Alice 3
Δημιουργία εικονικών κόσμων

Εφαρμογή του σεναρίου του σχολικού βιβλίου της Α’ ΓΕΛ

Ευτύχιος Δ. Γλαμπεδάκης
Υπεύθυνος ΚΕ.ΠΛΗ.ΝΕ.Τ. Ανατ. Θεσ/νίκης

Τι είναι η Alice

- Ένα προγραμματιστικό περιβάλλον για τη δημιουργία εφαρμογών εικονικών κόσμων
- Με αυτό οι μαθητές μπορούν να έχουν μια πρώτη επαφή με:
  - Τη δημιουργία σεναρίων
  - Την ανάλυση προγραμματιστικών απαιτήσεων
  - Τον αντικειμενοστραφή προγραμματισμό
  - Τον οπτικό προγραμματισμό
  - Και φυσικά με όλες τις τεχνικές του δομημένου προγραμματισμού (ακολουθία, επιλογή, επανάληψη, υποπρογράμματα)

Μέσα από ένα ευχάριστο εργαλείο που δίνει την αίσθηση παιχνιδιού.

© 2014 - Ε. Γλαμπεδάκης

Ξεκινώντας

- Η δημιουργία μιας εφαρμογής που κινεί αντικείμενα σε έναν εικονικό κόσμο, είναι μια διαδικασία που απαιτεί τα ακόλουθα στάδια:
  - Δημιουργία σεναρίου (περιγραφή προβλήματος)
  - Σχεδιασμός ενεργειών (ανάλυση)
  - Υλοποίηση (συγγραφή προγράμματος)
  - Δοκιμή (έλεγχος σωστής λειτουργίας)

© 2014 - Ε. Γλαμπεδάκης

Δημιουργία σεναρίου

Οι ερωτήσεις που πρέπει να απαντηθούν για να κατασκευαστεί ένα σενάριο είναι:

- Ποια είναι η «ιστορία» που πρέπει να παρασταθεί;
- Τι αντικείμενα χρειάζονται;
  - Μερικά αντικείμενα θα παίξουν ηγετικό ρόλο στην ιστορία και μερικά άλλα θα χρησιμοποιηθούν ως μέρος του σκηνικού.
- Τι ενέργειες πρέπει να γίνουν;
  - Οι ενέργειες στην «ιστορία» θα αποτελέσουν τις πληροφορίες του προγράμματος.

© 2014 - Ε. Γλαμπεδάκης

Σενάριο σχολικού βιβλίου Α’ ΓΕΛ

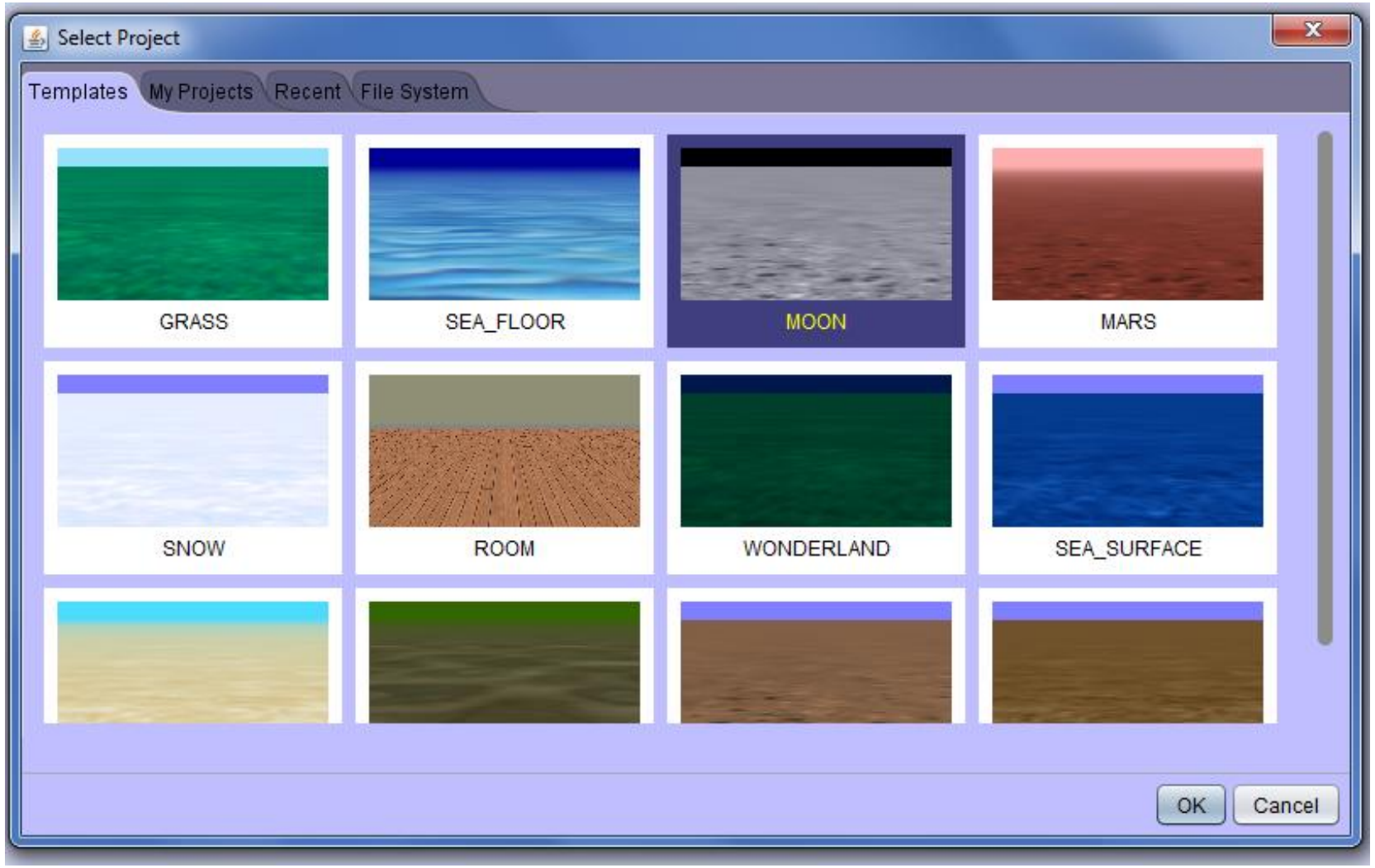
- Ένας αστροναύτης (AdultPerson) στη Σελήνη, αντιλαμβάνεται την παρουσία ενός σκάφους και στρέφεται προς εκείνη την κατεύθυνση.
- Το σκάφος είναι Άγνωστης Ταυτότητας Ιπτάμενο Αντικείμενο (UFO).
- Το σκάφος πλησιάζει και απ’ αυτό κατεβαίνει ένας εξωγήινος (Alien) και πλησιάζει τον αστροναύτη.
- Ο αστροναύτης καλεί σε βοήθεια το κέντρο ελέγχου και ζητάει από τον χρήστη να τον απομακρύνει από τον εξωγήινο .
- Αφού ο διάλογος μεταξύ τους δεν έχει κανένα αποτέλεσμα, ο εξωγήινος απαγάγει τον αστροναύτη στο σκάφος και αναχωρούν για τον πλανήτη του.

© 2014 - Ε. Γλαμπεδάκης

Βήμα 1: Επιλογή σκηνής (scene)

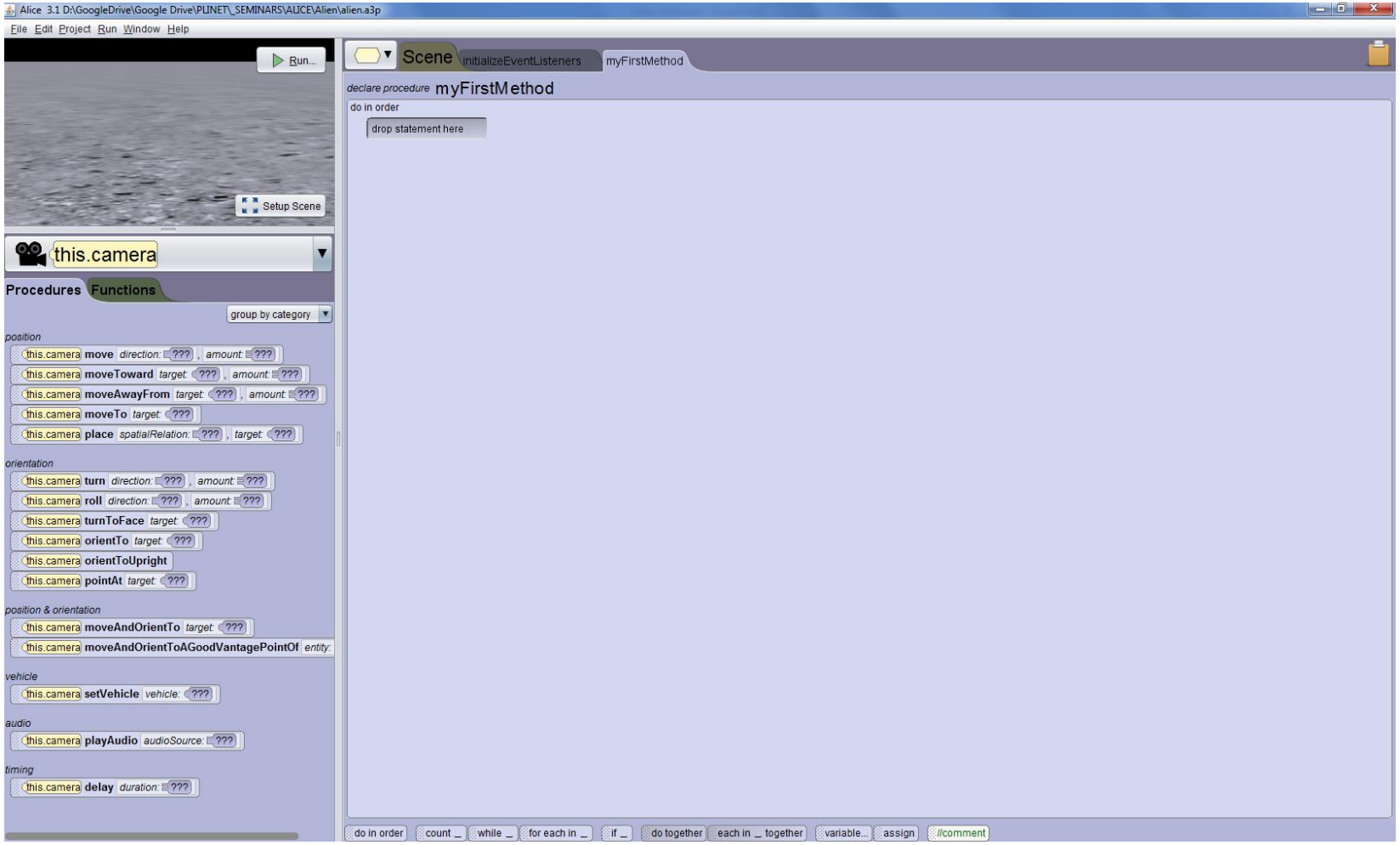
- Εκτελούμε την Alice και από το παράθυρο Select project που εμφανίζεται πρώτο, στην καρτέλα templates επιλέγουμε ως σκηνή (background) την επιλογή MOON (σελήνη)
- ΠΡΟΣΟΧΗ: Επειδή η Alice χρησιμοποιηθεί το Java Runtime Enviroment, μπορεί ανά πάσα στιγμή να σταματήσει να λειτουργεί (να κολλήσει)
- Γι’ αυτό καλό είναι να αποθηκεύουμε κάθε βήμα της εργασίας μας. (Μενού File → Save)
- Η κατάληξη των αρχείων της Alice 3, είναι .a3p

© 2014 - Ε. Γλαμπεδάκης



© 2014 - Ε. Γλαμπεδάκης

Βασικό περιβάλλον σχεδίασης του κόσμου και προγραμματισμού συμπεριφορών



© 2014 - Ε. Γλαμπεδάκης

Ανάπτυξη εφαρμογής

- Η Alice παρέχει ένα περιβάλλον οπτικού προγραμματισμού με αντικειμενοστραφή σχεδίαση.
- Η αντικειμενοστραφής σχεδίαση μοντελοποιεί το λογισμικό με όρους παρόμοιους εκείνων που χρησιμοποιούν οι άνθρωποι για να περιγράψουν πραγματικά αντικείμενα
- Στη Java (η Alice είναι java based περιβάλλον):
  - «Μονάδα προγραμματισμού» είναι η **κλάση**
  - Από μια κλάση προκύπτουν στιγμιότυπα αντικειμένων
  - Οι κλάσεις περιέχουν **μεθόδους** (που υλοποιούν λειτουργίες-συμπεριφορές)
  - Και **πεδία** (που υλοποιούν χαρακτηριστικά)
- Ο προγραμματιστής
  - Δημιουργεί νέες κλάσεις και χρησιμοποιεί στιγμιότυπά τους (υπαρχόντων ή νέων κλάσεων)
  - Κάθε κλάση περιέχει πεδία και μεθόδους που χειρίζονται αυτά τα πεδία και παρέχουν υπηρεσίες

© 2014 - Ε. Γλαμπεδάκης

Μια μικρή επανάληψη στον αντικειμενοστραφή προγραμματισμό (1/2)

Κάθε αντικείμενο :

- Έχει χαρακτηριστικά (attributes ► fields)
  - Μέγεθος, Σχήμα, Χρώμα, Βάρος, Ύψος, Πυκνότητα, Ηλικία, κλπ
- Υλοποιεί συμπεριφορές (behaviors ► methods)
  - Μια μπάλα κυλά, αναπηδά, φουσκώνει, ξεφουσκώνει…
  - Ένα μωρό κλαίει, κοιμάται, μπουσουλάει, περπατά, ανοιγοκλείνει τα μάτια…
  - Ένα αυτοκίνητο επιταχύνει, φρενάρει, στρέφεται…
  - Μια πετσέτα απορροφά νερό…
  - Οι δοκοί φορτίζονται, παραμορφώνονται…
  - Τα κτίρια διεγείρονται σεισμικά, ταλαντώνονται…
  - Οι δεξαμενές γεμίζουν ή αδειάζουν…

© 2014 - Ε. Γλαμπεδάκης

Μια μικρή επανάληψη στον αντικειμενοστραφή προγραμματισμό (2/2)

- Διαφορετικά αντικείμενα:
  - μπορεί να έχουν:
    - Παρόμοια χαρακτηριστικά
  - μπορεί να παρουσιάζουν:
    - Παρόμοιες συμπεριφορές
- Παραδείγματα:
    - Αυτοκίνητα – Πλοία
    - Άνθρωποι - Ζώα

© 2014 - Ε. Γλαμπεδάκης

Κλάσεις (1/2)

- Μια κλάση ορίζει ένα συγκεκριμένο είδος αντικειμένου.
- Στην Alice, οι κλάσεις είναι προκαθορισμένα 3D μοντέλα και βρίσκονται στην βιβλιοθήκη, όπου και κατηγοριοποιούνται σε ομάδες όπως Animals, People, Buildings, Sets and Scenes, Space κτλ.
- Σημειώστε ότι το όνομα της κλάσης ξεκινάει με κεφαλαίο.

© 2014 - Ε. Γλαμπεδάκης

Κλάσεις (2/2)

- Κάθε κλάση περιέχει τον κατάλληλο κώδικα που λέει στην Alice ακριβώς πώς να δημιουργήσει και να εμφανίσει ένα αντικείμενο της κλάσης.
- Όταν ένα αντικείμενο δημιουργηθεί και χρησιμοποιηθεί, ονομάζεται **στιγμιότυπο** (instance) της κλάσης.
- Το όνομα ενός αντικειμένου ξεκινάει με μικρό γράμμα
- Όλα τα αντικείμενα της ίδιας κλάσης έχουν ομοιότητες.

© 2014 - Ε. Γλαμπεδάκης

Ιδιότητες αντικειμένων κλάσεων

- Όλα τα αντικείμενα της κλάσης Person έχουν ιδιότητες όπως δυο πόδια, δυο χέρια, ύψος και χρώμα ματιών.
- Τα αντικείμενα - Person μπορούν να κάνουν ενέργειες όπως περπάτημά και ομιλία.
- Όλα τα αντικείμενα της κλάσης Dog έχουν ιδιότητες όπως τέσσερα πόδια, ύψος, χρώμα τριχώματος και έχουν την ικανότητα να τρέξουν και να γαβγίσουν.
- Παρόλο που κάθε αντικείμενο ανήκει σε μια κλάση, είναι μοναδικό με τον δικό του τρόπο:
  - Ο joe είναι ψηλός και έχει πράσινα μάτια.
  - Η cindy είναι κοντή και έχει μπλε μάτια.
  - Ο spike έχει καφέ τρίχωμα και το γάβγισμά του είναι αδύνατο γρύλισμα
  - Ο scamp έχει χρυσαφί τρίχωμα και το γάβγισμα του είναι τσιριχτό.

© 2014 - Ε. Γλαμπεδάκης

Μέθοδοι

- Οι μέθοδοι είναι μια συντονισμένη ακολουθία εντολών που μπορεί να εκτελεστεί όταν ζητηθεί.
- Κάθε αντικείμενο στην Alice έχει ένα ρεπερτόριο εντολών που ξέρει πως θα τις εκτελέσει (move, turn, turn to face, κτλ).
- Αυτές οι εντολές είναι στην πραγματικότητα οι αρχικές μέθοδοι που είναι ενσωματωμένες στο λογισμικό.
- Οι αρχικές μέθοδοι μπορούν να συνδυαστούν σε μια νέα μέθοδο (για να εκτελεστεί ένα μικρό κομμάτι του προγράμματος).
- Οι μέθοδοι new (δηλ. αυτές που το όνομα τους αρχίζει από new) ουσιαστικά δημιουργούν αντικείμενα από τις κλάσεις.
  - Στο σενάριο μας, η newAdult θα δημιουργήσει τον αστροναύτη στο επόμενο βήμα.

© 2014 - Ε. Γλαμπεδάκης

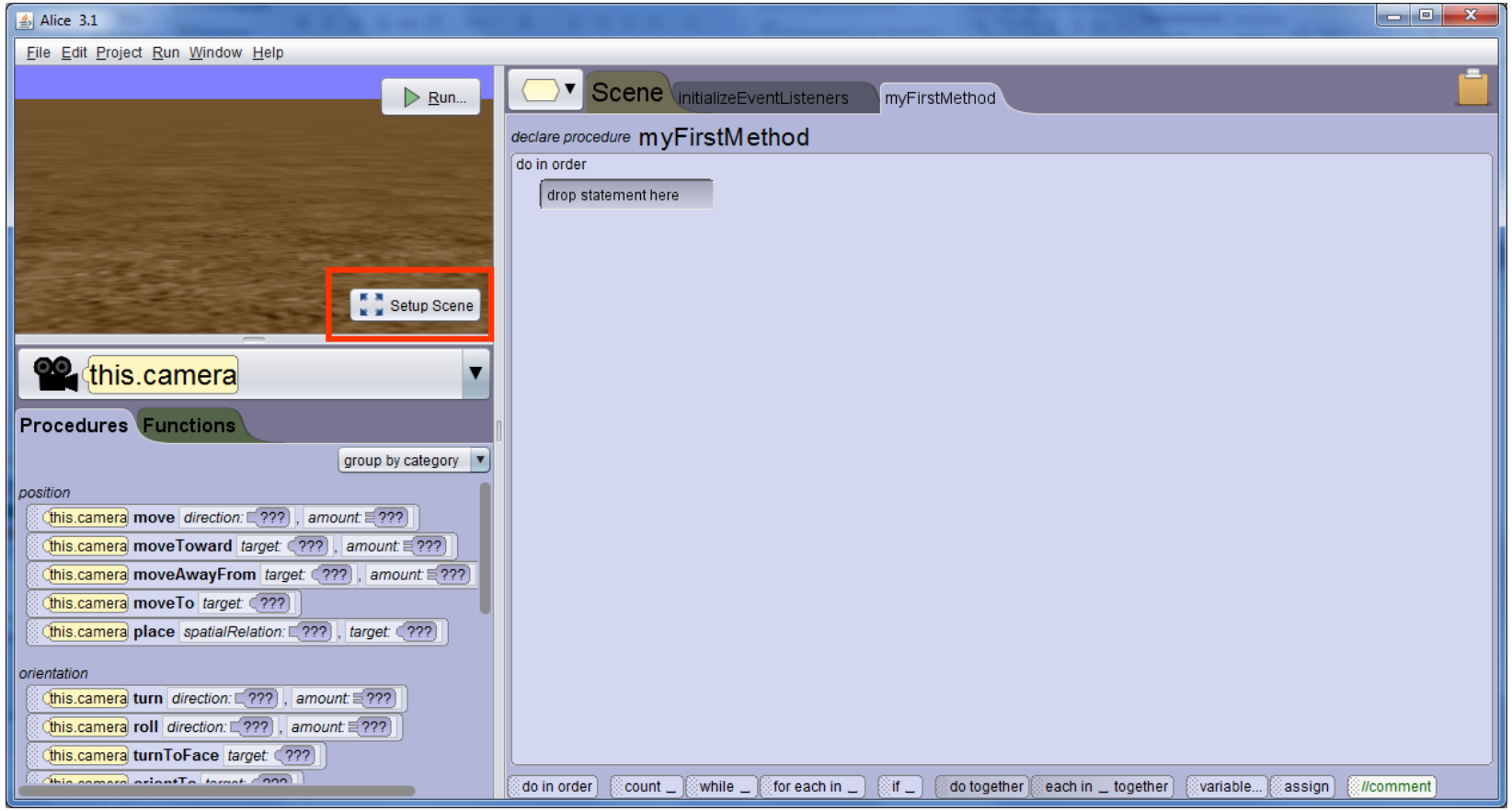
Εμβέλεια μεθόδων

- Στην Alice, μπορείτε να ορίσετε μεθόδους για ένα μόνο αντικείμενο, ή για δυο ή περισσότερα αντικείμενα που αλληλοεπιδρούν.
- Αυτό μοιάζει με τον τρόπο που δουλεύει ένας σκηνοθέτης με το σύνολο των ηθοποιών σε ένα έργο. Ο σκηνοθέτης δίνει οδηγίες κάποιες φορές σε ένα μόνο ηθοποιό και άλλες φορές σε μια ομάδα ηθοποιών ώστε να συνεργαστούν για την σκηνή.
- Οι μέθοδοι που αναφέρουν περισσότερα από ένα αντικείμενα είναι μέθοδοι επιπέδου κόσμου.
- Οι μέθοδοι που καθορίζουν την συμπεριφορά ενός μόνο αντικειμένου μπορούν να θεωρηθούν μέθοδοι επιπέδου κλάσης.

© 2014 - Ε. Γλαμπεδάκης

Προσθήκη αντικειμένων στον χώρο

- Κλικ στο Setup Scene



© 2014 - Ε. Γλαμπεδάκης

Βήμα 2: Προσθήκη αστροναύτη

- Εισάγουμε τον αστροναύτη από την **κλάση** Biped, χρησιμοποιώντας τη **μέθοδο** new Adult(…)
- Αφού του φορέσουμε τη στολή, τον τοποθετούμε στη σκηνή μπροστά και δεξιά. Τον μετονομάζουμε σε astronaut

© 2014 - Ε. Γλαμπεδάκης

© 2014 - Ε. Γλαμπεδάκης

© 2014 - Ε. Γλαμπεδάκης

Βήμα 3: Προσθήκη εξωγήινου

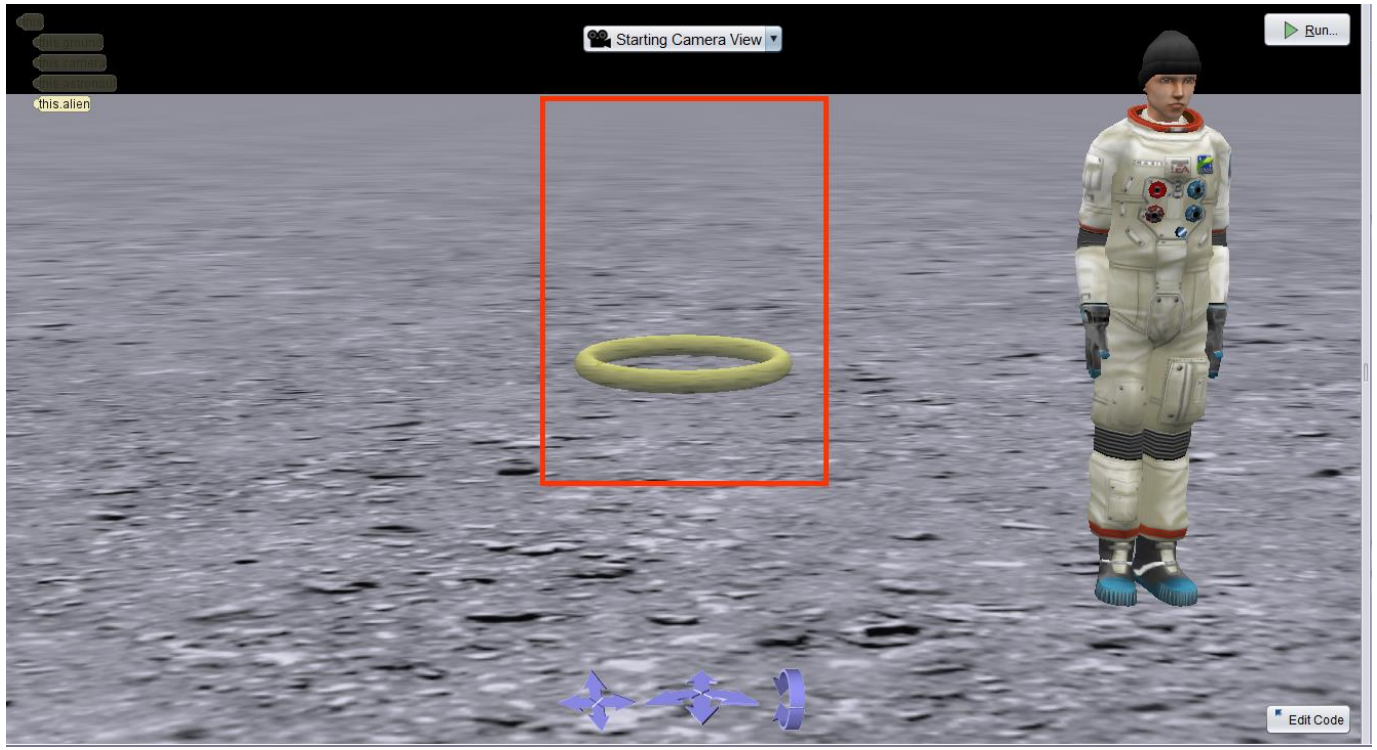
- Εισάγουμε τον εξωγήινο από την κλάση Biped, χρησιμοποιώντας την μέθοδο new Alien()
- Τον τοποθετούμε στη μέση της σκηνής και ορίζουμε την ιδιότητα Opacity (αδιαφάνεια) σε 0.0
- Μεταβάλλοντας αυτή την ιδιότητα, αρχικά ο alien δεν φαίνεται στη σκηνή, με σκοπό να μπορέσουμε να τον εμφανίσουμε αργότερα και να δημιουργείται η εντύπωση ότι βγαίνει από το UFO

© 2014 - Ε. Γλαμπεδάκης

© 2014 - Ε. Γλαμπεδάκης

Αλλαγή ιδιοτήτων αντικειμένου

- Αλλαγή διαφάνειας (opacity) εξωγήινου

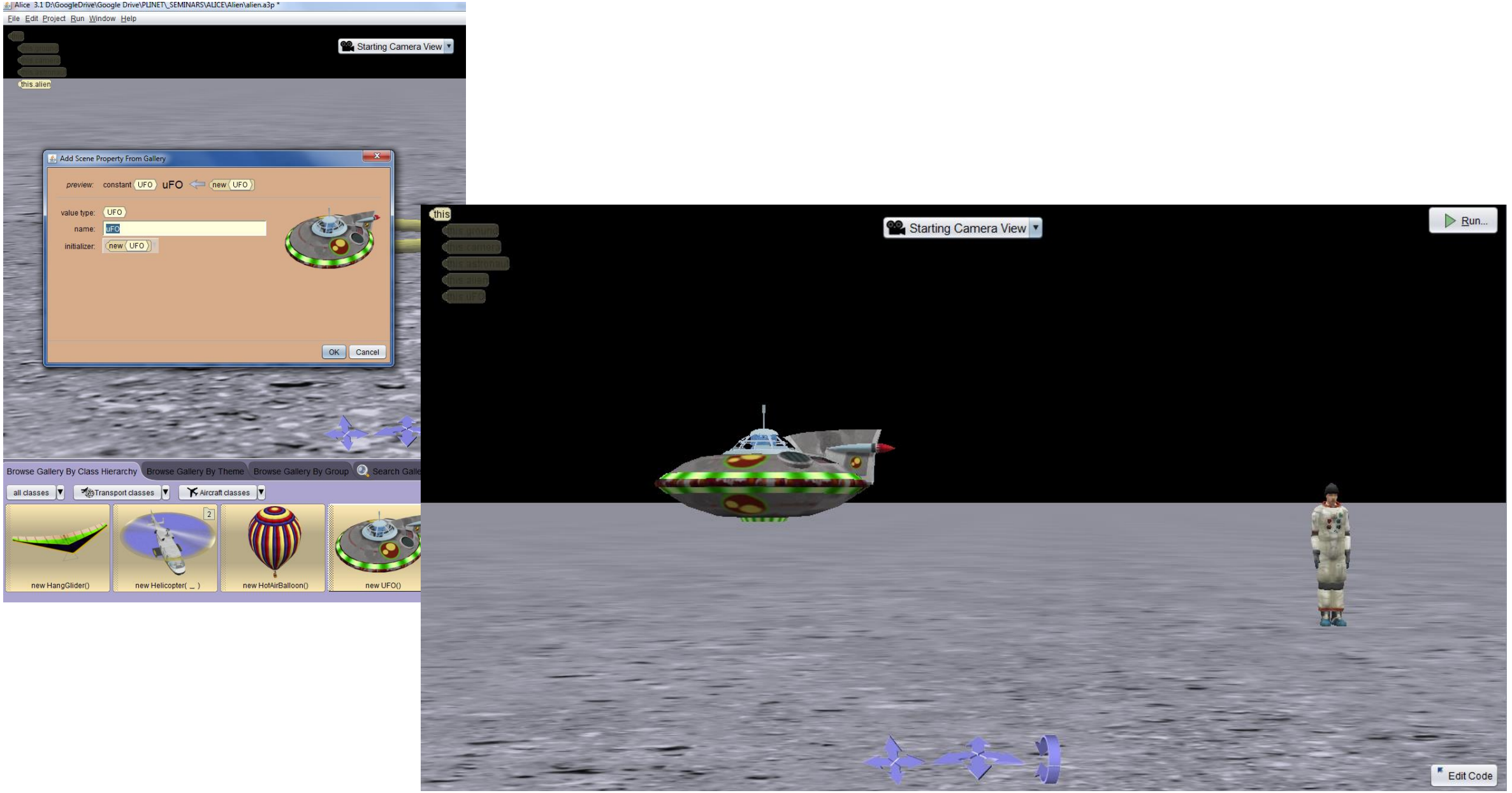


© 2014 - Ε. Γλαμπεδάκης

Βήμα 4: Προσθήκη UFO

- Εισάγουμε το UFO από την κλάση Transport και την υποκλάση Aircraft, χρησιμοποιώντας τη μέθοδο new UFO()
- Τον τοποθετούμε στη σκηνή πίσω και αριστερά.

© 2014 - Ε. Γλαμπεδάκης

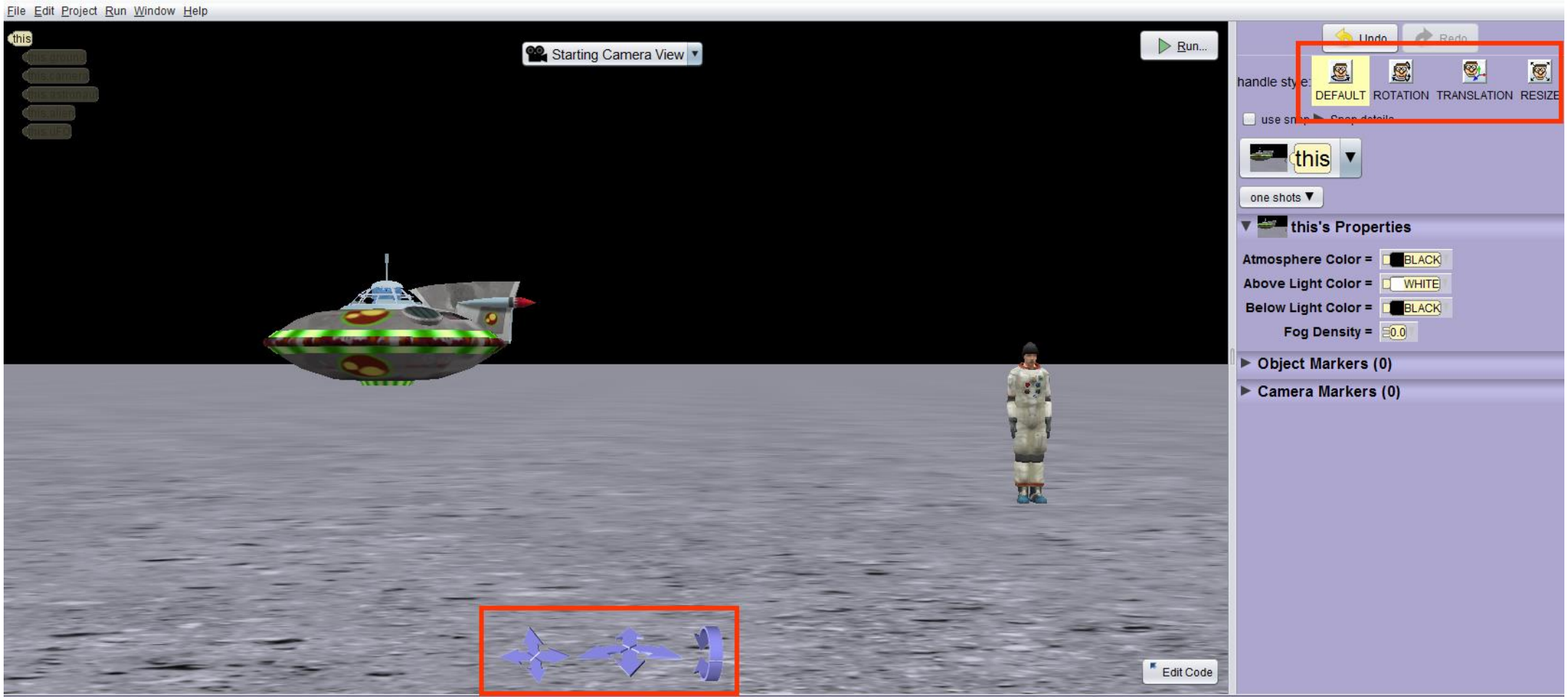


© 2014 - Ε. Γλαμπεδάκης

Βήμα 5: Θέση και μέγεθος αντικειμένων

- Μπορούμε να μετακινήσουμε, περιστρέψουμε, ανυψώσουμε, αλλάξουμε μέγεθος κ.ά. στα αντικείμενά μας:
  - με τις επιλογές Default, Rotation, Translation, Resize, επάνω δεξιά
  - και με τα μπλε βελάκια που βρίσκονται μπροστά και στο κέντρο της σκηνής

© 2014 - Ε. Γλαμπεδάκης



© 2014 - Ε. Γλαμπεδάκης

Βήμα 6: Εφαρμογή συμπεριφορών σε αντικείμενα (σύνταξη κώδικα)

- Για να επιστρέψουμε στο αρχικό παράθυρο και να μπορέσουμε να γράψουμε τον κώδικα του σεναρίου μας, πατάμε το κουμπί Edit Code

© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.1: Συμπεριφορές “say” & “think”

- Ο astronaut «μιλάει» και «σκέφτεται» το αντίστοιχο κείμενο που είναι γραμμένο.

© 2014 - Ε. Γλαμπεδάκης

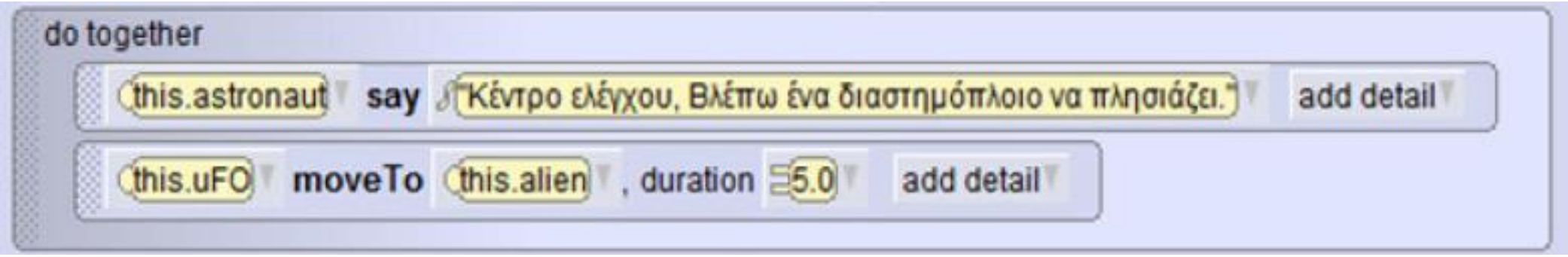
Βήμα 7.2: Συμπεριφορά “turn”

- Ο astronaut στρίβει προς μια κατεύθυνση
- Το 1.0 είναι μια πλήρης περιστροφή 360^{o}

© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.3: Ταυτόχρονη εφαρμογή συμπεριφορών (do together)

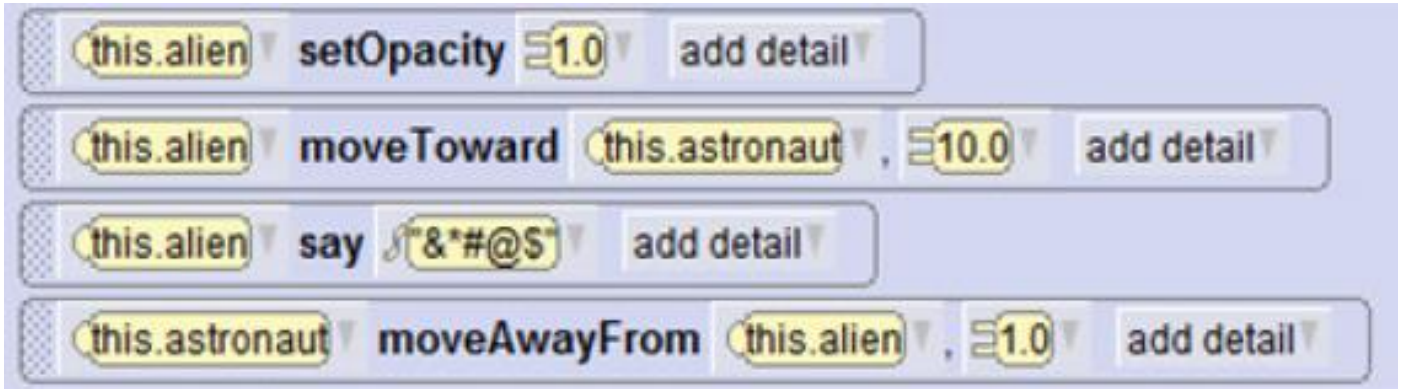
- Οι συμπεριφορές “**say**” και “**moveTo**” με τη χρήση της μεθόδου “**do together**” εκτελούνται ταυτόχρονα.
- Η “**moveTo**” μετακινεί ένα αντικείμενο από την τρέχουσα θέση του μέχρι τη θέση ενός άλλου αντικειμένου
- Η διάρκεια της κίνησης (duration) είναι σε δευτερόλεπτα.



© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.4: Πρόοδος σεναρίου

- Υπενθύμιση: Για να κρύψουμε τον alien όταν τον τοποθετήσαμε στη σκηνή είχαμε ορίσει την αδιαφάνεια του (opacity) σε 0.0
- Εμφανίζεται ο alien. Μέθοδος **setOpacity=1.0**
- Μετακινείται μπροστά στον astronaut. Μέθοδος **moveToward**. Η παράμετρος 10.0 δηλώνει την απόσταση μετακίνησης. Την μεταβάλλουμε κατάλληλα για να ταιριάζει στη σκηνή μας.
- Του μιλάει. Μέθοδος **say**.
- Ο astronaut απομακρύνεται προς τα πίσω. Μέθοδος **moveAwayFrom**.



© 2014 - Ε. Γλαμπεδάκης

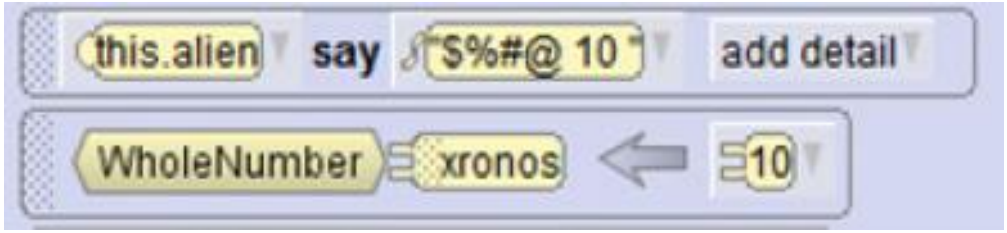
Βήμα 7.5: Η δομή επανάληψης **count up to**

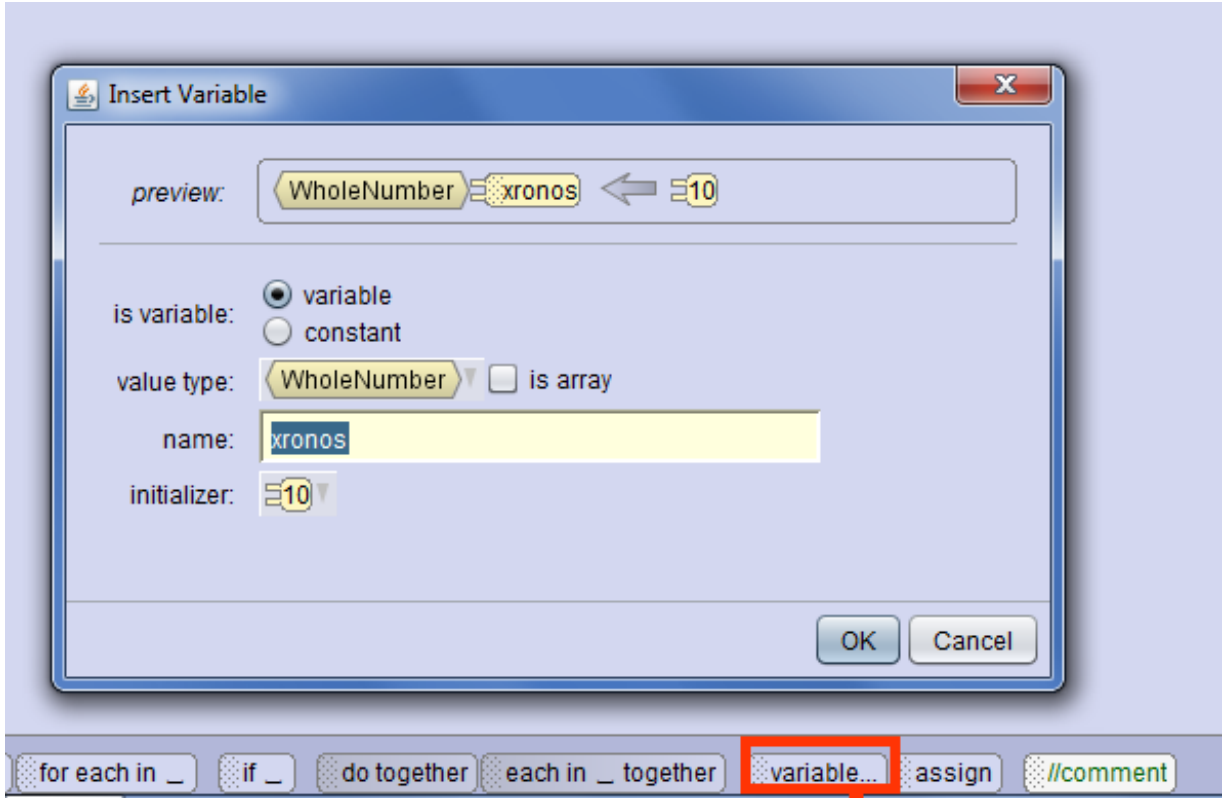
- Ο αστροναύτης καλεί σε βοήθεια το κέντρο ελέγχου
- Επανάληψη της κλήσης 2 φορές (με τη δομή count up to)

© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.6: Ορισμός αρχικής τιμής μεταβλητής

- Ο alien λέει κάτι που τελειώνει σε 10. Δίνει δηλ. στον astronaut 10 δευτερόλεπτα για να συνετιστεί
- Ο χρόνος αυτός εκχωρείται ως αρχική τιμή σε μια μεταβλητή με το όνομα “xronos” από την κάτω οριζόντια γραμμή εργαλείων, με το κουμπί **variable**

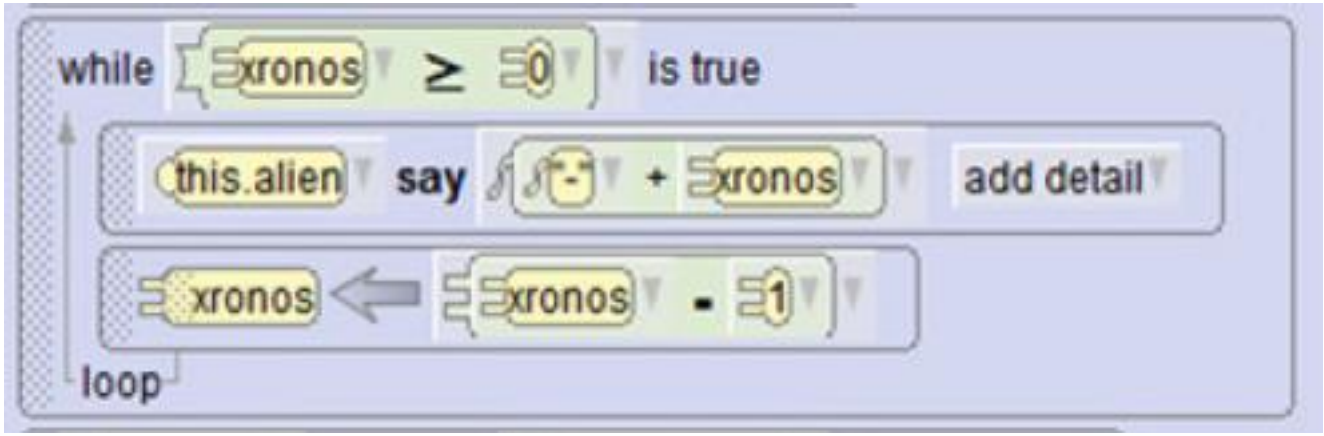




© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.7: Η δομή επανάληψης **while**

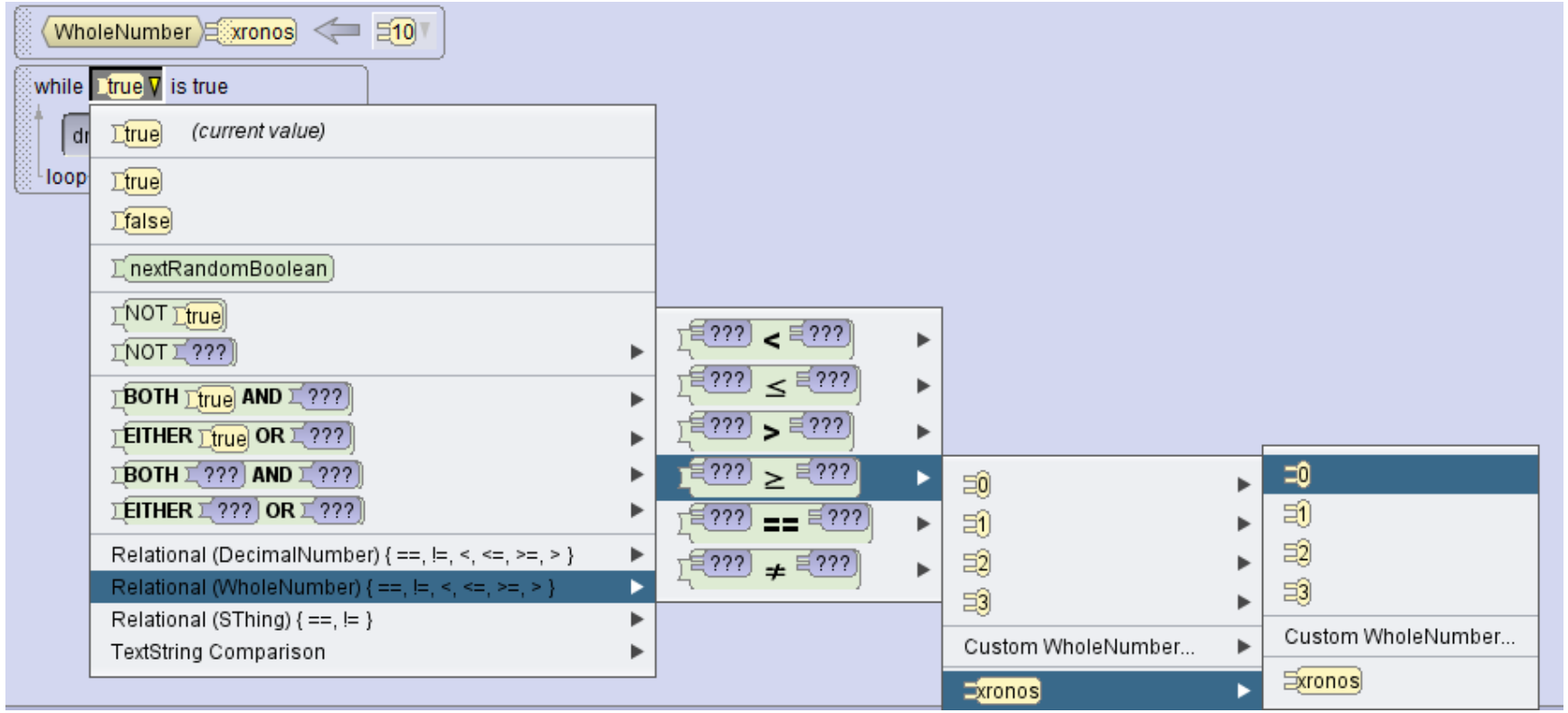
- Με τη χρήση της δομής επανάληψης **while,** ο alien μετράει αντίστροφα τον εναπομείναντα χρόνο.
- Η μεταβλητή **xronos** μειώνεται με βήμα ένα σε κάθε επανάληψη, μέχρι να μηδενιστεί.
- Ακολουθούν αναλυτικές διαφάνειες για τη συνθήκη και τις εντολές του loop



© 2014 - Ε. Γλαμπεδάκης

Επεξήγηση εισαγωγής συνθήκης στη δομή επανάληψης **while**

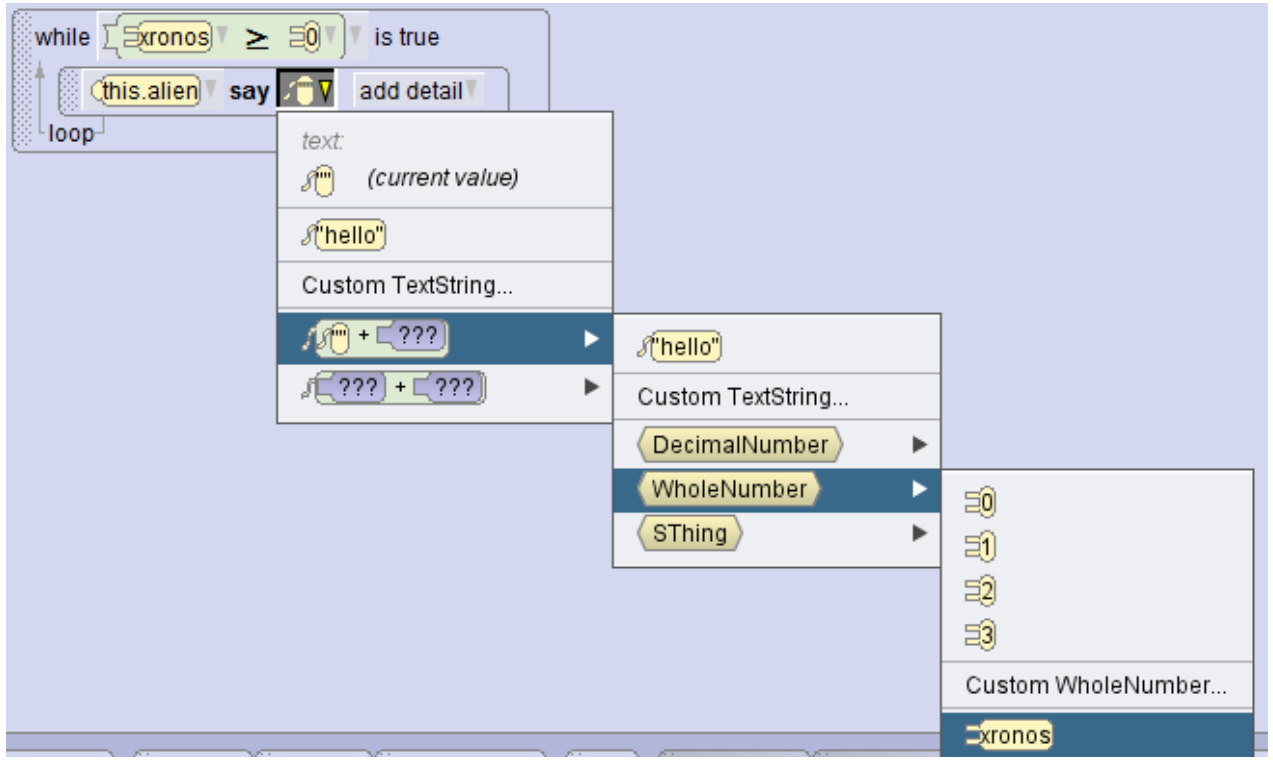
- Μετά το drag & drop της while στη myFirstMethod επιλέγουμε διαδοχικά:



© 2014 - Ε. Γλαμπεδάκης

Επεξήγηση εισαγωγής της μεθόδου “**say**” στη δομή επανάληψης **while**

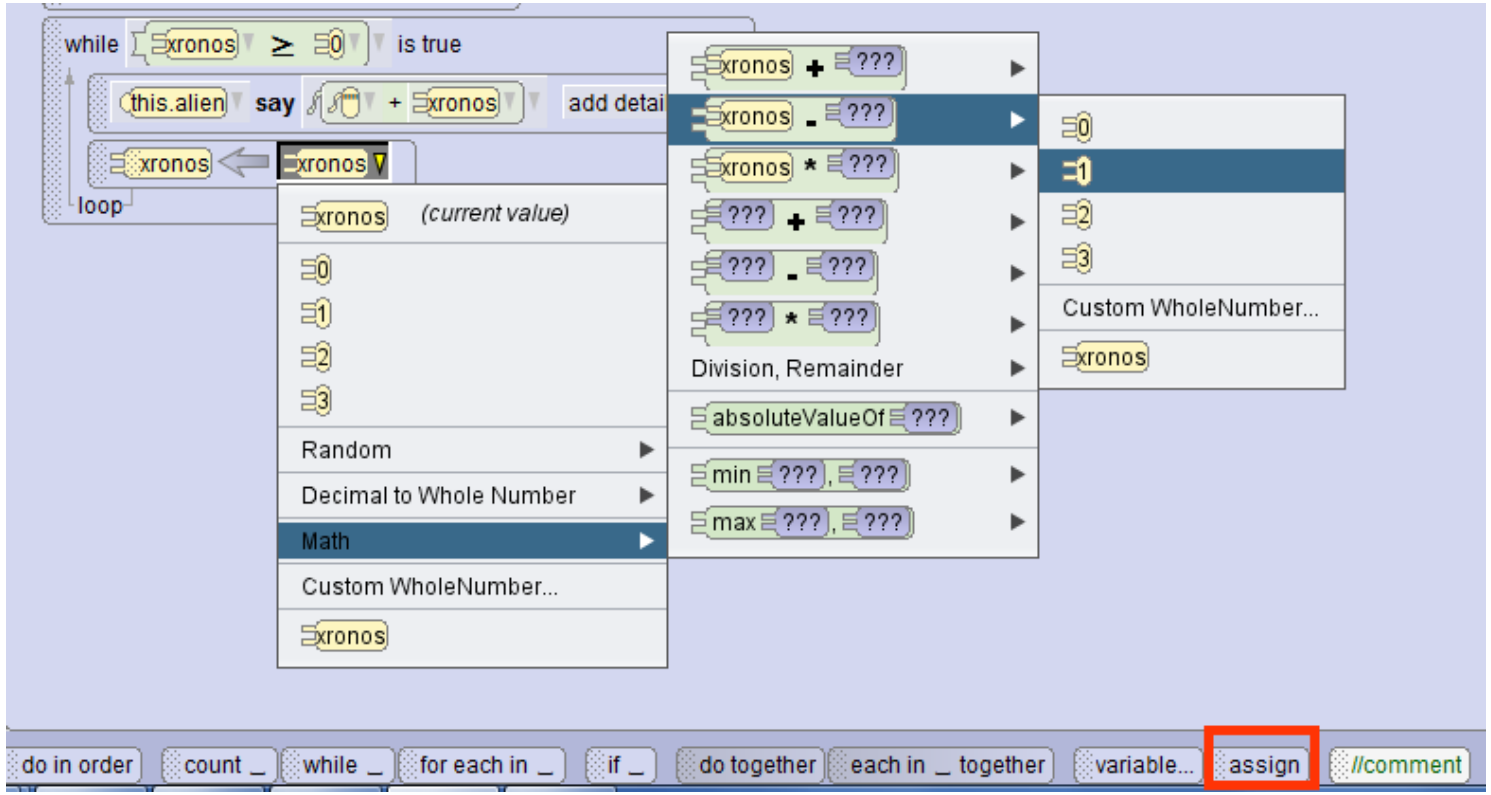
- Εισάγω τη μέθοδο **say** χωρίς κείμενο και μετά δεξιότερα με το κίτρινο κάτω βέλος επιλέγω



© 2014 - Ε. Γλαμπεδάκης

Επεξήγηση εισαγωγής εντολών εκχώρησης τιμής σε μεταβλητές μέσα στη δομή επανάληψης **while**

- Για να μεταβάλλουμε την τιμή της μεταβλητής **xronos** μέσα στο loop, χρησιμοποιούμε τη μέθοδο “assign” από την κάτω οριζόντια γραμμή εργαλείων



© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.8: Πρόοδος σεναρίου

- Ο alien κινείται προς τον astronaut

© 2014 - Ε. Γλαμπεδάκης

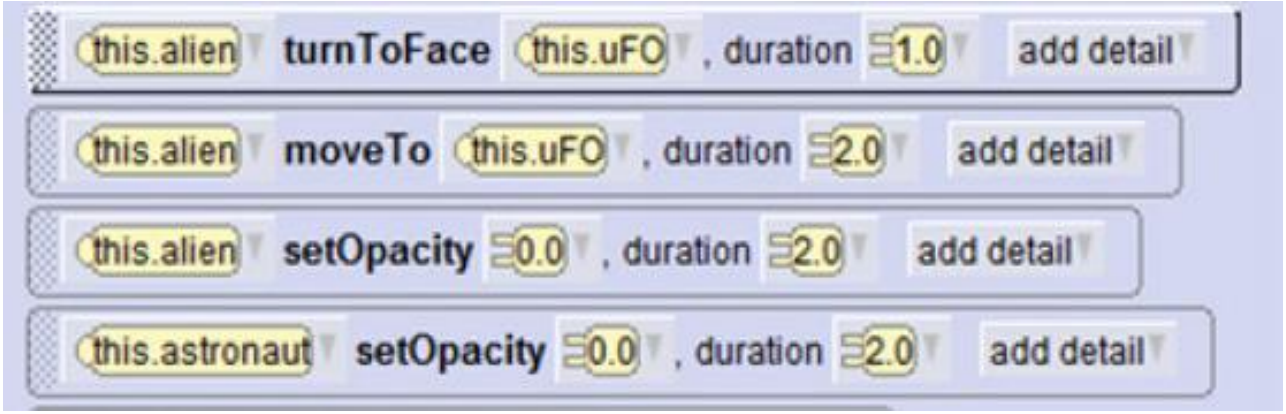
Βήμα 7.9: Η μέθοδος **setVehicle**

- Με τη μέθοδο **setVehicle** τα δύο αντικείμενα αντιμετωπίζονται σαν ένα (ομαδοποίηση).
- Πρακτικά το ένα αντικείμενο γίνεται το «όχημα» του άλλου
- Οποιαδήποτε αλλαγή επομένως στη συμπεριφορά του alien, προκαλεί την ίδια συμπεριφορά και στον astronaut.

© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.10: Πρόοδος σεναρίου

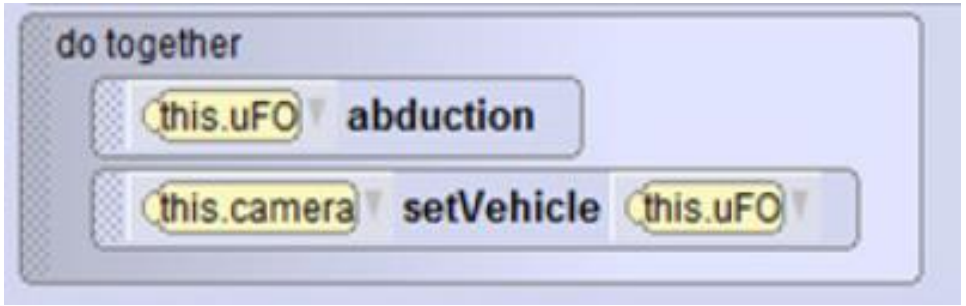
- Ο alien (άρα και ο astronaut μετά την εφαρμογή της **setVehicle**) στρέφεται προς το διαστημόπλοιο (μέθοδος **turnToFace**)
- Κινείται προς το UFO με διάρκεια κίνησης 2 sec
- Στο τέλος εξαφανίζονται και οι δυο.
- Προσοχή: Αν και κινούνται μαζί, δεν αποτελούν ένα αντικείμενο. Γι αυτό και πρέπει να εξαφανιστούν χωριστά με τη μέθοδο **setOpacity**



© 2014 - Ε. Γλαμπεδάκης

Βήμα 7.11: Δημιουργία υποπρογράμματος (διαδικασία-procedure)

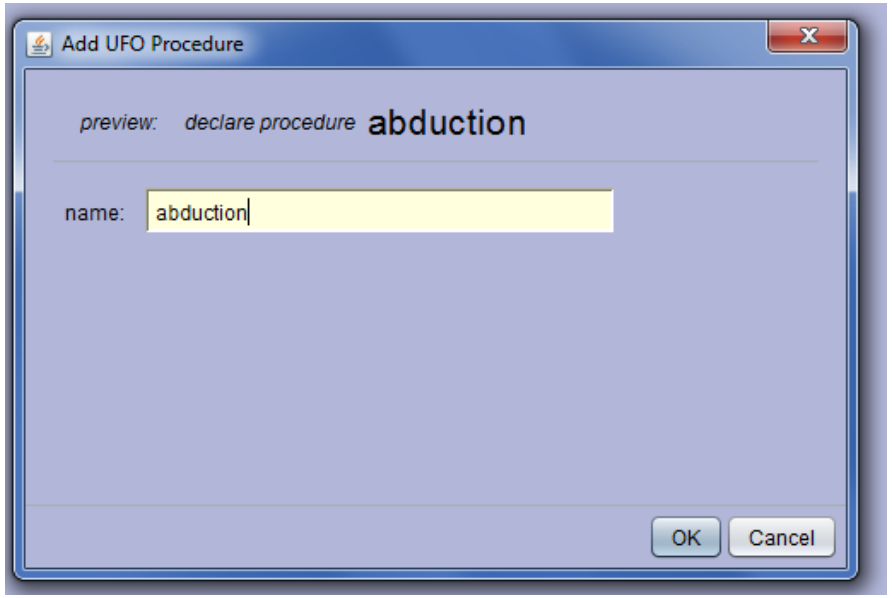
- Με τη διαδικασία abduction (απαγωγή) το UFO απομακρύνεται από τη σκηνή μας
- Εκτελούνται παράλληλα: η κλήση της procedure: abduction (απαγωγή) και η ομαδοποίηση της κάμερας με το UFO.

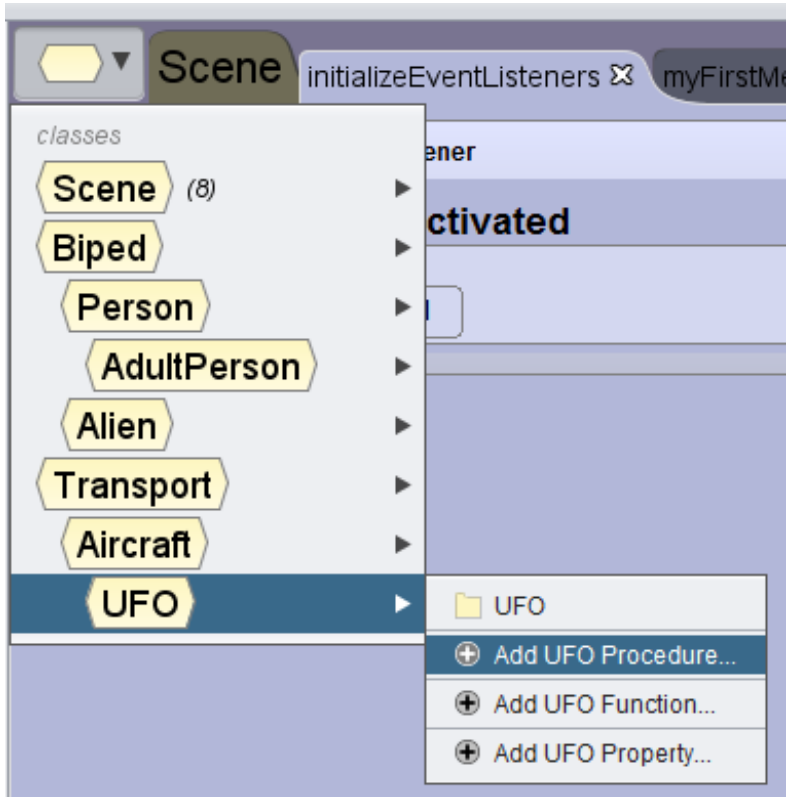


© 2014 - Ε. Γλαμπεδάκης

Επεξήγηση δημιουργίας διαδικασίας (procedure)

- Από την οριζόντια γραμμή καρτελών (επάνω) επιλέγω να δημιουργήσω μια νέα διαδικασία για το UFO
- Την ονομάζω abduction



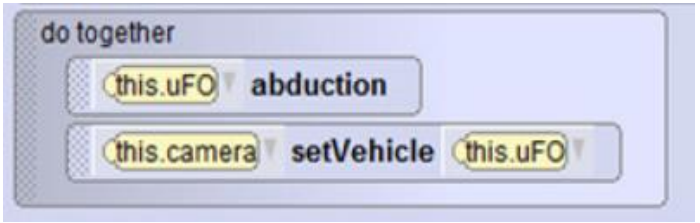


© 2014 - Ε. Γλαμπεδάκης

Επεξήγηση δημιουργίας διαδικασίας (procedure)

- Στο παράθυρο κώδικα της διαδικασίας που μόλις δημιουργήθηκε, χρησιμοποιώντας τις μεθόδους **do together** και **move**, επιβάλλω στο διαστημόπλοιο να κινηθεί προς τα επάνω και δεξιά στη σκηνή

- Μετά τη δημιουργία της **abduction**, την εισάγω στον κώδικα της **myFirstMethod** (βήμα 7.11) και ταυτόχρονα τοποθετώ την κάμερα πάνω στο UFO



© 2014 - Ε. Γλαμπεδάκης

Αλληλεπίδραση της εφαρμογής με το χρήση

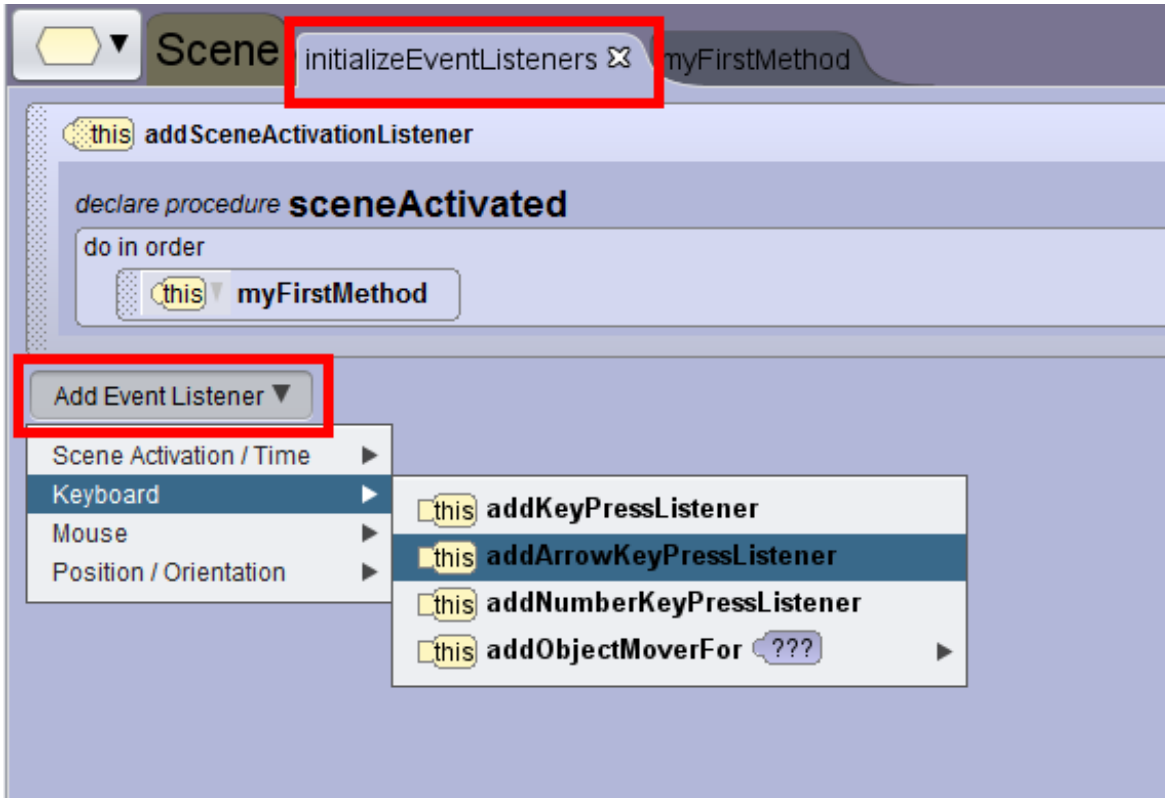
- Μέχρι τώρα στο σενάριο μας, όλο το επεισόδιο διαδραματίζεται σαν μια ταινία.
- Θα προσπαθήσουμε να εισάγουμε **διαδραστικότητα** στην εφαρμογή
- Έτσι:
  - Ο astronaut ζητά βοήθεια από το χρήστη της εφαρμογής
  - Εμφανίζεται μήνυμα για τη χρήση των βελών του πληκτρολογίου
  - Ο χρήστης πατώντας το αριστερό ή το δεξί βελάκι μετακινεί τον astronaut ώστε να ξεφύγει από τον alien

© 2014 - Ε. Γλαμπεδάκης

© 2014 - Ε. Γλαμπεδάκης

Διαδραστικότητα: Add Event Listener

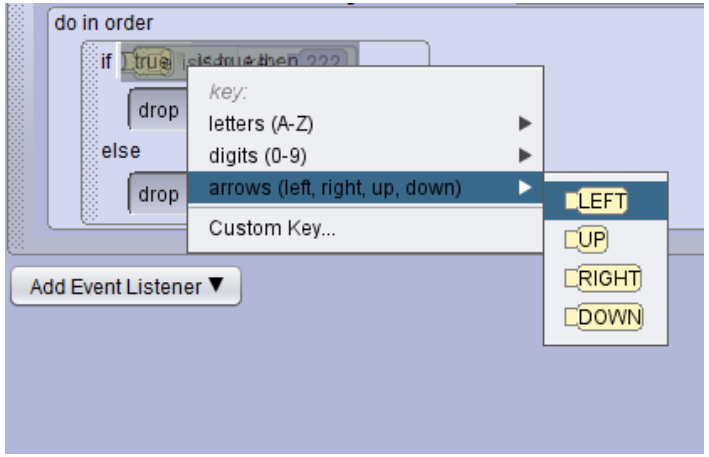
- Επιλέγω την καρτέλα InitializeEventListeners
- Προσθέτω ένα νέο Event Listener
- Για να χρησιμοποιήσω τα βελάκια για κίνηση του astronaut, επιλέγω **addArrowKeyPressListener**

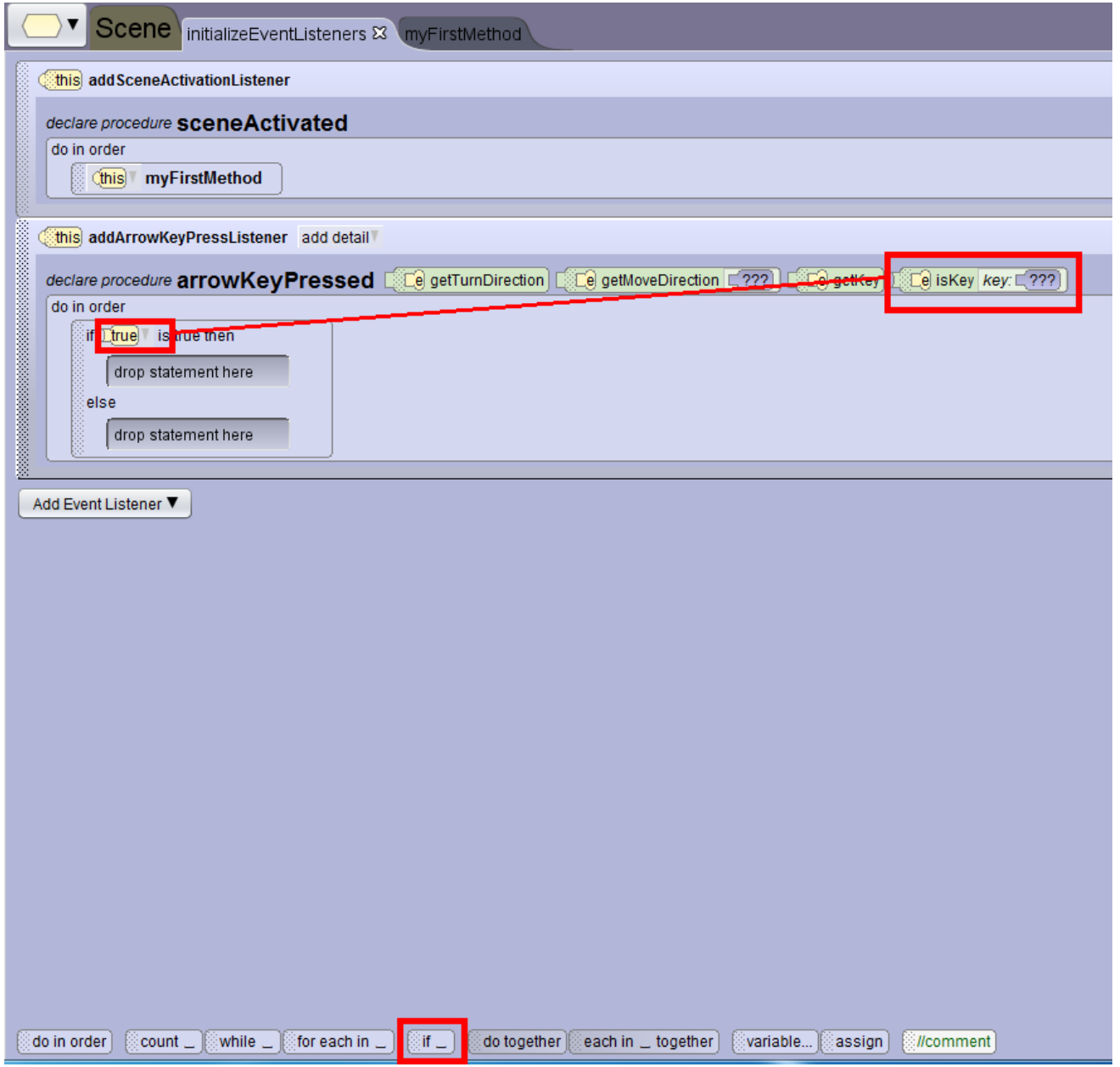


© 2014 - Ε. Γλαμπεδάκης

Βήματα

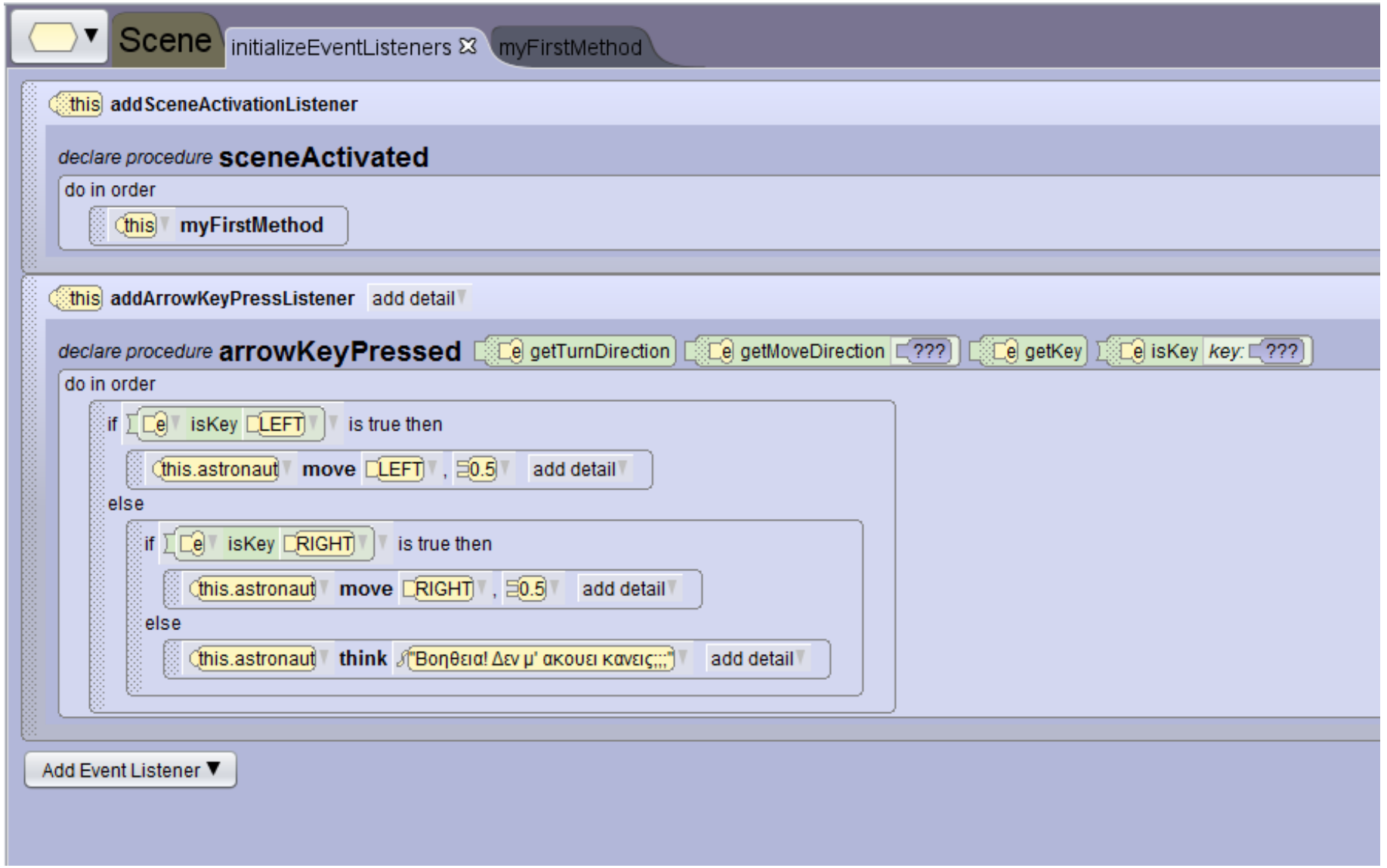
1. Εισάγω μια **if** statement (από την κάτω γραμμή εργαλείων)
2. Σέρνω επάνω στη συνθήκη true, το γεγονός “**e iskey key ???**”
3. Ορίζω τι θα γίνει αν πατηθεί το **αριστερό** βέλος
4. Ορίζω τι θα γίνει αν πατηθεί το **δεξί** βέλος (με δεύτερη **if** statement)
5. Ορίζω τι θα γίνει σε κάθε άλλη περίπτωση (με **else**)





© 2014 - Ε. Γλαμπεδάκης

Τελικά, αφού καλύψω τις ενδεχόμενες περιπτώσεις, πρέπει να έχω φτιάξει το διπλανό block κώδικα



© 2014 - Ε. Γλαμπεδάκης

Εμβέλεια της μεθόδου

- Η μέθοδος που κινεί τον astronaut, είναι καθολική
- Επομένως εφαρμόζεται σε οποιαδήποτε στιγμή του σεναρίου
- Ωστόσο, ο χρήστης πατάει τα βελάκια, όταν τον προτρέψει ο astronaut
- Αν αυξήσω τη διάρκεια το μηνύματος *«ΒΟΗΘΕΙΑ χρήστη! Πατά τα βελάκια να με σώσεις!»* μπορώ δώσω περισσότερο χρόνο στο χρήστη να κινήσει τον astronaut.

© 2014 - Ε. Γλαμπεδάκης

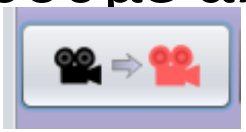
Σενάριο εξάσκησης

Ένα πειρατικό καράβι έχει εντοπίσει ένα ιστιοφόρο μέσα στη θάλασσα. Το προσεγγίζει και αφού κάνει μερικές στροφές γύρω του, επιτίθεται και το βουλιάζει.

ΠΑΡΑΡΤΗΜΑ

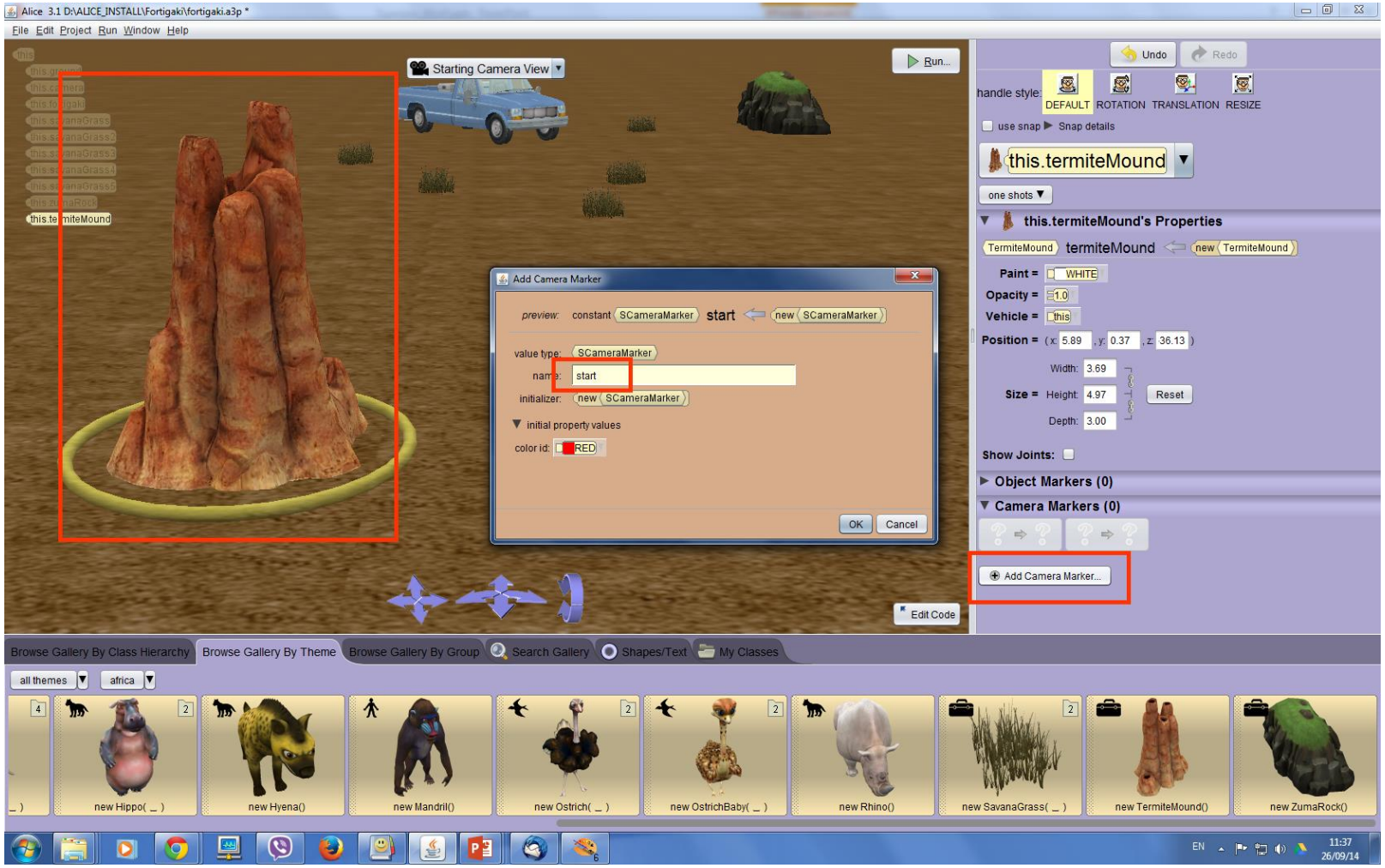
Κάμερες και γωνία θέασης

Εισαγωγή αρχικής κάμερας

- Επιλέγουμε ένα αντικείμενο κοντά στο οποίο θα τοποθετήσουμε την αρχική μας κάμερα
- Κάνουμε κλικ στο δεξί μέρος Camera markers – Add camera marker
- Την ονομάζουμε start (έτσι ξέρουμε από που ξεκινάει το σενάριο μας σε περίπτωση που χαθούμε) 
- Τα τρία βέλη στο κάτω κεντρικό μέρος της σκηνής μας, ελέγχουν πλέον την αρχική μας κάμερα
- Αν μετακινηθούμε λίγο προς τα πίσω, θα δούμε και την ίδια την κάμερα σε κόκκινο χρώμα

© 2014 - Ε. Γλαμπεδάκης

Αρχική κάμερα



© 2014 - Ε. Γλαμπεδάκης

© 2014 - Ε. Γλαμπεδάκης

Σας ευχαριστώ.

ΑΝΑΦΟΡΕΣ

- Jenna Hayesunder the direction of Professor Susan Rodger Duke University, June 2009
- Adams J., Alice in Action with Java, Thomson Course Technology, 2007.
- Adams J., “Alice, Middle Schoolers and the Imaginary World Camps”, 38th SIGCSE technical symposium on Computer science education, 2007
- Cooper S., Dann W., Pausch R., “Teaching objects-first in Introductory Computer Science”, 34th SIGCSE technical symposium on Computer science education, 191-195, 2003.
- Dann W., Cooper S., Pausch R., Learning to Program with Alice, Pearson Prentice Hall, 2006.
- Herbert C., An Introduction to Programming Using Alice, Thomson Course Technology, 2006.
- Kelleher C., Pausch R., Kiesler S., “Storytelling Alice motivates middle school girls to learn computer programming”, SIGCHI conference on Human factors in computing systems, 1455-1464, 2007.
- Moskal B., Lurie D., Cooper S., “Evaluating the Effectiveness of a New Instructional Approach”, 35th SIGCSE technical symposium on Computer science education, 75-79, 2004.
- Shelly G., Cashman T., Herbert C., Alice 2.0: Introductory Concepts and Techniques, Thomson Course Technology, 2006.
- http://www.acm.org/
- http://www.alice.org
- http://www.alicetik2o.com/rpp/index.htm

© 2014 - Ε. Γλαμπεδάκης