

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2015)

4ο Πανελλήνιο Συνέδριο «Ένταξη και Χρήση των ΤΠΕ στην Εκπαιδευτική Διαδικασία»



Alice ή App Inventor. Ποιό Προγραμματιστικό Περιβάλλον να επιλέξω για τη Διδασκαλία του Προγραμματισμού στην Α' Λυκείου;

Στ. Παπαδάκης, Β. Ορφανάκης

Βιβλιογραφική αναφορά:

Παπαδάκης Σ., & Ορφανάκης Β. (2022). Alice ή App Inventor. Ποιό Προγραμματιστικό Περιβάλλον να επιλέξω για τη Διδασκαλία του Προγραμματισμού στην Α' Λυκείου;. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση, 1*, 10–17. ανακτήθηκε από <https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/4290>

Alice ή App Inventor. Ποιό Προγραμματιστικό Περιβάλλον να επιλέξω για τη Διδασκαλία του Προγραμματισμού στην Α' Λυκείου;

Στ. Παπαδάκης¹, Β. Ορφανάκης²

¹ ΥΠΙΑΙΘ, stpapakis@gmail.com

² ΥΠΙΑΙΘ, vorfan@gmail.com

Περίληψη

Η διδασκαλία του προγραμματισμού είναι δύσκολη και ως εκ τούτου, είναι φυσικό, οι εμπλεκόμενοι στην εκπαιδευτική διαδικασία να αναζητούν τρόπους να διευκολύνουν τη διδακτική δραστηριότητα μέσω της ανάπτυξης νέων διδακτικών εργαλείων. Τα οπτικά περιβάλλοντα προγραμματισμού είναι από τα πιο δημοφιλή νέα εργαλεία καθώς παρέχουν ένα γραφικό περιβάλλον διεπαφής για την κατασκευή του κώδικα και προβολή ενός προγράμματος. Στην παρούσα εργασία, διερευνούμε τη χρήση του Alice και του App Inventor for Android, τα οποία αναφέρονται στο βιβλίο της Πληροφορικής της Α' Λυκείου, ως οπτικά περιβάλλοντα προγραμματισμού, από παιδαγωγική και τεχνολογική σκοπιά, προκειμένου να καταλήξουμε ποιο από τα δυο είναι καταλληλότερο για τη διδασκαλία του προγραμματισμού.

Λέξεις κλειδιά: *Alice, App Inventor, αρχάριοι προγραμματιστές*

1. Εισαγωγή

Παρότι ο αριθμός των θέσεων εργασίας στον τομέα της τεχνολογίας των πληροφοριών είναι σε άνοδο, το ποσοστό των νεοεισερχόμενων σπουδαστών στον κλάδο της Πληροφορικής μειώνεται σταθερά από την κορύφωση του στα τέλη της δεκαετίας του 1990 (Cooper, Dann & Harrison, 2010). Ένας από τους λόγους είναι ότι οι μαθητές θεωρούν τον προγραμματισμό, ήδη από τα σχολικά έτη, ως μια βαρετή και επίπονη διαδικασία, με αποτέλεσμα η στάση τους αυτή να διαμορφώνει και τις μελλοντικές επαγγελματικές τους επιλογές (Cooper, 2010). Έρευνες δείχνουν ότι οι μαθητές κατά τη σχολική ηλικία δείχνουν ενδιαφέρον στον προγραμματισμό μόνο αν ασχοληθούν με αυτόν με έναν ελκυστικό τρόπο (Hylton & Otoupal, 2005).

Σε μια προσπάθεια να αυξηθεί το ενδιαφέρον των μαθητών στην επιστήμη των υπολογιστών, αρκετή προσπάθεια έχει γίνει στον τομέα ανάπτυξης εργαλείων και δραστηριοτήτων για τους αρχάριους προγραμματιστές όλων των εκπαιδευτικών βαθμίδων. Οι προσπάθειες αυτές περιλαμβάνουν τη πραγματοποίηση κιναισθητικών δραστηριοτήτων όπως το CS Unplugged, καθώς και την ανάπτυξη οπτικών περιβαλλόντων προγραμματισμού για αρχάριους (Initial Learning Environments - ILEs) όπως το Scratch, το Alice και πρόσφατα το App Inventor for Android (AIA), προκειμένου να μειώσουν επιτυχώς το φράγμα της πρώτης εισόδου στον

προγραμματισμό (Roy, 2012; Meerbaum-Salant, Armoni & Ben-Ari, 2013). Τα ILE's διευκολύνουν την ανάπτυξη λογισμικού εντός ενός πλαισίου το οποίο είναι διασκεδαστικό και δε λειτουργεί απωθητικά στον αρχάριο προγραμματιστή (Papadakis et al., 2014).

Στην παρούσα εργασία, με αφορμή το νέο βιβλίο πληροφορικής της Α' τάξης Γενικού και Επαγγελματικού Λυκείου στο οποίο προτείνονται για διδασκαλία τα ILE's Alice και AIA, επιχειρούμε μια περιγραφή και μια σύγκριση των δυο περιβαλλόντων, από παιδαγωγική και τεχνολογική άποψη, προκειμένου να επιλέξουμε το βέλτιστο για τη διδασκαλία των βασικών προγραμματιστικών εννοιών.

2. Οι δυσκολίες στην εκμάθηση του προγραμματισμού

Οι δυσκολίες που αντιμετωπίζουν οι αρχάριοι προγραμματιστές στην εκμάθηση του προγραμματισμού είναι επαρκώς τεκμηριωμένες και εντοπίζονται στη χρήση των μεταβλητών, στους βρόχους, στη χρήση των λογικών τελεστών και δομών ελέγχου, στη μεταβίβαση τιμών και στην εκτέλεση πολλών ταυτόχρονων υπολογισμών (Meerbaum-Salant et al., 2013). Οι Cooper, Dann & Pausch, (2003) σχολιάζουν ότι συχνά οι αρχάριοι προγραμματιστές παρότι αρχικά έλκονται από τον προγραμματισμό λόγω της πρότερης εμπειρίας τους με τα βιντεοπαιχνίδια και τις πολυμεσικές εφαρμογές, γρήγορα αποθαρρύνονται καθώς διαπιστώνουν ότι είναι εξαιρετικά δύσκολο και χρονοβόρο να δημιουργήσουν τις δικές τους εφαρμογές.

Ο Soloway (Soloway, 1986 παράθεση σε Cooper, 2010) επισημαίνει ότι η πραγματική δυσκολία για τους αρχάριους προγραμματιστές είναι ότι, όχι μόνο χρειάζεται να μάθουν πώς να σχεδιάσουν έναν αλγόριθμο για την επίλυση ενός προβλήματος, αλλά επίσης, χρειάζεται να μάθουν πώς να μετατρέψουν τα βήματα του αλγορίθμου σε ένα ολοκληρωμένο λειτουργικό πρόγραμμα (Cooper, 2010). Στα πλαίσια αυτά, η συνδυασμένη χρήση των γραφικών και της οπτικοποίησης αναγνωρίζεται ως ένα αποτελεσματικό εργαλείο διδασκαλίας στην επιστήμη των υπολογιστών προκειμένου να βοηθηθούν οι μαθητές να συνδέσουν αποτελεσματικά τον αλγόριθμο με τις προγραμματιστικές εντολές (Cooper et al., 2003).

3. Το προγραμματιστικό περιβάλλον Alice

Το Alice αναπτύχθηκε στο πανεπιστήμιο Carnegie Mellon στις ΗΠΑ, υπό τη διεύθυνση της Randy Pausch και είναι ελεύθερο για δωρεάν χρήση. Παιδαγωγικά, έλκει την καταγωγή του από το έργο του Papert με την γλώσσα Logo και από το έτερο προγραμματιστικό περιβάλλον Karel the Robot. Αντίστοιχα, όπως με τη χελώνα και το ρομπότ, στο Alice οι μαθητές βλέπουν τα προγράμματα τους να εκτελούνται σε κινούμενη μορφή παρατηρώντας άμεσα τον αντίκτυπο κάθε διαφορετικής εντολής στη ροή του προγράμματος (Cooper, 2010). Ωστόσο το Alice διαφοροποιείται από αντίστοιχα περιβάλλοντα δεδομένου ότι περιλαμβάνει μία μηχανή δημιουργίας τρισδιάστατων γραφικών (3D) και μια μεγάλη βιβλιοθήκη από κινούμενα τρισδιάστατα μοντέλα (Parsons & Haden, 2007).

Ο σκοπός τη δημιουργίας του Alice, ήταν η εισαγωγή των αρχάριων προγραμματιστών στον αντικειμενοστραφή προγραμματισμό μέσω της χρήσης οπτικών αντικειμένων σε ένα τρισδιάστατο περιβάλλον (Cooper, 2010). Τα αντικείμενα θα προσέρχονταν από τον καθημερινό κόσμο όπως δέντρα, ζώα, κ.α. Οι μαθητές κατά τη δημιουργία ενός εικονικού κόσμου μαθαίνουν τις βασικές προγραμματιστικές δομές μέσω μιας κατάστασης που κατά τη Randy Pausch ονομάζεται «head fake». Ως «head fake» ορίζεται η κατάσταση κατά την οποία οι μαθητές ενθαρρύνονται να μάθουν κάτι θεωρώντας ότι μαθαίνουν κάτι διαφορετικό (Dann & Cooper, 2009). Στο Alice, οι μαθητές δεν είναι υποχρεωμένοι να γράψουν καθόλου κώδικα διότι, μέσω της χρήσης του συντάκτη, μπορούν απλά να σύρουν και να αποθέσουν τις εντολές που απαιτούνται για τη λειτουργία του προγράμματος.

Η αρχική ομάδα – στόχος του Alice ήταν οι πρωτοετείς φοιτητές πανεπιστημιακών τμημάτων πληροφορικής και κατά συνέπεια ήταν απαραίτητο για τους δημιουργούς του να σκεφτούν την ομαλή μετάβαση των φοιτητών αυτών στην Java ή σε κάποια άλλη αντικειμενοστραφή γλώσσα προγραμματισμού. Στο Alice αυτό επιτυγχάνεται με δυο τρόπους. Καταρχήν το Alice περιέχει τις τυπικές προγραμματιστικές δομές (π.χ. αντικείμενα) και δεύτερον παρέχει τη σύνταξη των εντολών με ένα «Java like» συντακτικό, δίχως οι μαθητές να χρειάζεται να ασχοληθούν με συντακτικές δυστροπίες όπως τη χρήση ερωτηματικών. Η τρίτη έκδοση του Alice παρέχει πιο σαφή υποστήριξη για τη μετάβαση στη Java. Είναι δυνατόν ένα έργο στο Alice να χρησιμοποιηθεί ως μια βιβλιοθήκη στην Java, με την χρήση του NetBeans IDE (Dann, Cooper & Ericson, 2010).

4. Το προγραμματιστικό περιβάλλον App Inventor for Android

Αντίστοιχα με το Alice, το AIA προσφέρει ένα οπτικό περιβάλλον προγραμματισμού στο οποίο οι μαθητές μπορούν να δημιουργήσουν τις εφαρμογές τους, σύροντας και συνδέοντας πλακίδια εντολών ως κομμάτια ενός παζλ. Το προγραμματιστικό περιβάλλον είναι αρκετά πρόσφατο, καθώς δημιουργήθηκε ως ένα πειραματικό έργο στα εργαστήρια της εταιρείας Google (Google Labs), μόλις το 2009. Ωστόσο, σε αντίθεση με το περιβάλλον του Alice, στο AIA, οι μαθητές δε δημιουργούν εφαρμογές οι οποίες εκτελούνται στους συμβατικούς υπολογιστές αλλά σε έξυπνες φορητές συσκευές με λειτουργικό σύστημα Android (smartphones / tablets) (Zaranis, Kalogiannakis & Papadakis, 2013). Πορίσματα ερευνών δείχνουν πως οι μαθητές έχουν αυξημένα κίνητρα για τη χρήση νέων τεχνολογιών και ότι προσεκτικά σχεδιασμένες παιδαγωγικές δραστηριότητες με τη χρήση των έξυπνων φορητών συσκευών παρακινούν τους μαθητές να εμπλακούν με αυτές (Bradley & Holley, 2011). Ουσιαστικά, το AIA εκμεταλλεύεται τη ραγδαία απήχηση που έχουν οι φορητές συσκευές στον νεανικό πληθυσμό προκειμένου να κεντρίσει το ενδιαφέρον των μαθητών στον προγραμματισμό.

Το μεγαλύτερο πλεονέκτημα του AIA είναι ότι δεν απαιτεί ιδιαίτερες γνώσεις προγραμματισμού (Papadakis et al., 2014). Η διεπαφή προγραμματισμού με πλακίδια

θυμίζει έντονα το Scratch, το δημοφιλές εργαλείο προγραμματισμού του MIT. Στο AIA, όπως και στο Alice, είναι δυνατή η δημιουργία εφαρμογών κάνοντας χρήση της Java και του Android SDK μέσω του Java Bridge το οποίο επιτρέπει στους χρήστες να χρησιμοποιήσουν συστατικά μέρη του AIA σε προγράμματα της Java. Αυτό σημαίνει ότι ένας εκπαιδευτικός μπορεί να χρησιμοποιήσει το AIA προκειμένου να εισάγει τους μαθητές στην προγραμματιστική λογική με ένα απλό και ευχάριστο τρόπο και καθώς οι μαθητές κατακτούν τις βασικές προγραμματιστικές έννοιες να τους «μεταφέρει» άμεσα στον παραδοσιακό προγραμματισμό (Roy, 2012).

5. Σύγκριση AIA & Alice ως προς την παιδαγωγική τους χρήση

Σε αντίθεση με τη χρήση συμβατικών γλωσσών προγραμματισμού και την παραδοσιακή διδακτική προσέγγιση στην οποία τα παραδείγματα εκμάθησης είναι ξένα προς τα ενδιαφέροντα των μαθητών και τα δύο προς εξέταση ILE's κινητοποιούν τους μαθητές μέσω της συγγραφής προγραμμάτων τα οποία είναι σχετικά με τα ενδιαφέροντα τους. Για αυτό το λόγο, άλλωστε, και τα δυο περιβάλλοντα είναι εξίσου ελκυστικά τόσο στα δυο φύλλα (Cooper, 2010) όσο και στους «μαθητές υψηλού κινδύνου», δηλαδή στους μαθητές ή στους σπουδαστές οι οποίοι χαρακτηρίζονται από μικρή επιτυχία στη κατανόηση των μαθηματικών και έχουν μικρή εμπειρία στον προγραμματισμό (Dann et al., 2012).

Κοινός στόχος στον σχεδιασμό του Alice και του AIA είναι η παροχή ενός πλούσιου και ελκυστικού περιβάλλοντος στους αρχάριους προγραμματιστές (Cooper, 2010). Και οι δυο σχεδιαστικές ομάδες θεωρούν ότι τα συντακτικά λάθη (π.χ. η παράλειψη ενός ερωτηματικού) δεν είναι σημαντικά κατά την φάση εκμάθησης του προγραμματισμού, σε αντίθεση με τα σημασιολογικά λάθη (όπως λάθη στη χρήση των μεθόδων). Ως εκ τούτου, είναι σημαντικό για έναν αρχάριο προγραμματιστή να επικεντρωθεί στις σημασιολογικές δυσκολίες ενώ δεν είναι αναγκαίο να επικεντρωθεί και στην αντιμετώπιση των τυχαίων συντακτικών λαθών τα οποία πηγάζουν από την χρήση της εκάστοτε γλώσσας προγραμματισμού (Cooper, 2010). Και στα δυο περιβάλλοντα, ο ενσωματωμένος συντάκτης με δυνατότητες «drag and drop» αποτρέπει μια μεγάλη ποικιλία συντακτικών λαθών τα οποία κάνουν συχνά οι αρχάριοι προγραμματιστές (Cooper, 2010) προσανατολίζοντάς τους στην επίλυση προβλημάτων και λιγότερο στα ιδιαίτερα χαρακτηριστικά της εκάστοτε γλώσσας προγραμματισμού. Και τα δυο περιβάλλοντα υποστηρίζουν τη χρήση εντολών τόσο υψηλού όσο και χαμηλού επιπέδου όπως τη χρήση των επαναληπτικών βρόγχων. Επίσης, το αποτέλεσμα της εκτέλεσης εντολών είναι άμεσα ορατό ως προς τις μεταβολές του υπό εξέταση αντικειμένου. Επιπλέον, η μετάβαση από την πρωτοβάθμια στη δευτεροβάθμια ή/και στην τριτοβάθμια εκπαίδευση μπορεί να επιτευχθεί μέσω του σχήματος Scratch → Alice/AIA → Αντικειμενοστραφής γλώσσα προγραμματισμού.

Μια διαφοροποίηση μεταξύ των δυο περιβαλλόντων είναι ότι το Alice υπερτερεί του AIA, σε σχέση με τις ποικίλες δυνατότητες που διαθέτει αναφορικά με την

παρακολούθηση της εκτέλεσης ενός προγράμματος, οι οποίες σε αρκετές περιπτώσεις είναι ανώτερες από επαγγελματικές γλώσσες προγραμματισμού. Ενδεχόμενα το Alice παρέχει την καλύτερη υποστήριξη εντοπισμού σφαλμάτων από όλα τα ILE's (Cooper, 2010). Επιπρόσθετα, το Alice είναι το μοναδικό περιβάλλον το οποίο υποστηρίζει τη χρήση των 3D γραφικών. Η χρήση των 3D γραφικών είναι ιδανική για ορισμένους τύπους έργων όπως ιστορίες, παιχνίδια και εικονικούς κόσμους. Το ΑΙΑ επίσης, υποστηρίζει τη δημιουργία ιστοριών και παιχνιδιών αλλά δεν μπορεί να υποστηρίζει τη δημιουργία εικονικών κόσμων. Ωστόσο, άλλοι τύποι έργων όπως απλά παιχνίδια ή τεστ γνώσεων είναι ευκολότερο να υλοποιηθούν στον δισδιάστατο κόσμο. Είναι πολύ εύκολο για έναν μαθητή να εισάγει εικόνες δυο διαστάσεων παρά να δημιουργήσει ή να ψάξει να βρει τρισδιάστατα μοντέλα.

Παρότι και τα δυο περιβάλλοντα είναι ισοδύναμα με κριτήριο το βαθμό κάλυψης της διδακτέας ύλης της Α' Λυκείου, το ΑΙΑ θεωρείται ως ιδανικότερο εκπαιδευτικό περιβάλλον συγκριτικά με το Alice καθώς προσφέρει επιπλέον κίνητρα στους μαθητές, εξαιτίας της φορητότητας και της πρακτικής χρήσης των εφαρμογών που δημιουργούν. Στο Alice, οι μαθητές δημιουργούν εικονικά κινούμενα σχέδια, παιχνίδια και ιστορίες. Στο ΑΙΑ οι μαθητές δημιουργούν αρκετά έργα όπως και στο Alice αλλά επιπλέον έχουν τη δυνατότητα να δουν τις εφαρμογές να τρέχουν και στις φορητές τους συσκευές. Επιπλέον, η ύπαρξη προσομοιωτή προσδίδει ευελιξία στη διδασκαλία του ΑΙΑ, αφού δεν καθιστά υποχρεωτική τη χρήση φορητών συσκευών εντός του σχολικού εργαστηρίου πληροφορικής. Επιπρόσθετα η μετάβαση από το Alice στο ΑΙΑ είναι πολύ ομαλή, επειδή τα δυο περιβάλλοντα έχουν παρόμοια εμφάνιση και αίσθηση. Ως εκ τούτου, το ΑΙΑ θεωρείται ότι συμπληρώνει την υφιστάμενη εκπαιδευτική εμπειρία των χρηστών του Alice.

Ένα μειονέκτημα χρήσης του Alice είναι ότι το ιδιαίτερο γραφικό περιβάλλον του μπορεί να λειτουργήσει αντιπαραγωγικά όταν το Alice χρησιμοποιείται ως εργαλείο διδασκαλίας και όχι ως εργαλείο δημιουργίας animation (Parsons & Haden, 2007). Σε έρευνα τους, οι Parsons & Haden (2007), διαπίστωσαν ότι ορισμένοι μαθητές αφαιρέθηκαν με τα στοιχεία της διαδικασίας εικονικών κόσμων και απέτυχαν να αναγνωρίσουν τις δυνατότητες προγραμματισμού του Alice. Ένα αντίστοιχο παράδειγμα είναι η κοινότητα του Scratch. Παρότι υπάρχουν περισσότερα από 3 εκατομμύρια έργα διαθέσιμα έχει βρεθεί ότι στην πλειονότητα τους οι χρήστες του Scratch εστιάζουν κυρίως στο σχεδιασμό ή/και στην τροποποίηση των χαρακτήρων δίνοντας λιγότερη βαρύτητα στην ανάπτυξη των προγραμματιστικών τους ικανοτήτων. Για παράδειγμα, μόνο το 10% των έργων που έχει ανέβει στη δικτυακή κοινότητα του Scratch χρησιμοποιούν την προγραμματιστική δομή «repeat until (condition)» και μόνο το 20% των έργων κάνουν χρήση μεταβλητών (Meerbaum-Salant et al., 2013). Για ένα εκπαιδευτικό πληροφορικής είναι σημαντικό να διερευνήσει αν η χρήση τέτοιων περιβαλλόντων διευκολύνει την εκμάθηση των προγραμματιστικών εννοιών ή αν ο μαθητής αποπροσανατολίζεται στις πολυμεσικές δυνατότητες του περιβάλλοντος.

6. Σύγκριση AIA & Alice ως προς τις απαιτήσεις χρήσης

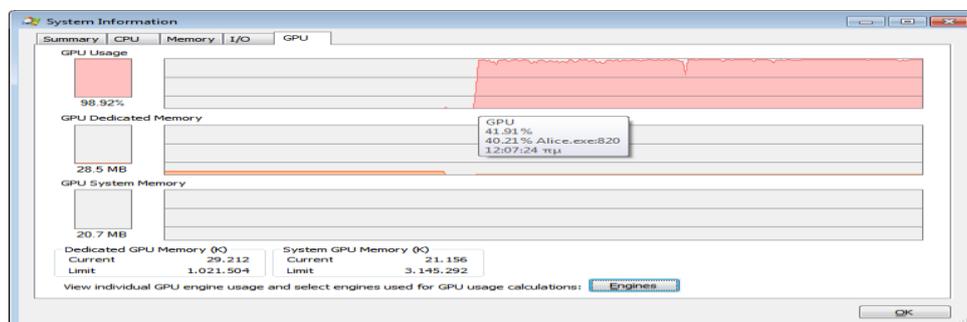
Τα δυο περιβάλλοντα λόγω της διαφορετικής τους φύσης διαφέρουν ως προς τις τεχνολογικές απαιτήσεις των συσκευών στις οποίες απευθύνονται. Για τη χρήση του Alice από πλευράς υλικού συνιστάται η χρήση ενός συμβατικού ηλεκτρονικού υπολογιστή. Αν και αρκετά netbooks έχουν σχεδιαστεί και δουλεύουν ικανοποιητικά με γραφικά 2D, δεν είναι σίγουρο ότι μπορούν να αποδώσουν γραφικά 3D. Ως προς τη μνήμη RAM, προτείνονται 2GB ή περισσότερα αν και το 1GB είναι το ελάχιστο. Η κάρτα γραφικών πρέπει να υποστηρίζει 32bit βάθος χρώματος με ελάχιστη ανάλυση τα 1024x768 pixels. Οι 3D κάρτες γραφικών προσφέρουν καλύτερα αποτελέσματα αλλά θεωρητικά δεν απαιτούνται. Το Alice 2.4 τρέχει σε όλες τις νεότερες εκδόσεις των γνωστών λειτουργικών συστημάτων.

Το AIA όπως αναφέρεται και στον επίσημο ιστότοπό του, τρέχει εξίσου στις νεότερες εκδόσεις των γνωστών λειτουργικών συστημάτων. Υποστηρίζονται όλοι οι γνωστοί φυλλομετρητές πλην του Microsoft Internet Explorer. Για τη δοκιμή των εφαρμογών που δημιουργούνται μέσω του AIA (αν δε χρησιμοποιηθεί ο προσομοιωτής) απαιτείται μία έξυπνη φορητή συσκευή με λειτουργικό σύστημα Android 2.3 ή νεότερο. Στον πίνακα 1 παρουσιάζονται συγκεντρωτικά οι απαιτήσεις σε υλικό και λογισμικό του Alice και του AIA.

Πίνακας 1: Απαιτήσεις σε υλικό-λογισμικό των Alice 2.4 και AIA

Προδιαγραφή	Alice 2.4	AIA
Μνήμη	>=1GB	-
Δίσκος	590MB	-
Κάρτα Γραφικών	Βάθος χρώματος 32bit Ανάλυση 1024x768	-
Λειτουργικό Σύστημα	Windows (>XP) Mac OS X (>10.6) Ubuntu ή Red Hat	Windows (>XP) Mac OS X (>10.5) Ubuntu (>8), Debian (>5)
Φυλλομετρητής	-	Mozilla Firefox (>3.6), Safari Apple (>5.0), Google Chrome (>4.0)
Φορητή Συσκευή	-	ΛΣ Android (>2.3)

Στην Εικόνα 1, παρουσιάζεται η επιβάρυνση ενός τυπικού υπολογιστικού συστήματος διατίας και ιδιαίτερα της κάρτας γραφικών από τη χρήση του Alice, όπως προκύπτει από την εφαρμογή Microsoft Process Explorer v16.05.



Εικόνα 1: Χρήση κάρτας γραφικών κατά την εκτέλεση του Alice

Είναι φανερό ότι η χρήση του ΑΙΑ αποτελεί τη βέλτιστη επιλογή, καθώς απαιτεί λιγότερους υπολογιστικούς πόρους. Στο περιβάλλον ενός παλαιωμένου σχολικού εργαστηρίου πληροφορικής η χρήση του Alice φαντάζει ουτοπική.

7. Συμπεράσματα - Συζήτηση

Η παρούσα εργασία ασχολείται με την συγκριτική επισκόπηση των δύο περιβαλλόντων. Σαφέστατα για την πληρέστερη αξιολόγησή τους απαραίτητη είναι η συστηματική καταγραφή της χρήσης τους στο ρεαλιστικό περιβάλλον της σχολικής τάξης. Εντούτοις, από παιδαγωγική και τεχνολογική σκοπιά, το ΑΙΑ σε σχέση με το Alice φαίνεται να αναδύεται ως ισχυρότερο προς χρήση περιβάλλον στους δυο τύπους Λυκείων. Οι σαφέστατα λιγότερες απαιτήσεις σε τεχνολογικούς πόρους του ΑΙΑ σε σχέση με το Alice αλλά και οι δυνατότητες εφαρμογής λύσεων BYOD, το καθιστούν ως ιδανική λύση για τα τεχνολογικά ανεπαρκή σχολικά εργαστήρια πληροφορικής. Από παιδαγωγική άποψη και τα δυο ΙΛΕ's έχουν τα δυνατά και αδύναμα σημεία τους. Το ΑΙΑ είναι πιο κατάλληλο για διδασκαλία σε προγράμματα σπουδών των οποίων ο πρωταρχικός στόχος είναι η ευχάριστη ενασχόληση και η εισαγωγική γνωριμία με τον προγραμματισμό. Το ΑΙΑ θεωρείται ιδανικό εργαλείο για την κινητοποίηση των ενδιαφερόντων των νέων μαθητών μέσω της σύνδεσης του μαθήματος με τις κινητές τους συσκευές. Δεν είναι τυχαίο ότι το ΑΙΑ θεωρείται ως το σωστό εργαλείο στο σωστό χρόνο, συμπληρώνοντας τη δίψα των χρηστών για φορητές συσκευές. Η παρούσα εργασία ευελπιστούμε να αποτελέσει ένα χρήσιμο οδηγό για τους εκπαιδευτικούς, οι οποίοι σχεδιάζουν εισαγωγικά μαθήματα και δραστηριότητες προγραμματισμού προσελκύοντας περισσότερους μαθητές στον κόσμο της πληροφορικής.

Βιβλιογραφία

Bradley, C., & Holley, D. (2011). Empirical research into students' mobile phones and their use for learning. *International Journal of Mobile and Blended Learning*, 3(4), 38-53.

Cooper, S. (2010). The design of Alice. *ACM Transactions on Computing Education*

Β. Δαγδύλης, Α. Λαδιάς, Κ. Μπίκος, Ε. Ντρενογιάννη, Μ. Τσιτουρίδου (επιμ.), Πρακτικά Εργασιών 4ου Πανελληνίου Συνεδρίου «Ένταξη των ΤΠΕ στην Εκπαιδευτική Διαδικασία» της Ελληνικής Επιστημονικής Ένωσης ΤΠΕ στην Εκπαίδευση (ΕΤΠΕ), Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης & Πανεπιστήμιο Μακεδονίας, Θεσσαλονίκη, 30 Οκτωβρίου – 1 Νοεμβρίου 2015

- (*TOCE*), 10(4), 15.
- Cooper, S., Dann, W., & Harrison, J. (2010). A K-12 College Partnership. In *Proceedings of the 41st SIGCSE Technical Symposium on Computer Science Education* (Milwaukee, WI, USA, March 10 - 13). SIGCSE '10. ACM, NY.
- Cooper, S., Dann, W., & Pausch, R. (2003). Using animated 3D graphics to prepare novices for CS. *Computer Science Education*, 13(1).
- Dann, W. & Cooper, S. (2009). Alice3: Concrete to abstract. *Communications of the ACM*, 52(8), 27-29.
- Dann, W., Cooper, S., & Ericson, B. (2010). *Exploring Wonderland: Java Programming Using Alice and Media Computation*. Prentice Hall.
- Dann, W., Cosgrove, D., Slater, D., Culyba, D., & Cooper, S. (2012). Mediated transfer: Alice 3 to Java. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12)*. ACM, NY, USA, 141-146.
- Hylton, P. & Otoupal, W. (2005). Preparing urban secondary school students for entry into engineering and technology programs. In *Proc. of IEEE Conference on Frontiers in Education*.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239-264.
- Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice Programming Environments. Scratch & App Inventor: a first comparison. In *Proceedings of the 2014 Workshop on Interaction Design in Educational Environments*. ACM.
- Parsons, D., & Haden, P. (2007). Programming osmosis: Knowledge transfer from imperative to visual programming environments. In *Conference of the National Advisory Committee on Computing Qualifications*.
- Roy, K. (2012). App inventor for android: report from a summer camp. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 283-288). ACM.
- Zaranis, N., Kalogiannakis, M., & Papadakis, S. (2013). Using Mobile Devices for Teaching Realistic Mathematics in Kindergarten Education. *Creative Education (Special Issue in Preschool Education)*, 4(7A1), 1-10