει την ικανότητα εφαρμογής των συντακτικών και εννοιολογικών γνώσεων για την επίλυση αυθεντικών προβλημάτων στον προγραμματισμό.

Για καθεμιά από τις παραπάνω κατηγορίες, εκτός από το θεωρητικό κομμάτι το οποίο ο μαθητής καλείται να γνωρίζει και να κατανοεί, υπάρχει και ένα πακέτο δεξιοτήτων που αφορούν στο πρακτικό κομμάτι του προγραμματισμού και απαιτεί μια παιδαγωγική προσέγγιση που θα εστιάζει στην πρόσκτηση της λεγόμενης διαδικαστικής γνώσης.

### **1.6.2 Δυσκολίες των αρχάριων προγραμματιστών**

Η επιστήμη του Προγραμματισμού εμφανίζει έως έναν βαθμό συνάφεια με άλλες γνωστικές περιοχές, όπως είναι τα μαθηματικά και η φυσική, λόγω της αξιοποίησης των αλγοριθμικών στρατηγικών, τις οποίες έχουν ήδη διδαχθεί οι μαθητές στα πλαίσια αυτών των μαθημάτων. Παρόλα αυτά, δε θεωρείται σε καμία περίπτωση ότι αυτά τα γνωστικά κεκτημένα καθιστούν αυτονόητη και εύκολη διαδικασία τη διδασκαλία του Προγραμματισμού. Αντίθετα, στις περισσότερες βιβλιογραφικές αναφορές, παρουσιάζεται εκτενώς ένα σύνολο δυσκολιών, τις οποίες αντιμετωπίζει η πλειονότητα των αρχάριων προγραμματιστών. Οι δυσκολίες αυτές κωδικοποιούνται από τους Τζιμογιάννη και Κόμη (2004) ως εξής:

- δυσκολία των αρχάριων προγραμματιστών να κατανοήσουν τη χρήση και λειτουργία των δομικών στοιχείων της γλώσσας για την επίλυση ενός προβλήματος
- δυσκολία των αρχάριων προγραμματιστών στη διαφοροποίηση του τρόπου σκέψης σχετικά με δεδομένα και προβλήματα σε αντιδιαστολή με άλλους γνωστικούς τομείς
- δυσκολία των αρχάριων προγραμματιστών στην προσαρμογή τους στην αυστηρότητα της σύνταξης και δόμησης ενός προγράμματος, η οποία

απαιτείται για την αποτελεσματική χρήση ενός προγραμματιστικού περιβάλλοντος με στόχο την επίλυση προβλημάτων

- δυσκολία των αρχάριων προγραμματιστών στην κατανόηση της δομής του προγράμματος και της βηματικής φύσης του αλγορίθμου, οι οποίες παρεμποδίζουν την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων σε προγραμματιστικό περιβάλλον
- δυσκολία των αρχάριων προγραμματιστών στην κατανόηση των απρόσιτων μηχανισμών που κρύβονται πίσω από την εκτέλεση ενός προγράμματος στον υπολογιστή, λόγω της έλλειψη επαρκών αναπαραστάσεων για τη ροή των δεδομένων, τον ρόλο και τη λειτουργία των βασικών μονάδων του υπολογιστή, ο οποίος γίνεται αντιληπτός ως μαύρο κουτί (black box)
- δυσκολία των αρχάριων προγραμματιστών στην ανάπτυξη αφαιρετικού συλλογισμού, τη σύνθεση νέων αλγορίθμων και την προσαρμογή εντολών ή υπολογιστικών διαδικασιών πάνω σε ένα νέο πρόβλημα με την αποτελεσματική αξιοποίηση των προηγούμενων γνώσεών τους
- δυσκολία των αρχάριων προγραμματιστών λόγω της φύσης των συνηθισμένων προγραμματιστικών περιβαλλόντων, τα οποία έχουν σχεδιαστεί για την ανάπτυξη εφαρμογών και όχι για τη διδασκαλία και, επομένως, προορίζονται για χρήση από έμπειρους προγραμματιστές
- δυσκολία των αρχάριων προγραμματιστών στη διάκριση του γλωσσικού κώδικα των προγραμματιστικών περιβαλλόντων, στον οποίο αποδίδονται συχνά λανθασμένα ανθρωπομορφικά χαρακτηριστικά.

### **1.6.3 Διδακτικές προσεγγίσεις του προγραμματισμού**

Παραδοσιακά, η διδασκαλία του Προγραμματισμού στα πλαίσια του μαθήματος της Πληροφορικής, επικεντρωνόταν στην εκμάθηση συγκεκριμένων προγραμματιστικών γλωσσών. Αυτή η παραδοσιακή διδακτική προσέγγιση θα μπορούσε να χαρακτηριστεί ως εμπειρική. Το επικοινωνιακό σχήμα μεταξύ διδάσκοντος και μαθητή που αναπτύσσεται στα πλαίσια μιας παραδοσιακής διδασκαλίας του συμπεριφοριστικού μοντέλου είναι κλειστό και δασκαλοκεντρικό.

Ο διδάσκων αναδεικνύεται σε πρωταγωνιστή της διαδικασίας, καθώς αποτελεί την πηγή όλων των πληροφοριών και γνώσεων, τις οποίες καλείται να μεταδώσει στους μαθητές. Ο μαθητής από την πλευρά του δέχεται παθητικά τις προσφερόμενες από τον παντογνώστη εκπαιδευτικό πληροφορίες. Στη συνέχεια, τις απομνημονεύει και εξασκείται στη χρήση του στα πλαίσια ενός απόλυτα ελεγχόμενου και τυποποιημένου περιβάλλοντος. Σε αυτό το μοντέλο, η αποτελεσματικότητα του εκπαιδευτικού εξαρτάται από την ποσότητα των γνώσεων που διαθέτει πάνω στο αντικείμενό του. Ωστόσο, όπως επισημαίνεται από τους Major, Kyriacou et Brereton (2012), η αποτελεσματική διδασκαλία του προγραμματισμού προϋποθέτει, πέρα από τη λεπτομερή και εις βάθος γνώση της επιστήμης της πληροφορικής, και τη γνώση των βασικών αρχών της παιδαγωγικής επιστήμης, ώστε να ανταποκρίνεται στις σύγχρονες απαιτήσεις της εκπαίδευσης, αλλά και να επιτυγχάνει τη μάθηση στον μέγιστο δυνατό βαθμό.

Στα πλαίσια της αλλαγής των διδακτικών προσεγγίσεων που εφαρμόζονται ευρύτερα στην εκπαίδευση τις τελευταίες δεκαετίες, επιδιώχθηκε η τροποποίηση των παραδοσιακών μεθόδων αυτού του εμπειρικού – τεχνοκρατικού μοντέλου, με σκοπό την υπέρβαση των αδιεξόδων που εμπεριείχαν. Σύμφωνα με τους Jerinic, Ivanovic, Putnik, Budimac et Savic (2014), εφόσον δεχτούμε ότι κάθε μαθητής αποτελεί ένα ευφυές, αυτόνομο και σκεπτόμενο άτομο με δικό του τρόπο επεξεργασίας της μάθησης, τότε θα συνειδητοποιήσουμε πως υπάρχουν τόσο διαφορετικοί και απρόβλεπτοι τρόποι μάθησης, όσοι είναι και οι μαθητές. Επομένως, απαιτείται ο εκ νέου σχεδιασμός των διδακτικών προσεγγίσεων που επιχειρούνται στη διδασκαλία όλων των γνωστικών αντικειμένων και, φυσικά, του Προγραμματισμού. Στα πλαίσια αυτών των προσπαθειών, αναδύεται η ιδέα της ανακαλυπτικής προσέγγισης του Προγραμματισμού (programming by discovery). Σύμφωνα με αυτό το μοντέλο διδασκαλίας, βασικός στόχος είναι να μετατραπεί ο μαθητής – αρχάριος προγραμματιστής σε ενεργό υποκείμενο της μάθησης και να οδηγηθούν στην κατανόηση των δυναμικών συμπεριφορών των εκάστοτε προγραμμάτων μέσα από διαδικασίες δοκιμής και άμεσης παρατήρησης των αποτελεσμάτων στην οθόνη του υπολογιστή. Με αυτόν τον τρόπο, μέσα από την αυτόνομη εξερεύνηση του προγραμματιστικού περιβάλλοντος, ανιχνεύουν πιο

εύκολα τα σφάλματα και τις παρανοήσεις τους. Σε αυτή τη μορφή διδασκαλίας, οι ρόλοι διδάσκοντος και μαθάνοντος διαφοροποιούνται ριζικά σε σχέση με προηγούμενες προσεγγίσεις. Πιο συγκεκριμένα, ο διδάσκων παρουσιάζεται ως διευκολυντής των μαθησιακών δραστηριοτήτων, παρέχοντας τόσο τα κατάλληλα εργαλεία, όσο και τις απαιτούμενες συνθήκες για την ανακάλυψη των βασικών χαρακτηριστικών των αλγορίθμων και των άλλων θεμελιωδών εννοιών του προγραμματισμού. Αντίστοιχα, ο ρόλος του μαθητή αναβαθμίζεται και τον καθιστά από παθητικό δέκτη των αέριων πληροφοριών που του παρείχε άλλοτε ο εκπαιδευτής σε ενεργό υποκείμενο της μάθησης με δυνατότητες πειραματισμού και δημιουργίας. Σε αυτό το νέο σχήμα, ο μαθητής ανακαλύπτει έννοιες και συσχετίσεις, αναπτύσσει προγράμματα, ελέγχει και διορθώνει τα ίδια του τα λάθη και σχεδιάζει λύσεις και επιχειρήματα. Μέσα από αυτήν τη διαδικασία, επιτυγχάνεται πέρα από τον αντικειμενικό στόχο της εξοικείωσης με τη θεωρία και πράξη του προγραμματισμού και ένας ανώτερος παιδαγωγικός στόχος της εξοικείωσης με μεθοδολογίες και στρατηγικές επίλυσης προβλημάτων που θα αξιοποιηθούν σε ποικίλες περιστάσεις, εντός και εκτός μαθητικής ζωής.

Ένα ακόμη μοντέλο που εφαρμόστηκε για τη διδασκαλία του προγραμματισμού είναι το εποικοδομιστικό. Σε αυτή τη μέθοδο, γίνεται προσπάθεια να αξιοποιηθούν οι αρχές των κονστрукτιβιστικών θεωριών μάθησης, με βάση τις οποίες η γνώση δε μεταδίδεται, αλλά οικοδομείται και αναδομείται ατομικά από τον μαθητή, ο οποίος αναλαμβάνει στον μεγαλύτερο βαθμό την υπευθυνότητα για τη μάθησή του. Ο σχεδιασμός μιας διδασκαλίας με βάση αυτό το μοντέλο προϋποθέτει τη μελέτη των διαδικασιών μέσα από τις οποίες οικοδομείται η γνώση και επιτυγχάνεται η μάθηση. Επομένως, είναι προφανές ότι το κέντρο βάρους αυτού του μοντέλου μετατοπίζεται και πάλι, όπως και στο ανακαλυπτικό, στις ανάγκες και τις ενέργειες του μαθητή. Ο διδάσκων συντονίζει και κατευθύνει τις μαθητικές δραστηριότητες, λαμβάνοντας υπόψη τις γνωστικές δυσκολίες και τους βασικούς μηχανισμούς με τους οποίους είναι δυνατόν αυτές να ξεπεραστούν. Με αυτήν την έννοια, ο ρόλος του θεωρείται πιο σημαντικός από αυτόν που προτείνεται στο ανακαλυπτικό μοντέλο, καθώς οι παιδαγωγικές του γνώσεις τον καθιστούν σημαντική συνιστώσα της μάθησης. Ο ρόλος του μαθητή διαμορφώνεται στο ίδιο ενεργό, συμμετοχικό

πλαίσιο με έμφαση στην καλλιέργεια μεταγνωστικών δεξιοτήτων, αυτών δηλαδή που του επιτρέπουν να γνωρίζει τους ίδιους τους μηχανισμούς που χρησιμοποιεί για να οικοδομήσει τη γνώση. Με άλλα λόγια, σε συνεργασία με συμμαθητές και διδάσκοντα, ο μαθητής μαθαίνει πώς να μαθαίνει (τεχνικές ανάλυσης, σύνθεσης και αξιολόγησης) και οικοδομεί ένα ικανό ρεπερτόριο ρουτινών που θα αξιοποιηθεί τόσο εντός των πλαισίων του μαθήματος, όσο και έξω από αυτά.

Σύμφωνα με τον Τζιμογιάννη (2008), κατά τον σχεδιασμό των εποικοδομηστικών μαθησιακών δραστηριοτήτων, επιδιώκεται η αφομοίωση στοιχείων και αρχών της διερευνητικής και της συνεργατικής μάθησης. Στόχος της ενσωμάτωσης αυτών των νέων θεωριών παραμένει η μετατόπιση από τη διδασκαλία στην ενεργητική μάθηση μέσω δραστηριοτήτων και από την εκμάθηση του συντακτικού ορισμένων γλωσσών στην καλλιέργεια δεξιοτήτων, όπως η αναλυτική σκέψη, αφαιρετική ικανότητα, μοντελοποίηση λύσεων, οι οποίες θα εφαρμοστούν από τον μαθητή για την επίλυση προβλημάτων ευρύτερα. Πιο αναλυτικά, η διερεύνηση μπορεί να οριστεί ως η συνειδητή διαδικασία διάγνωσης προβλημάτων, κριτικής θεώρησης και διάκρισης εναλλακτικών λύσεων, σχεδιασμού και διερεύνησης εικασιών, αναζήτησης πληροφοριών, κατασκευής μοντέλων και διατύπωσης λογικών επιχειρημάτων. Στόχος αυτού του νέου μοντέλου είναι η βελτίωση της μαθησιακής διαδικασίας και η υπερκάλυψη των εμποδίων που προέκυπταν κατά την εφαρμογή των προηγούμενων διδακτικών προσεγγίσεων. Η διδασκαλία ακολουθεί μια κυκλική διαδικασία, που αποτελείται από πέντε διακριτά, αλλά επαναλαμβανόμενα στάδια (εκμείωση ερωτήσεων, ενεργός έρευνα, δημιουργία, συζήτηση, αξιολόγηση). Η σύνδεση των δύο μοντέλων – διερευνητικού και συνεργατικού- στη βιβλιογραφία βασίζεται στο γεγονός ότι η δομή της διερευνητικής διαδικασίας ανάγει σε πρωτεύον συστατικό τη συζήτηση με τους συνομηλίκους.

## ΚΕΦΑΛΑΙΟ 2

### 2 ΕΚΠΑΙΔΕΥΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

#### 2.1 Εισαγωγή

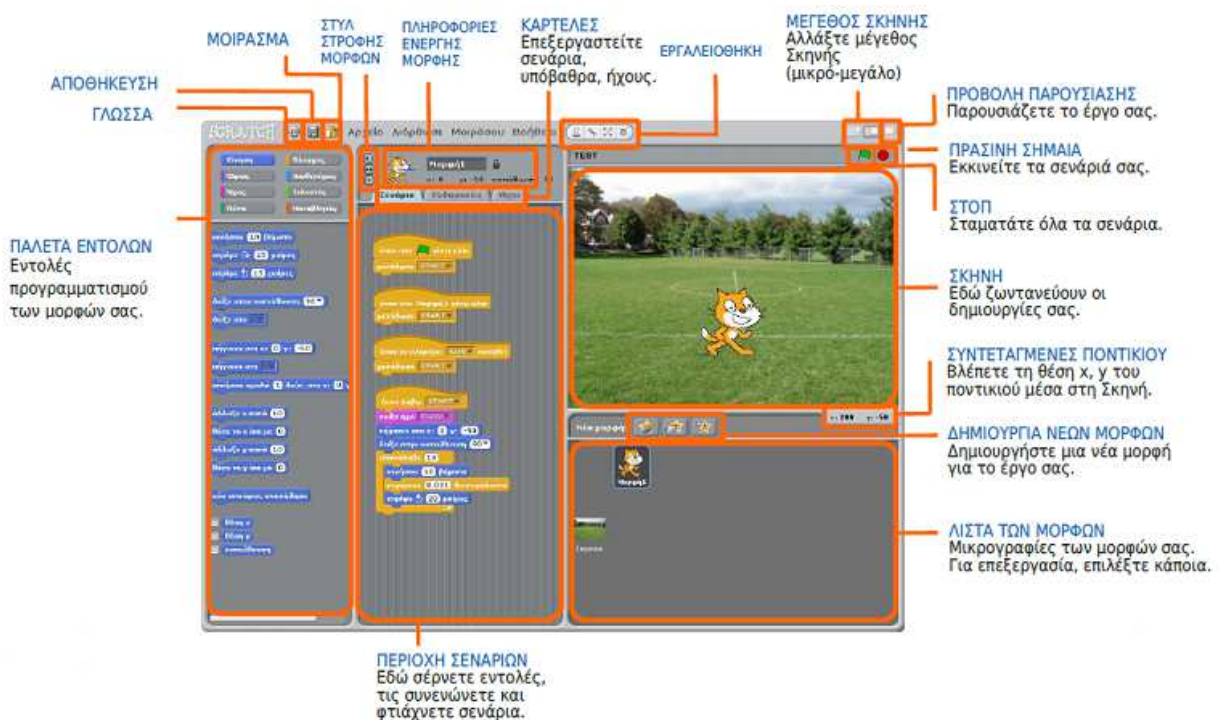
Λόγω της τόσο αυξημένης ανάγκης για εκπαίδευση με τη χρήση του υπολογιστή συνδυασμένη και με τις δυνατότητες που παρέχει το ίντερνετ, ο κάθε εκπαιδευόμενος έχει πλέον στη διάθεση του πολλά προγραμματιστικά περιβάλλοντα που του επιτρέπουν να αναπτύξει τα δικά του λογισμικά χωρίς να γνωρίζει κάποια συγκεκριμένη γλώσσα προγραμματισμού. Οι δυνατότητες των περιβαλλόντων αυτών είναι τόσο μεγάλες όσο και τα ονόματα των οργανισμών, πανεπιστημίων και εταιρειών-κολοσσών που τα έχουν αναπτύξει. (Το scratch αναπτύσσεται στο μεγαλύτερο πολυτεχνικό πανεπιστήμιο των Η.Π.Α., το Μ.Ι.Τ., το App Inventor εντάσσεται στη Google ενώ το Kodu στη Microsoft.). Μια επίσης μεγάλη ανάγκη της εποχής είναι η ανάπτυξη εφαρμογών και παιχνιδιών για τα κινητά τηλέφωνα και τις ταμπλέτες. Το γεγονός αυτό έχει προσφέρει και σε αυτόν τον τομέα αρκετά περιβάλλοντα που μας επιτρέπουν να δημιουργήσουμε τις δικές μας εφαρμογές απευθείας μέσα σε αυτά αλλά και από τον υπολογιστή, με απώτερο αποτέλεσμα πάντα την επαφή μας με τον κόσμο του προγραμματισμού. Θα παρουσιάσουμε παρακάτω μερικά από αυτά τα περιβάλλοντα ενώ θα χρησιμοποιήσουμε το App Inventor για την μελέτη περίπτωσης μας.

#### 2.2 Scratch

Το Scratch αναπτύχθηκε από το Lifelong Kindergarten group στο MIT με επικεφαλή τον Mitchel Resnick και πρωτοεμφανίστηκε το καλοκαίρι του 2007 (Resnick et al., 2009). Το λογισμικό διανέμεται δωρεάν για διαφορετικά λειτουργικά συστήματα (Windows, Mac OS X ή Linux) και η εγκατάστασή του είναι πολύ απλή. Σήμερα χρησιμοποιείται ευρέως για τη διδασκαλία του προγραμματισμού, ενώ η διάδοσή του είναι ταχύτατη. Ενδεικτικά μπορούμε να αναφέρουμε ότι στην ιστοσελίδα του Scratch (<http://scratch.mit.edu/>) υπάρχουν γύρω στα 700.000 εγγεγραμμένα μέλη και γύρω στους 200.000 προγραμματιστές που δημοσιεύουν τα προγράμματά τους, που ξεπερνούν τα 8.000.000!

Ενδιαφέρον είναι ότι το Scratch πήρε το όνομά του από την τεχνική των DJ's (scratching). Το βασικό χαρακτηριστικό της τεχνικής των DJ's είναι η επαναχρησιμοποίηση των μουσικών κομματιών. Αντίστοιχα στο Scratch όλα τα αντικείμενα, γραφικά, ήχοι, και κείμενα μπορούν εύκολα να εισαχθούν σε ένα νέο πρόγραμμα και να συνδυαστούν με ποικίλους τρόπους για την παραγωγή ενός προγράμματος, κάτι το οποίο δίνει κίνητρο για περαιτέρω ενασχόληση με αυτό.

Το Scratch είναι μία νέα γλώσσα προγραμματισμού με την οποία μπορούμε να φτιάχνουμε τις δικές μας διαδραστικές ιστορίες, τα δικά μας παιχνίδια εύκολα και γρήγορα. Αυτό επιτυγχάνεται αφενός με τον οπτικό προγραμματισμό, που μας δίνει τις εντολές σε ψηφίδες που συνδυάζονται μόνο σωστά συντακτικά, και αφετέρου με το περιβάλλον του Scratch (εικόνα 2) όπου ο χειρισμός ήχων και εικόνων γίνεται χαρακτηριστικά εύκολος (Νικολός & Κόμης, 2010). Τέλος, το Scratch μας δίνει τη δυνατότητα να εξετάζουμε πολύ γρήγορα και εύκολα τα αποτελέσματα οποιασδήποτε εντολής.



ΕΙΚΟΝΑ 2.1 - Το ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ SCRATCH

Καθώς το Scratch γιγαντώνεται στο χώρο, ένας άκρως ενδιαφέρον συνδυασμός έχει προκύψει με το Scratch και τον αισθητήρα της Microsoft, Kinect. Στο ονομαζόμενο



**Kinect2Scratch** μπορούμε να προγραμματίσουμε το Kinect με τη βοήθεια του Scratch, γεγονός που μετατρέπει τον προγραμματισμό ενός τόσο εξελιγμένου αισθητήρα σε πολύ απλή διαδικασία. Έτσι μπορούμε να ελέγξουμε την εφαρμογή που έχουμε αναπτύξει με το Scratch χρησιμοποιώντας όχι με το ποντίκι του υπολογιστή ή με κάποια οθόνη αφής, αλλά με τις κινήσεις του σώματός μας.

### 2.3 Snap(παλιότερα γνωστό ως byob)

Το Snap είναι μια οπτική, “drag and drop” γλώσσα προγραμματισμού, που ουσιαστικά επεκτείνει το Scratch, δίνοντας τη δυνατότητα να δημιουργήσουμε τα δικά μας μπλοκ εντολών (BYOB). Αναπτύχθηκε από τον Jens Mönig της MioSoft Corporation με συνδρομή και του πανεπιστημίου του Berkley. Τρέχει στον περιηγητή ιστού και χρησιμοποιεί τη γλώσσα προγραμματισμού Javascript, που μας επιτρέπει να δουλεύουμε με ασφάλεια σε οποιοδήποτε project στο διαδίκτυο.

### 2.4 Alice 3

Το Alice είναι ένα καινοτόμο περιβάλλον τρισδιάστατου προγραμματισμού (Εικόνα 2.2) που κάνει τη δημιουργία ενός παιχνιδιού, μιας ιστορίας κινουμένων σχεδίων ή ενός βίντεο αρκετά εύκολη και κυρίως χωρίς να απαιτεί εμπειρία στον προγραμματισμό. Σε αντίθεση μάλιστα, λειτουργεί ως εργαλείο διδακτικής του προγραμματισμού καθώς ο χρήστης έρχεται σε επαφή με τον αντικειμενοστραφή προγραμματισμό και τις βασικές αρχές του προγραμματισμού γενικά.

Διαδραστική του προγραμματισμού – Μια μελέτη περίπτωσης με χρήση του App Inventor

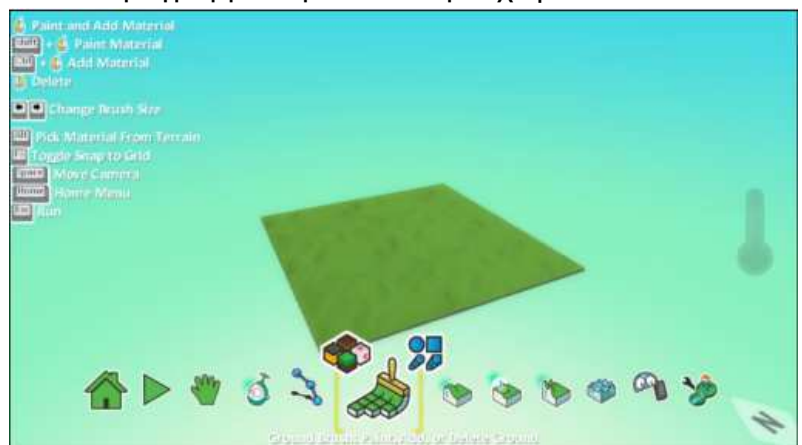
Ακολουθεί και αυτό τη λογική “drag and drop” για τη δημιουργία των προγραμμάτων, που μέσα από αυτά ο χρήστης μπαίνει στη νοοτροπία πολλών βασικών γλωσσών προγραμματισμού, όπως η Java, η C++ και η C#, αφού κάθε αντικείμενο έχει ιδιότητες και μεθόδους (Cooper et al., 2000)



ΕΙΚΟΝΑ 2.2 - ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ ALICE

## 2.5 Kodu

Το Kodu είναι μία απλή γλώσσα οπτικού προγραμματισμού που φτιάχτηκε ώστε να επιτρέψει σε παιδιά, κάθε ηλικίας, να δημιουργήσουν τα δικά τους παιχνίδια για τον υπολογιστή αλλά και για τη παιχνιδομηχανή Xbox. Το ότι είναι φτιαγμένη για παιδιά προδίδει το γεγονός ότι ο εκάστοτε προγραμματιστής δε απαιτείται να έχει γνώσεις προγραμματισμού.



ΕΙΚΟΝΑ 2.3 – ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ KODU

Δημιουργήθηκε το 2009 και έκτοτε έχει αποτελέσει σημείο αναφοράς σε πολλές έρευνες ακαδημαϊκού επιπέδου, έχει τιμηθεί από τον Λευκό Οίκο και έχουν γραφτεί βιβλία για αυτό. Δύο χρόνια μόλις μετά τη δημιουργία του, χαρακτηρίστηκε σαν επιτυχία στο Xbox

Live και χρησιμοποιείται σε 60 και πλέον εκπαιδευτικά ιδρύματα παγκοσμίως για την εισαγωγή των εκπαιδευμένων στον προγραμματισμό.



ΕΙΚΟΝΑ 2.4 – ΜΟΡΦΗ ΕΝΤΟΛΩΝ ΤΟΥ KODU

## 2.6 Stencyl

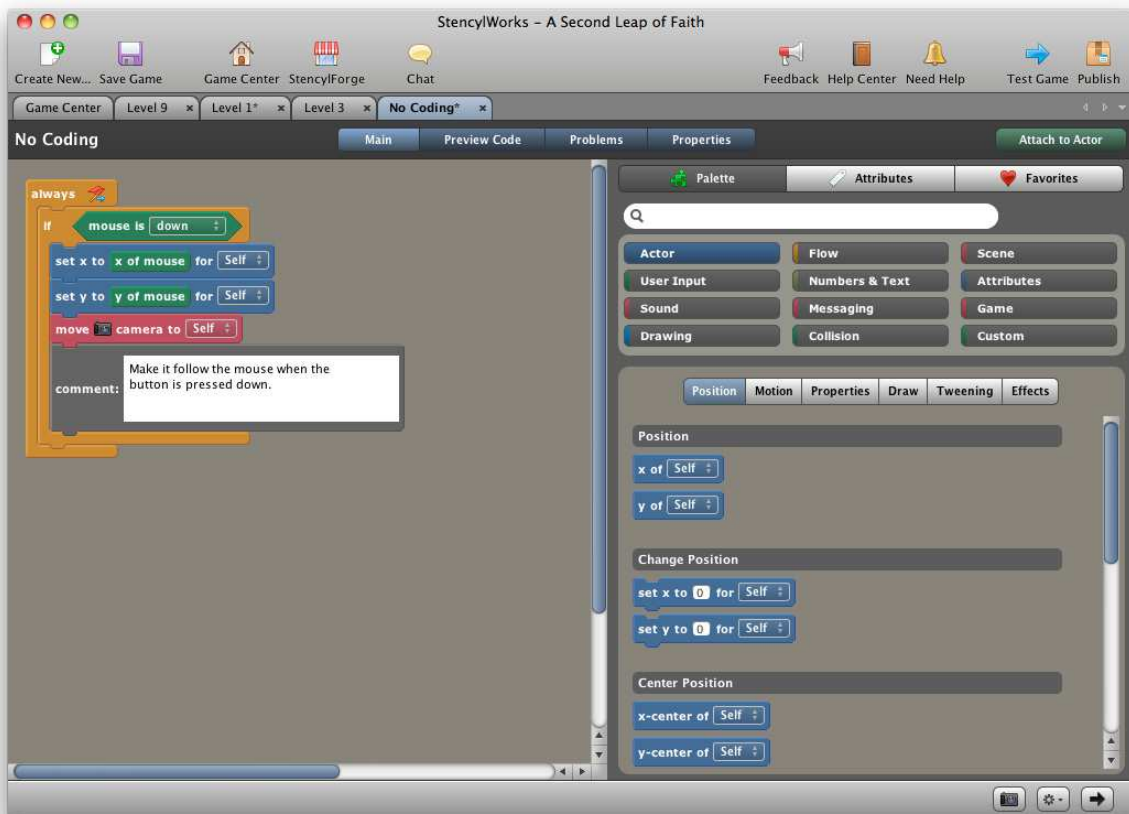
Οι δημιουργοί του Stencyl κατάφεραν, όπως ισχυρίζονται, να δημιουργήσουν ένα περιβάλλον που μας επιτρέπει να δημιουργήσουμε με τον γρηγορότερο και τον πιο εύκολο τρόπο, μιας και είναι στηριγμένο στο Scratch, τα δικά μας παιχνίδια. Αλλά αυτό που κάνει το Stencyl πραγματικά πρωτοπόρο και μοναδικό είναι ότι τα παιχνίδια που μπορούμε να δημιουργήσουμε με αυτό, μπορούμε να τα απολαύσουμε σε οποιαδήποτε υπολογιστικό περιβάλλον, δηλαδή στον υπολογιστή με λειτουργικό σύστημα Windows, Linux αλλά και με Mac, σε οποιοδήποτε κινητό τηλέφωνο ή ταμπλέτα, είτε iOS είτε Android (εικόνα 6) και τέλος και με χρήση της τεχνολογίας Flash.

Ο προγραμματισμός και εδώ επιτυγχάνεται με τα ιδιαίτερα εύχρηστα μπλοκ εντολών που δίνουν στον χρήστη την ευκαιρία να μη χρονοτριβεί με ότι έχει να κάνει με το συντακτικό ή την σωστή χρήση των εντολών και να ρίξει το βάρος του στο κατά πόσο θα κάνει πιο ενδιαφέρον το παιχνίδι του. Στο Stencyl ωστόσο οι πιο έμπειροι προγραμματιστές μπορούν να εντάξουν κώδικα, να εισάγουν βιβλιοθήκες και να γράψουν τις δικές τους κλάσεις, και όλα αυτά σε απόλυτη συμφωνία με τα μπλοκ εντολών.

Οι τόσες δυνατότητες του Stencyl έχουν καταφέρει να το βγάλουν από τον αυστηρά εκπαιδευτικό χαρακτήρα, αφού τα παιχνίδια που δημιουργούνται είναι διαθέσιμα στα μεγάλα καταστήματα εφαρμογών και οι δημιουργοί τους μπορούν από αυτά να αποκομίσουν κέρδη.



ΕΙΚΟΝΑ 2.5 - ΠΑΙΧΝΙΔΙΑ ΦΤΙΑΓΜΕΝΑ ΜΕ ΤΟ STENCYL

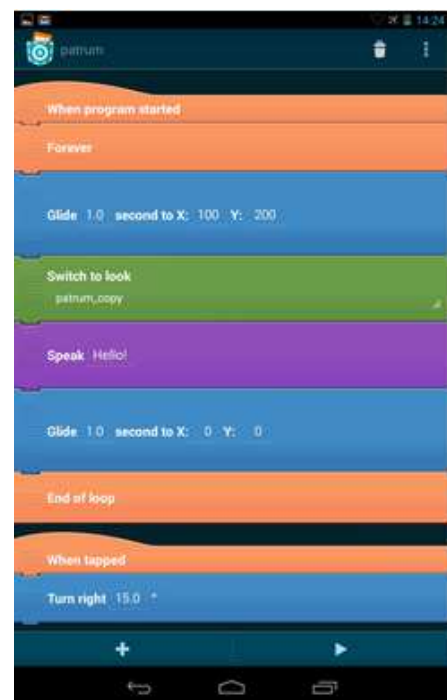


ΕΙΚΟΝΑ 2.6 - ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ STENCYL

## 2.7 Pocket code

Με το Pocket Code, της Catrobat, ο χρήστης μπορεί να δημιουργήσει τα δικά του παιχνίδια, κινούμενα σχέδια, διαδραστικά βίντεο μουσικής και οποιαδήποτε άλλη εφαρμογή απευθείας από το κινητό του ή την ταμπλέτα που χρησιμοποιεί το λειτουργικό σύστημα Android.

Όπως η πλειοψηφία των περιβαλλόντων που δεν χρειάζονται γνώσεις προγραμματισμού, και προορίζονται κυρίως για εκπαιδευτικούς σκοπούς, έτσι και το Pocket Code χρησιμοποιεί τις ψηφίδες εντολών, με τα προαναφερθέντα πλεονεκτήματα.



ΕΙΚΟΝΑ 2.7 - ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΤΟ POCKET CODE

## ΚΕΦΑΛΑΙΟ 3

### **3. Μελέτη περίπτωσης της διδακτικής του προγραμματισμού με χρήση του app inventor**

#### **3.1 Εισαγωγή**

Δεδομένου του ότι αναφερθήκαμε στο πόσο απαραίτητος έχει γίνει ο προγραμματισμός στις μέρες, στα προβλήματα της διδακτικής του προγραμματισμού αλλά και παρουσιάσαμε τα διάφορα περιβάλλοντα που προσφέρονται στους εκπαιδευτικούς και στους εκπαιδευόμενους για να διδάξουν και να διδαχτούν, αντίστοιχα, τον προγραμματισμό, φτάσαμε στο σημείο να παρουσιάσουμε το App Inventor. Το περιβάλλον με το οποίο θα αναπτυχθούν τα παραδείγματα.

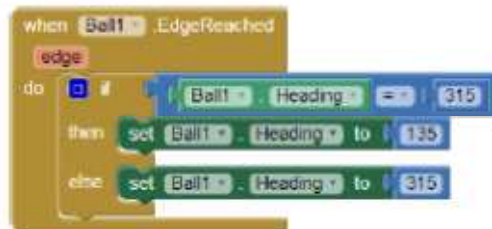
Το App Inventor είναι ένα νέο σύστημα της Google που μας επιτρέπει να σχεδιάσουμε και να προγραμματίσουμε εφαρμογές για Android συσκευές, σε μια σελίδα του περιηγητή μας και μια διεπαφή γραμμένη στη γλώσσα προγραμματισμού Java. Η δημιουργία των εφαρμογών μπορεί να επιτευχθεί από όλους αφού δεν απαιτείται προγραμματιστική εμπειρία ή γνώση. Εναντιθέσει, μάλιστα, με το App Inventor κάποιος ξεκινώντας από πολύ βασικό επίπεδο, μπορεί μετά από την συνεχή τριβή να συνθέσει πολύπλοκες και πολυμήχανες εφαρμογές.

Με τη χρήση των «έξυπνων» συσκευών και των εφαρμογών που φέρουν αυτές είμαστε όλοι αρκετά εξοικειωμένοι. Όπως με όλα τα πράγματα που εντάσσονται στην καθημερινότητα μας, έτσι και σε αυτήν την περίπτωση, μπορεί όλοι λίγο πολύ να έχουμε σκεφτεί μια εφαρμογή που ίσως να μην υπάρχει ή να είναι έστω μια βελτιωμένη έκδοση μιας προϋπάρχουσας. Με το App Inventor υπάρχει η δυνατότητα να τη δημιουργήσουμε. Και αυτό γίνεται εύκολο χάρις στη μέθοδο προγραμματισμού με τα μπλοκ, την οποία χρησιμοποιεί και το App Inventor, που έχει αποδειχθεί η πιο επιτυχημένη μέθοδος ακόμα και για εκπαιδευόμενους πολύ μικρής ηλικίας.

#### **3.2 Τι είναι το app inventor**

Το App Inventor είναι, λοιπόν, ένα οπτικό, «σύρε-και-άσε» εργαλείο δημιουργίας εφαρμογών στην πλατφόρμα των Android. Σχεδιάζουμε την διεπαφή του χρήστη μιας

εφαρμογής, χρησιμοποιώντας ένα γραφικό περιβάλλον με βάση του το διαδίκτυο, έπειτα διέπουμε την συμπεριφορά της διεπαφής αυτής χρησιμοποιώντας τα μπλοκ εντολών, σαν να δημιουργούμε ένα πάζλ.



ΕΙΚΟΝΑ 3.2 - ΠΑΡΑΔΕΙΓΜΑ ΚΩΔΙΚΑ APP INVENTOR

Και η χρήση της λέξης πάζλ είναι αυτή ακριβώς που περιγράφει το πώς λειτουργούν τα μπλοκ εντολών. Όπως στα πάζλ, συγκεκριμένα κομμάτια κολλάνε μόνο μεταξύ τους, έτσι και με τα μπλοκ εντολών, για να μην υπάρχουν προβλήματα σύνταξης, κολλάνε μεταξύ τους μόνο όταν βγάζει νόημα και δημιουργείται μια ακολουθία εντολών με σκοπό την επίτευξη των συμπεριφορών που θέλουμε να αποκτήσουν τα διάφορα αντικείμενα που έχουμε εισάγει στην διεπαφή του χρήστη.

Επειδή τα μπλοκ γίνονται πιο εύκολα κατανοητά από τον παραδοσιακό τρόπο προγραμματισμού με κώδικα, ο χρήστης εισάγεται πιο γρήγορα στη φιλοσοφία του App Inventor και του προγραμματισμού γενικότερα.

### 3.3 Γιατί χρησιμοποιούμε το app inventor

Όπως αναφέρθηκε και παραπάνω, το App Inventor είναι ένας εύκολος τρόπος προγραμματισμού, λόγω της οπτικής, «σύρε-και-άσε» διεπαφής του. Ας εξηγήσουμε όμως γιατί είναι πραγματικά τόσο εύκολο να προγραμματίσουμε με το App Inventor.

- **Δεν χρειάζεται ο χρήστης να απομνημονεύει και να γράφει τις εντολές.**

Ένα από τα μεγαλύτερα προβλήματα που αντιμετωπίζει κανείς όταν κάνει τα πρώτα του βήματα στον προγραμματισμό είναι ότι πρέπει να γράψει κώδικα, όπως γράφει ένα κείμενο. Αυτό συνεπάγεται ότι πρέπει να γνωρίζει αρχικά τις εντολές, αλλά και την ορθογραφία των εντολών αυτών, το σωστό συντακτικό της γλώσσας και τις διάφορες «ιδιοτροπίες» της. Αλλά δε φτάνει μόνο να τα γνωρίζει, θα πρέπει να τα γράψει και σωστά. Το γεγονός αυτό έχει ως αποτέλεσμα οι

πρώτες, αλλά και όχι μόνο, προσπάθειες να τρέξουμε ένα πρόγραμμα να επιστρέφουν ένα σωρό λάθη, το οποίο μας βάζει να λύνουμε αυτά τα λογικά ή πρακτικά προβλήματα πριν καν απολύσουμε αυτό που έχουμε προσπαθήσει να δημιουργήσουμε.

- **Οι εντολές είναι αρχειοθετημένες και οργανωμένες σε σετ με κριτήριο την εφαρμογή και τον σκοπό τους.**

Στη διεπαφή του App Inventor, τα στοιχεία και τα μπλοκ είναι οργανωμένα σε «συρτάρια» σεσημασμένα κατάλληλα ώστε να μπορεί ο χρήστης να βρίσκει εύκολα και γρήγορα τον τύπο της εντολής που θέλει να χρησιμοποιήσει. Έτσι, για να προγραμματίσει ο χρήστης αρκεί να βρει το μπλοκ εντολών που ψάχνει μέσα από τις διάφορες κατηγορίες εντολών, που τον βοηθά να συνειδητοποιήσει και την λειτουργία της εντολής και της εφαρμογής γενικότερα, και να το «σύρει» απλά στο πρόγραμμα. Δεν χρειάζεται να απομνημονεύσει τις οδηγίες της σύνταξης των εντολών ή να ανατρέξει για τους κανόνες σε κάποιο εγχειρίδιο ή στο διαδίκτυο.

- **Συγκεκριμένα μπλοκ μπορούν να ενωθούν μόνο με συγκεκριμένα μπλοκ**

Το App Inventor αποτρέπει τον χρήστη να πέσει σε λάθη συντακτικά από την πρώτη στιγμή με τους περιορισμούς που έχει η γλώσσα με τη χρήση των μπλοκ. Για παράδειγμα, αν ένα μπλοκ εντολών περιμένει σαν είσοδο του έναν αριθμό και όχι ένα χαρακτήρα, δεν επιτρέπει στο χρήστη να «κολλήσει» μπλοκ που μπορεί να επιτρέψει την εισαγωγή χαρακτήρα. Με αυτόν τον τρόπο αποφεύγει, αν όχι όλα, τα περισσότερα λάθη, που σε προγραμματισμό με χρήση κώδικα θα επέστρεφαν πολλές σειρές από κρυπτογραφημένα μηνύματα λαθών.

- **Χειριζόμαστε τα γεγονότα πιο εύκολα**

Οι περισσότερες «παραδοσιακές» γλώσσες προγραμματισμού δημιουργήθηκαν σε μια εποχή που τα γεγονότα που έπρεπε να χειριστούν, ήταν είτε προκαθορισμένα είτε εύκολο να προβλεφθεί ο χρόνος που θα έπρεπε το πρόγραμμα να τα διαχειριστεί. Στη σύγχρονη εποχή και με τη ανάγκη για προγραμματισμό εφαρμογών για κινητά τηλέφωνα, που πρέπει διαχειριστούν γεγονότα όπως μια κλήση ή ένα γραπτό μήνυμα, τα προγράμματα πρέπει να έχουν διαχειριστή γεγονότων (event handler). Ο διαχειριστής αυτός είναι ο

τρόπος που λέμε στην εφαρμογή μας, «όταν συμβεί αυτό, κάνε το άλλο», και σε γλώσσες όπως η Java είναι μια δομή που απαιτεί αφενός να έχουμε κατανοήσει απόλυτα τον συντακτικό και τις εντολές της γλώσσας και αφετέρου έννοιες όπως κλάσεις, αντικείμενα και ειδικά αντικείμενα που ονομάζονται «ακροατές» (listeners). Στο App Inventor μας δίνεται η δυνατότητα να χρησιμοποιήσουμε μπλοκ εντολών που γράφει «Όταν ο χρήστης κάνει...», που περιλαμβάνει όλα τα προηγούμενα χωρίς όμως ο χρήστης να πρέπει ούτε να τα γνωρίζει, ούτε να τα τροποποιήσει, στο ελάχιστο, για να δουλέψουν. Αρκεί να ψάξει για ένα μπλοκ εντολών στην κατηγορία «Όταν», που ουσιαστικά είναι ένας διαχειριστής γεγονότων, και να «σύρει» το μπλοκ στο πρόγραμμα του.

- **Χρήση των εφαρμογών με οθόνη αφής**

Πολλοί ερευνητές έχουν διερευνήσει τα διάφορα εργαλεία, η χρήση των οποίων θα μας κάνει να χρησιμοποιήσουμε τις ηλεκτρονικές συσκευές πιο εύκολα. (Romeo κ.ά., 2003). Οι περισσότερες μελέτες που έχουν γίνει περιελάμβαναν το ποντίκι, όμως τα τελευταία χρόνια η οθόνη αφής δείχνει να κερδίζει κατά κράτος το κοινό και απόδειξη αυτού είναι η ευρεία χρήση της μέσω των κινητών τηλεφώνων που παράγονται καθώς και των ταμπλετών. Η οθόνη αφής είναι ένα πολύ εύχρηστο εργαλείο στα χέρια των ενηλίκων, αλλά βοηθά πολύ και την εξοικείωση των μικρότερων χρηστών των υπολογιστών κάνοντας την επαφή τους με αυτούς ιδιαίτερα εύκολη και άμεση. Το Android είναι ένα λειτουργικό σύστημα που η συντριπτική πλειοψηφία των συσκευών που το χρησιμοποιούν διαθέτουν οθόνη αφής. Αφού, λοιπόν, με το App Inventor αναπτύσσουμε εφαρμογές για Android είναι σχεδόν σίγουρο ότι ο χρήστης θα κάνει χρήση των εφαρμογών με οθόνη αφής. Αυτό αποτελεί ένα περαιτέρω κίνητρο για τους δημιουργούς των εφαρμογών, αφού οι εφαρμογές τους έχουν τη δυνατότητα να είναι πιο άμεσες και πιο εύχρηστες από ανθρώπους όλων των ηλικιών.

Με περισσότερους από 400.000 μοναδικούς μηνιαίους ενεργούς χρήστες που προέρχονται από 195 χώρες και έχουν δημιουργήσει σχεδόν 22 εκατομμύρια εφαρμογές αλλάζει τα δεδομένα στη δημιουργία εφαρμογών και τη διδακτική του προγραμματισμού.



### **3.4 Τι εφαρμογές μπορούμε να δημιουργήσουμε με το app inventor**

Το App Inventor μας δίνει τη δυνατότητα να δημιουργήσουμε μια πληθώρα εφαρμογών, τόσο διασκεδαστικές όσο και χρήσιμες. Ας δούμε τα είδη των εφαρμογών αυτών.

- **Παιχνίδια**

Τα παιχνίδια είναι συνήθως οι πρώτες εφαρμογές που φτιάχνουν οι νέοι προγραμματιστές. Αυτά μπορεί να είναι απλά παιχνίδια όπως μια μπάλα που αναπηδά στα τοιχώματα της οθόνης αλλά και πιο περίπλοκα όπως το πολύ γνωστό σε όλους rac-man. Επίσης, ένα άλλο είδος παιχνιδιών είναι αυτά που δίνουν τη δυνατότητα στον χρήστη να ζωγραφίζει σε μια λευκή σελίδα ή ακόμα και τις φωτογραφίες των φίλων του. Τα παιχνίδια αυτά μπορούν να γίνουν ακόμα πιο συναρπαστικά αφού μπορεί να χρησιμοποιηθούν οι διάφοροι σένσορες των συσκευών για να παίζουμε τα παιχνίδια.

- **Εκπαιδευτικές εφαρμογές**

Αν και τα παιχνίδια μπορούν να χρησιμοποιηθούν για εκπαιδευτικούς λόγους, με το App Inventor μπορούμε να φτιάξουμε εφαρμογές που να ενημερώνουν και να εκπαιδεύσουν. Μπορούμε να δημιουργήσουμε εφαρμογές που μπορούν να δημιουργήσουν κουίζ πάνω σε διάφορα μαθήματα και να ελέγξουν τις γνώσεις των χρηστών. Αξίζει να αναλογιστούμε πόσο χρήσιμο θα ήταν αυτό για ένα δάσκαλο που θέλει να προετοιμάσει την τάξη του για ένα διαγώνισμα, ή ακόμα καλύτερα για μια ομάδα εκπαιδευόμενων που θέλει να εξετάσει τις γνώσεις της σε ένα θέμα.

- **Τοπογραφικές εφαρμογές**

Εκ του ότι το App Inventor μας δίνει η δυνατότητα να κάνουμε χρήση του GPS των συσκευών μας, συνεπάγεται πως μπορούμε να κατασκευάσουμε εφαρμογές

που θα γνωρίζουν την τοποθεσία που βρισκόμαστε, το που έχουμε παρκάρει το αυτοκίνητό μας, να μας γνωστοποιήσουν τη τοποθεσία που βρίσκονται οι φίλοι και οι συγγενείς ή και τέλος, να δημιουργήσουμε εφαρμογές που θα κάνουν περιήγηση σε έναν χώρο, όπως ένα μνημείο, ένα χώρο εργασίας ή ένα εκπαιδευτικό ίδρυμα.

- **Εφαρμογές υψηλών προδιαγραφών**

Οι συσκευές που χειριζόμαστε καθημερινά έχουν αποκτήσει φοβερές δυνατότητες, των οποίων τον έλεγχο, οι μηχανικοί της Google, κατάφεραν να κάνουν πολύ απλή υπόθεση αφού μπορούμε να τις προγραμματίσουμε εύκολα με το App Inventor. Εφαρμογές που σαρώνουν bar codes, μπορούν να μιλήσουν, να αναγνωρίσουν αυτά που ακούν και δράσουν αναλόγως, να παίξουν μουσική και βίντεο, να δημιουργήσουν μουσική, να αναγνωρίσουν την κατάσταση του κινητού (λειτουργία σε θέση πορτραίτο ή κάθετη λειτουργία) καθώς και τις κινήσεις που κάνει ο χρήστης κρατώντας το, να τραβήξει φωτογραφίες και να κάνει ένα τηλεφώνημα.

- **Εφαρμογές που μπορούν να ελέγξουν ρομπότ**

Τα ρομπότ ήταν κάποτε καρποί της φαντασίας κάποιων ανθρώπων, συνήθως σκηνοθετών και σεναριογράφων. Με τη ραγδαία ανάπτυξη της τεχνολογίας, τα ρομπότ έχουν εισέλθει σε πάρα πολλούς τομείς, κυρίως παραγωγικούς, και έχουν φτάσει στο σημείο να αντικαθιστούν την ανθρώπινη παρουσία στις γραμμές παραγωγής αλλά και ελέγχου. Όπως έχει αναφερθεί και παραπάνω, ότι απασχολεί την καθημερινότητά μας, απασχολεί και την εκπαίδευση, είτε συμβάλλοντας και βοηθώντας είτε ως εκπαιδευτικό αντικείμενο. Η LEGO τα τελευταία χρόνια έχει εισάγει στην αγορά ρομπότ που μπορούν να δημιουργηθούν από όλους και να προγραμματιστούν εξίσου. Το App Inventor ακλουθώντας αυτήν την τάση και εκμεταλλευόμενο, για ακόμη μια φορά, τις δυνατότητες των «έξυπνων» συσκευών μπορεί να δημιουργήσει εφαρμογές που

θα μπορούν να ελέγξουν το ρομπότ από απόσταση, αλλά και να προγραμματίσουν τον «εγκέφαλό» του, επικοινωνώντας μέσω του πρωτοκόλλου Bluetooth.

- **Πιο περίπλοκες εφαρμογές**

Μπορεί το App Inventor να έχει έτοιμα μπλοκ εντολών που κάνουν τη σύνθεση προχωρημένων εφαρμογών θέμα ωρών, αλλά έχει επίσης και μπλοκ με δομές επανάληψης, δομές επιλογής και άλλα προγραμματιστικές δομές που μας επιτρέπουν, έχοντας μπει πια στον τρόπο σκέψης του προγραμματισμού, να δημιουργήσουμε και εφαρμογές που επιβάλλουν την επίλυση πιο σύνθετων λογικών προβλημάτων.

- **Εφαρμογές που κάνουν χρήση το διαδίκτυο**

Ένα περιβάλλον που είναι βασισμένο στο διαδίκτυο και δημιουργεί εφαρμογές για συσκευές που επικοινωνούν με το διαδίκτυο είναι αναμενόμενο θα λέγαμε να έχει τη δυνατότητα να επικοινωνήσει με το διαδίκτυο και να παίρνει δεδομένα από δίκτυα μαζικής επικοινωνίας αλλά και να κάνει χρήση του πρωτοκόλλου RSS και άλλων χρήσιμων εργαλείων που παρέχονται από το διαδίκτυο ώστε να κάνουμε τις εφαρμογές μας πιο δυνατές.

### **3.5 Σε ποιον απευθύνεται το app inventor**

Το App Inventor είναι ένα δωρεάν εργαλείο διαθέσιμο για όλους. Λειτουργεί διαδικτυακά από τον περιηγητή μας και όχι με εγκατάσταση στον υπολογιστή του χρήστη. Δεν απαιτεί να έχει ο χρήστης συσκευή Android, αφού διαθέτει προσομοιωτή που επιτρέπει τον έλεγχο των εφαρμογών που δημιουργεί. Όλα αυτά έχουν συντελέσει στη δημιουργία των εκατοντάδων χιλιάδων εφαρμογών από τον Ιανουάριο του 2011.

Το ποιοί έχουν δημιουργήσει αυτές τις εφαρμογές παρουσιάζει ιδιαίτερο ενδιαφέρον. Η πιο ενδιαφέρουσα περίπτωση προκύπτει από τον David Wolber, καθηγητή του Πανεπιστημίου του Σαν Φραντσίσκο, ο οποίος διδάσκει γενική παιδεία της επιστήμης των υπολογιστών κάνοντας χρήση του App Inventor. Σύμφωνα με τον καθηγητή το μάθημα το επιλέγουν, στην πλειοψηφία τους, φοιτητές από κατευθύνσεις οικονομικών και ανθρωπιστικών επιστημών. Το γεγονός αυτό το αποδίδει στο ότι το App Inventor καταρρίπτει το τείχος που ορθώνουν τα μαθηματικά στην πιθανότητα να συνθέσουν αυτοί οι φοιτητές κάποιο πρόγραμμα.

Παρά την έλλειψη εμπειρίας, φοιτητής γλωσσολογίας ανέπτυξαν μια εφαρμογή που δεν επιτρέπει τη συγγραφή μηνυμάτων όταν ο κάτοχος της συσκευής οδηγεί, αποτρέποντας τυχών ατυχήματα. Επιπλέον, φοιτητές επικοινωνιολογίας δημιούργησαν μια εφαρμογή που δείχνει στον κάτοχο του κατοικίδιου που βρίσκεται το κατοικίδιο ανά πάσα στιγμή. Άλλα, ο David Wolber, συνειδητοποίησε ότι το App Inventor αλλάζει πραγματικά το τοπίο στην επαφή των ανθρώπων με τον προγραμματισμό, όταν ένας φοιτητής καλών τεχνών χτύπησε την πόρτα του γραφείου του, σε προχωρημένες ώρες, για να τον ρωτήσει πως λειτουργεί η δομή επανάληψης «όσο». (David Wolber, 2011)

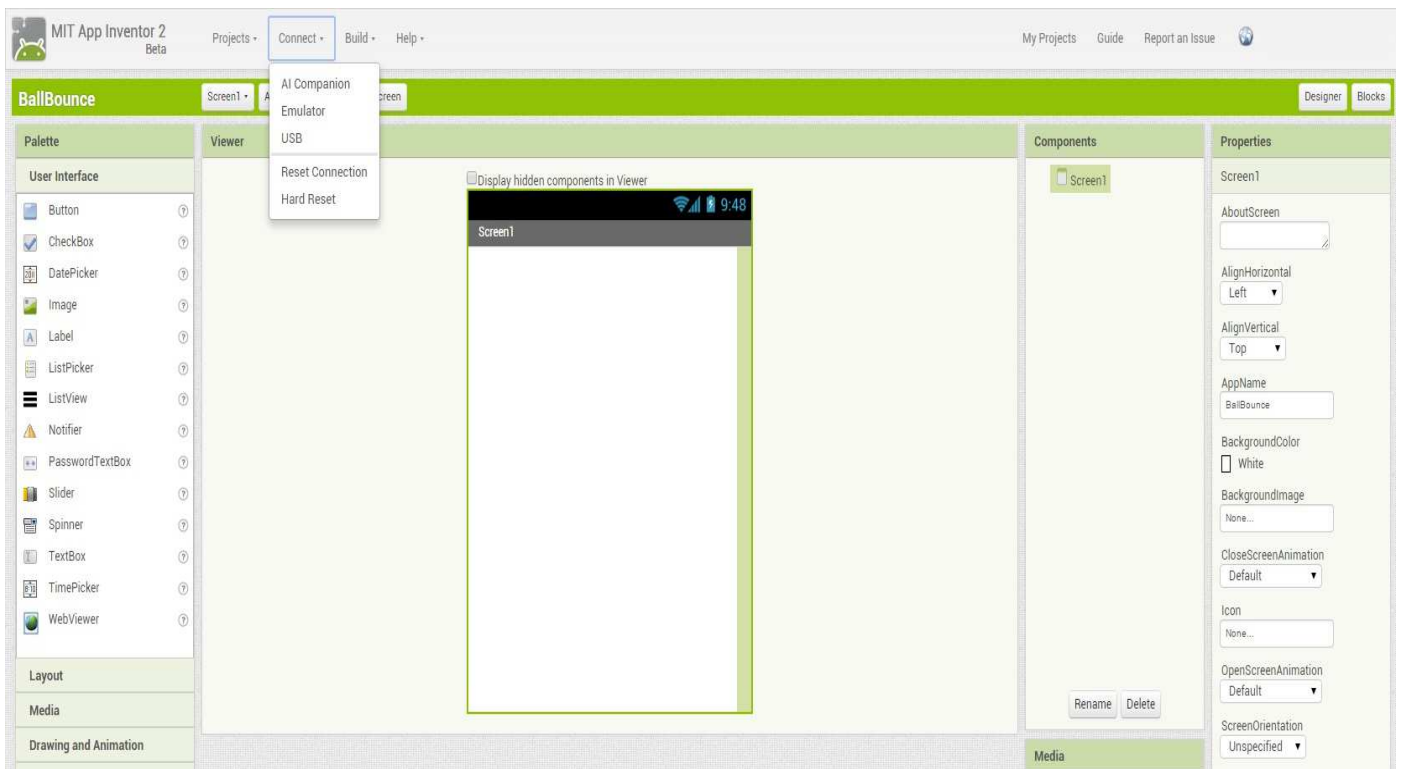
#### **3.5.1 Ξενάγηση στο περιβάλλον εργασίας του app inventor**

Θα παρουσιάσουμε τα βασικά χαρακτηριστικά του περιβάλλοντος εργασίας, τις διάφορες παλέτες εντολών, το πως μπορούμε να εισάγουμε μια εντολή, πως δημιουργούμε αντικείμενα, σκηνικά και σενάρια.

Το App Inventor τρέχει κυρίως στον περιηγητή μας, αλλά ενδεχομένως κάποια προγράμματα να χρειαστούν εγκατάσταση πριν γίνει εφικτό το να το χρησιμοποιήσουμε.

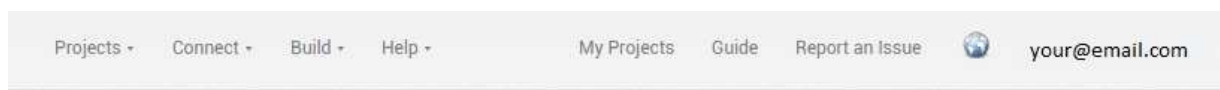
Τα κύρια μέρη του προγραμματιστικού περιβάλλοντος είναι τρία (εικόνα 10) και είναι τα εξής:

1. Το μέρος σχεδιασμού της διεπαφής της εφαρμογής (Designer Button). Όπου το χρησιμοποιούμε για να διαλέξουμε τα αντικείμενα της εφαρμογής μας και να ορίσουμε τις ιδιότητές τους.
2. Το μέρος που διαχειριζόμαστε τα μπλοκ εντολών. (Blocks Button). Όπου το χρησιμοποιούμε για να δημιουργήσουμε τις συμπεριφορές των αντικειμένων.
3. Τον προσομοιωτή-την σύνδεση μέσω USB-την χρήση της εφαρμογής Ai Companion (Connect Button). Όπου μας δίνεται η δυνατότητα με χρήση του ενσωματωμένου προσομοιωτή ή με χρήση του κινητού μας τηλεφώνου με λειτουργικό Android συνδεδεμένο ενσύρματα (USB) ή ασύρματα με χρήση της εφαρμογής Ai Companion, να ελέγξουμε τις εφαρμογές που δημιουργούμε στα δύο προηγούμενα μέρη που περιεγράφηκαν παραπάνω.



ΕΙΚΟΝΑ 3.2 – ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ APP INVENTOR

Επίσης, σημαντικό είναι να παρουσιάσουμε εδώ και τα υπόλοιπα στοιχεία του μενού (εικόνα 3.3), που είναι:



ΕΙΚΟΝΑ 3.3 - ΤΟ ΜΕΝΟΥ ΤΟΥ APP INVENTOR

- Στο κουμπί *Projects* μας δίνεται η δυνατότητα να πλοηγηθούμε στα διάφορα αποθηκευμένα έργα μας, να ανοίξουμε ένα έργο μας, να αποθηκεύσουμε ή και να εξάγουμε στον υπολογιστή μας το τρέχων έργο ή και να αποθηκεύσουμε προσωρινά τη τρέχουσα κατάσταση του έργου μας για να επανέλθουμε σε αυτή (checkpoint).
- Το κουμπί *Connect* το περιγράψαμε παραπάνω (βλ. Κύρια μέρη περιβάλλοντος εργασίας).
- Στο κουμπί *Build* μας δίνεται η δυνατότητα να δημιουργήσουμε την εφαρμογή μας, με όλα τα της τα αρχεία, έτοιμη για αυτόνομη χρήση σε περιβάλλον Android. (Δημιουργούνται τα αρχεία με κατάληξη .apk).

- Με το κουμπί *Help* μας παρέχεται τυχόν βοήθεια για το περιβάλλον χρήσης του App Inventor.
- Το κουμπί *My Projects*, το *Guide* και το *Report an Issue* είναι συντομεύσεις των επιλογών που μας δίνουν τα πιο πάνω κουμπιά.
- Στο εικονίδιο με την υδρόγειο μπορούμε να επιλέξουμε τη γλώσσα εμφάνισης της εφαρμογής.
- Και στο τελευταίο σημείο δεξιά εμφανίζεται το email με το οποίο έχουμε συνδεθεί , δίνοντας μας δυνατότητες διαχείρισης του λογαριασμού μας.

### 3.5.2 Αντικείμενα συμπεριφορές και γεγονότα

Όπως στην καθημερινή μας ζωή ερχόμαστε σε επαφή και αλληλεπίδραση με αντικείμενα, έτσι συμβαίνει και στον προγραμματισμό. Η βασική διαφορά ανάμεσα στα αντικείμενα της πραγματικής ζωής και σε αυτά του προγραμματισμού είναι ότι τα πρώτα έχουν ιδιότητες από τη φύση και βάσει αυτών συμπεριφέρονται αναλόγως, ενώ στα δεύτερα πρέπει να αποδώσουμε εμείς τις ιδιότητες και τις συμπεριφορές τους. Για παράδειγμα, αν αφήσουμε μια μπάλα μπάσκετ από τα χέρια μας αυτή θα πέσει στο έδαφος, λόγω του νόμου της βαρύτητας. Αν δημιουργήσουμε ένα παιχνίδι όμως και εισάγουμε μια φωτογραφία μπάλας σε αυτό πρέπει σαν προγραμματιστές να γράψουμε και ένα σύνολο εντολών που θα καθορίζουν την συμπεριφορά της μπάλας, για να μεταφέρουμε ουσιαστικά τον πραγματικό κόσμο στον υπολογιστή.

Έτσι, για τον προγραμματισμό ως *Αντικείμενο* ορίζουμε οποιαδήποτε μορφή φιγούρα που παίζει κάποιο ρόλο στο πρόγραμμα μας. Ως *Συμπεριφορά*, μια σειρά εντολών, σε μορφή μπλοκ στην περίπτωση του App Inventor, με τη βοήθεια των οποίων ένας προγραμματιστής μπορεί αντιστοιχίσει «δράσεις και αντιδράσεις» σε ένα αντικείμενο. Τέλος, ως *Γεγονός* ορίζουμε ένα συμβάν από το αντικείμενο για το οποίο έχει το αντικείμενο συγκεκριμένες συμπεριφορές.

### **3.5.3 Σχεδιάζοντας τα αντικείμενα**

Το πρώτο από τα εργαλεία που θα χρησιμοποιήσουμε είναι το πεδίο Σχεδίασης των Αντικειμένων (Designer). Μερικά αντικείμενα είναι αρκετά απλά, όπως ένα αντικείμενο ετικέτας (Label), το οποίο εμφανίζει κείμενο στην οθόνη, ή ένα αντικείμενο κουμπιού (Button) που ακουμπάμε για να ξεκινήσει μια ενέργεια. Υπάρχουν όμως και πιο σύνθετα αντικείμενα όπως το πίνακας (Canvas) που μπορούμε να ζωγραφίσουμε μέσα στα πλαίσια του, το επιταχυνσιόμετρο ή άλλα αντικείμενα που χρησιμοποιούν κάποιον αισθητήρα των έξυπνων συσκευών και αντικείμενα που μπορούν να στείλουν ή να λάβουν SMS.

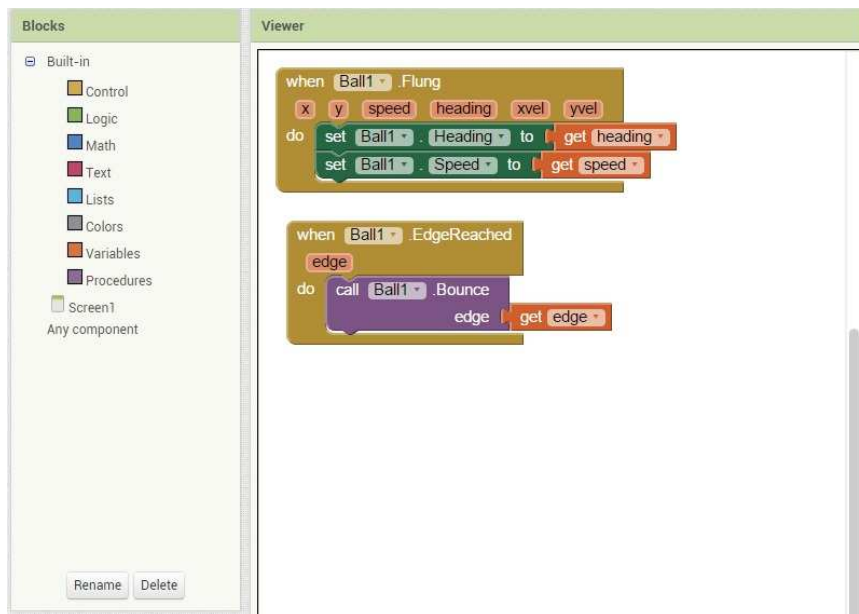
Εισάγοντας ένα αντικείμενο, εμφανίζονται στο δεξί τμήμα του παραθύρου οι ιδιότητες που έχει το αντικείμενο, όπου μπορούμε από εκεί να τις τροποποιήσουμε. Οι ιδιότητες ποικίλουν σύμφωνα με τα αντικείμενα.



### 3.5.4 Δίνοντας συμπεριφορές στα αντικείμενα

Εφόσον έχουμε εισάγει ένα αντικείμενο και έχουμε παραμετροποιήσει τις ιδιότητές του, πρέπει να επεξεργαστούμε τις συμπεριφορές των αντικειμένων αυτών και αυτό γίνεται στο τμήμα Διαχείρισης των Μπλοκ Εντολών (Blocks – εικόνα 12).

Στο αριστερό μέρος του παραθύρου εμφανίζονται τα αντικείμενα που έχουμε εισάγει, αλλά και η επιλογή να εισάγουμε μια συμπεριφορά σε όλα τα αντικείμενα επιλέγοντας το All component. Επιπλέον, εμφανίζονται σε κατηγορίες και σεσημασμένα με διαφορετικά χρώματα όλα τα είδη των μπλοκ εντολών χωρισμένα σε λογικά συρτάρια σύμφωνα με το σκοπό, τη φύση και τη λειτουργία τους.



ΕΙΚΟΝΑ 3.4 - Το Πεδίο Block του App Inventor

### 3.5.5 Οι εντολές στο app inventor

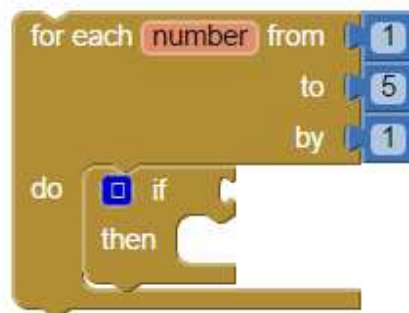
Για να δώσουμε οδηγίες δράσεις στα αντικείμενα που έχουμε δημιουργήσει, πρέπει να τοποθετήσουμε εντολές στο χώρο των σεναρίων κάθε αντικειμένου. Ένα δυνατό σημείο του App Inventor είναι ότι προωθεί τον παράλληλο προγραμματισμό καθώς ένα έργο μπορεί να περιέχει πολλά σενάρια και εντολές για κάθε αντικείμενο. Επιλέγοντας ένα αντικείμενο από κάποια οθόνη (Screen 1 στην εικόνα 13) μας εμφανίζονται όλες οι δυνατές εντολές που το αφορούν. Έτσι, για παράδειγμα, έχοντας επιλέξει το αντικείμενο Ball1 μας εμφανίζονται όλες οι εντολές που μπορεί να χρειαστούμε για να διαχειριστούμε όλα τα πιθανά συμβάντα που αφορούν το αντικείμενο αυτό. Οι εντολές αυτές πλαισιωμένες από λογικές εντολές που παίρνουμε από την κατηγορία Built-In δημιουργούν μια αλληλουχία εντολών και έχουν ως αποτέλεσμα το πρόγραμμα μας.



ΕΙΚΟΝΑ 3.5.5.5 - ΤΑ ΣΕΤ ΕΝΤΟΛΩΝ ΚΑΙ ΤΑ ΑΝΤΙΚΕΙΜΕΝΑ

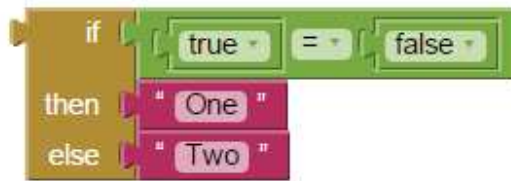
Συγκεκριμένα, οι διαθέσιμες κατηγορίες εντολών είναι:

**Control** : βασικές αλγοριθμικές εντολές, επιλογής, επανάληψης, αρχικοποίησης και λήξης μιας οθόνης ή και μιας εφαρμογής. Στο παρακάτω παράδειγμα (εικόνα 14), παρουσιάζεται μια δομή επανάληψης «για» με εμφωλευμένη δομή επιλογής. Για κάθε αριθμό (number) από το 1 μέχρι το 5 με βήμα αύξησης κατά 1, θα πραγματοποιείται η δομή επιλογής.



ΕΙΚΟΝΑ 3.5.5 - ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ ΚΑΙ ΔΟΜΗ ΕΠΙΛΟΓΗΣ ΣΤΟ APP INVENTOR

**Logic** : εδώ υπάρχουν οι βασικές λογικές πράξεις που μπορεί να χρησιμοποιήσουμε για να κάνουμε έναν έλεγχο σε μια δομή επιλογής ή σε μια δομή επανάληψης. Τα μπλοκ αυτά δεν μπορούν να ενταχθούν μόνα τους, αφού δεν είναι εντολές, αλλά πρέπει να τα εισάγουμε σε ένα μπλοκ που θα χρειάζεται κάποια λογική πράξη. Για παράδειγμα στην εικόνα που ακολουθεί, υπάρχει μια δομή επιλογής όπου αν αληθεύει η λογική πράξη που είναι σε πράσινο φόντο θα εμφανίσει στην οθόνη μας «One» αλλιώς θα εμφανίσει «Two». Στην προκειμένη περίπτωση η λογική πράξη που έχουμε βάλει θα είναι πάντα λανθασμένη άρα θα εκτυπώνεται συνέχεια «Two».



ΕΙΚΟΝΑ 3.5.5 - ΔΟΜΗ ΕΠΙΛΟΓΗΣ ΜΕ ΛΟΓΙΚΕΣ ΠΡΑΞΕΙΣ

**Math** : στην κατηγορία αυτή εντάσσονται όλα τα εργαλεία που μπορεί να χρειαστούν για οποιαδήποτε μαθηματική πράξη ή μαθηματικό έλεγχο. Μπορούμε να ελέγξουμε αν μια μεταβλητή είναι αριθμός, να εισάγουμε έναν σταθερό (εικόνα 16) ή κάποιον τυχαίο αριθμό, να κάνουμε όλες τις μαθηματικές πράξεις από τις πιο απλές όπως η πρόσθεση αλλά και πιο σύνθετες όπως η στρογγυλοποίηση κάποιου αριθμού ή ο υπολογισμός της τετραγωνικής ρίζας.

**Text** : ότι είναι η κατηγορία Math για τους αριθμούς είναι και η κατηγορία Text για τους χαρακτήρες. Υπάρχουν εντολές λοιπόν που απλά εμφανίζουν χαρακτήρες (εικόνα 15), που ελέγχουν αν η μεταβλητή έχει χαρακτήρες, αν περιέχει έναν συγκεκριμένο χαρακτήρα και να συγκρίνει τους χαρακτήρες δύο μεταβλητών. Επιπλέον μπορεί να μεταβάλλει μια μεταβλητή και να την χωρίσει σε μικρότερες ή και να κάνει αναγραμματισμό.

**Lists** : το App Inventor μας δίνει τη δυνατότητα να δημιουργούμε και να επεξεργαζόμαστε λίστες και σε αυτήν την κατηγορία των μπλοκ εντάσσονται όλες οι εντολές που μας επιτρέπουν να το κάνουμε (εικόνα 16).

**Colors** : εδώ μπορούμε να εισάγουμε σαν μεταβλητή ένα χρώμα εμφάνισης, επιλέγοντας από κάποια βασικά έτοιμα χρώματα ή μπορούμε ακόμα και να δημιουργήσουμε ένα χρώμα με το σύστημα RGB δημιουργώντας ένα πίνακα (εικόνα 16), όπου εισάγουμε το μαύρο χρώμα.



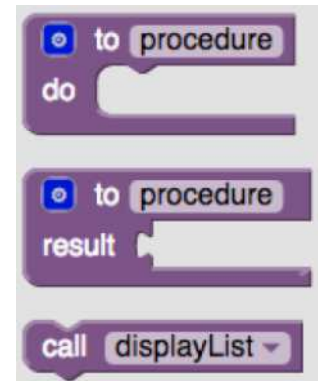
ΕΙΚΟΝΑ 3.5.5 ΔΗΜΙΟΥΡΓΙΑ ΧΡΩΜΑΤΟΣ RGB ΜΕ ΧΡΗΣΗ ΛΙΣΤΑΣ

**Variables** : ένας πολύ σημαντικός παράγοντας για τον προγραμματισμό, όπως έχουμε ήδη αναφέρει, είναι οι μεταβλητές. Σε αυτή την κατηγορία, έχουμε τις εντολές που δημιουργούν και αρχικοποιούν τις μεταβλητές που θέλουμε να χρησιμοποιήσουμε στο πρόγραμμά μας. Μπορούμε να δημιουργήσουμε μια μεταβλητή για όλες τις «οθόνες» και τα αντικείμενα της εφαρμογής μας ή να δημιουργήσουμε μεταβλητές που θα αφορούν συγκεκριμένη «οθόνη» ή αντικείμενο. Στην εικόνα που ακολουθεί, βλέπουμε την δημιουργία μιας μεταβλητής που θα είναι κοινή για όλα τα αντικείμενά μας, θα έχει το όνομα «var1» και την αρχικοποιούμε δίνοντάς της την τιμή «variable\_one».



ΕΙΚΟΝΑ 3.5.5 - ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΑΡΧΙΚΟΠΟΙΗΣΗ ΜΕΤΑΒΛΗΤΗΣ

**Procedures** : το App Inventor, όπως έχουμε αναφέρει, προσφέρει ένα μεγάλο αριθμό από έτοιμες εντολές που λύνουν τα χέρια και μειώνουν στο ελάχιστο τη αποπεράτωση πολλών προγραμματιστικών απαιτήσεων. Ωστόσο, το App Inventor δίνει την δυνατότητα να επεκτείνει ο χρήστης τις προγραμματιστικές του ικανότητες δημιουργώντας ο ίδιος τις δικές του εντολές ή μάλλον τις δικές του ακολουθίες εντολών, που θα χρησιμοποιεί διαρκώς στα προγράμματά του. Σε πολλές γλώσσες, οι ακολουθίες αυτές είναι γνωστές ως συναρτήσεις ή υποπρογράμματα, εδώ ονομάζονται διαδικασίες. Μια διαδικασία, λοιπόν, είναι μια σειρά από μπλοκ εντολών που μπορούμε να «καλέσουμε» σε οποιοδήποτε σημείο της εφαρμογής το χρειαστούμε. Σε αυτό το σετ εντολών, έχουμε τα μπλοκ που δημιουργούν και καλούν μια διεργασία. Αν θέλουμε η διεργασία να μας επιστρέφει μια τιμή, τότε χρησιμοποιούμε το δεύτερο μπλοκ που έχει το «result», διαφορετικά επιλέγουμε το πρώτο μπλοκ.



ΕΙΚΟΝΑ 3.10 – Μπλοκ ΧΕΙΡΙΣΜΟΥ ΤΩΝ ΔΙΑΔΙΚΑΣΙΩΝ

### **3.6 Παράδειγμα χρήσης: Παρουσίαση βασικών εννοιών του προγραμματισμού, με τη βοήθεια μιας εφαρμογής/παιχνιδιού σχεδιασμένη με το App Inventor, σε μη ειδικευμένους στον προγραμματισμό φοιτητές.**

Όπως σχολιάσαμε και παραπάνω, είναι ευρέως γνωστό το πόσο η τεχνολογία έχει εισχωρήσει στις ζωές μας και κατά συνέπεια και σε όλα τα επαγγέλματα. Γίνεται έτσι ολοένα και πιο επιτακτική η ανάγκη για όλους να μπορούν όχι μόνο να είναι χρήστες των εφαρμογών, τόσο στον υπολογιστή όσο και σε συσκευές όπως τα smart phones, αλλά και να μπορούν να δημιουργήσουν τις εφαρμογές που ενδεχομένως να χρειάζονται.

Μέχρι και λίγα χρόνια πίσω, αυτό ήταν κάτι αρκετά δύσκολο για κάποιον που δεν ήταν στο άμεσο γνωστικό του πεδίο ο προγραμματισμός. Πλέον, με την ανάπτυξη των διδακτικών γλωσσών προγραμματισμού και πιο συγκεκριμένα με το App Inventor αυτό από ακατόρθωτο που ίσως να φάνταζε, γίνεται όλο και πιο απλή υπόθεσή.

Στο συγκεκριμένο παράδειγμα χρήσης, θα παρουσιάσουμε μια εφαρμογή που στόχο έχει την εκπαίδευση του χρήστη στα ισοδύναμα κλάσματα. Μέσω αυτής θα στοχεύσουμε να γίνουν πιο εύκολα κατανοητές βασικές έννοιες όπως:

- ✓ η κατανόηση της αλληλουχίας του μπλοκ: δηλαδή, η ιδέα ότι η εκτέλεση ρέει διαδοχικά όταν τα μπλοκ συνδέονται μεταξύ τους κάθετα
- ✓ αυτή των μεταβλητών: με την χρήση και την επεξεργασία αλφαριθμητικών μεταβλητών αλλά και μετρητών καθώς και με την δημιουργία λιστών και μικρών βάσεων δεδομένων
- ✓ των δομών επιλογής: η χρήση των δομών επιλογής “if” με τα αντίστοιχα μπλοκ έτσι ώστε να ελέγξουν την εκτέλεση του προγράμματος
- ✓ των δομών επαναλήψεων: το πως και πότε χρειάζεται η επανάληψη μερών του κώδικα
- ✓ την διεκπεραίωση αριθμητικών και λογικών πράξεων με την χρήση των μπλοκ
- ✓ την διαχείριση πολυμέσων, όπως εικόνες και ήχους

- ✓ της λειτουργικής αποσύνθεσης: το «σπάσιμο» του κώδικα της εφαρμογής σε μικρότερα κομμάτια κώδικα, με την χρήση διαδικασιών και συναρτήσεων, για την απλοποίηση της διαχείρισής του
- ✓ του προγραμματισμού με χειρισμό γεγονότων ή συμβάντων: το να ανταποκρίνεται στα συμβάντα που προκύπτουν από τον χρήστη ή από κάποιον αισθητήρα
- ✓ η αποσφαλμάτωση του κώδικα

### 3.6.1 Περιγραφή της Εφαρμογής

Η εφαρμογή που δημιουργήθηκε για αυτόν το σκοπό είναι μια «ζυγαριά κλασμάτων» όπου ο χρήστης σέρνει πάνω στη ζυγαριά δύο κλάσματα και ελέγχει εάν είναι ισοδύναμα. Τα κλάσματα, που καλείται να συγκρίνει ο χρήστης, χωρίζονται σε επίπεδα με βάση την δυσκολία της σύγκρισης και την πολυπλοκότητά τους. Ο χρήστης διαλέγει το επίπεδο που επιθυμεί και εισέρχεται σε αυτό. Κατόπιν σύρει τα κλάσματα στη ζυγαριά και ελέγχει αν είναι ισοδύναμα. Όταν δυο κλάσματα είναι ισοδύναμα μεταφέρονται στο πάνω μέρος της οθόνης και έτσι μένουν στον χρήστη τα υπόλοιπα προς σύγκρισή.

Μόλις τα διαθέσιμα κλάσματα τελειώσουν η εφαρμογή ενημερώνει τον χρήστη ότι ολοκλήρωσε το επίπεδο και τον παροτρύνει να συνεχίσει στο επόμενο επίπεδο. Ταυτόχρονα υπολογίζεται η βαθμολογία που πετυχαίνει ο χρήστης στο εκάστοτε επίπεδο και εμφανίζεται στην αρχική οθόνη, όπου ο χρήστης επιλέγει το επίπεδο, με την μορφή αστεριών. Όταν ένας χρήστης ολοκληρώνει ένα επίπεδο χωρίς κανένα σφάλμα στις συγκρίσεις, λαμβάνει και τον μέγιστο αριθμό αστεριών (3). Τέλος, δίνεται η δυνατότητα να ξαναπροσπαθήσει ο χρήστης σε όποιο επίπεδο επιθυμεί.

### 3.6.2 Δομή της εφαρμογής

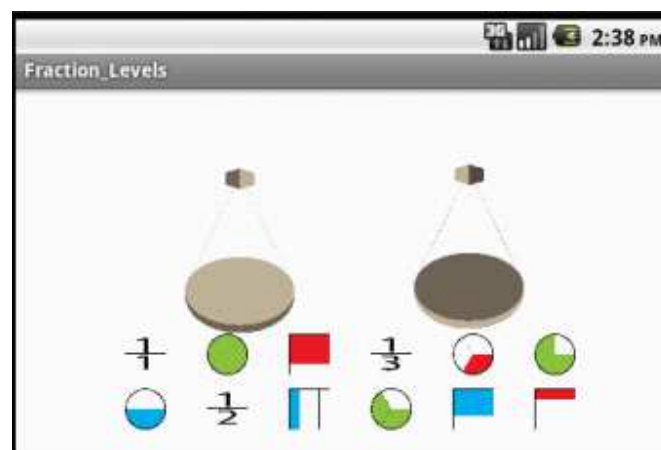
Δύο οθόνες αποτελούν το βασικό κορμό της εφαρμογής μας. Στην πρώτη οθόνη εμφανίζονται τα τέσσερα επίπεδα που μπορεί να επιλέξει ο χρήστης. Οι

τέσσερις εικόνες των επιπέδων μεταβάλλονται όταν ολοκληρώνεται το αντίστοιχο επίπεδο από τον χρήστη σύμφωνα με την βαθμολογία που κατάφερε να συγκεντρώσει.



ΕΙΚΟΝΑ 3.6.2 – Η ΠΡΩΤΗ ΟΘΟΝΗ ΜΕ ΣΥΜΠΛΗΡΩΜΕΝΟ ΤΟ ΠΡΩΤΟ ΕΠΙΠΕΔΟ

Η δεύτερη οθόνη χωρίζεται σε τρία τμήματα. Στο κάτω μέρος της οθόνης εμφανίζονται οι εικόνες των κλασμάτων που είναι προς «ζύγισμα». Κάθε φορά που φορτώνεται η οθόνη, φορτώνονται και διαφορετικές εικόνες κλασμάτων, ανάλογα με το επίπεδο που έχει επιλέξει ο χρήστης στην προηγούμενη οθόνη.

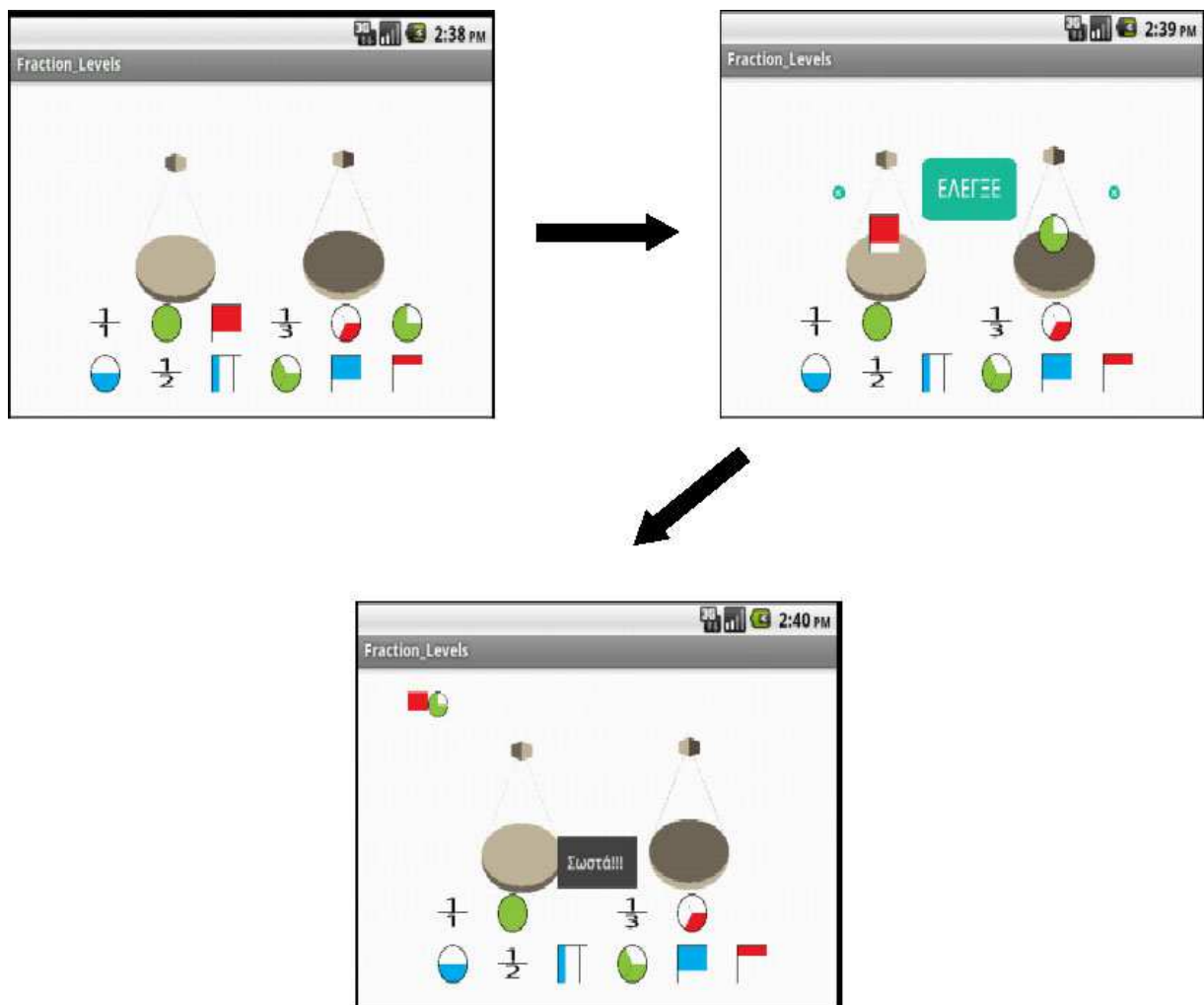


ΕΙΚΟΝΑ 3.6.22 – Η ΔΕΥΤΕΡΗ ΟΘΟΝΗ ΟΠΩΣ ΕΜΦΑΝΙΖΕΤΑΙ ΟΤΑΝ ΕΠΙΛΕΓΕΙ Ο ΧΡΗΣΤΗΣ ΕΠΙΠΕΔΟ

Στο μεσαίο τμήμα βρίσκεται η ζυγαριά, όπου ο χρήστης πρέπει να μεταφέρει τις εικόνες για να πραγματοποιηθεί η σύγκρισή τους. Μόλις ο χρήστης



τοποθετήσει και στις δύο πλάστιγγες από ένα κλάσμα εμφανίζεται ένα κουμπί όπου ο χρήστης καλείται να πατήσει αν είναι σίγουρος για την επιλογή των κλασμάτων. Όταν το κουμπί πατηθεί πραγματοποιείται η σύγκριση των κλασμάτων. Αν τα κλάσματα δεν είναι ισοδύναμα ο χρήστης θα πρέπει να αφαιρέσει το ένα ή και τα δύο από τις πλάστιγγες και να ξαναπροσπαθήσει. Αν τα κλάσματα είναι ισοδύναμα τότε το ζευγάρι αυτό μεταφέρεται στο άνω, και τελευταίο μέρος της οθόνης όπου τοποθετούνται τα ζεύγη από τις πετυχημένες



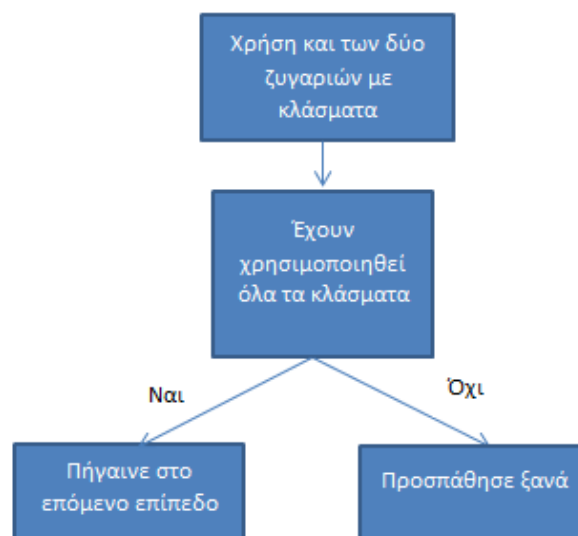
συγκρίσεις.

ΕΙΚΟΝΑ 3.6.23 – Η ΔΕΥΤΕΡΗ ΟΘΟΝΗ ΟΤΑΝ Ο ΧΡΗΣΤΗΣ «ΖΥΓΙΖΕΙ» ΔΥΟ ΑΠΟ ΤΑ ΚΛΑΣΜΑΤΑ

### 3.6.3 Παραδείγματα χρήσης και εμφάνιση των προγραμματιστικών εννοιών στην εφαρμογή.

Σε αυτό το σημείο θα παραθέσουμε κάποια παραδείγματα χρήσης των προγραμματιστικών εννοιών που αναφέραμε παραπάνω που συναντάμε για τις ανάγκες της εφαρμογής.

Καθώς ο χρήστης μπαίνει στο εκάστοτε επίπεδο, γίνεται συνέχεια ο έλεγχος αν τα κλάσματα που έχει στη διάθεσή του για να ζυγίσει έχουν τελειώσει. Αν τελειώσουν, θα πρέπει να του δοθεί η δυνατότητα να προχωρήσει στο επόμενο επίπεδο, ή να επιστρέψει στο αρχικό μενού. Αυτό επιτυγχάνεται με τους διαχειριστές συμβάντων.

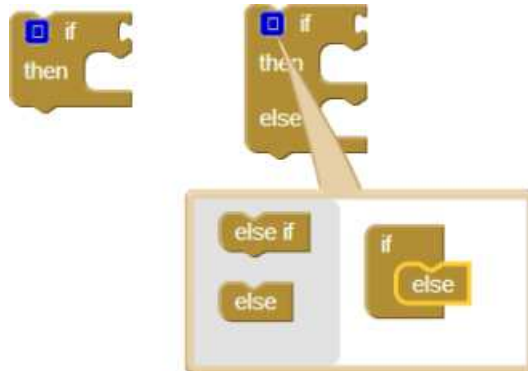


ΕΙΚΟΝΑ 3.14 ΔΙΑΓΡΑΜΜΑ ΔΙΑΘΡΩΣΗΣ ΔΟΜΗΣ ΕΠΙΛΟΓΗΣ

Οι διαχειριστές συμβάντων εκτελούν συγκεκριμένες μεθόδους σε απόκριση ενός συγκεκριμένου συμβάντος. Η απόκριση δεν είναι απαραίτητο να είναι γραμμική, ωστόσο, μερικές μέθοδοι θα εκτελεστούν μόνο υπό κάποιες ορισμένες συνθήκες. Στο δικό μας παράδειγμα, η εφαρμογή ελέγχει τα λάθη του χρήστη και αναλόγως του ανταποδίδει τα αντίστοιχα αστεράκια για το συγκεκριμένο επίπεδο.

Για να επιτευχθούν οι έλεγχοι αυτοί, το App Inventor παρέχει το αντίστοιχο μπλοκ στον χώρο διαχείρισης των εντολών. Το μπλοκ μπορεί να επεκταθεί σύμφωνα με

τις απαιτήσεις της εφαρμογής πατώντας το μπλε κουμπί όπως φαίνεται στην παρακάτω εικόνα.



ΕΙΚΟΝΑ 3.15 ΜΠΛΟΚ ΕΝΤΟΛΗΣ ΔΟΜΗΣ ΕΠΙΛΟΓΗΣ ΣΤΟ APP INVENTOR

Μέσα στις «εσοχές ελέγχου» των if και else if μπορούμε να εισάγουμε οποιαδήποτε «λογική έκφραση» («boolean expression»). Με τον όρο «λογική έκφραση» εννοούμε κάθε μαθηματική ισότητα που μας επιστρέφει σαν αποτέλεσμα το λογικό «αληθές» ή «ψευδές». Έτσι, στις εσοχές ελέγχου γίνονται οι έλεγχοι των τιμών των μεταβλητών και των ιδιοτήτων χρησιμοποιώντας τους σχετικούς και λογικούς τελεστές, όπως αυτοί στην παρακάτω εικόνα.



ΕΙΚΟΝΑ 3.16 ΜΠΛΟΚ ΣΧΕΤΙΚΩΝ ΚΑΙ ΛΟΓΙΚΩΝ ΤΕΛΕΣΤΩΝ ΣΤΟ APP INVENTOR

Τα μπλοκ κώδικα που τοποθετούμε μέσα στις εσοχές του «then» των «if» μπλοκ είναι αυτά που θα εκτελεστούν αν οι έλεγχοι βγουν αληθείς. Αν οι έλεγχοι είναι ψευδείς, η εφαρμογή συνεχίζει χωρίς να εκτελεστούν οι εμφολευμένες στο if εντολές.

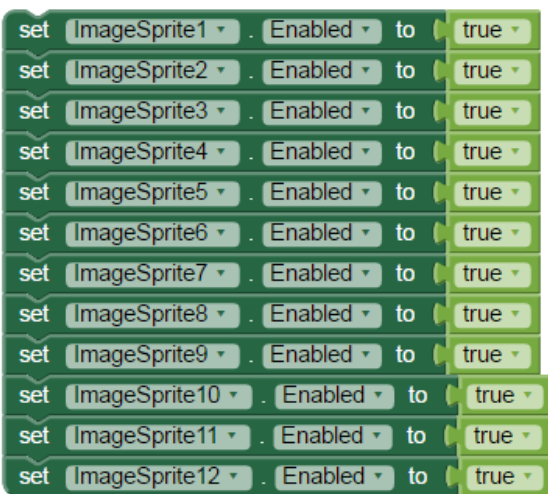
Πιο συγκεκριμένα, στην εφαρμογή μας η συγκεκριμένη δομή επιλογής θα έχει την παρακάτω μορφή.



ΕΙΚΟΝΑ 3.17 ΜΠΛΟΚ ΕΝΤΟΛΩΝ ΣΤΗΝ ΕΦΑΡΜΟΓΗ ΜΑΣ ΣΤΟ APP INVENTOR

Αν ο έλεγχος που έχουμε βάλει στην εσοχή του if είναι αληθής τότε θα εκτελεστεί η μέθοδος εμφάνισης του Notifier στην οθόνη του χρήστη που θα τον συγχαίρει αρχικά για την ολοκλήρωση του επιπέδου («Μπράβο» και «Ολοκλήρωσες το επίπεδο!») και στην συνέχεια θα τον βάζει πατώντας το κουμπί να προχωρήσει στο επόμενο επίπεδο («Πήγαινε στο επόμενο»).

Όταν αρχικοποιείται κάθε επίπεδο εμφανίζονται τα κλάσματα σαν εικόνες, όπου



ΕΙΚΟΝΑ 3.18 ΠΑΡΑΔΕΙΓΜΑ ΕΝΤΟΛΩΝ ΧΩΡΙΣ ΤΟ ΜΠΛΟΚ ΕΝΤΟΛΩΝ ΕΠΑΝΑΛΗΨΗΣ

μπορεί ο χρήστης να τις σύρει για να τις «ζυγίσει» πάνω στην ζυγαριά.

Θα πρέπει να εμφανίσουμε τις εικόνες των κλασμάτων 12 (δώδεκα) φορές κάθε φορά που ο χρήστης μπαίνει σε κάθε επίπεδο. Θα εκτελεστεί τις αντίστοιχες φορές το ίδιο κομμάτι κώδικα. Συμβατικά, για να συμβεί αυτό, θα γράφαμε τον κώδικα εμφάνισης της μια εικόνας και στην καλύτερη περίπτωση θα κάναμε αντιγραφή και επικόλληση του ίδιου κομματιού κώδικα, παραμετροποιημένο αναλόγως, δώδεκα φορές.

Αυτή η μέθοδος, αντιγραφής-επικόλλησης, μπορεί να ήταν κατάλληλη αν είχαμε να κάνουμε με λιγότερες εικόνες. Αλλά αν, όπως στην περίπτωση μας, έχουμε να διαχειριστούμε μεγαλύτερο αριθμό δεδομένων, αν χρειαστεί να μεταβάλουμε κάτι στον κώδικα μπορεί να αποδειχτεί αρκετά χρονοβόρο. Για να μην βάζουμε τα ίδια κομμάτια

κώδικα ξανά και ξανά, κάνοντας τον κώδικα μας πιο δύσκολο τόσο στην επεξεργασία όσο και στην αποσφαλμάτωση αλλά και στην κατανόηση, χρησιμοποιούμε τις δομές επανάληψης.

Όπως αναφέρθηκε και πιο πάνω οι εφαρμογές δεν εκτελούν απαραίτητα γραμμικά εντολές και μπορεί να εκτελεί ένα σύνολο εντολών ξανά και ξανά χωρίς απαραίτητα να εκτελούνται όλες κάθε φορά. Το πρώτο πραγματοποιείται με χρήση των δομών επιλογής ενώ το δεύτερο με την χρήση δομών επανάληψης. Στις τελευταίες, αφού εκτελεστεί μια σειρά μεθόδων και συναρτήσεων, επιστρέφει στην αρχή και εκτελούνται ξανά. Το App Inventor παρέχει πληθώρα μπλοκ επαναλήψεων, συμπεριλαμβανομένου της for each, της while κ.α.. Στην for each συγκεκριμένα, όπου θα δούμε και παρακάτω να χρησιμοποιούμε στην εφαρμογή μας, εκτελούνται οι εντολές που εμφωλεύουμε για κάθε ένα αντικείμενο μιας λίστας.

Στην εφαρμογή μας έχουμε φτιάξει μια λίστα με τις φωτογραφίες των κλασμάτων και θέλουμε να εμφανίζονται 12 από αυτές στα αντίστοιχα πεδία. Επομένως, θα χρησιμοποιήσουμε μια δομή επανάληψης for each μέχρι να τελειώσουν οι φωτογραφίες προς εμφάνιση στη λίστα μας. Έτσι ο κώδικας μας θα πάρει την παρακάτω μορφή.



ΕΙΚΟΝΑ 3.19 ΜΠΛΟΚ ΕΝΤΟΛΩΝ ΔΟΜΗΣ ΕΠΑΝΑΛΗΨΗΣ ΣΤΗΝ ΕΦΑΡΜΟΓΗ ΜΑΣ ΣΤΟ APP INVENTOR

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. [http://users.sch.gr/tsakarak/Yliko\\_Blog/Gel/Tzimogiannis\\_8ewria\\_didaktikis\\_Programmatismou.pdf](http://users.sch.gr/tsakarak/Yliko_Blog/Gel/Tzimogiannis_8ewria_didaktikis_Programmatismou.pdf)
2. [http://dide.ilei.sch.gr/keplinet/education/docs/syn\\_syrou2009\\_xynogalos\\_c.pdf](http://dide.ilei.sch.gr/keplinet/education/docs/syn_syrou2009_xynogalos_c.pdf)
3. <http://tinyurl.com/hr9t8rz>
4. [https://www.academia.edu/9828622/e-Education\\_in\\_Teaching\\_Programming\\_-\\_Forty\\_Years\\_of\\_Promises](https://www.academia.edu/9828622/e-Education_in_Teaching_Programming_-_Forty_Years_of_Promises)
5. [https://www.academia.edu/2994305/Shapemaker\\_A\\_game-based\\_introduction\\_to\\_programming](https://www.academia.edu/2994305/Shapemaker_A_game-based_introduction_to_programming)
6. [https://www.academia.edu/2184149/Teaching\\_Novices\\_Programming\\_Using\\_a\\_Robot\\_Simulator\\_Case\\_Study\\_Protocol](https://www.academia.edu/2184149/Teaching_Novices_Programming_Using_a_Robot_Simulator_Case_Study_Protocol)
7. [https://www.academia.edu/1110423/Experiences\\_of\\_Prospective\\_High\\_School\\_Teachers\\_Using\\_a\\_Programming\\_Teaching\\_Tool](https://www.academia.edu/1110423/Experiences_of_Prospective_High_School_Teachers_Using_a_Programming_Teaching_Tool)
8. [https://www.academia.edu/8887324/MIT\\_Scratch\\_A\\_Powerful\\_Tool\\_for\\_Improving\\_Teaching\\_of\\_Programming](https://www.academia.edu/8887324/MIT_Scratch_A_Powerful_Tool_for_Improving_Teaching_of_Programming)
9. [https://www.academia.edu/8887203/Computer\\_Supported\\_Collaborative\\_Learning\\_CSCL\\_in\\_Virtual\\_Classes\\_VCs\\_The\\_Effect\\_of\\_Systematic\\_Teaching\\_on\\_Collaborative\\_Learning](https://www.academia.edu/8887203/Computer_Supported_Collaborative_Learning_CSCL_in_Virtual_Classes_VCs_The_Effect_of_Systematic_Teaching_on_Collaborative_Learning)
10. [http://sepima.gr/FILES/PEKAP\\_7o\\_Synedrio/ergasies/10Karipidis.pdf](http://sepima.gr/FILES/PEKAP_7o_Synedrio/ergasies/10Karipidis.pdf)
11. <http://www.cetl.edmedu.upatras.gr/proc2/proceedings.html>
12. [http://di.ionio.gr/cie/images/documents13/CIE2013\\_proceedings/data/cie2013\\_2\\_25.pdf](http://di.ionio.gr/cie/images/documents13/CIE2013_proceedings/data/cie2013_2_25.pdf)
13. [http://users.auth.gr/users/2/5/126752/public\\_html/index\\_files/ETPE08\\_Papadopoulos.pdf](http://users.auth.gr/users/2/5/126752/public_html/index_files/ETPE08_Papadopoulos.pdf)
14. [http://hmathia14.ekped.gr/praktika14/VoID/VoID\\_233\\_246.pdf](http://hmathia14.ekped.gr/praktika14/VoID/VoID_233_246.pdf)
15. <http://tinyurl.com/z74oz9n>

16. Κόμης Β. (2001), Μελέτη βασικών εννοιών του προγραμματισμού στο πλαίσιο μιας οικοδομηστικής διδακτικής προσέγγισης, ΘΕΜΑΤΑ στην Εκπαίδευση, 2(2-3), 243-270
17. Τζιμογιάννης Α. (2000), Η διδασκαλία του Προγραμματισμού Η/Υ στη Δευτεροβάθμια Εκπαίδευση. Δυσκολίες και αντιλήψεις των μαθητών για την έννοια της μεταβλητής, Η Βάση, 2, 35-42, Ιωάννινα
18. Τζιμογιάννης Α. (2002), Διδακτική Πληροφορικής, Προγράμματα Σπουδών και διδακτικές πρακτικές στο Ενιαίο Λύκειο, Πρακτικά 3ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή "Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση", Τόμος Α', 229-238, Ρόδος
19. Τζιμογιάννης Α. (2003), Η διδασκαλία του Προγραμματισμού στο Ενιαίο Λύκειο: προς ένα ολοκληρωμένο πλαίσιο με στόχο την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων, Πρακτικά 2ου Πανελληνίου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ: «Αξιοποίηση των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών στη Διδακτική Πράξη», Τόμος Α', 706-720, Σύρος
20. Τζιμογιάννης Α. (2005), Προς ένα παιδαγωγικό πλαίσιο διδασκαλίας του προγραμματισμού στη δευτεροβάθμια εκπαίδευση, στο Α. Τζιμογιάννης (επιμ.), Πρακτικά 3ου Πανελληνίου Συνεδρίου "Διδακτική της Πληροφορικής", 99-111, Κόρινθος
21. Τζιμογιάννης, Α. και 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", 28-34, Αθήνα: ΕΠΥ
22. Τζιμογιάννης, Α. και Γεωργίου, Β. (1999), Οι δυσκολίες μαθητών της Δευτεροβάθμιας Εκπαίδευσης στην εφαρμογή της δομής ελέγχου για την ανάπτυξη απλών αλγορίθμων, Πρακτικά Πανελληνίου Συνεδρίου "Πληροφορική και Εκπαίδευση", 183-192, Ιωάννινα
23. Τζιμογιάννης Α. & Γιούνης Α. (2003), Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, Τόμος Α', Αθήνα: Εκδόσεις Σαββάλα

24. Τζιμογιάννης Α. και Κόμης Β. (2000), Η έννοια της μεταβλητής στον Προγραμματισμό: δυσκολίες και παρανοήσεις μαθητών του Ενιαίου Λυκείου, Πρακτικά 2ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή “Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση”, 103-114, Πάτρα.
25. Τζιμογιάννης Α. και Κόμης Β. (2004), Μελέτη των αναπαραστάσεων μαθητών του Ενιαίου Λυκείου για τη ροή δεδομένων και το ρόλο των βασικών μονάδων του υπολογιστή, Πρακτικά 2ης Διημερίδας “Διδακτική της Πληροφορικής”, 73-85, Βόλος
26. Τζιμογιάννης Α., Πολίτης Π. και Κόμης Β. (2005), Μελέτη των αναπαραστάσεων τελειόφοιτων μαθητών Ενιαίου Λυκείου για την έννοια της μεταβλητής, στο Α. Τζιμογιάννης (επιμ.), Πρακτικά 3ου Πανελληνίου Συνεδρίου “Διδακτική της Πληροφορικής”, 61-70, Κόρινθος.
27. Τζιμογιάννης Α., Τσιωτάκης Π. και Sajaniemi J. (2006), Μελετώντας το ρόλο των προσομοιώσεων αλγορίθμων στη διδασκαλία του προγραμματισμού στο Ενιαίο Λύκειο, στο Ε. Σταυρίδου & Χ. Σολομωνίδου (επιμ.), Πρακτικά Πανελληνίου Συνεδρίου “Ψηφιακό Εκπαιδευτικό Υλικό: Ζητήματα