



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Υποστήριξη της διδακτικής διαδικασίας του μαθήματος
«Εφαρμογές Πληροφορικής»**

Γεωργία Π. Γιώτη

Κωνσταντίνα Π. Χατζηελευθερίου

**Επιβλέποντες: Μαρία Γρηγοριάδου, Ομότιμη Καθηγήτρια
Ηλίας Βεργίνης, Διδάκτωρ ΕΚΠΑ**

ΑΘΗΝΑ

ΙΟΥΝΙΟΣ 2017

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υποστήριξη της διδακτικής διαδικασίας του μαθήματος «Εφαρμογές Πληροφορικής»

Γεωργία Π. Γιώτη
A.M.: 1115201200026

Κωνσταντίνα Π. Χατζηλευθερίου
A.M.: 1115201200198

Επιβλέποντες: **Μαρία Γρηγοριάδου, Ομότιμη Καθηγήτρια**
Ηλίας Βεργίνης, Διδάκτωρ ΕΚΠΑ

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας πτυχιακής εργασίας είναι η υποστήριξη της διδακτικής και μαθησιακής διαδικασίας του μαθήματος επιλογής «Εφαρμογές Πληροφορικής», που διδάσκεται στην Α΄ τάξη του Γενικού Λυκείου. Στο πλαίσιο της συγκεκριμένης πτυχιακής σχεδιάσαμε δραστηριότητες και φύλλα εργασίας. Με τα φύλλα αυτά εργασίας υποστηρίξαμε το μάθημα «Εφαρμογές Πληροφορικής» το οποίο διεξήχθη στο Πρότυπο ΓΕΛ της Βαρβακείου Σχολής, με υπεύθυνο καθηγητή τον κ. Βεργίνη Ηλία όπου παρακολούθησαμε και υποστηρίξαμε και τα αντίστοιχα εβδομαδιαία εργαστήρια. Η δομή της παρούσας πτυχιακής είναι η παρακάτω:

Στο πρώτο κεφάλαιο, παρουσιάζεται το πρόγραμμα σπουδών του μαθήματος καθώς και οι ιδιαίτερες απαιτήσεις και οι στόχοι του.

Στο δεύτερο κεφάλαιο παρουσιάζονται ιστορικά στοιχεία για το πρόγραμμα Alice και γίνεται αναφορά σε ορισμένες έννοιες του αντικειμενοστραφούς προγραμματισμού. Επιπλέον παρουσιάζονται δραστηριότητες και φύλλα εργασίας, τα οποία χρησιμοποιήθηκαν με στόχο να μάθουν οι μαθητές εύκολα και ευχάριστα αντικειμενοστραφή προγραμματισμό.

Στο τρίτο κεφάλαιο, παρουσιάζεται μία εισαγωγή στα περιβάλλοντα Pivot και Game Maker, τα οποία δεν αποτελούν μέρος του προγράμματος σπουδών, καθώς και δραστηριότητες οι οποίες βοήθησαν στην κατανόηση της λειτουργίας του εκάστοτε περιβάλλοντος.

Στο τέταρτο κεφάλαιο, παρατίθεται μια εισαγωγή στην οποία γίνεται μία αναδρομή της εξέλιξης του προγράμματος Appinventor. Όμοια με το δεύτερο κεφάλαιο, σχεδιάσαμε φύλλα εργασίας, με σκοπό την εξοικείωση των μαθητών με την ανάπτυξη εφαρμογών.

Ακόμη, στο πέμπτο κεφάλαιο, γίνεται μια σύνοψη της παρούσας πτυχιακής εργασίας και παρατίθενται ορισμένα συμπεράσματα που προέκυψαν μετά από την ενασχόληση των μαθητών με τα προαναφερθέντα περιβάλλοντα.

Τέλος, το ΠΑΡΑΡΤΗΜΑ I περιλαμβάνει μία αξιολόγηση των μαθητών σχετικά με το περιβάλλον Alice αλλά και τον τρόπο διδασκαλίας του, ενώ στα ΠΑΡΑΡΤΗΜΑΤΑ II και III παρουσιάζονται αντίστοιχα οι εργασίες των μαθητών και οι χριστουγεννιάτικες κάρτες που δημιούργησαν.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Διδακτική διαδικασία, προγραμματισμός, εκπαιδευτικά περιβάλλοντα μάθησης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αντικειμενοστραφής προγραμματισμός, Photoshop, Alice, GameMaker, Appinventor

ABSTRACT

The aim of this BA thesis, is to support the educational process of the High School first grade optional course "IT Applications". Within this BA thesis, we designed activities and worksheets. With these worksheets, we supported the course "IT Applications", which was conducted in the Varvakio High School, with Professor Verginis Ilias, where we attended and supported the weekly lessons. The structure of this BA thesis is the following:

The first chapter, presents the course curriculum and its specific requirements and objectives.

The second chapter provides some historical information about the Alice educational environment. Furthermore, certain concepts of the object-oriented programming are presented. We designed activities and worksheets so that students can learn easily and pleasantly object-oriented programming.

The third chapter presents an introduction to the Pivot and Game Maker environments, which are not part of the curriculum, in addition we designed activities that helped students in understanding the function of each one.

In the fourth section there is an introduction to the Appinventor program development. Similar to the second chapter, we designed worksheets, in order to familiarize students with the development of applications.

Also, the fifth chapter provides some conclusions about the abovementioned environments.

Finally, ANNEX I contains a students' evaluation on Alice environment and their opinion on the way of teaching, whereas ANNEX II and III present students' work on Alice environment and some Christmas cards which they created.

SUBJECT AREA: Educational process, programming, educational learning environments

KEYWORDS: Object-oriented programming, Photoshop, Alice, GameMaker, Appinventor

ΣΤΙΣ ΟΙΚΟΓΕΝΕΙΕΣ ΜΑΣ

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα εργασία διενεργήθηκε στο πλαίσιο των μαθημάτων Πτυχιακή Εργασία I και Πτυχιακή Εργασία II του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Για την εκπόνηση της συγκεκριμένης εργασίας, με αφορμή το μάθημα του προγράμματος σπουδών της Α' Λυκείου, Εφαρμογές Πληροφορικής, φιλοξενηθήκαμε στο Πρότυπο ΓΕΛ της Βαρβακείου Σχολής, από τη διευθύντρια κα. Σερέτη Βάλια, την οποία κι ευχαριστούμε θερμά.

Ευχαριστούμε πολύ την επιβλέπουσα Ομότιμη Καθηγήτρια κ. Γρηγοριάδου Μαρία του ΕΚΠΑ για την εμπιστοσύνη που μας έδειξε δίνοντάς μας τη δυνατότητα να υλοποιήσουμε την εργασία μας.

Θα θέλαμε να απευθύνουμε θερμές ευχαριστίες στον Καθηγητή Πληροφορικής του ΓΕΛ Βαρβακείου Σχολής, Βεργίνη Ηλία , για τις πολύτιμες γνώσεις και συμβουλές που μας παρείχε καθ' όλη τη διάρκεια της χρονιάς, καθώς και το αμείωτο ενδιαφέρον που έδειξε, σε συνδυασμό με την πολυετή έρευνα του και τις γνώσεις που μας προσέφερε, διότι χωρίς αυτές δε θα ήταν δυνατή η ολοκλήρωση της εργασίας μας.

Συνοψίζοντας, ευχαριστούμε πολύ τους προαναφερθέντες για την πολύτιμη συμβολή τους στην ομαλή διεξαγωγή της όλης διαδικασίας, καθώς και για την άκρως ενδιαφέρουσα και ξεχωριστή εμπειρία του να διδάξουμε σε μία τάξη.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΡΟΛΟΓΟΣ	15
ΕΙΣΑΓΩΓΗ.....	16
1. ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΚΑΙ ΔΟΜΗ ΜΑΘΗΜΑΤΟΣ	17
1.1 Αναλυτικό Πρόγραμμα Σπουδών	17
1.2 Δομή Μαθήματος	19
1.3 Ημερολόγιο Μαθημάτων.....	20
2. ΕΙΣΑΓΩΓΗ ΣΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ALICE – ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΚΑΙ ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ	22
2.1 Δραστηριότητες και Φύλλα Εργασίας [3]	25
2.1.1 1 ^ο Φύλλο εργασίας	25
2.1.2 Ενδεικτικές απαντήσεις στο 1ο φύλλο εργασίας	31
2.1.3 2 ^ο Φύλλο εργασίας	32
2.1.4 1 ^η Δραστηριότητα	34
2.1.5 2 ^η Δραστηριότητα	34
2.1.6 3 ^η Δραστηριότητα	35
2.1.7 4 ^η Δραστηριότητα	35
2.1.8 Φύλλο εργασίας για το διαγώνισμα τετραμήνου	37
3. ΕΠΙΠΛΕΟΝ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΚΑΙ ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ ΣΕ ΆΛΛΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ	39
3.1 Εισαγωγή στο προγραμματιστικό περιβάλλον Pivot Animator	39
3.1.1 Δραστηριότητες και Φύλλα Εργασίας	39
3.1.1.1 1 ^ο Φύλλο εργασίας	39
3.1.1.2 1 ^η Δραστηριότητα	40
3.1.1.3 2 ^η Δραστηριότητα	40
3.1.1.4 3 ^η Δραστηριότητα	40
3.2 Εισαγωγή στο προγραμματιστικό περιβάλλον Game Maker	40
3.2.1 Δραστηριότητες και Φύλλα Εργασίας	44
3.2.1.1 1 ^ο Φύλλο εργασίας	44
3.2.1.2 1 ^η Δραστηριότητα	49
4. ΕΙΣΑΓΩΓΗ ΣΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ APPINVENTOR – ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΚΑΙ ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ	58
4.1 Δραστηριότητες και Φύλλα Εργασίας.....	59
4.1.1 1 ^ο Φύλλο εργασίας [12]	59

4.1.2	2 ^ο Φύλλο εργασίας [15]	71
4.1.3	3 ^ο Φύλλο Εργασίας [15]	81
4.1.4	4 ^ο Φύλλο εργασίας [15]	85
4.1.5	5 ^ο Φύλλο Εργασίας [15]	89
4.1.6	6 ^ο Φύλλο Εργασίας [15]	98
4.1.7	1 ^η Δραστηριότητα [15]	104
5.	ΣΥΜΠΕΡΑΣΜΑΤΑ	107
	ΠΑΡΑΡΤΗΜΑ Ι	108
	ΠΑΡΑΡΤΗΜΑ ΙΙ	119
	ΠΑΡΑΡΤΗΜΑ ΙΙΙ	130
	ΑΝΑΦΟΡΕΣ	134

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Βασική οθόνη περιβάλλοντος Alice 3.2.....	23
Εικόνα 2: Λίστα (σχηματικά)	25
Εικόνα 3: Δομή επιλογής (σχηματικά).....	25
Εικόνα 4: Επιλογή περιβάλλοντος για τον κόσμο που θα δημιουργήσουμε	26
Εικόνα 5: Δημιουργία μεταβλητής.....	26
Εικόνα 6: Αρχικοποίηση μεταβλητής	26
Εικόνα 7: Δημιουργία του μπλοκ If	27
Εικόνα 8: Δημιουργία λίστας.....	29
Εικόνα 9: Δημιουργία λίστας (συνέχεια).....	29
Εικόνα 10: Προσθήκη αντικειμένων στη λίστα	30
Εικόνα 11: Εντολή if.....	30
Εικόνα 12: Εντολή else.....	30
Εικόνα 13: Το αντικείμενο Alice στο τέλος.....	32
Εικόνα 14: Το αντικείμενο Alice αρχικά.....	32
Εικόνα 15: Τα αντικείμενα της σκηνής	33
Εικόνα 16: Το σκηνικό της δραστηριότητας	35
Εικόνα 17: Σκηνή frame 1	37
Εικόνα 18: Σκηνή frame 2.....	38
Εικόνα 19: Σκηνή frame 3.....	38
Εικόνα 20: Σκηνή frame 4	38
Εικόνα 21: Αντικείμενο realistic_man.....	39
Εικόνα 22: Σκηνή 3 ^{ης} δραστηριότητας	40
Εικόνα 23: Μενού επιλογών και μπάρα εργαλείων	41
Εικόνα 24: Πόροι του παιχνιδιού	41
Εικόνα 25: Παράθυρο ιδιοτήτων γραφικού στοιχείου.....	42
Εικόνα 26: Παράθυρο ιδιοτήτων προγραμματιζόμενου αντικειμένου	42
Εικόνα 27: Παράθυρο ιδιοτήτων των δωματίου	43
Εικόνα 28: Τελική μορφή παιχνιδιού.....	44
Εικόνα 29: Προσθήκη των sprites.....	45
Εικόνα 30: Δημιουργία των objects (1)	45
Εικόνα 31: Δημιουργία των objects (2)	45
Εικόνα 32: Δημιουργία πίστας	47
Εικόνα 33: Η πίστα	47
Εικόνα 34: Προσθήκη φόντου.....	47
Εικόνα 35: Προσθήκη ήχου	48

Εικόνα 36: Ο λαβύρινθος.....	50
Εικόνα 37: Προσθήκη ιδιοτήτων	51
Εικόνα 38: Ο λαβύρινθος μέχρι τώρα	52
Εικόνα 39: Προσθήκη ήχου	53
Εικόνα 40: Οθόνη εκκίνησης παιχνιδιού	54
Εικόνα 41: Score παιχνιδιού.....	55
Εικόνα 42: I have a dream application 1	59
Εικόνα 43: Άνοιγμα Browser.....	59
Εικόνα 44: Starter app.....	60
Εικόνα 45: Open starter app.....	60
Εικόνα 46: Algin.....	61
Εικόνα 47: Image property.....	62
Εικόνα 48: Button text.....	62
Εικόνα 49: Labels	63
Εικόνα 50: Player component.....	63
Εικόνα 51: Block editor	63
Εικόνα 52: Button.Click event handler	64
Εικόνα 53: Player1.Start block.....	64
Εικόνα 54: I have a dream application 2.....	65
Εικόνα 55: SaveAs	65
Εικόνα 56: Adding images 1	66
Εικόνα 57: Adding images 2	66
Εικόνα 58: Second player component	67
Εικόνα 59: Screen title and background color	67
Εικόνα 60: Start/Pause code	68
Εικόνα 61: If/Else start/pause code	68
Εικόνα 62: Duplicate event handler	69
Εικόνα 63: Changing Blocks 1	69
Εικόνα 64: Changing blocks 2	70
Εικόνα 65: Properties Screen1 AppInventor.....	71
Εικόνα 66 : Background Color	71
Εικόνα 67: Property Scrollable.....	71
Εικόνα 68: Οθόνη Designer.....	72
Εικόνα 69: Select Background Image.....	73
Εικόνα 70: Upload Image	73
Εικόνα 71: Build-in πλακίδια	75

Εικόνα 72: Πλακίδια που έχουν εισαχθεί	75
Εικόνα 73: Οθόνη Blocks.....	76
Εικόνα 74: when.____.Touchdown.....	76
Εικόνα 75: call.____.Play	76
Εικόνα 76: Συναρμολόγηση πλακιδίων.....	77
Εικόνα 77: Πλακίδιο κενού κειμένου	77
Εικόνα 78: TouchDown.....	77
Εικόνα 79: TouchUp	77
Εικόνα 80: Συντεταγμένες σημείου επαφής	78
Εικόνα 81: call____.PointInDirection	78
Εικόνα 82: x συντεταγμένη	78
Εικόνα 83: Προσδιορισμός Ταχύτητας.....	79
Εικόνα 84: Ολοκλήρωση εντολής TouchDown	79
Εικόνα 85: Ολοκλήρωση εντολής TouchUp.....	79
Εικόνα 86: Εφαρμογή Guard Dog.....	80
Εικόνα 87: Εφαρμογή Guard Dog-AngryDog.....	80
Εικόνα 88: Αποτελέσματα καρτέλας Blocks.....	82
Εικόνα 89: Βοηθητική εικόνα 3ου φύλλου εργασίας	82
Εικόνα 90: Δήλωση global μεταβλητής.....	83
Εικόνα 91: Αρχικοποίηση global μεταβλητής.....	83
Εικόνα 92: Αποτελέσματα καρτέλας Blocks(2)	83
Εικόνα 93: Αποτελέσματα καρτέλας Blocks(2a)	84
Εικόνα 94: Τελική εικόνα εφαρμογής HitTheMole.....	84
Εικόνα 95: Σχεδιασμός application	85
Εικόνα 96: Προγραμματισμός Blocks 1	86
Εικόνα 97: Προγραμματισμός Blocks 2	87
Εικόνα 98: Αποτέλεσμα (α).....	87
Εικόνα 99: Αποτέλεσμα (β).....	88
Εικόνα 100: Βοηθητικό υλικό	89
Εικόνα 101 : Σχεδίαση	90
Εικόνα 102 : Προγραμματισμός (α)	90
Εικόνα 103 : Προγραμματισμός (β)	91
Εικόνα 104 : Προγραμματισμός (γ).....	91
Εικόνα 105 : - Προγραμματισμός (ε).....	92
Εικόνα 106 : Προγραμματισμός (στ).....	92
Εικόνα 107 : Προγραμματισμός (ζ).....	93

Εικόνα 108 : Προγραμματισμός (η).....	93
Εικόνα 109 : Αποτέλεσμα (α).....	95
Εικόνα 110 : Αποτέλεσμα (β).....	96
Εικόνα 111 : Αποτέλεσμα (γ).....	97
Εικόνα 112 : Αποτελέσματα.....	100
Εικόνα 113: Βοηθητική εικόνα	101
Εικόνα 114 : Προγραμματισμος (α)	101
Εικόνα 115 : Προγραμματισμος (β)	101
Εικόνα 116 : Γεγονός ενέγεια (α)	102
Εικόνα 117: Γεγονός ενέγεια (β)	102
Εικόνα 118 : Αποτέλεσμα (α).....	102
Εικόνα 119 : Αποτέλεσμα (β).....	103
Εικόνα 120 – Σχεδίαση (α) Δραστηριότητας 9	104
Εικόνα 121 - Σχεδίαση (β) Δραστηριότητας 9.....	104
Εικόνα 122 - Σχεδίαση (γ) Δραστηριότητας 9	105
Εικόνα 123 - Σχεδίαση (δ) Δραστηριότητας 9.....	105
Εικόνα 124 - Προγραμματισμός (α) Δραστηριότητας 9.....	105
Εικόνα 125: Ερωτήσεις αξιολόγησης 1-7.....	108
Εικόνα 126: Ερωτήσεις αξιολόγησης 8-16.....	109
Εικόνα 127: Απαντήσεις στις ερωτήσεις 1-2.....	110
Εικόνα 128 :Απαντήσεις στις ερωτήσεις 3-4.....	111
Εικόνα 129: Απαντήσεις στις ερωτήσεις 5-6.....	112
Εικόνα 130: Απαντήσεις στις ερωτήσεις 7-8.....	113
Εικόνα 131: Απαντήσεις στις ερωτήσεις 9-10.....	114
Εικόνα 132: Απαντήσεις στις ερωτήσεις 11-12.....	115
Εικόνα 133: Απαντήσεις στις ερωτήσεις 13-14.....	116
Εικόνα 134 Απάντηση στην ερώτηση 15.....	117
Εικόνα 135: Απάντηση στην ερώτηση 16	117
Εικόνα 136: Ομαδική εργασία 1.....	119
Εικόνα 137: Ομαδική εργασία 2.....	119
Εικόνα 138: Ομαδική εργασία 3.....	120
Εικόνα 139: Ομαδική εργασία 4.....	120
Εικόνα 140: Ομαδική εργασία 5.....	121
Εικόνα 141: Ομαδική εργασία 6.....	121
Εικόνα 142: Ομαδική εργασία 7.....	122
Εικόνα 143: Ομαδική εργασία 8.....	122

Εικόνα 144: Ομαδική εργασία 9.....	123
Εικόνα 145: Ομαδική εργασία 10.....	123
Εικόνα 146: Ομαδική εργασία 11.....	124
Εικόνα 147: Ομαδική εργασία 12.....	124
Εικόνα 148: Ομαδική εργασία 13.....	125
Εικόνα 149: Ομαδική εργασία 14.....	125
Εικόνα 150: Ομαδική εργασία 15.....	126
Εικόνα 151: Ομαδική εργασία 16.....	126
Εικόνα 152: Ομαδική εργασία 17.....	127
Εικόνα 153: Ομαδική εργασία 18.....	127
Εικόνα 154: Ομαδική εργασία 19.....	128
Εικόνα 155: Ομαδική εργασία 20.....	128
Εικόνα 156: Ομαδική εργασία 21.....	129
Εικόνα 157: Ομαδική εργασία 22.....	129
Εικόνα 158: Πίνακας και κάρτα 1.....	130
Εικόνα 159: Πίνακας και κάρτα 2.....	130
Εικόνα 160: Πίνακας και κάρτα 3.....	131
Εικόνα 161: Πίνακας και κάρτα 4.....	131
Εικόνα 162: Πίνακας και κάρτα 5.....	132
Εικόνα 163: Πίνακας και κάρτα 6.....	132
Εικόνα 164: Πίνακας και κάρτα 7.....	132
Εικόνα 165: Πίνακας και κάρτα 8.....	133

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1 : Πρόγραμμα Σπουδών	17
Πίνακας 2 : Ημερολόγιο μαθημάτων	20

ΠΡΟΛΟΓΟΣ

Σκοπός αυτής της πτυχιακής εργασίας είναι η υποστήριξη του μαθήματος επιλογής «Εφαρμογές Πληροφορικής» της Α΄ τάξης του Γενικού Λυκείου μέσω της ανάπτυξης, εφαρμογής και αξιολόγησης δραστηριοτήτων που αξιοποιούν διδακτικά τα εκπαιδευτικά περιβάλλοντα Alice και Appinventor στα οποία δόθηκε ιδιαίτερη έμφαση. Επιπλέον, αναπτύξαμε δραστηριότητες στα προγραμματιστικά περιβάλλοντα Pivot και Game Maker, τα οποία, αν και δεν αποτελούν μέρος του προγράμματος σπουδών αποφασίσαμε σε συνεργασία με τον υπεύθυνο καθηγητή, να τα εντάξουμε στο πλαίσιο της διδασκαλίας του μαθήματος. Σ' αυτό, οδηγηθήκαμε λόγω του υπέρμετρου ζήλου των μαθητών για τέτοιου είδους προγράμματα και τις αντίστοιχες δραστηριότητες τους. Τέλος, για να δώσουμε μία ευχάριστη νότα κατά την διάρκεια του σχολικού έτους, ενόψει των Χριστουγεννιάτικων εορτών, δώσαμε στα παιδιά την ευκαιρία να δημιουργήσουν τις δικές τους κάρτες χρησιμοποιώντας το περιβάλλον Photoshop.

Καθ' όλη τη διάρκεια του σχολικού έτους, υποστηρίξαμε το μάθημα με την εβδομαδιαία παρουσία μας, όπου γινόταν η παρουσίαση του θεωρητικού μέρους του μαθήματος σε συνεργασία με τον υπεύθυνο καθηγητή και η εκπόνηση από τους μαθητές των δραστηριοτήτων που είχαμε αναπτύξει. Επιπλέον, παρουσιάσαμε στα παιδιά έννοιες που αφορούν την εξέλιξη του διαδικτύου και της Πληροφορικής εν γένει

ΕΙΣΑΓΩΓΗ

Με δεδομένη την αλματώδη ανάπτυξη των τεχνολογιών Πληροφορικής και Επικοινωνιών τα τελευταία χρόνια και τη διείσδυσή τους στις περισσότερες εκφάνσεις της καθημερινής μας ζωής, κρίνεται απαραίτητη η μύηση των νέων στον προγραμματισμό, τις νέες τεχνολογίες και τις εφαρμογές τους, καθώς και η εξοικείωση τους με αυτές. Ταυτόχρονα, γίνεται όλο και πιο συχνή η ανάγκη να κρατηθεί αμείωτο το ενδιαφέρον των μαθητών. Ένας τρόπος για να επιτευχθεί το συγκεκριμένο είναι η ενεργή εμπλοκή τους σε δραστηριότητες, οι οποίες έχουν ως στόχο αρχικά, την αξιοποίηση των υπάρχοντων γνώσεών τους, σε συνδυασμό με την ανάπτυξη νέων δεξιοτήτων, όπως η κριτική σκέψη, η συνεργατικότητα, η λήψη αποφάσεων, η αυτοπαρακολούθηση και η αυτορύθμιση, με την ταυτόχρονη υποστήριξή τους από κατάλληλα εκπαιδευτικά εργαλεία. Σε αυτήν την κατεύθυνση, προσπαθήσαμε κι αναπτύξαμε σχετικές δραστηριότητες, οι οποίες αφορούν την απόκτηση νέων γνώσεων και δεξιοτήτων σε εισαγωγικά μαθήματα προγραμματισμού. Βασικός μας στόχος ήταν η ενεργή εμπλοκή των μαθητών με τρόπο τέτοιο, ώστε να μπορούν να εφαρμόσουν αυτές τις νεοαποκτηθείσες γνώσεις και στη συνέχεια να είναι ικανοί να αναπτύξουν δικά τους προγράμματα. ^[13]

Η χρήση των ηλεκτρονικών παιχνιδιών και η δημιουργία κινούμενων γραφικών για την αφήγηση μιας ιστορίας ως εκπαιδευτικό εργαλείο για τη διδασκαλία του προγραμματισμού έχει πολλά πλεονεκτήματα, ενώ προσθέτουν ένα στοιχείο διασκέδασης και δημιουργικότητας στον προγραμματισμό. Σύμφωνα με τον Overmars (2004), οι μαθητές που δημιουργούν ηλεκτρονικά παιχνίδια μπορεί να έχουν περισσότερα κίνητρα για να δουν το τελικό προϊόν του παιχνιδιού, σε αντίθεση με μια τυπική εργασία προγραμματισμού, επειδή τα παιχνίδια είναι περισσότερο διαδραστικά και ελκυστικά οπτικά.

Στα ηλεκτρονικά παιχνίδια και στις ιστορίες, το περιβάλλον, οι χαρακτήρες και τα σκηνικά είναι αντικείμενα (objects). Σε κάθε τέτοιο αντικείμενο, ο δημιουργός του προσθέτει ορισμένες συμπεριφορές. Η αξιοποίηση, λοιπόν, της σχεδίασης και ανάπτυξης τέτοιου είδους δραστηριοτήτων ως εκπαιδευτικό εργαλείο, εμπλέκει άμεσα τους μαθητές στη διαδικασία του αντικειμενοστραφούς προγραμματισμού. Οι μαθητές εξάλλου παίρνουν άμεσα οπτική ανατροφοδότηση για την πρόοδό τους, καθώς δημιουργούν το παιχνίδι ή την ιστορία που επιθυμούν. Η συγκεκριμένη ανατροφοδότηση, τους επιτρέπει να εντοπίζουν και να επιλύουν γρήγορα τα σφάλματα τους. Τελικά, οι μαθητές έχουν ένα απτό, πραγματικό παιχνίδι, το οποίο μπορούν να παίξουν με τους φίλους ή την οικογένειά τους, ή να το μοιραστούν στο Διαδίκτυο και να το εξελίξουν όσο αυτοί επιθυμούν. Το ίδιο μπορεί να συμβεί, τέλος, με οποιαδήποτε ιστορία σκεφτούν οι ίδιοι.

1. ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΚΑΙ ΔΟΜΗ ΜΑΘΗΜΑΤΟΣ

1.1 Αναλυτικό Πρόγραμμα Σπουδών

Το μάθημα «Εφαρμογές Πληροφορικής» αποτελεί ένα από τα μαθήματα επιλογής της Α' τάξης του Γενικού Λυκείου. Το γεγονός ότι πρόκειται για ένα μάθημα επιλογής σημαίνει ότι η διδασκαλία του θα πρέπει να προσαρμοστεί με τέτοιο τρόπο, έτσι ώστε η διεξαγωγή του να είναι σχετικά ευχάριστη πάντα όμως μέσα στο πλαίσιο του απαιτητικού αναλυτικού προγράμματος σπουδών.

Λαμβάνοντας υπόψιν το αναλυτικό πρόγραμμα σπουδών Πληροφορικής του Λυκείου, όπως αυτό παρουσιάζεται και στην ιστοσελίδα του υπουργείου ^[13], τις ταχύτατες αλλαγές στο χώρο των τεχνολογιών της Πληροφορίας και των Επικοινωνιών, και έχοντας ως στόχο την ενίσχυση των γνώσεων και των δεξιοτήτων Πληροφορικής που έχουν αποκτηθεί στις προηγούμενες τάξεις του Γυμνασίου, συμμετείχαμε στη διεξαγωγή του μαθήματος επιλογής της Α' τάξης «Εφαρμογές Πληροφορικής», για το σχολικό έτος 2016-2017.

Το αναλυτικό πρόγραμμα σπουδών του μαθήματος φαίνεται στον παρακάτω πίνακα :

Πίνακας 1 : Πρόγραμμα Σπουδών

ΘΕΜΑΤΙΚΗ ΕΝΟΤΗΤΑ	ΚΕΦΑΛΑΙΑ	ΠΕΡΙΓΡΑΦΗ	ΕΝΔΕΙΚΤΙΚΕΣ ΩΡΕΣ ΔΙΔΑΣΚΑΛΙΑΣ
2. Προγραμματιστικά περιβάλλοντα- Δημιουργία εφαρμογών	Κεφάλαιο 7 Υλοποίηση εφαρμογών σε προγραμματιστικά περιβάλλοντα	Οι μαθητές αναμένεται ότι θα έχουν στοιχειώδεις γνώσεις προγραμματισμού, από το Δημοτικό και το Γυμνάσιο, κυρίως μέσα από Logo-like περιβάλλοντα. Η ενότητα αυτή έρχεται να επεκτείνει τις γνώσεις των μαθητών και να τους δώσει την ευκαιρία να γνωρίσουν και άλλα εκπαιδευτικά περιβάλλοντα μέσα από ενδεικτικές δραστηριότητες ανάπτυξης μικροεφαρμογών.	16

3. Επικοινωνία και Διαδίκτυο	<p>Κεφάλαιο 9 (9.3) Από τον Web 1.0 στον Web X.0</p> <p>Κεφάλαιο 10 Υπηρεσίες κι εφαρμογές διαδικτύου</p> <p>Κεφάλαιο 11 Εισαγωγή στην HTML</p>	<p>Η ενότητα αυτή έχει ως στόχο οι μαθητές να εμβαθύνουν στις υπηρεσίες του Διαδικτύου και τις Web 2.0 εφαρμογές, να αναγνωρίζουν κώδικα HTML, να μπορούν να τον επεξεργαστούν και να τον ενσωματώσουν σε Διαδικτυακές εφαρμογές.</p>	18
4. Συνεργασία κι ασφάλεια στο διαδίκτυο	<p>Κεφάλαιο 13 Εφαρμογές νέφους</p> <p>Κεφάλαιο 14 (14.2) Επικοινωνία και συνεργασία από απόσταση</p> <p>Κεφάλαιο 15 Κοινωνικά Δίκτυα</p> <p>Κεφάλαιο 16 Ασφάλεια και προστασία στο διαδίκτυο</p>	<p>Η ενότητα αυτή έχει σκοπό να εισάγει τους μαθητές στη χρήση των εφαρμογών νέφους που προσφέρονται στο Διαδίκτυο για τη δημιουργία - διαχείριση εγγράφων και τη συνεργασία από απόσταση. Θέλουμε οι μαθητές να αποσαφηνίσουν τη λειτουργία του υπολογιστικού νέφους και να πειραματιστούν με τις υπηρεσίες του. Επίσης στην ενότητα αυτή εισάγονται θέματα κοινωνικών δικτύων, πνευματικών δικαιωμάτων, ασφάλειας και προστασίας στο Διαδίκτυο. Θέλουμε οι μαθητές να έρθουν σε επαφή με αντίστοιχες εφαρμογές, να γνωρίσουν τις επιπτώσεις από την κακή χρήση τους, να είναι σε θέση να</p>	16

		διαχειριστούν ανάλογα ζητήματα και να τα αξιολογήσουν.	
--	--	--	--

1.2 Δομή Μαθήματος

Σε συνεργασία με τον διδάσκοντα οργανώσαμε τον προγραμματισμό του μαθήματος με βάση τα μαθησιακά χαρακτηριστικά, τα ιδιαίτερα ενδιαφέροντα και το υπόβαθρο των μαθητών. Κάθε μάθημα είχε το αντίστοιχο θεωρητικό μέρος, μετά την ολοκλήρωσή του οποίου, οι μαθητές, ως επί το πλείστον χωρισμένοι σε ομάδες, αναλάμβαναν την ολοκλήρωση κάποιας δραστηριότητας. Με αυτόν τον τρόπο, καλλιεργήθηκε ένα πνεύμα αλληλοσεβασμού και ομαδικότητας ανάμεσα στους μαθητές. Επιπλέον, πριν από κάθε πρακτικό μέρος υπήρχε σχετική παρουσίαση της δραστηριότητας τόσο στον πίνακα, όσο και με προβολή σχετικού υλικού. Σε καμία περίπτωση δε ζητήθηκε από τους μαθητές να αποστηθίσουν τεχνικές λεπτομέρειες, καθώς και ιστορικές ή άλλου τύπου πληροφορίες που παρουσιάζονται στο σχολικό εγχειρίδιο. Αντιθέτως, επικεντρωθήκαμε στην κατανόηση των εννοιών και την εφαρμογή τους στο εργαστήριο. Με βάση λοιπόν το παραπάνω πρόγραμμα σπουδών, σκεφτήκαμε και δημιουργήσαμε εργασίες, είτε ατομικές είτε ομαδικές, οι οποίες εκπονήθηκαν από τους μαθητές υπό την μορφή φύλλων εργασίας. Στο πλαίσιο του μαθήματος, ενισχύθηκε η διερευνητική προσέγγιση, η αυτενέργεια και η συνεργατική μάθηση μέσω των ομαδικών δραστηριοτήτων που υλοποιήθηκαν. Ακόμη, αξίζει να τονίσουμε πως χρειάστηκε να εμπλουτίσουμε το πρόγραμμα σπουδών με επιπλέον περιβάλλοντα για την καλύτερη κατανόηση του προγραμματισμού, αφού το επίπεδο των μαθητών ήταν υψηλό, και υπήρχε άμεση ανταπόκριση σε ό,τι τους ζητούσαμε. Τέλος, αξιοσημείωτο είναι ότι μέρος των διαγωνισμάτων των τετράμηνων ήταν εργασίες προγραμματιστικού περιεχομένου. Οι δραστηριότητες που περιγράφονται στη συνέχεια έχουν αναπτυχθεί σύμφωνα με τις αρχές που διδάσκονται στο μάθημα «Διδακτική της Πληροφορικής» το οποίο προσφέρονταν ως μάθημα επιλογής του 8^{ου} εξαμήνου στο τμήμα Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ.

1.3 Ημερολόγιο Μαθημάτων

Πίνακας 2: Ημερολόγιο μαθημάτων

ΗΜΕΡΟΜΗΝΙΑ	ΠΕΡΙΛΗΨΗ ΜΑΘΗΜΑΤΟΣ
04/10/16	Γνωριμία με τους μαθητές
11/10/16	Εισαγωγή στο προγραμματιστικό περιβάλλον Alice
18/10/16	1 ^ο και 2 ^ο φύλλο εργασίας στο προγραμματιστικό περιβάλλον Alice
25/10/16	Ενότητα 3 Κεφάλαιο 9, 1 ^η και 2 ^η ομαδική δραστηριότητα στο προγραμματιστικό περιβάλλον Alice
01/11/16	3 ^η και 4 ^η ομαδική δραστηριότητα στο προγραμματιστικό περιβάλλον Alice
08/11/16	Ενότητα 3 Κεφάλαιο 10, παρουσίαση ομαδικών δραστηριοτήτων στο προγραμματιστικό περιβάλλον Alice
15/11/16	Εισαγωγή στο προγραμματιστικό περιβάλλον Pivot Animator, 1 ^ο φύλλο εργασίας και 1 ^η ομαδική δραστηριότητα
22/11/16	3 ^η και 4 ^η ομαδική δραστηριότητα στο προγραμματιστικό περιβάλλον Pivot Animator
29/11/16	Εισαγωγή στο προγραμματιστικό περιβάλλον Game Maker, 1 ^ο φύλλο εργασίας και 1 ^η ομαδική δραστηριότητα
06/12/16	Ενότητα 4 Κεφάλαιο 16, Χριστουγεννιάτικη εργασία στο Photoshop
20/12/16	Συνέχεια ομαδικών δραστηριοτήτων στο προγραμματιστικό περιβάλλον Game Maker
17/01/17	Δραστηριότητα για το διαγώνισμα Α' τετραμήνου στο προγραμματιστικό περιβάλλον Alice
24/01/17	Εισαγωγή στο προγραμματιστικό περιβάλλον Appinventor

31/01/17	Άσκηση στο περιβάλλον Appinventor
07/02/17	Εφαρμογή στο περιβάλλον Appinventor
14/02/17	Εφαρμογή στο περιβάλλον Appinventor
21/02/17	Εισαγωγή στην HTML
28/02/17	Ασκήσεις στην HTML
7/03/17	Ασκήσεις στην HTML
14/03/17	Ασκήσεις στην HTML
21/03/17	Ασκήσεις στην HTML

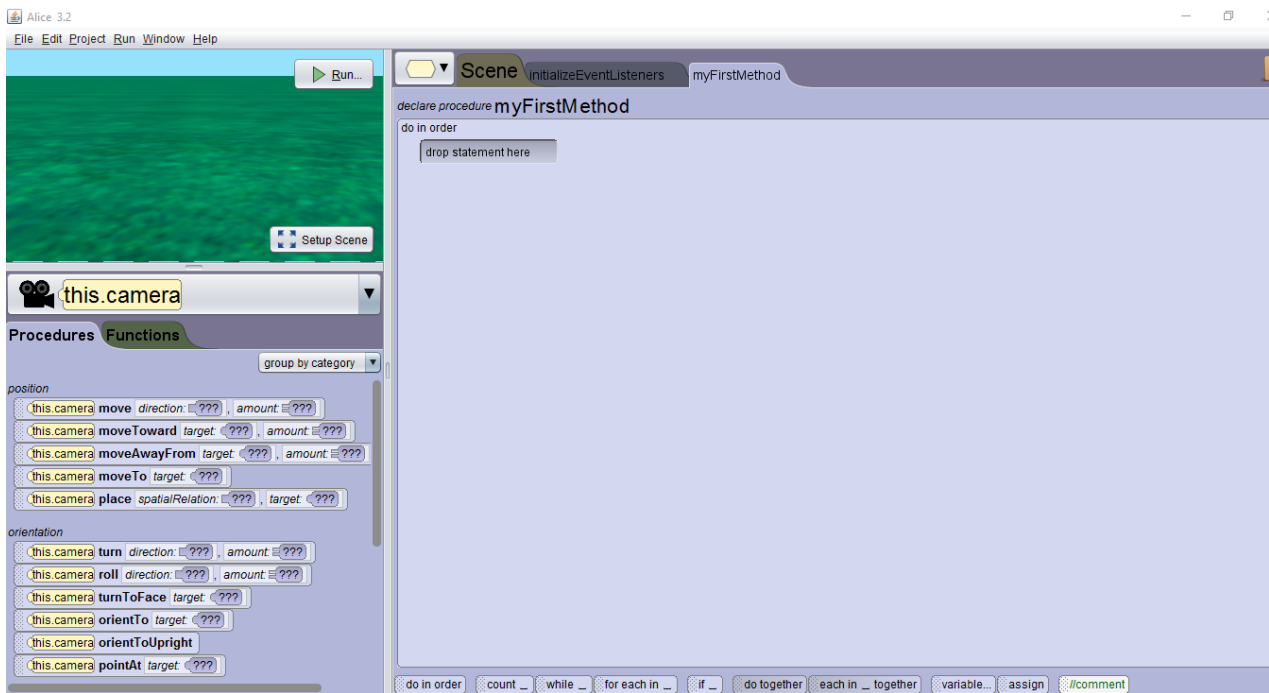
2. Εισαγωγή στο προγραμματιστικό περιβάλλον Alice – Δραστηριότητες και φύλλα εργασίας

Πρόκειται για ένα περιβάλλον οπτικού προγραμματισμού το οποίο αναπτύχθηκε το 1999 στο Πανεπιστήμιο Carnegie Mellon για εκπαιδευτικούς σκοπούς [6]. Παρέχει τη δυνατότητα στο χρήστη να δημιουργήσει κινούμενα γραφικά ή διαδραστικά παιχνίδια και βίντεο χωρίς αυτό να προϋποθέτει γνώσεις στον προγραμματιστικό τομέα. Το Alice δημιουργήθηκε για να διευκολύνει αρχάριους προγραμματιστές πάνω στον αντικειμενοστραφή προγραμματισμό. Συγκεκριμένα απλοποίησε τη διδασκαλία προγραμματισμού καθώς αντιμετώπισε τη συντακτική και σημασιολογική πολυπλοκότητα άλλων γλωσσών προγραμματισμού.

Μέσω ειδικών πλακιδίων – εντολών ο χρήστης μπορεί να δημιουργήσει αντικείμενα ή ακόμα κι έναν ολόκληρο τρισδιάστατο εικονικό κόσμο και στην συνέχεια να τον προγραμματίσει. Το περιβάλλον προσφέρει μια σειρά από έτοιμα αντικείμενα συνδυασμένα με συγκεκριμένες μεθόδους, συναρτήσεις και ιδιότητες με τη μορφή πλακιδίων, για να τα χρησιμοποιήσει ο χρήστης. Υποστηρίζεται από γλώσσες προγραμματισμού όπως η Java, C++, C# και έχει χρησιμοποιηθεί επιτυχώς από δασκάλους και μαθητές στη δευτεροβάθμια εκπαίδευση καθώς και σε ορισμένα Πανεπιστήμια.

Ο χρήστης στο συγκεκριμένο περιβάλλον έχει τη δυνατότητα να προγραμματίζει με μία πλήρως αντικειμενοστραφή γλώσσα χωρίς παρ' όλα αυτά να χρειάζεται να θυμάται εντολές και συντακτικούς κανόνες. Επιπλέον, προσφέρει τη δυνατότητα στο μαθητή να βλέπει άμεσα και διαδραστικά τα προγράμματα του φτιάχνει καθιστώντας τον ικανό να αντιλαμβάνεται τη σχέση μεταξύ των εντολών και των αντιδράσεων των αντικειμένων που έχει δημιουργήσει μέσω μιας κατάστασης που κατά τη Randy Pausch ονομάζεται «head fake». Ως «head fake» ορίζεται η κατάσταση κατά την οποία οι μαθητές ενθαρρύνονται να μάθουν κάτι θεωρώντας ότι μαθαίνουν κάτι διαφορετικό (Dann & Cooper, 2009). Αυτό έχει ως αποτέλεσμα οι μαθητές να αποκτούν μια πρώτη επαφή με βασικές προγραμματιστικές έννοιες.

Το Alice είναι διαθέσιμο για περιβάλλοντα Windows, Mac και Linux, όχι όμως στα ελληνικά [16]. Στη βασική οθόνη εμφανίζεται το μενού επιλογών μαζί με τα κουμπιά εκτέλεσης, αναίρεσης/επανάληψης και τον κάδο ανακύκλωσης. Το περιβάλλον είναι χωρισμένο σε περιοχές μερικές από τις οποίες αναφέρονται στα αντικείμενα, στις λεπτομέρειες τους και στον προγραμματισμό τόσο αντικειμένων όσο και γεγονότων. Επιπλέον, εμφανίζεται η περιοχή προεπισκόπησης του κόσμου που έχει δημιουργηθεί. Τέλος, είναι διαθέσιμη στον χρήστη η σκηνή που εμφανίζεται όταν εκτελείται το πρόγραμμα μέσω του Play. Ως προς το υλικό, το περιβάλλον Alice λειτουργεί άριστα σε ένα συμβατικό ηλεκτρονικό υπολογιστή με μνήμη RAM τουλάχιστον 2GB και κάρτα γραφικών τουλάχιστον 32bit βάθος χρώματος με ελάχιστη ανάλυση τα 1024x768 pixels.



Εικόνα 1: Βασική οθόνη περιβάλλοντος Alice 3.2

Το συγκεκριμένο περιβάλλον απευθύνεται σε αρχάριους χρήστες ή ακόμα και σε αυτούς που δεν έδειχναν ιδιαίτερο ενδιαφέρον για τον προγραμματισμό. Ως συνέπεια αυτού, δημιουργήθηκε το Storytelling Alice το οποίο αξιοποιώντας την αλληλεπίδραση δημιουργεί και αφηγείται ιστορίες ώστε να προσελκύσει το ενδιαφέρον των μαθητών.

Το προγραμματιστικό περιβάλλον Alice πέρα από την χρήση του σε εκπαιδευτικά προγράμματα στα σχολεία συνεργάστηκε με την Electronic Arts για την δημιουργία και την παραγωγή videogames. Χαρακτηριστικό παράδειγμα αποτελεί η συνεργασία του Alice με τους χαρακτήρες του παιχνιδιού Sims. Ύστερα από αυτή τη συνεργασία, το 2006 στο Πανεπιστήμιο Pittsburg του Carnegie Mellon έγινε μια νέα συμφωνία με την Electronic Arts η οποία είχε σκοπό την εξέλιξη και αναβάθμιση της χρήσης υπολογιστών στο εκπαιδευτικό σύστημα της Αμερικής. Μέσω αυτής της συμφωνίας, η Electronic Arts αποφάσισε να συμβάλει στην αναβάθμιση του Alice 3.0 παρέχοντας τους χαρακτήρες του παιχνιδιού Sims. Μετά την αναβάθμιση του Alice, το περιβάλλον έχει τις προδιαγραφές να είναι το βασικό εργαλείο διδασκαλίας της πληροφορικής στο εκπαιδευτικό σύστημα της Αμερικής κάνοντας ταυτόχρονα τον προγραμματισμό διαδραστικό και πιο διασκεδαστικό. Σύμφωνα με έρευνες που έγιναν διαπιστώθηκε ότι οι μαθητές όχι μόνο αυξήθηκαν στα μαθήματα που χρησιμοποιήθηκε το Alice αλλά ταυτόχρονα είχαν και καλύτερες επιδόσεις. Αυτό αποδόθηκε στο γεγονός ότι το περιβάλλον προγραμματισμού είναι εξαιρετικά προσιτό, εύχρηστο και διασκεδαστικό. Για αυτό το λόγο χρησιμοποιείται ευρέως σε σχολεία και πανεπιστήμια. Επιπλέον διαπιστώθηκε πως οι μαθητές κατανόησαν καλύτερα την έννοια της κληρονομικότητας του αντικειμενοστραφούς προγραμματισμού. Η επιτυχία του Alice οφείλεται στην οπτική αναπαράσταση των αντικειμένων. Οι μαθητές μπορούσαν να δουν και να σχετίσουν τα αντικείμενα με τις ενέργειες αναπτύσσοντας έτσι καλή διαίσθηση τόσο για αυτά, όσο και για τον αντικειμενοστραφή προγραμματισμό. Ακόμη, τους κατέστησε ικανούς να ελέγξουν την πολυπλοκότητα του προγράμματος που ανέπτυσσαν, εξοικειώνοντάς τους με την χρήση των αντικειμένων και των μεθόδων επίκλησής τους. Οι μαθητές είχαν τη δυνατότητα να παρακολουθήσουν τι πήγε στραβά στα προγράμματά τους και εύκολα να το διορθώσουν, μαθαίνοντας έτσι από τα λάθη τους. Παρ' όλα αυτά μετά από έρευνες των Parsons & Haden (2007) , διαπιστώθηκε πως ένα ποσοστό

μαθητών έχαναν την συγκέντρωσή τους εξαιτίας των στοιχείων του εικονικού κόσμου με αποτέλεσμα να μην εμβαθύνουν στις προγραμματιστικές έννοιες.

Οι ερευνητές συνιστούν ότι η Alice θα πρέπει να ενσωματωθεί στην διδασκαλία εισαγωγικών εννοιών του αντικειμενοστραφούς προγραμματισμού, προκειμένου να βελτιωθεί το επίπεδο συμμετοχής των μαθητών και η ικανότητά τους να αναπτύξουν μια διαισθητική κατανόηση των εννοιών, ειδικά της κληρονομικότητας, σε μια οπτική ανάδραση με το περιβάλλον. Εξαιτίας της ευρείας αποδοχής τους οι δημιουργοί του συνεχίζουν να το αναβαθμίζουν ώστε να γίνεται όλο και καλύτερο.

2.1 Δραστηριότητες και Φύλλα Εργασίας [3]

2.1.1 1^ο Φύλλο εργασίας

Ενδεικτική Διάρκεια: 45 λεπτά (1 διδακτική ώρα)

Διδακτικός στόχος

Σκοπός αυτού του φύλλου εργασίας είναι οι μαθητές να διδαχθούν την έννοια της λίστας και της δομής επιλογής. Αφού εξοικειωθούν με το πρόγραμμα, μέσω των δραστηριοτήτων που τους έχουμε αναθέσει να υλοποιήσουν, καλούνται να κατασκευάσουν οι ίδιοι μία δραστηριότητα. Δίνονται όσο το δυνατόν πιο ακριβείς οδηγίες για τον τρόπο κατασκευής του προγράμματος, έτσι δεν είναι απαραίτητο να έχουν ιδιαίτερες προγραμματιστικές γνώσεις.

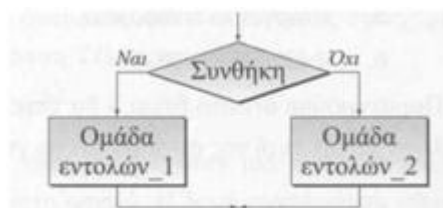
Τι είναι η λίστα;

Μία λίστα (ή ένας πίνακας σε άλλες γλώσσες προγραμματισμού) είναι ένα εργαλείο που μπορεί να χρησιμοποιηθεί για να αποθηκεύουμε πολλές πληροφορίες ταυτόχρονα. Μπορεί επίσης να οριστεί σαν μια μεταβλητή η οποία περιέχει πολλές άλλες μεταβλητές. Μια λίστα αποτελείται από αριθμούς ζευγαρωμένους με αντικείμενα και κάθε αντικείμενο μπορεί να ανακτηθεί από τον αριθμό – ζευγάρι του.



Τι είναι η δομή επιλογής;

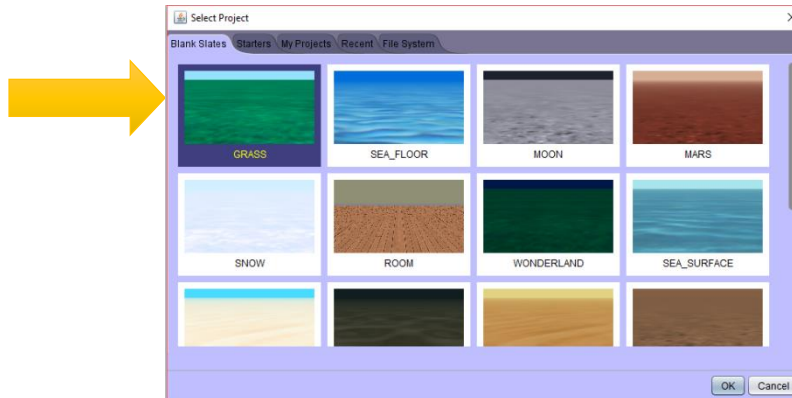
Πολλές φορές, ανάλογα με τα δεδομένα του προβλήματος καλούμαστε να πάρουμε αποφάσεις μέσα στον αλγόριθμο και να εκτελέσουμε κάποια συγκεκριμένα κομμάτια του αλγορίθμου και να αγνοήσουμε κάποια άλλα. Προκύπτει, λοιπόν, η έννοια της δομής επιλογής. Η δομή επιλογής είναι από τις βασικότερες δομές για την ανάπτυξη ενός αλγορίθμου. Χρησιμοποιείται για τη λήψη απόφασης μεταξύ δύο διαφορετικών καταστάσεων, εκ των οποίων η μία είναι αληθής και η άλλη ψευδής.



Εικόνα 3: Δομή επιλογής (σχηματικά)

1. Άνοιξε το πρόγραμμα Alice.

2. Επίλεξε να δημιουργήσεις τον αντίστοιχο κόσμο και πάτησε **Open**.



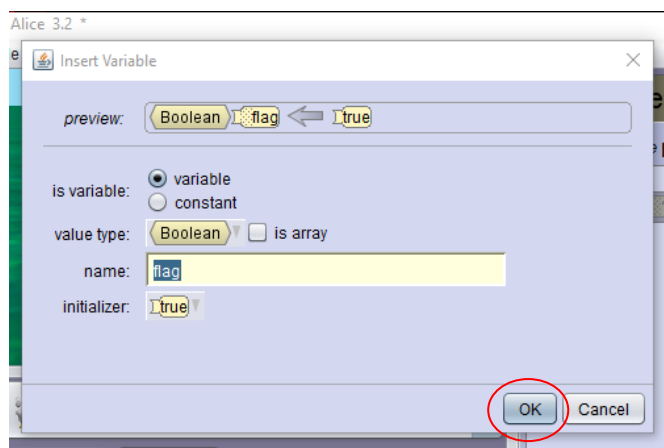
Εικόνα 4: Επιλογή περιβάλλοντος για τον κόσμο που θα δημιουργήσουμε

3. Στη συνέχεια επέλεξε **Setup Scene**

4. Ψάξε το αντικείμενο **Chicken** μέσω του **Search Gallery** και πρόσθεσέ το στον κόσμο.

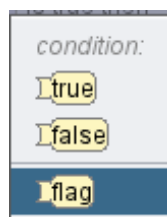
5. Επίλεξε **Edit Code**

6. Δημιούργησε μια μεταβλητή αφού σύρεις την μέθοδο **variable** στο κάτω μέρος της οθόνης όπως φαίνεται παρακάτω:



Εικόνα 5: Δημιουργία μεταβλητής

7. Από τον κατάλογο εντολών στο κάτω μέρος της σελίδας σύρε την εντολή **If/Else** και επέλεξε τα παρακάτω :



Εικόνα 6: Αρχικοποίηση μεταβλητής

8. Στον κατάλογο αντικειμένων επέλεξε το **this.Chicken**. Σύρε κάτω από την εντολή **If** από το μενού **procedure** την εντολή **say**. Αντίστοιχα κάτω από την εντολή **else** σύρε την εντολή **turn** επιλέγοντας το **LEFT**.



Εικόνα 7: Δημιουργία του μπλοκ If
Με αυτό τον τρόπο δημιουργήσαμε μια δομή επιλογής.

9. Πάτησε **Run** και δες τα αποτελέσματα των εντολών σου στον κόσμο.

Πειραμάτισου αλλάζοντας την τιμή της μεταβλητής **flag** από **true** σε **false**.

1. Τι παρατηρείς;

.....
.....
.....
.....
.....
.....

Εφαρμογή

ΒΗΜΑ 1

1. Με τον ίδιο τρόπο που δημιουργήσαμε το αντικείμενο `chicken` ψάξε το αντικείμενο **alien** μέσω του **Search Gallery** και πρόσθεσέ τον στον κόσμο. Επαναλαμβάνοντας αυτή τη διαδικασία 4 φορές δημιούργησε 4 ίδια αντικείμενα και τοποθέτησε τα το ένα δίπλα στο άλλο.

Παρατήρησε πως το κάθε αντικείμενο έχει διαφορετικό όνομα.

2. Να επαναλάβετε το βήμα 4 και 5 για να δημιουργήσεις μια μεταβλητή `flag` και την εντολή επιλογής.
3. Με τον ίδιο τρόπο που έκανες στο βήμα 6, πατώντας πάνω στο πρώτο αντικείμενο **this.alien**, σύρε κάτω από το **If** την εντολή **say** και επίλεξε **hello**. Αντίστοιχα, πατώντας πάνω στο αντικείμενο **this.alien2**, σύρε κάτω από το **Else** την εντολή **resizeHeight (2.0)**.
4. Πάτησε **Run**.

Πειραματίσου αλλάζοντας την τιμή της μεταβλητής `flag` από **true** σε **false**.

2. Τι παρατηρείς;

.....

.....

.....

.....

.....

.....

.....

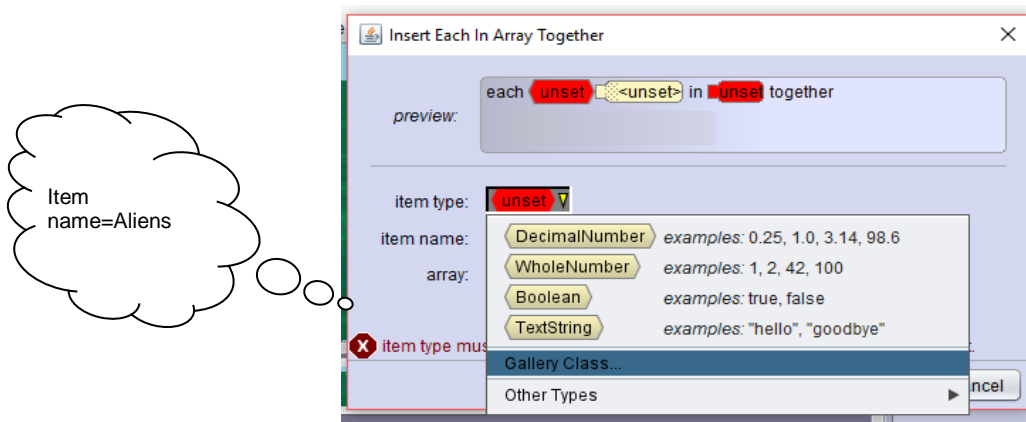
.....

ΒΗΜΑ 2

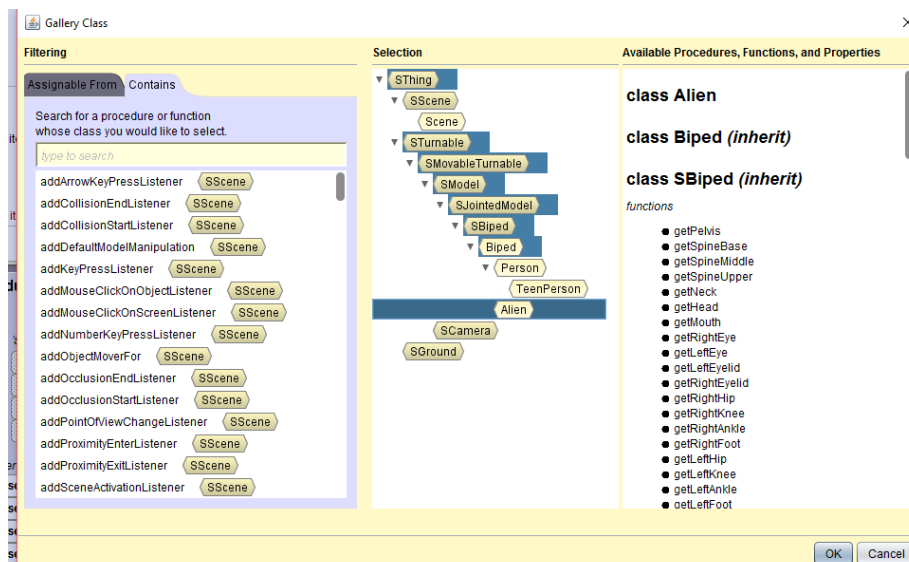
1. Διέγραψε όλο το κομμάτι εντολών **If/Else**.

Υποστήριξη της διδακτικής διαδικασίας του μαθήματος «Εφαρμογές Πληροφορικής»

2. Σύρε την εντολή **each_in_together** και δημιούργησε μία λίστα με όνομα Aliens, όπως φαίνεται παρακάτω:

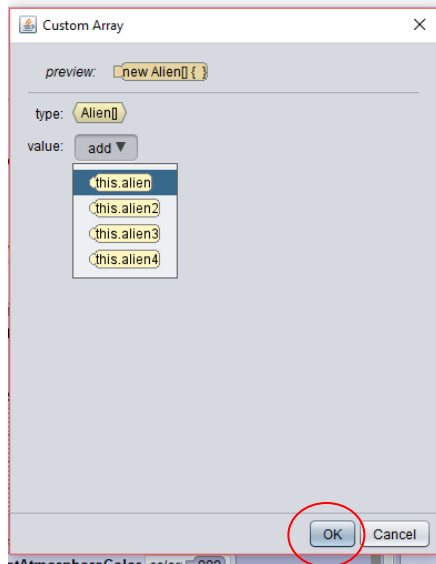


Εικόνα 8: Δημιουργία λίστας



Εικόνα 9: Δημιουργία λίστας (συνέχεια)

3. Πρόσθεσε όλα τα αντικείμενα και πάτησε OK :



Εικόνα 10: Προσθήκη αντικειμένων στη λίστα

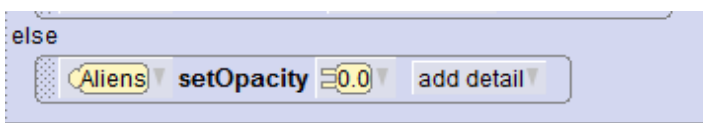
Με αυτό τον τρόπο δημιουργήσαμε μια λίστα από αντικείμενα.

4. Δημιούργησε μια καινούρια δομή επιλογής **If/Else** όπως στο βήμα 5 του παραδείγματος. Εδώ θα πρέπει να σύρεις την εντολή **if** στο πεδίο **drop statement** της μεθόδου **each_in_together**, που μόλις δημιούργησες.
5. Σύρε κάτω από το **If** την μέθοδο **turnToFace**, όπως φαίνεται παρακάτω:



Εικόνα 11: Εντολή if

6. Αντίστοιχα κάτω από το **Else** σύρε την μέθοδο **setOpacity**, όπως φαίνεται στην εικόνα:



Εικόνα 12: Εντολή else

7. Πάτησε **Run**.

Πειραμάτισου αλλάζοντας την τιμή της μεταβλητής **flag** από **true** σε **false**.

3. Τι παρατηρείς;

.....
.....
.....
.....
.....
.....
.....

4. Πώς θα μπορούσαν οι εντολές μέσα στην If/Else να εκτελούνται για ένα μόνο αντικείμενο, παρότι υπάρχει η λίστα αντικειμένων;

.....
.....
.....
.....
.....

2.1.2 Ενδεικτικές απαντήσεις στο 1ο φύλλο εργασίας

1.Τι παρατηρείς;

Όταν η μεταβλητή flag έχει την τιμή true εκτελείται το block εντολών κάτω από το If ενώ αν έχει την τιμή false εκτελούνται οι εντολές κάτω από το Else. Δηλαδή κάθε φορά εκτελείται μόνο ένα από τα δύο κομμάτια της δομής επιλογής.

2.Τι παρατηρείς;

Οι εντολές εκτελούνται μόνο για το ένα αντικείμενο alien και alien2 αντίστοιχα. Όταν η μεταβλητή flag έχει την τιμή true εκτελείται το block εντολών κάτω από το If ενώ αν έχει την τιμή false εκτελούνται οι εντολές κάτω από το Else. Δηλαδή κάθε φορά εκτελείται μόνο ένα από τα δύο κομμάτια της δομής επιλογής.

3.Τι παρατηρείς;

Σε αυτό το βήμα οι εντολές εκτελούνται για όλα τα αντικείμενα της κάθε λίστας αντίστοιχα. Όταν η μεταβλητή flag έχει την τιμή true εκτελείται το block εντολών κάτω από το If ενώ αν έχει την τιμή false εκτελούνται οι εντολές κάτω από το Else. Δηλαδή κάθε φορά εκτελείται μόνο ένα από τα δύο κομμάτια της δομής επιλογής.

4.Πώς θα μπορούσαν οι εντολές μέσα στην If/Else να εκτελούνται για ένα μόνο αντικείμενο, παρότι υπάρχει η λίστα αντικειμένων;

Οι εντολές συνεχίζουν να εκτελούνται μόνο για το ένα αντικείμενο παρόλο που έχουμε δημιουργήσει μια λίστα αντικειμένων, αν αντί για Aliens, στην 1η επιλογή των μεθόδων turnToFace και setOpacity, επιλεγεί ένα από τα αντικείμενα this.alien, this.alien2, this.alien3, this.alien4

2.1.3 2^ο Φύλλο εργασίας

Ενδεικτική Διάρκεια: 45 λεπτά (1 διδακτική ώρα)

Διδακτικός στόχος

Σκοπός αυτού του φύλλου εργασίας είναι η εξοικείωση των μαθητών με την έννοια της διαδικασίας και της συνάρτησης. Οι μαθητές να είναι σε θέση να μάθουν να συντάσσουν μία διαδικασία, να την καλούν σε κάθε πρόγραμμα κι επιπλέον να γνωρίσουν τις διαφορές της σε σχέση με την συνάρτηση.

Τι είναι συνάρτηση και τι είναι διαδικασία;

Το είδος κάθε υποπρογράμματος καθορίζεται από το είδος της λειτουργίας που επιτελεί. Υπάρχουν δύο κατηγορίες υποπρογραμμάτων, οι διαδικασίες και οι συναρτήσεις.

Η συνάρτηση είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομά της (όπως οι μαθηματικές συναρτήσεις).

Η διαδικασία είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος και να επιστρέφει πολλές τιμές και να παράγει πολλά μηνύματα. ^[2]

Εφαρμογή

1. Δημιουργήστε ένα νέο Έργο και δώστε του το όνομα «Gymnastics». Διαλέξτε για σκηνικό το γρασίδι (GRASS) και τοποθετήστε μέσα την Alice.
2. Στο χώρο του κώδικα δώστε τις εντολές που θα χρειαστούν ώστε τα χέρια του ανθρώπου να βρεθούν σε έκταση.

Θα πρέπει να χρησιμοποιήσετε τις κλειδώσεις των Ωμων (`getLeftShoulder`, `getRightShoulder`) και να φροντίσετε έτσι ώστε και τα δύο χέρια να σηκωθούν ταυτόχρονα.



Εικόνα 14: Το αντικείμενο Alice αρχικά



Εικόνα 13: Το αντικείμενο Alice στο τέλος

3. Κάντε μικρή έκταση των ποδιών όπως φαίνεται στην παραπάνω εικόνα. Για να γίνει αυτό θα χρειαστεί να εργασθείτε με τις κλειδώσεις του Ισχίου (`getLeftHip`, `getRightHip`). Η έκταση θα πρέπει να γίνει ταυτόχρονα και για τα δύο πόδια. Για ένα πιο ρεαλιστικό αποτέλεσμα (για να μην φαίνεται ότι στέκεται στον αέρα) μπορείτε ταυτόχρονα να την μετακινήσετε λίγο προς τα κάτω. Πειραματιστείτε με την απόσταση έτσι ώστε τα πόδια να μην μπαίνουν μέσα στο γρασίδι.

4. Προγραμματίστε την Alice να κάνει ελαφρά επίκλιση. Για να το πετύχετε θα χρειαστείτε την κλείδωση της μέσης (getSpineBase).
5. Δώστε καθυστέρηση ενός δευτερολέπτου και στη συνέχεια ακολουθήστε την αντίστροφη σειρά εντολών ώστε η Alice να επανέλθει στην αρχική της θέση. Οι κινήσεις αυτές μπορούν να εκτελεστούν από κάθε ον που ανήκει στην κλάση των δίποδων (Biped). Αντί να αντιγράψετε κάθε φορά τις εντολές για κάθε νέο αντικείμενο υπάρχει ένας άλλος ευκολότερο τρόπος. Μπορείτε να φτιάξετε μια διαδικασία (ομάδα εντολών) που να εφαρμόζεται στην κλάση Biped και να μπορεί να εκτελεστεί για οποιοδήποτε μέλος της.
6. Από το παράθυρο κώδικα πατήστε στο εικονίδιο των κλάσεων και από τον κατάλογο επιλέξτε την κλάση «Biped» και μετά «Add Biped Procedure». Στο πλαίσιο που θα εμφανιστεί δώστε το όνομα «LeanForward» και πατήστε OK. Παρατηρείστε ότι δημιουργήθηκαν δύο επιπλέον καρτέλες στο πάνω μέρος, η μία για την κλάση Biped και η άλλη για την διαδικασία.
7. Μεταβείτε στην καρτέλα myFirstMethod όπου είχατε φτιάξει το πρόγραμμά σας. Θα χρησιμοποιήσουμε το Πρόχειρο (Clipboard) για να μεταφέρουμε τις εντολές στη νέα διαδικασία. Αντιγράψτε τις εντολές στο Πρόχειρο (Copy to Clipboard). Σύρτε από το εικονίδιο του πρόχειρου στα δεξιά του πάνω μέρους της σελίδας κάθε μπλόκ εντολών στην καρτέλα LeanForward.

Παρατηρείστε ότι όλες οι αναφορές στο συγκεκριμένο άνθρωπο έχουν γίνει κόκκινες. Αυτό συμβαίνει διότι οι εντολές αυτές τώρα πια ισχύουν για όλα τα μέλη της κλάσης Biped. Αλλάξτε λοιπόν το this.Alice με τη σκέτη λέξη this.

8. Μεταβείτε στην καρτέλα myFirstMethod. Επιλέξτε την Alice και δείτε ότι στην αριστερή στήλη φαίνεται η νέα διαδικασία. Σύρτε τη διαδικασία στην περιοχή εντολών και τρέξτε το πρόγραμμα. Η Alice θα πρέπει να κάνει τις ίδιες ακριβώς κινήσεις.
9. Εισάγετε στη σκηνή έναν Λαγό (Hare), μία χελώνα (tortoise) και τον MadHatter και κάντε τους να εκτελούν όλοι μαζί την άσκηση γυμναστικής.



Εικόνα 15: Τα αντικείμενα της σκηνής

2.1.4 1^η Δραστηριότητα

1. Δημιουργήστε ένα νέο Έργο και δώστε του το όνομα «Jump». Διαλέξτε για σκηνικό το χώμα (DIRT) και τοποθετήστε μέσα έναν Λαγό.
2. Φτιάξτε μια διαδικασία με την οποία θα μπορείτε να κάνετε κάθε δίποδο ον (Biped) να αναπηδά. Η αναπήδηση αποτελείται από ανέβασμα με ταυτόχρονη μετακίνηση προς τα εμπρός και στη συνέχεια κατέβασμα, πάλι με ταυτόχρονη κίνηση προς τα εμπρός.
3. Εισάγετε τις εντολές που θα υψώνουν το όν κατά 0.25 και θα το μετακινούν μπροστά κατά 0.25 ταυτόχρονα. Ρυθμίστε την κάθε κίνηση να διαρκεί 0.5 και να ξεκινά και να σταματά ομαλά (BEGIN_AND_END_GENTLY).
4. Δώστε τώρα τις εντολές που θα κάνουν το δεύτερο μισό της κίνησης, όπως περιγράφεται στο προηγούμενο βήμα.
5. Πατήστε στην καρτέλα MyFirstMethod και κάντε το Λαγό να γυρίσει προς τα δεξιά του. Στη συνέχεια προγραμματίστε το Λαγό να αναπηδήσει τρεις φορές, χρησιμοποιώντας τη διαδικασία που μόλις φτιάξατε.
6. Βάλτε στη σκηνή μία χελώνα (tortoise) και προγραμματίστε την να κάνει την ίδια κίνηση με το Λαγό.

2.1.5 2^η Δραστηριότητα

Ανοίξτε ένα νέο έργο. Διαλέξτε ένα σκηνικό με στέρεο έδαφος. Αποθηκεύστε το σαν «Walking». Στο έργο αυτό θα κάνουμε έναν άνθρωπο (ή οποιοδήποτε δίποδο) να περπατά με (σχεδόν) ρεαλιστικό τρόπο.

Το περπάτημα στα δίποδα όντα αναλύεται στις εξής κινήσεις (το κάθε μπλόκ εντολών εκτελείται ταυτόχρονα):

- α. Στροφή του αριστερού ώμου (LeftShoulder) προς τα Αριστερά κατά 0.07
- β. Στροφή του δεξιού ώμου (RightShoulder) προς τα Αριστερά κατά 0.07
- γ. Στροφή του αριστερού γοφού (LeftHip) προς τα εμπρός κατά 0.05
- δ. Στροφή του δεξιού γοφού (RightHip) προς τα πίσω κατά 0.05
- ε. Στροφή του αριστερού γόνατου (LeftKnee) προς τα εμπρός κατά 0.05
- στ. Μετακίνηση προς τα εμπρός κατά 0.25
- ζ. Επαναφορά στην αρχική στάση (straightenOutJoints)
- η. Μετακίνηση προς τα εμπρός κατά 0.25
- θ. Οι εντολές α έως και στ με αντίθετη κατεύθυνση (εκτός από την στ. που εξακολουθεί να μετακινεί προς τα εμπρός)
- ι. Οι εντολές ζ και η

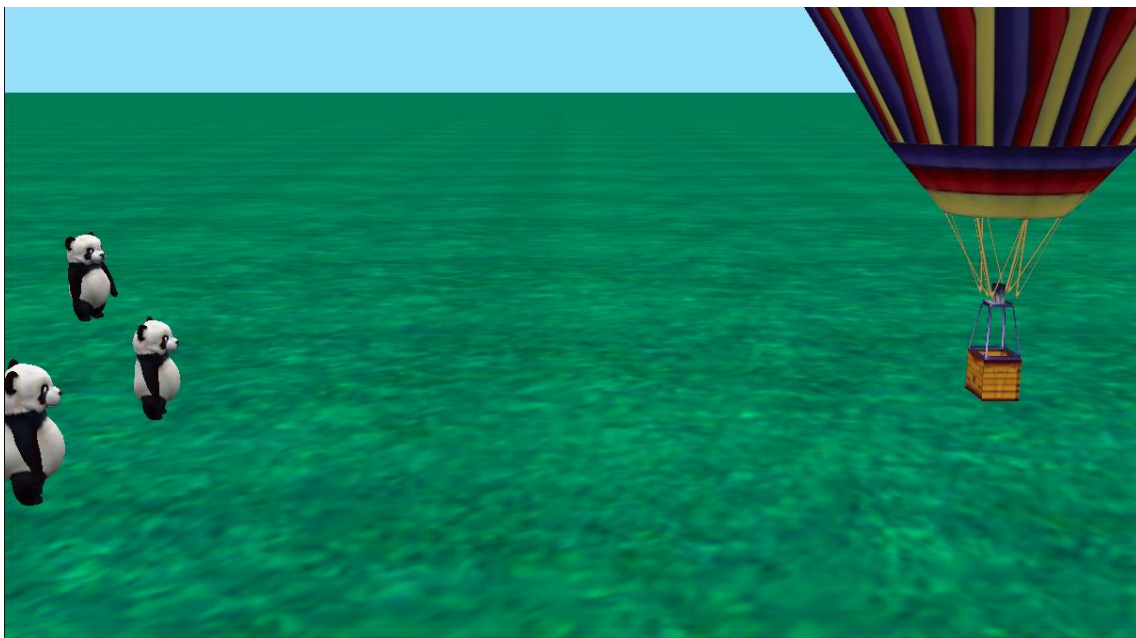
Υλοποιείστε τις προηγούμενες εντολές απευθείας στην κλάση Biped (διαδικασία walking). Στη συνέχεια πρώτα σε ένα δίποδο ον και μετά αντιγράψτε τις στην κλάση Biped (διαδικασία walking)

Τοποθετήστε στη σκηνή δύο ή τρία δίποδα όντα και βάλτε τα να κάνουν 3 επαναλήψεις των βημάτων σε διαφορετικές κατευθύνσεις. Για την επανάληψη χρησιμοποιείτε την εντολή `count` που θα σύρετε από το κάτω μέρος της εφαρμογής.

2.1.6 3^η Δραστηριότητα

Ανοίξτε ένα νέο έργο. Διαλέξτε ένα σκηνικό με στέρεο έδαφος. Αποθηκεύστε το σαν «Hot_Air_Balloon». Στο έργο αυτό θα χρειαστεί να χρησιμοποιήσετε όσα μάθατε στα 2 φύλλα εργασίας!

Αρχικά, προσθέστε στα δεξιά της σκηνής ένα `hotAirBalloon` μέσω του `SearchGallery`, ρυθμίστε το ύψος του και προσαρμόστε τη σκηνή ώστε να φαίνεται όσο το δυνατόν καλύτερα. Τοποθετήστε τρία `Panda` στα αριστερά της σκηνής σε απόσταση περίπου 10 μέτρων από την μπάλα και κάντε τα να κοιτάζουν προς την μπάλα. (Για ευκολία δημιουργήστε μια λίστα(`Pandas`) και χρησιμοποιήστε τη διαδικασία μίας φορές `turnToFace`).



Εικόνα 16: Το σκηνικό της δραστηριότητας

Προγραμματίστε τα `panda` να κινούνται με μικρά πηδηματάκια προς το αερόστατο. Για να το κάνετε αυτό γράψτε μια διαδικασία για την κλάση `Biped`. (Χρησιμοποιήστε τη μέθοδο `count` από το κάτω μέρος της οθόνης για να ορίσετε πόσα πηδηματάκια θα κάνουν τα `panda`). Προγραμματίστε τα `Panda` να μπαίνουν στο αερόστατο. Τι θα χρησιμοποιήσετε για να γίνει αυτό πιο εύκολα; Στη συνέχεια μετακινήστε το αερόστατο ώστε να μην φαίνεται στη σκηνή. Στο τέλος του έργου θα πρέπει να φαίνεται μόνο το γρασίδι!

2.1.7 4^η Δραστηριότητα

Εφόσον έχετε κατανοήσει όσα είδατε ως τώρα στο περιβάλλον `Alice`, τώρα ήρθε η ώρα να συζητήσετε σε ομάδες μια δική σας ιστορία που νομίζετε ότι είναι εφικτό να υλοποιηθεί στο

συγκεκριμένο περιβάλλον. Η ιστορία σας θα έχει όποια αντικείμενα εσείς θέλετε και θα διαρκεί περίπου ένα λεπτό! Προσοχή όμως! Στην ιστορία σας θα πρέπει να χρησιμοποιείται τις παρακάτω μεθόδους:

- **say/think**
- **move**
- **turn**
- **count_**
- **do together**
- **setOpacity**

Ανάλογα με την ιστορία σας σε κάθε μέθοδο βάλτε τις κατάλληλες λεπτομέρειες(set detail)! Σημείωση: μπορείτε να χρησιμοποιήσετε μεθόδους για κλειδώσεις/ξεχωριστά μέλη του σώματος. Τέλος, στην ιστορία σας θα πρέπει να δημιουργείτε και μια διαδικασία όπως μάθατε στο φύλλο παραπάνω.

2.1.8 Φύλλο εργασίας για το διαγώνισμα τετραμήνου

Ενδεικτική Διάρκεια: 90 λεπτά (2 διδακτικές ώρες)

Διδακτικός στόχος

Σκοπός αυτής της δραστηριότητας είναι η αξιολόγηση των μαθητών στις έννοιες που διδάχθηκαν στο προγραμματιστικό περιβάλλον Alice. Μέσω αυτής της δραστηριότητας μπορούμε να καταλάβουμε σε ποιο βαθμό οι μαθητές κατανόησαν τις έννοιες που διδάχθηκαν κι αν είναι σε θέση να τις εφαρμόσουν.

Άσκηση

Επιλέξτε το περιβάλλον Snow. Για να φτιάξετε τη σκηνή χρησιμοποιήστε αντικείμενα που βρίσκονται στο θέμα snow (έκτος από τη λίμνη-αναζητήστε τη με το όνομα pond). Υλοποιήστε την ιστορία που μόλις είδατε χρησιμοποιώντας μία συνάρτηση για την εμφάνιση των πιγκουίνων στη σκηνή και κατάλληλα Camera Markers ώστε να επιτύχετε το αντίστοιχο αποτέλεσμα.

Υπόδειξη:

Για την περιστροφή του ice scater χρειάζονται δύο σεντ εντολών : κίνηση αριστερά και σήκωμα των χεριών, κίνηση δεξιά και κατέβασμα των χεριών.



Εικόνα 17: Σκηνή frame 1



Εικόνα 18: Σκηνή frame 2



Εικόνα 19: Σκηνή frame 3



Εικόνα 20: Σκηνή frame 4

3. Επιπλέον δραστηριότητες και φύλλα εργασίας σε άλλα προγραμματιστικά περιβάλλοντα

3.1 Εισαγωγή στο προγραμματιστικό περιβάλλον Pivot Animator

Το Pivot Animator (αλλιώς Pivot Stickfigure Animator ή σύντομα Pivot) είναι μια δωρεάν εφαρμογή ^[9], η οποία επιτρέπει στους χρήστες να δημιουργούν φιγούρες και αντικείμενα με τη χρήση μόνο γραμμών και κύκλων, και να τα αποθηκεύουν σαν κινούμενα GIF (Graphics Interchange Format).

3.1.1 Δραστηριότητες και Φύλλα Εργασίας

3.1.1.1 1^ο Φύλλο εργασίας

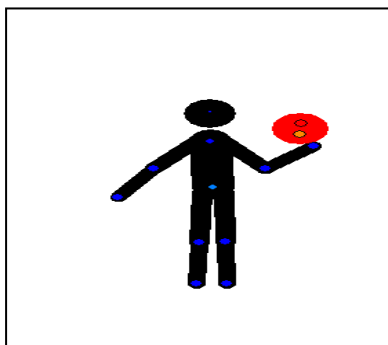
Ενδεικτική Διάρκεια: 45 λεπτά (1 διδακτική ώρα)

Διδακτικός στόχος

Μετά την ενασχόληση των μαθητών με το περιβάλλον Alice, και αφού δημιούργησαν τις δικές τους ιστορίες, σκεφτήκαμε να τους παρουσιάσουμε ακόμη ένα περιβάλλον, το **Pivot Animator**, στο οποίο μπορούν να δημιουργήσουν αντικείμενα, να τα μετακινούν όπως εκείνοι επιθυμούν και να δημιουργούν και πάλι τις δικές τους ιστορίες. Στόχος είναι μετά την ενασχόληση των μαθητών με το περιβάλλον το Pivot Animator, οι μαθητές να είναι σε θέση να δημιουργούν κινούμενα σχέδια.

Αρχικά, από τον κοινόχρηστο φάκελο του υπολογιστή, καλείστε να βρείτε τα εξής animation: Ball, Girl, Skater Boy, Stairs και Skateboard.

1. Ανοίξτε το Pivot Animator
2. Επιλέξτε File → Load Figure Type και επιλέξτε το realistic_man.
3. Επίσης, προσθέστε στη σκηνή το αντικείμενο Ball και δώστε του ένα χρώμα της επιλογής σας. Η αρχική εικόνα πρέπει να είναι η εξής:



Εικόνα 21: Αντικείμενο realistic_man

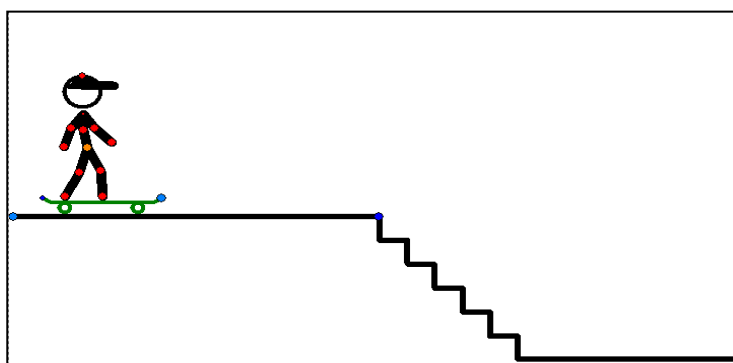
4. Στη συνέχεια, αλλάζοντας τη θέση των χεριών και της μπάλας θα προσθέτετε καρτέ στο επάνω μέρος της οθόνης πατώντας το **Add Frame**. Η μπάλα θα πρέπει περνώντας πάνω από το κεφάλι του ανθρώπου να φτάνει στο δεξί του χέρι και μετά να επανέρχεται πάλι στο αριστερό. Προσοχή! Θα πρέπει να φαίνονται όλες οι θέσεις των χεριών και της μπάλας!
5. Μόλις ολοκληρώσετε τις σκηνές, πατήστε το κουμπί Play.
6. Αποθηκεύστε το Animation που δημιουργήσατε με το όνομα play_ball.

3.1.1.2 1^η Δραστηριότητα

1. Επιλέξτε όλα τα καρτέ που προσθέσατε στην ιστορία σας, εκτός από το πρώτο και πατήστε το **X** στα αριστερά, ώστε να διαγραφούν.
2. Προσθέστε στη σκηνή το animation Girl (Load Figure Type → Girl → Άνοιγμα).
3. Το κορίτσι θα πρέπει να κοιτάει προς τον άνθρωπο. Αν κάτι τέτοιο δεν συμβαίνει, επιλέξτε Flip/Mirror Figure (M) από το μενού αριστερά.
4. Σε αυτή την δραστηριότητα, ο άνθρωπος θα πρέπει να πετάει την μπάλα στο κορίτσι και εκείνο να του επιστρέφει.
5. Αφού προσθέσετε τα καρτέ που χρειάζονται, πατήστε Play και αποθηκεύστε το Animation με όνομα play_with_the_girl.

3.1.1.3 2^η Δραστηριότητα

1. Δημιουργήστε ένα νέο Animation (File → New). Επιλέξτε το ανθρωπάκι που εμφανίζεται και διαγράψτε το.
2. Πρόσθεσε στην σκηνή τα animation Skater Boy, Stairs και Skateboard. Η αρχική σας σκηνή θα πρέπει να είναι η εξής:



Εικόνα 22: Σκηνή 3^{ης} δραστηριότητας

3. Στη συνέχεια, αλλάζοντας τη θέση του αγοριού και του skate θα προσθέσετε καρτέ στο επάνω μέρος της οθόνης πατώντας το **Add Frame**, μέχρις ότου το αγόρι φτάσει στο τέλος της γραμμής. Προσπαθήστε το αποτέλεσμα να είναι όσο το δυνατόν πιο ρεαλιστικό, ειδικά όταν το αγόρι θα κατέβει τις σκάλες με το skate.
4. Πατήστε Play και τέλος, αποθηκεύστε το έργο σας με όνομα Skater_Boy.

3.1.1.4 3^η Δραστηριότητα

1. Δημιουργήστε ένα νέο έργο και διαγράψτε το ανθρωπάκι που εμφανίζεται.
2. Επιλέξτε File → Create New Animation. Δημιουργήστε τον δικό σας χαρακτήρα και βάλτε τον να κάνει ό, τι εσείς επιθυμείτε.
3. Πατήστε Play και αποθηκεύστε το έργο σας με όποιο όνομα επιθυμείτε.

3.2 Εισαγωγή στο προγραμματιστικό περιβάλλον Game Maker

Το GameMaker είναι ένα περιβάλλον ανάπτυξης παιχνιδιών που αρχικά δημιουργήθηκε το 1999 από τον Mark Overmars σε γλώσσα προγραμματισμού Delphi ^[10]. Τώρα, το

περιβάλλον πλέον αναπτύσσεται και διατίθεται από την εταιρεία YoYo Games. Το GameMaker σχεδιάστηκε έτσι, ώστε να επιτρέπει την γρήγορη δημιουργία παιχνιδιών ακόμα και από χρήστες που δεν έχουν εμπειρία στον προγραμματισμό. Το 2012 κυκλοφόρησε η πρώτη έκδοση του GameMaker Studio, η οποία γράφτηκε σε C#, και επιτρέπει την δημιουργία παιχνιδιών και την έκδοσή τους σε πολλές διαφορετικές πλατφόρμες. Η τρέχουσα έκδοση του GameMaker Studio υποστηρίζει την δημιουργία παιχνιδιών για Windows, Mac-OS, iOS, Android, σε HTML5, Windows 8, Windows 8 phone και Linux.

Στο περιβάλλον αλληλεπίδρασης του GameMaker γίνεται χρήση ενός συστήματος συρσίματος και εναπόθεσης, επιτρέποντας σε χρήστες που δεν είναι εξοικειωμένοι με τον προγραμματισμό να δημιουργούν διαισθητικώς παιχνίδια, οργανώνοντας με οπτικό τρόπο εικονίδια στην οθόνη του υπολογιστή. Αυτά τα εικονίδια, αντιπροσωπεύουν ενέργειες που συμβαίνουν σε ένα παιχνίδι, όπως κινήσεις, σχεδίαση και απλές δομές ελέγχου. Για πιο προχωρημένους χρήστες, το GameMaker περιέχει μια ενσωματωμένη γλώσσα προγραμματισμού την Game Maker Language (GML), επιτρέποντας έτσι την δημιουργία πιο πολύπλοκων παιχνιδιών.

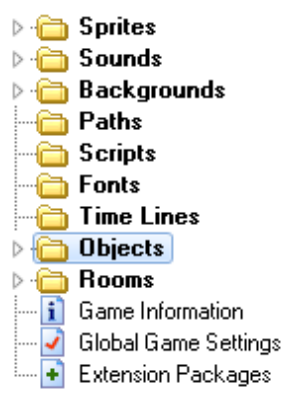
Κάποια από τα βασικά συστατικά του περιβάλλοντος δημιουργίας παιχνιδιών GameMaker είναι:

1. Το μενού επιλογών και η μπάρα εργαλείων με εικονίδια - συντομεύσεις λειτουργιών του μενού επιλογών.



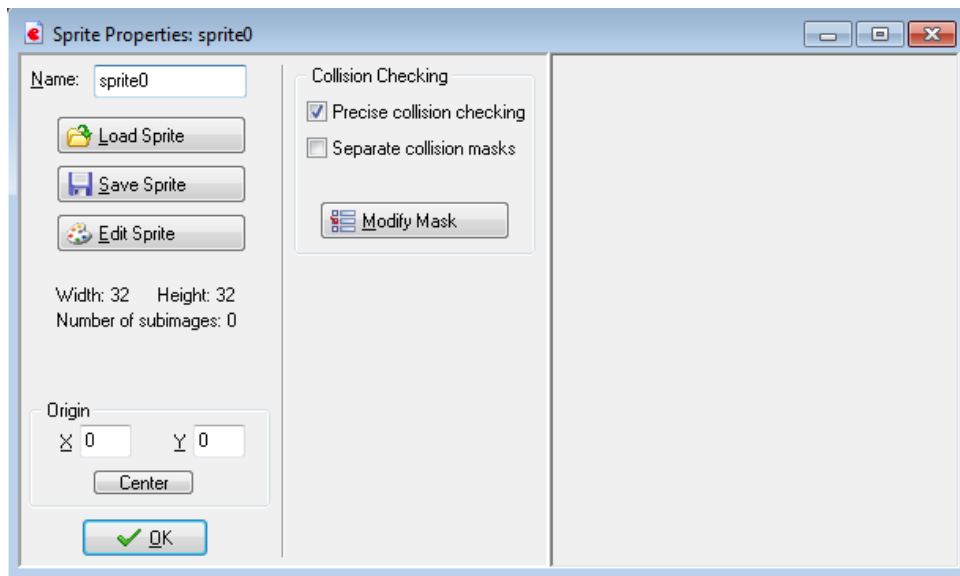
Εικόνα 23: Μενού επιλογών και μπάρα εργαλείων

2. Η περιοχή με τους πόρους του παιχνιδιού όπου παρατίθενται, σε μορφή λίστας, όλα τα είδη των πόρων του παιχνιδιού (γραφικά στοιχεία, ήχοι, φόντα, προγραμματιζόμενα αντικείμενα, δωμάτια, πληροφορίες για το παιχνίδι κ.ά.).



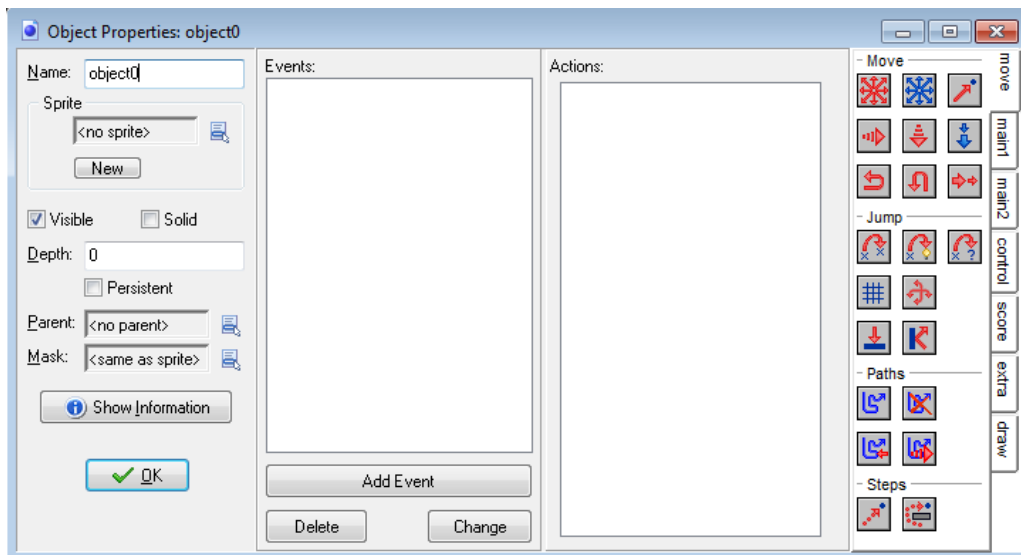
Εικόνα 24: Πόροι του παιχνιδιού

3. Παράθυρο ιδιοτήτων γραφικού στοιχείου όπου επιλέγεται/σχεδιάζεται/τροποποιείται η εικόνα που θα αντιστοιχεί σε αυτό.



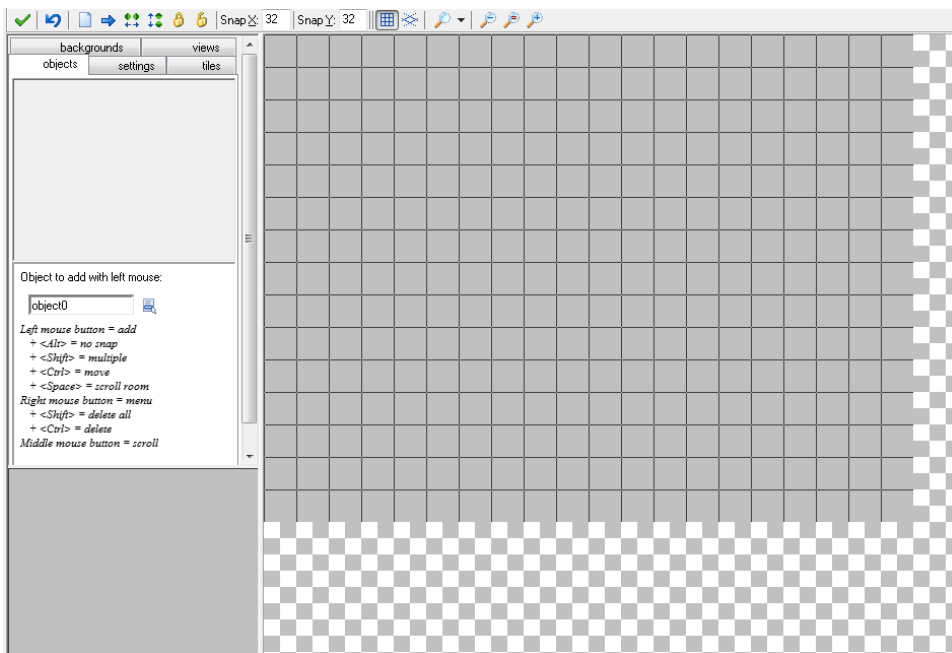
Εικόνα 25: Παράθυρο ιδιοτήτων γραφικού στοιχείου

4. Παράθυρο ιδιοτήτων προγραμματιζόμενου αντικείμενου όπου λαμβάνει χώρα ο προγραμματισμός αυτού (ορισμός γεγονότος και των ενεργειών που θα εκτελεστούν όταν αυτό συμβεί). Εδώ επίσης επιλέγεται αν το αντικείμενο αντιστοιχεί σε κάποιο ήδη ορισμένο γραφικό στοιχείο και αν κληρονομεί τις ιδιότητες από κάποιο άλλο (γονικό) αντικείμενο.



Εικόνα 26: Παράθυρο ιδιοτήτων προγραμματιζόμενου αντικείμενου

5. Παράθυρο ιδιοτήτων των δωματίου (πάνω στο οποίο θα εξελίσσεται το παιχνίδι) όπου συμβαίνει η σχεδίαση αυτού.



Εικόνα 27: Παράθυρο ιδιοτήτων των δωματίου

6. Παράθυρο εκτέλεσης του παιχνιδιού. Το παράθυρο αυτό εμφανίζεται επιλέγοντας Run Normally/Run in Debug Mode, αφού δημιουργήσουμε το παιχνίδι μας, όπως θα δούμε και στις παρακάτω δραστηριότητες.

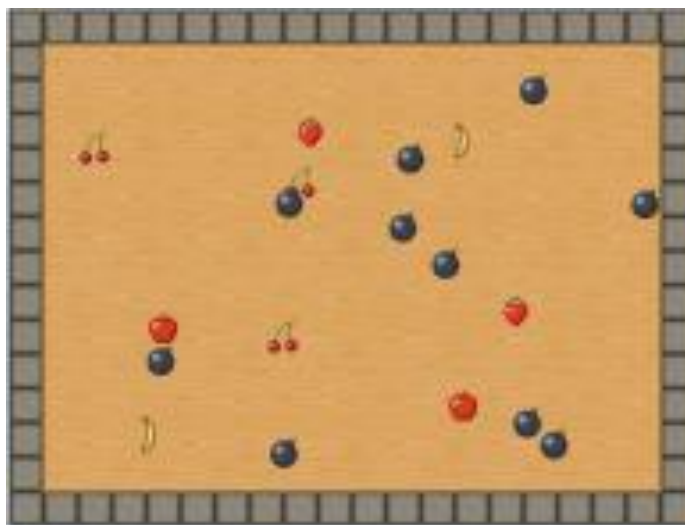
3.2.1 Δραστηριότητες και Φύλλα Εργασίας

3.2.1.1 1^ο Φύλλο εργασίας

Ενδεικτική Διάρκεια: 90 λεπτά (2 διδακτικές ώρες)

Διδακτικός στόχος

Για το πρώτο φύλλο εργασίας χρησιμοποιήθηκε ένα παιχνίδι tutorial του GameMaker που βοηθάει στην εισαγωγή στο περιβάλλον. Δημιουργήσαμε λοιπόν το παρακάτω φύλλο εργασίας ως κατευθυντήρια γραμμή για τους μαθητές. Σκοπός του φύλλου εργασίας είναι η ανάπτυξη ενός παιχνιδιού, στο οποίο ο χρήστης καλείται να επιλέξει με το ποντίκι αντικείμενα σε μια πίστα και με τον τρόπο αυτό να κερδίσει πόντους (ή να χάσει το παιχνίδι).



Εικόνα 28: Τελική μορφή παιχνιδιού

➤ Εισαγωγή των sprites

Στα παιχνίδια του GAME MAKER χρησιμοποιούνται διάφορα εικονίδια για τα αντικείμενα που υπάρχουν σε αυτά (ανθρωπάκια, τοίχος, βόμβες, κ.λπ.). Οι εικόνες αυτές ονομάζονται sprites. Η προσθήκη των sprites γίνεται από την επιλογή Resources-> CreateSprite ή από το αντίστοιχο εικονίδιο. Στη συνέχεια επιλέγετε LoadSprite στο μενού που θα εμφανιστεί και δίνεται το κατάλληλο όνομα.

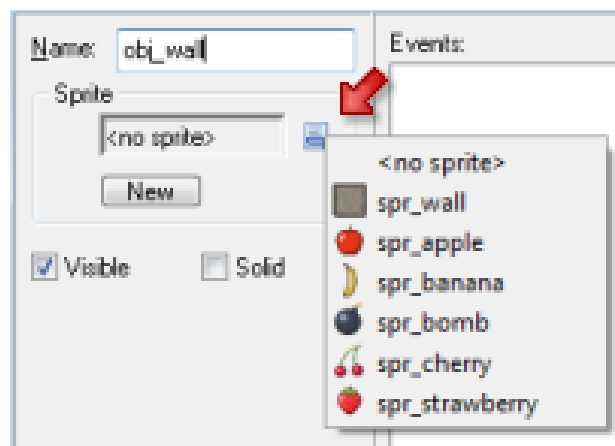


Εικόνα 29: Προσθήκη των sprites

Εισάγετε τις εικόνες που αντιστοιχούν στα παρακάτω αντικείμενα: μήλο, μπανάνα, κεράσι, φράουλα, βόμβα και τοίχος. Στα αντικείμενα αυτά δώστε τα ονόματα spr_apple, spr_banana, spr_cherry, spr_strawberry, spr_bomb και spr_wall.

➤ Δημιουργία objects

Τα sprites είναι μόνο εικόνες. Δεν μπορούν να μετακινηθούν, να τρέξουν κ.λπ. . Για να γίνει αυτό πρέπει να δημιουργηθούν τα κατάλληλα αντικείμενα (objects) και να συνδεθούν με τα αντίστοιχα sprites. Η δημιουργία objects γίνεται από την επιλογή Resources-> CreateObject ή από το αντίστοιχο εικονίδιο. Η αντιστοίχιση με το κατάλληλο sprite γίνεται από το μενού που εμφανίζεται



Εικόνα 30: Δημιουργία των objects (1)

Δημιουργήστε ένα αντικείμενο για το sprite spr_apple. Δώστε στο αντικείμενο το όνομα obj_apple.



Εικόνα 31: Δημιουργία των objects (2)

➤ Δημιουργία συμπαγών αντικειμένων

Κάποια από τα αντικείμενα που υπάρχουν στα παιχνίδια είναι συμπαγή (solid). Αυτό σημαίνει ότι μέσα από αυτά δεν μπορούν να περάσουν άλλα αντικείμενα. Στο συγκεκριμένο παιχνίδι το μόνο συμπαγές αντικείμενο είναι ο τοίχος. Δημιουργείστε το αντικείμενο obj_wall και στην αντίστοιχη επιλογή ,επιλέγετε solid.

➤ Προσθήκη συμπεριφορών στα αντικείμενα

Τα περισσότερα από τα αντικείμενα που υπάρχουν στα παιχνίδια δεν είναι ακίνητα, αλλά αντιδρούν ή μετακινούνται στο κλικ του ποντικιού. Για να γίνει αυτό , πρέπει στα αντικείμενα να προστεθούν οι αντίστοιχες συμπεριφορές. Αυτό γίνεται μέσω του καθορισμού γεγονότων (events). Events συμβαίνουν, όταν για παράδειγμα δύο αντικείμενα συγκρουστούν ή όταν πατήσουμε ένα πλήκτρο από το πληκτρολόγιο. Τα events συνοδεύονται από κάποιες ενέργειες (actions), όπως για παράδειγμα η μετακίνηση προς κάποια κατεύθυνση, η αύξηση του σκορ κ.λπ. .

Συμπεριφορά: Κατά τη δημιουργία του obj_apple, το αντίστοιχο αντικείμενο να μετακινείται αυτόματα και τυχαία προς κάθε κατεύθυνση.

Event: Δημιουργία obj_apple

Action: Τυχαία μετακίνηση προς μια κατεύθυνση

Διπλό κλικ στο obj_apple- AddEvent-> Create και επιλογή του action MoveFixed(Drag and Drop στο αντίστοιχο παράθυρο). Επιλέγεται όλα τα 8 βελάκια και στο πεδίο speed εισάγετε την τιμή 8.

Συμπεριφορά: Όταν το obj_apple συγκρούεται με τον τοίχο, να αναπηδά

Event: Collision event με το obj_wall

Action: Bounce (είναι στο tab move, στο πεδίο jump)

AddEvent-> Collision και επιλογή του Action Bounce

Συμπεριφορά: όταν κάνουμε κλικ στο obj_apple τότε αυτό να μετακινείται σε μια τυχαία θέση και ο παίχτης να κερδίζει 50 πόντους.

Event: Mouse event Left pressed

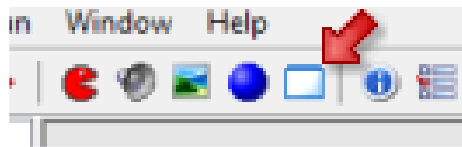
Action1: jump to random position

Action2: set score –τιμή 50 και επιλογή το relative

AddEvent-> Mouse Event (left pressed) και εισαγωγή των actions: jump to a random position και set the score relative to 50.

➤ Δημιουργία πίστας

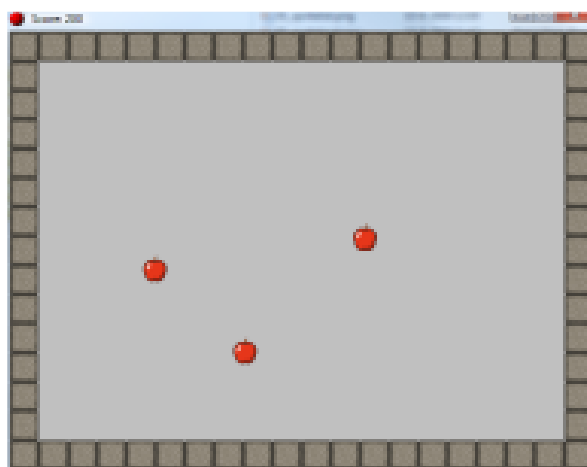
Για να δημιουργήσουμε μια πίστα (room) στην οποία θα υπάρχουν τα objects του παιχνιδιού, επιλέγουμε create room



Εικόνα 32: Δημιουργία πίστας

Τα sprites που χρησιμοποιούμε σε αυτό το παιχνίδι είναι 32x32 . Για αυτό επιλέγουμε SnapX 32 και SnapY 32.

Τοποθετείστε τα obj_wall και obj_apple όπως φαίνεται στο παρακάτω σχήμα.

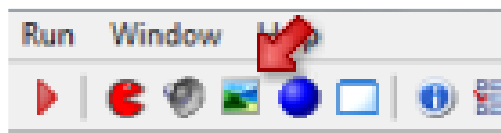


Εικόνα 33: Η πίστα

Αποθηκεύστε τις αλλαγές και στη συνέχεια πατήστε play για να παίξετε το παιχνίδι.

➤ Προσθήκη Φόντου

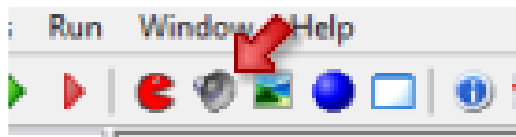
Για να προσθέσουμε ένα φόντο στο room επιλέγουμε Create Background



Εικόνα 34: Προσθήκη φόντου

LoadBackground-> Επιλογή wood.gif και ok για αποθήκευση. Προσθέστε το background στο room.

➤ Προσθήκη Ήχου



Εικόνα 35: Προσθήκη ήχου

Με τον ίδιο τρόπο που προσθέσαμε το φόντο μπορούμε να προσθέσουμε έναν ήχο(από το εικονίδιο με το megafono). Προσθέστε τον ήχο click.wav και δώστε το όνομα klik. Επιλέξτε το obj_apple και στο event LeftPressed(mouse) προσθέστε το action Play Sound (είναι στο tab main1) και τον ήχο klik. Τρέξτε το παιχνίδι και στη συνέχεια ελέγξτε τη λειτουργία του ήχου προσθέτοντας και άλλα αντικείμενα. Τα υπόλοιπα φρούτα έχουν παρόμοιες ιδιότητες με αυτές του obj_apple. Αντιγράψτε το obj_apple (δεξί κλικ-> duplicate), και αλλάξτε το sprite, ταχύτητα κίνησης καθώς και τους βαθμούς του κάθε κλικ (όσο μικρότερη η ταχύτητα, τόσο μικρότεροι οι βαθμοί). Στη συνέχεια επιλέξτε το room και προσθέστε τα καινούργια αντικείμενα.

➤ Προσθήκη Βομβών

Όταν παίζουμε ένα παιχνίδι, θα πρέπει να υπάρχει ένας τρόπος οι παίκτες να χάνουν. Για αυτό το λόγο θα προσθέσουμε βόμβες ώστε όταν ο παίχτης κάνει κλικ πάνω σε μία βόμβα, να χάνει. Για να δυσκολέψουμε το παιχνίδι, ο αριθμός των βομβών θα αυξάνεται συνεχώς.

Δημιουργείστε ένα αντικείμενο obj_bomb με τις παρακάτω ιδιότητες:

Όταν δημιουργείται να μεταπηδά προς μία τυχαία διεύθυνση (jump to random) και όταν ο παίχτης κάνει κλικ σε αυτό να ακούγεται ο ήχος explosion.wav (δημιουργία ήχου όπως πριν), να περιμένει για 1000ms, να εμφανίζει τον πίνακα με τα high scores και να τελειώνει το παιχνίδι.

Κάθε δύο δευτερόλεπτα θέλουμε να προστίθεται μία βόμβα στο παιχνίδι.

Η παραπάνω συμπεριφορά θα προστεθεί σε ένα αντικείμενο το οποίο δε θα το συνδέσουμε με κάποιο sprite. Ονομάστε το obj_alarm. Η λογική της παραπάνω συμπεριφοράς θα υλοποιηθεί με ένα χρονόμετρο(alarm) το οποίο θα μετράει αντίστροφα. Όταν φτάνει στο μηδέν θα δημιουργούμε την βόμβα.

Δημιουργήστε ένα αντικείμενο obj_alarm και μη το συνδέσετε με κάποιο sprite.

Στο αντικείμενο αυτό προσθέστε τις παρακάτω ιδιότητες:

Event: Create

Action: Set alarm 0 to 60

Event: Alarm Event

Action1: Create instance of object obj_bomb

Action2: Set alarm 0 to 60

Με τον τρόπο αυτό κάθε δύο δευτερόλεπτα (όταν το alarm 0 θα παίρνει την τιμή 0 και θα ενεργοποιείται) θα δημιουργείται μια νέα βόμβα στο room και το alarm 0 θα αρχίζει πάλι τα αντίστροφη μέτρηση. Προσθέτω το αντικείμενο obj_alarm σε ένα σημείο της πίστας (δε θα εμφανίζεται γιατί δεν έχει κάποιο sprite). Δοκιμάστε το παιχνίδι. Θα πρέπει κάθε δύο sec να δημιουργείται μια νέα βόμβα.

Αφού έχουν φτιαχτεί οι βόμβες στη συνέχεια σκοπός είναι όταν συγκρούονται να εξαφανίζονται.

Διπλό κλικ στο obj_bomb -> add event -> collision με obj_bomb και προσθέτουμε actions στο event που δημιουργήσαμε.

Θέλουμε να εισαχθεί και ήχος.

Drag and drop:

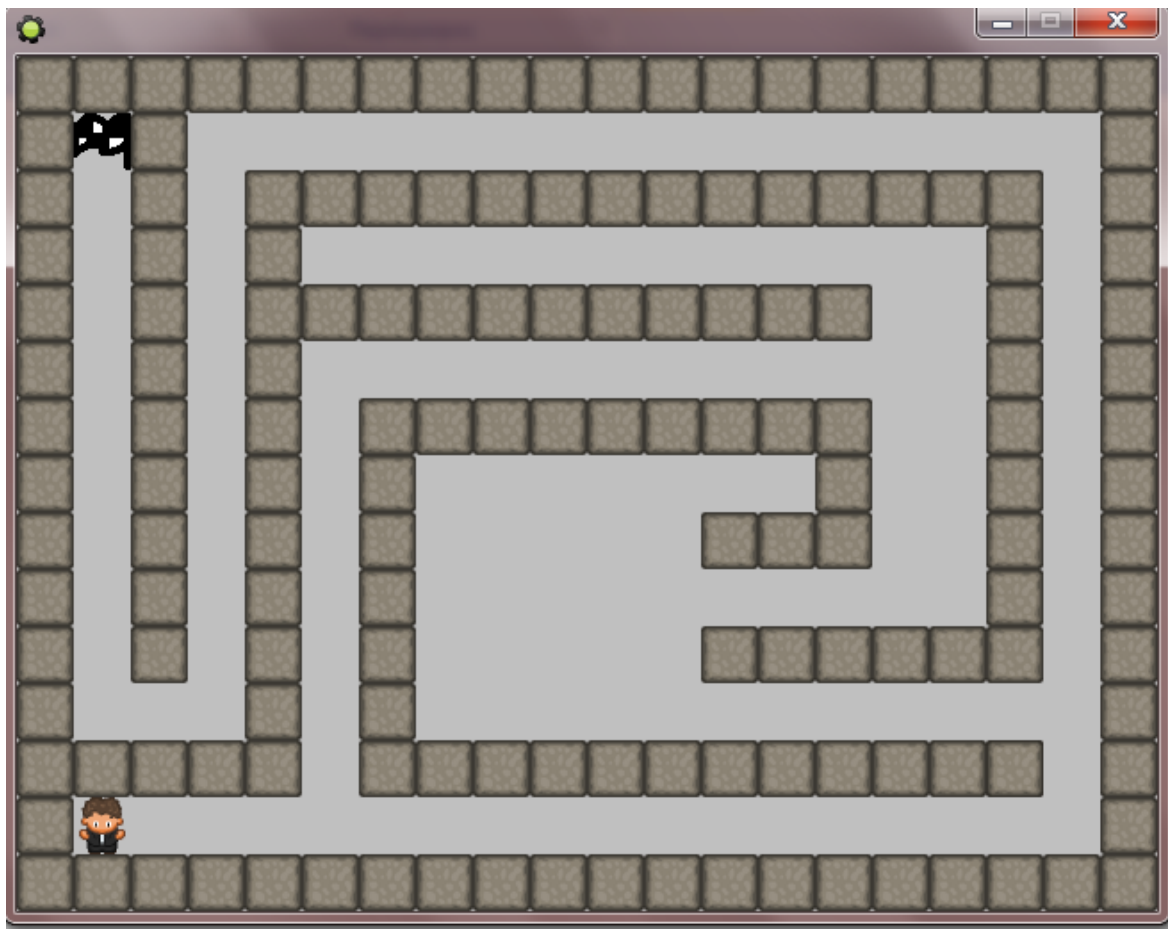
Play sound-> snd_explode (ήχος έκρηξης)

Destroy the instance.

3.2.1.2 1^η Δραστηριότητα

ΣΗΜΕΙΩΣΗ: Για τη δραστηριότητα αυτή θα πρέπει να έχει επιλεγεί Advanced Mode.

Σκοπός αυτής της δραστηριότητας είναι να σχεδιάσουμε το παιχνίδι που φαίνεται στην εικόνα, δηλαδή να φτιάξουμε ένα λαβύρινθο, όπου ο χαρακτήρας μας θα κινείται μέχρι να φτάσει στον τερματισμό, ενώ στη δεύτερη φάση θα μαζεύει διαμάντια ώστε να ανοίξει η πόρτα και να φτάσει στο σημείο του τερματισμού.



Εικόνα 36: Ο λαβύρινθος

➤ **Εισαγωγή των sprites που χρειαζόμαστε**

Για άλλη μια φορά, πρέπει να εισάγουμε τις εικόνες που αντιστοιχούν στα αντίστοιχα αντικείμενα. Πηγαίνουμε, λοιπόν, στο βασικό μενού και επιλέγουμε το εικονίδιο create sprite και δημιουργούμε τα εξής : spr_npc_guy, spr_wall, spr_goal (Για το 3^ο sprite, θα πρέπει να επιλέξετε edit sprite και να το δημιουργήσετε μόνοι σας!).

Στη συνέχεια σκοπός μας είναι να δημιουργήσουμε τα objects.

➤ **Εισαγωγή των objects που χρειαζόμαστε.**

Όπως έχουμε αναφέρει και παραπάνω, τα sprites είναι μόνο εικόνες. Δεν μπορούν να μετακινηθούν να τρέξουν κ.τ.λ., για αυτό θα πρέπει να δημιουργηθούν και τα κατάλληλα αντικείμενα και να συνδεθούν με τα sprites.

Δημιουργείτε τα αντικείμενα object. Πηγαίνετε στο βασικό μενού και επιλέξτε το εικονίδιο create object και δημιουργούμε τα εξής : obj_guy, obj_wall, obj_goal.

Στη συνέχεια προσθέστε, στα αντικείμενα που φτιάξατε, τις παρακάτω ιδιότητες:

1. obj_wall -> solid (αφού ο τοίχος μας είναι συμπαγής)
2. Ύστερα για το obj_guy:

Add_event -> collision με obj_wall.

-> Actions -> Move Fixed

Add_event -> keyboard -> no key (ακινησία)

-> Actions(control) -> Check Grid (32,32)

-> Actions(move) -> Move Fixed (Προσοχή! Επιλέξτε στη σωστή ταχύτητα 1)

Με την ίδια διαδικασία θα επιλέξουμε το left,right,up,down και στο Actions(move)->Move Fixed θα επιλέξουμε την αντίστοιχη κατεύθυνση.

1. Για το obj_goal:

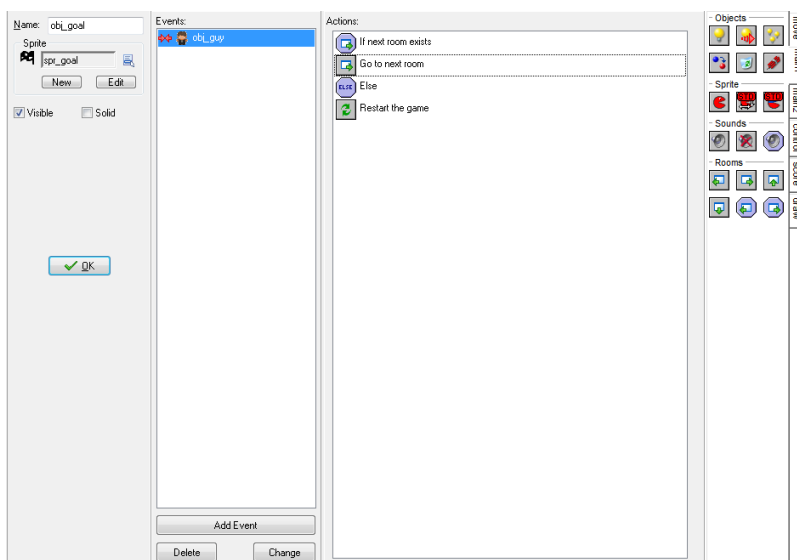
Add_event -> collision με obj_guy

-> Actions(main1) -> check next

-> Next room

-> Actions(control) -> else

-> Actions(main2) -> restart game



Εικόνα 37: Προσθήκη ιδιοτήτων

➤ Δημιουργία των rooms

Από το βασικό μενού επιλέγουμε το create room και βάζουμε διαστάσεις X:32 και Y:32. Στη συνέχεια κάνουμε διπλό κλικ στο room που δημιουργήσαμε επιλέγουμε την ετικέτα objects και χτίζουμε το δωμάτιο.

➤ Δημιουργία background

Πάλι από το βασικό μενού επιλέγουμε το εικονίδιο create background και το δημιουργούμε σύμφωνα με αυτά που έχουμε αναφέρει.

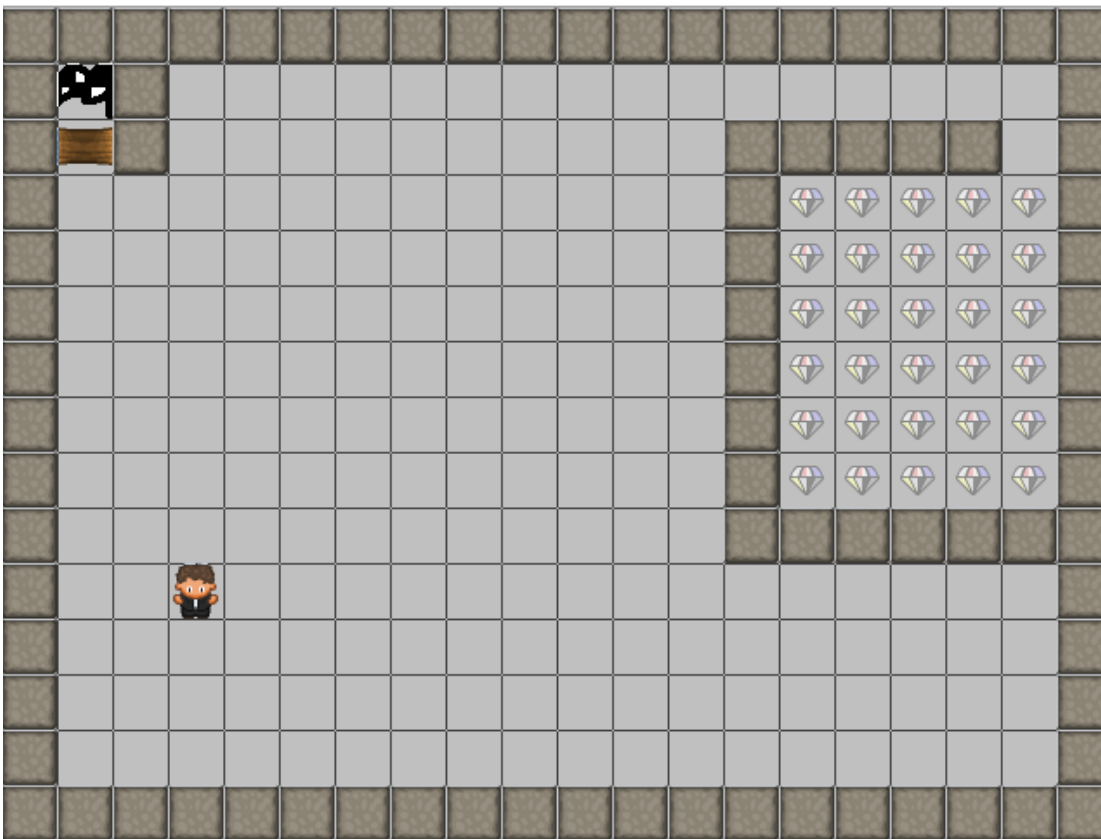
➤ Δημιουργία δεύτερου επιπέδου

Στο δεύτερο επίπεδο σκοπός μας είναι να φτιάξουμε μία πίστα όπου ο παίχτης θα πρέπει να συλλέγει τα διαμάντια(obj_diamond) και στο τέλος να τερματίζει.

Αυτό για να πραγματοποιηθεί αρκεί να κάνουμε τα εξής:

Δημιουργία Sprites και Objects (ακολουθούμε τα παραπάνω βήματα) ,για τα sprt_diamond, sprt_door και obj_diamond, obj_door αντίστοιχα.

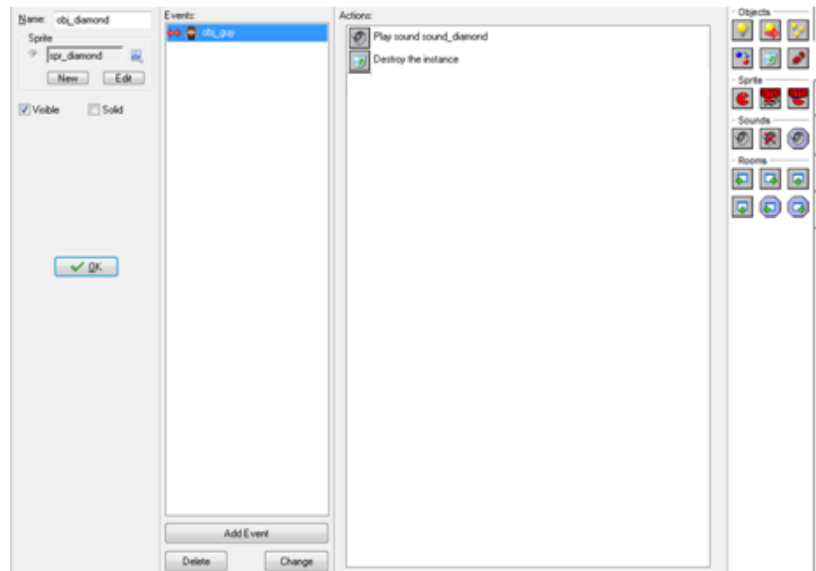
Προσθέστε τις απαραίτητες ιδιότητες στα διαμάντια ώστε να εξαφανίζεται όταν το obj_guy τα μαζέψει ο χαρακτήρας σας, ενώ για κάθε διαμάντι ο παίκτης θα παίρνει 10 πόντους.



Εικόνα 38: Ο λαβύρινθος μέχρι τώρα

➤ Προσθήκη ήχου

Πηγαίνουμε στο βασικό μενού και πατάμε το εικονίδιο ->create sounds και δημιουργούμε έναν ήχο για τη συλλογή των διαμαντιών (found_item και αποθηκεύστε το ως sound_diamond) και έναν ήχο για το άνοιγμα της πόρτας (door_rough και αποθηκεύστε το ως sound_door).



Εικόνα 39: Προσθήκη ήχου

Προσθέστε στο obj_diamond-> actions(main1)-> play sound και επιλέξτε τον αντίστοιχο ήχο. Ο ήχος θα πρέπει να ακούγεται πριν εξαφανιστεί το διαμάντι!

Θεωρείστε ότι το obj_wood είναι μια πόρτα και θα πρέπει να έχει τις εξής ιδιότητες:

Add event -> step

- > Actions(control) ->Test instance Count (obj_diamond equal to 0)
- > Actions(control) -> start of the block
- > Actions(main1) -> play sound(false)
 - > destroy the instance
- > Actions(control) -> end of block.

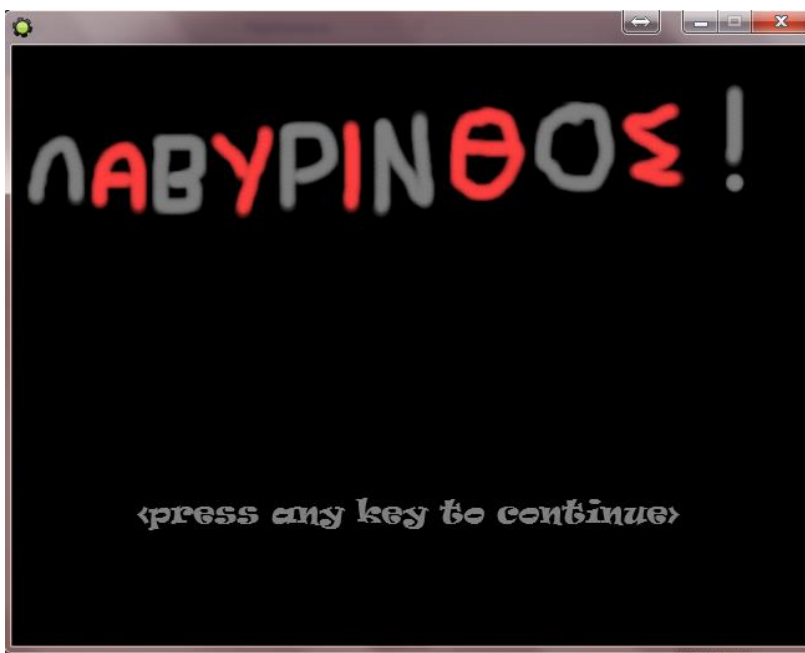
➤ Δημιουργία οθόνης εκκίνησης του παιχνιδιού

Ουσιαστικά η οθόνη εκκίνησης του παιχνιδιού θα είναι ένα επιπλέον room που θα δημιουργήσουμε πριν από τις πίστες μας. Αυτό θα έχει ως background τον τίτλο του παιχνιδιού μας και το μήνυμα <press any key to start>.

Θέλουμε λοιπόν όταν φορτώσει η αρχική οθόνη όταν πατήσουμε ένα οποιοδήποτε κουμπί να μπορούμε στη πρώτη πίστα του παιχνιδιού. Πρέπει λοιπόν να έχουμε ένα object, το οποίο ωστόσο δε θα φαίνεται, το οποίο θα έχει τις ανάλογες ιδιότητες.

Δημιουργούμε λοιπόν το object controller_start και πάμε να του προσθέσουμε τις ιδιότητες:

Add Event -> keyboard-> any key
-> Actions(main1)-> next room



Εικόνα 40: Οθόνη εκκίνησης παιχνιδιού

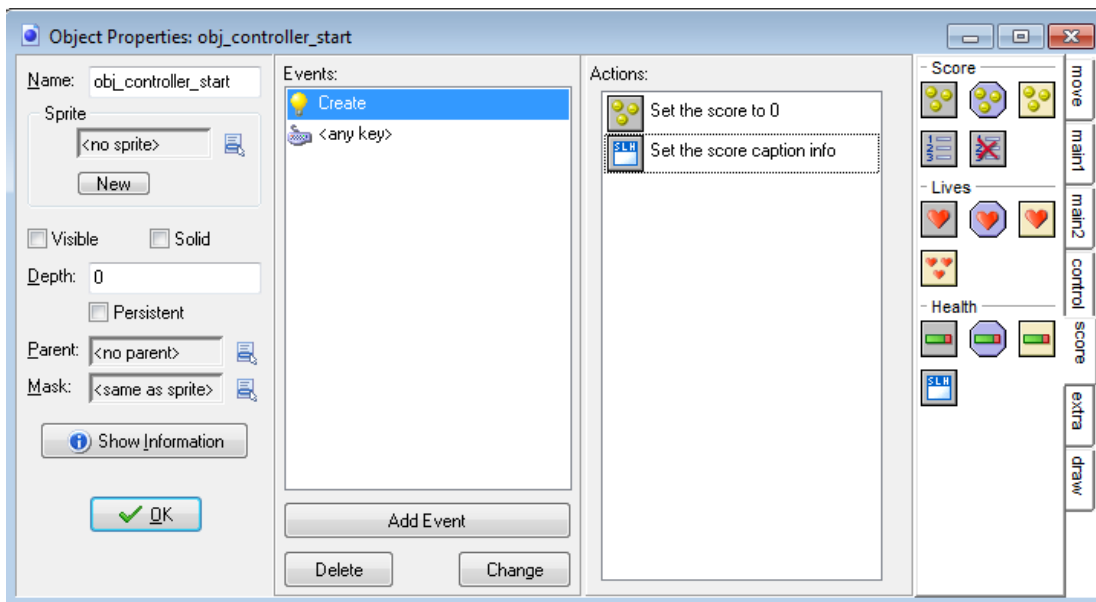
➤ Εμφάνιση του σκορ

Θέλουμε με το που θα ανοίγει το παιχνίδι να γίνεται αρχικοποίηση του σκορ. Δηλαδή :
Στο controller_start-> Add Event-> Create

-> Actions(score)-> set score (0)

-> Score Caption (don't show)

Επιλέγουμε don't show για να μη φαίνεται στο περίγραμμα του παραθύρου.



Εικόνα 41: Score παιχνιδιού

Για να εμφανίζεται το score μέσα στις πίστες θα δημιουργήσουμε ένα object που θα το κάνει αυτό, το obj_controller_main και θα το τοποθετήσουμε σε κάθε room στο σημείο που θέλουμε να φαίνεται το σκορ.

Αφού δημιουργήσουμε το obj_controller_main, προσθέτουμε τις ιδιότητες που θέλουμε να έχει:

Add Event -> Draw -> Actions(draw)-> Set Color

Σε αυτό το σημείο θα πρέπει να φτιαχτεί μια γραμματοσειρά για το πως θα φαίνεται το σκορ που θα φτιάξουμε στην πίστα μας. Οπότε θα πρέπει να φτιάξουμε από το εικονίδιο στο βασικό μενού Create a Font τα χαρακτηριστικά που επιθυμούμε. (Font, μέγεθος γραμματοσειράς και τα λοιπά).

Αφού φτιάξουμε το Font συνεχίζουμε στις actions του controller_main:

Actions(draw)-> Set Font (επιλέγουμε το font που φτιάξαμε)

Actions(score)-> Draw Score (τσεκάρουμε την επιλογή relative)

➤ **Εμφάνιση μηνύματος λήξης του παιχνιδιού και πίνακα highscores**

Όταν ολοκληρώνονται όλα τα επίπεδα του παιχνιδιού, θέλουμε σε ένα παράθυρο να εμφανιστεί το μήνυμα «Congratulations! All levels are cleared!!!». Αυτό θα γίνει όταν φτάσουμε στο goal του τελευταίου επιπέδου. Πρέπει λοιπόν να προσθέσουμε μία επιπλέον action στο event collision with obj_guy του obj_goal. Αφού είναι όμως το τελευταίο επίπεδο, θέλουμε η action να πραγματοποιηθεί μετά το else που έχουμε ήδη βάλει και επειδή υπάρχουν κ άλλες actions που θέλουμε να γίνουν μετά το else, θα τις βάλουμε όλες μαζί μέσα σε ένα block εντολών. Οπότε, μετά το else, θα πρέπει να προσθέσετε τα εξής:

Actions(control)-> Start Block

Actions(main2) -> Display message

Actions(control)-> End Block (στο τέλος)

Για την εμφάνιση τώρα των highscores θέλουμε ακόμα μία action που θα γίνει μετά την εμφάνιση του προηγούμενου μηνύματος. Οπότε μέσα στο ίδιο block εντολών:

Actions(score) -> Show highscore (επιλέγουμε τα χαρακτηριστικά που θέλουμε).

➤ Ζωές

Στο σημείο αυτό θα πρέπει να κάνετε κάποια βήματα, έτσι ώστε να προστεθούν ζωές στο παιχνίδι σας!

1. Πηγαίνετε στο obj_controller_start και επιλέξτε:
 - > Add Event -> Create
 - > Actions(score) -> Set lives (3)
2. Πηγαίνετε στο obj_controller_main και επιλέξτε:
 - > Add Event -> Draw
 - > Actions(draw) -> Draw text (x: 300, y: 0, check relative)
 - > Actions(score) -> Draw life images (x: 345, y:0)
 - > Add Event -> Other -> No more Lives
 - > Actions(main2) -> Display message(<No more lives>)
 - > Actions(score) -> Show highscore
 - > Actions(main2) -> Restart Game
3. Τέλος, θα δημιουργήσετε το sprite spr_lives, επιλέγοντας από τα διαθέσιμα το αντίστοιχο.

Πηγαίνετε στο obj_controller_main και στο Draw live images και επιλέξτε το spr_lives.

Μπορείτε να διαμορφώσετε το μέγεθος του εικονιδίου ως εξής: spr_lives -> Edit -> Transfom -> Scale (σμίκρυνση με τη μπάρα).

➤ Αντίπαλοι

Τώρα, θα εξελίξουμε το παιχνίδι μας, προσθέτοντας αντιπάλους!

Αρχικά, θα δημιουργήσετε το εικονίδιο spr_skull, spr_skeleton, spr_statue2 και μετέπειτα ένα αντικείμενο obj_skull, χρησιμοποιώντας το αντίστοιχο από τα παραπάνω sprite.

Το αντικείμενο που δημιουργήσατε θα πρέπει να έχει τις εξής ιδιότητες:

Add event -> Create

-> Actions(move) -> Move fixed (αριστερά, δεξιά, ταχύτητα 10)

Add event -> End Step

-> Actions(control) -> Test variable

-> Actions(main1) -> Change Sprite (spr_skeleton)

-> Actions(control) -> else

-> Actions(main1) -> Change Sprite (spr_statue2)

Add event -> Collision με obj_wall

-> Actions(move) -> Reverse Horizontal

Επίσης, το obj_guy θα πρέπει να έχει, επιπρόσθετα, τις εξής ιδιότητες:

Add event -> Collision με obj_skull

-> Actions(main1) -> Play sound (φτιάχνουμε αντίστοιχο ήχο)

Υποστήριξη της διδακτικής διαδικασίας του μαθήματος «Εφαρμογές Πληροφορικής»

- > Actions (main2)-> Sleep
- > Actions(score)-> Set lives (-1, check relative)
- > Actions(main1)-> Restart Room

Προσθέστε αντικείμενα obj_skull και στα 2 δωμάτια!

➤ **Αύξηση ζωής**

Δημιουργούμε ένα εικονίδιο spr_smile και αντίστοιχο αντικείμενο obj_smile με τις εξής ιδιότητες:

- Add Event -> Collision με obj_guy
- > Actions(score)-> Set lives (1, relative)
 - > Actions(main1)-> Destroy the instance

Προσθέστε αντικείμενα obj_skull και στα 2 δωμάτια!

4. Εισαγωγή στο προγραμματιστικό περιβάλλον Appinventor – Δραστηριότητες και φύλλα εργασίας

Το App Inventor for Android αποτελεί ένα νέο δωρεάν οπτικό περιβάλλον προγραμματισμού με πλακίδια (blocks), για τη δημιουργία εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα Android ^[11]. Οι συγκεκριμένες εφαρμογές τρέχουν και σε emulator. Αναπτύχθηκε στα εργαστήρια της Google από μια ομάδα με επικεφαλής τον καθηγητή του MIT Hal Abelson (Abelson, 2009). Το App Inventor χρησιμοποιείται και δοκιμάζεται ήδη ως πλατφόρμα διδασκαλίας και εισαγωγής στον προγραμματισμό, τόσο στην τριτοβάθμια όσο και στη σχολική εκπαίδευση. Το περιβάλλον του App Inventor έχει πολλές ομοιότητες με το περιβάλλον του Scratch και του Alice, με τη διαφορά ότι οι εφαρμογές που δημιουργούνται τρέχουν σε έξυπνα τηλέφωνα (smart phones).

Σε πρόσφατη έρευνα η οποία πραγματοποιήθηκε από την οργάνωση Ανθρωπιστική δράση με Δωρεάν Λογισμικό Ανοικτού Κώδικα (Humanitarian Free and Open Source Software –HFOSS) και χρηματοδοτήθηκε από το Εθνικό Ίδρυμα Επιστημών των Ηνωμένων Πολιτειών (National Science Foundation), διαπιστώθηκε πως το app inventor είναι κατάλληλο για τη διδασκαλία του προγραμματισμού και ειδικότερα της υπολογιστικής σκέψης στο σχολείο, αφού αρχικά είναι εύκολο στη χρήση περιβάλλον με πολλές δυνατότητες. Επιπλέον, είναι ένα αντικειμενοστραφές μοντέλο οπτικού προγραμματισμού με δομές ελέγχου καθοδηγούμενες από γεγονότα (event-driven) το οποίο βοηθάει στη μάθηση μέσω της λύσης προβλημάτων. Ακόμη, δίνει επιπλέον κίνητρα στους μαθητές σε σχέση με τα περιβάλλοντα Scratch και Alice εξαιτίας της φορητότητας και της πρακτικής χρήσης των εφαρμογών που δημιουργούνται. Τέλος, η ύπαρξη emulator σημαίνει ότι δεν χρειάζονται πολλές συσκευές για την εισαγωγή στη σχολική τάξη.

4.1 Δραστηριότητες και Φύλλα Εργασίας

4.1.1 1^ο Φύλλο εργασίας [12]

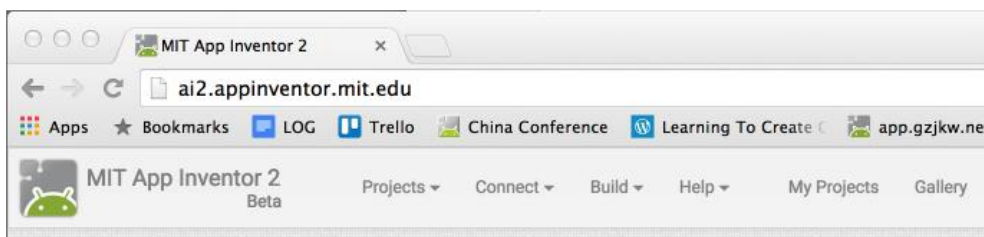
Καθώς οι μαθητές είναι πολύ εξοικειωμένοι με την αγγλική γλώσσα και μπορούν με ευκολία να κατανοήσουν οδηγίες και να τις ακολουθήσουν, η συγκεκριμένη δραστηριότητα τους δόθηκε στα αγγλικά. Τα παιδιά μπόρεσαν και ολοκλήρωσαν επιτυχώς κάθε βήμα της εργασίας με την ελάχιστη δική μας καθοδήγηση.

This is the way your I Have a Dream app will appear. When the user clicks on the images, the app will play a speech by Martin Luther King.



Εικόνα 42: I have a dream application 1

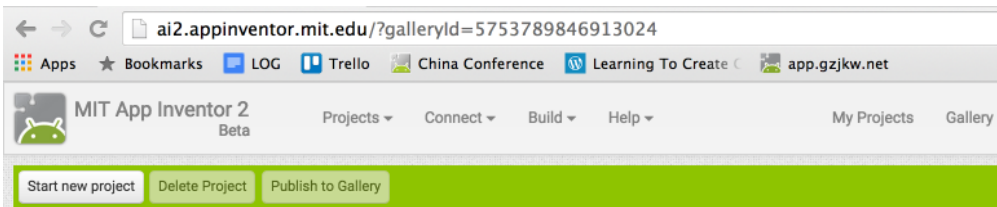
1. Begin by opening a browser window to App Inventor: ai2.appinventor.mit.edu.



Εικόνα 43: Άνοιγμα Browser

If you have WiFi and an Android device, you can build apps with App Inventor completely "in the cloud", i.e., without downloading any software to your computer. You login to the system with a Google account. If you were going to start an app from scratch, you would begin from here. But we'll use a "starter" app (see below).

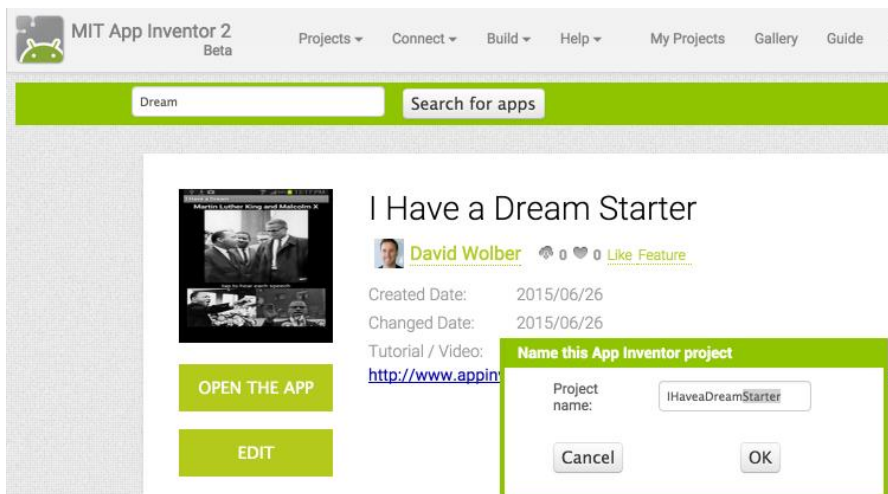
2. For this tutorial, you'll begin with a "starter app" from the App Inventor Gallery



Εικόνα 44: Starter app

App Inventor has a gallery for sharing apps. Click this link <http://ai2.appinventor.mit.edu/?galleryId=5753789846913024> and App Inventor will open to the I Have a Dream Starter app. The starter app has no code-- it just contains the image files and sound clips you need for the app. It's provided so you can learn the coding without spending time finding media files on the Internet. Later, you can build your own soundboard app with your own images and sound clips.

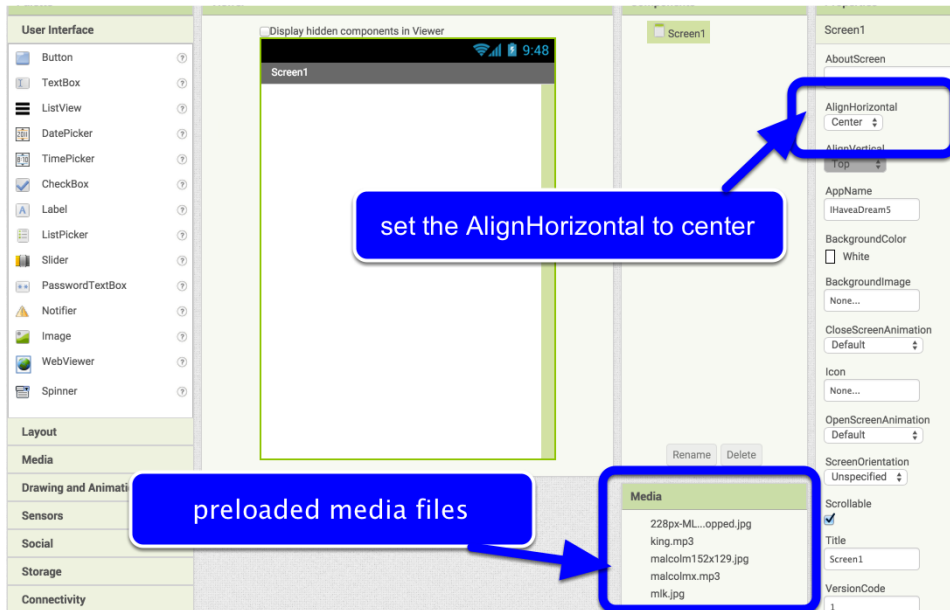
3. Open the starter app



Εικόνα 45: Open starter app

Click "Open the App" to open the starter app. You'll be prompted to name your app. Name it something like "IHaveADream", e.g., remove the "Starter" from the name. Note that app names cannot have spaces or other punctuation.

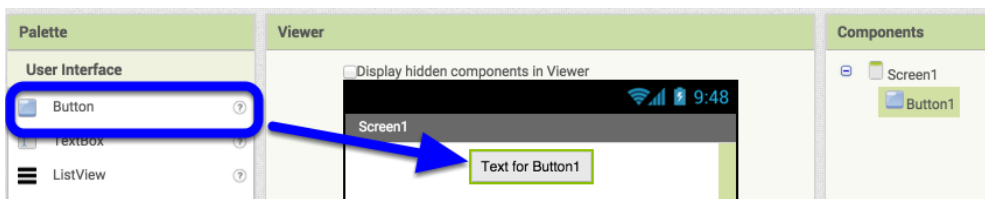
4. Check out the Designer and align the screen



Εικόνα 46: Align

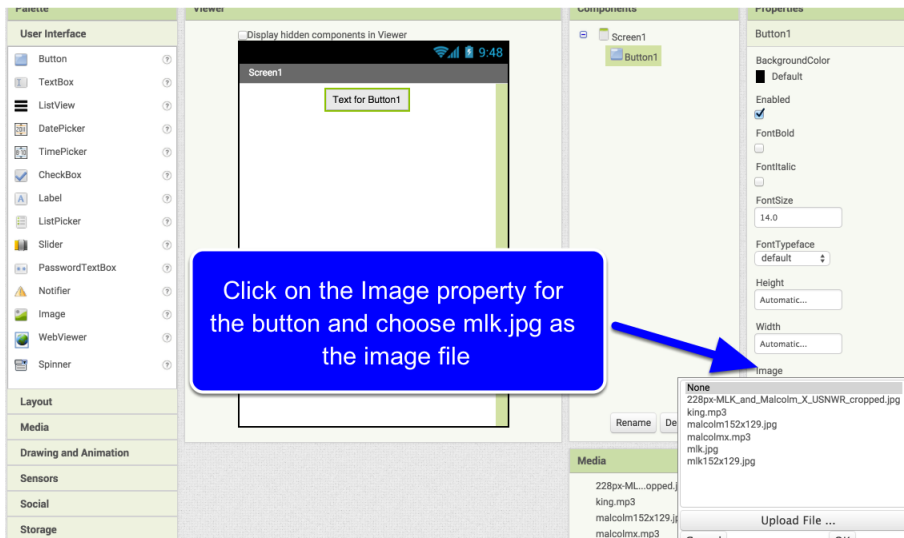
You'll specify how the app should look, and all the components you'll use, in the App Inventor Designer, which is shown above. As you can see, the screen starts out as blank, but the starter app has some media files preloaded to help you get going. Note that you can add your own media files as well, you're just not doing that in this tutorial. To get things started, set the Screen's AlignHorizontal property to "Center". When we add the pictures of the app, they'll automatically be centered within the screen.

5. Drag a button onto the screen



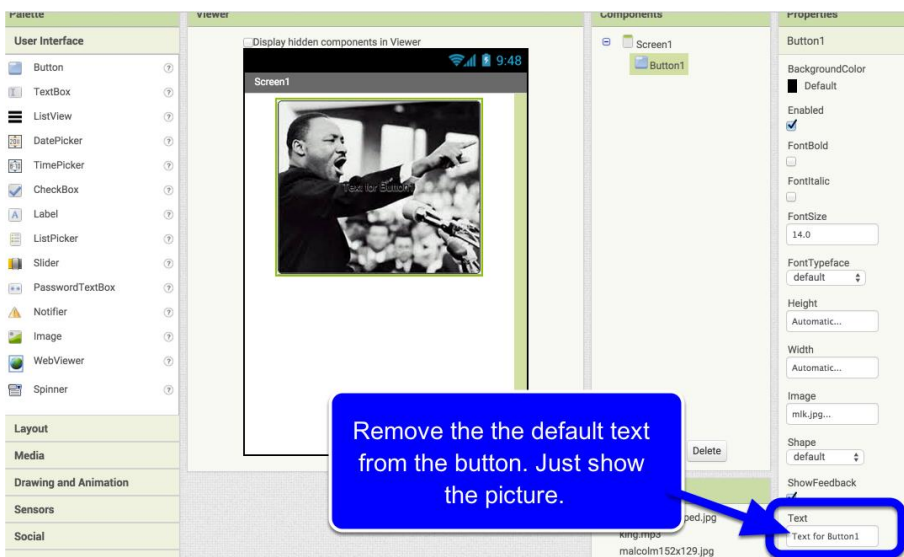
Next, add a Button to the screen. Click down on the Button in the palette, then drag one out and into the screen. The button will appear with some default text. In the next step, you'll change it so it shows a picture.

6. Set the button's image property so that it shows MLK



Εικόνα 47: Image property

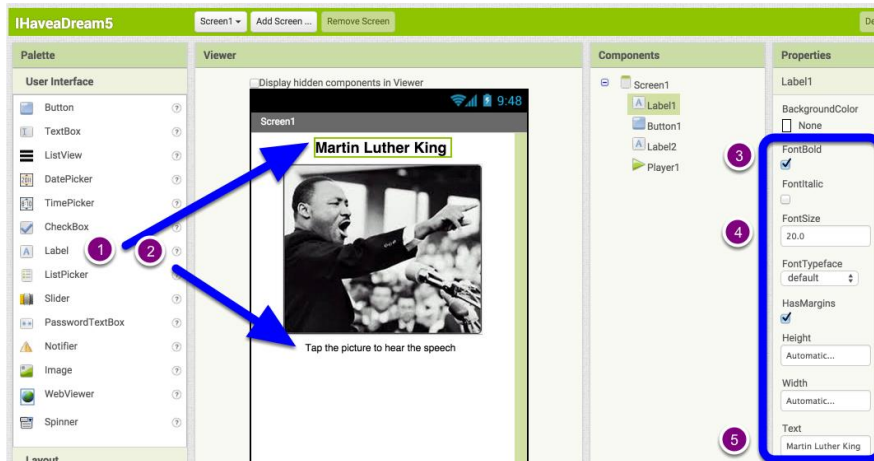
7. Remove the default text from the button



Εικόνα 48: Button text

When you set the Image property for the button, the picture appears but the default text for the button is still there. Click on the Text property and set it to blank (backspace to beginning).

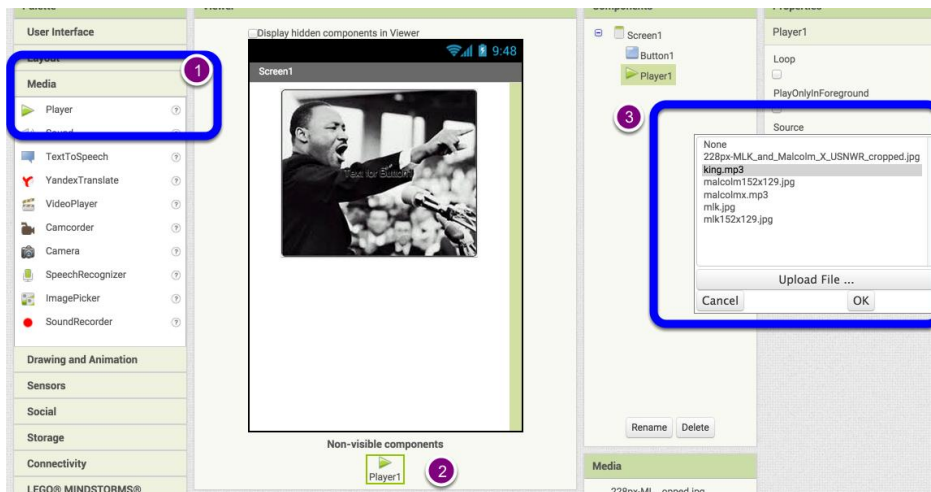
8. Add two labels above and below the button.



Εικόνα 49: Labels

Drag two labels onto the screen, one above the button and one below. Set the Text property of the labels as shown. For the top label, check the FontBold property and change the FontSize to 20.0.

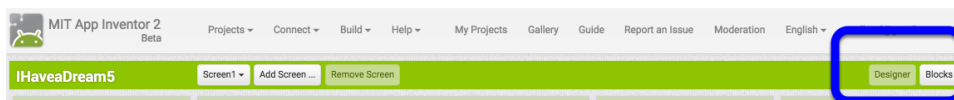
9. Add a Player component for playing the speech



Εικόνα 50: Player component

Open the Media drawer (1) and drag a Player component into the app (2). A Player component knows how to play sound clips and is considered a "non-visible" component because it doesn't show up in the user interface of the app. After dragging in the Player component, set its Source property to king.mpg (the sound clip of MLK's speech).

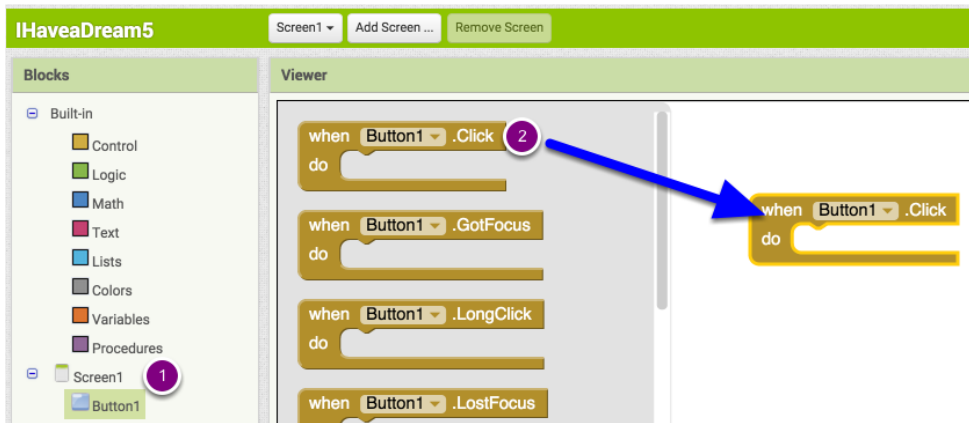
10. Open the Blocks Editor to program the interactive behavior



Εικόνα 51: Block editor

Thus far, you have completed the design of part 1 of the app, but you still need to code the interactive behavior of the app. In the right top-corner, you can switch between the Design view and the Blocks view. Click on "Blocks" and open the empty Blocks view.

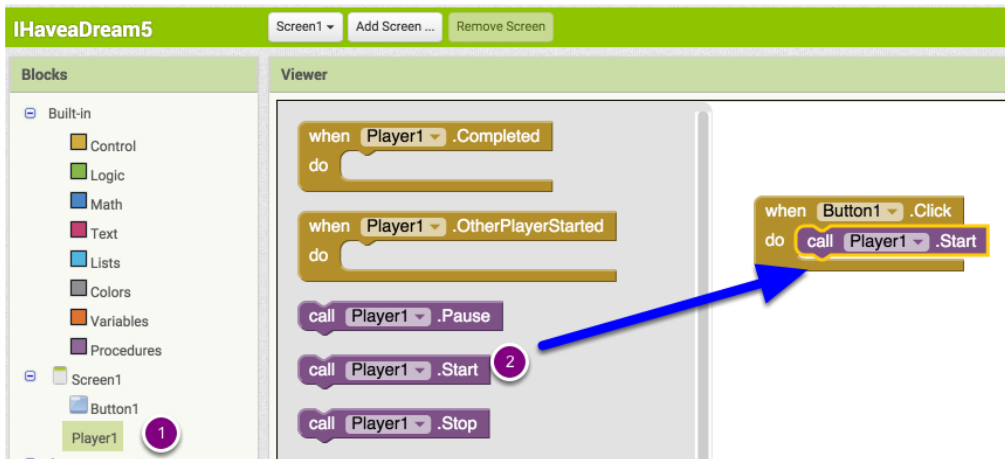
11. Drag out a Button.Click event handler



Εικόνα 52: Button.Click event handler

In the Blocks editor, click on Button1 (1) and then drag out a Button1.Click event handler (2). You are preparing to show that when the button is clicked, the speech will be played.

12. Place a Player1.Start block within the Button1.Click event handler.



Εικόνα 53: Player1.Start block

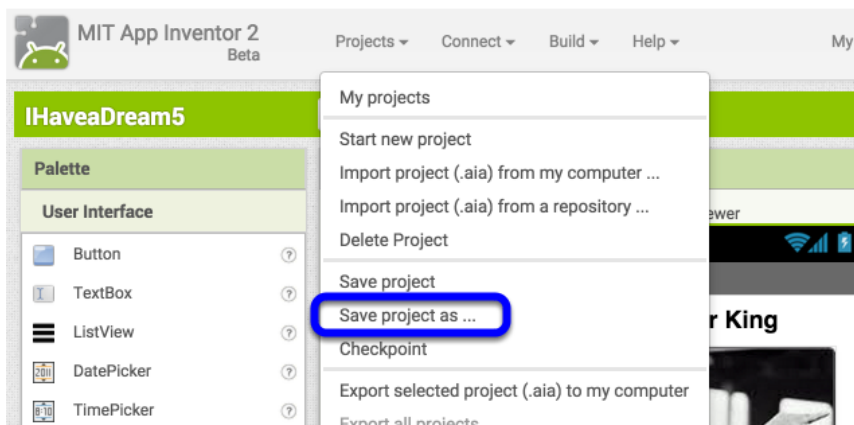
Click on the Player1 drawer (1) and then drag out a "call Player1.Start" block and place it in the event handler. You are specifying that when the button is clicked, the speech should be played. Remember, the Player1.Source property is set to the king.mpg file, so that will play.

Now that you have your feet wet, let's make the app more fun. MLK and Malcolm X were Civil Rights leaders of great contrast. Let's build an app that also plays an app by Malcolm X in order to show this contrast. You'll need to modify the UI, then the behavior. Behavior-wise, things will become more complicated as you'll need to make sure the speeches don't overlap.



Εικόνα 54: I have a dream application 2

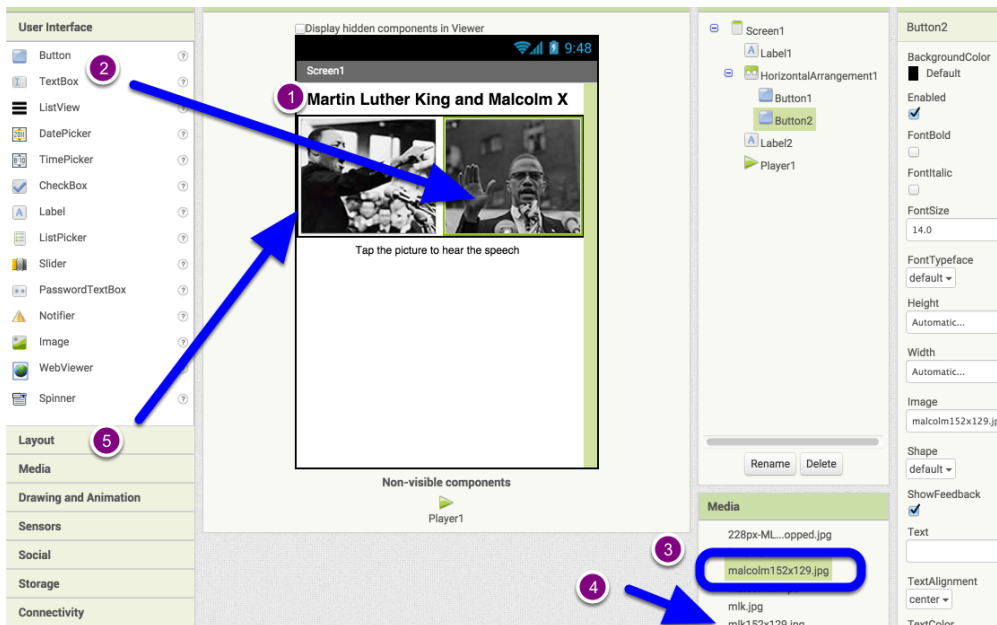
13. SaveAs your project to create a new project



Εικόνα 55: SaveAs

Choose Project | SaveAs and name your new app something else (e.g., DreamWithNoName).

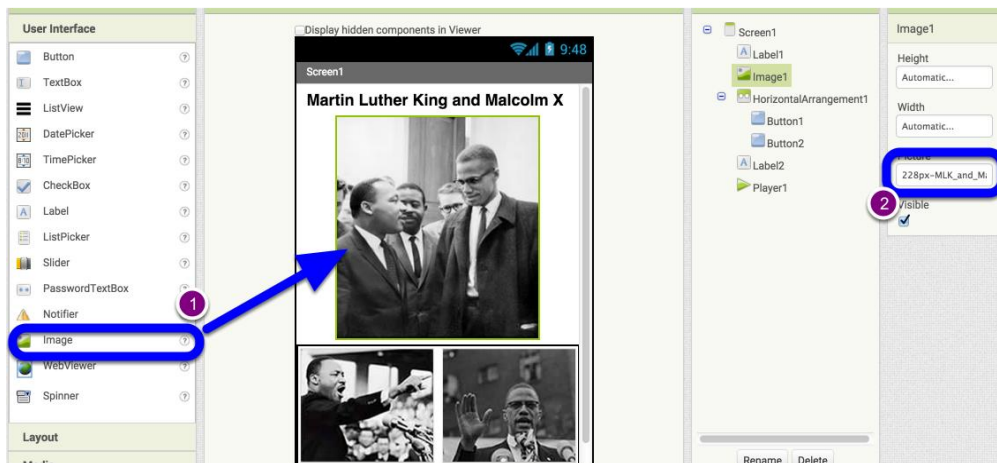
14. Choose a smaller image for MLK and add an image of Malcolm X.



Εικόνα 56: Adding images 1

In the designer, modify the user interface. First, change the text of the top label so it mentions Malcolm X (1). Then drag out a second button for Malcolm X (2) and set its picture property as shown (3). Then change the Image for MLK's button to the smaller image (4). Finally, drag out a HorizontalArrangement from the Layout drawer (5), and drag the two buttons into it so that they appear side-by-side as shown.

15. Add the image of both leaders



Εικόνα 57: Adding images 2

Add an Image component (1) then set its Picture property to the picture of both leaders. Note that the top component is an Image not a Button because nothing should happen when the user clicks it.

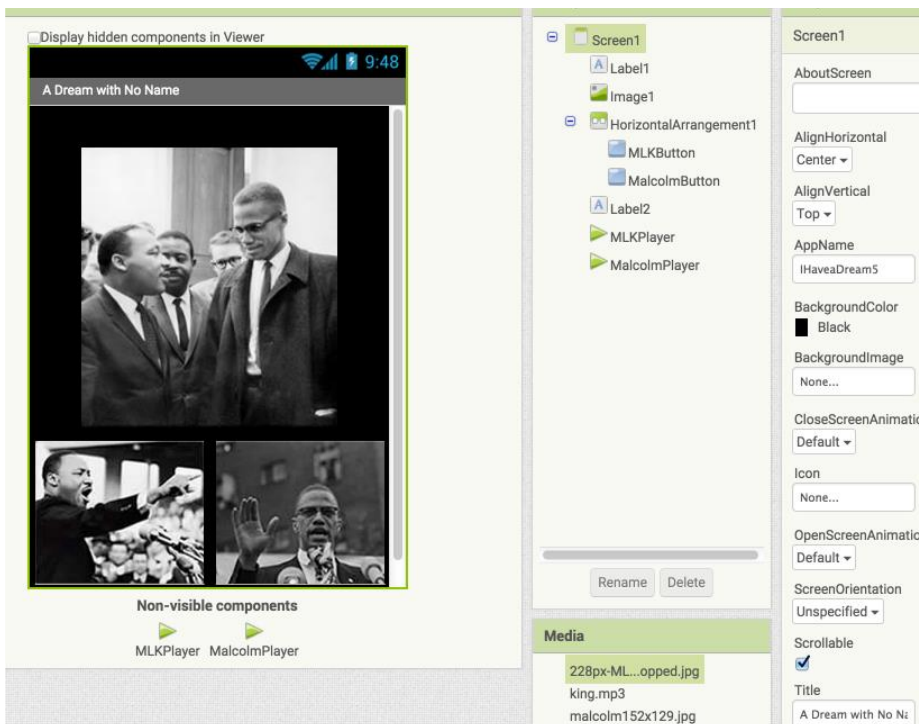
16. Add a second player and rename components



Εικόνα 58: Second player component

Drag out another Player component (1) and set its Source property to Malcolm's speech (2). Then rename the Button and Player components (3) so we can distinguish them easily in the blocks editor. A rule-of-thumb is to give a descriptive name with a suffix which is the component type, e.g., MalcolmPlayer.

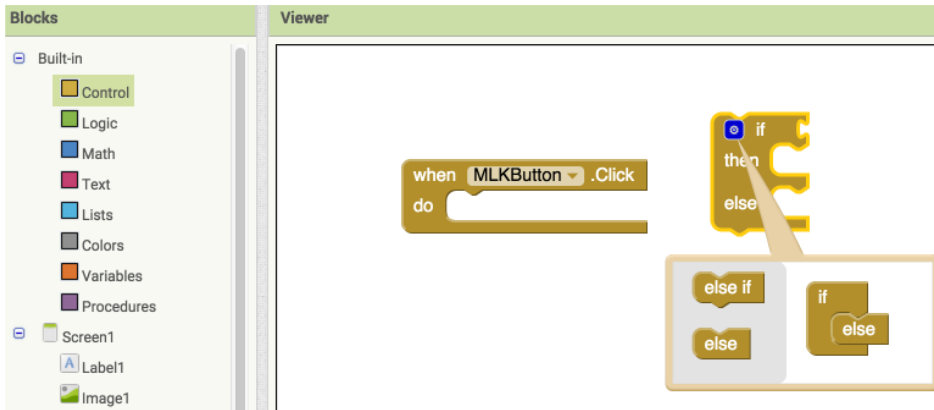
17. Add a Screen Title and Background Color



Εικόνα 59: Screen title and background color

To complete the UI, set the BackgroundColor to black and set the screen's Title property. You can also set the Screen's Icon property to an image that you want to show on a device representing the app. Now you're ready to program the behavior.

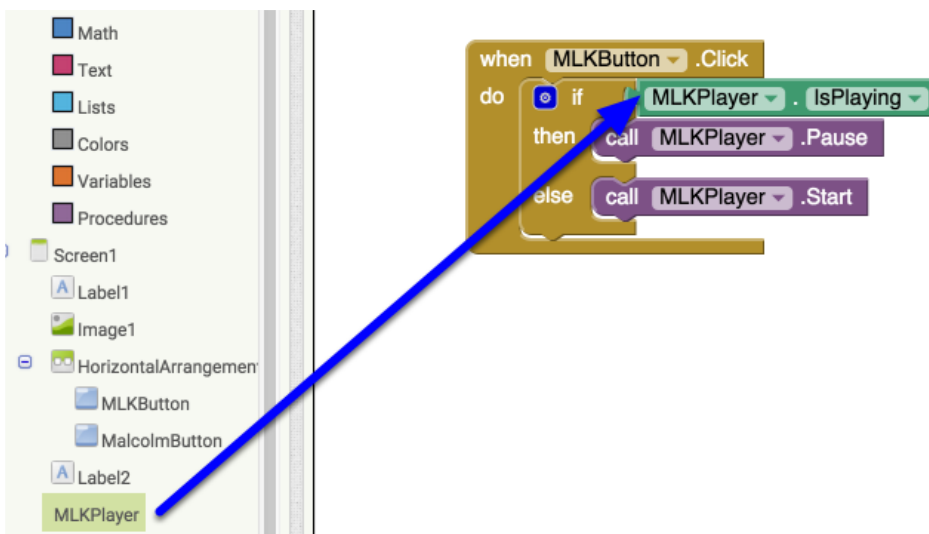
18. Code it so you start/pause the MLK speech



Εικόνα 60: Start/Pause code

Open the Blocks editor. Note that the Button1.Click event and other blocks have been renamed based on your renames in the designer. First, focus on the MLK button. The behavior we want is for the first click to start the speech, the next one pause it, next one start it, and so on. So you don't always want the same thing to happen when the button is clicked. To program this, you use an if-else block. If-else allows the app to ask questions, such as, "is the speech already playing?". To code this, drag out an if-block from the Control drawer, then click on the blue "modifier" icon. The modifier allows you to add branches. In this case, there are two possibilities, start or pause, so drag an "else" within the picture to its right so that there are two branches as shown.

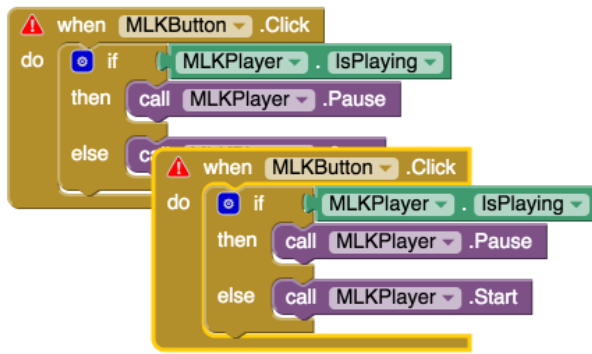
19. Code it so the if-else starts/pauses MLK's speech.



Εικόνα 61: If/Else start/pause code

Place the if-else block in the event handler. Then drag out an MLKPlayer.IsPlayingproperty block from the MLKPlayer drawer. This block is true if the speech is playing, false if not. If it is playing, you want to pause the speech, so drag out this block from MLKPlayer. If it is not playing, the "else" branch will be taken, so MLKPlayer.Start is called. In this code, you're seeing two new things, both fundamental to software. First is the conditional (if-else) block. Conditional blocks provide the mechanism for the app to ask questions, the basis of artificial intelligence. The second new thing is you are checking a component property in the blocks. You set a few properties in the designer (e.g., the Screen.Title). Here you are checking the Player's IsPlaying property in the blocks.

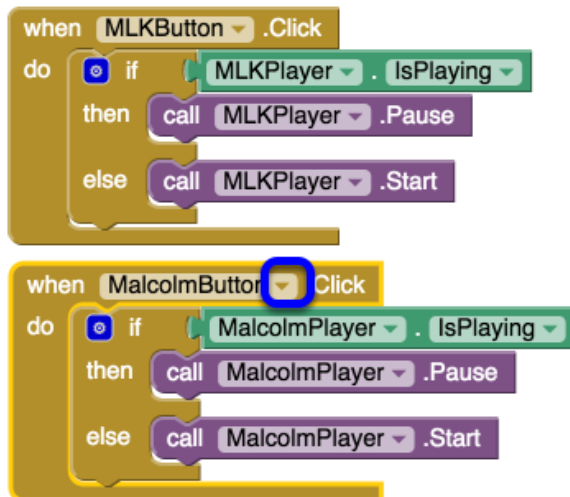
20. Copy-Paste the MLKButton event handler



Εικόνα 62: Duplicate event handler

You want the Malcolm button to behave similarly to the MLK button, so copy-paste the event handler. Select the MLKButton.Click event handler and copy/paste it using command-c and command-v on Mac (ctrl-c and v on Windows). A red error triangle will appear because you can't have two handlers for the same event. Next, you'll fix this.

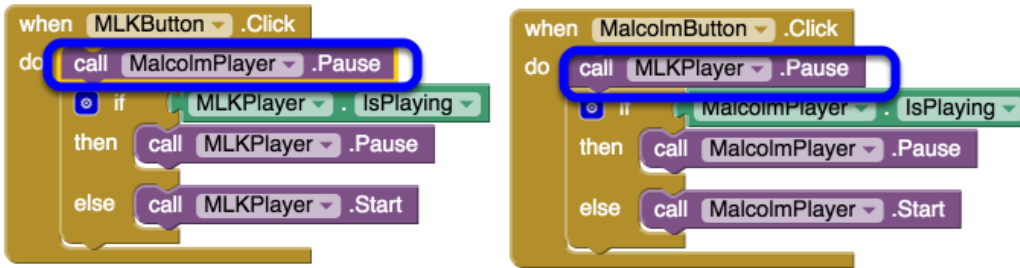
21. Change the blocks to refer to MalcolmButton and MalcolmPlayer



Εικόνα 63: Changing Blocks 1

Using the upside down triangle widget circled above, switch the copied blocks to refer to MalcolmButton and MalcolmPlayer.

22. Program it so that you pause the "other" speech



Εικόνα 64: Changing blocks 2

You just need to pause Malcolm when MLK speaks, and vice-a-versa. The final blocks for the app are shown above with the additional blocks circled. Test the app again (you should still be connected). Does it work how you'd expect? This app is simple, but indicative of the architecture of most apps. This one consists of two event handlers. Each event handler consists of an event (a click in this case) and a response which is a sequence of blocks (they're executed in order). Some blocks in the response are only conditionally executed. So you've not only created an app, you've learned some programming lingo. Next time your hanging out with your friends, smack talk a bit about "event handlers" and "if-else conditionals".

4.1.2 2^ο Φύλλο εργασίας ^[15]

Ενδεικτική Διάρκεια: 45 λεπτά (1 διδακτική ώρα)

Διδακτικός στόχος

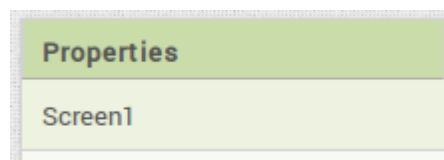
Ακολουθώντας τα βήματα αυτού του φύλλου εργασίας, ζητήθηκε από τους μαθητές να αναπτύξουν μια μικρή εφαρμογή για Android συσκευές (κινητά ή tablets), η οποία θα τους εξοικειώσει με χρήσιμες λειτουργίες που θα αξιοποιήσουμε και στη συνέχεια.

Θα αναπτυχθεί σταδιακά μια **εφαρμογή «συναγερμός»**, χρησιμοποιώντας έναν σκύλο που δεν θα επιτρέπει σε κανέναν να πειράζει την συσκευή μας.

Η παρακάτω εφαρμογή είναι απλή, πρόκειται για ένα εισαγωγικό παράδειγμα. Οι οδηγίες που ακολουθούν είναι αναλυτικές, με σκοπό να προσφέρουν μια στοιχειώδη εξοικείωση με το περιβάλλον του AppInventor.

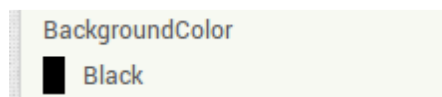
1. Ανοίξτε μια νέα εφαρμογή και ονομάστε την “GuardDog”

Αρχικά, θα πρέπει να βρίσκεστε στην ενότητα **Designer**, στην οποία σχεδιάζουμε τη *διεπαφή* (interface) της εφαρμογής μας, προσθέτοντας τα απαραίτητα *συστατικά* (components) και ορίζοντας *ιδιότητες* (properties) για αυτά. Το μοναδικό συστατικό μέχρι στιγμής, είναι η οθόνη (Screen1). Προτού προσθέσετε άλλα συστατικά, θα κάνετε ορισμένες απαραίτητες τροποποιήσεις στις ιδιότητες της οθόνης, που βρίσκονται στο πλαίσιο **Properties**, στα δεξιά.



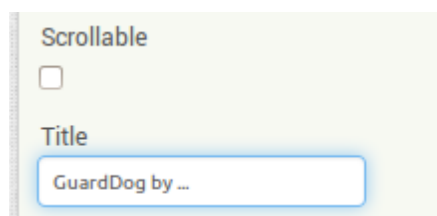
Εικόνα 65: Properties Screen1 AppInventor

2. Επιλέξτε το χρώμα του φόντου (BackgroundColor) να είναι μαύρο.



Εικόνα 66 : Background Color

3. Η ιδιότητα Scrollable δεν πρέπει να είναι επιλεγμένη. Στον τίτλο (Title) προσθέστε το όνομα της ομάδας σας, τα ονόματά σας ή τα αρχικά σας.



Εικόνα 67: Property Scrollable

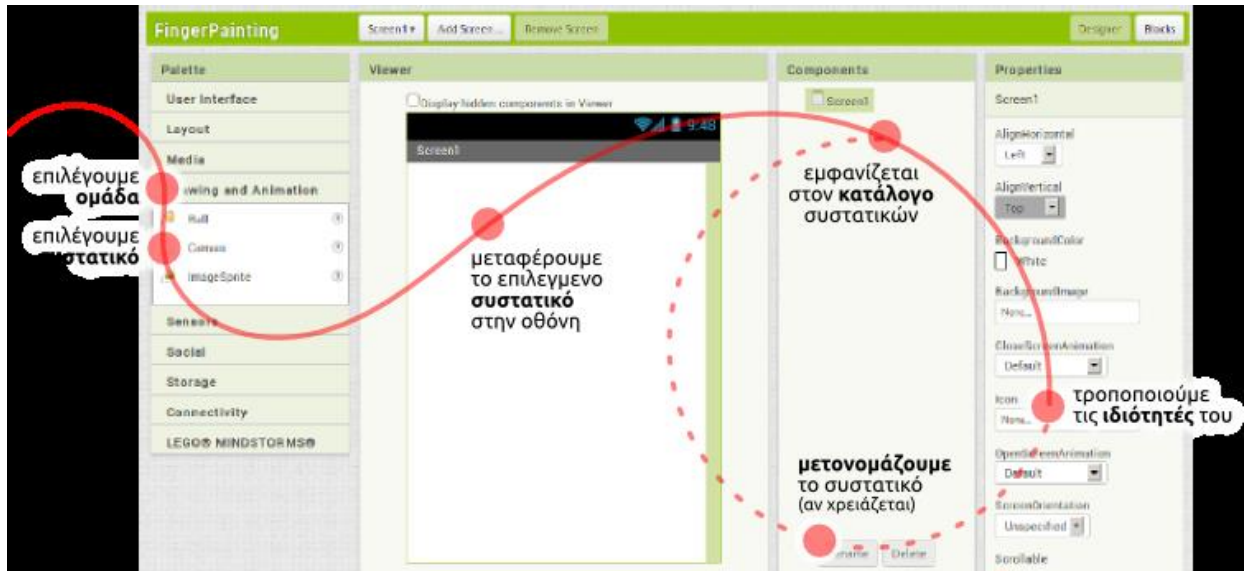
Συνεχίζοντας, θα προσθέσετε το συστατικό που θα αποτελέσει το χώρο όπου θα κινείται ο σκύλος και θα καθορίσετε τις ιδιότητες του χώρου αυτού.

Ορισμός

Ο καμβάς (Canvas) είναι μια ορθογώνια επιφάνεια, εντός της οποίας μπορούμε να χειριζόμαστε φιγούρες (Sprites) ή να σχεδιάζουμε αγγίζοντάς την.

Προκειμένου να ορίσουμε μια περιοχή στην οποία θα είναι δυνατή η μετακίνηση της φιγούρας του σκύλου, θα προσθέσουμε στην οθόνη μας έναν καμβά.

4. Σύρετε, λοιπόν, από την παλέτα αριστερά (Palette), από την ομάδα πλακιδίων Drawing and Animation, το συστατικό Canvas, και αποθέστε το στην οθόνη.



Εικόνα 68: Οθόνη Designer

Είναι σημαντικό (αλλά όχι υποχρεωτικό), να ονομάζουμε τα συστατικά που χρησιμοποιούμε με τέτοιο τρόπο, ώστε να τα αναγνωρίζουμε ευκολότερα.

5. Μετονομάστε τον καμβά Canvas1 σε DogCanvas.

Παρατήρηση: Τα ονόματα των συστατικών πρέπει να αποτελούνται από λατινικούς χαρακτήρες, αριθμούς ή κάτω παύλες και πρέπει να ξεκινάνε με χαρακτήρα, οπότε δεν μπορούμε να χρησιμοποιήσουμε π.χ. ελληνικούς χαρακτήρες ή κενά.

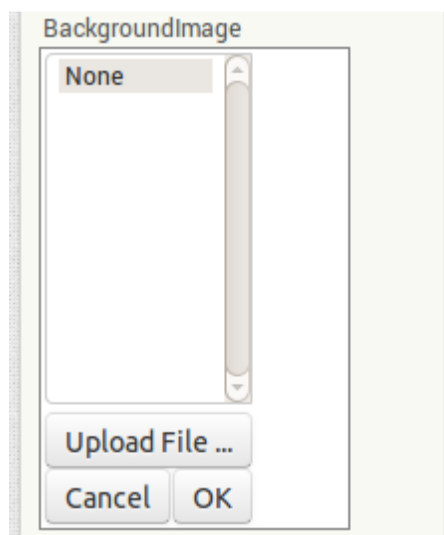
6. Μεταβάλλετε τις ιδιότητες του καμβά DogCanvas ως εξής:

- Width: Fill Parent
- Height: Fill Parent
- BackgroundImage: Grass.jpg

Η επιλογή Fill Parent για τις ιδιότητες Width και Height του καμβά του επιτρέπει να επεκταθεί και να καταλάβει όλο το διαθέσιμο χώρο. Έτσι, ο σκύλος θα μπορεί να κινείται κατά μήκος και κατά πλάτος όλης της οθόνης.

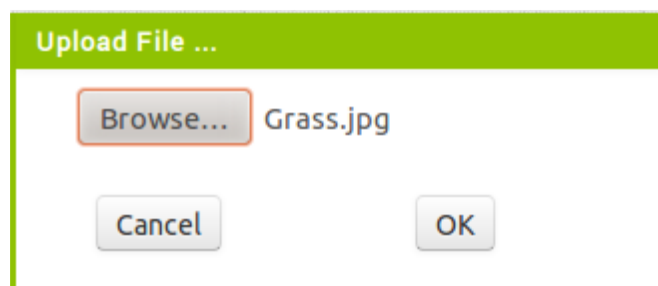
Για να χρησιμοποιηθεί ως φόντο του καμβά DogCanvas η εικόνα με το γρασίδι(Grass.jpg), θα πρέπει πρώτα αυτή να «ανέβει» στο project μας.

7. Κάντε κλικ στην ιδιότητα BackgroundImage και επιλέξτε Upload File.



Εικόνα 69: Select Background Image

8. Κάνετε Αναζήτηση (Browse...) κι επιλέξτε το αρχείο Grass.jpg.



Εικόνα 70: Upload Image

Παρόμοιες κινήσεις κάνουμε για κάθε αρχείο εικόνας ή ήχου που σκοπεύουμε να χρησιμοποιήσουμε στο project, ενώ παρακάτω περιγράφεται κι ένας εναλλακτικός τρόπος να ανεβάζουμε αρχεία στο project μας.

Οι φιγούρες τοποθετούνται εντός του καμβά και μπορούν να αλληλεπιδρούν με τον καμβά, με άλλες φιγούρες εντός του καμβά και φυσικά να αντιδρούν στις δικές μας ενέργειες.

9. Προκειμένου να προσθέσετε τον σκύλο, σύρετε μια φιγούρα (ImageSprite) μέσα στον καμβά. Μέσω της ιδιότητας Picture της φιγούρας, ορίστε σαν εικόνα της την ήρεμη εκδοχή του σκύλου, κατά παρόμοιο τρόπο με παραπάνω.

10. Μεταβάλλετε τις ιδιότητες της φιγούρας ως εξής:

- Interval: 10
- Picture: SleepyDog.jpg
- Rotates: no

Αν ο σκύλος φαίνεται πολύ μεγάλος, μπορούμε να αλλάξουμε τις διαστάσεις του σε pixel, μέσω των ιδιοτήτων Width και Height.

Συνεχίζοντας, θα προστεθεί ο ήχος του γαβγίσματος, καθώς και οι πρώτες εντολές, ώστε ο σκύλος να γαβγίζει όταν αγγίζουμε την οθόνη.

Όταν η εφαρμογή μας χρησιμοποιεί ήχους τότε χρειάζεται να προσθέσουμε τους ήχους αυτούς, καθώς και τα συστατικά ήχου (Sound) που αναλαμβάνουν την αναπαραγωγή τους.

11. Για να μπορεί ο σκύλος να γαβγίζει, προσθέστε ένα συστατικό Sound της ομάδας Media, σέρνοντάς το στην οθόνη. Αυτό θα αναπαράγει τον ήχο, όταν το θελήσουμε. Παρατηρήστε ότι δεν εμφανίζεται εντός της οθόνης, αλλά κάτω από αυτήν, αποτελώντας για τον χρήστη της εφαρμογής ένα μη-ορατό συστατικό (non-visible component).

12. Μετονομάστε το σε DogBarkingSound και ορίστε ως πηγή του ήχου το αρχείο Bark.mp3, μέσω της ιδιότητας Source.

13. Ορίστε την ιδιότητα MinimumInterval σε 300.

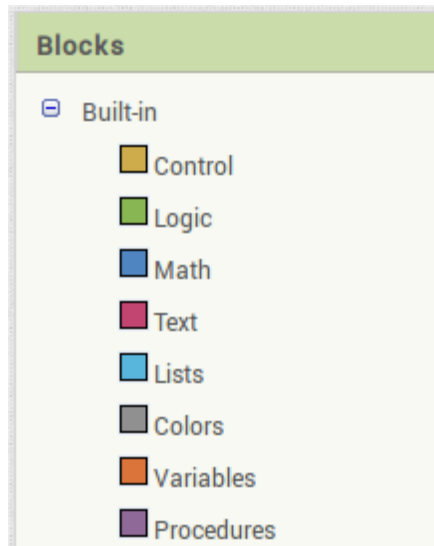
Η ιδιότητα MinimumInterval είναι ο ελάχιστος χρόνος πριν την επανάληψη του ήχου. Αν, λοιπόν, θέσουμε MinimumInterval: 300 msec (0,3 sec) τότε ο ήχος δεν θα μπορεί να ξαναπαίξει προτού περάσουν τουλάχιστον 0,3 δευτερόλεπτα.

Παρατήρηση: Το συστατικό Sound, ακόμα κι αν δεν ορίσουμε συγκεκριμένο αρχείο ήχου για αναπαραγωγή, είναι απαραίτητο, αν θέλουμε η συσκευή να δονείται σαν αποτέλεσμα κάποιας ενέργειας.

Μέχρι στιγμής, στην καρτέλα Designer έχουμε σχεδιάσει ένα μέρος της «οθόνης» της εφαρμογής μας, αλλά δεν έχουμε ορίσει καθόλου την συμπεριφορά της.

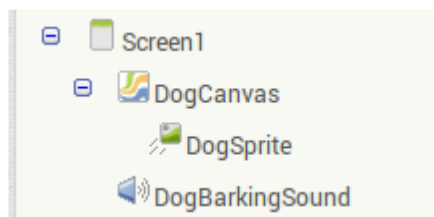
14. Μεταβείτε στην καρτέλα Blocks (το κουμπί μετάβασης βρίσκεται πάνω δεξιά) για να συσχετίσετε ενέργειες με γεγονότα και, ουσιαστικά, να προγραμματίσετε, προσθέτοντας τις κατάλληλες εντολές.

Στα αριστερά της οθόνης διακρίνουμε τρεις ομάδες πλακιδίων (blocks). Πρώτα βρίσκουμε τα ενσωματωμένα (built-in) πλακίδια, την λειτουργία των οποίων θα εξετάσουμε αργότερα.



Εικόνα 71: Build-in πλακίδια

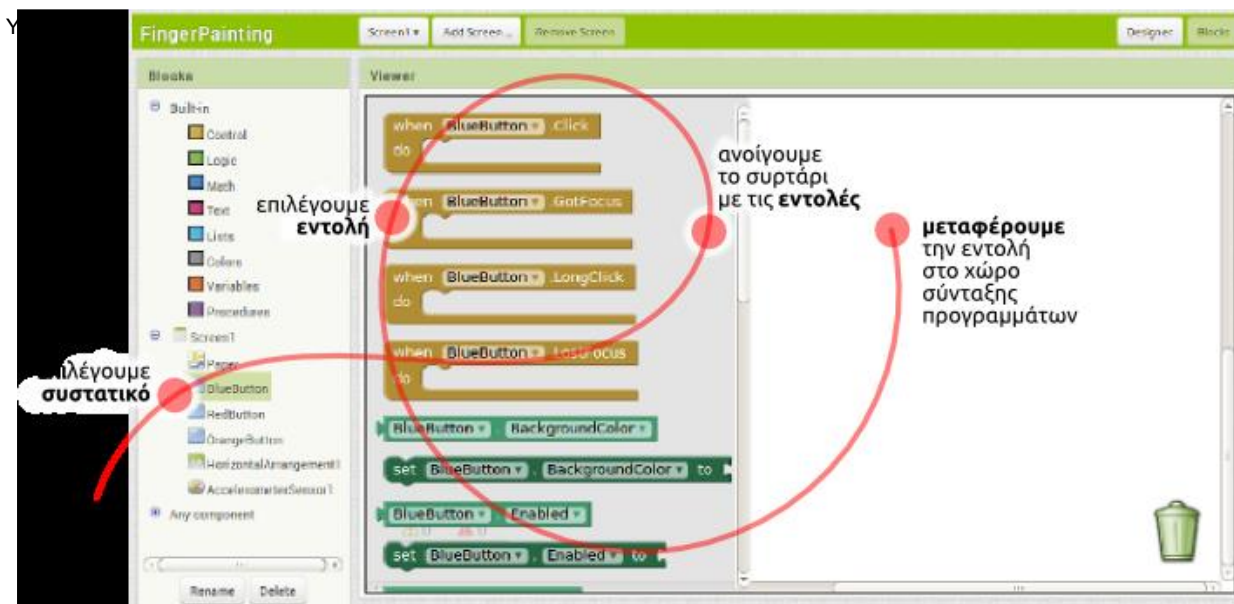
Στη συνέχεια, βλέπουμε τα πλακίδια που αφορούν τα συστατικά που έχουμε προσθέσει στην εφαρμογή μας.



Εικόνα 72: Πλακίδια που έχουν εισαχθεί

Ακολουθεί η τελευταία ομάδα, που δε θα μας απασχολήσει εδώ.

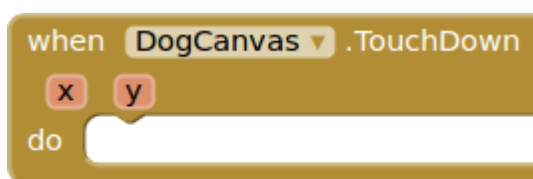
Όταν προγραμματίζουμε μια συγκεκριμένη συμπεριφορά για την εφαρμογή μας, πρακτικά συναρμολογούμε πλακίδια εντολών (και όχι μόνο). Για κάθε\ πλακίδιο που θέλουμε να προσθέσουμε ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης, ανοίγει το αντίστοιχο «συρτάρι» με τις διαθέσιμες εντολές, αναζητούμε και επιλέγουμε το πλακίδιο που χρειαζόμαστε και το σέρνουμε στον χώρο σύνταξης των προγραμμάτων.



Εικόνα 73: Οθόνη Blocks

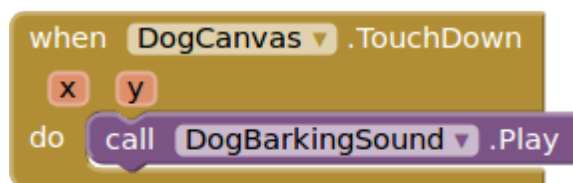
Στην συγκεκριμένη περίπτωση, θέλουμε να ακούγεται ο ήχος του γαβγίσματος DogBarkingSound κάθε φορά που ο χρήστης αγγίζει τον καμβά DogCanvas.

- Επιλέξτε το συστατικό DogCanvas, ανοίξτε το αντίστοιχο συρτάρι και μεταφέρετε το πλακίδιο *when DogCanvas.TouchDown*.



Εικόνα 74: when.....Touchdown

- Επιλέξτε το συστατικό DogBarkingSound, εντοπίστε το πλακίδιο *call DogBarkingSound.Play* και το «κουμπώστε» το μέσα στο προηγούμενο πλακίδιο.



Εικόνα 75: call.....Play

Παρατήρηση: Σημειώστε ότι το πλακίδιο *when DogCanvas.TouchDown* επιστρέφει και δύο νούμερα, τα x και y. Αυτά τα νούμερα προσδιορίζουν που ακριβώς ακούμπησε το δάχτυλό του ο χρήστης στην οθόνη και θα τα χρησιμοποιήσουμε παρακάτω.

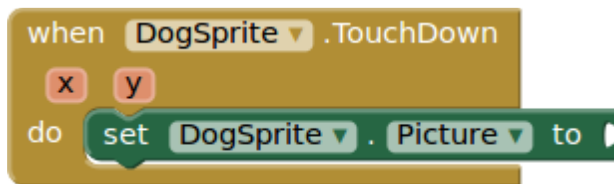
- Επιστρέψτε ξανά στην καρτέλα Designer.

- Προσθέστε στο project τη θυμωμένη εκδοχή του σκύλου (AngryDog.png).

Εδώ θέλουμε η φιγούρα του σκύλου DogSprite να αλλάζει μορφή όταν ο χρήστης την αγγίζει και να επανέρχεται όταν η επαφή σταματά. Όλα τα πλακίδια αφορούν την μορφή DogSprite και θα τα βρούμε στο συρτάρι που εμφανίζεται όταν την επιλέξουμε στην καρτέλα Blocks.

19. Επιστρέψτε στην καρτέλα Blocks.

20. Εντοπίστε και συναρμολογήστε τα παρακάτω πλακίδια.



Εικόνα 76: Συναρμολόγηση πλακιδίων

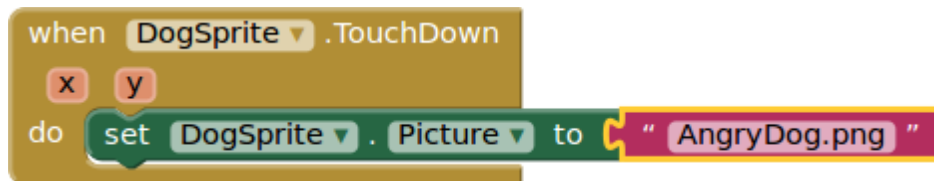
Παρατηρήστε ότι οι εντολές μας είναι ελλειπείς. Πώς θα προσδιορίσουμε την εικόνα που θα έχει η φιγούρα DogSprite όταν αγγιχθεί; Θα πρέπει να παρέχουμε το αντίστοιχο όνομα αρχείου.

21. Ανατρέξτε στην ενσωματωμένη ομάδα πλακιδίων Text και επιλέξτε το πλακίδιο του κενού κειμένου.



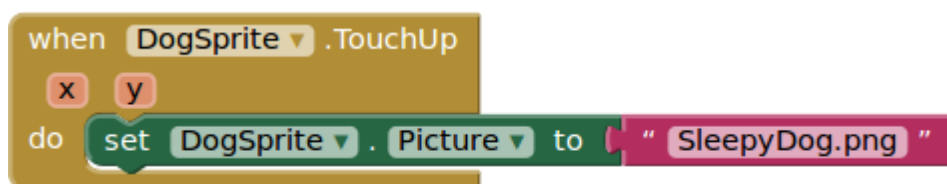
Εικόνα 77: Πλακίδιο κενού κειμένου

22. Συμπληρώστε το όνομα του αρχείου εικόνας "AngryDog.png" και προσθεσέ το στις εντολές μας.



Εικόνα 78: TouchDown

23. Προσδιορίστε την εικόνα στην οποία θα επανέρχεται ο σκύλος όταν σταματάει η επαφή.



Εικόνα 79: TouchUp

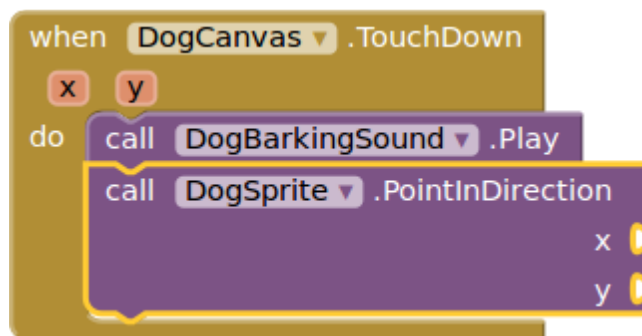
Θέλουμε, ακόμη, όταν ο χρήστης αγγίζει τον καμβά DogCanvas η φιγούρα DogSpite να στρέφεται και να κινείται προς το σημείο επαφής. Παραπάνω, είχαμε προσδιορίσει πως όταν ο χρήστης ακουμπά τον καμβά DogCanvas τότε ακούγεται ο ήχος του γαβγίσματος.

Διαθέτουμε λοιπόν ήδη το `when DogCanvas.TouchDown`, το οποίο προσδιορίζει και το σημείο στο οποίο έγινε η επαφή, δηλαδή την οριζόντια απόσταση x από το αριστερό άκρο του καμβά και την κάθετη απόσταση y από το πάνω άκρο του καμβά. Οι αριθμοί x και y ονομάζονται και *συντεταγμένες* του σημείου επαφής. Θα χρησιμοποιήσουμε τώρα τις συντεταγμένες x και y όπου έγινε η επαφή, για να καθορίσουμε προς τα που πρέπει να στραφεί ο σκύλος.



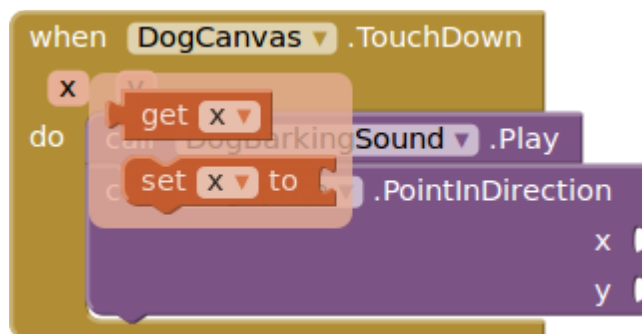
Εικόνα 80: Συντεταγμένες σημείου επαφής

24. Προσθέστε το πλακίδιο `call DogSprite.PointInDirection`, το οποίο στρέφει τη φιγούρα προς ένα συγκεκριμένο σημείο.



Εικόνα 81: `call __.PointInDirection`

25. Για να πάρετε την τιμή της x στην οποία έγινε η επαφή, αφήστε το δείκτη του ποντικιού πάνω από την ετικέτα x κι εμφανίζεται το πλακίδιο `get x`.



Εικόνα 82: x συντεταγμένη

26. Σύρετε και «κουμπώστε» το `get x` στο `x` της `PointInDirection`.

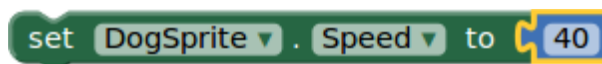
27. Κάνετε το ίδιο και για την `y` συντεταγμένη.

Τώρα απομένει να ασχοληθούμε με την κίνηση του σκύλου. Θα πρέπει, όταν ο χρήστης ακουμπά τον καμβά, να τροποποιήσουμε την ταχύτητά του με το πλακίδιο `set DogSprite.Speed`.

Χρειάζεται ένας αριθμός για να προσδιορίσουμε την ταχύτητα.

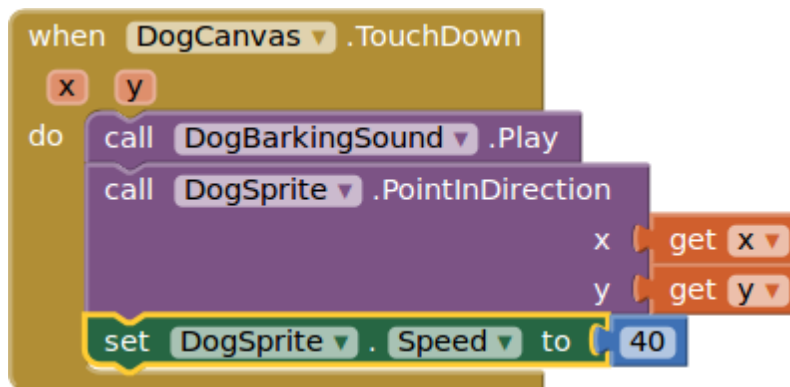
28. Ανατρέξτε στην ενσωματωμένη ομάδα πλακιδίων `Math` κι επιλέξτε το πλακίδιο με τον αριθμό 0.

29. Τροποποιήστε σε 40 και «κουμπώστε» στην `set DogSprite.Speed`.



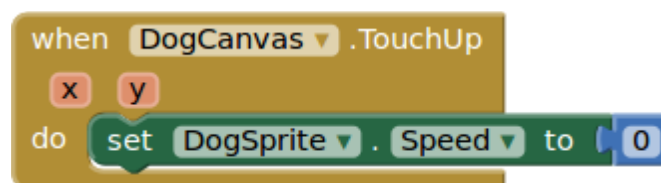
Εικόνα 83: Προσδιορισμός Ταχύτητας

30. Εισάγετε ολόκληρη την εντολή στην `when DogCanvas.TouchDown`.



Εικόνα 84: Ολοκλήρωση εντολής TouchDown

31. Με παρόμοιο τρόπο, μηδενίστε την ταχύτητα του σκύλου όταν το δάχτυλο του χρήστη δεν είναι πια σε επαφή με την οθόνη.



Εικόνα 85: Ολοκλήρωση εντολής TouchUp

Παρακάτω φαίνεται μια εικόνα της τελικής εφαρμογής:



Εικόνα 86: Εφαρμογή Guard Dog



Εικόνα 87: Εφαρμογή Guard Dog-AngryDog

4.1.3 3^ο Φύλλο Εργασίας ^[15]

Ενδεικτική διάρκεια: 2 διδακτικές ώρες(90 λεπτά)

Διδακτικός στόχος

Η εφαρμογή που θα υλοποιήσετε θα πρέπει να περιέχει μια φιγούρα και μία μπάλα. Η **φιγούρα θα κινείται διαρκώς σε τυχαία θέση**. Όταν την **αγγίζουμε** θέλουμε να ακούγεται ένας ήχος. Η μπάλα θα είναι κόκκινη και **θα την σέρνουμε με το δάχτυλό μας** ώστε να αποφύγουμε την σύγκρουση με τη φιγούρα. Όταν **συγκρούεται** με τη φιγούρα θέλουμε να ακούγεται ένας άλλος ήχος. Η εφαρμογή θα μετράει επιτυχημένες προσπάθειες- σκορ όταν πετυχαίνουμε ένα στόχο- φιγούρα.

Μετά την εφαρμογή του φύλλου εργασίας θα πρέπει να είστε σε θέση να:

- Να δημιουργήσετε κίνηση μιας μορφής στην οθόνη
- Να αλλάζετε δυναμικά (καθώς εκτελείται το πρόγραμμα) τις ιδιότητες συστατικών με χρήση `set` και `get` μπλοκ
- Να επιλέγετε διαδικασίες από τις υπάρχουσες για κάθε συστατικό
- Να αναλύεται συμβάντα και να επιλέγετε κατάλληλους χειριστές αυτών
- Να ορίζεται τυχαία θέση στην οθόνη
- Να ορίζετε και να χρησιμοποιείτε μεταβλητές
- Να ορίζετε και να χρησιμοποιείτε λίστες

Προσπαθήστε να κάνετε την άσκηση μόνοι σας, πριν δείτε τις οδηγίες.

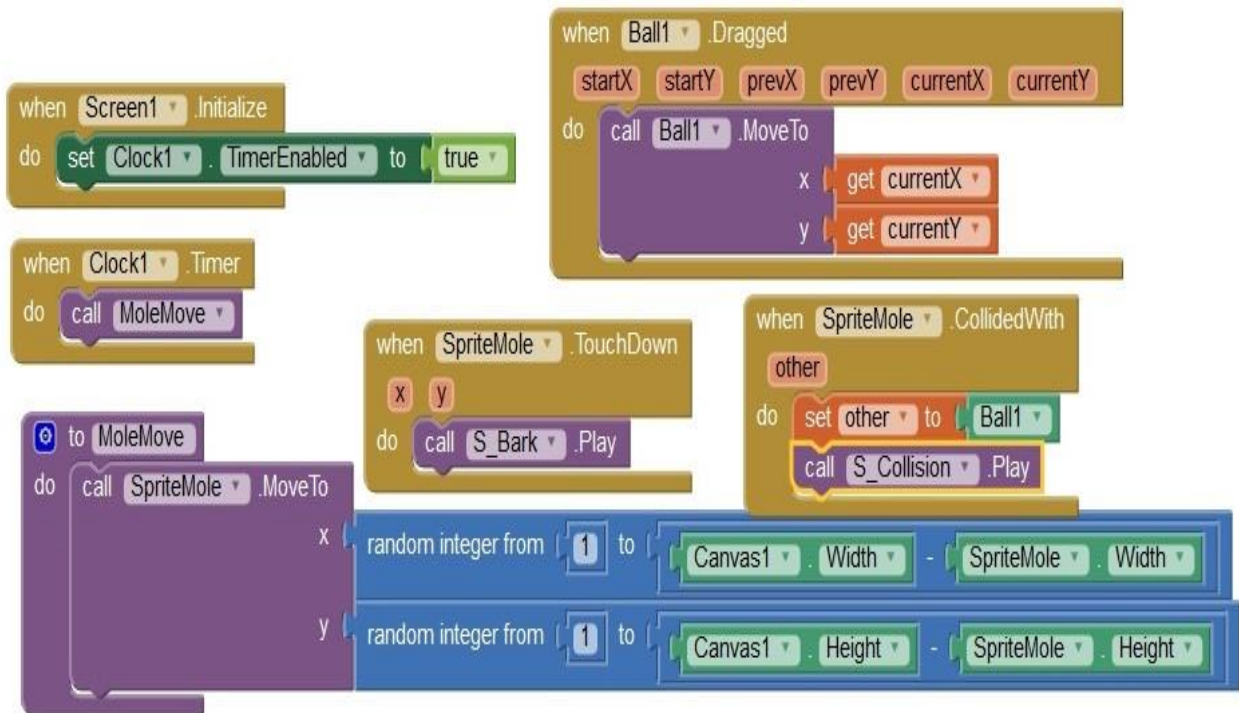
Εφαρμογή

1. Επικολλήστε στο φυλλομετρητή σας την διεύθυνση <http://appinventor.mit.edu/explore/sites/all/files/ai2tutorials/moleMash/mole.png> και αποθηκεύστε την εικόνα `mole.png`.
2. Δημιουργήστε μια νέα εφαρμογή με όνομα `HitMole1` στην οποία θα ανεβάσετε τον ήχο του γαυγίσματος (`bark.mp3`) και την εικόνα `mole.png` που θα περιέχει 1 καμβά, που θα καλύπτει όλη την οθόνη και 1 φιγούρα (`ImageSprite`), με εικόνα το `mole.png` και ένα συστατικό `Sound` με πηγή το `bark.mp3`.
3. Επίσης, προσθέστε ένα ρολόι (`clock`) από την ομάδα `Sensor` με ιδιότητα: `Time Interval = 500`.
4. Προγραμματίστε κατάλληλα(καρτέλα `Blocks`), έτσι ώστε με την εκκίνηση της εφαρμογής η φιγούρα να τοποθετείται σε τυχαία θέση στην οθόνη, η φιγούρα να μετακινείται διαρκώς τυχαία κάθε 0,5", ενώ όταν πετυχαίνουμε να κάνουμε πάνω της κλικ, να ακούγεται ένα γαύγισμα.
5. Προσθέστε μια μπάλα στην εφαρμογή σας με κόκκινο χρώμα και ακτίνα 10, την οποία θα σέρνετε με το δάχτυλό σας πάνω στην οθόνη, με σκοπό να αποφύγετε τη

σύγκρουση με το Mole. (Για να σύρουμε μια μορφή στην οθόνη υπάρχει το χειριστήριο συμβάντων when Μορφή.Dragged με παραμέτρους τις αρχικές (start)συντεταγμένες, τις τωρινές (current) και τις προηγούμενες(prev))

6. Προγραμματίστε κατάλληλα, ώστε όταν συγκρούεται με τη φιγούρα να ακούγεται ένας άλλος ήχος.

Αποτέλεσμα



Εικόνα 88: Αποτελέσματα καρτέλας Blocks

Επειδή δεν υπάρχει κάποια ιδιότητα στον καμβά ή στη μορφή, η οποία να επηρεάζεται από το πόσες φορές ακουμπάμε τον καμβά, θα πρέπει να ορίσετε μια «μεταβλητή» για να «θυμάται», να αποθηκεύεται δηλαδή εκεί το πλήθος των επιτυχιών, το σκορ.

Ορισμός

Μια μεταβλητή έχει όνομα και αρχική τιμή. Η αρχική τιμή μπορεί να είναι αριθμός, γράμματα, ή ένα σύνολο από μεμονωμένα στοιχεία τακτοποιημένα με σειρά 1ο, 2ο, κλπ σε μορφή λίστας.

Η τιμή μιας μεταβλητής μπορεί να αλλάξει με το μπλοκ set και μπορεί να αναγνωστεί με το μπλοκ get.

Στο κομμάτι blocks θα χρησιμοποιήσετε τα παρακάτω:



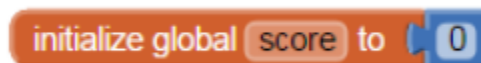
Εικόνα 89: Βοηθητική εικόνα 3ου φύλλου εργασίας

7. Εισάγετε τον κώδικα της εφαρμογής HitMole1.aia που υλοποιήσατε παραπάνω και αποθηκεύστε τον με όνομα : HitMole2.
8. Εισάγετε ένα Label με όνομα ScoreLabel στο οποίο θα φαίνεται το σκορ, δηλ το πλήθος των φορών που ο παίχτης πετυχαίνει να χτυπήσει το Mole.
9. Εισάγετε ένα Button με όνομα ResetButton που θα μηδενίζει το σκορ για να ξεκινήσει και πάλι το παιχνίδι.
10. Μεταβείτε στην καρτέλα Blocks και από την ομάδα πλακιδίων variables σύρετε την :



Εικόνα 90: Δήλωση global μεταβλητής

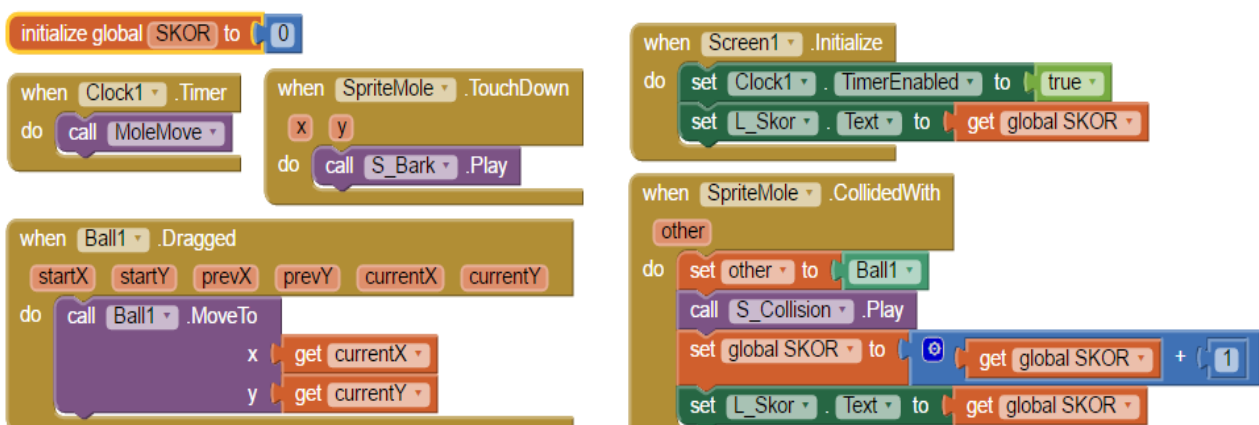
11. Δώστε στην μεταβλητή την αρχική τιμή 0.



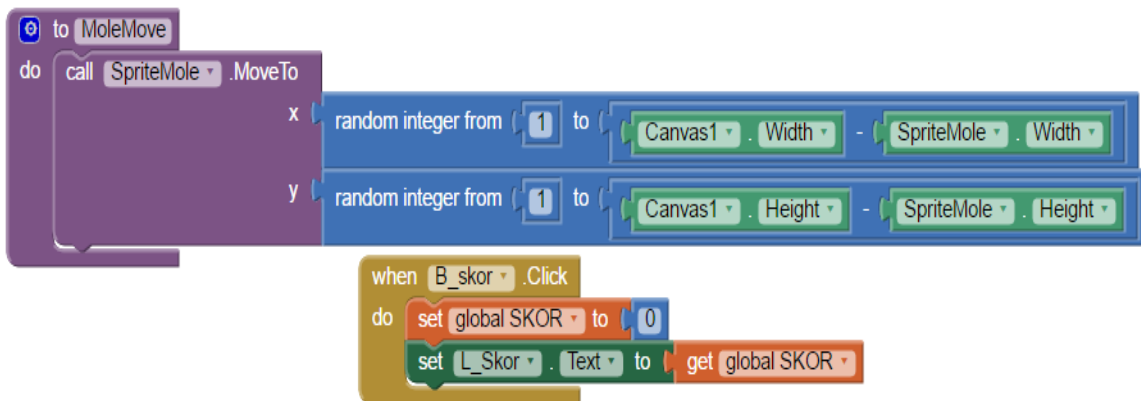
Εικόνα 91: Αρχικοποίηση global μεταβλητής

12. Προγραμματίστε κατάλληλα, ώστε κάθε φορά που θα αγγίζουμε το Mole, το σκορ να αυξάνει κατά 1 (για το άθροισμα θα χρειαστούμε το αντίστοιχο πλακίδιο από την ομάδα math) και να εμφανίζεται στο ScoreLabel.
13. Όταν πατιέται το κουμπί μηδενισμού, το Text του ScoreLabel να γίνεται 0.

Αποτέλεσμα



Εικόνα 92: Αποτελέσματα καρτέλας Blocks(2)



Εικόνα 93: Αποτελέσματα καρτέλας Blocks(2a)

Τελική εικόνα εφαρμογής



Εικόνα 94: Τελική εικόνα εφαρμογής HitTheMole

4.1.4 4^ο Φύλλο εργασίας ^[15]

Ενδεικτική Διάρκεια: 45 λεπτά (1 διδακτική ώρα)

Η συγκεκριμένη εφαρμογή, είναι ένα παιχνίδι σκοποβολής όπου ένας πύραυλος ρίχνει σφαίρες προσπαθώντας να πετύχει έναν ιπτάμενο δίσκο. Αρχικά, έγινε μία εισαγωγή στα μπλοκ εντολών if-else μέσω μικρών παραδειγμάτων στον πίνακα, ώστε οι μαθητές να κατανοήσουν τη λειτουργία όπως και τη χρήση τους, για να τις αξιοποιήσουν στην εφαρμογή που ακολουθεί.






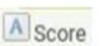

Διδακτικός στόχος

Μετά την εφαρμογή του φύλλου εργασίας οι μαθητές θα είναι σε θέση να δημιουργούν κίνηση αντικειμένων χρησιμοποιώντας τις δυνατότητες animation του AppInventor, να εισάγουν μουσική υπόκρουση σε μία εφαρμογή appinventor και να ενεργοποιούν γεγονότα ανά τακτά χρονικά διαστήματα (time intervals) χρησιμοποιώντας το συστατικό clock.

- Για την εφαρμογή θα χρειαστείτε εικόνες και ήχοι για τα διαστημόπλοια. Θα τα βρείτε στον σύνδεσμο:


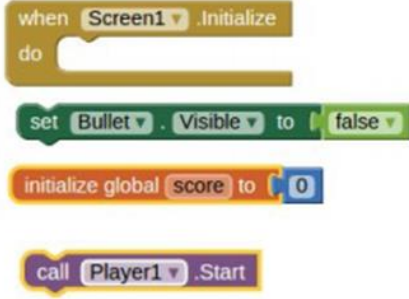
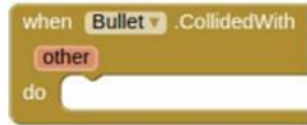


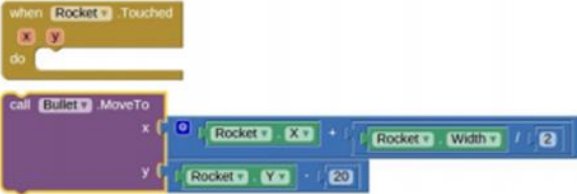
https://drive.google.com/folderview?id=0B19WMZZCCKT0VTdlS3VqRF9DN0E&usp=sharing_, ωστόσο μπορείτε να χρησιμοποιήσετε δικό σας υλικό.

1. Σχεδίαση (Designer)

	 Screen1	Title : SpacInvaders BackgroundColor : None
	 Canvas1 Δίνει τη δυνατότητα της ζωγραφικής.	BackgroundColor : Black Width : Fill Parent Height : Fill Parent
	 Rocket Alien	Picture : rocket.png και saucer.png
	 Bullet	PaintColor: Yellow
	 Score	Text : "Σκορ:"
	 Reset	Text : "Reset:"

Εικόνα 95: Σχεδιασμός application

2. Προγραμματισμός Blocks

<p>Όταν ο χρήστης σύρει τον πύραυλο τότε αυτός μετακινείται στον άξονα των X.</p>	
<p>Όταν ξεκινά η εφαρμογή η σφαίρα δεν εμφανίζεται ενώ το σκορ είναι 0. Επίσης θα ξεκινά να παίζει η μουσική που ανεβάσατε. (συμπληρώστε κατάλληλα τα διπλανά κομμάτια κώδικα)</p>	
<p>Όταν η σφαίρα χτυπά σε άλλο αντικείμενο (στην περίπτωση μας στον ιπτάμενο δίσκο πρέπει:</p> <ul style="list-style-type: none"> • Να εξαφανίζεται η σφαίρα • Να αυξάνει κατά 1 η μεταβλητή score • Να αλλάζει ανάλογα και το κείμενο της ετικέτας του σκορ. 	 <p>(Λείπουν μερικά κομμάτια κώδικα)</p>
<p>Όταν η σφαίρα φτάσει στην άκρη χωρίς να χτυπήσει τον ιπτάμενο δίσκο πρέπει να εξαφανίζεται.</p>	 <p>(Λείπουν μερικά κομμάτια κώδικα)</p>
<p>Όταν ο χρήστης κάνει κλικ στο Reset τότε πρέπει η μεταβλητή score να μηδενίζει και να αλλάζει ανάλογα και το κείμενο της ετικέτας του σκορ.</p>	 <p>(Λείπουν μερικά κομμάτια κώδικα)</p>
<p>Όταν ο χρήστης κάνει κλικ στον πύραυλο πρέπει να:</p> <ul style="list-style-type: none"> • μετακινείται η σφαίρα μέσα στον πύραυλο • να γίνεται ορατή η σφαίρα • να αλλάζει η ταχύτητα της σφαίρας σε 5 • να αλλάζει η κατεύθυνση της σφαίρας σε 90 μοίρες. 	 <p>(Λείπουν μερικά κομμάτια κώδικα)</p>

Εικόνα 96: Προγραμματισμός Blocks 1

<p>Επιπλέον δυσκολία. Κάθε φορά που ενεργοποιείται το ρολόι clock1 ο ιπτάμενος δίσκος μετακινείται τυχαία αριστερά ή δεξιά</p>	 <p>(Λείπουν μερικά κομμάτια κώδικα)</p>
---	--

Εικόνα 97: Προγραμματισμός Blocks 2

3. Αποτέλεσμα

```

initialize global score to 0

when Button1.Click
do
  set global score to 0
  set Label2.Text to get global score

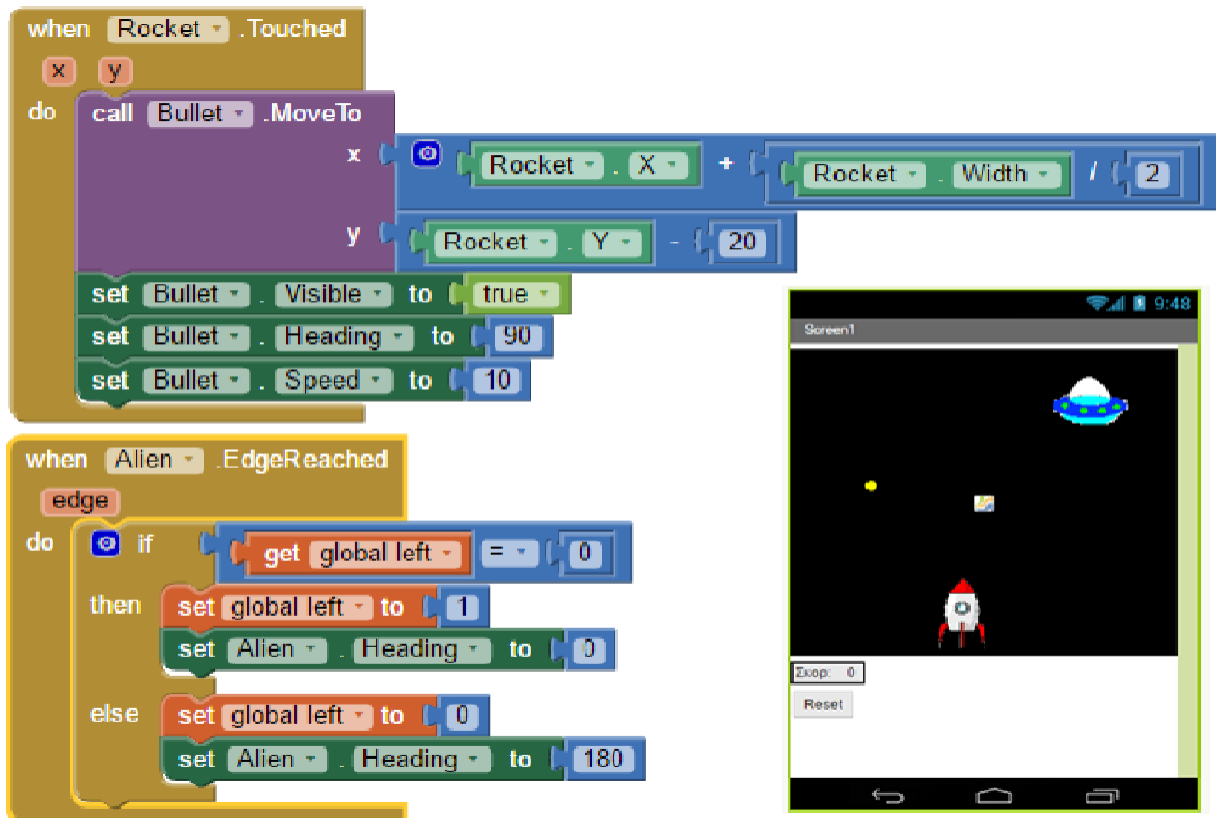
when Rocket.Dragged
  startX startY prevX prevY currentX currentY
do
  set Rocket.X to get currentX

when Bullet.CollidedWith other
do
  set other to Alien
  set Bullet.Visible to false
  set global score to get global score + 1
  set Label2.Text to get global score

when Bullet.EdgeReached edge
do
  set Bullet.Visible to false

when Screen1.Initialize
do
  set global score to 0
  set Bullet.Visible to false
  call Player1.Start
  
```

Εικόνα 98: Αποτέλεσμα (α)



Εικόνα 99: Αποτέλεσμα (β)

4.1.5 5^ο Φύλλο Εργασίας ^[15]

Ενδεικτική Διάρκεια: 45 λεπτά (1 διδακτική ώρα)

Η δραστηριότητα έχει στόχο την ανάπτυξη ενός παιχνιδιού τύπου ράλλυ στο οποίο ο παίκτης θα προσπαθεί να πετύχει με σφαίρες το αυτοκίνητο του αντιπάλου χωρίς να συγκρουστεί μαζί του.

Διδακτικός Στόχος

Μετά την εκπόνηση της δραστηριότητας θα είστε σε θέση να

- να σχεδιάζετε μια εφαρμογή με Labels, Buttons, Notifier και Canvas
- αξιοποιείτε τις δυνατότητες κίνησης και animation
- να ελέγχετε την κίνηση αντικειμένων αξιοποιώντας τις ιδιότητες του OrientationSensor

Εφαρμογή

Εισάγετε στο περιβάλλον ως media τα ακόλουθα gif αρχεία (μπορείτε να χρησιμοποιήσετε δικά σας με την βοήθεια του Google)

			
car1.gif	car1_crash.gif	sportcar.gif	sportcar_crash.gif

Εικόνα 100: Βοηθητικό υλικό

Σχεδίαση (Designer)

Εισάγετε τα παρακάτω συστατικά με τις αντίστοιχες ιδιότητες



Συστατικό	Ιδιότητες
Screen1 (Screen)	Title: Racing Game
Canvas1 (Canvas)	BackgroundColor: Green Height: FillParent Width: FillParent
Bullet (Ball)	Radius: +5 PaintColor: Black
mycar (ImageSprite)	Picture: sportcar.gif
comcar (ImageSprite)	Picture: car1.gif
orientationSensor	
racingtimer (Timer)	TimerInterval: 33
Notifier1 (Notifier)	
Συστατικά που βρίσκονται μέσα στο <u>HorizontalArrangement1</u>	
Button1 (Button)	Text: Continue
Restart (Button)	Text: Restart
Label1 (Label)	Text: Score
Score (Label)	Text: 0
Fire (Button)	Text: Fire

Εικόνα 101 : Σχεδίαση

Προγραμματισμός (Blocks)

A. Κώδικας που αφορά την κίνηση του comcar

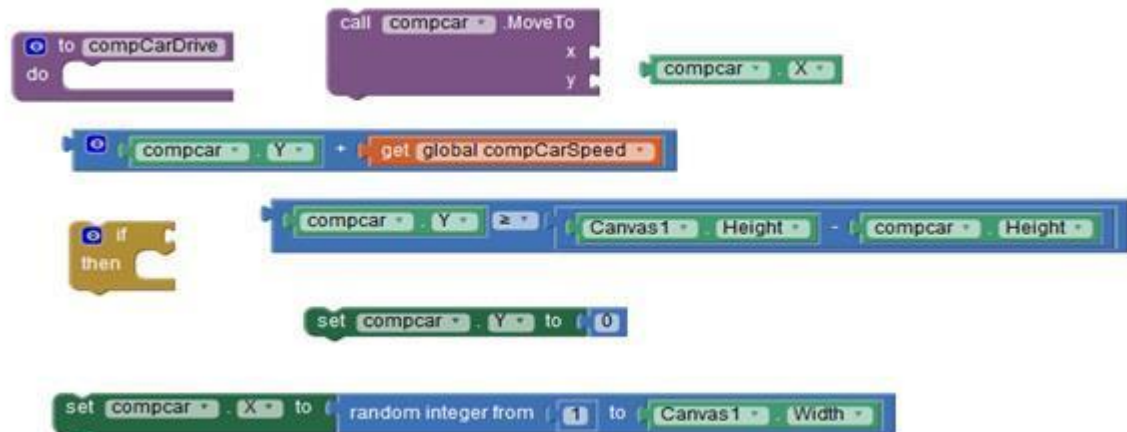
Το αυτοκίνητο του αντιπάλου (comcar) θα πρέπει να διασχίζει τον Canvas από πάνω προς τα κάτω (άξονας y) σε τυχαίες θέσεις (στον άξονα των x). Δηλώστε μια μεταβλητή με όνομα compCarSpeed και αρχικοποιήστε την με την τιμή 20

- Η κίνηση του comcar θα γίνεται με τη βοήθεια της διαδικασίας comCarDrive. Η διαδικασία αυτή θα καλείται κάθε φορά που ενεργοποιείται το Timer racingtimer



Εικόνα 102 : Προγραμματισμός (α)

- Συνδυάστε τα παρακάτω συστατικά έτσι ώστε η κάθε φορά που γίνεται κλήση της διαδικασίας `compCarDrive` κινείται να μετακινείται το `compCar` κατά 20 βήματα προς τα κάτω στον άξονα των `y` (τιμή της μεταβλητής `compCarSpeed`). Όταν το `compCar` φτάσει στο κάτω μέρος του καμβά τότε θα πρέπει να ξεκινάει πάλι από το πάνω μέρος του καμβά σε διαφορετικό (τυχαίο) σημείο του άξονα των `x`.

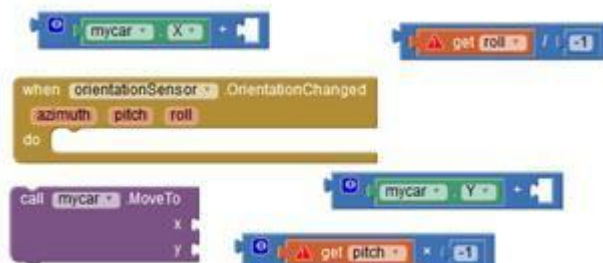


Εικόνα 103 : Προγραμματισμός (β)

Το αυτοκίνητο του παίκτη (`mycar`) θα πρέπει να διασχίζει τον `Canvas` ανάλογα με την κλήση της συσκευής θα μετακινεί το `mycar`

- Δεξιά – Αριστερά (ιδιότητα `roll`)
- Μπροστά – Πίσω (ιδιότητα `pitch`),

Συνδυάστε τα συστατικά που φαίνονται πιο κάτω έτσι ώστε να πετύχετε την κατάλληλη συμπεριφορά στο `mycar`



Εικόνα 104 : Προγραμματισμός (γ)

Γ. Κώδικας που αφορά την κίνηση της σφαίρας Bullet

Συνδυάστε τις παρακάτω εντολές έτσι ώστε η σφαίρα να ενεργοποιείται με το πάτημα του κουμπιού Fire και να εκτελούνται τα παρακάτω:

- Μετακίνησε τη σφαίρα στο μπροστά κέντρο μέρος του mycar
- Κάνε τη σφαίρα ορατή (ιδιότητα visible ← true)
- Θέσε την κατεύθυνση της σφαίρας προς το πάνω μέρος της οθόνης (ιδιότητα Heading ← 90)
- Θέσε την ταχύτητα κίνησης της σφαίρας στο 20 (ιδιότητα Speed ← 20)



Εικόνα 6 - Προγραμματισμός (δ) Δραστηριότητας 10

Επιπλέον θα πρέπει όταν η σφαίρα φτάσει το πάνω μέρος του καμβά να εξαφανίζεται

- Κάνε τη σφαίρα αόρατη (ιδιότητα visible ← false)



Εικόνα 105 : - Προγραμματισμός (ε)

Κώδικας που αφορά τη σύγκρουση της σφαίρας Bullet με το compcar

Στην περίπτωση που η σφαίρα Bullet πετύχει το compcar, τότε θα πρέπει να αλλάζει η εικόνα του σε car1_crash.gif και το σκορ να αυξάνεται κατά 1.



Εικόνα 106 : Προγραμματισμός (στ)

- Κάνε τη σφαίρα αόρατη (ιδιότητα visible ← false)
- θέσε ως εικόνα στο comcar την εικόνα car1_crash.gif
- αύξησε το σκορ (Label score) κατά 1

Ε. Κώδικας που αφορά τη σύγκρουση του comcar με το mycar

Στην περίπτωση που συγκρουστεί το mycar με το comcar θα πρέπει να εκτελούνται τα παρακάτω:

- απενεργοποίησε το racingtimer θέτοντας την ιδιότητα TimeerEnabled σε false
- θέσε ως εικόνα στο comcar την εικόνα car1_crash.gif
- θέσε ως εικόνα στο mucar την εικόνα sportcar_crash.gif
- κάνε το comcar αόρατο
- κάνε το mycar αόρατο
- με τη βοήθεια του Notifier1 εμφάνισε το μήνυμα Game Over



Εικόνα 107 : Προγραμματισμός (ζ)

ΣΤ. Κώδικας που αφορά την αρχικοποίηση του παιχνιδιού και το κουμπί Restart



Εικόνα 108 : Προγραμματισμός (η)

Κατά την αρχική εκκίνηση του παιχνιδιού (καθώς και με το πάτημα του κουμπιού Restart) θα πρέπει να εκτελούνται τα παρακάτω

- κάνε το comcar ορατό
- κάνε το mycar ορατό
- κάνε τη σφαίρα αόρατη
- ενεργοποίησε το racingtimer
- θέσε ως εικόνα στο comcar την εικόνα car1.gif
- θέσε ως εικόνα στο mucar την εικόνα sportcar.gif
- θέσε το Text του Label score σε 0

ΣΤ. Κώδικας που αφορά την το κουμπί Button1

Το κουμπί Button1 αφορά τις ενέργειες που πρέπει να γίνουν όταν η σφαίρα Bullet πετύχει το comcar. Θα πρέπει με το πάτημα του κουμπιού να εκτελούνται τα παρακάτω:

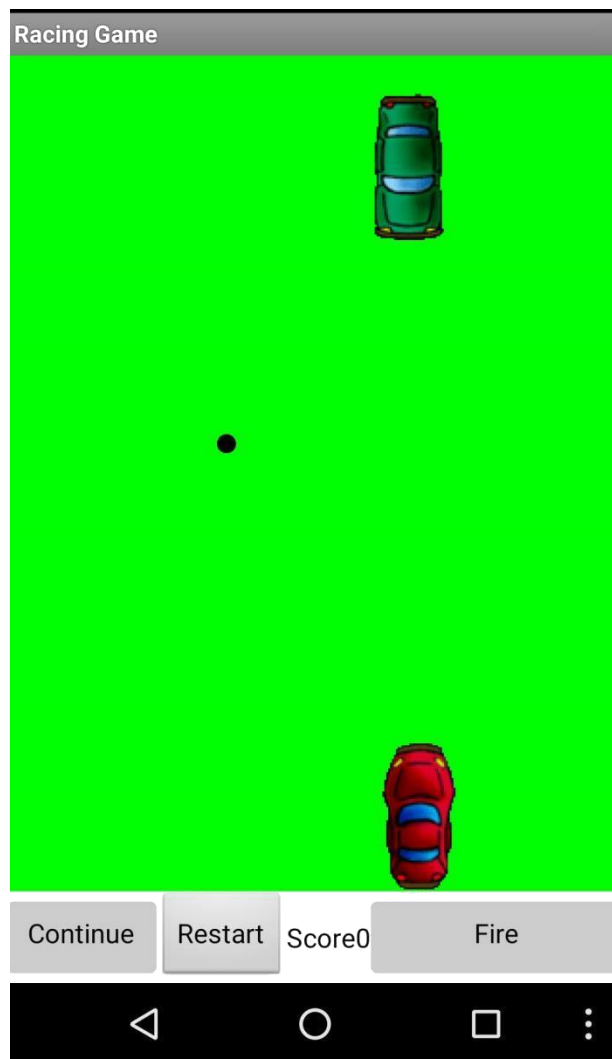
- ενεργοποίησε το racingtimer
- κάνε τη σφαίρα ορατή
- θέσε ως εικόνα στο comcar την εικόνα car1.gif

Ζ. Προαιρετικός κώδικας

Αναζητήστε στο διαδίκτυο ήχους που να μπορούν να ενσωματωθούν

- ως μουσική στο υπόβαθρο (που να παίζει κατά τη διάρκεια του παιχνιδιού)
- ως ήχοι σύγκρουσης
- ως ήχοι εκτόξευσης της σφαίρας

Αποτέλεσμα



Εικόνα 109 : Αποτέλεσμα (α)

```

when orientationSensor .OrientationChanged
  azimuth pitch roll
do
  call mycar .MoveTo
  x mycar . X + (get roll / -1)
  y mycar . Y + (get pitch * -1)

initialize global compCarSpeed to 20

when Button1 .Click
do
  set racingtimer .TimerEnabled to true
  set Bullet .Visible to false
  set compcar .Picture to car1.gif
  set mycar .Picture to sportcar.gif

when racingtimer .Timer
do
  call compCarDrive

when compcar .CollidedWith
  other
do
  set other to mycar
  set racingtimer .TimerEnabled to false
  set compcar .Picture to car1_crash.gif
  set mycar .Picture to sportcar_crash.gif
  set racingtimer .TimerEnabled to false
  set compcar .Visible to false
  set mycar .Visible to false
  call Notifier1 .ShowAlert
  notice Game Over

when Screen1 .Initialize
do
  set compcar .Visible to true
  set mycar .Visible to true
  set Bullet .Visible to false
  set racingtimer .TimerEnabled to true
  set compcar .Picture to car1.gif
  set mycar .Picture to sportcar.gif
  set Score .Text to 0

when Restart .Click
do
  set compcar .Visible to true
  set mycar .Visible to true
  set Bullet .Visible to false
  set racingtimer .TimerEnabled to true
  set compcar .Picture to car1.gif
  set mycar .Picture to sportcar.gif
  set Score .Text to 0

when Fire .Click
do
  call Bullet .MoveTo
  x mycar . X + (mycar . Width / 2)
  y mycar . Y - 20
  set Bullet .Visible to true
  set Bullet .Heading to 90
  set Bullet .Speed to 20
  
```

Εικόνα 110 : Αποτέλεσμα (β)

```
when Bullet . CollidedWith other
do
  set other to compcar
  set Bullet . Visible to false
  set compcar . Picture to car1_crash.gif
  set Score . Text to Score . Text + 1

when Bullet . EdgeReached edge
do
  set Bullet . Visible to false

to compCarDrive
do
  call compcar . MoveTo
  x: compcar . X
  y: compcar . Y + get global compCarSpeed
  if compcar . Y >= Canvas1 . Height - compcar . Height
  then
    set compcar . Y to 0
    set compcar . X to random integer from 1 to Canvas1 . Width
```

Εικόνα 111 : Αποτέλεσμα (γ)

4.1.6 6^ο Φύλλο Εργασίας ^[15]

Η εφαρμογή περιέχει μια φιγούρα και μία μπάλα. Η φιγούρα θα κινείται διαρκώς σε τυχαία θέση. Όταν την αγγίζουμε να ακούγεται ένας ήχος. Η μπάλα θα είναι κόκκινη και θα την σέρνουμε με το δάχτυλό μας ώστε να αποφύγουμε την σύγκρουση με τη φιγούρα. Όταν συγκρούεται με τη φιγούρα να ακούγεται ένας άλλος ήχος.

Διδακτικός Στόχος

Μετά την εφαρμογή του φύλλου εργασίας θα είστε σε θέση να:

- Να δημιουργήσετε κίνηση μιας μορφής στην οθόνη
- Να αλλάζετε δυναμικά (καθώς εκτελείται το πρόγραμμα) τις ιδιότητες συστατικών με χρήση set και get μπλοκ
- Να επιλέγετε διαδικασίες από τις υπάρχουσες για κάθε συστατικό
- Να αναλύεται συμβάντα και να επιλέγετε κατάλληλους χειριστές αυτών
- Να ορίζεται τυχαία θέση στην οθόνη

Προσπαθήστε να κάνετε την άσκηση μόνοι σας, πριν δείτε τις οδηγίες.

Εφαρμογή

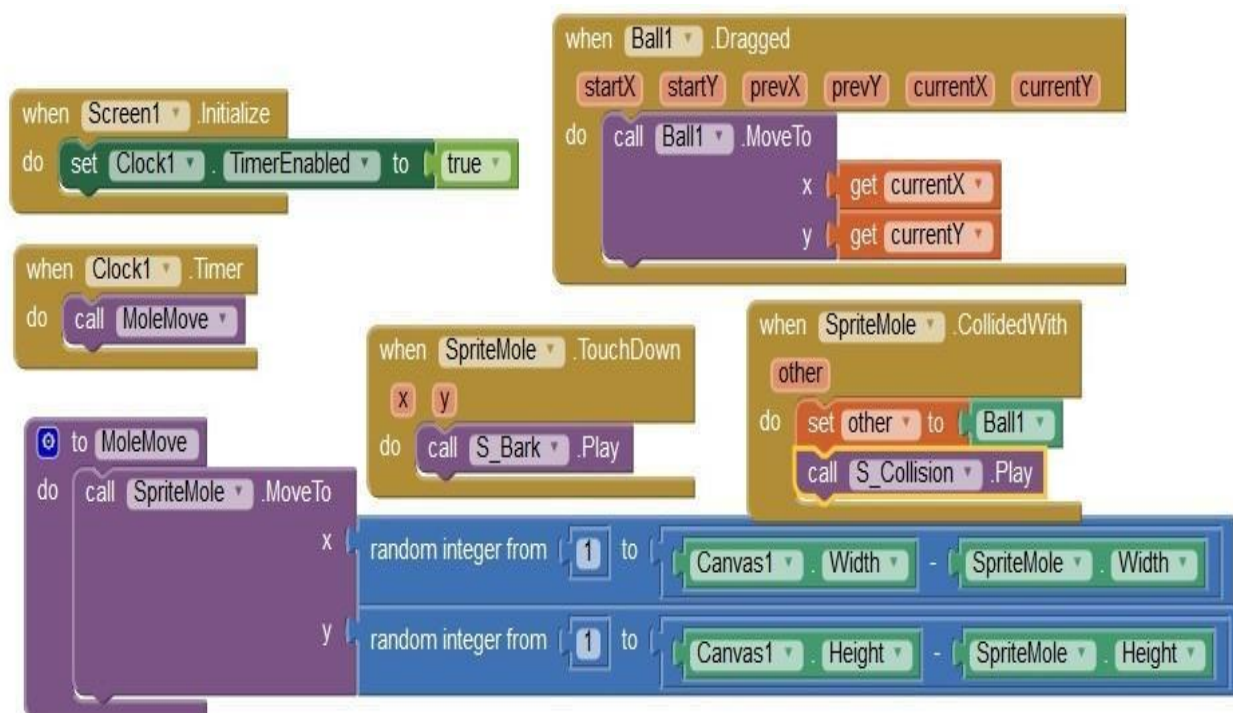
Σχεδίαση (Designer) – Προγραμματισμός (Blocks)

- Επικολλήστε στο φυλλομετρητή σας την διεύθυνση <http://appinventor.mit.edu/explore/sites/all/files/ai2tutorials/moleMash/mole.png> και αποθηκεύστε την εικόνα mole.png.
- Δημιουργήστε μια νέα εφαρμογή με όνομα HitMole1 στην οποία θα ανεβάσετε τον ήχο του γαυγίσματος (bark.mp3) και την εικόνα mole.png
- Η εφαρμογή θα περιέχει 1 καμβά, που θα καλύπτει όλη την οθόνη και 1 φιγούρα (ImageSprite), με εικόνα το mole.png, ένα συστατικό Sound με

πηγή το bark.mp3 και ένα ρολόι (clock) από την ομάδα Sensor με ιδιότητα: Time Interval =500

- Θα προγραμματίσετε έτσι ώστε με την εκκίνηση της εφαρμογής η φιγούρα να τοποθετείται σε τυχαία θέση στην οθόνη,
- Η φιγούρα να μετακινείται διαρκώς τυχαία κάθε 0,5''
- Όταν πετυχαίνουμε να κάνουμε πάνω της κλικ, να ακούγεται ένα γαύισμα
- Προσθέστε μια μπάλα στην εφαρμογή σας με κόκκινο χρώμα και ακτίνα 10 , την οποία θα σέρνετε με το δάχτυλό σας πάνω στην οθόνη, με σκοπό να αποφύγετε τη σύγκρουση με το Mole. (Για να σύρουμε μια μορφή στην οθόνη υπάρχει το χειριστήριο συμβάντων when Morφή.Dragged με παραμέτρους τις αρχικές (start) συντεταγμένες, τις τωρινές (current) και τις προηγούμενες (prev)).
- Όταν συγκρούεται με τη φιγούρα να ακούγεται ένας άλλος ήχος.

Αποτέλεσμα



Εικόνα 112 : Αποτελέσματα

Επειδή δεν υπάρχει κάποια ιδιότητα στον καμβά ή στη μορφή, η οποία να επηρεάζεται από το πόσες φορές ακουμπάμε τον καμβά, θα ορίσουμε μια «μεταβλητή» για να «θυμάται», να αποθηκεύουμε δηλαδή εκεί το πλήθος των επιτυχιών, το σκορ.

Μια μεταβλητή έχει όνομα και αρχική τιμή. Η αρχική τιμή μπορεί να είναι αριθμός, γράμματα, ή ένα σύνολο από μεμονωμένα στοιχεία τακτοποιημένα με σειρά 1ο, 2ο, κλπ σε μορφή λίστας.

Η τιμή μιας μεταβλητής μπορεί να αλλάξει με το μπλοκ set και μπορεί να αναγνωστεί με το μπλοκ get.

Στο κομμάτι blocks θα χρησιμοποιήσετε τα παρακάτω



Εικόνα 113: Βοηθητική εικόνα

Σχεδίαση (Designer)

- Εισάγετε την κώδικα της εφαρμογής HitMole1.aia από το Φύλλο Εργασίας 4 και αποθηκεύστε τον με όνομα : HitMole2 .
- Εισάγετε ένα Label με όνομα ScoreLabel στο οποίο θα φαίνεται το σκορ, δηλ το πλήθος των φορών που ο παίχτης πετυχαίνει να χτυπήσει το Mole .
- Εισάγετε ένα Button με όνομα ResetButton που θα μηδενίζει το σκορ για να ξαναξεκινήσει το παιχνίδι.

Προγραμματισμός (Blocks)

Από την ομάδα πλακιδίων variables σέρνουμε την :



Εικόνα 114 : Προγραμματισμος (α)

και της δίνουμε αρχική τιμή 0 και όνομα “score” (αρχικοποίηση)



Εικόνα 115 : Προγραμματισμος (β)

Κάθε φορά που θα αγγίζουμε το Mole, το σκορ να αυξάνει κατά 1. (για το άθροισμα θα χρειαστούμε το αντίστοιχο πλακίδιο από την ομάδα math)

Συμβάν-γεγονός	Συμπεριφορά-ενέργεια	Δηλ
Αγγίζουμε το Mole	Το σκορ να αυξάνεται κατά 1	Θέσε το Score <- σε όσο ήταν +1
	Το σκορ εμφανίζεται στο ScoreLabel	Θέσε στο Text του ScoreLabel την τιμή της μεταβλητής score

Εικόνα 116 : Γεγονός ενέργεια (α)

Συμβάν- γεγονός	Συμπεριφορά- ενέργεια	Δηλ
όταν πατιέται το κουμπί μηδενισμός	Text του ScoreLabel να γίνεται 0	Θέσε στο Text του ScoreLabel την τιμή 0

Εικόνα 117: Γεγονός ενέργεια (β)

Αποτέλεσμα

```

initialize global SKOR to 0

when Clock1 .Timer
do call MoleMove

when SpriteMole .TouchDown
do call S_Bark .Play

when Ball1 .Dragged
startX startY prevX prevY currentX currentY
do call Ball1 .MoveTo
x get currentX
y get currentY

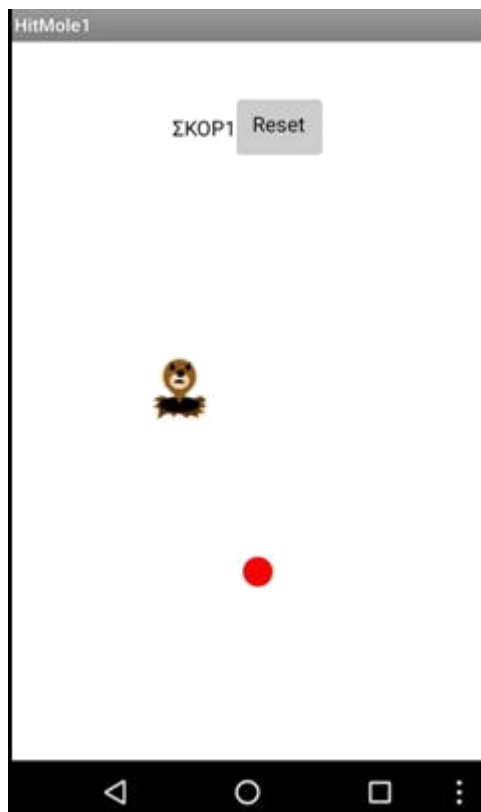
when Screen1 .Initialize
do set Clock1 .TimerEnabled to true
set L_Skor .Text to get global SKOR

when SpriteMole .CollidedWith
other
do set other to Ball1
call S_Collision .Play
set global SKOR to get global SKOR + 1
set L_Skor .Text to get global SKOR

to MoleMove
do call SpriteMole .MoveTo
x random integer from 1 to Canvas1 .Width - SpriteMole .Width
y random integer from 1 to Canvas1 .Height - SpriteMole .Height

when B_skor .Click
do set global SKOR to 0
set L_Skor .Text to get global SKOR
    
```

Εικόνα 118 : Αποτέλεσμα (α)



Εικόνα 119 : Αποτέλεσμα (β)

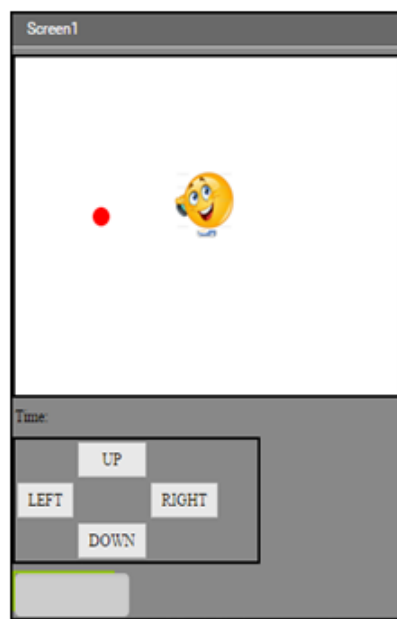
4.1.7 1^η Δραστηριότητα [15]

Πληροφορίες

Ένα κόκκινο μπαλάκι θα κινείται τυχαία στην οθόνη του κινητού, ο παίχτης καθοδηγώντας με τέσσερα κουμπιά μια μικρή φατσούλα θα προσπαθεί να ακουμπήσει το κόκκινο μπαλάκι. Όταν ο παίχτης καταφέρει να ακουμπά το μπαλάκι θα ακούγεται ένας ήχος και θα αυξάνεται ένας μετρητής που κρατά το σκορ επαφών. Όταν το σκορ των επαφών φτάσει στην τιμή 20 τότε το παιχνίδι τερματίζεται και ο παίχτης βλέπει τον χρόνο του. Σκοπός του κάθε φορά είναι να ακουμπάει το μπαλάκι 20 φορές στον λιγότερο χρόνο.

Σχεδιασμός (Designer)

Αφού συνδεθείτε με το περιβάλλον του App Inventor με την βοήθεια του φυλλομετρητή σας, δημιουργήστε ένα καινούργιο project και δώστε το όνομα Game. Σχεδιάστε την οθόνη σας, παρόμοια με την εικόνα που ακολουθεί, με εικόνες που θα βρείτε χρησιμοποιώντας το Google



Εικόνα 120 – Σχεδίαση (α) Δραστηριότητας 9

Φορτώστε τα ακόλουθα μη ορατά στοιχεία όπως φαίνονται στην εικόνα που ακολουθεί



Εικόνα 121 - Σχεδίαση (β) Δραστηριότητας 9

Στην συνέχεια και πάλι με την βοήθεια του Google βρείτε και ανεβάστε τα ακόλουθα αρχεία:

- ένα ήχο με αστείο γέλιο (χρησιμοποιήστε το : “crazy laugh mp3”)
- ένα ήχο με χειροκρότημα (χρησιμοποιήστε το: “applause mp3”)
- μία εικόνα που να δίνει συγχαρητήρια (χρησιμοποιήστε το : “congartulations image”).

Υποστήριξη της διδακτικής διαδικασίας του μαθήματος «Εφαρμογές Πληροφορικής»

- μια εικόνα με μια χαμογελαστή φατσούλα (χρησιμοποιήστε το : “smiley face”).
θα χρειαστεί να μικρύνετε την φατσούλα στο 50%

και ονομάστε τα αρχεία αυτά σύμφωνα με την εικόνα που ακολουθεί:



Εικόνα 122 - Σχεδίαση (γ) Δραστηριότητας 9

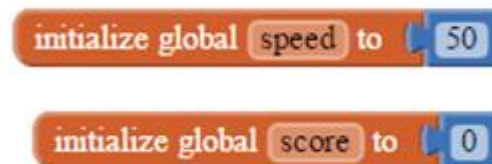
Η σύνθεση των αντικειμένων της εφαρμογής σας θα πρέπει να έχει την παρακάτω μορφή:



Εικόνα 123 - Σχεδίαση (δ) Δραστηριότητας 9

Προγραμματισμός (Blocks)

Αρχικοποίηση μεταβλητών :



Εικόνα 124 - Προγραμματισμός (α) Δραστηριότητας 9

Αρχικοποιήστε την οθόνη του παιχνιδιού έτσι ώστε:

- το imagesprite1 να είναι ορατό
- το Ball1 να είναι ορατό
- το κείμενο του Label1 να είναι 0
- η εικόνα φόντου του Canvas1 να είναι "None"

- η θέση της Ball1 να είναι ένα τυχαίο σημείο μέσα στον Canvas1. χρησιμοποιήστε τον κώδικα της δραστηριότητας Hit the mole

Προγραμματίστε την κίνηση της φατσούλας με τα τέσσερα κουμπιά στο κλικ του κουμπιού left να μειώνεται το η X συντεταγμένη κατά την τιμή του global speed

(αντίστοιχα τα άλλα τρία κουμπιά)

Προγραμματίστε την τυχαία κίνηση της μικρής κόκκινης μπάλας, και μετρήστε τον χρόνο αυξάνοντας την τιμή της ετικέτας Label1 κατά 1. (κάθε φορά που θα χτυπάει το timer του Clock1

Προγραμματίστε τη συμπεριφορά της εφαρμογής όταν υπάρχει επαφή της μικρής κόκκινης μπάλας με την φατσούλα που καθοδηγεί ο παίχτης (να ακούγεται ο ήχος crazy laugh) καθώς και την αύξηση του σκορ κάθε φορά.

Τέλος, όταν το σκορ φτάσει στην τιμή 20:

- εξαφανίστε τη φατσούλα και το μπαλάκι
- εμφανίστε την εικόνα congrats.jpg ως backgroundImage του Canvas1
- παίξτε τον ήχο των χειροκροτημάτων και
- σταματήστε τον χρόνο.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η εμπειρία από αυτό το μάθημα ήταν ιδιαίτερα θετική και θεωρούμε ότι οι εκπαιδευτικοί στόχοι καλύφθηκαν στο έπακρον. Μετά το πέρας των μαθημάτων, παρατηρήσαμε ότι μαθητές, παρόλο που επρόκειτο για ένα μάθημα επιλογής κι ενώ, παράλληλα, είχαν αυξημένο φόρτο διαβάσματος και άγχος, είχαν αρκετή διάθεση και όρεξη για την δημιουργία των δικών τους ιστοριών και εφαρμογών.

Πρώτα από όλα, από το ερωτηματολόγιο αξιολόγησης των μαθητών σχετικά με το περιβάλλον Alice που παρατίθεται στο πρώτο παράρτημα, και την ανάλυση των απαντήσεων που έδωσαν, συμπεράναμε ότι οι μαθητές κατανόησαν τις βασικές έννοιες και αρχές του αντικειμενοστραφούς προγραμματισμού με τρόπο εύκολο και ευχάριστο. Μέσα από την ενασχόλησή τους με αυτό, αλλά και με το περιβάλλον Appinventor, είναι πλέον πιο εξοικειωμένοι με έννοιες όπως αντικείμενα, κληρονομικότητα, ιδιότητες των αντικειμένων και αλληλεπίδραση μεταξύ αυτών, δημιουργία συναρτήσεων και διαδικασιών, καθώς και ό,τι άλλο χρειάζεται για τη δημιουργία ιστοριών και εφαρμογών.

Με την παρουσία μας μέσα στην τάξη και μέσα από συζητήσεις με τους μαθητές, παρατηρήσαμε ότι η ενασχόληση με τα συγκεκριμένα περιβάλλοντα τους έδωσε την ευκαιρία να χρησιμοποιήσουν την εμπειρία, την φαντασία και το χιούμορ τους για ένα αποτέλεσμα που μας εξέπληξε όλους ευχάριστα τόσο στη δημιουργία ιστοριών μέσω του περιβάλλοντος Alice, όσο και στην δημιουργία εφαρμογών με τη βοήθεια του Appinventor. Με αφορμή το ενδιαφέρον που έδειξαν οι μαθητές, επιλέξαμε σε συνεργασία με τον διδάσκοντα, να τους παρουσιάσουμε διάφορα επιπλέον περιβάλλοντα. Αρχικά, ασχοληθήκαμε με το Pivot Animator, με τη βοήθεια του οποίου μπόρεσαν να δημιουργήσουν κινούμενα σχέδια. Στη συνέχεια, δημιούργησαν και επεξεργάστηκαν εικόνες της επιλογής τους στο περιβάλλον Photoshop, και τέλος, μέσω της ενασχόλησης με το GameMaker, μυήθηκαν στην δημιουργία δικών τους παιχνιδιών.

Κατά τη διάρκεια της διδασκαλίας, επικεντρωθήκαμε κυρίως, στη σωστή διεξαγωγή του μαθήματος, στην καλύτερη δυνατή τεχνική του οργάνωση, προσπαθώντας να διατηρήσουμε αμείωτο το ενδιαφέρον των μαθητών, προάγοντας ταυτόχρονα το ομαδικό πνεύμα και την συνεργατικότητα.

Εν κατακλείδι, ύστερα από αυτή μας την εμπειρία, θεωρούμε ότι μια τέτοιου είδους διδασκαλία θα ήταν πολύ ωφέλιμο να υιοθετηθεί, ακόμα και σε μικρότερες τάξεις, διότι βοηθάει στην καλύτερη κατανόηση και αφομοίωση δύσκολων προγραμματιστικών εννοιών. Αν η επίδραση αυτού του τρόπου διδασκαλίας ήταν τόσο θετική στους μαθητές της Α΄ Λυκείου, πιστεύουμε ότι θα έχει τα ίδια κι ίσως καλύτερα αποτελέσματα σε μαθητές μικρότερων ηλικιών.

ΠΑΡΑΡΤΗΜΑ Ι

Ερωτηματολόγιο αξιολόγησης μαθήματος

Ερωτηματολόγιο Αξιολόγησης Μαθήματος
** Αναγκίτση*

1. Φύλο *
Να επισημαίνεται μόνο μία έλλειψη.

Αγόρι
 Κορίτσι

2. Τμήμα *
Να επισημαίνεται μόνο μία έλλειψη.

ΗΥ1
 ΗΥ2

3. Οι στόχοι των μαθημάτων για το περιβάλλον Αλίοε ήταν σαφείς: *
Να επισημαίνεται μόνο μία έλλειψη.

1 2 3 4 5
Καθόλου Πόρα πολύ

4. Το εκπαιδευτικό-διδακτικό υλικό που χρησιμοποιήθηκε βοήθησε στην καλύτερη κατανόηση του περιβάλλοντος Αλίοε: *
Να επισημαίνεται μόνο μία έλλειψη.

1 2 3 4 5
Καθόλου Πόρα πολύ

5. Το περιεχόμενο των μαθημάτων ήταν χρήσιμο: *
Να επισημαίνεται μόνο μία έλλειψη.

1 2 3 4 5
Καθόλου Πόρα πολύ

6. Σας δόθηκε επαρκές υλικό στο πλαίσιο των μαθημάτων: *
Να επισημαίνεται μόνο μία έλλειψη.

1 2 3 4 5
Καθόλου Πόρα πολύ

7. Υπάρχει επαρκής καθοδήγηση: *
Να επισημαίνεται μόνο μία έλλειψη.

1 2 3 4 5
Καθόλου Πόρα πολύ

Εικόνα 125: Ερωτήσεις αξιολόγησης 1-7

8. Οι εργασίες που κάνετε σας βοήθησαν να κατανοήσετε το θέμα το οποίο αφορούν; *

Να επισημαντείται μόνο μία έλλειψη.

	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

9. Πιστεύετε πως οργανώθηκε καλά η παρουσίαση της ώλης στα μαθήματα; *

Να επισημαντείται μόνο μία έλλειψη.

	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

10. Πιστεύετε πως αναλύθηκαν και παρουσιάστηκαν οι έννοιες με απλό και ενδιαφέροντα τρόπο χρησιμοποιώντας παραδείγματα; *

Να επισημαντείται μόνο μία έλλειψη.

	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

11. Το περιεχόμενο των φύλλων εργασίας ήταν κατανοητό; *

Να επισημαντείται μόνο μία έλλειψη.

	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

12. Το περιεχόμενο των φύλλων εργασίας ήταν ενδιαφέρον; *

Να επισημαντείται μόνο μία έλλειψη.

	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

13. Το επίπεδο δυσκολίας των δραστηριοτήτων ήταν αποδοτικό για την τάξη στην οποία αντιστοιχεί; *

Να επισημαντείται μόνο μία έλλειψη.

	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

14. Το μάθημα συνέβαλε στην απόκτηση νέων γνώσεων; *

Να επισημαντείται μόνο μία έλλειψη.

	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

15. Το μάθημα συνέβαλε στην καλλιέργεια συνεργατικότητας; *

Να επισημαντείται μόνο μία έλλειψη.

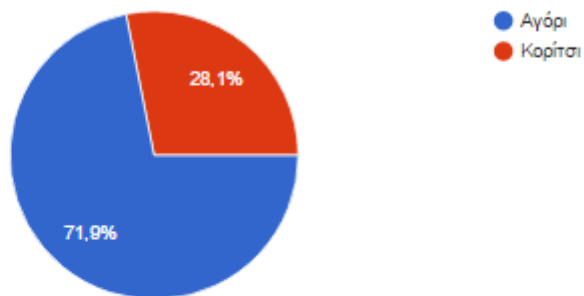
	1	2	3	4	5	
Καθόλου	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Πάρα πολύ

16. Σας παρακαλούμε σημειώστε οποιαδήποτε παρατήρηση –σχόλιο έχετε αναφορικά με την αξιολόγηση του μαθήματος που νομίζετε ότι δεν καλύπτεται από τις παραπάνω ερωτήσεις

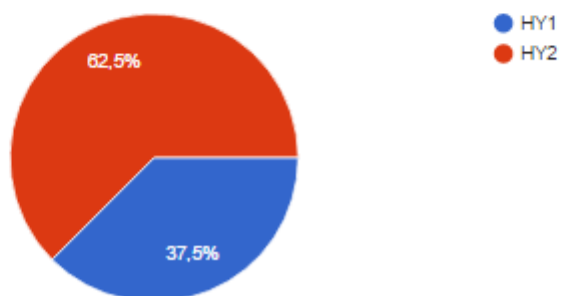
Εικόνα 126: Ερωτήσεις αξιολόγησης 8-16

Απαντήσεις των μαθητών

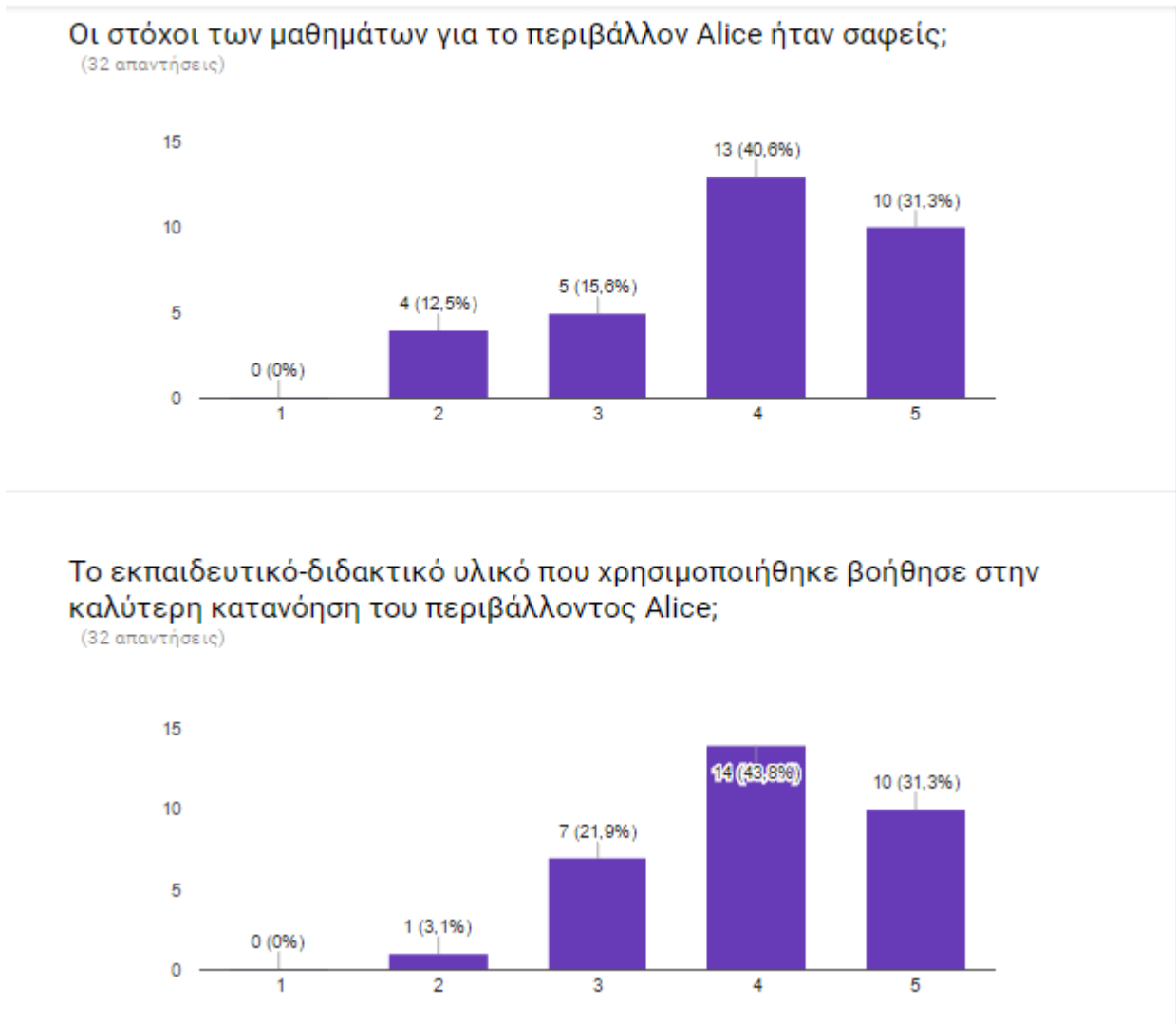
Φύλο (32 απαντήσεις)



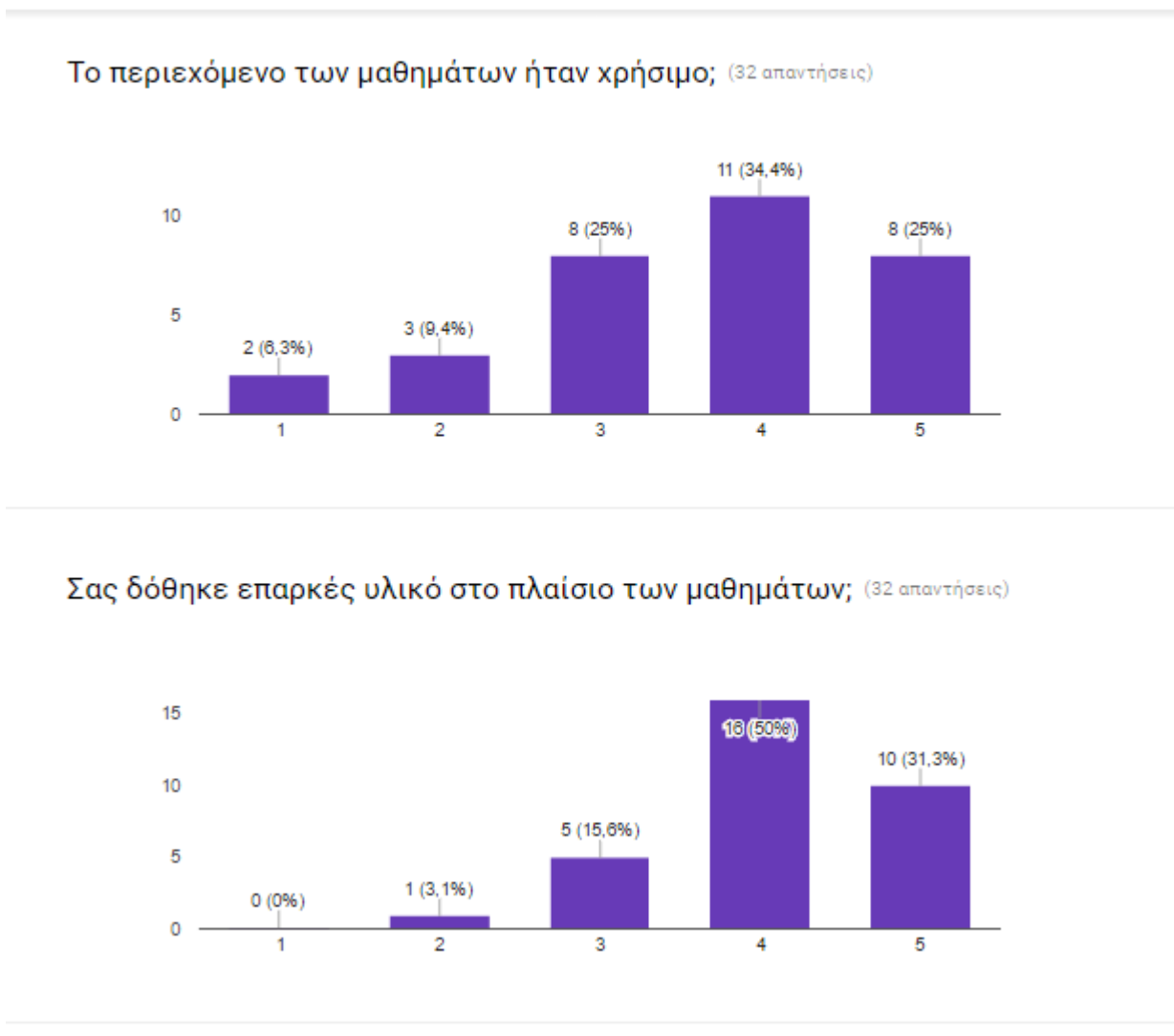
Τμήμα (32 απαντήσεις)



Εικόνα 127: Απαντήσεις στις ερωτήσεις 1-2

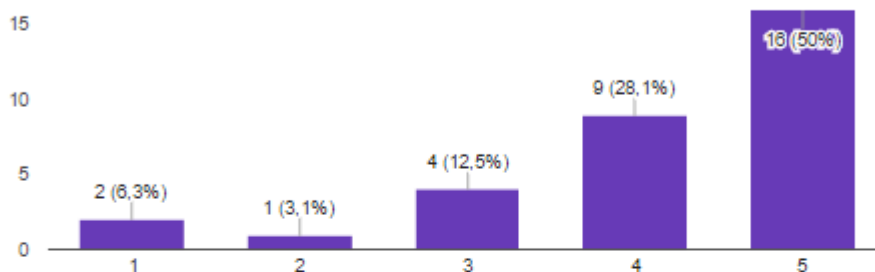


Εικόνα 128 :Απαντήσεις στις ερωτήσεις 3-4



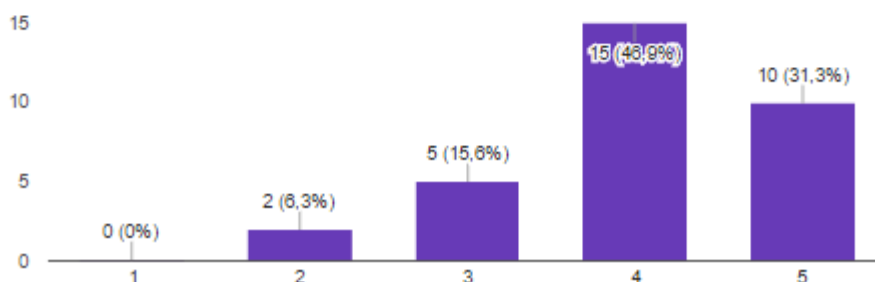
Εικόνα 129: Απαντήσεις στις ερωτήσεις 5-6

Υπήρχε επαρκής καθοδήγηση; (32 απαντήσεις)

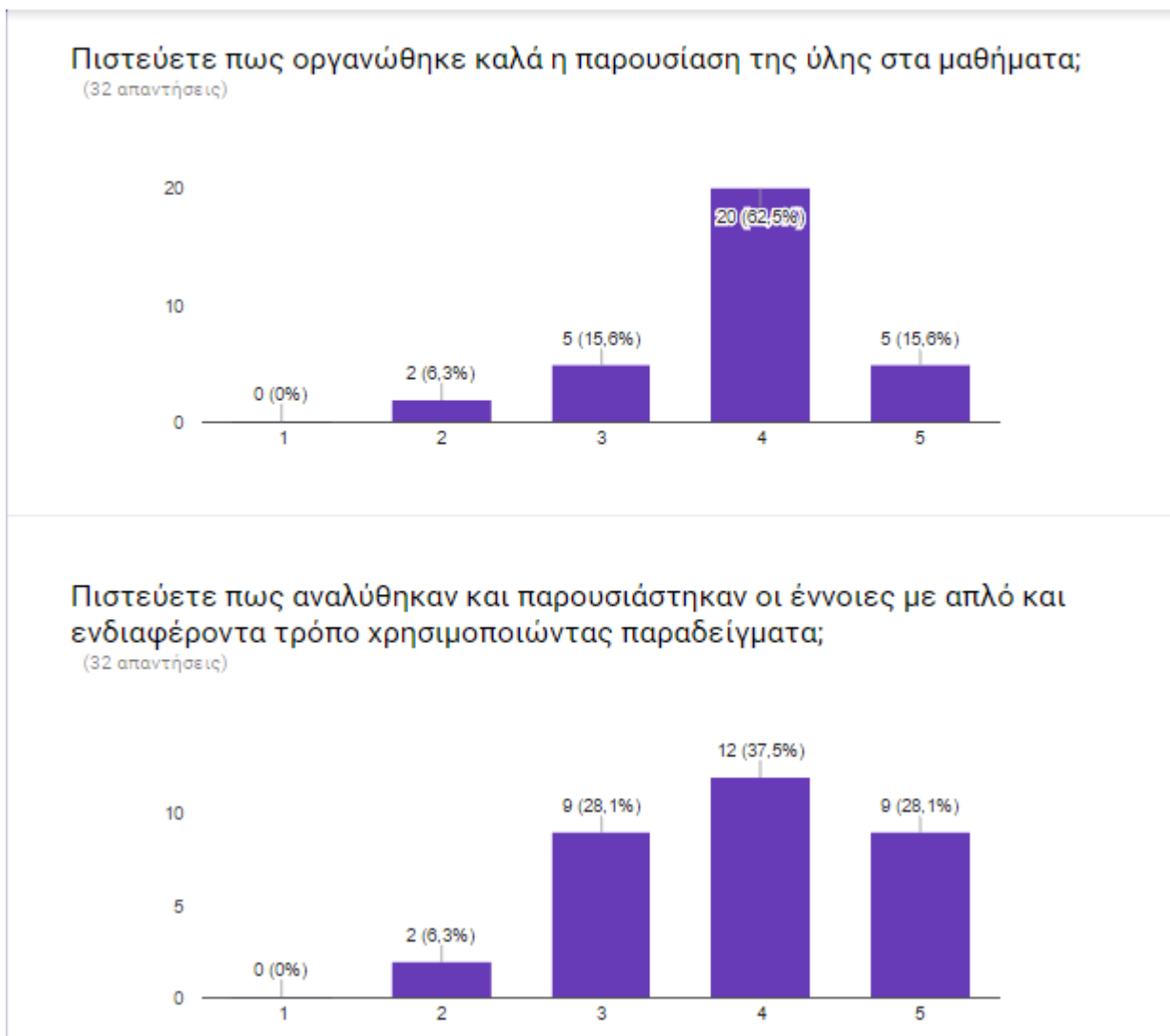


Οι εργασίες που κάνετε σας βοήθησαν να κατανοήσετε το θέμα το οποίο αφορούν;

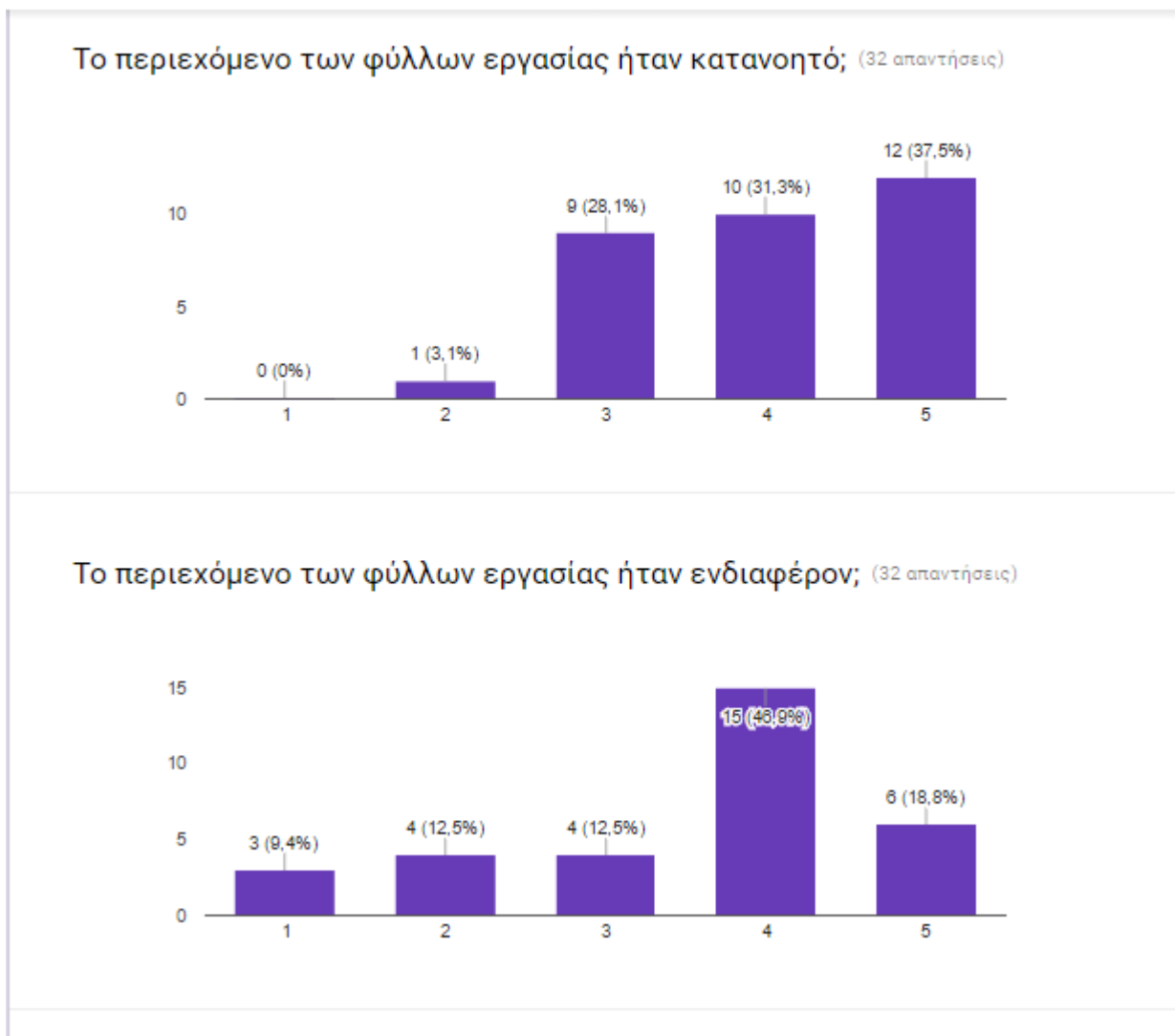
(32 απαντήσεις)



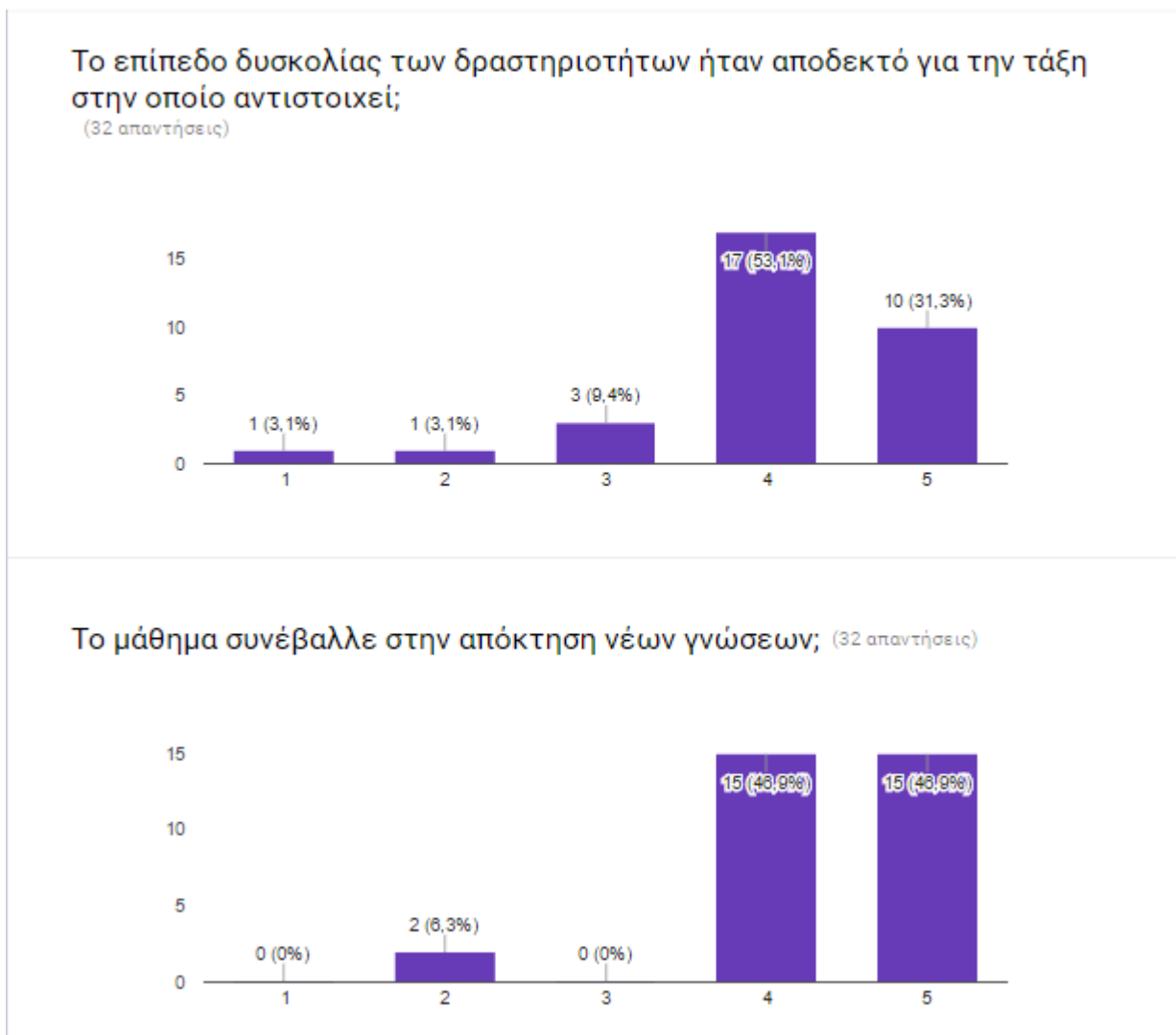
Εικόνα 130: Απαντήσεις στις ερωτήσεις 7-8



Εικόνα 131: Απαντήσεις στις ερωτήσεις 9-10



Εικόνα 132: Απαντήσεις στις ερωτήσεις 11-12

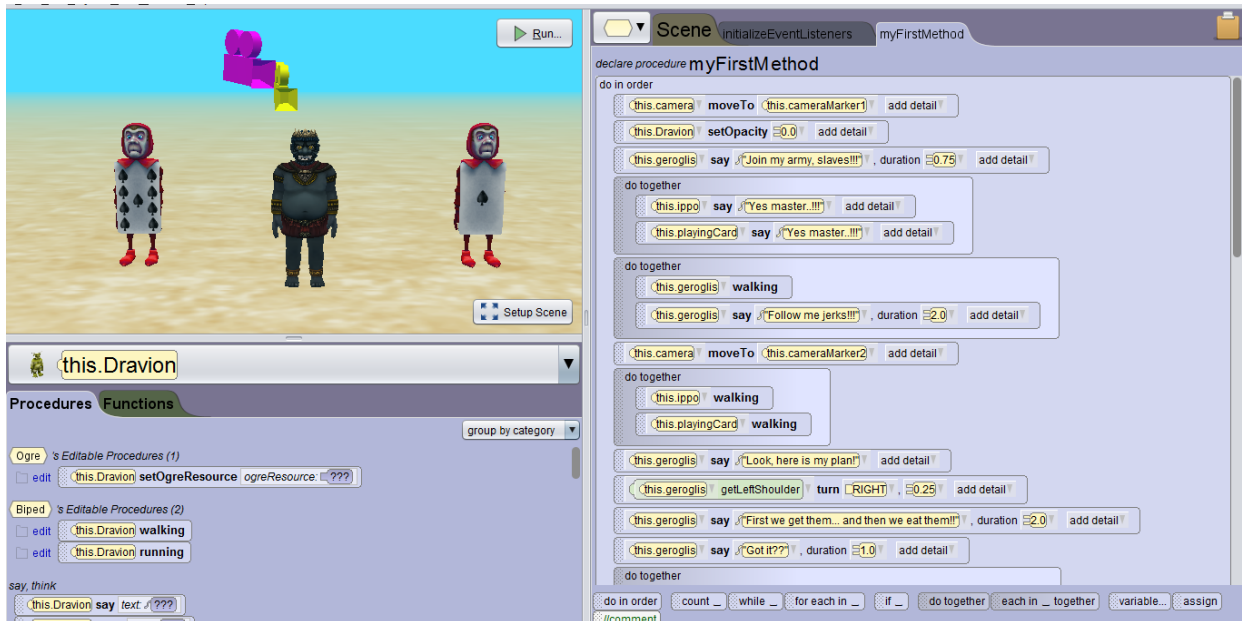


Εικόνα 133: Απαντήσεις στις ερωτήσεις 13-14

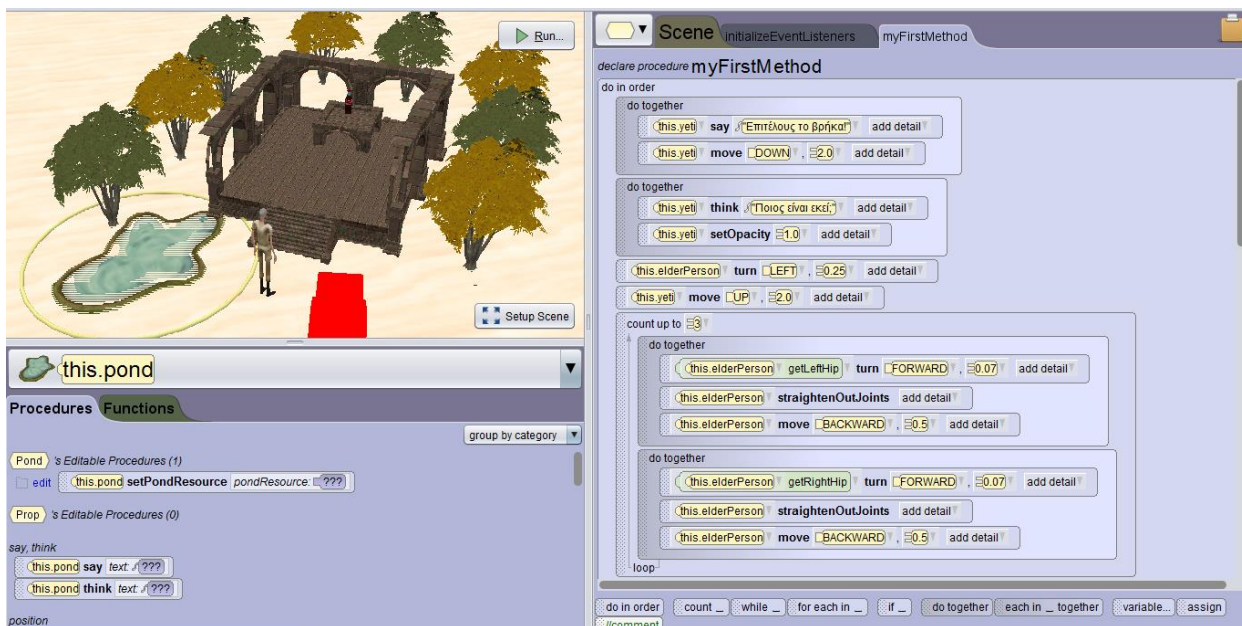
Στις παραπάνω εικόνες παρουσιάζονται τα διαγράμματα που αφορούν τις απαντήσεις των μαθητών στις ερωτήσεις του ερωτηματολογίου. Στις περισσότερες ερωτήσεις παρατηρούμε τα μεγαλύτερα ποσοστά στην στήλη 4 του διαγράμματος (απάντηση «Πολύ») ενώ ακολουθούν οι στήλες 5 και 3 (απαντήσεις «Πάρα Πολύ» και «Αρκετά» αντίστοιχα). Από τα διαγράμματα αυτά, συμπεραίνουμε ότι οι στόχοι των μαθημάτων για το περιβάλλον Alice ήταν σαφείς, το εκπαιδευτικό – διδακτικό υλικό που χρησιμοποιήθηκε βοήθησε στην καλύτερη κατανόηση του περιβάλλοντος και ήταν επαρκές, ενώ η πλειοψηφία των μαθητών θεώρησε και το περιεχόμενο των μαθημάτων «πολύ» χρήσιμο (εικόνα 45,46). Επιπλέον, φαίνεται ότι η καθοδήγηση τους φάνηκε «πάρα πολύ» επαρκής ενώ οι εργασίες που έκαναν τους βοήθησαν «πολύ» να κατανοήσουν το θέμα που αφορούσε η καθεμία (εικόνα 47). Τέλος, οι μαθητές θεώρησαν «πολύ» οργανωτική την παρουσίαση της ύλης στα μαθήματα και «πολύ» χρήσιμα τα παραδείγματα που χρησιμοποιήθηκαν για την ανάλυση και την παρουσίαση διαφόρων εννοιών.

ΠΑΡΑΡΤΗΜΑ II

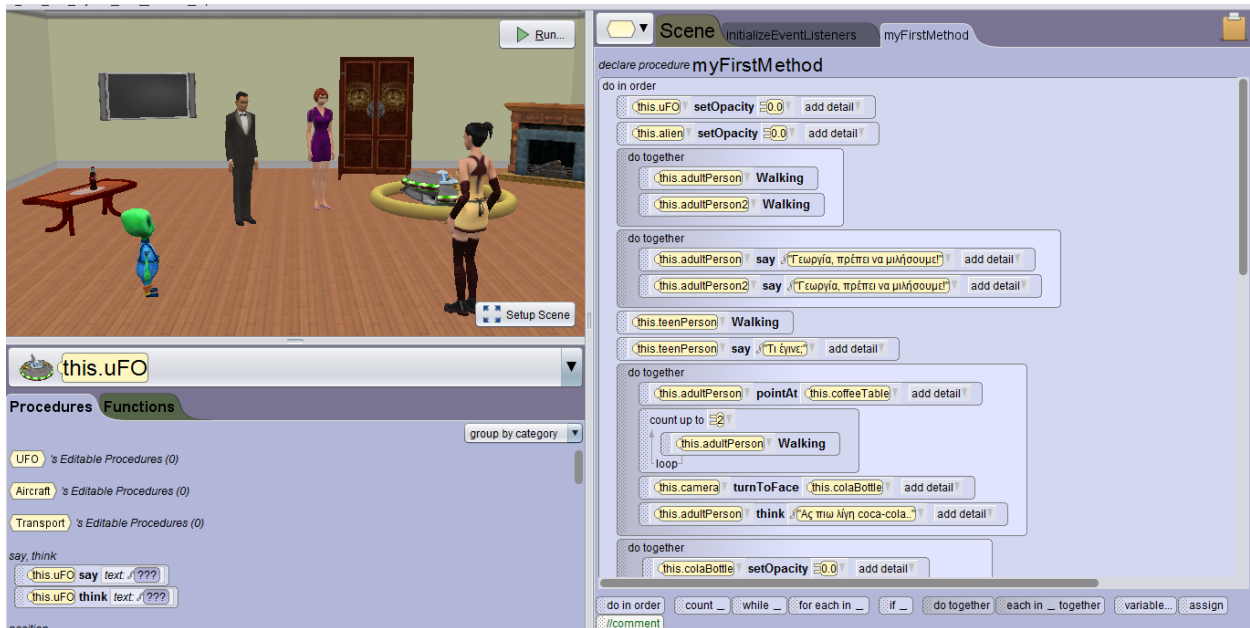
Οι εργασίες των μαθητών



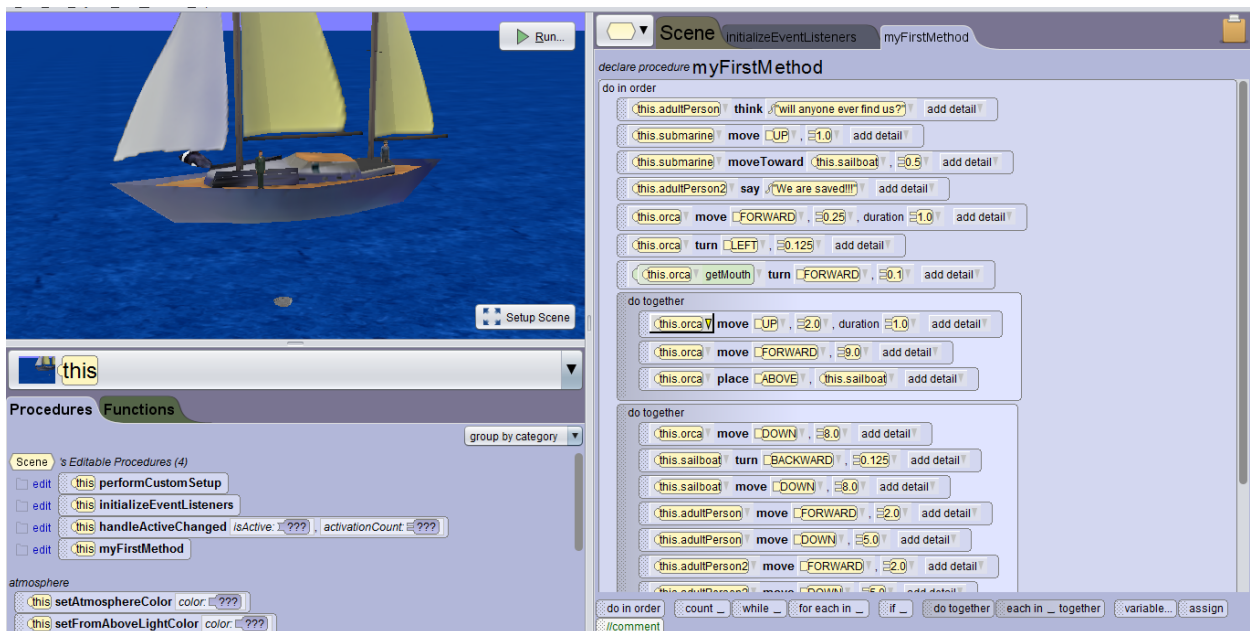
Εικόνα 136: Ομαδική εργασία 1



Εικόνα 137: Ομαδική εργασία 2



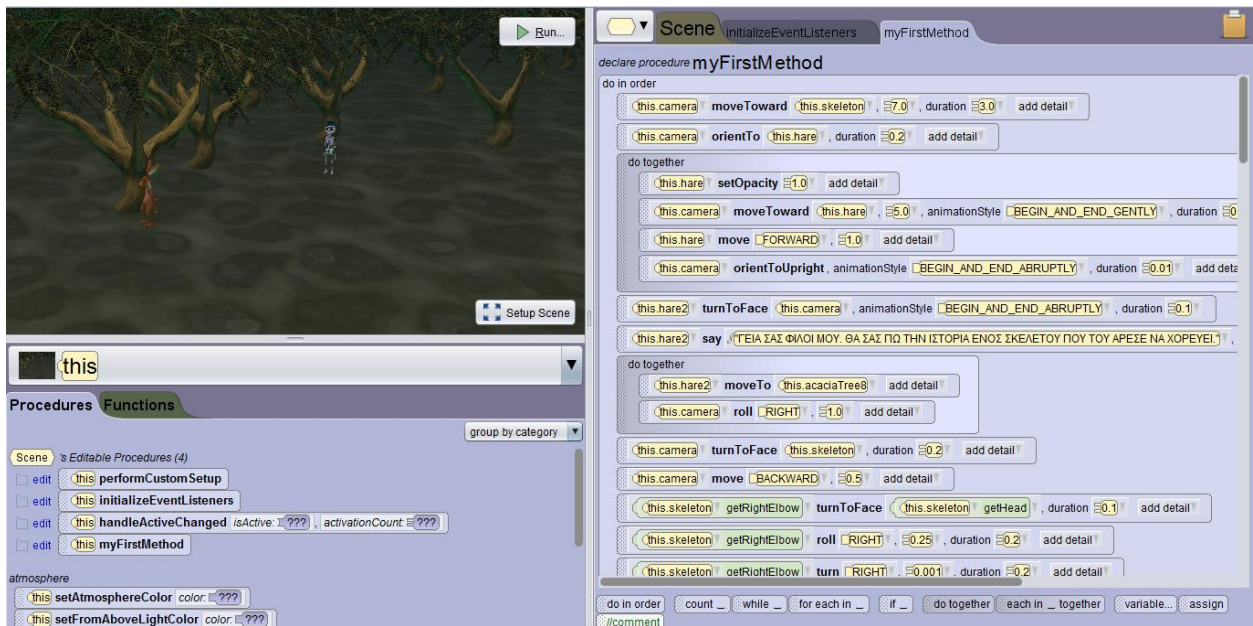
Εικόνα 138: Ομαδική εργασία 3



Εικόνα 139: Ομαδική εργασία 4



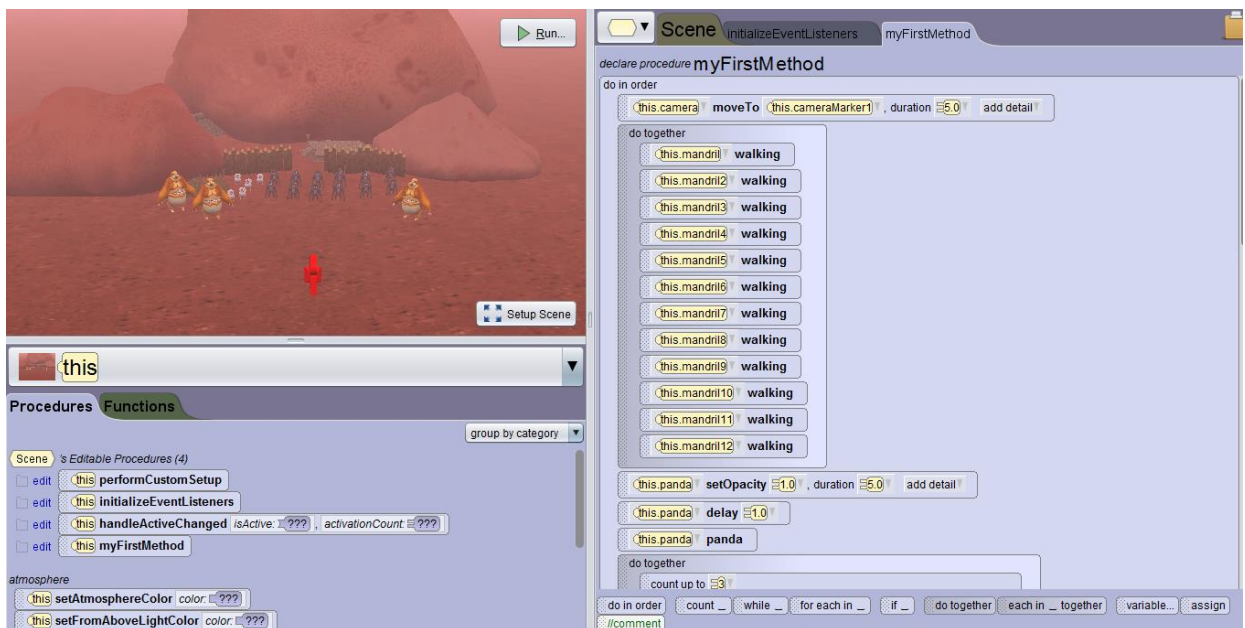
Εικόνα 140: Ομαδική εργασία 5



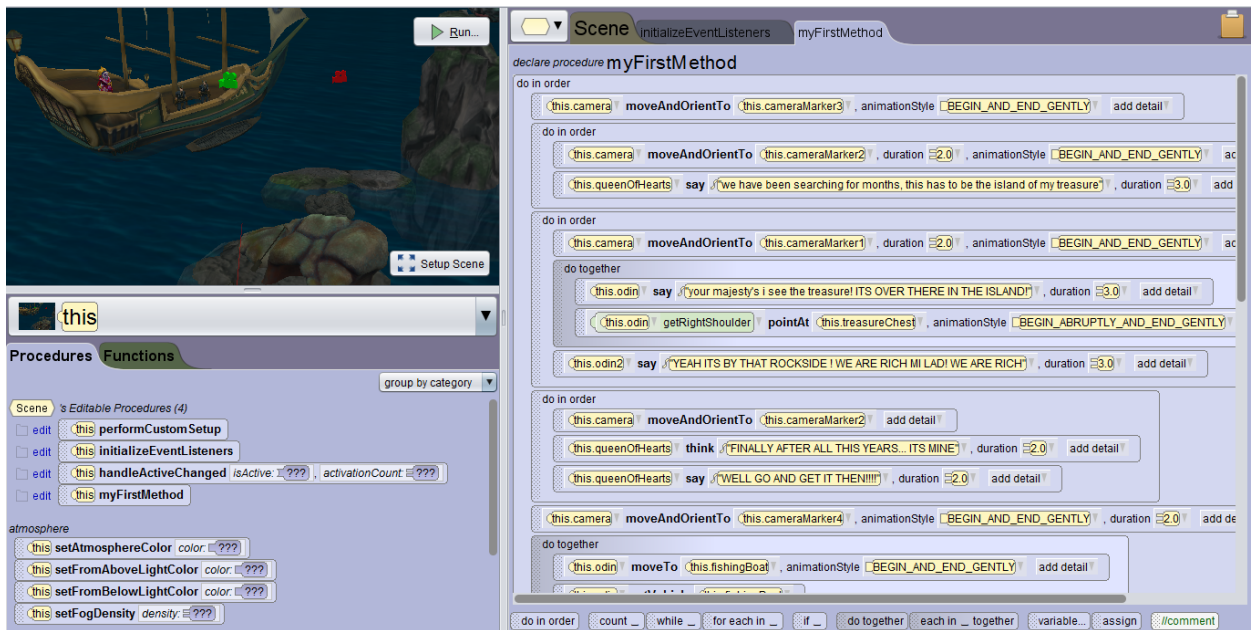
Εικόνα 141: Ομαδική εργασία 6



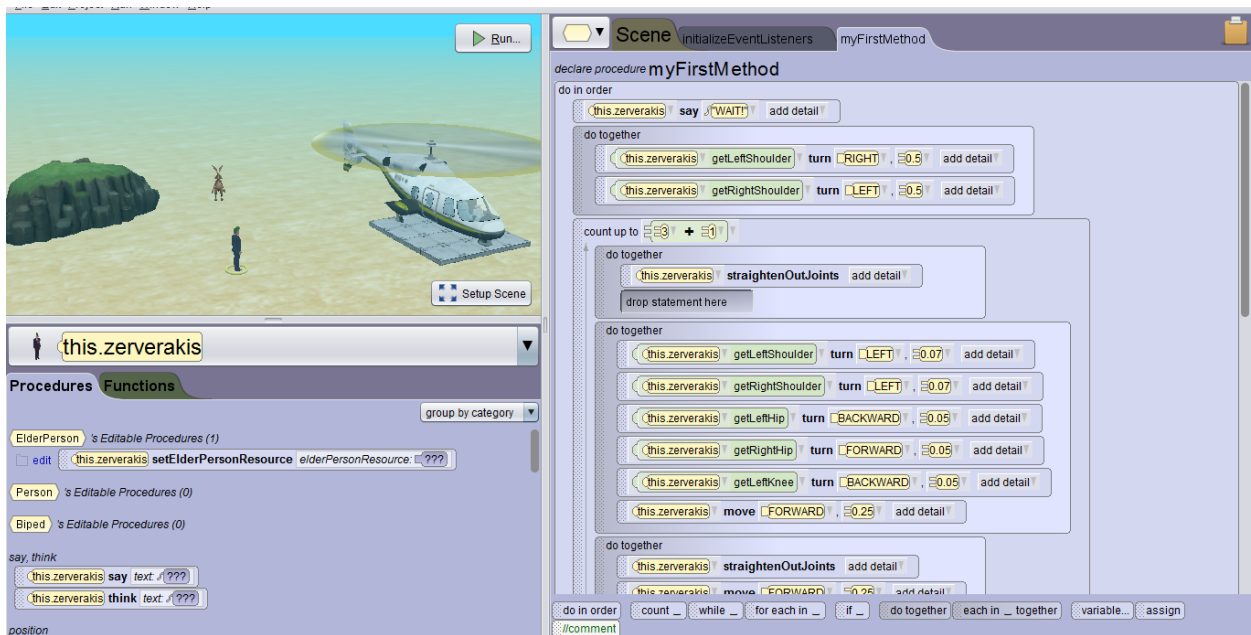
Εικόνα 142: Ομαδική εργασία 7



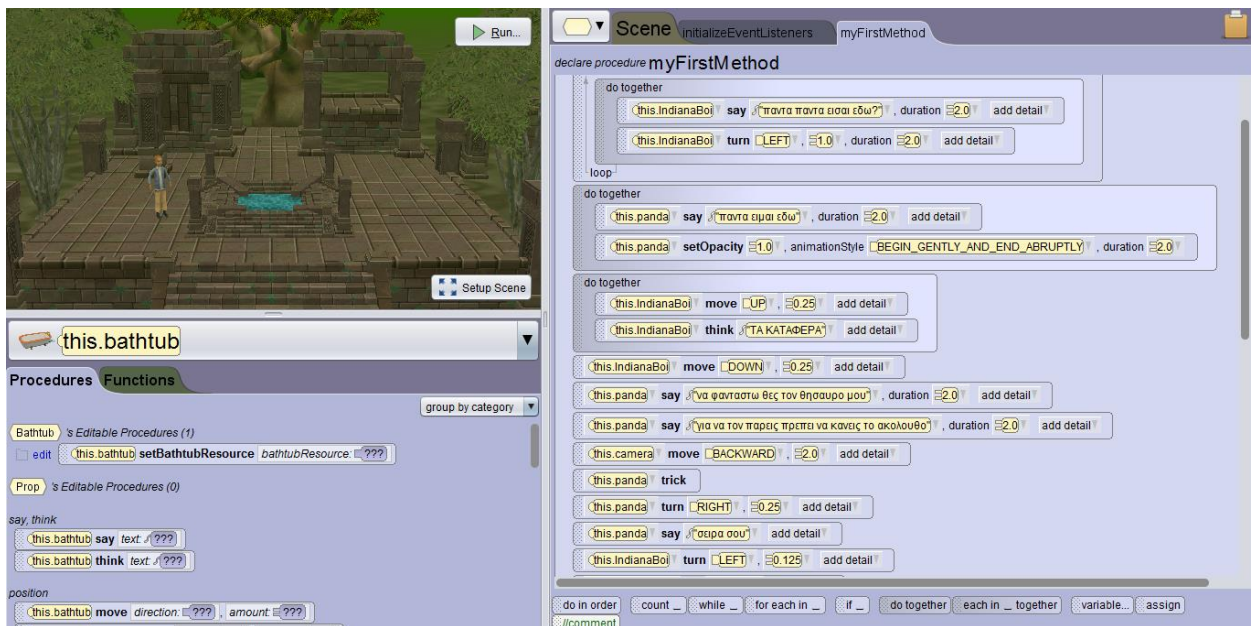
Εικόνα 143: Ομαδική εργασία 8



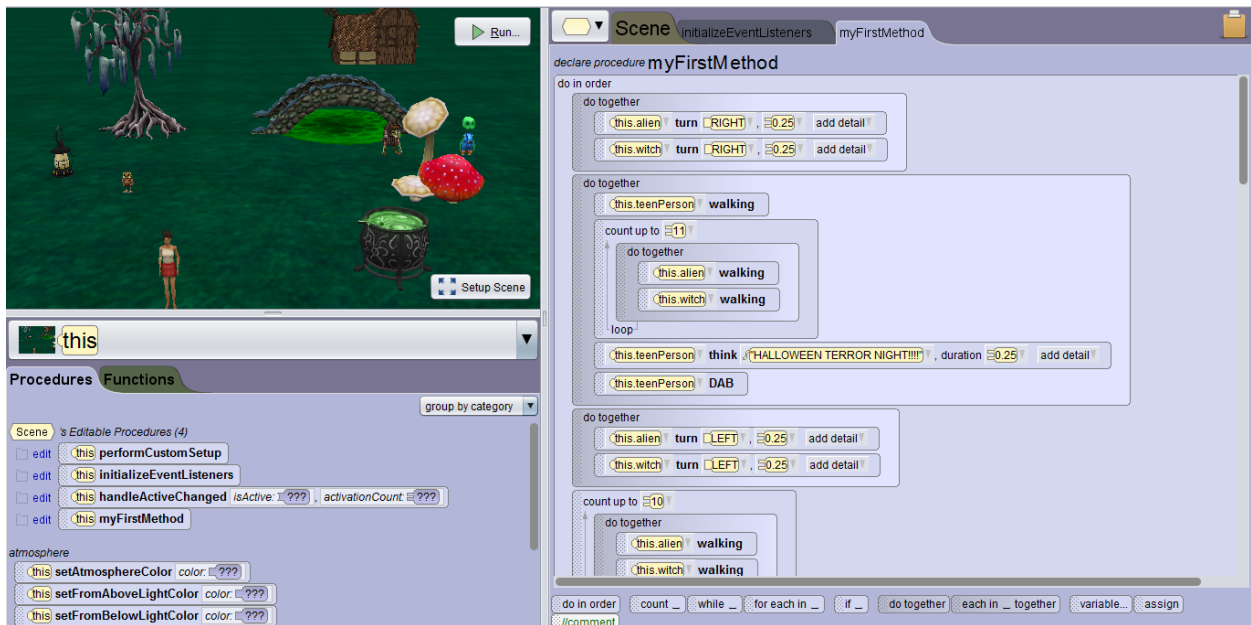
Εικόνα 144: Ομαδική εργασία 9



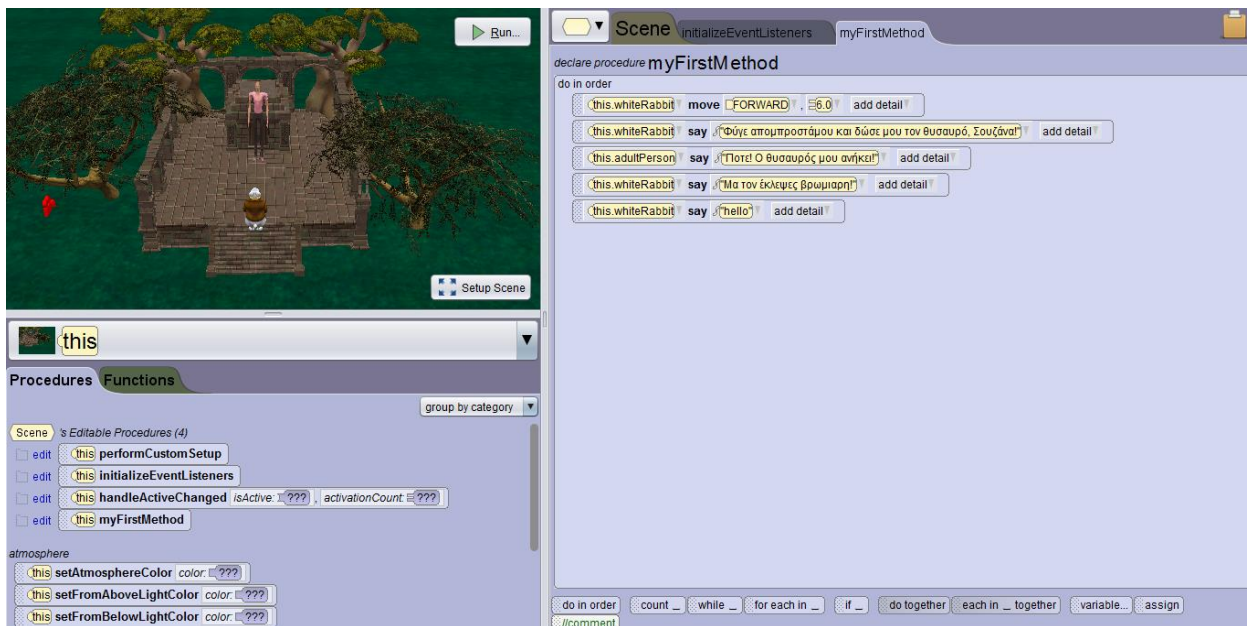
Εικόνα 145: Ομαδική εργασία 10



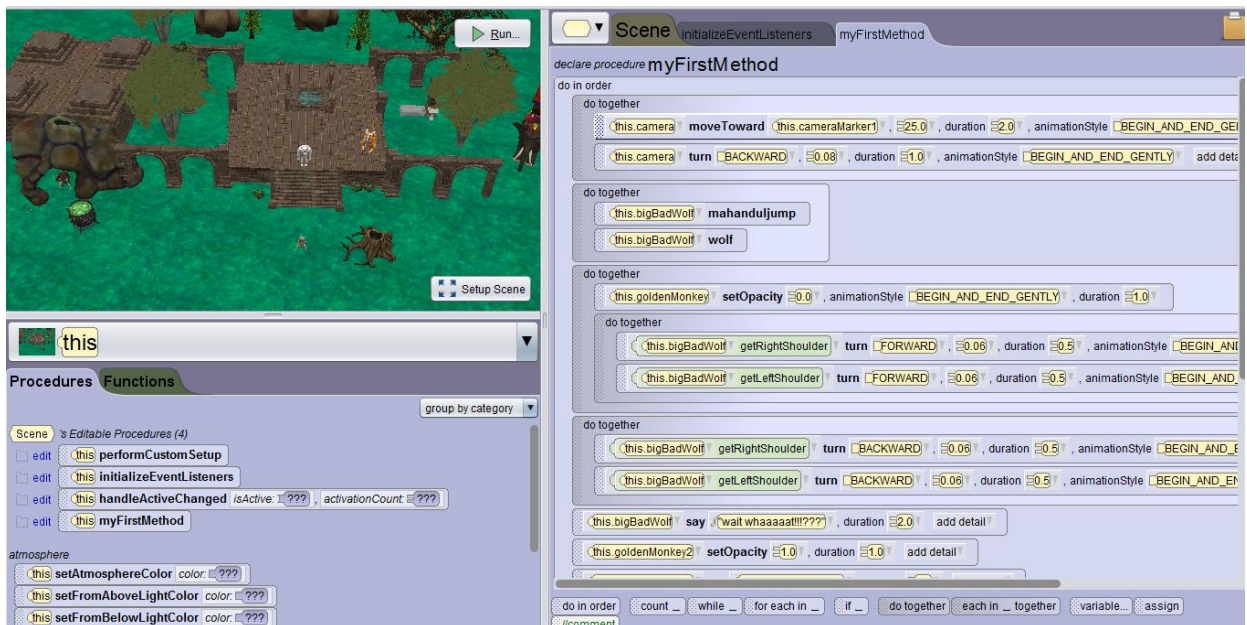
Εικόνα 146: Ομαδική εργασία 11



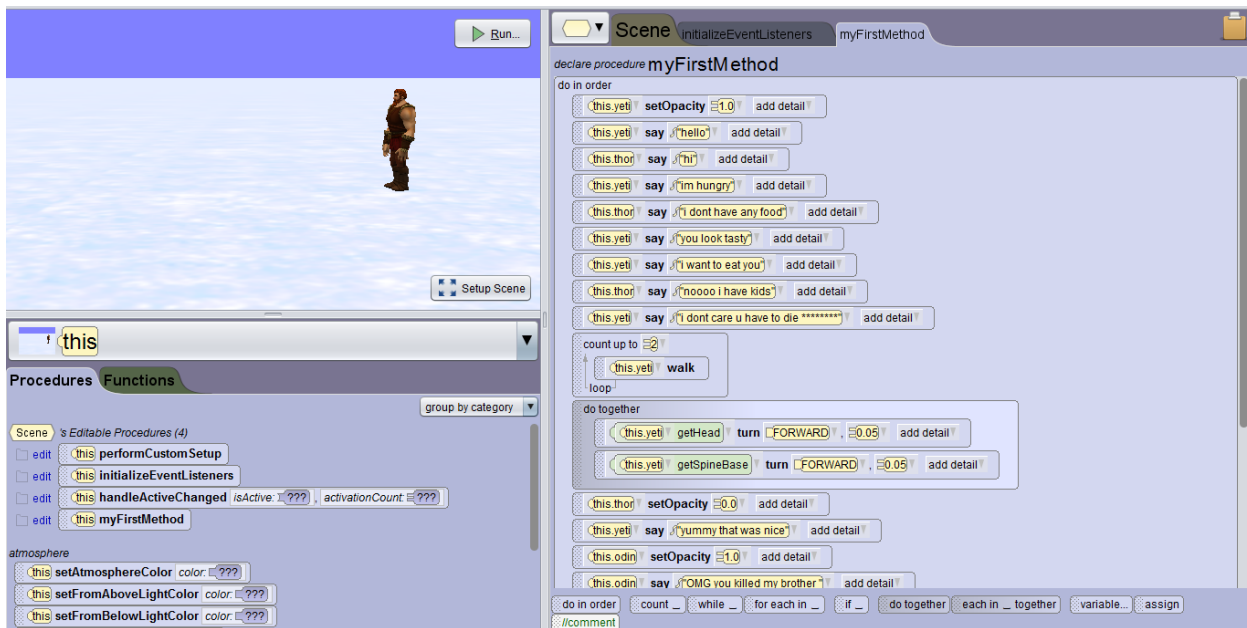
Εικόνα 147: Ομαδική εργασία 12



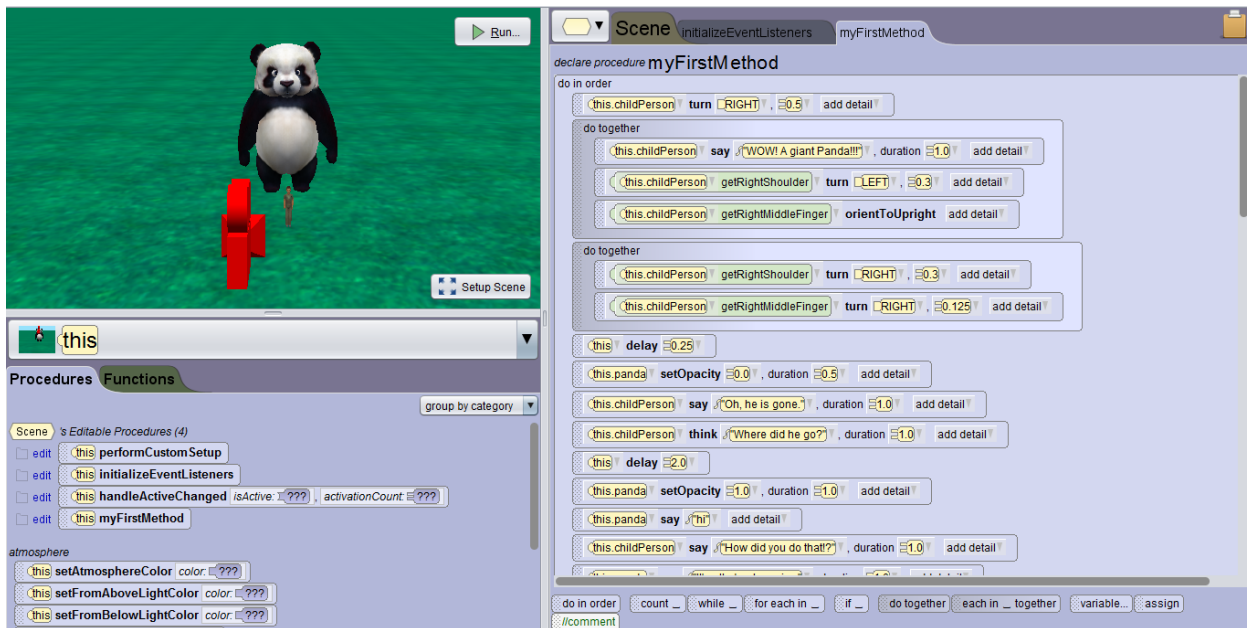
Εικόνα 148: Ομαδική εργασία 13



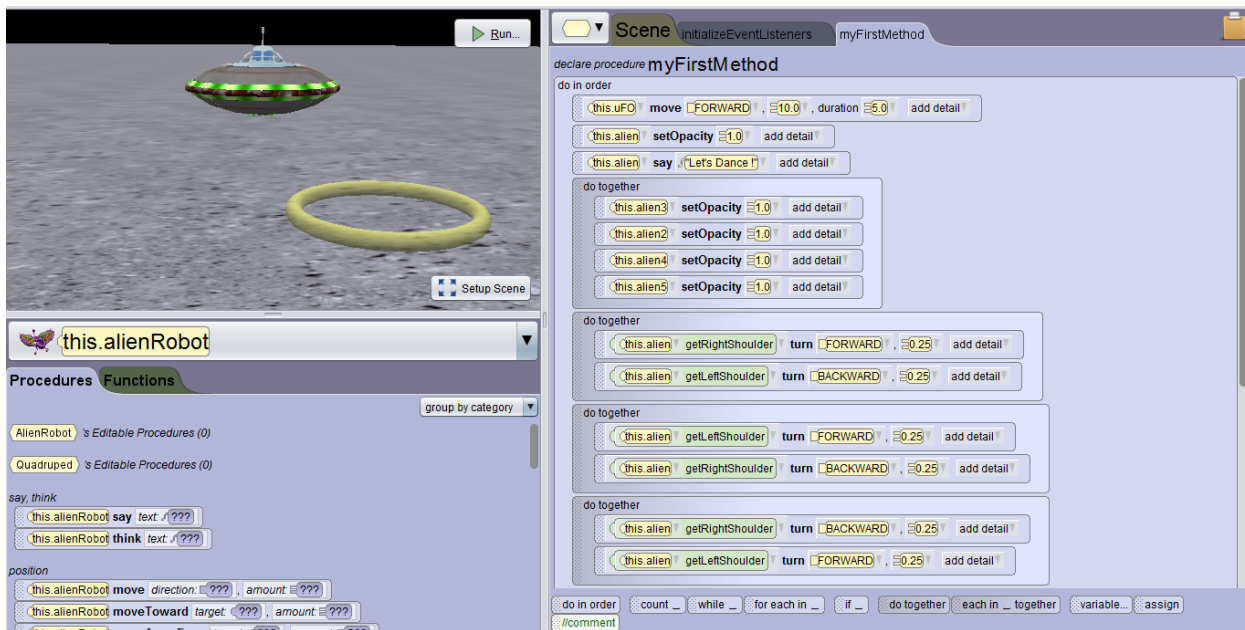
Εικόνα 149: Ομαδική εργασία 14



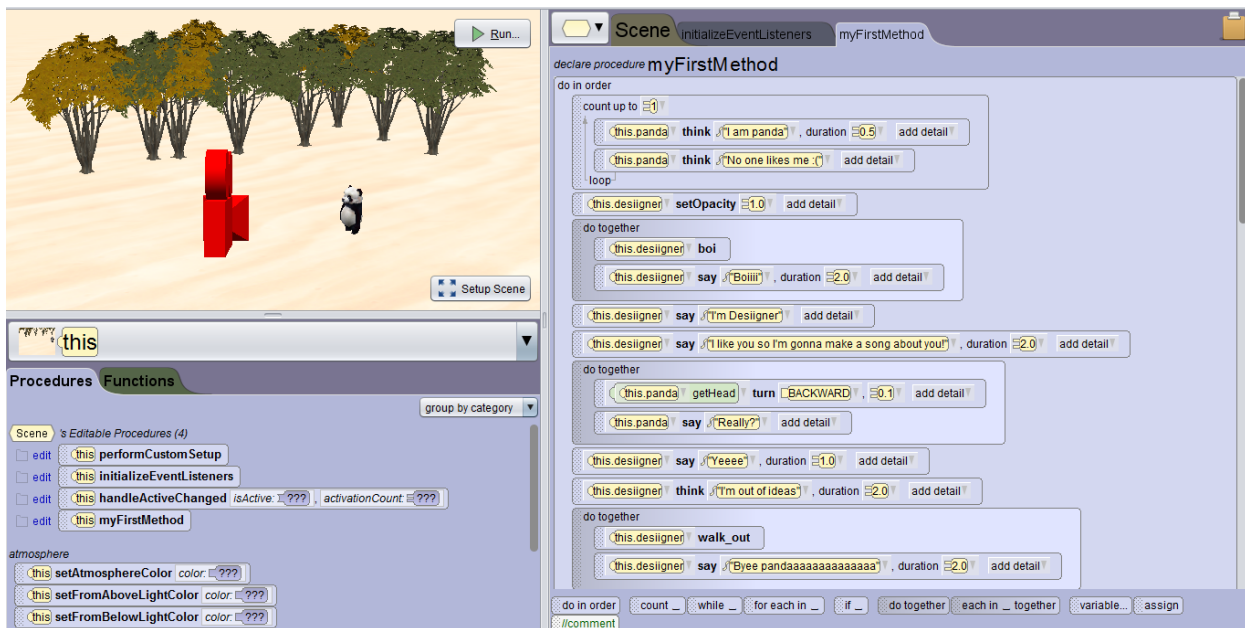
Εικόνα 150: Ομαδική εργασία 15



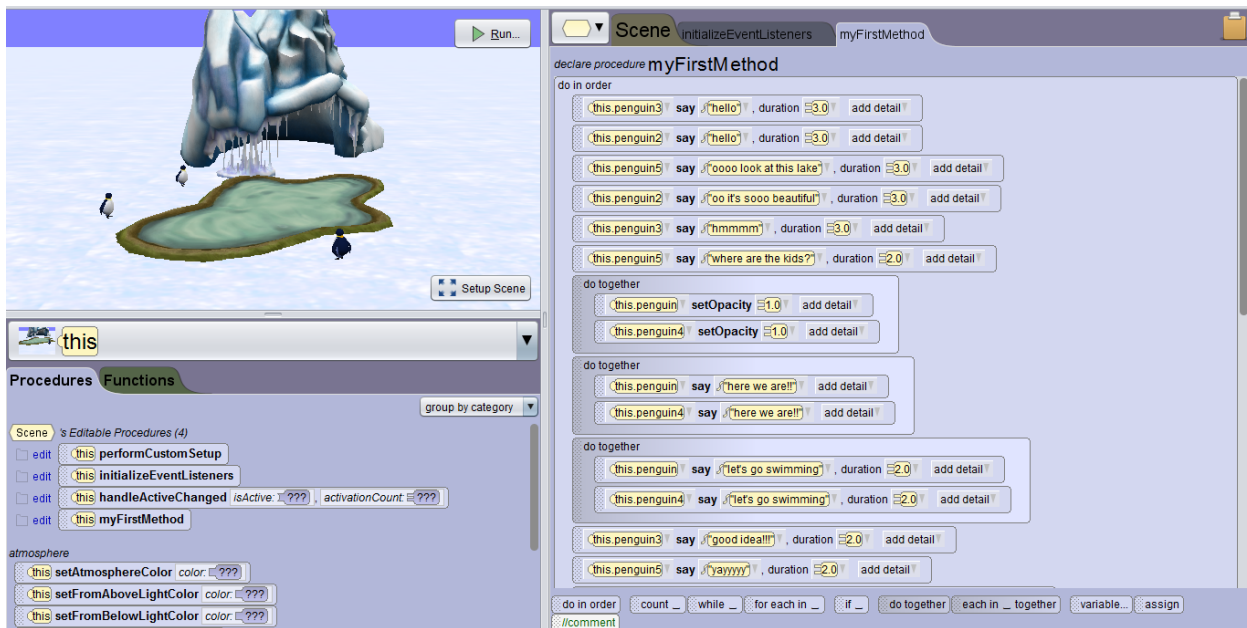
Εικόνα 151: Ομαδική εργασία 16



Εικόνα 152: Ομαδική εργασία 17



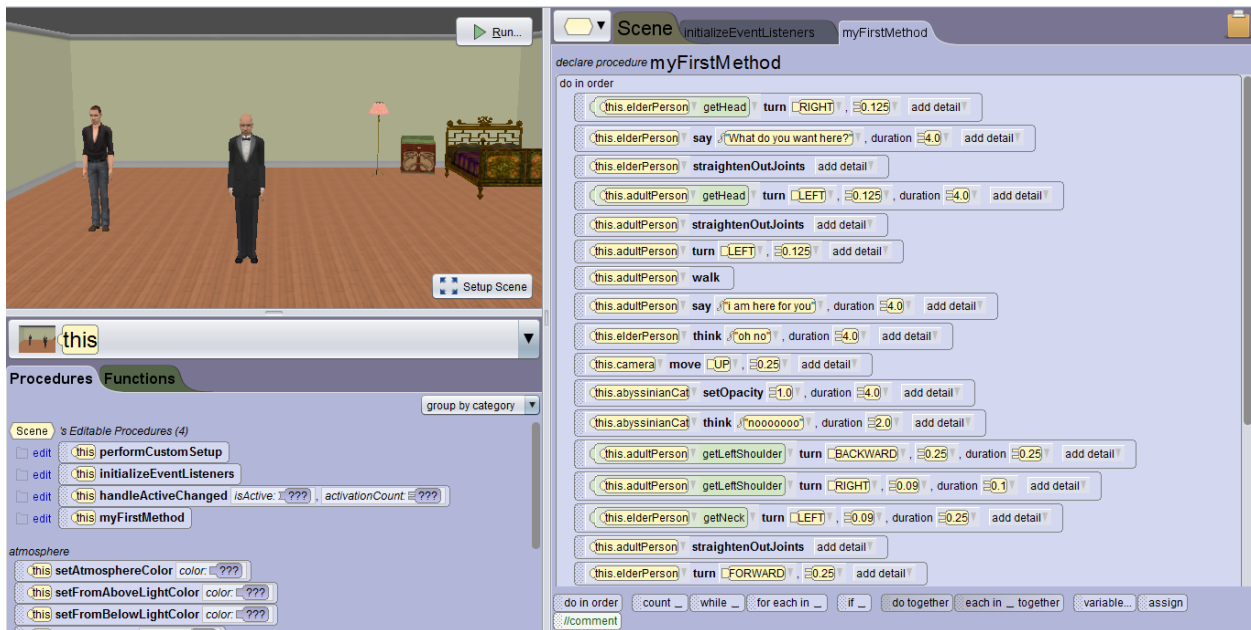
Εικόνα 153: Ομαδική εργασία 18



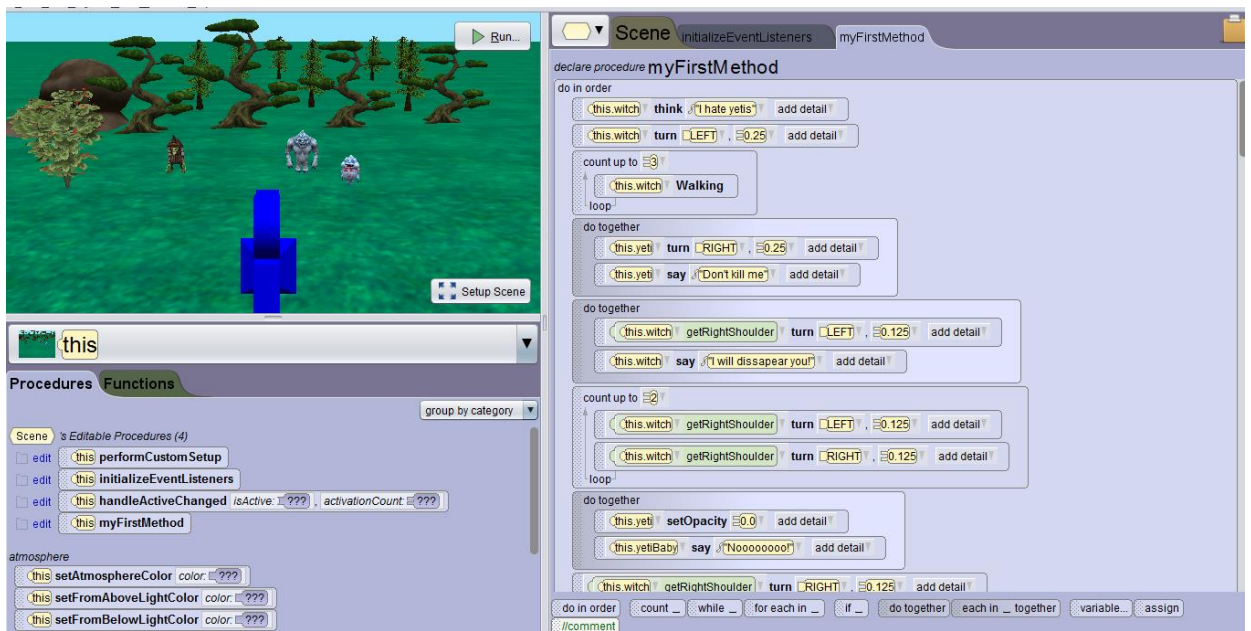
Εικόνα 154: Ομαδική εργασία 19



Εικόνα 155: Ομαδική εργασία 20



Εικόνα 156: Ομαδική εργασία 21



Εικόνα 157: Ομαδική εργασία 22

ΠΑΡΑΡΤΗΜΑ ΙΙΙ

Οι χριστουγεννιάτικες κάρτες των μαθητών

1. ΗΛΙΑΣ ΜΑΝΟΣ – ΓΙΑΝΝΗΣ ΚΑΚΟΥΛΙΔΗΣ



Εικόνα 158: Πίνακας και κάρτα 1

2. ΕΙΡΙΑΝΝΑ ΣΟΚΟΛΗ – ΤΣΙΓΚΑ ANNA



*Σας ευχόμαστε
Χρόνια καλά
& Δημιουργικά!!*

Βαρβάκειο Πρότυπο Λύκειο

Εικόνα 159: Πίνακας και κάρτα 2

3. ΜΙΧΑΛΗΣ ΚΛΕΑΜΑΚΗΣ – ΚΩΣΤΗΣ ΧΑΤΖΑΚΗΣ

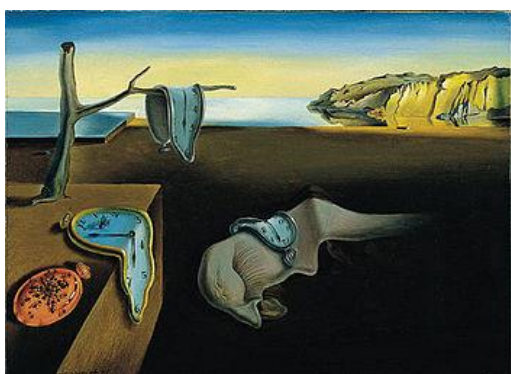


*Πρότυπο Λύκειο
Βαρβακείου Σχολής*

*Σας ευχόμαστε
Χρόνια καλά και Δημιουργικά*

Εικόνα 160: Πίνακας και κάρτα 3

4. ΒΟΥΓΙΟΥΚΑΣ ΗΛΙΑΣ – ΒΑΣΣΑΛΟΣ ΚΩΣΤΑΣ



Πρότυπο Λύκειο
Βαρβακείου Σχολής

Σας ευχόμαστε

Χρόνια καλά
και
δημιουργικά

Εικόνα 161: Πίνακας και κάρτα 4

5. ΑΠΟΣΚΙΤΗ ΖΩΗ



*Πρότυπο
Λύκειο
Βαρβακείου
Σχολής*

Σας ευχόμαστε

*Χρόνια καλά
και
δημιουργικά*

Εικόνα 162: Πίνακας και κάρτα 5

6. ΔΡΥΣ ΟΔΥΣΣΕΑΣ – ΘΕΟΧΑΡΗΣ ΣΑΡΑΝΤΟΣ



*Πρότυπο Λύκειο
Βαρβακείου Σχολής*

Σας ευχόμαστε

*Χρόνια Καλά
και
Δημιουργικά*

**Εικόνα 163:
Πίνακας και κάρτα 6**

7. ΒΑΣΣΗΣ ΧΡΗΣΤΟΣ – ΓΙΑΝΝΗΣ ΓΑΛΙΑΤΣΑΤΟΣ



*Πρότυπο Λύκειο
Βαρβακείου Σχολής*

Σας ευχόμαστε

*Χρόνια καλά
και
δημιουργικά*

Εικόνα 164: Πίνακας και κάρτα 7

8. ΑΔΑΜ ΦΩΤΟΥ – ΣΤΑΜΑΤΟΠΟΥΛΟΣ ΝΙΚΟΣ

Η συγκεκριμένη κάρτα επιλέχθηκε να εκπροσωπήσει το σχολείο για την αποστολή ευχών κατά την εορταστική περίοδο.



*Πρότυπο Λύκειο Βαρβακείου
Σχολής*

Σας ευχόμαστε

Χρόνια καλά και δημιουργικά!



Εικόνα 165: Πίνακας και κάρτα 8

ΑΝΑΦΟΡΕΣ

- [1] http://ys15programmatisosmepolymesa.blogspot.gr/2009_08_01_archive.html
- [2] <http://dide.chi.sch.gr/Themata-Askiseis/AEPP/kefalaio%2010.pdf>
- [3] <http://2laplirforiki.blogspot.gr/2015/03/alice.html>
- [4] <http://www.ct.aegean.gr/people/kordaki/conf-gr/S52.pdf>
- [5] http://www.kau.edu.sa/Files/320/Researches/57381_27616.pdf
- [6] <http://www.etpe.gr/custom/pdf/etpe2342.pdf>
- [7] <http://docplayer.gr/3556349-Diplomatiki-ergasia-thema-meleti-kai-axiologisi-toy-programmatistikoy-perivallontos-alice-kai-anaptyxi-didaktikoy-ylikoy.html>
- [8] <http://oakdome.com/k5/lesson-plans/pivot-animations/introductory-pivot-animator-lessons.php>
- [9] https://en.wikipedia.org/wiki/Pivot_Animator
- [10] <http://users.sch.gr/panosgots/index.php/introprogrammingenvironments/gymnasio/gamemaker>
- [11] <http://users.sch.gr/vorfan/index.php/app-inventor>
- [12] <http://www.appinventor.org/apps2/IHaveADream/IHaveADream.pdf>
- [13] https://xenesglosses.eu/wp-content/uploads/2016/09/%CE%9F%CE%94%CE%97%CE%93%CE%99%CE%95%CE%A3_%CE%95%CE%A6_%CE%A0%CE%9B%CE%97%CE%A1%CE%9F%CE%A6%CE%9F%CE%A1%CE%99%CE%9A%CE%97%CE%A3_%CE%91_%CE%93%CE%95%CE%9B_2016_17.pdf
- [14] Δήμητρα Γ. Μπόμπου, Σταύρος Θ. Κρέτσος, «Ανάπτυξη εκπαιδευτικού υλικού για τη διδασκαλία του αντικειμενοστραφή προγραμματισμού που βασίζεται σε Ηλεκτρονικά Εκπαιδευτικά παιχνίδια», Πτυχιακή Εργασία, Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, 2016.
- [15] Αδάμος Ξ. Κυριακίδης, «Υποστήριξη του μαθήματος Εφαρμογές Πληροφορικής με δραστηριότητες που ακολουθούν τις προδιαγραφές του App Inventor», Πτυχιακή Εργασία, Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, 2016.
- [16] http://www.alice.org/index.php?page=downloads/download_alice