

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
«ΡΟΜΠΟΤΙΚΗ, STEAM ΚΑΙ ΝΕΕΣ ΤΕΧΝΟΛΟΓΙΕΣ  
ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ»**

Διπλωματική Εργασία

**ΕΦΑΡΜΟΓΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΡΟΜΠΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ STEAM ΜΕ ΧΡΗΣΗ ΤΟΥ LEGO SPIKE PRIME.**

της

ΜΑΡΙΑΣ ΠΙΝΑ

Επιβλέπων Καθηγητής

Στέλιος Ξανθός

Υποβλήθηκε ως απαιτούμενο για την απόκτηση του μεταπτυχιακού διπλώματος  
ειδίκευσης Ρομποτική, STEAM και Νέες Τεχνολογίες στην Εκπαίδευση  
Θεσσαλονίκη, Φεβρουάριος 2023



Η παρούσα Διπλωματική Εργασία καλύπτεται στο σύνολό της νομικά από δημόσια άδεια πνευματικών δικαιωμάτων CreativeCommons:

Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή



Μπορείτε να:

- Μοιραστείτε: αντιγράψετε και αναδιανέμετε το παρόν υλικό με κάθε μέσο και τρόπο
- Προσαρμόστε: αναμείξτε, τροποποιήστε και δημιουργήστε πάνω στο παρόν υλικό

Υπό τους ακόλουθους όρους:

- Αναφορά Δημιουργού: Θα πρέπει να καταχωρίσετε αναφορά στο δημιουργό, με σύνδεσμο της άδειας, και με αναφορά αν έχουν γίνει αλλαγές. Μπορείτε να το κάνετε αυτό με οποιονδήποτε εύλογο τρόπο, αλλά όχι με τρόπο που να υπονοεί ότι ο δημιουργός αποδέχεται το έργο σας ή τη χρήση που εσείς κάνετε.
- Μη Εμπορική Χρήση: Δε μπορείτε να χρησιμοποιήσετε το υλικό για εμπορικούς σκοπούς.
- Παρόμοια Διανομή: Αν αναμείξετε, τροποποιήσετε, ή δημιουργήσετε πάνω στο παρόν υλικό, πρέπει να διανείμετε τις δικές σας συνεισφορές υπό την ίδια άδεια CreativeCommonsόπως και το πρωτότυπο.

Αναλυτικές πληροφορίες νομικού κώδικα στην ηλεκτρονική διεύθυνση:

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

## Υπεύθυνη Δήλωση

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις που προβλέπονται από τον Κανονισμό Σπουδών του Μεταπτυχιακού Προγράμματος Ρομποτική, STEAM και Νέες Τεχνολογίες στην Εκπαίδευση του Διεθνούς Πανεπιστημίου Ελλάδος, δηλώνω υπεύθυνα ότι:

- Η παρούσα Διπλωματική Εργασία αποτελεί έργο αποκλειστικά δικής μου δημιουργίας, έρευνας, μελέτης και συγγραφής.
- Για τη συγγραφή της Διπλωματικής μου Εργασίας δεν χρησιμοποίησα ολόκληρο ή μέρος έργου άλλου δημιουργού ή τις ιδέες και αντιλήψεις άλλου δημιουργού χωρίς να γίνεται σαφής αναφορά στην πηγή προέλευσης(βιβλίο, άρθρο από επιστημονικό περιοδικό, ιστοσελίδα κλπ.).

Θεσσαλονίκη, 11 Φεβρουαρίου 2023

Η Δηλούσα: Μαρία Πίνα

*Στην Ανθή και στον Γεράσιμο.*

Θα ήθελα αρχικά να ευχαριστήσω τους καθηγητές μου, κύριο Ξανθό Στέλιο και κύριο Καμούτση Κωνσταντίνο για την καθοδήγησή τους και την υποστήριξη καθ' όλη τη διάρκεια της εργασίας, καθώς και για την ευκαιρία που μου έδωσαν να συνεργαστώ μαζί τους. Επιπρόσθετα, θέλω να ευχαριστήσω την οικογένειά μου και τον αγαπημένο μου Κωνσταντίνο για την αμέριστη αγάπη και συμπαράσταση σε όλη την διάρκεια των μεταπτυχιακών μου σπουδών.

## Περίληψη

Η τεχνολογία έχει γίνει πλέον αναπόσπαστο κομμάτι της ζωής του ανθρώπου, διεισδύοντας σε όλους τους τομείς της ανθρώπινης ύπαρξης, πλέον και στον χώρο της εκπαίδευσης. Στα πλαίσια της προσπάθειας εξέλιξης της εκπαιδευτικής διαδικασίας για αποδοτικότερα μαθησιακά αποτελέσματα και απόκτησης δεξιοτήτων με βάση τις κοινωνικές και οικονομικές εξελίξεις παγκοσμίως, εμφανίστηκε μία νέα διδακτική προσέγγιση, η STEAM εκπαίδευση. Όντας πλέον σε μια τεχνολογική εποχή, ένας από τους στόχους της STEAM εκπαίδευσης είναι να φέρει σε επαφή τους νεαρούς μαθητές με τις αναδυόμενες τεχνολογίες, μια εκ των οποίων είναι και ο κλάδος της ρομποτικής.

Με την παρούσα διπλωματική εργασία αποφασίσαμε λοιπόν να επικεντρωθούμε στο κομμάτι της STEAM εκπαίδευσης που αφορά την εκπαιδευτική ρομποτική, ρομποτικούς δηλαδή μηχανισμούς που μπορούν να χρησιμοποιηθούν στα πλαίσια ενός μαθήματος για την καλύτερη και πιο αποτελεσματική επίτευξη των μαθησιακών στόχων, τόσο στα πεδία του STEAM όσο και στο κομμάτι που αφορά τον προγραμματισμό, η γνώση του οποίου θεωρείται βασική στον 21<sup>ο</sup> αιώνα. Σκοπός μας είναι να παρουσιάσουμε 5 ρομποτικά συστήματα, καθώς και τα προγράμματα αυτών, τα οποία μπορούν να χρησιμοποιηθούν στα πλαίσια της STEAM εκπαίδευσης. Το συνολικό εγχείρημα βασίζεται στη βιβλιογραφία που υπάρχει για την κατασκευή και τον προγραμματισμό ρομποτικών συστημάτων, ενώ οι υλοποίηση των δραστηριοτήτων πραγματοποιήθηκε στο εκπαιδευτικό Kit της LEGO, το Spike Prime. Οι δραστηριότητες που υλοποιήσαμε είναι μία αυτόματη αναπηρική καρέκλα, μία ρομποτική γάτα, ένας ρομποτικός βραχίονας, μία μηχανή διαλογής και μία μηχανή αντιγραφής μοτίβων.

Σαν αποτέλεσμα είχαμε την εκτενή μελέτη σε θεωρητικό επίπεδο των όσων προαναφέρθηκαν, εμβαθύνοντας στην STEAM εκπαίδευση, στις μεθόδους διδασκαλίας που διέπουν αυτή, στη σχέση της με την εκπαιδευτική ρομποτική, στην παρουσίαση εκπαιδευτικών ρομποτικών συστημάτων, στην ανάλυση του προγραμματισμού του LEGO Spike Prime κ.α. Επιπρόσθετα, κάθε μια από τις εφαρμογές που προαναφέραμε, τόσο σε κατασκευαστικό επίπεδο όσο και σε επίπεδο προγραμματισμού υλοποιήθηκε με αίσια έκβαση. Το συνολικό εγχείρημα ολοκληρώθηκε με μεγάλη επιτυχία, έχοντας όμως πάντα περιθώρια βελτίωσης για μελλοντικές προτάσεις και εξελίξεις.

## **Abstract**

Technology has now become an integral part of human life, penetrating into all areas of our existence, now also in the field of education. In the context of the effort to develop the educational process for more efficient learning results and the acquisition of skills based on social and economic developments worldwide, a new teaching approach appeared, named STEAM education. As we live in a technological age, one of the STEAM education's goals is to bring young students closer to the emerging technologies, one of which being the field of robotics.

So, with this thesis, we decided to focus on the part of STEAM education that concerns educational robotics, i.e. robotic mechanisms that can be part of a teaching class to achieve our learning goals as good and effective as possible, in the fields of STEAM but also of Computer Programming, the knowledge of which is considered essential in the 21st century. Our goal is to present 5 robotic systems, as well as their programs, which can be used in the context of STEAM education. The overall project is based on the existing literature regarding the construction and programming of robotic systems, while the activities' implementation was carried out using the LEGO Spike Prime Educational Kit. These activities are: an automatic wheelchair, a robotic cat, a robotic arm, a sorting machine and a pattern copying machine.

As a result, we had an extensive study at a theoretical level of what was mentioned earlier, delving into STEAM education and teaching methods, its relationship with educational robotics, the presentation of educational robotic systems, the analysis of the programming of LEGO Spike Prime, etc. In addition, each of the applications we mentioned above, regarding both their construction and their programming, were implemented with great outcome. The overall project had the desired results, always keeping in mind that there is always a lot of room for improvement, future proposals and further development.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Κατάλογος Πινάκων.....	9
Κατάλογος Εικόνων .....	9
Κεφάλαιο 1 – Εισαγωγή .....	13
1.1 Αντικείμενο της διπλωματικής εργασίας.....	13
1.2 Σκοπός της διπλωματικής εργασίας .....	14
1.3 Διάρθρωση της διπλωματικής εργασίας.....	15
Κεφάλαιο 2 – STEAM Εκπαίδευση .....	16
2.1 Τι είναι η STEM Εκπαίδευση.....	16
2.2 Τα αντικείμενα της STEM Εκπαίδευσης.....	20
2.3 Από το STEM στο STEAM.....	24
2.4 Επεξηγώντας το A στο STEAM.....	24
2.5 Πλεονεκτήματα STEAM εκπαίδευσης.....	26
2.6 Θεωρίες μάθησης και STEAM.....	28
2.7 Ο ρόλος του εκπαιδευτικού στην STEAM Εκπαίδευση. ....	29
2.8 Οι μέθοδοι διδασκαλίας στην STEAM Εκπαίδευση. ....	30
Κεφάλαιο 3 – Ρομποτικά Συστήματα .....	32
3.1 Ορισμός της Ρομποτικής. ....	32
3.2 Ιστορική αναδρομή.....	33
3.3 Χρήσεις Ρομποτικής.....	36
3.4 Ανατομία και έλεγχος ρομποτικών συστημάτων. ....	39
Κεφάλαιο 4 – Εκπαιδευτική Ρομποτική .....	41
4.1 Ορισμός Εκπαιδευτικής Ρομποτικής.....	41
4.2 Θεωρίες Μάθησης και Εκπαιδευτική Ρομποτική.....	42
4.3 Μέθοδοι Διδασκαλίας στην Εκπαιδευτική Ρομποτική.....	44
4.4 Εκπαιδευτικά Ρομποτικά Συστήματα.....	46
4.5 Πλεονεκτήματα Εκπαιδευτικής Ρομποτικής. ....	53
4.6 Εκπαιδευτική Ρομποτική και Διδασκαλία Προγραμματισμού.....	54
Κεφάλαιο 5 – Παρουσίαση του Spike Prime.....	57
5.1 Πληροφορίες για το Spike Prime. ....	57
5.2 Επισκόπηση του Kit. ....	58
5.3 Αποκτώντας Πρόσβαση στο λογισμικό του Spike Prime. ....	67
5.4 Επισκόπηση της εφαρμογής του Spike Prime.....	68
5.5 Το Lego Spike Prime Set Επέκταση.....	70
Κεφάλαιο 6 – Προγραμματισμός του Spike Prime.....	72



6.1 Τρόποι προγραμματισμού του Lego SPIKE Prime. ....	73
6.2 Προγραμματισμός με βάση εικονίδια (Icon-Based Programming).....	74
6.3 Το Icon-Based περιβάλλον προγραμματισμού στην εφαρμογή του Spike. ....	74
6.4 Προγραμματισμός με Blocks (Block-Based Programming). ....	75
6.5 Τύποι των Blocks και Ορολογία. ....	76
6.6 Κατηγορίες των Blocks. ....	80
6.7 Περισσότερες Επεκτάσεις. ....	97
6.8 Το Block-Based περιβάλλον προγραμματισμού στην εφαρμογή του Spike.....	97
6.9 Ξεκινώντας με τη Γλώσσα Προγραμματισμού Python. ....	98
6.10 Προγραμματισμός με Python στην εφαρμογή του Spike Prime. ....	100
6.11 Το περιβάλλον Προγραμματισμού με Python στην Εφαρμογή του Spike.....	114
Κεφάλαιο 7 – Σχεδιασμός και Προγραμματισμός Ρομποτικών Συστημάτων.....	117
7.1 Διαδικασία Σχεδιασμού Εφαρμοσμένης Μηχανικής. ....	117
7.2 Δημιουργία Εκπαιδευτικών Ρομποτικών Συστημάτων. ....	119
Κεφάλαιο 8 – Παρουσίαση των Εφαρμογών.....	124
8.1 Αναπηρική Καρέκλα ..... 124	124
8.2 Ρομποτική γάτα. .... 136	136
8.3 Ρομποτικός Βραχίονας. .... 146	146
8.4 Μηχανή διαλογής. .... 152	152
8.5 Μηχανή Αντιγραφής Εικονιδίων..... 159	159
Κεφάλαιο 9 – Συμπεράσματα ..... 170	170
Βιβλιογραφία ..... 172	172
Πηγές Εικόνων..... 196	196
Παράρτημα..... 197	197
Κατάλογος με τα Δομικά στοιχεία του Spike Prime. .... 197	197
Κατάλογος με τα Ηλεκτρονικά Εξαρτήματα του Spike Prime ..... 197	197
Κατάλογος με όλα τα στοιχεία του Set Επέκτασης..... 198	198

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

### Κατάλογος Πινάκων

Πίνακας 5.1 : Σύγκριση των βασικών χαρακτηριστικών των κινητήρων.....	[67]
Πίνακας 6.1 Η ορολογία των Block Εντολών.....	[79]
Πίνακας 6.2 Οι βασικές εντολές της Μπλε Κατηγορίας.....	[81]
Πίνακας 6.3 Οι εντολές της Επέκτασης της Μπλε Κατηγορίας.....	[82]
Πίνακας 6.4 Οι βασικές εντολές της Ροζ Κατηγορίας.....	[83]
Πίνακας 6.5 Οι εντολές της Επέκτασης της Μπλε Κατηγορίας.....	[84]
Πίνακας 6.6 Οι εντολές της Μωβ κατηγορίας.....	[85]
Πίνακας 6.7 Οι εντολές της Λιλιά Κατηγορίας.....	[86]
Πίνακας 6.8 Οι εντολές της Κίτρινης Κατηγορίας.....	[87]
Πίνακας 6.9 Οι εντολές της Πορτοκαλί Κατηγορίας.....	[89]
Πίνακας 6.10 Οι εντολές της Τirkουάζ Κατηγορίας.....	[90]
Πίνακας 6.11 Οι εντολές της Επέκτασης της Τirkουάζ Κατηγορίας.....	[91]
Πίνακας 6.12 Οι εντολές της Πράσινης Κατηγορίας.....	[92]
Πίνακας 6.13 Οι εντολές της Κατηγορίας των Μεταβλητών.....	[94]
Πίνακας 6.14 Οι εντολές της Κατηγορίας των Λιστών.....	[94]

### Κατάλογος Εικόνων

Εικόνα 2.1 Οι τρεις διαστάσεις της STEM Εκπαίδευσης.....	[19]
Εικόνα 4.1 Η μέλισσα Bee-bot. ....	[47]
Εικόνα 4.2 Το εκπαιδευτικό Kit: Kids First Coding & Robotics.....	[48]
Εικόνα 4.3 Το εκπαιδευτικό Kit: Vex Robotics Design System. ....	[49]
Εικόνα 4.4 Το ρομπότ Nao. ....	[50]
Εικόνα 4.5 Το εκπαιδευτικό Kit: Arduino & Nezha Inventor's Kit for Arduino.....	[51]
Εικόνα 4.6 Το εκπαιδευτικό Kit: Lego WeDo 2.0. ....	[52]
Εικόνα 5.1 Το εκπαιδευτικό Kit: Lego Spike Prime. ....	[53]
Εικόνα 5.2 Απεικόνιση του Hub. ....	[59]
Εικόνα 5.3 Ο Φωτεινός Πίνακας/Light Matrix. ....	[60]

<i>Εικόνα 5.4 Οι όροι Yaw, Pitch και Roll στην πράξη.</i>	[62]
<i>Εικόνα 5.5 Ο αισθητήρας χρώματος.</i>	[63]
<i>Εικόνα 5.6 Ο αισθητήρας απόστασης.</i>	[64]
<i>Εικόνα 5.7 Ο αισθητήρας αφής.</i>	[65]
<i>Εικόνα 5.8 Ο μεγάλος και οι μικροί κινητήρες.</i>	[66]
<i>Εικόνα 5.9 Η αρχική σελίδα της εφαρμογής.</i>	[68]
<i>Εικόνα 5.10 Το κεντρικό Menu της εφαρμογής.</i>	[69]
<i>Εικόνα 5.11 Το Menu βοήθειας.</i>	[70]
<i>Εικόνα 5.12 Το Set Επέκτασης.</i>	[70]
<i>Εικόνα 5.13 Απεικόνιση του Maker Plate.</i>	[71]
<i>Εικόνα 6.1 Η Εντολή "Απενεργοποίησε τον Κινητήρα A" σε α) Icon Based Blocks β) Word Blocks γ) Python.</i>	[73]
<i>Εικόνα 6.2 Παράδειγμα δημιουργίας προγράμματος με Icon-Based Προγραμματισμό.</i>	[74]
<i>Εικόνα 6.3 Το περιβάλλον Icon-Based Προγραμματισμού.</i>	[75]
<i>Εικόνα 6.4 Παράδειγμα της μη-αυτονομίας των δύο κατηγοριών.</i>	[79]
<i>Εικόνα 6.5 Οι 10 κατηγορίες εντολών του Spike Prime ανά χρώμα.</i>	[80]
<i>Εικόνα 6.6 Οι επεκτάσεις της Εφαρμογής.</i>	[81]
<i>Εικόνα 6.7 Δημιουργία συναρτήσεων σε Block-Based προγράμματα.</i>	[96]
<i>Εικόνα 6.8 Παράδειγμα Συνάρτησης σε Block-Based προγράμματα.</i>	[96]
<i>Εικόνα 6.9 Περισσότερες Επεκτάσεις της εφαρμογής.</i>	[97]
<i>Εικόνα 6.10 Το περιβάλλον Block-Based Προγραμματισμού.</i>	[98]
<i>Εικόνα 6.11 Απεικόνιση της κληρονομικότητας των κλάσεων.</i>	[100]
<i>Εικόνα 6.12 Παράδειγμα των εσοχών και των blocks.</i>	[102]
<i>Εικόνα 6.13 Απεικόνιση της βοηθητικής γκρι γραμμής.</i>	[102]
<i>Εικόνα 6.13 Παράδειγμα εισαγωγής σχόλιου.</i>	[104]
<i>Εικόνα 6.14 Παράδειγμα της εντολής While.</i>	[105]
<i>Εικόνα 6.15 Παράδειγμα 2 της εντολής While.</i>	[105]
<i>Εικόνα 6.16 Παράδειγμα της εντολής while not.</i>	[106]
<i>Εικόνα 6.17 Παράδειγμα της εντολής while True.</i>	[106]
<i>Εικόνα 6.18 Παράδειγμα της επανάληψης for.</i>	[107]
<i>Εικόνα 6.19 Παράδειγμα της επανάληψης for με αρχική τιμή, τελική τιμή και βήμα.</i>	[108]

<i>Εικόνα 6.20 Παράδειγμα της επανάληψης for με αρνητικό βήμα.</i>	[108]
<i>Εικόνα 6.21 Παράδειγμα της εντολής if.</i>	[109]
<i>Εικόνα 6.22 Παράδειγμα της δομής if/else.</i>	[109]
<i>Εικόνα 6.23 Παράδειγμα της δομής if/elif/else.</i>	[110]
<i>Εικόνα 6.24 Παράδειγμα διαίρεσης στον προγραμματισμό με Python.</i>	[113]
<i>Εικόνα 6.25 Το περιβάλλον προγραμματισμού με Python.</i>	[114]
<i>Εικόνα 7.1 Παράδειγμα της διαδικασίας Σχεδιασμού Εφαρμοσμένης Μηχανικής...</i>	[117]
<i>Εικόνα 8.1.1 Οι λειτουργίες του Αναπηρικού Αμαξιδίου...</i>	[125]
<i>Εικόνα 8.1.2 Το προσχέδιο του Αναπηρικού Αμαξιδίου.</i>	[126]
<i>Εικόνα 8.1.3 Το πρωτότυπο του Αναπηρικού Αμαξιδίου.</i>	[126]
<i>Εικόνα 8.1.4 Το τελικό αποτέλεσμα του Αναπηρικού Αμαξιδίου.</i>	[127]
<i>Εικόνα 8.1.5 Ο κώδικας της πρώτης εφαρμογής του Αναπηρικού Αμαξιδίου...</i>	[128]
<i>Εικόνα 8.1.6 Η στοιβάδα μπροστινή κίνησης.</i>	[129]
<i>Εικόνα 8.1.7 Η στοιβάδα πίσω κίνησης.</i>	[130]
<i>Εικόνα 8.1.8 Η στοιβάδα περιστροφικής κίνησης.</i>	[130]
<i>Εικόνα 8.1.9 Οι στοιβάδες επιταχυνόμενης κίνησης.</i>	[131]
<i>Εικόνα 8.1.10 Ο κώδικας της δεύτερης εφαρμογής του Αναπηρικού Αμαξιδίου....</i>	[132]
<i>Εικόνα 8.1.11 Εισαγωγή Βιβλιοθηκών και κλάσεων για το Αναπηρικό αμαξίδιο...</i>	[133]
<i>Εικόνα 8.1.12 Δημιουργία Αντικειμένων για το Αναπηρικό αμαξίδιο.</i>	[134]
<i>Εικόνα 8.1.13 Το βασικό κομμάτι του προγράμματος του Αναπηρικού Αμαξιδίου..</i>	[134]
<i>Εικόνα 8.2.1 Οι λειτουργίες της Ρομποτικής Γάτας.</i>	[136]
<i>Εικόνα 8.2.2 Το προσχέδιο της Ρομποτικής Γάτας.</i>	[137]
<i>Εικόνα 8.2.3 Το πρωτότυπο της Ρομποτικής Γάτας.</i>	[138]
<i>Εικόνα 8.2.4 Το τελικό αποτέλεσμα της Ρομποτικής Γάτας.</i>	[139]
<i>Εικόνα 8.2.5 Το πρώτο πρόγραμμα της Ρομποτικής Γάτας.</i>	[140]
<i>Εικόνα 8.2.6 Αρχικοποίηση Μεταβλητών.</i>	[140]
<i>Εικόνα 8.2.7 Στοιβάδα Κατανάλωσης Τροφής.</i>	[141]
<i>Εικόνα 8.2.8 Στοιβάδα Κατανάλωσης Νερού.</i>	[141]
<i>Εικόνα 8.2.9 Στοιβάδα Διασκέδασης.</i>	[142]
<i>Εικόνα 8.2.10 Στοιβάδα Τερματισμού του Παιχνιδιού.</i>	[142]
<i>Εικόνα 8.2.11 Το δεύτερο πρόγραμμα της Ρομποτικής Γάτας.</i>	[143]
<i>Εικόνα 8.2.12 Η Στοιβάδα Ορισμού Ηρεμίας.</i>	[143]

<i>Εικόνα 8.2.13 Η Στοιβάδα ορισμού διαθέσεων.</i>	[144]
<i>Εικόνα 8.2.14 Η Στοιβάδα επιλογής διάθεσης.</i>	[145]
<i>Εικόνα 8.3.1 Οι λειτουργίες του Ρομποτικού Βραχίονα.</i>	[146]
<i>Εικόνα 8.3.2 Το προσχέδιο του Ρομποτικού Βραχίονα.</i>	[147]
<i>Εικόνα 8.3.3 Το πρωτότυπο του Ρομποτικού Βραχίονα.</i>	[148]
<i>Εικόνα 8.3.4 Το τελικό αποτέλεσμα του Ρομποτικού Βραχίονα.</i>	[149]
<i>Εικόνα 8.3.5 Ολοκληρωμένος ο κώδικας του Ρομποτικού Βραχίονα.</i>	[149]
<i>Εικόνα 8.3.6 Οι βιβλιοθήκες και κλάσεις του Ρομποτικού Βραχίονα.</i>	[149]
<i>Εικόνα 8.3.7 Αρχικοποίηση των Αντικειμένων.</i>	[150]
<i>Εικόνα 8.3.8 Το βασικό κομμάτι του Προγράμματος.</i>	[150]
<i>Εικόνα 8.4.1 Οι λειτουργίες της Μηχανής Διαλογής.</i>	[153]
<i>Εικόνα 8.4.2 Το προσχέδιο της Μηχανής Διαλογής.</i>	[154]
<i>Εικόνα 8.4.3 Το πρωτότυπο της Μηχανής Διαλογής.</i>	[154]
<i>Εικόνα 8.4.4 Το τελικό αποτέλεσμα της Μηχανής Διαλογής.</i>	[155]
<i>Εικόνα 8.4.5 Ολοκληρωμένος ο Κώδικας της Μηχανής Διαλογής.</i>	[156]
<i>Εικόνα 8.4.6 Βιβλιοθήκες και κλάσεις της Μηχανής Διαλογής.</i>	[156]
<i>Εικόνα 8.4.7 Αρχικοποίηση Αντικειμένων της Μηχανής Διαλογής.</i>	[156]
<i>Εικόνα 8.4.8 Το βασικό κομμάτι του κώδικα της Μηχανής Διαλογής.</i>	[157]
<i>Εικόνα 8.5.1 Οι λειτουργίες της Μηχανής Αντιγραφής Εικονιδίων.</i>	[159]
<i>Εικόνα 8.5.2 Το προσχέδιο της Μηχανής Αντιγραφής Εικονιδίων.</i>	[160]
<i>Εικόνα 8.5.3 Το πρωτότυπο της Μηχανής Αντιγραφής Εικονιδίων.</i>	[161]
<i>Εικόνα 8.5.4 Το νέο προσχέδιο της Μηχανής Αντιγραφής Εικονιδίων.</i>	[162]
<i>Εικόνα 8.5.5 Το νέο πρωτότυπο της Μηχανής Αντιγραφής Εικονιδίων.</i>	[162]
<i>Εικόνα 8.5.6 Το τελικό αποτέλεσμα της Μηχανής Αντιγραφής Εικονιδίων.</i>	[163]
<i>Εικόνα 8.5.7 Ολοκληρωμένο το πρόγραμμα της Μηχανής Αντιγραφής Εικονιδίων.</i>	[164]
<i>Εικόνα 8.5.8 Βιβλιοθήκες και κλάσεις της Μηχανής Αντιγραφής Εικονιδίων.</i>	[164]
<i>Εικόνα 8.5.9 Τα αντικείμενα της Μηχανής Αντιγραφής Εικονιδίων.</i>	[164]
<i>Εικόνα 8.5.10 Η αρχή του προγράμματος.</i>	[165]
<i>Εικόνα 8.5.11 Χρωματική αντιστοίχιση των δύο επαναλήψεων.</i>	[166]
<i>Εικόνα 8.5.12 Η επανάληψη «σκαναρίσματος» μιας σειράς.</i>	[167]
<i>Εικόνα 8.5.13 Η εξωτερική επανάληψη.</i>	[168]

## **Κεφάλαιο 1 – Εισαγωγή**

Στην προσπάθεια της εκπαίδευσης να συμβαδίσει με την 4η βιομηχανική επανάσταση και να καλλιεργήσει στους μαθητές δεξιότητες όπως η υπολογιστική σκέψη, η ενεργητική μάθηση, η επίλυση προβλημάτων και η σύνδεση των μαθητών με την τεχνολογία γύρω τους, εμφανίστηκε ο κλάδος της STEAM (Science, Technology, Engineering, Arts, Mathematics) εκπαίδευσης στα πλαίσια της οποίας η ανάπτυξη αυτών των δεξιοτήτων συμβαίνει χρησιμοποιώντας και ρομποτικά εκπαιδευτικά εργαλεία. Η εκπαιδευτική ρομποτική όντας ένα αντικείμενο το οποίο κεντρίζει το ενδιαφέρον των μαθητών και τους ενώνει με τον χώρο της μάθησης πολύ αποτελεσματικά. Ταυτόχρονα αποτελεί ένα πεδίο που συνδυάζει γνώσεις από τα STEAM πεδία και τον κόσμο της επιστήμης των υπολογιστών. Για το λόγο αυτό βρίσκεται στον επίκεντρο του εκπαιδευτικού επιστημονικού ενδιαφέροντος. Έτσι λοιπόν, καθίσταται ανάγκη η συνεχής παραγωγή νέου εκπαιδευτικού υλικού στον τομέα της εκπαιδευτικής ρομποτικής.

Στα πλαίσια λοιπόν αυτής της εργασίας αποφασίσαμε να αναπτύξουμε 5 ρομποτικά συστήματα, καθώς και τον προγραμματισμό τους, τα οποία μπορούν να ενσωματωθούν μέσα σε STEAM εκπαιδευτικά πλαίσια. Το Kit στο οποίο αποφασίσαμε να υλοποιήσουμε αυτές τις εφαρμογές είναι το εκπαιδευτικό Kit ρομποτικής Lego Spike Prime. Για την δημιουργία των ρομποτικών συστημάτων ακολουθήσαμε μια παραλλαγή της διαδικασίας της εφαρμοσμένης μηχανικής ενώ για τον προγραμματισμό τους χρησιμοποιήσαμε το προγραμματιστικό περιβάλλον της εφαρμογής του Lego Spike Prime.

### **1.1 Αντικείμενο της διπλωματικής εργασίας**

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι τα ρομποτικά συστήματα και ο προγραμματισμός τους στα πλαίσια της STEAM εκπαίδευσης. Η μελέτη αυτή περιέλαβε μία βιβλιογραφική έρευνα για τα προς μελέτη αντικείμενα και τον τρόπο με τον οποίο μπορεί κανείς να υλοποιήσει ολοκληρωμένα projects εκπαιδευτικής ρομποτικής, ενώ ταυτόχρονα κομμάτι της εργασίας είναι και η πρακτική εφαρμογή των αντικειμένων που μελετήσαμε θεωρητικά.

Αρωγός μας σε αυτήν την προσπάθεια στάθηκε το kit εκπαιδευτικής ρομποτικής Spike Prime, όπου αξιοποιώντας τις δυνατότητές του καταφέραμε να μελετήσουμε τον τρόπο δημιουργίας και προγραμματισμού εκπαιδευτικών ρομποτικών συστημάτων. Το Spike Prime είναι ένα kit εκπαιδευτικής ρομποτικής με μεγάλες δυνατότητες, καθώς αποτελείται από ένα πλήθος εξαρτημάτων, δίνοντας τη δυνατότητα στον χρήστη να δημιουργήσει πολλές και ενδιαφέρουσες εκπαιδευτικές εφαρμογές. Φυσικά ιδιαίτερη βαρύτητα δίνεται και στην παραγωγή του κώδικα, καθώς είναι το κομμάτι που ολοκληρώνει κάθε ρομποτικό σύστημα. Έπειτα από εμβάθυνση στη θεωρία και το εκπαιδευτικό kit δομήσαμε 5 ολοκληρωμένα projects, δημιουργώντας τόσο τον μηχανισμό τους όσο και τα αντίστοιχα προγράμματα τα οποία απαιτούνται για την ολοκλήρωσή τους.

## **1.2 Σκοπός της διπλωματικής εργασίας**

Σκοπός της διπλωματικής εργασίας είναι η πρακτική εφαρμογή των προαναφερθέντων αντικειμένων που μελετήθηκαν εκτενώς κατά την διάρκεια του μεταπτυχιακού προγράμματος, όπως η εκπαίδευση STEAM, η εκπαιδευτική ρομποτική και η συγγραφή κώδικα. Απώτερος σκοπός αυτής της ανάλυσης είναι η παρούσα εργασία να αποτελέσει σημαντική πηγή πληροφοριών για όσους ασχολούνται με την STEAM εκπαίδευση και την Εκπαιδευτική Ρομποτική, συνεπάγοντας την βελτίωση της παρεχόμενης εκπαίδευσης. Σαν τελικό έμμεσο αποδέκτη αυτής της εργασίας θεωρούμε όλους τους μαθητές οι οποίοι θα εμπλακούν σε αυτές τις δραστηριότητες. Έτσι, μέσα από την πρόταση για την δημιουργία ολοκληρωμένων projects εκπαιδευτικής ρομποτικής, επιθυμούμε να τονιστεί και η σημαντικότητα του να μπορεί κανείς να υλοποιεί τα δικά του ρομποτικά συστήματα και τις δικές του εφαρμογές προγραμματισμού, καθώς η απελευθέρωση από την εξάρτηση οποιουδήποτε υλικού παρέχεται μέσω διαδικτύου ή άλλου μέσου ανοίγει τον δρόμο σε άπειρες εκπαιδευτικές και δημιουργικές δυνατότητες.

### 1.3 Διάρθρωση της διπλωματικής εργασίας

Για την αποδοτικότερη κατανόηση της παρούσας εργασίας προσπαθήσαμε η συγγραφή αυτής να ξεκινήσει από τα θεωρητικά κεφάλαια, προκειμένου κάποιος να αποκτήσει τις γνώσεις που χρειάζεται για να την κατανοήσει στο έπακρο, και στη συνέχεια προχωρήσαμε στο πρακτικό κομμάτι, όπου έχοντας αναλύσει σε βάθος όλα όσα θα χρησιμοποιηθούν θεωρούμε ότι μπορεί να γίνει εύκολα κατανοητό από τον αναγνώστη. Στο 1<sup>ο</sup> Κεφάλαιο γίνεται μία πρώτη εισαγωγή στην εργασία, αναλύοντας το θέμα με το οποίο θα ασχοληθούμε. Στο Κεφάλαιο 2 γίνεται μια πλήρης ανάλυση της STEAM Εκπαίδευσης, ενώ στο Κεφάλαιο 3 παρουσιάζουμε κάποιες βασικές πληροφορίες για τον κλάδο της ρομποτικής και τα ρομποτικά συστήματα γενικότερα. Στο 4<sup>ο</sup> Κεφάλαιο παρουσιάζεται αναλυτικά ο κλάδος της Εκπαιδευτικής Ρομποτικής και στο Κεφάλαιο 5 ακολουθεί μια αναλυτική περιγραφή του εξοπλισμού του οποίου θα χρησιμοποιηθεί στα πλαίσια της εργασίας. Στο 6<sup>ο</sup> Κεφάλαιο θεωρήσαμε σκόπιμο να μιλήσουμε διεξοδικά για τους πιθανούς τρόπους προγραμματισμού του Spike Prime. Στο Κεφάλαιο 7 παρουσιάζουμε μια πρόταση για τη δημιουργία μαθησιακών δραστηριοτήτων στην STEAM εκπαίδευση, αναλύοντας τον τρόπο με τον οποίο μπορεί κανείς να σχεδιάσει και να προγραμματίσει τα δικά του ρομποτικά συστήματα. Στο Κεφάλαιο 8 παρουσιάζουμε αναλυτικά όλες τις δραστηριότητες τις οποίες υλοποιήσαμε στα πλαίσια αυτής της εργασίας, και τέλος, στο Κεφάλαιο 9 ακολουθούν τα συμπεράσματα στα οποία καταλήξαμε με την περάτωση της παρούσας εργασίας.



## **Κεφάλαιο 2 – STEAM Εκπαίδευση**

### **Εισαγωγή Κεφαλαίου.**

Κάθε βιομηχανική επανάσταση μεταφέρει στο εσωτερικό της νέες προσδοκίες στους τομείς της ανθρώπινης ανάπτυξης, ειδικά στον τομέα της εκπαίδευσης. Έτσι, η τέταρτη βιομηχανική επανάσταση (4IR – Fourth Industrial Revolution) επηρέασε τα εκπαιδευτικά προγράμματα σε ολόκληρο τον κόσμο και οδήγησε στον εκμοντερνισμό του εκπαιδευτικού συστήματος. Γι' αυτό, από τα τέλη του 20<sup>ου</sup> αιώνα και μετά, το ενδιαφέρον της εκπαιδευτικής έρευνας επικεντρώνεται στην εκπαίδευση STEAM. Ο πρωτοποριακός τρόπος προσέγγισης της γνώσης, σε αντιπαράθεση με τις παραδοσιακές μεθόδους, είναι ο βασικότερος λόγος για τον οποίο η STEAM προσέγγιση έχει εισαχθεί σε παγκόσμιο επίπεδο στον χώρο της εκπαίδευσης, καθώς και το γεγονός ότι η υιοθέτηση μιας τέτοιας προσέγγισης παρέχει ευκαιρίες στους μαθητές να αναπτύξουν τόσο τις γνώσεις τους στα πεδία που την απαρτίζουν, όσο και τις δεξιότητες τους, προετοιμάζοντάς τους ικανοποιητικά για τον ρόλο τους ως μελλοντικούς πολίτες μιας κοινωνίας.

### **2.1 Τι είναι η STEM Εκπαίδευση.**

Ο όρος STEM αποτελεί ένα ακρωνύμιο που αποτελείται από τους τέσσερις τομείς: της Φυσικής (Science), της Τεχνολογίας (Technology), της Μηχανικής (Engineering) και των Μαθηματικών (Mathematics). Αρχικά ο όρος εμφανίστηκε ως "SMET", που σημαίνει Επιστήμη, Μαθηματικά, Μηχανική, και Τεχνολογία. Ο όρος επινοήθηκε το 1990 από το Εθνικό Ίδρυμα Επιστημών (NSF), στη συνέχεια όμως αντικαταστάθηκε με μια πολύ πιο ελκυστική σειρά γραμμάτων και μετατράπηκε σε STEM από την βιολόγο Judith A. Ramaley με σκοπό την πιο εύκολη προώθησή του.

Δεδομένου ότι η εκπαίδευση είναι μια βαθιά πολιτική (με την Αριστοτελική έννοια) πράξη και ένας από τους βασικούς σκοπούς της είναι να προετοιμάσει μελλοντικούς πολίτες, που θα δρουν αποτελεσματικά σε μια κοινωνία, η STEM εκπαίδευση εμφανίστηκε ως μια ανάγκη να συμβαδίσει η εκπαίδευση με τις κοινωνικές, οικονομικές και πολιτικές εξελίξεις που συντελούνται σε παγκόσμιο

επίπεδο. Μερικοί από τους λόγους που οδήγησαν σε αυτή την μεταρρυθμιστική πρωτοβουλία είναι:

- Η φυσική, τα μαθηματικά, η μηχανική και η τεχνολογία είναι πεδία στα οποία επικεντρώνεται το ενδιαφέρον των ΗΠΑ (όπου πρωτοεμφανίστηκε και ο όρος STEM) με σκοπό να παράγουν ικανούς εργαζομένους να ανταπεξέλθουν στην ανταγωνιστική μελλοντική παγκόσμια αγορά (Chesky, 2015).
- Η φυσική, τα μαθηματικά, η μηχανική και η τεχνολογία είναι από τη φύση τους πεδία τα οποία είναι άρρηκτα συνδεδεμένα μεταξύ τους. Συνεπώς ένας μαθητής που μελετά αυτά τα πεδία μέσα από πρακτικές δραστηριότητες θα μπορέσει να τα κατανοήσει ευκολότερα αξιοποιώντας τη διασύνδεσή τους.
- Αποκτώντας υψηλού επιπέδου γνώσεις σε αυτούς τους τέσσερις επιστημονικούς τομείς θα δημιουργηθούν ικανοί μελλοντικοί πολίτες που θα μπορούν να δρουν αποτελεσματικά σε μια δημοκρατική κοινωνία. Με αυτό τον τρόπο θα ωφεληθούν όλοι οι μαθητές, ειδικά αυτοί που ανήκουν σε μειονοτικές ομάδες και είχαν περιορισμένη πρόσβαση στην γνώση, με αποτέλεσμα τη δυσκολία εύρεσης εργασίας, αλλά και την αναποτελεσματική συμπεριφορά τους ως πολίτες (π.χ. δυσκολία στο να κατανοήσουν τη διαδικασία του δανείου) (Brown et al., 2011; Bybee, 2010).
- Τέλος, η ζήτηση εργαζομένων αλλά και η αύξηση των θέσεων εργασίας σε πεδία που σχετίζονται με τα STEM πεδία είναι ένας από τους λόγους που οδήγησαν σε αυτή τη μεταρρύθμιση. Λύνοντας προβλήματα του πραγματικού κόσμου κεντρίζεται το ενδιαφέρον των μαθητών στους χώρους της φυσικής, των μαθηματικών, της μηχανικής και της τεχνολογίας, καθώς οι μαθητές γνωρίζουν βιωματικά τα συγκεκριμένα πεδία. Το πλεονέκτημα που θα προσφέρει η αύξηση αυτή του ενδιαφέροντος είναι ότι υπολογίζεται αντίστοιχα να αυξηθεί και ο αριθμός των μαθητών που θα ασχοληθούν με επαγγέλματα που σχετίζονται με τα STEM πεδία (Brown et al., 2011).

Παρότι όπως προαναφέρθηκε ο όρος STEM αποτελεί ένα ακρωνύμιο που αποτελείται από τους τέσσερις τομείς της Φυσικής (Science), της Τεχνολογίας (Technology), της Μηχανικής (Engineering) και των Μαθηματικών (Mathematics), δεν είναι εύκολο να δοθεί ένας ακριβής ορισμός για το τι σημαίνει πραγματικά το STEM.

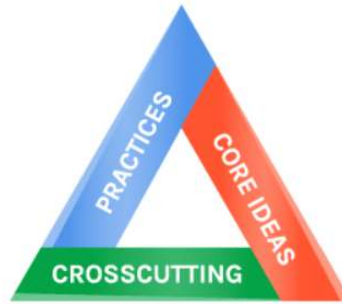
Σε έρευνα που πραγματοποίησαν οι Breiner et al. (2012), τόσο σε μέλη κλάδων STEM όσο και σε μέλη από κλάδους που δεν ανήκουν στο STEM, προέκυψε πως δεν είναι δυνατό να υπάρξει ένας κοινός ορισμός για την εννοιολόγηση του STEM. Επιπρόσθετα, ο Marrero (2014) παραθέτοντας τον Bybee (2010) αναφέρει πως «..ακόμη και οι επαγγελματίες σε τομείς STEM που ερωτήθηκαν δεν φαίνεται να κατανοούν τι σημαίνει το ακρωνύμιο, υποθέτοντας συχνά ότι σχετίζεται με τη βοτανική ή την έρευνα σε βλαστοκύτταρα».

Για κάποιους λοιπόν, το STEM είναι απλά η αντικατάσταση των παραδοσιακών μεθόδων και διδακτικών πρακτικών με δραστηριότητες που περιλαμβάνουν περισσότερη έρευνα και είναι βασισμένες σε projects. Από την άλλη πλευρά, υπάρχουν κι αυτοί που υποστηρίζουν πως για να χαρακτηριστεί ένα πρόγραμμα σπουδών ως STEM πρόγραμμα θα πρέπει να ενσωματώνει προγράμματα της επιστήμης, της τεχνολογίας, της μηχανικής και των μαθηματικών που θυμίζουν και παραλληλίζουν το έργο ενός επιστήμονα ή μηχανικού. Τέλος, για κάποιους το STEM είναι η δημιουργία προγραμμάτων που ενσωματώνουν αυτά τα τέσσερα πεδία με σκοπό το να ωθήσουν περισσότερους φοιτητές να αποφοιτήσουν από τα συγκεκριμένα πεδία, ώστε να μπορούν οι Ηνωμένες Πολιτείες να διατηρήσουν την ανταγωνιστικότητά τους και να διατηρήσουν την κυριαρχία τους στις αναδυόμενες χώρες (Marrero, 2014).

Παρόλη τη δυσκολία εννοιολόγησης του όρου STEM, σε παιδαγωγικό επίπεδο, για να χαρακτηριστεί ένα πρόγραμμα σπουδών ως STEM πρόγραμμα θα πρέπει να κινείται πάνω σε ένα πλαίσιο το οποίο αποτελείται από τρεις διαστάσεις (Quinn et al., 2012):

Η πρώτη διάσταση αφορά τις επιστημονικές πρακτικές και τις πρακτικές της μηχανικής.

1. Η δεύτερη διάσταση αφορά τις διασταυρούμενες επιστημονικές ιδέες που ισχύουν σε όλους τους επιστημονικούς κλάδους και αφορούν τον επιστημονικό γραμματισμό (Duschl, 2012).
2. Η τρίτη διάσταση αφορά τις βασικές ιδέες στα επιστημονικά πεδία και στις σχέσεις μεταξύ της επιστήμης, της μηχανικής και της τεχνολογίας.



Εικόνα 2.1 Οι τρεις διαστάσεις της STEM Εκπαίδευσης.

Πρώτη διάσταση: Πρακτικές.

Η πρώτη διάσταση, οι πρακτικές, χρησιμοποιείται για να περιγραφούν οι βασικές πρακτικές που χρησιμοποιούν οι επιστήμονες στην έρευνα, προκειμένου να δημιουργήσουν μοντέλα για τη διερεύνηση ενός φαινομένου και να αναπτύξουν θεωρίες για το πώς λειτουργεί ο κόσμος. Ταυτόχρονα, χρησιμοποιείται για να δηλώσει τις πρακτικές της μηχανικής που χρησιμοποιούν οι μηχανικοί προκειμένου να αναπτύξουν και να σχεδιάσουν νέα συστήματα. Ο όρος «πρακτικές» χρησιμοποιείται για να δηλώσει το γεγονός πως η επιστημονική έρευνα απαιτεί συγκεκριμένες γνώσεις και πρακτικές, και πως δεν επαρκούν μόνο οι δεξιότητες σε έναν τομέα. Σκοπός είναι να κατανοήσουν οι μαθητές σε βάθος το τι σημαίνει η διερεύνηση φαινομένων μέσα από επιστημονικές πρακτικές και να τις κατακτήσουν συμμετέχοντας οι ίδιοι σε δραστηριότητες επίλυσης πραγματικών προβλημάτων, προσεγγίζοντας τις πρακτικές αυτές βιωματικά.

Δεύτερη διάσταση: Διασταυρούμενες έννοιες.

Το National Research Council's Framework for K-12 Science Education (2012) δηλώνει ότι οι διασταυρούμενες έννοιες είναι «έννοιες που γεφυρώνουν τα όρια του πυρήνα των πεδίων, έχοντας επεξηγηματική αξία σε μεγάλο μέρος της επιστήμης και της μηχανικής. Αυτές οι έννοιες παρέχουν στους μαθητές ένα οργανωτικό πλαίσιο για τη σύνδεση της γνώσης από τους διάφορους κλάδους σε μια συνεκτική και επιστημονικά βασισμένη άποψη για τον κόσμο». Οι 7 βασικές διασταυρούμενες ιδέες είναι η εξής: 1) Μοτίβα, 2) Αίτιο-Αποτέλεσμα: μηχανισμοί και εξηγήσεις, 3)

Κλίμακα/Αναλογία/Ποσότητα, 4) Συστήματα και Μοντέλα Συστημάτων, 5) Ενέργεια και Ύλη: ροή/κύκλος/διατήρηση, 6) Δομή και Λειτουργικότητα και τέλος 7) Σταθερότητα και Αλλαγή (Duschl, 2012). Βασικός στόχος αυτής της διάστασης είναι να παρέχει στους μαθητές μια συνεκτική και επιστημονικά βασισμένη άποψη για τον κόσμο, δημιουργώντας ένα οργανωτικό πλαίσιο για τη σύνδεση της γνώσης από τους διάφορους κλάδους.

Τρίτη διάσταση: Βασικές ιδέες των επιστημονικών πεδίων.

Η συνεχής επέκταση των επιστημονικών γνώσεων, όπως επίσης και ο καταιγισμός πληροφοριών που βιώνουν οι άνθρωποι σε καθημερινή βάση, καθιστά απίθανη τη διδασκαλία όλων των εννοιών και ιδεών που συσχετίζονται με ένα επιστημονικό πεδίο, αλλά και την εμβάθυνση σε αυτές τις ιδέες σε ένα μεγάλο επίπεδο λεπτομερειών. Γι' αυτό και σκοπός της STEM εκπαίδευσης δεν είναι να διδάξει όλες ή όσες περισσότερες ιδέες και έννοιες δύναται, αλλά να παρέχει στους μαθητές κάποιες βασικές επιστημονικές ιδέες και γνώσεις, ώστε μετέπειτα να μπορέσουν να εξελίξουν αυτές τις γνώσεις προσθέτοντας από μόνοι τους νέες πληροφορίες. Η επικέντρωση αυτή σε έναν περιορισμένο αριθμό γνώσεων που πρέπει να διδαχθούν δίνει στους μαθητές τη δυνατότητα να αποκτήσουν κατά τη διάρκεια των μαθητικών τους χρόνων τις βασικές ιδέες, μετέπειτα να τις εξελίξουν και ενδεχομένως να γίνουν και παραγωγοί τέτοιων γνώσεων ως νέοι επιστήμονες. Οι Quinn et al. (2012) προκειμένου να τονίσουν ποιες είναι οι βασικές ιδέες που πρέπει να διδαχθούν τις κατηγοριοποίησαν σε τέσσερις ομάδες, τις οποίες αναφέρουν ως: «φυσικές επιστήμες, επιστήμες της ζωής, επιστήμες της Γης και του διαστήματος, και τέλος, επιστήμες της μηχανικής, της τεχνολογίας και των εφαρμογών της επιστήμης».

## **2.2 Τα αντικείμενα της STEM Εκπαίδευσης.**

Όπως προαναφέρθηκε, η εκπαίδευση STEM είναι μια διδακτική προσέγγιση που συνδυάζει την επιστήμη, την τεχνολογία, τη μηχανική και τα μαθηματικά. Θα ήταν όμως ανακριβές να υποθέσουμε ότι η εκπαίδευση STEM αντιμετωπίζει την επιστήμη, την τεχνολογία, την μηχανική και τα μαθηματικά ως ξεχωριστά και διακριτά

εκπαιδευτικά θέματα. Αντίθετα, ο βασικός πυρήνας της είναι η ενσωμάτωση αυτών των τεσσάρων πεδίων σε μια ενιαία και επιστημονική μαθησιακή διαδικασία που βασίζεται στην επίλυση προβλημάτων του πραγματικού κόσμου. Δηλαδή, η εκπαίδευση STEM στοχεύει στη μετατόπιση του παραδείγματος από την παραδοσιακή διδασκαλία σε μια ερευνητική διδασκαλία βασισμένη σε projects που έχουν ως βάση το πρόβλημα, το οποίο αποτελεί το μέσο εξερεύνησης και απόκτησης διεπιστημονικών μαθησιακών εμπειριών σε δύο ή περισσότερα από τα τέσσερα κύρια πεδία που απαρτίζουν την εκπαίδευση STEM (Mansour, 2015).

Αναλυτικά τα πεδία της STEM εκπαίδευσης:

- Science (S): Ως «Επιστήμη» ορίζεται η μελέτη του φυσικού κόσμου και των νόμων που διέπουν τον φυσικό κόσμο και συνδέονται με τη φυσική, τη χημεία και τη βιολογία. Αφορά την επεξεργασία ή τις εφαρμογές γεγονότων, την ανακάλυψη των αρχών, των εννοιών και των συμβάσεων που σχετίζονται με αυτούς τους κλάδους (Bahrum, 2017).
- Technology (T): Ο όρος "τεχνολογία" αποτελεί έναν συνδυασμό των λέξεων "τέχνη", που αναφέρεται κυρίως στην κατασκευαστική τέχνη, και από τη λέξη «λόγος», που σημαίνει λέξη, σκέψη, ομιλία ή αρχή. Από αυτή την ανάλυση γίνεται σαφές ότι ως τεχνολογία ορίζεται η προσπάθεια ή η μελέτη της τέχνης των κατασκευών (Koc, 2018). Όταν αναφερόμαστε στον όρο τεχνολογία στην εκπαίδευση συνήθως εννοούμε μόνο τα τεχνολογικά εργαλεία και τα χαρακτηριστικά τους. Στην STEM εκπαίδευση όμως ο όρος χρησιμοποιείται ως «ομπρέλα» που αγκαλιάζει ολόκληρο το σύστημα δημιουργίας και οργάνωσης της διαδικασίας παραγωγής τεχνουργημάτων, από τους ανθρώπους που χρειάζονται για να κατασκευαστεί ένα νέο τεχνολογικό προϊόν, στις γνώσεις, διεργασίες και συσκευές που απαιτούνται για τη δημιουργία και τη σωστή λειτουργία τους, καθώς και τα ίδια τα τεχνουργήματα. Σκοπός της ενσωμάτωσης της τεχνολογίας σε ένα τέτοιο εκπαιδευτικό πλαίσιο είναι οι μαθητές να αναπτύξουν τεχνολογικούς γραμματισμούς, να ανακαλύψουν τι είναι η τεχνολογία γύρω τους, πώς παράγεται και πώς λειτουργεί, προκειμένου να μπορέσουν με τη σειρά τους να ανταπεξέλθουν ως τεχνολογικά εγγράμματοι πολίτες σε μια τεχνολογική κοινωνία. Αναφορικά με την τεχνολογία είναι επίσης πολύ σημαντικό οι μαθητές να απομυθοποιήσουν την «μαγεία» που κρύβεται πίσω από την τεχνολογία, για παράδειγμα το άνοιγμα των

αυτόματων πορτών, και να κατανοήσουν πως είναι όλα αποτέλεσμα ανθρώπινης σκέψης και δράσης (Cetin et al.2018).

- **Engineering (E):** Η μηχανική είναι το επιστημονικό πεδίο που ασχολείται με την διερεύνηση και ανακάλυψη βέλτιστων λύσεων σχεδιασμού και διαδικασιών για την παραγωγή προϊόντων και διαδικασιών, μελετώντας τους φυσικούς και μη περιορισμούς (π.χ. αντοχή των υλικών, κόστος) των μηχανικών σχεδιασμών και ανακαλύπτοντας τρόπους για να τους ξεπεράσει. Ταυτόχρονα ασχολείται με την παραγωγή διαδικασιών για την επίλυση προβλημάτων. Η μηχανική είναι από τη φύση της ένα πολύ ισχυρό STEM εργαλείο, καθώς χρησιμοποιεί ως εργαλεία της τους νόμους της φυσικής, τα μαθηματικά και την τεχνολογία. (National Research Council, 2009). Είναι σκόπιμο να τονίσουμε πως η ενασχόληση με τη μηχανική δεν είναι κάτι δύσκολο για τα παιδιά, καθώς παρουσιάζει πολλά κοινά στοιχεία με την φυσική περιέργειά τους για εξερεύνηση. Τα παιδιά από τη φύση τους είναι περίεργα και δημιουργικά, γεγονός που ενισχύει τη διάθεσή τους να εξερευνήσουν και να πειραματιστούν. Τα χαρακτηριστικά αυτά αντικατοπτρίζουν εκείνα που διέπουν και τους μηχανικούς (Adams et al., 2011).
- **Mathematics (M):** Τα μαθηματικά είναι το πεδίο επιστήμης που ασχολείται με τη μελέτη των προτύπων και των σχέσεων μεταξύ μεγεθών, αριθμών, και σχημάτων (Bahrum, 2017). Είναι πολύ σημαντικό πως το άτομο, πέρα από μαθηματικούς γραμματισμούς, μέσα από την STEM εκπαίδευση καταφέρνει να διαπιστώσει τον τρόπο με τον οποίο τα μαθηματικά συνδέονται με τον πραγματικό κόσμο, καθώς εντάσσεται μέσα σε διάφορες προβληματικές του πραγματικού κόσμου. Χρησιμοποιώντας τα μαθηματικά ως εργαλείο το άτομο αναπτύσσει ικανότητες, ώστε να μπορεί να αναλύει αποτελεσματικά το περιβάλλον γύρω του, και να παρεμβαίνει σε κοινωνικά ζητήματα χρησιμοποιώντας τη μαθηματική σκέψη για τη λήψη αποφάσεων. Χαρακτηριστικό είναι το γεγονός πως είναι σχεδόν απίθανο να οδηγηθεί κανείς στην επίλυση ενός προβλήματος χωρίς τη συμβολή μαθηματικών πεποιθήσεων (Kwon, 2021).

Για να θεωρηθεί όμως πετυχημένο ένα STEM πρόγραμμα σπουδών θα πρέπει, όχι μόνο να συνδυάζει τα συγκεκριμένα πεδία, αλλά να πληροί και κάποια συγκεκριμένα χαρακτηριστικά:

Το πρώτο χαρακτηριστικό που πρέπει να ληφθεί υπόψιν για τη δημιουργία ενός ολοκληρωμένου προγράμματος σπουδών STEM, όντας ιδιαίτερα σημαντικό, είναι η δημιουργία ενός ελκυστικού προγράμματος που θα παρακινεί τους μαθητές στη γνώση μέσα από δομημένες δραστηριότητες σε μια ενότητα. Οι δραστηριότητες αυτές θα πρέπει να προωθούν την γνώση που θα δομηθεί πάνω στις ήδη υπάρχουσες γνώσεις των μαθητών (Guzey et al., 2016). Είναι σημαντικό επίσης οι δραστηριότητες να βρίσκονται μέσα σε ένα μαθησιακό πλαίσιο που θα έχει νόημα για τους μαθητές και θα σχετίζεται με την πραγματική ζωή (Bahrum, 2017).

Το δεύτερο χαρακτηριστικό είναι η δημιουργία προκλήσεων στο πεδίο της μηχανικής και της διαδικασίας μηχανικού σχεδιασμού. Μέσα από αυτές τις προκλήσεις οι μαθητές θα καταφέρουν να ανακαλύψουν τη γνώση που κρύβεται πίσω από αυτό το πεδίο ενώ ταυτόχρονα θα αναπτύξουν δεξιότητες όπως η κριτική και δημιουργική σκέψη και η επίλυση προβλημάτων. Για να θεωρηθεί εκπαιδευτικά ωφέλιμη μια μηχανική πρόκληση θα πρέπει να προκαλεί τους μαθητές να λάβουν υπόψιν και άλλους παράγοντες όπως είναι η ασφάλεια, ο κίνδυνος, οι περιορισμοί στο σχεδιασμό αλλά και να προτείνουν εναλλακτικές λύσεις.

Εξίσου σημαντικό στοιχείο μέσα σε ένα STEM πρόγραμμα σπουδών είναι η προώθηση των λαθών ως μέσο που οδηγεί στο δρόμο της γνώσης. Η μάθηση από την αποτυχία είναι ένα κρίσιμο μέρος μιας τέτοιας διαδικασίας (Kolodner et al., 2003; Wendell & Rogers, 2013). Οι μαθητές θα πρέπει να συμφιλιωθούν με την ιδέα των λαθών έτσι ώστε να ξεπεράσουν τον συγκεκριμένο φόβο και να μπορούν να πειραματίζονται ελεύθερα, μαθαίνοντας μέσα από τα λάθη τους.

Τέταρτο χαρακτηριστικό είναι ότι το ενδιαφέρον του μαθήματος μετατοπίζεται από τον δάσκαλο στον μαθητή, ο οποίος πλέον βρίσκεται στο επίκεντρο της μαθησιακής διαδικασίας. Στην STEM εκπαίδευση ο μαθητής σταματά να είναι παθητικός αποδέκτης της γνώσης και αναλαμβάνει ενεργό ρόλο, συμπεριφέροντας ταυτόχρονα και ως παράγωγος της γνώσης και αποδέκτης της.

Τέλος, είναι κρίσιμο οι μαθητές μέσα από τις STEM δραστηριότητες να αναπτύξουν την κοινωνικότητα τους, την επικοινωνία και το ομαδικό πνεύμα τους, δεξιότητες που θεωρούνται πολύ κρίσιμες για τον 21<sup>ο</sup> αιώνα, γι' αυτό και οι STEM δραστηριότητες είναι κατά βάση ομαδικές δραστηριότητες. Δηλαδή, οι μαθητές χωρισμένοι σε ομάδες αναλαμβάνουν κάθε φορά από μια αποστολή/πρόβλημα που εμπλέκει τη μηχανική, τη φυσική, την τεχνολογία και τα μαθηματικά που θα πρέπει να



επιλύσουν, και σαν ομάδα να φτάσουν σε ένα τελικό αποτέλεσμα (Carlson & Sullivan, 2004).

### **2.3 Από το STEM στο STEAM.**

Συνδυάζοντας λοιπόν αυτά τα τέσσερα πεδία και τα χαρακτηριστικά που προαναφέρθηκαν θεωρήθηκε ότι τα STEM προγράμματα σπουδών είναι ικανά να παρέχουν στους μαθητές τις γνώσεις και τις δεξιότητες που χρειάζονται για να λειτουργήσουν ως πολίτες μιας κοινωνίας. Στη συνέχεια όμως παρατηρήθηκε πως τα τέσσερα αυτά πεδία κι ένας προσεκτικός σχεδιασμός των προγραμμάτων δεν επαρκούν για να οδηγήσουν στην καινοτομία που χρειάζεται ο 21<sup>ος</sup> αιώνας. Θεωρείται πως για να επιτευχθεί η καινοτομία θα πρέπει να συνδυαστούν χαρακτηριστικά που μέχρι τώρα θεωρούνταν αντικρουόμενα, όπως είναι η λογική και η διαίσθηση ή η μοναδικότητα ενός προϊόντος με την τυποποιημένη αναπαραγωγιμότητα και επαναληψιμότητά του, καθώς η καινοτομία εμφανίζεται όταν στοχαστές οι οποίοι ακολουθούν μια συγκεκριμένη ευθεία για να φτάσουν στον στόχο τους συναντούν εκείνους οι οποίοι σκέφτονται και περιπλανώνται ελεύθερα (Maeda, 2013). Για τον λόγο αυτό τα τελευταία χρόνια υπήρξε μια στροφή από το STEM στο STEAM, προστέθηκε δηλαδή στα τέσσερα επιστημονικά πεδία και το πεδίο των Arts, των τεχνών και των ανθρωπιστικών επιστημών. Παρατηρήθηκε λοιπόν πως, όπως αναφέρουν και οι Perignat & Buonincontro (2018): «Η τέχνη και η επιστήμη – κάποτε άρρηκτα συνδεδεμένα, και τα δύο αφοσιωμένα στην εύρεση της αλήθειας και της ομορφιάς – είναι καλύτερα μαζί παρά χώρια.» Έτσι θεωρήθηκε κρίσιμο για να καταφέρουν οι μαθητές να αναπτύξουν δεξιότητες που θεωρούνται δεξιότητες του 21<sup>ου</sup>, όπως είναι η δημιουργικότητα, η συνεργασία, η κριτική και αναλυτική σκέψη, να έρθουν πιο κοντά τα πεδία του STEM με τις τέχνες (Piro, 2010).

### **2.4 Επεξηγώντας το Α στο STEAM.**

Παρότι η STEAM εκπαίδευση βρίσκεται στο επίκεντρο της επιστημονικής έρευνας, τα τελευταία χρόνια δεν υπάρχει ένας σαφής ορισμός για το τι ακριβώς σημαίνει η ενσωμάτωση του Α στην STEM εκπαίδευση. Οι διάφοροι ορισμοί και η

ασάφεια του όρου «Arts» δημιουργεί πρόβλημα στην ενσωμάτωση του όρου μέσα σε ένα εκπαιδευτικό πρόγραμμα, καθώς οι εκπαιδευτικοί δεν μπορούν να ορίσουν τι ακριβώς σημαίνει, πώς θα πετύχουν την ενσωμάτωση του και πώς οι μαθητές θα καταφέρουν να αναπτύξουν τη δημιουργικότητά τους και άλλες γνωστικές δεξιότητες. Γι' αυτό και οι Perignat και Buonincontro (2018) μέσα από μια βιβλιογραφική ανασκόπηση ανέδειξαν τρεις ορισμούς για το A στο STEAM:

- Ο όρος Arts χρησιμοποιείται για να περιγράψει όλα τα non-STEM/ μη-STEM πεδία. Χρησιμοποιείται δηλαδή σαν όρος που αγκαλιάζει όλα τα υπόλοιπα πεδία, όπως είναι οι ανθρωπιστικές επιστήμες, οι φιλελεύθερες τέχνες ή οι επιστήμες του περιβάλλοντος.
- Ο δεύτερος ορισμός που δίνεται συχνά είναι ότι το A αντιπροσωπεύει τις εικαστικές τέχνες, όπως είναι η μουσική, το θέατρο, η ζωγραφική και η γλυπτική.
- Τέλος, ο όρος χρησιμοποιείται ως συνώνυμο για τους όρους μάθηση βάσει έργου (Project based learning), μάθηση μέσω επίλυσης προβλημάτων (Problem-Based Learning) και μάθηση βασισμένη στην τεχνολογία (Technology-Based Learning/Making). Σε αυτή την περίπτωση ο όρος Arts αποκτά έναν πιο τεχνολογικό ορισμό και δεν αναφέρεται τόσο στις εικαστικές τέχνες όπως συνηθίζεται, παρά συνδέεται περισσότερο με τη διαδικασία σχεδιασμού και κατασκευής διαφόρων έργων.

Συνεπώς, ο όρος Arts προσεγγίζεται διαφορετικά από την επιστημονική κοινότητα και ο τρόπος με τον οποίο θα ενσωματωθεί σε ένα εκπαιδευτικό πρόγραμμα και το τι θα προσφέρει στους μαθητές επαφίεται στον εκάστοτε εκπαιδευτικό και στον ορισμό με τον οποίο θα προσεγγίσει κάθε φορά τον όρο.

Όμως, η δυσκολία αυτή του ορισμού δεν μπορεί οδηγήσει στο να αγνοηθούν τα πλεονεκτήματα που προσφέρει η ενσωμάτωση του όρου Arts (όπως κι αν αυτός προσεγγίζεται), και πώς η παρουσία του είναι αναγκαία για να επιτευχθούν μέγιστα μαθησιακά αποτελέσματα και να αποκτήσουν οι μαθητές τις γνώσεις και τις δεξιότητες που θα χρειαστούν στη ζωή τους μετά το σχολείο.

Ένα βασικό πλεονέκτημα που προσφέρει η ενσωμάτωση των τεχνών είναι ότι τα STEM πεδία παρέχουν μια αντικειμενική προσέγγιση του κόσμου, παρουσιάζοντας τους νόμους που διέπουν τον φυσικό κόσμο για παράδειγμα, οι οποίοι είναι

αντικειμενικοί και έχουν καθολική ισχύ, ενώ οι τέχνες μάς βοηθούν να δούμε τον κόσμο μέσα από μια πιο υποκειμενική ματιά. Σε πολλές περιπτώσεις ο εγκέφαλος ενός ατόμου χρειάζεται να μπορεί να κρίνει τα δεδομένα τόσο υποκειμενικά όσο και αντικειμενικά για να λάβει τις κατάλληλες αποφάσεις (Sousa & Pilecki, 2018).

Επιπρόσθετα, ο 21<sup>ος</sup> αιώνας άνοιξε τον δρόμο για δεξιότητες που εκτείνονται πέρα από τους κειμενικούς γραμματισμούς. Οι νέοι γραμματισμοί που αναδύθηκαν κυρίως μέσα από τον χώρο του διαδικτύου και των social media αφορούν την κατανόηση αλλά και παραγωγή οπτικών υλικών, όπως για παράδειγμα βίντεο εκπαιδευτικού περιεχομένου ή προσομοιώσεις. Οι νέοι αυτοί γραμματισμοί μπορούν εύκολα να εξερευνηθούν και προσεγγιστούν μέσα από την τέχνη (Land, 2013).

Σε επίπεδο δεξιοτήτων, οι τέχνες ενισχύουν τη δημιουργική σκέψη και τη δημιουργικότητα στην επίλυση προβλημάτων, καθώς επίσης προωθούν και την αυτοέκφραση (Malley & Silverstein, 2014). Επιπρόσθετα, οι τέχνες επιτρέπουν στους μαθητές να αναπτύξουν ταυτόχρονα πολλαπλές ικανότητες και να αναπτύξουν τις γνωστικές και κοινωνικές του δεξιότητες (Lloyd, 2017), ενώ ταυτόχρονα εξερευνούν και γνωρίζουν τον εαυτό τους, γεγονός που μπορεί να τους οδηγήσει σε μια διαδικασία συνεχής αυτοβελτίωσης. Σημαντικό είναι επίσης το γεγονός ότι οι τέχνες συμβάλλουν στην βελτίωση της μακροπρόθεσμης μνήμης, εισάγουν την καινοτομία και μειώνουν το άγχος (Sousa, Pilecki 2018).

Τέλος, αξίζει να αναφερθεί πως ο όρος Arts, είτε προσεγγίζεται με την έννοια των τεχνών, είτε με των ανθρωπιστικών επιστημών, προσφέρει στους μαθητές την ευκαιρία να εξερευνήσουν την ανθρώπινη φύση και εμπειρία. Ενθαρρύνουν τους μαθητές να αναζητήσουν νέες λύσεις στα ανθρώπινα προβλήματα χρησιμοποιώντας την ανθρώπινη εμπειρία, και σε συνδυασμό με την τεχνολογία να μπορέσουν να προτείνουν αποτελεσματικές ενέργειες έναντι σε προβλήματα βασικών αναγκών ή επιθυμιών με τρόπο που δεν έχει προταθεί ξανά μέσα στην ιστορία (Aithal, 2020).

## **2.5 Πλεονεκτήματα STEAM εκπαίδευσης.**

Όπως προαναφέρθηκε, μια σημαντική λειτουργία του εκπαιδευτικού συστήματος είναι να εξοπλίζει τους μαθητές με εφόδια, και να τους προετοιμάζει για τις προκλήσεις της ζωής. Είναι κατανοητό λοιπόν πως η παραδοσιακή διδασκαλία

επιδιώκει να προσφέρει στους μαθητές μόνο τις βασικές ικανότητες της ανάγνωσης, γραφής, απομνημόνευσης και ανάκλησης πληροφοριών και δεν επαρκεί προκειμένου οι μαθητές να αποκτήσουν τις δεξιότητες και τις ικανότητες που χρειάζονται τον 21<sup>ο</sup> αιώνα.

Σε αντίθεση με τις παραδοσιακές μεθόδους διδασκαλίας, η STEAM εκπαίδευση παρέχει στους μαθητές μια πληθώρα πλεονεκτημάτων, και για αυτούς τους λόγους θα πρέπει να προωθείται μια εκπαίδευση τέτοιου τύπου έναντι των παραδοσιακών μεθόδων.

Αρχικά, η STEAM εκπαίδευση παρέχει την ευκαιρία στους μαθητές να ανακαλύπτουν την γνώση μέσα από αυθεντικές εργασίες και προβλήματα που τίθενται σε πραγματικό πλαίσιο (Reeves et al., 2004). Ενσωματώνοντας διάφορους κλάδους, οι αυθεντικές εργασίες αποτελούνται από σύνθετες ερωτήσεις ή ερωτήσεις πολλαπλών βημάτων και προωθούν τους πολλαπλούς τρόπους προσέγγισης ενός πραγματικού προβλήματος (Bertrand et al., 2020). Το σημαντικό πλεονέκτημα που προσφέρουν αυτές οι βασισμένες σε πραγματικά προβλήματα δραστηριότητες είναι ότι δημιουργούν μια πρακτική σύνδεση των φυσικών επιστημών, της τεχνολογίας, της μηχανικής, των τεχνών και των μαθηματικών με την πραγματική ζωή, δημιουργούν μια αυθεντική εμπειρία διδασκαλίας-μάθησης και προωθούν την συνεχή ανακάλυψη του κόσμου, άρα και κατά συνέπεια την βιωματική μάθηση (Belbase et al., 2019). Έτσι, οι μαθητές αναπτύσσουν την περιέργεια τους και αναζητούν δημιουργικές και αποδοτικές λύσεις σε προβλήματα του πραγματικού κόσμου (Okeke et al., 2020).

Επιπρόσθετα, μέσα από την STEAM εκπαίδευση οι μαθητές αποκτούν ένα σύνολο δεξιοτήτων που θα τους βοηθήσουν στην οικοδόμηση του χαρακτήρα τους, και οι οποίες μπορούν να μεταφερθούν και να φανούν χρήσιμες και σε άλλα πλαίσια της ζωής τους, όπως είναι η τριτοβάθμια εκπαίδευση ή η ενσωμάτωσή τους στο εργατικό δυναμικό μιας κοινωνίας (Bertrand et al., 2020). Τέτοιες δεξιότητες για παράδειγμα είναι η κριτική σκέψη, η δημιουργικότητα και η καινοτομία, η επικοινωνία, η συνεργασία μέσω παρατήρησης, η έρευνα και επίλυση προβλημάτων (May et al., 2019). Επιπλέον, η διεπιστημονική προσέγγιση που προωθεί η STEAM εκπαίδευση, συνδυάζοντας τις τέχνες και τις ανθρωπιστικές επιστήμες με τα πεδία STEM, μπορεί να δημιουργήσει ένα δημιουργικό εργατικό δυναμικό στο μέλλον, ενισχύοντας την αποκλίνουσα σκέψη με σκοπό την επίλυση προβλημάτων (Shi et al., 2021). Τέλος,

παρατηρήθηκε πως όταν οι μαθητές συνδέουν τη μάθησή τους με πρακτικές της τέχνης, εφαρμόζουν τα εργαλεία που χρειάζονται για τη δημιουργία και την έρευνα ενός έργου τέχνης και σε άλλα έργα. Με αυτό τον τρόπο, οι μαθητές βασίζονται λιγότερο στην απομνημόνευση και περισσότερο στην πραγματική κατανόηση των όσων μελετούν (Okeke et al., 2022).

Μια άλλη σημαντική πτυχή της εκπαίδευσης STEAM είναι ότι προωθεί την συλλογική εργασία και την συνεργατική μάθηση όχι μόνο ανάμεσα στους μαθητές, αλλά και μεταξύ των δασκάλων. Αρχικά, η εκπαίδευση STEAM δημιουργεί ένα συμπεριληπτικό περιβάλλον χωρίς αποκλεισμούς, όπου όλοι οι μαθητές μπορούν να συμμετέχουν και να συνεισφέρουν (Okeke et al., 2020). Η συνεργατική φύση της επιτρέπει σε όλους τους μαθητές να μοιράζονται τις ιδέες και τις εμπειρίες τους στην τάξη (Belbase et al., 2019). Μέσω της συνεργασίας στη διαδικασία διδασκαλίας και μάθησης, οι μαθητές εκπαιδεύονται να είναι πιο ανεκτικοί, πιο συνεργάσιμοι, να σέβονται τους ανθρώπους, αλλά και το περιβάλλον στο οποίο βρίσκονται, ενισχύοντας έτσι και την κοινωνικοποίησή τους (Shi et al, 2021). Τέλος, η σύγκλιση όλων αυτών των επιστημονικών πεδίων ανοίγει νέους δρόμους για συνεργασία των δασκάλων διαφορετικών ειδικοτήτων, απελευθερώνοντάς τους από την πίεση της ανάπτυξης μεμονωμένων μαθημάτων, δημιουργώντας έτσι μια ατμόσφαιρα δημιουργικότητας και καινοτομίας και στην κοινότητα των εκπαιδευτικών (Belbase et al., 2019).

## **2.6 Θεωρίες μάθησης και STEAM.**

Με τον όρο «θεωρία μάθησης» εννοούμε μια ολοκληρωμένη, συστηματική και εμπειρική άποψη που προσπαθεί να περιγράψει τον τρόπο με τον οποίο συντελείται η διαδικασία της μάθησης, το πώς δηλαδή απορροφούνται οι πληροφορίες, πώς επεξεργάζονται και πώς διατηρούνται κατά τη διάρκεια της μάθησης. Οι θεωρίες μάθησης μέσα από εμπειρικά και πειραματικά παραδείγματα που αντλήθηκαν από διαδικασίες διδασκαλίας και μάθησης επιχειρούν να περιγράψουν τους τρόπους με τους οποίους οι άνθρωποι κατασκευάζουν τη γνώση, βασιζόμενοι στην εμπειρία, στην προηγούμενη γνώση και τον τρόπο με τον οποίο οργανώνουν την προσωπική τους μάθηση (Sanders, 2008). Έτσι, για την οργάνωση του μαθήματος και για τις ενέργειες που θα εκτελεστούν μέσα στην τάξη, ένας δάσκαλος στηρίζεται στις θεωρίες μάθησης.

Η εκπαίδευση STEAM βασίζεται στην θεωρία μάθησης που ονομάζεται «Εποικοδομητισμός», η οποία συχνά συναντάται και ως «Κονστρουκτιβισμός». Ο «Εποικοδομητισμός» βασίζεται στην κονστρουκτιβιστική θεωρία του Piaget, όπου η απόκτηση της γνώσης βασίζεται σε προϋπάρχουσες γνώσεις, εμπειρίες και πεποιθήσεις, πάνω στις οποίες οι μαθητές στηρίζονται για να οικοδομήσουν νέα γνώση. Για τον Piaget, η γνώση δεν είναι απλά πληροφορίες που πρέπει να μεταδοθούν στα παιδιά, να απομνημονευθούν και να εφαρμοστούν όταν πρέπει. Αντίθετα, η γνώση είναι η εμπειρία που κατακτάται μέσω της αλληλεπίδρασης με τον κόσμο, τους ανθρώπους και τα πράγματα που τα περιβάλλουν (Ackermann, 2001). Η θεωρία του εποικοδομητισμού δίνει έμφαση στον ενεργό ρόλο του μαθητή και στην απόκτηση της γνώσης που βασίζεται στην επίλυση πραγματικών προβλημάτων.

## **2.7 Ο ρόλος του εκπαιδευτικού στην STEAM Εκπαίδευση.**

Ο ρόλος του εκπαιδευτικού σε μια STEAM τάξη διαφέρει από την παραδοσιακή τάξη, στην οποία ο δάσκαλος ήταν το επίκεντρο της μαθησιακής διαδικασίας. Σε μια κονστρουκτιβιστική τάξη το ενδιαφέρον μετατοπίζεται στον μαθητή, ο οποίος ενθαρρύνεται να κατασκευάσει τη γνώση μέσω της έρευνας, της εξερεύνησης και της εύρεσης του δικού του μονοπατιού για τη λύση ενός πραγματικού προβλήματος (Simon, 1995). Πιο συγκεκριμένα, ο ρόλος του δασκάλου δεν είναι πλέον να μεταδώσει τη γνώση, αλλά να ενθαρρύνει, να διευκολύνει και να βοηθά τους μαθητές καθώς εκείνοι οικοδομούν την γνώση. Η παιδαγωγική θεώρηση του εποικοδομητισμού απαιτεί από τον εκπαιδευτικό να λαμβάνει υπόψη τις προϋπάρχουσες γνώσεις των μαθητών, τις επιθυμητές γνώσεις που θέλουν να κατακτήσουν οι μαθητές, καθώς και το πώς να οδηγήσουν τους μαθητές στην γνώση (Drape, 2002). Μέσα σε μια κονστρουκτιβιστική τάξη, η γνώση των εκπαιδευτικών εξελίσσεται ταυτόχρονα με την γνώση των μαθητών. Δηλαδή, όπως οι μαθητές μαθαίνουν μέσα σε μια STEAM τάξη, έτσι και ο δάσκαλος με αυτόν τον τρόπο αναπτύσσει τις γνώσεις του επί του θέματος, αλλά ταυτόχρονα μαθαίνει και για τον τρόπο σκέψης των μαθητών (Patel, 2019).

## 2.8 Οι μέθοδοι διδασκαλίας στην STEAM Εκπαίδευση.

Ο όρος «μέθοδος διδασκαλίας» χρησιμοποιείται για να δηλώσει τις συλλογιστικές διαδικασίες που θα βοηθήσουν τον μαθητή στην κατανόηση του περιεχόμενου της διδασκόμενης έννοιας, καθώς και στις διδακτικές και μαθησιακές δραστηριότητες που θα οδηγήσουν τους μαθητές στην γνώση.

Η πιο βασική μέθοδος διδασκαλίας στην STEAM εκπαίδευση είναι η μάθηση βάσει προβλημάτων (Problem-Based Learning - PBL). Είναι μια διδακτική προσέγγιση όπου οι μαθητές κατακτούν την γνώση λύνοντας συνεργατικά προβλήματα του πραγματικού κόσμου. Πρόκειται για μία μέθοδο που χρησιμοποιούταν κατά κύριο λόγο στην ιατρική εκπαίδευση από το 1970, και έχει αποδειχτεί πως είναι μια εξαιρετικά αποτελεσματική εναλλακτική λύση στην εκπαίδευση έναντι των παραδοσιακών μεθόδων, αλλά και πως είναι ιδιαίτερα χρήσιμη στην εκπαίδευση STEAM (Dischino, Michele et al. 2011).

Βασίζεται στο κονστρουκτιβιστικό μοντέλο μάθησης και αποτελείται από τέσσερα βασικά συστατικά (Dischino, Michele et al. 2011):

1. Βασίζεται σε δομημένα προβλήματα που μπορούν να δημιουργήσουν στους μαθητές πολλαπλές υποθέσεις για τις αιτίες που τα προκαλούν, και στη συνέχεια να αναπτύξουν πολλαπλές προσεγγίσεις για την λύση τους.
2. Τα προβλήματα αυτά θα πρέπει να είναι αυθεντικά και να προέρχονται από τον πραγματικό κόσμο.
3. Η προσέγγιση της γνώσης είναι μαθητοκεντρική: Οι μαθητές είναι αυτοί που καθορίζουν τις πληροφορίες που θα διδαχθούν, ενώ ταυτόχρονα θα πρέπει να μάθουν να βρίσκουν τους κατάλληλους πόρους από τους οποίους θα τις αντλούν. Ο καθορισμός των πληροφοριών που θα διδαχθούν οι μαθητές καθορίζεται από το ίδιο το πρόβλημα.
4. Ο δάσκαλος ενεργεί ως διαμεσολαβητής της γνώσης και όχι ως αυθεντία υπεύθυνη για τη μετάδοση της.

Ο συνδυασμός αυτών των τεσσάρων αρχών βοηθάει τους μαθητές στην απόκτηση γνώσεων που έχουν εφαρμογή στον πραγματικό κόσμο, και ενισχύει την προετοιμασία τους για την μεταδευτεροβάθμια εκπαίδευση, καθώς δημιουργεί

συνδέσεις με το τι κάνουν πραγματικά οι επαγγελματίες των STEM πεδίων στη δουλειά τους (Carrao, 2013). Τέλος, η μάθηση βάσει προβλημάτων δημιουργεί ένα δυναμικό περιβάλλον μάθησης, το οποίο ενσωματώνει διάφορα ερεθίσματα και οδηγεί στην απόκτηση πολύτιμων εμπειριών και γνώσεων, και γι' αυτό θα πρέπει να θεωρείται η κύρια μέθοδος διδασκαλίας στην εκπαίδευση STEAM (Euefueno, 2019).



## Κεφάλαιο 3 – Ρομποτικά Συστήματα

### Εισαγωγή.

Η τέταρτη βιομηχανική επανάσταση οδήγησε στην εμφάνιση νέων εργαλείων και τεχνολογιών, επηρεάζοντας κάθε πτυχή της ζωής μας, ακόμα και την εκπαίδευση. Στα πλαίσια λοιπόν αυτής της ραγδαίας τεχνολογικής ανάπτυξης αναπτύχθηκε και η επιστήμη της ρομποτικής, η οποία έχει ήδη εφαρμοστεί σε διάφορους τομείς, συμπεριλαμβανομένης της ιατρικής, της αυτοκινητοβιομηχανίας, της αναζήτησης και της διάσωσης, ακόμα και της εκπαίδευσης. Συνδυάζοντας όλα τα πεδία της STEAM εκπαίδευσης σε ένα μόνο αντικείμενο, η ρομποτική παρέχει την ευκαιρία στους μαθητές να εξερευνήσουν πώς τα πεδία αυτά συνδυάζονται και στην πραγματική ζωή, πώς λειτουργεί η τεχνολογία γύρω τους, ενώ ταυτόχρονα τα ρομποτικά συστήματα αποτελούν ένα ελκυστικό εργαλείο, καθώς είναι προφανές ότι οι μαθητές θεωρούν ενδιαφέροντα τα ρομπότ τόσο σε πραγματικές καταστάσεις όσο και σε εικονικές (Badmus, 2020).

### 3.1 Ορισμός της Ρομποτικής.

Αναζητώντας κάποιος τον ορισμό της λέξης «ρομπότ» θα συναντήσει μια πληθώρα από ορισμούς, πολλές φορές και αντικρουόμενους ως προς κάποια χαρακτηριστικά. Για παράδειγμα, το Λεξικό της Κοινής Νεοελληνικής ορίζει το ρομπότ ως «ανθρωπόμορφη μηχανική κατασκευή που εκτελεί κινήσεις ανάλογες με του ανθρώπου και μπορεί να τον αντικαταστήσει σε ορισμένες εργασίες», ενώ η Encyclopaedia Britannica ορίζει το ρομπότ ως «κάθε συσκευή που λειτουργεί αυτόματα και αντικαθιστά τον άνθρωπο. Μπορεί να μην μοιάζει σωματικά με άνθρωπο ή να εκτελεί λειτουργίες με τον ανθρώπινο τρόπο και για αυτό το λόγο η διάκριση ανάμεσα στα ρομποτικά συστήματα και στα απλά αυτοματοποιημένα μηχανήματα είναι δύσκολο να εντοπιστεί». Ένας ακόμα ορισμός σύμφωνα με το Ινστιτούτο Ρομποτικής της Αμερικής (Robot Institute of America) ο οποίος δόθηκε το 1979 περιγράφει ένα ρομπότ ως «ένας επαναπρογραμματιζόμενος, πολυλειτουργικός χειριστής σχεδιασμένος για να μετακινεί υλικά αντικείμενα, μέρη αντικειμένων, εργαλεία ή εξειδικευμένες συσκευές μέσω διάφορων προγραμματισμένων κινήσεων

για την εκπόνηση διάφορων καθηκόντων». Παρότι λοιπόν υπάρχει μια πληθώρα ορισμών για τη λέξη ρομπότ, οι περισσότεροι ορισμοί συμφωνούν στο ότι ως ρομπότ ορίζεται οποιαδήποτε μηχανή μπορεί να υποκαθιστά τον άνθρωπο σε διάφορες εργασίες.

Ετυμολογικά, ο όρος «ρομπότ» έχει σλαβική προέλευση και προέρχεται από την τσέχικη λέξη *robota* (ρομπότα) που σημαίνει καταναγκαστική εργασία, δουλειά, αγγαρεία. Η λέξη «ρομπότ» απέκτησε την σημερινή της σημασία από τον Τσέχο θεατρικό συγγραφέα Karel Capek στις αρχές του 20<sup>ου</sup> αιώνα. Σε ένα θεατρικό έργο με τίτλο *R.U.R. (Rosum's Universal Robots)*, ο Capek δημιούργησε αυτοματοποιημένα υποκατάστατα των ανθρώπινων εργαζομένων, τα οποία έχουν ανθρώπινη συμπεριφορά και μπορούν να νιώσουν ανθρώπινα συναισθήματα.

Έτσι λοιπόν, η ρομποτική είναι ένας σύγχρονος τεχνολογικός και διεπιστημονικός κλάδος που έχει ως αντικείμενο την μελέτη, τον σχεδιασμό, την ανάπτυξη και κατασκευή ρομποτικών συστημάτων. Περιλαμβάνει επίσης και το κομμάτι της έρευνα προκειμένου να επιτευχθεί η περαιτέρω ανάπτυξη τους. Η διεπιστημονικότητα του κλάδου οφείλεται στο γεγονός ότι για να αναπτυχθεί ένα ρομποτικό σύστημα και να επιτευχθεί η αυτοματοποίηση στη λειτουργία του θα πρέπει να συνδυάσει κανείς αποτελεσματικά τους κλάδους της Μηχανικής και της Μηχανολογίας, της Ηλεκτρονικής, καθώς και της Επιστήμης των Υπολογιστών, της Πληροφορικής, αφού η ανάπτυξη ενός ρομποτικού συστήματος απαιτεί το συνδυασμό ηλεκτρικών και μηχανικών στοιχείων για την κατασκευή του «σώματος» με «εγκεφάλους» της επιστήμης των υπολογιστών (Birk, 2011).

### **3.2 Ιστορική αναδρομή.**

Ανατρέχοντας κανείς στην ιστορία των ρομπότ και της ρομποτικής παρατηρεί πως, παρότι η ρομποτική επιστήμη είναι ένα σύγχρονο επιστημονικό πεδίο, το ενδιαφέρον για τη δημιουργία μηχανών που θα εκτελούσε αυτόματα λειτουργίες προς όφελος του ανθρώπου είναι κάτι που μπορεί να εντοπιστεί ήδη από την αρχαιότητα. Ήδη στην μυθολογία γίνεται αναφορά σε ένα μυθικό ρομπότ φτιαγμένο από χαλκό, τον Τάλω, ο οποίος είχε κατασκευαστεί από τον Ήφαιστο με σκοπό να προφυλάσσει το

νησί της Κρήτης. Παρακάτω παρουσιάζεται ένα συνοπτικό χρονολογικό διάγραμμα με μερικές βασικές χρονολογίες - «σταθμούς» στην εξέλιξη της ρομποτικής:

- 1023–957 π.Χ.: Στην Κίνα ένας τεχνίτης-μηχανικός, ο Yan Shi, παρουσίασε στον βασιλιά Mu of Zhou μία φυσικού μεγέθους, σε σχήμα ανθρώπου, μηχανική φιγούρα.
- 4<sup>ος</sup> αιώνας π.Χ. (428 – 347 π.Χ. ): Ο Έλληνας φιλόσοφος και μαθηματικός Αρχύτας ο Ταραντίνος σχεδίασε ένα μηχανικό πουλί («το περιστέρι»), το οποίο με τη βοήθεια ατμού ή συμπιεσμένου ατμού κουνούσε τα φτερά του.
- 10-70 μ.Χ.: Ο μαθηματικός, φυσικός και μηχανικός Ήρων από την Αλεξάνδρεια έγραψε ένα βιβλίο με τίτλο "Automata" που συμπεριλάμβανε μια συλλογή από αυτόματες κατασκευές που θα μπορούσαν να χρησιμοποιηθούν σε ναούς, συμπεριλαμβανομένης μιας η οποία φέρεται να μπορούσε να μιλήσει (Yates, 2011).
- 1136–1206 μ.Χ.: Ο Άραβας εφευρέτης Al Jazari σχεδίασε και κατασκεύασε διάφορους αυτόματους μηχανισμούς, καθώς και το πρώτο προγραμματιζόμενο ανθρωποειδές ρομπότ, μια ρομποτική βάρκα, με αυτοματοποιημένους μουσικούς. Σκοπός του ρομποτικού αυτού μηχανισμού ήταν να διασκεδάζει τους επισκέπτες σε βασιλικά πάρτι.
- 1495 μ.Χ.: Ο Leonardo da Vinci σχεδιάζει ένα ανθρωποειδές ρομπότ, ικανό να κινεί το κεφάλι του, να κάθεται και να σηκώνεται.
- 1700–1900 μ.Χ.: Διάφορες αυτόματες μηχανικές μαριονέτες από ζώα ή ανθρώπους γίνονται ιδιαίτερα δημοφιλείς. Η πιο διάσημη από αυτές ήταν μια μηχανική πάπια σχεδιασμένη από τον Jacques de Vaucanson (1709), η οποία μπορούσε να τεντώσει το λαιμό της, να κουνήσει τα φτερά της, να καταπιεί και να χωνέψει κόκκους.
- 1725 μ.Χ.: Ο Lorenz Rosenegege εγκαθιστά στο Παλάτι Χέλμπρουν ένα μηχανικό θέατρο που αποτελείται από 256 φιγούρες. Οι 119 από αυτές κινούνταν με τη βοήθεια ενός υδροστόβιλου.
- 1772 μ.Χ.: Ο Pierre Jaquet-Droz κατασκευάζει τον «Γραφέα» (Writer), μια παιδική φιγούρα που μπορούσε να γράψει, χωρίς όμως να χρησιμοποιεί κενά ή σημεία στίξης, η οποία ελεγχόταν από μια προγραμματιζόμενη μηχανή αυξημένης πολυπλοκότητας.

- 1833 μ.Χ.: Ο Βρετανός μαθηματικός Charles Babbage ξεκίνησε να δουλεύει πάνω στην Αναλυτική Μηχανή, μια από τις πρώτες προγραμματιζόμενες μηχανές.
- 1921 μ.Χ.: Χρησιμοποιείται για πρώτη φορά ο όρος «ρομπότ» από τον θεατρικό συγγραφέα Karel Capek. Στο έργο «Rossum's Universal Robots» ένας εφευρέτης δημιουργεί ανθρωποειδείς μηχανές που ονομάζονται ρομπότ.
- 1932 μ.Χ.: Δημιουργείται στην Ιαπωνία το πρώτο ρομποτικό παιχνίδι που ονομάζεται «Lilliput». Ήταν ένα παιχνίδι από κασσίτερο που μπορούσε να περπατήσει.
- 1942 μ.Χ.: Ο Αμερικάνος συγγραφέας Isaac Asimov διατυπώνει τους τρεις νόμους της ρομποτικής στο έργο του “Runaround”:  
**Πρώτος Νόμος:** Ένα ρομπότ δεν μπορεί να τραυματίσει έναν άνθρωπο ή, μέσω αδράνειας, να επιτρέψει σε έναν άνθρωπο να βλάψει.  
**Δεύτερος Νόμος:** Ένα ρομπότ πρέπει να υπακούει στις εντολές που του δίνονται από τους ανθρώπους, εκτός από τις περιπτώσεις όπου οι εντολές θα έρχονται σε αντίθεση με τον Πρώτο Νόμο.  
**Τρίτος Νόμος:** Ένα ρομπότ πρέπει να προστατεύει την ύπαρξή του, εφόσον αυτή η προστασία δεν έρχεται σε αντίθεση με τον Πρώτο ή τον Δεύτερο Νόμο.
- 1961 μ.Χ.: Το πρώτο βιομηχανικό ρομπότ με όνομα «Unimate» χρησιμοποιείται στη γραμμή συναρμολόγησης της General Motors για την κατασκευή κινητήρων.
- 1975 μ.Χ.: Ο Αμερικάνος μηχανικός Victor Scheinman αναπτύσσει έναν προγραμματιζόμενο ρομποτικό βραχίονα, ο οποίος στη συνέχεια χρησιμοποιείται ευρέως στη βιομηχανία.
- 1981 μ.Χ.: Το “Canadarm”, ένα προγραμματιζόμενο ρομποτικό «χέρι» ελεγχόμενο εξ αποστάσεως πραγματοποιεί την πρώτη του διαστημική αποστολή.
- 1997 μ.Χ.: Το αμερικάνικο ρομποτικό διαστημικό σκάφος «Mars Pathfinder» πραγματοποιεί μια σειρά από πειράματα στον Άρη.
- 1998 μ.Χ.: Η LEGO παρουσιάζει την πρώτη έκδοση του Lego Mindstorms που πωλείται ως Robotics Invention System (RIS).

- 2009 μ.Χ.: Η εταιρεία «Intuitive Surgical» κυκλοφορεί το σύστημα da Vinci Si (da Vinci Si System), ένα ρομποτικό σύστημα που επιτρέπει τη συνεργασία ανάμεσα σε δύο χειρουργούς.
- 2016 μ.Χ.: Ενεργοποιείται για πρώτη φορά το ανθρωποειδές ρομπότ «Sophia». Το 2017 έλαβε την ιθαγένεια της Σαουδικής Αραβίας και έτσι αποτέλεσε το πρώτο ρομπότ που απέκτησε εθνικότητα.

### 3.3 Χρήσεις Ρομποτικής.

Ένας από τους βασικούς χώρους χρήσης των ρομποτικών συστημάτων είναι η βιομηχανία. Η παγκοσμιοποίηση και η ολοένα αυξανόμενη ζήτηση για καινοτόμα προϊόντα έχει οδηγήσει σε νέες βιομηχανικές προκλήσεις και μαζική παραγωγή προϊόντων. Έτσι, οι βιομηχανίες στράφηκαν στην αντικατάσταση της ανθρώπινης και σωματικής εργασίας από ρομπότ, τα οποία μπορούν να χρησιμοποιηθούν ως βοηθοί για την υποστήριξη των εργαζομένων. Τα ρομπότ είναι κατασκευασμένα με τέτοιο τρόπο ώστε να αντικαταστήσουν ή να υποστηρίξουν τους εργαζομένους σε εργασίες όπως για παράδειγμα ο χειρισμός βαρέων φορτίων, εργασίες δηλαδή όπου απαιτείται μεγάλη σωματική προσπάθεια. Η χρήση τους είναι ιδιαίτερα χρήσιμη προκειμένου να προβλεφθούν και να ελαχιστοποιηθούν μυοσκελετικές διαταραχές που προέρχονται από επαναλαμβανόμενες και δυναμικές εργασίες (Bragança et al., 2019).

Σημαντικά πλεονεκτήματα έχει προσφέρει και η χρήση των ρομποτικών συστημάτων στον στρατό. Στην αρχή η χρήση τους περιοριζόταν σε αποστολές αποναρκοθέτησης και διάσωσης, η έρευνα όμως τώρα επικεντρώνεται και στην αποστολή των ρομπότ σε πεδία μάχης ή αποστολές κατασκοπείας. Σκοπός της χρήσης των ρομποτικών συστημάτων στις στρατιωτικές αποστολές είναι να βελτιώσουν τη λήψη αποφάσεων υπό τακτική πίεση και κάτω από δύσκολες συνθήκες για τον άνθρωπο, λαμβάνοντας αποφάσεις με τη βοήθεια της τεχνητής νοημοσύνης, συλλέγοντας, αναλύοντας, και επεξεργάζοντας γρήγορα δεδομένα, να λαμβάνουν τις κατάλληλες αποφάσεις και να τις διαδίδουν. Επιπρόσθετα, σε περίπτωση σύλληψης του ρομποτικού συστήματος, αυτό θα έχει τη δυνατότητα να αυτοκαταστραφεί, διασφαλίζοντας έτσι την μυστικότητα της αποστολής (Khurshid, 2004). Όσον αφορά

την κατασκοπεία, τα ρομπότ θα μπορούσαν να κρύβονται για μεγάλα χρονικά διαστήματα χωρίς φαγητό και με μικρότερη ανάγκη για ενέργεια.

Ένα παράδειγμα ρομποτικού συστήματος που δημιουργήθηκε για στρατιωτικούς σκοπούς είναι το τετράποδο ρομπότ “BigDog”, το οποίο αναπτύχθηκε για να μεταφέρει υπολογιστικές μονάδες εκεί όπου τα συμβατικά οχήματα μεταφοράς δεν μπορούν να λειτουργήσουν, καθώς είναι σχεδιασμένο να περνά μέσα από χιόνι, νερό και βραχώδες έδαφος (Szegedi et al., 2017),

Επιπρόσθετα, ιδιαίτερα σπουδαία είναι η συνεισφορά των ρομποτικών συστημάτων στην εξερεύνηση του διαστήματος. Αρχικά, οι ρομποτικές αποστολές εξερεύνησης του διαστήματος ως επί το πλείστον αφορούσαν μεγάλες ιπτάμενες κάμερες με σκοπό να στέλνουν εικόνες από πλανήτες, φεγγάρια, δορυφόρους και άλλα εξωγήινα σώματα (Katz, 2003). Πλέον, οι ρομποτικές αποστολές επικεντρώνονται κυρίως σε διασώσεις και απομάκρυνση διαστημικών απορριμμάτων, διόρθωση, ανεφοδιασμό και υποστήριξη αποστολών. Επιπλέον, η ρομποτική χρησιμοποιείται για να ξεπεράσει τους ανθρώπινους περιορισμούς και να παρέχει πρόσβαση στο σκληρό και ενδεχομένως επικίνδυνο περιβάλλον του διαστήματος, επεκτείνοντας τις δυνατότητες των αστροναυτών για να εξερεύνηση (Gao, 2017).

Δύο τομείς της ανθρώπινης ζωής ιδιαίτερα σημαντικοί για την ανθρώπινη επιβίωση και ανάπτυξη, στους οποίους η ενσωμάτωση της ρομποτικής έχει προσφέρει σημαντικά οφέλη, είναι ο χώρος της υγείας και της ιατρικής, καθώς και ο χώρος της αγροτικής παραγωγής.

Αναφορικά με τους τομείς υγείας, η ρομποτική εισήχθη για πρώτη φορά στους τομείς υγείας με τη δημιουργία των χειρουργικών ρομπότ που εκτελούν βοηθητικό ρόλο στις χειρουργικές επεμβάσεις. Μέχρι σήμερα τα χειρουργικά ρομπότ - βοηθοί είναι τα πιο δημοφιλή στους κλάδους υγείας και τα πιο εγκεκριμένα. Ένα καινούργιο αναδυόμενο πεδίο στην υγειονομική περίθαλψη με χρήση ρομπότ, είναι τα ρομπότ – συνοδοί, τα οποία χρησιμοποιούνται στη θέση των ζώων σαν συνοδοί θεραπείας. Τα αποτελέσματα μιας τέτοιου είδους θεραπείας έχουν αποδειχθεί πολύ θετικά και ελπιδοφόρα, ενθαρρύνοντας την περαιτέρω έρευνα σε αυτόν τον τομέα. Τέλος, η υποβοηθούμενη από ρομπότ τηλεχειρουργική, είναι ένα είδος χειρουργικής επέμβασης όπου ο χειρουργός γιατρός μπορεί να βρίσκεται σε διαφορετική τοποθεσία από αυτή στην οποία διενεργείται η επέμβαση. Αυτού του είδους χειρουργικές επεμβάσεις

μπορούν επίσης να ωφελήσουν επίσης τον στρατό σε απομακρυσμένες περιοχές, όπου η υγειονομική περίθαλψη είναι περιορισμένη (Khan and Yumna Anwar, 2020).

Σχετικά με την αγροτική παραγωγή, τα ρομποτικά συστήματα έχουν αλλάξει το πρόσωπο της γεωργίας τα τελευταία χρόνια, με την δημιουργία τηλεχειριζόμενων και αυτόνομων τρακτέρ και θεριζοαλωνιστικών μηχανών με δυνατότητα GPS, με ρομποτικούς σταθμούς αρμέγματος, με ρομποτικούς συλλέκτες φρούτων και λαχανικών και με τη χρήση αυτόνομων αεροσχημάτων, γνωστών και ως Drones, για εξ αποστάσεως έλεγχο αγροτικών υποδομών, και ιδιαίτερα περιφραγμένων περιοχών και συστημάτων άρδευσης (Sparrow and Howard, 2020).

Δύο ακόμη τομείς πιο κοντινούς στην ανθρώπινη καθημερινότητα στους οποίους συναντά κανείς χρήση ρομποτικών συστημάτων είναι η εξυπηρέτηση και η ψυχαγωγία πελατών, αλλά και η συμπερίληψη των ρομπότ στα ξενοδοχειακά συστήματα φαίνεται να αποτελεί το μέλλον της φροντίδας των πελατών. Ένα παράδειγμα ρομποτικού συστήματος που χρησιμοποιείται σε ξενοδοχεία είναι το Saviok's Relay, το οποίο εκτελεί κυρίως εργασίες παράδοσης και είναι εξοπλισμένο με κάμερες και αισθητήρες ώστε να μπορεί να αναγνωρίζει τους αριθμούς των δωματίων, να κινείται σε πολυσύχναστους διαδρόμους αποφεύγοντας τις συγκρίσεις, ενώ μπορεί να χρησιμοποιεί μόνο του και το ασανσέρ. Με την άφιξη στον προορισμό του, το Relay ανοίγει το καπάκι του ώστε να προμηθευτούν οι επισκέπτες τα είδη και τρόφιμα που έχουν παραγγείλει, καθώς και κάθε είδους ανέσεις. (Choi et al., 2019). Τα τελευταία χρόνια, στο κομμάτι της ψυχαγωγίας είναι ιδιαίτερα προσφιλή τα ρομποτικά πρωταθλήματα ποδοσφαίρου (Keshav et al., 2020), ενώ μπορεί κανείς να παρακολουθήσει ρομποτικό θέατρο, που συχνά αναφέρεται ως «robot acting», σε ζωντανές σκηνές και περιβάλλοντα ψυχαγωγίας (Lytridis et al., 2019).

Τέλος, ακόμα και στον τομέα της εκπαίδευσης ποικίλουν τα οφέλη της ρομποτικής. Μέσα από προγράμματα που βασίζονταν σε projects στα οποία γινόταν χρήση ρομποτικών εκπαιδευτικών συστημάτων, οι μαθητές κατάφεραν να αναπτύξουν πολύ εφευρετικές λύσεις στα προβλήματα (Barak, 2009), ενώ αποδείχθηκε πως βοήθησαν πάρα πολύ και στην ανάπτυξη της ομαδικότητας μεταξύ των μαθητών (Causo and Chen, 2016). Παράλληλα, η χρήση των κοινωνικών ρομπότ (Social Robots) έχει αποδειχθεί πως είναι ένα εξαιρετικό εργαλείο στην εκμάθηση μιας ξένης γλώσσας. Για παράδειγμα, έρευνα που έγινε σε παιδιά μεταναστών έδειξε πως η σύνδεση

αντικειμένων και γεγονότων με χειρονομίες στη γλώσσα-στόχο με την βοήθεια κοινωνικών ρομπότ ήταν ιδιαίτερα σημαντική στην εκμάθηση της δεύτερης γλώσσας (Konijn et al., 2020).

### **3.4 Ανατομία και έλεγχος ρομποτικών συστημάτων.**

Κάθε ρομποτικός μηχανισμός είναι ένα ολοκληρωμένο σύστημα, το οποίο όμως αποτελείται από διάφορα υποσυστήματα τα οποία συνεργάζονται ώστε να έχουμε το τελικό αποτέλεσμα. Τα βασικά υποσυστήματα από τα οποία αποτελείται ένας ρομποτικός μηχανισμός είναι: το υποσύστημα της τροφοδοσίας, το υποσύστημα των αισθητήρων, του ελέγχου/επεξεργασίας και των ενεργοποιητών.

Σχετικά με το σύστημα της τροφοδοσίας, αυτό είναι υπεύθυνο για την τροφοδότηση του ρομποτικού συστήματος, παρέχοντας σωστή τάση, ρεύμα και συχνότητα ηλεκτρικού ρεύματος, το οποίο προέρχεται από μια πηγή.

Το υποσύστημα των αισθητήρων περιλαμβάνει το κομμάτι με το οποίο «αισθάνεται» το ρομπότ και «αντιλαμβάνεται» το περιβάλλον γύρω του. Ως άνθρωποι έχουμε την ικανότητα να επιβιώνουμε και να επιδρούμε στο περιβάλλον χρησιμοποιώντας τις πέντε αισθήσεις μας: όραση, ακοή, όσφρηση, αφή και γεύση. Με τον ίδιο τρόπο τα ρομπότ χρησιμοποιώντας αισθητηριακούς μετατροπείς είναι ικανά να λειτουργούν μέσα σε ένα περιβάλλον. Οι αισθητήρες μετατρέπουν ένα φυσικό μέγεθος σε ηλεκτρικό σήμα, το οποίο μπορεί να αναγνωριστεί και να χρησιμοποιηθεί κατάλληλα από τον ρομποτικό μηχανισμό. Δρώντας συχνά σε μη δομημένα και μεταβαλλόμενα περιβάλλοντα, ένα σημαντικό πλεονέκτημα των αισθητήρων είναι ότι μεταφέρουν πληροφορίες στο ρομποτικό σύστημα σε πραγματικό χρόνο, με αποτέλεσμα να μπορούν να αντιληφθούν και να μεταδώσουν αποτελεσματικά τις μεταβολές του περιβάλλοντος χώρου. Οι πιο συνήθεις κατηγορίες αισθητήρων είναι οι απτικοί αισθητήρες, οι αισθητήρες απόστασης, φωτός και ήχου. Το υποσύστημα αυτό είναι υπεύθυνο για την λήψη πληροφοριών και την μετατροπή τους σε ηλεκτρικό σήμα, αλλά και για την παραγωγή σημάτων εξόδου για την μετάδοση των πληροφοριών.

Αναφορικά με το υποσύστημα ελέγχου/επεξεργασίας, αυτό είναι υπεύθυνο για τον προσδιορισμό και την λήψη πληροφοριών από τους αισθητήρες, την επεξεργασία τους, αλλά και τον υπολογισμό των εντολών ελέγχου που πρέπει να σταλούν στα



στοιχεία του ρομποτικού συστήματος με σκοπό να πραγματοποιηθεί οποιαδήποτε εργασία. Οι ενέργειες που θα εκτελέσει το ρομποτικό σύστημα καθορίζονται μέσω του προγραμματισμού. Ο προγραμματισμός δεν είναι τίποτα άλλο από τη σύνταξη οδηγιών που πρέπει να ακολουθήσει ένας υπολογιστής για την επίλυση ενός «προβλήματος». Η δουλειά ενός προγραμματιστή είναι να βρει μια λύση σε ένα πρόβλημα, να μετατρέψει τη λύση σε αλγόριθμο και στη συνέχεια να κωδικοποιήσει τον αλγόριθμο σε πρόγραμμα ώστε να μπορεί να διαβαστεί από έναν υπολογιστή. Δηλαδή, ένα πρόγραμμα είναι ένας αλγόριθμος που έχει μεταφραστεί-κωδικοποιηθεί σε οδηγίες για έναν υπολογιστή. Όπως αναφέρει και ο Blackwell (2002:1) παραθέτοντας τον McCracken (1957): «Ο προγραμματισμός... είναι βασικά μια διαδικασία μετάφρασης από τη γλώσσα που είναι κατάλληλη για τον άνθρωπο στη γλώσσα που είναι κατάλληλη για τον υπολογιστή».

Τέλος, οι ενεργοποιητές, ή αλλιώς συσκευές εξόδου, είναι ηλεκτρομηχανικές συσκευές που μετατρέπουν την ενέργεια που προσλαμβάνουν σε μηχανικό έργο. Σκοπός τους είναι να παράγουν πληροφορίες ή ενέργειες από το ρομπότ προς το εξωτερικό περιβάλλον. Συνήθως οι ενέργειές τους συνδέονται με τις πληροφορίες που δέχονται από τα αισθητήρια όργανα.

## **Κεφάλαιο 4 – Εκπαιδευτική Ρομποτική**

### **Εισαγωγή.**

Στα πλαίσια της διερεύνησης καινοτόμων εργαλείων που θα βοηθήσουν την εκπαιδευτική διαδικασία, αλλά και θα προετοιμάσουν τους μαθητές κατάλληλα για την σημερινή κοινωνία, εμφανίστηκε ένα σύνολο «νέων τεχνολογιών» ικανών να υποστηρίξουν ικανοποιητικά την ανάπτυξη των μαθητών και να κεντρίσουν το ενδιαφέρον τους για τη μάθηση. Στα πλαίσια της ενσωμάτωσης νέων τεχνολογιών στην εκπαίδευση εμφανίστηκε και η εκπαιδευτική ρομποτική. Έχοντας διεισδύσει στον χώρο της εκπαίδευσης τα τελευταία χρόνια, η εκπαιδευτική ρομποτική έχει αποδειχθεί ένα πολύ ισχυρό εργαλείο για τους μαθητές, καθώς τους παρέχει ένα ενεργό περιβάλλον μάθησης μέσα από το οποίο οι μαθητές θα κατακτήσουν δεξιότητες αναγκαίες για τον 21<sup>ο</sup> αιώνα.

### **4.1 Ορισμός Εκπαιδευτικής Ρομποτικής.**

Παρότι τα τελευταία χρόνια έχει αυξηθεί το ενδιαφέρον και η ενασχόληση της εκπαιδευτικής κοινότητας με το πεδίο της εκπαιδευτικής ρομποτικής, στις περισσότερες περιπτώσεις παρατηρείται πως σαφής ορισμός δεν είναι δυνατός να καταστεί. Στα πλαίσια της αναζήτησής μας, παρατηρήσαμε ότι στην υπάρχουσα βιβλιογραφία υπάρχουν δύο οπτικές γωνίες τοποθέτησης της ρομποτικής σε ένα εκπαιδευτικό πλαίσιο.

Ο πρώτος ορισμός, που είναι και ο πιο συνηθισμένος, είναι αυτός που βλέπει την εκπαιδευτική ρομποτική σαν το πεδίο το οποίο ασχολείται με τη χρήση των ρομπότ ως εργαλεία στην εκπαίδευση. Στο πλαίσιο αυτού του ορισμού, η ρομποτική θεωρείται ως ένα αποτελεσματικό εργαλείο για να παρακινήσει τους μαθητές και να τους παρέχει από υλικό για μάθηση. Σε αυτήν την εργαλειακή θεώρηση της ρομποτικής, η δημιουργία προγραμμάτων σπουδών ρομποτικής έχει υποστηρικτικό ρόλο και δεν διδάσκονται αυτόνομα, αλλά ως συμπληρωματικό υλικό που μπορεί να υποστηρίξει τα ήδη υπάρχοντα προγράμματα σπουδών πάνω σε ένα αντικείμενο (Jung & Won, 2018). Παρόλα αυτά, αυτός ο ορισμός είναι ασαφής και ανοιχτός σε παρερμηνείες, καθώς οποιοδήποτε μάθημα στο οποίο θα χρησιμοποιούταν ένας ρομποτικός μηχανισμός ως

εργαλείο θα μπορούσε να θεωρηθεί ως ένα μάθημα εκπαιδευτικής ρομποτικής (Fernandez & Vincze, 2018).

Ο δεύτερος ορισμός ορίζει την εκπαιδευτική ρομποτική ως το πεδίο στο οποίο τα ρομποτικά συστήματα χρησιμοποιούνται ως εργαλεία για να διδάξουν την ίδια την επιστήμη της ρομποτικής. Αυτή η οπτική επικεντρώνεται στην διεπιστημονική φύση της εκπαιδευτικής ρομποτικής σχετίζοντας την με την STEM εκπαίδευση. Επεξεργάζοντας αναλυτικά τους διδακτικούς στόχους, το περιεχόμενο, τις μεθόδους διδασκαλίας της εκπαιδευτικής ρομποτικής (με βάση αυτόν τον ορισμό) και της εκπαίδευσης STEM, παρατηρήθηκε πως παρουσιάζουν πολλά κοινά και υποστηρίχθηκε πως η εκπαιδευτική ρομποτική μπορεί να είναι και το καλύτερο πεδίο που μπορεί να προσφέρει πρόσβαση στην STEM εκπαίδευση (Jung & Won, 2018).

Λαμβάνοντας υπόψιν αυτή την ασάφεια στον ορισμό της εκπαιδευτικής ρομποτικής, οι Fernandez και Vincze προσπάθησαν να δώσουν έναν ορισμό της ρομποτικής που θα περιλαμβάνει όλους τους υπόλοιπους πιθανούς ορισμούς. Έτσι σύμφωνα με τους Fernandez και Vincze (2018:1):

*«Η Εκπαιδευτική Ρομποτική είναι ένα πεδίο σπουδών που στοχεύει στη βελτίωση της μαθησιακής εμπειρίας των ανθρώπων μέσω της δημιουργίας, της υλοποίησης, της βελτίωσης και επικύρωσης παιδαγωγικών δραστηριοτήτων, εργαλείων (π.χ. οδηγίες και πρότυπα) και τεχνολογιών, όπου τα ρομπότ παίζουν ενεργό ρόλο και κάθε απόφαση ενημερώνεται από παιδαγωγικές μεθόδους».*

Αυτός ο ορισμός καλύπτει όλους του πιθανούς ορισμούς που έχουν δοθεί για το πεδίο της εκπαιδευτικής ρομποτικής: και την ρομποτική ως διδασκόμενο πεδίο επιστήμης, αλλά και την ρομποτική ως εργαλείο και βοήθημα στην εκπαιδευτική διαδικασία.

#### **4.2 Θεωρίες Μάθησης και Εκπαιδευτική Ρομποτική.**

Οι δύο θεμελιώδεις θεωρίες πίσω από την εκπαιδευτική ρομποτική είναι ο κονστρουκτιβισμός (Constructivism) και ο εποικοδομισμός (Constructionism). Η βασική αρχή του κονστρουκτιβισμού είναι πως η γνώση είναι μια εμπειρία που κατασκευάζεται μέσω της ενεργής αλληλεπίδρασης των ατόμων με το περιβάλλον. Η

εμπειρία των ατόμων και οι προϋπάρχουσες γνώσεις τους είναι τα θεμέλια πάνω στα οποία θα δομηθεί η νέα γνώση (Piaget, 1970). Βάσει του κονστρουκτιβισμού, οι μαθητές χωρισμένοι σε μικρές ομάδες αποκτούν την γνώση εργαζόμενοι προς την επίλυση αυθεντικών προβλημάτων. Η ενασχόληση αυτή των μαθητών με αυθεντικά προβλήματα του πραγματικού κόσμου χρησιμοποιώντας τεχνολογικά εργαλεία ενθαρρύνει την δημιουργία δημιουργικών λύσεων, ενώ ταυτόχρονα δίνει κίνητρα στους μαθητές να εμπλακούν στη διαδικασία της γνώσης (Papert, 1993). Σε τέτοια περιβάλλοντα μάθησης, η διαδικασία κατασκευής της γνώσης, η διαμορφωτική αξιολόγηση και το τελικό προϊόν είναι εξίσου σημαντικά και δεν δίνεται παραπάνω έμφαση σε κάποιο από αυτά (Anwar et al, 2019).

Η δεύτερη θεωρία, η οποία αναπτύχθηκε από τον Papert βασισμένος στις αρχές του κονστρουκτιβισμού του Piaget, είναι ο εποικοδομισμός (Papert, 1980; 1993). Ο Papert πίστευε πως η δόμηση της γνώσης συμβαίνει πιο αποτελεσματικά μέσα σε ένα πλαίσιο όπου οι μαθητές εμπλέκονται συνειδητά στην κατασκευή μιας οντότητας, είτε αυτή είναι απλά ένα κάστρο στην άμμο, είτε ένα τεχνολογικό εργαλείο (Papert, 1980). Σύμφωνα με αυτή τη θεωρία, η γνώση κατασκευάζεται στο μυαλό του μαθητή μέσω της ενεργητικής μάθησης και δεν μεταφέρεται απλά από τον δάσκαλο στον μαθητή (Sisman & Kucuk, 2019). Η κύρια διαφορά αυτής της θεώρησης με τον κονστρουκτιβισμό είναι πως ο κονστρουκτιβισμός εστιάζει κυρίως στις νοητικές διαδικασίες των μαθητών, ενώ ο εποικοδομισμός βασίζεται στην φυσική υπόσταση των διαδικασιών (π.χ. κατασκευή ενός φυσικού μοντέλου) (Ackermann, 2001).

Έτσι, στην περίπτωση της Εκπαιδευτικής Ρομποτικής οι μαθητές χωρισμένοι σε ομάδες γίνονται ενεργοί κατασκευαστές αντί για παθητικούς χρήστες προϊόντων και εργαλείων, και οικοδομούν τη γνώση σχεδιάζοντας και κατασκευάζοντας τόσο καινοτόμα ρομποτικά συστήματα όσο και προγράμματα για τη λειτουργία των ρομποτικών συστημάτων. Με αυτό τον τρόπο, οι μαθητές θα αποκτήσουν γνώσεις για τα πεδία του STEM, ενώ ταυτόχρονα θα αναπτύξουν δεξιότητες όπως η επίλυση προβλημάτων, η κριτική σκέψη, η λήψη αποφάσεων και η ομαδική εργασία, ενώ θα έρθουν και σε επαφή με επιστημονικές διαδικασίες (Sisman et al, 2019).

#### 4.3 Μέθοδοι Διδασκαλίας στην Εκπαιδευτική Ρομποτική.

Όπως προαναφέρθηκε, ως «μέθοδος διδασκαλίας» ορίζεται η συλλογιστική διαδικασία μέσα από την οποία αναμένεται ο μαθητής να κατανοήσει το περιεχόμενο του διδασκόμενου θέματος και των διδασκόμενων εννοιών.

Η πιο βασική μέθοδος διδασκαλίας στην Εκπαιδευτική Ρομποτική είναι η Μάθηση βασισμένη σε Project, Project-Based Learning (PBL). Είναι μια διδακτική προσέγγιση όπου οι μαθητές λειτουργούν συνεργατικά προκειμένου να αντιμετωπίσουν προκλήσεις και να λύσουν αυθεντικά προβλήματα, τα οποία βασίζονται στο πρόγραμμα σπουδών τους και τις περισσότερες φορές έχουν διεπιστημονικό χαρακτήρα. Σε αυτή τη διδακτική προσέγγιση, ο μαθητής βρίσκεται στο επίκεντρο της μαθησιακής διαδικασίας, είναι υπεύθυνος για τη δική του μάθηση και δεν αντιμετωπίζεται ως παθητικός αποδέκτης της γνώσης (Cocota, 2015). Η μέθοδος αυτή χαρακτηρίζεται από την αυτονομία των μαθητών, την ανάπτυξη της ικανότητας για καθορισμό των επιδιωκόμενων στόχων, τη συνεργασία, την επικοινωνία και τον προβληματισμό πάνω σε αυθεντικά προβλήματα. Η Project- Based Learning διαδικασία βασίζεται σε τρεις κοστροκτιβιστικές αρχές: α) η μάθηση ορίζεται μέσα ένα συγκεκριμένο πλαίσιο (context – specific learning), β) οι μαθητές συμμετέχουν ενεργά στην διαδικασία της μάθησης και γ) οι μαθητές επιτυγχάνουν τους στόχους τους μέσα από κοινωνικές αλληλεπιδράσεις και μέσω της κοινοποίησης των γνώσεων. (Kokotsaki et al 2016). Μέσα από την ανάθεση ενός Project που αφορά μια πραγματική πρακτική, οι μαθητές μαθαίνουν να συλλέγουν πληροφορίες από διάφορες και έγκυρες πηγές, αναλύοντάς τις και συνθέτοντας νέα γνώση, την οποία μετά θα πρέπει να παρουσιάσουν και στους υπόλοιπους μαθητές (Blumenfeld et al., 2000). Σε όλη τη διαδικασία της μάθησης ο μαθητής βρίσκεται στο επίκεντρο, ενώ ο ρόλος του δασκάλου είναι περισσότερο υποστηρικτικός και καθοδηγητικός, έχοντας ως καθήκον να συμβουλεύει τους μαθητές παρά να διαχειρίζεται το Project που τους έχει ανατεθεί.

Το 2004 οι Carbonaro, Rex και Chambers, πιστεύοντας πως θα έπρεπε να υπάρξουν αλλαγές στην εκπαίδευση και, αντί να μαθαίνουμε από την τεχνολογία (δηλαδή, με τη χρήση προγραμμάτων υπολογιστών), θα πρέπει να προσπαθήσουμε να μαθαίνουμε με την τεχνολογία (δηλαδή, ρομποτική), παρουσίασαν ένα μοντέλο το οποίο προέκυψε μετά από έρευνα σε Project-Based περιβάλλοντα και το οποίο

αποτελεί το πιο διαδεδομένο μοντέλο εκπαίδευσης στην εκπαιδευτική ρομποτική. Το μοντέλο αυτό αποτελείται από πέντε στάδια τα οποία είναι τα εξής:

- **Στάδιο Ενεργοποίησης:** Σε αυτή τη φάση παρουσιάζεται στους μαθητές το πρόβλημα το οποίο θα πρέπει να αναλύσουν. Αφού χωριστούν οι μαθητές σε ομάδες προχωρούν στην ανάλυση του προβλήματος και μοιράζονται τις απόψεις και τις ιδέες τους, με σκοπό να προσδιοριστούν και να εμπλουτιστούν οι λεπτομέρειες του προβλήματος. Υπάρχουν διάφορες πρακτικές που διέπουν αυτό το στάδιο, όπως είναι η συζήτηση με χρήση ερωτήσεων, ο καταγιγισμός ιδεών και η δημιουργία πινάκων.
- **Στάδιο Εξερεύνησης:** Κατά τη διάρκεια του σταδίου της εξερεύνησης οι μαθητές εκτελούν συγκεκριμένες ενέργειες και εργασίες, προκειμένου να αποκτήσουν συγκεκριμένες γνώσεις και δεξιότητες που θα τους βοηθήσουν να φτάσουν στην επίλυση του προβλήματος. Σε αυτό το στάδιο οι μαθητές γνωρίζουν τους πιθανούς εξοπλισμούς.
- **Στάδιο Διερεύνησης:** Οι μαθητές σε αυτή τη φάση λειτουργούν κάνοντας προβλέψεις και σχεδιάζοντας πειράματα, τα οποία εκτελούν προκειμένου να κάνουν διάφορες δοκιμές. Σκοπός αυτού του σταδίου είναι να μπορέσουν οι μαθητές να πειραματιστούν και να δοκιμάσουν διάφορες λύσεις ελεύθερα, εντοπίζοντας ενδεχόμενα προβλήματα που μπορούν να προκύψουν σε κάθε λύση και διορθώνοντάς τα.
- **Στάδιο Σύνθεσης και Δημιουργίας:** Οι μαθητές προχωρούν στη σχεδίαση και στη δοκιμή του τελικού τους σχεδίου, καθώς και στην τροποποίησή του σε περίπτωση που αυτό απαιτείται.
- **Στάδιο Παρουσίασης και Αξιολόγησης:** Στο στάδιο της παρουσίασης οι μαθητές παρουσιάζουν το τελικό project και την πρότασή τους για την επίλυση του προβλήματος. Σε αυτό το στάδιο υπάρχει και μια άτυπη αυτό-αξιολόγηση, αλλά και μια ετερο-αξιολόγηση της ομάδας, της λύσης του προβλήματος, καθώς και της γνώσης που αποκτήθηκε.

Είναι πολύ σημαντικό να αναφερθεί πως η εκτέλεση των παραπάνω σταδίων δεν είναι αυστηρώς γραμμική, καθώς σε πολλές περιπτώσεις μπορεί οι μαθητές να χρειαστεί να ξαναγυρίσουν σε ένα προηγούμενο στάδιο και να επανεκτιμήσουν τις

κινήσεις τους. Εάν για παράδειγμα στο στάδιο της Σύνθεση και Δημιουργίας οι μαθητές δεν είναι ικανοποιημένοι από τα αποτελέσματα της δοκιμής του τελικού σχεδίου, μπορούν να επιστρέψουν στο στάδιο της Διερεύνησης και να δοκιμάσουν εκ νέου πειράματα και πιθανές λύσεις.

Τέλος, είναι σημαντικό να αναφερθεί πως, παρότι δεν υπάρχει αξιολόγηση αυτή καθαυτή από τον εκπαιδευτικό μέσω ενός διαγωνίσματος για παράδειγμα, υπάρχουν κάποια κριτήρια τα οποία θα πρέπει να θέτουν οι εκπαιδευτικοί στα PBL περιβάλλοντα, προκειμένου να αξιολογήσουν ότι οι μαθητές απέκτησαν όλα τα επιδιωκόμενα μαθησιακά αποτελέσματα. Αρχικά, θα πρέπει να σιγουρευτούν πως οι μαθητές κατανοούν την διαδικασία με την οποία προσεγγίζουν το πρόβλημα και πως αντιλαμβάνονται γιατί κάνουν τις συγκεκριμένες επιλογές ώστε να φτάσουν στην επίλυση του προβλήματος. Επιπρόσθετα, θα πρέπει να δουν πως οι μαθητές είναι σε θέση να σκέφτονται και να κατανοούν τα αποτελέσματα των εργασιών τους. Αυτό μπορεί να επιτευχθεί και μέσα από συνδυασμούς προφορικού λόγου με βίντεο, ή παρουσιάσεις, ή μέσω γραπτών ημερολογίων. Τέλος, οι κατευθυντήριες γραμμές του project θα πρέπει να περιλαμβάνουν ορισμένα κριτήρια σε διάφορα σημεία κατά τη διάρκεια της διαδικασίας. Ελέγχοντας στο τέλος ο εκπαιδευτικός πως πληρούνται τα κριτήρια όλων των σημείων του Project μπορεί να εξασφαλίσει πως και το Project ολοκληρώθηκε με επιτυχία.

#### **4.4 Εκπαιδευτικά Ρομποτικά Συστήματα.**

Η ανάδειξη της εκπαιδευτικής ρομποτικής, όντας πλέον ένα χρήσιμο εργαλείο στην εκπαίδευση του 21<sup>ου</sup> αιώνα, είχε σαν αποτέλεσμα την ενσωμάτωση διαφόρων ρομποτικών συστημάτων τόσο στον χώρο της εκπαίδευσης, αλλά και την παραγωγή νέων συστημάτων ειδικά διαμορφωμένων για εκπαιδευτικούς σκοπούς. Σε αυτό το σημείο θα παρουσιάσουμε μερικούς από τους πιο διαδεδομένους ρομποτικούς μηχανισμούς που χρησιμοποιούνται σε όλες τις εκπαιδευτικές βαθμίδες.

### Bee – bot.

Η δυσκολία των δασκάλων στην διδασκαλία του προγραμματισμού στις αρχές του 21<sup>ου</sup> αιώνα οδήγησε στην εμφάνιση μιας καινοτόμου και ελκυστικής συσκευής, της «έξυπνης μέλισσας» γνωστής ως Bee-bot, η οποία αποτελεί πλέον την πιο διαδεδομένη ρομποτική πλατφόρμα για την διδασκαλία προγραμματισμού και την ανάπτυξη της αλγοριθμικής σκέψης, τόσο σε παιδιά του νηπιαγωγείου αλλά και στις πρώτες τάξεις του δημοτικού. Το Bee-bot είναι ένα προγραμματιζόμενο ρομπότ δαπέδου, το οποίο έχει τη μορφή μιας μέλισσας και διαθέτει πλήκτρα στο επάνω μέρος της όψης του, με τα οποία οι μαθητές προγραμματίζουν το ρομπότ να κινείται μέσα στο χώρο.



*Εικόνα 4.1 Η μέλισσα Bee-bot.*

Οι κινήσεις που μπορεί να εκτελέσεις είναι μπροστά, πίσω, δεξιά και αριστερά, και είναι κατασκευασμένο ώστε να τις εκτελεί με μεγάλη ακρίβεια μέσα στον χώρο. Συνήθως χρησιμοποιείται από τους καθηγητές σε συνδυασμό με διάφορους χάρτες, πάνω στους οποίους κινείται η μέλισσα προκειμένου να εκτελέσει διάφορες αποστολές. Στόχος είναι να εισαχθούν αποτελεσματικά οι μαθητές των μικρότερων ηλικιών στον κόσμο του προγραμματισμού, να αναπτύξουν την αλγοριθμική σκέψη που είναι βασική για την επίλυση προβλημάτων, αλλά και να αποκτήσουν μια πρώτη επαφή για τον τρόπο με τον οποίο επικοινωνούν οι άνθρωποι με τις μηχανές.



### Kids First Coding & Robotics.

Το “Kids First Coding & Robotics” είναι ένα kit που απευθύνεται σε παιδιά προσχολικής και πρωτοσχολικής ηλικίας, έχοντας ως στόχο να εισάγει τα παιδιά στις βασικές έννοιες προγραμματισμού (όπως είναι η ακολουθία, η επανάληψη, τα συμβάντα κ.α.) μέσα από τον συνδυασμό της χρήσης ρομποτικών συστημάτων και της αφήγησης (storytelling). Αποτελείται από μικρά ρομποτικά συστήματα τα οποία προσομοιάζουν διάφορες φιγούρες, για παράδειγμα ένα ποντίκι, έναν πιγκουίνο, αλλά και το ρμπότ Sammy, ένα ρομποτικό σύστημα που θυμίζει ένα παιδί – ρμπότ.



*Εικόνα 4.2 Το εκπαιδευτικό Kit: Kids First Coding & Robotics.*

Ταυτόχρονα, διαθέτει και μικρά τουβλάκια με τα οποία οι μαθητές μπορούν να κατασκευάσουν τις δικές τους ρομποτικές φιγούρες. Χρησιμοποιώντας τις κάρτες, οι μαθητές δημιουργούν ένα πρόγραμμα – μονοπάτι το οποίο στη συνέχεια θα εκτελέσει το ρμπότ. Το ρμπότ διαβάζει την ακολουθία περνώντας πάνω από τις κάρτες με τη βοήθεια ενός αισθητήρα που διαθέτει στο κάτω μέρος, και έτσι γνωρίζει το πρόγραμμα το οποίο θα πρέπει να εκτελέσει. Στη συνέχεια, οι μαθητές τοποθετούν το ρμπότ μέσα σε μια πίστα στην οποία εκτελεί το πρόγραμμα που έχει διαβάσει. Δεν απαιτεί χρήση Υ/Η ή tablet, καθώς ολόκληρος ο προγραμματισμός του συστήματος γίνεται αποκλειστικά με τη χρήση καρτών. Επιπρόσθετα, διαθέτει προγραμματιζόμενα φώτα LED, ηχείο για αναπαραγωγή ήχων αλλά και επιπλέον δομικά μέρη όπως γρανάζια, τα οποία επιτρέπουν την δημιουργία απλών ρομποτικών κατασκευών. Το 2018 κέρδισε το Βραβείο Σχεδιασμού “Golden Pin 2018”, ένα από τα σημαντικότερα βραβεία που αφορούν στον σχεδιασμό προϊόντος.

### Vex Robotics Design System.

Το Vex Robotics Design System δημιουργήθηκε από την εταιρεία Innovation First, Inc., και θεωρείται ως μια από τις κορυφαίες πλατφόρμες για την προώθηση της εκπαιδευτικής ρομποτικής. Διαθέτει μεταλλικά στοιχεία, αισθητήρες και έναν προγραμματιζόμενο μικροελεγκτή, τα οποία δίνουν τη δυνατότητα στους μαθητές να δημιουργήσουν τηλεκατευθυνόμενα, αυτόνομα ή και ημιαυτόνομα ρομποτικά συστήματα.



*Εικόνα 4.3 Το εκπαιδευτικό Kit: Vex Robotics Design System.*

Είναι βασισμένο και σχεδιασμένο πάνω στις αρχές της STEM εκπαίδευσης, και σκοπός του είναι να εισάγει αποτελεσματικά τα παιδιά τόσο στο πεδίο της επιστήμης, της τεχνολογίας, της μηχανικής και των μαθηματικών, όσο και να καλλιεργήσει την συνεργατικότητα, την επίλυση προβλημάτων και την ικανότητα της ηγεσίας σε μικρές ομάδες μαθητών. Τέλος, είναι σχεδιασμένο ώστε να δίνει στους εκπαιδευτικούς τη δυνατότητα να τροποποιούν και να προσαρμόζουν εύκολα τα έργα τους στο επίπεδο των ικανοτήτων των μαθητών.

### Nao Robot.

Το ρομπότ Nao είναι ένα αυτόνομο, προγραμματιζόμενο ανδροειδές ρομπότ, το οποίο δημιουργήθηκε από την εταιρεία Aldebaran Robotics. Αποτελεί το πιο δημοφιλές ανδροειδές ρομπότ τόσο στον χώρο της εκπαίδευσης, καθώς χρησιμοποιείται σε πάνω

από 70 χώρες σε όλον τον κόσμο, όσο και στον χώρο της εκπαιδευτικής έρευνας. Αποτελείται από αισθητήρες που του επιτρέπουν να λειτουργεί στον χώρο, να αναγνωρίζει πρόσωπα και αντικείμενα, ενώ ταυτόχρονα διαθέτει και διάφορους ενεργοποιητές που του επιτρέπουν να μιλά, να περπατά και να χορεύει.



*Εικόνα 4.4 Το ρομπότ Nao.*

Το ρομπότ Nao διαθέτει επίσης και τη δική του προσωπικότητα, την οποία χρησιμοποιεί για να εμπνεύσει τους μαθητές να παραμείνουν συγκεντρωμένοι και να συνεχίσουν να προσπαθούν. Χρησιμοποιείται ως συντονιστής της εκπαιδευτικής διαδικασίας, ως βοηθός καθηγητή και μαθητών αλλά και σαν εργαλείο στα πλαίσια του μαθήματος. Οι πιο διαδεδομένες χρήσεις του είναι η αξιοποίησή του ως βοηθός στην εκμάθηση ξένων γλωσσών, αλλά και ως υποστηρικτής στην εκπαίδευση παιδιών που ανήκουν στο φάσμα του αυτισμού.

#### *Arduino & Nezha Inventor's Kit for Arduino.*

Το Arduino είναι μια πλακέτα ανοικτής πηγής (Open-Source Hardware) με έναν ενσωματωμένο μικροελεγκτή, η οποία μπορεί να προγραμματιστεί με τη χρήση της γλώσσας Wiring. Οι πλακέτες είναι εξοπλισμένες με αναλογικές και ψηφιακές εισόδους και εξόδους που μπορούν να συνδεθούν με διάφορες πλακέτες επέκτασης breadboards, αισθητήρες και συσκευές εξόδου (π.χ. οθόνη LCD, LEDs) για την κατασκευή πρωτοτύπων και άλλων κυκλωμάτων. Χρησιμοποιείται στην εκπαίδευση προκειμένου οι μαθητές να μάθουν να δημιουργούν κυκλώματα, να έρθουν σε επαφή με τον χώρο της ηλεκτρονικής, αλλά και να μάθουν να κατασκευάζουν έξυπνες συσκευές που θα μπορούν να αντιληφθούν το περιβάλλον γύρω τους και να

ανταποκρίνονται αναλόγως. Ταυτόχρονα χρησιμοποιείται και ως εργαλείο για την διδασκαλία του προγραμματισμού.



*Εικόνα 4.5 Το εκπαιδευτικό Kit: Arduino & Nezha Inventor's Kit for Arduino.*

Το Arduino όμως δεν δημιουργήθηκε για εκπαιδευτικούς σκοπούς, γι' αυτό και στη συνέχεια εμφανίστηκε το Nezha Inventor's Kit for Arduino, ένα Kit ρομποτικής το οποίο έχει ενσωματωμένη μια πλακέτα Arduino. Το πακέτο περιλαμβάνει πάνω από 400 δομικά στοιχεία (τουβλάκια), κινητήρες, αισθητήρες και ενεργοποιητές, καθώς και έτοιμες δραστηριότητες για τους μαθητές. Ιδιαίτερα σημαντικό είναι το γεγονός ότι πάνω στην πλακέτα μπορούν να κουμπώσουν δομικά στοιχεία της Lego, τουβλάκια δηλαδή, με τα οποία είναι ήδη εξοικειωμένα τα παιδιά από μικρή ηλικία. Το πλεονέκτημα που προσφέρει αυτό το Kit έναντι του απλού Arduino είναι ότι επιτρέπει στους μαθητές να κατασκευάσουν τα δικά τους ρομποτικά συστήματα και έτσι να αποκτήσουν και γνώσεις μηχανικής. Τέλος, οι μαθητές μπορούν να προγραμματίσουν στο περιβάλλον του Arduino οποιονδήποτε ρομποτικό μηχανισμό έχουν κατασκευάσει.

### *Lego Robotics Systems.*

Η ιστορία των ρομποτικών συστημάτων της Lego ξεκινά το 1984, όταν ο ιδιοκτήτης της Kjeld Kirk Kristiansen παρακολούθησε μια τηλεοπτική εκπομπή, όπου ο καθηγητής Seymour Papert από το MIT (Ινστιτούτο Τεχνολογίας της

Μασαχουσέτης) παρουσίαζε την γλώσσα προγραμματισμού του για παιδιά, γνωστή και ως LOGO. Οι απόψεις του Papert για τον Εποικοδομισμός και το πώς τα παιδιά μαθαίνουν μέσα από το παιχνίδι είχαν ιδιαίτερη απήχηση στον Kjeld Kirk, καθώς παρουσίαζαν μεγάλες ομοιότητες με την φιλοσοφία της εταιρείας του. Μετά από συνεργασία του Kjeld Kirk με τον Seymour Papert κυκλοφόρησαν τα πρώτα ρομποτικά συστήματα της LEGO. Το πιο αξιοσημείωτο προϊόν που σχεδιάστηκε από τη συνεργασία τους είναι το LEGO MINDSTORMS: the Robotics Invention System, το οποίο κυκλοφόρησε το 1998. Στην συνέχεια, το 2006 κυκλοφόρησε στην ίδια σειρά το Mindstorms NXT, και το 2013 το Mindstorms EV3. Αξίζει να αναφερθεί πως, μαζί με αυτές τις κυκλοφορίες, κυκλοφορούσαν ταυτόχρονα ξεχωριστές παραλλαγές αυτών των kit για εκπαιδευτικούς σκοπούς.

Το 2009 η LEGO κυκλοφόρησε το πρώτο set της σειράς WeDo, ένα εκπαιδευτικό kit το οποίο αποτελούνταν από 158 δομικά στοιχεία, δύο αισθητήρες, ένα hub και έναν κινητήρα. Το 2016 όμως αντικαταστάθηκε από το LEGO WeDo 2.0, το οποίο ήταν μια πιο εξελιγμένη μορφή της πρώτης έκδοσης και έδινε περισσότερες δυνατότητες στους μαθητές να κατασκευάζουν πιο εξελιγμένους ρομποτικούς μηχανισμούς. Ένα από τα πιο σημαντικά πλεονεκτήματα ήταν ότι η δεύτερη έκδοση περιλάμβανε και δύο set από ρόδες, γεγονός που έδινε στους μαθητές την δυνατότητα να κατασκευάζουν και κινητά ρομπότ.



*Εικόνα 4.6 Το εκπαιδευτικό Kit: Lego WeDo 2.0.*

Παρότι τα συγκεκριμένα sets έχουν γνωρίσει μεγάλη επιτυχία και έχουν χρησιμοποιηθεί – και σε μερικές περιπτώσεις χρησιμοποιούνται ακόμα – τόσο στην εκπαίδευση όσο και σε εκπαιδευτικούς διαγωνισμούς, η LEGO έχει αποφασίσει την

απόσυρσή τους, έχοντας πλέον μόνο δύο διαθέσιμα εκπαιδευτικά ρομποτικά kits: το Spike Prime και το Spike Essential. Το Spike Prime κυκλοφόρησε το 2020 και απευθύνεται σε παιδιά ηλικίας 7 ετών και άνω. Εφόσον αποτελεί και το kit πάνω στο οποίο στηρίχθηκε η παρούσα εργασία θα παρουσιαστεί αναλυτικότερα σε επόμενο κεφάλαιο. Για τις μικρότερες ηλικίες, το 2021 κυκλοφόρησε το Spike Essential. Αποτελεί την πιο πρόσφατη κυκλοφορία της LEGO και σχεδιάστηκε προκειμένου οι μαθητές να αναπτύξουν τόσο τις γνώσεις και τις δεξιότητες τους στα πεδία του STEAM, όσο και για την ανάπτυξη κοινωνικών δεξιοτήτων. Το kit, εκτός από τα στοιχεία που χρησιμοποιούνται για την δημιουργία ρομποτικών μηχανισμών, περιλαμβάνει και 4 χαρακτήρες, καθένας με τη δική του ξεχωριστή προσωπικότητα. Βασική αρχή του συγκεκριμένου kit είναι να εισάγονται οι μαθητές σε αφηγήσεις, όπου ο βασικός χαρακτήρας είναι μια από τις 4 φιγούρες, στη συνέχεια να τους παρουσιάζεται ένα πρόβλημα που αντιμετωπίζει ο συγκεκριμένος χαρακτήρας, και τελικά οι μαθητές να ξεκινούν μια διαδικασία επίλυσης του προβλήματος.

#### **4.5 Πλεονεκτήματα Εκπαιδευτικής Ρομποτικής.**

Όπως προαναφέρθηκε, η ραγδαία ανάπτυξη των τεχνολογιών της ρομποτικής τα τελευταία χρόνια ανέπτυξε πολυάριθμα ανεπτυγμένα εργαλεία, τα οποία έχουν ενσωματωθεί πλέον και στον χώρο της εκπαίδευσης και έχουν οδηγήσει σε πιο αποτελεσματική διδασκαλία σε πληθώρα μαθημάτων.

Ένα σημαντικό πλεονέκτημα της χρήσης της ρομποτικής στην εκπαίδευση είναι ότι οι αφηρημένες έννοιες μπορούν να παρουσιαστούν έμπρακτα και να μετατραπούν σε προβλήματα και λύσεις του πραγματικού κόσμου. Η εμπλοκή των μαθητών σε πρακτικές επιδείξεις και η πρακτική εμπειρία που αποκτούν συνεπάγεται την ευκολότερη κατανόηση των εννοιών που θέλουμε να διδαχθούν οι μαθητές, αλλά επίσης και αύξηση των κινήτρων τους για εμπλοκή στη μάθηση. Η βελτίωση του κινήτρου σίγουρα λαμβάνει χώρα όταν περιλαμβάνονται και αντικείμενα του πραγματικού κόσμου. Καθώς λοιπόν οι μαθητές μπορούν να δουν, να αγγίξουν και να παίξουν με το ρομποτικό σύστημα το οποίο οι ίδιοι έχουν αναπτύξει αυξάνεται το κίνητρο τους για μάθηση (Gerecke & Wagner, 2013).

Επιπρόσθετα, μέσω της μάθησης βάσει προβλημάτων, την κατασκευή και τον προγραμματισμό σύνθετων ρομποτικών μηχανισμών, οι μαθητές έχουν βελτιώσει σημαντικά τις δεξιότητές τους στα μαθηματικά, τη φυσική, τον στρατηγικό σχεδιασμό και την επικοινωνία (Nag et al., 2013). Εκτός από την ανάπτυξη δεξιοτήτων στα πεδία των θετικών επιστημών, μέσα από την κατασκευή μοντέλων ρομποτικής οι μαθητές αποκτούν και μια βαθύτερη κατανόηση των μαθηματικών και των φυσικών εννοιών (Varney et al., 2012). Τέλος, οι μαθητές αποκτούν δεξιότητες που θεωρούνται σημαντικά συστατικά του επιστημονικού γραμματισμού, όπως η μοντελοποίηση και η μάθηση με μοντέλα (Cuperman et al., 2013).

Σημαντικό πλεονέκτημα αποτελεί επίσης και η ανάπτυξη της υπολογιστικής σκέψης. Η υπολογιστική σκέψη αντιπροσωπεύει πλέον μια καθολικά εφαρμόσιμη δεξιότητα, η οποία θα έπρεπε να αποτελεί στόχο όλων μας προς εκμάθηση και όχι μόνο από τους επιστήμονες των υπολογιστών (Wing, 2006). Για το λόγο αυτό θα πρέπει να διδάσκεται από πολύ μικρή ηλικία, μαζί με την ανάγνωση, την γραφή και την εκμάθηση των μαθηματικών (Ιοαννου, 2018). Η Υπολογιστική σκέψη και η Εκπαιδευτική Ρομποτική έχουν μια φυσική συμβιωτική σχέση, και μπορούν να εργαστούν μαζί προκειμένου να προσφέρουν μεγάλες εκπαιδευτικές ευκαιρίες στους μαθητές (Catlin & Woollard, 2014).

Τέλος, σημαντική είναι η συνεισφορά της εκπαιδευτικής ρομποτικής και στην ανάπτυξη κοινωνικών δεξιοτήτων, όπως η ομαδική εργασία και η ικανότητα για συνεργασία, αλλά και επικοινωνιακών δεξιοτήτων, όπως η ικανότητα τους να μοιράζονται τις ιδέες τους και να παρουσιάζουν τα ευρήματά τους στο κοινό (Khanlari, 2015). Όλα αυτά έχουν σαν αποτέλεσμα την ανάπτυξη των διαπροσωπικών σχέσεων των μαθητών, αλλά και την απόκτηση κινήτρων ώστε να γίνουν δια βίου μαθητές.

#### **4.6 Εκπαιδευτική Ρομποτική και Διδασκαλία Προγραμματισμού.**

Ο προγραμματισμός υπολογιστών είναι ένας κλάδος της επιστήμης της πληροφορικής, όπου οι μαθητές μαθαίνουν να λύνουν προβλήματα μέσω της αλληλεπίδρασης με έναν υπολογιστή. Επί του παρόντος, μια τέτοια αλληλεπίδραση πραγματοποιείται μέσα από ένα λογισμικό που ονομάζεται Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environments – IDE), το οποίο

αποτελεί και το κύριο περιβάλλον μάθησης. Η διεπαφή τους όμως είναι αρκετά πολύπλοκη, άρα και κατ' επέκταση μη φιλική για νέους μαθητές, καθιστώντας τα έτσι μη θελκτικά εργαλεία εκμάθησης, αποθαρρύνοντάς τους από την ενασχόληση με την επιστήμη των υπολογιστών.

Για τον λόγο αυτό, η κοινότητα των επιστημόνων και των εκπαιδευτικών έχει στραφεί προς την χρήση της εκπαιδευτικής ρομποτικής ως εργαλείο εκμάθησης του προγραμματισμού. Αυτά τα εναλλακτικά περιβάλλοντα, στα οποία προγραμματίζονται τα εκπαιδευτικά ρομποτικά συστήματα, διαθέτουν ένα οπτικό περιβάλλον προγραμματισμού πιο κατάλληλο για παιδιά μικρά που μαθαίνουν προγραμματισμό για πρώτη φορά (Fagin et al., 2001). Το περιβάλλον τους είναι σχεδιασμένο με τέτοιο τρόπο ώστε να είναι πιο φιλικό και εύπεπτο προς τις μικρές ηλικίες, σε αντίθεση με τα περιβάλλοντα όπου μαθαίνουν και εργάζονται οι προγραμματιστές που χρησιμοποιούν γλώσσες υψηλού επιπέδου.

Βασικό όφελος που έχει η εκμάθηση προγραμματισμού με τη χρήση εκπαιδευτικών ρομποτικών συστημάτων είναι η άμεση ανατροφοδότηση των αποτελεσμάτων που λαμβάνουν οι μαθητές όταν γράφουν κώδικα, καθώς η εκτέλεση και τα αποτελέσματα του προγραμματισμού αντικατοπτρίζονται σε ένα φυσικό και απτό σύστημα. Η εργασία αυτή πάνω σε ένα φυσικό μοντέλο που παρουσιάζει παρατηρήσιμη συμπεριφορά είναι πιο ελκυστική για τους μαθητές από την επίλυση προβλημάτων σε κοινά περιβάλλοντα προγραμματισμού (Athanasiou et al., 2017). Με αυτόν τον τρόπο, η χρήση του ρομποτικού μηχανισμού βοηθά τους μαθητές να οπτικοποιήσουν και να συνειδητοποιήσουν καλύτερα τις λύσεις τους, ενώ ταυτόχρονα μαθαίνουν και για την μεθοδολογία επίλυσης προβλημάτων, όπου θα πρέπει να αναλύσουν προβλήματα, να σχεδιάσουν πιθανές λύσεις, να τις εφαρμόσουν, και στο τέλος να αξιολογήσουν τις λύσεις τους (Flowers & Gossett, 2002).

Το σημαντικότερο όφελος όμως που προσφέρει η χρήση ρομποτικών συστημάτων στην εκμάθηση του προγραμματισμού είναι η εστίαση στην εκμάθηση του προγραμματισμού, η οποία είναι ανεξάρτητη των γλωσσών. Οι μαθητές δεν μαθαίνουν απλά μια γλώσσα προγραμματισμού, αλλά καθολικές αλήθειες και τεχνικές προγραμματισμού (Lawhead et al., 2002), όπως οι βρόγχοι επανάληψης, η διακλάδωση, η καταχώρηση δεδομένων κ.α. Το γεγονός αυτό είναι πολύ σημαντικό, καθώς στην εκμάθηση του προγραμματισμού απαραίτητη προϋπόθεση για να μπορέσει



να δημιουργήσει κανείς ένα πρόγραμμα δεν είναι απλά η εκμάθηση των εντολών που χρησιμοποιεί η εκάστοτε γλώσσα, αλλά η ανάπτυξη της αλγοριθμικής σκέψης, του τρόπου δηλαδή με τον οποίο πρέπει να σκέφτεται ώστε να μπορέσει να αναπτύξει έναν αλγόριθμο και να επιλύσει το εκάστοτε πρόβλημα.

## Κεφάλαιο 5 – Παρουσίαση του Spike Prime

### Εισαγωγή.

Σε αυτό το κεφάλαιο θα ακολουθήσει η ανάλυση του Kit εκπαιδευτικής ρομποτικής που χρησιμοποιήσαμε, του Lego Spike Prime. Αφού παρουσιάσουμε κάποιες γενικές πληροφορίες για αυτό, θα αναλύσουμε το περιεχόμενό του, δίνοντας ιδιαίτερη έμφαση στα εξαρτήματά του.

### 5.1 Πληροφορίες για το Spike Prime.

Το LEGO Education SPIKE Prime Set κυκλοφόρησε στις 22 Μαΐου 2020 και θεωρείται πλέον ένα βασικό εργαλείο εκμάθησης STEAM για μαθητές από 11 έως 13 ετών. Συνδυάζει πολύχρωμα δομικά στοιχεία LEGO, βοηθητικό υλικό για μαθητές και δασκάλους, καθώς και τρεις διαφορετικές επιλογές προγραμματισμού.



Εικόνα 5.1 Το εκπαιδευτικό Kit: Lego Spike Prime.

Σκοπός του είναι να καταφέρει να εισάγει τους μαθητές στην STEAM εκπαίδευση μέσα από ευχάριστες μαθησιακές δραστηριότητες, κάτι που θα τους βοηθήσει στην ανάπτυξη της κριτικής τους σκέψης και στην επίλυση σύνθετων προβλημάτων ανεξάρτητα από το μαθησιακό τους επίπεδο. Το 2021, τα Toys Awards το ανακήρυξαν ως το καλύτερο παιχνίδι για το 2021, λαμβάνοντας τόσο την GOLD όσο και την τιμητική PLATINUM βράβευση.

## 5.2 Επισκόπηση του Kit.

Αγοράζοντας κανείς το Lego SPIKE Prime Kit θα βρει μέσα στο κουτί όλα όσα χρειάζεται για να μπορέσει να ξεκινήσει την κατασκευή και τη δημιουργία λειτουργικών ρομποτικών συστημάτων. Πιο συγκεκριμένα, το kit απαρτίζεται από:

- Ένα φυλλάδιο, στο μπροστινό μέρος του οποίου αναγράφεται μια [ηλεκτρονική διεύθυνση](#) που προτείνεται να επισκεφθεί κανείς, ειδικά αν είναι αρχάριος στον τομέα της ρομποτικής, για να ξεκινήσει να χρησιμοποιεί το Spike Prime και να βρει περισσότερες χρήσιμες πληροφορίες γι' αυτό. Στο πίσω μέρος του φυλλαδίου απεικονίζονται όλα τα διαφορετικά κομμάτια Lego που συναντά κανείς μέσα στο κουτί, το πλήθος τους, αλλά και όλα τα ηλεκτρονικά εξαρτήματα που περιέχονται.
- Δύο λευκούς δίσκους, οι οποίοι συνδυάζονται με τα αυτοκόλλητα που απεικονίζουν όλα τα κομμάτια του Spike κατανεμημένα σε ομάδες. Χρησιμοποιούνται για την αποθήκευση και την ταξινόμηση των κομματιών του πριν και μετά την κατασκευή.
- Ένα micro USB καλώδιο για σύνδεση του Spike με άλλες συσκευές.
- 523 στοιχεία (για παράδειγμα τουβλάκια, δοκούς, γρανάζια) με τα οποία μπορεί να σχεδιάσει και να κατασκευάσει έναν ρομποτικό μηχανισμό.
- Ένα Έξυπνο Τούβλο/Hub.
- Μια επαναφορτιζόμενη μπαταρία για τη λειτουργία του Hub.
- Τρεις εξωτερικούς αισθητήρες: έναν αισθητήρα χρώματος, έναν αισθητήρα απόστασης και έναν αισθητήρα αφής.
- Δύο μεσαίους κινητήρες και έναν μεγάλο κινητήρα.

Ακολουθεί η ανάλυση των ηλεκτρονικών στοιχείων που το αποτελούν.

### Έξυπνο Τούβλο (Hub):

Το Έξυπνο Τούβλο/Hub είναι μια επαναπρογραμματιζόμενη μονάδα, υπεύθυνη για την επεξεργασία δεδομένων και τον έλεγχο των προγραμμάτων. Είναι πολύ σημαντικό εξάρτημα καθώς ουσιαστικά λειτουργεί σαν τον «εγκέφαλο» του ρομποτικού συστήματος. Αποτελείται από μια επαναφορτιζόμενη μπαταρία, η οποία τροφοδοτεί τον μηχανισμό με την ενέργεια που χρειάζεται. Έχει ενσωματωμένο έναν μικροεπεξεργαστή και μνήμη 32 MB, ικανό να αποθηκεύσει μέχρι και 20 προγράμματα.



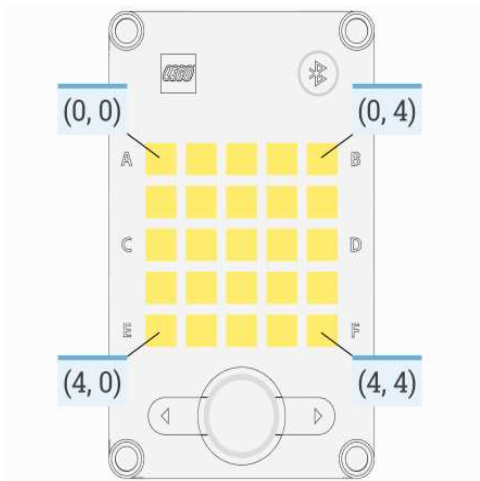
*Εικόνα 5.2 Απεικόνιση του Hub.*

Στο πλάι, διαθέτει 6 θύρες σύνδεσης περιφερειακών (ονομάζονται με τα αγγλικά γράμματα A, B, C, D, E και F) στις οποίες μπορούν να συνδεθούν αισθητήρες και κινητήρες, και μπορούν να χρησιμοποιηθούν είτε ως είσοδοι είτε ως έξοδοι του ρομποτικού συστήματος. Σε προγράμματα που χρειαζόμαστε υψηλή ταχύτητα και απόκριση από κάποιο εξάρτημα συνιστάται η χρήση της θύρας E ή F. Τέλος, διαθέτει και μια θύρα σύνδεσης USB για τη σύνδεση του Hub με τον υπολογιστή, ώστε να μεταφέρονται τα προγράμματα από τον υπολογιστή από και προς το Hub, καθώς και για την επαναφόρτιση της μπαταρίας.

Στο επάνω μέρος διαθέτει ένα κουμπί για σύνδεση με Bluetooth, μια οθόνη 5x5 με LEDs και τρία κουμπιά πλοήγησης, ένα αριστερό κουμπί, ένα δεξί κουμπί και ένα κεντρικό κουμπί το οποίο περιβάλλεται από ένα προγραμματιζόμενο φως, μπορεί δηλαδή να προγραμματιστεί από τον χρήστη ώστε να αλλάζει χρώμα. Για την ενεργοποίηση του Hub ο χρήστης θα πρέπει να πατήσει το μεσαίο στρόγγυλο πλήκτρο και για την απενεργοποίηση του θα πρέπει να το κρατήσει πατημένο για μερικά δευτερόλεπτα. Επίσης, με τα κουμπιά ο χρήστης μπορεί να περιηγηθεί στο menu του Hub και να εκτελέσει διάφορες εργασίες, όπως το να επιλέξει ποιο από τα αποθηκευμένα προγράμματα θέλει να εκτελεστεί. Επιπρόσθετα, το αριστερό και το δεξί κουμπί είναι προγραμματιζόμενα, μπορούν δηλαδή να χρησιμοποιηθούν ως είσοδοι μέσα στα προγράμματα και να καθορίσουν τις ενέργειες του ρομποτικού συστήματος, για παράδειγμα να μπορεί ο χρήστης να δώσει την εντολή «Όταν πατηθεί το αριστερό κουμπί απενεργοποίησε τον κινητήρα».

## Η οθόνη 5x5 LEDs (Φωτεινός Πίνακας/Light Matrix):

Στην επάνω πλευρά του Hub θα παρατηρήσει κανείς έναν φωτεινό πίνακα που αποτελείται από 25 λευκά LEDs. Κάθε ένα από τα LEDs μπορεί να προγραμματιστεί είτε ξεχωριστά, είτε και μέσα σε ένα σύνολο, προκειμένου να σχηματίσουν κάτι το οποίο θα εμφανιστεί στην οθόνη. Η φωτεινότητα του κάθε LED μπορεί να αυξομειωθεί από τον χρήστη ανά 10 μονάδες κάθε φορά, ξεκινώντας από το 0 και φτάνοντας μέχρι το 100, που είναι και η μέγιστη φωτεινότητα. Έτσι, όλα τα LEDs μαζί σχηματίζουν ένα φωτεινό πίνακα όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 5.3 Ο Φωτεινός Πίνακας / Light Matrix.

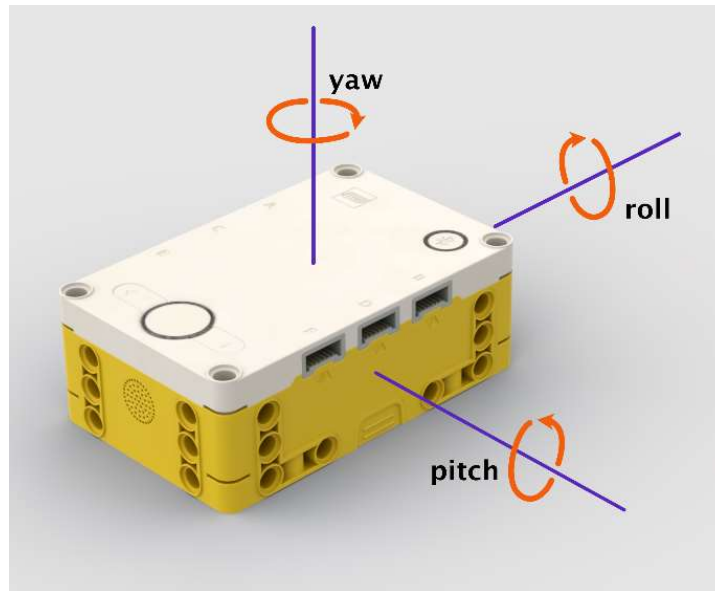
Απαραίτητη προϋπόθεση για να προγραμματίσει κανείς τον φωτεινό πίνακα είναι να κατανοήσει τον τρόπο με τον οποίο οργανώνεται. Παρότι ο πίνακας χωρίζεται σε γραμμές και στήλες δεν πρέπει να προσεγγίζεται σαν ένα σύστημα από γραμμές και στήλες, αλλά σαν ένα σύστημα αξόνων  $x$  και  $y$ , όπου ο άξονας  $x$  είναι ο κάθετος άξονας και  $y$  είναι ο οριζόντιος. Έτσι, για να προγραμματίσει κάποιος ένα LED αρκεί να υπολογίσει τη θέση του μέσα στο σύστημα αξόνων. Ο αριθμός της τιμής του  $x$  και του  $y$  αυξάνεται κάθε φορά κατά 1, ξεκινώντας από το LED πάνω αριστερά που αποτελεί το αρχικό σημείο του άξονα, δηλαδή το σημείο  $(0,0)$ . Ο άξονας  $x$  αυξάνεται όσο κινούμαστε κάθετα στο Hub, ενώ ο άξονας  $y$  όσο κινούμαστε οριζόντια. Επειδή το αρχικό LED είναι το σημείο  $(0,0)$ , η μέγιστη τιμή κάθε άξονα είναι ο αριθμός 4, παρότι κάθε άξονας αποτελείται από 5 LEDs. Έτσι, το τελευταίο LED του άξονα  $x$  βρίσκεται στη θέση  $(0,4)$  ενώ του άξονα  $y$  στη θέση  $(4,0)$ . Το τελευταίο LED κάτω δεξιά βρίσκεται στη θέση  $(4,4)$ .

Τέλος, μέσα στο Hub βρίσκονται ενσωματωμένα ένα ηχείο για αναπαραγωγή ήχων και ένας αισθητήρας κίνησης. Αξίζει να αναφερθούμε αναλυτικά σε αυτά, καθώς μπορούν να φανούν πολύ χρήσιμα στη δημιουργία προγραμμάτων.

Αναφορικά με το ηχείο, παρότι βρίσκεται ενσωματωμένο στο Hub, μπορεί να δει κανείς την έξοδο του στην μπροστινή πλευρά του Hub. Προγραμματίζοντας το ηχείο, ο χρήστης μπορεί να αναπαράγει διάφορες νότες. Κάθε νότα αντιστοιχεί σε έναν αριθμό, ο οποίος προέρχεται από την αρίθμηση του MIDI (Musical Instrument Digital Interface), ένα πρωτόκολλο που ακολουθούν οι ηλεκτρονικές συσκευές για να αναπαραστήσουν τη μουσική. Καλώντας δηλαδή κανείς την μέθοδο «beep» για το ηχείο «speaker», και δίνοντας ως παράμετρο τον αριθμό 60, στην πραγματικότητα ζητά από το ηχείο να παίζει την νότα Ντο/C (οι μέθοδοι και οι παράμετροι κάθε εξαρτήματος θα εξηγηθούν αναλυτικά σε παρακάτω κεφάλαιο). Σε αυτό το σημείο είναι πολύ σημαντικό να αναφερθεί ότι το ηχείο του Hub μπορεί να αναπαράγει μόνο νότες και ακολουθίες από νότες, ενώ οποιοδήποτε άλλο αρχείο ήχου θα αναπαράγεται μέσω της εφαρμογής του SPIKE, δηλαδή από το ηχείο του υπολογιστή. Επομένως, οι ήχοι που βρίσκονται στη βιβλιοθήκη της εφαρμογής του SPIKE δεν θα αναπαραχθούν από το ηχείο του Hub, αλλά από την εφαρμογή, όπως για παράδειγμα ο ήχος «Cat Sound».

Τέλος, το Hub διαθέτει ενσωματωμένο έναν γυροσκοπικό αισθητήρα και ένα επιταχυνσιόμετρο, τα οποία αναφέρονται μαζί ως ένας αισθητήρας κίνησης (Motion Sensor). Ο αισθητήρας αυτός αναγνωρίζει 6 άξονες (αποτελούμενος στην ουσία από ένα επιταχυνσιόμετρο 3 αξόνων και ένα γυροσκόπιο 3 αξόνων) και είναι πολύ σημαντικός στο να μας δίνει χρήσιμες πληροφορίες για τη θέση και τον προσανατολισμό του Hub, βοηθώντας πολύ στη δημιουργία ποικίλων προγραμμάτων. Για να περιγραφεί ο προσανατολισμός του ρομποτικού συστήματος χρησιμοποιούνται 3 όροι: Yaw, Pitch και Roll (η αναφορά στους συγκεκριμένους όρους θα γίνεται στα αγγλικά, καθώς η ακριβή αντιστοίχιση με ελληνικούς όρους που να τους περιγράφουν επακριβώς δεν είναι δυνατή). Ο καθένας από αυτούς τους όρους συνδέεται και με έναν από τους τρεις άξονες x, y και z. Ο όρος Yaw χρησιμοποιείται για να περιγράψει την περιστροφή γύρω από τον άξονα z, ο όρος Pitch αναφέρεται στην περιστροφή γύρω από τον άξονα y και ο όρος Roll σχετίζεται με την περιστροφή γύρω από τον άξονα x. Προσπαθώντας να εξηγήσουμε τους όρους με απλά λόγια θα λέγαμε πως ο όρος Yaw αφορά την περιστροφή του Hub προς τα δεξιά – αριστερά, ο όρος Pitch πάνω – κάτω και όρος Roll την περιστροφή του Hub προς την μια πλευρά ή προς την άλλη πλευρά.

Έτσι, κάνοντας χρήση του συγκεκριμένου αισθητήρα, το ρομποτικό σύστημα μπορεί να μας επιστρέψει ανά πάσα στιγμή ως τιμή τον προσανατολισμό του.



Εικόνα 5.4 Οι όροι Yaw, Pitch και Roll στην πράξη.

Ένα ακόμη πλεονέκτημα του γυροσκοπικού αισθητήρα είναι ότι έχει τη δυνατότητα να αναγνωρίσει συγκεκριμένες «χειρονομίες», δίνοντάς μας τη δυνατότητα να δημιουργήσουμε και να ελέγξουμε προγράμματα με αυτές. Οι χειρονομίες που αναγνωρίζει είναι το «κούνημα», το «χτύπημα», και η «ελεύθερη πτώση» του Hub (συναντώνται στις εντολές του προγραμματισμού ως Shaken, Tapped και Falling αντίστοιχα).

### **Αισθητήρες/Sensors:**

Οι αισθητήρες είναι ηλεκτρικά εξαρτήματα που λειτουργούν κυρίως ως είσοδοι στις συσκευές. Χρησιμοποιούνται με στόχο να μπορεί το ρομπότ να «αντιλαμβάνεται» το περιβάλλον στο οποίο βρίσκεται «διαβάζοντας» κάποια φυσικά μεγέθη, π.χ. τη φωτεινότητα του περιβάλλοντος, και στη συνέχεια μετατρέπουν την πληροφορία από το φυσικό περιβάλλον σε ηλεκτρικό σήμα. Έτσι, τα ρομποτικά συστήματα έχουν τη δυνατότητα να συλλέγουν δεδομένα και να προσαρμόζουν τις ενέργειες τους με βάση αυτά.

### **Αισθητήρας Χρώματος/Φωτός (Color Sensor):**

Ο Αισθητήρας Χρώματος/Φωτός έχει ουσιαστικά τρεις διαφορετικές λειτουργίες. Χρησιμοποιείται για να αναγνωρίσει χρώματα, την ένταση του ανακλώμενου φωτός και την ένταση του φωτός στο περιβάλλον. Επιπρόσθετα, χρησιμοποιείται και ως φακός για να εκπέμψει λευκό φως.



*Εικόνα 5.5 Ο αισθητήρας χρώματος.*

Όταν λειτουργεί ως αισθητήρας χρώματος μπορεί να εντοπίσει 8 χρώματα: άσπρο, μπλε, μαύρο, πράσινο, κίτρινο, γαλάζιο, μωβ/βιολετί και κανένα χρώμα. Σε μερικές περιπτώσεις είναι πιθανό να μην αναγνωρίσει ένα από τα προαναφερθέντα χρώματα, καθώς τα χρώματα αυτά αντιστοιχούν σε συγκεκριμένη τυπολογία της LEGO. Η βέλτιστη λειτουργία του πραγματοποιείται όταν το αντικείμενο το χρώμα του οποίου θέλουμε να αναγνωρίσουμε βρίσκεται σε απόσταση 16 εκατοστών. Το μέγεθος, το χρώμα και το υλικό της επιφάνειας μπορεί επίσης να επηρεάσει την λειτουργία του αισθητήρα.

Στην λειτουργία ανίχνευσης της αντανάκλασης φωτός, ο αισθητήρας λειτουργεί και ως πομπός που στέλνει μια δέσμη φωτός προς μια συγκεκριμένη κατεύθυνση, όπου το φως αντανακλάται σε κάποιο αντικείμενο ή επιφάνεια. Στη συνέχεια, ο αισθητήρας μετρά την τιμή της αντανάκλασης που επιστρέφει στον αισθητήρα η επιφάνεια ή το αντικείμενο στο οποίο αντανακλάται. Η τιμή που αποθηκεύεται εκφράζεται σε ποσοστό επί τοις εκατό, με εύρος από 0% (όπου δεν υπάρχει καθόλου αντανάκλαση) έως 100% (μέγιστη αντανάκλαση). Όπως και στη λειτουργία φωτός, έτσι και τώρα, η βέλτιστη απόδοσή του πραγματοποιείται στα 16 εκατοστά, με το μέγεθος, το χρώμα και το υλικό της επιφάνειας να μπορούν να επηρεάσουν την αποδοτικότητα του αισθητήρα.



Στην λειτουργία ανίχνευσης φωτός του περιβάλλοντος, ο αισθητήρας επιστρέφει ποσοστιαίες τιμές σε ένα εύρος από 0% - 100%. Όπως είναι φυσικό, όσο μικρότερο είναι το ποσοστό επί τοις εκατό, τόσο πιο σκοτεινός είναι ο χώρος που βρισκόμαστε, και όσο μεγαλύτερο είναι το ποσοστό τόσο πιο φωτεινός είναι.

Τέλος, όπως προαναφέρθηκε ο αισθητήρας μπορεί να χρησιμοποιηθεί και σαν φακός για να εκπέμψει ένα λευκό φως. Η λειτουργία αυτή δεν μπορεί να ενεργοποιηθεί όσο ο αισθητήρας χρησιμοποιείται ως αισθητήρας χρώματος ή αισθητήρας μέτρησης της έντασης του φωτός του περιβάλλοντος.

### **Αισθητήρας Απόστασης (Distance Sensor):**

Ο αισθητήρας αυτός χρησιμοποιείται για να μετρήσει την απόσταση του ρομποτικού μηχανισμού από ένα αντικείμενο ή μια επιφάνεια χρησιμοποιώντας υπερήχους. Από το ρομποτικό σύστημα, με τη βοήθεια του αισθητήρα, εκπέμπεται ένας υπέρηχος συχνότητας 100Hz και, εκτιμώντας τον χρόνο που χρειάστηκε να φτάσει σε μια επιφάνεια και να επιστρέψει, υπολογίζει την απόσταση από αυτή.



*Εικόνα 5.6 Ο αισθητήρας απόστασης.*

Το εύρος τιμών που ο αισθητήρας μπορεί να ανιχνεύσει κάποιο αντικείμενο ή μία επιφάνεια είναι από 5-200cm, με ακρίβεια +/-2cm. Η γωνία στην οποία μπορεί να ανιχνεύσει ο αισθητήρας είναι +/-35 μοίρες, με την τιμή αυτή όμως να μπορεί να διαφέρει ανάλογα με την απόσταση. Τιμές έξω από το εύρος ανίχνευσης, η γωνία που βρίσκεται το αντικείμενο, το μέγεθος και το υλικό που είναι φτιαγμένο το αντικείμενο μπορούν να επηρεάσουν την αποδοτικότητα του αισθητήρα. Για παράδειγμα, αν η επιφάνεια του αντικειμένου είναι πολύ μαλακή ο υπέρηχος δεν θα ανακλαστεί πίσω.

Επιπρόσθετα, ο συγκεκριμένος αισθητήρας διαθέτει 4 προγραμματιζόμενα LEDs στην μπροστινή πλευρά λευκού χρώματος το καθένα από τα οποία μπορεί να ενεργοποιηθεί ξεχωριστά από το χρήστη. Η ένταση του φωτός ορίζεται σε μια ποσοστιαία κλίμακα από 0% έως 100% και η αύξηση μπορεί να γίνει κατά 1%.

### **Αισθητήρας Αφής (Force Sensor):**

Ο αισθητήρας αφής είναι ικανός να ανιχνεύσει από ένα απλό πάτημα, μέχρι και την πίεση που ασκείται πάνω στον αισθητήρα.



*Εικόνα 5.7 Ο αισθητήρας αφής.*

Η συχνότητα απόκρισης του αισθητήρα ορίζεται στα 100Hz και έχει τρεις διαφορετικές λειτουργίες:

1. Λειτουργία αισθητήρα αφής/Touch Sensing: Σε αυτή την λειτουργία, η ζώνη ενεργοποίησης του είναι 0-2mm. Μέσω της εφαρμογής ορίζεται ένα όριο ενεργοποίησης με ακρίβεια 1mm +/-0.5mm. Ως δύναμη ενεργοποίησης ορίζεται το 0.5-1 Newton με απόκλιση +/-10%. Η τιμή που επιστρέφει ο αισθητήρας είναι δυαδικού τύπου, επιστρέφει δηλαδή δύο τιμές: την τιμή 1 όταν είναι ενεργοποιημένος/πατημένος, και την τιμή 0 όταν δεν έχει ενεργοποιηθεί/δεν είναι πατημένος.
2. Λειτουργία πατήματος: Σε αυτή τη λειτουργία, η ζώνη ενεργοποίησης είναι 0-2mm. Στην εφαρμογή ο χρήστης μπορεί να ορίσει ένα όριο ενεργοποίησης με ακρίβεια 1mm +/-0.5mm. Ως δύναμη ενεργοποίησης ορίζεται το 0.5-1 Newton με απόκλιση +/-10%. Στην συγκεκριμένη λειτουργία ο αισθητήρας επιστρέφει 4 τιμές από το 0 έως το 3. Η τιμή 0 αντιστοιχεί στην κατάσταση όπου δεν είναι πατημένος ο

αισθητήρας, δεν υπάρχει δηλαδή κάποιο πάτημα. Στην τιμή 1 αντιστοιχεί το μόνο πάτημα, ενώ στην τιμή 2 το διπλό πάτημα. Τέλος, στην τιμή 3 αντιστοιχεί η κατάσταση όπου ο αισθητήρας έχει πατηθεί και κρατιέται πατημένος για ένα χρονικό διάστημα.

3. Λειτουργία πίεσης: Στην τελευταία λειτουργία, η ζώνη ενεργοποίησης είναι 2-8mm, ενώ ως δύναμη ενεργοποίηση ορίζεται το εύρος 2.5-10 Newton. Σε αυτή την λειτουργία ο αισθητήρας επιστρέφει τον αριθμό της πίεσης που ασκείται σε αυτόν σε Newton με βήμα 0.1 Newton. Η μέγιστη λειτουργία που μπορεί να επιστρέψει είναι 10 Newton, ενώ η τιμή μπορεί να δέχεται απόκλιση +/-0.65 Newton.

### **Κινητήρες (Motors):**

Το συγκεκριμένο Kit διαθέτει τρεις κινητήρες, οι δύο από τους οποίους ονομάζονται μεσαίοι κινητήρες και ο ένας μεγάλος κινητήρας, η ονομασία των οποίων προκύπτει από το μέγεθος τους. Τόσο ο μικρός, όσο και οι μικροί κινητήρες, είναι σχεδιασμένοι με τρόπο ώστε να λειτουργούν τόσο ως κινητήρες όσο και ως αισθητήρες. Με τον ενσωματωμένο αισθητήρα περιστροφής που διαθέτουν επιτρέπουν στον χρήστη, με τη βοήθεια της εφαρμογής, να συλλέξει δεδομένα που αφορούν την ταχύτητα και την θέση του κινητήρα. Σε περίπτωση που η περιστροφή του κινητήρα δεν γίνεται μηχανικά, αλλά με το χέρι, ο αισθητήρας περιστροφής μπορεί να αντιληφθεί και την φορά κατά την οποία γίνεται η περιστροφή.



*Εικόνα 5.8 Ο μεγάλος και οι μικροί κινητήρες.*

Αναφορικά με τον αισθητήρα, ο αριθμός της ανάλυσής του είναι 360 μετρήσεις ανά περιστροφή, όπου σαν ανάλυση ορίζεται ο αριθμός των μετρήσεων που κάνει ο αισθητήρας για κάθε πλήρη περιστροφή του άξονά του, ενώ η ακρίβεια των μετρήσεων είναι περίπου +/-3 μοίρες. Η συχνότητα με την οποία ενημερώνεται ο αριθμός των μετρήσεων από τον αισθητήρα είναι 100Hz.

Τα βασικά χαρακτηριστικά των δύο κινητήρων παρουσιάζονται στον παρακάτω πίνακα:

	Μεγάλος Κινητήρας	Μεσαίος Κινητήρας
Εύρος τάσης λειτουργίας	Ελάχιστο: 5V Μέγιστο: 9V	Ελάχιστο: 5V Μέγιστο: 9V
Μέγιστη απόδοση: Ταχύτητα	135rpm με απόκλιση +/-15%	135rpm με απόκλιση +/-15%
Μέγιστη απόδοση: Ροπή	8Ncm	3.5Ncm
Μέγιστη απόδοση: Κατανάλωση ρεύματος	430mA με απόκλιση +/-15%	280mA με απόκλιση +/-15%

Πίνακας 5.1 : Σύγκριση των βασικών χαρακτηριστικών των κινητήρων.

### 5.3 Αποκτώντας Πρόσβαση στο λογισμικό του Spike Prime.

Απαραίτητη προϋπόθεση για να προγραμματίσει κάποιος τον εκάστοτε ρομποτικό μηχανισμό που θα δημιουργήσει χρησιμοποιώντας το LEGO Spike Prime είναι να διαθέτει το απαραίτητο λογισμικό, μέσω του οποίου θα γίνεται η επικοινωνία με το ρομπότ. Για τον σκοπό αυτό, το Spike Prime διαθέτει μια δικιά του εφαρμογή, η πρόσβαση στην οποία γίνεται με δύο τρόπους:

a) Είτε κατεβάζοντας δωρεάν την εφαρμογή από τον παρακάτω σύνδεσμο:

<https://education.lego.com/en-us/downloads/spike-app/software>.

b) Είτε χρησιμοποιώντας την εφαρμογή online στην διεύθυνση :

<https://spike.legoeducation.com/>

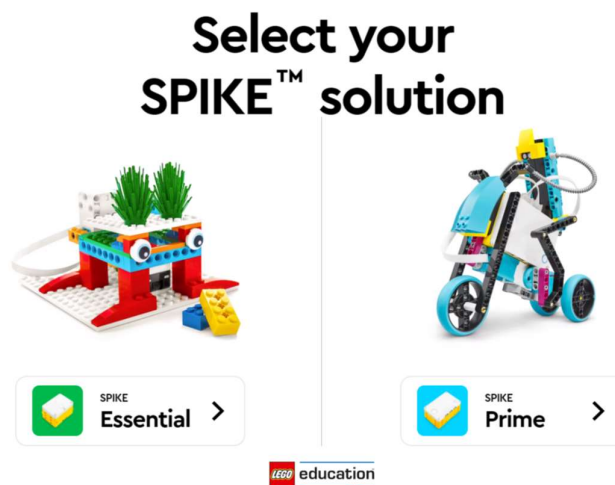
Οι δύο εκδοχές της εφαρμογής δεν παρουσιάζουν πολλές διαφορές, πέρα από το γεγονός ότι κατεβάζοντας την εφαρμογή στον υπολογιστή ο χρήστης έχει τη δυνατότητα να αποθηκεύει στην ενότητα «My Projects» τα προγράμματα που δημιουργεί. Σε αντίστοιχη περίπτωση, χρησιμοποιώντας την online εκδοχή ο χρήστης

θα πρέπει να αποθηκεύσει το πρόγραμμα τοπικά στον υπολογιστή του, καθώς δεν του δίνεται η δυνατότητα να αποθηκεύσει το πρόγραμμα μέσα στην εφαρμογή.

Η εφαρμογή υποστηρίζεται στα πιο διαδεδομένα λειτουργικά συστήματα (iOS, Microsoft Windows, macOS, Android και Chrome OS), απαιτεί τουλάχιστον 3GB διαθέσιμο αποθηκευτικό χώρο, καθώς επίσης και την τελευταία έκδοση του Bluetooth (Bluetooth 4.0). Δεν απαιτείται σύνδεση στο διαδίκτυο για την χρήση της εφαρμογής, παρόλα αυτά η σύνδεση είναι απαραίτητη για την λήψη ενημερώσεων και υλικού που προστίθεται στην εφαρμογή, για την πρόσβαση σε υλικό υποστήριξης για τους εκπαιδευτικούς και για τη χρήση συγκεκριμένων χαρακτηριστικών που προσφέρει η εφαρμογή, όπως η ζωντανή ενημέρωση και η παροχή δεδομένων για τον καιρό.

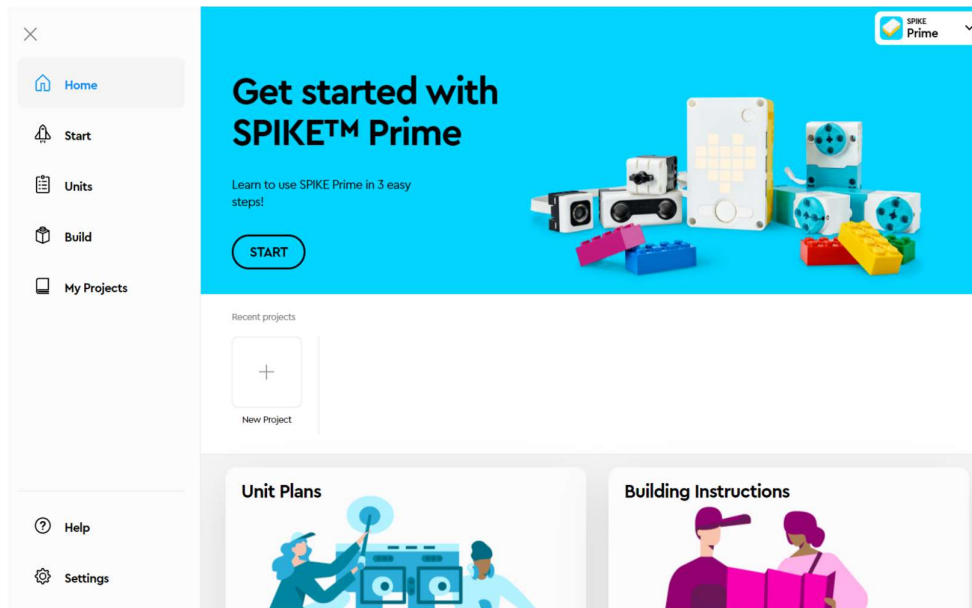
#### 5.4 Επισκόπηση της εφαρμογής του Spike Prime.

Ανοίγοντας την εφαρμογή του Spike θα συναντήσει κανείς το συγκεκριμένο μενού, όπου δίνεται η επιλογή να διαλέξουμε ανάμεσα στον προγραμματισμό του Spike Prime Kit και του Spike Essential Kit.



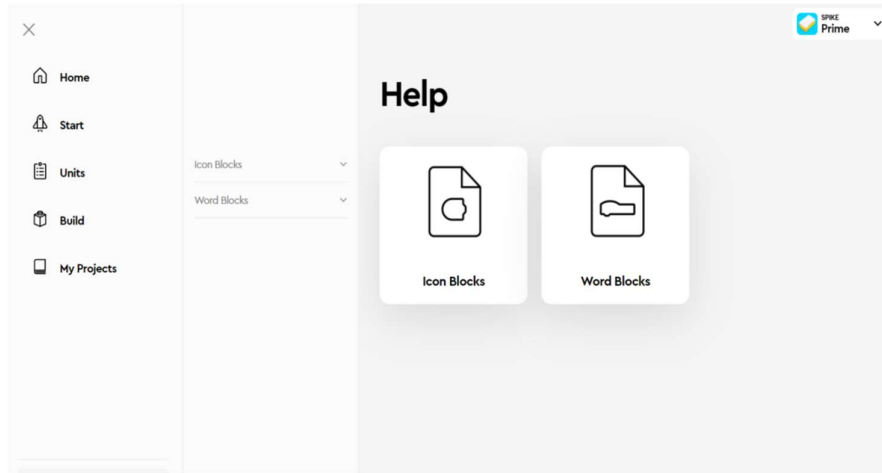
Εικόνα 5.9 Η αρχική σελίδα της εφαρμογής.

Επιλέγοντας κανείς να συνεχίσει με το Spike Prime μεταφέρεται στο αρχικό περιβάλλον της εφαρμογής.



Εικόνα 5.10 Το κεντρικό Menu της εφαρμογής.

Στο κυρίως menu, στην αριστερή πλευρά βρίσκονται όλες οι κατηγορίες με τις δυνατότητες που προσφέρονται από την εφαρμογή. Η κατηγορία «Start» αποτελείται από μικρά tutorials, μέσω των οποίων ο χρήστης μπορεί να γνωρίσει τα εξαρτήματα του Spike και να αποκτήσει μια πρώτη επαφή με αυτά. Οι κατηγορίες «Units» και «Build» αποτελούνται από διδακτικά μαθήματα και οδηγίες για τις κατασκευές αντίστοιχα. Συνολικά, το Spike Prime αποτελείται από 44 μαθήματα με τις αντίστοιχες κατηγορίες τους. Στο «My Projects» ο χρήστης θα βρει ένα αρχείο με όλα τα προγράμματα που έχει δημιουργήσει. Τέλος, πολύ σημαντική κατηγορία είναι η «Help», όπου επιλέγοντάς την ο χρήστης θα βρει πληροφορίες για τον Icon-Based και τον Block-Based προγραμματισμό όπου παρουσιάζονται αναλυτικά όλες οι εντολές που χρησιμοποιούνται στο Spike Prime.



Εικόνα 5.11 Το menu βοήθειας.

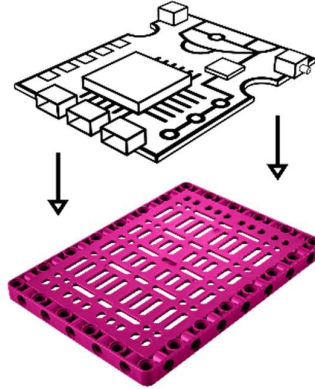
### 5.5 Το Lego Spike Prime Set Επέκτασης.

Για περισσότερες και πιο εξελιγμένες δημιουργίες στην εκπαίδευση STEAM παράλληλα με το βασικό Kit του Spike Prime, κυκλοφορεί και το set επέκτασης LEGO Education SPIKE Prime Expansion Set. Περιέχει περισσότερα από 600 στοιχεία LEGO, συμπεριλαμβανομένων μεγάλων τροχών, έναν επιπρόσθετο αισθητήρα χρώματος και καθώς και έναν μεγάλο κινητήρα.



Εικόνα 5.12 Το Set Επέκτασης.

Η ιδιαιτερότητα του Set Επέκτασης είναι ότι περιέχει ένα μοναδικό Maker Plate στο οποίο μπορεί κανείς να ενσωματώσει υπολογιστικές συσκευές μονής πλακέτας (SBC - Single-Board Computer).



*Εικόνα 5.13 Απεικόνιση του Maker Plate.*

Μαζί με το Spike Prime και την δωρεάν εφαρμογή του χρησιμοποιείται για εισάγει τους μαθητές στον κόσμο της ρομποτικής τόσο σε εκπαιδευτικά περιβάλλοντα όσο και σε διαγωνισμούς ρομποτικής.



## Κεφάλαιο 6 – Προγραμματισμός του Spike Prime

### Εισαγωγή.

Απαραίτητη προϋπόθεση για να μπορέσει το ρομπότ να διεκπεραιώσει κάποιες ενέργειες είναι να μπορέσουμε να του μεταδώσουμε τις ενέργειες αυτές σαν εντολές μέσω του προγραμματισμού. Όπως η επικοινωνία μεταξύ των ανθρώπων γίνεται με τη χρήση διάφορων κωδίκων (για παράδειγμα ο γλωσσικός κώδικας), με τον ίδιο τρόπο χρησιμοποιούμε διάφορους κώδικες για να μεταφέρουμε πληροφορίες στα ρομποτικά συστήματα. Γι' αυτό το λόγο και η συγγραφή προγραμμάτων συναντάται συχνά και ως συγγραφή «κώδικα». Στην πραγματικότητα, ο προγραμματισμός είναι ένας τρόπος επικοινωνίας με το ρομπότ, μέσω του οποίου του παρέχουμε μια σειρά από οδηγίες – βήματα – εντολές που πρέπει να εκτελέσει το ρομπότ προκειμένου να πετύχουμε το επιθυμητό αποτέλεσμα.

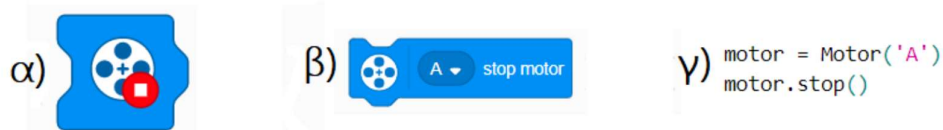
Γενικότερα, για να επιτευχθεί η επικοινωνία είναι απαραίτητο αυτός ο κώδικας να είναι κατανοητός τόσο από τον πομπό όσο και από τον δέκτη. Καθώς τα ρομπότ δεν μπορούν να καταλάβουν τις ανθρώπινες γλώσσες, χρειάζεται κανείς να μάθει μία από τις γλώσσες των ρομπότ, γνωστές και ως γλώσσες προγραμματισμού. Χρησιμοποιούμε διαφορετικά είδη γλωσσών προγραμματισμού, αλλά και ποικίλες γλώσσες, όπως είναι οι Python, Java, C++, Scratch κλπ. Κάποιες γλώσσες είναι σχεδιασμένες για να είναι πιο φιλικές και οικείες στον άνθρωπο, ενώ άλλες απαιτούν αρκετή μελέτη και εξάσκηση ώστε να μπορέσει κάποιος να τις χρησιμοποιήσει αποτελεσματικά.

Δημιουργώντας λοιπόν ένα πρόγραμμα, ή αλλιώς κώδικα, δημιουργούμε ουσιαστικά μια σειρά από οδηγίες που κατευθύνουν τις ενέργειες και την συμπεριφορά του ρομπότ, υλοποιούμε δηλαδή μια λογική σειρά από εντολές-οδηγίες, η οποία μπορεί να μας οδηγήσει σε ένα αποτέλεσμα. Είναι πολύ σημαντικό να θυμάται κανείς όταν γράφει ένα πρόγραμμα πως το ρομπότ εκτελεί ακριβώς τις οδηγίες και με τη σειρά που του έχουν δοθεί. Γι' αυτό το λόγο, για να είναι αποδοτικό ένα πρόγραμμα και να επιτευχθεί το επιθυμητό αποτέλεσμα, θα πρέπει οι οδηγίες που του δίνουμε να είναι στη σωστή σειρά και να είναι όσο το δυνατόν πιο σωστές, συγκεκριμένες και αναλυτικές. Συνεπώς, ενδεχόμενα λάθη στον κώδικα, ελλιπείς οδηγίες ή μια λανθασμένη σειρά μπορεί να οδηγήσει σε διαφορετικό αποτέλεσμα, καθώς το ρομπότ

δεν έχει την δυνατότητα να αναγνωρίσει τα «λάθη» μας ως λάθη, εκτελεί απλά τις οδηγίες που εμείς του έχουμε δώσει.

### 6.1 Τρόποι προγραμματισμού του Lego SPIKE Prime.

Ανοίγοντας κανείς την εφαρμογή του Lego Spike Prime συναντά 3 είδη γλωσσών με τις οποίες μπορεί να προγραμματίσει ένα ρομποτικό σύστημα: α) μια γλώσσα προγραμματισμού βασισμένη σε εικονίδια (Icon-Based) όπου κάθε εντολή παρουσιάζεται με τη μορφή μιας εικόνας. Β) μια γλώσσα προγραμματισμού με λεξικά blocks (Word- Blocks) στην οποία όλες οι εντολές παρουσιάζονται με την μορφή γραπτών εντολών γ) μια Text-Based γλώσσα προγραμματισμού, την Python.



Εικόνα 6.1 Η Εντολή "Απενεργοποίησε τον Κινητήρα Α" σε α) Icon Based Blocks β) Word Blocks γ) Python.

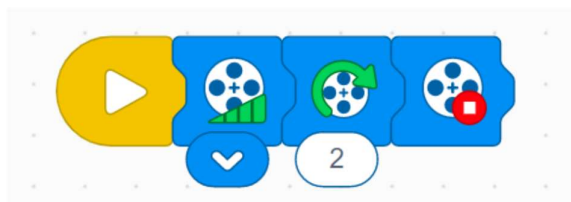
Είναι σημαντικό να αναφερθεί πως οι εντολές που λαμβάνει το ρομπότ δεν διαφοροποιούνται ανάλογα με τη γλώσσα που επιλέγει κάθε φορά ο χρήστης. Η ουσία της κάθε εντολής και το αποτέλεσμα που θα πραγματοποιήσει το ρομπότ παραμένει το ίδιο και στους τρεις τύπους προγραμματισμού. Αυτό που διαφοροποιείται είναι ο τρόπος διεπαφής με τον χρήστη.

Τέλος, όπως προαναφέρθηκε, ένα πρόγραμμα είναι μια σειρά από οδηγίες που πρέπει να εκτελέσει το ρομπότ. Γι' αυτό το λόγο αξίζει να αναφερθούμε στον τρόπο με τον οποίο εκτελείται το πρόγραμμά μας, στην ροή δηλαδή του προγράμματος, την σειρά με την οποία εκτελεί το ρομπότ αυτές τις οδηγίες. Στις γλώσσες εικονιδίων (Icon-Based Languages) το πρόγραμμα «στήνεται» οριζόντια και η ροή του προγράμματος ξεκινάει από τα αριστερά και πηγαίνει προς τα δεξιά, ενώ στις γλώσσες Γραπτών Εντολών (Word-Block Languages) και στις υπόλοιπες γλώσσες προγραμματισμού το πρόγραμμα «στήνεται» κάθετα και η ροή του προγράμματος ξεκινά από πάνω προς τα κάτω.

## 6.2 Προγραμματισμός με βάση εικονίδια (Icon-Based Programming).

Η γλώσσα προγραμματισμού βασισμένη σε εικονίδια (Icon-Based Language) είναι μια γλώσσα οπτικού προγραμματισμού. Στα περιβάλλοντα οπτικού προγραμματισμού, αντί λεξικών εντολών ο χρήστης συναντά εικονίδια ή γραφικά, τα οποία μπορούν να χρησιμοποιηθούν για την κατασκευή ενός προγράμματος. Στοιβάζοντας τις εντολές την μια μετά την άλλη σαν κομμάτια από puzzle, παράγουμε μια ακολουθία ενεργειών που πρέπει να ακολουθήσει το ρομπότ (Biggs 2003). Ένα πλεονέκτημα των γραφικών συστημάτων είναι η ευκολία τους στη χρήση τους καθώς κάθε ένα από τα block-εικονίδια αντιπροσωπεύει και μια εντολή για το ρομπότ καθιστώντας τον προγραμματισμό πιο απλό και ευκολονόητο (Bischoff et al., 2002).

Παρόλο που ο οπτικός προγραμματισμός έχει πολλά πλεονεκτήματα, δεν θα χρησιμοποιηθεί και δεν θα αναλυθεί περαιτέρω στα πλαίσια της παρούσας εργασίας, καθώς το ηλικιακό εύρος στο οποίο απευθύνονται οι πιο διαδεδομένες εφαρμογές εκμάθησης προγραμματισμού που χρησιμοποιούν οπτικό προγραμματισμό (π.χ. Scratch Jr.) αφορά κυρίως παιδιά 5-7 ετών, ενώ το LEGO SPIKE Prime έχει σχεδιαστεί για παιδιά ηλικιών 7-8 ετών. Παρόλα αυτά, ορθώς διαθέτει και την συγκεκριμένη επιλογή, καθώς μπορεί να χρησιμοποιηθεί για να εισάγει αποτελεσματικά τα παιδιά που δεν έχουν καμία πρότερη εμπειρία στον προγραμματισμό και την ρομποτική. Γι' αυτό και οι εντολές, και κατά συνέπεια οι δυνατότητες και τα προγράμματα που μπορεί να εκτελέσει κανείς χρησιμοποιώντας την Icon-Based επιλογή, περιορίζονται σε πολύ βασικές ενέργειες του ρομποτικού συστήματος.

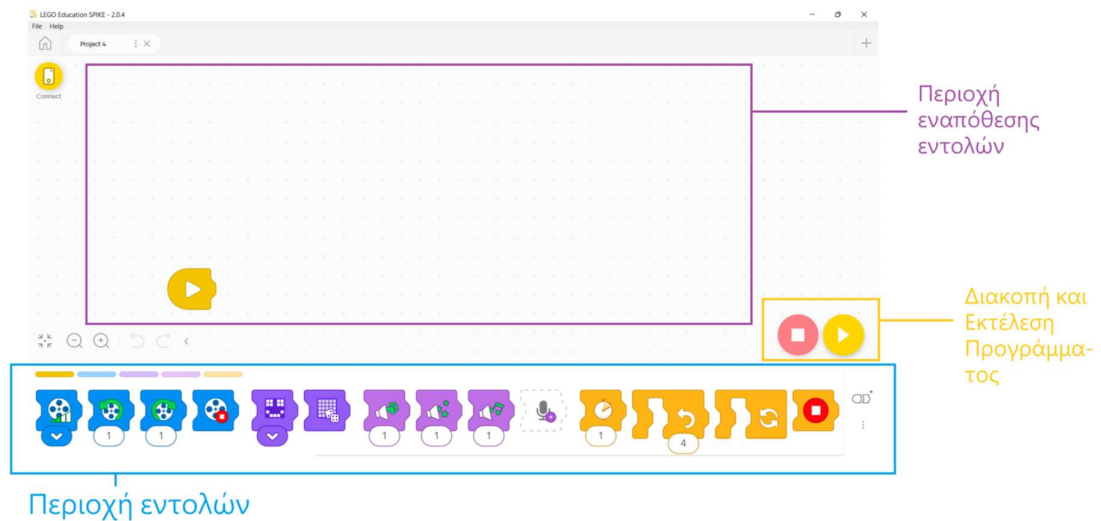


Εικόνα 6.2 Παράδειγμα δημιουργίας προγράμματος με Icon-Based Προγραμματισμό.

## 6.3 Το Icon-Based περιβάλλον προγραμματισμού στην εφαρμογή του Spike.

Το περιβάλλον προγραμματισμού βάσει εικονιδίων στην εφαρμογή του Spike Prime αποτελείται από δύο βασικές περιοχές: την «Περιοχή Εναπόθεσης» των

εικονιδίων, όπου ο χρήστης μπορεί να τοποθετεί τις εντολές και να δημιουργεί τα δικά του προγράμματα, και την «Περιοχή των Εντολών» από όπου ο χρήστης επιλέγει τις εντολές που χρειάζεται κάθε φορά, επιλέγοντάς τες και «τραβώντας» τες μέσα στην Περιοχή Εναπόθεσης. Για να διαγράψει ο χρήστης μια εντολή την αποσυνδέει από το πρόγραμμα και την τοποθετεί ξανά μέσα στην «Περιοχή των Εντολών». Στην παρακάτω εικόνα απεικονίζεται η ανάλυση του περιβάλλοντος:



Εικόνα 6.3 Το περιβάλλον Icon-Based Προγραμματισμού.

#### 6.4 Προγραμματισμός με Blocks (Block-Based Programming).

Ο προγραμματισμός με blocks είναι μια μέθοδος προγραμματισμού, όπου κάθε εντολή απεικονίζεται με την μορφή ενός «κουτιού» που ονομάζεται block. Οι εντολές στοιβάζονται η μία μετά την άλλη σαν κομμάτια puzzle και όλες μαζί παράγουν μια ακολουθία. Κάθε ακολουθία είναι δυνατό να συνδυαστεί με μια άλλη ακολουθία και να σχηματίσουν μαζί μια απλή ιεραρχία. Η σύνταξη ενός προγράμματος σε block-based προγράμματα βασίζεται στην λειτουργία «Μεταφοράς και Απόθεσης (Drag and Drop)», κατά την οποία ο χρήστης, για να δημιουργήσει μια ακολουθία, αρκεί να διαλέξει κάθε φορά τις εντολές που χρειάζεται και να τις σύρει από την περιοχή των εντολών στην περιοχή του προγραμματισμού. Στον block-based προγραμματισμό κάθε εντολή, ανάλογα με το σχήμα της, δίνει οπτικές πληροφορίες για το πώς μπορεί να χρησιμοποιηθεί και με ποιες εντολές μπορεί να συνδυαστεί, διασφαλίζοντας με αυτόν

τον τρόπο ότι αταίριαστες εντολές δεν μπορούν να συνδυαστούν, αποτρέποντας έτσι τα συντακτικά λάθη στο πρόγραμμα (Weintrop & Wilensky, 2017), ενώ ταυτόχρονα διατηρεί με αυτό τον τρόπο την πρακτικότητα στη συγγραφή των προγραμμάτων, καθώς οι εντολές συναρμολογούνται μία προς μία (Weintrop, 2019).

## 6.5 Τύποι των Blocks και Ορολογία.


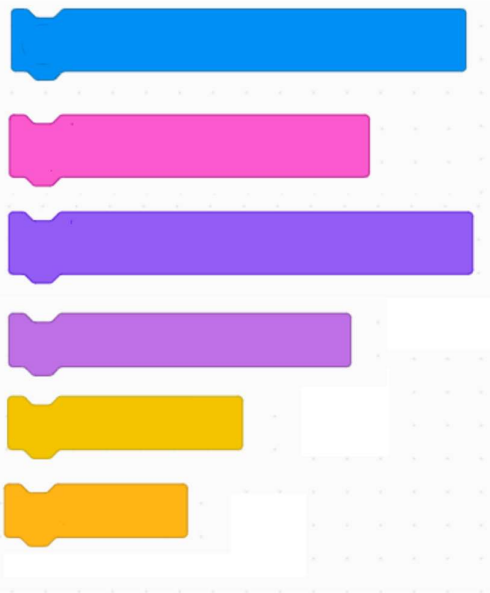
Η βασική λειτουργία του προγραμματισμού που βασίζεται σε blocks κειμένου έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι πολύ διαισθητική. Βασίζεται στην πιο διαδεδομένη εφαρμογή προγραμματισμού με blocks, το [Scratch 3.0](#). Τα προγράμματα αντιπροσωπεύονται από μια ακολουθία block προγραμματισμού, τα οποία εκτελούνται το ένα μετά το άλλο. Αυτή η ακολουθία ονομάζεται στοίβα ή block στοίβα στο Scratch.

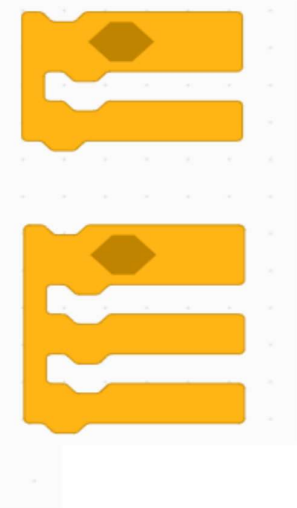

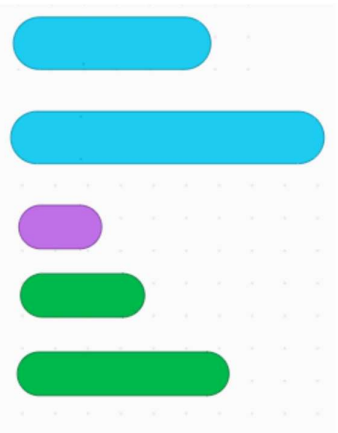
Όπως προαναφέρθηκε, τα Blocks είναι εντολές που έχουν τη μορφή puzzle και συνδέονται το ένα μετά το άλλο κάθετα, ενώνοντας τες όπως ακριβώς τα κομμάτια ενός puzzle. Η κάθετη αυτή συναρμολόγηση αντιστοιχεί και στην ροή του προγράμματος, η οποία ξεκινάει από πάνω προς τα κάτω. Όταν δύο εντολές “κολλήσουν” μεταξύ τους, ο χρήστης ειδοποιείται με έναν χαρακτηριστικό ήχο. Φυσικά, η σύνδεση μεταξύ των Blocks δεν είναι μόνιμη. Ο χρήστης μπορεί να την επεξεργαστεί μετακινώντας Blocks, να προσθέσει κάποιο καινούργιο ανάμεσα στα ήδη υπάρχοντα Blocks που απαρτίζουν το πρόγραμμα του, ακόμα και να διαγράψει κάποιο από το πρόγραμμα.


Για την δημιουργία ενός προγράμματος είναι πολύ σημαντικό ο χρήστης να γνωρίζει τους τύπους των Blocks, καθώς ο τύπος του κάθε Block θα καθορίσει στην συνέχεια τους επιτρεπτούς συνδυασμούς του με κάποιο άλλο. Στον προγραμματισμό με Blocks συναντάμε 6 διαφορετικούς τύπους από Block:

1. Hat Blocks
2. Stack Blocks
3. C Blocks
4. Boolean Blocks
5. Reporter Blocks.
6. Cap Blocks

Στον παρακάτω πίνακα ακολουθεί η ανάλυση του κάθε τύπου:

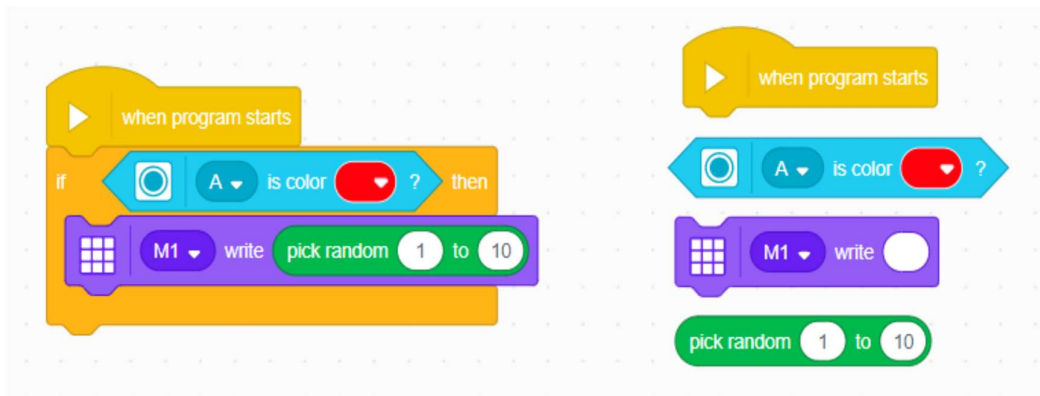
Hat Blocks / Block Καπέλο		<p>Χρησιμοποιούνται για να ξεκινήσει κάποιο πρόγραμμα όταν το συμβάν το οποίο αναγράφεται στο block είναι αληθές. Το σχήμα τους θυμίζει καπέλο και δείχνει πως δεν μπορεί κάποιος να προσθέσει εντολές στο πάνω μέρος τους, παρά μόνο στο κάτω μέρος τους, το οποίο θυμίζει κομμάτι από puzzle.</p>
Stack Blocks / Block Στοιβάδα		<p>Είναι τα βασικά blocks που αποτελούν και τις βασικές εντολές του προγράμματος. Μπορεί να τα βρει κάποιος σε όλες τις κατηγορίες. Έχουν μια εσοχή στο επάνω μέρος τους, κι ένα εξόγκωμα στο κάτω, που θυμίζουν κομμάτια από puzzle. Αυτό σημαίνει πως μπορούν να ενωθούν με κάποιο άλλο block είτε στο πάνω μέρος είτε στο κάτω, ή να μπουν μέσα σε κάποιο C Block.</p>

<p>C Blocks / Block σε σχήμα «C»</p>		<p>Σε αυτή την κατηγορία βρίσκονται τα blocks που θυμίζουν το αγγλικό γράμμα C. Κυρίως είναι Blocks που βοηθούν στο να ελέγξουμε την ροή του προγράμματος. Το ιδιαίτερο σχήμα τους επιτρέπει στα Stack Blocks να μπουκν ανάμεσά τους.</p>
<p>Boolean Blocks / Block δεδομένων Αληθείας</p>		<p>Τα συγκεκριμένα Blocks αντιστοιχούν σε μια τιμή που είναι είτε αληθής είτε ψευδής (boolean values). Το εξαγωνικό τους σχήμα τους επιτρέπει να τοποθετούνται μέσα σε blocks με κενά σημεία που έχουν το αντίστοιχο σχήμα. Συνήθως τοποθετούνται μέσα σε Control Blocks για να δημιουργήσουν σημεία «αποφάσεων» ή λογικών ελέγχων μέσα στο πρόγραμμα.</p>
<p>Reporter Blocks / Value Blocks / Blocks Τιμών</p>		<p>Είναι blocks που αντιπροσωπεύουν μια συγκεκριμένη τιμή, η οποία είναι είτε κείμενο (string), είτε αριθμός που εισάγεται από τον χρήστη, είτε τιμή από κάποιον αισθητήρα, είτε κάποια μεταβλητή. Το οβάλ τους σχήμα τους επιτρέπει να εισχωρήσουν μέσα σε blocks που έχουν θέσεις αντίστοιχου σχήματος.</p>

Cap Blocks / Blocks Καπάκια		Τα Cap Blocks χρησιμοποιούνται για να σταματήσουν τη ροή της στοιβάδας που είναι συνδεδεμένα. Είναι σχεδιασμένα με τέτοιο τρόπο ώστε να θυμίζουν το τελικό κομμάτι σε ένα puzzle (πρόγραμμα). Στο LEGO Spike Prime συναντάμε μόνο ένα τέτοιο block.
--------------------------------	---	---

Πίνακας 6.1 Η ορολογία των Block Εντολών.

Αξίζει να αναφερθεί πως, από τις 6 κατηγορίες, μόνο οι 4 είναι αυτόνομες, δηλαδή, τα Blocks των κατηγοριών μπορούν να συνδεθούν και να λειτουργήσουν στο πρόγραμμα χωρίς την υποστήριξη ενός άλλου Block. Αντίθετα, οι δύο κατηγορίες, δηλαδή τα Boolean Blocks και τα Reporter Blocks χρειάζονται οπωσδήποτε να συνδυαστούν με κάποια Blocks των υπολοίπων 4 κατηγοριών για να λειτουργήσουν στο πρόγραμμα.



Εικόνα 6.4 Παράδειγμα της μη-αυτονομίας των δύο κατηγοριών.

Στο παραπάνω παράδειγμα βλέπουμε τη μη-αυτονομία των δύο κατηγοριών. Στην αριστερή πλευρά, το Boolean Block και το Reporter Block έχουν συνδυαστεί με ένα C-Block και ένα Stack Block αντίστοιχα, και έχει δημιουργηθεί ένα εκτελέσιμο πρόγραμμα. Στην δεξιά πλευρά είναι εμφανές πως τα Blocks δεν έχουν ενωθεί μεταξύ



τους, καθώς τα σχήματά τους δεν επιτρέπουν την μεταξύ τους σύνδεση, και έτσι δεν έχει δημιουργηθεί μια στοιβάδα/πρόγραμμα που να μπορεί να εκτελεστεί από το ρομποτικό μας σύστημα.

## 6.6 Κατηγορίες των Blocks.


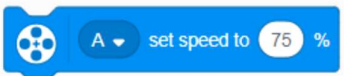
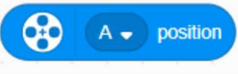

Στην έκδοση 2.0.4 όλα τα διαθέσιμα Blocks στο Spike Prime είναι κατηγοριοποιημένα σε χρωματικές κατηγορίες που αντιστοιχούν σε διαφορετικούς σκοπούς. Στο Spike Prime συναντάμε 10 κατηγορίες εντολών: 8 βασικές με 8 διαφορετικά χρώματα όπου περιέχουν τις βασικές εντολές που μπορεί να χρησιμοποιήσει κάποιος για να δημιουργήσει ένα πρόγραμμα και 2 πρόσθετες κατηγορίες, μια κατηγορία όπου ο χρήστης δημιουργεί τις δικές του μεταβλητές και μια ακόμη όπου δημιουργεί τις δικές του εντολές/συναρτήσεις.



Εικόνα 6.5 Οι 10 κατηγορίες εντολών του Spike Prime ανά χρώμα.

Μπλε Κατηγορία/Κατηγορία Κινητήρων (Motors): Η κατηγορία αυτή περιέχει εντολές για την λειτουργία μεμονωμένων κινητήρων. Ο χρήστης μπορεί με τις συγκεκριμένες εντολές να ξεκινήσει έναν κινητήρα, να τον σταματήσει, να ορίσει την ταχύτητά του, ακόμη και να ζητήσει πληροφορίες από τους κινητήρες, για παράδειγμα την ταχύτητα με την οποία έχει οριστεί να κινείται.

	<p>Ενεργοποιεί τον κινητήρα και τον κινεί αριστερόστροφα ή δεξιόστροφα για συγκεκριμένες περιστροφές, δευτερόλεπτα ή μοίρες.</p>
	<p>Δίνει εντολή στον κινητήρα να λειτουργήσει προς το συντομότερο μονοπάτι, δεξιόστροφα ή αριστερόστροφα προς την επιλεγμένη θέση (0-360 μοίρες).</p>
	<p>Ενεργοποιεί τον κινητήρα που έχουμε ορίσει με αριστερόστροφη ή δεξιόστροφη κατεύθυνση.</p>

	Σταματάει τον κινητήρα που έχουμε επιλέξει.
	Ορίζει μια συγκεκριμένη ταχύτητα σε έναν κινητήρα.
	Το block αναφέρει την τρέχουσα θέση ενός κινητήρα. Η τιμή που επιστρέφει είναι από 0 έως 359 μοίρες.
	Πρόκειται για ένα Report Block με το οποίο μπορεί κάποιος να ανακτήσει την τιμή της ταχύτητας ενός συγκεκριμένου κινητήρα.

Πίνακας 6.2 Οι βασικές εντολές της Μπλε Κατηγορίας.

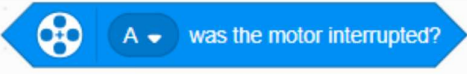

Οι εντολές όμως που μπορεί να χρησιμοποιήσει κανείς για να προγραμματίσει έναν κινητήρα δεν περιορίζονται σε αυτές τις 7 εντολές. Πηγαίνοντας κανείς στη γραμμή των εργαλείων και διαλέγοντας το κουμπί των Επεκτάσεων μπορεί να βρει επιπρόσθετες εντολές και λειτουργίες που μπορεί να χρησιμοποιήσει στα προγράμματα του.



Εικόνα 6.6 Οι επεκτάσεις της Εφαρμογής.

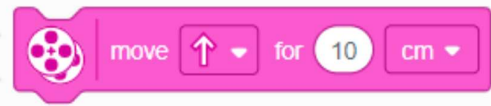
Έτσι, επιλέγοντας την κατηγορία «More Motors» μπορεί να προσθέσει ακόμα 11 εντολές διαθέσιμες για χρήση στα προγράμματά του.

	<p>Κινεί τον κινητήρα δεξιόστροφα ή αριστερόστροφα για έναν συγκεκριμένο αριθμό από περιστροφές, δευτερόλεπτα ή μοίρες με καθορισμένη ταχύτητα (ως ποσοστό ή στροφές ανά λεπτό).</p>
	<p>Ενεργοποιεί τον κινητήρα δεξιόστροφα ή αριστερόστροφα για πάντα με την καθορισμένη ταχύτητα.</p>
	<p>Κινεί τον κινητήρα σε μια σχετική θέση με μια καθορισμένη ταχύτητα.</p>
	<p>Ορίζει τη σχετική θέση του επιλεγμένου κινητήρα σε μια καθορισμένη τιμή. Το "0" επαναφέρει τη σχετική θέση.</p>
	<p>Συγκρατεί τον αριθμό των μοιρών που έχει γυρίσει ο κινητήρας από την αρχή του προγράμματος ή από τη στιγμή που επαναφέρθηκε στο 0.</p>
	<p>Κινεί τον κινητήρα με ένα καθορισμένο ποσοστό ισχύος για πάντα.</p>
	<p>Επιστρέφει το επίπεδο ισχύος που χρησιμοποιεί ένας καθορισμένος κινητήρας σε ποσοστό.</p>
	<p>Καθορίζει τον τρόπο με τον οποίο θα σταματήσει ο κινητήρας. Οι επιλογές είναι τρεις: το «Brake» αντιστοιχεί σε «Φρένο», το «Coast» όπου σταματά τον κινητήρα ομαλά και «Hold» όπου ο κινητήρας απενεργοποιείται και δεν μπορεί να περιστραφεί.</p>
	<p>Χρησιμοποιείται για να απενεργοποιήσετε τη λειτουργία «Stall Detection» σε block κινητήρα και κίνησης. Εάν η λειτουργία είναι απενεργοποιημένη, ο κινητήρας θα συνεχίζει ώστε να ολοκληρώσει μια κίνηση, ακόμη και όταν αποτραπεί σωματικά. Εάν είναι ενεργοποιημένη, σε αυτή την περίπτωση θα προχωρήσει στο επόμενο block.</p>

	Επιστρέφει «αληθές» εάν μια κίνηση ορισμένης διάρκειας που έχουμε ορίσει σε έναν κινητήρα δεν ολοκληρωθεί.
	Ορίζει την τιμή της επιτάχυνσης και της επιβράδυνσης ενός κινητήρα.

Πίνακας 6.3 Οι εντολές της Επέκτασης της Μπλε Κατηγορίας.

Ροζ Κατηγορία/Κατηγορία Κίνησης (Movement): Περιέχει εντολές οι οποίες χρησιμοποιούνται για την ταυτόχρονη λειτουργία δύο κινητήρων ίδιου τύπου. Ονομάζεται κατηγορία «Κίνησης» διότι συνήθως είναι εντολές που χρησιμοποιούνται για τον έλεγχο τροχοφόρων ρομπότ που χρησιμοποιούν δύο κινητήρες προκειμένου να κινηθούν. Οι εντολές που συναντά κανείς σε αυτή την κατηγορία είναι παρόμοιες με τις εντολές που βρίσκονται στην Μπλε Κατηγορία, δεν πρέπει όμως να συγχέονται καθώς αυτό θα μπορούσε να δημιουργήσει προβλήματα στο αποτέλεσμα του προγράμματος.

	Κινεί ταυτόχρονα τους κινητήρες, και κατ' επέκταση το όχημα προς μια κατεύθυνση για συγκεκριμένα εκατοστά, ίντσες, περιστροφές, μοίρες κινητήρων ή για συγκεκριμένα δευτερόλεπτα.
	Ενεργοποιεί τους κινητήρες με αποτέλεσμα την κίνηση του οχήματος.
	Ενεργοποιεί τους κινητήρες, προκειμένου το ρομπότ να κινηθεί προς τα μπροστά, αριστερά ή δεξιά για εκατοστά, ίντσες, περιστροφές, μοίρες ή δευτερόλεπτα με μια δεδομένη πηδαλιούχηση. Οι τιμές που αντιστοιχούν στην κάθε κίνηση είναι: αριστερά (-100), δεξιά (100) ή ευθεία (0).
	Ενεργοποιεί τους κινητήρες, προκειμένου το ρομπότ να κινηθεί με μια δεδομένη πηδαλιούχηση.
	Σταματά την κίνηση των κινητήρων.
	Ρυθμίζει τους κινητήρες να κινούνται με ένα προκαθορισμένο ποσοστό της μέγιστης ταχύτητάς τους.

	Ορίζει ποιοι δύο κινητήρες χρησιμοποιούνται για κίνηση του ρομποτικού συστήματος.
	Ορίζει σε πόσα εκατοστά ή ίντσες αντιστοιχεί μία περιστροφή κινητήρα.

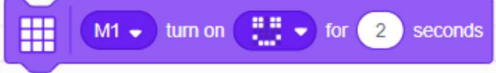
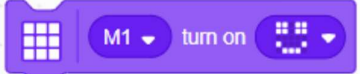

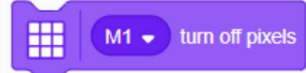
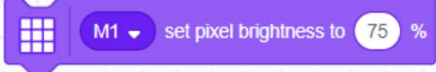
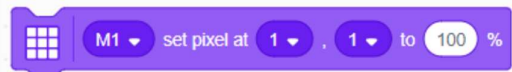
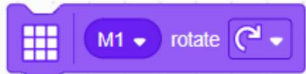
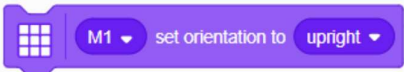
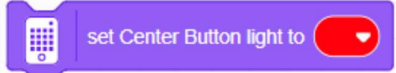
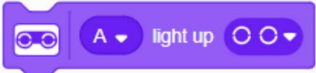
Πίνακας 6.4 Οι βασικές εντολές της Ροζ Κατηγορίας.

Χρησιμοποιώντας τις «Επεκτάσεις» μπορεί κανείς να βρει εννέα ακόμα εντολές που μπορούν να χρησιμοποιηθούν για τη δημιουργία προγραμμάτων.

	Μετακινεί το ρομποτικό σύστημα για συγκεκριμένα εκατοστά, ίντσες, δευτερόλεπτα, μοίρες ή περιστροφές με μια συγκεκριμένη ταχύτητα για κάθε κινητήρα.
	Μετακινεί το ρομποτικό σύστημα για πάντα με μια καθορισμένη ταχύτητα για κάθε κινητήρα.
	Κινεί το ρομποτικό σύστημα για ένα συγκεκριμένο διάστημα μια με συγκεκριμένη ταχύτητα και πηδαλιούχηση.
	Κινεί το ρομποτικό σύστημα για πάντα με μια συγκεκριμένη ταχύτητα και πηδαλιούχηση (steering).
	Ορίζει την ταχύτητα με την οποία θα ξεκινήσει κάθε κινητήρας.
	Μετακινεί το ρομποτικό σύστημα με μια καθορισμένη ισχύ για πάντα και με συγκεκριμένη πηδαλιούχηση: αριστερά (-100), δεξιά (100) ή ευθεία (0).
	Καθορίζει τον τρόπο με τον οποίο θα σταματήσουν οι κινητήρες όταν χρησιμοποιηθεί ένα stop block.
	Επιστρέφει «αληθές» εάν μια κίνηση ορισμένης διάρκειας που έχουμε ορίσει στους κινητήρες δεν ολοκληρωθεί.
	Ορίζει την τιμή της επιτάχυνσης και της επιβράδυνσης των κινητήρων.

Πίνακας 6.5 Οι εντολές της Επέκτασης της Μπλε Κατηγορίας

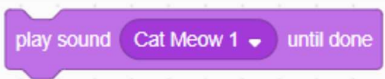
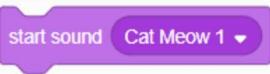
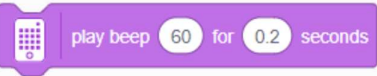
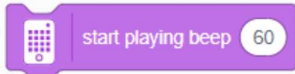

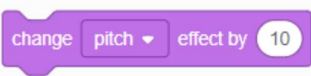
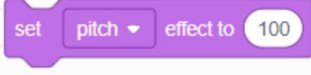
Μωβ Κατηγορία/Κατηγορία των Φώτων: Σε αυτή την κατηγορία βρίσκονται οι εντολές που χρησιμοποιούνται για να ελέγξει κανείς όλα τα φώτα που διαθέτει το Spike Prime, δηλαδή τον Φωτεινό Πίνακα/Light Matrix που βρίσκεται πάνω στο Hub, το φως που βρίσκεται στο κεντρικό κουμπί του Hub και τα φώτα του αισθητήρα απόστασης.

	<p>Δημιουργεί ένα μοτίβο-εικονίδιο στο Light Matrix για συγκεκριμένο χρονικό διάστημα. Επιλέγοντας κανείς το μοτίβο μπορεί να το επεξεργαστεί ενώ ταυτόχρονα θα συναντήσει κι έναν μοχλό με τον οποίο μπορεί να ορίσει την ένταση του φωτός στο μοτίβο.</p>
	<p>Ενεργοποιεί ένα μοτίβο-εικονίδιο στο Light Matrix. Το φωτεινό μοτίβο παραμένει ενεργό μέχρι το τέλος, ή μέχρι να δοθεί κάποια άλλη εντολή στο Light Matrix να κάνει κάτι άλλο.</p>
	<p>Απεικονίζει ένα μήνυμα (συμβολοσειρά) στην οθόνη του Light Matrix. Το μήνυμα δεν εμφανίζεται ολοκληρωμένο στην οθόνη, αλλά τα γράμμάτα του σχηματίζονται και κυλούν ένα-ένα στην οθόνη.</p>
	<p>Απενεργοποιεί όλα τα pixels που είναι ενεργοποιημένα στην οθόνη του Light Matrix.</p>
	<p>Ρυθμίζει την φωτεινότητα του Light Matrix.</p>
	<p>Ορίζει τη φωτεινότητα ενός συγκεκριμένου pixel στην οθόνη του Light Matrix.</p>
	<p>Περιστρέφει οτιδήποτε απεικονίζεται στο Light Matrix αριστερόστροφα ή δεξιόστροφα.</p>
	<p>Ορίζει τον προσανατολισμό για οτιδήποτε απεικονίζεται στο Light Matrix.</p>
	<p>Αλλάζει το χρώμα του φωτός του κεντρικού κουμπιού του Hub σε ένα από τα εξής χρώματα: φούξια, ροζ, μπλε, γαλάζιο, λαχανί, πράσινο, κίτρινο, πορτοκαλί, κόκκινο, άσπρο και μωβ.</p>
	<p>Αφορά τον αισθητήρα απόστασης, όπου κάθε ένα από τα φώτα του χωρίζεται σε δύο τμήματα. Με αυτή την</p>

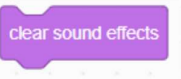
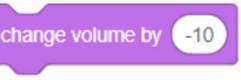
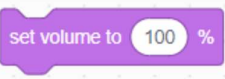

	εντολή ο χρήστης μπορεί να επιλέξει ποια από τα τμήματα των φώτων θέλει να ενεργοποιήσει.
--	---

Πίνακας 6.6 Οι εντολές της Μωβ κατηγορίας.

Λιλά Κατηγορία/Κατηγορία Ήχων: Περιέχει εντολές που αφορούν τους ήχους που δημιουργούνται από το Hub, την εφαρμογή του υπολογιστή και την επεξεργασία τους. Οι περισσότεροι επιλεγόμενοι ήχοι αναπαράγονται από τον υπολογιστή και όχι από το Hub, με εξαίρεση τον ήχο «μπιπ»/«beep». Επιπρόσθετα, χρησιμοποιώντας ο χρήστης αυτές τις εντολές μπορεί να προβεί σε επεξεργασία των ήχων επιταχύνοντάς τους, επιβραδύνοντάς τους, προσθέτοντας μια ηχώ σε κάποιον ήχο, μετατρέποντας τους σε ρομποτική φωνή, ή να τον επεξεργαστεί κάνοντας τον απαλότερο, δυνατότερο ή αντιστρέφοντας τον. Μπορεί επίσης να μειώσει τη διάρκεια ενός ήχου καθώς και να ηχογραφήσει έναν δικό του ήχο και να τον αναπαράγει.

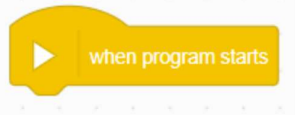

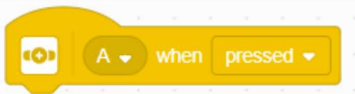
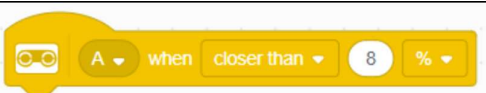

	Αναπαράγει έναν συγκεκριμένο ήχο από ένα παράθυρο επιλογών μέχρι το τέλος του ενώ ταυτόχρονα η ροή του προγράμματος σταματάει μέχρι να ολοκληρωθεί ο ήχος
	Χρησιμοποιείται για την αναπαραγωγή ενός ήχου χωρίς διακοπή του προγράμματος. Δηλαδή, το πρόγραμμα συνεχίζει στην επόμενη εντολή του προγράμματος χωρίς να περιμένει να ολοκληρωθεί ο ήχος.
	Αναπαράγει έναν «μπιπ» ήχο, μια συγκεκριμένη νότα, για συγκεκριμένα δευτερόλεπτα.
	Αναπαράγει έναν «μπιπ» ήχο μέχρι να δοθεί μια εντολή στο πρόγραμμα που θα τον διακόψει.
	Όταν η ροή του προγράμματος φτάσει σε αυτήν την εντολή τότε σταματάει όλους τους ήχους που αναπαράγονται εκείνη τη στιγμή.
	Αλλάζει το τονικό ύψος των ήχων ή το pan right/left εφέ κατά έναν συγκεκριμένο αριθμό.
	Ορίζει το τονικό ύψος των ήχων ή το pan right/left εφέ σε ένα προκαθορισμένο ποσό της μέγιστης τιμής του.



	Καθαρίζει όλα τα εφέ και επαναφέρει το τονικό ύψος στη φυσιολογική του τιμή.
	Αλλάζει την ένταση του ήχου που αναπαράγεται αυτήν τη στιγμή κατά μια συγκεκριμένη τιμή από την ένταση στην οποία βρίσκεται αυτή τη στιγμή.
	Ορίζει την ένταση του ήχου σε ένα προκαθορισμένο ποσοστό.
	Αναφέρει την τρέχουσα ένταση του ήχου.

Πίνακας 6.7 Οι εντολές της Λιλά Κατηγορίας.

Κίτρινη Κατηγορία/Κατηγορία των Γεγονότων: Οι εντολές που βρίσκονται σε αυτή την περιοχή ορίζουν το τι θα συμβεί στο πρόγραμμα όταν συμβεί ένα συγκεκριμένο γεγονός. Αυτή η κατηγορία αποτελείται κυρίως από block εκκίνησης (Hat Blocks). Τέλος, σε αυτή την κατηγορία μπορεί να βρει κανείς τις εντολές μετάδοσης και λήψης «μηνυμάτων».


	Εκτελεί όλα τα blocks που βρίσκονται συνδεδεμένα από κάτω του, όταν ο χρήστης επιλέξει το εικονίδιο «εκκίνησης».
	Όταν ο αισθητήρας χρώματος ανιχνεύσει το χρώμα που έχει οριστεί, τότε αναπαράγονται όλα τα blocks που είναι τοποθετημένα σε αυτό το block.
	Η εντολή αυτή σχετίζεται με τον αισθητήρα αφής. Σε περίπτωση που αυτός πατηθεί ή πατηθεί δυνατά ή απελευθερωθεί ή ανιχνευθεί οποιαδήποτε αλλαγή στην πίεση του αισθητήρα, τότε ξεκινά η αναπαραγωγή των συνδεδεμένων blocks.
	Αναπαράγει όλα τα blocks που βρίσκονται τοποθετημένα σε αυτή, όταν ο αισθητήρας απόστασης ανιχνεύσει ένα αντικείμενο σε απόσταση μικρότερη, μεγαλύτερη ή ίση με αυτή που έχει ορίσει ο χρήστης. Η απόσταση μπορεί να οριστεί σε ποσοστό επί τοις εκατό, σε εκατοστά ή ίντσες.
	Η αναπαραγωγή της στοιβάδας ξεκινά όταν το Hub αποκτήσει μια κλίση προς τα δεξιά, αριστερά, μπροστά, πίσω ή προς οποιαδήποτε κατεύθυνση, είτε παραμένει ακίνητο.



	<p>Αναπαράγει όλα τα blocks που συνδέονται με αυτό όταν το Hub αποκτά έναν συγκεκριμένο προσανατολισμό (εμπρός, πίσω, πάνω, κάτω, αριστερή πλευρά ή δεξιά).</p>
	<p>Αναπαράγει όλα τα συνημμένα Blocks όταν το Hub κουνιέται, χτυπιέται ή πέφτει.</p>
	<p>Αναπαράγει όλα τα blocks όταν το αριστερό ή το δεξί κουμπί είναι είτε πατημένα είτε απελευθερωθούν από κάποιο πάτημα.</p>
	<p>Όλα τα blocks που είναι συνδεδεμένα σε αυτό αναπαράγονται όταν το χρονόμετρο ή η ηχηρότητα υπερβεί την καθορισμένη τιμή.</p>
	<p>Ο χρήστης ορίζει μια συνθήκη μέσα στην εξαγωνική περιοχή και όταν αυτή αρχίσει να ισχύει (γίνει αληθής) τότε αναπαράγονται όλα τα blocks που έχει τοποθετήσει κάποιος σε αυτό το block.</p>
	<p>Όλα τα blocks που επισυνάπτονται σε αυτό αναπαράγονται όταν ένα συγκεκριμένο μήνυμα μεταδίδεται με τις εντολές broadcast [message1], ή broadcast [message1] and wait.</p>
	<p>Μεταδίδει ένα συγκεκριμένο μήνυμα και μεταφέρει τη ροή του προγράμματος σε κάποιο άλλο σημείο, κάποια άλλη στοιβάδα, ή στοιβάδες που ξεκινάνε με την εντολή When I receive [message1].</p>
	<p>Η λειτουργία αυτής της εντολής είναι παρόμοια με την λειτουργία της προηγούμενης. Η διαφορά τους είναι πως χρησιμοποιώντας αυτή την εντολή μεταδίδεται το μήνυμα και το πρόγραμμα περιμένει μέχρι οι εντολές των στοιβάδων που ξεκινούν με το block When I receive [message1] ολοκληρωθούν προτού προχωρήσει στην επόμενη εντολή.</p>

Πίνακας 6.8 Οι εντολές της Κίτρινης Κατηγορίας.

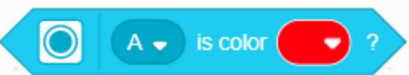



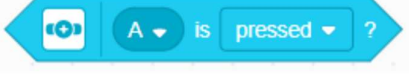

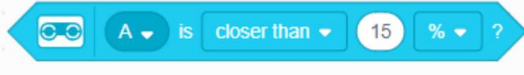
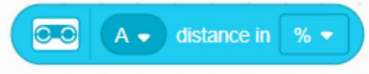
Πορτοκαλί Κατηγορία/Κατηγορία Ελέγχου: Η πορτοκαλί κατηγορία περιέχει blocks που σχετίζονται με τον έλεγχο της ροής του προγράμματος. Μπορεί κανείς σε αυτή την κατηγορία να βρεις blocks που τον βοηθούν, για παράδειγμα, να καθυστερήσει το πρόγραμμα, να εισάγει διακλαδώσεις ή και επαναλήψεις.

	<p>Σταματάει τη ροή του προγράμματος για συγκεκριμένο χρόνο. Μπορεί να λάβει ως τιμές τόσο ακέραιους, όσο και δεκαδικούς αριθμούς.</p>
	<p>Όλα τα blocks που βρίσκονται μέσα σε αυτό το block θα επαναληφθούν τόσες φορές όσες ορίσει ο χρήστης.</p>
	<p>Επαναλαμβάνει τα blocks που βρίσκονται μέσα του για πάντα, για όλη τη διάρκεια του προγράμματος</p>
	<p>Είναι η πρώτη εντολή διακλάδωσης που συναντά κάποιος. Ελέγχει αν μια συγκεκριμένη συνθήκη (αυτή που έχει ορίσει κάποιος μέσα στην εξαγωγική περιοχή) είναι αληθής. Αν είναι αληθής θα εκτελέσει τις εντολές που βρίσκονται μέσα του.</p>
	<p>Είναι η δεύτερη εντολή διακλάδωσης και ελέγχει την τιμή μιας συνθήκης. Αν είναι αληθής, εκτελεί τις εντολές που βρίσκονται μέσα στην if &lt;&gt; then περιοχή (πρώτο κενό). Αν δεν ισχύει η συνθήκη, είναι ψευδής, και θα εκτελέσει τις εντολές που βρίσκονται μέσα στην else περιοχή (δεύτερο κενό).</p>
	<p>Σταματάει την στοιβάδα που βρίσκεται μέχρι να γίνει αληθής η συνθήκη που έχει ορίσει κάποιος μέσα στην εξαγωγική περιοχή.</p>
	<p>Όλες οι εντολές που βρίσκονται μέσα στο συγκεκριμένο block θα επαναλαμβάνονται μέχρι η συνθήκη που βρίσκεται στην εξαγωγική περιοχή να γίνει αληθής. Ουσιαστικά, τα blocks επαναλαμβάνονται όσο δεν ισχύει η συνθήκη.</p>
	<p>Όταν η ροή της συγκεκριμένης στοιβάδας φτάσει σε αυτή την εντολή σταματάει τη ροή όλων των υπολοίπων στοιβάδων (stacks), ενώ η ίδια συνεχίζει να εκτελείται.</p>
	<p>Το μοναδικό Cap Block που έχει το Spike Prime. Χρησιμοποιείται για να σταματήσει όλες τις στοιβάδες (stacks) που τρέχουν εκείνη τη στιγμή ή την ροή της στοιβάδας που βρίσκεται ή μπορεί να</p>

	σταματήσει τη ροή και να πραγματοποιήσει έξοδο από το πρόγραμμα.
--	--

Πίνακας 6.9 Οι εντολές της Πορτοκαλί Κατηγορίας.

Τιρκουάζ Κατηγορία/Κατηγορία Αισθητήρων: Η κατηγορία αυτή περιέχει εντολές που προσφέρουν άμεση πρόσβαση στις τιμές των αισθητήρων. Σε αυτή την κατηγορία συναντάμε κυρίως Boolean Blocks και Reporter Blocks, που σημαίνει ότι αυτές οι εντολές δεν μπορούν να σταθούν «μόνες» τους, και για να χρησιμοποιηθούν θα πρέπει να συνδυαστούν με κάποια άλλη εντολή (π.χ. με κάποιο C block). Αποτελείται από 15 blocks από τα οποία τα 7 είναι Boolean Blocks, τα 2 Stack Blocks και τα υπόλοιπα 6 είναι Reporter Blocks.

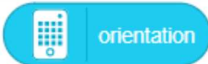

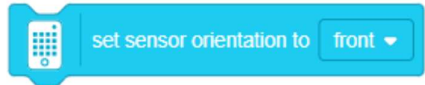
	Επιστρέφει την τιμή «Αληθές» όταν το χρώμα που ανιχνεύει ο αισθητήρας χρώματος είναι το ίδιο με αυτό που έχει ορίσει ο χρήστης.
	Συγκρατεί την τιμή, το χρώμα δηλαδή, που ανιχνεύεται από τον αισθητήρα χρώματος.
	Επιστρέφει την τιμή «αληθές» όταν η αντανάκλαση του φωτός που φτάνει στον αισθητήρα χρώματος είναι μεγαλύτερη, μικρότερη ή ίση από ένα αριθμητικό ποσοστό επί τοις εκατό που έχει οριστεί από τον χρήστη.
	Συγκρατεί την τιμή της αντανάκλασης του φωτός που φτάνει στον αισθητήρα χρώματος.
	Επιστρέφει την τιμή «αληθές» όταν ο αισθητήρας αφής είναι πατημένος (1-5 Newtons), είναι πολύ πατημένος (> 5 Newtons), απεθεί (0 Newtons).
	Αναφέρει την πίεση που ασκείται στον αισθητήρα αφής σε Newtons (2-10 Newtons) ή ως ποσοστό.
	Επιστρέφει την τιμή «αληθές», όταν ο αισθητήρας απόστασης ανιχνεύσει κάτι το οποίο βρίσκεται σε απόσταση μεγαλύτερη, μικρότερη ή ίση με μια προκαθορισμένη απόσταση. Η απόσταση μπορεί να οριστεί σε ποσοστό επί τοις εκατό, εκατοστά ή ίντσες.
	Αποθηκεύει την απόσταση που ανιχνεύεται από τον αισθητήρα απόστασης. Η απόσταση μπορεί να

	μετρηθεί σε ποσοστό επί τοις εκατό, εκατοστά ή ίντσες.
	Επιστρέφει «αληθές» όταν το Hub έχει μια κλίση προς μια συγκεκριμένη κατεύθυνση.
	Επιστρέφει «αληθές» όταν το Hub έχει έναν συγκεκριμένο προσανατολισμό που ορίζεται από τον χρήστη.
	Επιστρέφει «αληθές» όταν το Hub κουνιέται, χτυπιέται ή πέφτει.
	Συγκρατεί και αναφέρει την τιμή του yaw, pitch ή του roll.
	Ορίζει την τιμή του yaw σε μηδέν. Η προκαθορισμένη τιμή του yaw ορίζεται σε 0 στην κατεύθυνση στην οποία θα κοιτά το Hub όταν το πρόγραμμα ξεκινήσει.
	Επιστρέφει την τιμή της ηχηρότητας.
	Επιστρέφει «αληθές» αν το αριστερό ή το δεξί κουμπί πατηθεί ή απελευθερωθεί.
	Αναφέρει τον χρόνο από την έναρξη του προγράμματος σε δευτερόλεπτα.
	Επαναφέρει το χρονόμετρο.

Πίνακας 6.10 Οι εντολές της Τυκονάζ Κατηγορίας.


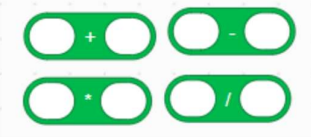


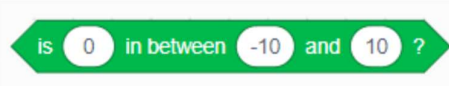
Επιλέγοντας κανείς να χρησιμοποιήσει τις Επεκτάσεις θα βρει 6 επιπλέον εντολές που μπορεί να χρησιμοποιήσει σε αυτή την κατηγορία:


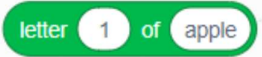
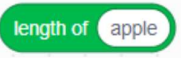




	Επιστρέφει την τιμή του ακατέργαστου κόκκινου, πράσινου ή μπλε χρώματος (0-255).
	Επιστρέφει την επιτάχυνση του Hub στον άξονα X, Y ή Z.
	Επιστρέφει τη γωνιακή ταχύτητα του Hub (ονομάζεται και ταχύτητα γυροσκοπίου) στον άξονα X, Y ή Z.

	<p>Επιστρέφει τον τρέχοντα προσανατολισμό του Hub (μπροστά, πίσω, πάνω, κάτω, αριστερή ή δεξιά πλευρά).</p>
	<p>Επιστρέφει την τρέχουσα χειρονομία του Hub (ανακίνηση, πάτημα ή πτώση).</p>
	<p>Ορίζει τον προσανατολισμό του γυροσκοπικού αισθητήρα 6 αξόνων σε μπροστά, πίσω, πάνω, κάτω, αριστερά πλευρά, ή δεξιά πλευρά.</p>

Πίνακας 6.11 Οι εντολές της Επέκτασης της Τικουάζ Κατηγορίας.

Πράσινη Κατηγορία/Κατηγορία Τελεστών: Σε αυτή την κατηγορία θα βρει κανείς blocks που αφορούν κυρίως μαθηματικούς υπολογισμούς και πράξεις. Σε αυτή την κατηγορία βρίσκονται τα blocks που βοηθούν τους χρήστες να πραγματοποιήσουν συγκρίσεις μεταξύ τιμών, εντολές που αφορούν πίνακες, εντολές boolean, γεννήτρια τυχαίων αριθμών κ.α.

	<p>Επιλέγει έναν τυχαίο αριθμό ανάμεσα στο εύρος που έχει ορίσει ο χρήστης, συμπεριλαμβανομένων και των άκρων. Ο αριθμός αυτός μπορεί να είναι είτε ακέραιος, είτε δεκαδικός. Αν τα άκρα του εύρους είναι ακέραιοι αριθμοί τότε και ο τυχαίος αριθμός θα είναι ακέραιος. Για να επιλέξει έναν τυχαίο δεκαδικό αριθμό θα πρέπει τα άκρα του εύρους να είναι δεκαδικοί αριθμοί επίσης.</p>
	<p>Τα συγκεκριμένα blocks χρησιμοποιούνται για πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση δύο τιμών, και επιστρέφουν το αποτέλεσμα.</p>
	<p>Χρησιμοποιούνται για σύγκριση μεταξύ δύο αριθμών. Επιστρέφουν την τιμή «αληθές» αν ισχύει το αποτέλεσμα της σύγκρισης.</p>
	<p>Οι εντολές αυτές ενώνουν δύο boolean blocks με σύζευξη «και» ή διάζευξης «ή». Η άρνηση «not» αντιστρέφει την τιμή αλήθειας της boolean πρότασης που εισάγουμε.</p>
	<p>Ελέγχει εάν η τιμή που ορίζει ο χρήστης βρίσκεται ανάμεσα σε ένα εύρος που ορίζεται επίσης από τον χρήστη. Μέσα σε αυτό το εύρος συμπεριλαμβάνονται και τα άκρα του.</p>

	Χρησιμοποιείται για να ενώσει δύο συμβολοσειρές σε μια και να επιστρέψει το αποτέλεσμα. Στην συγκεκριμένη περίπτωση το αποτέλεσμα θα ήταν «applebanana».
	Επιστρέφει τον χαρακτήρα της συμβολοσειράς που βρίσκεται στην αριθμητική θέση που έχει ορίσει ο χρήστης. Σε αυτό το παράδειγμα θα επέστρεφε το γράμμα που βρίσκεται στη θέση 1, δηλαδή το «a».
	Επιστρέφει τον αριθμό των χαρακτήρων από τους οποίους αποτελείται μια συμβολοσειρά. Για την λέξη apple θα επέστρεφε τον αριθμό «5».
	Επιστρέφει την τιμή «αληθές» αν ένας συγκεκριμένος χαρακτήρας περιέχεται σε μια συμβολοσειρά.
	Επιστρέφει το υπόλοιπο της διαίρεσης ανάμεσα σε δύο τιμές που ορίζει ο χρήστης, με διαιρετέο την πρώτη τιμή και διαιρέτη τη δεύτερη.
	Στρογγυλοποιεί έναν αριθμό που δίνεται από τον χρήστη στον πιο κοντινό ακέραιο αριθμό (από 0.5 έως 0.9 ο αριθμός στρογγυλοποιείται στον αμέσως επόμενο ακέραιο).
	Με αυτή την εντολή ο χρήστης μπορεί να επιλέξει ανάμεσα σε μια πληθώρα μαθηματικών συναρτήσεων που μπορεί να εκτελέσει σε έναν αριθμό, όπως για παράδειγμα, να ζητήσει την τετραγωνική του ρίζα. Αφού εκτελεστεί η συνάρτηση η εντολή επιστρέφει το αποτέλεσμα.





Πίνακας 6.10 Οι εντολές της Επέκτασης της Τυρκουάζ Κατηγορίας.

**Κατηγορία Μεταβλητών και Λιστών:** Σε αυτή την κατηγορία μπορεί κάποιος να δημιουργήσει τις δικές του μεταβλητές ή τις δικές του λίστες. Στον προγραμματισμό τόσο οι μεταβλητές όσο και οι λίστες είναι πολύ σημαντικά στοιχεία.

Μια μεταβλητή είναι ένας «περιέκτης» που μπορεί να αποθηκεύσει μια πληροφορία. Η πληροφορία αυτή μπορεί να αλλάξει και μπορεί να είναι είτε ένα κείμενο, είτε αριθμός ή και λίστες αριθμών και κειμένων. Οι μεταβλητές μας βοηθούν να κάνουμε πιο αποδοτικά τα προγράμματα μας καθώς η τιμή που λαμβάνουν είναι επεξεργάσιμη, και μπορούμε να ορίσουμε το πρόγραμμα μας να συμπεριφέρεται διαφορετικά ανάλογα με την τιμή της μεταβλητής. Επιπρόσθετα μπορούν να

λειτουργήσουν σαν ετικέτες πληροφοριών που λείπουν. Με αυτό τον τρόπο μπορούμε να συνεχίσουμε να δουλεύουμε στο πρόγραμμα μας, και οι πληροφορίες που λείπουν, να συμπληρωθούν αργότερα είτε από εμάς τους ίδιους, είτε από κάποιον άλλον.



Πηγαίνοντας κανείς σε αυτή την κατηγορία παρατηρεί ότι στην αρχή είναι άδεια από εντολές. Όταν όμως ο χρήστης δημιουργήσει μια νέα μεταβλητή ή μια νέα λίστα τότε, εμφανίζονται οι εντολές που σχετίζονται με τις μεταβλητές ή τις λίστες αντίστοιχα. Έτσι για παράδειγμα, δημιουργώντας μια μεταβλητή με όνομα “distance” εμφανίζονται οι ακόλουθες εντολές:

	Χρησιμοποιείται για τη δημιουργία νέων μεταβλητών.
	Επιστρέφει την τιμή που είναι αποθηκευμένη στην μεταβλητή που ονομάζεται “distance”.
	Χρησιμοποιείται για να δώσουμε σε μια μεταβλητή μια συγκεκριμένη τιμή. Η τιμή της μπορεί να είναι είτε αριθμός είτε συμβολοσειρά.
	Χρησιμοποιείται για να αυξομειώσει την τιμή μιας αριθμητικής μεταβλητής κατά έναν συγκεκριμένο αριθμό. Η εντολή αυτή μπορεί να λάβει μόνο αριθμητικές τιμές ως όρισμα.

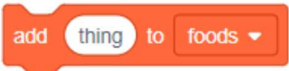
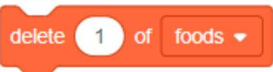
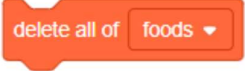

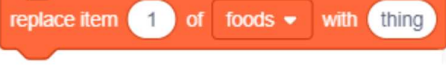


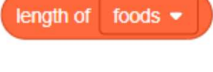

Πίνακας 6.13 Οι εντολές της Κατηγορίας των Μεταβλητών.

Μία λίστα (list) είναι μια διατεταγμένη συλλογή από τιμές που ονομάζονται στοιχεία της λίστας (elements). Δεν είναι απαραίτητο τα στοιχεία μιας λίστας να είναι του ίδιου τύπου, καθώς μια λίστα μπορεί να αποθηκεύσει και αριθμούς και συμβολοσειρές ταυτόχρονα, ενώ σαν δομή επιτρέπει και τις διπλοτυπίες, δηλαδή, μια τιμή σε μία λίστα μπορεί να υπάρχει περισσότερες από μία φορές. Κάθε στοιχείο μιας λίστας αντιστοιχεί σε μια θέση μέσα σε αυτή, η οποία αντιστοιχεί σε έναν δείκτη.

Όπως και στις μεταβλητές, η κατηγορία είναι άδεια από εντολές μέχρι ο χρήστης να δημιουργήσει μια νέα λίστα. Ας υποθέσουμε ότι δημιουργούμε μια λίστα με το όνομα “foods”.

	Χρησιμοποιείται για τη δημιουργία νέων λιστών.
	Συγκρατεί τις τιμές που είναι αποθηκευμένες στην λίστα που ονομάζει “foods” ως συμβολοσειρά.



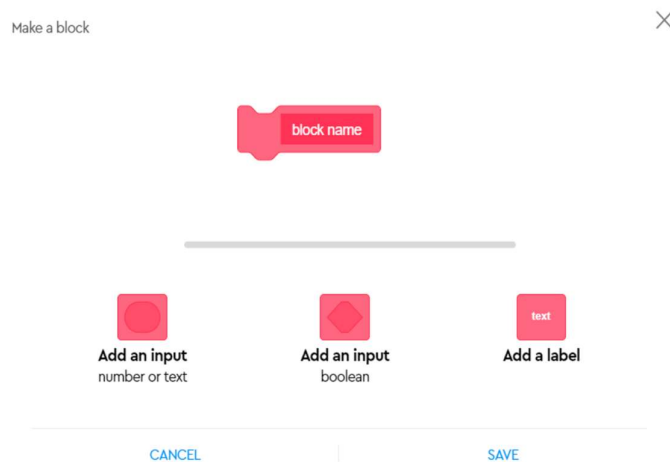
	Προσθέτει ένα καινούργιο στοιχείο στο τέλος μιας λίστας.
	Διαγράφει το στοιχείο που βρίσκεται στην θέση που ορίζει ο χρήστης.
	Διαγράφει όλα τα στοιχεία που είναι αποθηκευμένα στην συγκεκριμένη λίστα.
	Προσθέτει ένα στοιχείο σε μια θέση της λίστας που ορίζεται από τον χρήστη.
	Αντικαθιστά την τιμή που βρίσκεται στη θέση που επιλέγουμε με μια νέα τιμή που ορίζουμε εμείς.
	Επιστρέφει το στοιχείο που βρίσκεται στη θέση που του ζητά ο χρήστης.
	Αυτό το block επιστρέφει τον αριθμό της θέσης που πρωτοεμφανίζεται το στοιχείο που ζητάμε στη λίστα. Αν το στοιχείο δεν βρίσκεται στη συγκεκριμένη λίστα επιστρέφει την τιμή «0».
	Επιστρέφει τον αριθμό των στοιχείων που περιέχει μια λίστα.
	Αυτό το block επιστρέφει αληθές εάν το στοιχείο που ζητάμε βρίσκεται στην συγκεκριμένη λίστα. Για να αναγνωρίσει μια τιμή θα πρέπει να είναι ακριβώς ίδια με την τιμή που βρίσκεται μέσα στη λίστα.

Πίνακας 6.14 Οι εντολές της Κατηγορίας των Λιστών.

Κόκκινη Κατηγορία My blocks/Κατηγορία Συναρτήσεων: Σε αυτή την κατηγορία μπορεί να ανατρέξει ο χρήστης όταν θελήσει να δημιουργήσει τις δικές του συναρτήσεις. Στον προγραμματισμό οι συναρτήσεις είναι επαναχρησιμοποιήσιμα μέρη κώδικα που μπορούμε να τα χρησιμοποιήσουμε ξανά σε όποιο σημείο του κώδικα τα χρειαζόμαστε και όσες φορές θέλουμε. Ουσιαστικά είναι σαν να δημιουργούμε μια ομάδα εντολών στην οποία δίνουμε ένα συγκεκριμένο όνομα και στην συνέχεια, όταν θελήσουμε να εκτελέσουμε αυτό το κομμάτι κώδικα, αρκεί να καλέσουμε την συνάρτηση με το όνομα της. Είναι πολύ σημαντικό να κατανοήσει ο χρήστης πως όταν δημιουργεί μια συνάρτηση, το κομμάτι αυτό του κώδικα δεν εκτελείται εκείνη τη στιγμή παρά ζητάμε από το πρόγραμμα να αποθηκεύσει αυτό το κομμάτι και να το εκτελέσει όταν εμείς του ζητήσουμε.



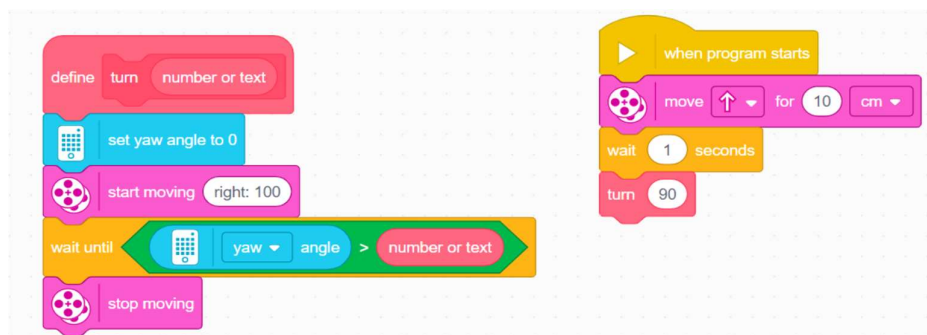
Πατώντας κανείς το κουμπί “Make a Block” εμφανίζεται το παρακάτω πλαίσιο.



Εικόνα 6.7 Δημιουργία συναρτήσεων σε Block-Based προγράμματα.

Στο πρώτο block που συναντά κανείς καλείται να ονομάσει την συνάρτηση. Στα blocks που βρίσκονται κάτω από τη γραμμή, ο χρήστης μπορεί να επιλέξει αν θέλει η συνάρτησή του να έχει παραμέτρους καθώς και τον τύπο των παραμέτρων. Οι παράμετροι βοηθούν στο να δοθούν τιμές ως είσοδος στη συνάρτηση, με σκοπό αυτή να τις αξιοποιήσει και να κάνει κάτι με αυτές τις τιμές. Όταν καλούμε μια συνάρτηση, οι τιμές που δίνουμε στις παραμέτρους ονομάζονται ορίσματα.

Για παράδειγμα, θέλουμε να δημιουργήσουμε μια συνάρτηση που θα ελέγχει κάθε φορά την στροφή που θα εκτελεί το ρομποτικό μας όχημα.



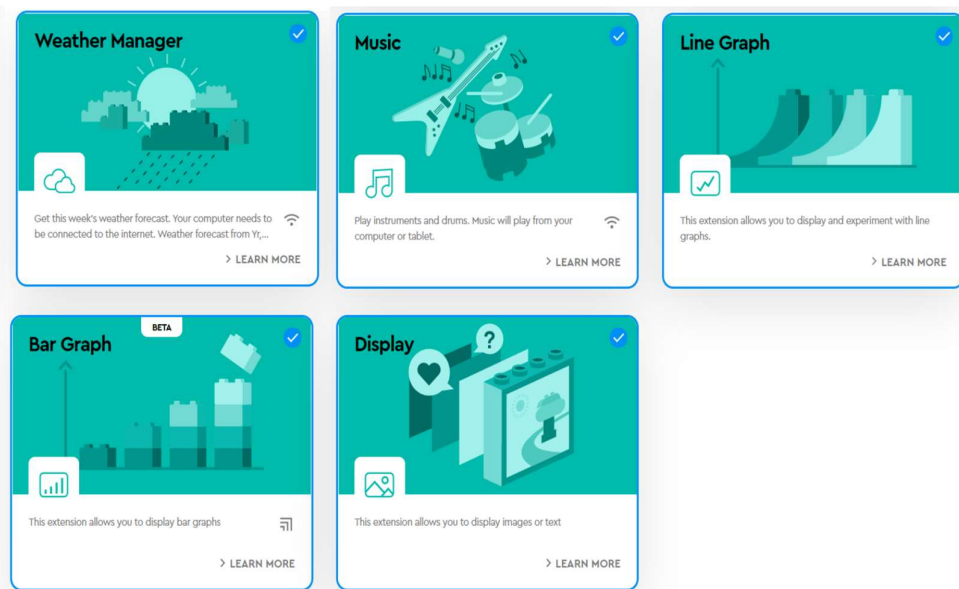
Εικόνα 6.8 Παράδειγμα Συνάρτησης σε Block-Based προγράμματα.

Αρχικά ορίζουμε μια συνάρτηση που την ονομάζουμε “turn”/στροφή, και ορίζουμε πως θα έχει μια παράμετρο που θα δέχεται ως τιμή αριθμό ή συμβολοσειρά.

Δημιουργούμε την συνάρτηση της στροφής, και στην συνέχεια όταν την καλούμε ορίζουμε πως οι στροφή που θέλουμε να εκτελέσει το ρομποτικό μας όχημα είναι στροφή 90 μοιρών. Το πλεονέκτημα που μας προσφέρει η συνάρτηση σε αυτή την περίπτωση είναι πως μπορούμε να τροποποιήσουμε τον αριθμό των μοιρών που θα στρίψει το όχημα μας αλλάζοντας απλά το 90 με κάποιον άλλον αριθμό.

## 6.7 Περισσότερες Επεκτάσεις.

Εκτός από τις Επεκτάσεις που αφορούν τους κινητήρες και τους αισθητήρες, υπάρχουν 5 πρόσθετες κατηγορίες με εντολές που αφορούν λειτουργίες όπως: πρόβλεψη καιρού, δημιουργία μουσικών κομματιών, δημιουργία γραφικών παραστάσεων και απεικόνιση εικόνων και κειμένων.

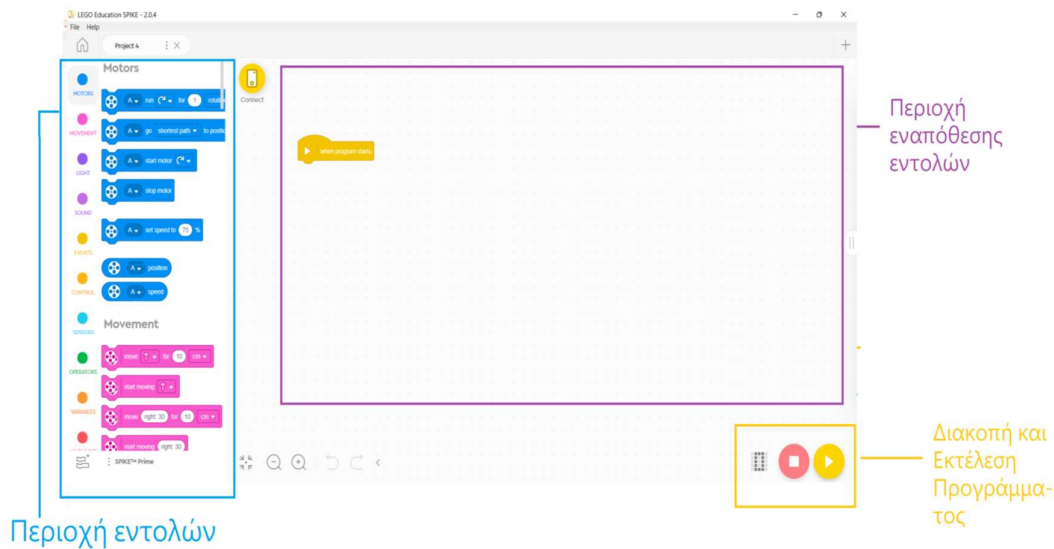


Εικόνα 6.9 Περισσότερες Επεκτάσεις της εφαρμογής.

## 6.8 Το Block-Based περιβάλλον προγραμματισμού στην εφαρμογή του Spike.

Το περιβάλλον του Block-Based Προγραμματισμού στην εφαρμογή του Spike Prime αποτελείται επίσης από δύο βασικές περιοχές: την «Περιοχή Εναπόθεσης» των εικονιδίων, όπου όπως προαναφέρθηκε και στον Icon-Based προγραμματισμό, λειτουργεί ως χώρος τοποθέτησης εντολών με σκοπό τη δημιουργία προγραμμάτων,

και την «Περιοχή των Εντολών» όπου κάθε εντολή είναι ταξινομημένη σε μια χρωματική κατηγορία. Ο χρήστης μπορεί να περιηγηθεί στις κατηγορίες και να διαλέξει τις εντολές «τραβώντας» τες. Τοποθετώντας τες μέσα στην περιοχή εναπόθεσης εντολών δημιουργεί τα δικά του προγράμματα. Προκειμένου να διαγραφεί μια εντολή, αρκεί να αποσυνδεθεί από το πρόγραμμα και να τοποθετηθεί ξανά στην «Περιοχή των Εντολών». Στην παρακάτω εικόνα απεικονίζεται η ανάλυση του περιβάλλοντος:



Εικόνα 6.10 Το περιβάλλον Block-Based Προγραμματισμού.

## 6.9 Ξεκινώντας με τη Γλώσσα Προγραμματισμού Python.

Η γλώσσα προγραμματισμού Python είναι μια text-based γλώσσα προγραμματισμού (γλώσσα που βασίζεται σε «κείμενο») υψηλού επιπέδου. Είναι μια γλώσσα πολύ ευέλικτη και αποτελεσματική, και αρκετά φιλική προς νέους προγραμματιστές κυρίως λόγω της απλής σύνταξης της. Το μεγαλύτερο πλεονέκτημα της προς τους νέους χρήστες είναι ότι είναι μια γλώσσα ευανάγνωστη, δηλαδή ένας κώδικας γραμμένος σε Python μπορεί να γίνει εύκολα κατανοητός ακόμα και από χρήστες που δεν έχουν μεγάλη εξοικείωση με τον προγραμματισμό.

Σημαντική πληροφορία για την Python αποτελεί το γεγονός ότι είναι μια κατεξοχήν αντικειμενοστραφής γλώσσα προγραμματισμού, αυτό σημαίνει ότι οτιδήποτε θέλουμε να εισάγουμε και να χρησιμοποιήσουμε στην Python είναι ένα αντικείμενο. Στην ουσία είναι σαν να μεταφέρουμε μια οντότητα του πραγματικού

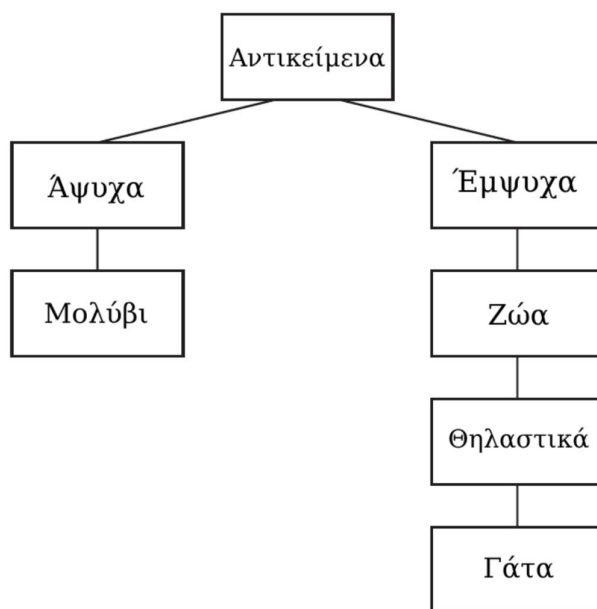
κόσμου μέσα στον υπολογιστή (και έπειτα στο πρόγραμμα) σαν ένα ψηφιακό αντικείμενο το οποίο μπορεί να χρησιμοποιηθεί από εμάς για να το προγραμματίσουμε. Για παράδειγμα, για να μπορέσουμε να προγραμματίσουμε έναν κινητήρα να εκτελέσει μια συγκεκριμένη ενέργεια θα πρέπει πρώτα να δημιουργήσουμε ένα ψηφιακό αντίγραφο του στον υπολογιστή. Για τον λόγο αυτό, καθώς κάθε εξάρτημα του SPIKE Prime αποτελεί έναν αντικείμενο η κατανόηση του αντικειμενοστραφή προγραμματισμού είναι απαραίτητη ώστε να μπορέσει κανείς να δημιουργήσει πρόγραμμα με τη γλώσσα Python.

Ο αντικειμενοστραφής προγραμματισμός αποτελεί μια πολύ διαδεδομένη προσέγγιση η οποία είναι «στραμμένη» (προσανατολισμένη) στα αντικείμενα (objects). Τα αντικείμενα (objects) είναι δομές της Python που προσομοιάζουν πραγματικά αντικείμενα του πραγματικού κόσμου.

Κεντρικό επίσης στοιχείο στην έννοια του αντικειμενοστραφή προγραμματισμού είναι η κλάση (Class). Στην ουσία η κλάση είναι ένα πρότυπο, ένα «καλούπι» (template) για τη δημιουργία αντικειμένων. Όλα τα αντικείμενα ανήκουν σε μια κλάση και ονομάζονται «περιπτώσεις» ή «υποστάσεις» (Instances) αυτής της κλάσης. Η κλάση από την οποία προέρχεται ένα αντικείμενο καθορίζει τα χαρακτηριστικά (Attributes, Properties) και ποιες ενέργειες/μέθοδοι θα περιέχονται σε αυτά (Behaviors, Methods). Για παράδειγμα εάν είχαμε την κλάση «Σκύλος» θα μπορούσαμε να έχουμε τα χαρακτηριστικά (Attributes, Properties) : Ηλικία, χρώμα, ράτσα ενώ στις ενέργειες θα μπορούσαμε να έχουμε: Περπατάω, γαβγίζω, τρώω κλπ.

Ένα βασικό χαρακτηριστικών των γλωσσών αντικειμενοστραφή προγραμματισμού είναι η κληρονομικότητα (Inheritance). Η κληρονομικότητα είναι η διεργασία μέσω της οποίας μια κλάση η οποία προέρχεται από μια άλλη κλάση μπορεί να κληρονομήσει τις ιδιότητες και τις μεθόδους της αρχικής κλάσης. Σε αυτή την περίπτωση γονική κλάση (Parent Class) ονομάζουμε την αρχική κλάση και κλάση-παιδί (Child Class) ή υποκλάση (Subclass) ονομάζουμε τη δευτερεύουσα κλάση, αυτή που κληρονομεί τις ιδιότητες και τις μεθόδους της γονικής κλάσης.

Παρακάτω παραθέτουμε ένα παράδειγμα με σκοπό να γίνουν τα προαναφερθέντα πιο κατανοητά:



Εικόνα 6.11 Απεικόνιση της κληρονομικότητας των κλάσεων.

Η βασική κλάση στο παράδειγμα αυτό είναι η κλάση «Αντικείμενα», η οποία είναι η γονική κλάση στις δύο υποκλάσεις «Έμψυχα» και «Άψυχα». Ξεκινώντας από κάτω προς τα πάνω, κάθε ορθογώνιο που βρίσκεται σε κατώτερο επίπεδο από το προηγούμενο είναι «υποκλάση» του προηγούμενου ορθογωνίου κλάσης. Αυτό σημαίνει ότι τα χαρακτηριστικά και οι ενέργειες κάθε ορθογωνίου κληροδοτούνται στο επόμενο: Έτσι για παράδειγμα, η κλάση «Ζώα» κληρονομεί από την κλάση «Έμψυχα» την ενέργεια «Αναπνέουν» και η κλάση «Καμηλοπάρδαλη» κληρονομεί από την κλάση «Θηλαστικά» την ενέργεια «Θηλάζουν». Όσον αφορά τα χαρακτηριστικά, θα μπορούσαμε να πούμε ότι η κλάση «Ζώα» κληρονομεί από την κλάση «Έμψυχα», το χαρακτηριστικό «Μάτια».

## 6.10 Προγραμματισμός με Python στην εφαρμογή του Spike Prime.

Όλες οι εντολές της Python που είναι διαθέσιμες την έκδοση 2.0.4 είναι ομαδοποιημένες μέσα σε μια Γνωσιακή Βάση/Knowledge Base που βρίσκεται στο δεξί μέρος της οθόνης. Για τον λόγο αυτό δεν είναι απαραίτητο να θυμάται κανείς όλες τις

ενέργειες που μπορεί να εκτελέσει το ρομποτικό σύστημα, καθώς ανά πάσα στιγμή μπορεί να τρέξει στην Γνωσιακή Βάση και να τη συμβουλευτεί για το πρόγραμμα του. Η μετάβαση από τον προγραμματισμό με Blocks στον προγραμματισμό με Python μπορεί να γίνει πολύ εύκολα, καθώς για κάθε ενέργεια που εκτελείται μέσω Blocks υπάρχουν και αντίστοιχες εντολές στην Python. Έτσι ο χρήστης δεν χρειάζεται να ξαναγνωρίσει τις εντολές και τις ενέργειες από την αρχή, παρά μόνο να ταιριάζει τις ενέργειες με τις εντολές. Επιπρόσθετα, καθώς οι εντολές της Python είναι γραμμένες στα αγγλικά, έχουν και υψηλό επίπεδο κατανόησης ακόμα και από πιο αρχάριους χρήστες. Για παράδειγμα, αν συναντήσει κανείς την εντολή «hub.light\_matrix.show\_image('HAPPY')» είναι εύκολο να κατανοήσει πως είναι μια εντολή που χρησιμοποιεί τον φωτεινό πίνακα του Hub για να δείξει ένα χαρούμενο εικονίδιο.

Στην Γνωσιακή Βάση θα βρει κανείς μια πληθώρα αναλυτικών πληροφοριών που αφορούν τον προγραμματισμό με Python, και για το πως μπορεί να εργαστεί χρησιμοποιώντας τους κινητήρες, τους αισθητήρες αλλά και τα διάφορα εξαρτήματα του Spike, καθώς και διάφορες άλλες πληροφορίες που είναι βασικές για τον προγραμματισμό με Python. Λόγω όμως του ότι μερικές βασικές πληροφορίες που χρειάζεται κανείς για να εργαστεί με την Python δεν είναι καλά οργανωμένες, είτε παρουσιάζονται ελλείψεις στην Γνωσιακή Βάση, θα παρουσιαστούν αναλυτικότερα στην συνέχεια του κεφαλαίου.

#### Εσοχές και διαστήματα:

Σε αντίθεση με πολλές γλώσσες προγραμματισμού η Python χρησιμοποιεί την εσοχή (indentation) ή κενά, για να ομαδοποιήσει τις εντολές που εκτελούνται στο ίδιο επίπεδο. Για να δημιουργήσει κάποιος χρήστης μια εσοχή αρκεί να πατήσει το Tab πλήκτρο ή το πλήκτρο Space τέσσερις φορές. Οι εντολές που είναι στην ίδια θέση, ξεκινούν δηλαδή με τον ίδιο αριθμό εσοχών, δημιουργούν ένα block κώδικα. Ένα block κώδικα είναι ένα ομαδοποιημένο σύνολο εντολών. Κάθε φορά που αλλάζουμε την εσοχή δημιουργούμε ένα νέο block κώδικα που στην ουσία είναι κομμάτι του προηγούμενου block.

Παράδειγμα 1:



Εικόνα 6.12 Παράδειγμα των εσοχών και των Blocks.

Στο συγκεκριμένο παράδειγμα έχουμε διαχωρίσει τη δομή ελέγχου if-else με τη δομή επανάληψης for χρησιμοποιώντας διαφορετικό αριθμό εσοχών. Όμως αυτές οι δύο δομές δεν είναι ανεξάρτητες καθώς το block εντολών που δημιουργεί η δομή επανάληψης for είναι κομμάτι του block που δημιουργείται από την δομή ελέγχου if-else.

Τέλος, αξίζει να αναφερθεί ότι το προγραμματιστικό περιβάλλον του Spike Prime μάς βοηθάει να ξεχωρίσουμε τις εντολές που ομαδοποιούνται εμφανίζοντας μπροστά μια γκρι γραμμή.

Παράδειγμα 2:

```
if speed1 == speed2:
    for x in range(5):
        hub.light_matrix.show_image('HAPPY')
        app.play_sound('Win')
else:
    app.play_sound('Oops')
```

Εικόνα 6.13 Απεικόνιση της βοηθητικής γκρι γραμμής.

Σε αυτό το παράδειγμα βλέπουμε πως η δεσμευμένη λέξη else δεν βρίσκεται στην ίδια ευθεία με την δεσμευμένη λέξη if, άρα δεν βρίσκονται στο ίδιο πεδίο ισχύος

όπως θα έπρεπε. Σε αυτή την περίπτωση, το περιβάλλον του Spike Prime θα μας εμφανίζει ένα μήνυμα σφάλματος.

#### Κεφαλαία και μικρά γράμματα:

Ένα χαρακτηριστικό της Python είναι ότι είναι case sensitive, δηλαδή αναγνωρίζει τα κεφαλαία και τα μικρά γράμματα σαν διαφορετικούς χαρακτήρες. Έτσι, το νόημα και η ορθότητα μιας εντολής επηρεάζεται από τη χρήση κεφαλαίων ή μικρών γραμμάτων. Για παράδειγμα, εάν κανείς πληκτρολογήσει την εντολή `hub.light_matrix.show_image ('Happy')` αντί της `hub.light_matrix.show_image('HAPPY')` τότε θα λάβει ένα μήνυμα σφάλματος.

Η διάκριση αυτή μεταξύ κεφαλαίων και μικρών γραμμάτων πρέπει να προσεχθεί ιδιαίτερα στη δημιουργία μεταβλητών. Για την Python η μεταβλητή “MotorSpeed” και η μεταβλητή “Motorspeed” είναι δύο τελείως διαφορετικά αντικείμενα. Καθώς η Python δεν επιτρέπει την χρήση κενών στα ονόματα μεταβλητών, η χρήση κεφαλαίων γραμμάτων στην αρχή κάθε λέξης από τις οποίες μπορεί να συνεισφέρει στο να είναι το πρόγραμμα μας πιο ευανάγνωστο, αρκεί το όνομα της μεταβλητής να παραμένει ακριβώς ίδιο σε ολόκληρο το πρόγραμμα ειδικά, όπως προαναφέρθηκε, θα δημιουργηθούν σφάλματα στον κώδικα μας.

#### Εκτέλεση εντολών:

Η ολοκλήρωση και κατά συνέπεια η εκτέλεση εντολών δεν γίνεται με χρήση χαρακτήρων τερματισμού της εντολής, αλλά με απλή αλλαγή γραμμής, πατώντας δηλαδή το πλήκτρο Enter.

#### Χρήση σχολίων:

Κατά τη δημιουργία ενός προγράμματος είναι πάρα πολύ βοηθητικό να χρησιμοποιεί κανείς σχόλια. Ένα σχόλιο είναι μια γραμμή κώδικα η οποία δεν εκτελείται, και μπορεί να χρησιμοποιηθεί σε ένα πρόγραμμα ώστε να καταγράψει κανείς τι κάνει ο κώδικας του ή ένα κομμάτι από κώδικα. Τα σχόλια στην Python δημιουργούνται βάζοντας στην αρχή της εντολής το σύμβολο #, ενώ σε περίπτωση που



κάποιος χρήστης θελήσει να χρησιμοποιήσει ως σχόλιο ένα μεγαλύτερο κομμάτι κώδικα αυτό που χρειάζεται να κάνει, είναι να το κλείσει μέσα σε τριπλά εισαγωγικά.

Παράδειγμα 3:

```
#this is a comment
hub = PrimeHub()

hub.light_matrix.show_image('HAPPY')
```

Εικόνα 6.13 Παράδειγμα εισαγωγής σχόλιου

Χρησιμοποιώντας κανείς σχόλια μπορεί να βοηθηθεί, καθώς είναι χρήσιμα τόσο στο να σημειώνουμε τους λόγους για τους οποίους χρησιμοποιήσαμε συγκεκριμένες εντολές, αλλά μπορούν και να βοηθήσουν άλλους χρήστες να καταλάβουν τι κάνει ο συγκεκριμένος κώδικας. Επιπρόσθετα, η χρήση των σχολίων μπορεί να φανεί χρήσιμη και στη διαδικασία της αποσφαλμάτωσης ή της δοκιμής ενός κώδικα. Εάν για παράδειγμα θέλουμε να δοκιμάσουμε ποιο θα ήταν το αποτέλεσμα του κώδικά μας χωρίς μια συγκεκριμένη εντολή, αλλά δεν θέλουμε να την σβήσουμε, μπορούμε να την μετατρέψουμε σε σχόλιο και η εντολή θα σταματήσει να εκτελείται.

### Δομές Επανάληψης:

Οι δομές επανάληψης (loops) μας βοηθούν στο να επαναλάβουμε μια ενέργεια ή ένα κομμάτι κώδικα πολλές φορές στη σειρά. Όταν η ροή του προγράμματος μπει σε μια δομή επανάληψης, οι εντολές που βρίσκονται μέσα της θα εκτελεστούν σειριακά ξανά και ξανά μέχρι να τελειώσει ο αριθμός των επαναλήψεων που έχουμε ορίσει, και έπειτα θα συνεχίσει στην κανονική ροή του προγράμματος. Στην Python συναντάμε δύο δομές επανάληψης: τις δομές επανάληψης while loops, όπου ένα κομμάτι κώδικα εκτελείται όσο είναι αληθής μια συγκεκριμένη συνθήκη, και τις δομές επανάληψης for loops, όπου χρησιμοποιούνται για να επαναλάβουμε ένα set εντολών για έναν συγκεκριμένο αριθμό επαναλήψεων.

#### 1) While loops:

Οι εντολές επανάληψης while loops επαναλαμβάνουν ένα set εντολών για όσο χρονικό διάστημα είναι αληθής μια συγκεκριμένη συνθήκη. Είναι χρήσιμες σε

περιπτώσεις όπου ο χρήστης δεν γνωρίζει τον ακριβή αριθμό επαναλήψεων που χρειάζεται να επαναληφθεί το κώματι του κώδικα.

Έτσι, ορίζοντας μια συνθήκη, όσο αυτή είναι αληθής το set εντολών θα επαναλαμβάνεται, και όταν η συνθήκη καταστεί ψευδής τότε θα σταματήσει η επανάληψη και θα συνεχιστεί η ροή του προγράμματος σειριακά.  
Παράδειγμα 4:

```
while hub.left_button.is_pressed():  
    hub.light_matrix.show_image ('HAPPY')
```

*Εικόνα 6.14 Παράδειγμα της εντολής While.*

Στο συγκεκριμένο παράδειγμα δημιουργούμε μια επανάληψη χρησιμοποιώντας την δεσμευμένη λέξη «while», και στη συνέχεια, ορίζοντας μια συνθήκη που πρέπει να είναι αληθής για να ξεκινήσει να εκτελείται ο κώδικας. Η συνθήκη στην συγκεκριμένη περίπτωση είναι «όσο είναι πατημένο το αριστερό πλήκτρο του Hub». Άρα όσο το αριστερό πλήκτρο είναι πατημένο, ο Φωτεινός Πίνακας/Light Matrix θα εμφανίζει το εικονίδιο HAPPY. Μόλις το κουμπί αφηθεί, τότε η συνθήκη θα γίνει ψευδής και η επανάληψη θα σταματήσει.

Η δομή επανάληψης «while» εκτελείται μόνο αν είναι αληθής τη συγκεκριμένη στιγμή που εκτελείται ο κώδικας. Εάν το συγκεκριμένο παράδειγμα βρισκόταν στην αρχή ενός προγράμματος και το αριστερό κουμπί δεν ήταν πατημένο κατά την εκκίνηση του προγράμματος ή δίναμε την εντολή να ξεκινήσει το πρόγραμμα και το πατούσαμε αργότερα, τότε δεν θα εκτελούνταν, καθώς το πρόγραμμα θα είχε ελέγξει την συνθήκη, θα την έβρισκε ψευδή και έτσι θα προχωρούσε στις υπόλοιπες εντολές.  
Παράδειγμα 5:

```
while hub.left_button.is_pressed():  
    hub.light_matrix.show_image ('HAPPY')  
    wait_for_seconds(3)  
    hub.light_matrix.show_image ('SAD')  
    wait_for_seconds (3)  
    hub.speaker.beep (60, 0.5)
```

*Εικόνα 6.15 Παράδειγμα 2 της εντολής While.*

Σε αυτό το παράδειγμα όλες οι εντολές στις οποίες προηγείται η εσοχή είναι κομμάτι της επανάληψης. Αντίθετα, η εντολή `hub.speaker.beep(60, 0.5)` είναι έξω από αυτή. Έτσι εκτελώντας τον συγκεκριμένο κώδικα, όσο το αριστερό πλήκτρο του Hub είναι πατημένο τότε στον Φωτεινό Πίνακα/ Light Matrix θα εναλλάσσονται τα εικονίδια HAPPY και SAD. Όταν το κουμπί αφηθεί, τότε το ηχείο του Hub θα παίξει έναν ήχο.

Χρησιμοποιώντας τη δομή «while not + συνθήκη» μπορούμε ανατρέψουμε το αποτέλεσμα.

Παράδειγμα 6:

```
while not hub.left_button.is_pressed():
    hub.light_matrix.show_image ('HAPPY')
    wait_for_seconds(3)
    hub.light_matrix.show_image ('SAD')
    wait_for_seconds (3)
hub.speaker.beep(60, 0.5)
```

*Εικόνα 6.16 Παράδειγμα της εντολής while not.*

Σε αυτό το παράδειγμα, η συνθήκη που πρέπει να είναι αληθής για να εκτελεστεί το πρόγραμμα είναι να μην είναι πατημένο το αριστερό πλήκτρο του Hub. Έτσι, σε αντίθεση με το προηγούμενο πρόγραμμα, τα εικονίδια θα εναλλάσσονται και μόλις πατηθεί το κουμπί τότε θα το ηχείο του Hub θα παίξει τον ήχο.

Τέλος, μια άλλη χρήση των δομών επανάληψης while είναι για να δημιουργήσουμε μια δομή επανάληψης που θα επαναλαμβάνεται για πάντα χρησιμοποιώντας τις δεσμευμένες λέξεις While και True.

Παράδειγμα 7:

```
while True:
    hub.light_matrix.show_image ('HAPPY')
    wait_for_seconds(3)
    hub.light_matrix.show_image ('SAD')
    wait_for_seconds (3)
```

*Εικόνα 6.17 Παράδειγμα της εντολής while True.*

Σε αυτή την περίπτωση δεν χρειαζόμαστε κάποια συνθήκη. Το block αυτό των εντολών θα επαναλαμβάνεται συνεχώς μέχρι τη διακοπή του προγράμματος, δηλαδή τα εικονίδια HAPPY και SAD θα εμφανίζονται εναλλάξ στο Hub για πάντα.

## 2) For loops:

Η δομή επανάληψης «for» επαναλαμβάνει ένα set εντολών για έναν συγκεκριμένο αριθμό επαναλήψεων. Σε αντίθεση με τη δομή επανάληψης while, ο χρήστης πρέπει να γνωρίζει τον αριθμό των επαναλήψεων που χρειάζεται για να χρησιμοποιήσει αυτού του είδους την επανάληψη. Σε αυτή την περίπτωση το πρόγραμμα θα συνεχίσει τη ροή του προγράμματος αφού ολοκληρωθεί ο αριθμός των επαναλήψεων που έχει οριστεί.

Παράδειγμα 8:

```
for i in range (5):  
    hub.light_matrix.show_image ('HAPPY')  
    wait_for_seconds(3)  
    hub.light_matrix.show_image ('SAD')
```

Εικόνα 6.18 Παράδειγμα της επανάληψης for.

Μέσα στην παρένθεση βρίσκεται ο αριθμός των επαναλήψεων που θέλουμε να εκτελεστούν. Σε αυτή την περίπτωση με το range (5) δηλώνουμε πως αυτό το κομμάτι του κώδικα θέλουμε να εκτελεστεί 5 φορές. Σε κάθε επανάληψη ο αριθμός της αποθηκεύεται μέσα στον επαναλήπτη, που στην συγκεκριμένη περίπτωση είναι το γράμμα i. Δηλαδή, σε κάθε περίπτωση το γράμμα i αντιστοιχεί σε έναν αριθμό ανάλογο του αριθμού επαναλήψεων που έχουν συμβεί, σαν ένας μετρητής. Έτσι στην πρώτη επανάληψη το i θα έχει τιμή 0, στην δεύτερη την τιμή 1 και ούτε κάθε εξής μέχρι την τελευταία επανάληψη όπου θα έχει την τιμή 4. Αυτό είναι μια ιδιαιτερότητα της Python, καθώς η μέτρηση ξεκινάει από τον αριθμό 0. Έτσι, ζητώντας της να μας δώσει πέντε τιμές παίρνουμε τις τιμές 0,1,2,3,4. Παρότι η τελευταία τιμή που παίρνει ο μετρητής είναι 4 ο αριθμός των επαναλήψεων είναι 5, καθώς μαζί με τον αριθμό 0 μας δίνει 5 τιμές.

Σε μια δομή επανάληψης for μπορούμε να αλλάξουμε 3 πράγματα: την αρχική της τιμή, να ορίσουμε την τελική της τιμή και να ορίσουμε και τον αριθμό-βήμα με τον

οποίο θα αυξάνεται η τιμή σε κάθε επανάληψη.

Παράδειγμα 9:

```
for i in range (1,10,2):
    hub.light_matrix.show_image ('HAPPY')
    wait_for_seconds(3)
    hub.light_matrix.show_image ('SAD')
```

*Εικόνα 6.19 Παράδειγμα της επανάληψης for με αρχική τιμή, τελική τιμή και βήμα.*

Σε αυτή την περίπτωση η αρχική τιμή είναι το 1, η τελική τιμή είναι το 10 και το βήμα της αλλαγής είναι 2. Έτσι, εφόσον η αλλαγή δεν γίνεται κατά 1 αλλά κατά 2 μονάδες, ο αριθμός των επαναλήψεων θα είναι 5 και όχι 10.

Τέλος, μπορούμε να χρησιμοποιήσουμε και αρνητικό βήμα στις επαναλήψεις μας. Αν για παράδειγμα θέλαμε να εκτυπώσουμε μια αντίστροφη μέτρηση των αριθμών από το 10 έως το 2 θα μπορούσαμε να γράψουμε το παρακάτω πρόγραμμα:

Παράδειγμα 10:

```
for i in range (10,1,-2):
    print (i)
```

*Εικόνα 6.20 Παράδειγμα της επανάληψης με αρνητικό βήμα.*

Δομές ελέγχου:

Οι δομές ελέγχου μας επιτρέπουν να ελέγξουμε διάφορες συνθήκες στο πρόγραμμά μας και να δράσουμε με διαφορετικό τρόπο ανάλογα με το αποτέλεσμα του ελέγχου. Στην Python οι πιο διαδεδομένες δομές ελέγχου είναι η if, η if/else και η if/elif/else.

### 1) Δομή ελέγχου if:

Η πιο απλή δομή ελέγχου είναι αυτή που ξεκινάει με τη δεσμευμένη λέξη if. Σε αυτή την περίπτωση το πρόγραμμα ελέγχει αν μια συνθήκη είναι αληθής ή ψευδής, και ανάλογα εκτελεί ή όχι ένα κομμάτι κώδικα.

Παράδειγμα 11:

```
if hub.left_button.is_pressed ():  
    hub.light_matrix.show_image('HAPPY')
```

Εικόνα 6.21 Παράδειγμα της εντολής if.

Το αποτέλεσμα αυτού του προγράμματος εξαρτάται από το αν είναι πατημένο το αριστερό κουμπί ή όχι. Αν τη στιγμή που η ροή του προγράμματος φτάσει στη δομή ελέγχου είναι πατημένο, τότε εκτελείται η παρακάτω εντολή και εμφανίζει το εικονίδιο HAPPY στην οθόνη του Hub, αν το κουμπί δεν είναι πατημένο τότε το πρόγραμμα θα παραβλέψει την εντολή και θα συνεχίσει τη ροή του προγράμματος χωρίς να περιμένει αν το κουμπί θα πατηθεί αργότερα ή όχι.

### 2) Δομή ελέγχου if/else:

Πιο σύνθετη δομή ελέγχου είναι αυτή που χρησιμοποιεί τις δεσμευμένες λέξεις if και else. Χρησιμοποιώντας μόνο την δεσμευμένη λέξη «if», δημιουργήσαμε μια δομή που εκτελεί ένα block εντολών όταν μια συνθήκη είναι αληθής τη στιγμή που η ροή του προγράμματος βρίσκεται σε εκείνο το σημείο. Όταν είναι ψευδής απλά δεν εκτελεί την εντολή και συνεχίζει τη ροή του προγράμματος. Συνδυάζοντάς την μαζί με την δεσμευμένη λέξη else τότε, μπορούμε να ορίσουμε τι θα γίνει και στην περίπτωση που η συνθήκη δεν είναι αληθής.

Παράδειγμα 12:

```
if hub.left_button.is_pressed ():  
    hub.light_matrix.show_image('HAPPY')  
else:  
    hub.light_matrix.show_image ('SAD')
```

Εικόνα 6.22 Παράδειγμα της δομής if/else.

Σε αυτή την περίπτωση η συνθήκη που ελέγχει το πρόγραμμα είναι «αν είναι πατημένο το αριστερό κουμπί». Αν είναι πατημένο τότε εμφανίζει ένα χαρούμενο εικονίδιο, αν δεν είναι αληθής τότε εμφανίζει ένα στεναχωρημένο εικονίδιο.

### 3) Δομή ελέγχου *if/elif/else*:

Η πιο σύνθετη δομή ελέγχου είναι αυτή που μπορούμε να ορίσουμε τόσο τι θα εκτελεστεί όταν μια συνθήκη είναι αληθής, όσο και μια δεύτερη συνθήκη που θα εκτελεστεί σε περίπτωση που είναι αληθής ενώ η πρώτη συνθήκη δεν είναι, και τέλος τι θα συμβεί στο πρόγραμμα μας αν καμία από τις προηγούμενες συνθήκες δεν είναι αληθείς. Ουσιαστικά, η δεσμευμένη λέξη *elif* (που είναι συνδυασμός των λέξεων *else* και *if*) εισάγει στο πρόγραμμα μια επιπλέον συνθήκη. Μπορεί να χρησιμοποιηθεί παραπάνω από μια φορά και να εισάγει πολλαπλούς ελέγχους στο πρόγραμμα.

Παράδειγμα 13:

```
if hub.left_button.is_pressed ():
    hub.light_matrix.show_image('HAPPY')
elif hub.right_button.is_pressed ():
    hub.light_matrix.show_image ('SAD')
else:
    hub.light_matrix.show_image ('SMILE')
```

*Εικόνα 6.23 Παράδειγμα της δομής if/elif/else.*

Στο συγκεκριμένο παράδειγμα, το πρόγραμμα ελέγχει αρχικά αν το αριστερό κουμπί είναι πατημένο. Αν η συνθήκη είναι αληθής, τότε εμφανίζει το εικονίδιο HAPPY στην οθόνη. Αν είναι ψευδής προχωράει στην επόμενη εντολή και πραγματοποιεί έναν δεύτερο έλεγχο. Αν η συνθήκη του δεύτερου ελέγχου είναι αληθής, τότε το εικονίδιο SAD εμφανίζεται στον φωτεινό πίνακα του Hub. Σε περίπτωση που καμία από τις προηγούμενες συνθήκες δεν είναι αληθείς τότε εκτελείται η τελευταία εντολή, όπου στο Hub εμφανίζεται το εικονίδιο SMILE.

Σε περίπτωση που παραπάνω από μια συνθήκες είναι αληθείς, τότε εκτελείται αυτή που βρίσκεται πιο ψηλά στην ροή του προγράμματος και οι υπόλοιπες παραλείπονται. Αν για παράδειγμα πατήσουμε το αριστερό και το δεξί πλήκτρο ταυτόχρονα, τότε το πρόγραμμα θα ελέγξει αν έχει πατηθεί το αριστερό και θα

εκτελέσει τις αντίστοιχες εντολές, καθώς αυτός ο έλεγχος βρίσκεται πιο ψηλά στην ροή του προγράμματος. Το γεγονός ότι πατήθηκε και το δεξί πλήκτρο θα αγνοηθεί.

### Τύποι δεδομένων:

Ο τύπος ενός δεδομένου είναι ένα χαρακτηριστικό που λέει σε ένα σύστημα υπολογιστή πώς να ερμηνεύσει την τιμή του. Κάθε τύπος δεδομένων έχει συγκεκριμένα χαρακτηριστικά και χρήσεις. Δεδομένου ότι ο προγραμματισμός ενός ρομποτικού συστήματος βασίζεται πολλές φορές σε δεδομένα τόσο εισαγόμενα (Input) όσο και εξαγόμενα (Output), όπως για παράδειγμα η απόσταση του ρομποτικού συστήματος από ένα αντικείμενο ή το χρώμα που θέλουμε να ανιχνεύσει ένας αισθητήρας, είναι πολύ σημαντικό να γνωρίζουμε τον τύπο αυτών των δεδομένων, ώστε να μπορούμε να τα χρησιμοποιούμε αποτελεσματικά μέσα στο πρόγραμμα. Παρότι γίνεται αναφορά στους τύπους δεδομένων στην Knowledge Base του Spike Prime, θεωρούμε ότι μια εκτενής αναφορά είναι απαραίτητη, καθώς οι τύποι δεδομένων είναι ένα πολύ σημαντικό στοιχείο σε ένα πρόγραμμα.

Στον προγραμματισμό του Spike Prime συναντάμε κυρίως τέσσερις βασικούς τύπους δεδομένων: τις συμβολοσειρές (Strings), τους αριθμούς (Numbers), τις μεταβλητές δυαδικού τύπου (Booleans) και τις None Type τιμές (τιμές που δεν έχουν ή δεν αναγνωρίζεται ο τύπος τους).

Οι συμβολοσειρές (Strings) είναι ακολουθίες από χαρακτήρες. Μπορούν να αποτελούνται από γράμματα, αριθμούς, σύμβολα ακόμα και κενά. Στην Python οι συμβολοσειρές τοποθετούνται πάντα ανάμεσα σε εισαγωγικά μονά (' ') ή διπλά ("").

Οι αριθμοί (Numbers) είναι αριθμητικού τύπου δεδομένα και χωρίζονται σε δύο βασικές κατηγορίες: τους ακέραιους αριθμούς (integers) και τους δεκαδικούς αριθμούς (floating numbers). Οι ακέραιοι αριθμοί είναι ολόκληρο αριθμοί, χωρίς δεκαδικό μέρος όπως το 10, το -2 κλπ. σε αντίθεση με τους δεκαδικούς που έχουν και το δεκαδικό μέρος, για παράδειγμα το 0.5.

Οι μεταβλητές δυαδικού τύπου (Booleans) ονομάζονται και Τύποι Δεδομένων Αληθείας καθώς μπορούν να πάρουν δύο τιμές: Αληθές και Ψευδές. Για παράδειγμα ,μπορούν να χρησιμοποιηθούν σε ένα πρόγραμμα ώστε να εκτελέσει τον ρομπότ μια



ενέργεια εάν μια συνθήκη είναι αληθής ή για να επιστρέψουν την τιμή «Ψευδές» ή «Αληθές» στο χρήστη στα πλαίσια ενός ελέγχου.

Τέλος, η τιμή None ανήκει στην κατηγορία None Type και χρησιμοποιείται μόνο όταν ένα αντικείμενο δεν έχει κάποια τιμή. Για παράδειγμα σε περίπτωση που έχουμε ζητήσει από κάποιο αισθητήριο όργανο να μας επιστρέψει μια τιμή, εάν η τιμή που μετρά είναι έξω από τα πιθανά όρια λειτουργίας του, είναι πιθανό να μας επιστρέψει την τιμή «None».

#### Αναγνώριση και μετατροπή τύπων δεδομένων:

Καθώς τα δεδομένα που εισάγονται και εξάγονται από ένα ρομποτικό σύστημα μπορεί να είναι είτε συμβολοσειρές, είτε αριθμοί (ακέραιοι και δεκαδικοί) ή δυαδικού τύπου είναι πολύ σημαντικό να γνωρίζουμε τον τύπο τους, ώστε να μπορέσουμε να τα χρησιμοποιήσουμε με κατάλληλο τρόπο σε ένα πρόγραμμα, αλλά και πώς μπορούμε να κάνουμε μετατροπές από έναν τύπο σε έναν άλλον σε περίπτωση που χρειαστεί.

Για να μπορέσουμε να ζητήσουμε από το πρόγραμμα να μας δώσει τον τύπο μιας μεταβλητής σε περίπτωση που δεν τον γνωρίζουμε ή θέλουμε να τον επιβεβαιώσουμε χρησιμοποιούμε τη δεσμευμένη λέξη type(), η οποία επιστρέφει τον τύπο των δεδομένων που εισάγουμε μέσα στην παρένθεση. Εάν ο τύπος των δεδομένων είναι συμβολοσειρά, το πρόγραμμα μας επιστρέφει την ένδειξη 'str', εάν είναι ακέραιος αριθμός την ένδειξη 'int', εάν είναι δεκαδικός μας επιστρέφει 'float', ενώ αν είναι δυαδικού τύπου ή δεν έχει τύπο μας επιστρέφει 'bool' και 'NoneType' αντίστοιχα. Ο λόγος για τον οποίο είναι σημαντικό να γνωρίζουμε τον τύπο των δεδομένων που χρησιμοποιούμε είναι γιατί πολλές φορές, εάν δεν χρησιμοποιούσε τον κατάλληλο τύπο σε διάφορα σημεία του προγράμματος, είναι δυνατόν το πρόγραμμα μας να μην μπορεί να εκτελεστεί.

Ας υποθέσουμε ότι θέλουμε να δημιουργήσουμε ένα πρόγραμμα όπου ως ταχύτητα ενός κινητήρα ορίζεται η ένταση του ανακλώμενου φωτός διαιρεμένη με τον αριθμό δύο.

Παράδειγμα 14:

```
1 motor = Motor ('A')
2 lightsensor = ColorSensor ('C')
3
4 while True:
5     brightness = lightsensor.get_reflected_light ()
6     speed = brightness / 2
7     motor.start (speed)
```

*Εικόνα 6.24 Παράδειγμα διαίρεσης στον προγραμματισμό με Python.*

Εάν προσπαθήσουμε να εκτελέσουμε το συγκεκριμένο πρόγραμμα θα εμφανιστεί στην κονσόλα μας μια ένδειξη σφάλματος. Ο λόγος είναι ότι η μέθοδος start που ενεργοποιεί τον κινητήρα χρειάζεται μια παράμετρο ακεραίου τύπου. Μια ιδιότητα της Python είναι ότι κάθε φορά που χρησιμοποιείται ο τελεστής διαίρεσης το αποτέλεσμα είναι δεκαδικός αριθμός, ακόμα και στην περίπτωση που η διαίρεση είναι τέλεια. Έτσι, ακόμα και αν η τιμή της μεταβλητής brightness ήταν ακέραιος αριθμός που διαιρείται τέλεια με το δύο, το αποτέλεσμα της θα ήταν δεκαδικός αριθμός. Εάν για παράδειγμα διαιρέσουμε τον αριθμό 10 με το 2, το αποτέλεσμα που θα πάρουμε θα είναι 5.0. Γι' αυτό το λόγο είναι σημαντικό να μπορούμε να γνωρίζουμε τον τύπο των δεδομένων που χρησιμοποιούμε, έτσι ώστε να μπορέσουμε να δημιουργήσουμε αποτελεσματικά προγράμματα.

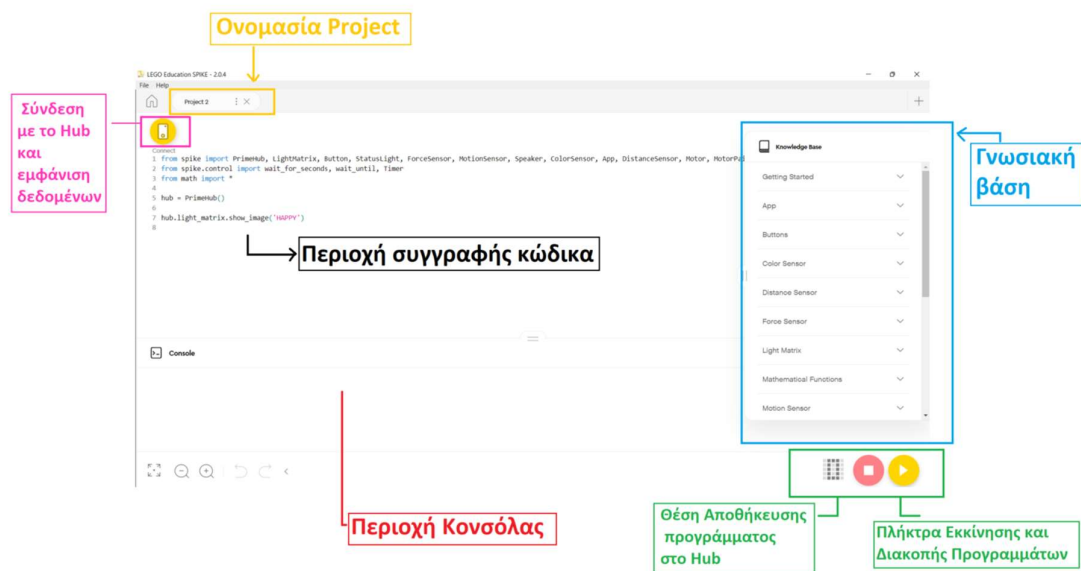
Για να αποφύγουμε τέτοια σφάλματα στο πρόγραμμα μας μπορούμε να χρησιμοποιήσουμε διάφορους τρόπους ώστε να μετατρέψουμε τα δεδομένα μας από έναν τύπο σε έναν άλλον. Οι βασικές συναρτήσεις - εντολές που χρησιμοποιούνται για αυτό τον σκοπό είναι η εξής:

- `round ()`: Μετατρέπει έναν δεκαδικό αριθμό στον πιο κοντινό ακέραιο.
- `floor()` και `ceil()`: Οι συγκεκριμένες συναρτήσεις χρησιμοποιούνται για να μετατρέψουν έναν δεκαδικό αριθμό στον πιο κοντινό ακέραιο είτε στρογγυλοποιώντας τον αριθμό προς τα πάνω (`ceil`) είτε στρογγυλοποιώντας τον προς τα κάτω (`floor`). Σημαντικό είναι πως αυτές οι συναρτήσεις είναι μέρος της βιβλιοθήκης `math`, οπότε για να χρησιμοποιηθούν από τον χρήστη, θα πρέπει πρώτα να εισαχθεί η βιβλιοθήκη με την χρήση της εντολής “`import math`”.
- `float ()`: Μετατρέπει ένα ακέραιο σε δεκαδικό αριθμό.

- `int()`: Χρησιμοποιείται για να μετατρέψει έναν δεκαδικό αριθμό σε ακέραιο αριθμό.
- `str()`: Αλλάζει τον τύπο ενός δεδομένου σε συμβολοσειρά.

### 6.11 Το περιβάλλον Προγραμματισμού με Python στην Εφαρμογή του Spike.

Το περιβάλλον όπου μπορεί να προγραμματίσει κανείς το Spike Prime χρησιμοποιώντας τη γλώσσα Python αποτελείται από της εξής περιοχές: Την «Περιοχή συγγραφής κώδικα», όπου γίνεται η συγγραφή των εντολών και η δημιουργία των προγραμμάτων, και την «Περιοχή της Κονσόλας» όπου εμφανίζονται διάφορα μηνύματα του προγράμματος, όπως ένα μήνυμα σφάλματος που προειδοποιεί τον χρήστη για σφάλματα που υπάρχουν στον κώδικα ή τα αποτελέσματα της εντολής `print()`. Επιπρόσθετα, διαθέτει και την «Γνωσιακή βάση» όπου όπως προαναφέρθηκε μπορεί κανείς να βρει πληροφορίες για πως να ξεκινήσει τον προγραμματισμό με Python, πληροφορίες για το πως να προγραμματίσει τα εξαρτήματα του Spike Prime καθώς και όλες τις εντολές που μπορεί να χρησιμοποιήσει για να δημιουργήσει προγράμματα γραμμένα σε Python.



Εικόνα 6.25 Το περιβάλλον προγραμματισμού με Python.

## Εισαγωγή Βιβλιοθηκών:

Καίρια στον προγραμματισμό του Spike Prime με Python είναι η έννοια της βιβλιοθήκης. Υπάρχουν δυο ειδών βιβλιοθηκών στην Python:

1. Βιβλιοθήκες που είναι εξωτερικές και τις χρησιμοποιεί η γλώσσα. Για να εισάγουμε τις εξωτερικές βιβλιοθήκες και να μπορέσουμε να τις χρησιμοποιήσουμε χρειάζεται να κάνουμε την εισαγωγή τους με τη δεσμευμένη λέξη «import».
2. Βιβλιοθήκες που είναι ενσωματωμένες στη γλώσσα. Για να χρησιμοποιήσουμε τις ενσωματωμένες βιβλιοθήκες δεν πρέπει να κάνουμε κάποιου είδος εισαγωγή. Χρησιμοποιήσουμε απλά τις εντολές, καθώς βρίσκονται ήδη ενσωματωμένες στην Python. Για παράδειγμα οι συναρτήσεις max, sub, range είναι ενσωματωμένες ήδη στην Python και δεν χρειάζεται να τις εισάγουμε, απλά τις καλούμε πληκτρολογώντας την εντολή.

Για να μπορέσουμε να χρησιμοποιήσουμε τα εξαρτήματα-αντικείμενα του Spike θα πρέπει πρώτα να εισάγουμε τις αντίστοιχες βιβλιοθήκες. Είναι σημαντικό πως οποιοδήποτε αντικείμενο θέλω να «περάσω» από τον πραγματικό κόσμο «μέσα» στον υπολογιστή πρέπει να το κάνω με τη εισαγωγή μιας βιβλιοθήκης. Άρα λοιπόν οι βιβλιοθήκες που εισάγω συνδέονται άμεσα με τα κομμάτια τα πραγματικά που μπορώ να βάλω στο πρόγραμμά μου. Υπάρχουν δύο τρόποι εισαγωγής των βιβλιοθηκών:

- Ο πρώτος τρόπος είναι να χρησιμοποιήσουμε την εντολή «import» και στη συνέχεια να γράψουμε το όνομα της βιβλιοθήκης που θέλουμε να εισάγουμε π.χ. Math. Αν χρησιμοποιήσουμε αυτόν τον τρόπο, για να αναφερθούμε σε μεθόδους της βιβλιοθήκης θα πρέπει όταν εκτελέσουμε τις εντολές, να γράψουμε το όνομα της βιβλιοθήκης, να προσθέσουμε μια τελεία και στη συνέχεια τη μέθοδο που θα χρησιμοποιήσουμε. Για παράδειγμα για να αναφερθούμε στον αριθμό π, θα πρέπει να γράψουμε πρώτα το όνομα της βιβλιοθήκης «Math» και μετά το «.pi».
- Ο δεύτερος τρόπος είναι να χρησιμοποιήσουμε κομμάτια της βιβλιοθήκης χρησιμοποιώντας την δεσμευμένη λέξη «from», στη συνέχεια το όνομα της βιβλιοθήκης, το «import» και μετά είτε να γράψουμε της μεθόδους που θέλουμε να εισάγουμε, είτε να γράψουμε έναν αστερίσκο που ερμηνεύεται ως εισαγωγή όλων των μεθόδων της κλάσης μιας της βιβλιοθήκης. Σε περίπτωση που χρησιμοποιήσουμε αυτόν τον τρόπο μπορούμε απευθείας να χρησιμοποιήσουμε

την μέθοδο χωρίς να προηγηθεί το όνομα της βιβλιοθήκης. Έτσι λοιπόν μειώνεται και ο όγκος του κώδικα που γράφουμε.

Μετά την εισαγωγή των βιβλιοθηκών ακολουθεί η αρχικοποίηση και η δημιουργία αντικειμένων. Για δημιουργήσω ένα αντικείμενο και να μπορέσω να το χρησιμοποιήσω, αρκεί να εισάγω την κλάση που χρειάζομαι μέσα σε μια μεταβλητή. Αν θέλω για παράδειγμα να χρησιμοποιήσω το Hub αρκεί να το εισάγω μέσα σε μια μεταβλητή με τον εξής τρόπο: `hub = PrimeHub()`. Στο τέλος, ο χρήστης πρέπει να συμβουλευτεί την Γνωσιακή Βάση προκειμένου να δει τις μεθόδους που υλοποιεί η κλάση και καλεί αυτές που χρειάζεται για το αντικείμενο που έχει δημιουργήσει.

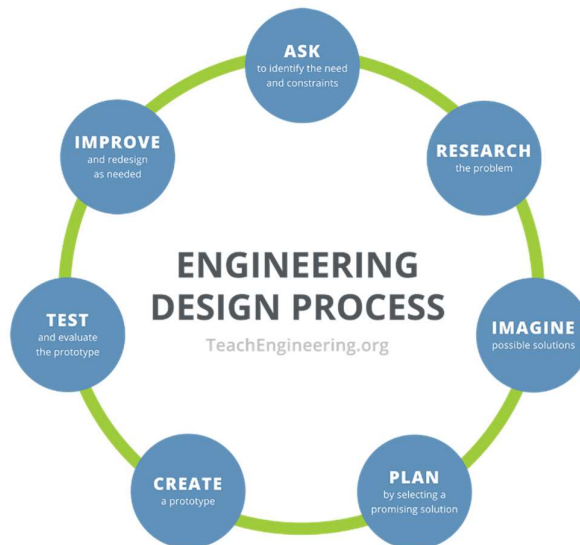
## Κεφάλαιο 7 – Σχεδιασμός και Προγραμματισμός Ρομποτικών Συστημάτων

### Εισαγωγή Κεφαλαίου.

Σκοπός αυτού του κεφαλαίου είναι να παρουσιάσει τον τρόπο τη διαδικασία που ακολουθήθηκε για την ανάπτυξη των ρομποτικών συστημάτων που υλοποιήσαμε στα πλαίσια αυτής της εργασίας. Η διαδικασία αυτή είναι μια σειρά από βήματα που ακολουθούν οι μηχανικοί για να βρουν μια λύση σε ένα πρόβλημα. Αρκετές φορές ως λύση ενός προβλήματος ορίζεται ο σχεδιασμός ενός προϊόντος που πληροί ορισμένα κριτήρια (όπως ένας υπολογιστής) ή ένα προϊόν που εκτελεί μια συγκεκριμένη εργασία.

### 7.1 Διαδικασία Σχεδιασμού Εφαρμοσμένης Μηχανικής.

Η Διαδικασία Σχεδιασμού Εφαρμοσμένης Μηχανικής (Engineering Design Process) είναι μια συστηματική προσέγγιση, μια επαναληπτική διαδικασία επίλυσης προβλημάτων που χρησιμοποιείται από τους μηχανικούς στην ανάπτυξη νέων και υφιστάμενων προϊόντων, συστημάτων και λύσεων σε διάφορα προβλήματα. Αυτή η διαδικασία χρησιμοποιείται συνήθως στο σχεδιασμό προϊόντων, υπηρεσιών και διαδικασιών. Αποτελείται από μια σειρά βημάτων που ακολουθούν οι μηχανικοί για να βρουν βιώσιμες λύσεις σε προβλήματα.



Εικόνα 7.1 Παράδειγμα της διαδικασίας Σχεδιασμού Εφαρμοσμένης Μηχανικής.

Ωστόσο δεν υπάρχει μια διαδικασία σχεδιασμού η οποία να είναι καθολικά αποδεκτή, καθώς ανατρέχοντας τη βιβλιογραφία μπορεί να εντοπίσει κανείς διάφορες παραλλαγές της διαδικασίας αυτής. Παρότι τα ενδιάμεσα βήματα της διαδικασίας μπορεί να διαφέρουν στις διάφορες προτάσεις, συνήθως η διαδικασία του σχεδιασμού ξεκινά με ένα πρόβλημα προς διερεύνηση και επίλυση, και καταλήγει στην πρόταση μιας λύσης για το πρόβλημα που έχουμε ορίσει.

Στην πλειονότητα των προτάσεων της διαδικασίας τα βήματα στη Διαδικασία Μηχανικού Σχεδιασμού περιλαμβάνουν:

1. Προσδιορισμός της Ανάγκης: Σε αυτό το βήμα προσδιορίζεται η ανάγκη ή το πρόβλημα που πρέπει να αντιμετωπιστεί.
2. Έρευνα και συλλογή πληροφοριών: Στη συνέχεια έπεται η συλλογή πληροφοριών οι οποίες είναι σχετικές και φωτίζουν πτυχές της ανάγκης ή του προβλήματος.
3. Καταιγισμός Ιδεών και Ανάπτυξη πιθανών λύσεων: Ο Καταιγισμός Ιδεών (Brainstorming) είναι μια τεχνική που χρησιμοποιείται για τη δημιουργία νέων ιδεών ή λύσεων για ένα συγκεκριμένο θέμα. Κατά τη διαδικασία του Brainstorming, όλα τα μέλη της ομάδας συμμετέχουν εκφράζοντας τις ιδέες τους, ενώ συνεχίζουν να αναζητούν και να διερευνούν διαρκώς για εναλλακτικές λύσεις. Η κεντρική ιδέα πάνω στην οποία βασίζεται η συγκεκριμένη μέθοδος είναι η εκτίμηση και η εξέταση των διάφορων ιδεών που θα προταθούν από μια διαφορετική γωνία, έτσι ώστε να διαπιστωθούν τα θετικά και τα αρνητικά σημεία των σκέψεων που έχουν προταθεί. Η διαδικασία επιτρέπει την εξέταση και την αναδιάταξη των ιδεών προκειμένου να διαπιστωθούν οι ιδανικές λύσεις και τα καλύτερα αποτελέσματα.
4. Επιλογή της Βέλτιστης Λύσης: Σε αυτό το στάδιο επιλέγεται η καλύτερη από τις λύσεις που έχουν προταθεί.
5. Σχεδιασμός και Κατασκευή Πρωτοτύπου: Στη συνέχεια σχεδιάζεται η λύση η οποία επιλέχθηκε ως βέλτιστη, αξιολογείται και στη συνέχεια έπεται η δοκιμή της. Αναφορικά με τον σχεδιασμό, οι μηχανικοί σχεδιάζουν μια δομή, ένα πρωτότυπο, οι οποία ικανοποιεί τις ανάγκες που αναδείχθηκαν. Ο όρος "πρωτότυπο" χρησιμοποιείται για να περιγράψει το πρώτο μοντέλο ή σχέδιο για ένα προϊόν ή έργο, και το οποίο μπορεί να χρησιμοποιηθεί ως βάση για την ολοκλήρωση του έργου καθώς προσφέρει μια γενική εικόνα για το προϊόν ή το έργο. Η κατασκευή του είναι ένας τρόπος για να εξερευνήσει κανείς και να εξετάσει τις ιδέες του και

πρόκειται για μια σύντομη και οικονομική μορφή δοκιμής, η οποία μπορεί να διευκολύνει τις διαδικασίες σχεδιασμού και ανάπτυξης. Η κατασκευή ενός πρωτότυπου μπορεί να αποτελέσει έναν σημαντικό εργαλείο για την κατανόηση των απαιτήσεων των χρηστών και την αποφυγή λαθών κατά τη διαδικασία ανάπτυξης.

6. Αξιολόγηση και δοκιμή της λύσης: Αυτό το βήμα γίνεται για να επαληθεύσει την επιτυχία του σχεδιασμού μετρώντας την απόδοση του συστήματος, και επιβεβαιώνοντας την αξιοπιστία της λύσης. Στην Διαδικασία Σχεδιασμού Εφαρμοσμένης Μηχανικής η δοκιμή είναι πολύ σημαντική για την επιτυχία του σχεδιασμού και την επιλογή της βέλτιστης λύσης.
7. Επανασχεδιασμός και Βελτίωση του Προϊόντος: Ο επανασχεδιασμός και η βελτίωση του προϊόντος είναι επίσης ένα σημαντικό βήμα για την επιτυχία του έργου. Σε περίπτωση που το τελικό προϊόν δεν ανταποκρίνεται στις απαιτήσεις που έχουμε θέσει ή θεωρούμε πως επιδέχεται βελτίωσης, τότε επιστρέφουμε στη φάση του σχεδιασμού, καθορίζονται οι αλλαγές που πρέπει να γίνουν στο σχέδιο και ακολουθεί η κατασκευή ενός νέου πρωτοτύπου και η εκ νέου δοκιμή.

Είναι πολύ σημαντικό να σημειωθεί πως στην παραγωγή προϊόντων, υπηρεσιών και διαδικασιών δεν υπάρχει απαραίτητα μια και μοναδική λύση αλλά πολυάριθμες παραλλαγές, για τον λόγο αυτό, η Διαδικασία Σχεδιασμού Εφαρμοσμένη Μηχανικής δεν είναι μια γραμμική διαδικασία αλλά μια διαδικασία η οποία μπορεί να απαιτεί την επιστροφή σε κάποιο προηγούμενο βήμα και την επανάληψη του.

## **7.2 Δημιουργία Εκπαιδευτικών Ρομποτικών Συστημάτων.**

Στα πλαίσια της παρούσας διπλωματικής η διαδικασία η οποία ακολουθήθηκε και συστήνεται για την ανάπτυξη των ρομποτικών συστημάτων αποτελείται από 9 βήματα. Πρόκειται για μια πρόταση δημιουργίας και σχεδιασμού η οποία συνδυάζει επιμέρους βήματα διαφόρων διαδικασιών που έχουν προταθεί για την ανάπτυξη ενός προϊόντος ή μιας ιδέας. Τα βήματα από τα οποία αποτελείται είναι τα εξής:

1. Ορισμός του Προβλήματος
2. Εξέταση και Ανάλυση του Προβλήματος
3. Καταιγισμός Ιδεών και Σύλληψη Πιθανής ιδέας



4. Διαίρεση του Συστήματος σε Λειτουργίες
5. Έρευνα και Επιλογή Μηχανισμών
6. Δημιουργία Προσχεδίου
7. Δημιουργία Πρωτοτύπου
8. Δοκιμή/Εξέταση
9. Βελτίωση/Επανασχεδιασμός

Παρότι τα ρομποτικά συστήματα τα οποία θέλουμε να αναπτύξουμε δεν αποτελούν άμεσα μια λύση σε ένα πρόβλημα του πραγματικού κόσμου, διατηρούμε αυτά τα βήματα καθώς επρόκειτο να τα εντάξουμε μέσα σε STEAM περιβάλλοντα. Η STEAM εκπαίδευσης, όπως προαναφέρθηκε, είναι μια κατεξοχήν μέθοδος διδασκαλίας βασιζόμενη στην επίλυση προβλημάτων, συνεπώς όταν θα τα εντάξουμε μέσα σε εκπαιδευτικά πλαίσια, τα ρομποτικά μας συστήματα θα πρέπει να αποτελούν απάντηση σε κάποιο πρόβλημα. Επιπρόσθετα, ένα πρόβλημα του πραγματικού κόσμου μπορεί να αποτελέσει και πηγή έμπνευσης σε αυτές τις περιπτώσεις. Στη συνέχεια ακολουθεί η παρουσίαση των βημάτων αναλυτικά:

1. Ορισμός του Προβλήματος: Όπως προαναφέρθηκε, όλες οι διαδικασίες σχεδιασμού ξεκινούν από το πρόβλημα το οποίο έρχεται να επιλύσει η διάταξη η οποία θα αναπτυχθεί. Πρόκειται για τον ξεκάθαρο προσδιορισμό του προβλήματος το οποίο καλούμαστε να επιλύσουμε το οποίο θα είναι αυστηρώς ορισμένο και αποτελεί τον τελικό στόχο του μαθήματος. Καθώς όπως προαναφέρθηκε η STEAM εκπαίδευση είναι μια προσέγγιση η οποία βασίζεται στην επίλυση προβλημάτων το συγκεκριμένο βήμα μας βοηθά στο να εντάξουμε το ρομποτικό σύστημα, το οποίο θα δημιουργήσουμε, ευκολότερα σε ένα εκπαιδευτικό πλαίσιο καθώς έχουμε ορίσει ήδη το πρόβλημα που θα αποτελέσει τον κεντρικό άξονα του μαθήματος.
2. Εξέταση και Ανάλυση του Προβλήματος: Στο συγκεκριμένο βήμα ορίζονται οι περιορισμοί που διέπουν το πρόβλημα και φωτίζονται οι πτυχές του που χρήζουν επίλυσης. Η ενδεδειγμένη ανάλυση του προβλήματος μας βοηθάει να αναπτύξουμε ιδέες και προσδοκίες για την λύση του.
3. Καταιγισμός Ιδεών και Σύλληψη Πιθανής ιδέας: Σε αυτό το στάδιο, με βάση τα προηγούμενα δύο στάδια προχωράμε στην σύλληψη διαφόρων ιδεών μέσω Brainstorming και στην επιλογή αυτής που φαίνεται ως βέλτιστη λύση για να χρησιμοποιηθεί ως ο βασικός ρομποτικός μηχανισμός που θα χρησιμοποιηθεί στα πλαίσια του μαθήματος.

4. Διαίρεση του Συστήματος σε Λειτουργίες: Η συνολική λειτουργία του συστήματος μπορεί να αναλυθεί σε περαιτέρω υπολειτουργίες που προσδιορίζουν σκόπιμες ενέργειες που επιθυμούμε να εκτελεί το ρομποτικό μας σύστημα. Οι λειτουργίες είναι μηχανικές ενέργειες, ουδέτερες ως προς τη λύση που θα εκτελέσει το προϊόν. Είναι σημαντικό να αναφερθεί πως οι λειτουργίες θα πρέπει να εξετάζουν το «τι» κάνει το ρομποτικό μας σύστημα και όχι το «πώς» το κάνει.
5. Έρευνα και Επιλογή Μηχανισμών: Στο στάδιο αυτό εξετάζουμε πλέον το «πώς» υλοποιεί το ρομποτικό μας σύστημα τις λειτουργίες που προαναφέραμε, αναζητώντας πιθανούς μηχανισμούς και λύσεις. Η αναζήτηση αυτή ασφαλώς μπορεί να περιλαμβάνει παρόμοιες κατασκευές που έχουν χρησιμοποιηθεί και εφαρμοστεί σε παρόμοια ρομποτικά συστήματα, είτε πρόκειται για εξ ολοκλήρου παρόμοιες κατασκευές, είτε για μέρη αυτών. Παρατηρώντας παρόμοιες υλοποιήσεις αφενός διευρύνουμε τις γνώσεις μας, παίρνουμε ιδέες και κατανοούμε καλύτερα το πρόβλημα, αφετέρου μπορούμε να εφαρμόσουμε ήδη υπάρχουσες υλοποιήσεις με δικές μας παρεμβάσεις και αλλαγές, ώστε να υλοποιήσουμε επιτυχώς το σύστημά μας, κερδίζοντας πολύτιμο χρόνο και ελαχιστοποιώντας το ρίσκο.

Είναι σημαντικό σε αυτό το σημείο να αναφερθεί πως ένας σημαντικός παράγοντας ο οποίος θα επηρεάσει την έρευνα και την επιλογή των μηχανισμών είναι το κατά πόσο το ρομποτικό σύστημα το οποίο έχουμε αποφασίσει να υλοποιήσουμε είναι ήδη υπαρκτό αντικείμενο του πραγματικού κόσμου ή όχι. Αν για παράδειγμα θελήσω να δημιουργήσω ένα ρομποτικό ζώο, θα πρέπει οι μηχανισμοί που έχω επιλέξει να υλοποιούν τις λειτουργίες του να είναι τέτοιοι ώστε εμφανισιακά να θυμίζει το συγκεκριμένο ζώο που έχω επιλέξει. Έτσι, εάν σε αυτή την περίπτωση έχω αποφασίσει να υλοποιήσω έναν ρομποτικό σκύλο τότε είναι προφανές πως δεν μπορώ να χρησιμοποιήσω στον μηχανισμό που θα υλοποιεί την λειτουργία της κίνησης, ρόδες.

Αντίθετα, σε περίπτωση που το ρομποτικό σύστημα που έχω επιλέξει να δημιουργήσω δεν είναι ένα μοντέλο ενός αντικειμένου του πραγματικού κόσμου, αλλά ένα σύστημα που δεν έχω προσδιορίσει κάπως την εμφάνιση του, και αυτό που γνωρίζω είναι πως θα εκτελεί κάποιες συγκεκριμένες ενέργειες για την επίλυση ενός προβλήματος, τότε η εμφάνιση του θα εξαρτηθεί σε μεγάλο βαθμό από τον συνδυασμό των μηχανισμών που έχω επιλέξει να υλοποιούν τις συγκεκριμένες λειτουργίες.

6. Δημιουργία προσχεδίου: Σε αυτό το σημείο, έχοντας προσδιορίσει τόσο τις ενέργειες τις οποίες θέλουμε να υλοποιεί το ρομποτικό μας σύστημα και τους μηχανισμούς οι οποίοι θα υλοποιούν τις λειτουργίες που έχουμε ορίσει, προχωρούμε στην δημιουργία ενός σχεδίου προκειμένου να προσδιορίσουμε τη γενική εικόνα για την εμφάνιση του ρομποτικού συστήματος.

Το σχέδιο το οποίο θα υλοποιηθεί σε αυτή την περίπτωση μπορεί να είναι είτε ένα χειρόγραφο σκίτσο, είτε ένα σκαρίφημα, είτε ένα 3D σχέδιο. Στην παρούσα εργασία το εργαλείο που χρησιμοποιήθηκε είναι το Studio 2.0 της εταιρείας Bricklink και είναι διαθέσιμο για εγκατάσταση σε αυτόν τον [σύνδεσμο](#). Είναι ένα αρκετά χρήσιμο εργαλείο για τη δημιουργία τρισδιάστατων κατασκευών, καθώς η υλοποίηση ενός σχεδίου βασίζεται στην ένωση δομικών στοιχείων της Lego που περιλαμβάνονται ήδη στην εφαρμογή. Επιλέχθηκε γιατί περιλαμβάνει όλα τα κομμάτια που διαθέτει το Spike Prime, επομένως δίνει στον χρήστη τη δυνατότητα να δημιουργήσει ακριβή προσχέδια πριν προχωρήσει στη δημιουργία του πρωτοτύπου.

7. Δημιουργία Πρωτοτύπου: Ο όρος "πρωτότυπο" αναφέρεται στο πρώτο μοντέλο ή σχέδιο για ένα προϊόν ή έργο που δημιουργείται, και που μπορεί να χρησιμοποιηθεί ως βάση για την ολοκλήρωση μιας κατασκευής. Είναι ένας τρόπος για να εξετάσει κανείς τις ιδέες του και να τις δοκιμάσει, ενώ προσφέρει και μια γενική εικόνα για το τελικό αποτέλεσμα του ρομποτικού συστήματος.

8. Δοκιμή/Εξέταση: Αυτό το βήμα έχει ως σκοπό να επιβεβαιώσει την επιτυχία του σχεδιασμού ή να αναδείξει τα σημεία που χρήζουν περαιτέρω μελέτης και διόρθωσης. Σκοπός αυτού του βήματος δηλαδή είναι να επιβεβαιώσει ή όχι την αξιοπιστία της λύσης.

9. Βελτίωση/Επανασχεδιασμός: Αν το τελικό προϊόν δεν ανταποκρίνεται στα κριτήρια που έχουμε θέσει, θα επιστρέψουμε σε προηγούμενες φάσεις, για να καθορίσουμε τις αλλαγές που πρέπει να γίνουν.

Έχοντας ολοκληρώσει τη διαδικασία σχεδιασμού και την ανάπτυξη ενός ρομποτικού μηχανισμού που θεωρούμε πως πληροί τις προδιαγραφές που θέσαμε και αποτελεί τη βέλτιστη λύση, συνεχίζουμε με την ολοκλήρωση της εκάστοτε δραστηριότητας δημιουργώντας τον κώδικα που θα πλαισιώνει την κατασκευή. Η κατασκευή εξάλλου από μόνη της δεν είναι ικανή να επιλύσει το πρόβλημα που έχουμε ορίσει, γι' αυτό χρειαζόμαστε και το προγραμματιστικό σκέλος. Ουσιαστικά, έχοντας

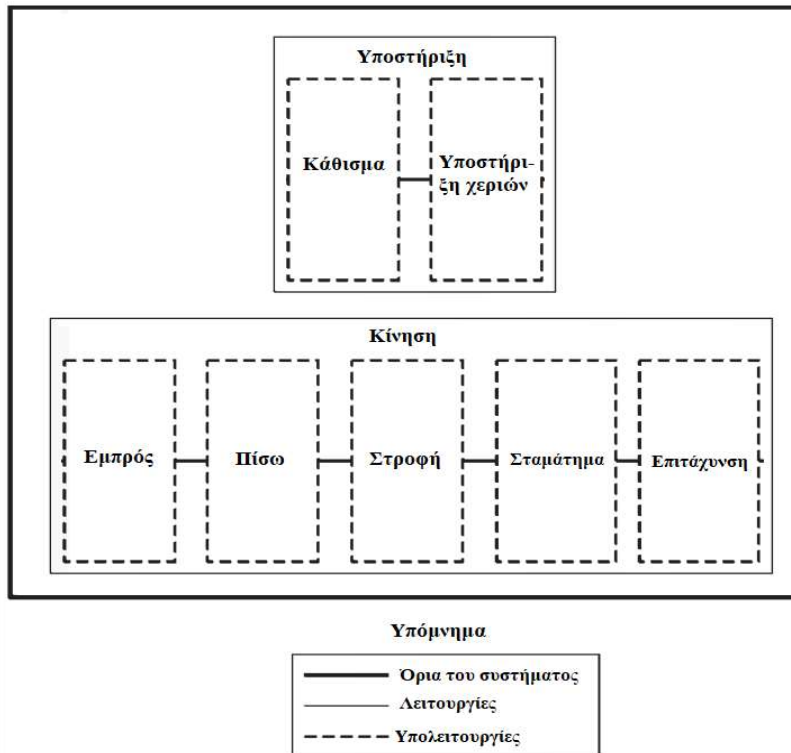
ήδη δομήσει την ιδέα που θα αποτελεί επίλυση σε ένα πρόβλημα, ταυτόχρονα έχουμε ορίσει και τις ενέργειες που επιθυμούμε να εκτελεί το ρομποτικό μας σύστημα. Άρα οι ενέργειες αυτές δεν μένει παρά να μεταφραστούν σε εντολές προγραμματισμού, συνδυάζοντας κατάλληλα του μηχανισμούς και τα εξαρτήματα που διαθέτει η κατασκευή μας με τις εντολές που προσφέρονται από το προγραμματιστικό περιβάλλον του Spike Prime.

## Κεφάλαιο 8 – Παρουσίαση των Εφαρμογών.

### 8.1 Αναπηρική Καρέκλα

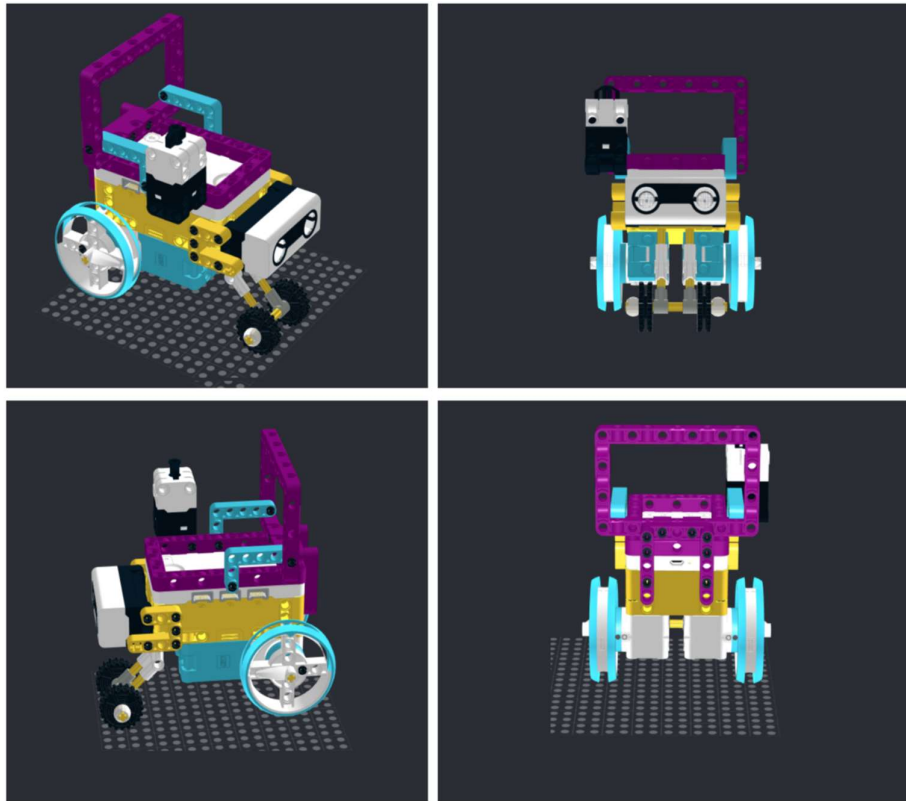
#### Διαδικασία Σχεδιασμού:

1. Ορισμός Προβλήματος: Ως πρόβλημα της συγκεκριμένης ενότητας ορίζεται η δυσκολία των ατόμων με αναπηρία να κινηθούν με ευκολία και άνεση.
2. Εξέταση και Ανάλυση του Προβλήματος: Οι πτυχές του προβλήματος που εντοπίσαμε και αποφασίσαμε πως χρήζουν αντιμετώπισης είναι οι εξής: η σωματική κούραση που προκαλείται από τον έλεγχο της καρέκλας, η καταπόνηση των χεριών τους, η ενδεχόμενη σύγκρουση σε εμπόδιο λόγω μειωμένης ορατότητας και η αυξημένη ένταση και προσπάθεια που πρέπει να καταβάλει κανείς σε κεκλιμένο επίπεδο.
3. Καταιγισμός Ιδεών και Σύλληψη Ιδέας: Με βάση τα δύο προηγούμενα στάδια καταλήξαμε στην ιδέα υλοποίησης ενός αναπηρικού αμαξιδίου, οι κινήσεις του οποίου θα ελέγχονται σε μεγάλο βαθμό από τον χρήστη ή αυτόματα από τον ίδιο τον ρομποτικό μηχανισμό, κάνοντας λιγότερο επίπονο και δύσκολο τον χειρισμό του. Ο χρήστης θα μπορεί να ελέγξει την μπροστινή και την πίσω κίνηση του αμαξιδίου, την περιστροφή του, ενώ ταυτόχρονα το ρομποτικό όχημα θα ελέγχει την ύπαρξη εμποδίων και την κίνηση του οχήματος σε ανηφορικό έδαφος.
4. Διαίρεση του Συστήματος σε Λειτουργίες: Οι λειτουργίες οι οποίες θεωρούμε ότι εξυπηρετεί το αναπηρικό αμαξίδιο είναι η υποστήριξη του καθήμενου και η λειτουργία της κίνησης της αναπηρικής καρέκλας. Κάθε μία από αυτές τις λειτουργίες μπορεί να αναλυθεί σε περαιτέρω απλούστερες υπολειτουργίες, και συγκεκριμένα:
  - i) Η λειτουργία της υποστήριξης αναλύεται στις υπολειτουργίες του καθίσματος και της υποστήριξης των χεριών.
  - ii) Η λειτουργία της κίνησης αναλύεται στις υπολειτουργίες της μπροστινής και πίσω κίνησης, στην στροφή, στο σταμάτημα και της επιτάχυνσης.



Εικόνα 8.1.1 Οι λειτουργίες του Αναπηρικού Αμαξιδίου.

5. Έρευνα και Επιλογή Μηχανισμών: Στη συνέχεια πραγματοποιήσαμε έρευνα για πιθανούς μηχανισμούς που θα μπορούσαν να εξυπηρετήσουν την υλοποίηση της ιδέας μας. Για τη λειτουργία της υποστήριξης επιλέξαμε να χρησιμοποιήσουμε το hub ως το κάθισμα της καρέκλας και με τη χρήση διαφόρων δομικών στοιχείων να προσομοιώσουμε την πλάτη και τα χέρια της καρέκλας. Σχετικά με τη λειτουργία της κίνησης αποφασίσαμε να πραγματοποιείται με τη χρήση τροχών και κινητήρων. Αναφορικά με τις υπολειτουργίες, όλες θα υλοποιούνται από τον ίδιο μηχανισμό κίνησης όμως θα διαφέρει ο τρόπος ενεργοποίησής τους. Δηλαδή, τα κουμπιά που βρίσκονται στο hub θα ελέγχουν την μπροστινή και πίσω κίνηση, η στροφή θα πραγματοποιείται με τη χρήση του αισθητήρα αφής, το σταμάτημα με τη χρήση του αισθητήρα απόστασης και η επιτάχυνση με τον αισθητήρα κίνησης που βρίσκεται ενσωματωμένος στο hub.
6. Δημιουργία προσχεδίου: Για τη δημιουργία του προσχεδίου όπως προαναφέραμε, χρησιμοποιήσαμε την εφαρμογή Studio 2.0 της BrickLink όπου δημιουργήσαμε ένα 3D προσχέδιο του ρομποτικού μας συστήματος. Το προσχέδιο που υλοποιήσαμε ήταν το εξής:



*Εικόνα 8.1.2 Το προσχέδιο του Αναπηρικού Αμαξιδίου.*

7. Δημιουργία Πρωτοτύπου: Στο σημείο αυτό παρουσιάζεται το πρωτότυπο που υλοποιήσαμε με βάση το προσχέδιο.



*Εικόνα 8.1.3 Το πρωτότυπο του Αναπηρικού Αμαξιδίου.*

8. Δοκιμή/Εξέταση: Κατά τη διάρκεια της δοκιμής στο αναπηρικό αμαξίδιο μας παρατηρήθηκαν τα εξής προβλήματα που έπρεπε να επανεξεταστούν: η απόσταση από το έδαφος των μπροστινών και πίσω τροχών ήταν άνιση με αποτέλεσμα το αμαξίδιο μας να παρουσιάζει μια ελαφρά προς τα μπροστά κλίση. Επιπλέον, τα

λάστιχα από τις μπροστινές ρόδες δεν επέτρεπαν στο ρομποτικό μας μηχανισμό να στρίψει με ευκολία λόγω αυξημένης τριβής.

9. Βελτίωση/Επανασχεδιασμός: Σε αυτό το στάδιο προχωρήσαμε στην επίλυση των προβλημάτων που προαναφέρθηκαν με επανασχεδιασμό του ρομποτικού μας συστήματος. Προκειμένου να λύσουμε το πρόβλημα της ισορροπίας αφαιρέσαμε τις μπροστινές ρόδες και προσθέσαμε σαν τρίτο τροχό την μπίλια που διαθέτει το kit του Spike Prime. Έτσι το ρομπότ μας ισορροπούσε καλύτερα, ενώ ταυτόχρονα μπορούσε να κινηθεί καλύτερα στις περιστροφικές κινήσεις. Νέες ρόδες προστέθηκαν σε μεγαλύτερο ύψος, ώστε να μην εμποδίζουν την κίνηση, κυρίως για λόγους εμφάνισης, ώστε να θυμίζει το ρομποτικό μας σύστημα όσο το δυνατόν περισσότερο ένα αναπηρικό αμαξίδιο. Στην παρακάτω εικόνα βλέπουμε το τελικό αποτέλεσμα.



*Εικόνα 8.1.4 Το τελικό αποτέλεσμα Αναπηρικού Αμαξιδίου.*

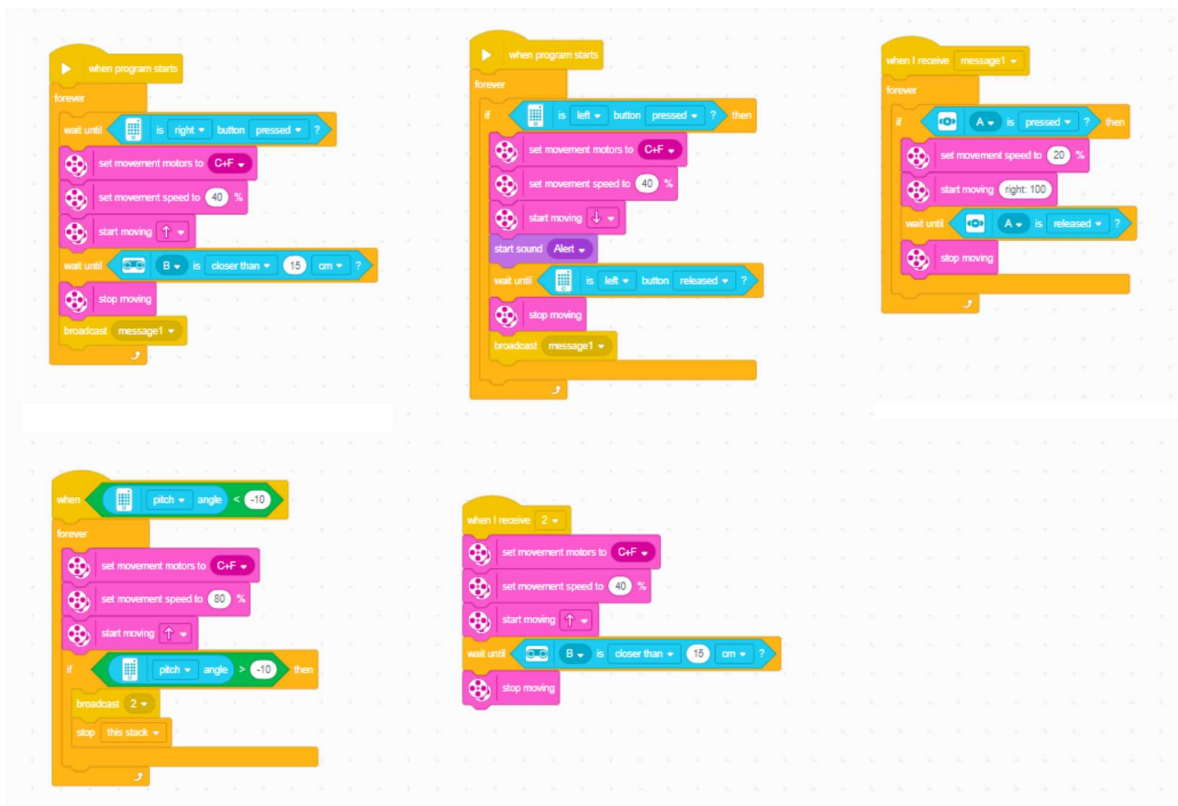


## Ανάλυση Κωδίκων:

### 1) Ανάλυση Πρώτου Κώδικα:

Για την υλοποίηση του προγράμματος επιλέξαμε να χρησιμοποιήσουμε Block-Based γλώσσα προγραμματισμού, καθώς μας δίνει την δυνατότητα να υλοποιούμε πολλά ταυτόχρονα υποπρογράμματα μέσα σε ένα πρόγραμμα, τα οποία μπορούν να εκτελούνται ταυτόχρονα.

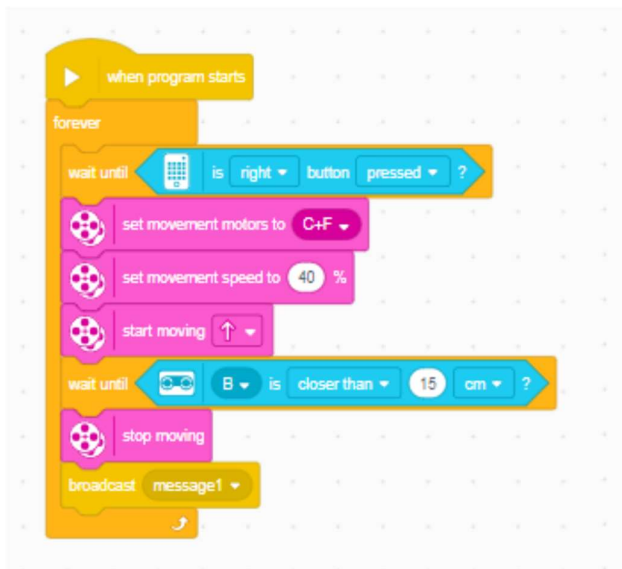
Ολοκληρωμένο το πρόγραμμα έχει την εξής μορφή:



Εικόνα 8.1.5 Ο κώδικας της πρώτης εφαρμογής του Αναπηρικού Αμαξιδίου.

Παρακάτω παρουσιάζεται η ανάλυση της κάθε στοιβάδας ξεχωριστά.

### Στοιβάδα μπροστινής κίνησης:

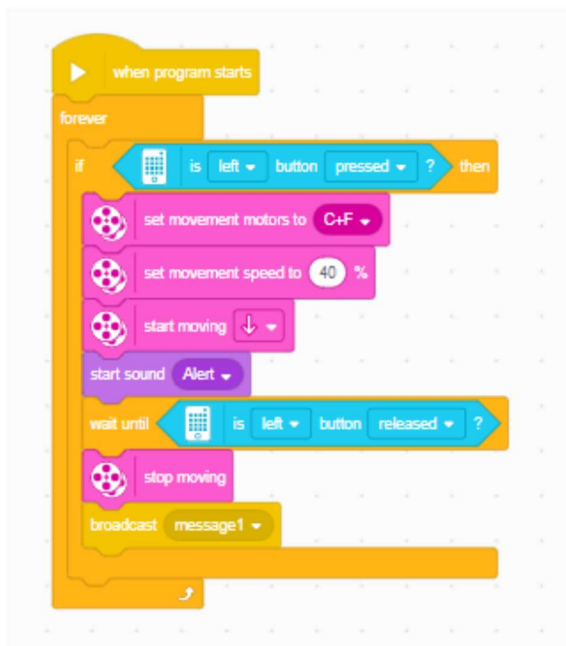


Εικόνα 8.1.6 Η στοιβάδα μπροστινή κίνησης.

Η εξής στοιβάδα ξεκινά όταν ξεκινήσει και το πρόγραμμα. Η στοιβάδα αυτή θα εκτελείται για πάντα, δηλαδή για όσο τρέχει το πρόγραμμα. Η πρώτη εντολή που δίνουμε είναι το ρομποτικό μας σύστημα να περιμένει μέχρι να πατηθεί το δεξί κουμπί. Όταν πατηθεί το δεξί κουμπί οι κινητήρες που βρίσκονται στις θέσεις C και F θα ξεκινήσουν να κινούνται με ταχύτητα 40% προς τα εμπρός. Εκτελώντας

την εντολή “start moving” η ροή του προγράμματος συνεχίζει στην επόμενη εντολή που στην ουσία ζητά από το πρόγραμμα να περιμένει μέχρι η απόσταση που μετρά ο αισθητήρας απόστασης, που βρίσκεται στην θέση «B», να γίνει μικρότερη από 15 εκατοστά πριν εκτελέσει την επόμενη εντολή. Αφού η απόσταση γίνει μικρότερη από 15 εκατοστά, το πρόγραμμα συνεχίζει στην επόμενη εντολή που απενεργοποιεί τους κινητήρες. Η επόμενη εντολή στέλνει ένα «εικονικό» μήνυμα στην στοιβάδα που ελέγχει την στροφή του οχήματος μας έτσι ώστε, αφού το αμαξίδιο μας έχει συναντήσει ένα εμπόδιο και έχει σταματήσει, να μπορέσει να στρίψει για να το αποφύγει.

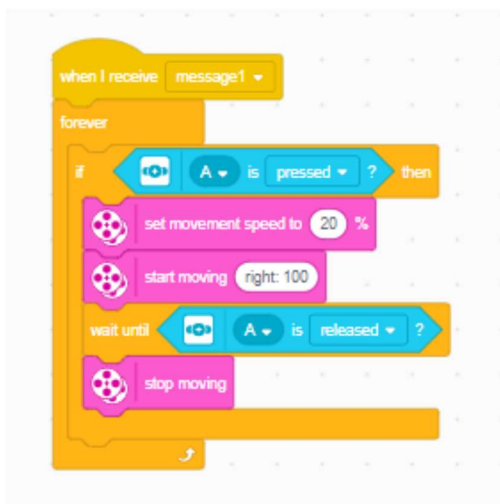
### Στοιβάδα κίνησης προς τα πίσω:



Εικόνα 8.1.7 Η στοιβάδα πίσω κίνησης.

κουμπί, τότε απενεργοποιούνται οι κινητήρες μας και ένα «εικονικό» μήνυμα αποστέλλεται ξανά στην στοιβάδα που ελέγχει την στροφή του οχήματος μας, έτσι ώστε αφού το αμαξίδιο μας έχει κινηθεί προς τα πίσω κατά την επιθυμητή απόσταση, να μπορέσει στη συνέχεια να στρίψει αν το επιθυμεί.

### Στοιβάδα περιστροφικής κίνησης:



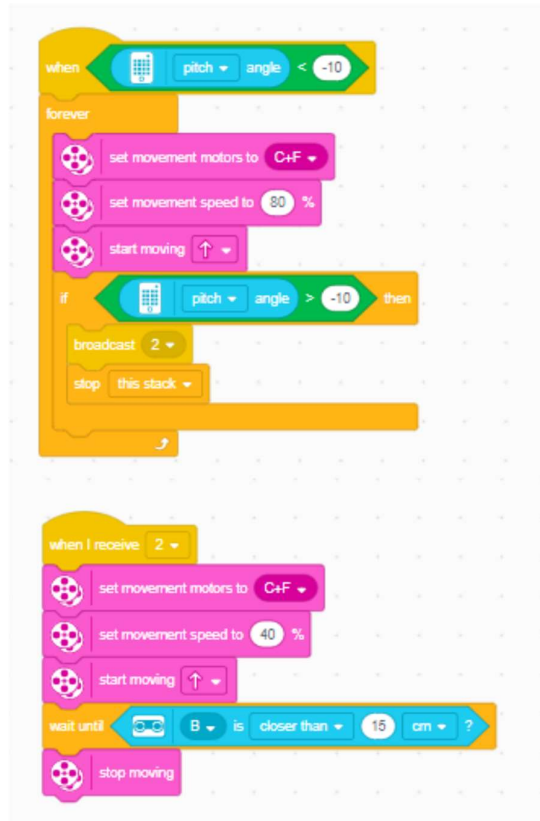
Εικόνα 8.1.8 Η στοιβάδα περιστροφικής κίνησης.

Και αυτή η στοιβάδα ξεκινά μαζί με την εκκίνηση του προγράμματος. Με την εκκίνηση του προγράμματος πραγματοποιείται για πάντα ο εξής έλεγχος: αν πατηθεί το αριστερό κουμπί τότε θα ενεργοποιηθούν ξανά οι κινητήρες που βρίσκονται στις θέσεις C και F με ταχύτητα 40% αλλά αυτή την φορά με αντίθετη κατεύθυνση, δηλαδή προς τα πίσω, και θα ενεργοποιηθεί ο ήχος “Alert”. Το κομμάτι αυτό εκτελείται μέχρι να αφησουμε ελεύθερο το αριστερό

Η στοιβάδα αυτή ξεκινάει σε περίπτωση που ληφθεί το “Message1” δηλαδή, είτε όταν το αμαξίδιο μας έχει δει ένα εμπόδιο και έχει σταματήσει, είτε όταν το όχημα μας έχει κινηθεί προς τα πίσω και έπειτα θελήσει να στρίψει. Όταν ληφθεί το “Message1” το πρόγραμμα μας θα πραγματοποιεί συνεχώς τον παρακάτω έλεγχο: αν ο πατηθεί ο Αισθητήρας Αφής που βρίσκεται στην θέση «Α» τότε θα ξεκινήσει μια κίνηση με ταχύτητα 20%

προς τα δεξιά. Η τιμή 100 που δίνεται ως όρισμα είναι αυτή που καθορίζει την τιμή της «στροφής/steering» του οχήματος, με το 100 να δηλώνει την στροφή προς τα δεξιά. Έτσι, όσο είναι πατημένος ο Αισθητήρας Αφής το όχημα θα κινείται προς τα δεξιά και θα περιμένει μέχρι να αφεθεί. Όταν αφεθεί το όχημα θα σταματήσει.

### Στοιβάδες επιταχυνόμενης κίνησης:



Εικόνα 8.1.9 Οι στοιβάδες επιταχυνόμενης κίνησης.

Σε αυτή την περίπτωση το πρόγραμμα χρησιμοποιεί τον γυροσκοπικό αισθητήρα που βρίσκεται ενσωματωμένος μέσα στο Hub. Η συγκεκριμένη στοιβάδα ξεκινά όταν η τιμή του pitch του αισθητήρα είναι μικρότερη του -10, δηλαδή έχοντας ξεκινήσει το όχημα μας πατώντας το δεξί κουμπί, αν η κλίση του pitch είναι μικρότερη από -10 (το όχημα δηλαδή βρίσκεται σε ανηφόρα), τότε οι κινητήρες που βρίσκονται στις θύρες C και F θα κινούνται πλέον προς τα μπροστά με ταχύτητα 80%. Η κίνηση αυτή θα εκτελείται συνεχώς, ενώ ταυτόχρονα συνέχεια το

πρόγραμμα μας θα ελέγχει αν η κλίση έχει γίνει μεγαλύτερη του -10, αν το όχημά μας δηλαδή έχει έρθει ξανά σε ευθεία. Όταν συμβεί αυτό, θα στείλει ένα «εικονικό» μήνυμα, το μήνυμα «2», και θα σταματήσει αυτήν την στοιβάδα. Όταν η επόμενη στοιβάδα λάβει το μήνυμα «2» θα επαναφέρει τους κινητήρες που βρίσκονται στις θύρες C και F στην αρχική τους ταχύτητα 40% και θα συνεχίζει να κινείται ευθεία. Όταν η απόσταση που μετρά ο αισθητήρας που βρίσκεται στη θύρα «B» γίνει μικρότερη από 15 εκατοστά, τότε το όχημα θα σταματήσει.

### 1) Ανάλυση Δεύτερου Κώδικα:

Αφού υλοποιήσαμε την βασική ιδέα προχωρήσαμε και στη συγγραφή ενός δεύτερου προγράμματος για το συγκεκριμένο ρομποτικό σύστημα. Το δεύτερο πρόγραμμα είναι γραμμένο στη γλώσσα προγραμματισμού Python και σκοπός του είναι να αυτοματοποιήσει το αμαξίδιό μας ως προς την αποφυγή εμποδίων που τυχόν θα συναντήσει. Παρακάτω παρουσιάζεται ολοκληρωμένος ο κώδικας:

```
1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
4
5 hub = PrimeHub()
6 motor_pair = MotorPair ('C', 'F')
7 distance_sensor = DistanceSensor ('B')
8 motor_pair.set_stop_action ('brake')
9
10 while True:
11     motor_pair.start (0,30)
12     distance_sensor.wait_for_distance_closer_than(10,'cm')
13     motor_pair.stop()
14     hub.motion_sensor.reset_yaw_angle ()
15     motor_pair.start (-100,10)
16     while True:
17         if hub.motion_sensor.get_yaw_angle () <= -90:
18             break
19     motor_pair.stop()
20     motor_pair.move (10,'cm',0,30)
21     hub.motion_sensor.reset_yaw_angle ()
22     motor_pair.start(100,10)
23     while True:
24         if hub.motion_sensor.get_yaw_angle () >= 90:
25             break
26     motor_pair.stop()
27
```

Εικόνα 8.1.10 Ο κώδικας της δεύτερης εφαρμογής του Αναπηρικού Αμαξιδίου.

### Επιμερισμός του κώδικα και ανάλυση:

```
1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
4
```

Εικόνα 8.1.11 Εισαγωγή Βιβλιοθηκών και κλάσεων για το Αναπηρικό Αμαξίδιο.

Πρωταρχική δουλειά στον προγραμματισμό με Python είναι η εισαγωγή των βιβλιοθηκών και των κλάσεων που θα χρειαστούμε στην κάθε εφαρμογή. Ανοίγοντας την εφαρμογή του Spike και επιλέγοντας τον προγραμματισμό με Python μας εμφανίζει ένα έτοιμο πρόγραμμα στο οποίο έχει εισάγει ήδη για εμάς όλες τις κλάσεις και τις βιβλιοθήκες που θα χρειαστούμε. Στην γραμμή 1 από την βιβλιοθήκη Spike εισάγουμε τις κλάσεις που αφορούν τα ηλεκτρονικά εξαρτήματα του Spike. Στην συνέχεια από την βιβλιοθήκη spike.control εισάγονται 3 κλάσεις που αφορούν κυρίως τον χρόνο. Τέλος, από τη βιβλιοθήκη math εισάγει όλες τις κλάσεις. (Το συγκεκριμένο κομμάτι

κώδικα θα χρησιμοποιηθεί χωρίς καμία διαφορά σε όλες τις εφαρμογές που έχουν υλοποιηθεί στην γλώσσα προγραμματισμού Python γι' αυτό και στις επόμενες εφαρμογές η ανάλυση του θα παραλείπεται.) Στη συνέχεια ακολουθεί η δημιουργία των αντικειμένων που θα χρειαστούμε και η αρχικοποίηση τιμών.

```
5 hub = PrimeHub()
6 motor_pair = MotorPair ('C', 'F')
7 distance_sensor = DistanceSensor ('B')
8 motor_pair.set_stop_action ('brake')
```

*Εικόνα 8.1.12 Δημιουργία Αντικειμένων για το Αναπηρικό Αμαξίδιο.*

Για τη δημιουργία ενός αντικείμενου χρησιμοποιώ μια μεταβλητή στην οποία αναθέτω την κλάση της οποία θέλω να δημιουργήσω ένα αντικείμενο και στο οποίο μπορώ να χρησιμοποιήσω τις ιδιότητες της. Έτσι σε αυτή την εφαρμογή δημιουργούμε τρία αντικείμενα: χρησιμοποιώντας την μεταβλητή “hub” και την κλάση “PrimeHub” δημιουργούμε ένα αντικείμενο για το Hub του συστήματος. Σε μια μεταβλητή που ονομάσαμε “motor\_pair” αναθέτουμε την κλάση “MotorPair” η οποία μας επιτρέπει να χρησιμοποιήσουμε δύο κινητήρες σαν να είναι ένα ζευγάρι κινητήρων και μέσα στην παρένθεση δίνουμε ως τιμές τις θύρες στις οποίες βρίσκονται οι κινητήρες μας, στην προκειμένη περίπτωση τις θύρες “C” και “F”. Τέλος, το τρίτο αντικείμενο που δημιουργούμε είναι ο αισθητήρας απόστασης με την χρήση της μεταβλητής “distance\_sensor” και την κλάση “DistanceSensor” όπου δίνουμε ξανά μέσα στην παρένθεση την θύρα που είναι συνδεδεμένος ο αισθητήρας μας.

Έχοντας δημιουργήσει πλέον το αντικείμενο μας για το ζευγάρι των κινητήρων της κλάσης MotorPair μπορούμε να χρησιμοποιήσουμε σε αυτό τις μεθόδους που υλοποιεί η κλάση. Χρησιμοποιούμε λοιπόν την μέθοδο “set\_stop\_action” για να ορίσουμε τον τρόπο με τον οποίο θα σταματάει το όχημα μας και δίνουμε ως τιμή στο όρισμα της την λέξη “brake” ορίζοντας έτσι ότι το όχημα μας θα σταματάει με φρενάρισμα.

```

9
10 while True:
11     motor_pair.start (0,30)
12     distance_sensor.wait_for_distance_closer_than(10,'cm')
13     motor_pair.stop()
14     hub.motion_sensor.reset_yaw_angle ()
15     motor_pair.start (-100,10)
16     while True:
17         if hub.motion_sensor.get_yaw_angle () <= -90:
18             break
19     motor_pair.stop()
20     motor_pair.move (10,'cm',0,30)
21     hub.motion_sensor.reset_yaw_angle ()
22     motor_pair.start(100,10)
23     while True:
24         if hub.motion_sensor.get_yaw_angle () >= 90:
25             break
26     motor_pair.stop()
27

```

*Εικόνα 8.1.13 Το βασικό κομμάτι του προγράμματος του Αναπηρικού Αμαξιδίου.*

Μπαίνοντας στο βασικό κομμάτι του κώδικα αρχικά δημιουργούμε μια επανάληψη “while True” προκειμένου το πρόγραμμα μας να εκτελεί συνεχώς τα κομμάτια του κώδικα που βρίσκονται μέσα σε αυτή. Η εσοχή που προηγείται όλων των εντολών που έπονται δηλώνει ότι είναι κομμάτια της επανάληψης και θα εκτελούνται συνεχώς. Έπειτα στο ζεύγος των κινητήρων καλούμε την μέθοδο “start” η οποία δέχεται σε αυτή την περίπτωση δύο ορίσματα: την τιμή της πηδαλιούχησης/steering, και την ταχύτητα με την οποία θα κινηθούν οι κινητήρες. Έχοντας ορίσει την τιμή της στροφής ως 0, αυτό υποδηλώνει πως το όχημα μας θα κινηθεί προς τα εμπρός, με ταχύτητα 30. Στην συνέχεια στον αισθητήρα απόστασης καλούμε την μέθοδο “wait\_for\_distance” που κάνει την ροή του προγράμματος μας να περιμένει σε αυτό το σημείο μέχρι η απόσταση που μετρά ο αισθητήρας να είναι μικρότερη των 10 εκατοστών. Όταν η απόσταση γίνει μικρότερη των 10 εκατοστών (δηλαδή το όχημα έχει συναντήσει εμπόδιο) καλώντας την μέθοδο “stop” στο ζεύγος των κινητήρων μας σταματάμε την κίνηση του αμαξιδίου μας.

Για την αποφυγή του εμποδίου το πρώτο πράγμα που θα χρειαστεί να κάνουμε είναι να επαναφέρουμε την τιμή του yaw του αισθητήρα κίνησης σε 0 προκειμένου να είμαστε σίγουροι ότι το όχημα μας θα περιστραφεί για 90 μοίρες. Για να το καταφέρουμε αυτό χρησιμοποιούμε τη μέθοδο “reset\_yaw\_angle” στον γυροσκοπικό



αισθητήρα που βρίσκεται ενσωματωμένος μέσα στο Hub και έπειτα ζητάμε από το όχημα μας να ξεκινήσει να περιστρέφεται με στροφή/steering -100, δηλαδή προς τα αριστερά με ταχύτητα 10. Με τη χρήση αυτής της εντολής, το όχημα μας αρχίζει να κινείται προς τα αριστερά ενώ ταυτόχρονα η ροή του προγράμματος συνεχίζει στην επόμενη εντολή. Στη συνέχεια δημιουργούμε μια νέα επανάληψη της μορφής “while True”. Με αυτό τον τρόπο «εγκλωβίζουμε» την ροή του προγράμματος σε αυτό το σημείο και μέσα σε αυτή ζητάμε από το πρόγραμμα να πραγματοποιεί συνεχώς έναν έλεγχο: Αν η τιμή που παίρνει από τον αισθητήρα κίνησης γίνει μικρότερη ή ίση με 90 μοίρες τότε να κάνει “break” από την επανάληψη, δηλαδή με αυτόν τον τρόπο η ροή του προγράμματος βγαίνει μέσα από την “while True” και συνεχίζει στην επόμενη εντολή που σταματά το ζεύγος των κινητήρων μας.

Έχοντας σταματήσει και περιστρέψει το αμαξίδιο μας για 90 μοίρες προς τα αριστερά το δεύτερο πράγμα που θα χρειαστεί να κάνουμε για να αποφύγουμε το εμπόδιο είναι μετακινήσουμε το αμαξίδιο μας προς τα εμπρός για 10 εκατοστά. Σε αυτή την περίπτωση καλούμε την μέθοδο “move” η οποία δέχεται τέσσερα ορίσματα: τον αριθμό που θέλει να κινηθεί το όχημα μας, την μονάδα μέτρησης με τη μορφή συμβολοσειράς, την πηδαλιούχηση/steering και την ταχύτητα με την οποία θα κινηθεί. Έτσι, αφού πραγματοποιήσουμε μια κίνηση προς τα εμπρός για 10 εκατοστά, με ταχύτητα 30 ζητάμε ξανά από το πρόγραμμα μας να επαναφέρει την τιμή του yaw του αισθητήρα κίνησης σε 0 και έπειτα ζητάμε από το όχημα μας να ξεκινήσει να περιστρέφεται με πηδαλιούχηση/steering 100, δηλαδή προς τα δεξιά, με ταχύτητα 10. Με τον ίδιο τρόπο που υλοποιήσαμε την στροφή 90 μοιρών προς τα αριστερά, δηλαδή με μια επανάληψη της μορφής “while True”, μια συνθήκη ελέγχου των μοιρών όπου σε αυτή την περίπτωση ελέγχουμε αν η τιμή του yaw είναι μεγαλύτερη ή ίση των 90 μοιρών και την χρήση της εντολής “brake” υλοποιούμε ξανά μια στροφή προς τα δεξιά αυτή τη φορά. Όταν η τιμή του yaw γίνει μεγαλύτερη ή ίση του 90 τότε η ροή του προγράμματος θα συνεχίσει στην επόμενη εντολή που απενεργοποιεί τον κινητήρα μας.

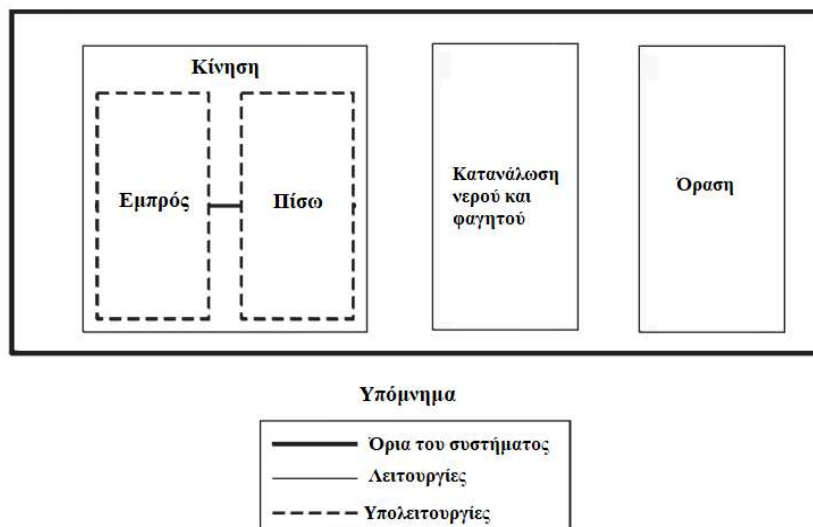
Τέλος, λόγω του ότι το πρόγραμμα βρίσκεται μέσα σε μια επανάληψη της μορφής “while True” θα ξεκινήσει ξανά από την αρχή ενεργοποιώντας ξανά τους κινητήρες και ελέγχοντας για προς αποφυγήν εμπόδια.



## 8.2 Ρομποτική γάτα.

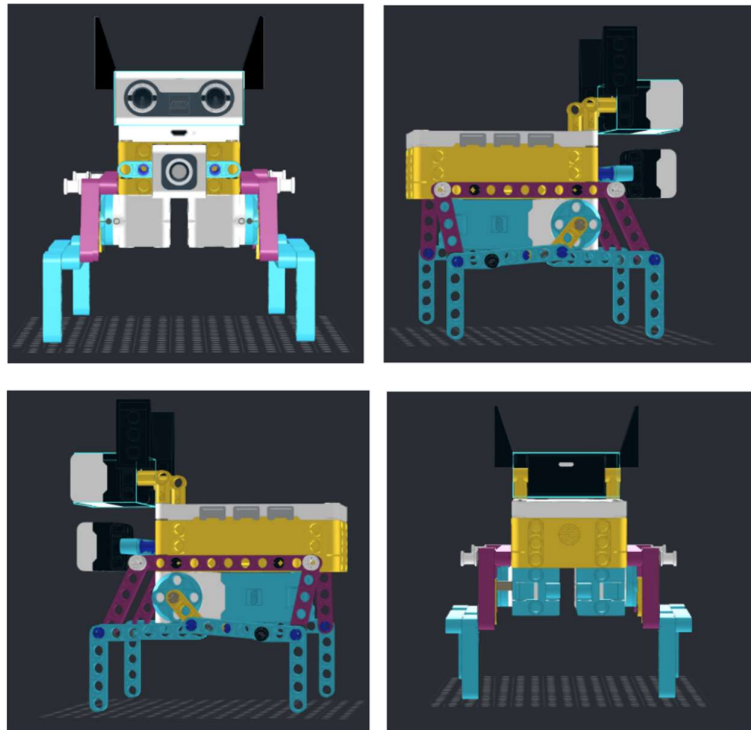
### Διαδικασία Σχεδιασμού:

1. Ορισμός Προβλήματος: Ως πρόβλημα της συγκεκριμένης ενότητας ορίζεται η έλλειψη υπευθυνότητας και η ασέβεια απέναντι στα κατοικίδια ζώα.  
Εξέταση και Ανάλυση του Προβλήματος: Οι πτυχές του προβλήματος που εντοπίσαμε και αποφασίσαμε πως χρήζουν αντιμετώπισης είναι οι εξής: η παραμέληση των κατοικίδιων ζώων και η μη κάλυψη των βασικών τους αναγκών, όπως επίσης και η έλλειψη ενσυναίσθησης προς τις επιθυμίες τους.
2. Καταιγισμός Ιδεών και Σύλληψη Πιθανής Ιδέας: Με βάση τα δύο προηγούμενα στάδια καταλήξαμε στην ιδέα υλοποίησης ενός ρομποτικού κατοικίδιου, και πιο συγκεκριμένα μιας ρομποτικής γάτας, η οποία θα χρήζει φροντίδας από τον χρήστη, ο οποίος θα πρέπει να καλύπτει τις βασικές της ανάγκες για τροφή, νερό και διασκέδαση. Επιπρόσθετα, αποφασίσαμε την υλοποίηση μιας εφαρμογής σχετικής με τα συναισθήματα του κατοικίδιου.
3. Διαίρεση του Συστήματος σε Λειτουργίες: Οι λειτουργίες οι οποίες θεωρούμε ότι εξυπηρετεί η ρομποτική μας γάτα είναι η κίνηση, η λειτουργία της κατανάλωσης τροφής και νερού, και της όρασης. Από αυτές, η λειτουργία της κίνησης μπορεί να διαιρεθεί σε στις υπολειτουργίες της μπροστινής και πίσω κίνησης.



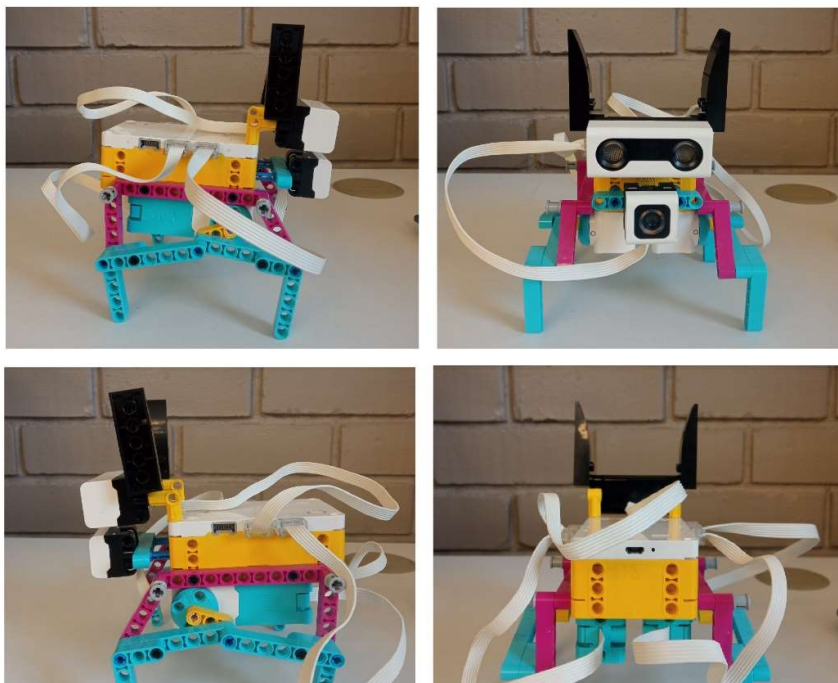
Εικόνα 8.2.1 Οι λειτουργίες της Ρομποτικής Γάτας.

4. Έρευνα και Επιλογή Μηχανισμών: Στη συνέχεια πραγματοποιήσαμε έρευνα για πιθανούς μηχανισμούς που θα μπορούσαν να εξυπηρετήσουν την υλοποίηση της ιδέας μας. Για τη λειτουργία της κατανάλωσης της τροφής επιλέξαμε να χρησιμοποιήσουμε τον αισθητήρα χρώματος, όπου κάθε χρώμα θα ικανοποιεί και μια από τις λειτουργίες, ενώ για την λειτουργία της όρασης επιλέξαμε να χρησιμοποιήσουμε τον αισθητήρα απόστασης. Σχετικά με τη λειτουργία της κίνησης, αποφασίσαμε η μπροστινή και πίσω κίνηση να πραγματοποιούνται με τη δημιουργία ενός μηχανικού μοχλού, μιας μανιβέλας, που θα κινεί το ζώο μας κατά τέτοιο τρόπο ώστε να θυμίζει όσο το δυνατό περισσότερο την κίνηση των ποδιών.
5. Δημιουργία προσχεδίου: Για τη δημιουργία του προσχεδίου χρησιμοποιήσαμε την εφαρμογή Studio 2.0 της BrickLink όπου δημιουργήσαμε ένα 3D προσχέδιο του ρομποτικού μας συστήματος. Το προσχέδιο που υλοποιήσαμε ήταν το εξής:



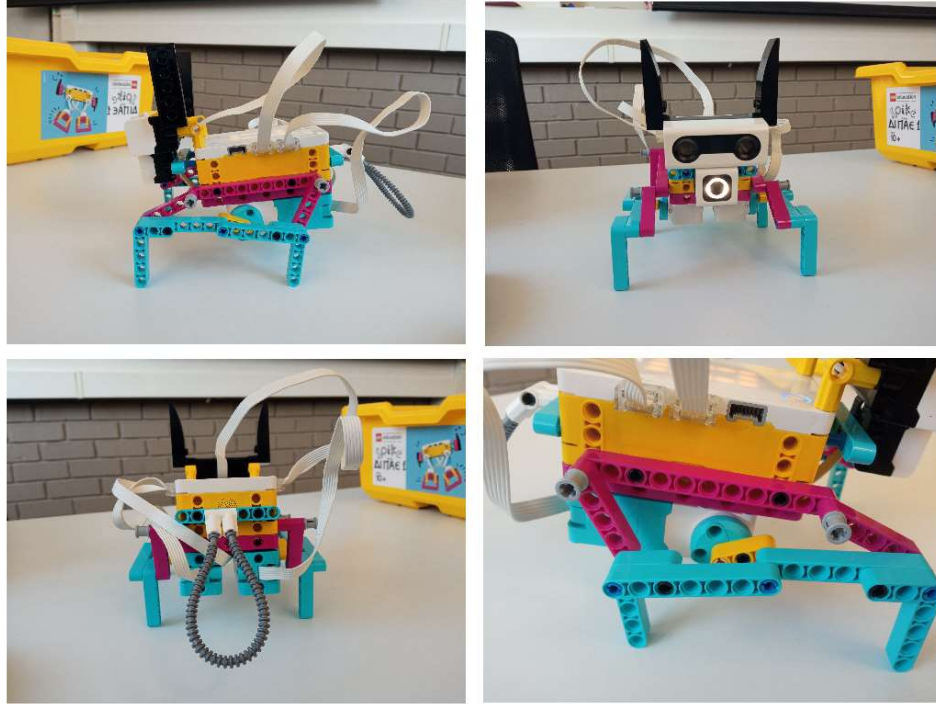
Εικόνα 8.2.2 Το προσχέδιο της Ρομποτικής Γάτας

6. Δημιουργία Πρωτοτύπου: Στο σημείο αυτό παρουσιάζεται το πρωτότυπο που υλοποιήσαμε με βάση το προσχέδιο.



*Εικόνα 8.2.3 Το πρωτότυπο της Ρομποτικής Γάτας.*

7. Δοκιμή/Εξέταση: Κατά τη διάρκεια της δοκιμής της ρομποτικής μας γάτας παρατηρήθηκε πως η απόσταση των κινητήρων με την ροζ δοκό 11 οπών που βρίσκεται πάνω στο Hub ήταν αρκετά κοντινή. Αυτό είχε ως αποτέλεσμα η κίτρινη δοκός 3 οπών που δίνει την κίνηση στα πόδια να μην μπορεί να περιστραφεί ελεύθερα. Έτσι, το ρομποτικό μας σύστημα δεν μπορούσε να εκτελέσει ούτε μπροστινή ούτε προς τα πίσω κίνηση.
8. Βελτίωση/Επανασχεδιασμός: Για την επίλυση του προαναφερθέντος προβλήματος επιλέξαμε να αυξήσουμε την απόσταση των κινητήρων με την ροζ δοκό 11 οπών χρησιμοποιώντας ένα ορθογώνιο πλαίσιο 7x11 οπών το οποίο μας έδωσε το απαραίτητο ύψος που χρειαζόμασταν για να μπορέσει να κινηθεί η ρομποτική γάτα μας. Προς βελτίωση της εμφάνισης της επιλέξαμε να τοποθετήσουμε σε ψηλότερο ύψος των αισθητήρα χρώματος καθώς και να προσθέσουμε μια ουρά στο πίσω μέρος. Το τελικό αποτέλεσμα ήταν το εξής:



*Εικόνα 8.2.4 Το τελικό αποτέλεσμα της Ρομποτικής Γάτας.*

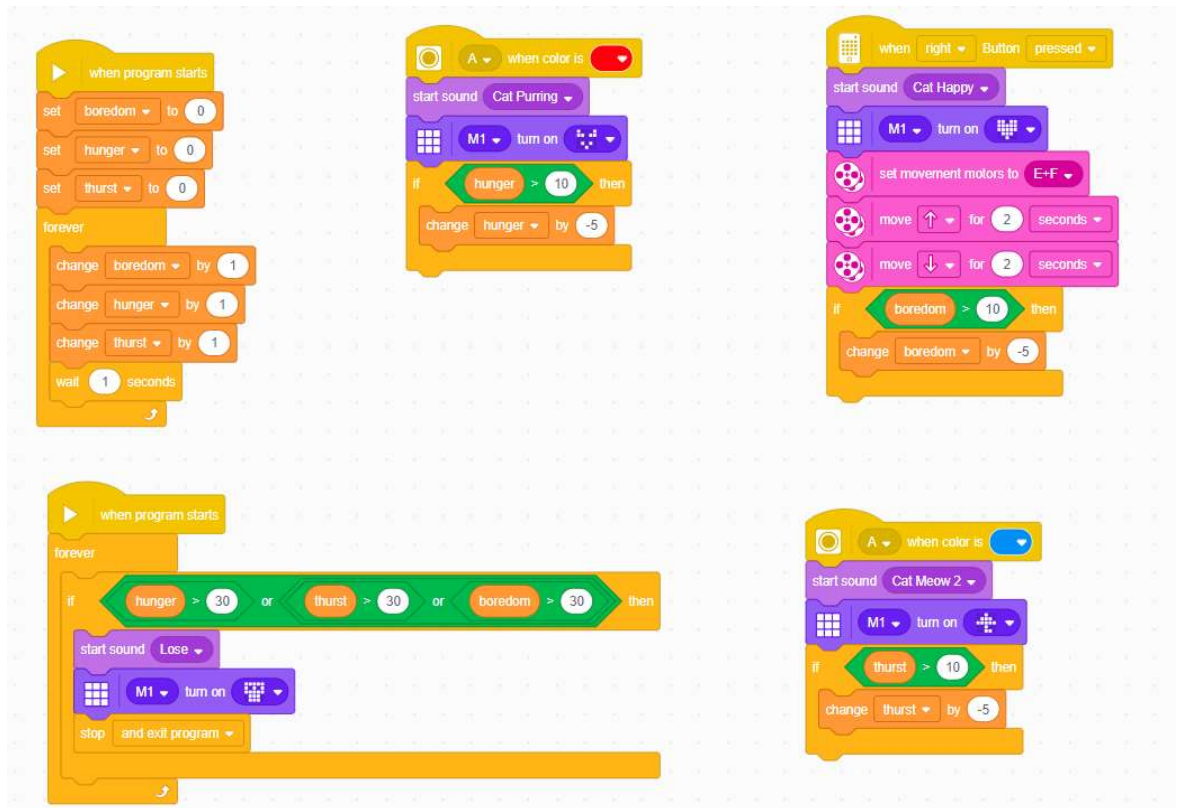
### Ανάλυση Κωδίκων:

Για την υλοποίηση των προγραμμάτων επιλέξαμε να χρησιμοποιήσουμε Block-Based γλώσσα προγραμματισμού καθώς μας δίνει την δυνατότητα να υλοποιούμε πολλά ταυτόχρονα υποπρογράμματα μέσα σε ένα πρόγραμμα τα οποία μπορούν να εκτελούνται ταυτόχρονα. Επιπρόσθετα μας δίνει την δυνατότητα να εκτελούμε αλλαγές με παράλληλους ελέγχους στα προγράμματα μας.

#### *1) Ανάλυση Πρώτου Κώδικα:*

Ο πρώτος μας κώδικας αποτελεί μια παραλλαγή ενός ψηφιακού παιχνιδιού όπου ο χρήστης πρέπει να φροντίσει ένα εικονικό κατοικίδιο.

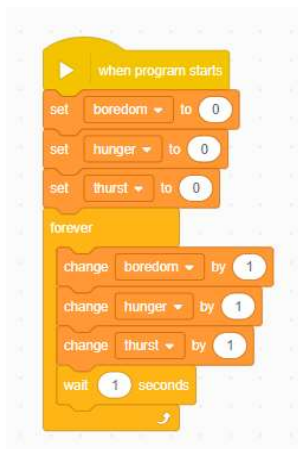
Ολοκληρωμένο το πρόγραμμα έχει την εξής μορφή:



Εικόνα 8.2.5 Το πρώτο πρόγραμμα της Ρομποτικής Γάτας.

Παρακάτω παρουσιάζεται η ανάλυση της κάθε στοιβάδας ξεχωριστά.

### Στοιβάδα Μεταβλητών:



Εικόνα 8.2.6 Αρχικοποίηση Μεταβλητών.

Αφού δημιουργήσαμε τρεις μεταβλητές με τις ονομασίες “boredom”, “hunger”, “thurst” περάσαμε στην αρχικοποίηση τους. Έτσι ορίσαμε πως και οι τρεις μεταβλητές θέλουμε να έχουν τιμή 0 όταν ξεκινήσει το πρόγραμμα μας. Στη συνέχεια, εισαγάγαμε μια επανάληψη της μορφής “forever” όπου κάθε μια από τις μεταβλητές θα αλλάζει κατά μια μονάδα κάθε ένα δευτερόλεπτο. Για να μην αλλάζουν συνεχώς οι μεταβλητές, αλλά κατά ένα δευτερόλεπτο όπως θέλουμε εμείς προσθέσαμε την εντολή “wait 1 seconds” η οποία καθυστερεί το πρόγραμμά μας με βάση τον χρόνο που έχουμε ορίσει. Η αλλαγή αυτή των

μεταβλητών θα εκτελείται συνεχώς μέχρι το τέλος του προγράμματος και παράλληλα με τις άλλες στοιβάδες που έχουμε δημιουργήσει.

#### Στοιβάδα Κατανάλωσης Τροφής:



Εικόνα 8.2.7 Στοιβάδα Κατανάλωσης Τροφής.

Η συγκεκριμένη στοιβάδα χρησιμεύει προκειμένου να ταΐσουμε την ρομποτική μας γάτα. Ενεργοποιείται κάθε φορά που ο αισθητήρας χρώματος θα εντοπίσει το κόκκινο χρώμα, δηλαδή μόνο σε αυτή την περίπτωση θα εκτελεστούν οι εντολές που βρίσκονται κάτω από αυτή. Έτσι, όταν ο αισθητήρας εντοπίσει το κόκκινο χρώμα θα ξεκινήσει ο ήχος “Cat Purring” και το πρόγραμμα θα συνεχίσει στις παρακάτω εντολές (δεν χρειάζεται να έχει ολοκληρωθεί ο ήχος). Έπειτα θα δείξει ένα εικονίδιο γάτας στο Light Matrix του

hub και αν η μεταβλητή “hunger” εκείνη τη στιγμή που ο αισθητήρας βλέπει το κόκκινο χρώμα είναι μεγαλύτερη από 10 τότε θα την μειώσει κατά 5 μονάδες.

#### Στοιβάδα Κατανάλωσης Νερού:



Εικόνα 8.2.8 Στοιβάδα Κατανάλωσης Νερού.

Με παρόμοιο τρόπο λειτουργεί και το κομμάτι του κώδικα που ελέγχει τη δίψα της γάτα μας. Η στοιβάδα ενεργοποιείται όταν ο αισθητήρας εντοπίσει το μπλε χρώμα και η πρώτη εντολή που εκτελείται είναι η εντολή ήχου. Αφού ξεκινήσει ο ήχος “Cat Meow 2” τότε, στο Light Matrix, εμφανίζεται ένα εικονίδιο ομπρέλας, και αν η μεταβλητή “thirst” είναι μεγαλύτερη του 10 τότε η τιμή της θα μειωθεί κατά 5.



### Στοιβάδα Διασκέδασης:

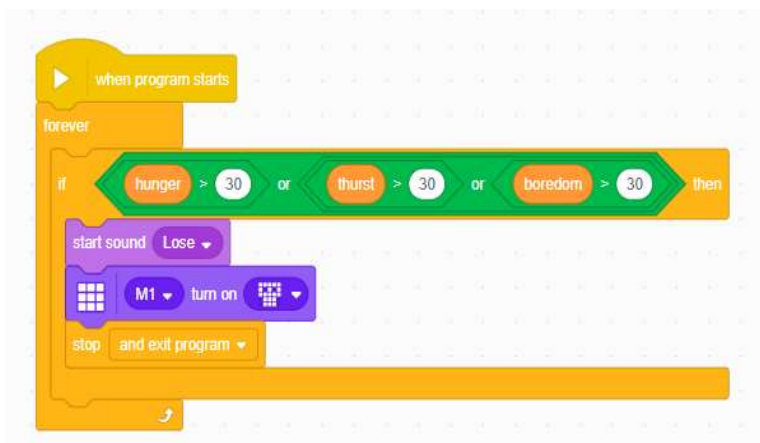


Εικόνα 8.2.9 Στοιβάδα Διασκέδασης.

Αυτή η στοιβάδα ενεργοποιείται όταν ο χρήστης πατήσει το δεξί κουμπί. Τότε ξεκινάει ένας ήχος χαρούμενης γάτα και στη συνέχεια απεικονίζεται μια καρδιά στην οθόνη του Hub. Στη συνέχεια η ρομποτική γάτα εκτελεί μια μικρή κίνηση προς τα μπροστά για 2 δευτερόλεπτα και επιστρέφει πίσω στην αρχική της θέση εκτελώντας ακόμη μια κίνηση 2 δευτερολέπτων αυτή τη φορά προς την αντίθετη κατεύθυνση. Αν η μεταβλητή “boredom” είναι μεγαλύτερη του 10 εκείνη τη στιγμή τότε αλλάζει κατά 5. (Για την κατανόηση του κώδικα είναι σημαντικό να θυμάται κανείς πως η

αλλαγή των μεταβλητών συμβαίνει κατά 1 δευτερόλεπτο οπότε η τιμή της μεταβλητής διαφέρει σε διάφορες φάσεις του προγράμματος).

### Στοιβάδα Τερματισμού του Παιχνιδιού:



Εικόνα 8.2.10 Στοιβάδα Τερματισμού του Παιχνιδιού.

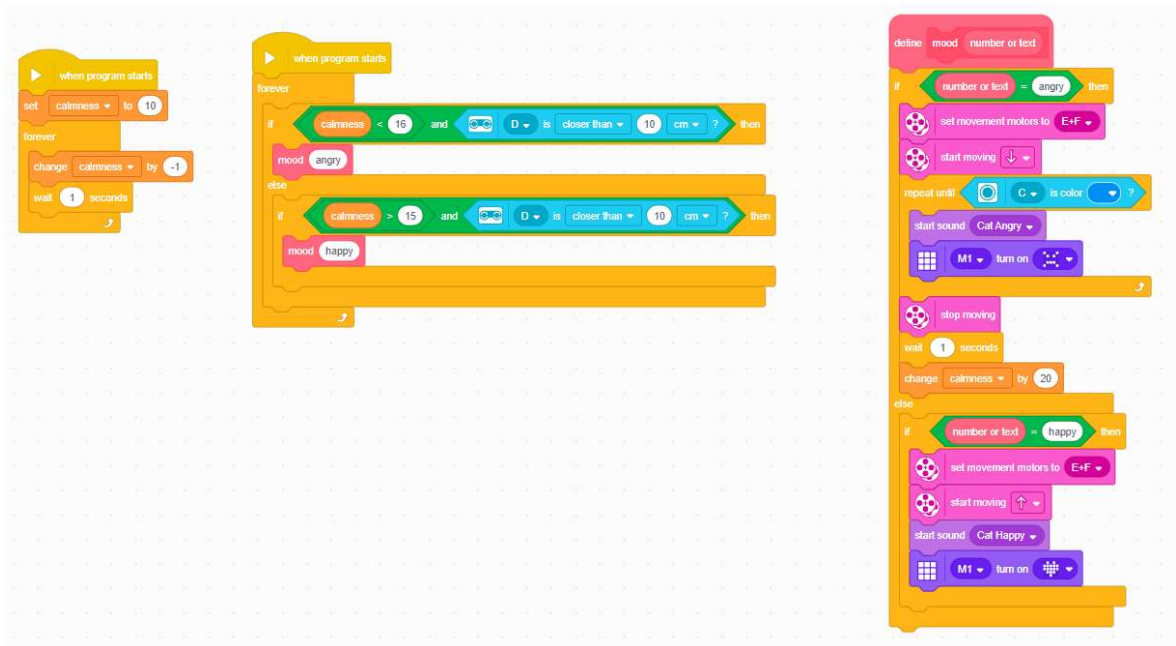
Η στοιβάδα αυτή χρησιμοποιείται για να ελέγχει διαρκώς την τιμή των μεταβλητών. Η εντολή “forever” χρησιμοποιείται ώστε να γίνεται διαρκώς ο εξής έλεγχος: αν κάποια από τις τρεις

μεταβλητές που έχουμε δημιουργήσει ξεπεράσει την τιμή 30, τότε ξεκινάει ένας ήχος «χασίματος», στην οθόνη του hub ενεργοποιείται ένα εικονίδιο νεκροκεφαλής και μετά, με τη χρήση της εντολής “stop and exit program” τερματίζεται το πρόγραμμα και

κατ' επέκταση το παιχνίδι μας. Φυσικά τα όρια των μεταβλητών μπορούν να τροποποιηθούν κι έτσι το παιχνίδι μας να διαρκεί περισσότερο ή να γίνει πιο δύσκολο.

### 1) Ανάλυση Πρώτου Κώδικα:

Ολοκληρωμένο το Πρόγραμμα έχει αυτή τη μορφή:



Εικόνα 8.2.11 Το δεύτερο πρόγραμμα της Ρομποτικής Γάτας.

Παρακάτω παρουσιάζεται η ανάλυση της κάθε στοιβάδας ξεχωριστά.

### Στοιβάδα Ορισμού Ηρεμίας:



Εικόνα 8.2.12 Η Στοιβάδα Ορισμού Ηρεμίας

Αυτή η στοιβάδα ξεκινά με την εκκίνηση του προγράμματος. Συνεπώς, με το που ξεκινά το πρόγραμμα η μεταβλητή μας που ορίζει την ηρεμία της γάτας (calmness) έχει την τιμή 10. Στη συνέχεια πραγματοποιούμε μια συνεχή επανάληψη όπου κάθε ένα δευτερόλεπτο το επίπεδο της ηρεμίας αλλάζει κατά -1.



### Στοιβάδα ορισμού διαθέσεων:



Εικόνα 8.2.13 Η Στοιβάδα ορισμού διαθέσεων.

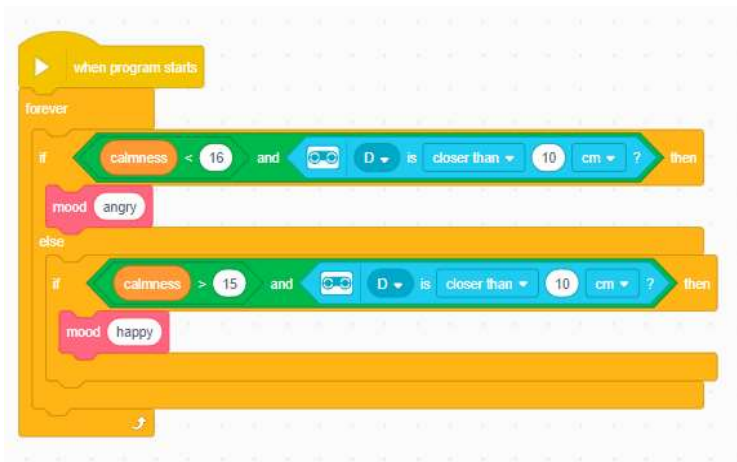
Σε αυτό το σημείο του κώδικα δημιουργούμε μια μεταβλητή η οποία ονομάζεται “mood” και δέχεται ένα όρισμα τύπου συμβολοσειρά ή αριθμός. Πρόκειται για μια δομή ελέγχου if-else η οποία ελέγχει την τιμή που θα δοθεί στο όρισμα από τον χρήστη. Πιο συγκεκριμένα:

- Εάν η τιμή που θα δοθεί είναι “angry” τότε ενεργοποιούνται οι κινητήρες “E” και “F” και το ρομποτικό σύστημα ξεκινά να κινείται προς τα πίσω. Στη συνέχεια μπαίνουμε σε έναν βρόγχο επανάληψης όπου η ρομποτική γάτα συνεχίζει να κινείται προς τα πίσω, παράγοντας έναν ήχο νευριασμένης γάτας, ενώ ταυτόχρονα ενεργοποιείται ένα θυμωμένο εικονίδιο στο Light Matrix του Hub. Ο βρόγχος αυτός τερματίζεται όταν ο αισθητήρας χρώματος δει μπροστά του το μπλε χρώμα (το οποίο υποθέτουμε ότι είναι ένα παιχνίδι που αρέσει στη γάτα). Η ροή του προγράμματος περνάει στην επόμενη εντολή όπου

σταματάει την κίνηση των κινητήρων, περιμένει για 1 δευτερόλεπτο και αυξάνει την τιμή της μεταβλητής “calmness” κατά 20.

- Εάν η τιμή που έχει δοθεί στην συνάρτηση είναι “happy” τότε ενεργοποιούνται ξανά οι κινητήρες “E” και “F” και η ρομποτική γάτα ξεκινάει να κινείται προς τα εμπρός παίζοντας έναν ήχο χαρούμενης γάτας και στην οθόνη του Hub εμφανίζεται πλέον ένα νέο εικονίδιο, μια καρδιά.

### Στοιβάδα επιλογής διάθεσης:



Εικόνα 8.2.14 Η Στοιβάδα επιλογής διάθεσης.

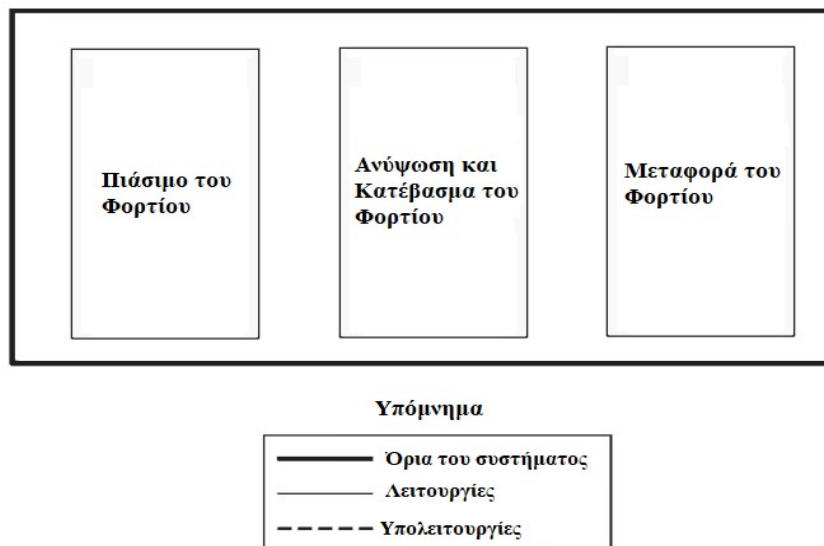
Και αυτή η στοιβάδα ξεκινά μαζί με το πρόγραμμα. Πρόκειται για έναν ατέρμονο βρόγχο επανάληψης όπου ελέγχει συνεχώς το επίπεδο της ηρεμίας της γάτας, δηλαδή την μεταβλητή “calmness” και ταυτόχρονα την τιμή της απόστασης

που μετρά ο αισθητήρας απόστασης. Εάν το επίπεδο της ηρεμίας είναι μικρότερο του 16 και η τιμή της απόστασης είναι μικρότερη από 10 εκατοστά, το πρόγραμμα καλεί την συνάρτηση mood με τιμή ορίσματος “angry” ενώ, εάν το επίπεδο της ηρεμίας είναι μεγαλύτερο του 15 και η απόσταση μικρότερη από 10 εκατοστά, τότε καλείται ξανά η συνάρτηση αλλά με την τιμή “happy”.

### 8.3 Ρομποτικός Βραχίονας.

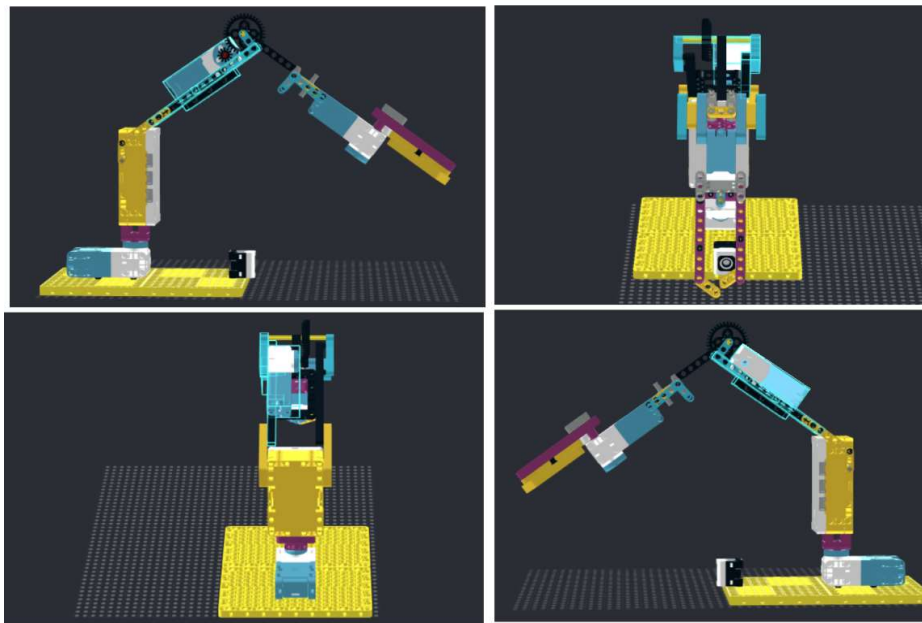
#### Διαδικασία Σχεδιασμού:

1. Ορισμός Προβλήματος: Ως πρόβλημα της συγκεκριμένης ενότητας ορίζεται η επίπονη εργασία στη μεταφορά φορτίων για τους εργαζόμενους μιας βιομηχανίας.
2. Εξέταση και Ανάλυση του Προβλήματος: Οι πτυχές του προβλήματος που εντοπίσαμε και αποφασίσαμε πως χρήζουν αντιμετώπισης είναι οι εξής: η δυσκολία του ανθρώπου να εκτελεί συνεχώς επαναλαμβανόμενες κινήσεις και οι περιορισμοί στην ανθρώπινη αντοχή, το ζήτημα του κινδύνου της σωματικής ακεραιότητας σε περιπτώσεις μεγάλου φορτίου ή μεγάλης απόστασης από το σημείο μεταφοράς και η ασφάλεια του φορτίου κατά τη μεταφορά.
3. Καταιγισμός Ιδεών και Σύλληψη Πιθανής ιδέας: Με βάση τα προηγούμενα στάδια καταλήξαμε στην ιδέα υλοποίησης ενός ρομποτικού βραχίονα, ικανού να εκτελεί επαναλαμβανόμενες μεταφορές φορτίων, ανυψώνοντας τα ελάχιστα προκειμένου να αποφευχθεί η τριβή τους με το έδαφος και έτσι να εξασφαλιστεί η ασφάλεια τους.
4. Διαίρεση του Συστήματος σε Λειτουργίες: Οι λειτουργίες οι οποίες θεωρούμε ότι πραγματοποιεί ο ρομποτικός βραχίονας είναι το πιάσιμο του φορτίου, η ανύψωση και το κατέβασμα του φορτίου, και η μεταφορά του στο επιθυμητό σημείο.



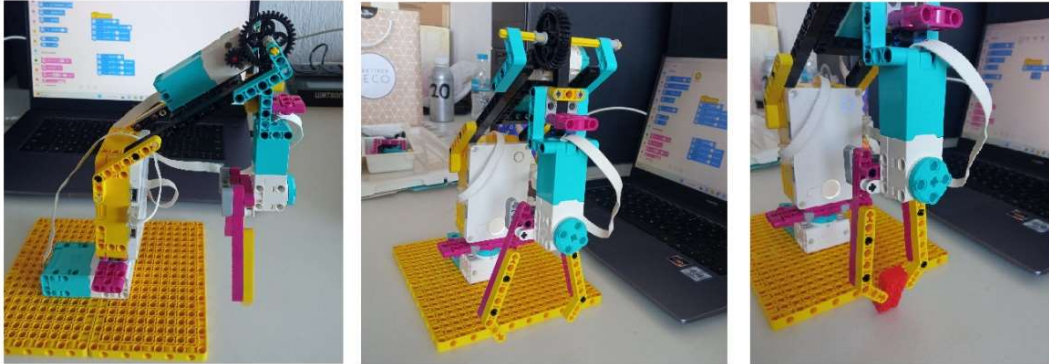
Εικόνα 8.3.1 Οι λειτουργίες του Ρομποτικού Βραχίονα.

5. Έρευνα και Επιλογή Μηχανισμών: Στη συνέχεια πραγματοποιήσαμε έρευνα για πιθανούς μηχανισμούς που θα μπορούσαν να εξυπηρετήσουν την υλοποίηση της ιδέας μας. Για τη λειτουργία του πιασίματος του φορτίου επιλέξαμε να κατασκευάσουμε μια δαγκάνα η οποία, με τη βοήθεια ενός κινητήρα θα ανοίγει και θα κλείνει προκειμένου να πιάσει το αντικείμενο. Σχετικά με τη λειτουργία ανύψωσης, επιλέξαμε να κατασκευάσουμε έναν μηχανισμό ο οποίος θα βασίζεται στην εμπλοκή δύο γριναζιών προκειμένου να πετύχουμε αύξηση της ισχύς και ελαχιστοποίηση της ταχύτητας άρα και ομαλότερη κίνηση κατά την ανύψωση. Συγκεκριμένα δημιουργήσαμε μια εμπλοκή ενός γριναζιού 12 δοντιών(γριναζί οδηγός) με ένα γριναζί 36 δοντιών(οδηγούμενο γριναζί). Η σχέση μετάδοσης μεταξύ των γριναζιών είναι 12:36 άρα 1:3 με αποτέλεσμα να έχουμε μικρότερη ταχύτητα καθώς για κάθε 3 περιστροφές του γριναζιού οδηγού έχουμε 1 περιστροφή στο οδηγούμενο γριναζί. Τέλος, για την μεταφορά των φορτίων επιλέξαμε να χρησιμοποιήσουμε τον μεγάλο κινητήρα πάνω στον οποίο θα βρίσκεται ο ρομποτικός μας βραχίονας και το οποίο θα περιστρέφει δεξιά και αριστερά.
6. Δημιουργία προσχεδίου: Για τη δημιουργία του προσχεδίου όπως προαναφέραμε, χρησιμοποιήσαμε την εφαρμογή Studio 2.0 της BrickLink όπου δημιουργήσαμε ένα 3D προσχέδιο του ρομποτικού μας συστήματος. Το προσχέδιο που υλοποιήσαμε ήταν το εξής:



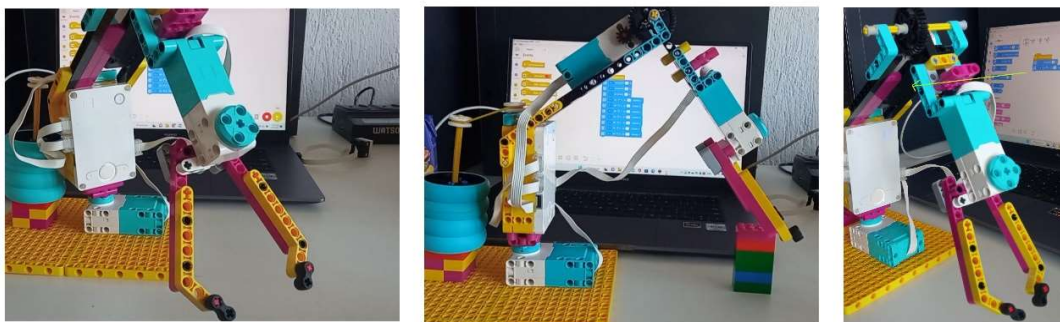
Εικόνα 8.3.2 Το προσχέδιο του Ρομποτικού Βραχίονα.

7. Δημιουργία Πρωτοτύπου: Στο σημείο αυτό παρουσιάζεται το πρωτότυπο που υλοποιήσαμε με βάση το προσχέδιο.



*Εικόνα 8.3.3 Το πρωτότυπο του Ρομποτικού Βραχίονα.*

8. Δοκιμή/Εξέταση: Κατά τη διάρκεια της δοκιμής στον ρομποτικό μας βραχίονα παρατηρήθηκαν τα εξής προβλήματα που έπρεπε να επανεξεταστούν: το βάρος του συνολικού ρομποτικού συστήματος ήταν αρκετά μεγάλο με αποτέλεσμα το σύστημα μας να παρουσιάζει μια ελαφρά κλίση προς τα μπροστά. Επιπρόσθετα, τα δομικά στοιχεία πάνω στα οποία στηριζόταν ο κινητήρας της ανύψωσης δεν ήταν κατάλληλα και δημιουργούσαν προβλήματα στη σταθερότητα του συστήματος.
9. Βελτίωση/Επανασχεδιασμός: Σε αυτό το στάδιο προχωρήσαμε στην επίλυση των προβλημάτων που προαναφέρθηκαν με επανασχεδιασμό του ρομποτικού μας συστήματος. Προκειμένου να λύσουμε το πρόβλημα του βάρους δημιουργήσαμε ένα αντίβαρο έτσι ώστε να ισορροπήσει το ρομποτικό μας σύστημα. Τέλος, προχωρήσαμε και στην αλλαγή των δομικών στοιχείων πάνω στα οποία στηρίζεται ο κινητήρας της ανύψωσης με άλλα πιο κατάλληλα, τα οποία προσέφεραν μια μεγαλύτερη επιφάνεια για μεγαλύτερη σταθερότητα του κινητήρα. Το τελικό αποτέλεσμα είναι το εξής:



Εικόνα 8.3.4 Το τελικό αποτέλεσμα του Ρομποτικού Βραχίονα.

### Ανάλυση Κώδικα:

Ολοκληρωμένο το πρόγραμμα έχει την εξής μορφή:

```

1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
4
5 motor_x = Motor ('A')
6 motor_y = Motor ('C')
7 grabber = Motor ('F')
8
9 motor_y.run_for_seconds(0.7,10)
10 grabber.run_for_seconds(1,-10)
11 motor_y.run_for_seconds(0.7,-10)
12 motor_x.run_for_degrees(45,10)
13 motor_y.run_for_seconds(0.7,10)
14 grabber.run_for_seconds(1,10)
15 motor_y.run_for_seconds(0.7,-10)
16

```

Εικόνα 8.3.5 Ολοκληρωμένος ο κώδικας του του Ρομποτικού Βραχίονα.

### Επιμερισμός του κώδικα και ανάλυση:

```

1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
4

```

Εικόνα 8.3.6 Οι βιβλιοθήκες και κλάσεις του Ρομποτικού Βραχίονα.

Όπως και στις υπόλοιπες εφαρμογές στις οποίες εργαστήκαμε με τη γλώσσα προγραμματισμού Python το πρώτο βήμα είναι η εισαγωγή των βιβλιοθηκών και των κλάσεων που θα χρειαστούμε στην κάθε εφαρμογή.

Στη συνέχεια ακολουθεί η δημιουργία των αντικειμένων που θα χρειαστούμε και η αρχικοποίηση τιμών.



```

4
5 motor_x = Motor ('A')
6 motor_y = Motor ('C')
7 grabber = Motor ('F')

```

*Εικόνα 8.3.7 Αρχικοποίηση των Αντικειμένων.*

Στη συγκεκριμένη εφαρμογή θα χρησιμοποιήσουμε τρία αντικείμενα: χρησιμοποιώντας την μεταβλητή “motor\_x” και την κλάση “Motor” δημιουργούμε ένα αντικείμενο για κινητήρα του συστήματος, ο οποίος βρίσκεται στην θύρα “A” και είναι υπεύθυνος για την κίνηση του βραχίονα στον x άξονα, δηλαδή για την περιστροφή του δεξιά και αριστερά. Το δεύτερο αντικείμενο που δημιουργούμε κάνοντας χρήση της μεταβλητής “motor\_y” και χρησιμοποιώντας την κλάση “Motor” είναι ο δεύτερος κινητήρας του βραχίονα που ελέγχει την ανύψωση και το κατέβασμα του αντικειμένου, δηλαδή την κίνηση στον y άξονα. Τέλος, το τρίτο αντικείμενο που θα χρειαστούμε είναι ένας τρίτος κινητήρας που θα ελέγχει το κλείσιμο και το άνοιγμα της αρπάγης, τον οποίο αρχικοποιούμε χρησιμοποιώντας την μεταβλητή “grabber” και την κλάση “Motor” και τον συνδέουμε στη θύρα “F”.

```

8
9 motor_y.run_for_seconds(0.7,10)
10 grabber.run_for_seconds(1,-10)
11 motor_y.run_for_seconds(0.7,-10)
12 motor_x.run_for_degrees(45,10)
13 motor_y.run_for_seconds(0.7,10)
14 grabber.run_for_seconds(1,10)
15 motor_y.run_for_seconds(0.7,-10)
16

```

*Εικόνα 8.3.8 Το βασικό κομμάτι του Προγράμματος.*

Στη συνέχεια προχωράμε στο κυρίως πρόγραμμα, όπου καλούμε την μέθοδο run\_for\_seconds στον κινητήρα motor\_y και ορίζουμε πως θα κινηθεί για 0.7 δευτερόλεπτα με ταχύτητα 10, δηλαδή θα εκτελέσει μία δεξιόστροφη κίνηση και θα κατεβάσει την αρπάγη ώστε να πιάσει το αντικείμενό μας. Έπειτα καλούμε ξανά την ίδια μέθοδο αλλά για την μεταβλητή grabber έτσι ώστε να κλείσει η αρπάγη και να

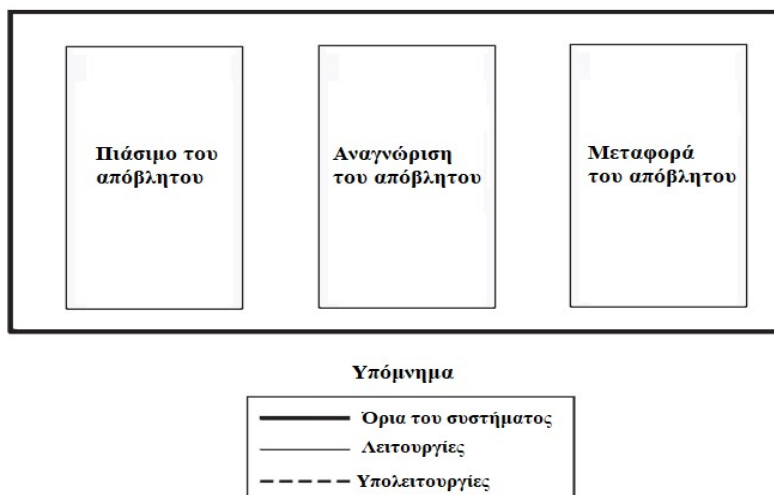
πιάσουμε το αντικείμενο, δίνοντας εντολή στον κινητήρα να περιστραφεί για 1 δευτερόλεπτο με ταχύτητα -10. Για να ανυψώσουμε το φορτίο καλούμε την ίδια συνάρτηση με την γραμμή 9 του κώδικα, με διαφορετικό όρισμα όμως όσον αφορά την ταχύτητα περιστροφής, βάζοντας ίση αλλά αντίθετη τιμή ώστε να κινηθεί προς την αντίθετη κατεύθυνση. Ακολούθως δίνουμε εντολή στον motor\_x να περιστραφεί δεξιόστροφα κατά 45 μοίρες και με ταχύτητα 10 καλώντας την μέθοδο run\_for\_degrees. Το φορτίο πλέον βρίσκεται στο σημείο το οποίο επιθυμούμε να τοποθετηθεί, επομένως καλούμε την ίδια εντολή της γραμμής 9 ώστε να κατέβει ξανά η αρπάγη και καλώντας την συνάρτηση της γραμμής 10 αλλά με αντίθετη τιμή στο όρισμα της ταχύτητας ώστε να ανοίξει η αρπάγη. Τέλος, καλούμε ξανά την εντολή της γραμμής 11 ώστε η αρπάγη να ανυψωθεί.



## 8.4 Μηχανή διαλογής.

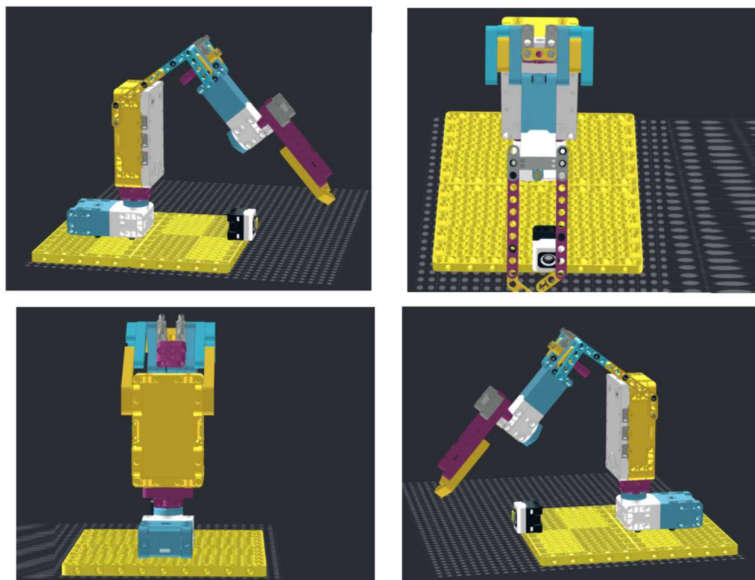
### Διαδικασία Σχεδιασμού:

1. Ορισμός της ανάγκης ή του προβλήματος : Ως πρόβλημα ορίζεται τα χαμηλά ποσοστά ανακύκλωσης των ανακυκλώσιμων υλικών.
2. Έρευνα της ανάγκης ή του προβλήματος: Οι πτυχές του προβλήματος που εντοπίσαμε και αποφασίσαμε πως χρήζουν αντιμετώπισης είναι οι εξής: Η διαλογή των υλικών είναι δυσκολότερη διαδικασία και απαιτεί περισσότερο χρόνο από την απόθεση όλων μαζί σε έναν κάδο αποβλήτων, η πιθανότητα να τοποθετηθούν μη ανακυκλώσιμα υλικά σε σακούλα που προορίζεται για ανακύκλωση, και εν τέλει να πεταχτεί σε κάδο απορριμμάτων, αποτρέπει αρκετούς από το να ασχοληθούν εξ αρχής με την ανακύκλωση.
3. Καταιγισμός Ιδεών και Σύλληψη Πιθανής ιδέας : Με βάση τα προηγούμενα δύο στάδια καταλήξαμε στην ιδέα υλοποίησης μιας ρομποτικής αρπάγης, η οποία θα λειτουργεί ως μηχανής διαλογής και θα ξεχωρίζει αυτόματα τα σκουπίδια από τα ανακυκλώσιμα υλικά, τοποθετώντας τα στους αντίστοιχους κάδους. Η μόνη εργασία που θα εκτελείται πλέον από τον χρήστη είναι η τοποθέτηση του απόβλητου μπροστά από τη μηχανή διαλογής. Η αυτοματοποίηση αυτή επιλύει ταυτόχρονα και τον χρόνο που απαιτεί η διαλογή αλλά και την πιθανότητα λανθασμένης τοποθέτησης.
4. Διαίρεση του συστήματος σε λειτουργίες: Οι λειτουργίες οι οποίες θεωρούμε ότι πραγματοποιεί το ρομποτικό μας σύστημα είναι η λειτουργία του πιασίματος του απόβλητου, η λειτουργία της Αναγνώρισης του και η λειτουργία Μεταφοράς στους αντίστοιχους κάδους.



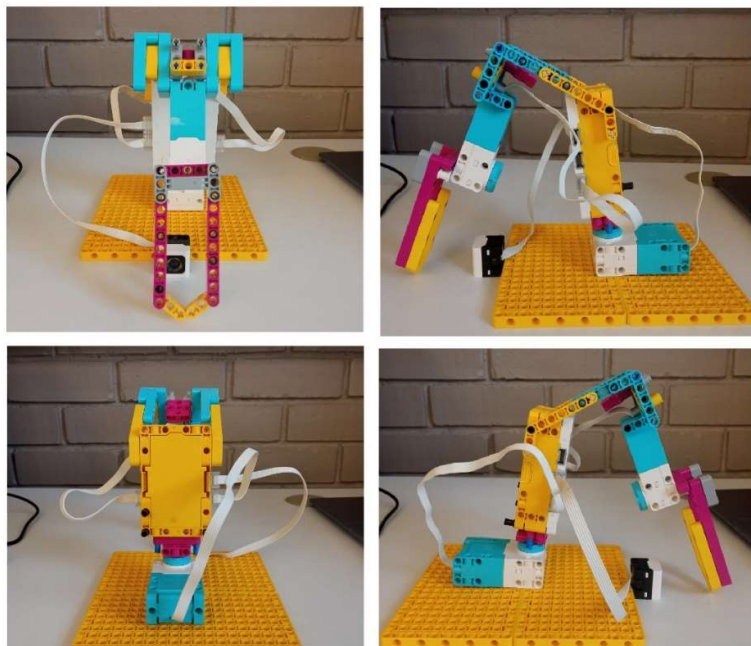
*Εικόνα 8.4.1 Οι λειτουργίες της Μηχανής Διαλογής.*

5. Έρευνα και Επιλογή Μηχανισμών: Στη συνέχεια πραγματοποιήσαμε έρευνα για πιθανούς μηχανισμούς που θα μπορούσαν να εξυπηρετήσουν την υλοποίηση της ιδέας μας. Για τη λειτουργία του πιασίματος του αποβλήτου επιλέξαμε να κατασκευάσουμε μια δαγκάνα η οποία με τη βοήθεια ενός κινητήρα θα ανοίγει και θα κλείνει προκειμένου να πιάσει το σκουπίδι. Σχετικά με τη λειτουργία της αναγνώρισης του αποβλήτου αποφασίσαμε να τοποθετήσουμε τον αισθητήρα απόστασης σε κεντρικό σημείο, ώστε να ξεχωρίζει ανάμεσα σε δύο χρώματα που θα συμβολίζουν τις δύο περιπτώσεις που ξεχωρίσουμε, τα ανακυκλώσιμα υλικά και τα υλικά που προορίζονται για τον κάδο απορριμμάτων. Τέλος, για την μεταφορά των αποβλήτων επιλέξαμε να χρησιμοποιήσουμε τον μεγάλο κινητήρα πάνω στον οποίο θα βρίσκεται το ρομποτικό μας σύστημα και το οποίο θα περιστρέφει δεξιά και αριστερά.
6. Δημιουργία προσχεδίου: Για τη δημιουργία του προσχεδίου όπως προαναφέραμε, χρησιμοποιήσαμε την εφαρμογή Studio 2.0 της BrickLink όπου δημιουργήσαμε ένα 3D προσχέδιο του ρομποτικού μας συστήματος. Το προσχέδιο που υλοποιήσαμε ήταν το εξής:



Εικόνα 8.4.2 Το προσχέδιο της Μηχανής Διαλογής.

#### 7. Δημιουργία Πρωτοτύπου:



Εικόνα 8.4.3 Το πρωτότυπο της Μηχανής Διαλογής.

8. Δοκιμή/Εξέταση: Κατά τη διάρκεια της δοκιμής δεν αντιμετωπίστηκαν προβλήματα στην λειτουργικότητα του ρομποτικού συστήματος.

9. Βελτίωση/Επανασχεδιασμός: Σε αυτό το στάδιο επανεξετάσαμε κάποια ζητήματα του συστήματος. Παρότι δεν υπήρχαν προβλήματα στην λειτουργικότητα για λόγους βελτιστοποίησης επιλέξαμε να προσθέσουμε τέσσερις ρόδες ως στηρίγματα στη βάση του μηχανισμού προκειμένου να δώσουμε ύψος στον μηχανισμό και να μην σέρνει τα τουβλάκια που μεταφέρει. Επιπρόσθετα, προσθέσαμε μια στηρικτική βάση πάνω στην οποία θα τοποθετείται το εκάστοτε τουβλάκι που θέλουμε να μετακινηθεί στη σωστή θέση ενώ προχωρήσαμε και στην δημιουργία δύο κουτιών προκειμένου να είναι πιο ξεκάθαρος ο διαχωρισμός τοποθετώντας το κάθε χρώμα στο δικό του κουτί. Το τελικό αποτέλεσμα είναι το εξής:



*Εικόνα 8.4.4 Το τελικό αποτέλεσμα της Μηχανής Διαλογής.*

## Ανάλυση Κώδικα:

Ολοκληρωμένο το πρόγραμμα έχει την εξής μορφή:

```
1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
4
5 hub = PrimeHub()
6 grabber = Motor ('F')
7 motor2 = Motor ('A')
8 colorSensor = ColorSensor ('B')
9
10 while True:
11     color = colorSensor.get_color()
12     if color == 'green':
13         hub.speaker.beep(60, 0.5)
14         grabber.run_for_seconds(1,-10)
15         motor2.run_for_degrees(90,10)
16         grabber.run_for_seconds(1,10)
17         motor2.run_for_degrees(90,-10)
18     elif color == 'red':
19         hub.speaker.beep(67, 0.5)
20         grabber.run_for_seconds(1,-10)
21         motor2.run_for_degrees(90,-10)
22         grabber.run_for_seconds(1,10)
23         motor2.run_for_degrees(90,10)
```

*Εικόνα 8.4.5 Ολοκληρωμένος ο Κώδικας της της Μηχανής Διαλογής.*

## Επιμερισμός του κώδικα και ανάλυση:

```
1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
4
```

*Εικόνα 8.4.6 Βιβλιοθήκες και κλάσεις της της Μηχανής Διαλογής.*

Πρωταρχική δουλειά στον προγραμματισμό με Python είναι η εισαγωγή των βιβλιοθηκών και των κλάσεων που θα χρειαστούμε στην κάθε εφαρμογή. Όπως προαναφέρθηκε το πρόγραμμα έχει ήδη εισάγει για εμάς τις πιθανές βιβλιοθήκες και κλάσεις που θα χρειαστούμε.

```
5 hub = PrimeHub()
6 grabber = Motor ('F')
7 motor2 = Motor ('A')
8 colorSensor = ColorSensor ('B')
```

*Εικόνα 8.4.7 Αρχικοποίηση Αντικειμένων της Μηχανής Διαλογής.*

Έπειτα προχωρήσαμε στη δημιουργία των αντικειμένων που θα χρειαστούμε για την υλοποίηση του προγράμματος μας. Για τη δημιουργία ενός αντικειμένου της κλάσης PrimeHub χρησιμοποιήσαμε την μεταβλητή “hub” . Για την δημιουργία της

δαγκάνας αναθέσαμε την κλάση “Motor” που βρίσκεται στη θύρα “F” σε μια μεταβλητή που ονομάσαμε “grabber” προκειμένου να μπορούμε να ξεχωρίσουμε εύκολα ποιος κινητήρας χρησιμοποιείται για να ενεργοποιήσει την δαγκάνα μας και ποιος χρησιμοποιείται για να δώσει την κίνηση στο σύστημα μας. Έτσι, ονομάσαμε τον κινητήρα που δίνει κίνηση “motor2” και αναθέσαμε σε αυτή τη μεταβλητή την κλάση του κινητήρα που βρίσκεται στη θύρα “A”. Τέλος, δημιουργήσαμε και ένα ψηφιακό αντικείμενο για τον αισθητήρα χρώματος χρησιμοποιώντας την μεταβλητή “colorSensor”.

```
9
10 while True:
11     color = colorSensor.get_color()
12     if color == 'green':
13         hub.speaker.beep(60, 0.5)
14         grabber.run_for_seconds(1, -10)
15         motor2.run_for_degrees(90, 10)
16         grabber.run_for_seconds(1, 10)
17         motor2.run_for_degrees(90, -10)
18     elif color == 'red':
19         hub.speaker.beep(67, 0.5)
20         grabber.run_for_seconds(1, -10)
21         motor2.run_for_degrees(90, -10)
22         grabber.run_for_seconds(1, 10)
23         motor2.run_for_degrees(90, 10)
```

Εικόνα 8.4.8 Το βασικό κομμάτι του κώδικα της Μηχανής Διαλογής.

Προχωρώντας στο βασικό κομμάτι του προγράμματος αρχικά δημιουργήσαμε μια επανάληψη της μορφής “while True” προκειμένου το πρόγραμμα μας να πραγματοποιεί συνεχώς τον έλεγχο του χρώματος. Όλες οι εντολές που βρίσκονται μέσα σε αυτή την επανάληψη ξεκινούν με τον ίδιο αριθμό εσοχών. Έπειτα δημιουργήσαμε μια μεταβλητή την οποία ονομάσαμε “color” η τιμή της οποία θα ορίζεται κάθε φορά από το χρώμα που ανιχνεύει ο αισθητήρας χρώματος καλώντας τη μέθοδο “get\_color”.

Στη συνέχεια δημιουργήσαμε τον πρώτο έλεγχο του προγράμματος που υλοποιείται όταν ο αισθητήρας μας ανιχνεύσει το χρώμα πράσινο. Δημιουργώντας έναν έλεγχο της μορφής “if” δίνουμε μια συνθήκη ελέγχου στην οποία ορίζουμε ότι αν η

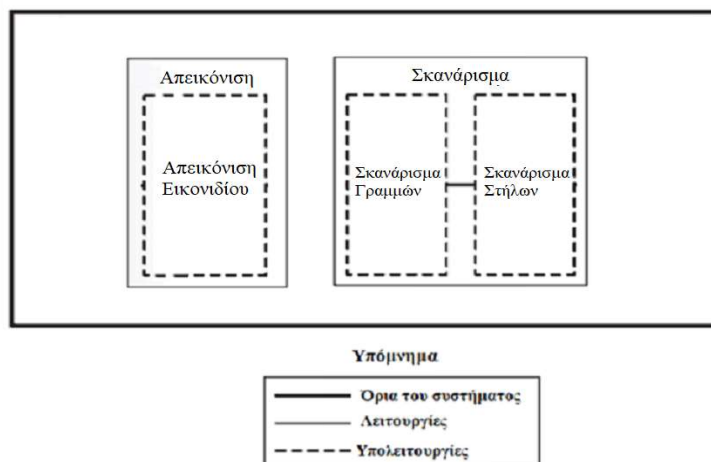
τιμή της μεταβλητής “color” είναι “green”, δηλαδή πράσινο τότε το πρόγραμμα μας θα εκτελέσει τις παρακάτω εντολές: Αρχικά χρησιμοποιώντας το ηχείο που βρίσκεται ενσωματωμένο στο hub θα παίξει έναν ήχο “beep” με τόνο 60 για 0.5 δευτερόλεπτα. Καλώντας τη μέθοδο “run\_for\_seconds” στη δαγκάνα μας, η οποία δέχεται δύο ορίσματα, το χρόνο σε δευτερόλεπτα και την ταχύτητα επιλέγουμε να κινησουμε τον κινητήρα μας για ένα δευτερόλεπτο με ταχύτητα -10 για να κλείσει η δαγκάνα μας. Το αρνητικό πρόσημο στην τιμή της ταχύτητας δείχνει πως ο κινητήρας μας θα κινηθεί αριστερόστροφα. Έπειτα, χρησιμοποιούμε τον δεύτερο κινητήρα που κινεί το ρομποτικό μας σύστημα και χρησιμοποιούμε τη μέθοδο “run\_for\_degrees” που μας επιτρέπει να ορίσουμε τις μοίρες που θέλουμε να κινηθεί ο κινητήρας και την ταχύτητα με την οποία θέλουμε να κινηθεί. Επιλέγουμε να κάνει μια δεξιόστροφη κίνηση για 90 μοίρες με ταχύτητα 10. Τέλος, χρησιμοποιώντας τις δύο τελευταίες εντολές πραγματοποιούμε δύο αντίθετες με τις προηγούμενες κινήσεις, δηλαδή ενεργοποιούμε τον κινητήρα της δαγκάνας για ένα δευτερόλεπτο με ταχύτητα 10 προς τα αριστερά προκειμένου να ανοίξει τη δαγκάνα και με την μέθοδο “run\_for\_degrees” κινούμε το ρομποτικό μας σύστημα 90 μοίρες προς τα αριστερά, με ταχύτητα -10, προκειμένου να έρθει στην αρχική του θέση και να είναι έτοιμο να εξετάσει το επόμενο χρώμα.

Με την χρήση της δεσμευμένης λέξης “elif” δημιουργούμε μια νέα συνθήκη ελέγχου, την οποία το πρόγραμμά μας θα ελέγξει σε περίπτωση που δεν ισχύει η πρώτη συνθήκη που ορίστηκε με την λέξη “if”. Έτσι, αν το χρώμα της μεταβλητής είναι κόκκινο το πρώτο πράγμα που θα κάνει το ρομποτικό μας σύστημα είναι να εκτελέσει έναν ήχο “beep” με τόνο 67 για 0.5 δευτερόλεπτα. Σε αυτή την περίπτωση, χρησιμοποιήσαμε διαφορετικό τόνο στον ήχο μας, ώστε να διακρίνεται από τον προηγούμενο ήχο, και έτσι το σύστημα μας να εκτελεί έναν διαφορετικό ήχο κάθε φορά που βλέπει ένα χρώμα. Στη συνέχεια, χρησιμοποιούμε την ίδια εντολή στην δαγκάνα μας προκειμένου να κλείσει, και πραγματοποιούμε μια αντίθετη κίνηση, αριστερόστροφη αυτή τη φορά, ώστε να γυρίσει 90 μοίρες προς την αντίθετη πλευρά και να τοποθετήσει το κόκκινο τουβλάκι στο κουτί των κόκκινων τούβλων. Τέλος καλούμε ξανά τη μέθοδο “run\_for\_seconds” με τιμές 1 δευτερόλεπτο και ταχύτητα 10 προκειμένου να ανοίξουμε την δαγκάνα μας, που εκφράζεται με τη μεταβλητή “grabber”, και επιστρέφουμε το ρομποτικό σύστημα στην αρχική του θέση χρησιμοποιώντας τη μέθοδο “run\_for\_degrees” ώστε να περιστραφεί για 90 μοίρες προς τα δεξιά με ταχύτητα 10, αντίθετη δηλαδή με αυτή της αρχικής κίνησης.

## 8.5 Μηχανή Αντιγραφής Εικονιδίων.

### Διαδικασία Σχεδιασμού:

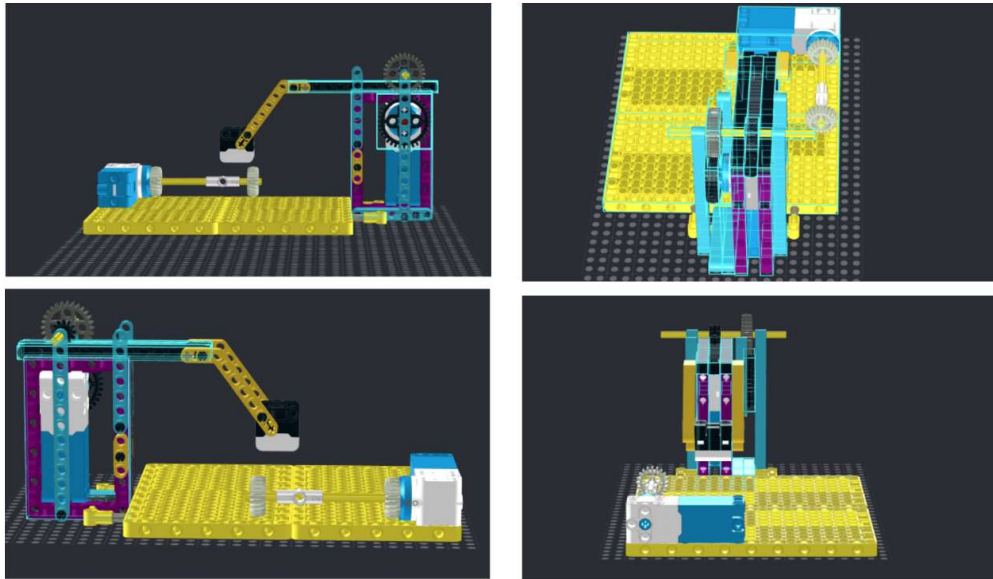
1. Ορισμός Προβλήματος: Ως πρόβλημα της συγκεκριμένης ενότητας ορίζεται το πρόβλημα αποθήκευσης έντυπων φωτογραφιών, καθώς απαιτείται συχνά αρκετός χώρος, όπως και η πιθανότητα είτε να χαθούν είτε να καταστραφούν.
2. Εξέταση και Ανάλυση του Προβλήματος: Οι πτυχές του προβλήματος που εντοπίσαμε και αποφασίσαμε πως χρήζουν αντιμετώπισης είναι οι εξής: οι φωτογραφίες που έχουν έντυπη μορφή είναι πιο επιρρεπείς στη φθορά, σε αντίθεση με τις ψηφιακές φωτογραφίες, οι οποίες όχι μόνο είναι πιο δύσκολο να χαθούν και να καταστραφούν, αλλά η μορφή τους επιτρέπει την αποθήκευση μεγάλου όγκου φωτογραφιών χωρίς να καταναλώνουν φυσικό χώρο.
3. Καταιγισμός Ιδεών και Σύλληψη Πιθανής ιδέας: Με βάση τα προηγούμενα στάδια καταλήξαμε στην ιδέα υλοποίησης ενός ρομποτικού μηχανισμού, ο οποίος θα «σκανάρει» εικονίδια και θα δημιουργεί ένα ψηφιακό αντίγραφό τους στην οθόνη του Hub.
4. Διαίρεση σε Λειτουργίες του Συστήματος: Οι λειτουργίες οι οποίες θεωρούμε ότι πραγματοποιεί ο ρομποτικός μηχανισμός είναι η λειτουργία του σκαναρίσματος των εικονιδίων και η λειτουργία της ψηφιακής απεικόνισης. Δεδομένου ότι η οθόνη του Hub αποτελείται από γραμμές και στήλες η λειτουργία του σκαναρίσματος διαιρείται σε δύο υπολειτουργίες, τη λειτουργία του σκαναρίσματος των γραμμών και τη λειτουργία του σκαναρίσματος των στηλών.



Εικόνα 8.5.1 Οι λειτουργίες της Μηχανής Αντιγραφής Εικονιδίων.

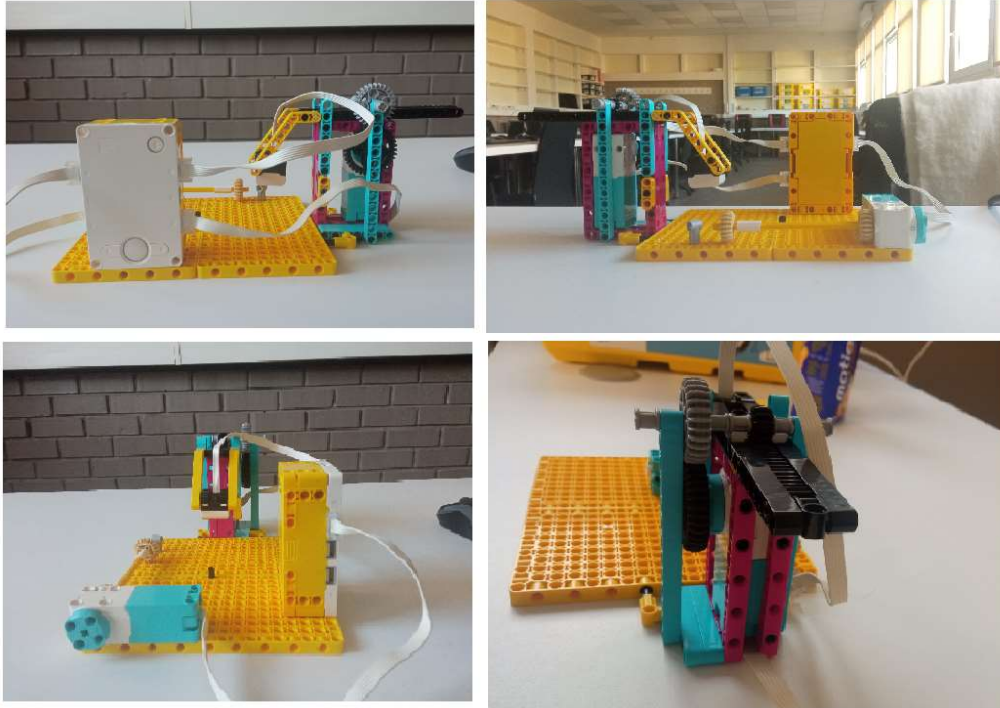


5. Έρευνα και Επιλογή Μηχανισμών: Στη συνέχεια πραγματοποιήσαμε έρευνα για πιθανούς μηχανισμούς που θα μπορούσαν να εξυπηρετήσουν την υλοποίηση της ιδέας μας. Για την υπολειτουργία του σκαναρίσματος των γραμμών, επιλέξαμε να χρησιμοποιήσουμε έναν οδοντωτό κανόνα και έναν οδοντωτό τροχό για την μεταφορά του αισθητήρα χρώματος στον οριζόντιο άξονα, καθώς η εμπλοκή ενός κυκλικού γραναζιού με ένα γραμμικό γρανάζι μετατρέπει την περιστροφική κίνηση σε γραμμική κίνηση. Για την υπολειτουργία του σκαναρίσματος των στηλών επιλέξαμε να υλοποιήσουμε έναν μηχανισμό ο οποίος θα μετακινεί το χαρτί μας προς τον κάθετο άξονα, προκειμένου να έρθει στη συνέχεια ο αισθητήρας χρώματος και να «σκανάρει» την γραμμή της συγκεκριμένης στήλης.
6. Δημιουργία προσχεδίου: Για τη δημιουργία του προσχεδίου όπως προαναφέραμε, χρησιμοποιήσαμε την εφαρμογή Studio 2.0 της BrickLink όπου δημιουργήσαμε ένα 3D προσχέδιο του ρομποτικού μας συστήματος. Το προσχέδιο που υλοποιήσαμε ήταν το εξής:



Εικόνα 8.5.2 Το προσχέδιο της Μηχανής Αντιγραφής Εικονιδίων.

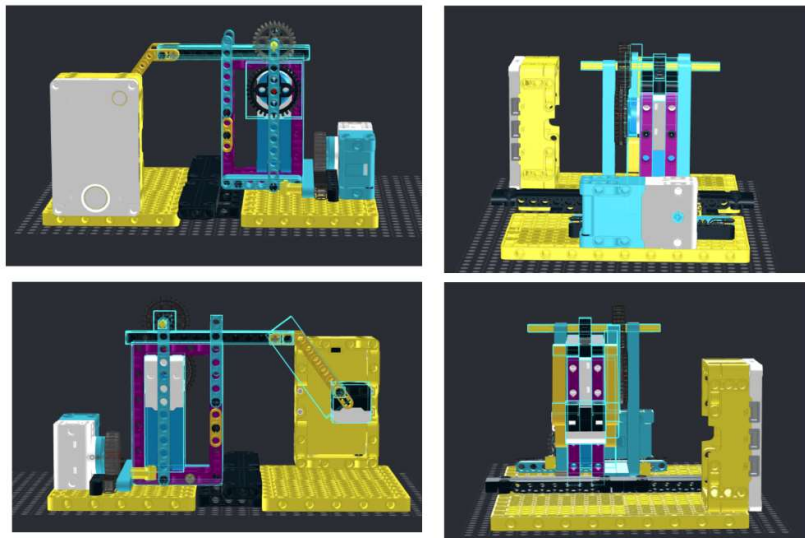
7. Δημιουργία Πρωτοτύπου: Στο σημείο αυτό παρουσιάζεται το πρωτότυπο που υλοποιήσαμε με βάση το προσχέδιο.



*Εικόνα 8.5.3 Το πρωτότυπο της Μηχανής Αντιγραφής Εικονιδίων.*

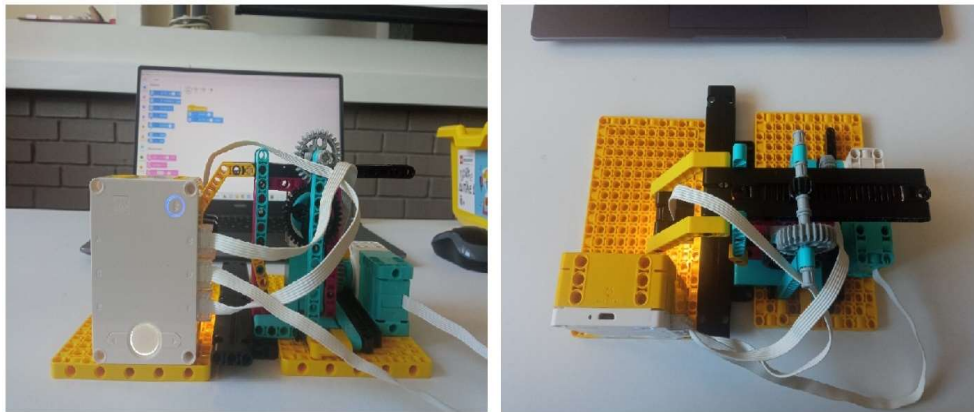
8. Δοκιμή/Εξέταση (Testing): Κατά τη διάρκεια της δοκιμής του ρομποτικού συστήματος μας αντιμετωπίσαμε τα εξής προβλήματα: Ο μηχανισμός ο οποίος θα μετέφερε το χαρτί στον κάθετο άξονα, δεν μπορούσε να μεταφέρει το χαρτί, καθώς το πάχος του ήταν ιδιαίτερα λεπτό για να μπορέσουν τα γρανάζια να φτάσουν σε αυτό. Επιπρόσθετα αποφασίσαμε πως ακόμα και αν αλλάζουμε το πάχος του χαρτιού ή τα γρανάζια ένας τέτοιος μηχανισμός δεν θα είναι ιδιαίτερα αποτελεσματικός και χρηστικός σε ένα σύστημα που απαιτεί μεγάλη ακρίβεια. Τέλος, πραγματοποιήσαμε και δοκιμή του συστήματος ανίχνευσης αναφορικά με το μέγεθος της εικόνας που μπορεί να σκανάρει. Ενώ στην αρχή προοριζόταν για ένα σύστημα που θα σκανάρει ένα μοτίβο 5x5 το σύστημα δεν κατάφερε να περάσει αυτή τη δοκιμή. Έχοντας βρει την απόσταση που διανύει το γρανάζι και διαιρώντας την δια 5 δημιουργήσαμε ένα δοκιμαστικό εικονίδιο το οποίο δεν κατάφερε να το αντιγράψει πιστά το ρομποτικό μας σύστημα. Μετά από αλληπάλληλες δοκιμές καταλήξαμε πως το πιο αποδοτικό σενάριο είναι το «σκανάρισμα» μιας εικόνας με διαστάσεις 3x3.

9. Βελτίωση/Επανασχεδιασμός: Σε αυτό το στάδιο προχωρήσαμε στην επίλυση των προβλημάτων που προαναφέρθηκαν με επανασχεδιασμό του ρομποτικού μας συστήματος. Στο συγκεκριμένο ρομποτικό σύστημα προχωρήσαμε σε αναδιαμόρφωση ολόκληρου του συστήματος και αλλαγή του μηχανισμού που θα εκτελεί μια από τις λειτουργίες. Έτσι, αντί να μετακινούμε το χαρτί προς το κάθετο άξονα, επιλέξαμε να υλοποιήσουμε έναν μηχανισμό όπου θα μετακινεί ολόκληρο τον μηχανισμό που χρησιμοποιεί τον αισθητήρα χρώματος προς τον κάθετο άξονα, χρησιμοποιώντας ξανά ένα κυκλικό και ένα γραμμικό γρανάτζι. Δημιουργήσαμε λοιπόν ένα προσχέδιο του νέου αυτού μηχανισμού.



*Εικόνα 8.5.4 Το νέο προσχέδιο της Μηχανής Αντιγραφής Εικονιδίων.*

Με βάση το καινούργιο προσχέδιο υλοποιήσαμε το εξής πρωτότυπο:

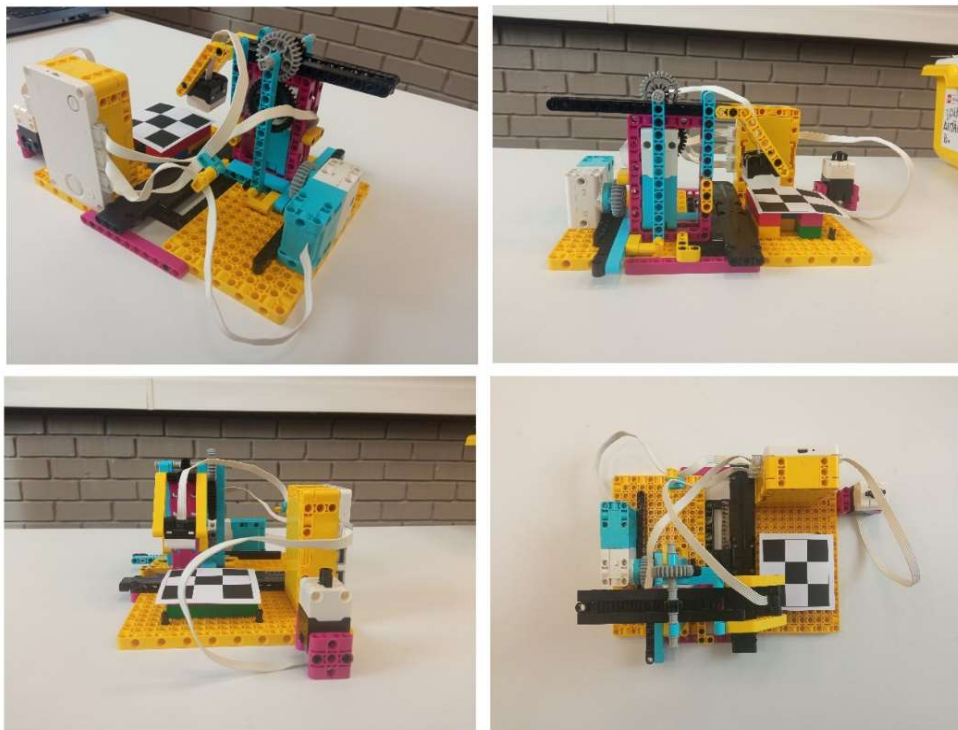


*Εικόνα 8.5.5 Το νέο πρωτότυπο της Μηχανής Αντιγραφής Εικονιδίων.*

Δοκιμάζοντας τον νέο μηχανισμό μας, σαν πρόβλημα παρατηρήθηκε η δυσκολία του ρομποτικού συστήματος να επιστρέψει στην αρχική του θέση, καθώς το ροζ πλαίσια που διαθέτει ο μηχανισμός «σκαναρίσματος» έμπαιναν μέσα σε μια από τις εσοχές που έχουν τα μαύρα πλαίσια/ panels που χρησιμοποιούνται για να περιορίσουν την κίνηση του. Για να λύσουμε αυτό το πρόβλημα προσθέσαμε μια μαύρη δοκό 9 οπών και μετατοπίσαμε τα πλαίσια προς τα αριστερά.

Για λόγους βελτίωσης, αποφασίσαμε τη δημιουργία ενός «κρεβατιού» για την τοποθέτηση της εικόνας την οποία θέλουμε να αντιγράψει το σύστημα, ώστε να δώσουμε στην εικόνα μας μια καλύτερη απόσταση από τον αισθητήρα χρώματος, ενώ προχωρήσαμε και στην προσθήκη του αισθητήρα αφής με το πάτημα του οποίου θα ξεκινάει η διαδικασία του σκαναρίσματος. Με αυτόν τον τρόπο το ρομποτικό μας σύστημα θα μπορεί να εκτελεί διαδοχικά «σκαναρίσματα» εικόνων, επειδή θα περιμένει την εντολή του χρήστη για να ξεκινήσει και έτσι ο χρήστης θα έχει τον χρόνο να αλλάξει τα εικονίδια χωρίς να χρειάζεται κάθε φορά να εκτελεί ξανά το πρόγραμμα.

Το τελικό αποτέλεσμα ήταν το εξής:



Εικόνα 8.5.6 Το τελικό αποτέλεσμα της Μηχανής Αντιγραφής Εικονιδίων.

## Ανάλυση Κώδικα:

Ολοκληρωμένο το πρόγραμμα έχει την εξής μορφή:

```
1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
4
5 hub = PrimeHub()
6 forceSensor = ForceSensor ('A')
7 colorSensor = ColorSensor ('F')
8 motorx = Motor ('B')
9 motory = Motor ('D')
10
11 while True:
12     hub.light_matrix.off ()
13     pixel = 1
14     pixel_y = 1
15     forceSensor.wait_until_pressed()
16     for i in range (3):
17         for j in range (3):
18             if colorSensor.get_reflected_light () < 60:
19                 hub.light_matrix.set_pixel (pixel, pixel_y,100)
20             else:
21                 hub.light_matrix.set_pixel (pixel,pixel_y,0)
22             wait_for_seconds(1)
23             if pixel < 3:
24                 motorx.run_for_seconds(2.5,-8)
25             pixel = pixel + 1
26             motorx.run_for_seconds (5,8)
27             pixel = 1
28             if pixel_y < 3:
29                 motory.run_for_seconds (1.4,-8)
30             pixel_y = pixel_y + 1
31             motory.run_for_seconds (2.8,8)
32
```

*Εικόνα 8.5.7 Ολοκληρωμένο το πρόγραμμα της Μηχανής Αντιγραφής Εικονιδίων.*

## Επιμερισμός του κώδικα και ανάλυση:

```
1 from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
2 from spike.control import wait_for_seconds, wait_until, Timer
3 from math import *
```

*Εικόνα 8.5.8 Βιβλιοθήκες και κλάσεις της Μηχανής Αντιγραφής Εικονιδίων.*

Ο κώδικας μας ξεκινάει όπως και κάθε φορά με την πολύ βασική λειτουργία της εισαγωγής των βιβλιοθηκών και των κλάσεων που θα χρειαστούμε στην εφαρμογή.

```
4
5 hub = PrimeHub()
6 forceSensor = ForceSensor ('A')
7 colorSensor = ColorSensor ('F')
8 motorx = Motor ('B')
9 motory = Motor ('D')
```

*Εικόνα 8.5.9 Τα αντικείμενα της Μηχανής Αντιγραφής Εικονιδίων.*



Στη συνέχεια ακολουθεί η δημιουργία των αντικειμένων που θα χρειαστούμε και η αρχικοποίηση τιμών. Στη συγκεκριμένη εφαρμογή δημιουργούμε πέντε αντικείμενα:

Το πρώτο αντικείμενο είναι το Hub του Spike Prime το οποίο το αρχικοποιούμε με τη μεταβλητή hub. Στη συνέχεια αρχικοποιούμε τον αισθητήρα αφής με την μεταβλητή forceSensor ο οποίος βρίσκεται στην θύρα A και με παρόμοιο τρόπο, αρχικοποιούμε και τον αισθητήρα χρώματος ο οποίος βρίσκεται στη θύρα F, κάνοντας χρήση της μεταβλητής colorSensor. Τέλος, δημιουργούμε και δύο διαφορετικά αντικείμενα για τους δύο μας κινητήρες: για τον κινητήρα που είναι υπεύθυνος για την κάθετη κίνηση και βρίσκεται στη θύρα B χρησιμοποιήσαμε την μεταβλητή motorX, ενώ για τον κινητήρα που είναι υπεύθυνος για την οριζόντια κίνηση και βρίσκεται στην θύρα D χρησιμοποιήσαμε την μεταβλητή motorY.

```
10
11 while True:
12     hub.light_matrix.off ()
13     pixel = 1
14     pixel_y = 1
15     forceSensor.wait_until_pressed()
```

*Εικόνα 8.5.10 Η αρχή του προγράμματος.*

Στο βασικό κομμάτι του κώδικα, το πρώτο πράγμα που κάνουμε είναι να δημιουργήσουμε μια μορφή επανάληψης της μορφής while True ώστε το πρόγραμμα μας να τρέχει συνεχώς και να μπορεί έτσι να εκτελεί διαδοχικά σκαναρίσματα χωρίς να χρειαστεί να το τρέχουμε ξανά και ξανά. Έπειτα, απενεργοποιούμε όλα τα pixels του Light Matrix ώστε κάθε φορά που θα τρέχει το πρόγραμμα μας το Light Matrix να είναι «καθαρό» και έτοιμο να απεικονίσει μια νέα εικόνα. Έπειτα, προχωρούμε στην δημιουργία και στην αρχικοποίηση δύο μεταβλητών, της μεταβλητής “pixel” που θα έχει την τιμή 1 και της μεταβλητής “pixel\_y” που θα έχει επίσης την τιμή 1. Ο λόγος για τον οποίο οι συγκεκριμένες μεταβλητές ξεκινούν με την τιμή 1 είναι καθώς θέλουμε η απεικόνιση του εικονιδίου να ξεκινήσει από το pixel (1,1) ώστε το αποτέλεσμα να βρίσκεται στον κέντρο του Light Matrix. Στην επόμενη σειρά ουσιαστικά δίνουμε εντολή στο πρόγραμμα να περιμένει, να σταματήσει δηλαδή η ροή του προγράμματος εκεί, μέχρι να πατηθεί ο αισθητήρας αφής.

```

16 for i in range (3):
17     for j in range (3):
18         if colorSensor.get_reflected_light () < 60:
19             hub.light_matrix.set_pixel (pixel, pixel_y,100)
20         else:
21             hub.light_matrix.set_pixel (pixel,pixel_y,0)
22             wait_for_seconds(1)
23             if pixel < 3:
24                 motorx.run_for_seconds(2.5,-8)
25             pixel = pixel + 1
26         motorx.run_for_seconds (5,8)
27         pixel = 1
28         if pixel_y < 3:
29             motory.run_for_seconds (1.4,-8)
30         pixel_y = pixel_y + 1
31     motory.run_for_seconds (2.8,8)
32

```

Εικόνα 8.5.11 Χρωματική αντιστοίχιση των δύο επαναλήψεων.

Στη συνέχεια το υπόλοιπο κομμάτι του κώδικα βασίζεται στην επανάληψη τριών κινήσεων: στο «σκανάρισμα» της γραμμής, στην επιστροφή του αισθητήρα στην αρχική θέση μετά το «σκανάρισμα», και στη μετακίνηση του αισθητήρα προς τα κάτω, προς τον κάθετο άξονα προκειμένου να προχωρήσει στο «σκανάρισμα» της επόμενης γραμμής. Δεδομένου ότι πρόκειται για μια φωτογραφία διαστάσεων 3x3 θέλουμε κάθε «σκανάρισμα» και κάθε κίνηση να εκτελεστεί 3 φορές. Για να το πετύχουμε αυτό δημιουργήσαμε μια δύο επαναλήψεις όπου η μια είναι κομμάτι της άλλης, έχουμε δηλαδή μια εμφωλευμένη επανάληψη. Στην φωτογραφία που απεικονίζεται ο κώδικας, παρότι το Spike Prime μας βοηθά με την προσθήκη γκρι οριζόντιων γραμμών οι οποίες περιλαμβάνουν τις ομαδοποιημένες εντολές που απεικονίζονται στο ίδιο επίπεδο, προσθέσαμε δύο διαφορετικά χρώματα για να γίνουν ακόμα πιο κατανοητές οι εκτάσεις των δύο επαναλήψεων.

Παρότι η εντολή της πρώτης επανάληψης προηγείται, θα προχωρήσουμε πρώτα στην ανάλυση της δεύτερης επανάληψης, που αποτελεί κομμάτι της πρώτης προκειμένου να γίνει όσο το δυνατό πιο κατανοητή η εξήγηση του κώδικα.

```

for j in range (3):
    if colorSensor.get_reflected_light () < 60:
        hub.light_matrix.set_pixel (pixel, pixel_y,100)
    else:
        hub.light_matrix.set_pixel (pixel,pixel_y,0)
    wait_for_seconds(1)
    if pixel < 3:
        motorx.run_for_seconds(2.5,-8)
    pixel = pixel + 1

```

Εικόνα 8.5.12 Η επανάληψη "σκαναρίσματος" μιας σειράς.

Έτσι, δημιουργούμε μια επανάληψη που για τον έλεγχο του βρόγχου χρησιμοποιεί την μεταβλητή *j* και η οποία θα εκτελεστεί τρεις φορές. Η επανάληψη ξεκινάει ελέγχοντας την ανάκλαση του φωτός. Αν το ποσοστό του ανακλώμενου φωτός που φτάνει στον αισθητήρα χρώματος είναι μικρότερο από 60% (άρα το κουτάκι είναι μαύρο) τότε το *pixel* που βρίσκεται στη θέση *pixel*, *pixel\_y* παίρνει σαν τιμή έντασης το 100 και ανάβει. Στη συγκεκριμένη περίπτωση, οι μεταβλητές αυτές έχουν και οι δύο την τιμή 1 όπως ορίσαμε στην αρχή του προγράμματος. Έτσι, το *pixel* που θα ενεργοποιηθεί βρίσκεται στη θέση 1,1. Εάν η τιμή είναι μεγαλύτερη του 60% (άρα το κουτάκι είναι άσπρο) τότε το *pixel* που βρίσκεται πάλι στη θέση *pixel*, *pixel\_y* (άρα στη θέση 1,1) παίρνει σαν τιμή φωτεινότητας το 0, και παραμένει σβηστό. Αφού εκτελεστεί ο έλεγχος, εισάγουμε μια παύση στο πρόγραμμα με τη χρήση της "wait\_for\_seconds()" όπου παίρνει σαν όρισμα τον αριθμό 1, προκειμένου το πρόγραμμα μας να περιμένει για 1 δευτερόλεπτο. Αφού έχουμε ολοκληρώσει τον έλεγχο του πρώτου τετραγώνου του εικονιδίου, πρέπει να μετακινήσουμε τον αισθητήρα μας προς τα δεξιά για να ελέγξει το επόμενο τετράγωνο της σειράς. Για να πετύχουμε αυτό δημιουργήσαμε έναν έλεγχο όπου, αν η τιμή της μεταβλητής *pixel* είναι μικρότερη του 3, τότε ο κινητήρας μας κινείται αριστερόστροφα για 2.5 δευτερόλεπτα προκειμένου να πάει στο επόμενο τετράγωνο. Μετά τον έλεγχο αυτό, αλλάζουμε την τιμή της μεταβλητής *pixel* κατά μία μονάδα χρησιμοποιώντας την δομή "pixel = pixel + 1". Η συγκεκριμένη δομή μπορεί να φαίνεται λίγο ανορθόδοξη καθώς μαθηματικά δεν θα μπορούσε να σταθεί, παρόλα αυτά είναι σημαντικό να θυμόμαστε πως στον προγραμματισμό το σύμβολο "=" δηλώνει εκχώρηση και όχι ισοδυναμία. Έτσι, η εντολή αυτή μεταφράζεται σαν «Πρόσθεσε στην τιμή της μεταβλητής *pixel* μία μονάδα και στη συνέχεια καταχώρησε αυτή την τιμή μέσα στην μεταβλητή». Σε αυτή



τη περίπτωση η τιμή της μεταβλητής pixel είναι 1, άρα το αποτέλεσμα θα είναι  $(\text{pixel}=1) + 1 = 2$ , και μετά αυτό το 2 θα καταχωρηθεί μέσα στη μεταβλητή, η οποία θα αποκτήσει νέα τιμή. Όταν η τιμή της μεταβλητής φτάσει στο 3, τότε ο κινητήρας μας δεν θα δώσει κίνηση στον αισθητήρα γιατί θέλουμε το σύστημα μας να πραγματοποιήσει 3 ελέγχους φωτεινότητας, αλλά 2 κινήσεις σε κάθε γραμμή του εικονιδίου.

```
16 for i in range (3):
17     for j in range (3):
18         if colorSensor.get_reflected_light () < 60:
19             hub.light_matrix.set_pixel (pixel, pixel_y,100)
20         else:
21             hub.light_matrix.set_pixel (pixel,pixel_y,0)
22         wait_for_seconds(1)
23         if pixel < 3:
24             motorx.run_for_seconds(2.5,-8)
25             pixel = pixel + 1
26         motorx.run_for_seconds (5,8)
27         pixel = 1
28         if pixel_y < 3:
29             motory.run_for_seconds (1.4,-8)
30         pixel_y = pixel_y + 1
31         motory.run_for_seconds (2.8,8)
```

Εικόνα 8.5.13 Η εξωτερική επανάληψη.

Ο κώδικας μας μέχρι αυτό το σημείο εκτελεί το σκανάρισμα της μιας γραμμής του εικονιδίου. Εμείς θέλουμε όμως να εκτελεί αυτό το κομμάτι για κάθε γραμμή του εικονιδίου, να εκτελεί δηλαδή αυτόν τον έλεγχο, να επιστρέφει στην αρχική θέση ο αισθητήρας, να αλλάζει στήλη, να ελέγχει την επόμενη γραμμή και ούτω καθεξής.

Για να το πετύχουμε αυτό, δημιουργούμε μια «εξωτερική» επανάληψη στην οποία η πρώτη επανάληψη που έχουμε δημιουργήσει αποτελεί το πρώτο της βήμα. Εκτελώντας λοιπόν τον έλεγχο της μια γραμμής, το επόμενο βήμα που θέλουμε να επαναλαμβάνει το ρομποτικό μας σύστημα μετά από κάθε έλεγχο, είναι η επιστροφή του αισθητήρα στην αρχική του θέση. Γι' αυτό το λόγο, καλούμε την μέθοδο `run_for_seconds` ξανά στον κινητήρα `motorx` η οποία στο όρισμα του χρόνου θα λάβει την τιμή 5, όσο δηλαδή και ο χρόνος που εκτέλεσε στον έλεγχο το σύστημα μας (έκανε 2 κινήσεις των 2.5 δευτερολέπτων) ενώ στο όρισμα της ταχύτητας θα λάβει την τιμή 8, για να εκτελέσει αυτή την φορά δεξιόστροφη κίνηση. Στην επόμενη γραμμή

επαναφέρουμε την τιμή της μεταβλητής `pixel` σε 1, καθώς ξεκινάει το «σκανάρισμα» μιας νέας γραμμής κι έτσι οι τιμές που θα που θα λάβει η μεταβλητή αυτή θέλουμε να είναι ξανά 1,2 και 3.

Έπειτα, δημιουργούμε ξανά μια συνθήκη ελέγχου: αν η τιμή της μεταβλητής `pixel_y` είναι μικρότερη του αριθμού 3, τότε καλούμε την μέθοδο `run_for_seconds` στον κινητήρα που ονομάσαμε με την μεταβλητή `motory` ώστε να εκτελέσει μια κίνηση στον κάθετο άξονα για 1.4 δευτερόλεπτα με ταχύτητα -8. Ο λόγος για τον οποίο δημιουργούμε αυτόν τον έλεγχο είναι γιατί, όπως και στον οριζόντιο άξονα, έτσι και στον κάθετο το σύστημα μας θέλουμε να εκτελέσει 3 ελέγχους αλλά δύο κινήσεις. Στην συνέχεια αυξάνουμε την τιμή της μεταβλητής `pixel_y` ξανά με τον ίδιο τρόπο, αυξάνοντας την τιμή της κατά ένα, και καταχωρώντας στη συνέχεια την νέα τιμή στην μεταβλητή. Έτσι, η τιμή των `pixels` θα αλλάζει αναλογικά με την θέση στηλών.

Τέλος, αφού ολοκληρώσει το σύστημα μας την αντιγραφή του εικονιδίου, του ζητάμε να επιστρέψει στην αρχική του θέση, ώστε να είναι έτοιμο να σκανάρει μια ενδεχόμενη δεύτερη εικόνα. Η λειτουργία αυτή επιτυγχάνεται με την χρήση της εντολής `motory.run_for_seconds (2.8,8)`, δηλαδή δίνοντας εντολή στον κινητήρα που ευθύνεται για την κάθετη κίνηση να περιστραφεί δεξιόστροφα για 2.8 δευτερόλεπτα, για όσα δηλαδή χρειάστηκε για να εκτελέσει τις απαιτούμενες κινήσεις στον κάθετο άξονα (πραγματοποιήσαμε 2 κινήσεις των 1.4 δευτερολέπτων).

## Κεφάλαιο 9 – Συμπεράσματα

Φτάνοντας στο τέλος της παρούσας διπλωματικής εργασίας, και με βάση τα αποτελέσματα που προέκυψαν από αυτήν, είμαστε πλέον σε θέση να εξάγουμε πολλά χρήσιμα συμπεράσματα. Για να φτάσουμε φυσικά εδώ, αναγκαία ήταν αρχικά η διεξοδική μελέτη της STEAM Εκπαίδευσης, ώστε να κατανοήσουμε πλήρως το τι προσβέυει, ποια είναι τα αντικείμενά της, ποια είναι τα προτερήματα και τα πλεονεκτήματα της εφαρμογής της, όπως και τους τρόπους με τους οποίους μπορεί να εφαρμοστεί στις απαιτήσεις της σύγχρονης εκπαίδευσης. Επιλέξαμε λοιπόν να ασχοληθούμε με το κομμάτι που αφορά την εκπαιδευτική ρομποτική, καθώς θεωρούμε πως είναι ένα από τα πιο αξιοσημείωτα και χρήσιμα εργαλεία για να προσεγγίσει κανείς την STEAM εκπαίδευση, υλοποιώντας τελικά 5 ολοκληρωμένες εφαρμογές ρομποτικών συστημάτων.

Τα συμπεράσματα που προκύπτουν είναι τα εξής:

- Η πρόταση δημιουργίας και σχεδιασμού των κατασκευών την οποία παρουσιάσαμε και εφαρμόσαμε για τις 5 κατασκευές μας ήταν άκρως λειτουργική και αποδοτική, οδηγώντας σε σωστές και αποτελεσματικές λύσεις στα προβλήματα που είχαμε θέσει.
- Σχετικά με την αναπηρική καρέκλα και τη ρομποτική γάτα θεωρούμε πως πετύχαμε επακριβώς, χωρίς ιδιαίτερες αλλαγές, τους στόχους τους οποίους είχαμε θέσει, καθώς καταλήξαμε σε ένα πολύ ικανοποιητικό αποτέλεσμα και στις δύο εφαρμογές, τόσο σε επίπεδο κατασκευής, όσο και σε επίπεδο προγραμματισμού.
- Αναφορικά με τον ρομποτικό βραχίονα και την μηχανή διαλογής, ξεπεράστηκαν σχετικά εύκολα τα προβλήματα που προέκυψαν στο κατασκευαστικό επίπεδο (στον ρομποτικό βραχίονα το βάρος εξισορροπήθηκε με τη δημιουργία ενός αντίβαρου, ενώ στη μηχανή διαλογής το σύρσιμο του φορτίου ξεπεράστηκε με την ανύψωση του συστήματος). Σαν αποτέλεσμα είχαμε δύο πολύ λειτουργικές κατασκευές, οι οποίες παισιώνονται από πολύ αποδοτικά προγράμματα που μας βοηθούν να επιτύχουμε τους συνολικούς στόχους των projects.

- Τέλος, στην μηχανή αντιγραφής εικονιδίων επανασχεδιάζοντας το σύστημα και δημιουργώντας το κατάλληλο για αυτό πρόγραμμα, φέραμε εις πέρας την πετυχημένη υλοποίηση ενός πιο απαιτητικού συστήματος.
- Συμπερασματικά, θεωρούμε ότι όλα τα projects τα οποία υλοποιήσαμε, τόσο σε επίπεδο κατασκευής, όσο και σε επίπεδο προγραμματισμού είναι αρκετά πετυχημένα ώστε να ενταχθούν σε STEAM εκπαιδευτικά περιβάλλοντα.

## Βιβλιογραφία

- Ackermann, E. (2001). *Piaget's Constructivism, Papert's Constructionism: What's the difference?*
- Adams, R., Evangelou, D., English, L., De Figueiredo, A. D., Mousoulides, N., Pawley, A. L., Schiefellite, C., Stevens, R., Svinicki, M., Trenor, J. M., & Wilson, D. M. (2011). Multiple Perspectives on Engaging Future Engineers. *Journal of Engineering Education*, 100(1), 48–88.  
<https://doi.org/10.1002/j.2168-9830.2011.tb00004.x>
- Adnan, M., Puteh, M., & Mistima, S. (2018). *INTEGRATING STEM EDUCATION THROUGH PROJECT-BASED INQUIRY LEARNING IN TOPIC SPACE AMONG YEAR ONE CHILDREN* *Information processing View project Learning Fractions Diagrammatically View project*.  
<https://doi.org/10.7456/1080SSE/185>
- Aguilera, D., & Ortiz-Revilla, J. (2021). STEM vs. STEAM Education and Student Creativity: A Systematic Literature Review. *Education Sciences*, 11(7), 331.  
<https://doi.org/10.3390/educsci11070331>
- Ah-Nam, L., Osman, K., & Bangi. (2017). *Developing 21 st Century Skills through a Constructivist-Constructionist Learning Environment*. 3(2), 205–216.
- Aithal, P., & Aithal, S. (2020). *Importance of Arts & Design in Liberal Education based STEAM model of Higher Education*.
- Alimisis, D. (2012). Robotics in Education & Education in Robotics: Shifting Focus from Technology to Pedagogy. In *Faculty of Mathematics and Physics* (pp. 7–14).

- Anwar, S., Bascou, N. A., Menekse, M., & Kardgar, A. (2019). A Systematic Review of Studies on Educational Robotics. *Journal of Pre-College Engineering Education Research (J-PEER)*, 9(2). <https://doi.org/10.7771/2157-9288.1223>
- Aran, W., & Moore, T. (2013). *Theoretical Foundations for Effective STEM Learning Environments*.
- Arslan, Ö. (2021). A Different Perspective on Design-Skill Workshops and STEM Education: Teachers' Opinions. *Journal of STEM Teacher Institutes*, 2021(1), 58–65.
- Athanasiou, L., Topali, P., & Mikropoulos, T. A. (2017). The Use of Robotics in Introductory Programming for Elementary Students. *Educational Robotics in the Makers Era*, 183–192. [https://doi.org/10.1007/978-3-319-55553-9\\_14](https://doi.org/10.1007/978-3-319-55553-9_14)
- Badmus, O. T., & Omosewo, E. O. (2020). Evolution of STEM, STEAM and STREAM Education in Africa: The Implication of the Knowledge Gap. *International Journal on Research in STEM Education*, 2(2), 99–106. <https://doi.org/10.31098/ijrse.v2i2.227>
- Bahrum, S. (2017). Integration of STEM Education in Malaysia and Why to STEAM  
Cite this paper A Measure of Student's Motivation, Attitude and Parental Influence towards Interest in STE... FAZILAH RAZALI STEM Integration Module In Teaching And Learning with Visual Arts Education: A Needs Of Analysis. *Integration of STEM Education in Malaysia and Why to STEAM*. <https://doi.org/10.6007/IJARBSS/v7-i6/3027>
- Barak, M., & Zadok, Y. (2009). Robotics projects and learning concepts in science, technology and problem solving. *International Journal of Technology and Design Education*, 19(3), 289–307. [https://doi.org/10.1007/s10798-007-9043-](https://doi.org/10.1007/s10798-007-9043-3)

- Beatty, A. (2011). *Successful STEM Education: A Workshop Summary Programs for K-12 STEM Education; National Research Council.*
- Belbase, S., Mainali, B. R., Kasemsukpipat, W., Tairab, H., Gochoo, M., & Jarrah, A. (2021). At the dawn of science, technology, engineering, arts, and mathematics (STEAM) education: prospects, priorities, processes, and problems. *International Journal of Mathematical Education in Science and Technology*, 53(11), 2919–2955.  
<https://doi.org/10.1080/0020739x.2021.1922943>
- Bell, D., Devés, R., Dyasi, H., Fernández De La Garza, G., Léna, P., Millar, R., Reiss, M., Rowell, P., & Yu, W. (2015). *Big Ideas of Science Education Edited by Wynne Harlen.*
- Benedettelli, D. (2021). *The LEGO MINDSTORMS Robot Inventor Activity Book.* No Starch Press.
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978–988.  
<https://doi.org/10.1016/j.compedu.2011.10.006>
- Bertrand, M. G., & Namukasa, I. K. (2020). STEAM education: student learning and transferable skills. *Journal of Research in Innovative Teaching & Learning*, 13(1), 43–56. <https://doi.org/10.1108/jrit-01-2020-0003>
- Bethke Wendell, K., & Rogers, C. (2013). Engineering Design-Based Science, Science Content Performance, and Science Attitudes in Elementary School. *Journal of Engineering Education*, 102(4), 513–540.  
<https://doi.org/10.1002/jee.20026>
- Biggs, G., & Macdonald, B. (2003). A Survey of Robot Programming Systems.
- Birk, A. (2011). What is Robotics? An Interdisciplinary Field Is Getting Even More

- Diverse [Education]. *IEEE Robotics & Automation Magazine*, 18(4), 94–95.  
<https://doi.org/10.1109/mra.2011.943235>
- Bischoff, R., Kazi, A., & Seyfarth, M. (2002, September 1). The MORPHA style guide for icon-based programming. *IEEE Xplore*.  
<https://doi.org/10.1109/ROMAN.2002.1045668>
- Blackwell, A. (2002). *What is Programming?*
- Blumenfeld, P., Fishman, B. J., Krajcik, J., Marx, R. W., & Soloway, E. (2000).  
Creating Usable Innovations in Systemic Reform: Scaling Up Technology-  
Embedded Project-Based Science in Urban Schools. *Educational  
Psychologist*, 35(3), 149–164. [https://doi.org/10.1207/s15326985ep3503\\_2](https://doi.org/10.1207/s15326985ep3503_2)
- Boy, G. (2013). *From STEM to STEAM: Toward a Human-Centred Education,  
Creativity & Learning Thinking*. <https://doi.org/10.1145/2501907.2501934>
- Bragança, S., Costa, E., Castellucci, I., & Arezes, P. M. (2019). A Brief Overview of  
the Use of Collaborative Robots in Industry 4.0: Human Role and Safety.  
*Studies in Systems, Decision and Control*, 641–650.  
[https://doi.org/10.1007/978-3-030-14730-3\\_68](https://doi.org/10.1007/978-3-030-14730-3_68)
- Bratzel, B., & Torok, R. (2022a). *Getting Started with LEGO® MINDSTORMS*. No  
Starch Press.
- Bratzel, B., & Torok, R. (2022b). *Mastering LEGO® MINDSTORMS*. No Starch  
Press.
- Breiner, J. M., Harkness, S. S., Johnson, C. C., & Koehler, C. M. (2012). What Is  
STEM? A Discussion About Conceptions of STEM in Education and  
Partnerships. *School Science and Mathematics*, 112(1), 3–11.  
<https://doi.org/10.1111/j.1949-8594.2011.00109.x>
- Brown, R., & Merrill, C. (2011). *Understanding STEM: Current perceptions*



*Teaching and Learning Truss Design View project Beam Design and Analysis View project.*

- Bybee, R. W. (2010). What Is STEM Education? *Science*, 329(5995), 996–996.  
<https://doi.org/10.1126/science.1194998>
- Çakır, R., Korkmaz, Ö., İdil, Ö., & Uğur Erdoğan, F. (2021). The effect of robotic coding education on preschoolers' problem solving and creative thinking skills. *Thinking Skills and Creativity*, 40, 100812.  
<https://doi.org/10.1016/j.tsc.2021.100812>
- Capraro, R., Capraro, M., & Morgan, J. (2013). *STEM Project-Based Learning An Integrated Science, Technology, Engineering, and Mathematics (STEM) Approach (2nd Edition)*.
- Carbonaro, M., Rex, M., & Chambers, J. (2004). *Using LEGO Robotics in a Project-Based Learning Environment*.
- Casler-Failing, S. L. (2018). The Effects of Integrating LEGO Robotics Into a Mathematics Curriculum to Promote the Development of Proportional Reasoning. *Proceedings of the Interdisciplinary STEM Teaching and Learning Conference*, 2(1). <https://doi.org/10.20429/stem.2018.020105>
- Catlin, D., & Woollard, J. (2014). *Educational Robots and Computational Thinking*.
- Causo, A. (2016). *A Review on the Use of Robots in Education and Young Children Show & Tell Robotic Puppet for Preschool Education View project Autonomous Picking Robot View project*.
- Çetin, M., & Demircan, H. Ö. (2018). Empowering technology and engineering for STEM education through programming robots: a systematic literature review. *Early Child Development and Care*, 190(9), 1323–1335.  
<https://doi.org/10.1080/03004430.2018.1534844>

- Changpetch, S., & Seechaliao, T. (2019). The Propose of an Instructional Model Based on STEM Education Approach for Enhancing the Information and Communication Technology Skills for Elementary Students in Thailand. *International Education Studies*, 13(1), 69.  
<https://doi.org/10.5539/ies.v13n1p69>
- Chesky, N. Z., & Wolfmeyer, M. R. (2015). *Philosophy of STEM Education*.  
<https://doi.org/10.1057/9781137535467.0001>
- Choi, Y., Choi, M., Oh, M. (Moon), & Kim, S. (Sam). (2019). Service robots in hotels: understanding the service quality perceptions of human-robot interaction. *Journal of Hospitality Marketing & Management*, 29(6), 613–635.  
<https://doi.org/10.1080/19368623.2020.1703871>
- Chronis, C., & Varlamis, I. (2022). FOSSBot: An Open Source and Open Design Educational Robot. *Electronics*, 11(16), 2606.  
<https://doi.org/10.3390/electronics11162606>
- Cocota, J. A. N., D'Angelo, T., & de Barros Monteiro, P. M. (2015). A Project-Based Learning Experience in the Teaching of Robotics. *IEEE Revista Iberoamericana de Tecnologias Del Aprendizaje*, 10(4), 302–309.  
<https://doi.org/10.1109/rita.2015.2486458>
- Colak, I., & Yildirim, D. (2009). *Evolving a Line Following Robot to Use in Shopping Centers for Entertainment*.
- Cooper, M. M. (2020). The Crosscutting Concepts: Critical Component or “Third Wheel” of Three-Dimensional Learning?. *Journal of Chemical Education*, 97(4), 903–909. <https://doi.org/10.1021/acs.jchemed.9b01134>
- Cuperman, D., & Verner, I. M. (2013). Learning through creating robotic models of biological systems. *International Journal of Technology and Design*

- Education*, 23(4), 849–866. <https://doi.org/10.1007/s10798-013-9235-y>
- Daugherty, M. (2013). *The Prospect of an “A” in STEM Education*.
- Daugherty, M. K., Carter, V., & Swagerty, L. (2014). Elementary STEM Education: The Future for Technology and Engineering Education? *Journal of STEM Teacher Education*, 49(1). <https://doi.org/10.30707/jste49.1daugherty>
- David, M. (2013). Organising, valuing and improving the engineering design process. *Journal of Engineering Design*, 24(7), 524–545. <https://doi.org/10.1080/09544828.2013.776214>
- Di, C., Zhou, Q., Shen, J., Li, L., Zhou, R., & Lin, J. (2021). Innovation event model for STEM education: A constructivism perspective. *STEM Education*, 1(1), 60. <https://doi.org/10.3934/steme.2021005>
- Dischino, M., Donnelly, J., & Massa, N. (2011). *Increasing the STEM Pipeline through Problem-Based Learning*.
- Donahoe, D. (2013). *"The Definition of STEM?"*.
- Draper, R. (2002). School Mathematics Reform, Constructivism, and Literacy: A Case for Literacy Instruction in the Reform-Oriented Math Classroom. *Source: Journal of Adolescent & Adult Literacy*, 45(6).
- Duschl, R. (2012). *The Second Dimension- Crosscutting Concepts Understanding A Framework for K-12 Science Education*.
- Eckerdal, A., & Berglund, A. (2005). *What Does It Take to Learn “Programming Thinking”?*
- Eguchi, A. (2014). *Robotics as a Learning Tool for Educational Transformation*.
- El-Deghaidy, H., & Mansour, N. (2015). Science Teachers’ Perceptions of STEM Education: Possibilities and Challenges. *International Journal of Learning and Teaching*. <https://doi.org/10.18178/ijlt.1.1.51-54>

- Erduran, S. (2020). Nature of “STEM”? Engineering, and Mathematics in Education. *Nature of “STEM”? Epistemic Underpinnings of Integrated Science, Technology, Engineering, and Mathematics in Education*.  
<https://doi.org/10.1007/s11191-020-00150-6>
- Euefueno, W. (2019). *Project-/Problem-Based Learning in STEM: Impacts on Student Project-/Problem-Based Learning in STEM: Impacts on Student Learning Learning*.
- Fagin, B. S., Merkle, L. D., & Eggers, T. W. (2001). Teaching computer science with robotics using Ada/Mindstorms 2.0. *ACM SIGAda Ada Letters*, XXI(4), 73–78.  
<https://doi.org/10.1145/507546.507592>
- Fernandez, J. M. A., & Vincze, M. (2018). *Towards a Definition of Educational Robotics*. <https://doi.org/10.15203/3187-22-1>
- Flowers, T., & Gossett, K. (2002). *TEACHING PROBLEM SOLVING, COMPUTING, AND INFORMATION TECHNOLOGY WITH ROBOTS*.
- Gao, Y., & Chien, S. (2017). Review on space robotics: Toward top-level science through space exploration. *Sci. Robot*, 2, 28.
- Gareth Branwyn. (2003). *Absolute Beginner’s Guide to Building Robots*. Que Publishing.
- Geist, E. (2016). Robots, Programming and Coding, Oh My! *Childhood Education*, 92(4), 298–304. <https://doi.org/10.1080/00094056.2016.1208008>
- Gerecke, U., & Wagner, B. (2013). The Challenges and Benefits of Using Robots in Higher Education. *Intelligent Automation & Soft Computing*, 13(1), 29–43.  
<https://doi.org/10.1080/10798587.2007.10642948>
- Gresnigt, R., Taconis, R., van Keulen, H., Gravemeijer, K., & Baartman, L. (2014). Promoting science and technology in primary education: a review of

integrated curricula. *Studies in Science Education*, 50(1), 47–84.

<https://doi.org/10.1080/03057267.2013.877694>

Gross, K., & Gross, S. (2016). TRANSFORMATION: Constructivism, Design

Thinking, and Elementary STEAM. *Art Education*, 69(6), 36–43.

<https://doi.org/10.1080/00043125.2016.1224869>

GÜLERYÜZ, H., & DİLBER, R. (2021). STEM ACTIVITIES WITH ROBOTIC CODING; THE EFFECT ON AWARENESS OF TEACHER CANDIDATES REGARDING ITS USE IN SCIENCE LESSONS. *International Journal of Engineering Technologies and Management Research*, 8(11), 37–54.

<https://doi.org/10.29121/ijetmr.v8.i11.2021.1063>

Guzey, S. S., Moore, T. J., & Harwell, M. (2016). Building Up STEM: An Analysis of Teacher-Developed Engineering Design-Based STEM Integration Curricular Materials. *Journal of Pre-College Engineering Education Research (J-PEER)*, 6(1). <https://doi.org/10.7771/2157-9288.1129>

Haik, Y., Shahin, T. M., & Sivaloganathan, S. (2010). Engineering Design Process. In *Google Books*. Cengage Learning.

[https://books.google.gr/books?id=HgwVzgEACAAJ&hl=el&source=gbs\\_book\\_other\\_versions](https://books.google.gr/books?id=HgwVzgEACAAJ&hl=el&source=gbs_book_other_versions)

Harris, C., Krajcik, J., Pellegrino, J., Mcelhaney, K., Debarger, A., Dahsah, C., Damelin, D., D', C., Louis, A., Dibello, V., Gane, B., & Lee, J. (2016). *Constructing Assessment Tasks that Blend Disciplinary Core Ideas, Crosscutting Concepts, and Science Practices for Classroom Formative Applications*.

Hildebrandt, C., von Stein, M., Woodlief, T., & Elbaum, S. (2022). Preparing software engineers to develop robot systems. *Proceedings of the ACM/IEEE*

*44th International Conference on Software Engineering: Software Engineering Education and Training.*

<https://doi.org/10.1145/3510456.3514161>

Hinojo-Lucena, F.-J., Dúo-Terrón, P., Ramos Navas-Parejo, M., Rodríguez-Jiménez, C., & Moreno-Guerrero, A.-J. (2020). Scientific Performance and Mapping of the Term STEM in Education on the Web of Science. *Sustainability*, *12*(6), 2279. <https://doi.org/10.3390/su12062279>

Hood, C. S., & Hood, D. J. (2005). Teaching programming and language concepts using LEGOs®. *ACM SIGCSE Bulletin*, *37*(3), 19–23.

<https://doi.org/10.1145/1151954.1067454>

Hurst, C. (2015). Thinking Big about Mathematics, Science, and Technology: Effective Teaching STEMS from big ideas. *International Journal of Innovation in Science and Mathematics Education*, *23*(3), 11–21.

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531–2544. <https://doi.org/10.1007/s10639-018-9729-z>

Jung, S., & Won, E. (2018). Systematic Review of Research Trends in Robotics Education for Young Children. *Sustainability*, *10*(4), 905.

<https://doi.org/10.3390/su10040905>

KALOTI-HALLAK, F., ARMONI, M., & BEN-ARI, M. (2019). The Effect of Robotics Activities on Learning the Engineering Design Process. *Informatics in Education*, *18*(1), 105–129. <https://doi.org/10.15388/infedu.2019.05>

Karahoca, D., Karahoca, A., & Uzunboylub, H. (2011). Robotics teaching in primary school education by project based learning for supporting science and

- technology courses. *Procedia Computer Science*, 3, 1425–1431.  
<https://doi.org/10.1016/j.procs.2011.01.025>
- Karch, M. (2014). *Build and Program Your Own LEGO Mindstorms EV3 Robots*.  
 Que Publishing.
- Karim, M., Lemaignan, S., & Mondada, F. (2015). *A review: Can robots reshape K-12 STEM education?*
- Katz, D. S., & Some, R. R. (2003). NASA advances robotic space exploration. *Computer*, 36(1), 52–61. <https://doi.org/10.1109/mc.2003.1160056>
- Kelley, T. R., & Knowles, J. G. (2016). A conceptual framework for integrated STEM education. *International Journal of STEM Education*, 3(1).  
<https://doi.org/10.1186/s40594-016-0046-z>
- Keshav, R., Koul Aditya V., Priyam, S., & K. S, M. (2020). Applications of Artificial Intelligence in the Game of Football: The Global Perspective. *International Refereed Social Sciences Journal*. <https://doi.org/10.18843/rwjasc/v11i2/03>
- Khamphroo, M., Kwankeo, N., Kaemarungsi, K., & Fukawa, K. (2017). *MicroPython-based Educational Mobile Robot for Computer Coding Learning*.
- Khan, A., & Anwar, Y. (2020). Robots in Healthcare: A Survey. *Advances in Intelligent Systems and Computing*, 280–292. [https://doi.org/10.1007/978-3-030-17798-0\\_24](https://doi.org/10.1007/978-3-030-17798-0_24)
- Khanlari, A. (2015). Teachers' perceptions of the benefits and the challenges of integrating educational robots into primary/elementary curricula. *European Journal of Engineering Education*, 41(3), 320–330.  
<https://doi.org/10.1080/03043797.2015.1056106>
- Khurshid, J., & Bing-Rong, H. (2004). *Military Robots -A Glimpse from Today and*

*Tomorrow.*

- Koc, M., & Demirbilek, M. (2018). *What is Technology (T) and What does it Hold for STEM Education? Definitions, Issues, and Tools.*
- Kokotsaki, D., Menzies, V., & Wiggins, A. (2016). Project-based learning: A review of the literature. *Improving Schools, 19*(3), 267–277.  
<https://doi.org/10.1177/1365480216659733>
- Kolberg, E., Reich, Y., & Levin, I. (2014). Designing winning robots by careful design of their development process. *Research in Engineering Design, 25*(2), 157–183. <https://doi.org/10.1007/s00163-014-0171-y>
- Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntambekar, S., & Ryan, M. (2003). Problem-Based Learning Meets Case-Based Reasoning in the Middle-School Science Classroom: Putting Learning by Design(tm) Into Practice. *Journal of the Learning Sciences, 12*(4), 495–547. [https://doi.org/10.1207/s15327809jls1204\\_2](https://doi.org/10.1207/s15327809jls1204_2)
- Konijn, E. A., Smakman, M., & Berghe, R. (2020). Use of Robots in Education. *The International Encyclopedia of Media Psychology*, 1–8.  
<https://doi.org/10.1002/9781119011071.iemp0318>
- Koonce, D., & Anderson, C. (2011). *AC 2011-289: WHAT IS STEM? What is STEM?*
- Koski, M.-I., Kurhila, J., & Pasanen, T. (2008). *Why Using Robots to Teach Computer Science can be Successful Theoretical Reflection to Andragogy and Minimalism.*
- Krajcik, J., & Delen, I. (2016). How to Support Learners in Developing Usable and Lasting Knowledge of STEM. *International Journal of Education in Mathematics, Science and Technology, 5*(1), 21.  
<https://doi.org/10.18404/ijemst.16863>



- Kubilinskiene, S., Zilinskiene, I., Dagiene, V., & Sinkevičius, V. (2017). Applying Robotics in School Education: a Systematic Review. *Baltic Journal of Modern Computing*, 5(1), 50–69. <https://doi.org/10.22364/bjmc.2017.5.1.04>
- Kvesko, S. B., Kvesko, N. G., Korniyenko, A. A., & Kabanova, N. N. (2018). Educational robotics as an Innovative teaching practice using technology: minimization of risks. *IOP Conference Series: Materials Science and Engineering*, 363, 012004. <https://doi.org/10.1088/1757-899x/363/1/012004>
- Kwon, H., Capraro, R. M., & Capraro, M. M. (2021). When I Believe, I Can: Success STEMs from My Perceptions. *Canadian Journal of Science, Mathematics and Technology Education*, 21(1), 67–85. <https://doi.org/10.1007/s42330-020-00132-4>
- LaForce, M., Noble, E., & Blackwell, C. (2017). Problem-Based Learning (PBL) and Student Interest in STEM Careers: The Roles of Motivation and Ability Beliefs. *Education Sciences*, 7(4), 92. <https://doi.org/10.3390/educsci7040092>
- LaForce, M., Noble, E., King, H., Century, J., Blackwell, C., Holt, S., Ibrahim, A., & Loo, S. (2016). The eight essential elements of inclusive STEM high schools. *International Journal of STEM Education*, 3(1). <https://doi.org/10.1186/s40594-016-0054-z>
- Land, M. H. (2013). Full STEAM Ahead: The Benefits of Integrating the Arts Into STEM. *Procedia Computer Science*, 20, 547–552. <https://doi.org/10.1016/j.procs.2013.09.317>
- Lawhead, P. B., Duncan, M. E., Bland, C. G., Goldweber, M., Schep, M., Barnes, D. J., & Hollingsworth, R. G. (2002). A road map for teaching introductory programming using LEGO© mindstorms robots. *ACM SIGCSE Bulletin*, 35(2), 191–201. <https://doi.org/10.1145/782941.783002>

- Lewis, T. (2006). Design and inquiry: Bases for an accommodation between science and technology education in the curriculum? *Journal of Research in Science Teaching*, 43(3), 255–281. <https://doi.org/10.1002/tea.20111>
- Li, Y. (2018). Four years of development as a gathering place for international researchers and readers in STEM education. *International Journal of STEM Education*, 5(1). <https://doi.org/10.1186/s40594-018-0153-0>
- Liang, H.-N., Fleming, C., Man, K., & Tillo, T. (2013). *A first introduction to programming for first-year students at a Chinese university using LEGO MindStorms*.
- Liao, C. (2016). From Interdisciplinary to Transdisciplinary: An Arts-Integrated Approach to STEAM Education. *Art Education*, 69(6), 44–49. <https://doi.org/10.1080/00043125.2016.1224873>
- Lloyd, K. (2017). Benefits of Art Education: A Review of the Literature. *Scholarship and Engagement in Education*, 1.
- Lozano-Perez, T. (1983). Robot programming. *Proceedings of the IEEE*, 71(7), 821–841. <https://doi.org/10.1109/proc.1983.12681>
- Lykke, M., Coto, M., Mora, S., Vandell, N., & Jantzen, C. (2014). *Motivating programming students by Problem Based Learning and LEGO robots*.
- Lytridis, C., Bazinas, C., Kaburlasos, V., Ferelis, V., Vassileva-Aleksandrova, V., Jaki, A., Youssfi, M., & Mestari, M. (2019). *Social Robots as Cyber-Physical Actors in Entertainment and Education*.
- Madden, L., Beyers, J., & O'brien, S. (2016). The Importance of STEM Education in the Elementary Grades: Learning from Pre-service and Novice Teachers' Perspectives. *Electronic Journal of Science Education*, 20(5).
- Maeda, J. (2013). STEM + Art = STEAM. *The STEAM Journal*, 1(1), 34.

<https://doi.org/10.5642/steam.201301.34>

- Malley, S. M., & Silverstein, L. B. (2014). Examining the Intersection of Arts Education and Special Education. *Arts Education Policy Review*, 115(2), 39–43. <https://doi.org/10.1080/10632913.2014.883894>
- Mangold, J., & Robinson, S. (2013). *The engineering design process as a problem solving and learning tool in K- 12 classrooms*.
- Marrero, M., Gunning, A., & Germain-Williams, T. (2014). *What is Stem Education? What is STEM Education?*
- Martin, F. G., Sargent, R., & Oberoi, P. (1992). *The 6.270 Robot Builder's Guide*.
- Martín-Páez, T., Aguilera, D., Perales-Palacios, F. J., & Vílchez-González, J. M. (2019). What are we talking about when we talk about STEM education? A review of literature. *Science Education*, 103(4), 799–822. <https://doi.org/10.1002/sce.21522>
- Mathers, N., Goktogen, A., Rankin, J., & Anderson, M. (2012). Robotic Mission to Mars: Hands-on, minds-on, web-based learning. *Acta Astronautica*, 80, 124–131. <https://doi.org/10.1016/j.actaastro.2012.06.003>
- Maurer, A. (2021). *Smart Robotics with LEGO MINDSTORMS Robot Inventor*. Packt Publishing Ltd.
- Maurer, A. (2022). *Design Innovative Robots with LEGO SPIKE Prime*. Packt Publishing.
- May, S., Todd, K., Paneto, S., Kipling, B., Kollmann, E., Reich, C., Rappolt-Schlichtmann Edtogether, G., & Kent, K. (2019). *Empowering Learners through Effective Emotional Engagement: Project Review*.
- Mccomb, G. (2001). *The robot builder's bonanza*. McGraw-Hill.
- Mcdonald, C. (2016). STEM Education: A review of the contribution of the

- disciplines of science, technology, engineering and mathematics. *Science Education International*, 27, 530–569.
- McDonald, S., & Howell, J. (2011). Watching, creating and achieving: Creative technologies as a conduit for learning in the early years. *British Journal of Educational Technology*, 43(4), 641–651. <https://doi.org/10.1111/j.1467-8535.2011.01231.x>
- Mesutoglu, C., & Baran, E. (2020). Examining the Development of Middle School Science Teachers' Understanding of Engineering Design Process. *International Journal of Science and Mathematics Education*, 18(8), 1509–1529. <https://doi.org/10.1007/s10763-019-10041-0>
- Metevier, A., Hunter, L., & Seagroves, S. (2022). *UC Santa Cruz Professional Development Title ISEE's Framework of Six Elements to Guide the Design, Teaching, and Assessment of Authentic and Inclusive STEM Learning Experiences* Publication Date.
- Miller, M. R., & Miller, R. (2017). *Robots and Robotics*.
- Mukhopadhyay, K. (2021). *ROBOTICS IN PHYSICAL EDUCATION AND SPORTS*.
- National Research Council. (2009). Learning Science in Informal Environments: People, Places, and Pursuits. In *nap.nationalacademies.org*. <https://nap.nationalacademies.org/catalog/12190/learning-science-in-informal-environments-people-places-and-pursuits>
- Nag, S., Katz, J. G., & Saenz-Otero, A. (2013). Collaborative gaming and competition for CS-STEM education using SPHERES Zero Robotics. *Acta Astronautica*, 83, 145–174. <https://doi.org/10.1016/j.actaastro.2012.09.006>
- Nicolescu, B. (2010). Methodology of Transdisciplinarity–Levels of Reality, Logic of the Included Middle and Complexity. *Transdisciplinary Journal of*

*Engineering & Science*, 1(1). <https://doi.org/10.22545/2010/0009>

- Noh, J., & Lee, J. (2019). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, 68(1), 463–484. <https://doi.org/10.1007/s11423-019-09708-w>
- Nor, W., Binti, F., Husin, W., Fadzilah, W., Nurazidawati, M., Arsad, O., Othman, L., Halim, M., Rasul, K., Osman, Z., & Iksan. (2016). Fostering students' 21st century skills through Project Oriented Problem Based Learning (POPBL) in integrated STEM education program Fostering students' 21st century skills through Project Oriented Problem Based Learning (POPBL) in integrated STEM education program Fostering students' 21st century skills through Project Oriented Problem Based Learning (POPBL) in integrated STEM education program. *Asia-Pacific Forum on Science Learning and Teaching*, 17(1).
- Oddie, A., Hazlewood, P., Blakeway, S., & Whitfield, A. (2010). Introductory Problem Solving and Programming: Robotics Versus Traditional Approaches. *Innovation in Teaching and Learning in Information and Computer Sciences*, 9(2), 1–11. <https://doi.org/10.11120/ital.2010.09020011>
- Oh, K.-M., Kim, J.-H., & Kim, M.-S. (2005). *Development of Humanoid Robot Design Process*.
- Okeke, N., & Unachukwu, G. (2022). The influence of STEAM education in post Covid-19 era on students' mastery and application of knowledge in music in secondary schools in Anambra State, Nigeria. *SOCIAL SCIENCES and EDUCATION RESEARCH REVIEW*, 9, 149–156. <https://doi.org/10.5281/zenodo.6795732>

- Osada, J., Sato, M., & Ohnaka, S. (2006). *The Scenario and Design Process of Childcare Robot, PaPeRo*.
- Pan, Z., Polden, J., Larkin, N., Van Duin, S., & Norrish, J. (2012). Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, 28(2), 87–94.  
<https://doi.org/10.1016/j.rcim.2011.08.004>
- Papert, S. (1980). *Mindstorms—Children, Computers and Powerful Ideas*. New York: Basic Books, Inc.
- Papert, S. (1993). *The children s machine: Rethinking school in the age of the computer*. New York: Basic Books.
- Park, N., & Ko, Y. (2012). Computer Education’s Teaching-Learning Methods Using Educational Programming Language Based on STEAM Education \* Namje Park. *LNCS*, 7513, 320–327.
- Patel, J. (2019). *The Constructivist Approach to Curriculum Integration of STEM Education*.
- Pawel Kmieć. (2017). *The unofficial LEGO technic builder’s guide*. No Starch Press.
- Perignat, E., & Katz-Buonincontro, J. (2019). STEAM in practice and research: An integrative literature review. *Thinking Skills and Creativity*, 31, 31–43.  
<https://doi.org/10.1016/j.tsc.2018.10.002>
- Piaget, J. (1970). Science of education and the psychology of the child. Trans. D. Coltman. Orion.
- Piro, J. (2010). *as Going From STEM to STEAM COMMENTARY Going From STEM to STEAM The Arts Have a Role in America’s Future, Too*.
- Quinn, H., Schweingruber, H., & Keller, T. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*.

<https://doi.org/10.17226/13165>

- Radziwill, N., Benton, M., & Moellers, C. (2015). From STEM to STEAM: Reframing What it Means to Learn. *STEAM*, 2(1), 1–7.  
<https://doi.org/10.5642/steam.20150201.3>
- Reeves, T. C., Herrington, J., & Oliver, R. (2004). A development research agenda for online collaborative learning. *Educational Technology Research and Development*, 52(4), 53–65. <https://doi.org/10.1007/bf02504718>
- Ringwood, J. V., Monaghan, K., & Maloco, J. (2005). Teaching engineering design through Lego Mindstorms. *European Journal of Engineering Education*, 30(1), 91–104. <https://doi.org/10.1080/03043790310001658587>
- RobotShop Distribution Inc. (2008). *History of Robotics: Timeline*.
- Rodríguez, C., Guzmán, J., Berenguel, M., & Dormido, S. (2016). *Teaching real-time programming using mobile robots*.  
<https://doi.org/10.1016/j.ifacol.2016.07.145>
- Rollins, M. (2013). *Practical LEGO Technics*. Apress.
- Rossano, G. F., Martinez, C., Hedelind, M., Murphy, S., & Fuhlbrigge, T. A. (2013). *Easy Robot Programming Concepts: An Industrial Perspective*.
- Sanders, M. (2008). *STEM, STEM Education, STEMmania*.  
<https://vtechworks.lib.vt.edu/bitstream/handle/10919/51616/STEMmania.pdf>
- Scaradozzi, D., Screpanti, L., & Cesaretti, L. (2019). Towards a Definition of Educational Robotics: A Classification of Tools, Experiences and Assessments. *Smart Learning with Educational Robotics*, 63–92.  
[https://doi.org/10.1007/978-3-030-19913-5\\_3](https://doi.org/10.1007/978-3-030-19913-5_3)
- Sen, C., Ay, Z. S., & Kiray, S. A. (2021). Computational thinking skills of gifted and talented students in integrated STEM activities based on the engineering

- design process: The case of robotics and 3D robot modeling. *Thinking Skills and Creativity*, 42, 100931. <https://doi.org/10.1016/j.tsc.2021.100931>
- Simarro, C., & Couso, D. (2021). Engineering practices as a framework for STEM education: a proposal based on epistemic nuances. *International Journal of STEM Education*, 8(1). <https://doi.org/10.1186/s40594-021-00310-2>
- Simon, M. A. (1995). Reconstructing Mathematics Pedagogy from a Constructivist Perspective. *Journal for Research in Mathematics Education*, 26(2), 114. <https://doi.org/10.2307/749205>
- Singer, J. E., Ross, J. M., & Jackson-Lee, Y. (2016). Professional Development for the Integration of Engineering in High School STEM Classrooms. *Journal of Pre-College Engineering Education Research (J-PEER)*, 6(1). <https://doi.org/10.7771/2157-9288.1130>
- Sisman, B., & Kucuk, S. (2019). An Educational Robotics Course: Examination of Educational Potentials and Pre-service Teachers' Experiences. *International Journal of Research in Education and Science (IJRES)*, 5(2), 510–531.
- Smith, K., Douglas, T., & Cox, M. (2009). *Supportive Teaching and Learning Strategies in STEM Education*.
- Sousa, D. A., & Pilecki, T. (2018). *From STEM to STEAM Second Edition*.
- Soysal, O. (2005). *A Brief History of Robotics*.
- Sparrow, R., & Howard, M. (2020). Robots in agriculture: prospects, impacts, ethics, and policy. *Precision Agriculture*, 22(3), 818–833. <https://doi.org/10.1007/s11119-020-09757-9>
- Spolaôr, N., & Benitti, F. B. Vavassori. (2017). Robotics applications grounded in learning theories on tertiary education: A systematic review. *Computers & Education*, 112, 97–107. <https://doi.org/10.1016/j.compedu.2017.05.001>



- Stief, P., Dantan, J.-Y., Etienne, A., & Siadat, A. (2018). *The Design Thinking Approach to students STEAM projects*.  
<https://doi.org/10.1016/j.procir.2020.03.100>
- Su, R., & Rounds, J. (2015). All STEM fields are not created equal: People and things interests explain gender disparities across STEM fields. *Frontiers in Psychology*, 6. <https://doi.org/10.3389/fpsyg.2015.00189>
- Sucheta, K. (2022). Effectiveness of “STREAM based Learning Approach” on Achievement in Science of Elementary School Students. *International Journal of Innovative Science and Research Technology*, 7(5).
- Szczecinski, N. S., Hunt, A. J., & Quinn, R. D. (2017). Design process and tools for dynamic neuromechanical models and robot controllers. *Biological Cybernetics*, 111(1), 105–127. <https://doi.org/10.1007/s00422-017-0711-4>
- SZEGEDI, P., KORONVÁRY, P., & BÉKÉSI, B. (2017). THE USE OF ROBOTS IN MILITARY OPERATIONS. *SCIENTIFIC RESEARCH and EDUCATION in the AIR FORCE*, 19(1), 221–230. <https://doi.org/10.19062/2247-3173.2017.19.1.25>
- The National Academy of Sciences. (2011). *Successful K-12 STEM Education - Identifying Effective Approaches in Science, Technology, Engineering, and Mathematics*.
- The contribution of Derek Bell, W., De La Garza, G., Léna, P., Millar, R., Reiss, M., & Rowell, P. (2010). *Principles and big ideas of science education Edited by Wynne Harlen*.
- Thomas, K., & Huffman, D. (2020). *Challenges and Opportunities for Transforming From STEM to STEAM Education A volume in the Advances in Educational Technologies and Instructional Design (AETID) Book Series*.

- Too, S., Shi, K., Siew Feon, N., Nordin, M., & Kelantan, U. (2021). Steam The Best Solution For Science Teaching ?: A Review. *International Journal of Aquatic Science*, 12, 2021.
- Varney, M. W., Janoudi, A., Aslam, D. M., & Graham, D. (2012). Building Young Engineers: TASEM for Third Graders in Woodcreek Magnet Elementary School. *IEEE Transactions on Education*, 55(1), 78–82.  
<https://doi.org/10.1109/te.2011.2131143>
- Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM*, 62(8), 22–25. <https://doi.org/10.1145/3341221>
- Weintrop, D., & Wilensky, U. (2017). Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. *ACM Transactions on Computing Education*, 18(1), 1–25.  
<https://doi.org/10.1145/3089799>
- Widya, Rifandi, R., & Laila Rahmi, Y. (2019). STEM education to fulfil the 21<sup>st</sup> century demand: a literature review. *Journal of Physics: Conference Series*, 1317(1), 012208. <https://doi.org/10.1088/1742-6596/1317/1/012208>
- Williams, A. B. (2003). The qualitative impact of using LEGO MINDSTORMS robots to teach computer engineering. *IEEE Transactions on Education*, 46(1), 206–206. <https://doi.org/10.1109/te.2002.808260>
- WINARNO, N. (2020). The Steps of the Engineering Design Process (EDP) in Science Education: A Meta-Analysis. *Journal for the Education of Gifted Young Scientists*. <https://doi.org/10.17478/jegys.766201>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33.  
<https://doi.org/10.1145/1118178.1118215>
- Wong, K.-W. (2001). *Teaching Programming with Lego RCX Robots*.

- Xie, Y., Fang, M., & Shauman, K. (2015). STEM Education. *Annual Review of Sociology*, 41(1), 331–357. <https://doi.org/10.1146/annurev-soc-071312-145659>
- Yates, D. R., Vaessen, C., & Roupret, M. (2011). *From Leonardo to da Vinci: the history of robot-assisted surgery in urology*.
- Yıldız, T., & Seferoğlu, S. (2021). The Effect of Robotic Programming on Coding Attitude and Computational Thinking Skills toward Self-Efficacy Perception. *Journal of Learning and Teaching in Digital Age*, 2021(2), 101–116.
- Yoshihito Isogawa. (2011). *The LEGO technic idea book. 2 Wheeled wonders*. San Francisco, Ca No Starch Press.
- Yoshihito Isogawa. (2015). *The LEGO Power Functions Idea Book, Volume 1*. No Starch Press.
- Yoshihito Isogawa. (2016a). *The LEGO power functions idea book. Volume 1, Machines and mechanisms*. No Starch Press.
- Yoshihito Isogawa. (2016b). *The LEGO power functions idea book. Volume 2, Cars and contraptions*. No Starch Press.
- Yoshihito Isogawa. (2021a). *LEGO Technic Non-Electric Models: Clever Contraptions*. No Starch Press.
- Yoshihito Isogawa. (2021b). *The LEGO MINDSTORMS Robot Inventor Idea Book*. No Starch Press.
- Zadok, Y. (2019). Project-based learning in robotics meets junior high school. *Journal of Engineering, Design and Technology*, 18(5), 941–958. <https://doi.org/10.1108/jedt-01-2019-0023>
- Zarske, M., Sullivan, J., Carlson, L., & Yowell, J. (2004). *Teachers Teaching Teachers: Linking K-12 Engineering Curricula with Teacher Professional*

*Development.*

Ziaeeafard, S., Miller, M. H., Rastgaar, M., & Mahmoudian, N. (2017). Co-robotics hands-on activities: A gateway to engineering design and STEM learning.

*Robotics and Autonomous Systems*, 97, 40–50.

<https://doi.org/10.1016/j.robot.2017.07.013>

Zollman, A. (2012). Learning for STEM Literacy: STEM Literacy for Learning.

*School Science and Mathematics*, 112(1), 12–19.

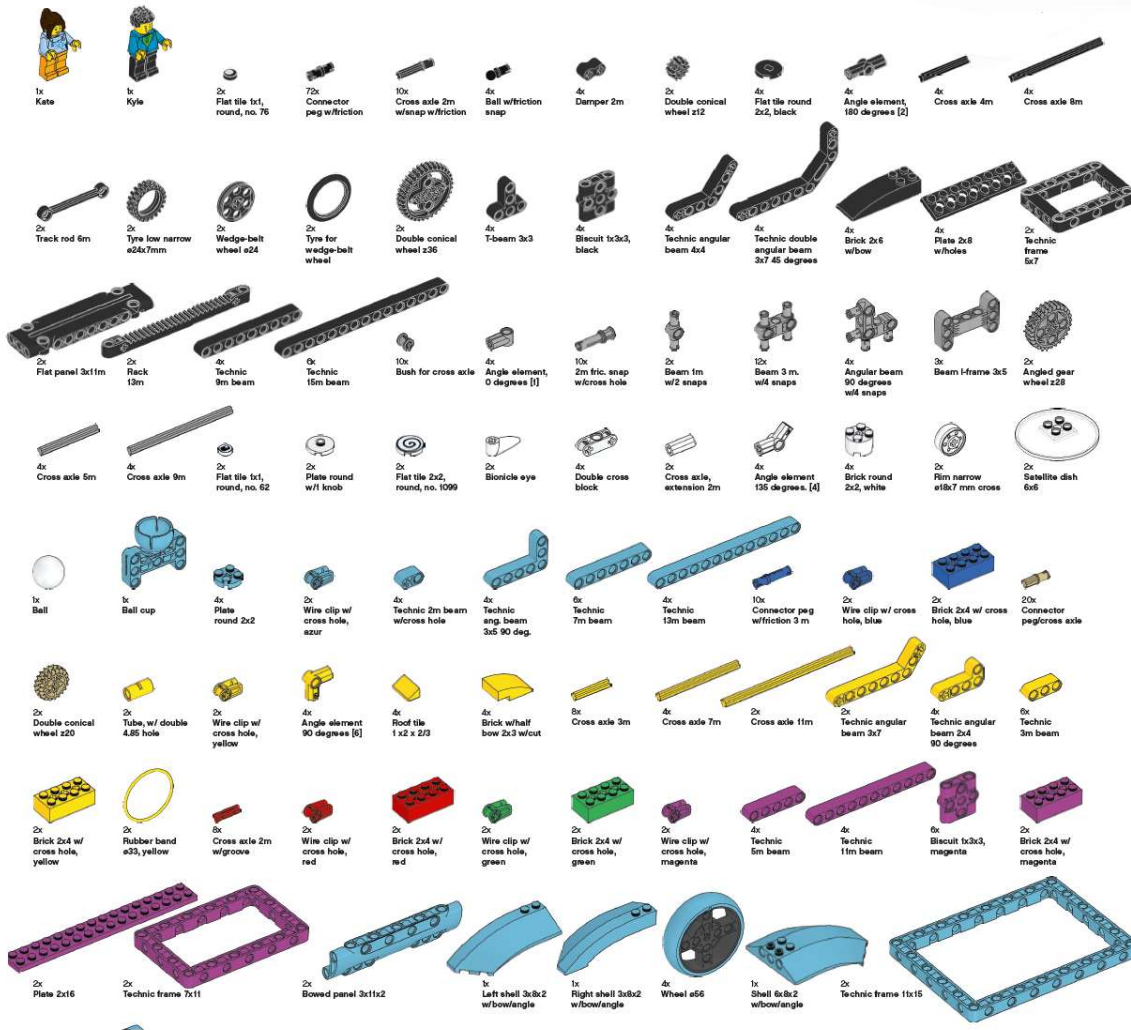
<https://doi.org/10.1111/j.1949-8594.2012.00101.x>

## Πηγές Εικόνων

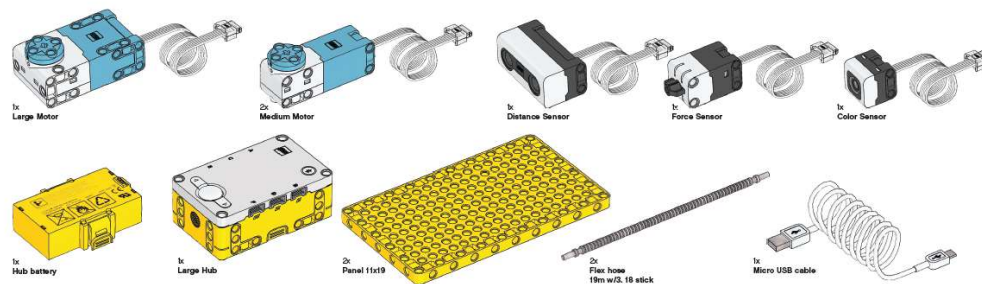
- [1] <https://www.knowatom.com/blog/making-sense-of-the-3-dimensions-of-ngss>
- [2] <https://www.kentromelissas.gr/product/bee-bot-programmable-floor-robot/>
- [3] <https://www.lavieenrose.gr/product/kids-first-coding-roboting-407072-4-stem-toys/>
- [4] [http://www.robotcombat.com/store\\_vex\\_starter.html](http://www.robotcombat.com/store_vex_starter.html)
- [5] <https://www.aldebaran.com/en/nao>
- [6] <https://www.why.gr/%CE%BA%CE%B1%CF%84%CE%B1%CF%83%CF%84%CE%B7%CE%BC%CE%B1/open-hardware/elec Freaks/elec Freaks-kits/nezha-inventors-kit-for-arduino/>
- [7] <https://www.robocamp.eu/en/blog/lego-education-wedo2-review/>
- [8] [https://live.staticflickr.com/65535/50044865547\\_90e1776edf\\_b.jpg](https://live.staticflickr.com/65535/50044865547_90e1776edf_b.jpg)
- [9] [https://live.staticflickr.com/65535/50020100516\\_c9e0b12f05\\_b.jpg](https://live.staticflickr.com/65535/50020100516_c9e0b12f05_b.jpg)
- [10] <https://docs.pybricks.com/en/latest/hubs/primehub.html>
- [11] <https://antonsmindstorms.com/2021/01/14/advanced-undocumented-python-in-spike-prime-and-mindstorms-hubs/>
- [12] <https://community.legoeducation.com/blogs/31/220>
- [13] [JimBricks](#)
- [14] <https://www.gkindex.com/python-tutorial/python-if-elif-else.jsp>
- [15] <https://www.teachengineering.org/populartopics/designprocess>
- [16] <https://education.lego.com/en-us/products/lego-education-spike-prime-expansion-set/45681>
- [17] <https://www.o3.gr/product/lego-education-spike-prime-expansion-set-2021/>
- [18] [LE SPIKE classroom posters 18x24inch V1.indd \(legocdn.com\)](#)
- [19] [LE SPIKE classroom posters 18x24inch V1.indd \(legocdn.com\)](#)

# Παράρτημα

## Κατάλογος με τα Δομικά στοιχεία του Spike Prime.



## Κατάλογος με τα Ηλεκτρονικά Εξαρτήματα του Spike Prime



## Κατάλογος με όλα τα στοιχεία του Set Επέκτασης

