



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

# Σχεδίαση και υλοποίηση κύβων *tangible computing* για εκμάθηση βασικών εννοιών προγραμματισμού σε παιδιά

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

Λάτσης Νικόλαος

Επιβλέπων Καθηγητής

Δρ. Μηνάς Δασυγένης

Επίκουρος Καθηγητής

Εργαστήριο Ρομποτικής, Ενσωματωμένων και Ολοκληρωμένων Συστημάτων

ΚΟΖΑΝΗ ΦΕΒΡΟΥΑΡΙΟΣ 2023





ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

# **Design and implementation of tangible computing cubes for learning basic programming concepts in children**

**DIPLOMA THESIS**

---

**Latsis Nikolaos**

**Supervising Professor**

Dr. Minas Dasygenis

Assistant Professor

**Robotics, Embedded and Integrated Systems Laboratory**

**KOZANI FEBRUARY 2023**





ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## **ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ**

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο “Σχεδίαση και Υλοποίηση Κύβων Tangible Computing για εκμάθηση βασικών εννοιών προγραμματισμού σε παιδιά” καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος Δρ. Μηνά Δασυγένη αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Νικόλαος Λάτσης & Δρ. Μηνάς Δασυγένης, 2022, Κοζάνη

Υπογραφή Φοιτητή: Νικόλαος Λάτσης



## Περίληψη

Τις τελευταίες δεκαετίες παρατηρούμε ότι έχει αρχίσει και εισάγεται η έννοια της πληροφορικής σε παιδιά μικρότερων ηλικιών στην Ελλάδα γεγονός πολύ θετικό για την εκπαιδευτική διαδικασία και την προσπάθεια εκσυγχρονισμού της χώρας, πόσο μάλλον σε θέματα που αφορούν τη νέα γενιά και την εκπαίδευσή της σχετικά με την τεχνολογία. Με σκοπό λοιπόν την ανάπτυξη αυτού του τομέα έχουν δημιουργηθεί διάφορες μέθοδοι οι οποίες να μεν διδάσκουν αλλά και ελκύουν το ενδιαφέρον των παιδιών σχετικά με την πληροφορική.

Στην παρούσα λοιπόν διπλωματική θα παρουσιαστεί και θα αναλυθεί μία τέτοια μέθοδος εκπαίδευσης, η οποία θα περιλαμβάνει διαδραστικούς κύβους μέσω των οποίων παιδιά ηλικίας μικρών τάξεων του δημοτικού κυρίως (αλλά και όχι μόνο), θα έχουν την ευκαιρία να κατανοήσουν εισαγωγικές έννοιες της πληροφορικής, αλλά και να έρθουν σε επαφή με το συγκεκριμένο αντικείμενο. Πιο ειδικά τα παιδιά θα μπορούν να συνδυάσουν με διαφορετικούς τρόπους τους κύβους τοποθετώντας τους πάνω σε μία επιφάνεια, ώστε να έχουν ένα από αποτέλεσμα, κάνοντας έτσι τα πρώτα τους βήματα στον προγραμματισμό. Το από αυτό αποτέλεσμα του συνδυασμού και του προγραμματισμού των κύβων θα εκφράζεται σε ένα άλλο μέσο, συγκεκριμένα σε μία οθόνη και θα επιτελείται μία συγκεκριμένη λειτουργία που θα γίνεται αντιληπτή από τα παιδιά μέσω των αισθήσεών τους κάνοντας έτσι πιο εύκολη την κατανόησή τους σχετικά με έννοιες όπως εντολή, προγραμματισμός, πρόγραμμα και κατ' επέκταση πληροφορική. Συνεπώς η μέθοδος μάθησης αυτή θα συμβάλλει στην εκπαιδευτική διαδικασία, αλλά και την ψυχαγωγία των ίδιων των παιδιών αφού θα εξοικειώνονται με την τεχνολογία και θα μαθαίνουν παράλληλα με έναν διαδραστικό και ευχάριστο για τα ίδια τρόπο.

## **Abstract**

In recent years we see that the concept of computer science has begun and is being introduced to younger children in Greece, which is very positive for the educational process and the effort to modernize the country, especially in matters concerning the new generation and its technology education. In order to develop this sector, various methods have been created which teach and attract children's interest in information technology.

In this thesis, such a method of education will be presented and analyzed, which will include interactive cubes through which children of small primary school years mainly (but not only), will have the opportunity to understand introductory concepts of informatics but also to come in contact with the specific object. More specifically, children will be able to combine the cubes in different ways by placing them on a surface and have a tangible result, thus taking their first steps in programming. This tangible result of combination and programming of the cubes will be expressed in another medium, namely on a screen and will perform a specific function that will be perceived by children through their senses, thus making it easier to understand concepts such as command, programming, programming and consequently computing. Therefore, this learning method will contribute to the educational process and the children's entertainment as they will become familiar with technology and learn at the same time in an interactive and enjoyable way.



## Ευχαριστίες

Θα ήθελα πρωτίστως να ευχαριστήσω την οικογένειά μου για την στήριξη και όλα όσα μου προσέφερε κατά τη διάρκεια των σπουδών μου. Ακόμη, θα ήθελα να ευχαριστήσω τον Δρ. Μηνά Δασυγένη για την καθοδήγηση και τη συνεργασία σε κάθε στάδιο της εκπόνησης της διπλωματικής εργασίας μου. Επίσης, θα ήθελα να επεκτείνω τις ευχαριστίες μου στους φίλους αλλά και συναδέλφους που με βοήθησαν στην ανάπτυξή μου ως άνθρωπο αλλά και ως μηχανικό μέσα από τις εμπειρίες που έζησα μαζί τους σε όλα μου τα φοιτητικά χρόνια.



## Περιεχόμενα

<b>Περίληψη</b> .....	3
<b>Abstract</b> .....	4
<b>Ευχαριστίες</b> .....	5
<b>Κατάλογος Εικόνων</b> .....	11
<b>Κεφάλαιο 1 – Εισαγωγή</b> .....	16
1.1 Σημερινή κατάσταση.....	16
1.2 Ιστορική Αναδρομή.....	17
1.3 Σκοπός του Έργου.....	19
1.4 Ηλεκτρονική Μάθηση.....	20
1.4.1 Βασικά χαρακτηριστικά.....	20
1.5 Απτός προγραμματισμός .....	23
1.5.1 Συστήματα απτού προγραμματισμού .....	23
1.5.2 Περιορισμοί των εργαλείων απτού προγραμματισμού.....	26
1.5.3 Προσέλκυση ευρύτερου κοινού .....	26
1.6 Εκπαιδευτικά Συστήματα Βασισμένα σε Κύβους .....	28
1.6.1 Block Jam.....	28
1.6.2 Algo Block .....	29
1.6.3 KIBO.....	30
1.6.4 Electronic Blocks - roBlocks.....	31
1.6.5 Music Blocks.....	31
1.6.6 Robo Blocks .....	32
1.6.7 Cubetto .....	33
1.7 Συμπεράσματα .....	33
1.8 Σύνοψη Διπλωματικής Εργασία .....	34
<b>Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο</b> .....	35

2.1 Ηλεκτρονικό Παιχνίδι-Σοβαρό Παιχνίδι.....	35
2.1.1 Car Racing Games .....	36
2.2 Μηχανική Όραση.....	38
2.2.1 Ψηφιακή Επεξεργασία Εικόνας.....	39
2.2.1.1 Μετατροπή Χρωματικού Μοντέλου.....	40
2.2.1.2 Κατωφλίωση (Thresholding).....	41
2.2.1.3 Γκαουσιανό Θόλωμα (Gaussian Blur).....	42
2.2.1.4 Ανιχνευτής Άκρων Canny.....	44
2.2.1.5 Διαστολή και Διάβρωση .....	45
2.2.2 Ανάλυση Εικόνας .....	46
2.2.2.1 Εύρεση Περιγραμμάτων (Contouring) .....	47
2.3 Πλατφόρμα Ανάπτυξης και Εργαλεία.....	48
2.3.1 Visual Studio Code.....	48
2.3.2 Python .....	49
2.3.3 Pygame.....	50
2.3.4 OpenCV .....	51
2.4 Τρισδιάστατος εκτυπωτής Ender-3.....	52
2.5 Πρόγραμμα Sketchup.....	53
2.6 Πρόγραμμα Cura.....	53
2.7 Σύνοψη Κεφαλαίου .....	53
<b>Κεφάλαιο 3 – Ανάλυση και Σχεδιασμός του Συστήματος.....</b>	<b>54</b>
3.1 Γενική Επισκόπηση και Ανάλυση .....	54
3.2 Απαιτήσεις Συστήματος .....	55
3.3 Είδη χρηστών και περιπτώσεις χρήσης.....	56
3.4 Διαγράμματα Επεξήγησης.....	57
3.5 Υλοποίηση Ηλεκτρονικού Παιχνιδιού .....	59
3.5.1 Ρύθμιση περιβάλλοντος .....	59

3.5.2 Κίνηση Αυτοκινήτου .....	63
3.5.2.1 Περιστροφή Αυτοκινήτου .....	64
3.5.2.2 Κίνηση υπό γωνία.....	65
3.5.3 Σύγκρουση με τα Σύνορα της Πίστας.....	66
3.5.4 Αλληλεπίδραση με τη γραμμή τερματισμού.....	68
3.6 Εντοπισμός και Καθορισμός σχημάτων.....	68
3.6.1 Ρύθμιση Κάμερας .....	68
3.6.2 Φιλτράρισμα της εικόνας.....	70
3.6.3 Εφαρμογή Κατωφλίωσης.....	71
3.6.4 Εφαρμογή εύρεσης περιγραμμάτων (Contouring Application).....	71
3.6.5 Εύρεση γωνιακών σημείων και Εγκατάσταση περιοριστικού τετραγώνου.....	73
3.6.6 Προσδιορισμός-Ταυτοποίηση σχήματος .....	76
3.6.7 Διεπαφή Εντοπισμού και Καθορισμού σχήματος .....	76
3.7 Κατασκευή των Κύβων .....	79
3.7.1 Σχεδιασμός Κύβου.....	79
3.7.2 Εκτύπωση κύβου και ολοκλήρωση κατασκευής.....	83
3.8 Διασύνδεση επί μέρους λειτουργιών .....	85
3.9 Σύνοψη κεφαλαίου.....	89
<b>Κεφάλαιο 4 – Πειραματική Διαδικασία .....</b>	<b>90</b>
4.1 Προσδιορισμός περιβάλλοντος.....	90
4.2 Πειραματισμός κατασκευής κύβων .....	95
4.2.1 Κατασκευή από μακετόχαρτο .....	95
4.2.2 Κατασκευή με τρισδιάστατο εκτυπωτή.....	97
4.3 Πειραματική επαλήθευση συστήματος .....	98
4.4 Σύνοψη κεφαλαίου.....	102
<b>Κεφάλαιο 5 – Επίλογος.....</b>	<b>103</b>
5.1 Ανακεφαλαίωση.....	103

5.2 Μετρικά συστήματος .....	103
5.3 Ανάλυση SWOT .....	104
5.4 Μελλοντικές επεκτάσεις.....	105
5.5 Συμπεράσματα.....	106
<b>Παράρτημα: Οδηγίες Εγκατάστασης .....</b>	<b>108</b>
<b>Βιβλιογραφία .....</b>	<b>109</b>

## Κατάλογος Εικόνων

Εικόνα 1.1: Digital Beads .....	18
Εικόνα 1.2: Χαρακτηριστικά ηλεκτρονικής μάθησης .....	23
Εικόνα 1.3: Ο κουλοχέρης Tortis .....	24
Εικόνα 1.4: Προγραμματισμός με χορδές.....	25
Εικόνα 1.5: Quetzal και Tern αριστερά και δεξιά αντίστοιχα.....	26
Εικόνα 1.6: Το LilyPad Arduino .....	27
Εικόνα 1.7: Εφαρμογή LilyPad Arduino σε παράσταση μόδας.....	28
Εικόνα 1.8: Block Jam .....	29
Εικόνα 1.9: AlgoBlock.....	30
Εικόνα 1.10: Το ρομπότ KIBO και τα μπλοκ του .....	30
Εικόνα 1.11: roBlocks (αργότερα Cubelets) .....	31
Εικόνα 1.12: Music Blocks .....	32
Εικόνα 1.13: Robo Blocks.....	32
Εικόνα 1.14: Cubetto .....	33
Εικόνα 2.1: Το βιντεοπαιχνίδι Super Mario Bros στις πρώτες εκδόσεις του.....	36
Εικόνα 2.2: Σοβαρό παιδικό παιχνίδι προσομοίωσης νοσοκομειακής περίθαλψης.....	36
Εικόνα 2.3: Racing game πρώτης γενιάς .....	37
Εικόνα 2.4: Racing game τελευταίας γενιάς .....	37
Εικόνα 2.5: Αναγνώριση και Ταυτοποίηση προσώπου αντίστοιχα μέσω χρήσης της Μηχανικής Όρασης.....	38
Εικόνα 2.6: Εφαρμογή Μηχανικής Όρασης σε ακτινογραφίες θώρακα.....	39
Εικόνα 2.7: Ζωντανό στιγμιότυπο εικόνας κάμερας όπισθεν αυτοκινήτου όπου εφαρμόζεται Μηχανική Όραση.....	39
Εικόνα 2.8: Ο κύκλος του μοντέλου RGB.....	41
Εικόνα 2.9: Παράδειγμα Greyscaling .....	41
Εικόνα 2.10: Εφαρμογή Κατωφλίωσης σε εικόνα ενός τοπίου .....	42
Εικόνα 2.11: Παράδειγμα ενός πυρήνα Gauss όπου οι τιμές των pixel αντιστοιχούν στις τιμές μιας καμπύλης Gauss. ....	43
Εικόνα 2.12: Παράδειγμα εφαρμογής Gaussian Blur.....	43
Εικόνα 2.13: Εφαρμογή φίλτρου Gaussian Blur .....	44

Εικόνα 2.14: Πριν και μετά την εφαρμογή ανίχνευσης γραμμών Canny σε φωτογραφία της ηθοποιού Emma Stone .....	45
Εικόνα 2.15: Χειροκίνητη εφαρμογή Διαστολής και Διάβρωσης σε εικόνα .....	46
Εικόνα 2.16: Ανάλυση εικόνας με χαμηλό φωτισμό και βελτίωσή της μέσω της αφαίρεσης της θολότητας .....	47
Εικόνα 2.17: Εύρεση περιγράμματος μετά την εφαρμογή κατωφλίωσης και εύρεσης γραμμών Canny .....	48
Εικόνα 2.18: Το σήμα και παράδειγμα γραφικού περιβάλλοντος σε VS Code .....	49
Εικόνα 2.19: Το σύμβολο της Python.....	50
Εικόνα 2.20: Το σύμβολο της Pygame .....	50
Εικόνα 2.21: Παιχνίδι σχεδιασμένο σε Pygame βασισμένο στο ευρέως διαδεδομένο arcade παιχνίδι της εποχής του '90 Wolfenstein .....	51
Εικόνα 2.22: Το σήμα της OpenCV και μία εικόνα επεξεργασμένη μέσω αυτής της βιβλιοθήκης .....	52
Εικόνα 2.23: Ο τρισδιάστατος εκτυπωτής Ender-3 .....	52
Εικόνα 3.1: Διάγραμμα Επεξήγησης λειτουργιών εκπαιδευτικού .....	58
Εικόνα 3.2: Διάγραμμα Επεξήγησης λειτουργιών μαθητή .....	58
Εικόνα 3.3: Το κόκκινο αυτοκίνητο του παιχνιδιού .....	60
Εικόνα 3.4: Το σχήμα του δρόμου και των ορίων της πίστας.....	60
Εικόνα 3.5: Η γραμμή τερματισμού .....	60
Εικόνα 3.6: Το γρασίδι της πίστας.....	61
Εικόνα 3.7: Παράδειγμα κενού παραθύρου pygame .....	62
Εικόνα 3.8: Κώδικας με την εισαγωγή και αλλαγή μεγέθους των εικόνων.....	62
Εικόνα 3.9: Τελικό αποτέλεσμα παραθύρου αναπαραγωγής.....	63
Εικόνα 3.10: Απόσπασμα του κώδικα της κλάσης AbstractCar .....	64
Εικόνα 3.11: Κώδικας της συνάρτησης blit_rotate_center .....	65
Εικόνα 3.12: Γράφημα παραδείγματος υπολογισμού κίνησης του αυτοκινήτου υπό γωνία ...	66
Εικόνα 3.13: Η Συνάρτηση της κίνησης του αυτοκινήτου σε κώδικα .....	66
Εικόνα 3.14: Δημιουργία μάσκας στην εικόνα με τα όρια της πίστας (γραμμές 14,15) και αλληλεπίδραση με αυτή (γραμμές 298,299) .....	67
Εικόνα 3.15: Πρόσβαση και λήψη βίντεο διαμέσου της συνάρτησης cv2.VideoCapture και αρχικοποίηση ρυθμίσεων κάμερας.....	69
Εικόνα 3.16: Προβολή της εικόνας με τη συνάρτηση cv2.imshow().....	69



Εικόνα 3.17: Παράδειγμα παραθύρου βίντεο με ζωντανή εικόνα μετά την εφαρμογή των παραπάνω εντολών .....	69
Εικόνα 3.18: Εφαρμογή των φίλτρων Γκαουσιανό Θόλωμα, Grayscale, Ανίχνευση Γραμμών Canny και Αναστολής της εικόνας στις γραμμές 216, 217, 223, 227 του κώδικα αντίστοιχα.....	70
Εικόνα 3.19: Οι μπάρες με αναγραφόμενες τις τιμές κατωφλίωσης η εισαγωγή των οποίων έγινε στις γραμμές 220 και 221 του κώδικα στην Εικόνα 3.18.....	71
Εικόνα 3.20: Απόσπασμα του κώδικα της συνάρτησης getContours.....	72
Εικόνα 3.21: Αναπαραγωγή του περιγράμματος ενός τετραγώνου σχήματος με μοβ χρώμα στη ζωντανή μετάδοση της κάμερας .....	73
Εικόνα 3.22: Απεικόνιση του περιοριστικού τετραγώνου με πράσινο και του περιγράμματος με μοβ αντίστοιχα .....	74
Εικόνα 3.23: Εντοπισμός ξεχωριστών σχημάτων εντός ξεχωριστών περιοριστικών τετραγώνων .....	75
Εικόνα 3.24: Το τελικό παράθυρο με τις μπάρες τιμών κατωφλίωσης 1, 2 και της περιοχής που καλύπτει το .....	75
Εικόνα 3.25: Αναπαράσταση της εικόνας με την ταυτοποίηση των σχημάτων (αριστερά τετράγωνο-δεξιά κύκλος).....	76
Εικόνα 3.26: Η αναπαραγόμενη αφιλτράριστη εικόνα.....	77
Εικόνα 3.27: Η εικόνα υπό την εφαρμογή του φίλτρου για την ανίχνευση γραμμών Canny..	77
Εικόνα 3.28: Το παράθυρο της διεπαφής του χρήστη για την διαδικασία του εντοπισμού και της αναγνώρισης των σχημάτων.....	78
Εικόνα 3.29: Το τελικό εικονικό σχέδιο του κύβου στο περιβάλλον του Sketchup .....	80
Εικόνα 3.30: Το νέο σχέδιο προς εκτύπωση στο περιβάλλον του Cura .....	80
Εικόνα 3.31: Το εσωτερικό των κύβων προς εκτύπωση.....	81
Εικόνα 3.32: Η ρύθμιση της παραμέτρου του infill στο Cura .....	82
Εικόνα 3.33: Η ρύθμιση των θερμοκρασιών μύτης και επιφάνειας εκτύπωσης στο Cura .....	82
Εικόνα 3.34: Στάδια εκτύπωσης των κύβων στον τρισδιάστατο εκτυπωτή .....	83
Εικόνα 3.35: Το τελικό προϊόν της εκτύπωσης.....	84
Εικόνα 3.36: Το τελικό προϊόν σε σχέση με ένα χέρι ενηλίκου.....	84
Εικόνα 3.37: Ολοκληρωμένη κατασκευή κύβου με τοποθέτηση του σχήματος του τετραγώνου.....	85
Εικόνα 3.38: Απόσπασμα κώδικα λούπας while.....	86
Εικόνα 3.39: Απόσπασμα κώδικα ορισμού σχήματος - κίνησης .....	86

Εικόνα 3.40: Απόσπασμα κώδικα για την ταξινόμηση των σχημάτων .....	87
Εικόνα 3.41: Μέρος κώδικα με τη διαχείριση λιστών για τη λειτουργία του αυτοματισμού..	88
Εικόνα 3.42: Διάγραμμα αυτοματισμού τοποθέτησης σχημάτων.....	89
Εικόνα 4.1: Αναγνώριση ενός, δύο και τριών σχημάτων αντίστοιχα με φυσικό φως πρωινών-μεσημεριανών ωρών .....	91
Εικόνα 4.2: Το σετ ρυθμίσεων κατωφλίωσης και περιοχής για την αναγνώριση σχημάτων στην εικόνα 4.1.....	92
Εικόνα 4.3: Στιγμιότυπο λανθασμένης αναγνώρισης σχημάτων τις απογευματινές ώρες με λιγότερο φωτισμό .....	92
Εικόνα 4.4: Επισημάσωση σημείων λανθασμένης αναγνώρισης μετά την αλλαγή των ρυθμίσεων.....	93
Εικόνα 4.5: Το σετ ρυθμίσεων κατωφλίωσης και περιοχής για την αναγνώριση των σχημάτων στην εικόνα 4.4.....	93
Εικόνα 4.6: Επιτυχής διαδικασία αναγνώρισης σχημάτων με τεχνητό φως.....	94
Εικόνα 4.7: Απόπειρες κατασκευής κύβων από μακετόχαρτο.....	96
Εικόνα 4.8: Παράδειγμα λανθασμένης αναγνώρισης σχημάτων λόγω παρεμβολών κατά την κατωφλίωση που φανερώνονται στα επισημασμένα σημεία. ....	96
Εικόνα 4.9: Προσέγγιση κατασκευής κύβου με ενσωματωμένο σχήμα.....	97
Εικόνα 4.10: Προσέγγιση κατασκευής κύβου από το ανάπτυγμα του .....	98
Εικόνα 4.11: Αποτέλεσμα εισαγωγής μεμονωμένου σχήματος-εντολής βέλους που αντιστοιχεί στην ευθεία κίνηση του αυτοκινήτου .....	99
Εικόνα 4.12: Αποτέλεσμα εισαγωγής συνδυασμού εντολών δεξιά και ευθεία μέσω των σχημάτων κύκλου και βέλους αντίστοιχα.....	99
Εικόνα 4.13: Αποτέλεσμα εισαγωγής τριών εντολών δεξιά-ευθεία-ευθεία μέσω των σχημάτων κύκλου-βέλους-βέλους αντίστοιχα .....	100
Εικόνα 4.14: Πρόβλημα αναγνώρισης κατά την εισαγωγή τεσσάρων σχημάτων .....	101
Εικόνα 4.15: Περίπτωση εισαγωγής πέντε σχημάτων τα ακριανά των οποίων ανακόπτονται .....	101
Εικόνα 4.16: Αποτυχημένη αναγνώριση σχημάτων μετά την τοποθέτηση της κάμερας σε μεγαλύτερο ύψος .....	102



## Κεφάλαιο 1 – Εισαγωγή

Σε αυτό το κεφάλαιο θα περιγραφεί το αντικείμενο της συγκεκριμένης διπλωματικής εργασίας. Αρχικά, θα γίνει αναφορά στην υπάρχουσα κατάσταση που επικρατεί και στον συνεχόμενα αναπτυσσόμενο και εξελισσόμενο χώρο της τεχνολογίας καθώς και το πως αυτό το γεγονός ανακλάται και στην εκπαίδευση. Στην συνέχεια, παρουσιάζονται οι προκλήσεις και ο σκοπός που οδήγησαν στην υλοποίηση της παρούσης διπλωματικής εργασίας. Μετέπειτα γίνεται αναφορά σε έννοιες αλλά και ορολογίες που αφορούν το έργο, αναλύονται αντίστοιχα συστήματα που έχουν πραγματωθεί από άλλους ερευνητές και λαμβάνεται ένα ολοκληρωμένο συμπέρασμα. Τέλος, πραγματοποιείται μια συνοπτική παρουσίαση της δομής της εργασίας.

### 1.1 Σημερινή κατάσταση

Τις τελευταίες δεκαετίες η τεχνολογική εξέλιξη στον χώρο των πληροφοριακών συστημάτων έχει γίνει αναπόσπαστο κομμάτι της εκπαιδευτικής διαδικασίας σε όλες τις βαθμίδες της εκπαίδευσης. Προσπαθώντας κάθε εκπαιδευτική οντότητα να κάνει το μάθημα ειδικά της πληροφορικής και τη διαδικασία της μάθησης πιο ελκυστική, αποτελεσματική και άμεση έχει υιοθετήσει κάποια τεχνολογικά εργαλεία. Εξάλλου οι δεξιότητες που οι νέοι και οι μικροί κυρίως σε ηλικία μαθητές αποκτούν ανέκαθεν, ανεξάρτητα από την εκπαίδευση, λόγω της ενασχόλησής τους με ανάλογα τεχνολογικά «gadgets» καθιστούν και τη συμμετοχή των τελευταίων στην εκπαιδευτική διαδικασία πλέον κατάλληλη καθώς υπάρχουν διαρκώς στην καθημερινότητα των παιδιών.

Φανερό επομένως γίνεται, η αναγκαιότητα της ανάπτυξης της έννοιας της τεχνολογίας και πιο ειδικά της πληροφορικής και του προγραμματισμού στα παιδιά και στους μαθητές από μικρές πλέον ηλικίες. Προκειμένου με αυτόν τον τρόπο, τα άτομα της νέας γενιάς να αρχίσουν να αντιλαμβάνονται τον τρόπο λειτουργίας σε ένα βαθμό των συσκευών που χρησιμοποιούν και κατ' επέκταση του κόσμου γύρω τους, καθώς καλώς ή κακώς πλέον ο κόσμος μας αποτελείται και είναι δομημένος με βάση την τεχνολογία. Συστήνεται λοιπόν από ειδικούς η ανάπτυξη παιχνιδιών και εργαλείων που θα βοηθούν τον εκπαιδευτικό να ενημερώσει και να εκπαιδεύσει τα παιδιά σε αυτό τον τομέα της κατανόησης και εμπέδωσης

καθώς η ικανότητα αυτή αποτελεί σημαντικό κομμάτι στην ολόπλευρη ανάπτυξη των ίδιων των παιδιών.

Ο όρος «tangible computing» σημαίνει απτός προγραμματισμός. Χρησιμοποιείται ώστε να περιγράψουμε έναν τρόπο με το οποίο, όπως λέει και η ίδια η λέξη, μπορεί κάποιος να αγγίζει αυτό που προγραμματίζει και να είναι χειροπιαστό. Επομένως, προκειμένου άτομα μικρών ηλικιών να αντιληφθούν προαναφερθείσες έννοιες και να τις κατανοήσουν, η μέθοδος του απτού προγραμματισμού αποτελεί έναν πολύ καλό τρόπο για να το κάνουν μέσω και των αντίστοιχων παιχνιδιών-συσκευών που θα χρησιμοποιηθούν τα οποία υιοθετούν και τη συγκεκριμένη μέθοδο.

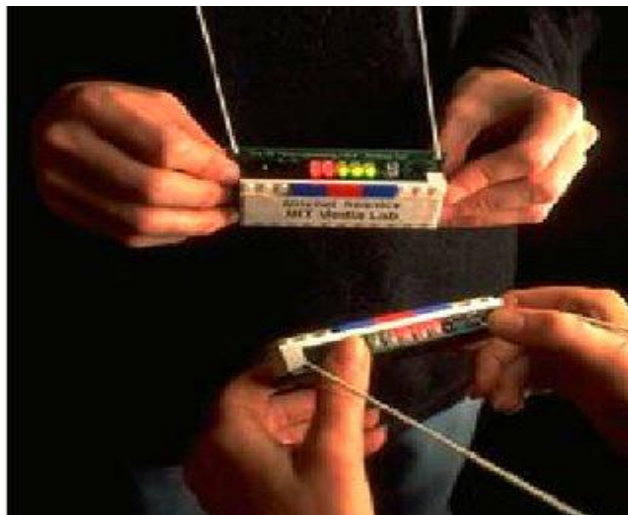
## 1.2 Ιστορική Αναδρομή

Όπως έχει προαναφερθεί μέχρι και τα τέλη του προηγούμενου αιώνα η εκπαίδευση της πληροφορικής σε ότι αφορά τη φυσική της υπόσταση γνώρισε μια αναζωπύρωση τόσο μέσω της τεχνολογικής προόδου όσο και της συνεχούς προσπάθειας των ερευνητών να προωθήσουν την ανθρώπινη αλληλεπίδραση με υπολογιστές πέρα από την οθόνη του υπολογιστή και μέσα στον πραγματικό κόσμο. Οι Ishii και Ulmer [1] επινόησαν τον όρο «απτό» για να περιγράψουν μια κατηγορία διεπαφών (interfaces) υπολογιστή που χρησιμοποιούν φυσικά αντικείμενα και επιφάνειες ως μέσο για να χειριστούν και να αντιπροσωπεύσουν ψηφιακές πληροφορίες. Η χρήση του όρου προοριζόταν να περιγράψει την ιδέα ότι μεγάλο μέρος του πλούτου της ανθρώπινης αλληλεπίδρασης με τον φυσικό κόσμο μέσω της χρήσης εργαλείων έχει αντικατασταθεί από την αντίστοιχη αλληλεπίδραση με συσκευές όπως ποντίκια, πληκτρολόγια και οθόνες αφής. Η ενσωμάτωση μιας ποικιλίας φυσικών αντικειμένων για να επιτευχθεί μια πολύ-αισθητηριακή ανατροφοδότηση (feedback) θεωρήθηκε ένας τρόπος για να ανακτηθεί αυτός ο πλούτος της ανθρώπινης επαφής και κατ' επέκταση να εξανθρωπιστεί η αλληλεπίδραση μεταξύ ανθρώπου υπολογιστή.

Αργότερα έννοιες και ορισμοί όπως η ενσωματωμένη αλληλεπίδραση του Dourish (embodied interaction) [2] και απτή αλληλεπίδραση (tangible interaction) των Hornecker and Buur [3] τόνισε τον βαθμό στον οποίο τα διαδραστικά συστήματα θα μπορούσαν να ενσωματωθούν ουσιαστικά στα φυσικά, κοινωνικά και πολιτιστικά περιβάλλοντα. Υπό αυτήν την έννοια, η απτότητα έγινε λιγότερο σχετική με τη φύση της διεπαφής και περισσότερο με την ιδέα ότι η αλληλεπίδραση με ψηφιακά συστήματα μπορεί να εμπλέκεται με υλικές και

κοινωνικές πραγματικότητες πέρα από αυτήν ενός ατόμου που κάθεται μπροστά από μία οθόνη υπολογιστή.

Αναμενόμενα λοιπόν μεγάλο μέρος της έρευνας που περιλαμβάνει απτή αλληλεπίδραση έχει δώσει έμφαση στην εκπαίδευση και τη μάθηση [4-6]. Για παράδειγμα, είναι αξιοσημείωτο το έργο του Resnick και των συνεργατών στα Media Lab του MIT που εστίαζε σε ψηφιακούς χειρισμούς, υπολογιστικά βελτιωμένες εκδόσεις παραδοσιακών παιδικών παιχνιδιών που δημιούργησαν νέες ευκαιρίες για τους μαθητές να αλληλεπιδράσουν με πολύπλοκες ιδέες. Το Digital Beads [7] (Εικόνα 1.1) ακόμη επέτρεψε στα παιδιά να δημιουργήσουν απλά προγράμματα σε μια γλώσσα με μονοδιάστατα κυψελοειδή αυτόματα συνδυάζοντας μικρές κάψουλες με ενσωματωμένα LED που θα μπορούσαν να μεταδώσουν, να απορροφήσουν ή να καταστρέψουν φως που περνά από γειτονικές μεταξύ τους χάντρες.



*Εικόνα 1.1: Digital Beads*

Το System Block Project [8] παρείχε μια παρόμοια διεπαφή για προσομοιωμένα δυναμικά συστήματα. Ξύλινα μπλοκ με ενσωματωμένα ηλεκτρονικά κομμάτια μπορούσαν να έχουν συμπεριφορές και να επιτελούν λειτουργίες αντίστοιχων πολύπλοκων συστημάτων όπως stocks, flows και feedback loops. Το έργο αυτής της ομάδας βοήθησε επίσης στο άνοιγμα της φυσικής υπολογιστικής και ρομποτικής σε ένα ευρύτερο και νεότερο κοινό.

Αντίστοιχα με προηγούμενα παραδείγματα το έργο LEGO / Logo κατάφερε τα παιδιά μπορούν να γράφουν προγράμματα υπολογιστή για τον έλεγχο κινούμενων κατασκευών LEGO ενσωματώνοντας αισθητήρες και κινητήρες [9]. Αυτό το έργο ακολούθησαν άλλα προγράμματα με επιρροή κυρίως στον εκπαιδευτικό τομέα και όχι μόνο, όπως το LEGO Mindstorms και το MIT Cricket [7,10]. Συστήματα όπως αυτά οδήγησαν στην εφεύρεση και άλλων πιο σύγχρονων όπως του Arduino [11].

Εν κατακλείδι επομένως βλέπουμε πως η έννοια του απτού προγραμματισμού εισήχθη αρκετά θα λέγαμε νωρίτερα, και λόγω κάποιων διορατικών ανθρώπων του χώρου κατάφερε να εφαρμοστεί και να ενσωματωθεί σε εργαλεία και εφαρμογές που με κάποιες εξελίξεις και βελτιώσεις χρησιμοποιούνται ακόμα έως σήμερα. Επίσης διαπιστώθηκε και η χρησιμότητα του σε όλους τους τομείς της εκπαίδευσης και όχι μόνο συνεισφέροντας έτσι κατ' επέκταση και στη διαμόρφωση ενδεχομένως μιας πιο τεχνολογικά εξελιγμένης κοινωνίας έως ένα βαθμό.

### 1.3 Σκοπός του Έργου

Ένας από τους τρόπους με τους οποίους μπορεί να εισαχθούν τα σύγχρονα ηλεκτρονικά μέσα και η πληροφορική στην εκπαίδευση, είναι τα εκπαιδευτικά λογισμικά αλλά και ειδικότερα οι γλώσσες προγραμματισμού. Οι γλώσσες προγραμματισμού φαίνεται να οδηγούν στην ανάπτυξη πνευματικών ικανοτήτων, που αφορούν κυρίως τον λογικό και μαθηματικό τομέα με αποτέλεσμα οι χρήστες να αποκομίζουν πολλαπλά οφέλη αναπτύσσοντας δομημένο και ιεραρχικό τρόπο σκέψης, πράγμα το οποίο μπορεί να βοηθήσει στην επίλυση προβλημάτων τους και εκτός μαθησιακού περιβάλλοντος αλλά και εκτός υπολογιστή .

Ο προγραμματισμός ωστόσο φαίνεται να είναι γενικά μια απαιτητική και δύσκολη ως επί το πλείστον διαδικασία για τους αρχάριους σε όλες τις ηλικίες. Οι χρήστες και πόσο μάλλον οι μαθητές συνήθως έχουν δυσκολία στο να κατανοήσουν μία περίεργη σύνταξη με εντολές με πολλά ονόματα αλλά και στο να μάθουν πώς να χειρίζονται το προγραμματιστικό περιβάλλον και να εκτελούν λειτουργίες για τον υπολογιστή μέσα από αυτό. Ο απτικός προγραμματισμός από την άλλη είναι μια μορφή γλώσσας η οποία για να εκφραστεί δεν απαιτεί απαραίτητα τη χρήση πληκτρολογίου, ποντικιού ή υπολογιστή, αλλά μόνο την αξιοποίηση και η διάταξη φυσικών αντικειμένων στον χώρο. Επομένως, μπορεί να συμπεράνει κανείς ότι τα απτικά συστήματα μειώνουν το γνωστικό φορτίο που απαιτείται

προκειμένου να μάθει κάποιος πως δουλεύει ένα σύστημα και έτσι η προσοχή του χρήστη εστιάζεται στο πεδίο εκμάθησης γύρω από τον προγραμματισμό, ιδιότητα που τα καθιστά ιδανικά εργαλεία εκμάθησης για άτομα και κοινωνικές ομάδες με μικρή εμπειρία στον προγραμματισμό όπως παιδιά.

Βάση των παραπάνω η διπλωματική εργασία θέτει τους ακόλουθους στόχους:

- Να παρουσιάσει την ανάλυση και υλοποίηση ενός ολοκληρωμένου συστήματος κύβων μέσω των οποίων τα παιδιά κυρίως μικρότερων ηλικιών θα έχουν την δυνατότητα να προγραμματίζουν και να έχουν ένα απτικό αποτέλεσμα των εντολών που έδωσαν μέσω των συνδυασμών των κύβων σε ένα άλλο οπτικοακουστικό μέσο.
- Να αξιοποιήσει το παραπάνω σύστημα κύβων και να παρέχει στοιχεία τα οποία θα μπορέσουν έμπρακτα να φωτίσουν τις συνθήκες και τα πιθανά οφέλη από τη χρήση απτικών διεπαφών, γενικά και ειδικά, στον εισαγωγικό προγραμματισμό σε παιδιά μικρών ηλικιών.
- Να παρουσιάσει τα οφέλη της χρήσης νέων τεχνολογιών και τέτοιων δημιουργικών τρόπων εκμάθησης στην εκπαιδευτική διαδικασία γενικότερα.

## 1.4 Ηλεκτρονική Μάθηση

### 1.4.1 Βασικά χαρακτηριστικά

Ο αιώνας που διανύουμε χαρακτηρίζεται από τη συνεχή ανάπτυξη των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών (Τ.Π.Ε.), του Διαδικτύου αλλά και των κινητών και «έξυπνων» συσκευών μετασχηματίζοντας τους τρόπους διδασκαλίας μάθησης και επικοινωνίας . Οι σύγχρονες ανάγκες για μια μορφή διδασκαλίας και μάθησης που προσφέρει μέγιστη αυτονομία και ευελιξία αναφορικά με τον τόπο, τον χρόνο και το ρυθμό μάθησης, διευκολύνοντας και διευρύνοντας ταυτόχρονα τη πρόσβαση στο κόσμο της γνώσης, έχουν οδηγήσει σε ένα νέο είδος εκπαιδευτικής καινοτομίας, την ηλεκτρονική μάθηση (e-learning)[13].

Η ηλεκτρονική εκπαίδευση και μάθηση, παρά το γεγονός ότι δεν πρόκειται για φαινόμενο των τελευταίων ετών, αλλά με ανιχνεύσιμα σημάδια ήδη από τη δεκαετία του 1980, σήμερα τυγχάνει ευρείας αποδοχής μεταξύ των διδασκόντων και των διδασκόμενων, αλλά και της



ανάλογης υποστήριξης από τα εκπαιδευτικά ιδρύματα που αναγνωρίζουν τις πλείστες δυνατότητες του συγκεκριμένου τρόπου μάθησης.

Κατά καιρούς έχουν δοθεί διαφορετικοί ορισμοί για την ηλεκτρονική μάθηση και έχουν προταθεί ισοδύναμες ή συμπληρωματικές έννοιες, χωρίς ωστόσο να έχει αποδοθεί ένας μοναδικός και περιεκτικός ορισμός, γεγονός που ενισχύει την πολυπλοκότητα και την πολυσημία της έννοιας αλλά και το πολύμορφο χαρακτήρα της.

Αναγνωρίζοντας τις δυσκολίες για την διατύπωση ενός κοινά αποδεκτού επιστημονικού ορισμού για έναν όρο που διαρκώς εξελίσσεται, πολλοί μελετητές προτιμούν να εντοπίσουν τα βασικά γνώρισμα και τα ιδιαίτερα χαρακτηριστικά της ηλεκτρονικής μάθησης, ως καινοτόμου εκπαιδευτικής διαδικασίας [12]:

- Η ηλεκτρονική μάθηση αξιοποιεί και συνδυάζει δημιουργικά τις Τεχνολογίες Πληροφορίας και Επικοινωνιών (Τ.Π.Ε.) και τα δίκτυα, ενώ ποικίλα διαδραστικά εργαλεία χρησιμοποιούνται για την παράδοση του απαραίτητου εκπαιδευτικού υλικού και την ανταλλαγή των μηνυμάτων μεταξύ των συμμετεχόντων στην διαδικασία.
- Οι εκπαιδευόμενοι οικοδομούν τη νέα γνώση ελεύθερα και αβίαστα αλληλεπιδρώντας τόσο με το εκπαιδευτικό υλικό, όσο και με τους συνδιδασκομένους τους αλλά και τον εκπαιδευτή τους, σε ένα περιβάλλον που σέβεται τους ατομικούς και αυτόνομους ρυθμούς μάθησης.
- Ο εκπαιδευτής σε ρόλο καθοδηγητικό, διευκολυντικό, υποστηρικτικό και διαμεσολαβητικό μεταξύ του εκπαιδευόμενου και της γνώσης, δεν αντιπαραβάλλεται με το συμβατικό τρόπο εκπαίδευσης, αλλά προσφέρει με τη βοήθεια της τεχνολογίας εναλλακτικές προσεγγίσεις στη διδασκαλία και στη μάθηση (συνδυασμός σύγχρονης – ασύγχρονης μορφής)
- Η ηλεκτρονική μάθηση δεν περιορίζεται μόνο στη πρόσβαση και εξερεύνηση εκπαιδευτικού υλικού μέσω του Διαδικτύου, αλλά δημιουργεί δυναμικά περιβάλλοντα μάθησης, αξιοποιώντας τις τεχνολογικές δυνατότητες συνδυασμού πολλαπλών μορφών πληροφορίας (κείμενο, ήχος, εικόνα, γραφικά, βίντεο), σε συνδυασμό με την αλληλεπιδραστική, αυτορυθμιζόμενη και συνεργατική μάθηση.
- Αν και ουσιαστικά η ηλεκτρονική μάθηση συνιστά μια στροφή από τις συμβατικές μορφές εκπαίδευσης σε πιο ευέλικτες και συνεργατικές μορφές μάθησης, στηριζόμενες σε ολοκληρωμένα Συστήματα Διαχείρισης της Μάθησης (Σ.Δ.Μ.),

ωστόσο μπορεί να χρησιμοποιηθεί συνδυαστικά με την παραδοσιακή διδασκαλία (υβριδικό μοντέλο) συνδυάζοντας συστηματικά τις εκπαιδευτικές διαδικασίες και την ανατροφοδότηση.

- Παράλληλα, χρησιμοποιώντας διαδραστικά πολυμεσικά εργαλεία, η ηλεκτρονική μάθηση επηρεάζει θετικά τους εκπαιδευόμενους προωθώντας τα μαθησιακά κίνητρα που ενεργοποιούν τη μεταφορά γνώσης μέσω διατήρησης, καθώς παρέχει καινοτόμες μαθησιακές εμπειρίες που δεν είναι εύκολα προσβάσιμες από ένα συμβατικό περιβάλλον εκπαίδευσης .
- Τα διακτυακά εργαλεία και περιβάλλοντα (ηλεκτρονικές πλατφόρμες, εργαλεία διαμοίρασης περιεχομένου, ιστολόγια, wikis, κ.λπ.) που έχει στη διάθεσή της η ηλεκτρονική μάθηση, καθιστούν δυνατή την αυτόνομη διαχείριση και την εύκολη κοινή χρήση και μετάδοση της γνώσης , δίνοντας τη δυνατότητα για τη δημιουργία δυναμικών κοινοτήτων μάθησης καθώς εκπαιδευόμενοι και εκπαιδευτές ανταλλάσσουν συλλογικά και συνεργατικά ερωτήματα, απόψεις, προβληματισμούς και εναλλακτικές λύσεις σχετικά με τη φύση ενός προβλήματος.
- Η ηλεκτρονική μάθηση δεν περιορίζεται μόνο στη μεταφορά του περιεχομένου των μαθημάτων, αλλά περιλαμβάνει και τις διαδικασίες σχεδιασμού (καθορισμός σκοπού και επιμέρους στόχων), υλοποίησης και τελικής αξιολόγησης των εκπαιδευτικών προγραμμάτων.

Οι εύκολα προσιτές και οικείες σε όλο και μεγαλύτερες ομάδες εκπαιδευομένων χρησιμοποιούμενες τεχνολογίες και η άμεση διαθεσιμότητά τους μέσω κινητών συσκευών σε διαρκή βάση, καθιστούν πλέον την ηλεκτρονική μάθηση ως ένα νέο, καινοτόμο και σύγχρονο εκπαιδευτικό μέσο. Στην παρακάτω εικόνα (Εικόνα 1.2) αναφέρονται συνοπτικά οι παραπάνω έννοιες μέσω ενός περιγραφικού διαγράμματος.



Εικόνα 1.2: Χαρακτηριστικά ηλεκτρονικής μάθησης

## 1.5 Απτός προγραμματισμός

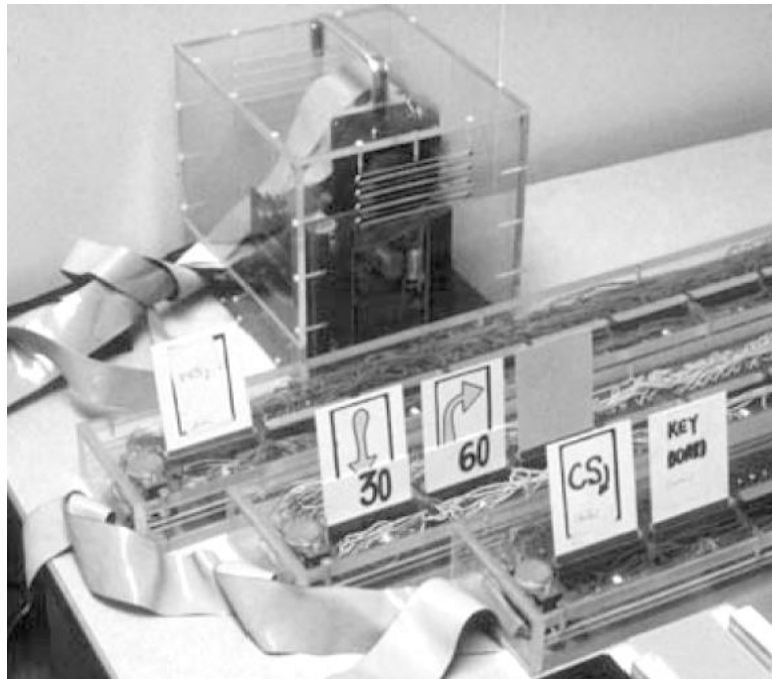
### 1.5.1 Συστήματα απτού προγραμματισμού

Η πρόσφατη έρευνα σχετικά με τις απτές διεπαφές χρήστη, όπως ορίστηκαν από τους Ishii και Uhlmer [14], άνοιξε το δρόμο για την καινοτόμο ανάπτυξη της τεχνολογίας στις τάξεις [15]. Αυτό το είδος τεχνολογίας φαίνεται να έχει ωφελήσει τα περιβάλλοντα απτικού προγραμματισμού, τα οποία έχουν εφαρμογές στην εκπαίδευση κυρίως, αλλά όχι μόνο [16]. Είτε με μια γλώσσα προγραμματισμού βασισμένη σε κείμενο, είτε με γραφικά, ένα απτό περιβάλλον προγραμματισμού μπορεί να αποφέρει τα ίδια αποτελέσματα.

Τα αντικείμενα του πραγματικού κόσμου, σε αντίθεση με τα γραφικά αντικείμενα επί της οθόνης ή τις οδηγίες επί της οθόνης, χρησιμοποιούνται για την ολοκλήρωση της διαδικασίας προγραμματισμού σε πραγματικές καταστάσεις. Ο προγραμματισμός φαίνεται να είναι ένα δύσκολο εγχείρημα για ερασιτέχνες όλων των ηλικιών [17]. Δεν είναι μόνο δύσκολο για τους χρήστες να κατανοήσουν ένα περίπλοκο συντακτικό με οδηγίες που ακούγονται αμήχανα, αλλά και το ίδιο το προγραμματιστικό περιβάλλον είναι επίσης δύσκολο να κατακτηθεί [18]. Ένα σημαντικό πλεονέκτημα των απτικών διεπαφών χρήστη είναι η ικανότητά τους να μειώνουν την καμπύλη εκμάθησης ενός προγραμματιστικού περιβάλλοντος. Δεδομένου ότι οι χρήστες δεν χρειάζεται πλέον να μάθουν πώς να χρησιμοποιούν ποντίκι ή πληκτρολόγιο, αρκεί να έχουν την έμφυτη ικανότητα να χειρίζονται αντικείμενα του πραγματικού κόσμου, όπως κύβους ή παζλ [19]. Κατά συνέπεια, αναμένεται ότι τα απτά συστήματα μειώνουν το βάρος της γνώσης ενός ατόμου που πρέπει να κατακτήσει ένα προγραμματιστικό περιβάλλον, εστιάζοντας έτσι την προσοχή του στη δουλειά της εκμάθησης του προγραμματισμού [20].

Στα τέλη της δεκαετίας του 1970, η Radia Perlman, ερευνήτρια στο εργαστήριο πολυμέσων του M.I.T., συνειδητοποίησε ότι οι περισσότεροι νέοι κάτω των 11 έως 14 ετών δεν ήταν έτοιμοι να αρχίσουν να προγραμματίζουν με τον συμβατικό τρόπο, π.χ. εισάγοντας εντολές Logo σε έναν υπολογιστή με πληκτρολόγιο. Μια από τις πιο δύσκολες πτυχές του προγραμματισμού για τα παιδιά δεν ήταν μόνο η δημιουργία του κώδικα αλλά και η διεπαφή χρήστη.

Η Perlman άρχισε τότε να δημιουργεί διεπαφές που θα επέτρεπαν ακόμη και σε παιδιά προσχολικής ηλικίας να μάθουν πώς να προγραμματίζουν μια χελώνα. Οι προσπάθειες αυτές οδήγησαν στην ανάπτυξη της πρώτης διεπαφής αυτού του τύπου, του Tortis - Slot machine [19] (Εικόνα 1.3). Από εκείνη την ημέρα και έπειτα, προέκυψε μια ποικιλία σχεδιαστικών προσεγγίσεων για τον απτό προγραμματισμό, οι οποίες περιγράφονται εν συντομία παρακάτω.



*Εικόνα 1.3: Ο κουλοχέρης Tortis*

Ο απτός προγραμματισμός με χορδές είναι μια τεχνολογία που έχει σχεδιαστεί για την ανάπτυξη βασικών προγραμμάτων για τον έλεγχο ρομπότ παιχνιδιών [21]. Προκειμένου οι χρήστες να κατασκευάσουν ένα πρόγραμμα σε αυτό το σύστημα, τα γεγονότα πρέπει να συνδυαστούν με ενέργειες που στη συνέχεια εκτελούνται ως απάντηση σε αυτά τα γεγονότα.

Αυτές οι συνδέσεις εμφανίζονται ως φωτογραφίες που συνδέονται με μια χορδή (στην πραγματικότητα πρόκειται για ένα σύρμα στριμμένο σαν χορδή). Ο χρήστης συνδέει γεγονότα που δημιουργούνται από αισθητήρες με ενέργειες που εκτελούνται ως αντίδραση σε αυτά τα ερεθίσματα (Εικόνα 1.4). Έτσι, το σύστημα παράγει ένα πρόγραμμα που μπορεί να τοποθετηθεί στη μνήμη του ρομπότ και είναι σχεδιασμένο να λειτουργεί με βάση τον τρόπο με τον οποίο συνδέθηκαν οι χορδές.



*Εικόνα 1.4: Προγραμματισμός με χορδές*

Η Quetzal και η Tern είναι δύο άλλες απτές γλώσσες προγραμματισμού με εκπαιδευτικό προσανατολισμό [22]. Αυτές οι γλώσσες χρησιμοποιούν ανόργανα αντικείμενα που στερούνται ηλεκτρονικών κυκλωμάτων και αναγκών ενέργειας. Τα μέρη προγραμματισμού κάθε γλώσσας είναι πανομοιότυπα και μοιάζουν με κομμάτια παζλ. Η Quetzal είναι μια γλώσσα που χρησιμοποιείται για τον χειρισμό ενός ρομπότ Lego Mindstorm, ενώ η Tern (Εικόνα 1.5) χρησιμοποιείται για τον έλεγχο ενός εικονικού ρομπότ που βασίζεται σε υπολογιστή. Λόγω του γεγονότος ότι και οι δύο γλώσσες δημιουργήθηκαν από τους ίδιους ανθρώπους κατά την ίδια χρονική περίοδο, οι φιλοσοφίες τους είναι παρόμοιες. Οι μαθητές που χρησιμοποιούν αυτές τις γλώσσες προγραμματίζουν σε λειτουργία εκτός σύνδεσης και χρησιμοποιούν μια φορητή συσκευή σάρωσης για τον εντοπισμό εντολών. Το σύστημα αυτό κατανοεί ποιες εντολές υπάρχουν και πώς συνδυάζονται για την κατασκευή ενός προγράμματος αναλύοντας τα κομμάτια του παζλ. Τέλος, είναι σημαντικό να σημειωθεί ότι

και οι δύο γλώσσες επιτρέπουν στον χρήστη να παρέχει επιχειρήματα στις προτροπές εντολών.



Εικόνα 1.5: Quetzal και Tern αριστερά και δεξιά αντίστοιχα

### 1.5.2 Περιορισμοί των εργαλείων απτού προγραμματισμού

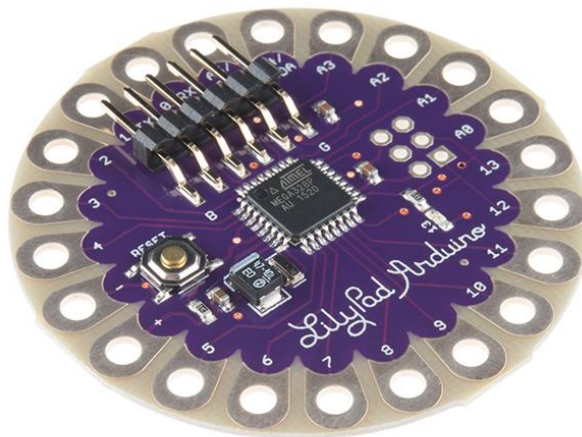
Αν και έχουν γίνει αρκετές προσπάθειες για τη δημιουργία απτών συστημάτων, υπάρχει έλλειψη απτών εργαλείων προγραμματισμού [23]. Οι ακόλουθοι περιορισμοί φαίνεται να υφίστανται αναλυτικότερα: α) μερικά συστήματα δεν διαθέτουν επαρκή αριθμό εντολών και παραμέτρων [24], γεγονός που εμποδίζει την αφομοίωση των εννοιών προγραμματισμού, β) η έλλειψη ελέγχου σε πραγματικό χρόνο εμποδίζει την ομαλή αλληλεπίδραση μεταξύ του προγραμματιστή και του προγράμματος [25], γ) ορισμένα συστήματα απαιτούν ειδικές επιφάνειες ή ράγες, που καθιστούν δύσκολη τη μετεγκατάστασή τους και, επομένως, δύσκολη τη χρήση τους σε πραγματικές αίθουσες διδασκαλίας, και δ) παρά το γεγονός ότι τα απτά συστήματα τείνουν να είναι πιο αποτελεσματικά από τις γραφικές διεπαφές χρήστη, απαιτείται περαιτέρω μελέτη για να διευκρινιστούν οι συνθήκες υπό τις οποίες εμφανίζονται τα οφέλη σε πολλούς τομείς [26].

### 1.5.3 Προσέλκυση ευρύτερου κοινού

Η ανεπάρκεια της εκπαίδευσης στην επιστήμη της πληροφορικής να προσλάβει και να διατηρήσει τη συμμετοχή διαφορετικών φοιτητών, τόσο εντός όσο και εκτός των ιδρυμάτων, είναι ένα διάχυτο και αποκαρδιωτικό ζήτημα.. Είναι πιθανό ότι έχουμε φτάσει σε ένα σημείο καμπής με την εισαγωγή νέων και πιο περιεκτικών μαθησιακών πλαισίων και προγραμμάτων που εμπλέκουν φοιτητές από ιστορικά υποεκπροσωπούμενα υπόβαθρα στην επιστήμη των

υπολογιστών και σε συναφή θέματα. Ωστόσο, η βαρύτητα των δεδομένων από προγράμματα μεταδευτεροβάθμιων σπουδών δείχνει ότι υπάρχει ακόμη σημαντική δουλειά να γίνει [27].

Ένα από τα πιο ελκυστικά χαρακτηριστικά της απτής πληροφορικής είναι η ικανότητά της να ασχολείται με ένα ευρύτερο κοινό, αντανακλώντας ένα ευρύτερο φάσμα πολιτισμικών παραδόσεων, εθίμων και αξιών. Στον τομέα της υλικής πληροφορικής, το LilyPad Arduino [28] (Εικόνα 1.6) και τα ηλεκτρονικά υφάσματα χρησιμοποιούν ευρηματικά σχέδια για να συνδέσουν τον υπολογιστή και τα ηλεκτρονικά υλικά με παραδόσεις χειροτεχνίας, όπως το ράψιμο, το σκίτσο και η μόδα. Οι εν λόγω βιοτεχνικές παραδόσεις χαρακτηρίζονται από διαγενεακές κοινότητες πρακτικής με βαθιές πολιτιστικές ρίζες. Αν και η μετάβαση μπορεί να μην είναι πλήρως κατανοητή, είναι λεπτή. Αυτές οι εργαλειοθήκες και οι πόροι καταλαμβάνονται ιδανικά από τις υπάρχουσες κοινότητες ως ένα νέο μέσο έκφρασης, που έχει τις ρίζες του σε διαφορετικές δυναμικές κοινότητες και πρακτικές. Αυτά τα καινοτόμα συστήματα αναδεικνύουν τη δυνητική δυνατότητα εφαρμογής της εκπαίδευσης των υπολογιστών σε ένα πολύ ευρύτερο φάσμα ενδιαφερόντων και δραστηριοτήτων (συμπεριλαμβανομένων των παραδόσεων της χειροτεχνίας, της μουσικής, του χορού, της μόδας, των εικαστικών τεχνών, της αφήγησης ιστοριών κ.ο.κ.) (Εικόνα 1.7).



Εικόνα 1.6: Το LilyPad Arduino



Εικόνα 1.7: Εφαρμογή LilyPad Arduino σε παράσταση μόδας

## 1.6 Εκπαιδευτικά Συστήματα Βασισμένα σε Κύβους

Μετά από έρευνα σε διάφορους διαδικτυακούς τόπους και τεχνολογικού και εκπαιδευτικού περιεχομένου θα παρατεθούν κάποια παρόμοια εκπαιδευτικά συστήματα βασισμένα και αυτά σε κύβους. Θα περιγραφούν τα χαρακτηριστικά τους καθώς και τις περιπτώσεις που καλύπτουν. Στο τέλος της ενότητας θα γίνει σύγκρισή τους με το σύστημα της εν λόγω εργασίας και θα παρατεθούν τα πλεονεκτήματα του τελευταίου έναντι των προαναφερθέντων δοθέντων παραδειγμάτων.

### 1.6.1 Block Jam

Το σύστημα Block Jam [29] (Εικόνα 1.8) είναι μια απτική διεπαφή και ταυτόχρονα μια οπτική γλώσσα η οποία επιτρέπει σε αρχάριους και μη, να τη λειτουργήσουν. Το σύστημα αποτελείται από 26 αντικείμενα τα οποία, ανάλογα με τη σύνδεση μεταξύ τους και τις ρυθμιζόμενες ιδιότητες τους, μπορούν να παράγουν μουσικές 'φράσεις'. Υπάρχουν δύο είδη κουτιών, τα play blocks και τα path blocks. Τα play blocks είναι υπεύθυνα για το ξεκίνημα, το σταμάτημα και τον έλεγχο της ταχύτητας της cue ball ενώ τα path blocks ελέγχουν τη διαδρομή της cue ball. Τα play block μπορούν να συνδεθούν μόνο με τη μία τους πλευρά. Τα path block όμως μπορούν να συνδεθούν και προς τις τέσσερις. Όταν τα κουτιά συνδεθούν μεταξύ τους δημιουργούν ένα σύμπλεγμα. Το σύμπλεγμα αυτό συνδέεται με τον υπολογιστή μέσω ενός καλωδίου και ενός κουτιού βάσης το οποίο λέγεται mother block. Το καλώδιο δίνει ρεύμα στο σύμπλεγμα και αποτελεί έναν δίαυλο επικοινωνίας με τον υπολογιστή. Η cue ball κινείται νοητά από τον ένα κύβο στο άλλο ανάλογα με τους κανόνες που επιβάλουν τα



path blocks. Ο κάθε κύβος περιέχει κάποιον ήχο και έτσι καθώς η cue ball ή οι cue balls κινούνται από κύβο σε κύβο παράγουν και δημιουργούν ήχους με αποτέλεσμα ο χρήστης κατά κάποιο τρόπο να συνθέτει μουσική.



Εικόνα 1.8: Block Jam

### 1.6.2 Algo Block

Το project AlgoBlock [30] είναι ένα εκπαιδευτικό περιβάλλον προγραμματισμού, προσαρμοσμένο για την πρωτοβάθμια και δευτεροβάθμια εκπαίδευση δηλαδή για μικρότερους μαθητές (Εικόνα 1.9). Το σύστημα αποτελείται από μια συλλογή από κύβους, οι οποίοι μπορούν να ενωθούν μεταξύ τους, με σκοπό να αναπαράγουν τον κώδικα του προγράμματος και να καθοδηγήσουν ένα εικονικό υποβρύχιο. Οι κύβοι αυτοί, συνδέονται με έναν υπολογιστή, στον οποίο μεταφέρουν το παραγόμενο πρόγραμμα. Κάθε κύβος αναπαριστά μια εντολή, η οποία είναι παρόμοια με μια εντολή της γλώσσας Logo. Χρησιμοποιώντας το σύστημα αυτό, οι χρήστες προσπαθούν να οδηγήσουν το υποβρύχιο το οποίο απεικονίζεται σε μια οθόνη CRT. Το AlgoBlocks αξιοποίησε και ενσάρκωσε για πρώτη φορά τον όρος απτός προγραμματισμός δημιουργώντας παράλληλα και την πρώτη απτική γλώσσα προγραμματισμού και έπαιξε καθοριστικό ρόλο στην ανάπτυξη τέτοιων συστημάτων.



Εικόνα 1.9: AlgoBlock

### 1.6.3 KIBO

Το KIBO επιτρέπει στους νέους να κατασκευάσουν, να ζωγραφίσουν και να προγραμματίσουν τα δικά τους ρομπότ χωρίς να χρειάζονται υπολογιστές, tablet ή smartphones (Εικόνα 1.10). Τα παιδιά προγραμματίζουν τα ρομπότ τους συναρμολογώντας ακολουθίες οδηγιών (προγράμματα) χρησιμοποιώντας ξύλινα μπλοκ KIBO, τα οποία στη συνέχεια σαρώνουν χρησιμοποιώντας έναν σαρωτή γραμμωτού κώδικα ενσωματωμένο στο σώμα του ρομπότ. Η γλωσσική σύνταξη του KIBO (δηλαδή μια διαδοχική σύνδεση των τούβλων) έχει ως στόχο να βοηθήσει και να ενισχύσει τις δεξιότητες αλληλουχίας των μικρών παιδιών [6].



Εικόνα 1.10: Το ρομπότ KIBO και τα μπλοκ του

#### 1.6.4 Electronic Blocks - roBlocks

Το project Electronic blocks [31] σχεδιάστηκε ώστε οι χρήστες να μπορούν να δημιουργήσουν προγραμματιζόμενες ρομποτικές κατασκευές και μηχανισμούς (Εικόνα 1.11). Τα παιδιά τοποθετώντας τους ηλεκτρονικούς κύβους τον ένα πάνω στον άλλο ή και σε διάφορες θέσεις, μπορούν να δημιουργήσουν κατασκευές με διαφορετικές λειτουργίες η κάθε μία. Υπάρχουν τρεις τύποι κύβων: οι κύβοι αισθητήρες, οι κύβοι δράσης και οι λογικοί κύβοι. Οι κύβοι αισθητήρες μπορούν να εντοπίζουν φως, επιφάνεια και ήχο. Οι λογικοί κύβοι παίζουν ρόλο του συνδετικού κρίκου και διαχειρίζονται τους κύβους δράσης. Με αυτόν τον τρόπο τα παιδιά μπορούν να φτιάξουν ένα ρομπότ το οποίο κινείται όταν εντοπίζει φως και ήχους ή και τα δύο ταυτόχρονα.



Εικόνα 1.11: roBlocks (αργότερα Cubelets)

#### 1.6.5 Music Blocks

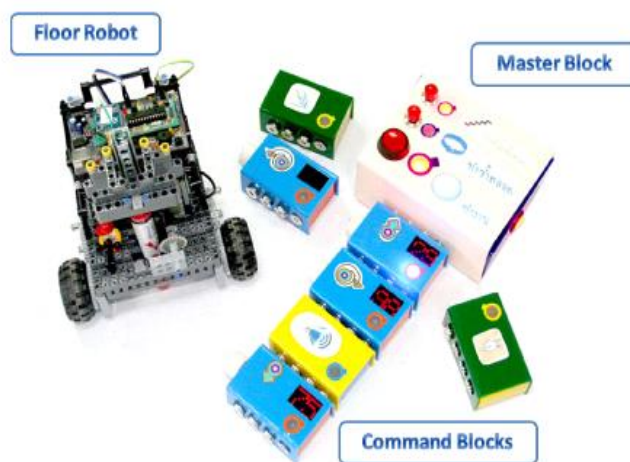
Το σύστημα Music Block [14] ονομάζεται πλέον και αλλιώς ως “Neurosmith Music Blocks” (Εικόνα 1.12). Η πρώτη εμπορική έκδοσή του πρωτοεμφανίστηκε πάνω από 15 χρόνια πριν. Το σύστημα αποτελείται από 5 χρωματιστούς κύβους, καθένας εκ των οποίων αναπαριστά ένα μουσικό κομμάτι από μελωδίες. Μόλις ο χρήστης τοποθετήσει τους κύβους στο σύστημα αναπαραγωγής, ακούγεται το αντίστοιχο μουσικό κομμάτι της μελωδίας που εκπροσωπεί ο κάθε κύβος. Αν ο χρήστης πατήσει το κουμπί “Αναπαραγωγή”, ακούγεται ολόκληρη η μουσική σύνθεση. Προφανώς το σύστημα είναι σχεδιασμένο έτσι ώστε να το αποτέλεσμα να εξαρτάται και από την σειρά που θα τοποθετηθούν οι κύβοι. Προκειμένου το σύστημα να γίνει περισσότερο διαδραστικό και ενδιαφέρον κάθε πλευρά του κύβου αντιστοιχεί σε διαφορετική τονική αναπαράσταση της μελωδίας.



Εικόνα 1.12: Music Blocks

### 1.6.6 Robo Blocks

Το σύστημα αυτό είναι ένα κλασικό παράδειγμα απτικού προγραμματισμού όπως το AlgoBlocks (Εικόνα 1.13). Αποτελεί μια συλλογή από κύβους κατασκευασμένους από πλέξιγκλας με ενσωματωμένους μικροελεγκτές και αισθητήρες που ελέγχουν την κίνηση ενός επιτραπέζιου ρομπότ, όταν αυτοί συνδεθούν μεταξύ τους [32]. Το σύστημα υποστηρίζει βασικές λειτουργίες κίνησης όπως “προχώρα μπροστά”, “στρίψε δεξιά” και “στρίψε αριστερά”. Προκειμένου να ορίσουν πόσα βήματα μπροστά θα προχωρήσει το ρομπότ ή πόσες φορές θα στρίψει, οι χρήστες μπορούν να χρησιμοποιήσουν έναν διακόπτη ώστε να αλλάξουν τον αριθμό που εμφανίζεται σε μια οθόνη πάνω σε κάθε εντολή. Το ρομπότ είναι εξοπλισμένο με μια πένα η οποία μπορεί να αφήνει μελάνι, σχηματίζοντας τη διαδρομή που αυτό ακολουθεί. Τέλος το σύστημα διαθέτει και λειτουργία αποσφαλμάτωσης η οποία επιτρέπει στο ρομπότ να εκτελεί τις εντολές κάθε μια ξεχωριστά.



Εικόνα 1. 13: Robo Blocks

### 1.6.7 Cubetto

Το cubetto είναι φτιαγμένο από απτό και σκληρό ξύλο χωρίς οθόνη και εισάγει τα παιδιά ηλικιών 3-5 ετών στα πρώτα βήματά τους στην κωδικοποίηση [32] (Εικόνα 1.14). Με μια γλώσσα κωδικοποίησης παρόμοια με αυτή της LEGO® κάθε μπλοκ στον ειδικά διαμορφωμένο πίνακα κίνησης του cubetto είναι μια ενέργεια. Τα παιδιά καλούνται να συνδυάσουν αυτές τις ενέργειες ώστε να δημιουργήσουν ένα πρόγραμμα. Υπάρχουν διαφορετικά μπλοκ με αντίστοιχα σχήματα και χρώματα που εκπροσωπούν διαφορετικές εντολές (Forward, Left, Right, Function Backward, Negation, Random). Τα παιδιά μπορούν να τοποθετήσουν τα τετράγωνα στο ταμπλό για να πουν στο Cubetto πού να πάει πατώντας ένα μπλε κουμπί. Διατίθεται με διάφορους χάρτες για εξερεύνηση στους οποίους μπορούν τα παιδιά να περιηγηθούν μέσω της κίνησης του cubetto ώστε η αλληλεπίδρασή τους με το σύστημα να είναι πιο ενδιαφέρουσα.



Εικόνα 1.14: Cubetto

## 1.7 Συμπεράσματα

Οι συσκευές με ευαισθησία αφής και οι οθόνες υπολογιστών έχουν σημειώσει σημαντική πρόοδο τις τελευταίες δύο δεκαετίες, αλλά οι φυσικές διεπαφές εξακολουθούν να βασίζονται κυρίως σε αλγόριθμους όρασης υπολογιστή ή σε ενσωματωμένα ηλεκτρικά εξαρτήματα. Η αποθήκευση και η αποκατάσταση προγραμμάτων είναι παρόμοια πρόκληση, τουλάχιστον με την τρέχουσα τεχνολογία, επειδή είναι αδύνατο να αναδημιουργηθούν αυτόματα φυσικές δομές. Οι λειτουργίες του πρόχειρου (π.χ. αντιγραφή, αποκοπή, επικόλληση και διαγραφή) των γλωσσών προγραμματισμού που βασίζονται σε κείμενο και των οπτικών γλωσσών προγραμματισμού θα ήταν δύσκολο να υλοποιηθούν με τις περισσότερες απτές γλώσσες για τους ίδιους λόγους, καθιστώντας τον προγραμματισμό δυσκίνητο για τους πιο εξελιγμένους

προγραμματιστές που επιθυμούν να αντιγράφουν και να επεξεργάζονται εύκολα τον υπάρχοντα κώδικα.

Παρόλο που βλέπουμε μεγάλες δυνατότητες στην απτική πληροφορική, η πλειονότητα των σημερινών ερευνών επικεντρώνεται σε μικρότερα παιδιά, με συγκριτικά περιορισμένη εστίαση σε μαθητές μεγαλύτερης ηλικίας. Υπάρχουν διάφορες αιτίες. Η μετάβαση από τα μέσα που βασίζονται σε μεγάλο βαθμό στην οθόνη συνδέεται με την εκπαίδευση στην πρώιμη παιδική ηλικία, και για τα μικρά παιδιά με αναπτυσσόμενες ικανότητες ανάγνωσης και λεπτής κινητικότητας, τα χειροπιαστά αντικείμενα παρέχουν ένα δελεαστικό και ίσως πιο προσιτό σημείο εισόδου στον ηλεκτρονικό γραμματισμό. Ωστόσο, ένα άλλο επιχείρημα σχετίζεται με τους τεχνολογικούς περιορισμούς των φυσικών διεπαφών και την καταλληλότητά τους για πιο εξελιγμένες εκπαιδευτικές, επαγγελματικές και πραγματικές καταστάσεις.

## **1.8 Σύνοψη Διπλωματικής Εργασίας**

Στη συνέχεια γίνεται η ανάλυση και η περιγραφή της διαδικασίας σχεδίασης αλλά και υλοποίησης του συστήματος, η οποία περιλαμβάνει τρία κεφάλαια. Το συγκεκριμένο κεφάλαιο αφορά την εισαγωγή και την επεξήγηση του αντικειμένου της μελέτης που πραγματεύεται η παρούσα εργασία. Πιο συγκεκριμένα έγινε αναφορά στην ηλεκτρονική μάθηση γενικότερα, καθώς και τα βασικά χαρακτηριστικά της και τέλος στον απτό προγραμματισμό, την χρησιμότητά του και τη σημασία που έχει για την εκπαιδευτική διαδικασία. Επίσης γίνεται αναφορά σε παρόμοια εκπαιδευτικά συστήματα που λειτουργούν με βάση το μοντέλο εκμάθησης μέσω του απτού προγραμματισμού.

## Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο

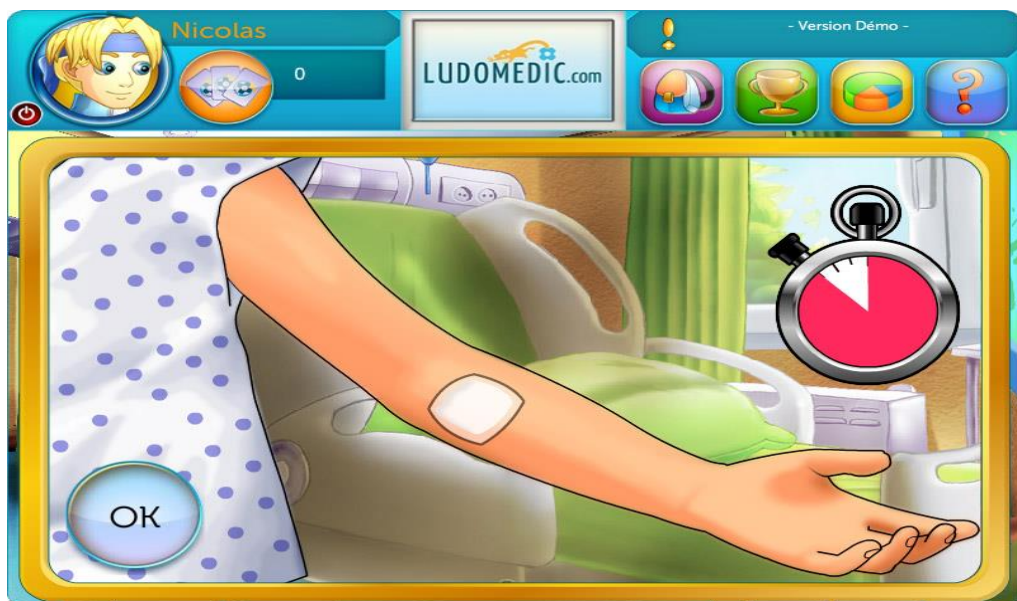
Σε αυτό το κεφάλαιο θα παρουσιαστεί το θεωρητικό υπόβαθρο που βασίστηκε η υλοποίηση του συστήματος και αναλυτικά μερικοί από τους όρους, έννοιες και αλγορίθμους που χρησιμοποιήθηκαν. Επίσης θα παρατεθεί ο εξοπλισμός και τα εργαλεία που αξιοποιήθηκαν για την κατασκευή κάποιων από τα επιμέρους στοιχεία που αποτελούν το σύστημα, καθώς και τα λογισμικά για την ανάπτυξή του.

### 2.1 Ηλεκτρονικό Παιχνίδι-Σοβαρό Παιχνίδι

Ηλεκτρονικό παιχνίδι ονομάζεται ένα παιχνίδι που χρησιμοποιεί ηλεκτρονικά στοιχεία για να δημιουργήσει ένα διαδραστικό σύστημα, μέσα στο οποίο μπορεί να παίξει ένας παίκτης. Σήμερα, η πιο κοινή μορφή ηλεκτρονικού παιχνιδιού είναι το βιντεοπαιχνίδι (Εικόνα 2.1), και για αυτόν το λόγο οι όροι συχνά χρησιμοποιούνται λανθασμένα, ως συνώνυμοι. Άλλα κοινά είδη ηλεκτρονικού παιχνιδιού είναι τα φορητά ηλεκτρονικά παιχνίδια, τα αυτόνομα συστήματα (π.χ. pinball, ή ηλεκτρομηχανικά παιχνίδια arcade), και συγκεκριμένα μη-οπτικά προϊόντα (π.χ. παιχνίδια ήχου). Το παιχνίδι το οποίο θα υλοποιηθεί στο σύστημα εντάσσεται στην κατηγορία των σοβαρών παιχνιδιών (Εικόνα 2.2) λόγω του ότι μεταφέρονται γνώσεις και δεξιότητες μέσω αυτού και έχει συγκεκριμένο εκπαιδευτικό χαρακτήρα. Τα σοβαρά παιχνίδια (serious games) είναι ένας όρος που γίνεται όλο και πιο δημοφιλής και έχει καταχωρηθεί χωρίς όμως να υπάρχει ένας μοναδικός ορισμός της έννοιας. Ο Zyda (2005) δίνει έναν επίσημο ορισμό οποίος αναφέρει ότι: "Σοβαρό παιχνίδι: ένας νοητικός διαγωνισμός, που παίζεται με έναν υπολογιστή σύμφωνα με συγκεκριμένους κανόνες, που χρησιμοποιεί την ψυχαγωγία για περαιτέρω κυβερνητική ή εταιρική εξάσκηση, εκπαίδευση, υγεία, δημόσια πολιτική και στρατηγικούς στόχους επικοινωνίας"[33]. Επομένως συμπεραίνουμε ότι το παιχνίδι που δημιουργήθηκε στην εργασία εκτός από ηλεκτρονικό λόγω της φύσης του μπορεί να χαρακτηριστεί και ως σοβαρό λόγω των ιδιοτήτων του .



Εικόνα 2.1: Το βιντεοπαιχνίδι Super Mario Bros στις πρώτες εκδόσεις του



Εικόνα 2.2: Σοβαρό παιδικό παιχνίδι προσομοίωσης νοσοκομειακής περίθαλψης

### 2.1.1 Car Racing Games

Τα car racing games (Εικόνα 2.3, Εικόνα 2.4) είναι ένα είδος βιντεοπαιχνιδιού στο οποίο ο παίχτης συμμετέχει σε έναν αγώνα μεταξύ αυτοκινήτων. Μπορεί να βασίζονται σε οτιδήποτε, από πρωτάθλημα με αγώνες του πραγματικού κόσμου μέχρι αγώνες σε φανταστικά περιβάλλοντα και καταστάσεις. Κατανέμονται σε ένα φάσμα μεταξύ πιο ρεαλιστικών προσομοιώσεων αγώνων και πιο φανταστικών παιχνιδιών τύπου arcade. Ο



κύριος στόχος σε αυτούς τους αγώνες είναι ο παίχτης από την γραμμή εκκίνησης να περάσει τη γραμμή τερματισμού και κατ' επέκταση να σημειώσει τον καλύτερο χρόνο ώστε να θεωρηθεί πρώτος.



*Εικόνα 2.3: Racing game πρώτης γενιάς*



*Εικόνα 2.4: Racing game τελευταίας γενιάς*

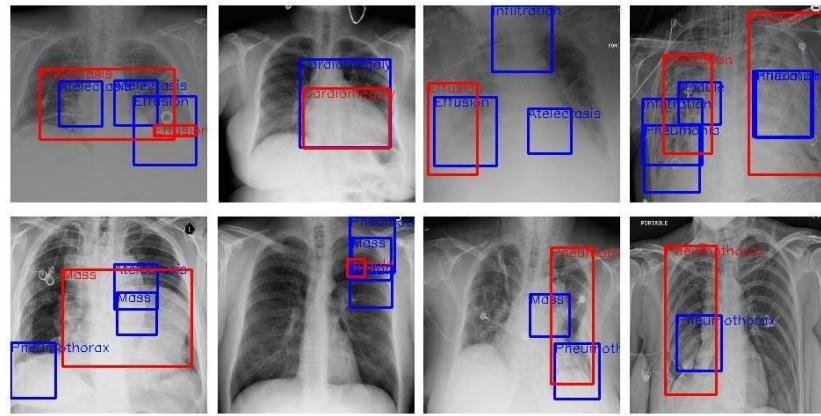
## 2.2 Μηχανική Όραση

Η Μηχανική Όραση αποτελεί ουσιαστικά ένα από τα πεδία της Τεχνητής Νοημοσύνης (Artificial Intelligence), που αναπτύσσεται ραγδαία τα τελευταία χρόνια λόγω της εξέλιξης της ίδιας της τεχνολογίας αλλά και των όλο και περισσότερων συστημάτων που τη χρησιμοποιούν. Επιτρέπει στην ουσία στον υπολογιστή αλλά και σε πιο πολύπλοκα συστήματα την άντληση πληροφοριών από μία ή περισσότερες ψηφιακές εικόνες, βίντεο αλλά και άλλες οπτικές εισροές, την επεξεργασία τους, και στη συνέχεια την εκτέλεση των αντίστοιχων ενεργειών, την παραγωγή συμβολικών δεδομένων ή ακόμη και τη σύσταση αλλαγών με βάση το περιεχόμενό τους. Εάν η Τεχνητή Νοημοσύνη δίνει τη δυνατότητα στον υπολογιστή να ‘σκέφτεται’, η Μηχανική Όραση του επιτρέπει να ‘παρατηρεί’, να ‘κατανοεί’ και να ‘καταλαβαίνει’ (Εικόνα 2.5). Οι εφαρμογές της Μηχανικής Όρασης πληθαίνουν τόσο σε τεχνολογίες που εφαρμόζεται, όσο και σε τομείς που χρησιμοποιείται. Τέτοιες εφαρμογές της Μηχανικής Όρασης μπορούμε να παρατηρήσουμε [34] :

- Στην ιατρική (ανάλυση ακτίνων X) (Εικόνα 2.6)
- Σε αυτοκινούμενα οχήματα τελευταίας γενιάς (Εικόνα 2.7)
- Στη διαδραστική ψυχαγωγία
- Στη μεταφορά ανθρώπων και προϊόντων
- Στην κατασκευή οποιουδήποτε οικοδομήματος (εκτίμηση 3D μοντέλων)



Εικόνα 2.5: Αναγνώριση και Ταυτοποίηση προσώπου αντίστοιχα μέσω χρήσης της Μηχανικής Όρασης



Εικόνα 2.6: Εφαρμογή Μηχανικής Όρασης σε ακτινογραφίες θώρακα



Εικόνα 2.7: Ζωντανό στιγμότυπο εικόνας κάμερας όπισθεν αυτοκινήτου όπου εφαρμόζεται Μηχανική Όραση

### 2.2.1 Ψηφιακή Επεξεργασία Εικόνας

Στην σημερινή εποχή η επεξεργασία εικόνας ανήκει στην κατηγορία με τις ραγδαία αναπτυσσόμενες τεχνολογίες και αποτελεί έναν από τους βασικούς ερευνητικούς τομείς στον κλάδο της μηχανικής και της επιστήμης των υπολογιστών γενικότερα. Πρόκειται στην ουσία για μία μέθοδο με την οποία μέσω της εκτέλεσης ορισμένων λειτουργιών στην εικόνα λαμβάνουμε μία βελτιωμένη έκδοσή της ή μπορούμε και να αποσπάσουμε ορισμένες πληροφορίες από αυτή. Αποτελεί έναν τύπο επεξεργασίας σήματος κατά τον οποίο η είσοδος είναι μία εικόνα και η έξοδος μία επίσης επεξεργασμένη εικόνα ή χαρακτηριστικά που

σχετίζονται με την εικόνα εισόδου [35]. Βασικά η Επεξεργασία Εικόνας περιλαμβάνει τα εξής τρία βήματα:

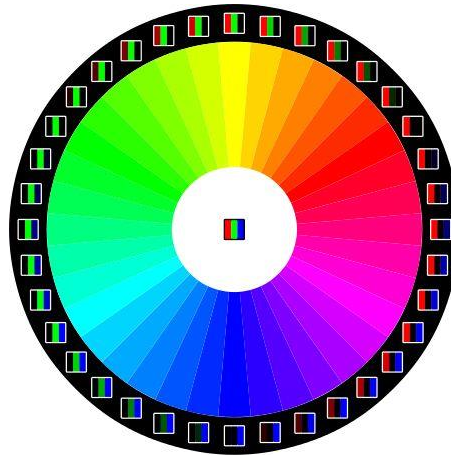
- Εισαγωγή Εικόνας μέσω ειδικών εργαλείων απόκτησης εικόνας
- Ανάλυση και διαχείριση της εικόνας
- Έξοδος η οποία μπορεί να είναι μία τροποποιημένη εικόνα ή μία αναφορά που βασίζεται στην ανάλυση της εικόνας

Η εικόνα κατά την είσοδο στο οποιοδήποτε μέσο επεξεργασίας της ορίζεται ως μία δυσδιάστατη συνάρτηση  $f(x,y)$ , στην οποία  $x$  και  $y$  είναι οι συντεταγμένες. Το πλάτος του  $f$  σε οποιοδήποτε ζεύγος μεταξύ  $x$  και  $y$  λέγεται ένταση για το κάθε σημείο της εικόνας. Εάν οι τιμές των  $x$  και  $y$  είναι διακριτές ποσότητες τότε λέμε ότι έχουμε μία ψηφιακή εικόνα.

### 2.2.1.1 Μετατροπή Χρωματικού Μοντέλου

Ξεκινώντας από το χρωματικό μοντέλο, ο όρος αυτός αποτελεί ουσιαστικά ένα αφηρημένο μαθηματικό μοντέλο το οποίο περιγράφει τον τρόπο με τον οποίο ένα ή και περισσότερα χρώματα μπορούν να αναπαρασταθούν ως πλειάδες αριθμών, στις πιο πολλές περιπτώσεις τριών ή τεσσάρων τιμών [36]. Ένα χαρακτηριστικό παράδειγμα χρωματικού μοντέλου αποτελεί το RGB το οποίο είναι ένα από τα πιο διαδεδομένα χρωματικά μοντέλα που εφαρμόζεται στα ηλεκτρονικά συστήματα του σήμερα (π.χ τηλεοράσεις και υπολογιστές). Κατά την εφαρμογή του RGB σε οποιοδήποτε σύστημα όλα τα χρώματα αναπαρίστανται με αριθμούς σε ένα εύρος τιμών από το 0 έως το 255 και κωδικοποιούνται με βάση τα τρία βασικά χρώματα κόκκινο-πράσινο-μπλε (Red-Green-Blue) [37].

Η Μετατροπή Χρωματικού Μοντέλου ωστόσο ορίζεται ως η μετάφραση της αναπαράστασης ενός χρώματος από μία χρωματική βάση (μοντέλο) σε μία άλλη. Αυτή η διαδικασία έχει ως στόχο, η μεταφρασμένη πλέον εικόνα μετά την χρωματική μετατροπή, να είναι όσο το περισσότερο παρόμοια με την αρχική. Για την υλοποίηση ορισμένων εργασιών μεσαίου επιπέδου που αφορούν εικόνες είναι απαραίτητη η αλλαγή και η μετατροπή του χρωματικού μοντέλου σε κλίμακες παραδείγματος χάρη του γκρι. Η διαδικασία αυτή της μετατροπής ονομάζεται Greyscaling.



Εικόνα 2.8: Ο κύκλος του μοντέλου RGB



Εικόνα 2.9: Παράδειγμα Greyscaling

### 2.2.1.2 Κατωφλίωση (Thresholding)

Η Κατωφλίωση αποτελεί μία μέθοδο επεξεργασίας εικόνας η οποία δημιουργεί μία δυτονική ή αλλιώς δυαδική εικόνα βασισμένη σε μία ορισμένη τιμή κατωφλίου στην ένταση των pixel της αρχικής εικόνας (Εικόνα 2.10). Η εφαρμογή αυτής της συνάρτησης χαμηλού επιπέδου γίνεται εν μέρει κατά κύριο λόγο σε εικόνες σε κλίμακα του γκρι, αλλά μπορεί να εφαρμοστεί και σε έγχρωμες εικόνες. Το όριο του threshold της εικόνας καθορίζεται και ρυθμίζεται ως μία τιμή. Στην περίπτωση που τα εικονοστοιχεία (pixels) της εικόνας στην οποία εφαρμόζεται thresholding είναι κάτω από αυτή την τιμή μετατρέπονται σε μαύρα και αντιθέτως τα pixels με τιμές πάνω σε λευκά. Η διαδικασία αυτή της κατωφλίωσης περιγράφεται και ως διαχωρισμός της εικόνας σε τιμές προσκηνίου (μαύρο) και τιμές φόντου (λευκό) [38].

Η ποιότητα της διαδικασίας και άρα ως αποτέλεσμα η τελική εικόνα μετά την επεξεργασία της αρχικής με κατωφλίωση έχει ιδιαίτερη σημασία κατά την εκτέλεση οπτικής αναγνώρισης χαρακτήρων (OCR) σε αυτή. Μετά από μία επιτυχημένη ή όχι και τόσο κατωφλίωση μπορούν να διακριθούν εύκολα ή αντίστοιχα δύσκολα τα δομικά στοιχεία που αποτελούν την εικόνα και να γίνει ο σωστός διαχωρισμός του προσκηνίου από το φόντο.



*Εικόνα 2.10: Εφαρμογή Κατωφλίωσης σε εικόνα ενός τοπίου*

### **2.2.1.3 Γκαουσιανό Θόλωμα (Gaussian Blur)**

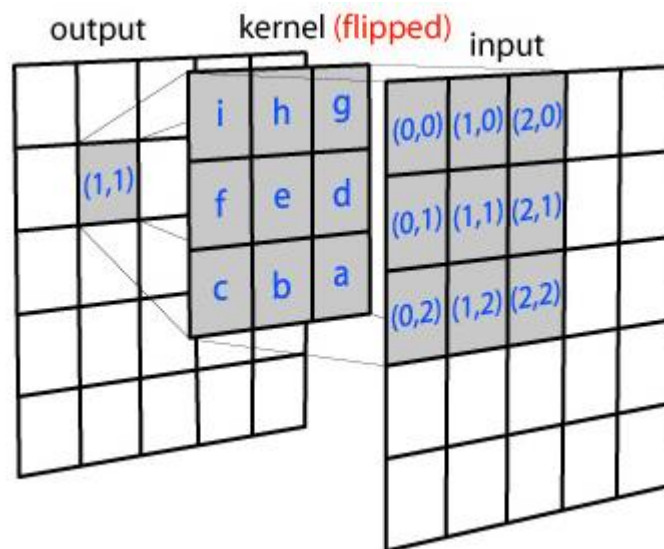
Το Γκαουσιανό Θόλωμα μπορεί να θεωρηθεί ως ένα χαμηλοπερατό φίλτρο κατά το οποίο η εικόνα που εισάγεται στο σύστημα θολώνεται (εξομαλύνεται) μέσω της συνάρτησης του Γκάους (Gauss) (Εικόνα 2.13). Σκοπός του φίλτρου αυτού είναι η διατήρηση της χαμηλής χωρικής συχνότητας και μείωση του επιπέδου του θορύβου της αρχικής εικόνας καθώς και των αμελητέων λεπτομερειών της [39].

Κατά την εκτέλεση του θολώματος ένα πλέγμα από εικονοστοιχεία επεξεργάζεται μέσω της μαθηματικής τεχνικής της συνέλιξης (convolution) (Εικόνα 2.11). Στη συνέλιξη δύο πυρήνες (kernels), που αποτελούν μία τετράγωνη συστοιχία pixel  $n \times n$ , συνδυάζονται και αθροίζονται για να παράγουν έναν τρίτο πυρήνα (Εικόνα 2.12). Αυτό γίνεται για να βεβαιωθεί ότι δεν προστίθεται η αφαιρείται ενέργεια από την εικόνα μετά την επέμβαση του θολώματος.

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Εικόνα 2.11: Παράδειγμα ενός πυρήνα Gauss όπου οι τιμές των pixel αντιστοιχούν στις τιμές μιας καμπύλης Gauss.



Εικόνα 2.12: Παράδειγμα εφαρμογής Gaussian Blur



Εικόνα 2.13: Εφαρμογή φίλτρου Gaussian Blur

#### 2.2.1.4 Ανιχνευτής Άκρων Canny

Ο ανιχνευτής άκρων Canny αναπτύχθηκε από τον John F. Canny το 1986 και αποτελεί έναν τελεστή ανίχνευσης που χρησιμοποιεί έναν αλγόριθμο πολλαπλών σταδίων με σκοπό την εύρεση ενός φάσματος ακμών σε ψηφιακές εικόνες [40]. Ο ανιχνευτής Canny δημιουργήθηκε για να ικανοποιεί τα εξής τρία κριτήρια:

- Χαμηλό ποσοστό σφάλματος, που σημαίνει την καλή ανίχνευση μόνο υπαρχόντων άκρων.
- Καλός εντοπισμός, που σημαίνει ότι η απόσταση μεταξύ των pixels των άκρων που ανιχνεύονται και των άκρων που εντοπίζονται πρέπει να ελαχιστοποιηθεί.
- Ελάχιστη απόκριση, δηλαδή μόνο μία απόκριση του ανιχνευτή ανά γραμμή που εντοπίζεται.

Μέσω της ανίχνευσης των άκρων επιτυγχάνεται η δραματική μείωση του όγκου των δεδομένων προς επεξεργασία γι' αυτό και εφαρμόζεται ευρέως σε διάφορα συστήματα που χρησιμοποιούν μηχανική όραση (Εικόνα 2.14).

Η διαδικασία της ανίχνευσης των άκρων Canny περιλαμβάνει τα εξής πέντε βήματα [41]:



- Εφαρμογή Γκαουσιανού φίλτρου για τη λείανση της εικόνας ώστε να αφαιρεθεί ο θόρυβος
- Εύρεση των διαβαθμίσεων της έντασης της εικόνας
- Εφαρμογή ορίου μεγέθους καταφλίωσης ή καταστολή αποκοπής χαμηλότερου ορίου για την απαλλαγή από τις ψευδείς ανιχνεύσεις ακμών
- Εφαρμογή διπλής καταφλίωσης για την εξακρίβωση και τον προσδιορισμό πιθανών ακμών
- Ολοκλήρωση της ανίχνευσης με την καταστολή όλων των άλλων ακμών που είναι πιο αδύναμες και δεν συσχετίζονται ή συνδέονται με ισχυρές ακμές

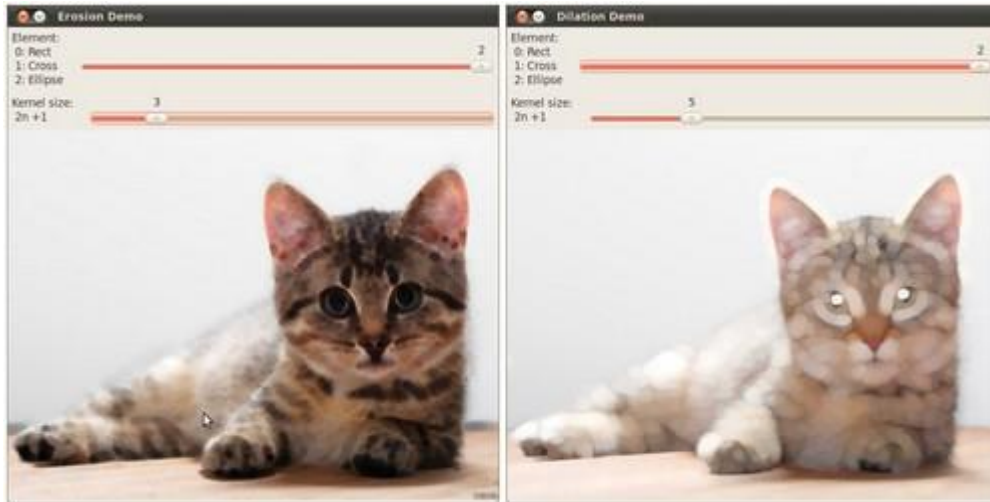


*Εικόνα 2.14: Πριν και μετά την εφαρμογή ανίχνευσης γραμμών Canny σε φωτογραφία της ηθοποιού Emma Stone*

### **2.2.1.5 Διαστολή και Διάβρωση**

Σε μια εικόνα, η διαστολή της περιλαμβάνει την προσθήκη εικονοστοιχείων στα όρια των αντικειμένων που εμπεριέχει και η διάβρωση περιλαμβάνει την απομάκρυνση των εικονοστοιχείων από τα όρια των αντικειμένων (Εικόνα 2.15). Το μέγεθος και το σχήμα του δομικού στοιχείου που χρησιμοποιείται για την επεξεργασία της εικόνας καθορίζει πόσα pixels προστίθενται ή αφαιρούνται από τα αντικείμενα της εικόνας. Πρόκειται δηλαδή για μορφολογικές διεργασίες οι οποίες δημιουργούν μία εικόνα εξόδου εφαρμόζοντας ένα δομικό

στοιχείο σε μία εικόνα εισόδου. Οι διεργασίες αυτές εφαρμόζονται κυρίως σε εικόνες που βασίζονται σε σχήματα. Μπορούν να χαρακτηριστούν και ως αδερφές διαδικασίες καθώς στο μεγαλύτερο ποσοστό των εργασιών που θα χρησιμοποιηθούν εφαρμόζονται μαζί και διαδοχικά η μία μετά την άλλη [42].



Εικόνα 2.15: Χειροκίνητη εφαρμογή Διαστολής και Διάβρωσης σε εικόνα

### 2.2.2 Ανάλυση Εικόνας

Η Ανάλυση Εικόνας είναι μία διαδικασία που χρησιμοποιείται για τον χειρισμό εικόνων για διάφορους σκοπούς (Εικόνα 2.16). Μεταξύ των τελευταίων είναι η εξαγωγή χαρακτήρων, η βελτίωση της ίδιας της εικόνας και άλλα. Η προηγμένη επεξεργασία εικόνας είναι πλέον δυνατή χάρη στους υπολογιστικούς αλγορίθμους [39]. Η Ανάλυση εικόνας λοιπόν αποτελεί έναν ευρύ όρο που καλύπτει μια σειρά τεχνικών που εντάσσονται γενικά σε αυτές τις υποκατηγορίες [43]:

- Βελτίωση εικόνας για προετοιμασία εικόνων για προβολή ή ανάλυση
- Τμηματοποίηση εικόνας για απομόνωση περιοχών και αντικειμένων ενδιαφέροντος
- Αφαίρεση θορύβου με χρήση μορφολογικού φιλτραρίσματος ή deep learning
- Ανάλυση περιοχής για εξαγωγή στατιστικών δεδομένων

Εξ ορισμού λοιπόν η διαδικασία της ανάλυσης της εικόνας έχει ως είσοδο και ως έξοδο μία εικόνα. Η είσοδος της εικόνας πρέπει να γίνεται υπό αυστηρές προϋποθέσεις καθώς η κατάλληλη επεξεργασία της κρίνεται απαραίτητη για την εξαγωγή δεδομένων. Η συμβολή άλλων επιστημών όπως η γεωμετρία, η φυσική και η στατιστική χρειάζεται για την ερμηνεία

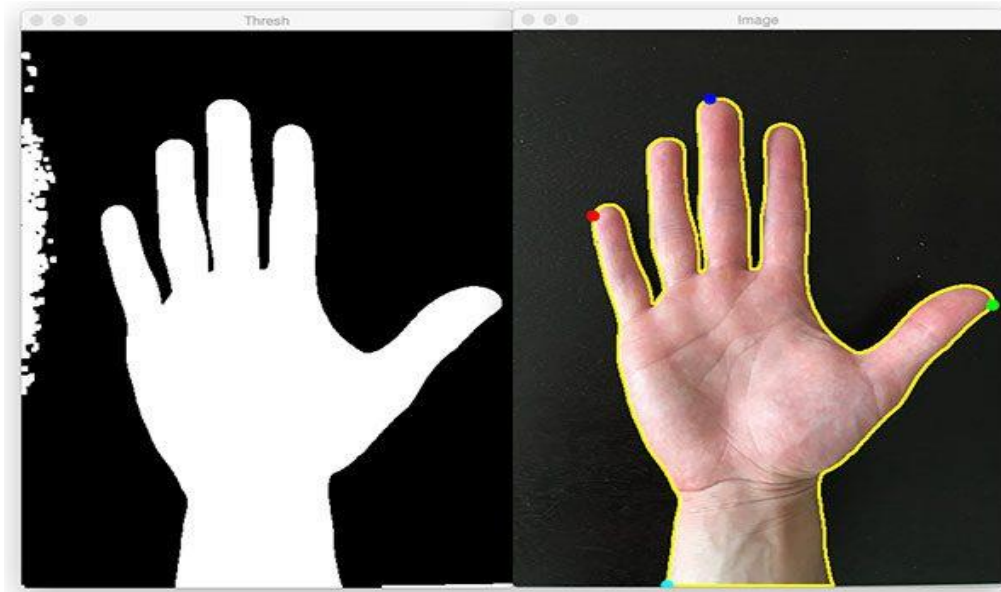
αυτών των δεδομένων και για την αντιμετώπιση προβλημάτων που καλείται να λύσει η μηχανική όραση.



*Εικόνα 2.16: Ανάλυση εικόνας με χαμηλό φωτισμό και βελτίωσή της μέσω της αφαίρεσης της θολότητας*

### **2.2.2.1 Εύρεση Περιγραμμάτων (Contouring)**

Με απλά λόγια, ένα περίγραμμα είναι μία καμπύλη που συνδέει όλα τα συνεχή σημεία κατά μήκος του ορίου που έχουν την ίδια απόχρωση ή ένταση (Εικόνα 2.17). Τα περιγράμματα είναι ένα χρήσιμο εργαλείο για τον εντοπισμό και την αναγνώριση αντικειμένων καθώς και για την ανάλυση φόρμας. Αντίθετα με την ανίχνευση ακμών όπου σκιαγραφείται μόνο το όριο ενός αντικειμένου, ένα περίγραμμα ανιχνεύει μια κλειστή φιγούρα. Για μεγαλύτερη ακρίβεια, πριν την εύρεση των περιγραμμάτων πρέπει να γίνει εισαγωγή μίας δυαδικής εικόνας και εφαρμογή κατωφλίωσης και ανίχνευση των ακμών Canny. Οι διάφορες τεχνικές εύρεσης περιγραμμάτων αφού εισαχθεί η εικόνα δεν εξάγουν εικόνα, αλλά αντικείμενα και πληροφορίες σχετικά με αυτά για χρήση από άλλες μεθόδους. Τα περιγράμματα χρησιμοποιούνται ευρέως σε τοπολογικές και ιατρικές εφαρμογές [44].

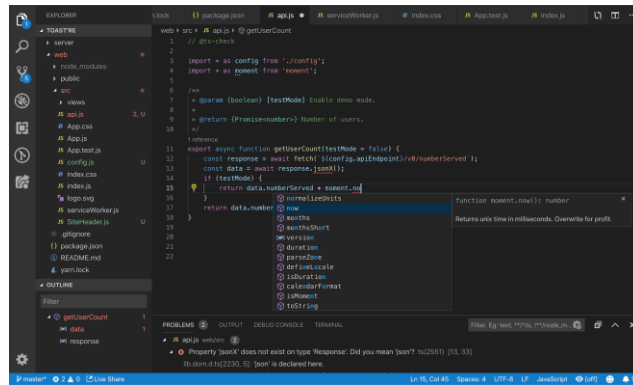


Εικόνα 2.17: Εύρεση περιγράμματος μετά την εφαρμογή κατοφλίωσης και εύρεσης γραμμών Canny

## 2.3 Πλατφόρμα Ανάπτυξης και Εργαλεία

### 2.3.1 Visual Studio Code

Το Visual Studio Code γνωστό και ως VS Code είναι ένα γρήγορο αλλά και αποτελεσματικό πρόγραμμα επεξεργασίας πηγαίου κώδικα που εκτελείται σε Windows, macOS και Linux (Εικόνα 2.18). Διαθέτει επεκτάσεις για διάφορες γλώσσες και χρόνους εκτέλεσης συμπεριλαμβανομένων των C++, C#, Java, Python, PHP και άλλα καθώς και ενσωματωμένη υποστήριξη για JavaScript, TypeScript και Node.js. Το Debugging, η επισήμανση της σύνταξης, η έξυπνη συμπλήρωση κώδικα, τα snippets, η ανακατασκευή κώδικα και το ενσωματωμένο Git είναι κάποιες από τις δυνατότητες που παρέχονται από το VS Code. Ακόμη επιτρέπει στους χρήστες να ανοίξουν έναν ή περισσότερους καταλόγους (directories), οι οποίοι μπορεί στη συνέχεια να αποθηκευτούν σε χώρους εργασίας για μελλοντική χρήση στη θέση ενός συστήματος έργου. Ως αποτέλεσμα μπορεί να λειτουργήσει ως γλωσσικά ουδέτερος επεξεργαστής κώδικα για οποιαδήποτε γλώσσα. Μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών για υπολογιστή αλλά και κινητό, ιστοσελίδων και εφαρμογών ιστού [45].



Εικόνα 2.18: Το σήμα και παράδειγμα γραφικού περιβάλλοντος σε VS Code

### 2.3.2 Python

Η Python (Εικόνα 2.19) αποτελεί μία προηγμένη γλώσσα δέσμης ενεργειών και είναι διαδραστική, ερμηνευόμενη, υψηλού επιπέδου και αντικειμενοστραφής. Εξαιρετικά κατανοητά κομμάτια κώδικα είναι φτιαγμένα μέσω της Python. Η γλώσσα περιέχει λιγότερες συντακτικές δομές από άλλες γλώσσες και συχνά χρησιμοποιεί αγγλικές λέξεις αντί για σημεία στίξης.

Η Python σχεδιάστηκε από τον Γκουίντο Βαν Ρόσουμ (Guido Van Rossum) στα τέλη της δεκαετίας του '80 στο Εθνικό Ινστιτούτο Ερευνών για τα Μαθηματικά και την Επιστήμη των Υπολογιστών στην Ολλανδία. Κατασκευάστηκε με βάση πολλές άλλες προϋπάρχουσες γλώσσες προγραμματισμού συμπεριλαμβανομένων των ABC, Modula-3, C, C++ και άλλων [46].

Βασικά χαρακτηριστικά της γλώσσας είναι ότι:

- Είναι εύκολη στην εκμάθηση διότι διαθέτει λίγες λέξεις-κλειδιά, απλή δομή και σαφώς καθορισμένη σύνταξη.
- Είναι ευκολοδιάβαστη καθώς ο πηγαίος κώδικας της Python είναι ξεκάθαρος και ευδιάκριτος από το μάτι
- Διαθέτει μία ευρεία τυπική βιβλιοθήκη καθώς το μεγαλύτερο μέρος της είναι φορητό και συμβατό με πολλές πλατφόρμες
- Είναι επεκτάσιμη, δηλαδή μπορεί ο οποιοσδήποτε να προσθέσει λειτουργικές μονάδες (modules) χαμηλού επιπέδου στον διερμηνέα της δίνοντας έτσι ελευθερία

στους προγραμματιστές να προσθέτουν και να προσαρμόζουν τα εργαλεία τους για να είναι πιο αποτελεσματικά.



*Εικόνα 2.19: Το σύμβολο της Python*

### 2.3.3 Pygame

Η Pygame (Εικόνα 2.20) αποτελεί μία συλλογή πολλαπλών πλατφόρμων από λειτουργικές μονάδες Python και χρησιμοποιείται κυρίως για τη δημιουργία βιντεοπαιχνιδιών (Εικόνα 2.21). Απαρτίζεται από βιβλιοθήκες ήχου και γραφικών που έχουν σχεδιαστεί ειδικά για χρήση μέσω της γλώσσας προγραμματισμού της Python. Ο Pete Shinnars δημιούργησε επίσημα την Pygame για να πάρει τη θέση της PySDL. Παρ' όλα αυτά η Pygame χρησιμοποιεί την βιβλιοθήκη PySDL για να επιτρέψει την ανάπτυξη παιχνιδιών σε πραγματικό χρόνο χωρίς τη χαμηλού επιπέδου γλώσσα προγραμματισμού C και τους μηχανισμούς των παραγώγων της. Αυτό βασίζεται στην ιδέα ότι πιο ακριβές ρουτίνες παιχνιδιών μπορούν να αφαιρεθούν από τη λογική του παιχνιδιού, καθιστώντας εύκολη τη δομή του χρησιμοποιώντας μια γλώσσα προγραμματισμού υψηλού επιπέδου όπως η Python. Η Pygame είναι κατάλληλη για την ανάπτυξη προγραμμάτων πελάτη (client-side programs) που ενδέχεται τελικά να περιέχονται σε ένα αυτόνομο εκτελέσιμο αρχείο [47].



*Εικόνα 2.20: Το σύμβολο της Pygame*

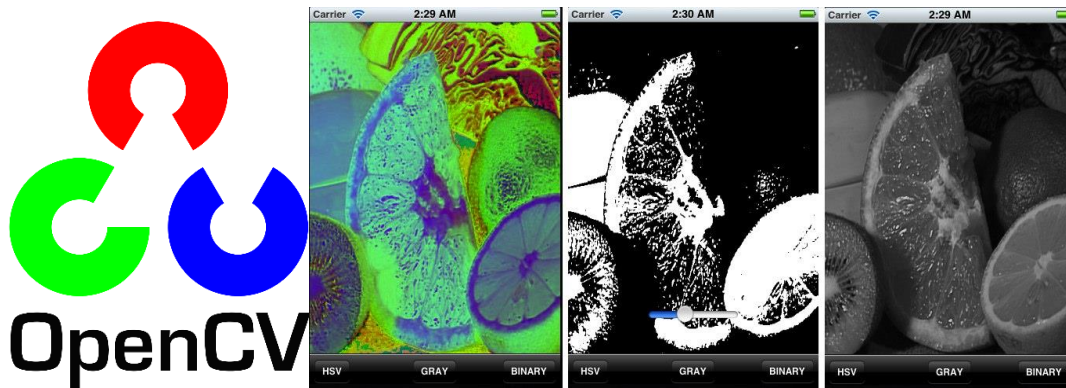


Εικόνα 2.21: Παιχνίδι σχεδιασμένο σε Pygame βασισμένο στο ευρέως διαδεδομένο arcade παιχνίδι της εποχής του '90 Wolfenstein

### 2.3.4 OpenCV

Μία βιβλιοθήκη λογισμικού ανοιχτού κώδικα υπολογιστικής όρασης και μηχανικής μάθησης που διατίθεται δωρεάν που χρησιμοποιείται ευρέως από την επιστημονική κοινότητα και όχι μόνο αποτελεί η OpenCV (Open Source Computer Vision Library) (Εικόνα 2.22). Η βιβλιοθήκη αυτή περιέχει μία πληθώρα βελτιστοποιημένων αλγορίθμων (απαριθμούν πάνω από 2500) οι οποίοι αποτελούν ένα ολοκληρωμένο, κλασικό και υπερσύγχρονο σύνολο. Μέσω αυτών των αλγορίθμων μπορεί να πραγματοποιηθούν από πιο απλές διεργασίες όπως η ανίχνευση και αναγνώριση προσώπων, η αναγνώριση αντικειμένων, εντοπισμός και ανίχνευση ανθρώπινων κινήσεων και άλλα, μέχρι σαφώς πιο περίπλοκες όπως η εξαγωγή τρισδιάστατων μοντέλων αντικειμένων, η δημιουργία τρισδιάστατων σύννεφων σημείων από στερεοφωνικές κάμερες και η συρραφή εικόνων ώστε να δημιουργηθεί μία εικόνα υψηλής ευκρίνειας και ανάλυσης. Μερικές από τις σύγχρονες εταιρίες κολοσσούς που χρησιμοποιούν έμπρακτα την συγκεκριμένη βιβλιοθήκη είναι η Google, Yahoo, Microsoft, Intel και διαφορετικές εταιρίες που ειδικεύονται σε διαφορετικούς τομείς όπως Sony, Toyota, Honda κτλ. Διαθέτει διεπαφές C++, Python, Java και Matlab γεγονός που την κάνει αυτόματα πιο ελκυστική και ευκολότερα αξιοποιήσιμη από μία ευρύτερη κλίμακα χρηστών

κάνοντάς την ένα από τα πιο δημοφιλή εργαλεία μηχανικής μάθησης με πάνω από 47 χιλιάδες ενεργούς χρήστες της επιστημονικής κοινότητας και εκτιμώμενα πάνω από 18 εκατομμύρια κατεβάσματα (downloads) ανά τον κόσμο [48].



Εικόνα 2.22: Το σήμα της OpenCV και μία εικόνα επεξεργασμένη μέσω αυτής της βιβλιοθήκης

## 2.4 Τρισδιάστατος εκτυπωτής Ender-3

Ο Ender-3 αποτελεί ένα εργαλείο ανοιχτού κώδικα τελευταίας γενιάς στον τομέα της εκτύπωσης τρισδιάστατων σχεδίων. Ο αναβαθμισμένος πίνακας ελέγχου του προσφέρει τεχνολογία αιχμής «Συνεχόμενης Εκτύπωσης» που μειώνει σημαντικά το ρίσκο για αποτυχημένες εκτυπώσεις. Είναι εξοπλισμένος με αθόρυβη σχεδίαση εκτύπωσης, με ομαλή ρυθμιζόμενη τροχαλία και στην περίπτωση διακοπής ρεύματος συνεχίζει να εκτυπώνει ακριβώς στο σημείο που άφησε το σχέδιο. Διαθέτει επίσης εύκολα προεγκατεστημένη τάση ιμάντα για την διατήρηση της απόδοσης της ακρίβειας της εκτύπωσης. Η κυκλοφορία του στην αγορά έγινε τον Φεβρουάριο του 2022 [49] (Εικόνα 2.23).



Εικόνα 2.23: Ο τρισδιάστατος εκτυπωτής Ender-3



## 2.5 Πρόγραμμα Sketchup

Το Sketchup αποτελεί μία διαισθητική εφαρμογή μοντελοποίησης τρισδιάστατων σχεδίων που επιτρέπει την επεξεργασία και δημιουργία 2D και 3D μοντέλων εφαρμόζοντας την πατενταρισμένη μέθοδο «Push and Pull». Η μέθοδος αυτή επιτρέπει ουσιαστικά στον ανέκαθεν δημιουργό να εξωθήσει οποιαδήποτε επίπεδη επιφάνεια σε τρισδιάστατο σχήμα. Το συγκεκριμένο πρόγραμμα έχει μεγάλο φάσμα εφαρμογών και χρησιμοποιείται στους κόσμους της αρχιτεκτονικής, της εσωτερικής διακόσμησης, της δημιουργίας βιντεοπαιχνιδιών. Ακόμη χρησιμοποιείται ευρέως από άτομα που θέλουν να δημιουργήσουν και να κατεβάσουν τρισδιάστατα μοντέλα προς χρήση με τρισδιάστατους εκτυπωτές [50].

## 2.6 Πρόγραμμα Cura

Το Cura είναι ένα λογισμικό ανοιχτού κώδικα και ανήκει στην κατηγορία των εφαρμογών τύπου «slicer». Το slicer αποτελεί ένα απαραίτητο εργαλείο λογισμικού σε οποιαδήποτε εργασία που αφορά τρισδιάστατη εκτύπωση. Ουσιαστικά προετοιμάζει ψηφιακά μοντέλα προς τρισδιάστατη εκτύπωση «κόβοντάς» τα σε οριζόντια στρώματα που μπορούν να εκτυπωθούν. Οργανώνει κατά κάποιο τρόπο τη μέθοδο εκτύπωσης και σχεδιάζει ένα πλάνο για τον τρισδιάστατο εκτυπωτή δοθέντος ενός τρισδιάστατου μοντέλου οποιουδήποτε μεγέθους ή σχήματος [51]. Επίσης μέσω των slicer και ειδικά του Cura τίθεται δυνατή η ρύθμιση των παραμέτρων του ίδιου του εκτυπωτή σε ότι αφορά θερμοκρασίες μύτης και επιφάνειας εκτύπωσης, ταχύτητας εκτύπωσης και άλλα.

## 2.7 Σύνοψη Κεφαλαίου

Στο παρόν κεφάλαιο εναποτέθηκαν όλες οι έννοιες και μέθοδοι που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας διπλωματικής εργασίας. Πιο συγκεκριμένα έγινε αναφορά στις έννοιες που περιγράφουν το παιχνίδι που δημιουργήθηκε καθώς και μελέτη των μηχανισμών και αλγορίθμων της μηχανικής όρασης που έπαιξαν καθοριστικό ρόλο στην αντίληψη των σχημάτων και του περιβάλλοντος υλοποίησης του project. Τέλος αναφέρθηκαν τα εργαλεία και η πλατφόρμα ανάπτυξης που έκαναν δυνατή την πραγματοποίηση της εργασίας.

## Κεφάλαιο 3 – Ανάλυση και Σχεδιασμός του Συστήματος

Σε αυτό το κεφάλαιο θα γίνει η περιγραφή και η ανάλυση της δομής του εκπαιδευτικού συστήματος και θα παρουσιαστούν τα στάδια αλλά και ο τρόπος υλοποίησης του. Αρχικά θα παρατεθούν η γενική επισκόπηση, οι προδιαγραφές που τηρήθηκαν για τη λειτουργικότητα του συστήματος καθώς και τα είδη χρηστών που καλούνται να το χρησιμοποιήσουν. Στη συνέχεια περιγράφεται η διαδικασία της υλοποίησης όλων των μερών του συστήματος μέσω της αναφοράς των βασικότερων τμημάτων του κώδικα που συγγράφηκε. Τέλος γίνεται αναφορά στο σύνολο ενεργειών πραγματοποιήθηκαν και των μέσων που χρησιμοποιήθηκαν για την κατασκευή των κύβων.

### 3.1 Γενική Επισκόπηση και Ανάλυση

Μία απαραίτητη προϋπόθεση για την κατανόηση του συστήματος και κατ' επέκταση της ανάπτυξης και υλοποίησής του είναι η ανάλυση των μερών από τα οποία απαρτίζεται. Συνεπώς το σύστημα αποτελείται από 4 επιμέρους μέρη τα οποία περιλαμβάνουν:

- τους κύβους με ενσωματωμένα σχήματα
- μια web κάμερα
- ένα πρόγραμμα αναγνώρισης σχημάτων
- ένα ηλεκτρονικό car racing παιχνίδι

Ο χρήστης καλείται μέσω της τοποθέτησης των κύβων να κατευθύνει το όχημα στην οθόνη κατά μήκος της πίστας μέχρι τη γραμμή τερματισμού.

Κατά την εκκίνηση του παιχνιδιού, και κατά συνέπεια όλης της διαδικασίας αλληλεπίδρασης με το σύστημα, ο χρήστης επιλέγει το πότε θα κινηθεί το όχημα. Αυτό πραγματοποιείται με την τοποθέτηση των κύβων σε ένα εύρος χώρου το οποίο αντιλαμβάνεται η κάμερα. Η κάμερα είναι τοποθετημένη σε ένα ύψος το οποίο καθορίζει ο εκάστοτε χρήστης με κατεύθυνση προς τα κάτω και καλύπτοντας ένα εύρος χώρου το οποίο αντιστοιχεί σε μία κόλλα χαρτί A3. Οι κύβοι με τη σειρά τους έχουν αποτυπωμένα στην επιφάνειά τους σχήματα μαύρου χρώματος τα οποία ορίζουν τις κινήσεις του αυτοκινήτου (εμπρός-πίσω-δεξιά-αριστερά) τα οποία είναι προκαθορισμένα σχετικά με την αντιστοιχία σχημάτων-κινήσεων (δηλαδή μπροστά-κύκλος, δεξιά-τετράγωνο κτλπ). Ανάλογα με την

τοποθέτησή τους από αριστερά προς τα δεξιά το σύστημα θα ερμηνεύσει τις αντίστοιχες κινήσεις.

Για την αντίληψη των σχημάτων από την κάμερα το σύστημα εφαρμόζει τεχνικές μηχανικής όρασης με εφαρμογές ψηφιακής επεξεργασίας και ανάλυσης της εικόνας. Ο χρήστης μπορεί ανά πάσα στιγμή να δει το εάν και το τι αντιλαμβάνεται η κάμερα από την οθόνη αφού κατά την εκτέλεση του προγράμματος εμφανίζεται το αντίστοιχο πλαίσιο με την εικόνα που απαθανατίζει η κάμερα. Επίσης μπορεί να καθορίσει χειροκίνητα την κατωφλίωση της εικόνας και το πλαίσιο στο οποίο θα εμπεριέχονται τα σχήματα.

### 3.2 Απαιτήσεις Συστήματος

Προκειμένου το έργο να λειτουργήσει σωστά και η όλη διαδικασία εκτέλεσής του να αποτελέσει μία ευχάριστη εμπειρία για τους μαθητές αλλά και τον εκπαιδευτικό θα πρέπει να καθοριστούν οι απαιτήσεις του. Στην παρούσα εργασία ο στόχος της είναι η δημιουργία ενός συστήματος με κύβους μέσω των οποίων οι μαθητές θα κατευθύνουν ένα εικονικό αυτοκίνητο σε μία εικονική πίστα. Στη συνέχεια, θα αναφερθούν και θα παρουσιαστούν αυτές οι απαιτήσεις και τα προαπαιτούμενα στοιχεία του έργου.

- **Απαιτείται επεξεργαστής πηγαίου κώδικα.** Εφόσον το λογισμικό (software) κομμάτι του έργου δεν διατίθεται ως κάποιου είδους εφαρμογή παρά μόνο ως πηγαίος κώδικας, θα πρέπει να υπάρχει το κατάλληλο περιβάλλον για την εκτέλεσή του. Συνεπώς είναι απαραίτητη η παρουσία κάποιου coding editor.
- **Τελευταία έκδοση Python-Pygame-OpenCV.** Για την εκτέλεση του κώδικα και τη λειτουργία του συστήματος εφαρμόζονται τεχνολογίες που αφορούν την επεξεργασία και αναγνώριση εικόνας καθώς επίσης και τη δημιουργία παιχνιδιού μέσω κώδικα. Επομένως οι βιβλιοθήκες OpenCV και Pygame που πραγματεύονται τέτοιες ευαίσθητες και πολύπλοκες τεχνολογίες, καθώς και κατ' επέκταση η γλώσσα Python μέσω της οποίας χειρίζονται, απαιτούν τελευταίες ή τουλάχιστον πρόσφατες εκδόσεις τόσο της γλώσσας προγραμματισμού όσο και των ίδιων των βιβλιοθηκών. Αυτό ισχύει προκειμένου να μην υπάρξουν τεχνικά προβλήματα και bugs που αφορούν την ορθή λήψη και επεξεργασία της εικόνας και τελικά την ορθή λειτουργία του συστήματος και του παιχνιδιού.

- **Φιλικό περιβάλλον για βέλτιστο αποτέλεσμα.** Το σύστημα της εργασίας περιλαμβάνει μία web κάμερα η οποία αποτελεί την είσοδο της εικόνας σε αυτό προκειμένου να γίνει η επεξεργασία της. Επομένως για να διευκολυνθεί ο αλγόριθμος επεξεργασίας της εικόνας απαιτείται η σωστή λήψη και εισαγωγή της στο σύστημα. Αυτό είναι εφικτό, όταν υπάρχει αρκετός και σωστός φωτισμός στο χώρο που βρίσκεται η κάμερα χωρίς να δημιουργούνται σκιές και επιπλέον θόρυβος κατά τη λήψη της εικόνας.
- **Θέση και φόντο κύβων.** Οι κύβοι της εργασίας είναι κατασκευασμένοι από χοντρό λευκό χαρτόνι και τα σχήματα πάνω τους έχουν μαύρο χρώμα. Για τη διευκόλυνση του αλγορίθμου επεξεργασίας επομένως οι κύβοι θα πρέπει να βρίσκονται σε κατάλληλη θέση ώστε να σχήματα να είναι ευδιάκριτα από την κάμερα και τοποθετημένοι σε ουδέτερου σχήματος λευκό φόντο.

### 3.3 Είδη χρηστών και περιπτώσεις χρήσης

Οι χρήστες του συστήματος θα είναι οι μαθητές και ο διδάσκων. Ο διδάσκων:

- Θα έχει το ρόλο του επιβλέποντα και διαχειριστή
- Θα είναι υπεύθυνος για την εγκατάσταση του έργου
- Θα είναι υπεύθυνος για την ορθή λειτουργία του

Καθώς οι μαθητές θα είναι οι κύριοι χρήστες και αυτοί που θα αλληλεπιδρούν άμεσα με το σύστημα, ο διδάσκων θα είναι ο αρμόδιος της ορθής διαχείρισης του υλικού μέρους του έργου. Ειδικότερα δηλαδή θα τοποθετεί και θα ελέγχει :

- Την κάμερα
- Το φόντο για την τοποθέτηση των κύβων
- Εάν το παιχνίδι λειτουργεί κανονικά

Και τέλος θα ελέγχει εάν γίνεται σωστή διαχείριση των κύβων σε ότι αφορά

- την χρήση τους
- την τοποθέτηση τους
- την αντίληψή τους από την κάμερα

ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα. Στη συνέχεια θα καλείται να αναλύσει του κανόνες και τον τρόπο λειτουργίας του συστήματος γενικότερα αλλά και του παιχνιδιού ειδικότερα καθώς και να επεξηγήσει τις ιδιότητες των σχημάτων στους κύβους.

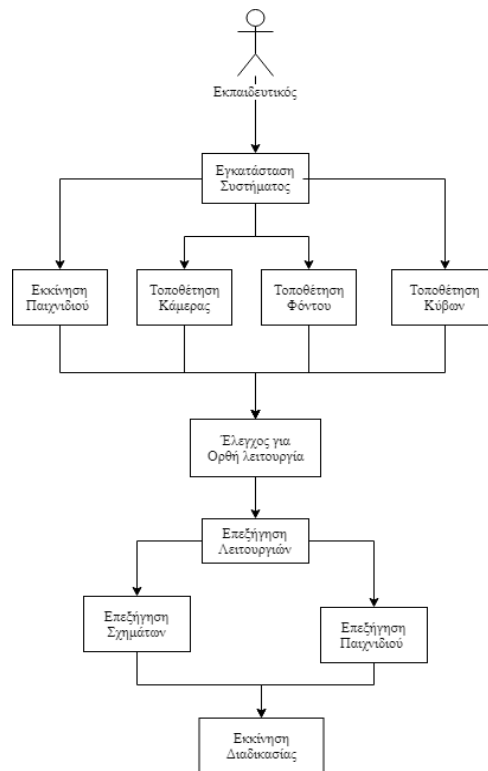
Οι μαθητές με τη σειρά τους:

- Θα καλούνται να τοποθετήσουν τους κύβους στην κατάλληλη θέση με σκοπό να κατευθύνουν το όχημα κατά μήκος της πίστας στην γραμμή τερματισμού
- Θα εναποθέτουν τους κύβους με σειρά από τα αριστερά προς τα δεξιά στο ουδέτερο φόντο λευκού χρώματος με τη σειρά που θέλουν να εκτελεστούν οι εντολές. Τα σχήματα που είναι εκτυπωμένα στις πλευρές τους θα πρέπει να είναι ευδιάκριτα, ώστε να γίνουν αντιληπτά από την κάμερα και μετέπειτα από το σύστημα κατά την επεξεργασία της εικόνας προκειμένου να ερμηνευτούν οι αντίστοιχες κινήσεις για το αυτοκίνητο, ενέργεια που θα επιβλέπει ο διδάσκων
- Θα έχουν τη δυνατότητα να βλέπουν άμεσα το αποτέλεσμα του προγραμματισμού της κίνησης του αυτοκινήτου μέσω των κύβων στην οθόνη ώστε να αποφασίσουν το πώς θέλουν να κινηθεί στη συνέχεια.

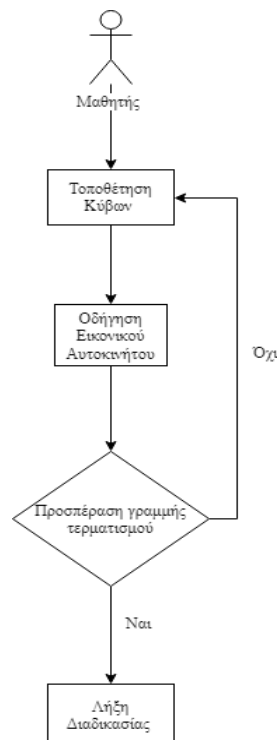
Ανάλογα με τις αποφάσεις τους θα πρέπει να επανατοποθετήσουν τους κύβους με την κατάλληλη σειρά ώστε να συνεχιστεί η πορεία του αυτοκινήτου προς τον τερματισμό. Μέσω αυτής της διαδικασίας οι μαθητές θα διδαχθούν και θα κατανοήσουν την έννοια του προγραμματισμού με απλό και δημιουργικό τρόπο.

### **3.4 Διαγράμματα Επεξήγησης**

Για μεγαλύτερη σαφήνεια και κατανόηση των ειδών των χρηστών καθώς και των ιδιοτήτων και των λειτουργιών στις οποίες εμπλέκονται, στις παρακάτω εικόνες (Εικόνα 3.1 και Εικόνα 3.2) παρουσιάζονται αναλυτικά διαγράμματα που αφορούν τον εκπαιδευτικό και τους μαθητές αντίστοιχα.



Εικόνα 3.1: Διάγραμμα Επεξήγησης λειτουργιών εκπαιδευτικού



Εικόνα 3.2: Διάγραμμα Επεξήγησης λειτουργιών μαθητή

### 3.5 Υλοποίηση Ηλεκτρονικού Παιχνιδιού

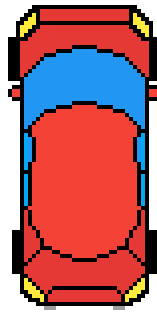
Σε αυτή την ενότητα περιγράφεται η διαδικασία υλοποίησης του ηλεκτρονικού παιχνιδιού. Καθώς το παιχνίδι αποτελεί ένα μέσο με το οποίο θα αλληλεπιδρούν άμεσα οι μαθητές, στην παρούσα διπλωματική εργασία αποφασίστηκε η δημιουργία ενός τύπου παιχνιδιού που να είναι οικείος με τα παιδιά της ηλικίας που απευθύνεται η εργασία, αλλά και αντίστοιχο της γενιάς που αντιπροσωπεύουν. Επομένως λήφθηκε η απόφαση δημιουργίας ενός ηλεκτρονικού παιχνιδιού, σαφώς πιο κοντινός με τα ενδιαφέροντα των μαθητών αυτής της γενιάς, αλλά και πιο ελκυστικό όντας ένα παιχνίδι εκπαιδευτικού χαρακτήρα. Ειδικότερα επιλέχθηκε η δημιουργία ενός car racing game δεδομένου ότι αυτού του τύπου παιχνίδια είναι ήδη οικεία σε παιδιά μικρών ηλικιών σε ότι αφορά τους κανόνες και το στόχο τους για την επίτευξη της νίκης. Επίσης λήφθηκε υπόψη, ότι το παιχνίδι μέσω των οπτικών του ερεθισμάτων, αλλά και γενικότερα ως δραστηριότητα θα βοηθήσει στην βαθιά κατανόηση και αφομοίωση των πληροφοριών που θα δεχθούν οι μαθητές. Άρα και αυτόματα το καθιστά και ένα από τα καταλληλότερα μέσα μετάδοσης της πληροφορίας ειδικά σε μικρές ηλικίες. Για τους παραπάνω λόγους το παιχνίδι επιλέχθηκε να αναπτυχθεί σε γλώσσα python και ειδικότερα μέσω της βιβλιοθήκης pygame.

#### 3.5.1 Ρύθμιση περιβάλλοντος

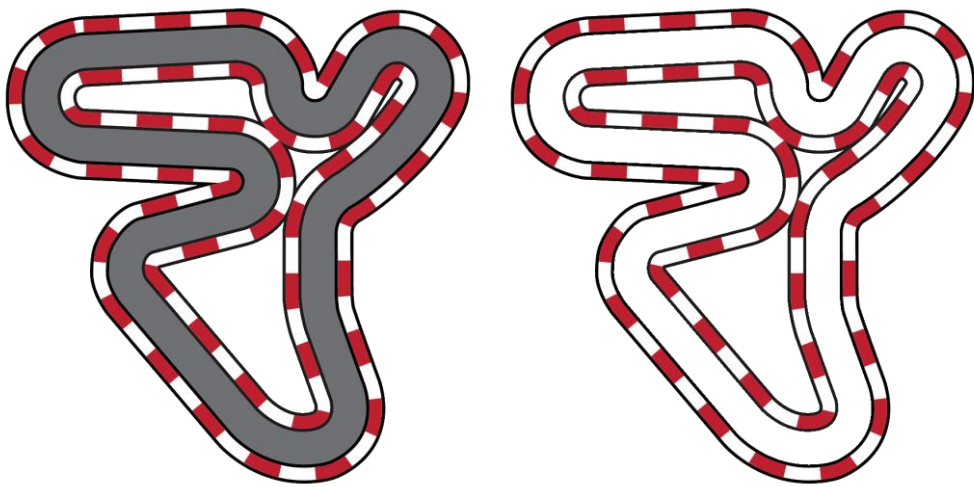
Ξεκινώντας τη δημιουργία του παιχνιδιού βασικό παράγοντα αποτελεί η δημιουργία του εικονικού περιβάλλοντος και των οπτικών στοιχείων από τα οποία θα απαρτίζεται. Τα στοιχεία αυτά αποτελούν ουσιαστικά ψηφιακές εικόνες σε μορφή αρχείων .jpg και .png οι οποίες είτε δημιουργήθηκαν είτε αντλήθηκαν από το διαδίκτυο. Πιο συγκεκριμένα το σύνολο των εικόνων αποτελείται από:

- Μία εικόνα ενός κόκκινου αυτοκινήτου
- Μία εικόνα του σχήματος του δρόμου της πίστας
- Μία εικόνα του σχήματος των ορίων-συνόρων της πίστας
- Μία εικόνα της γραμμής τερματισμού
- Μία εικόνα του γρασιδιού που περιβάλλει την πίστα

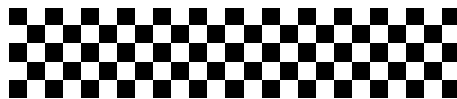
Μικρογραφίες των οποίων παρουσιάζονται στις εικόνες 3.3, 3.4, 3.5 και 3.6 αντίστοιχα.



Εικόνα 3.3: Το κόκκινο αυτοκίνητο του παιχνιδιού



Εικόνα 3.4: Το σχήμα του δρόμου και των ορίων της πίστας



Εικόνα 3.5: Η γραμμή τερματισμού

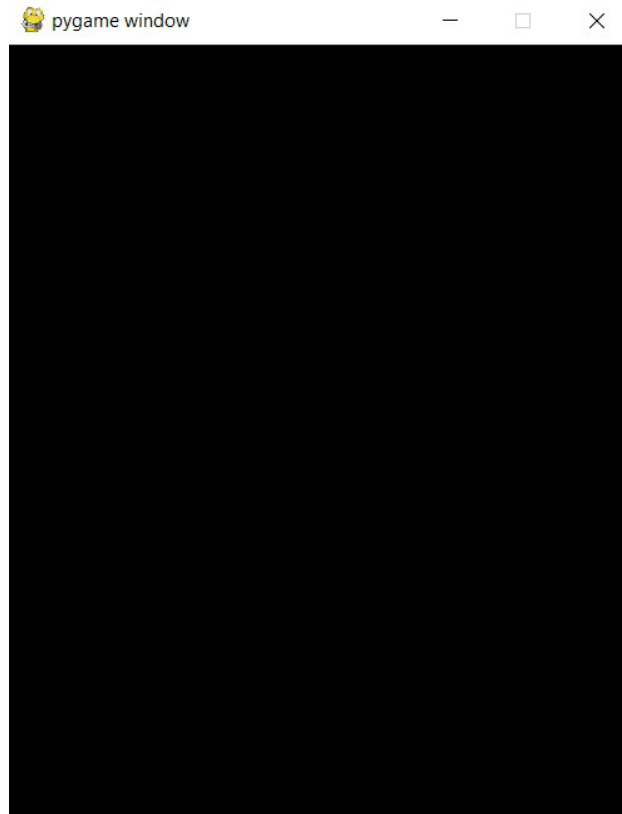




*Εικόνα 3.6: Το γρασίδι της πίστας*

Μετά τον καθορισμό και την επιλογή των εικόνων που θα χρησιμοποιηθούν είναι απαραίτητη η εισαγωγή (Εικόνα 3.8) και η τοποθέτησή τους. Προκειμένου να επιτευχθεί αυτό όμως, χρειάζεται πρώτα η δημιουργία ενός παραθύρου και πιο συγκεκριμένα μίας επιφάνειας προβολής και αναπαραγωγής των εικόνων. Μέσω αυτού του παραθύρου θα γίνονται αντιληπτές όλες οι μετέπειτα κινήσεις του αυτοκινήτου και οι μεταβολές στο περιβάλλον του παιχνιδιού. Μετά τη δημιουργία του παραθύρου (Εικόνα 3.7) γίνεται η τοποθέτηση των εικόνων, αφού τροποποιηθούν ως προς το μέγεθός τους για να εξυπηρετήσουν τις ανάγκες του παιχνιδιού. Ο καθορισμός της τοποθέτησης γίνεται μέσω των συντεταγμένων  $x$  και  $y$ . Επειδή όμως στην pygame το σύστημα συντεταγμένων ξεκινά στο πάνω αριστερά μέρος του παραθύρου οι εικόνες πρέπει να εναποτίθενται με βάση αυτή τη διαφοροποίηση. Οι εικόνες τοποθετούνται με την κατάλληλη σειρά η μία μετά την άλλη ώστε να γίνονται διακριτά τα απαραίτητα χαρακτηριστικά τους καθώς εάν αυτή η διαδικασία γινόταν αυθαίρετα θα συνέπιπταν μεταξύ τους με αποτέλεσμα να είχαμε λανθασμένο αποτέλεσμα. Συγκεκριμένα η εικόνες τοποθετούνται με την εξής σειρά:

1. Γρασίδι
2. Δρόμος
3. Όρια δρόμου
4. Αυτοκίνητο
5. Γραμμή τερματισμού

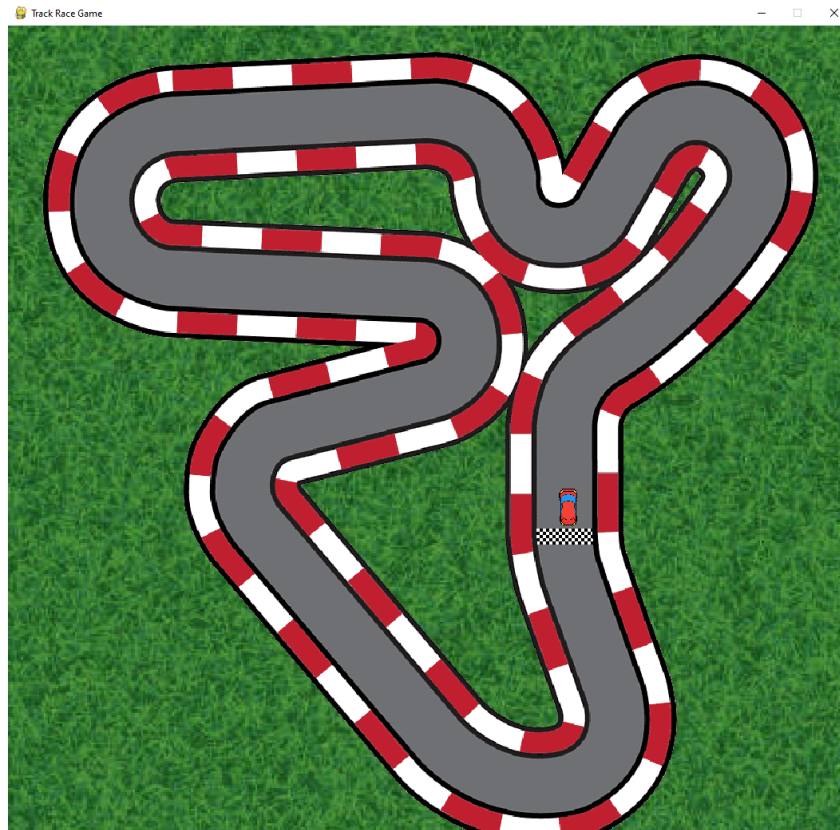


Εικόνα 3.7: Παράδειγμα κενού παραθύρου pygame

```
12 GRASS= scale_image(pygame.image.load('grass.jpg'), 2.6)
13 TRACK= scale_image(pygame.image.load('track2.png'), 1.15)
14 TRACK_BORDER= scale_image(pygame.image.load('track-border2.png'), 1.15)
15 TRACK_BORDER_MASK=pygame.mask.from_surface(TRACK_BORDER)
16 START_POS=(690,580)
17 CAR= scale_image(pygame.image.load('red-car.png'),0.6)
18 FINISH= pygame.image.load('finish.png')
```

Εικόνα 3.8: Κώδικας με την εισαγωγή και αλλαγή μεγέθους των εικόνων

Αξίζει να σημειωθεί ότι κατά την τοποθέτηση των εικόνων σε κάθε στάδιο, όπως προαναφέρθηκε, γίνεται και η αλλαγή του μεγέθους τους και συγκεκριμένα η μεγέθυνση ή σμίκρυνση τους. Μετά τις απαραίτητες τοποθετήσεις και αναδιαμορφώσεις λοιπόν έχουμε και το τελικό αποτέλεσμα του παραθύρου άρα και την τελική διεπαφή του παιχνιδιού. (Εικόνα 3.9)



Εικόνα 3.9: Τελικό αποτέλεσμα παραθύρου αναπαραγωγής

### 3.5.2 Κίνηση Αυτοκινήτου

Αφού πραγματοποιήθηκε ο η διαδικασία της ρύθμισης του περιβάλλοντος του παιχνιδιού σειρά έχει η υλοποίηση της βασικής λειτουργίας του, δηλαδή η κίνηση της εικόνας του αμαξιού. Η κίνηση του αυτοκινήτου καθορίζεται τόσο ως προς τον εαυτό του, με την περιστροφή του γύρω από έναν νοητό άξονα που διαπερνά το κέντρο της εικόνας του, όσο και μέσω της κίνησης της εικόνας κατά μήκος της πίστας προς τη γραμμή τερματισμού. Αυτό αποτελεί μία πολύπλοκη διεργασία που περιλαμβάνει την εφαρμογή πολλών παραμέτρων και μεθόδων της pygame σε συνδυασμό με βασικές γνώσεις μαθηματικών. Για την διαχείριση όλων αυτών των στοιχείων που χρειάζονται για την λειτουργία της κίνησης και περιστροφής του αυτοκινήτου δημιουργήθηκε η κλάση `AbstractCar` απόσπασμα της οποίας απεικονίζεται παρακάτω (Εικόνα 3.10).

```

89 class AbstractCar:      #image rotation
90     def __init__(self, max_vel,rotation_vel):
91         self.img=self.IMG
92         self.max_vel= max_vel
93         self.vel=0
94         #self.rotation_vel = rotation_vel
95         self.angle=0
96         self.x,self.y=self.START_POS
97         self.acceleration=1
98
99     def rotate(self,left=False,right=False):
100        if left:
101            self.angle -= ROTATION_VEL
102        elif right:
103            self.angle += ROTATION_VEL
104
105    def draw (self, win):
106        blit_rotate_center(win, self.img,(self.x,self.y), self.angle)
107
108    def move_forward(self):
109        self.vel = min(self.vel+self.acceleration,self.max_vel)
110        self.move()
111
112    def move_backward(self):
113        self.vel = max(self.vel-self.acceleration,-self.max_vel/2)
114        self.move()
115
116
117    def move(self):
118        radians = math.radians(self.angle)
119        vertical = math.cos(radians) * self.vel
120        horizontal= math.sin(radians) * self.vel
121
122        self.y -= vertical
123        self.x -= horizontal
124

```

Εικόνα 3.10: Απόσπασμα του κώδικα της κλάσης AbstractCar

### 3.5.2.1 Περιστροφή Αυτοκινήτου

Μετά την τοποθέτηση της εικόνας του αυτοκινήτου σε συγκεκριμένο σημείο του παραθύρου αναπαραγωγής, που καθορίστηκε με βάση τη μέθοδο της pygame για τους άξονες x,y όπως προαναφέρθηκε, σειρά έχει ο καθορισμός και η υλοποίηση της διαδικασίας περιστροφής του στον εικονικό χώρο. Για να πραγματοποιηθεί αυτή η διαδικασία δημιουργήθηκε η συνάρτηση blit\_rotate\_center που διαχειρίζεται την εικόνα με βάση μία γωνία (Εικόνα 3.11).

```

9  def blit_rotate_center(win, image, top_left, angle):
10     rotated_image = pygame.transform.rotate(image, angle)
11     new_rect=rotated_image.get_rect(center=image.get_rect(topleft=top_left).center)
12     win.blit(rotated_image, new_rect.topleft)

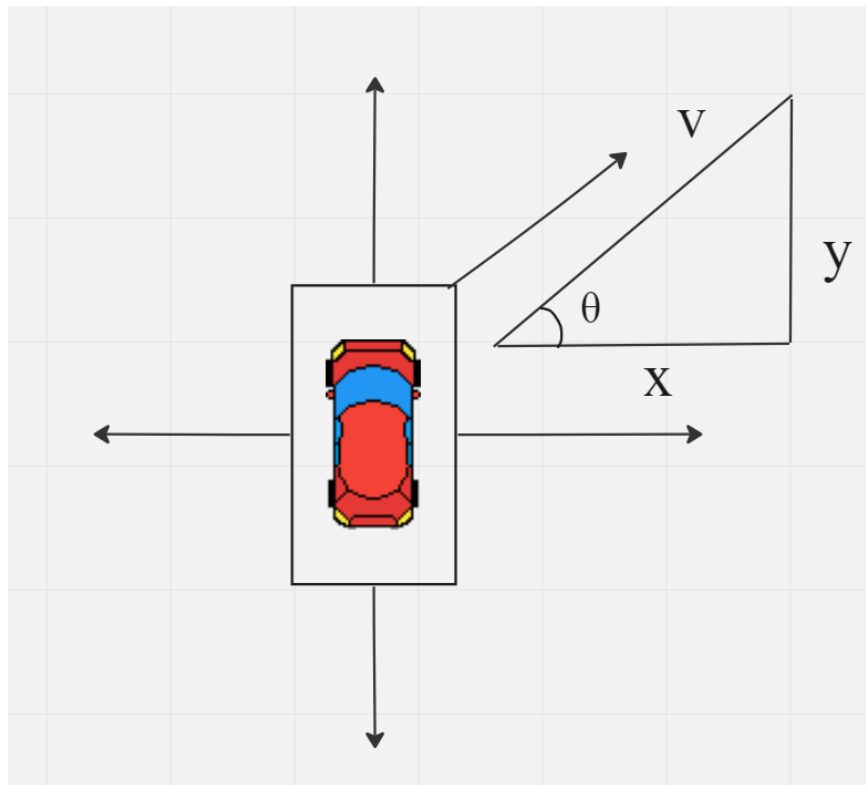
```

Εικόνα 3.11: Κώδικας της συνάρτησης `blit_rotate_center`

Σε αυτό το σημείο αξίζει να αναφερθεί ότι όλες οι εικόνες στην pygame αποτελούν ένα τετράγωνο στο οποίο εμπεριέχεται στην ουσία η εικόνα. Οπότε εάν η εικόνα περιστραφεί, περιστρέφεται το τετράγωνο και κατ' επέκταση η εικόνα που εμπεριέχεται σε αυτό. Επομένως για να αποφευχθούν τυχόν θόρυβοι και παρεμβολές δημιουργήθηκε και εφαρμόστηκε η παραπάνω συνάρτηση κατά την οποία περιστρέφεται το κέντρο της εικόνας παρά η πάνω αριστερά γωνία της όπως θα γινόταν σύμφωνα με μία απλή εφαρμογή έτοιμης συνάρτησης περιστροφής μιας εικόνας από την pygame .

### 3.5.2.2 Κίνηση υπό γωνία

Η κίνηση της εικόνας αυτή καθ' εαυτή κατά μήκος ή κατά πλάτος του παραθύρου προβολής αποτελεί μία σχετικά απλή διαδικασία για την pygame. Η κίνηση του αυτοκινήτου όμως στις διαστάσεις του παραθύρου προβολής, με το μπροστινό μέρος του να κοιτά στο σημείο που θα θέλει ο καθοδηγητής του να στρίψει και υπό τη γωνία που επιθυμεί αποτελεί αντιθέτως μία αρκετά περίπλοκη διαδικασία. Για τις ανάγκες της συγκεκριμένης κίνησης χρησιμοποιήθηκαν βασικές γνώσεις τριγωνομετρίας καθώς και σχετικές συναρτήσεις της (συνημίτονο και ημίτονο) οι οποίες αποτυπώθηκαν με τη μορφή κώδικα και εφαρμόστηκαν στη συνάρτηση κίνησης του αυτοκινήτου (Εικόνα 3.13). Στο παρακάτω γράφημα (Εικόνα 3.12) παρουσιάζεται η γενική ιδέα της μελέτης της κίνησης του αυτοκινήτου υπό γωνία. Στο συγκεκριμένο παράδειγμα απεικονίζονται με βέλη οι πιθανές πορείες του αυτοκινήτου με το διαγώνιο βέλος να αναπαριστά την επιθυμητή πορεία. Η θέση λοιπόν που γυρίζει η μπροστινή όψη του αυτοκινήτου ως προς τους άξονες x και y υπολογίζεται με το συνδυασμό του αποτελέσματος του πολλαπλασιασμού του ημιτόνου της γνωστής γωνίας  $\theta$  με το γνωστό  $V$  (που αντιστοιχεί στη λέξη *velocity* που μεταφράζεται ως ταχύτητα), με του αντίστοιχου αποτελέσματος του πολλαπλασιασμού του συνημίτονου της γωνίας  $\theta$  με το  $V$ .



Εικόνα 3.12: Γράφημα παραδείγματος υπολογισμού κίνησης του αυτοκινήτου υπό γωνία

```

116
117 def move(self):
118     radians = math.radians(self.angle)
119     vertical = math.cos(radians) * self.vel
120     horizontal = math.sin(radians) * self.vel
121
122     self.y -= vertical
123     self.x -= horizontal
124

```

Εικόνα 3.13: Η Συνάρτηση της κίνησης του αυτοκινήτου σε κώδικα

### 3.5.3 Σύγκρουση με τα Σύνορα της Πίστας

Η οδηγική πίστα σε έναν αγώνα αυτοκινήτων σαν έννοια καθορίζεται ως ο δρόμος στον οποίο κινούνται τα οχήματα που αγωνίζονται. Προκειμένου λοιπόν να υφίσταται αυτή η έννοια και στο εν λόγω ηλεκτρονικό παιχνίδι, κρίθηκε απαραίτητος ο καθορισμός των συνόρων της πίστας ώστε το αυτοκίνητο να μην κινείται ανεξέλεγκτα στον εικονικό χώρο παρά μόνο στο πλάτος του δρόμου και εντός των ορίων αυτών των συνόρων αυτών. Για την επίτευξη του στόχου αυτού αρχικά χρησιμοποιήθηκε η εικόνα των ορίων που αναφέρθηκε σε

προηγούμενη ενότητα, όπου αναγράφονται μόνο τα διακριτικά των συνόρων της πίστας περιμετρικά του δρόμου. Με αυτή την εικόνα θα έπρεπε να συγκρουστεί το όχημα ώστε να μην εξέρχεται των συνόρων της πίστας. Η υλοποίηση της σύγκρουσης στο εικονικό περιβάλλον της pygame έγινε μέσω της χρήσης των pygame μασκών και την εφαρμογή της διαδικασίας της τέλει σύγκρουσης των εικονοστοιχείων.

Κατά τη χρήση της pygame μάσκας μέσω της συνάρτησης `pygame.mask` (Εικόνα 3.13) δημιουργείται ένα αντικείμενο μάσκας από τη δεδομένη επιφάνεια, στη συγκεκριμένη περίπτωση των ορίων της πίστας, ορίζοντας όλα τα αδιαφανή εικονοστοιχεία και όχι τα διαφανή. Όπως έχει προαναφερθεί η εικόνα στην pygame ορίζεται ως ένα εικονικό τετράγωνο άρα οι όροι αδιαφανή και διαφανή περιγράφουν τα εικονοστοιχεία που φαίνονται και δεν φαίνονται από το μάτι αντίστοιχα. Επομένως εάν ένα κλειδί χρώματος χρησιμοποιείται από την επιφάνεια τότε το ίδιο το κλειδί χρησιμοποιείται για να αποφασιστεί από τη συνάρτηση ποιο κομμάτι από τη μάσκα που δημιουργείται θα οριστεί. Όλα τα εικονοστοιχεία που δεν είναι ίσα με το κλειδί χρώματος του ορίου της πίστας ορίζονται και αντιθέτως τα εικονοστοιχεία που είναι ίσα δεν ορίζονται. Η μάσκα δημιουργήθηκε τόσο για τα όρια της πίστας όσο και για το αυτοκίνητο.

```

14 TRACK_BORDER= scale_image(pygame.image.load('track-border2.png'), 1.15)
15 TRACK_BORDER_MASK=pygame.mask.from_surface(TRACK_BORDER)

298     if car.collision(TRACK_BORDER_MASK) != None:
299         car.bounceback()
    
```

Εικόνα 3.14: Δημιουργία μάσκας στην εικόνα με τα όρια της πίστας (γραμμές 14,15) και αλληλεπίδραση με αυτή (γραμμές 298,299)

Λαμβάνοντας υπόψη λοιπόν όλα τα παραπάνω η σύγκρουση ορίστηκε με βάση το άγγιγμα των δύο μασκών δηλαδή την επαφή των έγχρωμων εικονοστοιχείων των εικόνων. Κατά τη σύγκρουση ορίστηκε να γίνεται μία μικρή αναπήδηση του αυτοκινήτου προς τα πίσω ώστε να γίνεται αντιληπτή η επαφή από τους χρήστες και ευκολότερη η επαναπροσαρμογή της θέσης του. Με αυτό τον τρόπο διευκολύνεται η οδήγηση του αμαξιού και φυσικά δεν επιτρέπεται η έξοδος του από τα σύνορα της πίστας.

### 3.5.4 Αλληλεπίδραση με τη γραμμή τερματισμού

Αφού το ηλεκτρονικό παιχνίδι που δημιουργήθηκε αφορά έναν εικονικό αγώνα, εξ' ορισμού τέθηκε απαραίτητος ο προσδιορισμός και η δημιουργία μιας γραμμής εκκίνησης και μιας τερματισμού. Προκειμένου να τερματίσει τη διαδικασία του αγώνα ο χρήστης επομένως, θα πρέπει το αυτοκίνητο να διασχίσει την προαναφερθείσα εικονική γραμμή τερματισμού. Η εικόνα της γραμμής τερματισμού επομένως τοποθετήθηκε στοχευμένα σε συγκεκριμένο σημείο πίσω από την αρχική θέση της εικόνας του αυτοκινήτου.

Η αλληλεπίδραση του αυτοκινήτου με τη γραμμή τερματισμού σχεδιάστηκε να διαφέρει ανάλογα τον τρόπο με τον οποίο διαπερνά την τελευταία το αυτοκίνητο. Δηλαδή εάν τη διασχίσει κανονικά με το μπροστινό μέρος του να πραγματοποιεί πρώτο την επαφή, το αυτοκίνητο επαναφέρεται στην αρχική του θέση και εμφανίζεται μήνυμα νίκης (Εικόνα 3.14).

Σε μία διαφορετική περίπτωση όμως που το όχημα προγραμματίστηκε να κινηθεί αντίθετα από τη φυσιολογική πορεία που θα έπρεπε θεωρητικά να ακολουθήσει και τείνει να περάσει τη γραμμή τερματισμού με διαφορετικό τρόπο από τον καθορισμένο, το αυτοκίνητο έχει σχεδιαστεί να κάνει μία μικρή αναπήδηση από τη γραμμή παρόμοια με εκείνη της σύγκρουσης με τα όρια των συνόρων της πίστας (Εικόνα 3.14). Με αυτό τον τρόπο ο χρήστης περιορίζεται στο να κινηθεί μόνο προς τα εμπρός χωρίς να τερματίζεται πρόωρα το παιχνίδι. Λόγω του γεγονότος ότι το σύστημα απευθύνεται σε μικρά παιδιά, με αυτή την τεχνική κατά την αλληλεπίδραση του οχήματος με τη γραμμή τερματισμού μπορούν να αποφευχθούν τεχνικά λάθη από μέρους των μαθητών που θα οδηγούσαν ενδεχομένως σε σύγχυση.

## 3.6 Εντοπισμός και Καθορισμός σχημάτων

Ο εντοπισμός και ο καθορισμός των σχημάτων αποτελεί έναν από τους πυρήνες της διαδικασίας της υλοποίησης του συστήματος στο σύνολό του. Με βάση αυτή τη λειτουργία γίνεται δυνατή η οδήγηση του οχήματος μετέπειτα στο εικονικό περιβάλλον, γεγονός που την καθιστά το συνδεδετικό κρίκο ανάμεσα στο χρήστη και το παιχνίδι. Η υλοποίηση αυτής της διαδικασίας πραγματοποιήθηκε με τη χρήση μιας Web κάμερας και την αξιοποίηση των λειτουργιών της βιβλιοθήκης OpenCV μέσω της γλώσσας προγραμματισμού Python.

### 3.6.1 Ρύθμιση Κάμερας

Σε πρώτο στάδιο απαραίτητη είναι η είσοδος και η διαχείριση της κάμερας σε λογισμικό επίπεδο διαμέσου κώδικα. Η OpenCV διαθέτει πολλές λειτουργίες που αφορούν την κάμερα



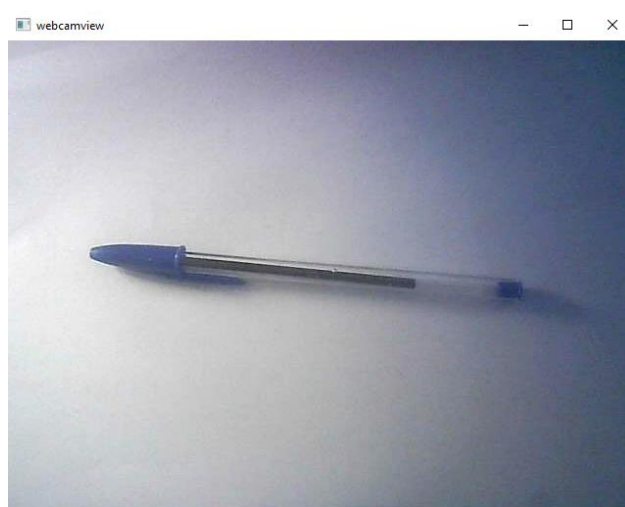
όπως το άνοιγμα και κλείσιμο της, την εγγραφή βίντεο, τη λήψη φωτογραφιών και άλλα. Για τις ανάγκες του συγκεκριμένου έργου η κάμερα θα έπρεπε να παραμένει συνέχεια ανοιχτή παράλληλα με την λειτουργία του παιχνιδιού ώστε να επιτρέπεται στο χρήστη να εναλλάσσει κύβους με σχήματα ανά πάσα στιγμή χωρίς να διακόπτεται συνεχώς η διαδικασία προκειμένου να την επανενεργοποιήσει. Επομένως αρχικά ενεργοποιήθηκε η πρόσβαση στην κάμερα και η αρχικοποίηση των ρυθμίσεών της, και στη συνέχεια η συνεχόμενη προβολή της εικόνας (Εικόνα 3.17) «διαβάζοντας» κάθε πλαίσιό της (Εικόνα 3.15 και Εικόνα 3.16).

```
30
31  fWidth=640
32  fHeight=480
33  cap = cv2.VideoCapture(0)
34
35  cap.set(3, fWidth)
36  cap.set(4, fHeight)
37
```

Εικόνα 3.15: Πρόσβαση και λήψη βίντεο διαμέσου της συνάρτησης `cv2.VideoCapture` και αρχικοποίηση ρυθμίσεων κάμερας

```
200  while run:
201      success, img= cap.read()
202      cv2.imshow(" ", imgStack)
```

Εικόνα 3.16: Προβολή της εικόνας με τη συνάρτηση `cv2.imshow()`



Εικόνα 3.17: Παράδειγμα παραθύρου βίντεο με ζωντανή εικόνα μετά την εφαρμογή των παραπάνω εντολών

### 3.6.2 Φιλτράρισμα της εικόνας

Βασική προϋπόθεση για τον εντοπισμό των σχημάτων είναι το φιλτράρισμα της εικόνας. Με τον όρο φιλτράρισμα εννοούμε την εφαρμογή διαδοχικών μεθόδων τροποποίησης και επεξεργασίας της εικόνας με σκοπό τη διευκόλυνση της διαδικασίας αναγνώρισης και εντοπισμού των σχημάτων στη συνέχεια. Σε αυτή την ενότητα θα περιγραφεί ο τρόπος και η σειρά εφαρμογής των προαναφερθέντων φίλτρων.

Η εφαρμογή των φίλτρων έγινε βηματικά και με συγκεκριμένη σειρά. Αυτό είχε μεγάλη σημασία καθώς τα φίλτρα έπρεπε να εφαρμοστούν διαδοχικά το ένα μετά το άλλο με τις αντίστοιχες συναρτήσεις τους να έχουν ως χαρακτηριστικά τους, την έξοδο της εικόνας από το προηγούμενο φίλτρο. Συγκεκριμένα η εφαρμογή των φίλτρων έγινε με την εξής σειρά (Εικόνα 3.18):

1. Γκαουσιανό Θόλωμα
2. Μετατροπή Χρωματικού Μοντέλου (Grayscale)
3. Ανίχνευση Ακρων Canny
4. Αναστολή της εικόνας

```
216     imgBlur = cv2.GaussianBlur(img,(7,7),1)
217     imgGray = cv2.cvtColor(imgBlur, cv2.COLOR_BGR2GRAY)
218
219
220     threshold1 = cv2.getTrackbarPos("Threshold1","Parameters")
221     threshold2 = cv2.getTrackbarPos("Threshold2","Parameters")
222
223     imgCanny = cv2.Canny(imgGray,threshold1,threshold2)
224
225     kernel=np.ones((5,5))
226
227     imgDil = cv2.dilate(imgCanny,kernel, iterations=1)
```

Εικόνα 3.18: Εφαρμογή των φίλτρων Γκαουσιανό Θόλωμα, Grayscale, Ανίχνευση Γραμμών Canny και Αναστολής της εικόνας στις γραμμές 216, 217, 223, 227 του κώδικα αντίστοιχα

Με τη χρήση των φίλτρων γίνεται άμεση υποβοήθηση των αλγορίθμων της Κατωφλίωσης και της Εύρεσης Περιγραμμάτων που είναι υπεύθυνα για τον καθορισμό της έντασης του θορύβου και την εύρεση των περιγραμμάτων του σχήματος προς αναγνώριση αντίστοιχα.

### 3.6.3 Εφαρμογή Κατωφλίωσης

Κατά την εφαρμογή του φίλτρου για την ανίχνευση γραμμών Canny είναι απαραίτητη η εισαγωγή δύο τιμών κατωφλίωσης. Η διαδικασία της κατωφλίωσης στο συγκεκριμένο έργο κατέχει ιδιαίζουσα σημασία αφού μέσω της εφαρμογής αυτής της μεθόδου:

- καθορίστηκε το προσκήνιο από το φόντο της εικόνας
- περιορίστηκε ο θόρυβος της εικόνας
- έγινε δυνατή η διάκριση και ο διαχωρισμός των στοιχείων της εικόνας

Η εισαγωγή των τιμών της κατωφλίωσης σχεδιάστηκε να πραγματοποιείται μέσω δύο μπαρών με αναγραφόμενα νούμερα (Εικόνα 3.19) τα οποία καθορίζει ο χρήστης σύμφωνα με τις συνθήκες του περιβάλλοντος που εκτελείται το σύστημα. Λόγω του γεγονότος ότι η κάμερα μπορεί να βρίσκεται σε αλλότροπα περιβάλλοντα (ενδεχομένως σε διαφορετικά δωμάτια), άρα και όλη η διαδικασία της εφαρμογής της κατωφλίωσης να γίνεται υπό διαφορετικές συνθήκες, μέσω των μπαρών που προαναφέρθηκαν ο χρήστης θα μπορεί να προσαρμόσει τις τιμές της κατωφλίωσης χειροκίνητα σύμφωνα με τις δικές του προτιμήσεις. Με αυτό τον τρόπο, καθιστάται ικανός να έχει τον έλεγχο της ποιότητας της διαδικασίας της εφαρμογής της κατωφλίωσης άρα και κατ' επέκταση της ικανότητας αναγνώρισης του σχήματος. Με την μεταβολή των τιμών των μπαρών ο χρήστης θα μπορεί να διακρίνει άμεσα το αποτέλεσμα της αλλαγής στην εικόνα του ζωντανού βίντεο από την κάμερα.



Εικόνα 3.19: Οι μπάρες με αναγραφόμενες τις τιμές κατωφλίωσης η εισαγωγή των οποίων έγινε στις γραμμές 220 και 221 του κώδικα στην Εικόνα 3.18

### 3.6.4 Εφαρμογή εύρεσης περιγραμμάτων (Contouring Application)

Όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο τα περιγράμματα αποτελούν τα όρια ενός σχήματος με την ίδια ένταση. Η εύρεση των περιγραμμάτων (contouring) λοιπόν, αποτέλεσε επίσης μια άκρως απαραίτητη διαδικασία, από την οποία θα κρινόταν ο εντοπισμός και μετέπειτα ο καθορισμός των σχημάτων σε άμεσο βαθμό

Η εφαρμογή της εύρεσης έγινε με τη δημιουργία της συνάρτησης `getContours` (Εικόνα 3.20) και ειδικότερα μέσω της συνάρτησης της βιβλιοθήκης OpenCV, `cv2.findContours`. Στη δεύτερη ως παράμετροι χρησιμοποιήθηκαν δύο πολύ βασικές με τη σειρά τους μέθοδοι, η `cv2.RETR_EXTERNAL` και η `cv2.CHAIN_APPROX_NONE`. Με τη χρήση της `RETR_EXTERNAL` από το αγγλικό `retrieve external` ανακτήθηκε μόνο το ακραίο εξωτερικό περίγραμμα του σχήματος της εικόνας και ακολούθως με την `CHAIN_APPROX_NONE` αποθηκεύτηκαν όλα τα οριακά σημεία του σχήματος που εντοπίστηκε. Ο συνδυασμός αυτών των δύο μεθόδων ανήκει στην ευρύτερη κατηγορία της μεθόδου προσέγγισης του περιγράμματος, η οποία καθορίστηκε με την επιλογή της χρήσης των δύο παραπάνω συναρτήσεων για τις ανάγκες της εργασίας.

```

50 def getContours(img, imgContour):
51
52     contours, hierarchy= cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
53
54     shapeList = []
55     shapeListPos = []
56
57     for cnt in contours:
58         area=cv2.contourArea(cnt)
59         areamin=cv2.getTrackbarPos("Area", "Parameters")
60         if area > areamin:
61             cv2.drawContours(imgContour, cnt, -1, (255,0,255), 7)
62             peri=cv2.arcLength(cnt, True)
63             approx = cv2.approxPolyDP(cnt, 0.02*peri, True)
64             #print(len(approx))
65             x,y,w,h=cv2.boundingRect(approx)
66
67             if len(approx)==3:
68                 cv2.putText(imgContour, "Triangle", (x+w+20,y+45), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 255, 0), 2)
69                 shapeList.append("Triangle")
70                 shapeListPos.append(x)

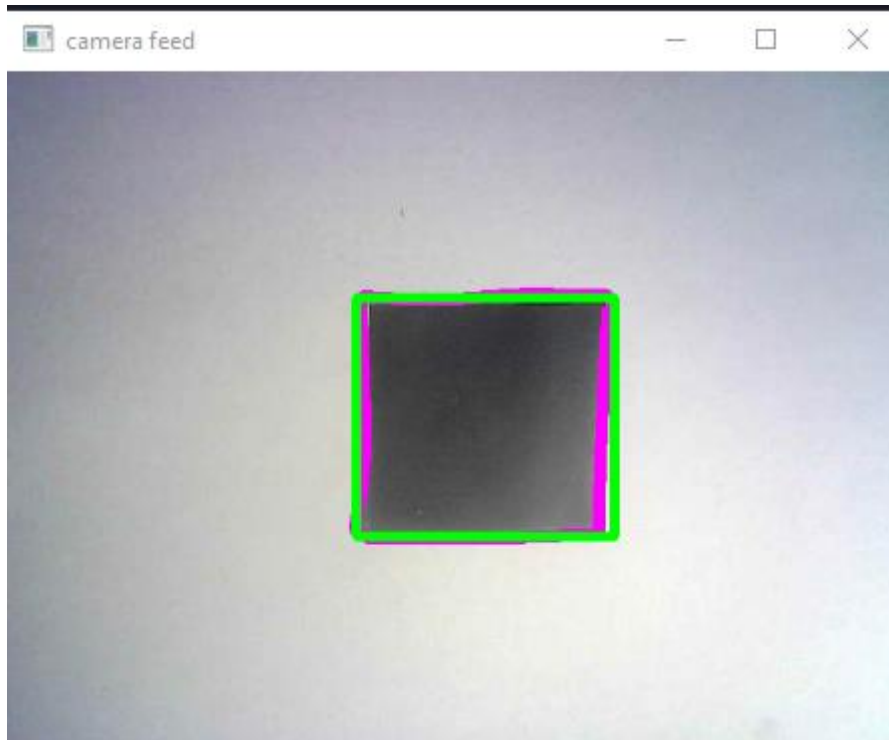
```

Εικόνα 3.20: Απόσπασμα του κώδικα της συνάρτησης `getContours`

Επίσης χρησιμοποιήθηκε η συνάρτηση `cv2.drawContours` της OpenCV η οποία τονίζει και κατά κάποιο τρόπο «ζωγραφίζει» το ευρεθέν περίγραμμα του σχήματος. Αυτό έγινε προκειμένου να επιτρέπεται ανά πάσα στιγμή στο χρήστη να αντιληφθεί εάν το ίδιο το σύστημα αντιλήφθηκε και βρήκε το περίγραμμα του σχήματος που εισήχθη.

Κατά τον εντοπισμό των περιγραμμάτων το σύστημα ενδέχεται να συμπεριλάμβανε και περιγράμματα που δεν χρειάζονταν πέρα του σχήματος που τοποθετήθηκε στην κάμερα λόγω θορύβου της εικόνας και ανωμαλιών του άσπρου φόντου εναπόθεσης του σχήματος. Για την αποφυγή του θορύβου και την ακόμη πιο ακριβή εύρεση ενός περιγράμματος χρησιμοποιήθηκε η συνάρτηση `cv2.contourArea` της OpenCV, με την οποία καθορίστηκε μέσω συγκεκριμένης κλίμακας μία περιοχή μόνο μέσα στην οποία θα επιτρέπεται να

σχεδιαστεί το περίγραμμα. Επομένως όλος ο θόρυβος της εικόνας εκτός του σχήματος που καλείται να γίνει εύρεση περιγράμματος αφαιρείται και δεν αναπαράγεται στη ζωντανή εικόνα της κάμερας. Το τελικό αποτέλεσμα της εύρεσης του περιγράμματος και απεικόνισης του φαίνεται στην παρακάτω εικόνα (Εικόνα 3.20).



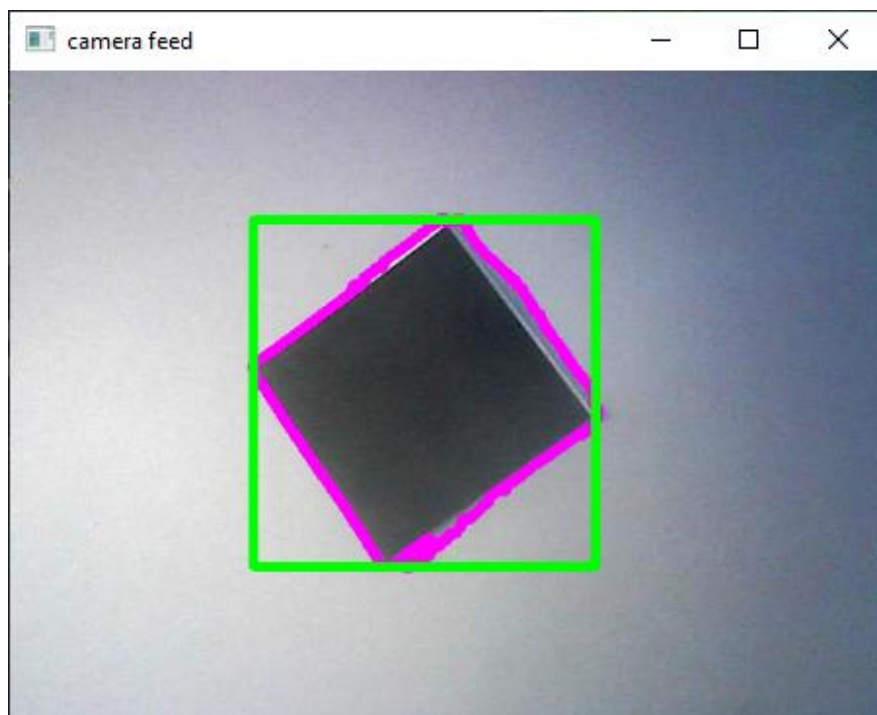
*Εικόνα 3.21: Αναπαραγωγή του περιγράμματος ενός τετραγώνου σχήματος με μοβ χρώμα στη ζωντανή μετάδοση της κάμερας*

### **3.6.5 Εύρεση γωνιακών σημείων και Εγκατάσταση περιοριστικού τετραγώνου**

Η διαδικασία της εύρεσης των γωνιακών σημείων αποτελεί καθοριστικό στάδιο για την αναγνώριση των σχημάτων. Με βάση τα γωνιακά σημεία θα γινόταν και η ταυτοποίησή τους στη συνέχεια. Για την υλοποίηση αυτής της λειτουργίας χρησιμοποιήθηκαν με αυτή τη σειρά

- η συνάρτηση `cv2.archLength` της OpenCV μέσω της οποίας υπολογίστηκε το μήκος του περιγράμματος του εντοπισμένου σχήματος
- η συνάρτηση `cv2.approxPolyDP` η οποία δέχεται ως παράμετρο την μεταβλητή της προηγούμενης συνάρτησης και προσεγγίζει έναν αριθμό πόντων για το μήκος του περιγράμματος του σχήματος. Οι πόντοι αυτοί μεταφέρονται και αποθηκεύονται σε μία λίστα.

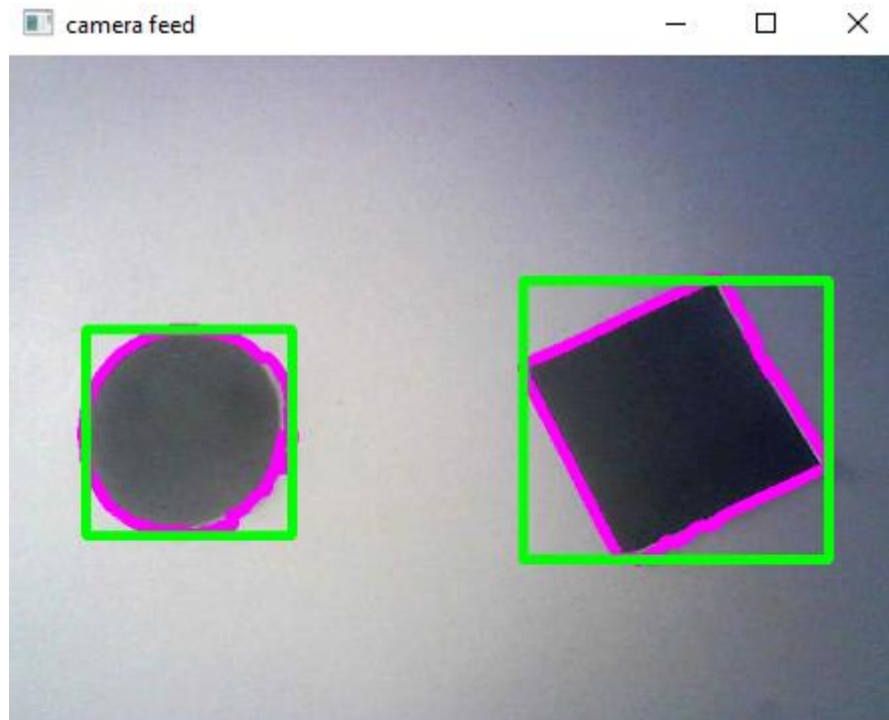
Με την εκτύπωση του μήκους της λίστας αυτής απεικονίζεται και ο αριθμός των πόντων που βρέθηκαν. Ανάλογα με τον αριθμό των πόντων έγινε και η ταυτοποίηση των σχημάτων. Για να γίνει πιο σαφής ο χώρος στον οποίο εντοπίζεται το σχήμα τόσο για το χρήστη όσο και για το ίδιο το σύστημα, αφού ενδέχεται να υπολογιστούν ως γωνιακά μη επιθυμητά σημεία λόγω θορύβου, δημιουργήθηκε ένα εικονικό περιοριστικό τετράγωνο. Αυτό το τετράγωνο υποδεικνύει την περιοχή που υπολογίζονται τα περιγράμματα και ο αριθμός των γωνιών του σχήματος που εμπεριέχεται μέσα τους. Η υλοποίηση του περιοριστικού τετραγώνου έγινε με τη χρήση της συνάρτησης `cv2.rectangle` με περασμένη ως παράμετρο το περίγραμμα του σχήματος.



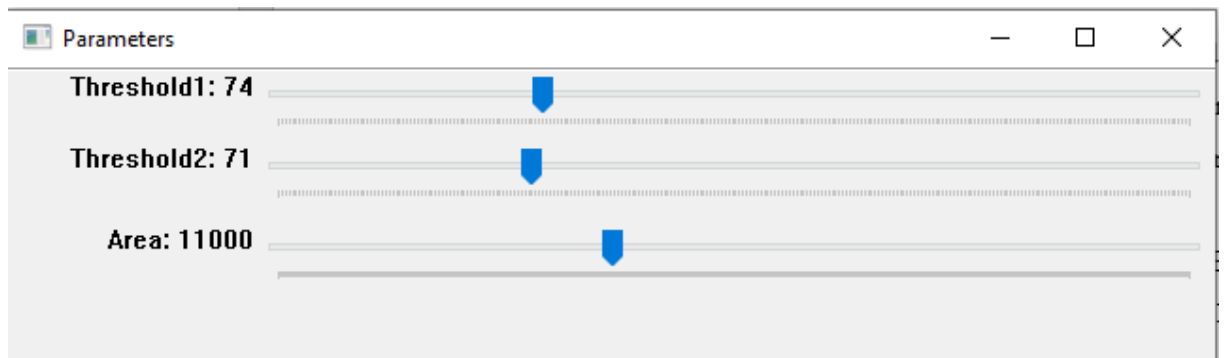
Εικόνα 3.22: Απεικόνιση του περιοριστικού τετραγώνου με πράσινο και του περιγράμματος με μοβ αντίστοιχα.

Λόγω του γεγονότος ότι στην περιοχή την οποία καλύπτει η κάμερα ενδέχεται να τοποθετηθούν πολλαπλά σχήματα, το σύστημα θα έπρεπε να είναι ικανό να τα διακρίνει και να τα ξεχωρίζει. Επομένως εφαρμόστηκε μία τεχνική για το φιλτράρισμα της περιοχής που καλύπτει η κάμερα. Με τη χρήση λοιπόν μίας ακόμη μπάρας με αναγραφόμενες τιμές μέσω της συνάρτησης `cv2.createTrackbar`, στην περίπτωση που το σύστημα δεν ξεχωρίζει από μόνο του σχήματα, ο χρήστης θα μπορεί να ρυθμίσει την περιοχή που καλύπτει το τετράγωνο περιορισμού ώστε να εξασφαλίσει την ομαλή λειτουργία της αναγνώρισης και της εύρεσης των γωνιών. Οι τιμές που αναγράφονται στην μπάρα τοποθετήθηκαν κατά προσέγγιση με

βάση τις προκαθορισμένες τιμές που προσδιόριζε την περιοχή που καλύπτει το περιοριστικό τετράγωνο η OpenCV. Στις παρακάτω εικόνες παρουσιάζεται η αναγνώριση περισσότερων σχημάτων με ξεχωριστά περιοριστικά τετράγωνα καθώς και η μπάρα με τις τιμές της περιοχής του περιοριστικού αυτού τετραγώνου.



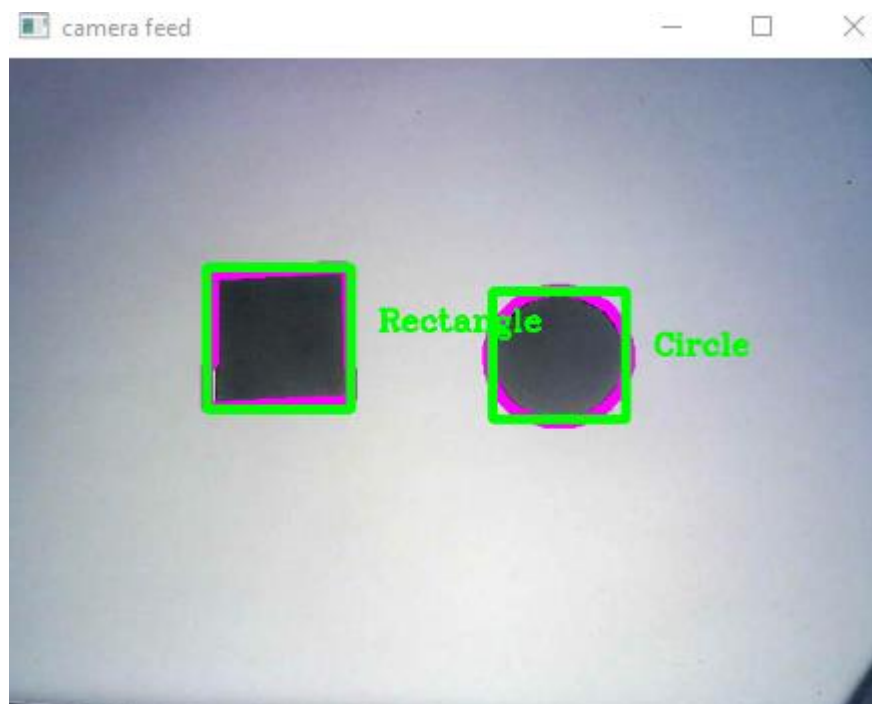
Εικόνα 3.23: Εντοπισμός ξεχωριστών σχημάτων εντός ξεχωριστών περιοριστικών τετραγώνων



Εικόνα 3.24: Το τελικό παράθυρο με τις μπάρες τιμών κατωφλίωσης 1, 2 και της περιοχής που καλύπτει το περιοριστικό τετράγωνο

### 3.6.6 Προσδιορισμός-Ταυτοποίηση σχήματος

Κατά την εμφάνιση της εικόνας με τα περιγράμματα, επιλέχθηκε να αναγράφεται το όνομα του σχήματος που αναγνωρίζεται (Εικόνα 3.25). Αυτό πραγματοποιήθηκε για την επιπλέον εξασφάλιση του ορθού εντοπισμού του σχήματος από το σύστημα μετά και την επιβεβαίωση του χρήστη. Επομένως μετά την απάντηση του ερωτήματος του «αν» ανιχνεύεται κάποιο σχήμα από το σύστημα, απαντάται και το «τι» δηλαδή το ποιο σχήμα ανιχνεύεται μέσω αυτού του είδους ταυτοποίησης. Η διαδικασία της ταυτοποίησης έγινε δια μέσω του εντοπισμού των γωνιακών σημείων στο σχήμα, ο οποίος αναφέρθηκε σε προηγούμενη ενότητα. Ανάλογα με τους πόντους που ανιχνεύονται, δηλαδή των γωνιών κατά κάποιο τρόπο που ανιχνεύονται, προσδιορίστηκε και το αντίστοιχο σχήμα (3 πόντοι-τρίγωνο, 4 πόντοι-τετράγωνο, 7 πόντοι-βέλος). Ο προσδιορισμός – ταυτοποίηση των σχημάτων έγινε εντός της συνάρτησης getContours.



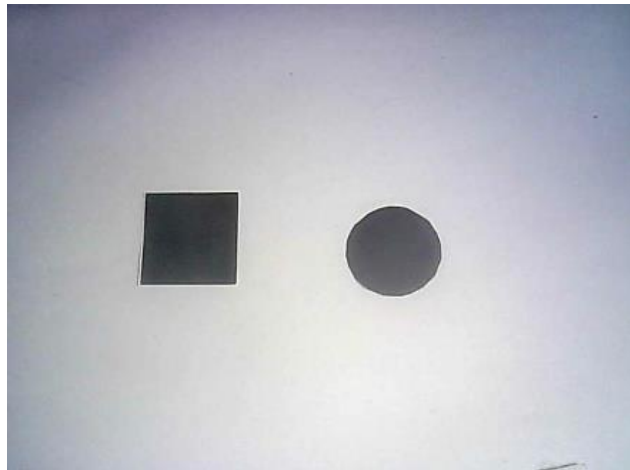
Εικόνα 3.25: Αναπαράσταση της εικόνας με την ταυτοποίηση των σχημάτων (αριστερά τετράγωνο-δεξιά κύκλος)

### 3.6.7 Διεπαφή Εντοπισμού και Καθορισμού σχήματος

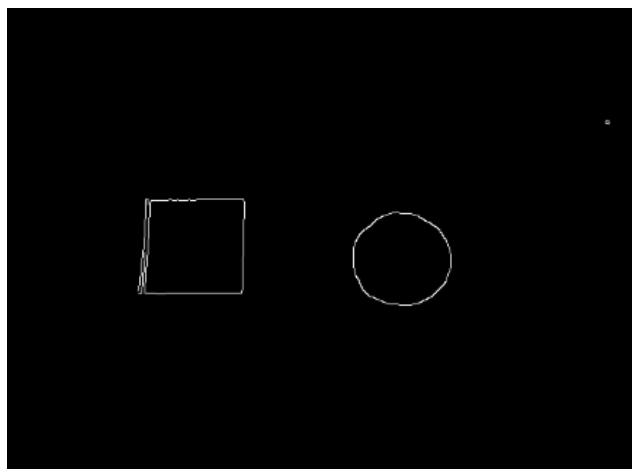
Η τελική διεπαφή του χρήστη σε ότι αφορά την αναγνώριση και τον εντοπισμό του σχήματος επιλέχθηκε να περιλαμβάνει τρεις εικόνες που πηγάζουν από την κάμερα. Συγκεκριμένα το παράθυρο της διεπαφής αποτελείται από τρία επιμέρους παράθυρα κάθετα στοιβαγμένα μεταξύ τους τα οποία περιλαμβάνουν με σειρά από πάνω προς τα κάτω:



- Την εικόνα στην οποία αναγράφονται τα περιγράμματα των σχημάτων με το περιοριστικό τετράγωνο και τα ονόματά τους, ώστε ο χρήστης να είναι πάντα ενήμερος για το εάν και τι σχήματα αντιλαμβάνεται το σύστημα. (Εικόνα 3.23)
- Την αφιльтраρίστη αναμετάδοση της εικόνας, ώστε ο χρήστης να αντιλαμβάνεται το εύρος χώρου το οποίο καλύπτει η κάμερα. (Εικόνα 3.26)
- Την εικόνα υπό την εφαρμογή του φίλτραρίσματος για την ανίχνευση των γραμμών Canny, δηλαδή ένα από τα βασικά εφαρμοζόμενα φίλτρα, ώστε ο χρήστης να έχει μία αίσθηση του τρόπου με τον οποίο γίνεται η διαδικασία φιλτραρίσματος της εικόνας και συγκεκριμένα του τρόπου με τον οποία γίνεται αντιληπτή η εικόνα από το σύστημα. (Εικόνα 3.27)

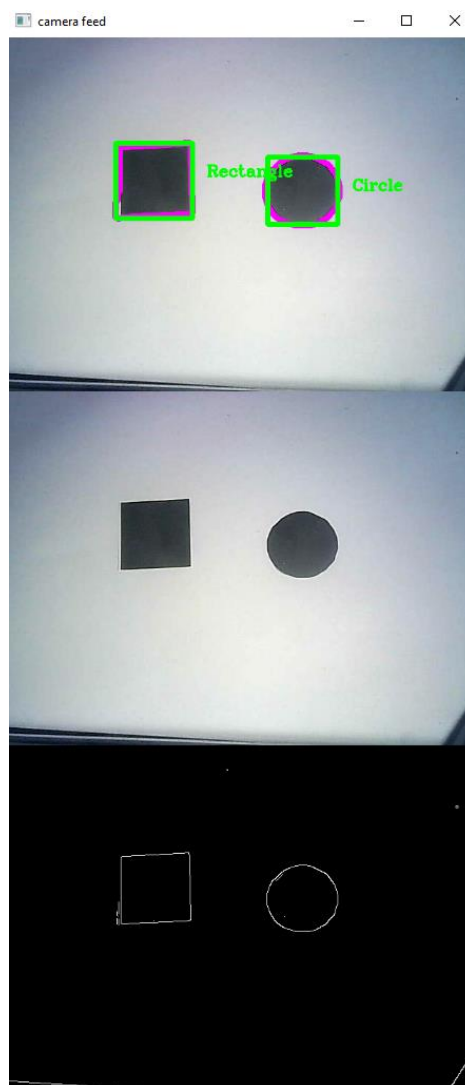


*Εικόνα 3.26: Η αναπαραγόμενη αφιльтраρίστη εικόνα*



*Εικόνα 3.27: Η εικόνα υπό την εφαρμογή του φίλτραρίσματος για την ανίχνευση γραμμών Canny*

Το ολοκληρωμένο παράθυρο με την ζωντανή μετάδοση των εικόνων όπως προαναφέρθηκε, σχεδιάστηκε να περιλαμβάνει όλες τις παραπάνω εικόνες ως ένα σύνολο σε κάθετη στοίχιση (Εικόνα 3.27). Η στοίχιση αυτή υλοποιήθηκε μέσω της συνάρτησης `stackimages`, η οποία μεταξύ άλλων περιλαμβάνει τις συναρτήσεις της OpenCV `np.vstack` και `np.hstack` όπου πραγματοποιούν τις στοιχίσεις των εικόνων κάθετα και οριζόντια αντίστοιχα (Εικόνα 3.28). Η κάθετη τοποθέτηση των εικόνων επιλέχθηκε για λόγους ευχρηστίας ώστε ο εκάστοτε χρήστης να μπορεί να επιβλέπει και να επιβεβαιώνει το αποτέλεσμα της ανίχνευσης και αναγνώρισης του σχήματος με ένα βλέμμα χωρίς την συνεχή ανακατεύθυνσή του.



Εικόνα 3.28: Το παράθυρο της διεπαφής του χρήστη για την διαδικασία του εντοπισμού και της αναγνώρισης των σχημάτων

### 3.7 Κατασκευή των Κύβων

Οι κύβοι αποτελούν το μέσο επικοινωνίας του χρήστη με το λογισμικό (software) μέρος του συστήματος. Η αλληλεπίδρασή του χρήστη με το σύστημα και η γενικότερη εμπειρία της χρήσης του εξαρτάται άμεσα από αυτούς. Επομένως τόσο το ίδιο το υλικό μέρος όσο και η διαδικασία κατασκευής των κύβων αποτέλεσαν καθοριστικό κομμάτι για την υλοποίηση του έργου στο σύνολό του.

Η κατασκευή των κύβων πραγματοποιήθηκε με τη μέθοδο της τρισδιάστατης εκτύπωσης. Συγκεκριμένα μέσω του τρισδιάστατου εκτυπωτή Ender-3 και την χρήση των προγραμμάτων Sketchup και Cura. Οι λόγοι για τους οποίους ο τρόπος κατασκευής επιλέχθηκε να γίνει με τρισδιάστατο εκτυπωτή είναι διότι οι παραγόμενοι κύβοι θα αποτελούσαν

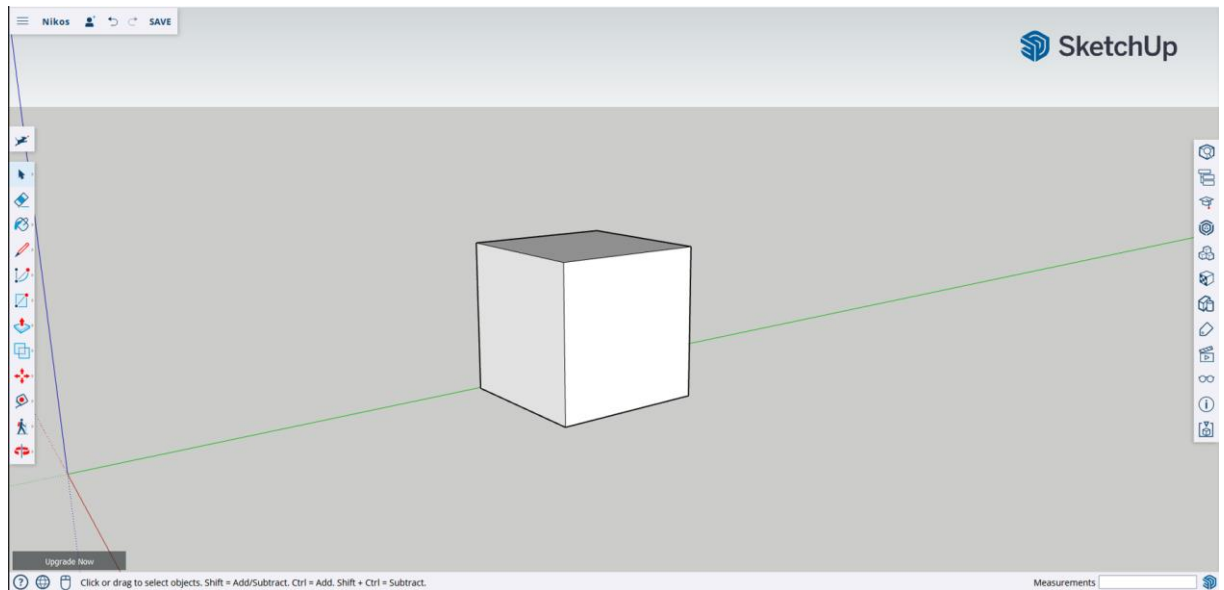
- στιβαρές κατασκευές
- με λεία επιφάνεια
- ακριβείς ορθές γωνίες
- και καθαρό άσπρο χρώμα

παράγοντες που θα έπαιζαν καθοριστικό ρόλο στην ανίχνευση και αναγνώριση των τοποθετημένων στις πλευρές τους σχημάτων. Η διαδικασία κατασκευής τους περιέλαβε δύο επιμέρους στάδια, του σχεδιασμού και της εκτύπωσής τους.

#### 3.7.1 Σχεδιασμός Κύβου

Προκειμένου να πραγματοποιηθεί η διαδικασία εκτύπωσης των κύβων και τελικά η ολοκλήρωση της κατασκευής τους έπρεπε πρώτα να προηγηθεί ο σχεδιασμός τους και στη συνέχεια, η μετάφραση του παραγόμενου σχεδίου σε γλώσσα μηχανής που να αντιλαμβάνεται και να ερμηνεύει ο εκτυπωτής.

Για το σχεδιασμό του κύβου χρησιμοποιήθηκε η διαδικτυακή εφαρμογή Sketchup, η οποία αποτελεί ένα χρήσιμο και εύκολο εργαλείο επεξεργασίας και δημιουργίας σχεδίων για τρισδιάστατους εκτυπωτές. Το σχέδιο των κύβων υλοποιήθηκε με τη χρήση των εργαλείων rectangle αρχικά για τη δημιουργία ενός τετραγώνου με πλευρές 5 χιλιοστών, και του push/pull για την προσφορά μίας εικονικής τρισδιάστατης υπόστασης στο είδη υπάρχον τετράγωνο. Το τελικό εικονικό σχέδιο μετά την χρήση των παραπάνω εργαλείων παρουσιάζεται στην παρακάτω εικόνα (Εικόνα 3.29).

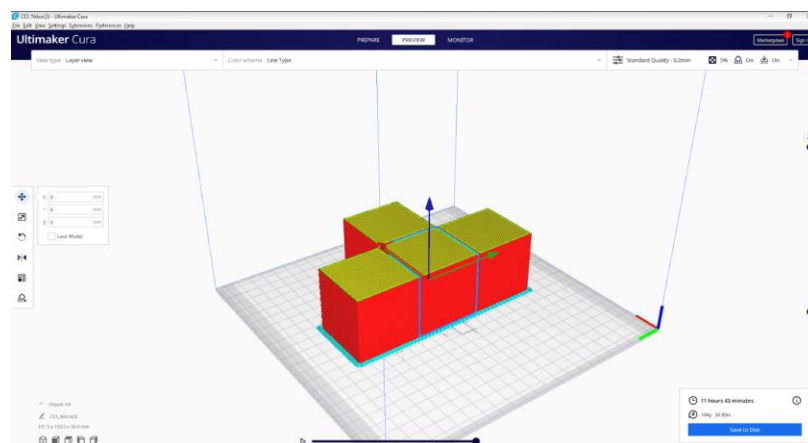


Εικόνα 3.29: Το τελικό εικονικό σχέδιο του κύβου στο περιβάλλον του Sketchup

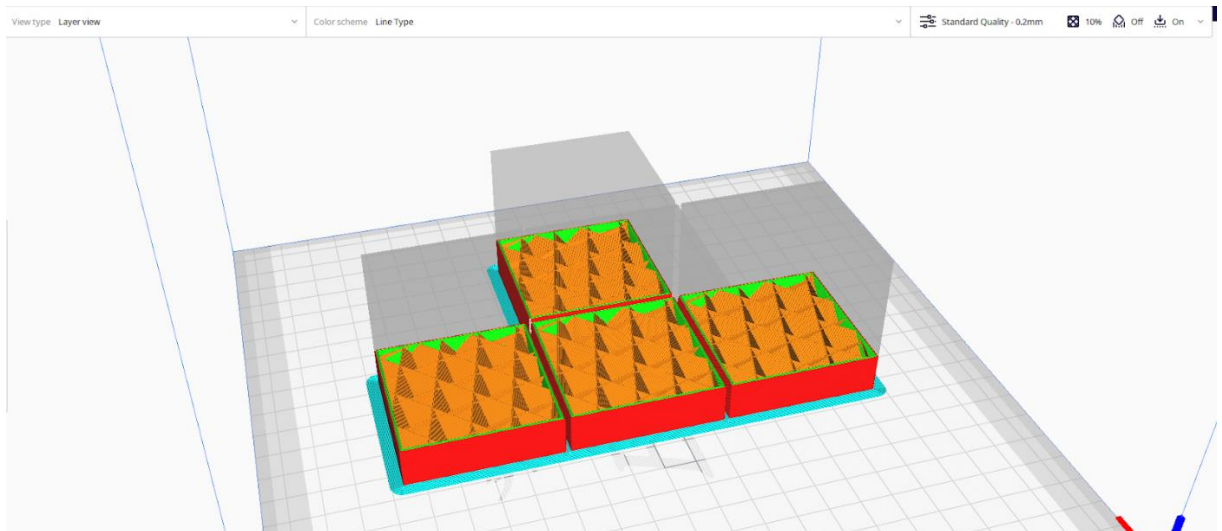
Μετά το σχεδιασμό το ολοκληρωθέν σχέδιο αποθηκεύθηκε σε μορφή αρχείου .stl και εισήχθη στο επόμενο πρόγραμμα επεξεργασίας, το Cura. Μέσω του Cura πραγματοποιήθηκε:

- Ο σχεδιασμός του τελικού σχήματος προς εκτύπωση
- Η ρύθμιση των παραμέτρων του εκτυπωτή
- Μετατροπή του αρχείου του σχεδίου σε αρχείο της μορφής .gcode και μεταφόρτωση του σε κάρτα μνήμης.

Για την εξοικονόμηση χρόνου αλλά και υλικού εκτύπωσης οι κύβοι σχεδιάστηκαν να εκτυπωθούν ως ομάδες τεσσάρων. Επομένως με βάση το αρχικό σχέδιο στο Sketchup έγινε ο εκ νέου προσδιορισμός και επαναπροσαρμογή του στο Cura (Εικόνα 3.30, Εικόνα 3.31).



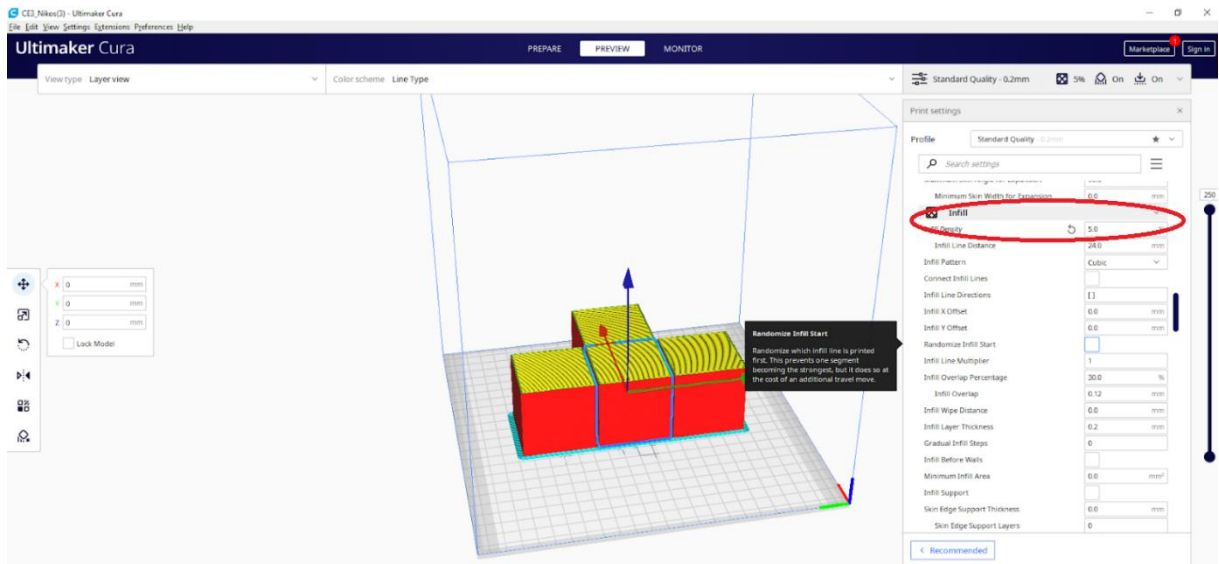
Εικόνα 3.30: Το νέο σχέδιο προς εκτύπωση στο περιβάλλον του Cura



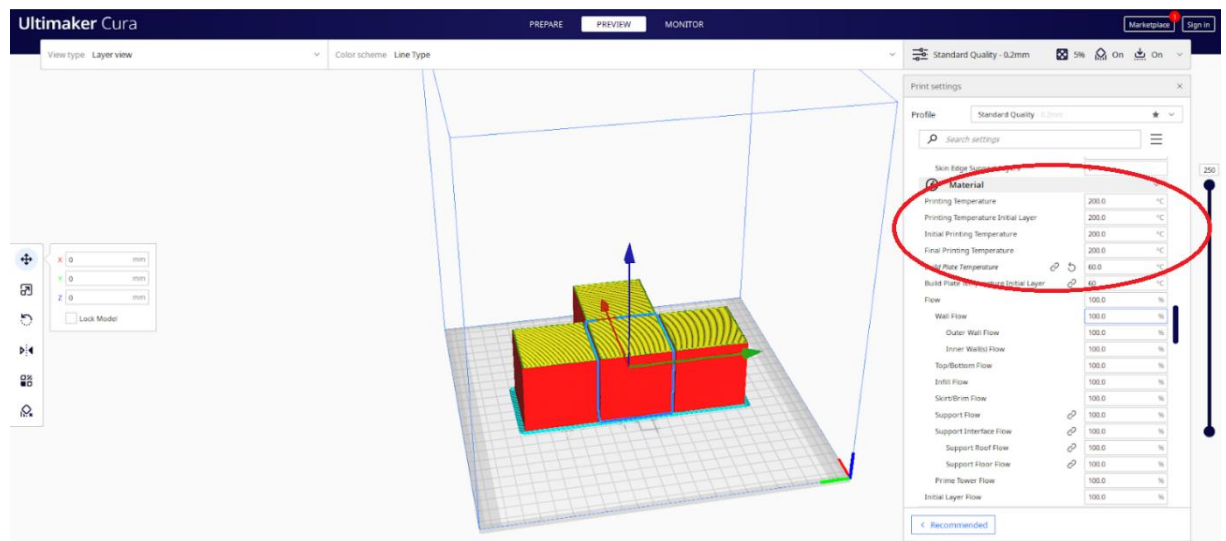
Εικόνα 3.31: Το εσωτερικό των κύβων προς εκτύπωση

Μετά το σχεδιασμό σειρά είχε η ρύθμιση των παραμέτρων του εκτυπωτή. Οι βασικές αυτές ρυθμίσεις του εκτυπωτή περιλάμβαναν:

- Το πλάτος της γραμμής εκτύπωσης στο 0.4 διότι η διάμετρος της μύτης (nozzle) του εκτυπωτή είναι 0.4 εκατοστά
- Η πλήρωση-γέμισμα (infill) της γραμμής και κατά συνέπεια όλου σχήματος στο 5 %. Η επιλογή αυτή έγινε για να μην κινδυνέψει η ακεραιότητα της ίδιας της εκτύπωσης, για λόγους εξοικονόμησης χρόνου εκτύπωσης και ώστε η τελική κατασκευή να είναι ανθεκτική.(Εικόνα 3.32)
- Η θερμοκρασία εκτύπωσης (Printing Temperature) στους 200 βαθμούς κελσίου.
- Η θερμοκρασία της επιφάνειας εκτύπωσης (Build Plate Temperature) στους 60 βαθμούς κελσίου (Εικόνα 3.33).
- Ο τύπος πρόσφυσης πλάκας κατασκευής (Build Plate Adhesion Type) στην επιλογή Brim, προκειμένου να δημιουργηθεί μια επιφάνεια γύρω από τη βάση της εκτύπωσης ώστε να κρατά σταθερό το σχήμα. Για την δημιουργία στην ουσία μίας ακόμη δικλείδας ασφαλείας κατά την εκτύπωση.



Εικόνα 3.32: Η ρύθμιση της παραμέτρου του infill στο Cura

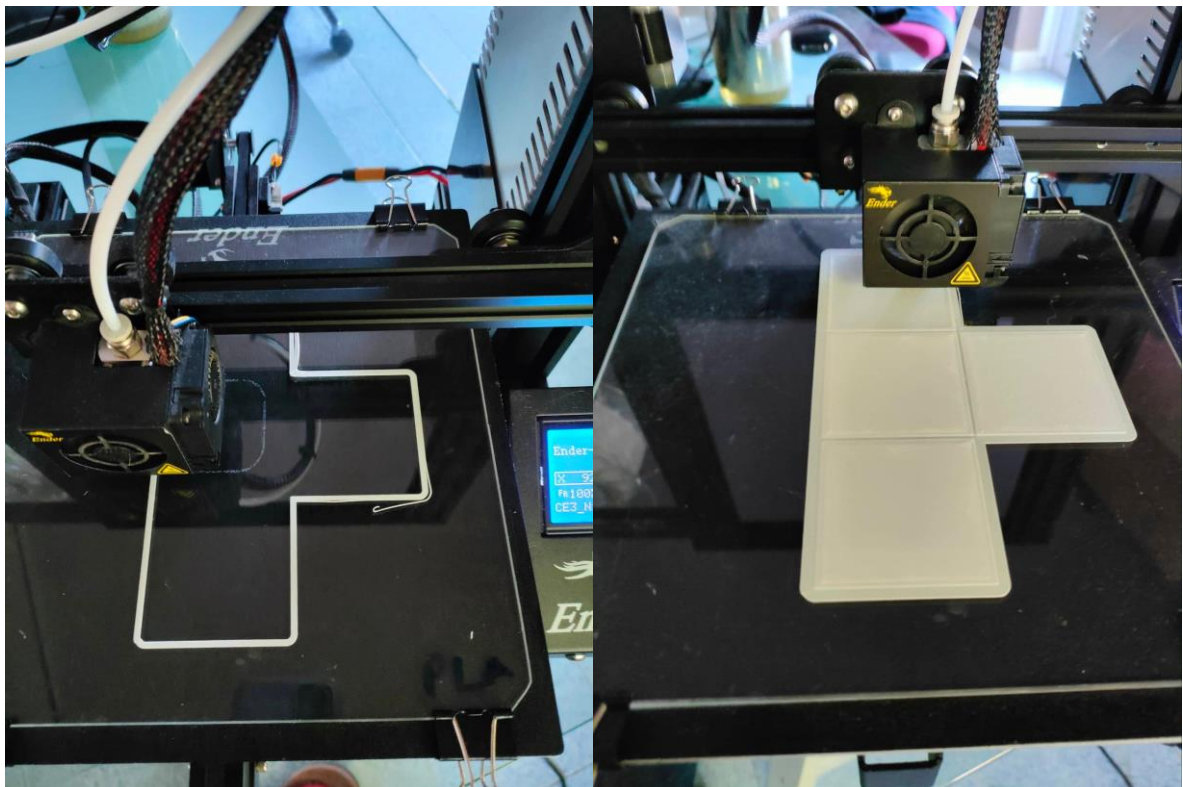


Εικόνα 3.33: Η ρύθμιση των θερμοκρασιών μύτης και επιφάνειας εκτύπωσης στο Cura

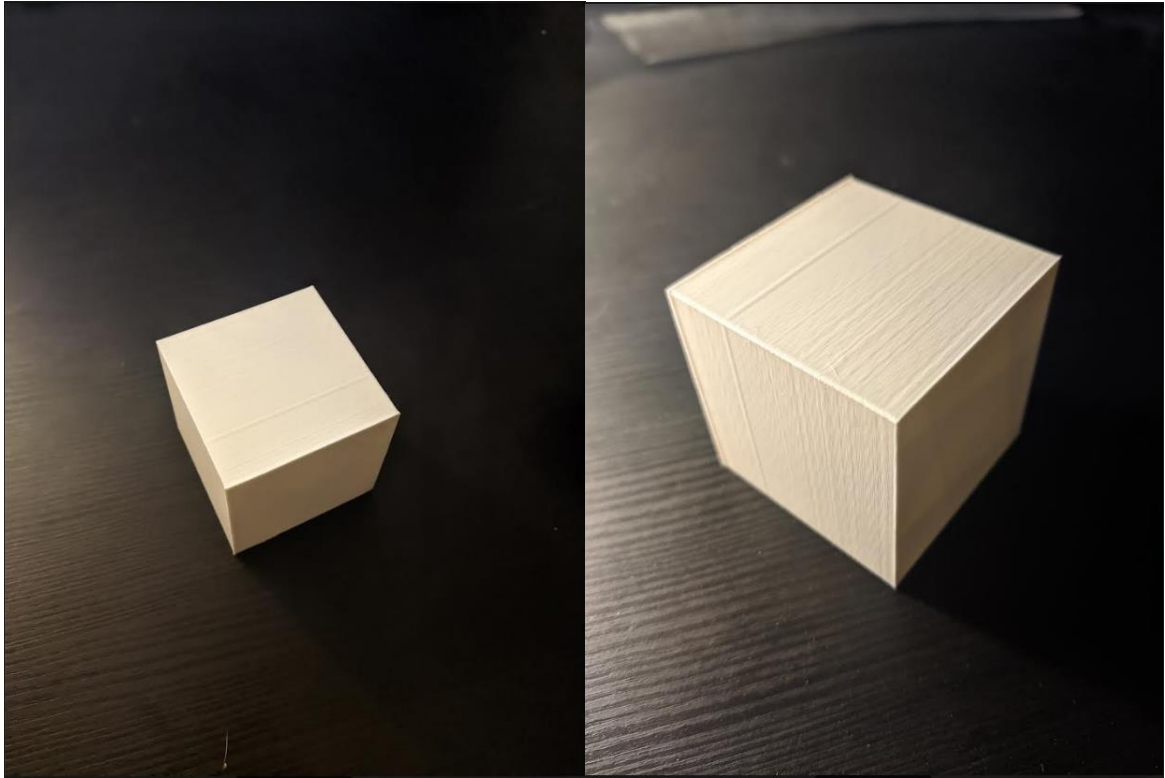
Οι παράμετροι κατά την διαδικασία των ρυθμίσεων προσδιορίστηκαν με βάση τη μέθοδο του trial and error (δοκιμής και λάθους) σε παρόμοιες εκτυπώσεις ώστε να γίνει η κατάλληλη προσέγγιση για την παραγωγή του βέλτιστου αποτελέσματος κατά την εκτύπωση. Μετά το σχεδιασμό των κύβων και τις ρυθμίσεις του εκτυπωτή έγινε η μεταφορά του αρχείου .gcode πλέον από το ίδιο το πρόγραμμα σε κάρτα μνήμης για την εισαγωγή της στον εκτυπωτή. Το αρχείο αυτό περιλαμβάνει τις πληροφορίες από όλα τα προηγούμενα στην γλώσσα μηχανής του εκτυπωτή.

### 3.7.2 Εκτύπωση κύβου και ολοκλήρωση κατασκευής

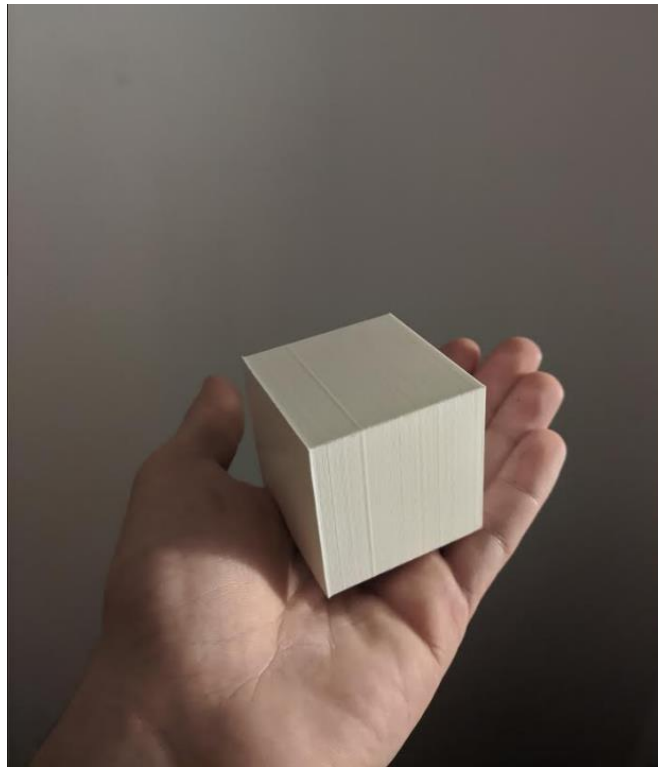
Μετά τη μεταφορά των πληροφοριών σχετικά με τις ρυθμίσεις και το τρισδιάστατο σχέδιο των κύβων σειρά είχε η ίδια η διαδικασία της εκτύπωσης. Το υλικό το οποίο χρησιμοποιήθηκε ήταν το PLA της εταιρίας Devil Design σε λευκό χρώμα, ένα από τα πιο εύχρηστα και οικονομικά υλικά της κατηγορίας του. Πριν την εκκίνηση της διαδικασίας εξασφαλίστηκε η επιφάνεια της εκτύπωσης να βρίσκεται σε ευθυγραμμισμένο επίπεδο μέσω της ρύθμισης των βιδών στις τέσσερις γωνίες κοντά στην επιφάνεια του εκτυπωτή. Αξίζει να σημειωθεί ότι για το σχεδιασμό των διαστάσεων των κύβων λήφθηκε υπόψη ότι θα έπρεπε να είναι άνετοι στο χειρισμό τους από ένα παιδί μικρής ηλικίας και κατασκευασμένοι από ένα ανθεκτικό υλικό. Η όλη διαδικασία εκτύπωσης τεσσάρων κύβων (Εικόνα 3.34) διήρκεσε έντεκα ώρες και σαράντα λεπτά με την εκτύπωση του ενός να διαρκεί περίπου στις τρεις ώρες και επτά λεπτά. Το τελικό προϊόν της εκτύπωσης παρουσιάζεται στην Εικόνα 3.35, 3.36.



Εικόνα 3.34: Στάδια εκτύπωσης των κύβων στον τρισδιάστατο εκτυπωτή



*Εικόνα 3.35: Το τελικό προϊόν της εκτύπωσης*



*Εικόνα 3.36: Το τελικό προϊόν σε σχέση με ένα χέρι ενηλίκου*



Για την ολοκλήρωση της κατασκευής έγινε η τοποθέτηση των σχημάτων στις πλευρές των κύβων. Τα σχήματα εκτυπώθηκαν αρχικά σε χαρτί σε μαύρο χρώμα, κόπηκαν και κολλήθηκαν στην κάθε πλευρά των κύβων. Ο απτός κύβος λοιπόν ως τελική ολοκληρωμένη κατασκευή παρουσιάζεται στην εικόνα 3.37



*Εικόνα 3.37: Ολοκληρωμένη κατασκευή κύβου με τοποθέτηση του σχήματος του τετραγώνου*

### **3.8 Διασύνδεση επί μέρους λειτουργιών**

Μετά την ολοκλήρωση της υλοποίησης των επιμέρους στοιχείων της εκτύπωσης , σειρά είχε η πραγματοποίηση της σύνδεσής τους. Το προϊόν της σύνδεσης αυτής, που αποτέλεσε και το γενικότερο αποτέλεσμα της εργασίας, αλλά και ο τρόπος υλοποίησής της παρουσιάζεται παρακάτω.

Αρχικά, ήδη από τη διαδικασία υλοποίησης του παιχνιδιού δημιουργήθηκε μία επαναλαμβανόμενη λούπα while η οποία είναι υπεύθυνη για την συνεχόμενη αναπαραγωγή των εικόνων άρα και για τη λειτουργία του παιχνιδιού γενικότερα. Αργότερα μέσα σε αυτή τη λούπα προστέθηκε ο κώδικας για τη λειτουργία της κάμερας αλλά και η εφαρμογή των φίλτρων για την λειτουργία της αναγνώρισης των σχημάτων (Εικόνα 3.38).

```

207 while run:
208     success,img= cap.read()
209
210     imgContour=img.copy()
211     clock.tick(FPS)
212     keys=pygame.key.get_pressed()
213
214
215     cv2.resizeWindow
216
217     for event in pygame.event.get():
218         if event.type==pygame.QUIT:
219             run=False
220             break
221

```

Εικόνα 3.38: Απόσπασμα κώδικα λούπας while

Στη συνέχεια σε αυτή τη δομή επανάληψης έγινε η αντιστοιχία σχήματος και κίνησης. Δηλαδή ορίστηκε ότι ανάλογα με το σχήμα που επιδεικνύεται στο ίδιο το σύστημα, να εκτελείται και η κατάλληλη κίνηση από το όχημα στην πίστα. Πιο συγκεκριμένα η αντιστοιχία σχημάτων κινήσεων ορίστηκε να είναι :

- Βέλος – Μπροστά
- Τετράγωνο – Αριστερά
- Κύκλος – Δεξιά
- Τρίγωνο – Πίσω

Απόσπασμα του συγκεκριμένου κώδικα παρουσιάζεται στην παρακάτω εικόνα (Εικόνα 3.39).

```

251 for shape in sortedInstructions:
252     if shape=='Arrow':
253         moved=True
254         for i in range(5):
255             if car.collission(TRACK_BORDER_MASK) != None:
256                 break
257             else:
258                 car.move_forward()
259
260     if shape=='Rectangle':
261         moved=True
262         if car.collission(TRACK_BORDER_MASK) != None:
263             break
264         else:
265             car.rotate(right=True)
266
267     if shape=='Circle':
268         moved=True
269         if car.collission(TRACK_BORDER_MASK) != None:
270             break
271         else:
272             car.rotate(left=True)
273
274     if shape=='Triangle':
275         moved=True
276         for i in range(10):
277             if car.collission(TRACK_BORDER_MASK) != None:
278                 break
279             else:
280                 car.move_backward()

```

Εικόνα 3.39: Απόσπασμα κώδικα ορισμού σχήματος - κίνησης

Πιο συγκεκριμένα, δημιουργήθηκε μία λίστα η οποία συμπληρώνεται με τα ονόματα των αναγραφόμενων στην εικόνα αναγνωρισμένων σχημάτων. Η λίστα αυτή στη συνέχεια ανατρέχεται και ανάλογα με το όνομα ή τα ονόματα του σχήματος που εμπεριέχει εκτελούνται οι ανάλογες κινήσεις.

Πριν αυτή τη διαδικασία όμως στη λίστα έγινε η εφαρμογή μίας συγκεκριμένης ταξινόμησης των ονομάτων των σχημάτων από αριστερά προς τα δεξιά (Εικόνα 3.40). Δηλαδή η συγκεκριμένη σειρά με την οποία παρουσιάζονται τα σχήματα στους κύβους να αναγράφεται αυστηρά και στη λίστα. Αυτή η λειτουργία εφαρμόστηκε λόγω του γεγονότος ότι έτσι και αλλιώς τα παιδιά που θα χρησιμοποιήσουν το σύστημα ασυναίσθητα θα τοποθετούν τα σχήματα με τη σειρά που επιθυμούν να εκτελεστούν από αριστερά προς τα δεξιά, επομένως προς διευκόλυνσή τους και για αποφυγή συγχύσεων και το σύστημα θα έπρεπε να ακολουθήσει αυτή τη σειρά κινήσεων.

```
instructions, instructionOrder = getContours(imgDil,imgContour)
```

```
sortedInstructions = [x for y,x in sorted(zip(instructionOrder,instructions))]
for shape in sortedInstructions:
    if shape=='Arrow':
        moved=True
        for i in range(5):
            if car.collision(TRACK_BORDER_MASK) != None:
                break
            else:
                car.move_forward()
```

Εικόνα 3.40: Απόσπασμα κώδικα για την ταξινόμηση των σχημάτων

Μία ακόμη λειτουργία του συστήματος που αξίζει να αναφερθεί είναι αυτή του αυτοματισμού που αφορά την εισαγωγή των σχημάτων προς επεξεργασία στο σύστημα και κατ' επέκταση της κίνησης του οχήματος στην πίστα. Ειδικότερα, δημιουργήθηκε μία διαδικασία η οποία περιλαμβάνει τρία στάδια.

Στο αρχικό στάδιο ορίστηκε ένας χρόνος για τη ρύθμιση των παραμέτρων της κατωφλίωσης. Συγκεκριμένα κατά την εκκίνηση της αναγνώρισης σχημάτων, όπως έχει προαναφερθεί ο χρήστης καλείται μέσω των αριθμημένων μπαρών που εμφανίζονται να ρυθμίσει την κατωφλίωση προς αποφυγή θορύβων στην εικόνα, για την ολοκλήρωση της διαδικασίας αυτής από το χρήστη ορίστηκε ένας χρόνο συγκεκριμένα είκοσι δευτερολέπτων.

Κατά το δεύτερο στάδιο μετά το πέρας αυτών των δευτερολέπτων το σύστημα ξεκινά και ανιχνεύει σχήματα. Στη συνέχεια εάν το σύστημα δεν ανιχνεύσει κανένα σχήμα το αυτοκίνητο παραμένει στην αρχική του θέση χωρίς να κινείται. Στην αντίθετη περίπτωση που ανιχνευτεί σχήμα όμως, καθορίστηκε ένας χρόνος πέντε δευτερολέπτων επεξεργασίας της παραγόμενης κίνησης από το χρήστη μέσω των σχημάτων που τοποθέτησε. Δηλαδή εάν το σύστημα ανιχνεύσει σχήμα ο χρήστης έχει στη διάθεση του πέντε δευτερόλεπτα για να αλλάξει την επιλογή του και να επανατοποθετήσει σχήματα ή να επιτρέψει στο σύστημα να μεταφράσει το σχήμα ή τον συνδυασμό σχημάτων που τοποθέτησε σε κίνηση για το αυτοκίνητο. Σε κάθε περίπτωση εάν το σύστημα ανιχνεύσει οποιοδήποτε σχήμα, μετά από πέντε δευτερόλεπτα που εξακολουθεί να το εντοπίζει, ορίστηκε να το αποθηκεύει ώστε να καταλήξει στο τρίτο στάδιο.

Στο τρίτο αυτό στάδιο, όπως προαναφέρθηκε γίνεται και η τελική μετάφραση του σχήματος σε κίνηση. Ανάλογα με το πόσα και ποια σχήματα τοποθετήθηκαν μέσω των κύβων μπροστά στην κάμερα εκτελείται και η ανάλογη κίνηση από το όχημα. Η υλοποίηση της παραπάνω διαδικασίας έγινε μεταξύ άλλων και μέσω διαχείρισης λιστών που πραγματεύονται χρόνο. Αποσπάσματα του κώδικα παρουσιάζονται παρακάτω (Εικόνα 3.41). Για μεγαλύτερη και περαιτέρω κατανόηση, παρακάτω παρουσιάζονται σχηματικά τα προαναφερθέντα στάδια (Εικόνα 3.42).

```

checkIfInstructionsChangedEvery = 5
prevInstructions = []
instructionsChangedAt = int( time.time() )

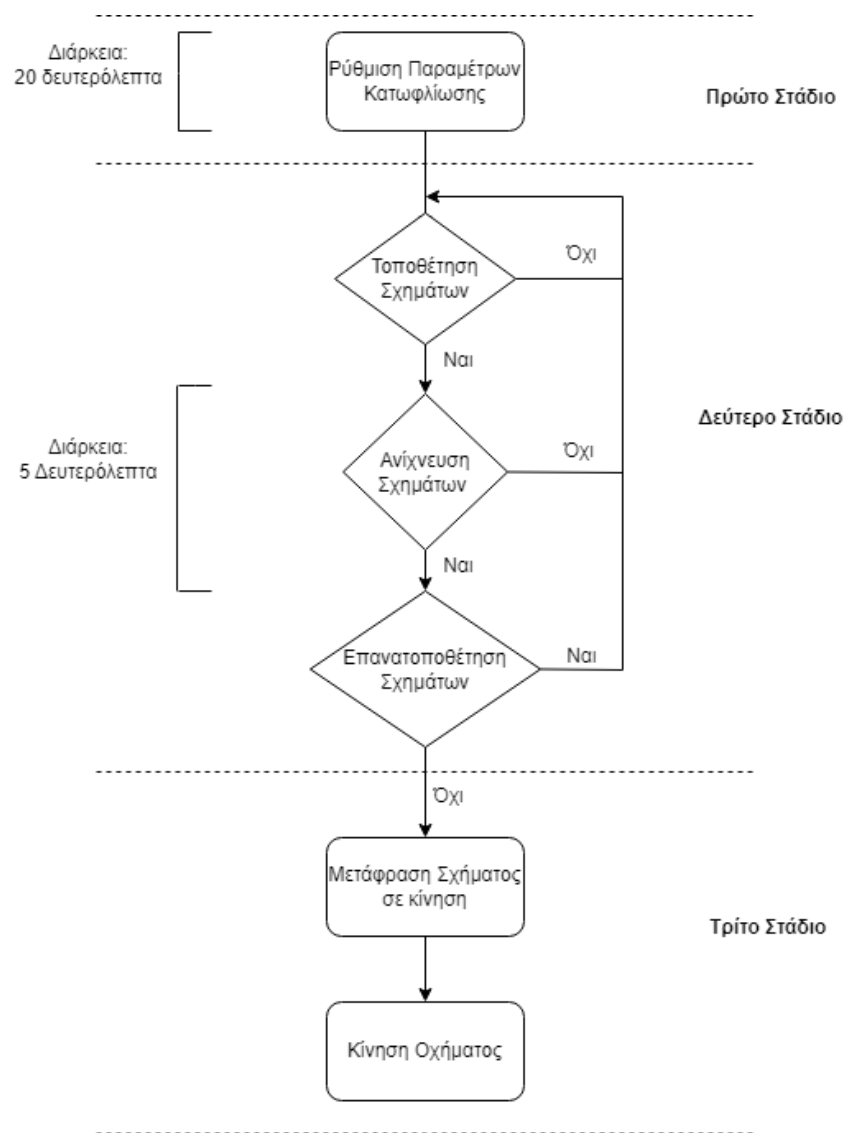
waitForSettings = int( time.time() )

hasRunned = False

if set(tmpInstructions) == set(prevInstructions) and len(instructions) and int( time.time() ) - instructionsChangedAt > 3 and int( time.time() ) - waitForSettings > 20 and not hasRunned:
    print("running...")
    hasRunned = True
    sortedInstructions = [x for y,x in sorted(zip(instructionOrder,instructions))]
    for shape in sortedInstructions:
        if shape=="Arrow":
            moved=True
            for i in range(5):
                if car_collision(TRACK_BORDER_MASK) != None:
                    break
                else:
                    car.move_forward()

```

Εικόνα 3.41: Μέρος κώδικα με τη διαχείριση λιστών για τη λειτουργία του αυτοματισμού



Εικόνα 3.42: Διάγραμμα αυτοματισμού τοποθέτησης σχημάτων

### 3.9 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο περιγράφηκε αναλυτικά και επεξηγήθηκε η διαδικασία υλοποίησης των τριών μερών που απαρτίζουν την εργασία από πλευράς τόσο λογισμικού όσο και του υλικού μέρους της εργασίας. Πιο συγκεκριμένα έγινε αναφορά στα είδη χρηστών του συστήματος, στον τρόπο υλοποίησης του εικονικού παιχνιδιού και στον εντοπισμό σχημάτων με μία web κάμερα μέσω αναλυτικής παρουσίασης σημαντικών μερών του κώδικα. Επίσης παρουσιάστηκε η τεχνική και το μέσο κατασκευής των κύβων καθώς και ο τρόπος σχεδίασής τους. Τέλος αναλύθηκε και παρουσιάστηκε ο τρόπος διασύνδεσης των επί μέρους στοιχείων του συστήματος και ο αυτοματισμός που αφορά την είσοδο των σχημάτων ως οδηγίες κατεύθυνσης του αυτοκινήτου.

## Κεφάλαιο 4 – Πειραματική Διαδικασία

Σε αυτό το κεφάλαιο παρουσιάζεται η διαδικασία του πειραματισμού του συστήματος. Ο πειραματισμός, περιλαμβάνει τόσο την δοκιμή της λειτουργίας του κάθε μέρους του συστήματος ξεχωριστά, όσο και τη δοκιμή του ως ενιαίο σύνολο. Εξετάζεται η απόκριση του συστήματος σε διαφορετικές συνθήκες περιβάλλοντος και αναφέρονται προγραμματιστικές τροποποιήσεις σχετικά με τη λειτουργία του. Επίσης παρουσιάζεται η τελική διεπαφή χρήστη και ο τρόπος παιχνιδιού, μετά την τελική έκβαση και τη μελέτη του αποτελέσματος της πειραματικής αυτής διαδικασίας.

### 4.1 Προσδιορισμός περιβάλλοντος

Η ορθή λειτουργία του συστήματος εξαρτάται άμεσα από την επιτυχή λήψη εικόνων μέσω της κάμερας. Η επιτυχής αυτή λήψη καθορίζεται από το αν τα σχήματα που παρουσιάζονται στην κάμερα γίνονται αντιληπτά σε αρχικό στάδιο και κατά συνέπεια ερμηνεύονται σωστά από το σύστημα. Για την εξασφάλιση των ιδανικών συνθηκών επομένως που θα καθόριζαν την επιτυχία της εκτέλεσης των λειτουργιών του συστήματος, έπρεπε να καθοριστεί και το κατάλληλο περιβάλλον. Ο καθορισμός του περιβάλλοντος περιλαμβάνει τρεις κυρίως βασικούς παράγοντες οι οποίοι αφορούν

- το φωτισμό
- το μέγεθος του σχήματος
- το ύψος της τοποθέτησης της κάμερας

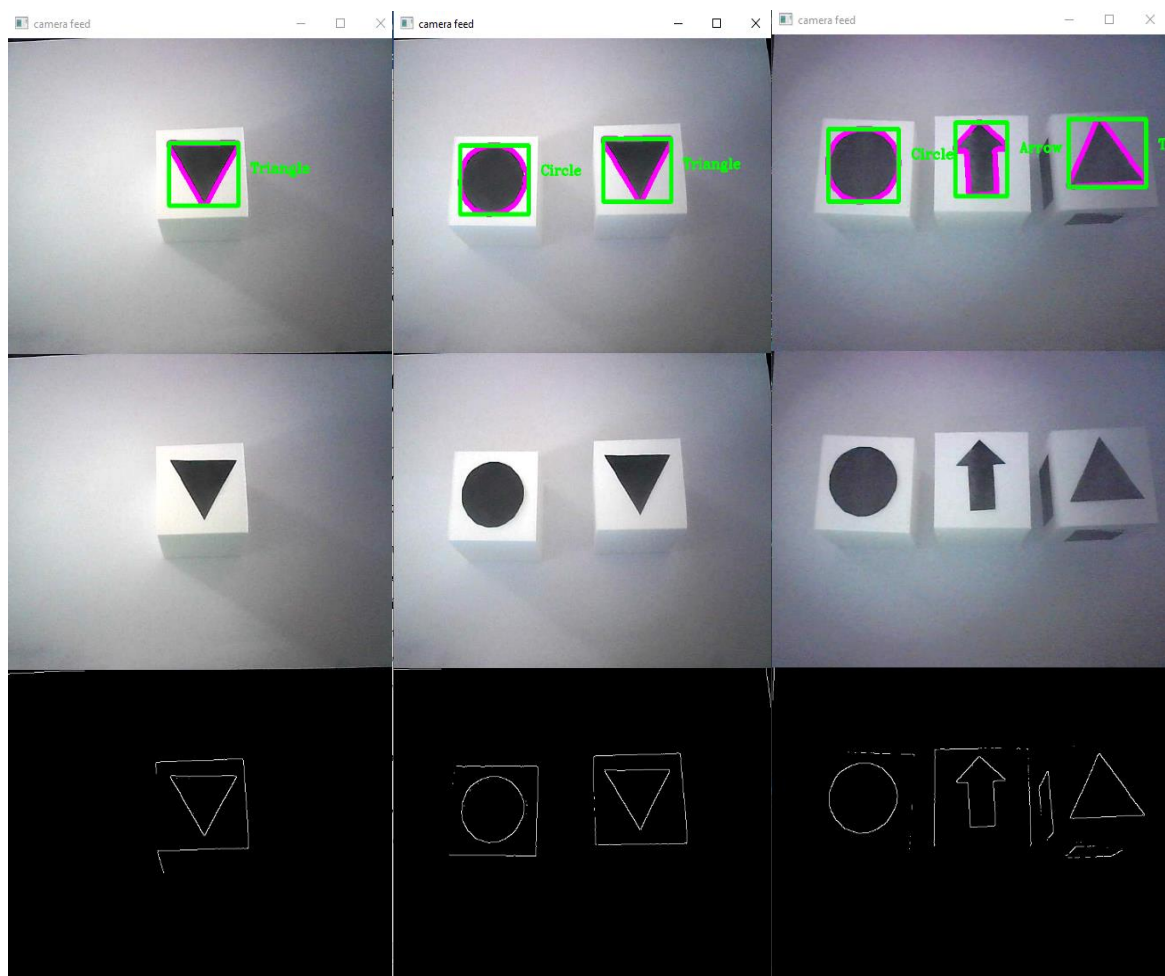
Δεδομένου του γεγονότος ότι η πειραματική διαδικασία γενικότερα πραγματοποιήθηκε σε συγκεκριμένο δωμάτιο (που ενδεχομένως θα μπορούσε να αντικατοπτρίσει πραγματικές συνθήκες δωματίου αίθουσας διδασκαλίας) η διαδικασία αναγνώρισης των σχημάτων εκτελέστηκε σε τρεις ενδεικτικές φάσεις της ημέρας. Ο σκοπός που η διαδικασία ακολούθησε αυτή τη λογική είναι το γεγονός ότι ανάλογα με τη χρονική φάση της ημέρας διέφερε και ο φυσικός φωτισμός, πράγμα που όπως έχει προαναφερθεί αποτελεί κρίσιμο παράγοντα για τη διαδικασία αναγνώρισης των σχημάτων.

Πέρα από το φυσικό φωτισμό του δωματίου όμως, σημαντικό ρόλο έπαιξε και ο καθορισμός του μεγέθους του ίδιου του σχήματος. Ένα μεγάλο ενδεχομένως σε μέγεθος σχήμα διευκολύνει κατά πολύ την διαδικασία αναγνώρισής του από το σύστημα. Δεδομένου

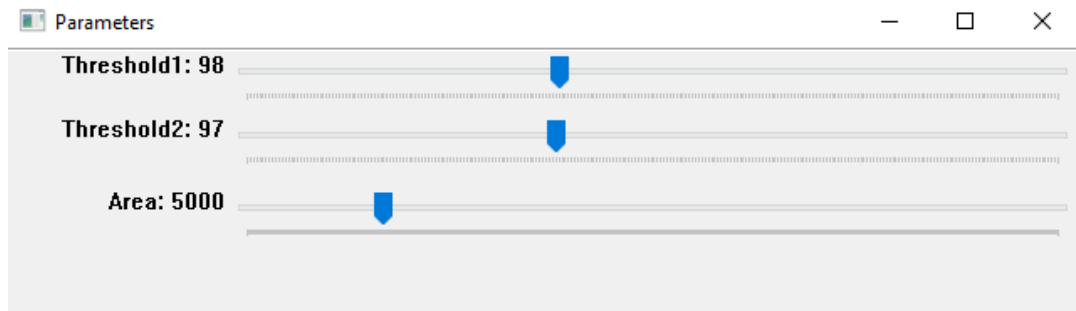
όμως του ότι το σχήμα αυτό έπρεπε να τοποθετηθεί στις πλευρές των κύβων δεν έπρεπε να υπερβαίνει τις διαστάσεις των πλευρών του και να προεξέχει. Επομένως το μέγεθος των σχημάτων που θα χρησιμοποιούνταν καθορίστηκε εμπειρικά από νωρίς λόγω πρακτικότητας.

Το ύψος τοποθέτησης της κάμερας σε σχέση με το επίπεδο που θα γινόταν η εναπόθεση των κύβων καθορίστηκε ανάλογα με το πόσους κύβους θα μπορούσε να συμπεριλάβει η εικόνα. Δεδομένου ότι η κάμερα θα είχε κατεύθυνση προς τα κάτω, προς το επίπεδο δηλαδή εναπόθεσης της των κύβων, είναι λογικό πως όσο πιο κοντά πλησίαζε προς αυτό τόσο θα περιοριζόταν η εικόνα. Επομένως το ύψος τοποθέτησης της κάμερας καθορίστηκε αρχικά στα 35 εκατοστά περίπου από το επίπεδο.

Έχοντας εξασφαλίσει λοιπόν το μέγεθος των σχημάτων και το ύψος τοποθέτησης της κάμερας παρατίθενται στη συνέχεια εικόνες από την πειραματική διαδικασία, με διαφορετικό φυσικό φωτισμό σε δύο διαφορετικές χρονικές στιγμές της ημέρας.



Εικόνα 4.1: Αναγνώριση ενός, δύο και τριών σχημάτων αντίστοιχα με φυσικό φως πρωινών-μεσημεριανών ωρών



Εικόνα 4.2: Το σετ ρυθμίσεων κατοφλίωσης και περιοχής για την αναγνώριση σχημάτων στην εικόνα 4.1

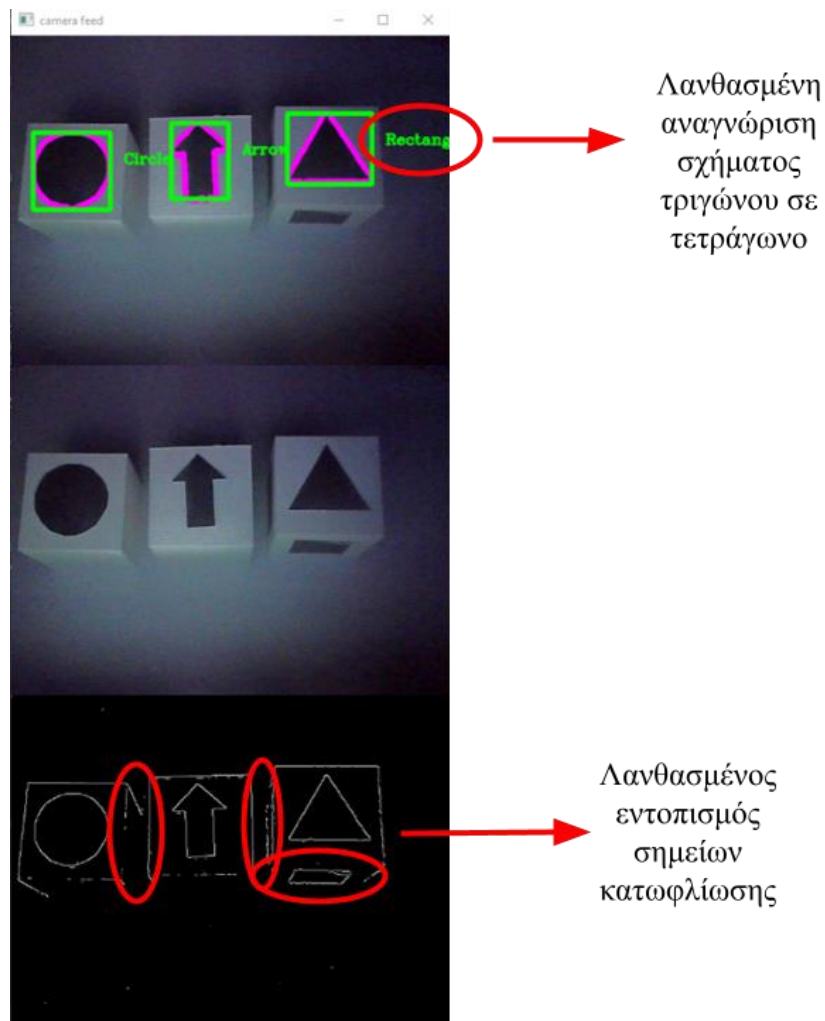
Κατά την εκτέλεση των πειραμάτων τις πρωινές ώρες (όπως φαίνεται και στην Εικόνα 4.1) με αρκετό φυσικό φως και μετά την εισαγωγή των ρυθμίσεων των παραμέτρων που καθορίζουν την αναγνώριση (όπως φαίνεται στην εικόνα 4.2), τα σχήματα αναγνωρίζονται επιτυχώς και η όλη διαδικασία ρύθμισης των παραμέτρων ώστε να εντοπίζονται τα σχήματα κρατά λίγα δευτερόλεπτα.



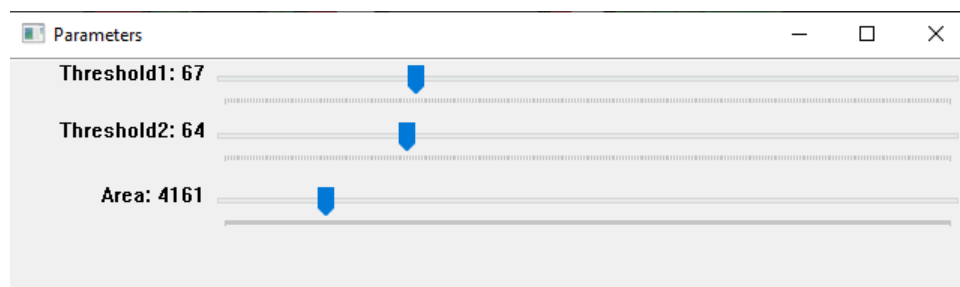
Εικόνα 4.3: Στιγμιότυπο λανθασμένης αναγνώρισης σχημάτων τις απογευματινές ώρες με λιγότερο φωτισμό



Μετά το πέρας μερικών ωρών και κατά τη μείωση του φυσικού φωτός (Εικόνα 4.3) παρατηρήθηκε ότι με το ίδιο σετ ρυθμίσεων της Εικόνας 4.2 γίνεται μερικώς η και καθόλου σωστή αναγνώριση των σχημάτων, γεγονός που οφείλεται στις φυσικές σκιάσεις και τη γενικότερη έλλειψη φωτισμού. Με την αλλαγή των μετρικών και των ρυθμίσεων γενικότερα που αναδεικνύονται στον πίνακα στην Εικόνα 4.5 παρατηρήθηκε ακόμη και τότε λανθασμένη αναγνώριση σε σημεία που επισημαίνονται και στην Εικόνα 4.4.

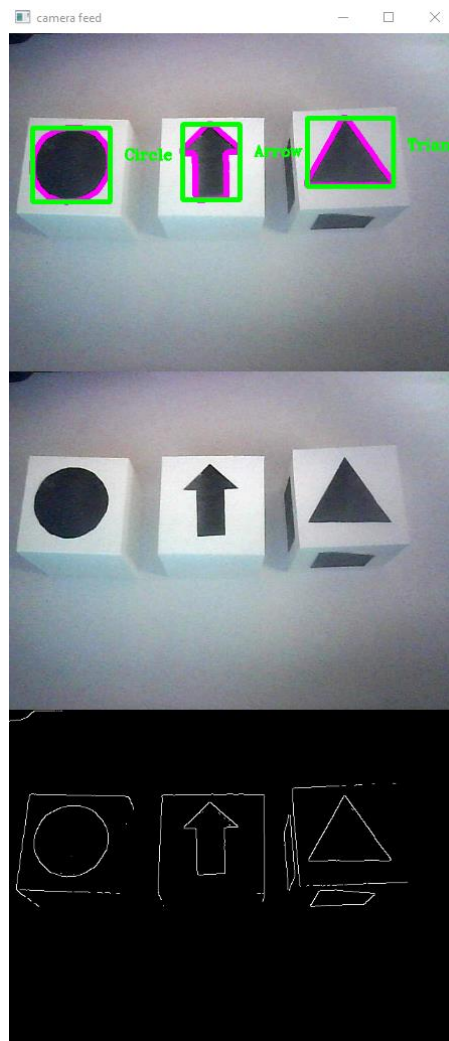


Εικόνα 4.4: Επισημάσση σημείων λανθασμένης αναγνώρισης μετά την αλλαγή των ρυθμίσεων



Εικόνα 4.5: Το σετ ρυθμίσεων κατωφλίωσης και περιοχής για την αναγνώριση των σχημάτων στην εικόνα 4.4

Αξίζει επίσης να σημειωθεί ότι η κατάλληλη ρύθμιση των παραμέτρων του σετ στην εικόνα 4.5 διήρκεσε μερικά λεπτά λόγω της ευαισθησίας του συστήματος εξαιτίας του λιγοστού φωτισμού. Με την εισαγωγή της παραμέτρου του τεχνητού φωτός βέβαια, παρατηρήθηκε ότι η διαδικασία αναγνώρισης διευκολύνεται αισθητά, αλλά παραμένει ασταθής (Εικόνα 4.6). Πιο συγκεκριμένα η αλλαγή στις ρυθμίσεις γίνεται πολύ πιο εύκολα και τα σχήματα αναγνωρίζονται επιτυχώς, αλλά η οποιαδήποτε αλλαγή στο περιβάλλον που εκτελείται η διαδικασία μπορεί να δημιουργήσει σκιάσεις και ενδεχομένως να μεταβάλει στιγμιαία την ορθότητα της διαδικασίας.



Εικόνα 4.6: Επιτυχής διαδικασία αναγνώρισης σχημάτων με τεχνητό φως

Το συμπέρασμα που προκύπτει από τον πειραματισμό με την εν λόγω λειτουργία αναγνώρισης είναι ότι η απόδοσή της εξαρτάται άμεσα από το φως του δωματίου. Επομένως η εκτέλεσή της και κατά προέκταση η εκτέλεση όλου σου συστήματος ενδείκνυται να γίνεται

ώρες της ημέρας με αρκετό φυσικό φως το οποίο διαχέεται στο δωμάτιο ή κατ' εξαίρεση με την παρουσία αρκετού τεχνητού φωτός το οποίο να διαχέεται επίσης ομοιόμορφα στο χώρο για την αποφυγή σκιάσεων που θα μπορούσαν να επηρεάσουν την αποδοτικότητά του.

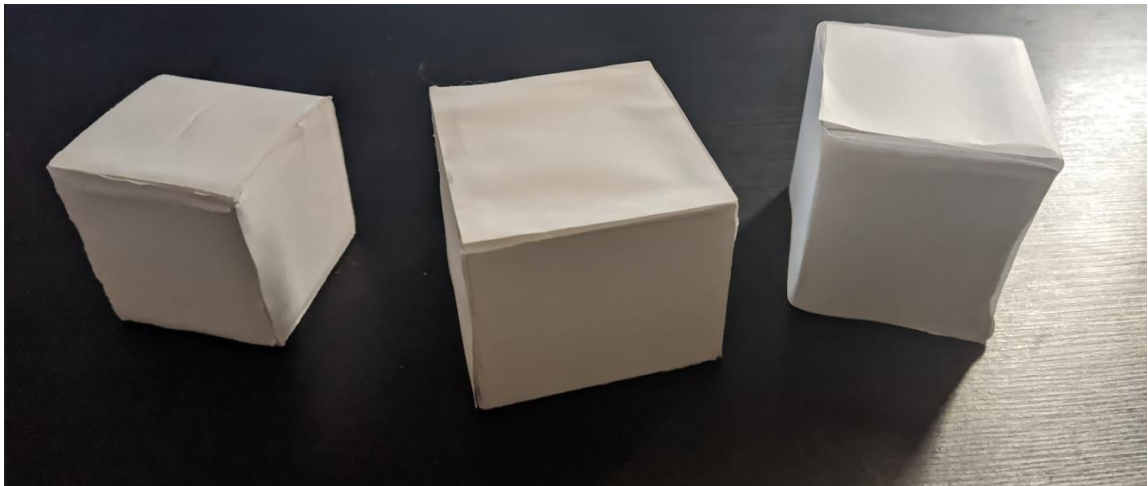
## 4.2 Πειραματισμός κατασκευής κύβων

Η τελική κατασκευή των κύβων δεν ήταν αποτέλεσμα έμπνευσης, αλλά μελέτης και σταδιακού πειραματισμού, που αφορούσε το υλικό της ίδιας της κατασκευής, αλλά και της ανταπόκρισης των λειτουργιών του συστήματος στην κατασκευή αυτή. Παρακάτω παρουσιάζονται αναλυτικά τα αποτελέσματα της μελέτης που αφορά το υλικό της κατασκευής των κύβων.

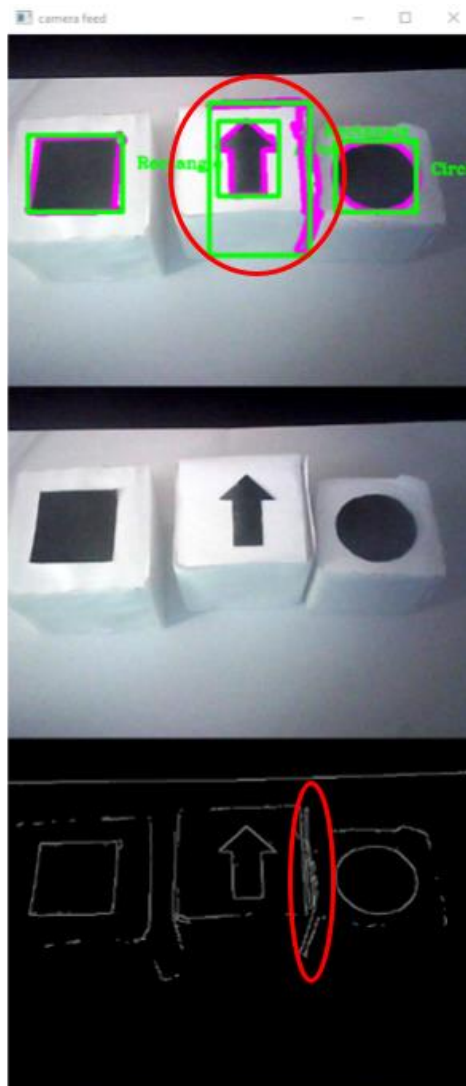
### 4.2.1 Κατασκευή από μακετόχαρτο

Εφόσον ήταν σαφές ότι το χρώμα της κατασκευής θα έπρεπε ήταν άσπρο για τη διευκόλυνση της διαδικασίας αναγνώρισης του σχήματος στις πλευρές του (ως ένα μέσο «καμουφλαρίσματος» κατά κάποιο τρόπο), η αρχική προσέγγιση της κατασκευής των κύβων ήταν να γίνει από άσπρο απλό χαρτί. Η σκέψη αυτή απορρίφθηκε γρήγορα όμως καθώς μια ενδεχομένως χάρτινη κατασκευή δεν θα ήταν καθόλου σταθερή και συμπαγής δεδομένου ότι θα χειριζόταν από μικρά παιδιά.

Το αμέσως επόμενο βήμα ήταν η προσέγγιση της κατασκευής των κύβων από άσπρο μακετόχαρτο. Το μακετόχαρτο είναι ένα εύχρηστο υλικό και το κόστος του αρκετά μικρό, γεγονός που το καθιστά κατάλληλο για τέτοιου είδους κατασκευές. Κατά την ολοκλήρωση της κατασκευής όμως παρατηρήθηκε ότι στα σημεία των γωνιών των κύβων δημιουργούνται αρκετές ανωμαλίες λόγω του πάχους του ίδιου του μακετόχαρτου, γεγονός που θα προκαλούσε ενδεχομένως δυσκολία στην αναγνώριση των σχημάτων στις πλευρές του λόγω των παρεμβολών στην εικόνα κατά τη διαδικασία της κατωφλίωσης. Επομένως επιλέχθηκε η ήδη υπάρχουσα κατασκευή να περιτυλιχθεί με κανονικό χαρτί για την εξομάλυνση των προαναφερθέντων ελαττωματικών σημείων. Τα αποτελέσματα της κατασκευής παρουσιάζονται παρακάτω (Εικόνα 4.7). Σειρά στη συνέχεια είχε και ο πειραματισμός τους με σχήματα κατά τη διαδικασία αναγνώρισης σχημάτων, τα αποτελέσματα της οποίας φαίνονται στην Εικόνα 4.8.



Εικόνα 4.7: Απόπειρες κατασκευής κύβων από μακετόχαρτο



Εικόνα 4.8: Παράδειγμα λανθασμένης αναγνώρισης σχημάτων λόγω παρεμβολών κατά την κατωφλίωση που φανερώνονται στα επισημασμένα σημεία.

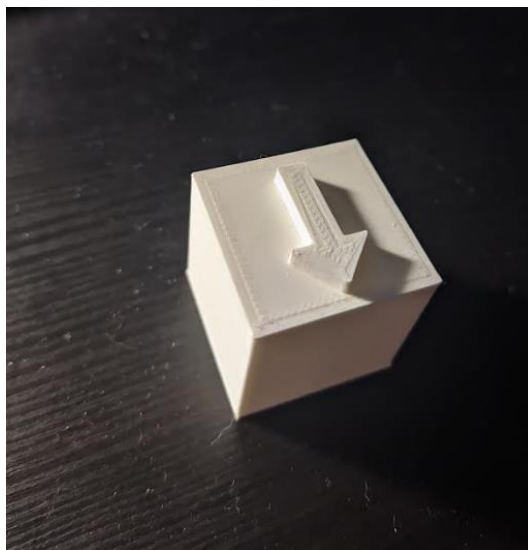
Από τον παραπάνω πειραματισμό με τους κύβους από μακετόχαρτο προέκυψε επομένως το συμπέρασμα ότι η διαδικασία αναγνώρισης παραμένει ασταθής λόγω ανωμαλιών στην ίδια την κατασκευή οι οποίες με τη σειρά τους δημιουργούν θόρυβο στην εικόνα. Η ανώμαλη επιφάνεια των κύβων σε συνδυασμό με τις αστάθειες στις γωνίες και στις άκρες των πλευρών τους κατέστησαν την κατασκευή ακατάλληλη προς χρήση για την επίτευξη του στόχου της εργασίας.

#### 4.2.2 Κατασκευή με τρισδιάστατο εκτυπωτή

Μετά την απόρριψη της επιλογής της κατασκευής των κύβων από μακετόχαρτο έγινε προσέγγιση του ενδεχομένου κατασκευής μέσω τρισδιάστατου εκτυπωτή. Η επιλογή αυτή έγινε με βάση το γεγονός ότι θα προέκυπτε ένα ολοκληρωμένο αποτέλεσμα με:

- λεία επιφάνεια
- καθαρές γωνίες και όρια πλευρών
- καθαρό άσπρο χρώμα

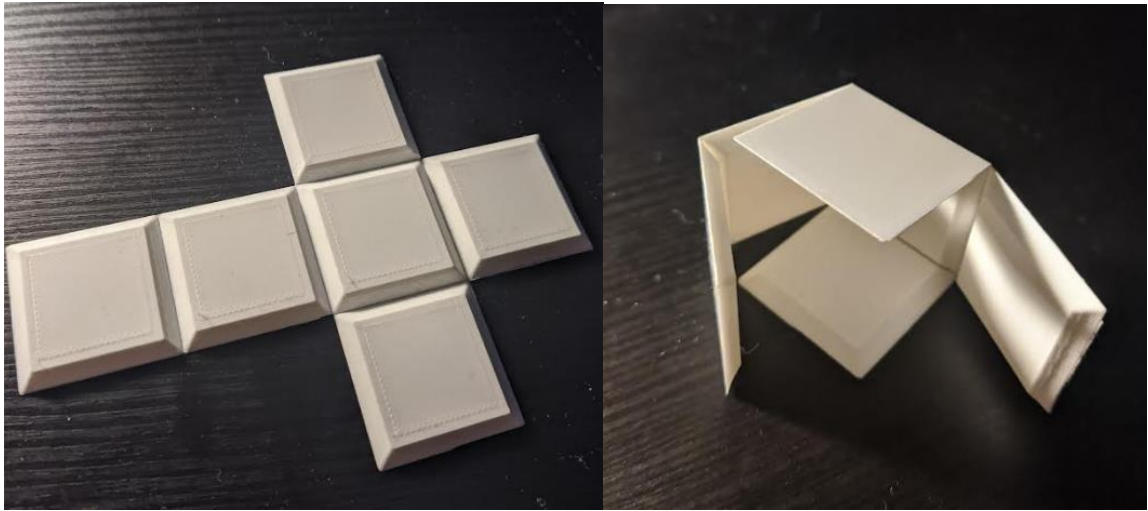
Το τελικό κατασκευαστικό αποτέλεσμα όμως δεν προέκυψε κατευθείαν, αλλά ήταν αποτέλεσμα περαιτέρω πειραματισμών. Στις παρακάτω εικόνες παρουσιάζονται δοκιμαστικοί εκτυπωμένοι κύβοι και οι λόγοι για τους οποίους δεν επιλέχθηκε η εκτύπωσή τους



*Εικόνα 4.9: Προσέγγιση κατασκευής κύβου με ενσωματωμένο σχήμα*

Στην παραπάνω εικόνα (Εικόνα 4.9) απεικονίζεται η προσπάθεια υλοποίησης του κύβου με ενσωματωμένο σχήμα. Η κατασκευή του όμως εκτιμήθηκε ως αρκετά χρονοβόρα για την εκτύπωση στο επιθυμητό μέγεθος κύβου άρα και περισσότερο κοστοβόρα ως προς την

κατανάλωση πόρων. Επιπλέον το γεγονός ότι το σχήμα έπρεπε να είναι κατεξοχήν μαύρου χρώματος περιέπλεκε ακόμη περισσότερο τη διαδικασία κατασκευής του αφού θα απαιτούσε και επιπλέον υλικό κατασκευής.



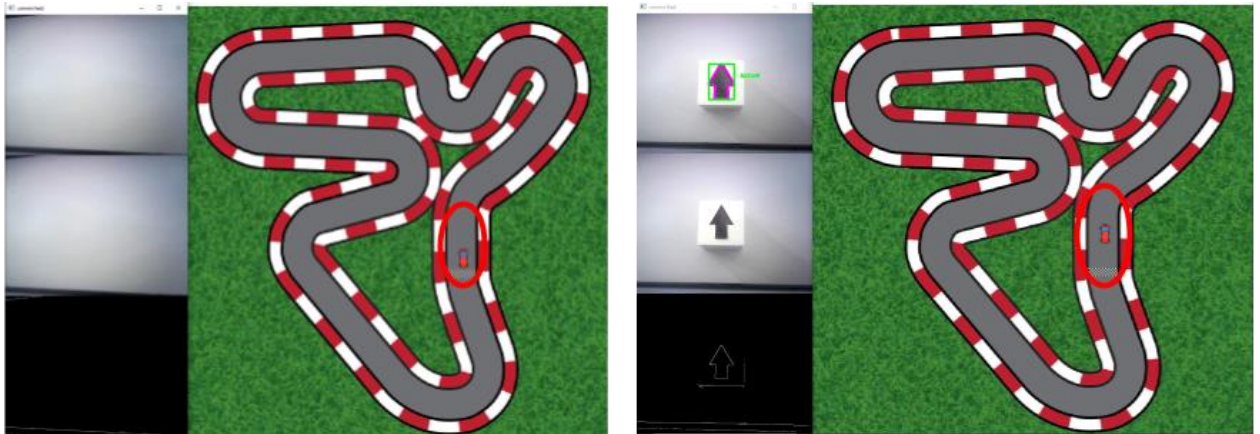
*Εικόνα 4.10: Προσέγγιση κατασκευής κύβου από το ανάπτυγμά του*

Στην παραπάνω εικόνα (Εικόνα 4.10) απεικονίζεται η προσπάθεια υλοποίησης του κύβου μέσω της συναρμολόγησης του αναπτύγματός του. Η ιδέα πίσω από αυτή την προσέγγιση ήταν ότι μετά τη συναρμολόγηση του τελικού κύβου οι επιφάνειές του θα ήταν εντελώς λείες. Αργότερα όμως διαπιστώθηκε πως το η συγκεκριμένη παράμετρος δεν έπαιξε ιδιαίτερο ρόλο κατά την αναγνώριση των σχημάτων αφού αρκούσε απλά το χρώμα της κατασκευής να ήταν άσπρο. Επιπλέον η συγκεκριμένη προσέγγιση της κατασκευής απαιτούσε αρκετό χρόνο τόσο για την υλοποίηση από τον εκτυπωτή όσο και για τη συναρμολόγηση του αναπτύγματος λόγω ακρίβειας κινήσεων.

### **4.3 Πειραματική επαλήθευση συστήματος**

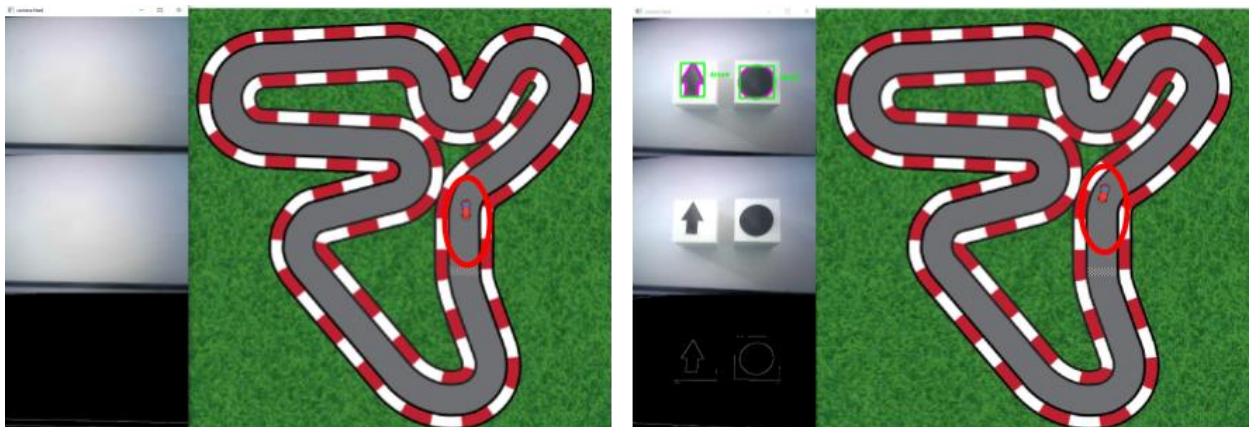
Η εξασφάλιση της ορθής αλλά και απρόσκοπτης λειτουργίας του συνόλου του συστήματος αποτέλεσε βασικό κομμάτι και της ευρύτερης υλοποίησης του. Σε αυτή την υποενότητα θα αναφερθεί το αποτέλεσμα της λειτουργίας αυτής του συστήματος, όπως διαπιστώθηκε μετά από επανειλημμένα πειράματα που πραγματοποιήθηκαν. Τα πειράματα αφορούσαν πρώτα την εισαγωγή σχημάτων μεμονωμένα και στη συνέχεια την εισαγωγή συνδυασμών σχημάτων. Οι συνδυασμοί εφαρμόστηκαν με σκοπό την οδήγηση του αυτοκινήτου σε μεγαλύτερο μέρος της πίστας προς εξοικονόμηση χρόνου, αλλά και προκειμένου οι πιθανοί χρήστες να εξασκηθούν με τη λογική της πολλαπλής εισαγωγής εντολών.

Παρακάτω παρουσιάζονται ενδεικτικά στιγμιότυπα από την οθόνη, που αφορούν το αποτέλεσμα της εισαγωγής διαφορετικών εντολών μέσω των κύβων για την κίνηση του αυτοκινήτου στην πίστα.



Εικόνα 4.11: Αποτέλεσμα εισαγωγής μεμονωμένου σχήματος-εντολής βέλους που αντιστοιχεί στην ευθεία κίνηση του αυτοκινήτου

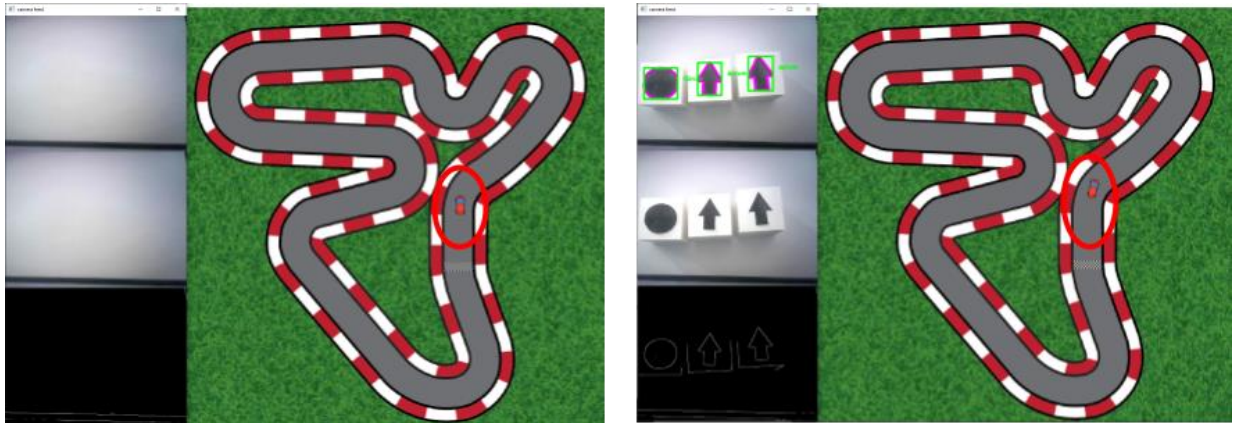
Όπως διαπιστώθηκε και παρουσιάζεται στην Εικόνα 4.11, μετά την εισαγωγή του σχήματος του βέλους που αντιστοιχεί στην ευθεία κίνηση, το αυτοκίνητο ορθώς έχει μετατοπιστεί σε άλλο σημείο της πίστας και συγκεκριμένα κάποια εικονοστοιχεία πιο μπροστά από την προηγούμενη θέση του. Επομένως το σύστημα αναγνωρίζει σωστά το σχήμα το όνομα του οποίου αναγράφεται στην οθόνη, μεταφράζει το σχήμα σε κίνηση του αυτοκινήτου και μετακινεί το αυτοκίνητο στο παιχνίδι.



Εικόνα 4.12: Αποτέλεσμα εισαγωγής συνδυασμού εντολών δεξιά και ευθεία μέσω των σχημάτων κύκλου και βέλους αντίστοιχα

Στη συνέχεια έγινε εισαγωγή δύο και τριών σχημάτων εντολών για να διαπιστωθεί η ανταπόκριση του συστήματος. Όπως φαίνεται στις εικόνες 4.12 και 4.13 το σύστημα

ανταποκρίνεται κανονικά. Πιο συγκεκριμένα στην εικόνα 4.12 γίνεται η εισαγωγή των εντολών ευθεία και δεξιά μέσω των σχημάτων βέλους και κύκλου. Όντως το αυτοκίνητο έχει μετακινηθεί σύμφωνα με τη σειρά εισαγωγής των εντολών. Το ίδιο ισχύει και στην εικόνα 4.13 όπου γίνεται εισαγωγή των εντολών.



Εικόνα 4.13: Αποτέλεσμα εισαγωγής τριών εντολών δεξιά-ευθεία-ευθεία μέσω των σχημάτων κύκλου-βέλους-βέλους αντίστοιχα

Αξίζει να αναφερθεί ότι εκτός από τα προηγούμενα δοκιμάστηκαν επιπλέον σενάρια με εισαγωγή τεσσάρων, αλλά και πέντε σχημάτων. Στην περίπτωση των τεσσάρων σχημάτων διαπιστώθηκε πως το σύστημα χάνει αρκετά την αξιοπιστία του καθώς δημιουργούνται προβλήματα σχετικά με την αναγνώριση των σχημάτων και την ανταπόκριση του παιχνιδιού. Λόγω του γεγονότος ότι η κάμερα καλύπτει συγκεκριμένο χώρο και η τοποθέτηση των σχημάτων μέσω των κύβων πρέπει να γίνεται σειριακά, διαπιστώθηκε πως το σύστημα δυσκολεύεται να αναγνωρίσει επιτυχώς όλα τα σχήματα (Εικόνα 4.14). Επιπλέον και στην περίπτωση που γίνει επιτυχής αναγνώριση μετά την εισαγωγή τεσσάρων σχημάτων, ο χρήστης είναι δύσκολο να προβλέψει και να υπολογίσει την πορεία του αυτοκινήτου χωρίς να συγκρουστεί με τα όρια της πίστας. Αυτό έχει ως αποτέλεσμα να απαιτούνται αρκετές διορθωτικές κινήσεις για την επαναφορά του αυτοκινήτου σε ορθή πορεία μειώνοντας έτσι την παραγωγικότητα του χρήστη. Επιπροσθέτως στην περίπτωση της εισαγωγής 5 σχημάτων, παρατηρήθηκε ότι λόγω του ύψους τοποθέτησης της κάμερας η εικόνα ανακόπτεται και τα εισαγόμενα σχήματα δεν αναγράφονται σωστά. Άρα κατά συνέπεια και όλη η διαδικασία της αναγνώρισης από το σύστημα παρεμποδίζεται (Εικόνα 4.15) και το παιχνίδι δεν αποκρίνεται όπως θα έπρεπε.

Ακόμη δοκιμάστηκε το σενάριο της αλλαγής του ύψους της τοποθέτησης της κάμερας έτσι ώστε να μπορεί να γίνει δυνατή η εισαγωγή μεγάλου σχετικά αριθμού σχημάτων.



Διαπιστώθηκε όμως πως το σύστημα αδυνατεί να εντοπίσει και κατά συνέπεια να αναγνωρίσει τα σχήματα που εισάγονται και χάνεται η αξιοπιστία, επομένως και αυτή η εναλλακτική απορρίφθηκε (Εικόνα 4.16).

Συμπερασματικά, μετά τις παραπάνω δοκιμές και πειραματισμούς, λήφθηκε το συμπέρασμα πως το σύστημα ανταποκρίνεται πληρέστερα με την εισαγωγή μέχρι και τριών κύβων με σχήματα. Σε άλλες περιπτώσεις κατά την εισαγωγή παραπάνω σχημάτων ή μεταβολής του ύψους της κάμερας το σύστημα δεν λειτουργεί με τον προβλεπόμενο και επιθυμητό τρόπο με αποτέλεσμα να παρεμποδίζονται οι λειτουργίες του.



Εικόνα 4.14: Πρόβλημα αναγνώρισης κατά την εισαγωγή τεσσάρων σχημάτων



Εικόνα 4.15: Περίπτωση εισαγωγής πέντε σχημάτων τα ακριανά των οποίων ανακόπτονται



*Εικόνα 4.16: Αποτυχημένη αναγνώριση σχημάτων μετά την τοποθέτηση της κάμερας σε μεγαλύτερο ύψος*

#### **4.4 Σύνοψη κεφαλαίου**

Σε αυτό το κεφάλαιο αναφέρθηκαν τα στάδια του πειραματισμού που αφορούσαν τα διάφορα στοιχεία του συστήματος. Παρουσιάστηκαν αναλυτικά τα βήματα που ακολουθήθηκαν για την επίτευξη του τελικού αποτελέσματος. Ειδικότερα αναφέρθηκαν οι πειραματισμοί που αφορούσαν τη μέθοδο αναγνώρισης σχημάτων υπό διαφορετικές συνθήκες περιβάλλοντος, την προσέγγιση της κατασκευής των κύβων και τέλος τη δοκιμή του γενικότερου συνόλου του συστήματος με διαφορετικά σενάρια εισαγωγής κύβων.

## Κεφάλαιο 5 – Επίλογος

### 5.1 Ανακεφαλαίωση

Η παρούσα διπλωματική εργασία πραγματεύεται τη δημιουργία και την ανάπτυξη ενός συστήματος με κύβους εκπαιδευτικού χαρακτήρα. Το σύστημα αποτελείται από τρία επιμέρους στοιχεία τα οποία περιλαμβάνουν ένα εικονικό παιχνίδι, ένα σύστημα αναγνώρισης σχημάτων μέσω κάμερας και τους κύβους.

Η υλοποίηση είχε ως κύριο σκοπό την εκμάθηση εννοιών που αφορούν τον προγραμματισμό και την πληροφορική σε μαθητές μικρής ηλικίας και ως στόχο την ευχάριστη εμπειρία των χρηστών σε ότι αφορά αυτή τη διαδικασία της εκμάθησης. Κύριοι χρήστες του συστήματος είναι ο εκάστοτε μαθητής και ο υπεύθυνος δάσκαλος. Ο μαθητής καλείται να αλληλεπιδρά άμεσα με το σύστημα με το να προγραμματίζει την κίνηση ενός αυτοκινήτου στο εικονικό παιχνίδι, μέσω της τοποθέτησης των κύβων μπροστά στην κάμερα. Οι κύβοι έχουν τοποθετημένα γεωμετρικά σχήματα στην κάθε πλευρά τους και αφού τα αντιληφθεί η κάμερα, το σύστημα τα ερμηνεύει σε κίνηση. Ο εκπαιδευτικός καλείται να στήσει το σύστημα και να επιβλέπει τη διαδικασία λειτουργίας του.

### 5.2 Μετρικά συστήματος

Το σύστημα αποτελείται από υλικό αλλά και από λογισμικό μέρος επομένως πρέπει να γίνει αναφορά στις παραμέτρους που αφορούν και χαρακτηρίζουν τα δύο αυτά μέρη. Στη συνέχεια παρουσιάζονται μετρήσεις που αφορούν τα δύο αυτά στοιχεία.

- Σε ότι αφορά τους κύβους οι διαστάσεις του κάθε ενός ξεχωριστά είναι στα 50x50x50 χιλιοστά και το βάρος τους ανέρχεται στα 25 γραμμάρια. Το κόστος του υλικού που αγοράστηκε για την κατασκευή τους ανέρχεται στα 20 ευρώ και το κόστος για την κατασκευή τους μεμονωμένα εκτιμάται γύρω στα 2 με 3 ευρώ.
- Τα αρχεία που δημιουργήθηκαν και αφορούν το λογισμικό του συστήματος είναι 2 και οι γραμμές του πηγαίου κώδικα ανέρχονται στις 325. Για τις μετρήσεις αυτές χρησιμοποιήθηκε το εργαλείο CLOC<sup>1</sup> και δεν συμπεριλήφθηκαν τα σχόλια και οι κενές γραμμές.
- Οι τεχνικές προδιαγραφές του υπολογιστή που πραγματοποιήθηκαν η δημιουργία της εργασίας, οι δοκιμές και οι πειραματισμοί είναι: Επεξεργαστής Intel Core i5-

---

<sup>1</sup> <https://codetabs.com/count-loc/count-loc-online.html>

3470 3.2 GHz, μνήμη RAM 16 GB, κάρτα γραφικών AMD Radeon HD 6670, μέγεθος οθόνης 21”, έκδοση windows W10 64-bit.

- Οι τεχνικές προδιαγραφές της κάμερας που χρησιμοποιήθηκε είναι: μοντέλο κάμερας Lamtech HD USB Webcam 720p, κύριο τσιπ ελέγχου Lingyang Innovation 2072, τσιπ κάμερας GC0307, φυσικά pixel κάμερας 30W, τύπος διασύνδεσης USB2.0
- Κατά την εκτέλεση του συστήματος το ποσοστό του υπολογιστικού φορτίου του επεξεργαστή (CPU load) ανέρχεται στο γύρω στο 30% στην περίπτωση που το παιχνίδι ρυθμίζεται να λειτουργεί στα 60 FPS (frames per second) και γύρω στο 20% εάν ρυθμίζεται στα 15. Η διαφορά των FPS δεν παίζει ιδιαίτερο ρόλο στην συνολική εμπειρία του παιχνιδιού, αλλά διευκολύνει τη γενικότερη λειτουργία του υπολογιστή καθώς η εκκίνηση του συστήματος διαρκεί γύρω στα 3 δευτερόλεπτα με το παιχνίδι ρυθμισμένο στα 60 και γύρω στο 1.5 ρυθμισμένο στα 15 FPS αντίστοιχα.
- Ο εντοπισμός του σχήματος κατά την εισαγωγή του γίνεται απευθείας και η ανταπόκριση του παιχνιδιού με τη μετάφραση του σχήματος σε κίνηση διαρκεί κάτω από 2 δευτερόλεπτα

### 5.3 Ανάλυση SWOT

Σε αυτή την ενότητα ακολουθεί η ανάλυση SWOT του υλοποιηθέντος συστήματος με τους κύβους, δηλαδή η εξέταση των δυνατών σημείων (Strengths), των αδυναμιών (Weaknesses), των ευκαιριών (Opportunities) αλλά και ενδεχομένως ορισμένων απειλών (Threats). Η αναφορά των παραπάνω γίνεται σε συνοπτικές λίστες οι οποίες βοηθούν στην βαθύτερη κατανόηση και την εξαγωγή συμπερασμάτων.

Δυνατά σημεία (Strengths):

- Πρωτότυπος τρόπος εκμάθησης ο οποίος συνδυάζει τεχνικές μηχανικής όρασης και απτού προγραμματισμού.
- Άμεσο οπτικό αποτέλεσμα του απτού προγραμματισμού στην οθόνη προς αντίληψη και κατανόηση από τον χρήστη.
- Απλός και εύχρηστος τρόπος προγραμματισμού μέσω των κύβων, αντιληπτός από χρήστες όλων των ηλικιών και ιδιαίτερα νεαρής ηλικίας.

- Ακρίβεια εντολών – κινήσεων αυτοκινήτου στο παιχνίδι.
- Ανοιχτού κώδικα προκειμένου οποιοσδήποτε να μπορεί να τροποποιήσει και να υλοποιήσει το σύστημα.

#### Αδυναμίες (Weaknesses):

- Για την εκτέλεση του συστήματος απαιτούνται ορισμένες συνθήκες στο περιβάλλον που εκτελείται.
- Απαιτείται συνεχής έλεγχος κατά την διαδικασία αναγνώρισης των σχημάτων για την ορθή λειτουργία της.

#### Ευκαιρίες (Opportunities):

- Δημιουργία ενός γενικότερου πλαισίου εναλλακτικού τρόπου προγραμματισμού για την ένταξη περισσότερων μέσων εισαγωγής εντολών μέσω μηχανικής όρασης.
- Επανασχεδιασμός του παιχνιδιού ώστε να γίνει ομαλότερη η κινούμενη εικόνα και η αλληλεπίδραση με τα στοιχεία της πίστας.
- Δημιουργία διαφορετικού παιχνιδιού με πολυπλοκότερους μηχανισμούς για την εισαγωγή εντολών μέσω μηχανικής όρασης.

#### Απειλές (Threats):

- Ύπαρξη πλήθους παρόμοιων συστημάτων (απτού και μη) προγραμματισμού για παιδιά.
- Πιθανή έλλειψη κατανόησης των λειτουργιών από τους χρήστες μικρών ηλικιών λόγω δυσκολίας σε κάποιο μηχανισμό του συστήματος.

## 5.4 Μελλοντικές επεκτάσεις

Στο σύστημα με τους κύβους απτού προγραμματισμού που αναπτύχθηκε και παρουσιάστηκε στην παρούσα διπλωματική εργασία, υπάρχουν δυνατότητες για πολλές μελλοντικές ενδεχομένως επεκτάσεις. Οι επεκτάσεις αυτές ενδέχεται να πραγματοποιούνται προσθήκες νέων λειτουργιών και τροποποιήσεις ήδη υπαρχόντων. Παρακάτω παρουσιάζονται ενδεικτικά κάποιες από αυτές τις πιθανές επεκτάσεις.

- Μία προσθήκη αρχικά στο σύστημα, θα μπορούσε να γίνει στα πλαίσια της ανάπτυξης του παιχνιδιού. Για παράδειγμα θα μπορούσε να δημιουργηθεί μία εικονική πίστα να μεν ενδεχομένως πιο απλή και με τους ίδιους μηχανισμούς,

αλλά εμπλουτισμένη με διάφορα είδη εμποδίων τα οποία ο χρήστης θα καλείται να αποφύγει. Η αλληλεπίδραση με το κάθε εμπόδιο μπορεί να είναι διαφορετική κάνοντας έτσι την πίστα και κατά προέκταση το παιχνίδι πιο ενδιαφέρον και ελκυστικό.

- Ακόμη μία προέκταση σε σχέση με το παιχνίδι ενδεχομένως να ήταν η δημιουργία ενός παιχνιδιού διαφορετικής φιλοσοφίας, παραδείγματος χάρη ενός παιχνιδιού ρόλων (RPG). Ο χρήστης θα κατείχε έναν εικονικό χαρακτήρα, ενδεχομένως σε έναν λαβύρινθο, και θα έπρεπε να τον καθοδηγήσει με εντολές κίνησης στο στόχο αλληλεπιδρώντας παράλληλα με το περιβάλλον του.
- Μία πολύ καλή προσθήκη στο κομμάτι της μηχανικής όρασης θα ήταν αυτή μεγαλύτερης γκάμας εντολών. Μέσω μίας διαφορετικής μεθόδου αναγνώρισης σχημάτων από την κάμερα ο χρήστης θα μπορούσε να επιλέξει ανάμεσα σε μία μεγαλύτερη γκάμα σχημάτων (όπως ρόμβος ή πολύγωνο) τα οποία θα εκτελούν πιο πολύπλοκες εντολές όπως επαναλήψεις κινήσεων ή στροφή υπό συγκεκριμένες μοίρες.
- Άλλη μία επέκταση στο ήδη υπάρχον σύστημα αφορά την προσθήκη ενός εικονικού αντιπάλου και αντίστοιχων επιπέδων. Ανάλογα με το επίπεδο δυσκολίας ο χρήστης θα καλείται να κερδίσει ένα αντίστοιχο αυτοκίνητο διαφορετικού χρώματος, το οποίο θα οδηγείται από τεχνητή νοημοσύνη αυτόματα, σε έναν αγώνα ταχύτητας. Αυτή η λειτουργία θα προσέδιδε μία νότα ανταγωνιστικού χαρακτήρα στο παιχνίδι κάνοντάς το έτσι και περισσότερο ενδιαφέρον.
- Τέλος μία ακόμη επέκταση, αφορά την κατασκευή των κύβων. Για παράδειγμα θα μπορούσε να τοποθετηθούν αισθητήρες ή κωδικοί QR στις πλευρές των κύβων οι οποίοι θα προσδιόριζαν και τις εντολές. Ανάλογα με ποια πλευρά εντόπιζε η κάμερα θα εκτελούσε και την αντίστοιχη εντολή. Οι εντολές αυτές θα μπορούσαν να είναι πιο πολύπλοκες και να επεξεργάζονται και να εκτελούνται πιο γρήγορα από το σύστημα.

## 5.5 Συμπεράσματα

Το σύστημα που υλοποιήθηκε απαρτίζεται από περαιτέρω υποσυστήματα και είναι πολυδιάστατο. Συνεπώς η ομαλή λειτουργία του εξαρτάται και από την ορθή λειτουργία αυτών των υποσυστημάτων. Με γεγονός το ότι στην παρούσα του μορφή διατηρεί μία αξιοπιστία, είναι βέβαιο πως σε ορισμένα σημεία μπορεί να δεχθεί αναβαθμίσεις και σημαντικές τροποποιήσεις. Παρ' όλα αυτά, σύμφωνα με όσα αναφέρθηκαν και αναλύθηκαν

σχετικά με την παρούσα διπλωματική εργασία, το αναφερθέν σύστημα κύβων επιτυγχάνει το σκοπό του. Παρουσιάζει μία εναλλακτική μέθοδο μάθησης και εκπαίδευσης σε όρους σχετικά με την πληροφορική και τον προγραμματισμό.

Η εφαρμογή τεχνικών μηχανικής όρασης για τον εντοπισμό των σχημάτων στις πλευρές των κύβων, η επεξεργασία των οποίων θα οδηγήσει στην εκτέλεση εντολών, αποτελεί μία διαφορετική και καινοτόμα προσέγγιση. Οι εντολές αυτές σχεδιάστηκε να εκτελούνται σε ακολουθία και να αφορούν κινήσεις. Δεν εξετάστηκαν πιο σύνθετες δομές προγραμματισμού καθώς θα ήταν δύσκολο να κατανοηθούν και να εφαρμοστούν από χρήστες νεαρών ηλικιών στους οποίους απευθύνεται το σύστημα. Επιπλέον, ο συνδυασμός της εκτέλεσης αυτής των εντολών με ένα διαδραστικό εικονικό παιχνίδι προσδίδει μία ευχάριστη και διασκεδαστική νότα στη συνολική εμπειρία χρήσης χωρίς να χάνεται ο εκπαιδευτικός του χαρακτήρας.

Παιδιά και μαθητές μικρών κυρίως ηλικιών μπορούν να χρησιμοποιήσουν το σύστημα ώστε να κατανοήσουν και να εμπεδώσουν τις προαναφερθείσες έννοιες, καθώς και να πάρουν μία ιδέα για τον προγραμματισμό με ένα ευχάριστο και οικείο για αυτά τρόπο. Τέλος, κάθε εκπαιδευτικός και όχι μόνο, έχει τη δυνατότητα να χρησιμοποιήσει και να υλοποιήσει το παρόν σύστημα εφαρμόζοντάς το στη δικιά του τάξη.

## Παράρτημα: Οδηγίες Εγκατάστασης

Σε αυτό το σημείο γίνεται αναφορά στη διαδικασία της εγκατάστασης των λογισμικών που χρησιμοποιήθηκαν και αναπτύχθηκαν στην παρούσα διπλωματική εργασία. Το λογισμικό αναπτύχθηκε σε περιβάλλον Windows 10, και τα προαπαιτούμενα πακέτα και βιβλιοθήκες για την εκτέλεσή του σε υπολογιστή με λειτουργικό σύστημα Windows είναι:

- Εγκατάσταση περιβάλλοντος Visual Studio Code<sup>2</sup>
- Εγκατάσταση της γλώσσας προγραμματισμού Python<sup>3</sup>
- Εγκατάσταση της βιβλιοθήκης OpenCV<sup>4</sup> από ένα τερματικό (π.χ. cmd) μέσω της εντολής “pip install opencv-python”
- Για την εγκατάσταση της βιβλιοθήκης Pygame από ένα τερματικό εκτέλεση της εντολής “pip install pygame”
- Εγκατάσταση του συστήματος διαχείρισης εκδόσεων git<sup>5</sup>
- Από ένα τερματικό (π.χ. cmd) εκτέλεση της εντολής “git clone <https://github.com/nicklatsis/diploma-thesis.git>”

Το σύστημα μπορεί να εκτελεστεί εφόσον βρίσκονται όλα τα αρχεία εικόνων και τα αρχεία με κατάληξη .py στον τον ίδιο φάκελο. Με την εκτέλεση του αρχείου project.py πυροδοτείται η εκκίνηση του συστήματος.

Στην περίπτωση που στην εκκίνηση του συστήματος, κατά την κλήση της βιβλιοθήκης pygame προκύψει κάποιο σφάλμα, ο χρήστης προτείνεται να υποβαθμίσει την βιβλιοθήκη κατεβάζοντας μία παλαιότερη έκδοσή της.

---

<sup>2</sup> <https://code.visualstudio.com/download>

<sup>3</sup> <https://www.python.org/downloads/>

<sup>4</sup> <https://opencv.org/releases/>

<sup>5</sup> <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>



## Βιβλιογραφία

- [1] Ishii H., Ullmer B., Tangible bits: Towards seamless interfaces between people, bits and atoms, Toronto Canada. pp. 234-241 (1997)
- [2] Dourish, P., Where the action is: the foundations of embodied interaction. Cambridge, MA: MIT press, (2004)
- [3] Hornecker E., Buur J., Getting a grip on tangible interaction: a framework on physical space and social interaction., Montréal, Québec, Canada pp. 437-446 (2006)
- [4] Orit Shaer, Eva Hornecker., "Tangible User Interfaces: Past, Present, and Future Directions", Foundations and Trends® in Human-Computer Interaction: vol. 3: no. 1-2, pp 4-137, (2009).
- [5] Claire O'Malley, Danae Stanton Fraser., Literature Review in Learning with Tangible Technologies. A NESTA Futurelab Research report (2004)
- [6] Baer RA, Smith GT, Lykins E, Button D, Krietemeyer J, Sauer S, Walsh E, Duggan D, Williams JM. Construct validity of the five faced mindfulness questionnaire in meditating and nonmeditating samples., pp.329-342, (2008)
- [7] Resnick M., Ocko S. & Papert, S., LEGO, Logo, and design. Children's Environments Quarterly, pp.14-18. (1988)
- [8] Zuckerman, O. Arida S. & Resnick M., Extending tangible interfaces for education: Digital montessori-inspired manipulatives., Portland, Oregon, USA. pp. 859-868., (2005)
- [9] Blikstein, Paulo. 'Multimodal Learning Analytics'. Proceedings of the Third International Conference on Learning Analytics and Knowledge, ACM Digital Library, pp. 102-06., (2013)
- [10] Chiu, Chi-Yue, and Ying-yi Hong. Social Psychology of Culture. Psychology Press, (2014).
- [11] Keating, Avril and Gabriella Melis., 'Youth Attitudes Towards Their Future: The Role of Resources, Agency and Individualism in the UK', vol. 5, no. 1, pp. 1-18, (2022)
- [12] Μακρή Αγορίτσα, Βλαχόπουλος Δημήτρης, Ηλεκτρονική μάθηση: η πολυσημία και πολυπλοκότητα της έννοιας, Διεθνές Συνέδριο για την Ανοικτή & εξ Αποστάσεως Εκπαίδευση, vol. 9, pp. 133-147, (2017)
- [13] Καρδάμη Δήμητρα, Η χρήση της επαυξημένης πραγματικότητας στην εκπαίδευση, Πολυτεχνική Σχολή Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πάτρα (2020).

- [14] Kan Zheng. (2016). *Tangible Interaction*. [Online]. Medium. Last Updated: November 10 2016. Available at: [https://medium.com/@wszk1992\\_34208/tangible-interaction-4b388df60dfe](https://medium.com/@wszk1992_34208/tangible-interaction-4b388df60dfe) [Accessed October 2022].
- [15] Itoh, Y., Akinobu, S., Ichida, H., Watanabe, R., Kitamura, Y., & Kishino, F. TSU. MI. KI: Stimulating children's creativity and imagination with interactive blocks, pp. 62-70, (2004)
- [16] McNerney T., Tangible computation bricks: Building-blocks for physical microworlds, (2001)
- [17] Kelleher C. & Pausch, R., Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, vol.37, no.2, pp.83-137, (2005)
- [18] Jeffrey Cockburn, Anne G.E. Collins and Michael J. Frank, A Reinforcement Learning Mechanism Responsible for the Valuation of Free Choice, vol. 83, no. 3, pp. 551–57, (2014).
- [19] Smith A. C., Using magnets in physical blocks that behave as programming objects, pp. 147-150, (2007)
- [20] Marshall, P., Do tangible interfaces enhance learning?, UK, pp. 163-170, (2007)
- [21] Patten, J., Griffith, L. & Ishii H., A tangible interface for controlling robotic toys., Hague, The Netherlands. pp. 277-278, (2000)
- [22] Horn M. S. & Jacob R. J. K., Tangible programming in the classroom with tern, Medford, USA, pp.2-6, (2007)
- [23] Kwon D., Kim, H., Shim J. & Lee W., Algorithmic bricks: A tangible robot programming tool for elementary school students., vol.55, no.4, (2012)
- [24] Cockburn A. & Bryant A. Leogo: An equal opportunity user interface for programming., vol. 8, no.5-6, pp.601-619, (1997)
- [25] Gallardo-Gallardo Eva & Dries Nicky & González-Cruz Tomás, The meaning of 'talent' in the world of work, pp.290-300, (2013)
- [26] Sapounidis, T. & Demetriadis, S., Tangible versus graphical user interfaces for robot programming: Exploring cross-age children's preferences. *Personal and Ubiquitous Computing*, vol. 17, no.8, pp.1775-1786, (2013)
- [27] Camp Tracy, Generation CS: The Growth of Computer Science., vol. 8, no. 2, pp. 44–50, (2017)

- [28] Jin, Qiao, Danli Wang, Xiaozhou Deng, Nan Zheng, Steve Chiu, AR-Maze: A Tangible Programming Tool for Children Based on AR Technology', Trondheim, Norway, pp.611–16, (2018)
- [29] Koushik, Varsha, Darren Guinness, and Shaun K. Kane, StoryBlocks, A Tangible Programming Game to Create Accessible Audio Stories, Department of Computer Science University of Colorado Boulder, USA, pp. 1–12, (2018)
- [30] Cecily Morrison, Nicolas Villar, Anja Thieme, Zahra Ashktorab, Eloise Taysom, Oscar Salandin, Daniel Cletheroe, Greg Saul, Alan F Blackwell, Darren Edge, Martin Grayson & Haiyan Zhang, Torino: A Tangible Programming Language Inclusive of Children with Visual Disabilities', vol. 35, no. 3, pp. 191–239, (2020).
- [31] Peta Wyeth & Purchase H. C., Programming without a computer: A new interface for children under eight. Paper presented at the User Interface Conference, AUIC First Australasian, pp. 141-148, (2000)
- [32] Μπούτσικας Χρήστος. (2020). *Σύστημα απτικής διεπαφής χρήστη με βάση ευφρείς κύβους: Ανάπτυξη*. Θεσσαλονίκη: Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης Σχολή Θετικών Επιστημών. pp.9-13.
- [33] Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer* vol.38, no.9, pp. 25– 32.
- [34] Γεωργίου- Παππάς Κωνσταντίνος, Καλιακούδας Μάριος. (2016). *Μηχανική Όραση και Τεχνολογικά Συστήματα Απεικόνισης Οφθαλμού*. Αίγιο: Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας. pp.11-12.
- [35] Gholamreza Anbarjafari. (2014). *Digital Image Processing*. [Online]. University of Tartu. Available at: <https://sisu.ut.ee/imageprocessing/book/1> [Accessed September 2022].
- [36] Wikipedia. (2007). *Color Model*. [Online]. Wikipedia. Available at: [https://en.wikipedia.org/wiki/Color\\_model](https://en.wikipedia.org/wiki/Color_model) [Accessed September 2022]
- [37] Wikipedia. (2020). *RGB color model*. [Online]. Wikipedia. Available at: [https://en.wikipedia.org/wiki/RGB\\_color\\_model](https://en.wikipedia.org/wiki/RGB_color_model) [Accessed September 2022].
- [38] Federal Agencies Digital Guidelines Initiative. (2022). *Thresholding*. [Online]. Digitization Guidelines. Available at: <https://www.digitizationguidelines.gov/term.php?term=thresholding> [Accessed October 2022].
- [39] Siddharth Misra, Hao Li and Jiabo He. (2019). *Machine Learning for Subsurface Characterization*. United States: Gulf Professional Publishing. pp.289-314. <https://doi.org/10.1016/B978-0-12-817736-5.00010-7>

- [40] Wikipedia. (2022). *Canny edge Detector*. [Online]. Wikipedia. Available at: [https://en.wikipedia.org/wiki/Canny\\_edge\\_detector#Development](https://en.wikipedia.org/wiki/Canny_edge_detector#Development) [Accessed October 2022].
- [41] Tatsiana Isakava. (2022). *Image processing techniques in computer vision*. [Online]. InData Labs. Available at: <https://indatalabs.com/blog/image-processing-techniques-in-computer-vision> [Accessed October 2022].
- [42] Megha Goyal. (2011). Morphological Image Processing. *International Journal of Computer Science & Technology.*, vol.2, no.4, pp.161-165. [Online]. Available at: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d0201e7e6de417ff683a3b4e853fdd4f9b39b> [Accessed October 2022].
- [43] MathWorks. (2022). *Image processing techniques for image analysis*. [Online]. MathWorks. Available at: <https://www.mathworks.com/discovery/image-analysis.html> [Accessed October 2022].
- [44] Meghana Kshirsagar, Gauri Vaidya. (2021). *Preprocessing with Computer Vision Part VI: Contour Detection*. [Online]. Medium. Available at: <https://medium.com/perspectivesondata-science/preprocessing-with-computer-vision-part-vi-contour-dete> [Accessed October 2022].
- [45] Wikipedia, the free encyclopedia. (2015). *Visual Studio Code*. [Online]. Wikipedia. Available at: [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code) [Accessed October 2022].
- [46] Coursera. (2022). *What Is Python Used For? A Beginner's Guide*. [Online]. Coursera. Available at: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> [Accessed 24 October 2022]
- [47] Pygame. (2022). *A Newbie Guide to pygame*. [Online]. Pygame. Last Updated: 2022. Available at: <https://www.pygame.org/docs/tut/newbieguide.html> [Accessed October 2022].
- [48] OpenCV. (2022). *About*. [Online]. OpenCV. Available at: <https://opencv.org/about/> [Accessed October 2022].
- [49] Amazon. (2022). *Ender 3 3d Printer*. [Online]. Amazon. Available at: <https://www.amazon.com/Official-Creality-3D-Printer-Source/dp/B07D218NX3> [Accessed November 2022].
- [50] Brady Gavin. (2018). *What is Sketchup (and How Do I Use It)*. [Online]. How-To Geek. Available at: <https://www.howtogeek.com/364232/what-is-sketchup/> [Accessed November 2022].

[51] Matt Jani. (2022). *Cura Settings Decoded – An Ultimaker Cura Tutorial*. [Online]. All3dp. Last Updated: 2022. Available at: <https://all3dp.com/1/cura-tutorial-software-slicer-cura-3d/> [Accessed November 2022].