



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΚΠΑΙΔΕΥΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ ΜΕ ΤΟ ΠΕΡΙΒΑΛΛΟΝ ALICE

ΓΕΩΡΓΙΟΥ ΕΥΑΓΓΕΛΙΑ
ΔΟΥΜΠΑ ΧΡΙΣΤΙΝΑ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΤΖΙΑΛΛΑΣ ΓΡΗΓΟΡΙΟΣ
ΚΑΘΗΓΗΤΗΣ

Λαμία έτος 2022-2023



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΚΠΑΙΔΕΥΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ ΜΕ ΤΟ ΠΕΡΙΒΑΛΛΟΝ ALICE

ΓΕΩΡΓΙΟΥ ΕΥΑΓΓΕΛΙΑ
ΔΟΥΜΠΑ ΧΡΙΣΤΙΝΑ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΤΖΙΑΛΛΑΣ ΓΡΗΓΟΡΙΟΣ
ΚΑΘΗΓΗΤΗΣ

Λαμία έτος 2022-2023



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

EDUCATIONAL SCENARIOS WITH THE PROGRAM ALICE

GEORGIOU EVANGELIA
DOUMPA CHRISTINA

FINAL THESIS

ADVISOR

TZIALLAS GRIGORIOS
PROFESSOR

Lamia year 2022-2023

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 20/3/2023

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Σε όλα τα προγράμματα σπουδών ο προγραμματισμός είναι άρρηκτα συνδεδεμένος με το μάθημα της πληροφορικής. Στην παρούσα πτυχιακή εργασία παρουσιάστηκε ένας εναλλακτικός τρόπος διδασκαλίας του αντικειμενοστραφή προγραμματισμού με τη δημιουργία ενός παιχνιδιού στο λογισμικό Alice 3. Στόχος του παιχνιδιού είναι η εκμάθηση εισαγωγικών εννοιών σε γλώσσα Java από τους μαθητές και η εξοικείωση τους με το προγραμματιστικό περιβάλλον του Alice. Επίσης έχουν συμπεριληφθεί κομμάτια κώδικα που έχουν γραφτεί εξ ολοκλήρου από εμάς και έχουν αναλυθεί εκτενέστερα σε αντίστοιχα κεφάλαια.

ABSTRACT

In all curricula, programming is inextricably linked to computing. In this thesis we presented an alternative way of teaching object-oriented programming by creating a game in Alice 3 software. The aim of the game is to teach students introductory concepts in Java language and to be conversant and familiar with the environment of Alice. There are also included pieces of code written entirely by us and are discussed in more detail in corresponding chapters.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	III
<u>ΠΕΡΙΕΧΟΜΕΝΑ.....</u>	0
<u>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ</u>	2
<u>ΚΕΦΑΛΑΙΟ 2 ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ</u>	3
(2.1) ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ.....	3
(2.2) Η ΠΛΗΡΟΦΟΡΙΚΗ ΣΤΟ ΠΕΡΑΣΜΑ ΤΟΥ ΧΡΟΝΟΥ	3
(2.3) ΟΦΕΛΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	3
(2.3) Η ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΤΩΝ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΛΟΓΙΣΜΙΚΩΝ	4
(2.4) ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ.....	4
<u>ΚΕΦΑΛΑΙΟ 3 ALICE 3</u>	5
(3.1)ΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΕΡΓΑΛΕΙΟ ALICE.....	5
(3.2) ΛΟΓΙΣΜΙΚΟ ALICE	5
(3.3)Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ ALICE	6
<u>ΚΕΦΑΛΑΙΟ 4 ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΣΕΝΑΡΙΟΥ</u>	8
(4.1)ΠΡΩΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ	8
(4.2)ΔΕΥΤΕΡΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ	9
(4.3)ΤΡΙΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ.....	10
(4.4)ΤΕΤΑΡΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ	11
<u>ΚΕΦΑΛΑΙΟ 5 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ</u>	12
(5.1)ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΠΑΙΧΝΙΔΙΟΥ ΣΤΟ ALICE	12
(5.2)ΈΝΑΡΞΗ ΠΑΙΧΝΙΔΙΟΥ	12
.....	15
.....	15
.....	16
(5.3)ΠΡΩΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ-ΜΕΤΑΒΛΗΤΕΣ.....	16
(5.4)ΔΕΥΤΕΡΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ-ΔΟΜΕΣ ΕΠΑΝΑΛΗΨΗΣ	25
(5.5)ΤΡΙΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ-ΤΕΛΕΣΤΕΣ ΚΑΙ ΔΟΜΕΣ ΕΛΕΓΧΟΥ	35
(5.6)ΤΕΤΑΡΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ-ΜΕΘΟΔΟΙ	44
<u>ΚΕΦΑΛΑΙΟ 6 ΑΞΙΟΛΟΓΗΣΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ.....</u>	58
(6.1) ΑΠΟΔΟΤΙΚΟΤΗΤΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ALICE ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ	58

(6.2) ΠΑΡΟΥΣΙΑΣΗ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ	59
(6.3) ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ	60
<u>ΚΕΦΑΛΑΙΟ 7 ΕΠΙΛΟΓΟΣ.....</u>	<u>62</u>
<u>ΒΙΒΛΙΟΓΡΑΦΙΑ</u>	<u>63</u>

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

Η συνεχής και ραγδαία ανάπτυξη της τεχνολογίας σε συνδυασμό με τη μετάδοση της πληροφορίας σε γρήγορους ρυθμούς έχει επιφέρει σημαντικές εξελίξεις σε διάφορους κλάδους, με την εκπαίδευση να είναι μέσα σε αυτούς. Τα προγράμματα σπουδών τις τελευταίες δεκαετίες σε όλο τον κόσμο έχουν ενσωματώσει το μάθημα του προγραμματισμού τόσο στην πρωτοβάθμια όσο και στη δευτεροβάθμια εκπαίδευση. Σύμφωνα με τον Papert η επιστήμη των υπολογιστών δεν είναι μονό εργαλείο εξήγησης του τρόπου με τον οποίο οι άνθρωποι μαθαίνουν και σκέφτονται, αλλά και εργαλείο αλλαγής που θα μπορέσει να βελτιώσει και να αλλάξει τους τρόπους αυτούς. (Folk, 1981)

Τα τελευταία χρόνια έχουν δημιουργηθεί διάφορα εργαλεία και προγραμματιστικά περιβάλλοντα που εμπλουτίζουν τη γνώση και τη διαδικασία της διδασκαλίας του προγραμματισμού. Ένας από τους σκοπούς αυτών των προγραμμάτων είναι η χρήση και ο συνδυασμός των αισθητηριακών τύπων πρόσληψης πληροφοριών από το περιβάλλον (οπτικός, ακουστικός, λεκτικός κλπ). Διάφορες μελέτες επισημαίνουν ότι μεγάλος όγκος πληροφοριών που λαμβάνει ο εγκέφαλος προέρχεται από οπτικά μέσα και η εφαρμογή αυτών στην εκπαίδευση βοηθάει στην καλύτερη κατανόηση της πληροφορίας από τους μαθητές. Η χρήση αυτών έχει ως αποτέλεσμα στην ανάπτυξη των προγραμματιστικών δεξιοτήτων των παιδιών, στην ενίσχυση της υπολογιστικής και λογικής σκέψης, της δημιουργικότητας καθώς και στην επίλυση προβλημάτων. Επιπλέον αξίζει να σημειωθεί ότι αυτές οι δεξιότητες είναι πλέον απαραίτητες για τη σύγχρονη ψηφιακή εποχή.

ΚΕΦΑΛΑΙΟ 2 Διδακτική της Πληροφορικής

(2.1) Αντικείμενο της Διδακτικής της Πληροφορικής

Η Διδακτική της πληροφορικής ορίζεται ως η μελέτη των διαδικασιών μετάδοσης και οικοδόμησης των γνώσεων έχοντας ως απώτερο στόχο τη βελτίωση αυτών των διαδικασιών (Vergnaud, 1994). Η αναπαράσταση της Διδακτικής συνήθως γίνεται με ένα τρίγωνο συμβολίζοντας το σύστημα που συνδέει τις γνώσεις, το μαθητή και τον εκπαιδευτικό. Είναι σημαντικό να αναφερθεί ότι κάθε κορυφή του τριγώνου αποτελεί ένα χώρο έρευνας που δεν αφορά αυτή καθαυτή την Διδακτική (Κόμης, 2001). Ωστόσο, από την συγκεκριμένη αναπαράσταση δεν καλύπτονται πλήρως οι μέθοδοι διδασκαλίας, εφόσον δεν συμπεριλαμβάνονται παράγοντες όπως οι διαπροσωπικές σχέσεις μεταξύ μαθητών και εκπαιδευτικού, το τεχνολογικό περιβάλλον της διδασκαλίας κ.ά.

(2.2) Η Πληροφορική στο πέρασμα του χρόνου

Η διδασκαλία της Πληροφορικής είναι μια ολιγόχρονη διαδρομή και η ένταξη της στα σχολεία έγινε τις τελευταίες δεκαετίες, επομένως αποτελεί κατηγορία σύγχρονης μικροϊστορίας. Η πληροφορική ως μάθημα δεν έχει ως στόχο την κατανόηση της λειτουργίας του μηχανήματος αλλά την απόκτηση των υπολογιστικών δεξιοτήτων. Συμπληρωματικά έχει δύο σκοπούς, να διδάσκεται και να διδάσκει, δηλαδή να συνίσταται αντικείμενο διδασκαλίας αλλά και μέσο διδασκαλίας.

Η εισαγωγή της πληροφορικής στην ελληνική εκπαίδευση ξεκίνησε στα:

- Τεχνικά- Επαγγελματικά και τα Πολυκλαδικά Λύκεια κατά την περίοδο 1983-1985
- Γυμνάσια, από το 1992 όπου και ολοκληρώθηκε μετά από μερικά χρόνια.
- Γενικό Λύκειο, από το 1998, και ολοκληρώθηκε μετά από μερικά χρόνια.
- Πρωτοβάθμια Εκπαίδευση & Προσχολική Εκπαίδευση (2003), με την καθιέρωση ενός προγράμματος σπουδών και τον σταδιακό εξοπλισμό των σχολείων με υπολογιστές. (Κόμης, 2004)

(2.3) Οφέλη του προγραμματισμού

«Η Επιστήμη των Υπολογιστών διαφέρει από τη Φυσική με την έννοια ότι δεν είναι απόλυτα επιστήμη, καθώς δεν μελετάει φυσικά αντικείμενα. Αντιθέτως, η Επιστήμη των Υπολογιστών μοιάζει με τη Μηχανική και δεν ασχολείται με αφαιρετικούς συλλογισμούς...» (Taylor and Francis, 2018)

Πολλοί επιστήμονες έχουν αναφερθεί στα πολλαπλά οφέλη του προγραμματισμού και παρακάτω θα αναφερθούν κάποια από αυτά.

- Πρώτο και κυριότερο τη μάθηση ενός δομημένου και ιεραρχικού τρόπου σκέψης και την εφαρμογή του σε καθημερινά προβλήματα που έρχονται αντιμέτωποι οι μαθητές.
- Επιπρόσθετα διευρύνουν τις διανοητικές τους ικανότητες και την φαντασία τους, έχουν την ικανότητα να κατασκευάζουν παιχνίδια και να εφαρμόζουν στην πράξη τις ιδέες τους. Χάρη σε αυτά οι μαθητές νιώθουν κυρίαρχοι της νέας τεχνολογίας.

- Στη συνέχεια ένα ακόμα πλεονέκτημα είναι να αποκτούν την ικανότητα να κατανοούν γενικές έννοιες που έχουν σχέση με το μάθημα των Μαθηματικών, όπως η συνάρτηση, η μεταβλητή, η διαδικασία κ.ά.
- Επιπλέον, οδηγούνται στη λύση με τη μέθοδο κατακερματισμού του προβλήματος. Δηλαδή τη διάσπαση του σε απλούστερα κομμάτια τα οποία μπορούν να συνδεθούν και να λυθούν ευκολότερα.
- Συμπληρωματικά ο προγραμματισμός βοηθάει στην καλύτερη κατανόηση και εμπέδωση των θεωρητικών εννοιών του μαθήματος σε σχέση με τα παραδοσιακά διδακτικά μέσα.
- Τέλος, η διαδικασία αποσφαλμάτωσης (Debugging) δίνει τη δυνατότητα στους μαθητές να εντοπίζουν και να διορθώνουν τα λάθη τους που υποδεικνύει ώριμο αναπτυξιακό επίτευγμα. Ο σημαντικότερος όμως παράγοντας είναι ότι γίνεται απενοχοποίηση των λαθών και τα παιδιά αντιλαμβάνονται ότι ο σκοπός τους είναι να μαθαίνουν μέσα από αυτά.

(2.3) Η Σημαντικότητα των Εκπαιδευτικών Λογισμικών

Τα σύγχρονα και εξελιγμένα εργαλεία προγραμματισμού που έχουν στην κατοχή τους οι εκπαιδευτικοί, έχουν δώσει τη δυνατότητα στους μαθητές να "καλλιεργήσουν" και να εμβαθύνουν αναγκαίες δεξιότητες γνωστές ως "4C" (Critical thinking and problem solving, Communication, Collaboration, and Creativity) που παραπέμπουν στην κριτική σκέψη και επίλυση προβλημάτων, στην επικοινωνία, στην συνεργασία και τέλος στη δημιουργία και στην καινοτομία. Βάση ερευνών η χρήση εκπαιδευτικών λογισμικών συμβάλει στην καλύτερη και πιο αποτελεσματική προσέγγιση του μαθήματος. Έχει παρατηρηθεί ότι όχι μόνο ο μαθητής παρακινείται να συμμετέχει πιο ενεργά στη διεξαγωγή του μαθήματος χρησιμοποιώντας τις προγραμματιστικές του ικανότητες, αλλά και αντιμετωπίζει την Πληροφορική ως τρόπο διασκέδασης. Επιπλέον, αντιλαμβάνεται ότι με τα εργαλεία προγραμματισμού που του δίνονται μπορεί να δώσει "ζωή" στις σκέψεις και στις ιδέες του.

(2.4) Προγραμματιστικά Εργαλεία που μπορούν να χρησιμοποιηθούν στην εκπαίδευση

Διάφορα προγραμματιστικά εργαλεία και περιβάλλοντα έχουν δημιουργηθεί για την ενίσχυση της διδασκαλίας στην Πληροφορική και πιο συγκεκριμένα για την διδασκαλία του Αντικειμενοστραφή Προγραμματισμού. Αρχικά το KarelJ Robot είναι ένα προγραμματιστικό περιβάλλον που δίνει τη δυνατότητα διαγραμματικής παρουσίασης του προγράμματος με τη χρήση διαγραμμάτων ροής. Στη συνέχεια το objectKarel είναι ένα εκπαιδευτικό περιβάλλον το οποίο προσφέρει επεξηγηματική οπτικοποίηση για την δημιουργία προγραμμάτων. Ένα ακόμα προγραμματιστικό περιβάλλον είναι το Jeroo που εστιάζει στις έννοιες «αντικείμενο» και «μέθοδος» και στις δομές ελέγχου. Επιπλέον το BlueJ το οποίο χρησιμοποιεί την «object-first» προσέγγιση και το Greenfoot που υποστηρίζει την τεχνική της οπτικοποίησης σε διδιάστατο πλέγμα. Τέλος είναι το Alice το οποίο θα αναλυθεί παρακάτω στην παρούσα πτυχιακή.

ΚΕΦΑΛΑΙΟ 3 ALICE 3

(3.1) Το εκπαιδευτικό εργαλείο Alice

Το Alice είναι ένα προγραμματιστικό περιβάλλον εκπαιδευτικού χαρακτήρα με τρισδιάστατα γραφικά και παρέχεται ελεύθερα στην επίσημη ιστοσελίδα του. Αναπτύχθηκε από το πανεπιστήμιο Carnegie Mellon του Pittsburgh στις αρχές του 1990 με κύριους ερευνητές τους Wanda Dann, Stefan Cooper και Randy Pausch και έχει την υποστήριξη μεγάλων διεθνών εταιρειών όπως της Google, της Oracle και της Sun.

Το Alice προτείνεται σε μαθητές που δεν έχουν προηγούμενη γνώση προγραμματισμού και θέλουν να αποκτήσουν εισαγωγικές γνώσεις σε αυτόν. Ειδικότερα θεωρείται εξαιρετική εναλλακτική των παραδοσιακών προγραμματιστικών γλωσσών και ο λόγος είναι ότι βοηθάει τους μαθητές στην ανάπτυξη αλγοριθμικής σκέψης που παρατηρείται ότι υστερούν οι αρχάριοι προγραμματιστές. Αξίζει να σημειωθεί ότι στο περιβάλλον Alice υπάρχει η δυνατότητα για την ανάπτυξη ενός προγράμματος να χρησιμοποιηθεί η μέθοδος drag and drop ή αλλιώς σύρε και άφησε τις εντολές/εκφράσεις. Ο χρήστης με το ποντίκι του επιλέγει την προγραμματιστική δομή που θέλει και την εντάσσει στον κώδικα του αντί να γράφει τον κώδικα λέξη προς λέξη. Επιπλέον, υπάρχει άμεση αλληλεπίδραση με τον κώδικα που κατασκευάζει καθώς μπορεί να δει οποιαδήποτε στιγμή πως τρέχει το πρόγραμμα. Ως επακόλουθο είναι πιο εύκολη η κατανόηση της σύνδεσης που έχουν τα αντικείμενα με τις δηλώσεις τους στον κώδικα και η διαχείριση των τεχνικών εμποδίων όπως η απομνημόνευση λεξιλογίου και του συντακτικού γλώσσας.

Η χρήση του animation ως εργαλείο για την προσομοίωση της εκτέλεσης ενός προγράμματος δεν είναι πρωτοπόρα ιδέα. Το visualization είναι μια προσέγγιση του προγραμματισμού που εξυπηρετεί το μαθητευόμενο ώστε να μπορεί να γνωρίζει τον τρόπο που λειτουργεί κάθε κομμάτι αλλά και το πως να συνδέσει τα κομμάτια ώστε να επιλύσει ένα πρόβλημα. Για αυτόν τον σκοπό έχει δημιουργηθεί το Alice που περιέχει αντικείμενα (που ανταποκρίνονται στα πρότυπα γλωσσών προγραμματισμού όπως Java, C++, C#) τα οποία μπορούν να προγραμματιστούν και να δημιουργήσουν μια animated ιστορία. Ανάμεσα σε πολλούς ερευνητές που ασχολούνται με το visualization, η άποψη που ξεχώρισε ήταν του ερευνητή Shu(1988). Υποστήριξε ότι για τον προγραμματισμό απαιτούνται και τα δύο μέρη του εγκεφάλου και εστίασε στην ανάγκη της ανάπτυξης του καλλιτεχνικού μέρους του.

Χρειάζεται επίσης να σημειωθεί ότι είναι πιο διαδεδομένο στο εξωτερικό και χρησιμοποιείται κυρίως από ξένους εκπαιδευτικούς. Ωστόσο, το Alice δίνει τη δυνατότητα αλλαγής του περιβάλλοντος του χρήστη στα Ελληνικά, μέσω της επιλογής window, preferences, locale και επιλέγοντας στη συνέχεια την γλώσσα Ελληνικά. Εύκολα λοιπόν, κάθε αρχάριος προγραμματιστής που δεν γνωρίζει την αγγλική γλώσσα μπορεί να το χρησιμοποιήσει.

(3.2) Λογισμικό Alice

Το Alice ως πρόγραμμα πήρε το όνομα του από τον Lewis Carroll ο οποίος ήταν ο δημιουργός του παραμυθιού "Alice in Wonderland". Ο Lewis Carroll υπήρξε μεταξύ άλλων φωτογράφος, ποιητής αλλά και μαθηματικός. Προς τιμήν του συγγραφέα οι δημιουργοί του Alice εμπνεύστηκαν τα εκπαιδευτικά παιχνίδια Alice's Adventure in Wonderland και Through the Looking Glass. Ιδιαίτερα σημαντικό είναι ότι κατά τη διάρκεια ενασχόλησης του με φιλοσοφικά αντικείμενα διαπίστωσε ότι το πιο σπουδαίο από όλα είναι να μπορείς να διατυπώσεις με σαφήνεια αλλά και με ενδιαφέρον τρόπο αυτό που θέλεις. Αυτό ήταν η αφορμή που παρακίνησε την ερευνητική ομάδα του Alice να παρουσιάσει τον προγραμματισμό με εύκολο και διασκεδαστικό τρόπο. Μερικοί ακόμα λόγοι που συνέβαλαν στην επιλογή αυτού του ονόματος, ήταν η εύκολη γραφή, ανάγνωση και προφορά του.

Η βασικότερη έκδοση του Alice είναι η έκδοση 2 η οποία είναι προσβάσιμη σε μεγάλο εύρος ηλικιών και έχει αποδειχθεί εξαιρετική σαν πρώτη γνωριμία με τον προγραμματισμό. Η συγκεκριμένη έκδοση υποστηρίζει μια μεγάλη συλλογή από αντικείμενα αλλά και Garfield χαρακτήρες που έχουν δημιουργηθεί από τη συνεργασία τους με το Αμερικάνικο στούντιο κόμικ Paws Inc.

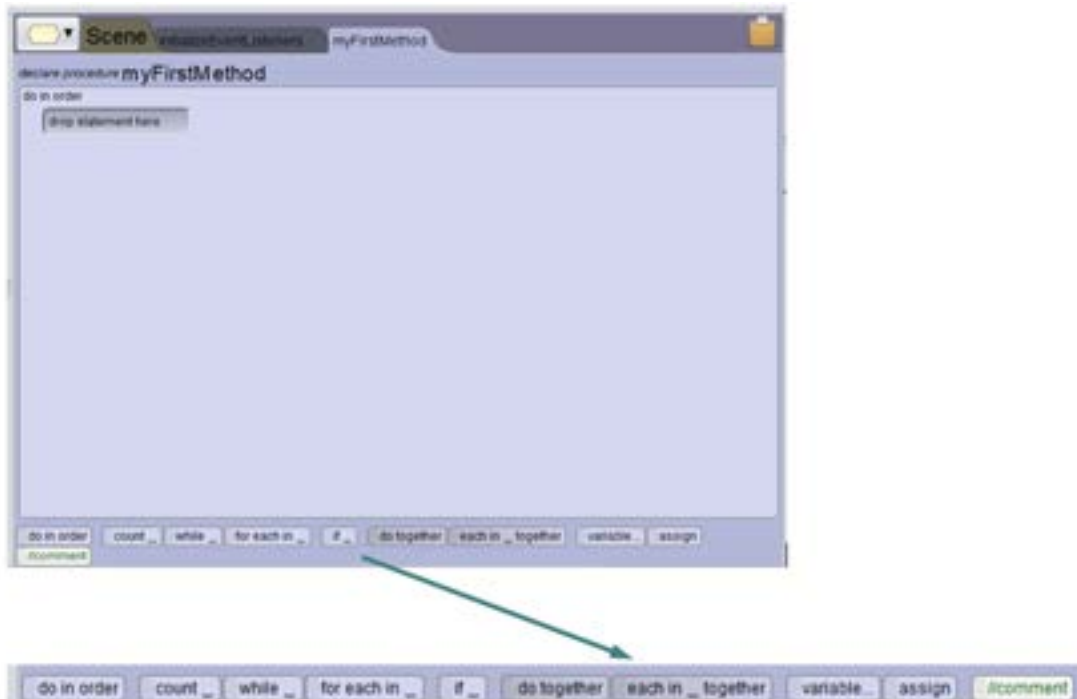
Το Alice 3 αποτελεί την πιο πρόσφατη έκδοση, εμπλουτισμένη με συναρπαστικά και δημιουργικά χαρακτηριστικά που βασίζονται σε αντικειμενοστραφείς έννοιες. Η συλλογή της περιλαμβάνει animated χαρακτήρες και σκηνές του The Sims2 που δώρισε η Electronic Arts(EA). Επιπλέον, το Alice 3 παρέχει τις ίδιες βασικές προγραμματιστικές δυνατότητες που προσφέρει και η Java καθώς και τη δυνατότητα προβολής του κώδικα σε αυτήν.

Υπάρχει ακόμα μια έκδοση, το Alice 3 Netbeans Plugin που απευθύνεται κυρίως στην ανάπτυξη Java εφαρμογών. Η ομάδα του Alice έχει δημιουργήσει μια βιβλιοθήκη που επιτρέπει στον χρήστη να αναπτύξει ή και να επεκτείνει «κόσμους» του Alice απευθείας στο περιβάλλον του NetBeans.

Οι εκδόσεις 2 και 3 του Alice έχουν διαφορετικούς παιδαγωγικούς σκοπούς και απευθύνονται σε διαφορετικά ηλικιακά κοινά. Για κάποιους το Alice 2 θεωρείται μια απλούστερη επιλογή για μάθηση του προγραμματισμού σε μικρότερες ηλικίες, όμως αρκετοί είναι οι εκπαιδευτικοί που επιλέγουν να χρησιμοποιήσουν και τις δύο εκδόσεις ανάλογα με τους στόχους και τις δεξιότητες των μαθητών τους.

(3.3)Ο Προγραμματισμός στο περιβάλλον της Alice

Ανοίγωντας το περιβάλλον Alice εμφανίζεται η περιοχή σύνταξης κώδικα και στο κάτω μέρος βρίσκονται οι προγραμματιστικές δομές που έχουν τη μορφή πλακιδίων.



Οι βασικότερες δομές που θα αναλυθούν παρακάτω είναι οι εξής :do in order, count, while, for each in, if, do together, each in together. Το Alice δίνει τη δυνατότητα στον χρήστη με τη μέθοδο σύρε και άφησε να μπορεί να προσθέσει τις προγραμματιστικές δομές που επιθυμεί στον κώδικα του.

Προγραμματιστική δομή do in order

Είναι από τις κυριότερες δομές στο Alice καθώς όλες οι μέθοδοι της ξεκινάνε εξ ορισμού με αυτήν. Ένα do in order πλακίδιο εκτελεί τις εντολές του κώδικα διαδοχικά, δηλαδή τη μια μετά την άλλη.

Δομή επανάληψης count

Ένα count πλακίδιο θα εκτελέσει τον κώδικα όσες φορές ορίσει ο χρήστης. Είναι μια πολύ εύχρηστη μέθοδος γιατί καθιστά εφικτή την πολλαπλή χρήση μια εντολής χωρίς να γραφτεί επανειλημμένα η ίδια. Για παράδειγμα αν θέλουμε ένας χαρακτήρας να κάνει μια συγκεκριμένη εντολή 15 φορές, το μόνο που χρειάζεται να κάνουμε είναι να βάλουμε σαν παράμετρο στη δομή count τον αριθμό 15.

Δομή επανάληψης while

Η δομή επανάληψης while χρησιμοποιείται στις περιπτώσεις που δεν γνωρίζουμε εξ αρχής τις φορές που θα εκτελεστεί μια εντολή. Ο χρήστης μέσα στην while ορίζει μια συνθήκη και επιλέγει αν θα είναι ψευδής(false) ή αληθής(true) και επαναλαμβάνεται όσο η συνθήκη επαληθεύεται.

Δομή επανάληψης for each in

Η συγκεκριμένη δομή εκτελεί τις εντολές που έχουν οριστεί σε αυτήν για κάθε στοιχείο του πίνακα. Για παράδειγμα αν ο χρήστης έχει δημιουργήσει έναν πίνακα με χαρακτήρες, μπορεί να επαναληφθεί η ίδια εντολή πολλές φορές από διαφορετικό χαρακτήρα κάθε φορά.

Δομή ελέγχου if

Πολλές φορές σε ένα πρόγραμμα θέλουμε να εκτελείται μια εντολή υπό συγκεκριμένες συνθήκες. Στο Alice αυτό εκφράζεται με τη δομή If_else, ειδικότερα ο χρήστης ορίζει μια συνθήκη και αν και μόνο αν είναι αληθής θα εκτελεστούν οι εντολές που ακολουθούν την if. Σε οποιαδήποτε άλλη περίπτωση θα εκτελεστούν οι εντολές που ακολουθούν την else.

Παράλληλος προγραμματισμός Do Together

Ένα do together πλακίδιο δίνει τη δυνατότητα να εκτελεστούν όλες οι εντολές που περιέχονται σε αυτήν ταυτόχρονα. Αυτή η δομή είναι πολύ χρήσιμη σε περιπτώσεις που είτε το αντικείμενο χρειάζεται να κάνει περίπλοκες κινήσεις (αντίστοιχο παράδειγμα θα αναλυθεί σε επόμενο κεφάλαιο), είτε όταν θέλουμε να προγραμματίσουμε πολλούς χαρακτήρες ταυτόχρονα.

Παράλληλος προγραμματισμός each In_together

Το πλακίδιο each in_together είναι μια do together προγραμματιστική δομή που κάνει χρήση έναν πίνακα. Οι εντολές του πλακιδίου πραγματοποιούνται ταυτόχρονα για όλα τα στοιχεία του πίνακα.

ΚΕΦΑΛΑΙΟ 4 Ανάπτυξη εκπαιδευτικού σεναρίου

Στην παρούσα πτυχιακή εργασία δημιουργήθηκαν τέσσερα εκπαιδευτικά σενάρια τα οποία έχουν ως στόχο την εισαγωγή των μαθητών στον αντικειμενοστραφή προγραμματισμό και την εξοικείωσή τους σε εισαγωγικές έννοιες. Τα τέσσερα εκπαιδευτικά σενάρια έχουν ως πρωταρχικό στόχο ο μαθητής να μπορεί να μάθει μόνος του προγραμματισμό με μικρή καθοδήγηση. Ο μαθητής μέσα από τις αλληλεπιδράσεις που έχουν δημιουργηθεί στο περιβάλλον του Alice αλλά και τους ορισμούς και τις οδηγίες που δίνονται αποκτά τις απαραίτητες γνώσεις για τον αντικειμενοστραφή προγραμματισμό.

(4.1) Πρώτο εκπαιδευτικό σενάριο

Το πρώτο εκπαιδευτικό σενάριο αναφέρετε στις μεταβλητές. Οι μεταβλητές παίζουν σημαντικό ρόλο στην γραφή ενός προγράμματος. Αντιπροσωπεύουν αριθμητικές και λογικές τιμές, χαρακτήρες, σειρές χαρακτήρων ή διευθύνσεις μνήμης και έχουν μοναδικό όνομα μέσα στο πρόγραμμα. Μια μεταβλητή συμβολίζει δεδομένα και κάνει τα προγράμματα πιο κατανοητά και ευανάγνωστα.

Γενική περιγραφή περιεχομένου

Στο σενάριο αυτό οι μαθητές εξασκούνται στις μεταβλητές και το διαχωρισμό των τύπων δεδομένων τους. Μέσα από μια σύντομη αφήγηση της θεωρίας, ο μαθητής έρχεται σε μια πρώτη επαφή με τις πρωταρχικές δομές μια μεταβλητής και στην συνέχεια με την ανάθεση τιμής σε αυτήν. Οι μαθητές δεν χρειάζεται να έχουν προαπαιτούμενες γνώσεις καθώς σε κάθε στάδιο του σεναρίου υπάρχουν επεξηγήσεις. Ο ρόλος του εκπαιδευτικού είναι υποβοηθητικός και δίνεται η ευκαιρία ανάλογα με τους εκπαιδευτικούς στόχους να δημιουργήσει ομάδες δύο ή παραπάνω ατόμων.

Διδακτικοί Στόχοι

- Να γνωρίσουν οι μαθητές τι είναι οι μεταβλητές
- Να διακρίνουν τους διάφορους τύπους δεδομένων που έχει μια μεταβλητή
- Να γνωρίζουν τη μορφή αποτίμησης μια τιμής στη μεταβλητή
- Να μπορούν να αναθέτουν τιμές ανάλογα με τον τύπο της μεταβλητής

Λέξεις κλειδιά που χαρακτηρίζουν το σενάριο

- Μεταβλητές
- Βασικοί τύποι
- Προγραμματισμός
- Alice
- Animation
- Αφήγηση

Τυπικός χρόνος αλληλεπίδρασης με το εκπαιδευτικό σενάριο

2 ώρες

Επίπεδο Δυσκολίας

Εύκολο

Επίπεδο Διαδραστικότητας

Μεσαίο

Προτεινόμενη ηλικία

Μαθητές Δημοτικού (9-12)

(4.2) Δεύτερο εκπαιδευτικό σενάριο

Το δεύτερο εκπαιδευτικό σενάριο αναφέρετε στις δομές επανάληψης. Αποτελεί βασικό εργαλείο για τη δημιουργία κώδικα. Σε πολλά προβλήματα που χρειάζεται να επαναληφθεί μια συγκεκριμένη εντολή πολλές φορές, η διαδικασία μπορεί να γίνει πιο απλή με την χρήση δομών επανάληψης.

Γενική περιγραφή περιεχομένου

Στο σενάριο αυτό ο μαθητής εξασκείται στις δομές επανάληψης. Αρχικά ασχολείται με τη δομή επανάληψης for και μαθαίνει τι χρειάζεται για την συγγραφή της. Στην συνέχεια γνωρίζει τη δομή επανάληψης while και μαθαίνει τα βασικά για αυτήν. Είναι σημαντικό να τονιστεί ότι ο μαθητής εξοικειώνεται με την σύνταξη αυτών των δύο δομών στην γλώσσα Java και εφαρμόζει στην πράξη αυτά που μαθαίνει.

Διδακτικοί Στόχοι

- Να γνωρίσουν οι μαθητές τις δομές επανάληψης
- Να μάθουν τη δομή και την συντάξη της δομής επανάληψης for
- Να μάθουν τη δομή και την συντάξη της δομής επανάληψης while
- Να μπορούν να κατανοήσουν και να ολοκληρώσουν την εκπαιδευτική δραστηριότητα του σεναρίου

Λέξεις κλειδιά που χαρακτηρίζουν το σενάριο

- Δομές επανάληψης
- For
- While
- Αντικειμενοστρεφής προγραμματισμός
- Κίνηση
- Alice
- Animation
- Αφήγηση

Τυπικός χρόνος αλληλεπίδρασης με το εκπαιδευτικό σενάριο

2-3 ώρες

Επίπεδο Δυσκολίας

Μεσαίο

Επίπεδο Διαδραστικότητας

Υψηλό

Προτεινόμενη ηλικία
Μαθητές Δημοτικού (9-12)

(4.3) Τρίτο εκπαιδευτικό σενάριο

Το τρίτο εκπαιδευτικό σενάριο αναφέρετε στους τελεστές και τις δομές επιλογής και έχει ως σκοπό ο μαθητής να κατανοήσει την έννοια των τελεστών (αριθμητικών, συγκριτικών και λογικών) καθώς και τις δομές ελέγχου. Οι δομές επιλογής αποτελούνται από μια συνθήκη και δύο ομάδες εντολών οι οποίες ανάλογα με την συνθήκη και τότε αυτή ικανοποιείται, εκτελείται είτε η πρώτη είτε η δεύτερη ομάδα. Οι τελεστές περιέχονται συνήθως στην συνθήκη που υπάρχει στις δομές επιλογής και για αυτόν τον λόγο προέκυψε και ο συνδυασμός αυτών των δύο εννοιών σε αυτό το διδακτικό σενάριο.

Γενική περιγραφή περιεχομένου

Στο σενάριο αυτό ο μαθητής εξασκείται στους τελεστές και στις δομές επιλογής. Αρχικά μέσα από μια σύντομη περιγραφή της θεωρίας μαθαίνει τα τρία είδη τελεστών. Στην συνέχεια γνωρίζει τη δομή επιλογής if και μαθαίνει τα βασικά για την σύνταξη και την χρήση της. Έπειτα αφού ολοκληρώσει με την if, συνεχίζει με τη δομή επιλογής switch και μελετάει τα αντίστοιχα πεδία και για αυτήν. Είναι σημαντικό να τονιστεί ότι ο μαθητής εξοικειώνεται με την σύνταξη αυτών των δύο δομών στην γλώσσα Java και εφαρμόζει πρακτικά αυτά που μαθαίνει.

Διδακτικοί Στόχοι

- Να γνωρίσουν οι μαθητές τους τελεστές και τις δομές επιλογής
- Να μάθουν τη δομή και την σύνταξη της δομής επιλογής if
- Να μάθουν τη δομή και την σύνταξη της δομής επιλογής switch
- Να μπορούν να κατανοήσουν και να ολοκληρώσουν τις εκπαιδευτικές δραστηριότητες του σεναρίου

Λέξεις κλειδιά που χαρακτηρίζουν το σενάριο

- Τελεστές
- Δομές επιλογής
- If
- Swich
- Συνθήκη ελέγχου
- Αντικειμενοστραφής προγραμματισμός
- Alice
- Animation
- Αφήγηση

Τυπικός χρόνος αλληλεπίδρασης με το εκπαιδευτικό σενάριο

2-3 ώρες

Επίπεδο Δυσκολίας

Μεσαίο

Επίπεδο Διαδραστικότητας
Υψηλό

Προτεινόμενη ηλικία
Μαθητές Δημοτικού (9-12)

(4.4) Τέταρτο εκπαιδευτικό σενάριο

Το τέταρτο εκπαιδευτικό σενάριο αναφέρετε στις μεθόδους σε γλώσσα Java. Οι μέθοδοι ή αλλιώς γνωστές και ως συναρτήσεις αποτελούν σπουδαίο κομμάτι του προγραμματισμού. Τα βασικά χαρακτηριστικά τους είναι ότι επιτρέπουν την επαναχρησιμοποίηση ενός κομματιού κώδικα όσες φορές θελήσει ο προγραμματιστής και κάνουν τον κώδικα πιο ευανάγνωστο.

Γενική περιγραφή περιεχομένου

Στο σενάριο αυτό ο μαθητής εξασκείται στις μεθόδους. Αρχικά γνωρίζει τα βασικά για τη μορφή που έχει μια μέθοδος και έπειτα τι χρειάζεται για την συγγραφή της. Αφότου αποκτήσει μια εξοικείωση με τη δομή της, ο μαθητής καλείτε να δημιουργήσει μια μέθοδο από την αρχή μόνος του.

Διδακτικοί Στόχοι

- Να γνωρίσουν οι μαθητές τις μεθόδους
- Να μάθουν τη δομή και την σύνταξη μιας μεθόδου
- Να μπορούν να γράψουν και να καλέσουν μια μέθοδο
- Να μπορούν να ολοκληρώσουν την εκπαιδευτική δραστηριότητα του σεναρίου

Λέξεις κλειδιά που χαρακτηρίζουν το σενάριο

- Μέθοδος
- Κλήση μεθόδου
- Αντικειμενοστραφής προγραμματισμός
- Alice
- Animation
- Αφήγηση

Τυπικός χρόνος αλληλεπίδρασης με το εκπαιδευτικό σενάριο
3 ώρες

Επίπεδο Δυσκολίας
Δύσκολο

Επίπεδο Διαδραστικότητας
Υψηλό

Προτεινόμενη ηλικία
Μαθητές Δημοτικού (9-12)

ΚΕΦΑΛΑΙΟ 5 Παρουσίαση της εφαρμογής

(5.1) Σχεδιασμός και δημιουργία ενός παιχνιδιού στο Alice

“Ο κόσμος του κώδικα” είναι ένα παιχνίδι που δημιουργήθηκε στο προγραμματιστικό περιβάλλον Alice χρησιμοποιώντας την πιο πρόσφατη έκδοση Alice 3.6. Η ιδέα του παιχνιδιού προήλθε από την ανάγκη να γίνει πιο ελκυστική η εκμάθηση του προγραμματισμού. Κύριος στόχος της ανάπτυξης του ήταν η γνωριμία και εισαγωγή των μαθητών του δημοτικού στον αντικειμενοστραφή προγραμματισμό.

Για την σχεδίαση και την υλοποίηση ενός προγράμματος στο περιβάλλον Alice απαιτούνται δύο στάδια. Το πρώτο στάδιο είναι η κατασκευή ενός φανταστικού τρισδιάστατου κόσμου, χρησιμοποιώντας τα αντικείμενα και τους χαρακτήρες που δίνονται στο Alice. Στη συνέχεια το δεύτερο στάδιο, αφού έχουν επιλεγθεί τα αντικείμενα και οι χαρακτήρες γίνεται ο προγραμματισμός της συμπεριφοράς τους. Αξίζει να σημειωθεί ότι οποιαδήποτε στιγμή μπορεί ο χρήστης να προσθέσει αντικείμενα και χαρακτήρες σε όποιο σημείο του κόσμου θέλει.

Η σύνταξη του κώδικα γίνεται στον χώρο συντάκτη κώδικα(editor) που εμφανίζεται με την έναρξη του προγράμματος Alice. Ο χρήστης εφόσον επιλέξει το αντικείμενο που θέλει να προγραμματίσει επιλέγει τις εντολές και τις μεθόδους που εμφανίζονται σε μορφή πλακιδίων. Η διαδικασία επιλογής των πλακιδίων γίνεται με την χρήση του ποντικιού και τη μέθοδο “σύρε και άφησε” ή αλλιώς “drag and drop”. Σε πολλές περιπτώσεις κατά την συγγραφή ενός προγράμματος προκύπτουν συντακτικά λάθη. Το Alice παρέχοντας αυτήν τη μέθοδο ελαχιστοποιεί την πιθανότητα λάθους και στην περίπτωση εμφάνισης του η διόρθωση του από την χρήστη είναι πιο εύκολη. Τέλος, κρίνεται σημαντικό να τονιστεί ότι μπορεί να εμφανιστεί η εκτέλεση του προγράμματος που έχει κατασκευάσει σε ένα παράθυρο που απεικονίζει σε τρισδιάστατη κινούμενη μορφή τον κόσμο.

(5.2) Έναρξη παιχνιδιού

Για την υλοποίηση του παιχνιδιού χρησιμοποιήθηκαν πολλά από τα χαρακτηριστικά που προσφέρει το πρόγραμμα Alice τόσο για την δημιουργία των κόσμων σε τρισδιάστατη απεικόνιση όσο και σε προγραμματιστικό επίπεδο. Η συγκεκριμένη εφαρμογή αποτελείται από το κύριο μενού που μεταφέρει τον μαθητή κάθε φορά σε έναν από τους τέσσερις κόσμους που έχουμε δημιουργήσει. Για τον κάθε κόσμο έχουμε χρησιμοποιήσει διαφορετικά αντικείμενα και χαρακτήρες κάνοντας έτσι το κάθε εκπαιδευτικό σενάριο μια μοναδική εμπειρία για αυτόν.

Το παιχνίδι αυτό περιέχει συνολικά 8 χαρακτήρες με κύριους και βασικότερους μια πριγκίπισσα, μια μαϊμού και έναν κάστορα. Επίσης έχει γίνει χρήση πολλών σκηνικών αντικειμένων αλλά και σχημάτων στους κόσμους που στήσαμε. Δόθηκε ιδιαίτερη σημασία στην ομαλή εναλλαγή των σκηνών και των σταδίων που έχει το κάθε σενάριο. Για την ακρίβεια τοποθετήθηκαν συνολικά 41 κάμερες σε διάφορα σημεία των κόσμων κάποιες από αυτές για να μετακινούνται μαζί με την κίνηση των χαρακτήρων και άλλες για την εστίαση σε συγκεκριμένα αντικείμενα.

Σε προγραμματιστικό επίπεδο ο κύριος στόχος μας ήταν να δώσουμε την αίσθηση ενός “πραγματικού” παιχνιδιού. Για αυτόν τον λόγο σε κάθε σενάριο προσπαθήσαμε να δημιουργήσουμε πέρα από εκπαιδευτικούς στόχους, δραστηριότητες που επικεντρώνονται στην αλληλεπίδραση του μαθητή με τους κόσμους. Αξίζει να επισημανθεί ότι με την έναρξη της εφαρμογής έχουμε εισάγει μουσική αλλά και ήχους σε διάφορες περιστάσεις μέσα στα σενάρια. Επιπλέον για την ομιλία των χαρακτήρων έχει γίνει χρήση τόσο της μεθόδου της αφήγησης όσο και της εισαγωγής ήχων(audio) για την ακρόαση τους. Έτσι ο μαθητής βλέπει και ακούει τους χαρακτήρες των σεναρίων κάνοντας όλη τη διαδικασία πιο ελκυστική και ενδιαφέρουσα.

Ανοίγοντας το πρόγραμμα εμφανίζονται αρχικά στην οθόνη κάποιες πληροφορίες για αυτό.



Στη συνέχεια με τη βοήθεια της κάμερας μεταφερόμαστε στο μενού επιλογής του παιχνιδιού. Μέσα από το μενού επιλογής δίνεται η δυνατότητα στο μαθητή να επιλέξει ένα από τα τέσσερα κουτάκια ανάλογα τους εκπαιδευτικούς στόχους που θέλει να ασχοληθεί και να καλύψει.



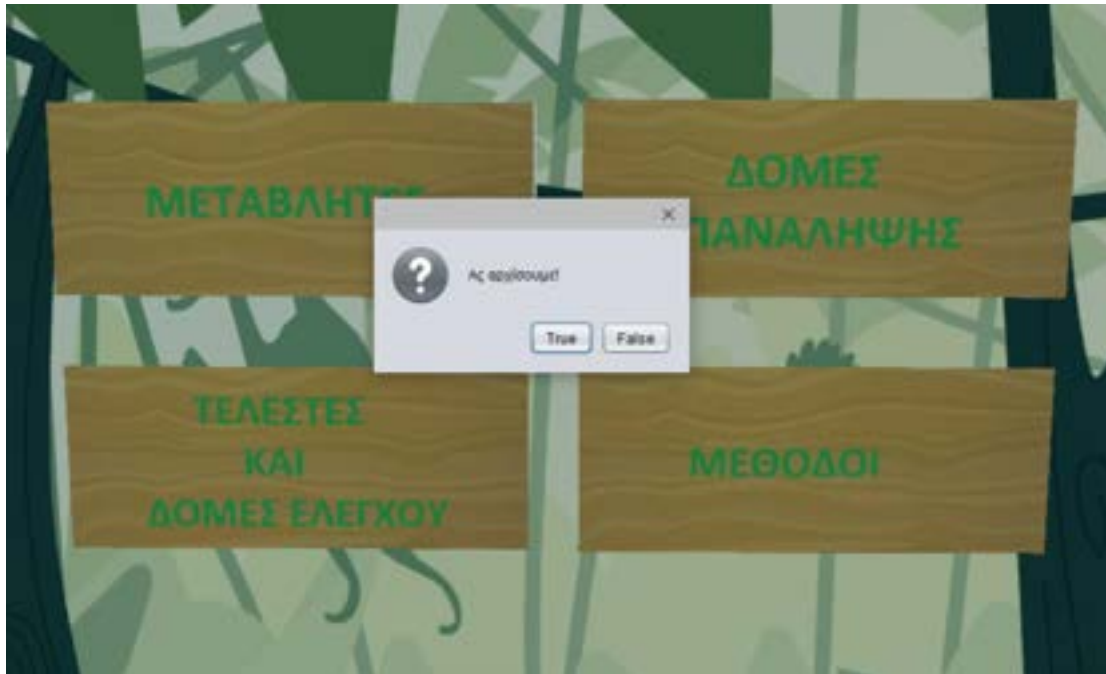
Για την υλοποίηση του μενού χρησιμοποιήθηκε κυρίως το Event Listener

```
this.addMouseClickedEventListener, multipleEventPolicy COMBINE add detail
```

Το `addMouseClickedEventListener` ενεργοποιείται κάθε φορά που ο χρήστης επιλέγει με το ποντίκι του ένα από τα αντικείμενα που ορίστηκαν μέσα σε αυτό.



Κάθε φορά που επιλέγεται ένας εκπαιδευτικός σκοπός εμφανίζεται ένα αναδυόμενο παράθυρο με το μήνυμα «Ας αρχίσουμε». Σε αυτό το σημείο ο μαθητής μπορεί να επιλέξει true στην περίπτωση που θέλει να συνεχίσει αλλιώς false για να παραμείνει στο μενού επιλογής.



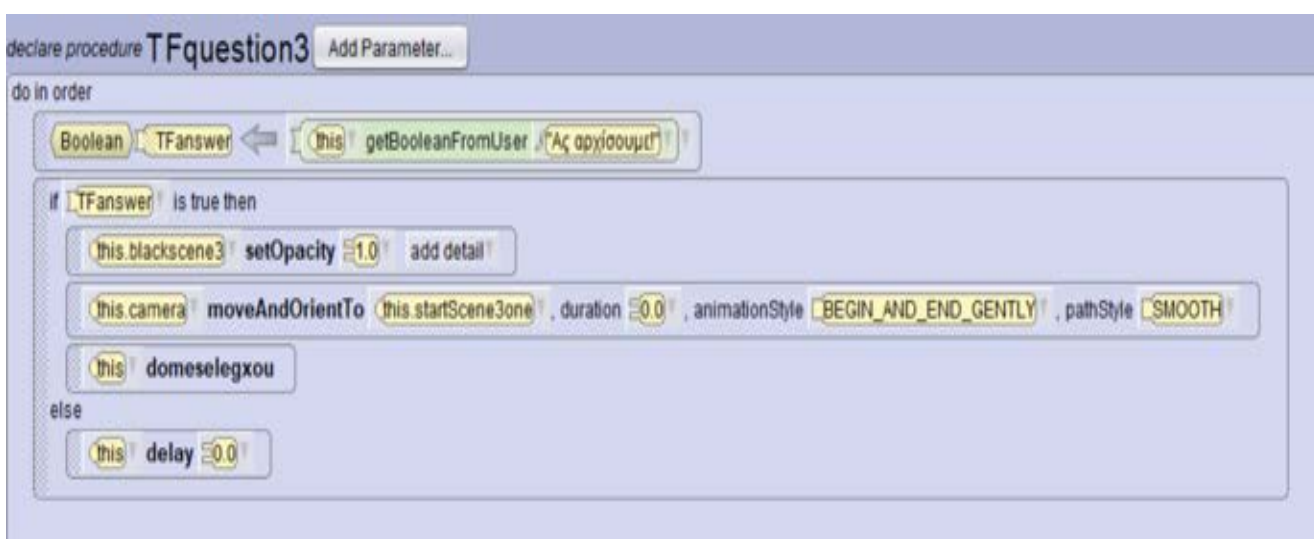
Για την πραγματοποίηση του παραπάνω, δημιουργήθηκαν τέσσερις συναρτήσεις οι οποίες όπως περιγράφηκε πιο πάνω ανάλογα με την απάντηση που δίνει ο χρήστης κάθε φορά τον οδηγούν είτε στο αντίστοιχο εκπαιδευτικό σενάριο που επιλέγει είτε παραμένοντας στο αρχικό μενού. Αναλυτικά οι συναρτήσεις ήταν οι εξής:



Εικόνα 1: Για το πρώτο εκπαιδευτικό σενάριο



Εικόνα 2: Για το δεύτερο εκπαιδευτικό σενάριο



Εικόνα 3: Για το τρίτο εκπαιδευτικό σενάριο



Εικόνα 4:Για το τέταρτο εκπαιδευτικό σενάριο

(5.3)Πρώτο εκπαιδευτικό σενάριο-Μεταβλητές

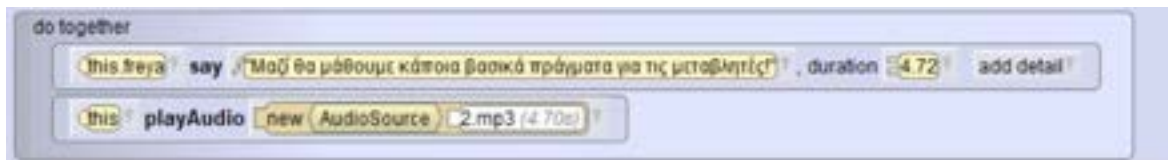
Στόχος του πρώτου εκπαιδευτικού σεναρίου είναι ο μαθητής να γνωρίσει τις μεταβλητές και την χρησιμότητά τους στον προγραμματισμό. Πρωταγωνίστρια του σεναρίου είναι μια πριγκίπισσα που με την τεχνική της αφήγησης κάνει μια σύντομη παρουσίαση της θεωρίας των μεταβλητών. Στην συνέχεια μέσω ενός παιχνιδιού ο μαθητής ελέγχει τις γνώσεις του στα βασικά σημεία της θεωρίας.



Στο κομμάτι που σχετίζεται με την αφήγηση χρησιμοποιήθηκαν κυρίως οι παρακάτω δυο εντολές για να μπορούμε παράλληλα να διαβάζουμε αλλά και να ακούμε τον χαρακτήρα.



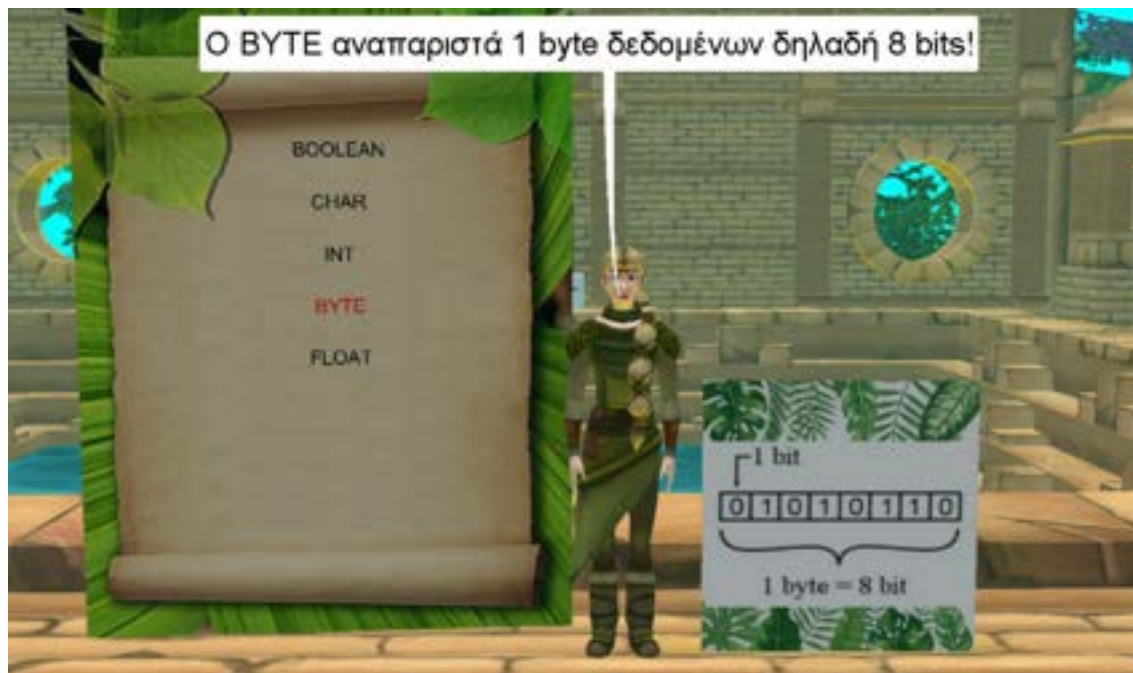
Για να επιτευχθεί επίσης το παραπάνω, έπρεπε οι δυο αυτές εντολές (say,play audio) να ενταχθούν μέσα σε μια δομή με τη μορφή πλακιδίου **do together**, ώστε να εκτελούνται ταυτόχρονα και να έχουμε συγχρονισμό εικόνας και ήχου.



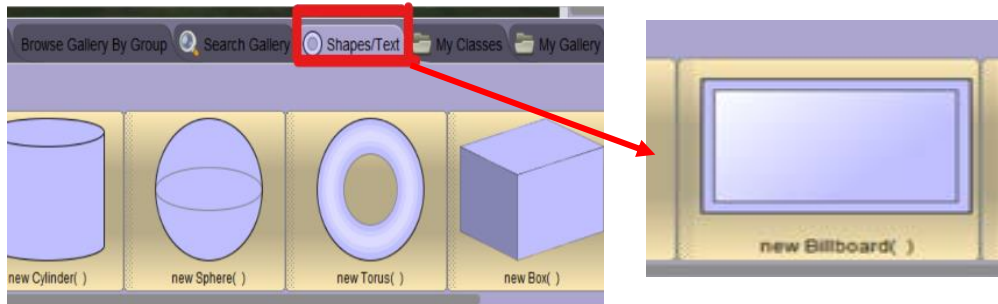
Αξίζει να σημειωθεί ότι εξαιρετικά χρήσιμη ήταν και η χρήση της δομής **do in order**, καθώς εισάγοντας μέσα σε αυτήν εμφωλευμένα πλακίδια **do together** καθορίζεται η χρονική αλληλουχία που έχει η αφήγηση.



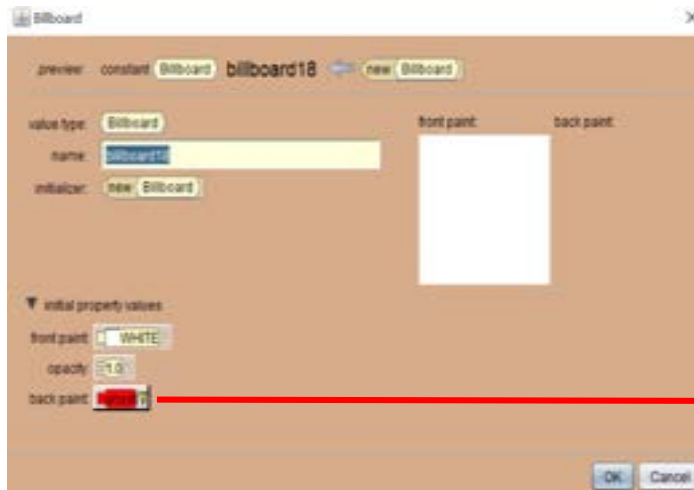
Η παρουσίαση της θεωρίας γίνεται με συνδυασμό της αφήγησης που αναλύσαμε πιο πάνω, αλλά και με ταυτόχρονη προβολή πινάκων που παρουσιάζουν συνοπτικά τα κύρια στοιχεία της θεωρίας.



Αυτοί οι πίνακες αποτελούν αντικείμενα στο περιβάλλον του Alice και η εισαγωγή τους γίνεται από την κατηγορία **Shapes/Text** και μετά επιλέγοντας το shape billboard.

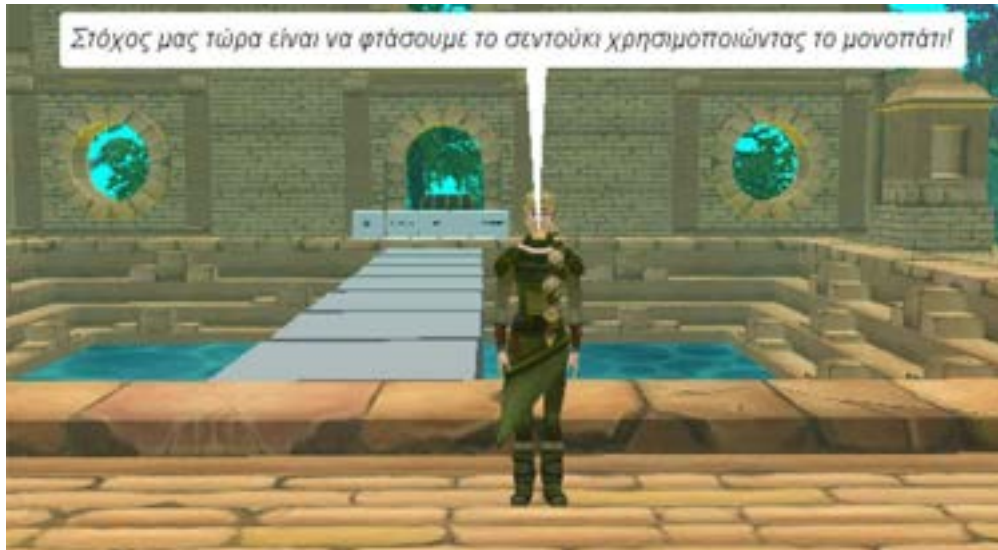


Κρίνεται σημαντικό να αναφερθεί ότι για την μορφοποίηση του billboard, υπάρχει ένα αναδυόμενο παράθυρο που εμφανίζεται μόλις ο χρήστης επιλέξει το billboard για να το εισάγει. Από αυτό ρυθμίζεις την εμφάνιση που θα έχει ο πίνακας. Υπάρχει η επιλογή τόσο της εισαγωγής κειμένου μέσα σε αυτόν όσο και η δυνατότητα προσθήκης μιας εικόνας της επιλογής του χρήστη.



Επιλέγοντας το βέλος, εμφανίζεται ένα dropdown μενού, όπου μπορεί ο χρήστης να εισάγει αρχεία της επιλογής του.

Μετά την ολοκλήρωση της θεωρίας, ξεκινάει η εκπαιδευτική δραστηριότητα η οποία αποτελεί το πιο διαδραστικό κομμάτι του σεναρίου και είναι απαραίτητη η συμμετοχή του μαθητή. Περιληπτικά το παιχνίδι βρίσκει την πρωταγωνίστρια μας να θέλει να περάσει στην απέναντι πλευρά του κάστρου ώστε να φτάσει στον θησαυρό και να κατακτήσει το διαμάντι.



Για να το καταφέρει αυτό ο μαθητής πρέπει να “βάλει” σε σωστή σειρά τα κουτιά ώστε να σχηματιστεί το μονοπάτι. Για να πραγματοποιηθεί ο στόχος αυτός πρέπει ο μαθητής να απαντήσει σωστά σε οκτώ ερωτήσεις θεωρίας σχετικά με τις μεταβλητές. Κάθε ερώτηση αντιστοιχεί και σε ένα κομμάτι από το μονοπάτι.



Για την υλοποίηση του παιχνιδιού σε προγραμματιστικό επίπεδο και πιο συγκεκριμένα για την μετακίνηση των κουτιών έγινε χρήση της εντολής **move**, καθορίζοντας απλά την κατεύθυνση και την απόσταση που θέλουμε να κουνηθεί το αντικείμενο. Επίσης παρέχεται η δυνατότητα να καθοριστούν και άλλες λεπτομέρειες όπως η διάρκεια που θα έχει η εντολή. Πιο αναλυτικά μέσα σε μια μέθοδο **do together** προστέθηκαν επτά εντολές **move** σε αντιστοιχία με τα επτά κουτάκια του παιχνιδιού. Επιπρόσθετα χρησιμοποιήθηκε και μια επιπλέον εντολή το **playAudio** με την οποία προσθέσαμε μουσική υπόκρουση. Αξίζει να παρατηρηθεί, ότι για να επιτευχθεί η τοποθέτηση των κουτιών όπως απεικονίζονται στην παραπάνω εικόνα, προγραμματίσαμε το κάθε ένα να κινείται σε αντίθετες κατευθύνσεις.



Εξίσου σημαντική προσθήκη στον κώδικα που αποτέλεσε σπουδαίο ρολό στην πραγματοποίηση του παιχνιδιού ήταν η κάμερα, που ήταν υπεύθυνη σχετικά με την οπτική που θα έχουμε στον “κόσμο” και ακολουθούσε την πριγκίπισσα κατά τη διάρκεια του παιχνιδιού. Με την εντολή **moveAndOrientTo** ορίζουμε τον στόχο και την κατεύθυνση που θέλουμε να έχει η κάμερα μας, την χρονική διάρκεια, το πόσο απότομη θα είναι η κίνηση της κλπ.



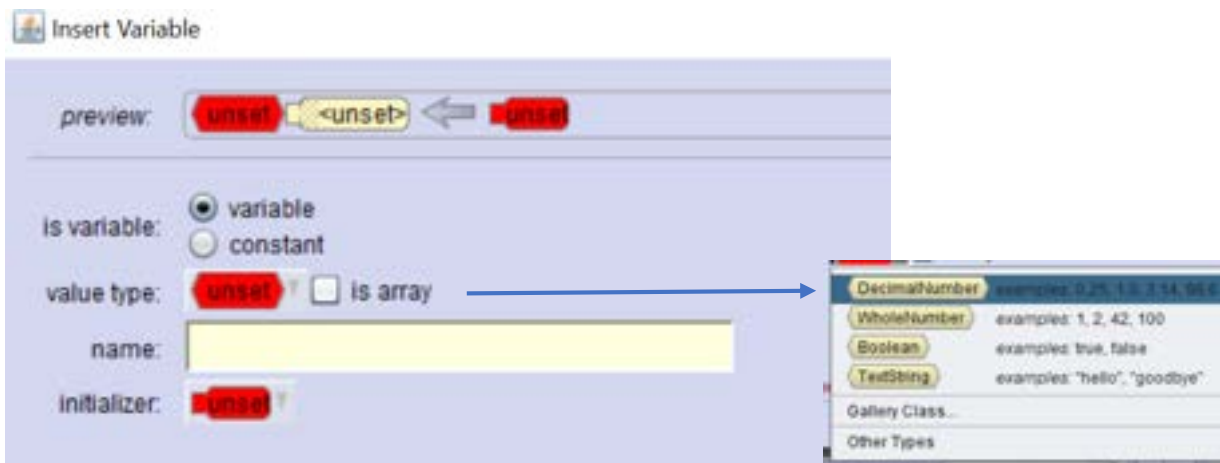
Στην προκειμένη περίπτωση ορίσαμε σαν βασική την κάμερα του κύριου χαρακτήρα δίνοντας έτσι την αίσθηση στον χρήστη ότι κινείται και ακολουθεί την πριγκίπισσα σε κάθε βήμα της καθόλη τη διάρκεια του παιχνιδιού.



Στην συνέχεια ακολουθεί ο προγραμματισμός του κάθε κουτιού. Η λογική που ακολουθήθηκε για τον προγραμματισμό τους είναι παρόμοια για όλα. Ένα παράδειγμα για την σύνταξη του κώδικα του κουτιού είναι το ακόλουθο.



Αναλυτικότερα για την σύνταξη του κώδικα, αρχικά ορίσαμε μια μεταβλητή τύπου **Boolean** στο προγραμματιστικό περιβάλλον Alice. Για να εισάγουμε μια μεταβλητή επιλέγουμε από το μενού το πεδίο variables όπου εμφανίζεται το παρακάτω μενού.



Στο πεδίο value type επιλέγουμε το είδος της μεταβλητής. Στο παρών παράδειγμα επιλέχθηκε ο τύπος μεταβλητής Boolean και στην συνέχεια στο πεδίο initializer ορίστηκε αν είναι true ή false.



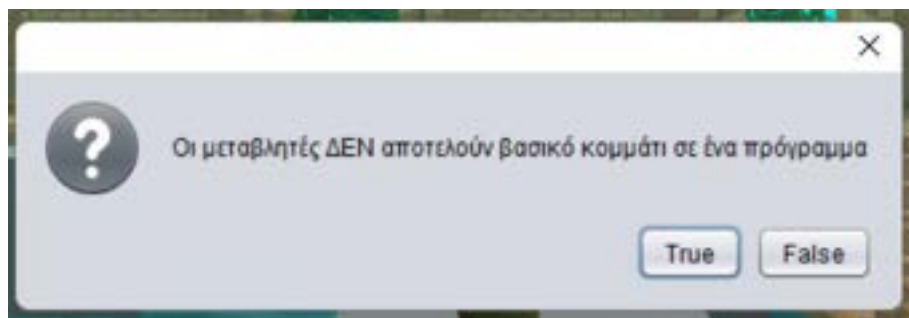
Παρατηρούμε ότι στο δεύτερο μέρος έχουμε τοποθετήσει την εντολή **getBooleanFromUser**. Αυτό σημαίνει ότι η μεταβλητή που έχουμε ορίσει λαμβάνει την τιμή (true ή false) από τον χρήστη, την οποία επιλέγει κατά την εκτέλεση του εκπαιδευτικού σεναρίου.



Στην συνέχεια προσθέτουμε μια δομή **while**. Η κεντρική ιδέα είναι να υπάρχει μια συνθήκη που θα ελέγχει την τιμή που δίνεται στη μεταβλητή. Στην περίπτωση μας απάντηση στο ερώτημα θεωρίας που δίνεται στον χρήστη αποτελεί η συνθήκη.



Το αποτέλεσμα της εκτέλεσης του κώδικα εμφανίζει ένα αναδυόμενο παράθυρο με την έρωτηση που έχουμε ορίσει.



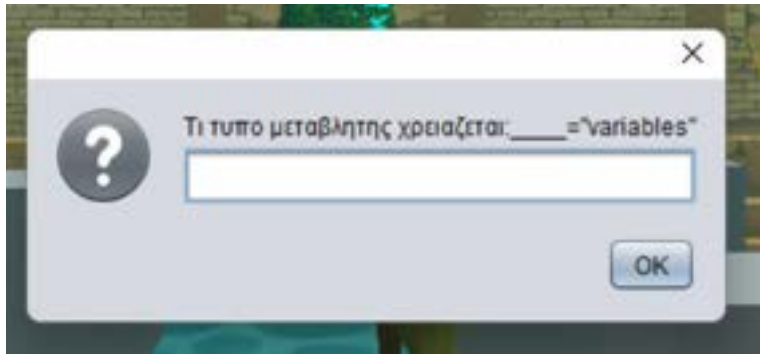
Όσο η απάντηση του είναι λάθος, εμφανίζεται στην οθόνη του χρήστη ένα μήνυμα που τον ενημερώνει να προσπαθήσει ξανά και του εμφανίζεται πάλι η ερώτηση. Μόλις ο χρήστης απαντήσει σωστά και "βγει" από τη δομή **while**, ακολουθεί ένα **do in order** που καθορίζει ποιες εντολές εκτελούνται σειριακά. Παρατηρούμε να επαναλαμβάνεται η εξής λογική: οι εντολές που θέλουμε να εκτελεστούν ταυτόχρονα τοποθετούνται μέσα σε μια δομή **do together**. Ένα παράδειγμα είναι το κείμενο και ο ήχος να μπαίνουν μαζί με σκοπό να δίνεται η αίσθηση στον χρήστη ότι ο χαρακτήρας μιλάει και ένα άλλο είναι οι εντολές που αφορούν τη μετακίνηση του κουτιού.

Παρόμοια λογική ακολουθείται και στον προγραμματισμό και των υπολοίπων κουτιών, με την μοναδική διαφορά ότι αλλάζει ο τύπος της μεταβλητής.

Στο συγκεκριμένο εκπαιδευτικό σενάριο χρησιμοποιούμε και μεταβλητές τύπου **textString**.

```
TextString erwthsh2 ← this.camera.getStringFromUser "Τι τυπο μεταβλητης χρειαζεται ____ =\variables"
```

Όπου με την εντολή **getStringFromUser**, ο χρήστης πληκτρολογεί τη σωστή απάντηση στο αναδυόμενο παράθυρο που του εμφανίζεται.



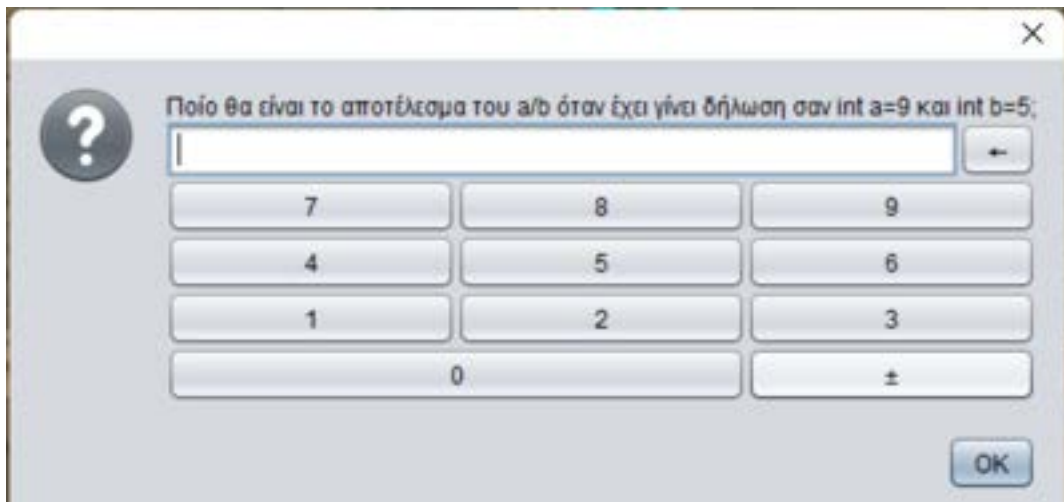
Έπειτα ο έλεγχος της απάντησης γίνεται πάλι σε μια δομή **while**.

```
while EITHER erwthsh2 contentEquals "char" OR erwthsh2 startsWith "CHAR" is true
```

Ένα επιπλέον παράδειγμα είναι το ακόλουθο:

```
WholeNumber erwthsh4 ← this.getIntegerFromUser "Ποιο θα είναι το αποτέλεσμα του a/b όταν έχω γίνει δήλωση σαν int a=9 και int b=5;"
```

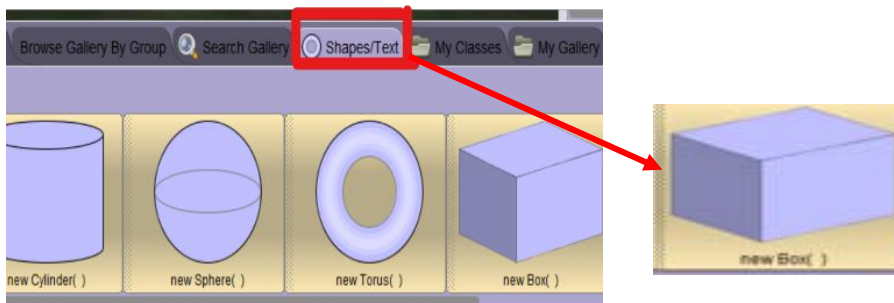
Το οποίο με την εντολή **getIntegerFromUser**, παίρνει έναν αριθμό που εισάγει ο χρήστης από το πληκτρολόγιο του.



Αφού ολοκληρωθούν και οι επτά ερωτήσεις, ζητείτε από τον χρήστη η ολοκλήρωση ακόμα μιας δραστηριότητας η οποία είναι η τελευταία σε αυτό το εκπαιδευτικό σενάριο. Εμφανίζονται 5 κουτιά που το κάθε ένα αναγράφει πάνω του ένα σύμβολο ή λέξη που αν τα βάλει ο χρήστης στη σωστή σειρά θα σχηματιστεί ο ορισμός της ανάθεσης μεταβλητής. Αυτό πραγματοποιείται αν ο χρήστης επιλέξει το κάθε κουτί και το "σύρει" με το ποντίκι του στη σωστή θέση.



Η τοποθέτηση των κουτιών γίνεται από την κατηγορία **Shapes/Text** και μετά επιλέγοντας το box.



Μετά την ολοκλήρωση και της τελευταίας δραστηριότητας, η πριγκίπισσα έχει καταφέρει να περάσει στο κάστρο και να φτάσει στο θησαυρό κατακτώντας έτσι το διαμάντι.

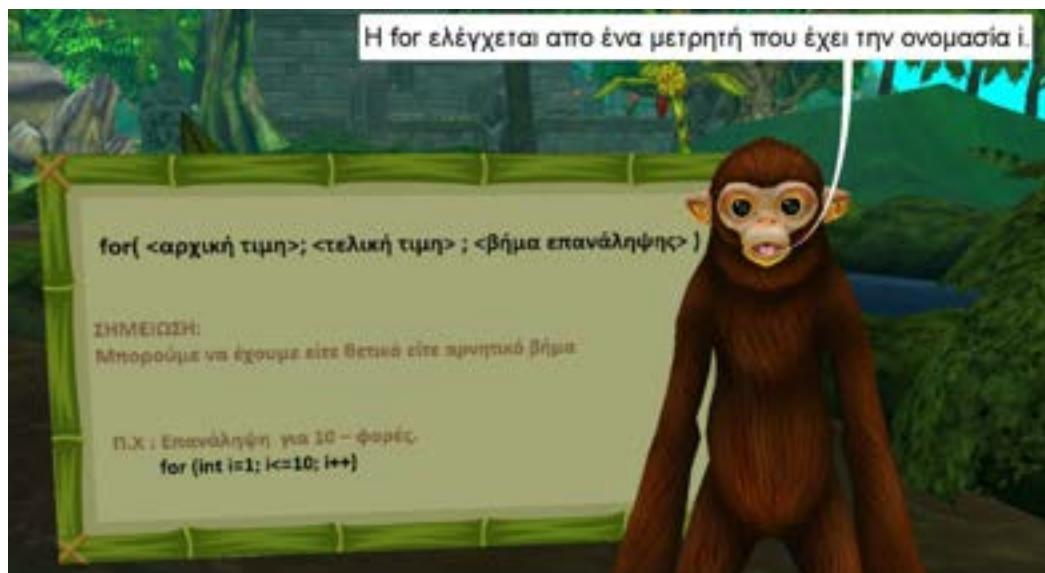


(5.4) Δεύτερο εκπαιδευτικό σενάριο-Δομές Επανάληψης

Στόχος του σεναρίου είναι ο μαθητής να μάθει κάποια βασικά πράγματα για τις δομές επανάληψης. Το δεύτερο εκπαιδευτικό σενάριο ξεκινάει με μια πρώτη γνωριμία του κύριου χαρακτήρα τον Μαρσελ όπου θα είναι και ο βοηθός του μαθητή στο ταξίδι της γνώσης του.

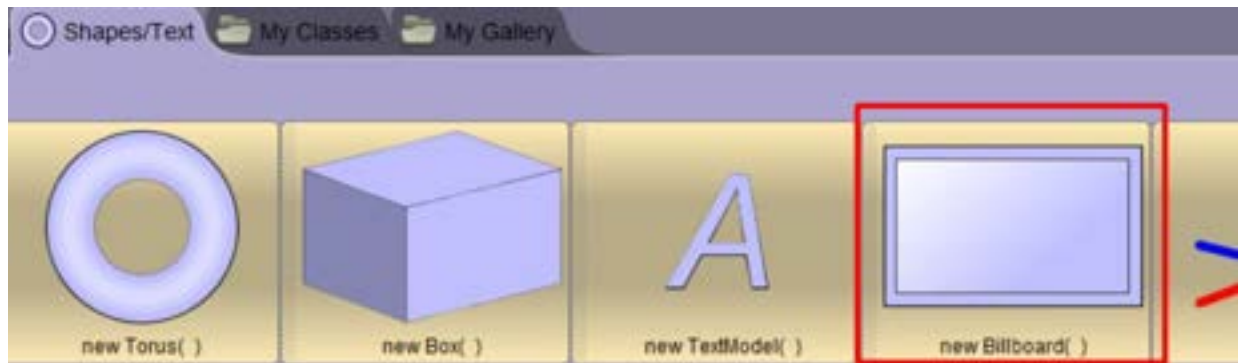


Στο πρώτο στάδιο του σεναρίου ο μαθητής βλέπει και ακούει τον χαρακτήρα Μαρσελ να εξηγεί πως δομείται αλλά και πως λειτουργεί μια δομή επανάληψης **for** και **while**.





Για την επίτευξη αυτού του σκοπού τοποθετήθηκαν αντικείμενα σε κατάλληλα σημεία στο περιβάλλον του Alice για την εμφάνιση πινάκων.



Η εισαγωγή εικόνων στα σχήματα πραγματοποιήθηκε με τις ιδιότητες



που παρέχει το περιβάλλον Alice για την επεξεργασία τους.

Επιπλέον εφαρμόστηκαν μέσα σε έναν παράλληλο προγραμματισμό **do together** οι μέθοδοι



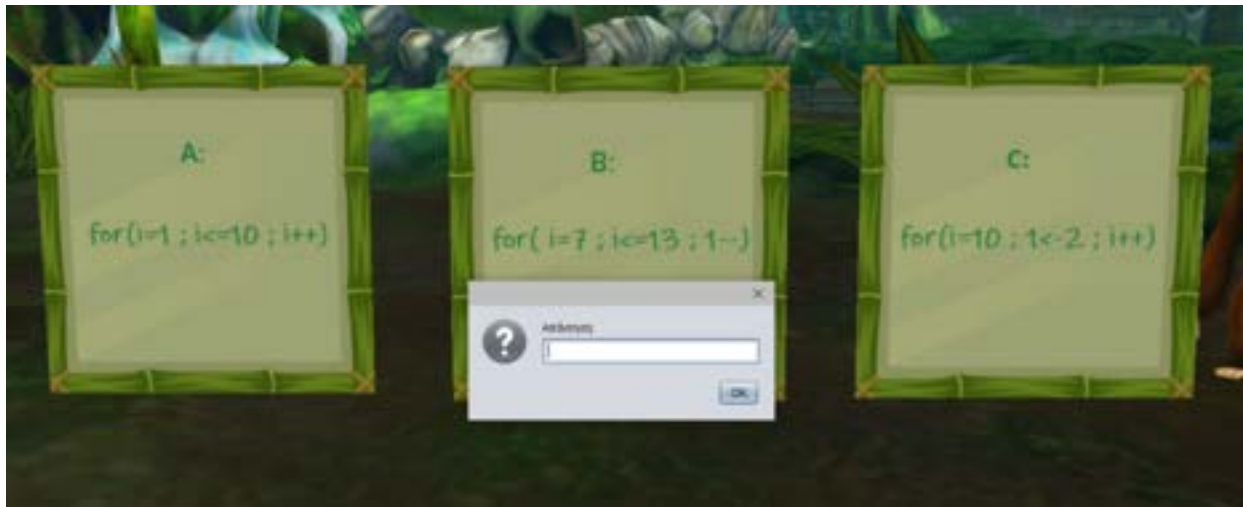
για την εμφάνιση και την ακρόαση ομιλίας του χαρακτήρα μας.



Χρειάζεται επίσης να τονιστεί ότι τοποθετήθηκαν σε πολλά σημεία κάμερες για την ομαλότερη συνοχή των σταδίων αλλά και για να γίνει πιο ελκυστική η εμπειρία του εκπαιδευτικού σεναρίου στο μαθητή. Ένα παράδειγμα κάμερας και χρήσης μιας μεθόδους της είναι το παρακάτω:



Στην συνέχεια αφότου ο μαθητής γνωρίσει τις δομές επανάληψης, εμφανίζονται στην οθόνη του τρεις πίνακες με διαφορετικά παραδείγματα για τη δομή **for**. Σε αυτό το στάδιο του εκπαιδευτικού σεναρίου ο μαθητής καλείται με τις γνώσεις που απέκτησε για τις δομές επανάληψης, να σκεφτεί και να πληκτρολογήσει τη σωστή απάντηση στο αναδυόμενο παράθυρο που του εμφανίστηκε.

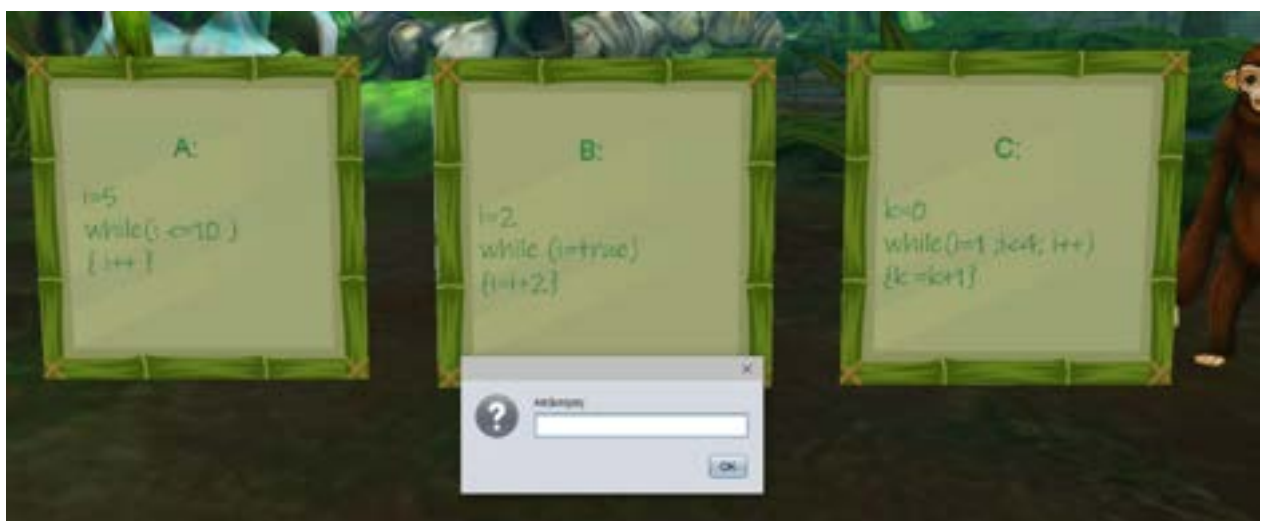


Για την υλοποίηση αυτού του σταδίου χρησιμοποιήθηκε αρχικά η **συνάρτηση(function)**

`this.monkey` `getStringFromUser` `/'Απάντηση/'` για να μπορέσει ο μαθητής να πληκτρολογήσει την απάντηση του. Έπειτα μέσα σε μια **while** ορίσαμε να γίνεται ο έλεγχος της ορθότητας της και στην περίπτωση λανθασμένης απάντησης να δίνεται στον μαθητή η ευκαιρία να προσπαθήσει ξανά.



Αντίστοιχα, με την προγραμματιστική λογική που παρουσιάστηκε παραπάνω εμφανίζονται πινακάκια επιλογής και για τη δομή επανάληψης **while**.



Η περιπέτεια του μαθητή όμως δεν σταματάει εδώ. Στο τρίτο στάδιο έχει ως στόχο να φτάσει στο κάστρο.



Για να το καταφέρει αυτό, ο μαθητής χρειάζεται να χρησιμοποιήσει ξανά τις γνώσεις του για τις δομές επανάληψης. Με την βοήθεια της **while** θα μπορέσει να προχωρήσει και να φτάσει το καράβι στο οποίο τον περιμένει και ο φίλος του ο Μαρσελ.

Για αυτόν τον σκοπό εφαρμόστηκε η **συνάρτηση (function)** `this.mike.getIntegerFromUser("steps")` η οποία εμφανίζει στην οθόνη του ένα αναδυόμενο παράθυρο. Σε αυτό ο μαθητής καλείται να πληκτρολογήσει τον αριθμό βημάτων που θεωρεί ότι χρειάζεται να κάνει για να φτάσει το καράβι.



Κατά τη διάρκεια κίνησης του προς το καράβι, ο μαθητής έχει τη δυνατότητα να δει πως λειτουργεί η δομή επανάληψης while. Σε κάθε βήμα του εμφανίζεται και ένα συννεφάκι που σχετίζεται με το κομμάτι κώδικα που βλέπει στο πινακάκι αριστερά.



Για τον προσανατολισμό και την κίνηση του χαρακτήρα, σε διάφορα σημεία του κώδικα

χρησιμοποιήθηκαν οι ακόλουθοι βασικοί μέθοδοι όπως `this.mike.pointAt(this.fishingBoat)`, `this.camera.moveAndOrientTo(this.humancamera2)`, `this.camera.setVehicle(this.mike)`,

για την πραγματοποίηση του παραπάνω στόχου. Επίσης ιδιαίτερα σημαντικό κρίνεται ότι συμπεριλήφθηκαν όλες οι περιπτώσεις αναλόγως των βημάτων που θα εισάγει ο μαθητής. Πιο συγκεκριμένα, αρχικά έχουν υπολογιστεί πόσα βήματα απαιτούνται για να φτάσει στο καράβι και δηλώθηκαν σε μια μεταβλητή

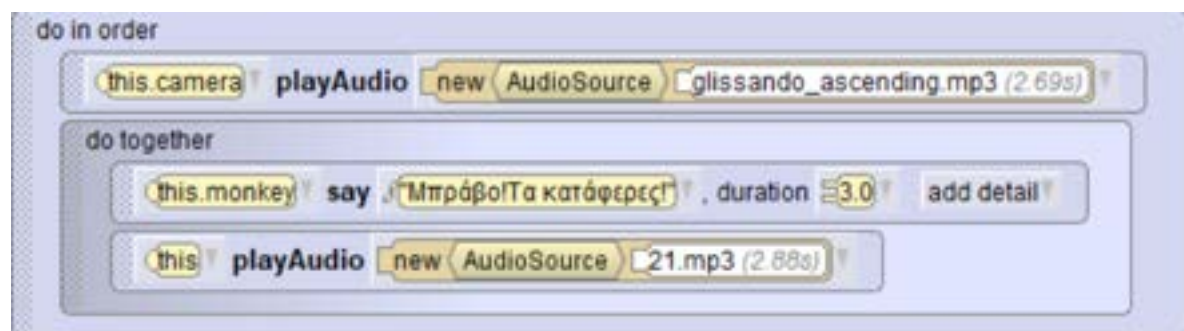
`WholeNumber steps`

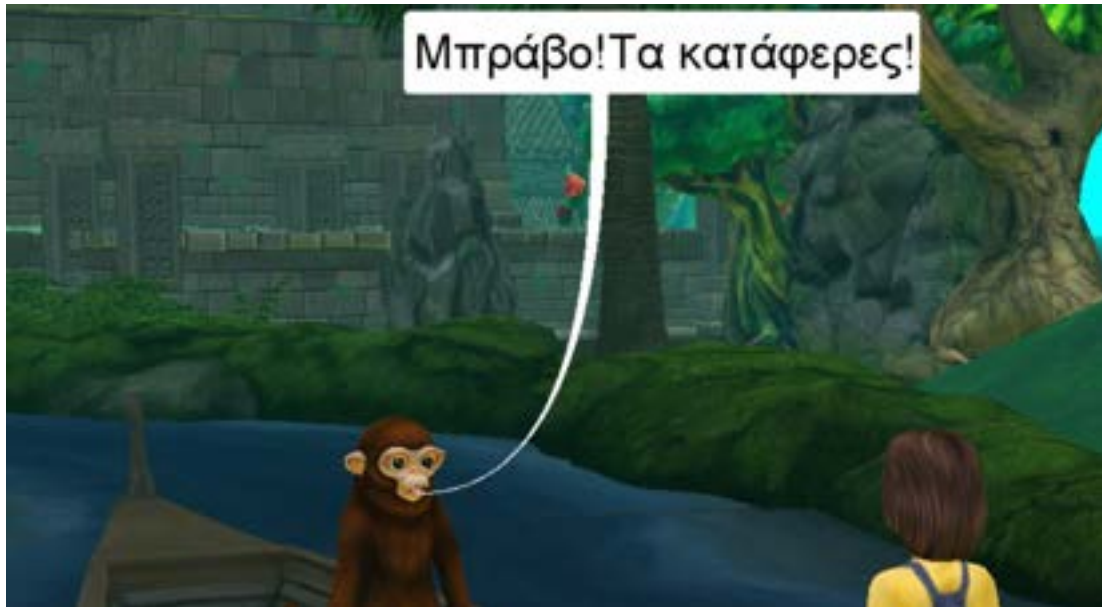
. Στην συνέχεια χρησιμοποιήθηκε η δομή ελέγχου **if** με διαφορετικές συνθήκες στις παρακάτω περιπτώσεις:

Για την πρώτη περίπτωση με συνθήκη `if (steps == steps)` δηλαδή αν ο μαθητής έχει πληκτρολογήσει επαρκή βήματα, χρησιμοποιείται η **μέθοδος**

`this.mike.moveToward(this.fishingBoat)`

μέσα σε μια δομή επανάληψης **while**. Ο μαθητής σε αυτήν την περίπτωση έχει καταφέρει τον στόχο του και μπαίνει στο καράβι ακούγοντας παράλληλα τη έναν ήχο επιτυχίας που έχουμε επιλέξει και βλέποντας το αντίστοιχο μήνυμα επιβράβευσης.





Στην περίπτωση της συνθήκης `if (steps < 0) is true then`, δηλαδή όταν τα βήματα που πληκτρολόγησε ο μαθητής δεν καλύπτουν την απόσταση, υπολογίζονται πόσα ακόμη βήματα χρειάζονται σε μια μεταβλητή `WholeNumber remaining ← steps - 0`. Έτσι, εμφανίζεται ξανά στην οθόνη του μαθητή ένα αναδυόμενο παράθυρο ενημερώνοντας τον πόσα βήματα του έχουν απομείνει να πληκτρολογήσει ώστε να φτάσει στο καράβι.



Στην τελευταία περίπτωση, που ο μαθητής έχει πληκτρολογήσει μεγαλύτερο αριθμό βημάτων από τον προαπαιτούμενο, ο χαρακτήρας του τον ενημερώνει ότι έχει επιλέξει παραπάνω από όσα θα έπρεπε.



Για αυτόν τον λόγο, του δίνεται η δυνατότητα να επιλέξει ξανά τα βήματα που θέλει, προτείνοντας του τον αριθμό βημάτων που απαιτούνται.



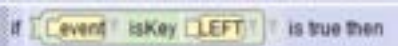
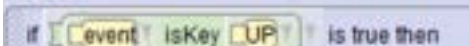

Πλέον οι χαρακτήρες του σεναρίου μας βρίσκονται στο καράβι και είναι έτοιμοι για το ταξίδι τους. Το μόνο που μένει είναι ο μαθητής να χρησιμοποιήσει τα βελάκια από το πληκτρολόγιο του για να ξεκινήσουν.



Για την επίτευξη αυτού του στόχου εφαρμόστηκαν τα κατάλληλα Event Listener.


Ειδικότερα έγινε η χρήση του **EventListener** `[this].addArrowKeyPressListener` και για κάθε πλήκτρο δηλώθηκε η κατάλληλη ενέργεια. Για παράδειγμα η σύνταξη των πλήκτρων έχει την παρακάτω δομή.

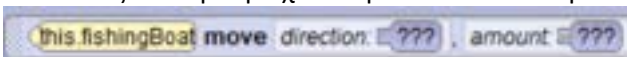


Η ίδια λογική εφαρμόστηκε και στις περιπτώσεις με συνθήκες , , .

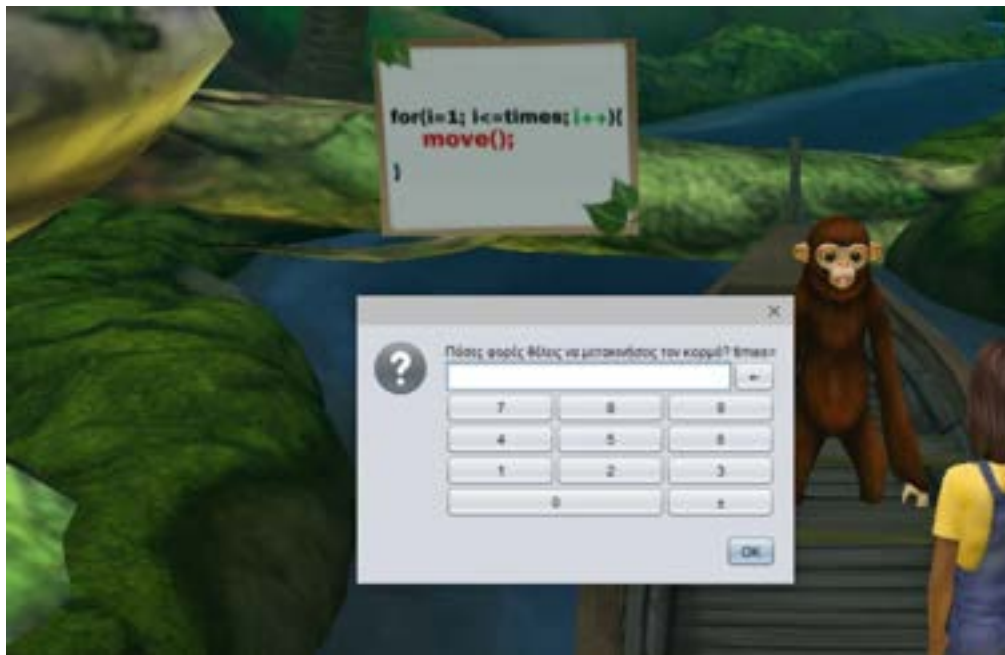
Ο μαθητής και ο φίλος του ο Μαρσελ απολαμβάνουν τη διαδρομή τους στο ποτάμι, όμως κάποια στιγμή το καράβι σταματάει να κινείται. Ένας κορμός δέντρου εμποδίζει τους χαρακτήρες να συνεχίσουν το ταξίδι τους στο ποτάμι του Αμαζονίου.



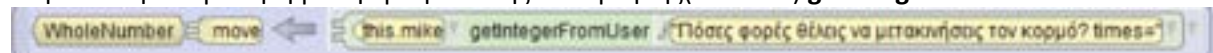
Για την ακινητοποίηση του καραβιού εφαρμόστηκε το **Event Listener** . Το **addCollisionListener** ενεργοποιείται όταν ένα αντικείμενο ακουμπάει ένα άλλο. Στην περίπτωση μας θα ενεργοποιηθεί όταν το καράβι ακουμπήσει τον κορμό δέντρου και στη συνέχεια θα εκτελεστούν οι εντολές που εμπεριέχονται μέσα σε αυτό. Η μέθοδος

 θα μετακινήσει το καράβι κάποια βήματα πίσω και θα το σταματήσει από την κίνηση του μπροστά.

Για ακόμη μια φορά ο μαθητής χρειάζεται να χρησιμοποιήσει αυτά που έμαθε. Ο χαρακτήρας Μαρσελ ζητάει την βοήθεια του μαθητή για τις γνώσεις του για τη δομή επανάληψης **for** ώστε να μπορέσουν να αντιμετωπίσουν το πρόβλημα που τους εμφανίστηκε.



Στην οθόνη του μαθητή με την βοήθεια της συνάρτησης(function) **getIntegerFromUser**



θα εμφανιστεί ένα αναδυόμενο παράθυρο ζητώντας του να πληκτρολογήσει πόσες φορές θέλει να μετακινήσει τον κορμό. Για την ολοκλήρωση αυτού του στόχου συμπεριλήφθηκαν δύο περιπτώσεις ανάλογα με την επιλογή του μαθητή για το πόσες φορές θέλει να μετακινήσει τον κορμό. Με την

βοήθεια της δομής ελέγχου **if** δημιουργήθηκαν οι περιπτώσεις:

if `move <= 4` **is true then**, όπου αν οι φορές που επέλεξε είναι λιγότερες από αυτές που χρειάζονται, του ζητείται να πληκτρολογήσει ξανά έναν αριθμό.

Έπειτα, ο κορμός θα μετακινηθεί με τη βοήθεια της μεθόδου **move**

`this.koutsouro` **move FORWARD**, και της `count`, η οποία θα επαναλάβει τις φορές μετακίνησης με βάση την επιλογή του χρήστη.

Σε οποιαδήποτε άλλη περίπτωση **else**, ο κορμός απομακρύνεται μπροστά από το καράβι ανοίγοντας ξανά τον δρόμο για να προχωρήσουν οι χαρακτήρες. Το κομμάτι κώδικα που αναλύσαμε προηγουμένως παρουσιάζεται παρακάτω.



Αξίζει να τονιστεί ότι μέσα από τη διαδικασία κίνησης του κορμού, ο μαθητής έχει τη δυνατότητα να δει σε πράξη τη δομή επανάληψης **for** και να κατανοήσει καλύτερα τη λειτουργία της.



Οι χαρακτήρες μας πλέον μπορούν να συνεχίσουν το ταξίδι τους χωρίς κανένα εμπόδιο. Στο τέλος του ποταμιού βρίσκεται το κάστρο και έτσι έχουν φτάσει στον προορισμό τους με επιτυχία. Σε αυτό το σημείο χρειάζεται να αναφερθεί ότι εφαρμόστηκε ξανά το **Event Listener**



με στόχο να ακινητοποιηθεί το καράβι όταν φτάσουν οι χαρακτήρες στο τέλος της διαδρομής.



(5.5) Τρίτο εκπαιδευτικό σενάριο-Τελεστές και Δομές Ελέγχου

Σκοπός του τρίτου εκπαιδευτικού σεναρίου είναι ο μαθητής να κατανοήσει την έννοια των τελεστών (αριθμητικών, συγκριτικών και λογικών) καθώς και τις δομές ελέγχου. Βασικός χαρακτήρας του σεναρίου αυτού είναι η πριγκίπισσα μαζί με το δράκο της. Σκοπός τους είναι να αντιμετωπίσει τα τρία τερατάκια τα οποία δεν την αφήνουν να περάσει στο κάστρο και να κατακτήσει το γαλάζιο διαμάντι του Αμαζονίου.



Προγραμματιστικό ενδιαφέρον έχει ο δράκος σαν αντικείμενο στο περιβάλλον του Alice. Καθόλη τη διάρκεια που εμφανίζεται ο δράκος κουνάει τα φτερά του, και αυτή η λεπτομέρεια έγινε με σκοπό να εμπλουτιστεί ο «κόσμος» που διαδραματίζεται σε αυτό το σενάριο, αλλά και να συνεισφέρει στην προσπάθεια να είναι πιο ρεαλιστικοί οι χαρακτήρες. Για την επίτευξη αυτού έγινε χρήση της εντολής **flapWind** που δίνεται από το Alice και σε συνδυασμό με την εντολή **count**.



Με την έναρξη του εκπαιδευτικού σεναρίου γίνεται μια σύντομη περιγραφή της θεωρίας. Για να γίνει πιο ενδιαφέρον στο μαθητή, η θεωρία χωρίζεται σε δυο κομμάτια των τελεστών και των δομών ελέγχου αντίστοιχα. Επίσης αυτό το σενάριο περιλαμβάνει και δυο παιχνίδια θεωρίας στα οποία ο μαθητής ελέγχει τις γνώσεις που απόκτησε από αυτό. Ξεκινώντας με το πρώτο σκέλος των τελεστών η πριγκίπισσα λέει στο χρήστη ότι θα χρειαστεί τη βοήθεια του ώστε να βρει τα βέλη και το τόξο της για να αντιμετωπίσει τα τέρατα στην πορεία. Ο μαθητής καλείται να απαντήσει σωστά σε σύντομες ερωτήσεις θεωρίας, οι οποίες αν απαντηθούν ορθά θα τον οδηγήσουν στην αποκάλυψη του τόξου της πριγκίπισσας.

Αρχικά γίνεται μια γνωριμία του μαθητή με τους τελεστές και τη θεωρία τους.



Η παρουσίαση της θεωρίας γίνεται με τη μέθοδο της αφήγησης και με τη χρήση βοηθητικών πινάκων που εμφανίζονται παράλληλα με την αφήγηση από την πρωταγωνίστρια. Για να επιτευχθεί αυτό χρησιμοποιήσαμε την εντολή **place** με την οποία τοποθετούμε το χαρακτήρα δίπλα ακριβώς σε όποιο αντικείμενο επιθυμούμε.



Και σε αυτό το σενάριο, όπως και στο σύνολο του παιχνιδιού χρησιμοποιήθηκε ο συνδυασμός κειμένου και ήχου, ώστε να γίνεται πιο ελκυστικό στο χρήστη. Για την αφήγηση όπως έχουμε αναλύσει και πιο πάνω, χρησιμοποιήθηκαν κυρίως οι εντολές **say**, τα αντικείμενα με τη μορφή **billboard** αλλά και τα πλακίδια εντολών **do together** και **do in order**. Η εμφάνιση και η εναλλαγή των πινάκων μέσα στον κώδικα, απαιτούν τη χρήση της εντολής **setOpacity**, με την οποία ανάλογα την τιμή που θα ορίσεις (μέγιστη τιμή 1, ελάχιστη τιμή 0) καθορίζεις πότε θα εμφανίζεται και πότε θα εξαφανίζεται αντίστοιχα ο πίνακας.



Για τη χρονική αλληλουχία που ακολουθεί η παρουσίαση των πινάκων θεωρίας γίνεται χρήση της εντολής **do together**. Επίσης παρατηρούμε ότι χρησιμοποιείται και η εντολή **delay** που καθορίζει τη χρονική διάρκεια που παραμένει ένα αντικείμενο, αλλά και ξανά η εντολή **place** με την οποία ορίζεται ο τρόπος με τον οποίο τοποθετούνται οι πίνακες στον "κόσμο" του παιχνιδιού.



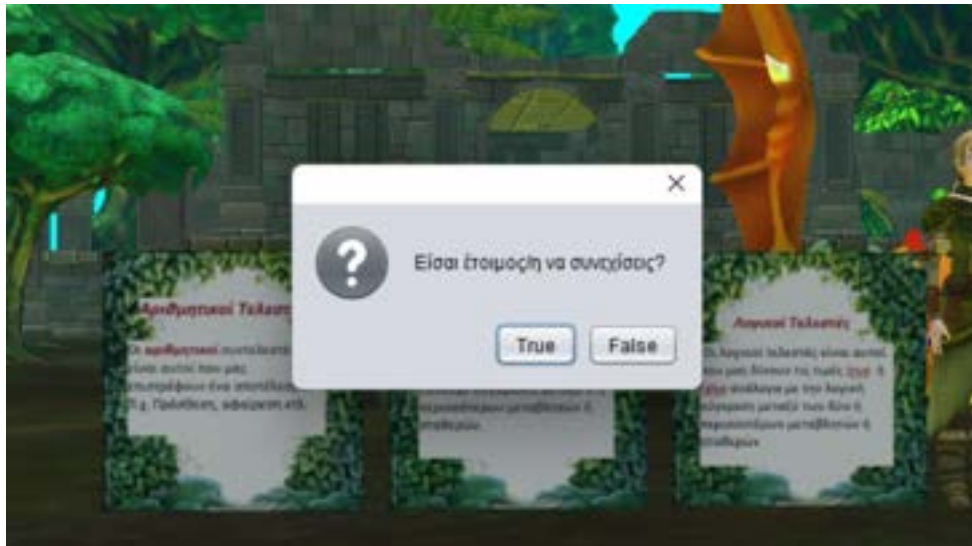
Το αποτέλεσμα της εκτέλεσης του παραπάνω κώδικα είναι το ακόλουθο.



Αξίζει να σημειωθεί ότι σε αυτό το εκπαιδευτικό σενάριο υπάρχει μεγάλος όγκος θεωρίας που εμφανίζεται σε πίνακες. Για αυτόν το λόγο δίνεται η δυνατότητα μέσω ενός αναδυόμενου παραθύρου που εμφανίζεται στην οθόνη, ο χρήστης να επιλέξει να παραμείνει σε αυτό το στάδιο έχοντας επιπλέον χρόνο στη θεωρία.



Για την πραγματοποίηση του παραπάνω δημιουργήθηκε μια συνάρτηση. Μέσα σε αυτή υπάρχει μια εντολή `delay` η οποία ανάλογα με την απάντηση του χρήστη του δίνει επιπλέον χρόνο ώστε να ολοκληρώσει την ανάγνωση του.



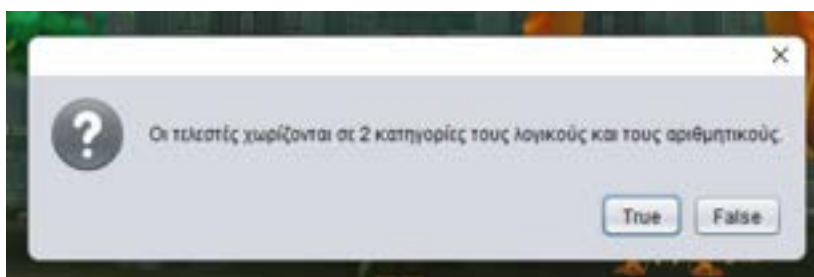
Στην συνέχεια η συνάρτηση καλείται από το κύριο πρόγραμμα, με τον ακόλουθο τρόπο.



Αφού ολοκληρωθεί η παρουσίαση η πριγκίπισσα μας ενημερώνει ότι είναι ώρα να αρχίσουμε το παιχνίδι. Το παιχνίδι αποτελείται από ένα σύνολο θεωρητικών ερωτήσεων που στηρίζονται σε όλα αυτά που αναλύθηκαν νωρίτερα. Ένα ενδεικτικό παράδειγμα αποτελεί το παρακάτω.



Η ερώτηση εμφανίζεται σε ένα ξεχωριστό πλαίσιο όπου ο μαθητής αφού διαβάσει και κατανοήσει την ερώτηση επιλέγει την αντίστοιχη απάντηση με το ποντίκι του.



Για την υλοποίηση του παιχνιδιού από προγραμματιστική άποψη, ακολουθήθηκε ένας συνδυασμός από πλακίδια εντολών **do in order** και **do together** παρόμοιο με αυτόν που έχει αναλυθεί και σε παραπάνω εκπαιδευτικό σενάριο. Για τη δημιουργία των ερωτήσεων χρησιμοποιήθηκαν οι **boolean** μεταβλητές και η δομή ελέγχου **while**. Πιο αναλυτικά δημιουργήσαμε μια μεταβλητή και την ορίσαμε να είναι τύπου **boolean** δηλαδή να δέχεται μόνο δυο απαντήσεις σωστό ή λάθος(true/false). Στην συνέχεια αναθέσαμε σε αυτήν μια εντολή που δίνεται από το προγραμματιστικό περιβάλλον του Alice την **getBooleanFromUser** με την οποία ανάλογα την απάντηση που θα επιλέξει ο χρήστης η μεταβλητή λαμβάνει και την αντίστοιχη τιμή. Τέλος προστέθηκε το κείμενο που αντιστοιχεί στην ερώτηση θεωρίας το οποίο και εμφανίζεται στο αναδυόμενο παράθυρο.



Για τον έλεγχο της απάντησης χρησιμοποιήθηκε η δομή ελέγχου **while** μέσα στην οποία εντάχθηκε ένα εμφωλευμένο **do together**. Στη συνθήκη ελέγχου τοποθετήθηκε η απάντηση από το χρήστη και όσο η απάντηση είναι λανθασμένη, εμφανίζεται η πριγκίπισσα που τον ενημερώνει να προσπαθήσει ξανά. Αυτό αποσκοπεί στο να μπορέσει ο μαθητής σε περίπτωση λάθους να ξανά διαβάσει την ερώτηση και να εντοπίσει το λάθος του. Με αυτόν τον τρόπο όχι μόνο κατανοεί καλύτερα την θεωρία αλλά γίνεται και απενοχοποίηση των λαθών που μπορεί να κάνει.



Μετά το πέρας των ερωτήσεων και εφόσον έχουν απαντηθεί σωστά, εμφανίζονται τα βέλη και το τόξο της πριγκίπισσας και είμαστε έτοιμοι για την συνέχεια του σεναρίου.

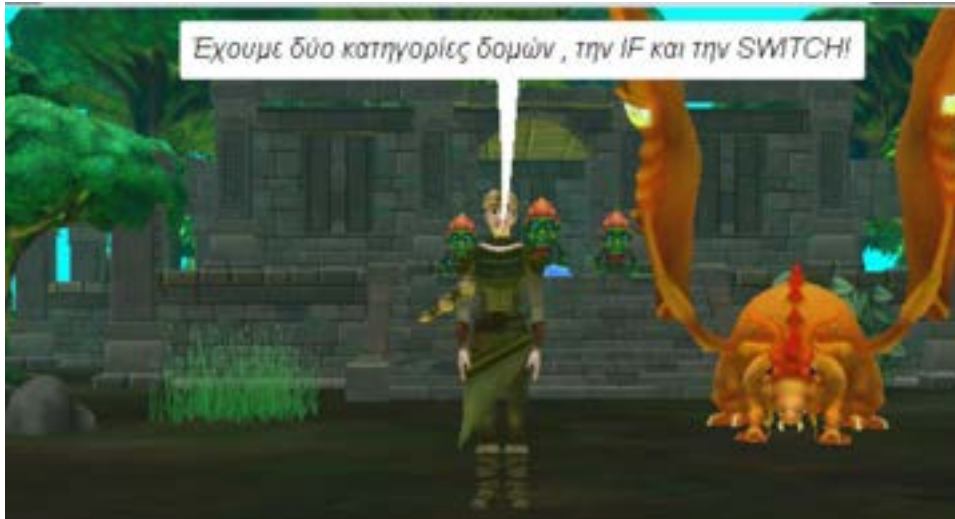


Για να υλοποιηθεί το παραπάνω τοποθετήσαμε ένα σύνολο από αντικείμενα μέσα στο περιβάλλον τα οποία προγραμματίστηκαν με κατάλληλό τρόπο. Αυτό είχε ως απώτερο σκοπό μετά την επιτυχή ολοκλήρωση των ερωτήσεων, ένας βράχος που βρίσκεται στο σκηνικό να μετακινηθεί και να αποκαλύψει τα βέλη και το τόξο. Στη συνέχεια θα δούμε αναλυτικότερα τον κώδικα.

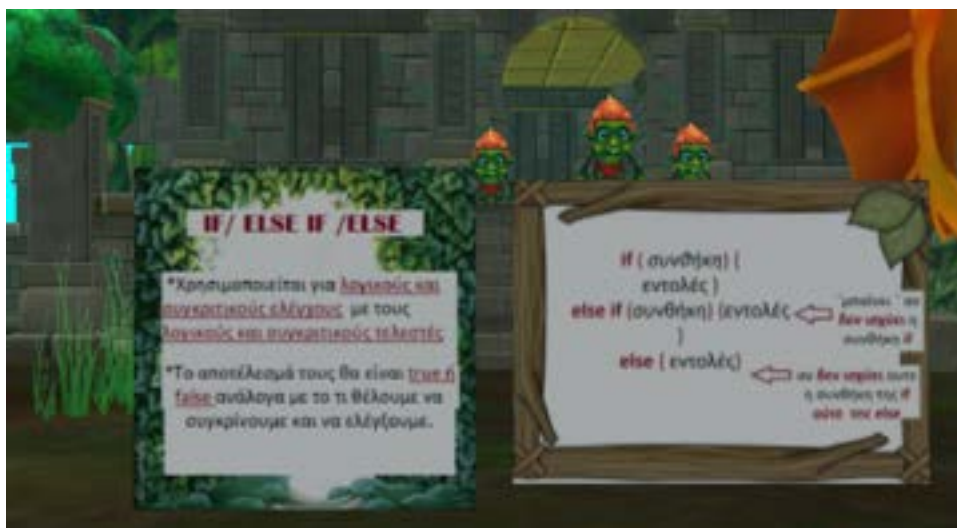


Στον κώδικα μας αρχικά πραγματοποιείται ένα **do together** στο οποίο γράφουμε τέσσερις εντολές οι οποίες γίνονται ταυτόχρονα. Ξεκινώντας έχουμε την εντολή **move** με την οποία μετακινούμε τον βράχο προς τα δεξιά και γράφουμε επίσης την ακριβή απόσταση που θέλουμε να μετακινηθεί. Έπειτα γίνεται χρήση της εντολής **setOpacity** με την οποία θέτοντας την τιμή ίση με το ένα εμφανίζονται τα βέλη και το τόξο. Μια λεπτομέρεια που με την οποία θα μπορούσαμε να καταλάβουμε την σημαντικότητα της εντολής **setOpacity** στην τελική εμφάνιση του παιχνιδιού είναι το ακόλουθο. Παρατηρούμαι ότι έχουμε ορίσει την τιμή της εντολής **setOpacity** για το διακοσμητικό γρασίδι που υπάρχει πίσω από τα υπόλοιπα αντικείμενα ίση με 0.6. Αυτό έχει ως αποτέλεσμα να είναι πιο αχνό από τα υπόλοιπα αντικείμενα(βέλη, τόξο) τα οποία θέλουμε να ξεχωρίζουν καθώς είναι τα βασικά. Αξίζει να τονιστεί επιπλέον και η χρήση της εντολής **place** με την βοήθεια της οποίας τοποθετούμε ακριβώς στο σημείο που θέλουμε τα αντικείμενα μας μέσα στο σκηνικό.

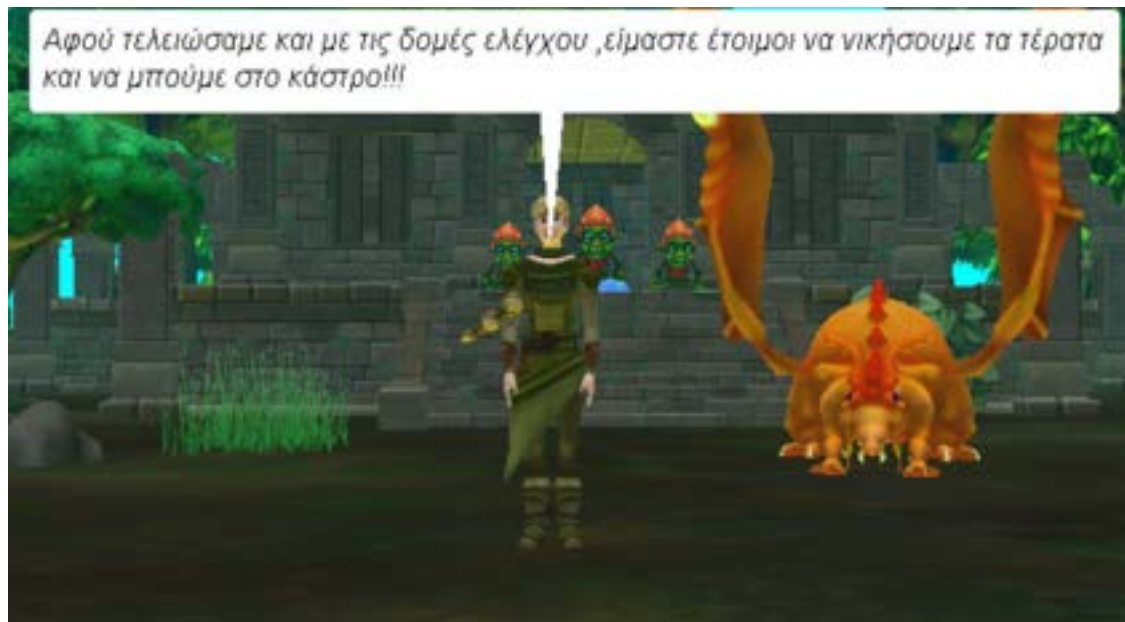
Στην συνέχεια του εκπαιδευτικού σεναρίου και εφόσον ολοκληρωθεί το προηγούμενο κομμάτι θεωρίας που αναλύθηκε πιο πάνω πηγαίνουμε στο δεύτερο μέρος του. Σε αυτό επικεντρωνόμαστε στις δομές ελέγχου και πιο συγκεκριμένα γίνεται η ανάλυση δυο εξ 'αυτών της **if** και της **switch** οι οποίες αποτελούν πολύ καλό παράδειγμα για την πρώτη γνωριμία των μαθητών με τις δομές επανάληψης.



Για την παρουσίαση της θεωρίας χρησιμοποιείται ξανά η αφήγηση σε συνδυασμό με την παρουσίαση πινάκων ώστε το εκπαιδευτικό σενάριο να έχει μια λογική ακολουθία στη δομή του. Επειδή το συγκεκριμένο παιχνίδι επικεντρώνεται κυρίως στην εισαγωγή της εκμάθησης του αντικειμενοστραφή προγραμματισμού προστέθηκαν σε billboard η σύνταξη της δομής ελέγχου **if** και **switch** σε γλώσσα προγραμματισμού Java. Έτσι ο μαθητής έρχεται σε μια πρώτη επαφή τόσο σε θεωρητικό όσο και σε προγραμματιστικό επίπεδο με αυτές.



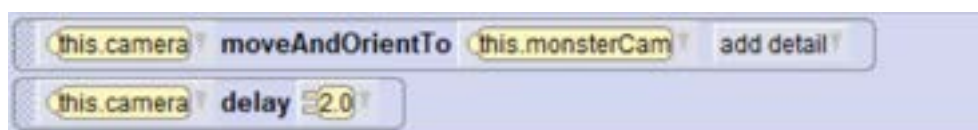
Στον κώδικα η υλοποίηση της αφήγησης έγινε σύμφωνα με τον τρόπο που έχει αναλυθεί και πιο πάνω εκτενέστερα. Για την επεξεργασία της εμφάνισης των πινάκων χρησιμοποιήθηκαν έτοιμες εικόνες που δημιουργήθηκαν από εμάς και εντάχθηκαν μετέπειτα στο περιβάλλον της Alice.



Στην συνέχεια αυτό που ακολουθεί είναι το δεύτερο παιχνίδι στο οποίο ο μαθητής καλείται να απαντήσει σε κάποιες απλές ερωτήσεις θεωρίας οι οποίες βασίζονται στην θεωρία που επεξηγήθηκε στην αρχή του εκπαιδευτικού σεναρίου. Στην οθόνη του εμφανίζονται τα τρία τερατάκα που προστατεύουν το διαμάντι και εμποδίζουν την πριγκίπισσα να φτάσει στο κάστρο για να το αποκτήσει.



Για να επιτευχθεί το παραπάνω τοποθετήθηκε μια κάμερα που δείχνει τα τρία νέα αντικείμενα στο εκπαιδευτικό σενάριο. Επίσης έγινε χρήση της εντολής **moveAndOrientTo** αλλά και της εντολής **delay** με σκοπό την εναλλαγή στο σκηνικό του παιχνιδιού.



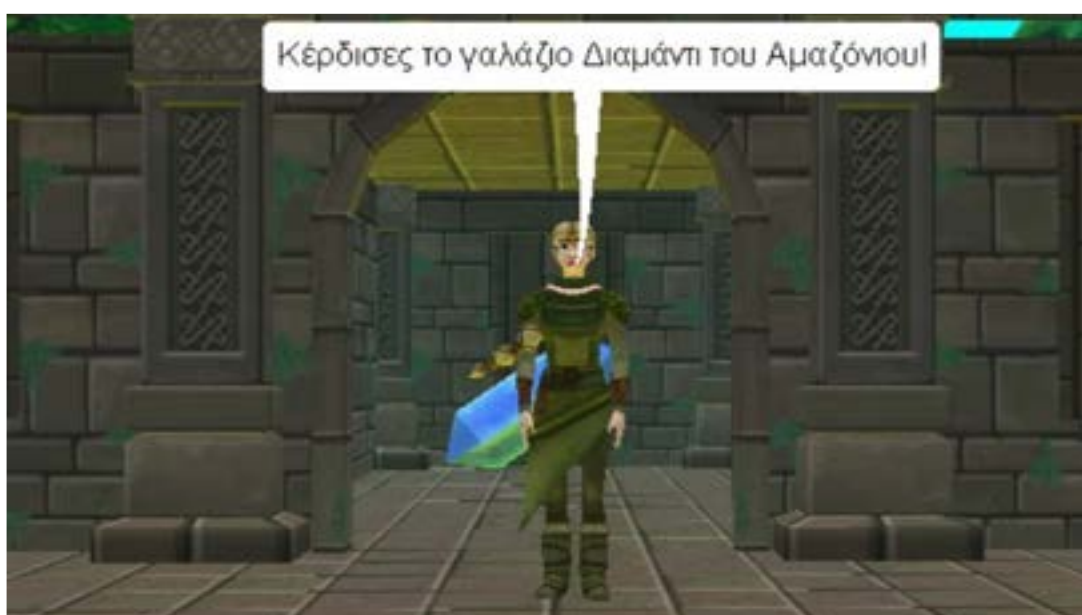
Για την εμφάνιση των ερωτήσεων ακολουθήθηκε ο παρακάτω κώδικας ο οποίος παρουσιάζεται αναλυτικά.

```
Boolean question5 ← (this.cunpira2.getBooleanFromUser /("Στην IF/ELSE IF αλλάζει πάντα η συνθήκη του else if"))
while (question5 is true
  (this.feya2 say /("Προσπάθησε (ανά)", duration 1.0) add detail)
  question5 ← (this.getBooleanFromUser /("Στην IF/ELSE IF αλλάζει πάντα η συνθήκη του else if"))
loop
(this.cunpira2 say /("ΟΧΑΙ!!", fontColor ORANGE, bubbleOutlineColor RED, textStyle BOLD, duration 2.0) add detail)
(this.cunpira2 setOpacity 0.0) add detail
```

Πρώτο βήμα είναι η δήλωση και ο ορισμός μιας μεταβλητής τύπου **Boolean**. Η συγκεκριμένη μεταβλητή έχει σαν όρισμα την συνάρτηση **getBooleanFromUser** η οποία έχει σαν αποτέλεσμα η τελική τιμή της να καθορίζεται από την απάντηση (true/false) που θα δώσει ο μαθητής. Έπειτα ακολουθεί η δομή επανάληψης **while** στην οποία ελέγχεται η απάντηση που δίνει ο χρήστης. Μόλις η απάντηση του είναι σωστή ακολουθούν οι επόμενες εντολές κατά τις οποίες το τερατάκι εκφράζει την δυσαρέσκεια του και εξαφανίζεται.



Το παιχνίδι ολοκληρώνεται μόλις απαντηθούν σωστά και οι τρεις ερωτήσεις και η πριγκίπισσα μπορεί να φτάσει στο κάστρο.



Για την εμφάνιση των τελευταίων αντικειμένων σε αυτό το εκπαιδευτικό σενάριο, ακολουθήθηκε παρόμοια προγραμματιστική λογική. Τόσο το διαμάντι όσο και το "ΤΕΛΟΣ" που εμφανίζονται τοποθετήθηκαν μέσα σε ένα πλακίδιο **do together** και έγινε χρήση της εντολής **move** καθώς στόχος μας ήταν να μην είναι στατικά για να κάνουν το παιχνίδι πιο ελκυστικό.



1. Κώδικας για το διαμάντι.



2. Κώδικας για το "ΤΕΛΟΣ"

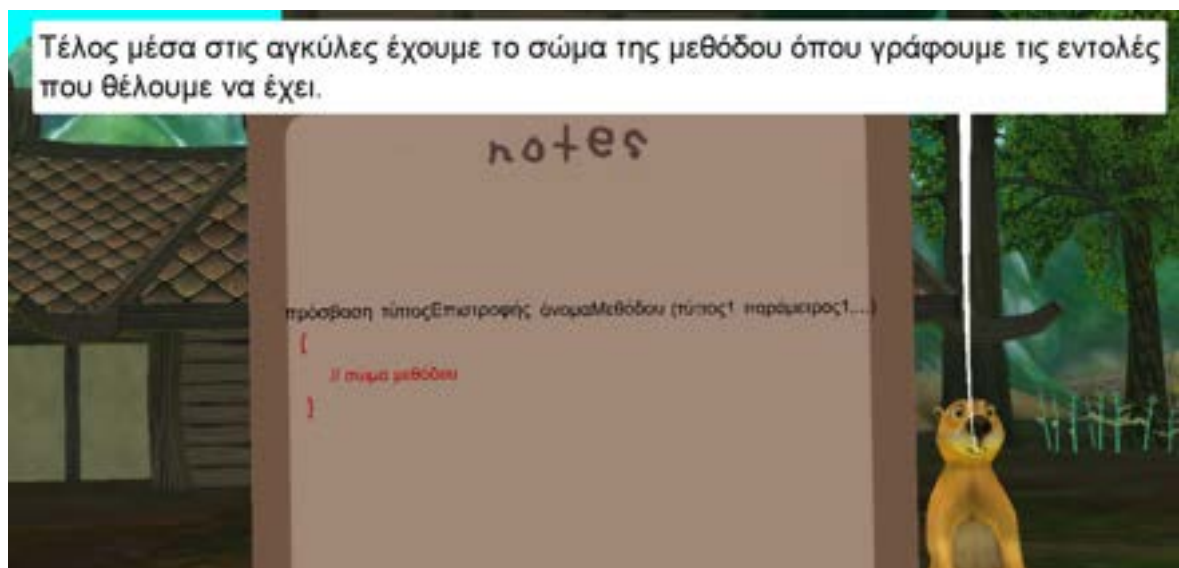
(5.6) Τέταρτο εκπαιδευτικό σενάριο-Μέθοδοι

Σκοπός του τέταρτου εκπαιδευτικού σεναρίου είναι ο μαθητής να γνωρίσει τις μεθόδους σε γλώσσα Java. Πρωταγωνιστής του σεναρίου είναι ένας κάστορας όπου θα μάθει στο μαθητή πως να δημιουργήσει αλλά και πως να καλέσει μια μέθοδο.

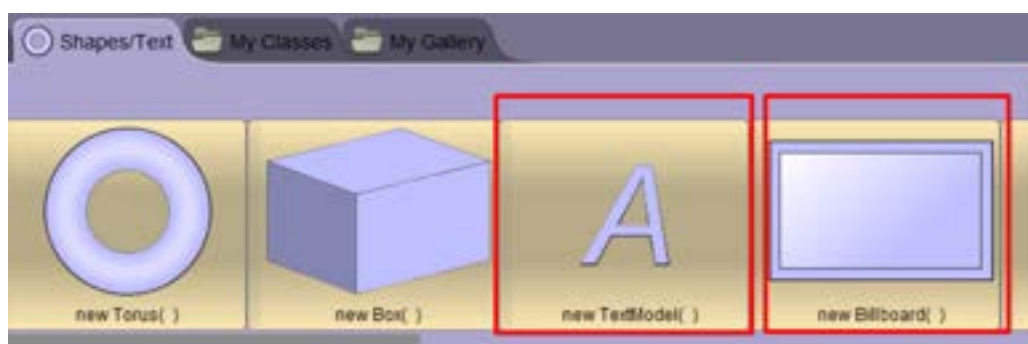


Στο πρώτο στάδιο του σεναρίου ο κάστορας με τη χρήση της αφήγησης εξηγεί βήμα βήμα τη δομή που έχει μια μέθοδος. Μερικά παραδείγματα από την αφήγηση των εισαγωγικών εννοιών της

μεθόδου είναι τα παρακάτω. Αξίζει να σημειωθεί ότι τα πινακάκια που θα παρουσιαστούν έχουν δημιουργηθεί από εμάς και μετέπειτα προστέθηκαν στον κώδικα.



Για την εμφάνιση των πινακίων αλλά και του κειμένου τοποθετήθηκαν στο περιβάλλον του Alice αντικείμενα και σχήματα. Για την ακρίβεια χρησιμοποιήθηκε το **TextModel** και το **Billboard** από τα **Shapes/Texts**.



Η τροποποίηση και η επεξεργασία τους έγινε μέσω των ιδιοτήτων(Properties) που παρέχονται από το πρόγραμμα Alice. Στη συνέχεια σε προγραμματιστικό επίπεδο για την αλλαγή του χρώματος του κάθε κειμένου σε διαφορετικά χρονικά διαστήματα χρησιμοποιήθηκε η μέθοδος **setPaint**. Επιλέγοντας το πεδίο paint εμφανίζεται ένα dropdown μενού δίνοντας πολλές διαφορετικές επιλογές χρωμάτων στο χρήστη για το κείμενο.



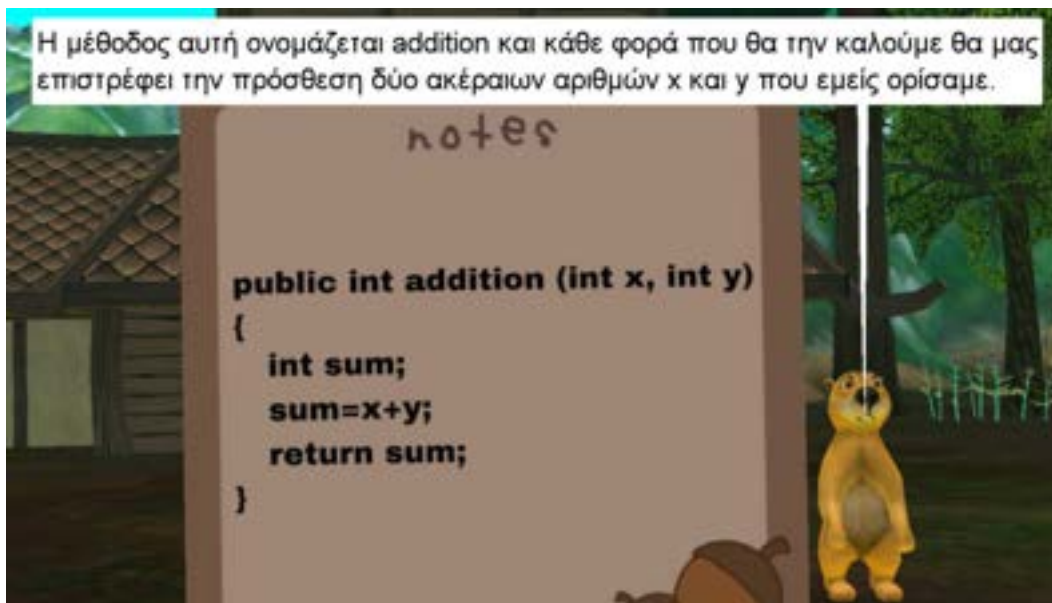
Σημαντική θεωρήθηκε ξανά η χρήση του **do together** σε πολλά σημεία για την αφήγηση της θεωρίας. Σε διαφορετικούς παράλληλους προγραμματισμούς τοποθετήθηκαν μέθοδοι για την ακρόαση της ομιλίας του χαρακτήρα και την αλλαγή χρώματος του κειμένου σε κόκκινο και έπειτα ξανά σε χρώμα μαύρο. Η εναλλαγή χρωμάτων κατά τη διάρκεια της αφήγησης της θεωρίας έγινε για την καλύτερη διευκρίνιση και κατανόηση του σημείου όπου γίνεται η παρουσίαση του.



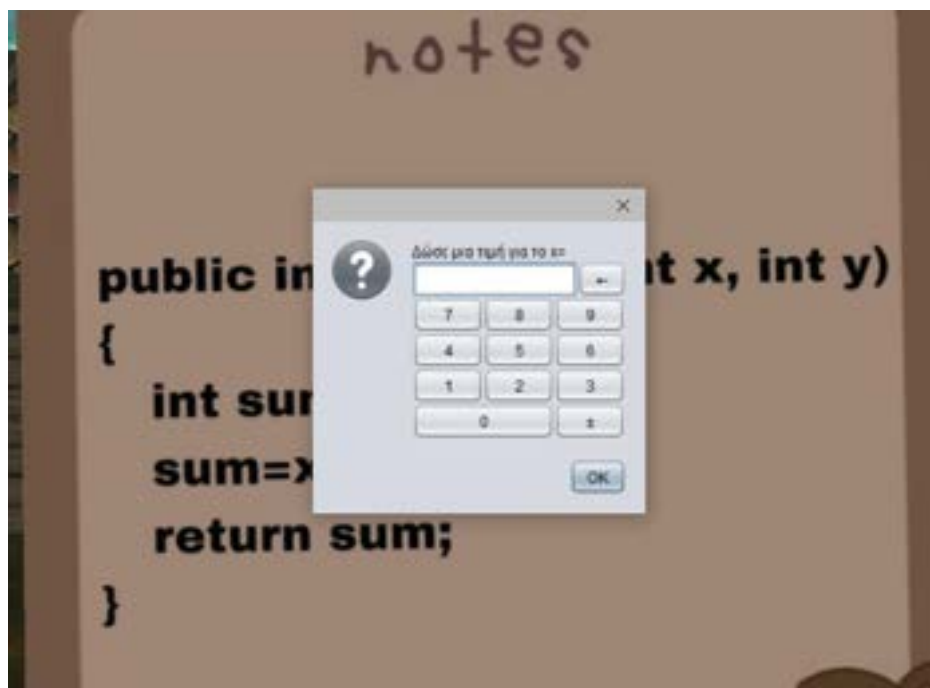
Κώδικας για την εναλλαγή χρώματος

Ο μαθητής πλέον έχει μάθει πως δομείται μια μέθοδος. Ο φίλος του ο κάστορας τον παρακινεί να δουν ένα παράδειγμα για να καταλάβει πως λειτουργεί η χρήση της.

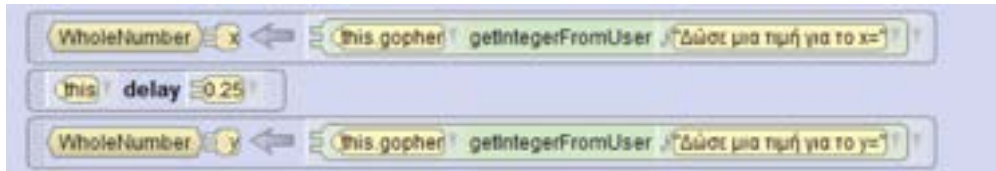
Σε αυτό το σημείο γίνεται μια σύντομη επεξήγηση της μεθόδου που βλέπει ο μαθητής στον πίνακα.



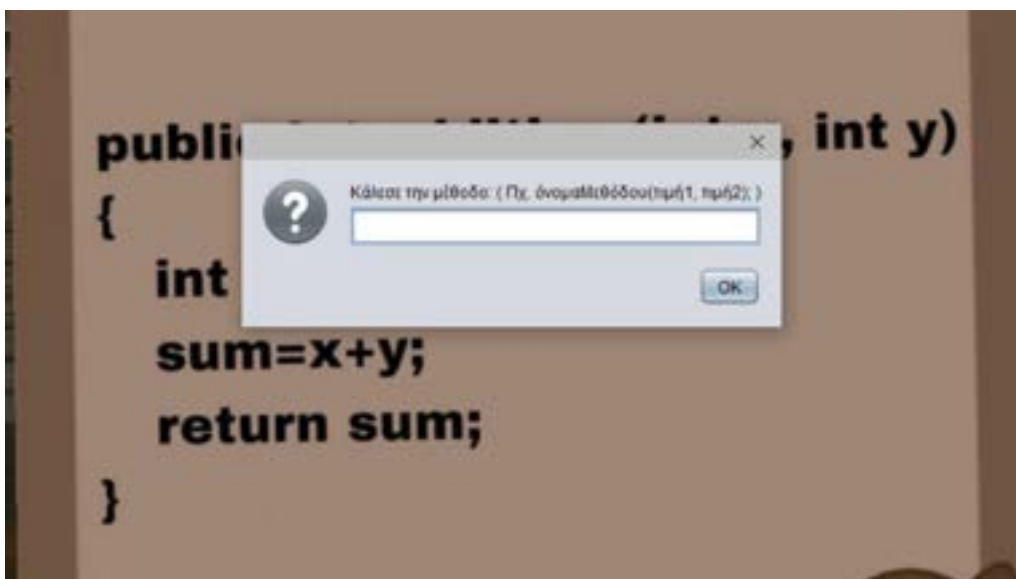
Εφόσον ο μαθητής έχει καταλάβει τι κάνει η μέθοδος, ζητείται από αυτόν να εισάγει κάποιες τιμές για τις παραμέτρους της. Με αυτόν τον τρόπο θα μπορέσει να δει σε πράξη τόσο τη λειτουργία της όσο και το τι θα επιστρέψει η μέθοδος.



Η εμφάνιση του παραθύρου για την εισαγωγή ή πληκτρολόγηση αριθμού που παρουσιάζεται παραπάνω έγινε με την ένταξη των παρακάτω εντολών μέσα σε μια **do in order**. Πρώτα θα ζητηθεί από το μαθητή να εισάγει μια τιμή για την πρώτη παράμετρο με όνομα x και έπειτα για την δεύτερη παράμετρο με όνομα y. Η ένταξη του **delay** σε πολλά σημεία του κώδικα κρίθηκε σημαντική για το αισθητικό κομμάτι της εφαρμογής.



Το μόνο που μένει τώρα μετά τον ορισμό των παραμέτρων είναι ο μαθητής να καλέσει τη μέθοδο που είδε. Ο χαρακτήρας κάστορας θα τον καθοδηγήσει, εξηγώντας και δείχνοντας του την μορφή που έχει το κάλεσμα μιας μεθόδου με παραμέτρους. Ο μαθητής θα κλιθεί ξανά να πληκτρολογήσει στο αναδυόμενο παράθυρο που θα του εμφανιστεί τον τρόπο με τον οποίο καλούμε μια μέθοδο. Για τη δική του διευκόλυνση στο αναδυόμενο παράθυρο πληκτρολόγησης έχουμε συμπεριλάβει βοηθητικά μηνύματα τα οποία αλλάζουν ανάλογα με τις λάθος προσπάθειες του. Πιο αναλυτικά στην πρώτη του προσπάθεια θα του εμφανιστεί το μήνυμα που δείχνει τη μορφή που έχει η κλήση της μεθόδου.



Σε προγραμματιστικό επίπεδο αρχικά δημιουργήθηκε μια μεταβλητή και επιλέχθηκε ο τύπος δεδομένων **TextString**. Η επιλογή αυτού του τύπου έγινε γιατί ο χρήστης θα πληκτρολογήσει συμβολοσειρά και όχι αριθμούς όπως προηγουμένως. Με την εισαγωγή μεταβλητής στον κώδικα εμφανίζεται ένα παράθυρο που δίνει τη δυνατότητα στον χρήστη να κάνει τις απαραίτητες αλλαγές.



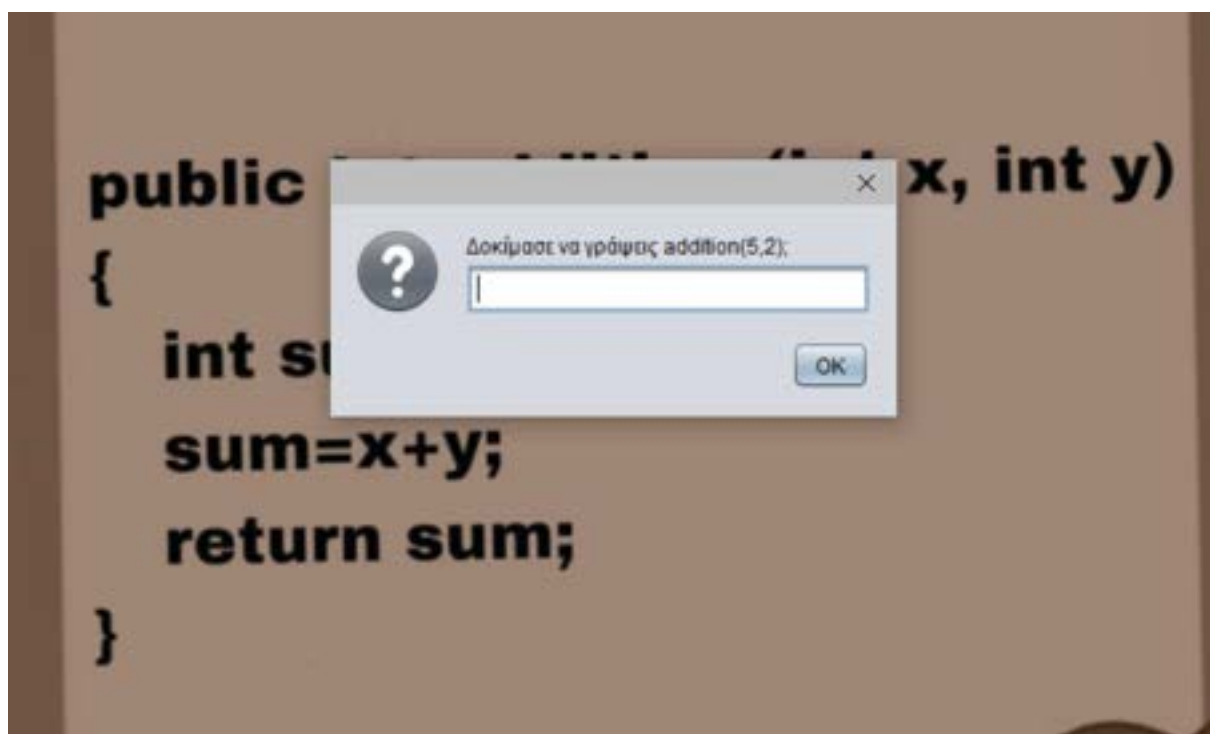
Για την εμφάνιση παραθύρου πληκτρολόγησης έγινε χρήση της συνάρτησης(function) `getStringFromUser` γιατί επιλέχθηκε το `TextString` ως τύπος δεδομένων.



Στην περίπτωση λανθασμένης απάντησης του μαθητή, ο κάστορας τον ενημερώνει ότι έχει γίνει κάποιο λάθος και του δίνεται η ευκαιρία να ξαναπροσπαθήσει.



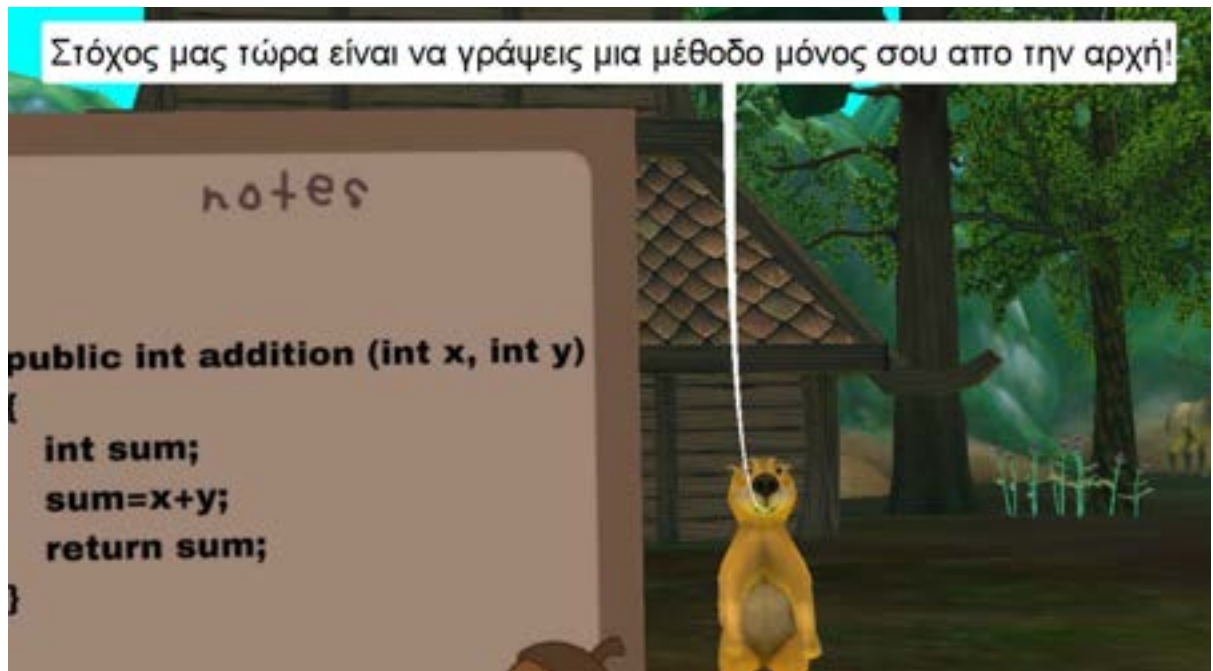
Επίσης έχουμε δημιουργήσει μια μεταβλητή που μετράει πόσες προσπάθειες έχει κάνει ο μαθητής. Μετά από έναν ορισμένο αριθμό εμφανίζεται σε αυτόν ένα καινούργιο μήνυμα κατατοπίζοντας τον πιο συγκεκριμένα πως πρέπει να γραφτεί η κλήση της μεθόδου με τις τιμές που έχει επιλέξει.



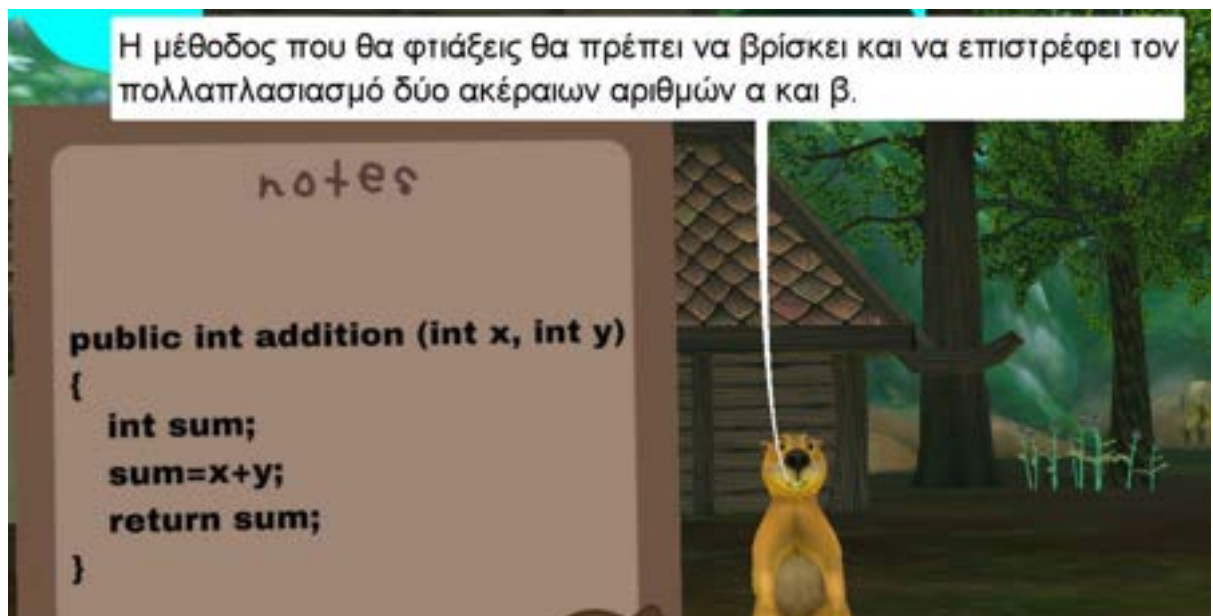
Όταν ο μαθητής πραγματοποιήσει σωστά αυτό που του ζητείται θα εμφανιστεί στην οθόνη του το αποτέλεσμα που θα επιστρέψει η μέθοδος.



Στη συνέχεια του εκπαιδευτικού σεναρίου, ο κάστορας ενημερώνει το μαθητή πως ήρθε η ώρα να γράψει μια μέθοδο από την αρχή μόνος του. Ιδιαίτερα βοηθητικό είναι ότι ο πίνακας με το παράδειγμα που είδανε προηγουμένως θα παραμείνει στο αριστερό μέρος της οθόνης του μαθητή. Αυτή η ενέργεια είχε ως στόχο ο μαθητής να μπορεί να κοιτάει τον πίνακα οποιαδήποτε στιγμή χρειάζεται βοήθεια σχετικά με τη δομή της μεθόδου.



Η αφήγηση από τον κάστορα προχωράει δίνοντας στο μαθητή τα απαραίτητα στοιχεία για την μέθοδο που έχει να φτιάξει. Ο κάστορας θα τον ενημερώσει για το όνομα της μεθόδου, τις παραμέτρους της αλλά και το τι θα κάνει. Παρακάτω διατυπώνεται ένα κομμάτι από αυτό το σημείο της αφήγησης του σεναρίου.



Στόχος του σεναρίου σε αυτήν την φάση της εφαρμογής είναι βήμα βήμα ο μαθητής να ξεκινήσει να πληκτρολογεί τα σωστά σύμβολα και τις σωστές εντολές για τη δημιουργία της μεθόδου. Όπως και προηγουμένως με την εμφάνιση του παραθύρου έχουμε εισάγει και τα κατάλληλα μηνύματα για να βοηθήσουμε το μαθητή σε όλη αυτήν τη διαδικασία. Κάθε φορά που ο μαθητής επιτυγχάνει τον στόχο του και γράφει σωστά το κομμάτι που του ζητείται συνεχίζει στον επόμενο στόχο με την εμφάνιση νέου παραθύρου πληκτρολόγησης. Και σε αυτό το κομμάτι έχουμε συμπεριλάβει την αλλαγή μηνυμάτων στο αναδυόμενο παράθυρο ανάλογα με τις προσπάθειες του μαθητή. Τα μηνύματα αλλάζουν για τη διευκόλυνση και την κατατόπιση του.

Ένα παράδειγμα του κώδικα για την εναλλαγή μηνυμάτων που αναφέραμε είναι τα παρακάτω. Η προγραμματιστική λογική αυτού του σημείου έχει αναλυθεί στα παραπάνω σενάρια.

```
TextString t1 ← (this.gopher) getStringFromUser ("Δήλωση μεθόδου με όνομα pol και παράμετρους a και b.")  
WholeNumber count1 ← 0
```

Εμφάνιση αρχικού μηνύματος

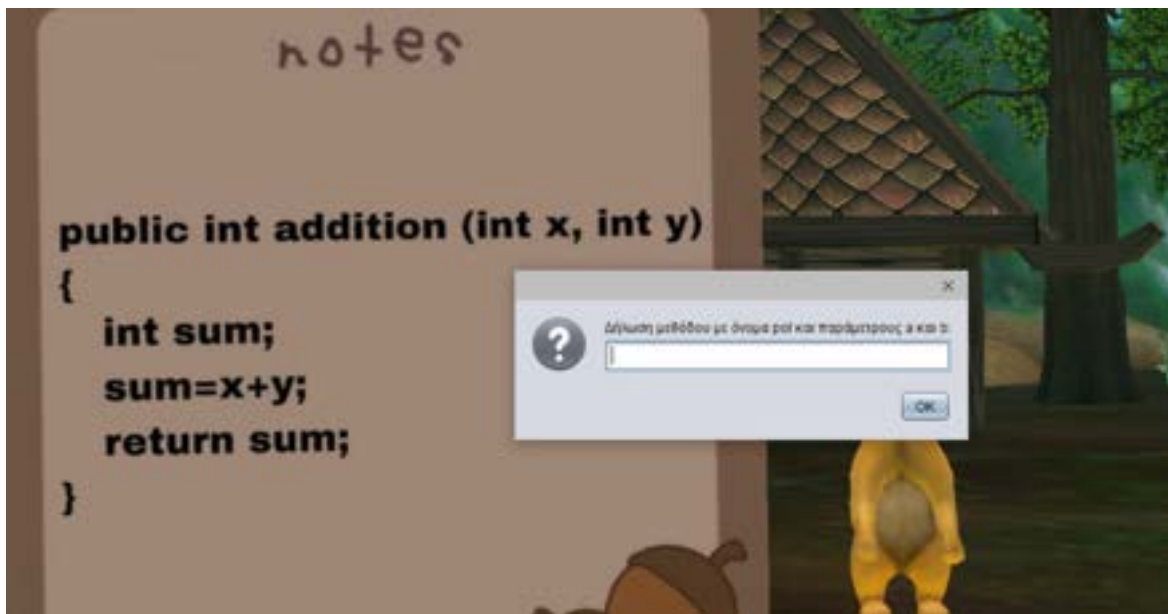
```
if (count1 == 1) is true then  
t1 ← (this.gopher) getStringFromUser ("Δήλωση μεθόδου με όνομα pol και παράμετρους a και b.")
```

Εμφάνιση μηνύματος στη δεύτερη προσπάθεια

```
else  
t1 ← (this) getStringFromUser ("Δοκίμασε να γράψεις public int pol(int a, int b)")
```

Εμφάνιση μηνύματος στη τρίτη προσπάθεια

Αναλυτικότερα παρουσιάζονται κάποια κομμάτια αλληλεπίδρασης του μαθητή με το παιχνίδι κατά τη διάρκεια δημιουργίας της μεθόδου.



int x, int y)

Χρησιμοποίησε το σύμβολο {

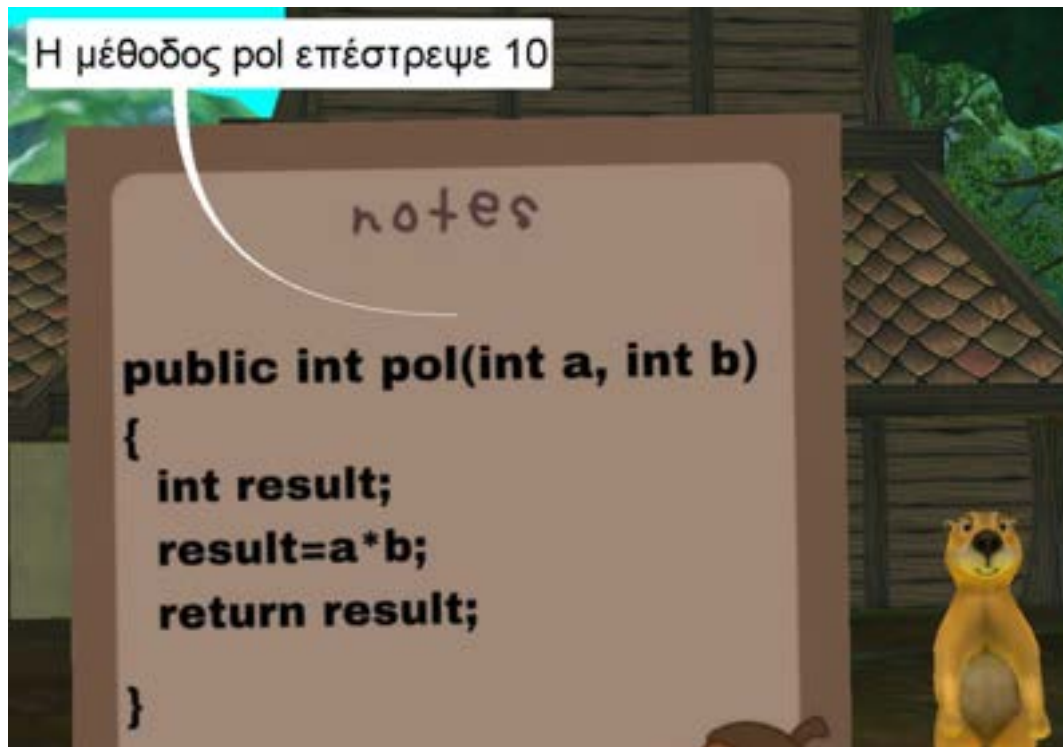
```
public int addition (int x, int y)  
{  
    int sum;  
    sum=x+y;  
    return sum;  
}
```

Δήλωσε ακέραιας μεταβλητής με όνομα result (Μην ξεχάσεις το γραμμικό στο τέλος)

```
public int addition (int x, int y)  
{  
    int sum;  
    sum=x+y;  
    return sum;  
}
```

Εκτέλεσε την πράξη που αποθήκευσε στην μεταβλητή result (Μην ξεχάσεις το γραμμικό στο τέλος)

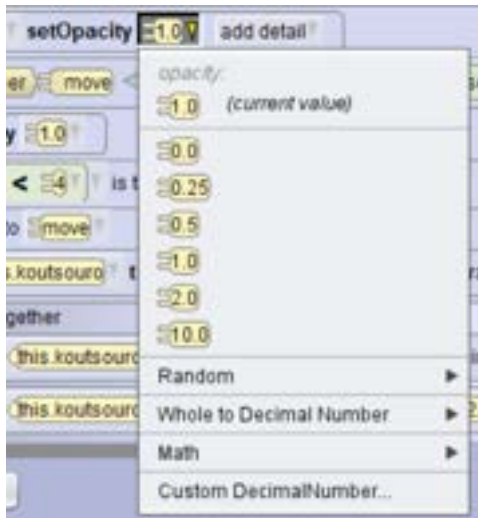
Αφότου ολοκληρώσει τη δημιουργία της νέας μεθόδου, εμφανίζεται στην οθόνη του ένα πινακάκι με τη μέθοδο που έφτιαξε. Σαν επόμενο στάδιο ο μαθητής μπορεί να πειραματιστεί με αυτήν δίνοντας της τιμές ώστε να δει και τι θα επιστρέψει κάθε φορά που την καλεί. Για παράδειγμα κατά την εκτέλεση του προγράμματος αν έχει επιλέξει να περάσει σαν παράμετρους τις τιμές 2 και 5, η μέθοδος pol θα υπολογίσει και θα εμφανίσει το αποτέλεσμα 10.



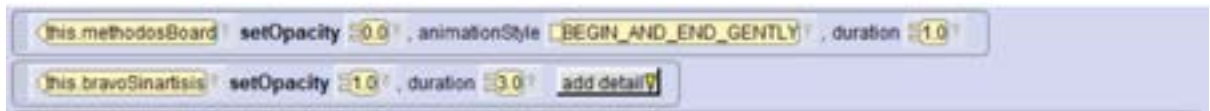
Ο μαθητής μετά την κλήση της μεθόδου του έχει ολοκληρώσει το εκπαιδευτικό σενάριο με επιτυχία. Ο χαρακτήρας κάστορας τον επιβραβεύει και ο μαθητής αποκτά το πράσινο διαμάντι του δάσους.



Για το αισθητικό κομμάτι της εφαρμογής έχουμε χρησιμοποιήσει σε πολλά αντικείμενα κινήσεις. Ένα παράδειγμα είναι η παρουσίαση του κειμένου και του διαμαντιού. Η εμφάνιση της λέξης “Μπράβο” γίνεται αρχικά με την επιλογή του **Text** αντικειμένου ορίζοντας στη συνέχεια το κείμενο που θέλουμε να εμφανίσουμε στον κόσμο. Έπειτα διαλέγοντας την κατάλληλη μέθοδο **setOpacity** μας δίνεται η δυνατότητα να επιλέξουμε την τιμή του opacity του αντικειμένου από 0.0 έως 1.0.



Επιλογή του opacity

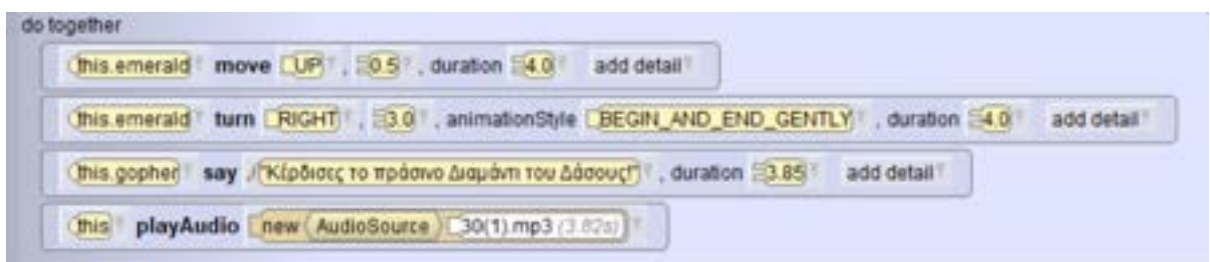


Κομμάτι κώδικα

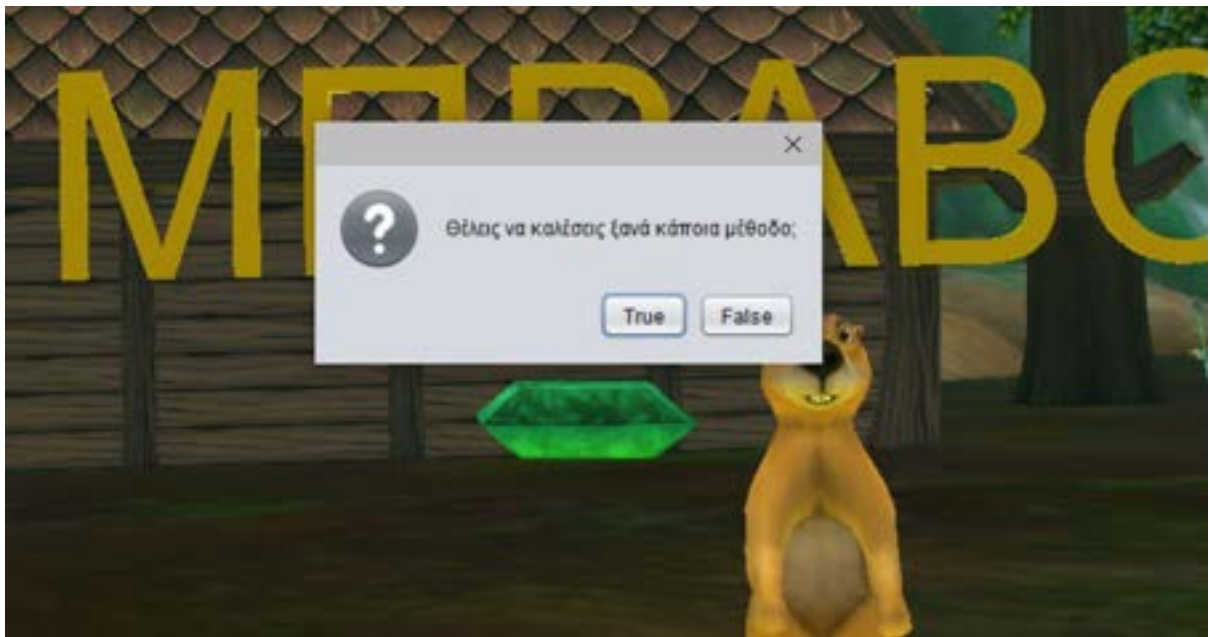
Με την ολοκλήρωση της εμφάνισης του κειμένου έχουμε προσθέσει κίνηση με φορά προς τα δεξιά εκτελώντας την εντολή **turn**.



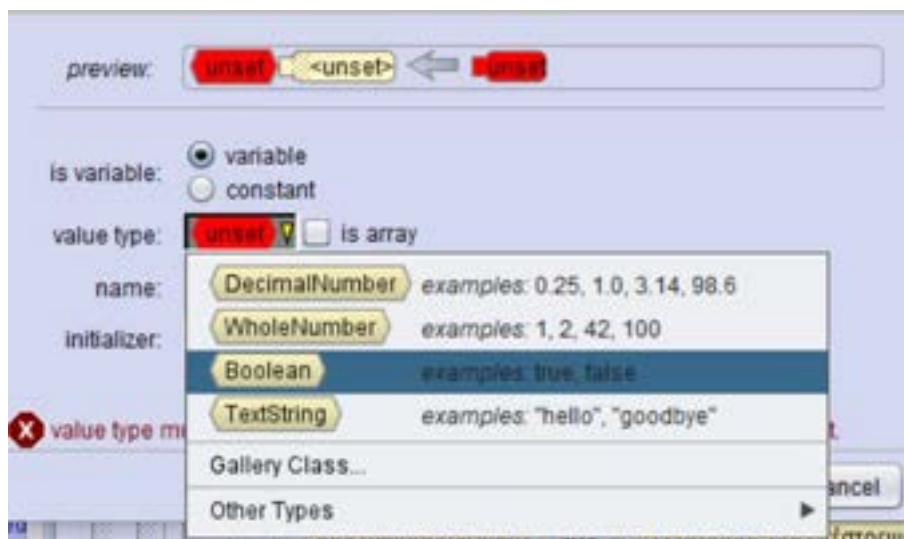
Η παρουσίαση του διαμαντιού γίνεται δεξιά του χαρακτήρα κάστορα. Το διαμάντι έχει τοποθετηθεί κάτω από το έδαφος του κόσμου με αποτέλεσμα όταν γίνεται η εμφάνιση του να κινείται από κάτω προς τα πάνω. Για να επιτευχθεί η παρουσίαση του διαμαντιού χρησιμοποιήθηκε σε ένα **do together** η μέθοδος **move** με κίνηση πάνω (up). Επιπλέον και για αυτό το αντικείμενο προστέθηκε η κίνηση με στροφή προς τα δεξιά με την εντολή **turn**.



Με το τέλος του σεναρίου δίνεται η δυνατότητα στο μαθητή να ξαναδεί και να ασχοληθεί με τις μεθόδους. Για αυτόν τον λόγο εμφανίζεται στην οθόνη του ένα αναδυόμενο παράθυρο με την ερώτηση αν θέλει να καλέσει ξανά κάποια μέθοδο.



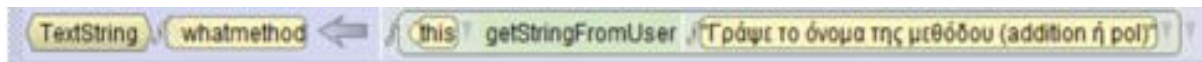
Για την εμφάνιση του παραθύρου με την επιλογή true ή false, τοποθετήθηκε μια μεταβλητή (variable) και ορίστηκε να έχει ως τύπο δεδομένων **Boolean**.



Στην περίπτωση που επιλέξει true εμφανίζεται ένα νέο παράθυρο για να πληκτρολογήσει ποια μέθοδο θέλει να καλέσει. Μετά την πληκτρολόγηση του ονόματος της μεθόδου ο μαθητής μπορεί να δει το πινακάκι με τη μέθοδο και να την καλέσει δίνοντας της διάφορες τιμές.



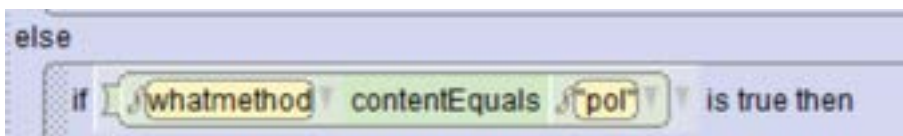
Από προγραμματιστική άποψη η επίτευξη του συγκεκριμένου κομματιού έγινε με τη βοήθεια της **while**. Έχοντας ως συνθήκη στην συγκεκριμένη δομή επανάληψης την **Boolean** μεταβλητή, όταν ο μαθητής επιλέξει true, έχουμε τοποθετήσει μέσα σε αυτήν δύο δομές ελέγχου με διαφορετικές συνθήκες. Οι δομές ελέγχου εκτελούνται ανάλογα με την τιμή που θα δώσει ο μαθητής σε μια μεταβλητή που ορίσαμε με τύπο δεδομένων **TextString**.



Η πρώτη περίπτωση είναι η δομή ελέγχου με συνθήκη ο μαθητής να έχει πληκτρολογήσει το όνομα της πρώτης μεθόδου και αντίστοιχα η δεύτερη περίπτωση είναι να έχει πληκτρολογήσει την δεύτερη μέθοδο.

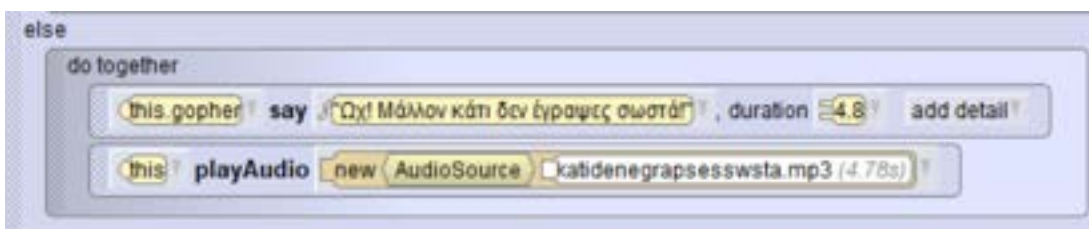


1η Συνθήκη

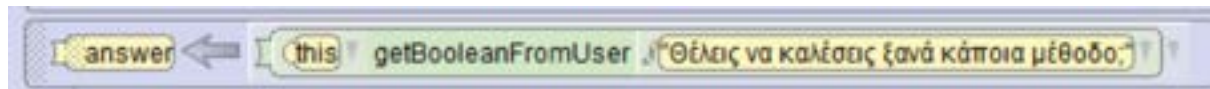


2η Συνθήκη

Σε οποιαδήποτε άλλη περίπτωση του εμφανίζεται το κατάλληλο μήνυμα λάθους.



Σε αυτό το σημείο είναι σημαντικό να τονιστεί ότι κάθε φορά μετά την εκτέλεση της μεθόδου εμφανίζεται ξανά ένα μήνυμα true ή false δίνοντας την ευκαιρία στο μαθητή να ξαναδεί τις μεθόδους όσες φορές επιθυμεί.



ΚΕΦΑΛΑΙΟ 6 Αξιολόγηση και Συμπεράσματα

(6.1) Αποδοτικότητα του προγράμματος Alice στην εκπαίδευση

Περισσότερα από 30 χρόνια έρευνας έχουν αφιερωθεί στην ανάπτυξη προγραμματιστικών συστημάτων και απευθύνονται σε μια ευρύτερη ομάδα ανθρώπων. Κύριο μέλημα τους είναι η συνεχής αναζήτηση καλύτερων τρόπων για να εμφυσήσουν βασικές δεξιότητες προγραμματισμού στους αυριανούς προγραμματιστές και όχι μόνο. Πολλές έρευνες έχουν δείξει ότι η χρήση του εκπαιδευτικού προγράμματος Alice έχει θετικά αποτελέσματα στην εκπαίδευση. Πιο συγκεκριμένα μια έρευνα που πραγματοποιήθηκε στο Miami University έδειξε ότι υπάρχει αύξηση του ποσοστού της ενασχόλησης των μαθητών με τον προγραμματισμό από 19% σε 48%, καθώς οι μαθητές φαίνεται να αισθάνονται περισσότερη ασφάλεια μετά τη χρήση του Alice. Ακόμα, η αυτοπεποίθηση τους για τον προγραμματισμό αυξήθηκε από 17% σε 29% και όλο και περισσότεροι μαθητές ανέφεραν ότι τους άρεσε η συγγραφή κώδικα σε αυτό. Σύμφωνα με μια άλλη μελέτη που έγινε στο Carnegie Mellon University, το εκπαιδευτικό πρόγραμμα έπαιξε σημαντικό ρόλο στην παρακίνηση των κοριτσιών να ασχοληθούν με τον προγραμματισμό. Ένα μεγάλο ποσοστό των κοριτσιών της τάξεως του γυμνασίου μετά τη δοκιμή του Alice εξέφρασαν ενδιαφέρον για την συνέχεια εκμάθησης του προγραμματισμού με το συγκεκριμένο εκπαιδευτικό λογισμικό. Το 83% τους κέντρισε την προσοχή και τους γεννήθηκε η επιθυμία να ακολουθήσουν τον κλάδο της πληροφορικής μελλοντικά. Τέλος, μέλη του Armstrong Atlantic State University ενώ έδειξαν ενθουσιασμό για το εκπαιδευτικό πρόγραμμα, διατύπωσαν τις αμφιβολίες τους για την αποδοτικότητα και τις προσεγγίσεις της εφαρμογής.



(6.2) Παρουσίαση Ερωτηματολόγιο

Αγαπητέ μαθητή,

Αγαπητή μαθήτριά,

Στο ερωτηματολόγιο αυτό, σου ζητάμε να εκφράσεις τη γνώμη σου για το εκπαιδευτικό πρόγραμμα Alice με το οποίο ασχολήθηκες.

Η συμπλήρωση του Ερωτηματολογίου διαρκεί περίπου 10 λεπτά.

ΔΕΝ υπάρχουν σωστές ή λάθος απαντήσεις.

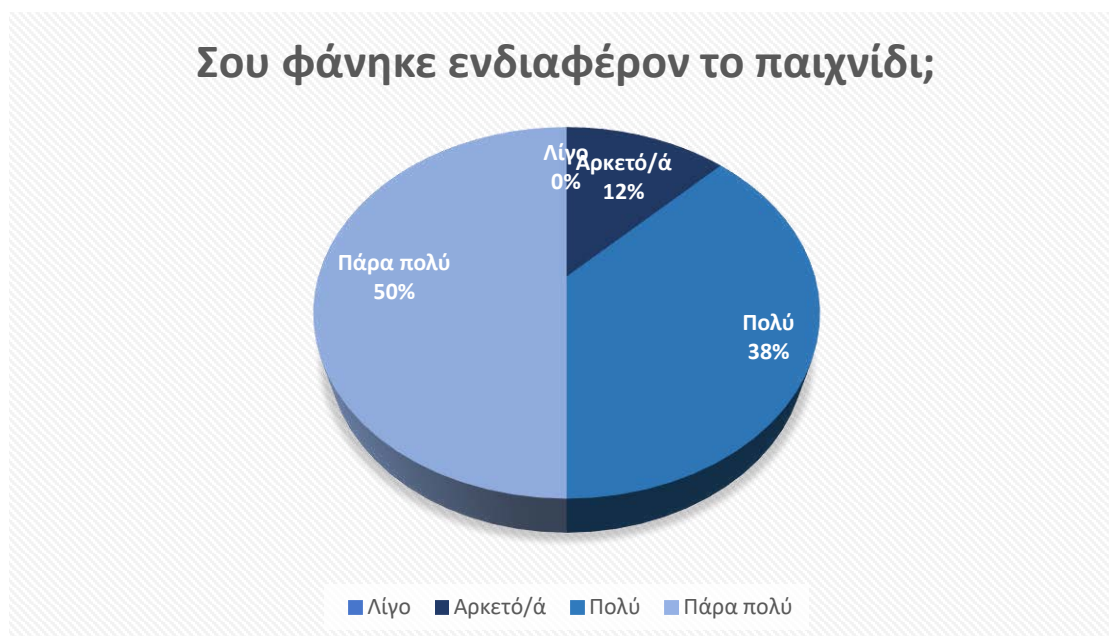
Το ερωτηματολόγιο αυτό είναι απόρρητο, για αυτό ΜΗΝ γράψεις το όνομά σου σε κανένα πεδίο.

Η συμπλήρωση του ερωτηματολογίου ΔΕΝ είναι υποχρεωτική. Εάν δεν νοιώθεις άνετα να απαντήσεις σε κάποια από τις ερωτήσεις, μπορείς να την παραλείψεις.

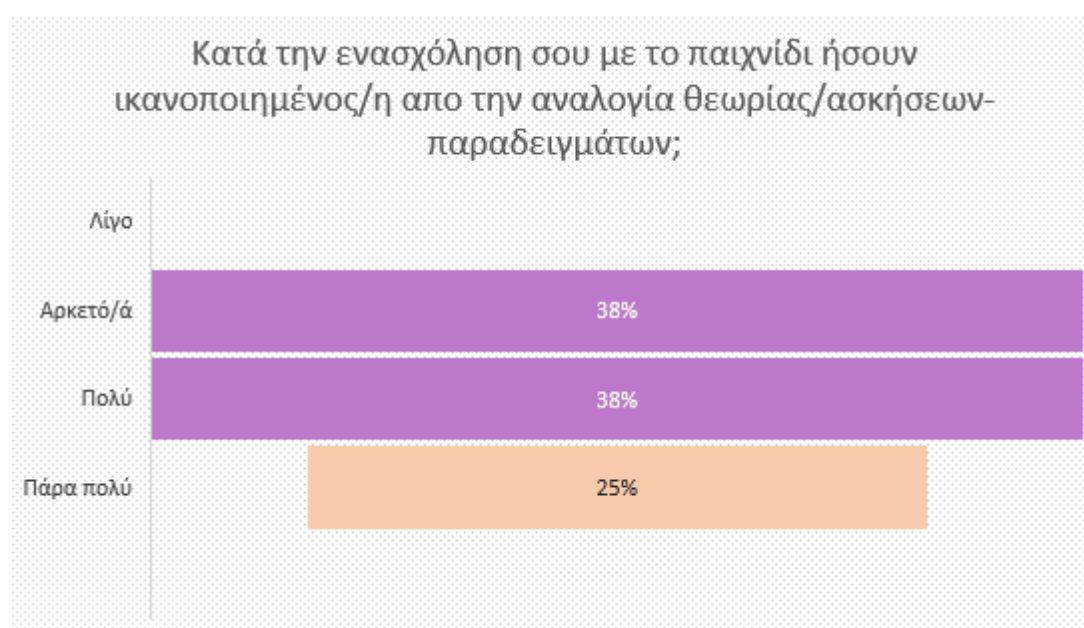
Κλίμακα αξιολόγησης		1.Λίγο	2.Αρκετό/ά	3.Πολύ	4.Πάρα πολύ
1	Σου φάνηκε ενδιαφέρον το παιχνίδι;				
2	Θεώρησες ότι ήταν κατανοητές οι οδηγίες;				
3	Βρήκες την καθοδήγηση από τους χαρακτήρες βοηθητική;				
4	Αντιλήφθηκες τον σκοπό κάθε σεναρίου;				
5	Κατά την ενασχόληση σου με το παιχνίδι ήσουν ικανοποιημένος/η από την αναλογία θεωρίας/ασκήσεων-παραδειγμάτων;				
6	Σου άρεσε ο τρόπος εκμάθησης της θεωρίας μέσα από το παιχνίδι;				
7	Σου άρεσε σαν εμπειρία;				
8	Θα ήθελες να ενταχθεί το Alice ή κάποιο άλλο εκπαιδευτικό πρόγραμμα στο μάθημα της πληροφορικής;				
9	Θα σε ενδιέφερε να συνεχίσεις να ασχολείσαι με το Alice;				

(6.3) Συμπεράσματα Ερωτηματολογίου

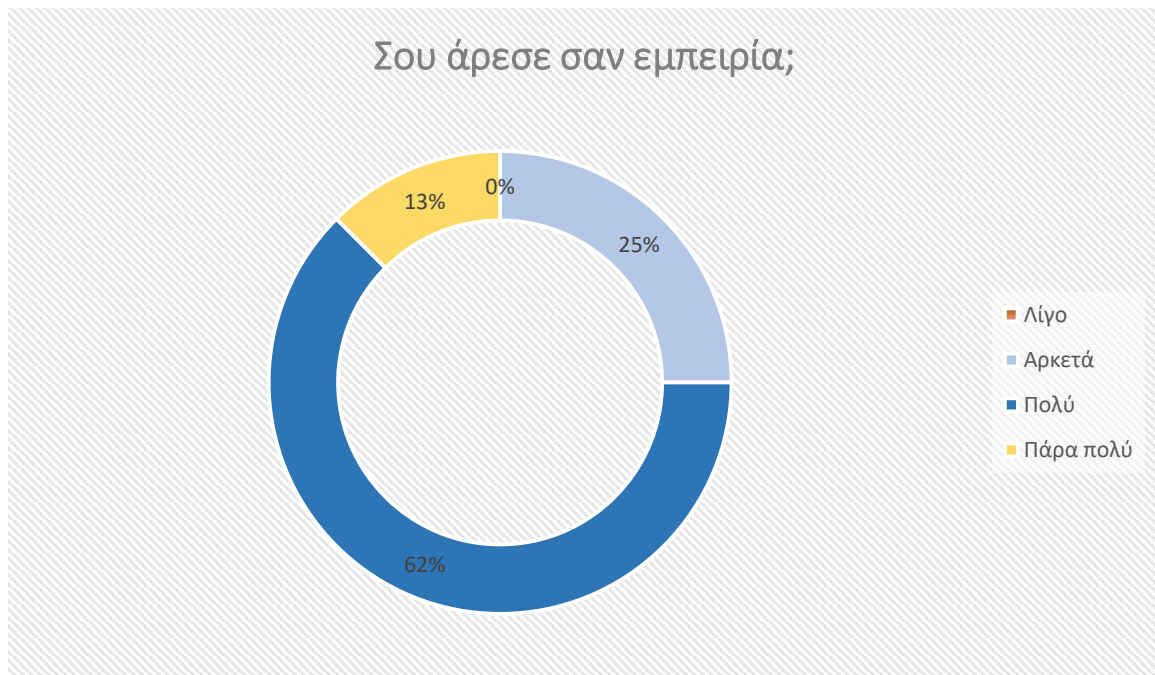
Με βάση το ερωτηματολόγιο που δόθηκε σε παιδιά διαφόρων ηλικιών κάτω των δεκατριών ετών προέκυψαν τα παρακάτω αποτελέσματα. Αρχικά ερωτήθηκαν αν τους φάνηκε ενδιαφέρον το παιχνίδι με το οποίο ασχολήθηκαν. Το 50% των συμμετεχόντων βρήκαν την εφαρμογή πάρα πολύ ενδιαφέρουσα ενώ ένα ποσοστό των 38% επέλεξαν την απάντηση πολύ.



Στην ερώτηση που ακολούθησε για το πόσο κατανοητές ήταν οι οδηγίες, η πλειοψηφία και πιο συγκεκριμένα το 63% απάντησε πάρα πολύ. Στη συνέχεια τόσο στην προηγούμενη ερώτηση όσο και στην επομένη που αφορά αν βρήκαν την καθοδήγηση από τους χαρακτήρες του παιχνιδιού βοηθητική το 38% των συμμετεχόντων απάντησε πολύ. Η επόμενη ερώτηση που κλήθηκαν να απαντήσουν τα παιδιά σχετικά με το αν αντιλήφθηκαν το σκοπό κάθε σεναρίου, έλαβε την απάντηση πάρα πολύ σε ποσοστό των 50%. Για την πέμπτη ερώτηση του ερωτηματολογίου τα αποτελέσματα των απαντήσεων απεικονίζονται στο παρακάτω διάγραμμα.



Το 50% των ερωτώμενων τους άρεσε ο τρόπος εκμάθησης της θεωρίας μέσα από το παιχνίδι καθώς επέλεξαν την απάντηση πάρα πολύ. Στην έβδομη ερώτηση τα αποτελέσματα που αναγράφονται στο ακόλουθο διάγραμμα ήταν ενθαρρυντικά.



Ιδιαίτερα σημαντικά θεωρήθηκαν τα στατιστικά που προέκυψαν από την όγδοη ερώτηση που αφορά την επιθυμία ένταξης του Alice ή κάποιου άλλου παρόμοιου εκπαιδευτικού προγράμματος στο μάθημα της πληροφορικής. Όλοι οι συμμετέχοντες ανταποκρίθηκαν θετικά και πιο συγκεκριμένα υπολογίστηκε το ίδιο ποσοστό των 50% τόσο για την απάντηση πολύ όσο και για την απάντηση πάρα πολύ. Τέλος, στην ερώτηση για το αν θα τους ενδιέφερε να συνεχίσουν την ενασχόληση τους με το Alice το 38% απάντησε αρκετά ενώ το 50% πάρα πολύ. Αναλυτικότερα οι απαντήσεις όλου του ερωτηματολογίου παρουσιάζονται στο παρακάτω διάγραμμα.



ΚΕΦΑΛΑΙΟ 7 Επίλογος

Μέσα από την πτυχιακή μας εργασία, μας δόθηκε η ευκαιρία για πρώτη φορά να ασχοληθούμε με το δωρεάν λογισμικό του Alice. Η εμπειρία μας καθόλη τη διάρκεια της ενασχόλησης και της γνωριμίας μας με το πρόγραμμα ήταν πολύ ευχάριστη και δημιουργική καθώς το Alice προσφέρει μια πληθώρα δυνατοτήτων τόσο σε προγραμματιστικό επίπεδο όσο και στη υλοποίηση ενός 3D περιβάλλοντος. Αρχικός στόχος για την εκπόνηση της πτυχιακής μας, ήταν η δημιουργία κάποιων εκπαιδευτικών σεναρίων, όμως μετά την καλύτερη εξοικείωση μας με το Alice μας γεννήθηκε η ιδέα για την δημιουργία ενός παιχνιδιού. Έτσι, υλοποιήθηκε το παιχνίδι “Ο Κόσμος του Κώδικα” με εισαγωγικές έννοιες του αντικειμενοστραφή προγραμματισμού που απευθύνεται σε μαθητές δημοτικής εκπαίδευσης. Έχει αποδειχθεί επιπλέον ότι το πρόγραμμα Alice μπορεί να είναι ωφέλιμο και για τους εκπαιδευτικούς πέρα από τους μαθητές. Βάση ερευνών το 58,7% των εκπαιδευτικών μετά την επιμόρφωση τους στο Alice παρουσίασε μεγάλο ενδιαφέρον για αυτό και σημειώθηκε κατάλληλο για την διδασκαλία αλγοριθμικών εννοιών και προγραμματισμού. Επίσης το 58% δήλωσαν ότι το εργαστήριο της Πληροφορικής του σχολείου τους ενδείκνυται για την εγκατάσταση του λογισμικού Alice.

Όραμα μας ήταν η εφαρμογή που δημιουργήσαμε να αποτελέσει εκπαιδευτική πρόταση και να μπορέσει να ενταχθεί στη διδακτική της πληροφορικής. Το παιχνίδι που παρουσιάσαμε σε αυτήν την πτυχιακή είναι μια πρωταρχική προσέγγιση και μπορεί να αποτελέσει θεμέλιο για τη δημιουργία και την εξέλιξη νέων εκπαιδευτικών σεναρίων. Ωστόσο, κατά τη διάρκεια ασχολίας με το Alice είναι πολύ πιθανόν να εμφανιστούν κάποιες προκλήσεις που μπορούν να δυσκολέψουν τη διαδικασία του προγραμματισμού. Από την προσωπική μας ενασχόληση, αντιμετωπίσαμε ζητήματα στο κομμάτι τοποθέτησης πολλαπλών αντικειμένων στο περιβάλλον του Alice καθώς η ικανότητα αναγνώρισης και επιλογής των αντικειμένων μειωνόταν σε μεγάλο βαθμό. Τέλος, η εφαρμογή μας μπορεί να εξελιχθεί προσθέτοντας είτε νέους κόσμους με υψηλότερο βαθμό δυσκολίας είτε δημιουργώντας ένα δεύτερο παιχνίδι που έχει ως σκοπό την εκμάθηση πιο εξειδικευμένων εννοιών.

Βιβλιογραφία

- BIBLIOGRAPHY Firat, Y. (2018). *The importance of coding and impacts to a country development in a computer technology teaching*. Istanbul.
- Folk, M. (1981). Review of "Mindstorms: Children, Computers, and Powerful Ideas by Seymour Papert", Basic Books: New York, 1980. *ACM Sigcue Outlook*, 15(1), 23-24. Retrieved 3 19, 2022, from <https://dl.acm.org/citation.cfm?id=1045074>
- Wanda Dann, S. C. (2009, August). Education: Alice 3: Concrete to Abstract. *Communications of the ACM*, pp. 27-29.
- Μπράττισης, Θ. (2013). *Η Πληροφορική στο Ελληνικό Σχολείο: Τάσεις, προσεγγίσεις, προοπτικές, 111-115*. Retrieved from <http://earthlab.uoi.gr/thete/index.php/thete>
- Μωράκης Διονύσιος, Ζ. Κ. (2017). *Εκπαιδευτικό περιβάλλον προγραμματισμού Alice. Καταγραφή χαρακτηριστικών και αξιολόγηση επιμορφωτικού σεμιναρίου*. Αθήνα .
- Sigcue Outlook*, 15(1), 23-24. Retrieved 3 19, 2022, from <https://dl.acm.org/citation.cfm?id=1045074>
- Wanda Dann, S. C. (2009, August). Education: Alice 3: Concrete to Abstract. *Communications of the ACM*, pp. 27-29.
- Μπράττισης, Θ. (2013). *Η Πληροφορική στο Ελληνικό Σχολείο: Τάσεις, προσεγγίσεις, προοπτικές, 111-115*. Retrieved from <http://earthlab.uoi.gr/thete/index.php/thete>
- Courte, Jill, Elizabeth V. Howard, and Cathy Bishop-Clark. "Using Alice in a computer science survey course." *Director 7* (2006).
- Cooper, Stephen, Wanda Dann, and Randy Pausch. "Developing algorithmic thinking with Alice." *The proceedings of ISECON*. Vol. 17. 2000.
- "Alice – Tell Stories. Build Games, from <http://www.alice.org/>
- Kim, Jackie Hee-Young, et al. "Determining the Effectiveness of the 3D Alice Programming Workshop for At-risk High School Students." *34thannual*: 388.
- Kelleher, Caitlin, Randy Pausch, and Sara Kiesler. "Storytelling alice motivates middle school girls to learn computer programming." *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2007.