

Μια «Κούρσα στα είκοσι» με Python: Μία διδακτική πρόταση εμπέδωσης βασικών εννοιών της γλώσσας προγραμματισμού

Αν. Σαριδάκη¹, Μ. Αγγελάκη², Π. Μουτσέλου³, Θ. Ντούρου⁴, Ελ. Πλυτά⁵

¹2ο ΕΠΑΛ Νέας Σμύρνης, ²4ο Λύκειο Νέας Σμύρνης, ³2ο Γυμνάσιο Μοσχάτου,
⁴Λεόντειο Λύκειο Νέας Σμύρνης, ⁵διάθεση ΠΥΣΔΕ Δ' Αθήνας
{asaridaki, epiangelaki, giotamouts, th.ntourou, eplyta}@gmail.com

Περίληψη

Η παρούσα εργασία αποτελεί μία διδακτική πρόταση για την εμπέδωση ήδη διδαγμένων αλγοριθμικών δομών της γλώσσας προγραμματισμού Python, με χρήση βασικών τύπων δεδομένων (ακέραιου και συμβολοσειράς), συναρτήσεων απλής μορφής με έμφαση στη χρήση του ακέραιου υπόλοιπου. Προτείνεται ως άσκηση εμπέδωσης και προσεγγίζει το θέμα που πραγματεύεται, με βιωματικό τρόπο σε ένα αυθεντικό πλαίσιο μάθησης. Οι μαθητές αρχικά καλούνται να παίξουν με πραγματικές πίστες, ανά δύο, το παιχνίδι στρατηγικής που είναι γνωστό με τα ονόματα «Κούρσα στα είκοσι». Στη συνέχεια, κλιμακωτά, αναπτύσσουν προγράμματα περνώντας από τις φάσεις ανάλυσης, σχεδίασης, υλοποίησης, εφαρμογής και αξιολόγησης ενός προγράμματος.

Λέξεις κλειδιά: python, αυθεντική μάθηση, παιχνίδι στρατηγικής, αλγοριθμικές δομές, ακέραιο υπόλοιπο

1. Εισαγωγή

1.1 Γενικά για τη διδακτική πρόταση

Η διδακτική πρόταση αφορά στην ανάπτυξη ενός ολοκληρωμένου παιχνιδιού με python. Αρχικά οι μαθητές παίζουν το παιχνίδι στρατηγικής «Κούρσα στα είκοσι» σε φυσική πίστα και στη συνέχεια αναπτύσσουν την ψηφιακή εκδοχή του που προσομοιώνει τη λειτουργία του παιχνιδιού. Η επιλογή της εκπαιδευτικής τεχνικής “του παιχνιδιού” έγινε για να προκαλέσει την ενεργητική συμμετοχή και να αυξήσει το ενδιαφέρον των μαθητών για το αντικείμενο του “Προγραμματισμού”. Αποτελεί επιστημονική πρόκληση να αναπροσαρμοστούν οι μέθοδοι διδασκαλίας στις τάξεις της νέας γενιάς και να ενσωματώσουν παιχνίδια στην εκπαιδευτική διαδικασία (Μαλλιαράκης, Ξυνόγαλος & Σατρατζέμη, 2012). Η διδασκαλία του προγραμματισμού βοηθά τους μαθητές να αποκτήσουν αναλυτική και συνθετική σκέψη, να αναπτύξουν ικανότητες μεθοδολογικού χαρακτήρα προκειμένου να είναι σε

θέση να επιλύουν απλά προβλήματα σε προγραμματιστικό περιβάλλον, να καλλιεργήσουν δεξιότητες αλγοριθμικής σκέψης (ανάλυση προβλήματος, σχεδίαση αλγορίθμου, δομημένη σκέψη, αυστηρότητα έκφρασης) και επιδρά θετικά στη δημιουργικότητα και τη φαντασία τους (Δογούλη, 2012). Η υπολογιστική σκέψη, κατά την Wing, είναι μια γενική ανθρώπινη δεξιότητα, ισότιμη με τις παραδοσιακές δεξιότητες αλφαριθμητισμού (γραφή, ανάγνωση, αριθμητική) και επομένως θα πρέπει να καλλιεργείται στον μαθητικό πληθυσμό προς όφελος των ατόμων αλλά και των κοινωνικών ομάδων γενικότερα (Δημητριάδης, 2016). Ο προγραμματισμός λοιπόν, εκτός από καθαρά γνωστικό αντικείμενο, αποτελεί και γνωστικό εργαλείο με σημαντικά οφέλη για τους μαθητές (Papert, 1991).

1.2 Γιατί παιχνίδι;

Σύμφωνα με το θεωρητικό μοντέλο του Kolb (Kolb, 1984) (όπως αναφέρεται στη Δελούδη, 2002) η διαδικασία της βιωματικής μάθησης μπορεί να περιγραφεί ως ένας κύκλος τεσσάρων σταδίων, όπου κάθε μαθησιακή διαδικασία ξεκινάει από μία εμπειρία επάνω στην οποία οι εκπαιδευόμενοι, αφού τη ζήσουν, καλούνται να αναστοχαστούν και να πειραματιστούν. Από τα συμπεράσματα που θα βγουν, θα προκύψουν θεωρητικές αρχές και γενικεύσεις για το τι τελικά έχει επιτευχθεί ως μάθηση. Μέσα από αυτή τη διαδικασία, οι εκπαιδευόμενοι με τον πειραματισμό θα δοκιμάσουν ό,τι έμαθαν στην πράξη. Έτσι δημιουργείται μια νέα εμπειρία και, κατά συνέπεια, με την ολοκλήρωση ενός πρώτου κύκλου βιωματικής μάθησης, δίνεται το έναυσμα για ένα νέο κύκλο. Ο τρόπος που μαθαίνουν οι μαθητές σχετίζεται με δώδεκα ψυχολογικές αρχές: ενεργός συμμετοχή, κοινωνική αλληλεπίδραση, δραστηριότητες που έχουν νόημα, σύνδεση των νέων πληροφοριών με τις προϋπάρχουσες γνώσεις, χρήση στρατηγικών, ανάπτυξη της αυτορρύθμισης και του αναστοχασμού, αναδόμηση της προϋπάρχουσας γνώσης, στόχος η κατανόηση κι όχι η απομνημόνευση, βοήθεια για να μάθουν οι μαθητές να εφαρμόζουν τις γνώσεις τους, διάθεση χρόνου για εξάσκηση, αναπτυξιακές και ατομικές διαφορές και καλλιέργεια της μάθησης με κίνητρα. Οι αρχές αυτές ανακεφαλαιώνουν ορισμένα από τα σημαντικά πορίσματα της τρέχουσας έρευνας για τη μάθηση και αφορούν στην εκπαίδευση. Προσπαθούν να ενσωματώσουν έρευνες από διαφορετικές περιοχές της ψυχολογίας, μεταξύ των οποίων συγκαταλέγονται η εκπαιδευτική, η εξελικτική, η γνωστική, η κοινωνική και η κλινική ψυχολογία (Βοσνιαδου, 2011). Οι αρχές αυτές υπηρετούνται σχεδόν όλες με την ανάπτυξη, προγραμματιστικά, προσομοιώσεων παιχνιδιών. Το παιχνίδι αποτελεί ένα νέο διδακτικό εργαλείο μέσα στη τάξη, ένα ισχυρό μέσο στα χέρια του εκπαιδευτικού. Ο εκπαιδευτικός μπορεί να δειλέψει τους μαθητές του να ασχοληθούν με τον προγραμματισμό, προκειμένου να δημιουργήσουν τα δικά τους παιχνίδια. Η δημιουργία παιχνιδιών απαιτεί την εξάσκηση ανωτέρων μορφών σκέψης, αλλά και την έκφραση της δημιουργίας. Οι μαθητές μπορούν να εμπλακούν στη δημιουργία ψηφιακών παιχνιδιών, ως μέσα διασκέδασης, χαλάρωσης, δημιουργίας, μάθησης και απόκτησης βασικών δεξιοτή-

των προγραμματισμού, απαραίτητων για την καθημερινή τους ζωή. Με το παιχνίδι μπορούμε να προσεγγίσουμε ένα νέο ή να εμβαθύνουμε σε ένα ήδη γνωστό εκπαιδευτικό θέμα, ή να το χρησιμοποιήσουμε ως συμπληρωματική δραστηριότητα για να αυξήσουμε τα κίνητρα των μαθητών να ανακαλύψουν, να εμβαθύνουν, να κρίνουν, να αλληλεπιδράσουν, να δημιουργήσουν. Αποτελεί βιωματική μαθησιακή προσέγγιση. Η προσομοίωση της λειτουργίας των παραδοσιακών, επιτραπέζιων, παιχνιδιών μπορεί να γίνει με κωδικοποίηση σε γλώσσες προγραμματισμού οι οποίες διδάσκονται στη δευτεροβάθμια εκπαίδευση στη χώρα μας.

1.3 Γιατί Python;

Η Python αναπτύσσεται ως ανοιχτό λογισμικό μέσω του μη κερδοσκοπικού οργανισμού Python Software Foundation. Είναι μια γλώσσα υψηλού επιπέδου, γενικής χρήσης, εξαιρετικά δημοφιλής. Χαρακτηριστικά λέγεται για την Python ότι έχει «χαμηλό κατώφλι» δηλαδή είναι εύκολη στην εκμάθησή της, ενώ «δεν έχει ταβάνι» ως υψηλού επιπέδου επαγγελματικό εργαλείο που εξελίσσεται συνεχώς. Η απλότητα και η ευελιξία της Python ενθαρρύνει τους μαθητές να δημιουργήσουν κατανοητό κώδικα. Ο κατανοητός κώδικας είναι ευκολότερος στη συντήρηση και είναι λιγότερο πιθανό να περιέχει σφάλματα (επειδή πολλά σφάλματα προκαλούνται από την παρανόηση του τι στην πραγματικότητα ο κώδικας κάνει σε σύγκριση με τι ο χρήστης εσφαλμένα νομίζει ότι θα έπρεπε να κάνει). Το να είναι σε θέση οι μαθητές απλά να δηλώσουν τις ιδέες τους στον κώδικα είναι μια πολύ σημαντική δυνατότητα (Tollervey, 2015). Η Python, προσφέρεται σε μια σειρά μοντέλων διδασκαλίας για τη διαφοροποίηση της διδακτικής προσέγγισης ανάλογα με τις εκάστοτε ανάγκες και χαρακτηριστικά των μαθητών (Δημητριάδης, 2016). Από την εμπειρία με χρήστες και μαθητές της python, η γλώσσα παρέχει: ποιότητα λογισμικού, παραγωγικότητα ως προς την ανάπτυξη, φορητότητα των προγραμμάτων, βιβλιοθήκες υποστήριξης, ενσωμάτωση και ικανοποίηση που μπορεί να φέρει μεγάλη παραγωγικότητα στον προγραμματιστή (Lutz, 2013).

2. Η διδακτική πρόταση

2.1 Το παιχνίδι “Κούρσα στα είκοσι”

Το επιτραπέζιο παιχνίδι “Κούρσα στα είκοσι” ανήκει στα παιχνίδια στρατηγικής και αλγοριθμικής σκέψης (E.E.E, 2012). Είναι επιτραπέζιο και παίζεται από δύο παίκτες, με ένα πιόνι σε μία πίστα είκοσι θέσεων. Η επιλογή του πρώτου παίκτη γίνεται τυχαία. Υπάρχει μια διαδρομή με 20 θέσεις (Εικόνα 1) στις οποίες προχωράει το πιόνι προς τα εμπρός κατά 1 ή 2 βήματα. Τη μετακίνηση του πιονιού την κάνει ο κάθε ένας από τους δύο παίκτες με τη σειρά του, μέχρι να φτάσει στην 20η

θέση. Ο παίκτης που (μετακινώντας το πόνι) θα φτάσει πρώτος στην 20η θέση, κερδίζει το παιχνίδι.



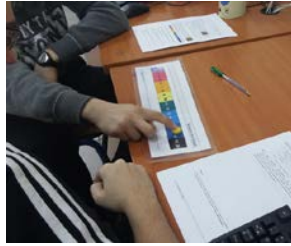
Εικόνα 1. Μία ενδεικτική πίστα του παιχνιδιού (E.E.E., 2012)

2.2 Γενικοί σκοποί της διδακτικής παρέμβασης

Ο πρώτος σκοπός είναι η ανάκληση και χρήση των διδαγμένων προγραμματιστικών εννοιών που αποσκοπεί στην εμπέδωση τους και στην απόκτηση ευχέρειας στη χρήση τους στη διαδικασία του προγραμματισμού. Ο δεύτερος σκοπός είναι η καλλιέργεια της “υπολογιστικής” σκέψης. Η ανάπτυξη του προγράμματος γίνεται σταδιακά, κλιμακωτά δημιουργώντας εκδοχές του προγράμματος. Οι έννοιες που ήδη έχουν διδαχθεί και αρκούν για την ολοκλήρωση μέχρι και το τελευταίο στάδιο είναι: η εισαγωγή δεδομένων από το πληκτρολόγιο (`input()`), η εκτύπωση στην οθόνη (`print()`), οι βασικοί τύποι δεδομένων (ακέραιος, λογικός, αλφαριθμητικός), οι αλγοριθμικές δομές της ακολουθίας, του ελέγχου (`if`) και της επανάληψης (`while`), η εισαγωγή βιβλιοθήκης (`import ...`), η χρήση έτοιμων συναρτήσεων `int()` και `str()`, η απλή μορφή συνάρτησης που δημιουργεί ο χρήστης (`def`) και δίνεται έμφαση στο ακέραιο υπόλοιπο.

2.3 Η ανάπτυξη του προγράμματος

Οι μαθητές αρχικά παίζουν το παιχνίδι, ανά δύο, σε φυσική μορφή (Εικόνα 2). Αυτό τους κεντρίζει το ενδιαφέρον, τους παρακινεί να ξαναπαίξουν προσέχοντας περισσότερο τις κινήσεις τους και κυρίως να αναστοχαστούν πάνω στην πορεία που ακολούθησαν. Αντιλαμβάνονται δηλαδή τον τρόπο παιχνιδιού (*gameplay*). Η έννοια *gameplay* (*game + play*) είναι μια ασαφής πλην όμως συχνά χρησιμοποιούμενη έννοια στον χώρο των παιχνιδιών, η οποία αναφέρεται σε οτιδήποτε σχετίζεται με τις εμπειρίες των χρηστών κατά την αλληλεπίδρασή τους με το παιχνίδι (αφήνοντας στην άκρη παράγοντες που σχετίζονται με την επένδυση του παιχνιδιού, όπως τα γραφικά και ο ήχος) (Δημητριάδης, 2015:192). Ο «τρόπος παιχνιδιού» συμπεριλαμβάνει επομένως τόσο τις δυνατότητες και τους περιορισμούς που θέτει το παιχνίδι, όσο και τη συμπεριφορά του χρήστη (τον τρόπο διάδρασης με το παιχνίδι που υιοθετεί ο χρήστης) σε σχέση με τα στοιχεία αυτά. Αφού οι μαθητές έχουν κατανοήσει τους κανόνες του παιχνιδιού και ενδεχομένως έχουν αρχίσει να προβληματίζονται για την ανάπτυξη στρατηγικών, καλούνται να αναπτύξουν τις ακόλουθες εκδοχές του προγράμματος.



Εικόνα 2. Οι μαθητές παίζουν το παιχνίδι

Α εκδοχή: Παίκτης εναντίον παίκτη. Γίνεται προσομοίωση της λειτουργίας του παιχνιδιού. Συμμετέχουν δύο πραγματικοί παίκτες. Το πρόγραμμα επιλέγει τυχαία τον ένα από τους δύο παίκτες (A ή B) που θα ξεκινήσει με χρήση της κατάλληλης συνάρτησης από την βιβλιοθήκη random της python. Στη συνέχεια, το πρόγραμμα δίνει τη σειρά, εναλλάξ, στους δύο παίκτες με κλήση συνάρτησης (Εικόνα 3α) που έχει δημιουργήσει ο προγραμματιστής περνώντας της ως παράμετρο τον παίκτη που μόλις έπαιξε. Κάθε παίκτης, με τη σειρά του, πληκτρολογεί πόσες θέσεις (1 ή 2) επιθυμεί να προχωρήσει το πόνι. Το πρόγραμμα ελέγχει την εφικτότητα της ζητούμενης κίνησης και ενημερώνει τον παίκτη εάν υπάρχει πρόβλημα. Κάθε φορά, μετά την μετακίνηση του πιονιού το πρόγραμμα εμφανίζει στην οθόνη μια μιμησιαστική οπτική αναπαράσταση της υφιστάμενης κατάστασης, για να μπορεί να αποφασίσει ο επόμενος παίκτης πόσες θέσεις να προχωρήσει το πόνι.

<pre>def next(p): if p == 1: return 2 else: return 1 def next(p): return (p % 2) + 1</pre> <p style="text-align: center;">(α)</p>	<p>Παίζει ο επόμενος παίκτης, δηλαδή A</p> <p>Πόσο θα προχωρήσεις 1 ή 2 βήματα παίκτη A: 1 -----3-----</p> <p>Παίζει ο επόμενος παίκτης, δηλαδή B</p> <p>Πόσο θα προχωρήσεις 1 ή 2 βήματα παίκτη B: 2 -----5-----</p> <p style="text-align: center;">(β)</p>	<pre>def showGrid(pos): grid = ((pos - 1) * "-" + str(pos) + (20 - pos) * "-") print(grid) return</pre> <p style="text-align: center;">(γ)</p>
--	--	--

Εικόνα 3. (α) Δύο εκδοχές κώδικα μαθητών για τη συνάρτηση του παίκτη που έχει σειρά (β) Κώδικας μαθητή για τη συνάρτηση εμφάνισης της οπτικής αναπαράστασης. (γ) Στιγμιότυπο από την εκτέλεση του προγράμματος, οπτική αναπαράσταση της πίστας και της θέσης του πιονιού.

Ο τρόπος υλοποίησης της οπτικής αναπαράστασης είναι αποτέλεσμα διαπραγματεύσεων. Μπορεί να είναι μόνο ένας αριθμός που θα δείχνει τη θέση που βρίσκεται το πόνι, ή ένα σύμβολο π.χ. "*" στη θέση του πιονιού ανάμεσα σε παύλες ή ο συνδυασμός των δύο (Εικόνα 3β). Η εμφάνιση γίνεται με χρήση συνάρτησης (Εικόνα 3γ) που ορίζει ο χρήστης και η οποία έχει ως παράμετρο τη θέση που βρίσκεται το πόνι. Η διαδικασία επαναλαμβάνεται μέχρι κάποιος παίκτης να φτάσει στην 20ή θέση, οπότε και κερδίζει. Στο τέλος ενημερώνονται οι παίκτες για το τελικό αποτέλεσμα με μήνυμα στην οθόνη. Στο περιβάλλον του διαδραστικού φλοιού (shell) της python φαίνεται το "τρέξιμο" του προγράμματος και η αλληλεπίδραση με την χρήστη. Έτσι, κατά την εκτέλεση του προγράμματος οι παίκτες έχουν τη

δυνατότητα να βλέπουν τις κινήσεις τους κατά τη διάρκεια του παιχνιδιού, να τις μελετήσουν και να αναστοχαστούν πάνω στον τρόπο που έπαιζαν. (Βελτίωση: Κατά την έναρξη του παιχνιδιού να εισάγουν οι παίκτες τα ονόματα τους από το πληκτρολόγιο, οπότε κάθε φορά που είναι η σειρά κάποιου παίκτη να τον προτρέπει το πρόγραμμα με το όνομα του αντί Α ή Β.)

Β εκδοχή: Παίκτης εναντίον υπολογιστή. Το πρόγραμμα λειτουργεί όπως στην Α εκδοχή, αλλά ο ένας παίκτης είναι εικονικός δηλαδή ο υπολογιστής. Όταν είναι η σειρά του να παίξει, θα προχωρήσει το πιόνι του τυχαία 1 ή 2 (αρκεί να επιτρέπεται) θέσεις προς τα εμπρός και θα εμφανιστεί στην οθόνη η τρέχουσα κατάσταση. Ο παίκτης-υπολογιστής (πρόγραμμα) παίζει τυχαία, χωρίς στρατηγική.

Γ εκδοχή: Ο αλγόριθμος με στρατηγική νίκης. Οι μαθητές, αφού παίζουν μερικές φορές το παιχνίδι και προσπαθήσουν να καταλάβουν τον αλγόριθμο που τους οδηγεί στη νίκη, θα διαπιστώσουν ότι κερδίζει ο παίκτης που θα φτάσει πρώτος στη θέση 17. Εάν προχωρήσουν αυτή τη σκέψη μέχρι το ξεκίνημα, θα ανακαλύψουν ότι οι “στρατηγικές θέσεις νίκης” είναι στις θέσεις 2, 5, 8, 11, 14, 17. Το ποιός παίκτης θα παίξει πρώτος είναι τυχαίο. Εάν λοιπόν ένας παίκτης παίζει πρώτος, μπορεί να κερδίζει πάντα. Εάν παίζει δεύτερος πρέπει να εκμεταλλευτεί κάποιο λάθος του αντιπάλου και να καταλάβει μια στρατηγική θέση, για να κερδίσει σίγουρα. Στην άσκηση αυτή καλούνται οι μαθητές να αναπτύξουν ένα πρόγραμμα στο οποίο ο υπολογιστής να κερδίζει τις περισσότερες φορές, είτε επειδή ξεκίνησε πρώτος, είτε επειδή αξιοποίησε τη λάθος κίνηση του αντιπάλου. Αυτές οι “στρατηγικές θέσεις” έχουν το ίδιο ακέραιο υπόλοιπο με την τελική θέση, το 20, όταν διαιρούνται με το τρία (3) που είναι η μέγιστη απόσταση που πρέπει να απέχει το πιόνι από το τέλος για να κερδίσει κάποιος σε δύο κινήσεις (μία του συμπαίκτη του και μία η δική του).

Δ εκδοχή: Παραμετροποίηση και επέκταση του αλγόριθμου, “Κούρσα στα Ν”.

Μετά την ολοκλήρωση των προηγούμενων εκδοχών ζητείται από τους μαθητές να παραμετροποιήσουν τον αριθμό των θέσεων της πίστας, χρησιμοποιώντας σε όλα τα σημεία το προγράμματος αντί για τη σταθερά 20, μία μεταβλητή, π.χ. $N=20$. Στη συνέχεια τους ζητείται να παίξουν πάλι το παιχνίδι, να διερευνήσουν εάν οι “στρατηγικές θέσεις νίκης” εξακολουθούν να είναι οι ίδιες και να επέμβουν στα σημεία του αλγόριθμου που επηρεάζεται από αυτές.

Οι παραπάνω εκδοχές επιδέχονται εναλλακτικές λύσεις σε επιμέρους σημεία (π.χ. η συνάρτηση που επιστρέφει τον επόμενο παίκτη (Εικόνα 3α)), παραλλαγές (π.χ. στην τυχαία επιλογή του πρώτου παίκτη), τροποποιήσεις και βελτιώσεις (π.χ. πως θα αποκαλούνται οι παίκτες). Ενδιαφέρον παρουσιάζει η αναπαράσταση της θέσης που βρίσκεται το πιόνι και η εκτύπωση της στην οθόνη, σε περιβάλλον τερματικού (όχι γραφικό) (Εικόνα 3β). Αυτό επιτυγχάνεται με τις προγραμματιστικές ιδιαιτερότητες της γλώσσας προγραμματισμού *python*. Στην Γ εκδοχή, για τον εντοπισμό των “στρατηγικών θέσεων νίκης” χρησιμοποιήθηκε το ακέραιο υπόλοιπο της θέ-

σης που βρίσκεται το πόνι με το 3. Στη Δ εκδοχή, αξιοποιήθηκε πάλι το ακέραιο υπόλοιπο του αριθμού των θέσεων N με το 3 (Εικόνα 4).

```
def metniki(pioni):
    if arithmos % 3 == 2:
        if (pioni + 1) % 3 == 2:
            pioni = pioni + 1
            print("Το πόνι βρίσκεται στη θέση ", pioni)
        elif (pioni + 2) % 3 == 2:
            pioni = pioni + 2
            print("Το πόνι βρίσκεται στη θέση ", pioni)
        else:
            pioni = pioni + random.randint(1,2)
            print("Το πόνι βρίσκεται στη θέση ", pioni)
    elif arithmos % 3 == 1:
        if (pioni + 1) % 3 == 1:
            pioni = pioni + 1
            print("Το πόνι βρίσκεται στη θέση ", pioni)
        elif (pioni + 2) % 3 == 1:
            pioni = pioni + 2
            print("Το πόνι βρίσκεται στη θέση ", pioni)
        else:
            pioni = pioni + random.randint(1,2)
            print("Το πόνι βρίσκεται στη θέση ", pioni)
    else:
        if (pioni + 1) % 3 == 0:
            pioni = pioni + 1
            print("Το πόνι βρίσκεται στη θέση ", pioni)
        elif (pioni + 2) % 3 == 0:
            pioni = pioni + 2
            print("Το πόνι βρίσκεται στη θέση ", pioni)
        else:
            pioni = pioni + random.randint(1,2)
            print("Το πόνι βρίσκεται στη θέση ", pioni)
    return pioni
```

Εικόνα 4. Κώδικας μαθητή. Χρήση «ακέραιου υπόλοιπου»

3. Η εφαρμογή της διδακτικής προσέγγισης

3.1 Πλαίσιο εφαρμογής

Η παραπάνω προσέγγιση έχει υλοποιηθεί, ως ένα τμήμα ενός ανοιχτού CodeClub (py4hs, 2017) κατά το σχολικό έτος 2016-2017. Το CodeClub απευθύνονταν σε μαθητές Γ Γυμνασίου, Α και Β ΓΕΛ και ΕΠΑΛ της Δ' Αθήνας και συμμετείχαν 11 μαθητές. Οι συμμετέχοντες ήταν όλα αγόρια και στην πλειονότητά τους είχαν προηγούμενες γνώσεις προγραμματισμού όπως συνάγεται από ένα αρχικό ερωτηματολόγιο που συμπλήρωσαν για το προφίλ τους.

Πίνακας 1. Το προφίλ μαθητών του CodeClub που συμμετείχαν στην 4^η συνάντηση

Πλήθος μαθητών	Τάξη	Γνώσεις προγραμματισμού
1	Β Γυμνασίου	Καλές
2	Γ Γυμνασίου	Πολύ καλές
2	Β ΕΠΑΛ, όχι Πληροφορικής	Όχι
3	Α Λυκείου	Πολύ καλές

Το CodeClub είχε διάρκεια 5 διώρων. Για κάθε δίωρο είχε δημιουργηθεί εκτενές φύλλο εργασίας. Η διδακτική παρέμβαση που παρουσιάζεται, χρησιμοποιήθηκε στο 4^ο δίωρο και οι μαθητές έπρεπε να φτάσουν μέχρι την Γ εκδοχή. Η Δ εκδοχή συμπεριλαμβάνονταν στις επιπλέον δραστηριότητες που διανεμήθηκαν, ως επέ-

κταση του φύλλου εργασίας. Το δείγμα ήταν μικρό, 11 μαθητές από τους οποίους οι 8 προσήλθαν στο 4^ο δάωρο / συνάντηση (Πίνακας 1).

3.2 Στόχοι

Γενικός σκοπός της λειτουργίας του CodeClub ήταν να αναπτύξει στους μαθητές θετική στάση ως προς τον προγραμματισμό. Υπήρχαν όμως επιμέρους στόχοι για κάθε συνάντηση. Οι στόχοι της 4ης συνάντησης που σχετίζεται με τη διδακτική παρέμβαση «Κούρσα στα 20» ήταν να μπορέσουν θα μαθητές να αναπτύξουν κλιμακωτά λύσεις που να αντιστοιχούν στις εκδοχές Α, Β και Γ όπως περιγράφονται στην ενότητα 2.3. Πιο αναλυτικά οι στόχοι αυτοί αντιστοιχούν:

- Επιτυχημένη λύση στην Α εκδοχή: Αποτύπωση του αλγόριθμου του παιχνιδιού, μετατροπή σε κώδικα python (επιλογή των κατάλληλων εντολών, συναρτήσεων, δομών κ.λπ.), έλεγχος σωστής λειτουργίας
- Επιτυχημένη λύση στην Β εκδοχή: τροποποίηση κώδικα με παρέμβαση στα κατάλληλα σημεία
- Επιτυχημένη λύση στην Γ εκδοχή: εντοπισμός στρατηγικών θέσεων αλγοριθμικά (όχι λίστα θέσεων), τροποποίηση κώδικα

Η επιτυχημένη λύση στην Α εκδοχή, σχετίζεται με τον πρώτο σκοπό της διδακτικής παρέμβασης δηλαδή στην ανάκληση και χρήση διδαγμένων εννοιών (ενότητα 2.2). Οι επιτυχημένες λύσεις στην Β και Γ εκδοχή σχετίζεται με τον πρώτο σκοπό δηλαδή την ανάπτυξη δεξιοτήτων επίλυσης προβλήματος και συγγραφής κώδικα που συντελεί στην καλλιέργεια του κριτικού τρόπου σκέψης, στην ανάπτυξη μεθοδικού τρόπου εργασίας, στην εξερεύνηση νέων ιδεών και στη βελτίωση της δημιουργικότητας, τόσο για την εργασία όσο και για την επικοινωνία (Αράπογλου κ.ά., 2015α).

3.3 Ευρήματα - Συζήτηση

Τα ευρήματα σχετικά με τους στόχους της 4ης συνάντησης φαίνονται στον Πίνακα 2. Οι μαθητές που έμειναν μόνο στην Α εκδοχή ήταν αυτοί που δεν είχαν προηγούμενες γνώσεις προγραμματισμού, γενικά προχωρούσαν με πιο αργό ρυθμό αλλά είχαν το ίδιο διαρκές ενδιαφέρον με όλους τους άλλους για τη διαδικασία και από τις εκδηλώσεις τους ήταν φανερό ότι ανέπτυξαν θετική στάση προς το προγραμματισμό. Οι περισσότεροι μαθητές (5/8) ολοκλήρωσαν τους στόχους της συνάντησης ενώ είναι αξιοσημείωτο ότι υπήρξε μαθητής που έφτασε μέχρι και τη Δ εκδοχή / γενίκευση (Εικόνα 4) που ξεπερνά τους στόχους της 4ης συνάντησης. Γενικά παρατηρήθηκε ότι ο βαθμός επιτυχίας δείχνει να σχετίζεται με το επίπεδο των γνώσεων προγραμματισμού που είχαν δηλώσει ότι είχαν.

Πίνακας 2. Συνοπτικά ευρήματα από την 4^η συνάντηση

Επιτυχημένη εκδοχή (στόχος)	Πλήθος μαθητών	Τάξη	Πρότερες γνώσεις προγραμματισμού
A εκδοχή	2	B ΕΠΑΛ, όχι Τομέα Πληροφορικής	Όχι
B εκδοχή	1	B Γυμνασίου	Καλές
Γ εκδοχή	4	Γ Γυμνασίου (2) + Α Λυκείου (2)	Πολύ καλές
Δ εκδοχή	1	A Λυκείου	Πολύ καλές

Οι δυσκολίες ως προς την ανομοιογένεια των μαθητών στην υλοποίηση (π.χ. τις πρότερες γνώσεις προγραμματισμού, το προσωπικό ενδιαφέρον, το διαθέσιμο χρόνο κ.λπ.) είχε ληφθεί υπόψη στον τρόπο που είχε αναπτυχθεί η πρόταση και είχε δημιουργηθεί το φύλλο εργασίας. Ήταν αρκετά καθοδηγητικό για τους αρχάριους ή/και αδύναμους μαθητές και προκαλούσε το ενδιαφέρον στους προχωρημένους που μπορούσαν να εξελίξουν τις ιδέες και τις σκέψεις τους με τις επιπλέον εκδόσεις, συνδεδεμένο με την πραγματικότητα, καθώς αφορά παιχνίδι.

Στο πλαίσιο του CodeClub είχαν προηγηθεί άλλα 3 φύλλα εργασίας που κάλυπταν σταδιακά τις έννοιες που αναφέρουμε στην παράγραφο 3 ως ήδη διδαγμένες. Η παρατήρηση αυτή, ενισχύει την άποψη ότι η διδακτική παρέμβαση που περιγράφεται μπορεί να χρησιμοποιηθεί ως επαναληπτική άσκηση στην τυπική εκπαίδευση, στον Τομέα Πληροφορικής των ΕΠΑΛ, χωρίς αυτό να σημαίνει ότι πρέπει να χρησιμοποιηθεί το ίδιο φύλλο εργασίας.

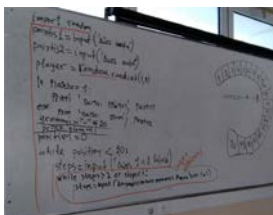
4. Μεταφορά της διδακτικής παρέμβασης στην τάξη

Από την εμπειρία του CodeClub σε συνδυασμό με ότι τα εκπαιδευτικά παιχνίδια είναι μια τεχνική που είναι εφικτό να χρησιμοποιηθεί στη δευτεροβάθμια εκπαίδευση ιδιαίτερα στα επαναληπτικά μαθήματα (Ρώσσιου, Παπαδάκης & Παπαμήτσιου, 2007) και παρατηρώντας ότι επιτεύχθηκαν οι στόχοι από ένα μέρος των μαθητών που συμμετείχαν στο CodeClub, δηλαδή δεν είναι ανεφάρμοστη η πρόταση, εκτιμούμε ότι θα είχε ενδιαφέρον να δοκιμαστεί και σε συνθήκες τάξης στην τυπική εκπαίδευση. Προτείνεται με την παρούσα εργασία ότι η υλοποίηση του παιχνιδιού, στο πλαίσιο της τυπικής εκπαίδευσης μπορεί να γίνει στον Τομέα Πληροφορικής του ΕΠΑΛ.

Από το 2015-2016, στον Τομέα Πληροφορικής των ΕΠΑΛ έχουν εισαχθεί τα μαθήματα “Αρχές Προγραμματισμού Υπολογιστών” στη Β και “Προγραμματισμός Υπολογιστών” στη Γ τάξη στα οποία χρησιμοποιείται η γλώσσα προγραμματισμού Python (Υ.ΠΑΙ.Θ., 2015). Η εφαρμογή της πρότασης μπορεί να γίνει σε δύο περιπτώσεις: α) ως επαναληπτική άσκηση για τη Β΄ τάξη όταν έχει καλυφθεί η διδακτέα ύλη μέχρι τις αρχές του 5^{ου} κεφαλαίου του σχολικού βιβλίου της Β΄ τάξης (πριν τη δομή δεδομένων λίστα) (Αράπογλου κ.ά., 2015β) και β) για τη Γ΄ τάξη

κατά την έναρξη της σχολικής χρονιάς ως εργαστηριακή άσκηση επανάληψης και σύνδεσης με την ύλη της προηγούμενης τάξης, για τα κεφάλαια 3 και 4 του σχολικού βιβλίου της Γ΄ τάξης (Αράπογλου κ.ά., 2016). Η προτεινόμενη διάρκειά της είναι δύο με τρία εργαστηριακά, διδακτικά δίωρα.

Όλοι οι μαθητές/τριες θα έχουν διδαχθεί την ίδια ύλη με βάση το ωρολόγιο πρόγραμμα, χωρίς να βασιζόμαστε στις πρότερες γνώσεις προγραμματισμού που τυχόν θα έχουν. Εκτιμάμε ότι όπως έχει σχεδιαστεί η διδακτική παρέμβαση (κλιμακωτά) οι μαθητές/τριες μπορούν να προχωρήσουν με το ρυθμό τους και να κατακτήσουν τουλάχιστον το Α επίπεδο σε αναλογία με τους μαθητές του CodeClub που δεν είχαν καμιά προηγούμενη εμπειρία στον προγραμματισμό, ενώ μπορούμε να περιμένουμε ότι ικανός αριθμός θα προχωρήσει στη Β και Γ εκδοχή.



Εικόνα 5. Στιγμιότυπο από την εφαρμογή στην τυπική τάξη

Στην εικόνα 5 φαίνεται ένα στιγμιότυπο από την εφαρμογή, ως άσκηση επανάληψης στα πρώτα μαθήματα της Γ τάξης το σχολικό έτος 2017-2018 στο 2^ο ΕΠΑΛ Νέας Σμύρνης. Στη φάση που φαίνεται στο στιγμιότυπο, οι μαθητές κωδικοποιούν σταδιακά τον αλγόριθμο και σε επόμενη φάση θα μετατρέψουν τμήματα του κώδικα που αντιστοιχούν σε λειτουργίες σε συναρτήσεις.

Αναφορές

Kolb D. (1984). *Experiential Learning*. New Jersey: Pentice Hall

Lutz M. (2013). *Learning Python (5th Edition)*, O'Reilly Media

Papert S. (1991). *Νοητικές Θύελλες*, Αθήνα: Εκδόσεις *Οδυσσέας*

py4hs (2017). Επιμορφωτικό πρόγραμμα <http://www.music.tuc.gr/projects/py4hs/>

Tollervey H. N. (2015). *Python in Education*. O'Reilly Media

Αράπογλου Α., Βραχνός Ε., Κανίδης Ε., Μακρυγιάννης Π., Μπελεσιώτης Β., Τζήμας Δ. (2015α). Οδηγίες για τον εκπαιδευτικό για τα μαθ/τα «Αρχές Προγραμματισμού Υπολογιστών» & «Προγραμματισμός Υπολογιστών», ΥΠΙΑΘ - ΙΕΠ

Αράπογλου Α., Βραχνός Ε., Κανίδης Ε., Μακρυγιάννης Π., Μπελεσιώτης Β., Τζήμας Δ. (2015β). «Αρχές Προγραμματισμού Υπολογιστών» Β & Γ ΕΠΑΛ, Τομέα Πληροφορικής, ΙΤΥΕ - Εκδόσεις «Διόφαντος»

Αράπογλου Α., Βραχνός Ε., Κανίδης Ε., Λέκκα Δ., Μακρυγιάννης Π., Μπελεσιώτης Β., Παπαδάκης Σπ., Τζήμας Δ. (2016). «Προγραμματισμός Υπολογιστών», Γ ΕΠΑΛ, Τομέα Πληροφορικής, ΙΤΥΕ - Εκδόσεις «Διόφαντος»

Βοσνιάδου Σ. (2001). Πώς μαθαίνουν οι μαθητές, *UNESCO: Διεθνής Ακαδημία της Εκπαίδευσης, Διεθνές Γραφείο Εκπαίδευσης*

Δελούδη Μ. (2002). Βιωματική μάθηση-Δυνατότητες αξιοποίησής της στο πλαίσιο της Ευέλικτης Ζώνης. *Επιθεώρηση Εκπαιδευτικών Θεμάτων*, 6, 145-159, Π.Ι.

Δημητριάδης Σ. (2015). Θεωρίες Μάθησης & Εκπαιδευτικό Λογισμικό, *Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα*

Δημητριάδης Σ. (2016). Η Python και η «τέχνη του υπολογίζεин»: πρόταση για ένα λεξικό μοντέλων διδασκαλίας της γλώσσας, *8ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»*, Ιωάννινα, 107-114

Δογούλη Ε. (2012). Ανάπτυξη εκπαιδευτικού λογισμικού για τη διδασκαλία των βασικών εννοιών του προγραμματισμού στις τελευταίες τάξεις του δημοτικού σχολείου, *Μεταπτυχιακή διπλωματική εργασία, ΠΜΣ, Κατ/νση: «Σχολική παιδαγωγική και νέες τεχνολογίες»*, ΠΤΝ Πανεπιστήμιο Δυτ. Μακεδονίας, Φλώρινα

Ε.Ε.Ε. (Εθνική Εστία Επιστημών) (2012). Ενημερωτικό φυλλάδιο εκθεμάτων, *Τ-δρυμα Νεότητας και Δια Βίου Μάθησης*

Μαλλιάρικης Χ., Ξυνόγαλος Σ., Σατρατζέμη Μ. (2012). Εκπαιδευτικά Παιχνίδια για την εκμάθηση του Προγραμματισμού, *8ο Πανελλήνιο Συνέδριο με διεθνή συμμετοχή «Τεχνολογίες της Πληροφορίας & Επικοινωνίας στην Εκπαίδευση»*, Βόλος, 471-478

Ρώσσιου Ε., Παπαδάκης Σ., Παπαμήτσιου Ζ. (2007). Μαθαίνουμε Παίζοντας: Το Εκπαιδευτικό Παιχνίδι στη Διδασκαλία της Πληροφορικής στο Γυμνάσιο, *4ο Συνέδριο ΤΠΕ στην Εκπαίδευση*, Σύρος, 364-373

Υ.ΠΑΙ.Θ (2015). Α.Π.Σ. των μαθημάτων Τομέα Πληροφορικής ΕΠΑΛ. ΦΕΚ 2010Β/16-9-2015

Abstract

This paper is a teaching proposal for embedding already-learned algorithmic structures of the Python programming language, using basic data types (integer and string), simple functions and emphasis on the use of the integral division reminder. It is proposed as an empowerment exercise and approaches the subject that it deals with, in an experiential way, in an authentic learning context. Students are initially invited to play on real slots, two at a time, the strategy game known as "Race to Twenty". Then, step by step, they develop programs through the design, implementation, and evaluation phases of a program.

Keywords: python, authentic learning, strategy game, algorithmic structures, integral division reminder