

# “Core Code”: η κωδικοποίηση και οπτικοποίηση της κίνησης ρομποτικών κατασκευών με προγραμματιστικές δομές και έλεγχο συμβάντων

Ιωάννης Κωτσάνης <sup>1</sup>, Αναστάσιος Λαδιάς <sup>2</sup>

[kotsanis@doukas.gr](mailto:kotsanis@doukas.gr), [ladiastas@gmail.com](mailto:ladiastas@gmail.com)

<sup>1</sup>Δρ. Μηχ. Πληροφορικής - Εκπαιδευτήρια Δούκα, <sup>2</sup>Δρ. Πληροφορικής

## Περίληψη

Διατρέχοντας τη διαχρονική πενήνταετή εκπαιδευτική ιδέα, της κίνησης ρομποτικών κατασκευών σε επίπεδες επιφάνειες, με μία ποικιλία από διαφορετικές αποστολές, η εισήγηση αυτή παρουσιάζει μία μοντελοποιημένη προσέγγιση βασισμένη σε 3 άξονες: (α) οπτικοποιημένη παρουσίαση των βασικών προγραμματιστικών δομών (ακολουθίας - επιλογής - επανάληψης) που χρησιμοποιούνται σε περιβάλλοντα προγραμματισμού με συμβάντα, (β) κωδικοποιημένες πληροφορίες σε ένα διαδιάστατο πίνακα, το "ΚωδικΌραμα", σχετικές με την ανατομία και τη λειτουργικότητα του προγράμματος, τόσο για τον δημιουργό του, όσο και για τον μαθητή ως προς την ενεργό εμπλοκή του στον προγραμματισμό, (γ) παιγνιώδεις διαδραστικές ψηφιακές δραστηριότητες κλιμακούμενης δυσκολίας, ώστε οι μαθητές να ανακαλύψουν πληροφορίες που είναι κρυμμένες στο "ΚωδικΌραμα" ή να δημιουργήσουν προτεινόμενες ή δικές τους αποστολές.

Βασικός σκοπός της προσέγγισης αυτής είναι να συγκεραστούν παράγοντες όπως η λειτουργικότητα του αυθεντικού προβλήματος-παραδείγματος, η απλότητα του κώδικα και η διδακτική πληρότητα του περιεχομένου του, με βάση δύο ψηφιακά μέσα που είναι το οπτικοποιημένο ψηφιακό μαθησιακό αντικείμενο (που συνιστά ένα διαδραστικό infographic) και μια φιλική γλώσσα προγραμματισμού (όπως το Scratch).

**Λέξεις κλειδιά:** Προγραμματιστικές δομές, Διδακτική Πληροφορικής, Scratch, Ρομπότ, ΚωδικΌραμα, STEAM.

## Εισαγωγή

Οι πρώτες ερευνητικές προσπάθειες σχεδιασμού και δημιουργίας ρομποτικών κατασκευών που κινούνται στο έδαφος ξεκίνησαν από τον G. Walter το 1948. Η πρώτη, διεθνώς, αξιοποίηση μιας ρομποτικής κατασκευής από παιδιά, ανάγεται στις αρχές του 1970 με πρωτεργάτη τον S. Papert (1980) και δημιουργό τον P. Wexelblat, με την κατασκευή της ασύρματης χελώνας εδάφους “Irving” (Chakraborty et al. 1999). Από τότε μέχρι σήμερα με τα εντυπωσιακά σύγχρονα ρομπότ (π.χ. “Robotsafe”, Gizelis Robotics 2020), έχουν περάσει 50 και πλέον χρόνια, αλλά η πρώτη και βασική προγραμματιστική δραστηριότητα κίνησης οποιασδήποτε ρομποτικής κατασκευής παραμένει η ίδια.

Παράλληλα το τρέχον ερευνητικό έργο Erasmus+ “Educational Infographics for STEAM” (2020), έχει ως βασικό σκοπό την αξιοποίηση οπτικών μέσων στη διδασκαλία, μέσω καινοτόμων μεθόδων για δραστηριότητες STEAM (Science, Technology, Engineering, Art, Maths). Στο πλαίσιο του έργου δημιουργούνται 40 και πλέον διαδραστικά εκπαιδευτικά infographics (σε επτά ευρωπαϊκές γλώσσες) από Εκπαιδευτικούς και Μαθητές, συνοδευόμενα και από αντίστοιχα σχέδια μάθησης-μαθήματος (learning-lesson plan). Ένα από τα 40 αυτά infographics είναι και το παρόν “Code Core”, στο οποίο οι δημιουργοί του επιχειρούν να συνδυάσουν τις παρακάτω τρεις οπτικοποιημένες προσεγγίσεις:

1. Κίνηση ενός εικονικού-ψηφιακού ρομπότ σε επίπεδες διαδρομές ή επιφάνειες χώρων με διάφορες αποστολές, όπως ακριβώς συμβαίνει στον πραγματικό κόσμο των ρομπότ.
2. Χρήση των έξι κύριων προγραμματιστικών δομών και των τριών κύριων δομών καθοδηγούμενων από συμβάντα, με τις οποίες επιλύονται όλα τα σχετικά προβλήματα.
3. Μεθοδολογική αποτύπωση του κώδικα που χρησιμοποιήθηκε, με το εργαλείο “ΚωδικόΧοράμα” το οποίο παρέχει οπτικοποιημένες πληροφορίες σχετικές με την ανατομία και τη λειτουργικότητα του κώδικα σε ένα δισδιάστατο πίνακα.
4. Το πρόγραμμα που επιλύει το πρόβλημα, για διδακτική αξιοποίηση ως project (με το αντίστοιχο σχέδιο μάθησης), έπρεπε:
  5. να έχει την απαραίτητη **λειτουργικότητα** ώστε να δημιουργεί ένα ελκυστικό παιχνίδι για το χρήστη (τάση αύξησης του πλήθους των αντικειμένων και των “καταστάσεων”),
  6. να διακρίνεται ο κώδικάς του για την **απλότητα** (τάση μείωσης του πλήθους των αντικειμένων και των “καταστάσεων”) και
  7. να έχει διδακτική **πληρότητα** περιλαμβάνοντας όλα τα είδη των προγραμματιστικών δομών.

### Διδακτικοί και εκπαιδευτικοί στόχοι

Βασικός σκοπός του project είναι οι μαθητές με μια σπειροειδή προσέγγιση αρχικά να πράττουν, να συσσωρεύουν εμπειρίες και να αποκτούν προσλαμβάνουσες παραστάσεις, δίνοντας έμφαση σε διαχρονικές έννοιες του προγραμματισμού (συγκλίνουσα - υπολογιστική - αλγοριθμική σκέψη, ανάλυση - σύνθεση, κλπ.) χωρίς να αποκτήσουν μόνο συγκεκριμένες και εξειδικευμένες γνώσεις σε εφήμερα προγραμματιστικά περιβάλλοντα.

Οι εννέα αυτές βασικές δομές είναι:

1. η δομή ακολουθίας (sequence),
2. η δομή επιλογής: εάν-τότε (if-then),
3. η δομή επιλογής: εάν-τότε-αλλιώς (if-then-else),
4. η δομή επανάληψης: επανάλαβε για πάντα (forever),
5. η δομή επανάληψης: επανάλαβε προκαθορισμένες φορές (for / repeat n times),
6. η δομή επανάληψης: επανάλαβε υπό συνθήκη (while / repeat until),
7. η δομή ανίχνευσης εξωτερικής δράσης επί συσκευής: Όταν γίνει κλικ στο πληκτρολόγιο (when space key pressed),
8. η δομή ανίχνευσης εξωτερικής δράσης επί αντικείμενου: Όταν γίνει κλικ στο αντικείμενο (when this sprite clicked),
9. η δομή ανίχνευσης εσωτερικής δραστηριότητας: Όταν λάβω μήνυμα ή Όταν ξεκινήσω ως κλώνος (when I receive message, when I start as a clone).

Οι μαθητές μετά την ενασχόλησή τους με τα θέματα του project, θα μπορούν:

- να αναγνωρίζουν τις προτεινόμενες διαχρονικές δομές σε προγράμματα και να περιγράφουν τη λειτουργία τους
- να περιγράφουν τις βασικές λειτουργίες κίνησης οποιασδήποτε ρομποτικής κατασκευής
- να εφαρμόζουν τις προτεινόμενες δομές σε προγράμματα που δημιουργούν οι ίδιοι
- να σχεδιάζουν μια εικονική ή πραγματική ρομποτική κατασκευή και να την κινούν

- να καταστρώνουν και να επιλύουν ποικίλα απλά προβλήματα ή και να κατασκευάζουν παιχνίδια χρησιμοποιώντας τις προτεινόμενες δομές.
- να περιγράφουν, να αναλύουν ή να συνθέτουν έναν απλό αλγόριθμο κίνησης ή σάρωσης μιας επιφάνειας για μια ρομποτική κατασκευή (π.χ. να ακολουθεί ένα δρόμο/μονοπάτι ή να αποφεύγει εμπόδια ή να σκουπίζει ή να βάφει ή να εκτοξεύει απολυμαντικό).

## Η δόμηση του infographic

Η οπτικοποίηση της κίνησης ρομποτικών κατασκευών με προγραμματιστικές δομές και έλεγχο συμβάντων ως infographic (όπως απεικονίζεται στο ΠΑΡΑΡΤΗΜΑ) συνιστά ένα μαθησιακό αντικείμενο (Zervas, et al., 2014) το οποίο δομείται στις παρακάτω τέσσερις διασυνδεδεμένες ενότητες:

1. Η πρώτη ενότητα αφορά στις **προγραμματιστικές δομές**: (α) την προγραμματιστική δομή *ακολουθίας* και με χρήση υποπρογραμμάτων, (β) την προγραμματιστική δομή *επιλογής* με το απλό “εάν/τότε” και το “εάν/τότε/αλλιώς” και (γ) την προγραμματιστική δομή *επανάληψης* “για πάντα”, “για προκαθορισμένο πλήθος επαναλήψεων” και “υπό συνθήκη” (Ladías, Karvounidis & Ladías, 2020). Η σειρά παράθεσης των επιμέρους εντολών δηλώνει την ευκολία κατανόησής τους από τους μαθητές.
2. Η δεύτερη ενότητα αφορά στον **κώδικα του προγράμματος** στο προγραμματιστικό περιβάλλον του Scratch αλλά οργανωμένο με την προσέγγιση “**ΚωδικΌραμα**” (τα οποία εξηγούνται στις επόμενες ενότητες), συνοδευόμενα όμως με λεκτική-γλωσσική περιγραφή του κώδικα (στο κάτω μέρος), όπου κάθε μία από τις πέντε προτάσεις της, σχετίζεται άμεσα με κάθε ένα από τα πέντε “blocks” προγραμμάτων.
3. Η τρίτη ενότητα αφορά σε ενδεικτικές **προτεινόμενες δραστηριότητες** με τον παρουσιαζόμενο κώδικα αλλά και τη δυνατότητα για επαναχρησιμοποίησή του (remix), συνοδευόμενες από ενδεικτικές “πίστες” αλλά και το διαχρονικό αλγόριθμο επίλυσης του προβλήματος λαβυρίνθου (Abelson & DiSessa, 1980; Bariamis, etc, 1993).
4. Η τέταρτη ενότητα αφορά σε ενδεικτικές **πηγές και αναφορές**, οι οποίες υποστηρίζουν τις προτεινόμενες προσεγγίσεις αλλά και από τις οποίες μπορούν να εξαχθούν πρόσθετες δραστηριότητες.

## Η επιλογή του Scratch ως προγραμματιστικό περιβάλλον

**Η ανάγκη που καλύπτει το Scratch:** Στη σύγχρονη εκπαιδευτική διαδικασία οι μαθητές εμπλέκονται σε αυθεντικές διαδικασίες μάθησης, όπως π.χ. είναι τα projects με προσέγγιση STEAM. Πολλές φορές σε αυτά τα projects απαιτείται οι μαθητές να προγραμματίσουν συσκευές και να αναπτύξουν κώδικες που συχνά είναι ογκώδεις και πολύπλοκοι. Έτσι αναδύεται η ανάγκη για υιοθέτηση **εκπαιδευτικών προγραμματιστικών περιβαλλόντων** στα οποία οι μικροί μαθητές να χειρίζονται **γλώσσες οπτικού προγραμματισμού με πλακίδια**, και αντιπροσωπευτικότερο παράδειγμα αυτών είναι το Scratch. Στο Scratch ο αρχάριος προγραμματιστής μπορεί να διαχειριστεί μέσω κώδικα -με σχετική ευκολία- πολυμεσικά στοιχεία και να εμπλακεί σε αυθεντικά ψηφιακά έργα δημιουργώντας animation, προσομοιώσεις και παιχνίδια (Resnick et. al., 2009). Στο περιβάλλον του Scratch το πρόγραμμα αποτελείται από **προγραμματιστικά σενάρια** (τμήματα-αρθρώματα κώδικα)

που είναι κατανεμημένα σε **αντικείμενα**. Στο Scratch ως **πρόγραμμα** νοείται το σύνολο των σεναρίων όλων των αντικειμένων.

**Η “μεταφορά” του Scratch:** Για την κατανόηση του τρόπου λειτουργίας του Scratch γίνεται χρήση της μεταφοράς που παραλληλίζει την ανάπτυξη ενός project με το “ανέβασμα” ενός θεατρικού έργου. Σύμφωνα με αυτή τη μεταφορά, το σενάριο / προγραμματιστικά σενάρια είναι εκείνα που προσδιορίζουν τη συμπεριφορά των ηθοποιών/αντικειμένων, οι οποίοι εμφανίζονται ως **ρόλοι** / sprites σε μια **σκηνή**, διαθέτοντας μια ποικιλία από κουστούμια / **ενδυμασίες**.

Η ενεργοποίηση των προγραμματιστικών σεναρίων προκαλείται από **συμβάντα** (events), που μπορεί να προέρχονται είτε από εξωτερικά γεγονότα που ανιχνεύονται από τις περιφερειακές συσκευές του υπολογιστή (π.χ. το πάτημα του ποντικιού), είτε από εσωτερικά γεγονότα όπως τα **μηνύματα** που εκπέμπονται από άλλα σενάρια ή κατά τη δημιουργία κλώνων. Η διαχείριση των συμβάντων στο Scratch αποτελεί παράδειγμα **προγραμματισμού βασισμένου σε συμβάντα** (event driven programming).

## Η Απεικόνιση του κώδικα με το ΚωδικΌραμα

**Η ανάγκη του ΚωδικΌραματος:** Η ίδια η φύση των αυθεντικών προβλημάτων που τίθενται από τα projects με προσέγγιση STEAM συνεπάγεται αυξημένη **πολυπλοκότητα των προγραμμάτων** που απαιτείται να αναπτύξουν οι αρχάριοι προγραμματιστές.

Σε αυτή την περίπτωση και όσον αφορά την **ανατομία του προγράμματος** στο περιβάλλον του Scratch, ο κατακερματισμός του κώδικα αφενός σε διαφορετικά αντικείμενα (που δεν επιτρέπουν την ταυτόχρονη εμφάνιση του συνόλου του κώδικα) και αφετέρου στη μη οργανωμένη χωροθέτηση των σεναρίων εντός της περιοχής προγραμματισμού ενός αντικειμένου, δημιουργούν την ανάγκη για μια εναλλακτική αναπαράσταση του κώδικα.

Επιπλέον όσον αφορά τη **λειτουργικότητα του προγράμματος**, σε αυτή την εναλλακτική αναπαράσταση του κώδικα θα πρέπει να γίνεται εμφανής η ροή της εκτέλεσης του προγράμματος και τα σημεία που αυτό αλληλεπιδρά με το χρήστη. Μια πρόταση που ικανοποιεί τα προηγούμενα είναι το ΚωδικΌραμα.

**Η οργάνωση του περιεχομένου στο ΚωδικΌραμα:** Το ΚωδικΌραμα είναι μια απεικόνιση του συνόλου των προγραμματιστικών σεναρίων όλων των αντικειμένων ενός προγράμματος σε έναν **πίνακα δύο διαστάσεων** (Λαδιάς & Λαδιάς, 2016; Ladias, Mikropoulos, Ladias & Bellou, 2021).

Όσον αφορά την **αναπαράσταση της ανατομίας** του προγράμματος, στις **στήλες του πίνακα** παρατίθενται τα σενάρια των αντικειμένων. Υιοθετώντας την προσέγγιση του “Μοντέλου Πεπερασμένων Καταστάσεων”, κάθε συμβάν αντιστοιχεί σε μια “κατάσταση” στην οποία περιέρχεται το σύστημα. Έτσι στις **γραμμές του πίνακα** αναπτύσσονται τα σενάρια που αντιστοιχούν στις “καταστάσεις” στις οποίες βρίσκονται τα αντικείμενα μετά την αναγνώριση ενός συμβάντος-αιτήματος που καλούνται να ικανοποιήσουν. Αποτέλεσμα αυτών είναι σε κάθε **κελί του πίνακα** να εμφανίζονται όλα εκείνα τα σενάρια που ανήκουν σε ένα αντικείμενο και ταυτόχρονα βρίσκονται στην ίδια κατάσταση.

Όσον αφορά την **αναπαράσταση της λειτουργικότητας** του προγράμματος, αυτή επιτυγχάνεται με τους **συνδέσμους** που μπορεί να υπάρχουν μεταξύ των σεναρίων συμβολίζοντας την επικοινωνία μεταξύ των σεναρίων του κώδικα και αναπαρίστανται με βέλη που δείχνουν την ενδεχόμενη **ροή του προγράμματος** κατά την εκτέλεσή του. Τα χρώματα που χρησιμοποιούνται για τους συνδέσμους κωδικοποιούν τους τρόπους επικοινωνίας, υιοθετώντας τη χρωματική κωδικοποίηση του Scratch. Οι διασυνδέσεις μεταξύ των τμημάτων του κώδικα αφορούν το **συχρονισμό της επικοινωνίας** (που γίνεται με την

αποστολή μηνυμάτων ή με τη μέθοδο της **διαμοιραζόμενης μνήμης**) και τη δημιουργία κλώνων.

**Η αναπαράσταση του προγράμματος με χρήση του ΚωδικΟράματος:** Η πληροφορία που απεικονίζεται στο ΚωδικΟραμα έχει σχέση με:

- Την ταυτόχρονη εποίπεια στην ολότητα του κώδικα (όλων των αντικειμένων).
- Το απόλυτο μέγεθος του κώδικα και την πυκνότητά του που σχετίζεται με την αναλογία του πλήθους των κενών κελιών ως προς το σύνολο.
- Τη σειρά παράθεσης των αντικειμένων (από αριστερά προς τα δεξιά) αναλογικά με το ιδιαίτερο βάρος του καθενός στο πρόγραμμα.
- Την κατανομή των προγραμματιστικών σεναρίων αφενός στα τρία αντικείμενα (ρομπότ, μήλα και πλήκτρο) του προγράμματος και αφετέρου στις τέσσερις διακριτές “καταστάσεις” του προγράμματος.
- Την ταυτοποίηση κάθε σεναρίου από τις συντεταγμένες του (αντικείμενο και κατάσταση).
- Τις περιοχές του κώδικα με ιδιαίτερα χαρακτηριστικά όπως η επάνω ζώνη με τα σενάρια αρχικοποίησης και η κάτω ζώνη που διαχειρίζεται το παίξιμο του παιχνιδιού.
- Την αντιστοιχία των πέντε βημάτων της λεκτικής περιγραφής της λειτουργίας του προγράμματος με τα αντίστοιχα δομικά στοιχεία (modules) του κώδικα.
- Την εξέλιξη της ροής του προγράμματος (από πάνω προς τα κάτω).
- Τον εκτεταμένο βαθμό τμηματοποίησης του προγράμματος σε συνδυασμό με τον περιορισμένο βαθμό ιεραρχικής σχεδίασης του προγράμματος.
- Τον χαρακτήρα του προγράμματος που βασίζεται σε συμβάντα.
- Την προέλευση των συμβάντων του προγράμματος που μπορεί να προέρχονται είτε από εξωτερικά αίτια όπως οι ενέργειες του χρήστη (when green flag clicked ή when this sprite clicked), είτε από εσωτερικά αίτια όπως η λήψη μηνυμάτων (when I receive message) ή η δημιουργία κλώνων (when I start as a clone).
- Επιπλέον παραδείγματα διαδοχικών (τα τέσσερα if στον ορισμό της διαδικασίας steer) και εμφωλευμένων δομών επιλογής (σενάριο “when I receive start game” του πλήκτρου “pen size”) και συνδυασμούς εμφωλευμένων δομών επιλογής στο εσωτερικό δομών επανάληψης (σενάριο when I start as a clone).

## Προτεινόμενες δραστηριότητες

Τα προαναφερθέντα μπορεί να δημιουργήσουν ενδιαφέροντες διαλόγους και ερωτήσεις που οι απαντήσεις τους είναι κρυμμένες σε κοινή θέα στο infographic και τις οποίες οι μαθητές θα κληθούν να εντοπίσουν. Παράλληλα με την προτεινόμενη προσέγγιση και τις υπάρχουσες πηγές στο infographic, μπορεί να πραγματοποιηθεί μια μεγάλη ποικιλία δραστηριοτήτων, όχι μόνο στο συγκεκριμένο προγραμματιστικό περιβάλλον του Scratch αλλά και σε οποιοδήποτε άλλη γλώσσα προγραμματισμού. Τέτοιες δραστηριότητες είναι:

- η διερεύνηση του κώδικα, η αναγνώριση - αντιστοίχιση με δομές, συμβάντα και αντικείμενα,
- η κίνηση εικονικού ρομπότ χρησιμοποιώντας πλήκτρα ή άλλα αντικείμενα,
- η δημιουργία οπτικο-ακουστικών εφέ για συγκρούσεις ή συλλογές αντικειμένων,
- η κατασκευή ή επιλογή άλλων διαδρομών ή χώρων σε επιφάνειες, και η δυνατότητα τυχαίων επιλογών,
- η κίνηση του ρομπότ σε διαφορετικούς λαβύρινθους και η αυτόματη εύρεση των εξόδων,

- η κατασκευή αγωνιστικών παιχνιδιών με ένα ή περισσότερα ρομπότ ή αντικείμενα,
- η κίνηση ρομπότ με ποικίλες αποστολές, π.χ. βιάσιμο ή καθαρισμό επιφανειών, εκνέφωση ("απολύμανση" χώρων) ή αποφυγή εμποδίων,
- η χρήση του ρομπότ για χαρτογράφηση χώρων και εμποδίων σε μια επιφάνεια.
- Επιπλέον είναι δυνατόν:
- να μελετηθεί η εξέλιξη ρομποτικών κατασκευών από το 1960 μέχρι σήμερα, μέσω μιας χρονογραμμής με σχετικά επιτεύγματα, αλλά και η διαφοροποίηση-εξέλιξη των λειτουργιών τους,
- να αναζητηθούν σημερινές μικρές και φθηνές ρομποτικές κατασκευές στις οποίες θα γίνει ο προγραμματισμός των κινήσεών τους.

## Συμπεράσματα

Το "Core Code" (ως project, σχέδιο μάθησης και infographic) οπτικοποιεί και παρουσιάζει ένα σύνολο πληροφοριών που όμως είναι κωδικοποιημένες και απαιτούν αφενός από τον θεατή-μαθητή την **ενεργό εμπλοκή** κατά την ανάγνωση και αφετέρου για το δημιουργό του προϋποθέτει την κατανόηση και εφαρμογή των κανόνων της **οπτικής γραφής** (Arnheim, 2004; Arnheim, 2007; Βακαλό, 1998; Κοζάκου, 2000; Μουζακίτη, 2003).

Συγκεκριμένα το "Core Code" σε ένα πρώτο επίπεδο παρουσιάζει τις προγραμματιστικές δομές (**ακολουθία - επιλογή - επανάληψη**) που χρησιμοποιούνται σε ένα περιβάλλον **προγραμματισμού που βασίζεται σε συμβάντα**. Στην παρουσίαση χρησιμοποιείται ως γλώσσα προγραμματισμού το Scratch που είναι ένα περιβάλλον **οπτικού προγραμματισμού με πλακίδια**. Για την αναπαράσταση της πολυπλοκότητας του κώδικα υιοθετείται το **κωδικόγραμμα** που παρέχει οπτικοποιημένες σε ένα διοδιάστατο πίνακα πληροφορίες σχετικές με την **ανατομία** και τη **λειτουργικότητα** του **κώδικα**.

Το infographic αυτό μπορεί να χρησιμοποιηθεί από τον εκπαιδευτικό, που ακολουθώντας τη **μαιευτική μέθοδο** με ερωτήσεις-υποδείξεις και σε μορφή παιχνιδιού, θα καθοδηγεί με κλιμακωτής δυσκολίας βήματα (**scaffolding**) τους μαθητές του, να **ανακαλύψουν** μόνοι τους τις προαναφερθείσες πληροφορίες που είναι κρυμμένες σε κοινή θέα.


Σχεδιάζεται έρευνα για τη μέτρηση της αποτελεσματικότητας όσον αφορά τα μαθησιακά αποτελέσματα της προαναφερθείσας διδακτικής προσέγγισης.

## Αναφορές

- Abelson, H., & DiSessa, A. A. (1980). *Turtle geometry: The computer as a medium for exploring mathematics*. Cambridge, Mass: MIT Press.
- Arnheim, Rudolf. (2004). *Τέχνη και οπτική αντίληψη, η ψυχολογία της δημιουργικής όρασης*. Αθήνα, Θεμέλιο.
- Arnheim, Rudolf. (2007). *Οπτική Σκέψη*. Θεσσαλονίκη, University Studio Press.
- Bariamis, G., Chaimantas, S., Kotsanis, Y., & Papatomaidi, L. (1993). A Microworld Oriented Approach in a Multi-Functional Logo-Based Curriculum. In Georgiadis, P., Gyftodimos, G., Kotsanis, Y., & Kynigos, X. (Eds). *Logo-like Learning Environments: Reflection & Prospects*, 4th European Logo Conference, Eurologo 93, Athens August 1993. Ανακτήθηκε 20/02/2021 <http://www.slideshare.net/YannisKotsanis/logomicroworlds1993>
- Chakraborty, A., Graebner, R., Stocky, T. (1999). *LOGO - A Project History* - MIT Publications. Ανακτήθηκε 20/02/2021 <http://web.mit.edu/6.933/www/LogoFinalPaper.pdf>
- "Educational Infographics for STEAM" (2020). *Erasmus+ Program*. Ανακτήθηκε 20/02/2021 <https://steam-edu.eu>
- Gizelis Robotics. (2020). *Robotosafe: the autonomous disinfection robot*. Ανακτήθηκε 20/02/2021 <https://grobotics.eu/disinfection-robot>

- Ladias, D., Karvounidis, T. & Ladias, A. (2020). Composition of programming structures into scenarios and their matching with the SOLO levels. *SEEDA-CECNSM 2020*, University of Piraeus.
- Ladias, A., Mikropoulos, Ar., Ladias, D. & Bellou, I. (2021). CodeOrama: A two dimensional visualization tool for Scratch code for teaching computer programming. *Themes in eLearning*, 14(1). Ανακτήθηκε 20/02/2021 <https://www.dropbox.com/s/wpd01hsdzbbzi09/2021%20CodeOrama.pdf>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books. (Ελληνική Έκδοση: Νοητικές Θύελλες, Εκδ. Οδυσσέας, 1991).
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, vol. 52, no. 11, pp. 60-67 (Nov. 2009). Ανακτήθηκε 20/02/2021 <http://www.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf>
- Zervas, P., Alifragkis, C., and Sampson, D. G. (2014) "A quantitative analysis of learning object repositories as knowledge management systems," *Knowledge Management & e-Learning Journal*, vol. 6, no. 2, pp. 156-170, 2014. Ανακτήθηκε 20/02/2021 <https://www.kmel-journal.org/ojs/index.php/online-publication/article/viewFile/240/241>
- Βακαλό, Εμμανουήλ-Γεώργιος. (1998). *Οπτική Σύνταξη: Λειτουργία και Παραγωγή Μορφών*. Αθήνα, Νεφέλη.
- Κοζάκου-Τσιάρα, Όλγα. (2000). *Εισαγωγή στην εικαστική γλώσσα*. Αθήνα. Gutenberg.
- Λαδιάς, Α. & Λαδιάς, Δ. (2016). Η αναπαράσταση του αλγορίθμου με τη βοήθεια του κωδικογράμματος σε περιβάλλοντα οπτικού προγραμματισμού. *Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση*, 9(2),103-117.
- Μουζακίτη, Φρόνη. (2003). Φόρμα. *Η οπτική γλώσσα στον σύγχρονο σχεδιασμό*. Αθήνα, Οδυσσέας.

## Παράρτημα: Infographic Code Core



## Moving a Robot with the 6 key structures & 3 event types of coding

**Commands**

**Programming Structures**

**sequence**

command  
command  
my block → define my block  
command  
command 1  
command 2

**decision (if - then - or else)**

if then  
if then  
else  
if then

**Iteration ('for' and 'while' loops)**

forever  
repeat  
repeat until

robot apples button ← Objects ('sprites')

*CodeOrama*

Initialization

when you click on the green flag...  
external event

when you click on the sprite ('apple')...  
external event

when I start as a clone ('apple')...  
Internal event

when I receive the message...  
Internal event


play & see inside  
[scratch.mit.edu/projects/3153906](https://scratch.mit.edu/projects/3153906)

HOUR of CODE  
[hourofcode.com/join/learn](http://hourofcode.com/join/learn)

*The robot moves continuously along a path. It is guided by two sensors. The goal is to eat all the apples along the path. Any apples outside the path can be dragged and dropped onto the path by the user. The user can also change the pen size/color of the robot track...*

**Event Driven Structures**

**tips & tricks!**

 Pledge algorithm

- Explore the code, find the programming and event structures, change the objects.
- Move the robot using keyboard keys or other objects.
- Create sound and light effects for collisions and/or collecting apples.
- Construct or choose your own tracks. Code to choose random tracks.
- Move the robot along different labyrinths. Find the exits.
- Create racing games with one or more robots or objects.
- Move the robot to clean or to rebulize the track. Put obstacles along the path.
- Use the robot for mapping different areas (rooms).

**remix ideas!**

QR Code

Logo: A Project History [web.mit.edu/6.933/www/LogoFinalPaper.pdf](http://web.mit.edu/6.933/www/LogoFinalPaper.pdf)

A Microworld Approach [dodshare.net/AnnexKatsani/lopmicroworlds41993](http://dodshare.net/AnnexKatsani/lopmicroworlds41993)

'CodeOrama' [corthlab.uoi.gr/tel/index.php/themeselearn](http://corthlab.uoi.gr/tel/index.php/themeselearn)

Scratch Projects [scratch.mit.edu/explore/projects/all](http://scratch.mit.edu/explore/projects/all)


Dry Fog Disinfection ROBOTS SAFE [robotics.eu/disinfection-robot](http://robotics.eu/disinfection-robot)

Core-Code Robot: Design by CARDET [www.cardet.org](http://www.cardet.org)

Y. Kotsanis - A. Ladias

This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained herein.

Co-funded by the Erasmus+ Programme of the European Union



Ιστότοπος Infographic: [Core Code](#)

- |   |  |
|---|--|
| • <a href="#">Play &amp; See Inside</a> | • <a href="#">Logo: A Project History</a>          |
| • <a href="#">Scratch Projects</a>      | • <a href="#">A Microworld Approach</a>            |
| • <a href="#">Hour of Code</a>          | • <a href="#">"CodeOrama"</a>                      |
| • <a href="#">Pledge algorithm</a>      | • <a href="#">Dry Fog Disinfection ROBOTS SAFE</a> |