



**ΑΝΟΙΚΤΑ** ακαδημαϊκά  
μαθήματα **ΠΠ**

# Εισαγωγή στις Τεχνολογίες της Πληροφορίας και των Επικοινωνιών

Εργαστηριακή Ενότητα 11 : Εισαγωγή στον Οντοκεντρικό  
Προγραμματισμό -Alice

Διδάσκων: Νικόλαος Τσέλιος  
Τμήμα Επιστημών της

Εκπαίδευσης και της Αγωγής στην Προσχολική Ηλικία



# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Σχολή Κοινωνικών και Ανθρωπιστικών Επιστημών, Τμήμα Επιστημών της Εκπαίδευσης και Αγωγής στην Προσχολική Ηλικία, Νικόλαος Τσέλιος, «Εισαγωγή στις Τεχνολογίες της Πληροφορίας και των Επικοινωνιών». Έκδοση: 1.0. Πάτρα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.upatras.gr/courses/PN1407/>

# Σκοποί ενότητας

- Να κατανοήσει ο φοιτητής τη χρησιμότητα του οντοκεντρικού προγραμματισμού για την εκπαίδευση μέσα από το παράδειγμα της Alice
- Να είναι σε θέση να χρησιμοποιήσει την Alice για τη δημιουργία μιας κινούμενης εικόνας, (animation) , την αφήγηση μιας ιστορίας, τη δημιουργία διαδραστικού παιχνιδιού
- Να εξοικειωθεί με τις βασικές λειτουργίες της Alice μέσα από απλές ασκήσεις

# Περιεχόμενα ενότητας

- Περιγραφή: Γίνεται συζήτηση για τον οντοκεντρικό προγραμματισμό παιχνιδιών και συγκεκριμένα για την Alice. Εξηγείται το πως σχεδιάζουμε μια ιστορία με την Alice και πως χρησιμοποιούμε κατάλληλα τα εργαλεία που προσφέρει. Τέλος, γίνεται πρακτική εξάσκηση μέσα από ασκήσεις/ερωτήματα.
- Λέξεις Κλειδιά: Alice, Αφήγηση Ιστορίας

# Τι είναι ο Οντοκεντρικός/Αντικειμενοστραφής Προγραμματισμός;

- Μία μεθοδολογία ανάπτυξης προγραμμάτων,
- Όλα περιστρέφονται γύρω από αντικείμενα τα οποία έχουν τη δική τους Ταυτότητα και Ιδιότητες!
- Εμφανίστηκε στα τέλη της δεκαετίας του **1960**
- Καθιερώθηκε κατά τη δεκαετία του **1990**

# Τι είναι η Alice;

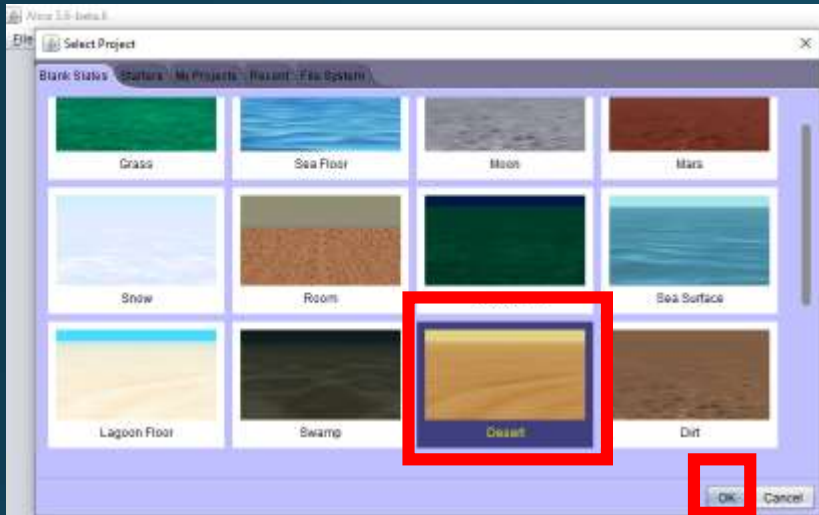
εκπαιδευτικό λογισμικό για τη διδασκαλία οντοκεντρικού προγραμματισμού σε 3D γραφικό περιβάλλον.

Χρησιμοποιείται για

- τη δημιουργία μιας κινούμενης εικόνας, (animation)
  - την αφήγηση μιας ιστορίας,
  - ένα διαδραστικό παιχνίδι
- Περιλαμβάνει 3-D αντικείμενα τα οποία ενσωματώνονται σε ένα εικονικό κόσμο.
  - Στη συνέχεια δημιουργούμε το πρόγραμμα για την κίνηση των αντικειμένων του εικονικού κόσμου που σχεδιάστηκε.

Περιβάλλον Alice

# Επιλογή Εικονικού Κόσμου:





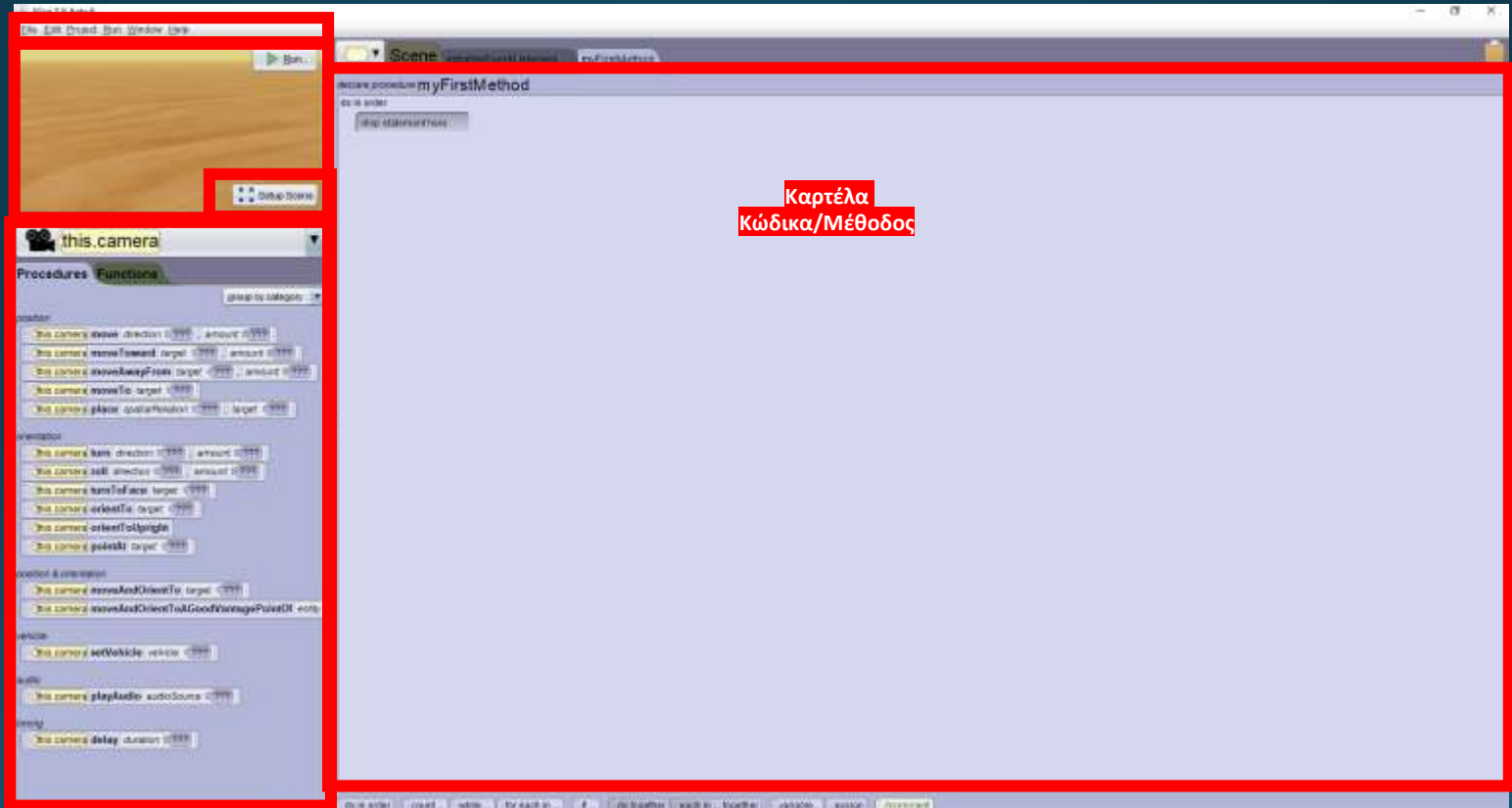
# Περιβάλλον Alice

Μενού

Σκηνή

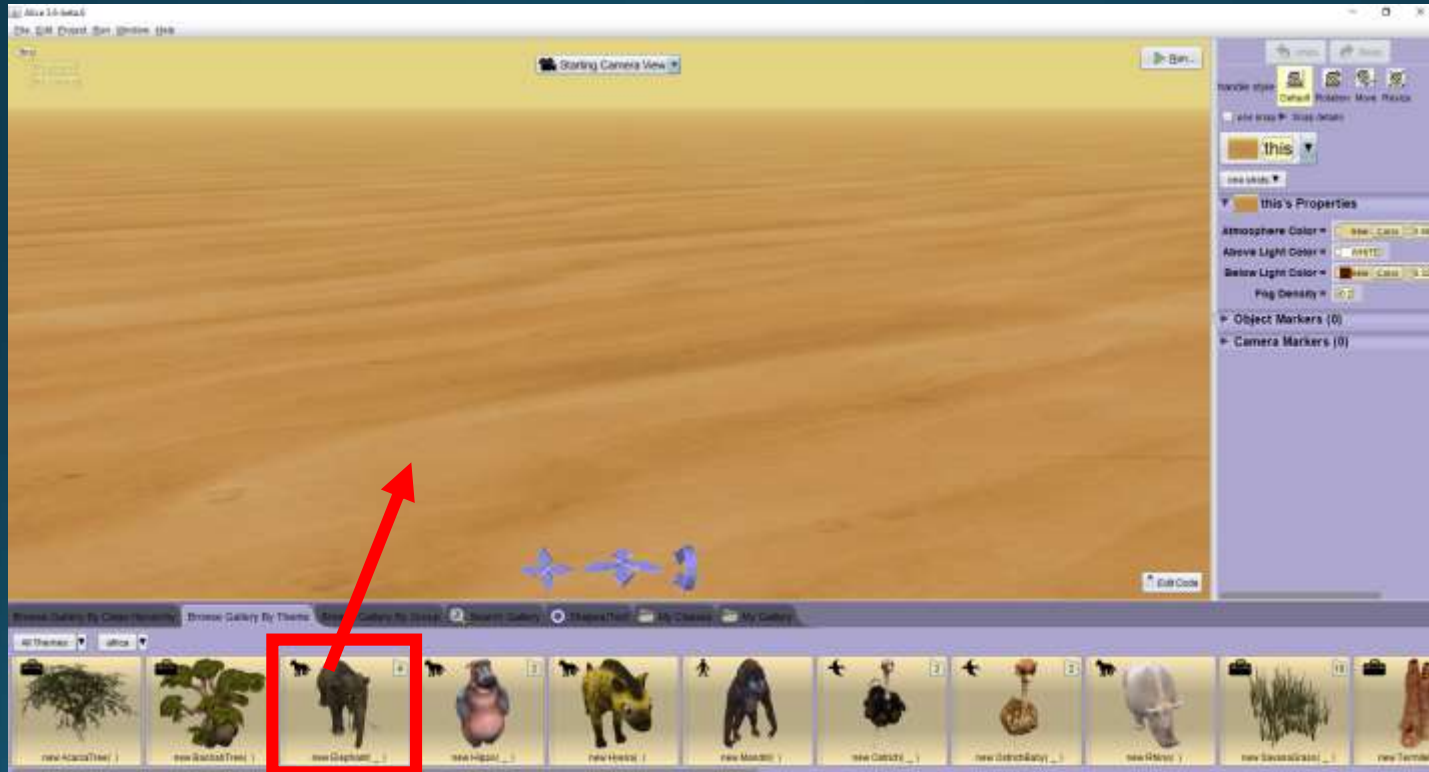
Καρτέλα  
Κώδικα/Μέθοδος

Καρτέλα  
Διαδικασιών &  
Συναρτήσεων





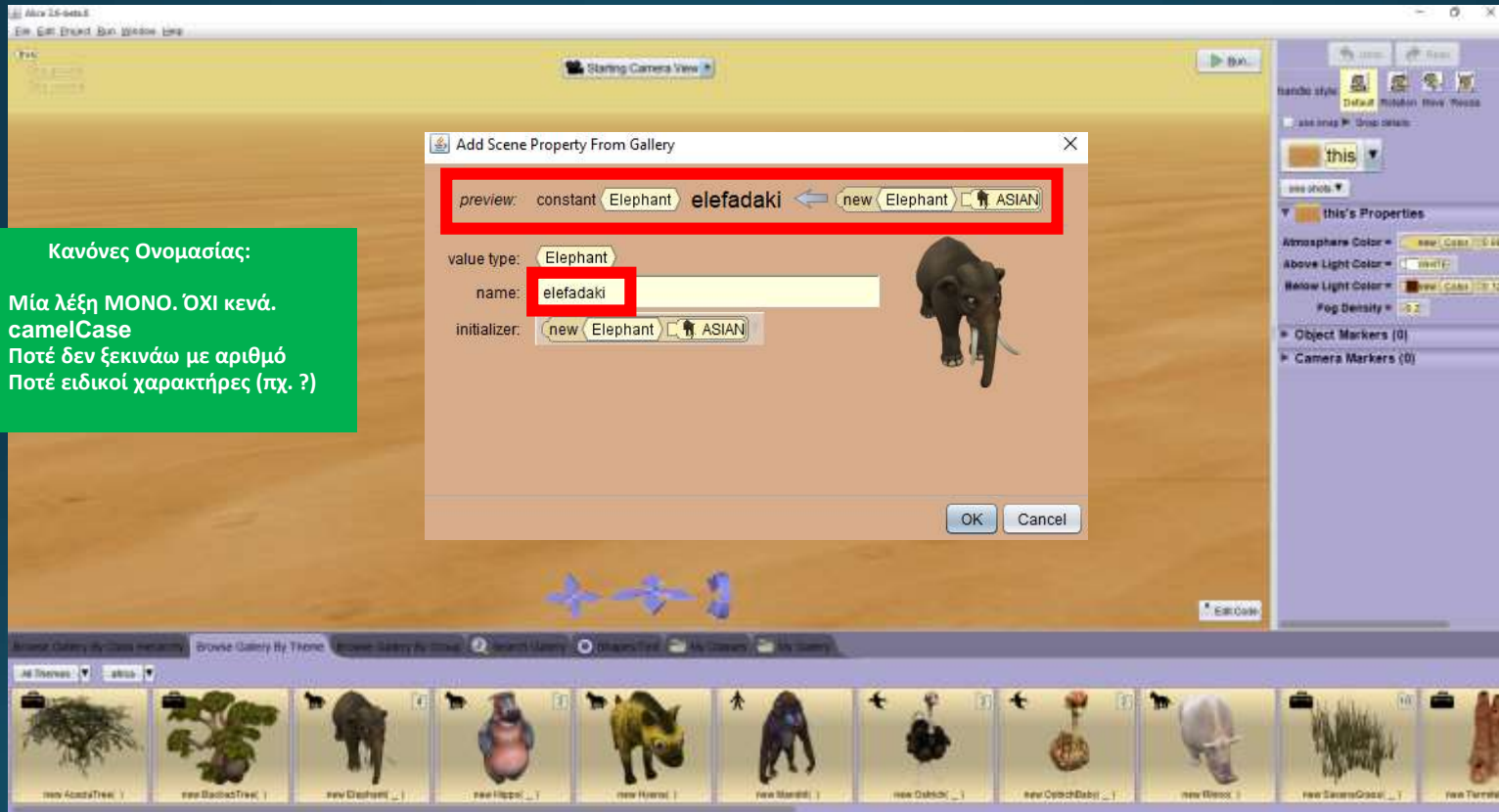
# Πώς θα προσθέσω στην σκηνή μου αντικείμενα;



Τεχνική «Σύρε και Άφησε»!

### Κανόνες Ονομασίας:

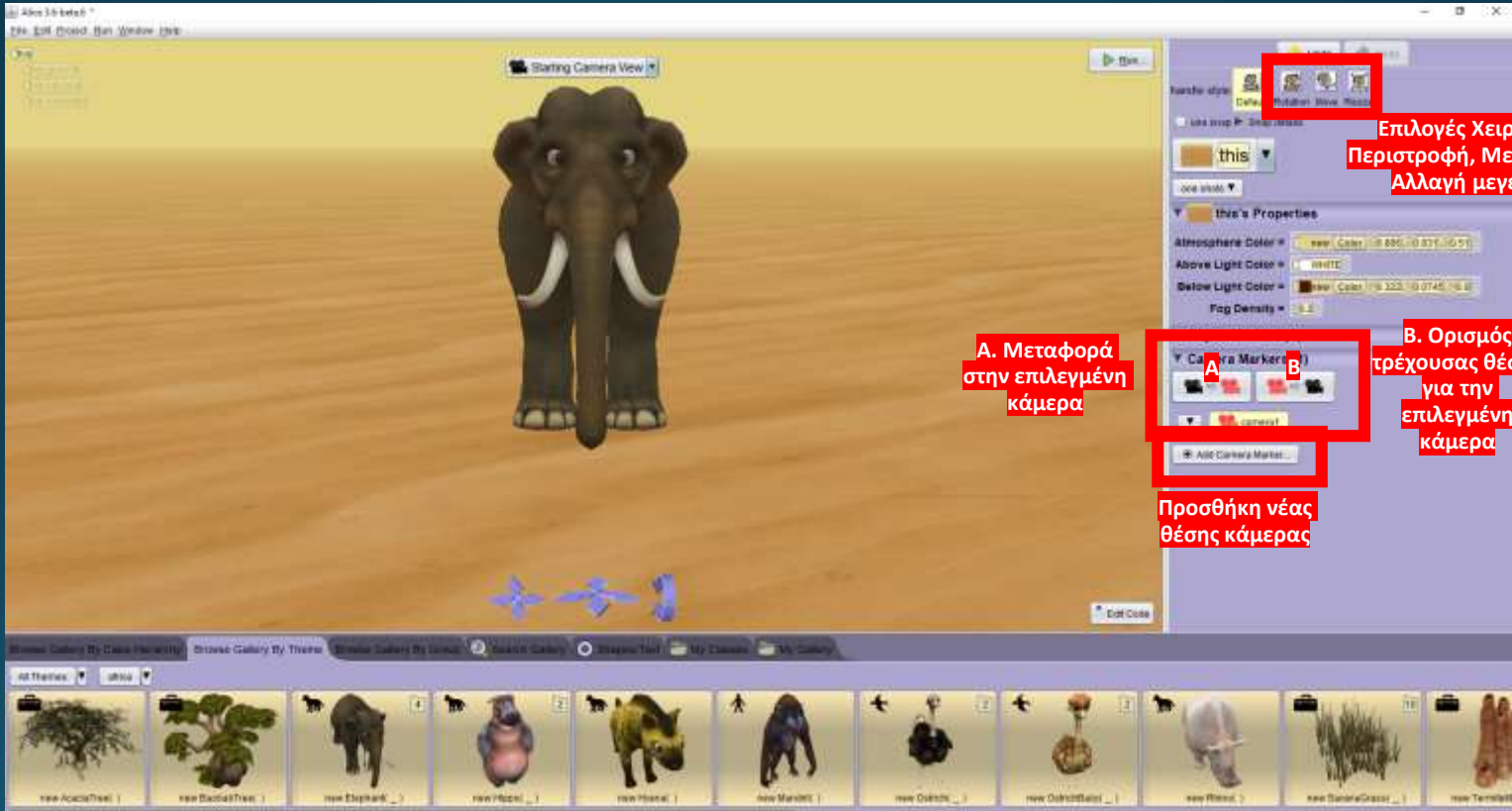
1. Μία λέξη ΜΟΝΟ. ΌΧΙ κενά.
2. camelCase
3. Ποτέ δεν ξεκινάω με αριθμό
4. Ποτέ ειδικοί χαρακτήρες (πχ. ?)



# ΩΧ! Ο Ελέφαντας είναι πραγματικά μεγάλος! Τί μπορώ να κάνω;



# Θέλω να τοποθετήσω μία κάμερα ώστε να μπορώ εύκολα να ξαναέρθω σε αυτό το σημείο!



Επιλογές Χειρισμού:  
Περιστροφή, Μετακίνηση,  
Αλλαγή μεγέθους

Α. Μεταφορά  
στην επιλεγμένη  
κάμερα

Β. Ορισμός  
τρέχουσας θέσης  
για την  
επιλεγμένη  
κάμερα

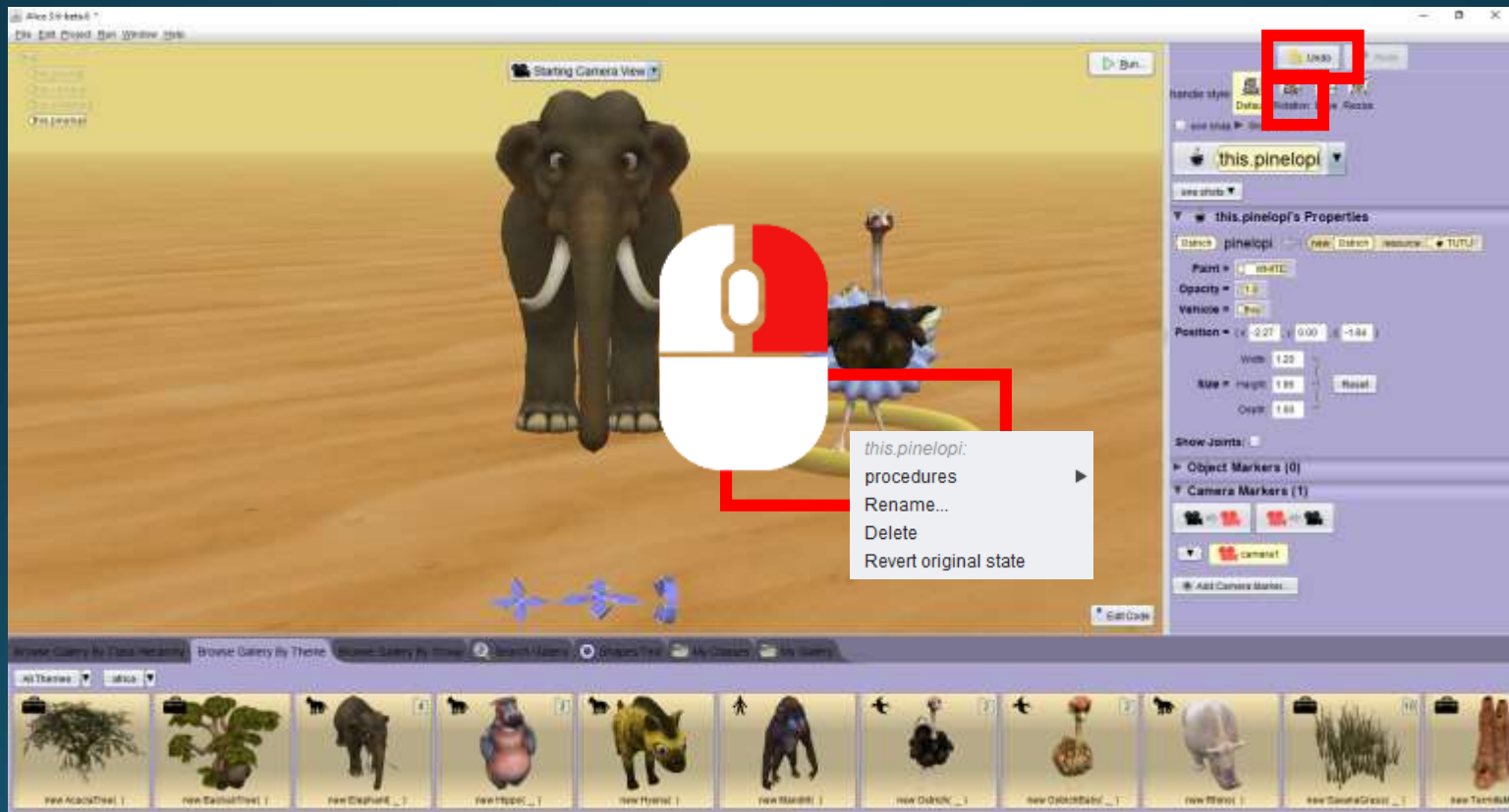
Προσθήκη νέας  
θέσης κάμερας



Θέλω να προσθέσω στην σκηνή και μία στρουθοκάμηλο! Τι πρέπει να κάνω;



# Θέλω η στρουθοκάμηλος να κοιτάξει τον ελέφαντα!





Τι “turn” πρέπει να κάνω για να γυρίσει η στρουθοκάμηλος προς τον ελέφαντα;



- Τι σημαίνουν οι αριθμοί «0.25, 2.5, 1, 2» ?



# Θέλω να μετακινήσω την στρουθοκάμηλο λίγο πιο δεξιά!



# Αποστολή 1<sup>η</sup> : Γυρίστε τον ελέφαντα ώστε να κοιτάζει την στρουθοκάμηλο



# Αποστολή 2<sup>η</sup> : Κάντε την στρουθοκάμηλο κίτρινη





# Αποστολή 3<sup>η</sup> : Χαιρετισμός Ελέφαντα

Starting Camera View

Run

this.pinelopi

this.pinelopi's Properties

Detach pinelopi new Detrich resource TUTU

Paint = YELLOW

Opacity = 1.0

Vehicle = Bird

Position = ( x: -2.77 y: 0.00 z: -1.84 )

Width: 0.72

Size = Height: 1.17 Depth: 0.59

Show Joints:

Object Markers (0)

Camera Markers (1)

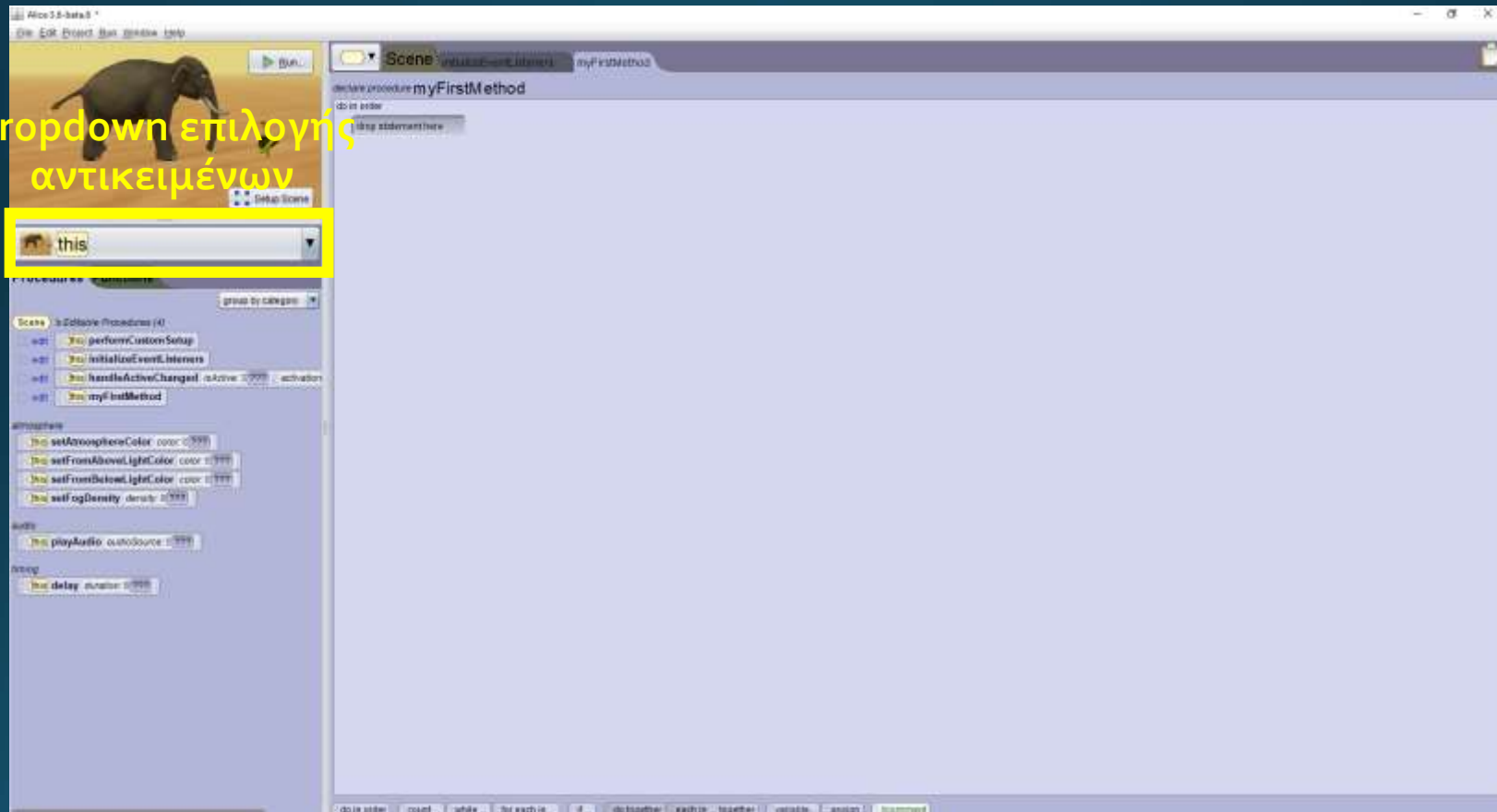
camera1

Add Camera Marker...

Edit Code

Επιστροφή στον κώδικα

Dropdown επιλογής  
αντικειμένων



# Θέλω όταν ξεκινάει το παιχνίδι ο ελέφαντας να λέει «Γεια σου Πηνελόπη!»

The screenshot shows the Unity Inspector window for a GameObject. A 'Custom TextString' dialog box is open, with 'preview' and 'value' fields both containing the text 'ΓΕΙΑ ΣΟΥ ΠΗΝΕΛΟΠΗ!'. The 'value' field is highlighted in yellow. A red arrow points from this field to a 'text' property in the Inspector, which currently contains 'hello'. Another red arrow points from the 'Custom TextString...' button in the Inspector to the dialog box. In the console window, a log entry is highlighted with a red box: 'this elephant say text: ΓΕΙΑ ΣΟΥ ΠΗΝΕΛΟΠΗ!'. A blue arrow points from this log entry to the text 'Τρέχω τον κώδικα!'.

Custom TextString

preview: ΓΕΙΑ ΣΟΥ ΠΗΝΕΛΟΠΗ!

value: ΓΕΙΑ ΣΟΥ ΠΗΝΕΛΟΠΗ!

OK Cancel

text: hello

Custom TextString...

Τρέχω τον κώδικα!

<αντικείμενο> <κάνει διαδικασία> <με παράμετρο>



# Αποστολή 4<sup>η</sup> & 5<sup>η</sup>

- Αποστολή 4<sup>η</sup>
  - Θέλω η φράση «Γειά σου Πηνελόπη» να μένει στην οθόνη για 5 δευτερόλεπτα.
  - Έπειτα η Πηνελόπη να σκέφτεται «Ωχ! Ο Ελέφαντας» για 5 δευτερόλεπτα.
  - Έπειτα να του λέει «Γειά σου ελέφαντα! Δεν σε είδα!» και να μένει στην οθόνη για **10** δευτερόλεπτα
- Αποστολή 5<sup>η</sup>
  - Θέλω η φράση «Γειά σου ελέφαντα! Δεν σε είδα!» να εμφανίζεται σαν δύο ομιλίες. Δηλαδή:
  - Να εμφανίζεται η φράση «Γειά σου ελέφαντα!» για 5 δευτερόλεπτα
  - Έπειτα η φράση «Δεν σε είδα!» για 5 δευτερόλεπτα.

Scene myFirstMethod

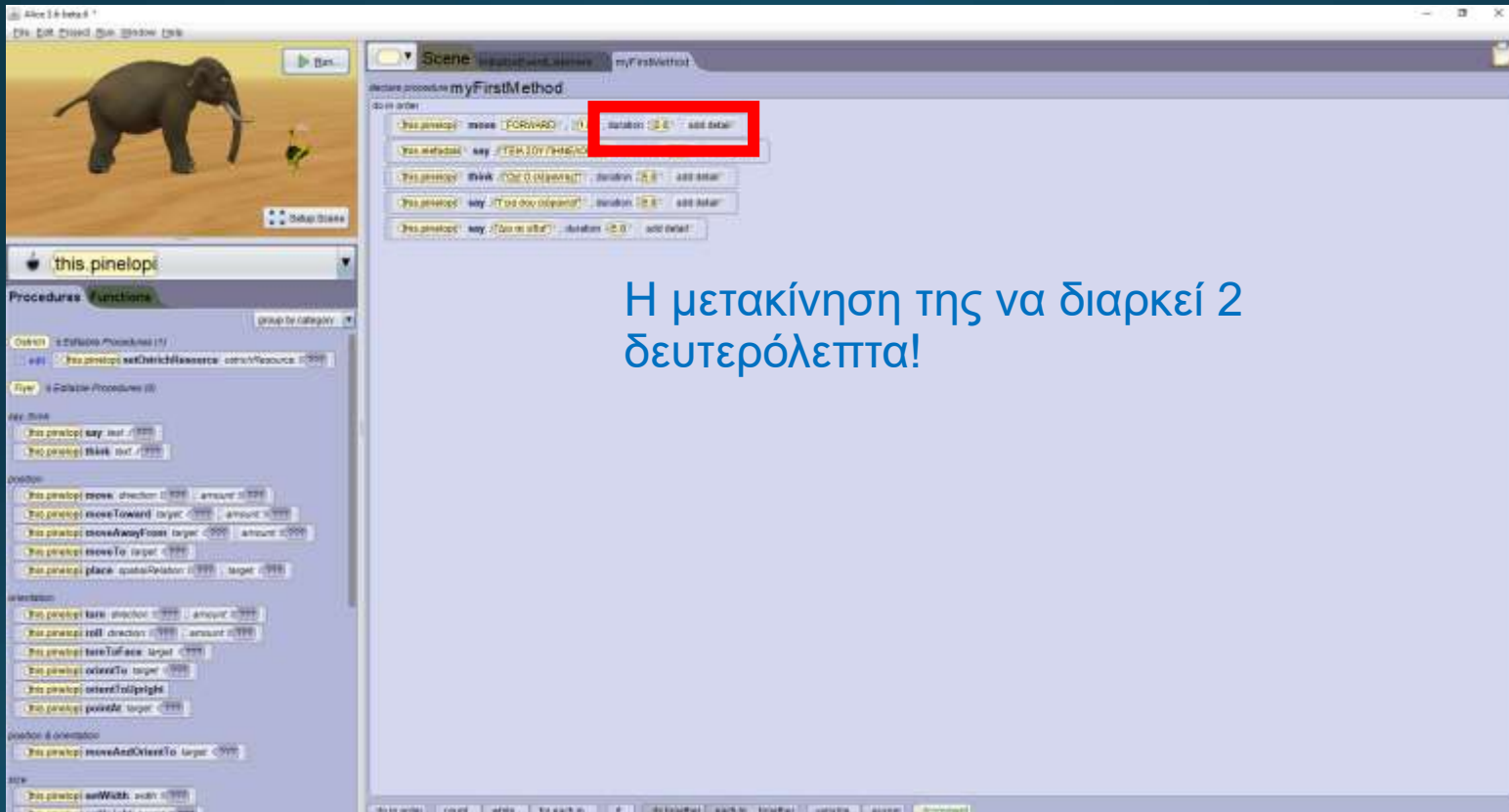
move FORWARD 1.0 add detail

think /Το ούλιπεντ/ duration 2.0 add detail

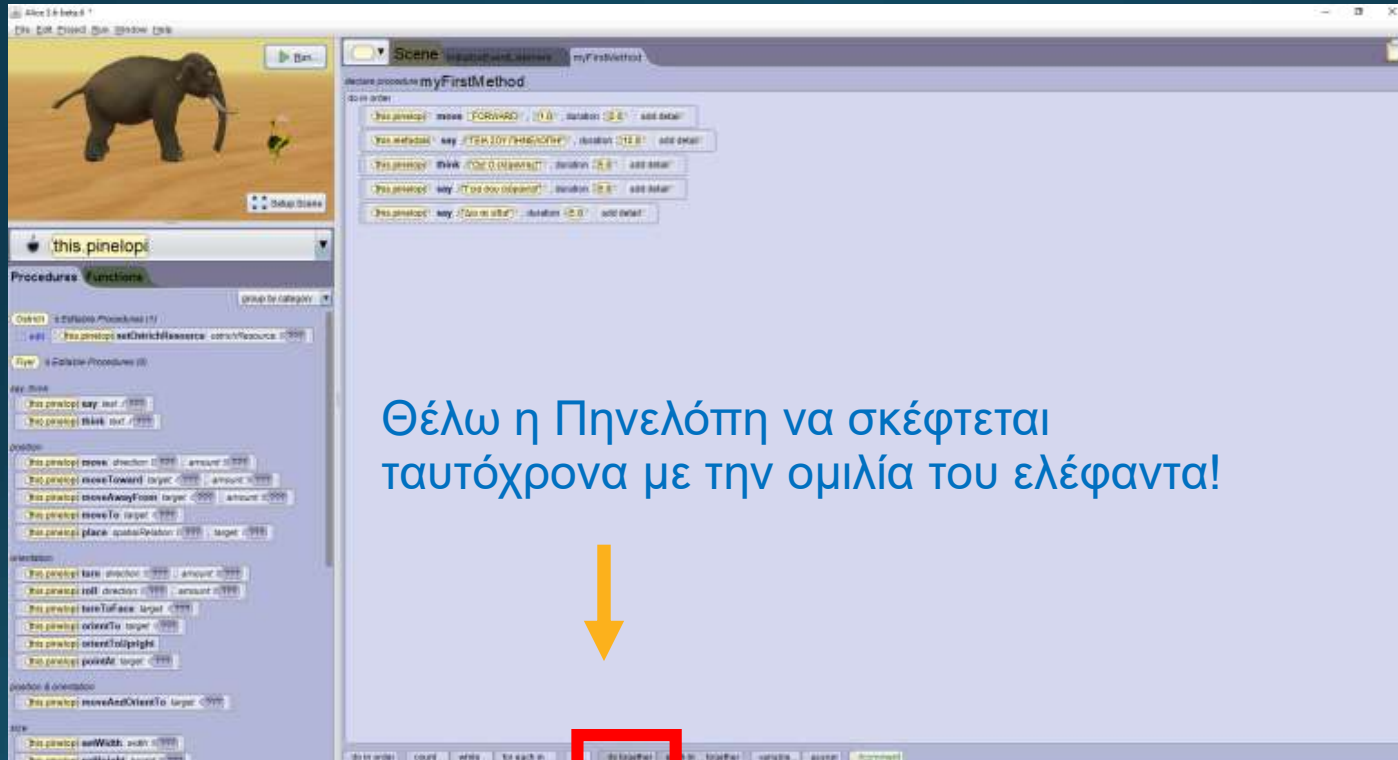
say /Το ούλιπεντ/ duration 2.0 add detail

say /Αν ούλιπεντ/ duration 2.0 add detail

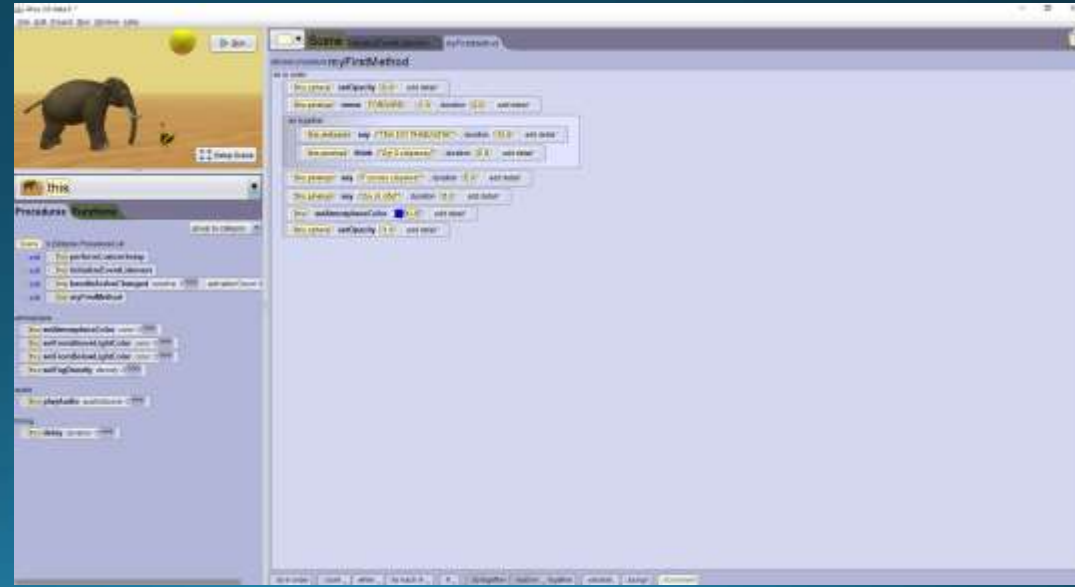
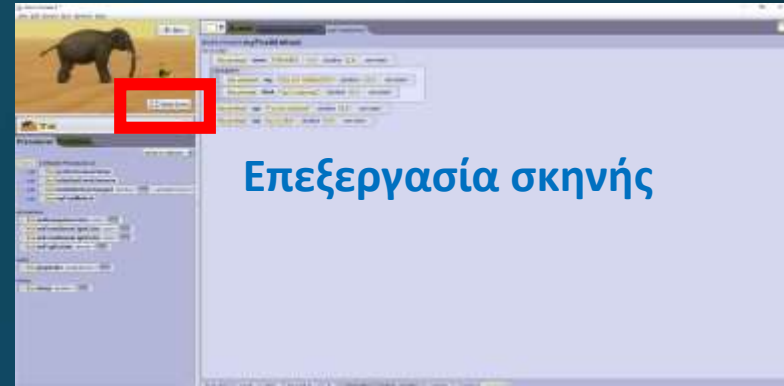
θέλω στην αρχή του παιχνιδιού να πλησιάσει τον ελέφαντα!



Η μετακίνηση της να διαρκεί 2 δευτερόλεπτα!



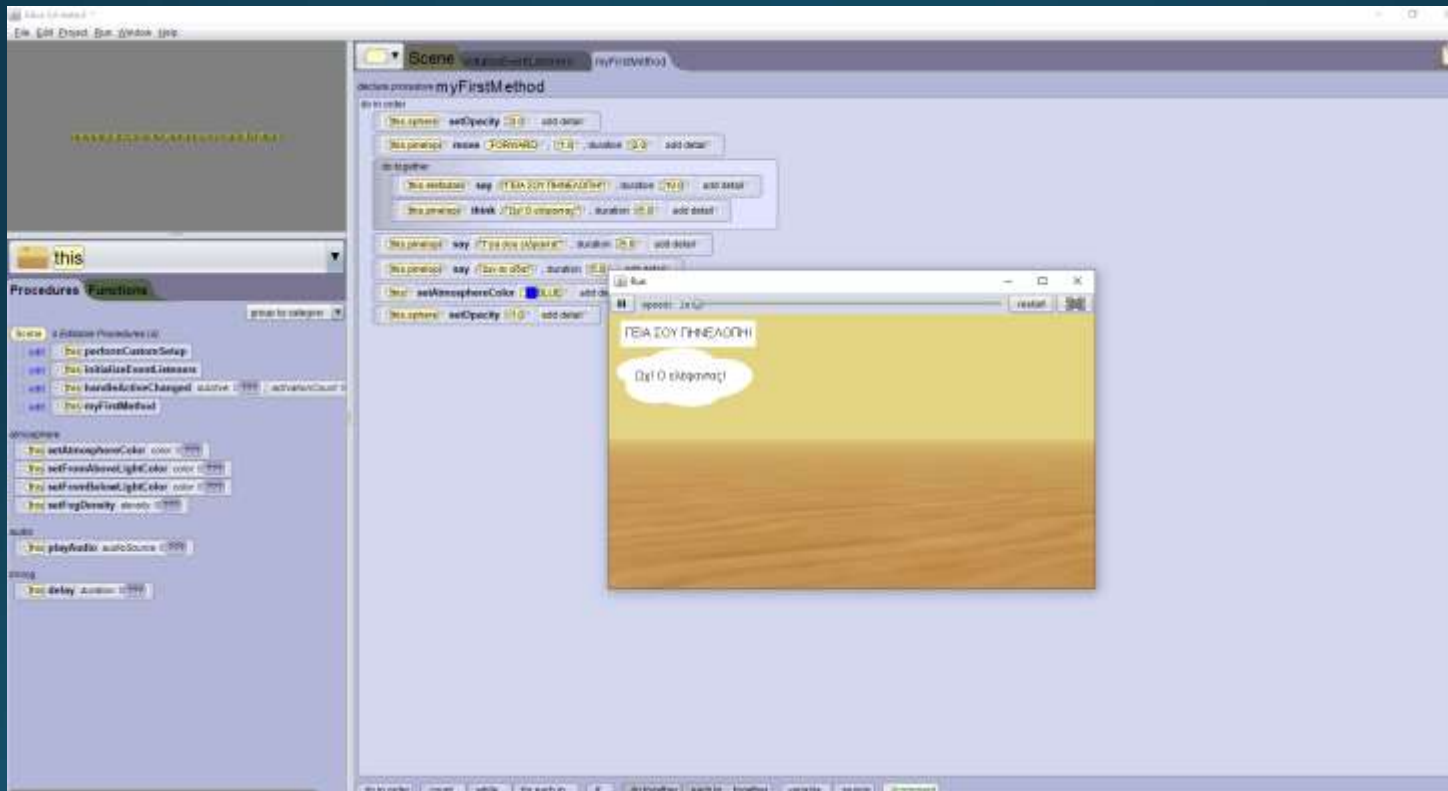
- Μόλις τελειώσουν οι συνομιλίες θέλω να νυχτώνει!
- Τι σημαίνει «νυχτώνει»;
- Αλλάζει χρώμα ο ουρανός;
- Βγαίνει το φεγγάρι;



- Τι θα γίνει εάν πειράξω την θέση της κάμερας μέσα στο **setup scene** και επιστρέψω στο **edit code** για να παίξω αυτό που έφτιαξα?



Τι θα γίνει εάν πειράξω την θέση της κάμερας μέσα στο `setup scene` και επιστρέψω στο `edit code` για να παίξω αυτό που έφτιαξα?





Επιλέξτε τον ελέφαντα και ενεργοποιήστε την επιλογή «Show Joints». Τι εμφανίζεται;





# Θέλω το ελεφαντάκι να κουνάει την ουρά του

The screenshot displays the Alice 3.6 software interface. On the left, a 3D scene window shows an elephant in a desert-like environment with a yellow sphere. Below the scene, there are buttons for 'this', 'Procedures', and 'Functions'. The 'Functions' list includes methods like 'this.getGround', 'this.getCamera', and 'this.getSphere'. The main workspace on the right shows a script for the 'Elephant' object. The script is organized into a 'myFirstMethod' block, which contains several 'say' and 'think' actions with durations, and a 'setOpacity' action for the sphere. The script is as follows:

```
myFirstMethod  
- say /ΓΕΜΙΣΤΗ ΠΑΡΕΛΑΘΗ! / duration: 10.0 / add detail  
- think /Ω! Ο σπέρμα! / duration: 5.0 / add detail  
- say /Το σου σπέρμα! / duration: 5.0 / add detail  
- say /Αν η πύλη! / duration: 5.0 / add detail  
- say /Αν η πύλη! / add detail  
- setAtmosphereColor BLUE / add detail  
- setOpacity 1.0 / duration: 10.0 / add detail
```



this

Procedures **Functions**

group by category ▾

Scene Editable Procedures (4)

- +01 my performCustomSetup
- +01 my initializeEventListeners
- +01 my handleActiveChanged (active: 0/100, activationCount: 0)
- +01 my myFirstMethod

atmosphere

- my setAtmosphereColor color: 0/0/0
- my setFromAboveLightColor color: 0/0/0
- my setFromBelowLightColor color: 0/0/0
- my setFogDensity density: 0/0/0

audio

- my playAudio audioSource: 0/0/0

timing

- my delay duration: 0/0/0

Scene initializeEventListeners myFirstMethod

declare procedure myFirstMethod

do in order

- my elephant getTail turn FORWARD 0.25 add detail!
- my camera moveAndOrientTo my camera duration: 0.0 add detail!
- my sphere setOpacity 0.0 add detail!
- my penelope move FORWARD 1.0 duration: 2.0 add detail!

together

- my elephant say TEA TOY THREEKITS duration: 0.0 add detail!

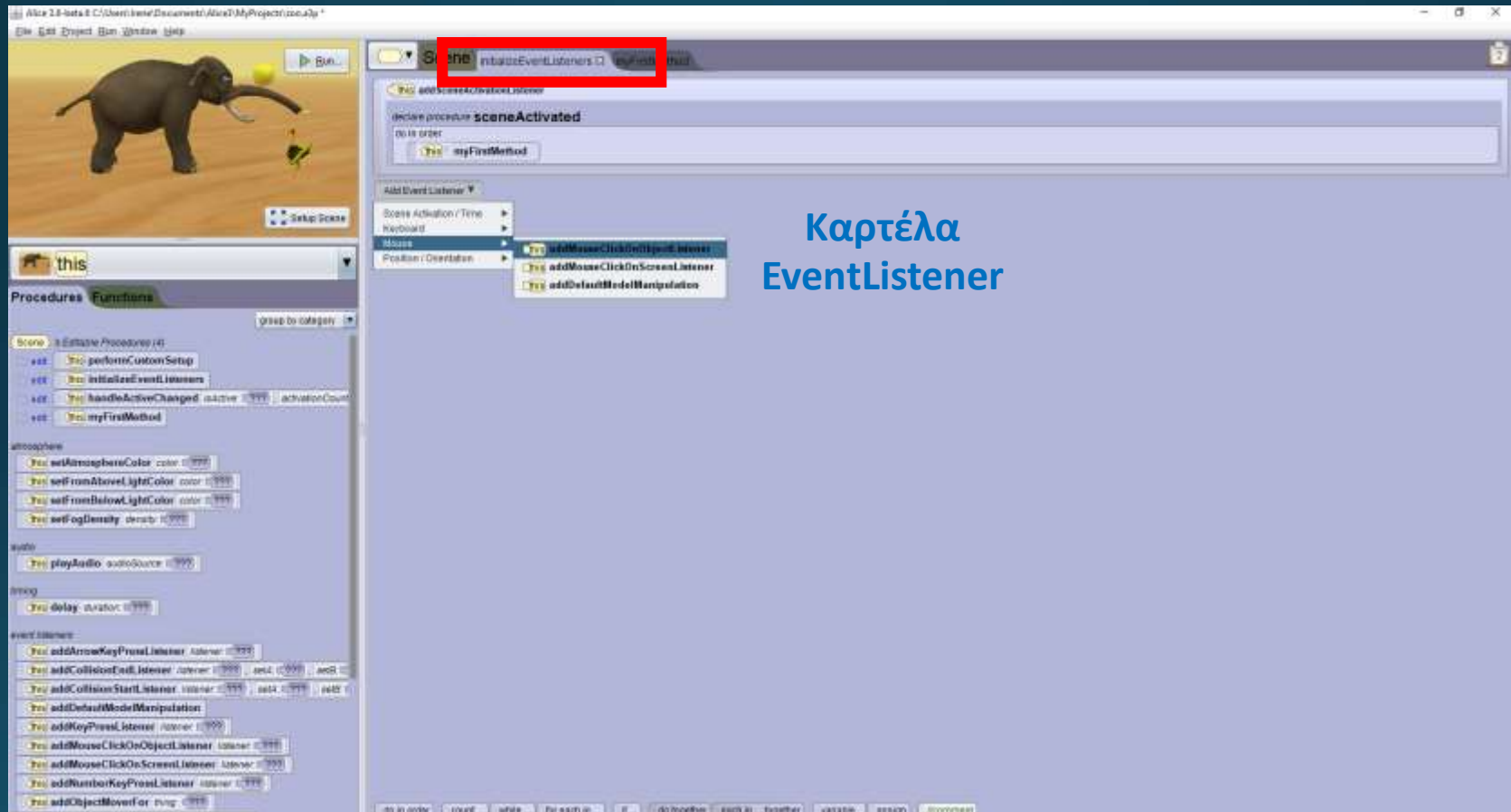
do in order

- my elephant getTail move RIGHT 1.0 duration: 2.0 add detail!
- my elephant getTail move LEFT 1.0 duration: 2.0 add detail!
- my penelope think OY O utdntep duration: 0.0 add detail!

- my penelope say Tux ou utdntep duration: 0.0 add detail!
- my penelope say Cov oc dntep duration: 0.0 add detail!
- my penelope say Cov oc dntep add detail!
- my setAtmosphereColor 0.0/0.0/0.0 add detail!
- my sphere setOpacity 1.0 duration: 0.0 add detail!

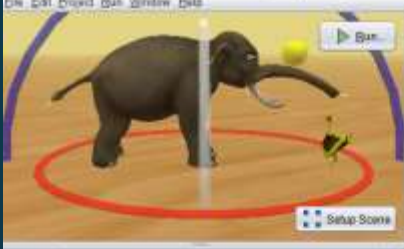
do in order count while for each in if do together each in together variable assign comment

Θέλω το ελεφαντάκι να κουνάει την ουρά του κάθε φορά που κάνω κλικ πάνω του!



Alice 3.0-beta.6 (C:\Users\jenf\Documents\Alice7\MyProject\zoo.a3p)

File Edit View Help Window Tools



Bar

Setup Scene

this.elefadaki

Procedures Functions

group by category

Elephant 3 Editable Procedures (1)

- setElephantResource resourceResource (???)

Sound 3 Editable Procedures (0)

think

- say text (???)
- think text (???)

motion

- move direction (???) amount (???)
- moveToward target (???) amount (???)
- moveAwayFrom target (???) amount (???)
- moveTo target (???)
- place spatialRelation (???) target (???)

orientation

- lean direction (???) amount (???)
- roll direction (???) amount (???)
- turnToFace target (???)
- orientTo target (???)
- orientToUpright
- pointAt target (???)

color & orientation

- moveAndOrientTo target (???)

size

- setWidth width (???)

Scene initializeEventListeners myFirstMethod

addSceneActivationListener

sceneActivated

Go in order

- myFirstMethod

addMouseEventListener add drag!

mouseClicked

- event.getGlobalMouseLocation
- event.getScreenDistanceFromLeft
- event.getScreenDistanceFromBottom

Go in order

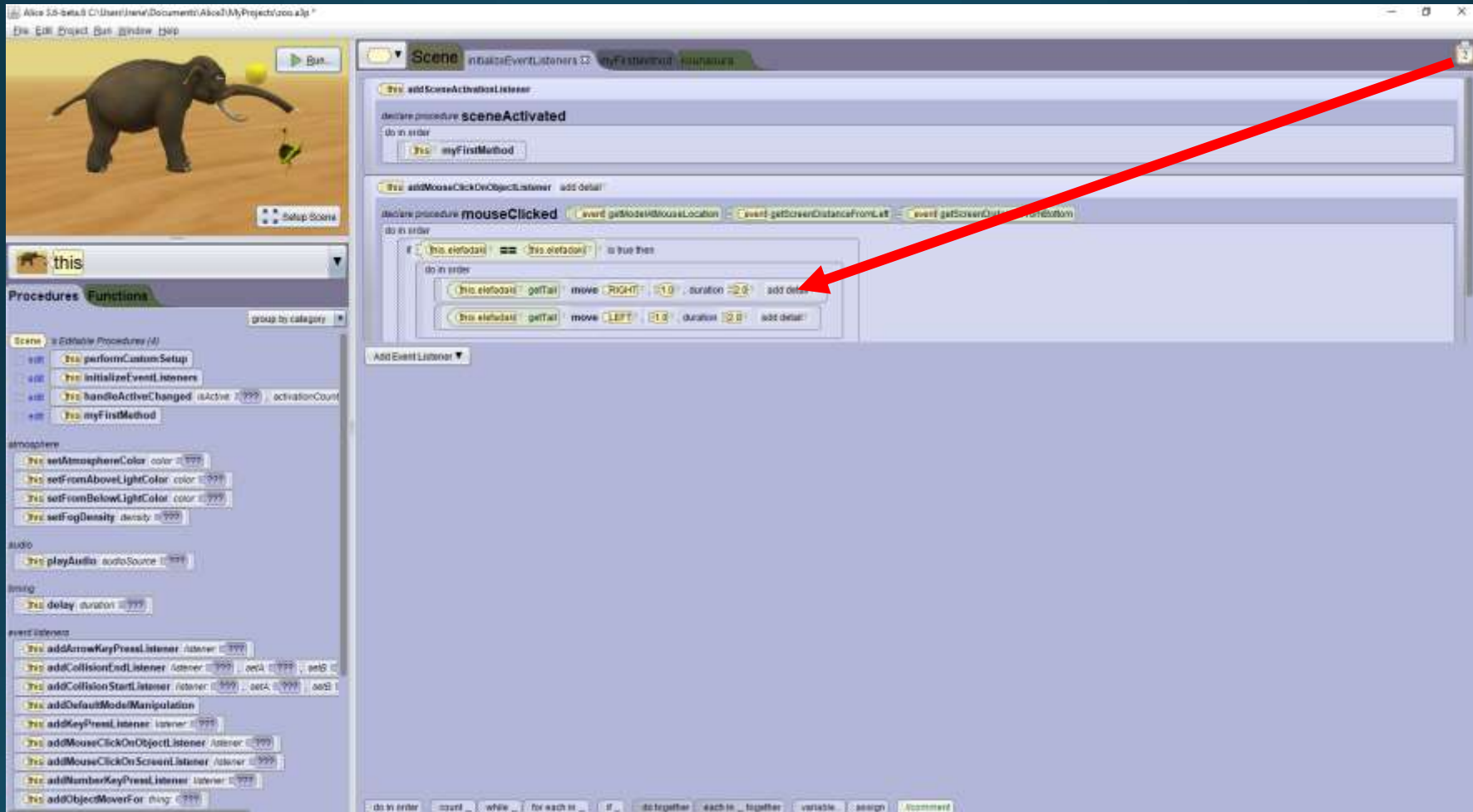
if (???) is true then

ADD

if

do in order count while for each if together each with together variable assign statement

File Edit Project Run Window Help



The screenshot shows the Unity 5.0-beta.3 interface. On the left, there is a scene view showing an elephant in a desert environment. Below the scene view is a console window with a search bar and a list of procedures and functions. The main area is the Hierarchy and Inspector, showing the 'Scene' object selected. The Inspector shows the 'Scene' object's properties and a script named 'Scene'. The script has two procedures: 'sceneActivated' and 'mouseClicked'. The 'mouseClicked' procedure has a conditional block that checks if the mouse is over the elephant and then moves the elephant to the right or left. A red arrow points to the 'move' block in the 'mouseClicked' procedure.

Scene

addSceneActivationListener

declare procedure sceneActivated

do in order

- myFirstMethod

addMouseClickOnObjectListener add detail

declare procedure mouseClicked

- event getMouseMouseLocation
- event getScreenDistanceFromLit
- event getScreenDistanceFromPosition

do in order

- if [this elephant] == [this elephant] is true then
  - do in order
    - [this elephant] getTail move RIGHT -1.0 duration 2.0 add detail
    - [this elephant] getTail move LEFT -1.0 duration 2.0 add detail

Add Event Listener

do in order count \_ while \_ for each in \_ if \_ do together each is \_ together variable assign comment



this.pinelopi

Procedures Functions

- Scene**
  - Settable Procedures (1)**
    - this.pinelopi.setDefaultResource: obj:objResource: 000
  - Key Bind**
    - this.pinelopi.key: key: 000
    - this.pinelopi.think: key: 000
  - motion**
    - this.pinelopi.move: direction: 000, amount: 000
    - this.pinelopi.moveToward: target: 000, amount: 000
    - this.pinelopi.moveAwayFrom: target: 000, amount: 000
    - this.pinelopi.moveTo: target: 000
    - this.pinelopi.place: quat:Quaternion: 000, target: 000
  - orientation**
    - this.pinelopi.turn: direction: 000, amount: 000
    - this.pinelopi.roll: direction: 000, amount: 000
    - this.pinelopi.lookToFace: target: 000
    - this.pinelopi.orientTo: target: 000
    - this.pinelopi.adjustUpAngle
    - this.pinelopi.pointAt: target: 000
  - position & orientation**
    - this.pinelopi.moveAndOrientTo: target: 000
  - size**
    - this.pinelopi.setWidth: width: 000
    - this.pinelopi.setHeight: height: 000

Scene initializeEventListeners myFirstMethod

```

addSceneActivatedListener
declare procedure sceneActivated
do in order
  this.myFirstMethod

addMouseButtonObjectListener add-data?
declare procedure mouseClicked
  event: getMouseButtonLocation | event: getScreenDistanceFromLeft | event: getScreenDistanceFromBottom
  if event.getMouseButtonLocation == this.stacked then
    this.stacked.getTail.move(RIGHT, 0.1, rotation: 0.1, add-data?)
    this.stacked.getTail.move(LEFT, 0.1, rotation: 0.1, add-data?)
  else
    stop statement here
  end if
end procedure
  
```

ADD Event Listener

do in order | count | while | for each in | if | do together | each in | together | variate | assign | comment

Θέλω η στρουθοκάμηλος να γίνεται κόκκινη κάθε φορά που κάνω κλικ πάνω της!

The screenshot shows the Unity Inspector window for a scene named 'Scene'. The main view displays a 3D scene with an elephant and a giraffe. The 'Scene' panel on the right shows a 'mouseClicked' event listener for the giraffe, which is highlighted with a red box. The event listener is set to 'is hit by' and triggers 'setParent' for the giraffe to 'RED' and 'YELLOW'. The 'this.pinelopi' object is selected in the Hierarchy panel. The 'Procedures' panel shows various functions for the giraffe, including 'turn', 'walk', 'look', 'move', 'setParent', 'setVolume', 'setOpacity', and 'setVelocity'.



Δημιουργήστε τη δική σας διαδραστική  
ιστορία διάρκειας 1 Λεπτού







# Τέλος Ενότητας



  
Ευρωπαϊκή Ένωση  
European Union

  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
ανάπτυξη στην κοινωνία της γνώσης  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΘΡΗΣΚΕΥΜΑΤΩΝ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

  
ΕΣΠΑ  
2007-2013  
Ευρωπαϊκό Κοινωνικό Ταμείο  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης.