



## Εγχειρίδιο Εκμάθησης Squeak / Squeak Etoys

Ομάδα Εργασίας  
Βασίλειος Πουλόπουλος  
Ιωάννης Παππάς

Πάτρα 2007

## Πρόλογος

Το παρόν κείμενο είναι ένα εγχειρίδιο εκμάθησης του Squeak / Squeak e-toys για αρχάριους χρήστες. Αποτελείται από απλά παραδείγματα που οδηγούν το χρήστη στην βήμα προς βήμα δημιουργία εφαρμογών οι οποίες μπορούν να αποτελέσουν τη βάση για κατασκευή σύνθετων εκπαιδευτικών προγραμμάτων. Σκοπός του παρόντος κειμένου είναι η εξοικείωση με το περιβάλλον του Squeak / Squeak e-Toys προκειμένου η κοινότητα που θα εμπλακεί με το OLPC (μαθητές, δάσκαλοι, καθηγητές) να είναι σε θέση να συμμετέχει ενεργά στην ανάπτυξη και εξάπλωση των εφαρμογών του.

Το εγχειρίδιο εκμάθησης αποτελείται από 4 βασικές ενότητες. Στην πρώτη ενότητα γίνεται συνοπτική παρουσίαση του squeak αλλά και της γλώσσας smalltalk πάνω στην οποία βασίζεται. Η δεύτερη ενότητα περιέχει στοιχεία για την εύκολη και γρήγορη εγκατάσταση του Squeak e-toys. Η τρίτη ενότητα κάνει συνοπτική παρουσίαση των βασικότερων στοιχείων του περιβάλλοντος squeak e-toys προκειμένου ο χρήστης να γνωρίσει πώς μπορεί να χειρίζεται τα βασικά στοιχεία του περιβάλλοντος. Εν συνεχεία, υπάρχει η τέταρτη και βασικότερη ενότητα του εγχειριδίου η οποία περιέχει τα απλά παραδείγματα ακολουθούμενα από εικόνες που δείχνουν την βήμα-βήμα κατασκευή απλών προγραμμάτων σε περιβάλλον squeak / squeak e-toys και μπορούν να χρησιμοποιηθούν για το olpc.

Για την κατασκευή του παρόντος εγχειριδίου εργάστηκε εθελοντικά ομάδα από το Πανεπιστήμιο Πατρών και από το Ερευνητικό Ακαδημαϊκό Ινστιτούτο Τεχνολογίας Υπολογιστών.

Περισσότερες πληροφορίες για την ομάδα μας μπορείτε να βρείτε στο δικτυακό τόπο <http://olpc.cti.gr>.

Επίσης σας προτείνουμε να δείτε και τη σελίδα της ομάδας πρωτοβουλίας για την ελληνοποίηση του μαθητικού υπολογιστή η οποία περιέχει πληθώρα πληροφοριών για τις δράσεις που λαμβάνουν χώρα σε όλη την Ελλάδα. <http://olpc.ellak.gr>.

Οι πληροφορίες που περιέχονται στο παρόν εγχειρίδιο μπορούν να χρησιμοποιηθούν και να διακινηθούν ελεύθερα αρκεί να γίνεται αναφορά στην πηγή και να μη χρησιμοποιηθούν για εμπορικούς σκοπούς.

Πάτρα, Φεβρουάριος 2007

## Περιεχόμενα

Πρόλογος.....	2
Περιεχόμενα .....	3
Λίστα Σχημάτων .....	4
1. Εισαγωγή .....	7
1.1. Χαρακτηριστικά του Squeak .....	7
1.2. Smalltalk .....	8
1.3. Squeak και oIrc .....	9
2. Εγκατάσταση του Squeak e-toys.....	10
3. Εξοικείωση με το περιβάλλον.....	12
3.1. Βασικά Εργαλεία.....	12
3.2. Δημιουργία-Αποθήκευση Αρχείου .....	18
3.3. Άνοιγμα Αρχείου.....	22
3.4. Workspace και Transcript .....	24
3.5. Δημιουργώντας Σημειώσεις .....	28
4. Παραδείγματα Χρήσης για Αρχάριους.....	30
4.1. Το πρώτο μου πρόγραμμα .....	30
4.1.1. Κάνοντας το ίδιο με κώδικα! .....	36
4.2. Αντικείμενα, Οπτικός Προγραμματισμός .....	38
4.2.1. Εκτέλεση με τη χρήση κουμπιού .....	42
4.2.2. Αντιγραφή και σβήσιμο του περιεχομένου του Obj1 .....	43
4.3. Φτιάχνοντας μία αριθμομηχανή.....	44
4.3.1. Επέκταση αριθμομηχανής με περισσότερες πράξεις .....	46
4.4. Φτιάχνοντας ένα κουίζ.....	46
4.5. Σχεδιάζοντας Δυσδιάστατα Αντικείμενα .....	52
4.5.1. Φτιάχνοντας το πρώτο μου αυτοκίνητο .....	54
4.6. Βάζοντας Κίνηση σε Αντικείμενα .....	57
4.7. Δημιουργώντας «συγκρούσεις» .....	59
4.8. Κινούμενα σχέδια .....	63
4.8.1. Ζωγραφίζοντας την κίνηση μιας μπάλας μπάσκετ.....	67
4.9. Εισαγωγή έτοιμων εικόνων στο πρόγραμμα .....	78
4.10. Kedama-εξομοίωση κίνησης ατόμων.....	81
5. Αναφορές.....	87

## Λίστα Σχημάτων

Εικόνα 1	Πρώτη εικόνα διαδικασίας εγκατάστασης .....	10
Εικόνα 2:	Τελική εικόνα εγκατάστασης.....	11
Εικόνα 3:	Αρχική εικόνα του περιβάλλοντος του Squeak.....	12
Εικόνα 4	Καρτέλα Navigator.....	13
Εικόνα 5	Καρτέλα Supplies .....	13
Εικόνα 6:	Επεξήγηση συμβόλων της περιοχής Supplies .....	14
Εικόνα 7:	Επεξήγηση συμβόλων της περιοχής Navigator.....	14
Εικόνα 8:	Ανάλυση των πιο συχνών εργαλείων του Squeak .....	15
Εικόνα 9:	Επιλογές ενεργειών αντικειμένου .....	16
Εικόνα 10:	Παλέτα ζωγραφικής .....	17
Εικόνα 11:	Εργαλεία κατασκευής κώδικα .....	17
Εικόνα 12:	Κατάλογος αντικειμένων του Squeak .....	18
Εικόνα 13:	Κεντρική Σελίδα του Squeak .....	19
Εικόνα 14:	Δημιουργία αντικειμένου Text.....	20
Εικόνα 15:	Εργαλεία επιλογών για το αντικείμενο Text .....	20
Εικόνα 16:	Αποθήκευση αρχείου – βήμα 1.....	21
Εικόνα 17:	Αποθήκευση αρχείου – βήμα 2.....	21
Εικόνα 18:	Το αρχείο που δημιουργήσαμε αποθηκευμένο στο σκληρό δίσκο .....	22
Εικόνα 19:	Άνοιγμα Αρχείου – βήμα 1 .....	23
Εικόνα 20:	Άνοιγμα Αρχείου – βήμα 2 .....	23
Εικόνα 21:	Άνοιγμα Αρχείου – βήμα 3 .....	24
Εικόνα 22:	Το αποθηκευμένο αρχείο ανοίγει στην επιφάνεια εργασίας.....	24
Εικόνα 23:	Η καρτέλα των αντικειμένων του squeak .....	25
Εικόνα 24:	Επιλέγοντας την καρτέλα εργαλείων από τα αντικείμενα .....	25
Εικόνα 25:	Workspace και Transcript στην επιφάνεια εργασίας μας .....	26
Εικόνα 26:	Γράφοντας κώδικα στο Workspace .....	27
Εικόνα 27:	Εμφανίζοντας τα αποτελέσματα στο transcript.....	27
Εικόνα 28:	Το αποτέλεσμα της εκτέλεσης κώδικα από το workspace στο transcript.....	28
Εικόνα 29:	Το αποτέλεσμα της εκτέλεσης κώδικα από το workspace στο transcript (κώδικας με σχόλια) .....	29
Εικόνα 30:	Το κεντρικό γραφικό περιβάλλον του Squeak.....	30
Εικόνα 31:	Επιλογές που εμφανίζονται με αριστερό κλικ .....	31
Εικόνα 32:	Επιλογές που εμφανίζονται με δεξί κλικ.....	31
Εικόνα 33:	Τα αντικείμενα (objects) του Squeak .....	32
Εικόνα 34:	Επιλέγοντας τα αντικείμενα τύπου Text (κείμενο) .....	32
Εικόνα 35:	Επιλέγουμε το αντικείμενο Text και το σύρουμε στην επιφάνεια εργασίας .....	33
Εικόνα 36:	Εμφανίζοντας τον κέρσορα γραψίματος.....	33
Εικόνα 37:	Γεια σου κόσμε!.....	33
Εικόνα 38:	Στοιχεία που αφορούν το αντικείμενο κειμένου "hello world!"	34
Εικόνα 39:	Σύροντας το κουμπί size από τα attributes ενός αντικειμένου	34
Εικόνα 40:	το attribute «μάτι» .....	34
Εικόνα 41:	Επεξήγηση attributes ενός αντικειμένου .....	35
Εικόνα 42:	Εμφανίζοντας το attribute Text's character στην επιφάνεια εργασίας του squeak.....	36
Εικόνα 43:	Το text's character στην επιφάνεια εργασίας .....	36
Εικόνα 44:	τα στοιχεία του κουμπιού Text .....	36
Εικόνα 45:	Επιλογή show code textually .....	37

Εικόνα 46: Επιλέγοντας την αποδοχή για το καινούριο κείμενο .....	37
Εικόνα 47: Αποδοχή και εφαρμογή της αλλαγής που κάναμε .....	37
Εικόνα 48: Κλείσιμο του παραθύρου .....	38
Εικόνα 49: Δημιουργώντας ένα αντικείμενο κειμένου με πλαίσιο .....	38
Εικόνα 50: Αλλάζοντας το όνομα του αντικειμένου .....	39
Εικόνα 51: Αντικείμενα τύπου text (border) τοποθετημένα στον κόσμο ...	39
Εικόνα 52: Εμφανίζοντας το Obj2 characters στον κόσμο (α) .....	40
Εικόνα 53: Εμφανίζοντας το Obj2 characters στον κόσμο (β) .....	40
Εικόνα 54: Τοποθετώντας το Obj1's characters στο abc του Obj2's characters.....	41
Εικόνα 55: Αποτέλεσμα τοποθέτησης Obj1's characters στο abc του Obj2's characters.....	41
Εικόνα 56: Γράφουμε Hello everyone στο Obj1 .....	41
Εικόνα 57: Αντιγραφή των χαρακτήρων του Obj1 στο Obj2 .....	42
Εικόνα 58: Δημιουργώντας ένα κουμπί για την εκτέλεση του script .....	42
Εικόνα 59: Το κουμπί για την εκτέλεση του κώδικα.....	42
Εικόνα 60: Ανοίγοντας τον κώδικα του Script1.....	43
Εικόνα 61: Κατασκευή τριών text (border) .....	44
Εικόνα 62: Τοποθέτηση του attribute numericValue στον κόσμο .....	45
Εικόνα 63: Σύνδεση Number1 με το Result .....	45
Εικόνα 64: Συνδεδεμένα Result, Number1 και Number2.....	46
Εικόνα 65: Έξι αντικείμενα Text για τη δημιουργία quiz.....	47
Εικόνα 66: επιλέγοντας τα miscellaneous attributes.....	48
Εικόνα 67: Δημιουργώντας ένα νέο Script.....	48
Εικόνα 68: Τα miscellaneous attributes του αντικειμένου Wrong.....	49
Εικόνα 69: προσθέτοντας ένα attribute μέσα στο script .....	49
Εικόνα 70: Το script μίας εκ των απαντήσεων (λάθος απάντηση) .....	49
Εικόνα 71: Τα scripts όλων των απαντήσεων .....	50
Εικόνα 72: Δημιουργία κουμπιού για το script.....	50
Εικόνα 73: Ερωτήσεις με τις απαντήσεις τους.....	51
Εικόνα 74: Το script και το κουμπί του script για το αντικείμενο question. ....	51
Εικόνα 75: Εκτέλεση του test.....	52
Εικόνα 76: Ανοίγοντας το παράθυρο ζωγραφικής .....	53
Εικόνα 77: Παράθυρο ζωγραφικής .....	54
Εικόνα 78: Σχεδιάζοντας το πρώτο μου αυτοκίνητο βήμα 1 .....	55
Εικόνα 79: Σχεδιάζοντας το πρώτο μου αυτοκίνητο βήμα 2 .....	55
Εικόνα 80: Σχεδιάζοντας το πρώτο μου αυτοκίνητο βήμα 3 .....	56
Εικόνα 81: αυτοκίνητο σχεδιασμένο με έτοιμα σχήματα .....	56
Εικόνα 82: ολοκλήρωση σχεδίου με προσθήκη χρώματος (γέμισμα) .....	57
Εικόνα 83: Αυτοκίνητο και σχέδιο/πλήκτρο μετακίνησης.....	57
Εικόνα 84: Διασυνδέοντας αυτοκίνητο και βέλος .....	58
Εικόνα 85: Επιλέγοντας πως θα εκτελείται το script. ....	59
Εικόνα 86: Αυτοκίνητο, βέλος, δρόμος και τοίχος .....	60
Εικόνα 87: διασυνδέοντας το Car με το Wall.....	61
Εικόνα 88: βάζοντας το script σε loop.....	62
Εικόνα 89: το φορτηγό συγκρούεται με τον τοίχο και τον καταστρέφει ...	62
Εικόνα 90: Ο κόσμος με δύο αντικείμενα: Holder και Playfield .....	63
Εικόνα 91: Τα δύο αντικείμενα στο Holder και το αντιγραμμένο αντικείμενο στο playfield .....	64
Εικόνα 92: Άνοιγμα επιλογών του αντικειμένου στο playfield.....	64
Εικόνα 93: Δημιουργία νέου script για τα κινούμενα σχέδια .....	65

Εικόνα 94: Βάζοντας το player at cursor μέσα στο script .....	65
Εικόνα 95: Επιλέγοντας το attribute cursor .....	66
Εικόνα 96: Προσθέτοντας το attribute cursor increase by μέσα στο script .....	66
Εικόνα 97: Το κινούμενο σχέδιο εν δράσει... .....	67
Εικόνα 98: Επιλέγοντας τα εργαλεία ζωγραφικής από το Squeak Etoys ...	67
Εικόνα 99: Σχεδιασμός μίας μπάλας μπάσκετ .....	68
Εικόνα 100: Επιλέγοντας τα αντικείμενα playfield από το supplies.....	68
Εικόνα 101: Τοποθετώντας τη μπάλα στο ένα αντικείμενο playfield .....	69
Εικόνα 102: Δημιουργία νέου αντικειμένου ζωγραφικής: δημιουργία τελείας .....	69
Εικόνα 103: Εμφανίζουμε τις επιλογές αντικειμένου και επιλογή του μενού .....	70
Εικόνα 104: Το μενού του αντικειμένου playfield.....	70
Εικόνα 105: Επιλογή του playfield options.....	71
Εικόνα 106: Επιλογή του Use Standard Texture .....	71
Εικόνα 107: Playfield με standard texture .....	72
Εικόνα 108: Τοποθέτηση του αντικειμένου (τελεία) μέσα στο playfield και μετονομασία του αντικειμένου .....	72
Εικόνα 109: Αλλαγή του grapher's heading .....	73
Εικόνα 110: Δημιουργία script και μετονομασία του .....	73
Εικόνα 111: Τοποθετώντας ένα attribute forward by μέσα στο script .....	74
Εικόνα 112: Επιλογή του grapher's γ.....	74
Εικόνα 113: Τοποθέτηση του grapher's γ μέσα στο script .....	75
Εικόνα 114: Επιλογή του basetball γ .....	75
Εικόνα 115: Τοποθέτηση του basetball γ μέσα στην τιμή του grapher's γ	76
Εικόνα 116: Κατηγορία pen use από τα attributes του grapher .....	76
Εικόνα 117: Επιλογή του attribute penDown .....	77
Εικόνα 118: Τοποθέτηση του attribute penDown μέσα στο script .....	77
Εικόνα 119: Αναπαράσταση της κίνησης της μπάλας .....	78
Εικόνα 120: Κρατώντας πατημένο το αριστερό κλικ πάνω στο Find.....	79
Εικόνα 121: Καρτέλα επιλογής αρχείου για άνοιγμα .....	80
Εικόνα 122: Επιλογή εικόνας για άνοιγμα .....	80
Εικόνα 123: Η εικόνα που επιλέξαμε μέσα στο περιβάλλον του squeak ...	81
Εικόνα 124: Δημιουργώντας ένα Kedama-World αντικείμενο .....	82
Εικόνα 125: Το εικονίδιο του μενού των attributes .....	83
Εικόνα 126: Επιλογή του add a new breed of turtle .....	83
Εικόνα 127: Το «χρωματιστό κουτί» .....	84
Εικόνα 128: Επιλέγοντας το Kedama Turtle Breed .....	85
Εικόνα 129: Πολλές κόκκινες χελώνες μέσα στο Kedama World.....	86

## 1. Εισαγωγή

Το Squeak είναι μια μοντέρνα, ανοιχτού λογισμικού υλοποίηση της γλώσσας προγραμματισμού Smalltalk. Το Squeak είναι ιδιαίτερα φορητό ως πρόγραμμα ενώ το γεγονός ότι η ιδεατή μηχανή του είναι και αυτή γραμμένη σε Smalltalk μας δίνει τη δυνατότητα για εύκολο debugging, ανάλυση και αλλαγή. Αποτελεί ένα εργαλείο για ένα μεγάλο εύρος από έργα που αφορούν πολυμεσικές εφαρμογές, εκπαιδευτικές πλατφόρμες μέχρι και ανάπτυξη εμπορικών Web εφαρμογών.

Το Squeak είναι ελεύθερα διαθέσιμο διαμέσου του διαδικτύου. Κάθε διάθεση του προγράμματος περιλαμβάνει υποστήριξη για χρώματα, ήχο και πρόσβαση στο δίκτυο ανεξαρτήτως πλατφόρμας ενώ παράλληλα περιλαμβάνει ολόκληρο τον κώδικά του. Επεκτείνει την θεμελιώδη φιλοσοφία της Smalltalk στην οποία όλα είναι διαθέσιμα να τα δεις, να τα κατανοήσεις, να τα αλλάξεις, και να τα επεκτείνεις για οποιοδήποτε σκοπό. Είναι ένα αυθεντικό, ολοκληρωμένο, συμπαγές, αποτελεσματικό και δυνατό περιβάλλον της Smalltalk. Δεν εξειδικεύεται για κάποιο συγκεκριμένο hardware ή λειτουργικό σύστημα. Έτσι λοιπόν το Porting, δηλαδή η μεταφορά του προγράμματος από ένα λειτουργικό σύστημα σε ένα άλλο, γίνεται εύκολα.

Το Squeak είναι μια πρακτική Smalltalk στο οποίο ο ερευνητής, ο καθηγητής ή ο μαθητής μπορεί να εξετάσει τον πηγαίο κώδικα οποιοδήποτε μέρους του συστήματος ακόμα και την ίδια την ιδεατή μηχανή. Έτσι κάποιος μπορεί να κάνει αλλαγές αμέσως χωρίς την ανάγκη να δει ή να ασχοληθεί με άλλη γλώσσα πέρα από την Smalltalk. Επίσης τρέχει ίδιου format εικόνες σε ολόκληρη τη φορητή του βάση, επιτυγχάνοντας τεράστια συνεργασία σε διαφορετικά περιβάλλοντα. Οποιοδήποτε αρχείο εικόνας μπορεί να τρέχει σε οποιοδήποτε διερμηνευτή(interpreter) ακόμα και αν έχει σωθεί σε τελείως διαφορετικό hardware με τελείως διαφορετικό λειτουργικό σύστημα.

Στη αναφορά αυτή θα γίνει μια εισαγωγή στο περιβάλλον Squeak e-toys καθώς και στον τρόπο ανάπτυξης απλών εφαρμογών. Το περιβάλλον Squeak e-toys έχει εγκατασταθεί στο λειτουργικό σύστημα του OLPC και είναι ιδιαίτερα εύχρηστο για ακόμα και για παιδιά μικρής ηλικίας.

### 1.1. Χαρακτηριστικά του Squeak

Τα χαρακτηριστικά του Squeak είναι εκπληκτικά. Έχει όλα τα χαρακτηριστικά ενός μοντέρνου περιβάλλοντος εργασίας ενώ χρησιμοποιείται για την εκπαίδευση, για παιχνίδια, internet, πολυμέσα, και έρευνα. Όσο αφορά τη γλώσσα Smalltalk είναι μια γλώσσα καθαρά αντικειμενοστραφής, διερμηνευόμενη(interpreted), ανοιχτού κώδικα και επεκτάσιμη. Όσο αφορά την ιδεατή μηχανή του Squeak, έχει αναπτυχθεί χρησιμοποιώντας το ίδιο το περιβάλλον Squeak. Έτσι είναι δυνατόν να κάνεις debug και να εκτελέσεις το Squeak μέσα από το ίδιο το Squeak. Επίσης, μπορεί να χρησιμοποιηθεί για την ανάπτυξη Web εφαρμογών. Ένα άλλο πλεονέκτημα είναι η δυνατότητά του να υποστηρίζει Unicode αλλά και πολύγλωσσες εφαρμογές. Από μόνο του αποτελεί ένα IDE, πράγμα που σημαίνει ότι μπορείς να γράφεις κώδικα, να τον τρέξεις αλλά και να τον κάνεις debug στο ίδιο περιβάλλον. Τέλος να αναφερθεί ότι χρησιμοποιεί αυξανόμενο compiling δηλαδή με την αποθήκευση του κώδικα, είναι έτοιμος για να τον τρέξεις.

Πιο συγκεκριμένα, το Squeak ως περιβάλλον ανάπτυξης κώδικα προσφέρει τονισμό στον κώδικα ενώ επίσης ο κώδικας που δεν είναι έγκυρος, τονίζεται και αυτός. Όσο αφορά τον Smalltalk compiler είναι build in και κάνει αυξητικό compiling καθώς φτιάχνεις κώδικα.

Όσο αφορά το εκπαιδευτικό κομμάτι του Squeak, υπάρχει το Squeakland (<http://www.squeakland.org/>). Το Squeakland είναι μια σελίδα στην οποία υπάρχουν έτοιμες εφαρμογές οι οποίες έχουν κατασκευαστεί είτε από ενήλικες είτε από παιδιά. Οι εφαρμογές αυτές έχουν κατασκευαστεί είτε για παιχνίδι είτε για εκπαίδευση είτε και για τα δύο. Έτσι το squeak είναι ένα αρκετά καλό παιχνίδι ανάπτυξης πλατφόρμας.

Επίσης παρέχεται προγραμματιστική πρόσβαση στο διαδικτυακό περιβάλλον του υπολογιστή σε όλα τα επίπεδα, από το πιο υψηλό επίπεδο του πέρασματος αντικειμένων και των πρωτοκόλλων όπως το HTTP/FTP μέχρι το χαμηλό επίπεδο της επικοινωνίας με sockets. Τα υπάρχοντα sockets είναι σχεδιασμένα να υποστηρίζουν τα TCP/IP και UDP πρωτόκολλα καθώς και υψηλότερου επιπέδου πρωτόκολλα όπως το POP3(mail retrieval), το SMTP(mail posting), το HTTP(web browsing), και το NTP(network news). Επίσης στο squeak περιλαμβάνονται χαρακτηριστικά τα οποία του δίνουν τη δυνατότητα να έχει πρόσβαση σε υπηρεσίες του διαδικτύου. Τα προγράμματα αυτά είναι:

- Celeste: πρόγραμμα ανάγνωσης e-mail
- Scamper: φυλλομετρητής
- Telnet: βασικός telnet client
- IRC: IRC client
- Nebraska: shared Morphic world, server and client
- Seaside: πρόγραμμα για την ανάπτυξη δυναμικών πολύπλοκων web εφαρμογών

Το Squeak μπορεί να παρουσιάσει την πληροφορία από μια ποικιλία format(ήχος, εικόνα, αλληλεπίδραση και κινούμενα σχέδια). Προσφέρει επεξεργασία σε 2D γραφικά ενώ προσφέρει τη δυνατότητα δημιουργίας 3D χώρων. Για την αλληλεπίδραση υπάρχει το πρόγραμμα Morphic το οποίο είναι ένα πρόγραμμα διαχείρισης της κατασκευής του User Interface (UI) στηριζόμενο στα display trees. Για τον ήχο έχει τη δυνατότητα να παίξει μια ποικιλία από διαφορετικά αρχεία ήχου καθώς επίσης να αναπαράγει πολλαπλά όργανα με πολλαπλές φωνές ταυτόχρονα. Επίσης υποστηρίζει mpeg2 και ένα εσωτερικό jpg συμπιεσμένο video format.

Τέλος, αξίζει να αναφερθεί ότι το Squeak είναι ένα πολύ ευέλικτο περιβάλλον που επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν τεχνικές όπως η εξερεύνηση διαφορετικών προγραμματιστικών παραδειγμάτων.

### **1.2. Smalltalk**

Η Smalltalk είναι μια καθαρά αντικειμενοστραφής γλώσσα, απλή και ομοιόμορφη. Η Smalltalk επηρέασε τις περισσότερες από τις μοντέρνες αντικειμενοστραφείς γλώσσες παρόλο που με τον καιρό έχασαν την κομψότητα και την απλότητα της.

Το συντακτικό της γλώσσας αλλά και το μοντέλο αντικειμένων είναι ιδιαίτερα απλό:

- Όλα είναι ένα αντικείμενο.
- Τα αντικείμενα επικοινωνούν μεταξύ τους με το πέρασμα μηνυμάτων.

- Οι κλάσεις περιγράφουν με όρους κατάστασης(στιγμιότυπες μεταβλητές) και συμπεριφοράς(μέθοδοι) τα αντικείμενα που παράγουν.
- Όταν ένα αντικείμενο λαμβάνει ένα μήνυμα, η αντίστοιχη μέθοδος βρίσκεται στην κλάση (ή στην υπερκλάση) του ληφθέντα.
- Οι μέθοδοι είναι public.
- Οι στιγμιότυπες μεταβλητές(instance variables) είναι private.
- Οι κλάσεις κληρονομούν διαμέσου της απλής κληρονομικότητας.

Για περαιτέρω πληροφορίες για τη γλώσσα αλλά και για το ίδιο το πρόγραμμα υπάρχουν στο <http://www.squeak.org/Smalltalk/> και στο <http://wiki.squeak.org/squeak/471>.

### **1.3. Squeak και olpc**

Στο olpc έχει εγκατασταθεί το πρόγραμμα etoys που είναι ένα περιβάλλον για εκπαίδευση. Η βασική θεωρία είναι ότι τα παιδιά μαθαίνουν καλύτερα εάν συναντήσουν, παίξουν και κατασκευάσουν τις ιδέες κινητικά, οπτικά, ηχητικά και συμβολικά.

Το etoys έχει σχεδιαστεί έτσι ώστε τα παιδιά 8 μέχρι 12 ετών να μπορούν εύκολα να δημιουργήσουν projects και να εκφράσουν τις ιδέες τους σε ένα ευκολονόητο περιβάλλον. Το μεγάλο πλεονέκτημα του etoys είναι ότι οποιοδήποτε πρόγραμμα κατασκευάζεται στο squeak, μπορεί να τρέξει στο etoys. Αυτό σημαίνει ότι μια βασική πηγή για υλικό εκπαιδευτικού περιεχομένου αποτελεί η <http://www.squeakland.org/>. Δίνεται η δυνατότητα πέρα από την κατασκευή, της χρήσης φτιαγμένων ήδη προγραμμάτων για το etoys τα οποία έρχονται μαζί με το λειτουργικό. Επίσης έχει τη δυνατότητα εύρεσης αρχείου για εκτέλεση σε οποιοδήποτε σημείο του file system, γεγονός που σημαίνει ότι ο μαθητής μπορεί να κατεβάσει είτε από το διαδίκτυο είτε από κάποιο φίλο του ένα πρόγραμμα για το etoys, και στη συνέχεια να το τρέξει στο δικό olpc.

## 2. Εγκατάσταση του Squeak e-toys

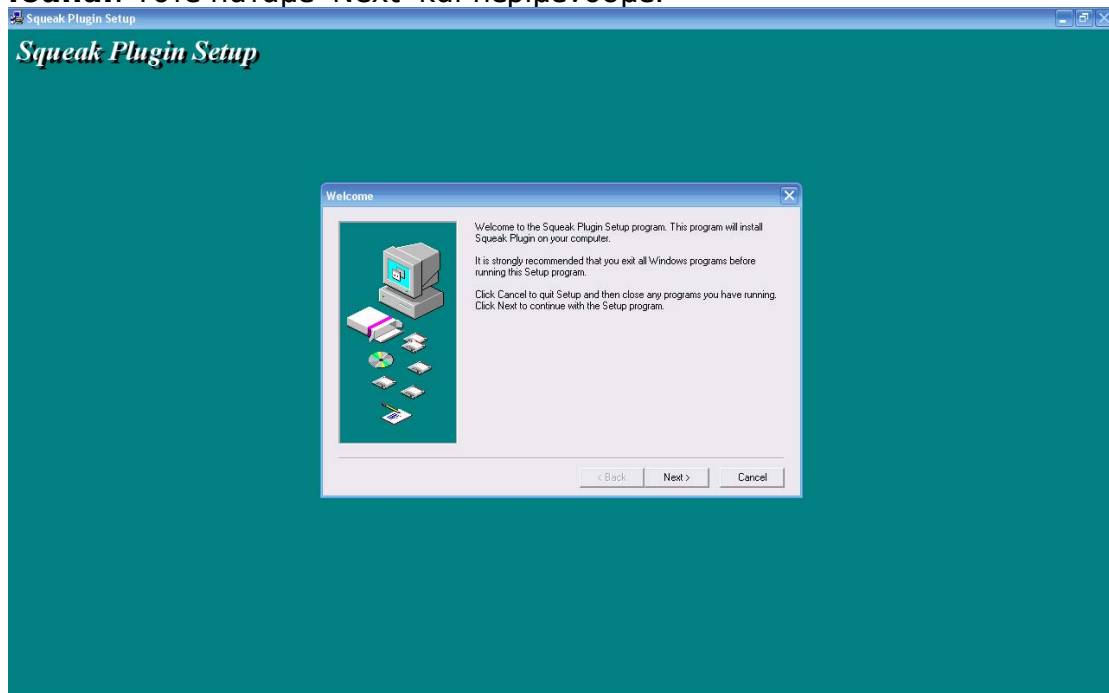
Για την ανάπτυξη εφαρμογών στο Etoys χρειάζεται η εγκατάσταση του προγράμματος στον υπολογιστή σας. Πηγαίνουμε με τη χρήση κάποιου φυλλομετρητή, με τη χρήση δηλαδή κάποιου προγράμματος όπως τον Internet explorer ή Mozilla Firefox, στην ιστοσελίδα [www.squeakland.org](http://www.squeakland.org).

Εκεί πατάμε πάνω στο εικονίδιο που λέει Download Squeak

Στη σελίδα στη οποία οδηγούμαστε, πατάμε με το δείκτη του ποντικιού την υπογραμμισμένη λέξη: Download Squeak.

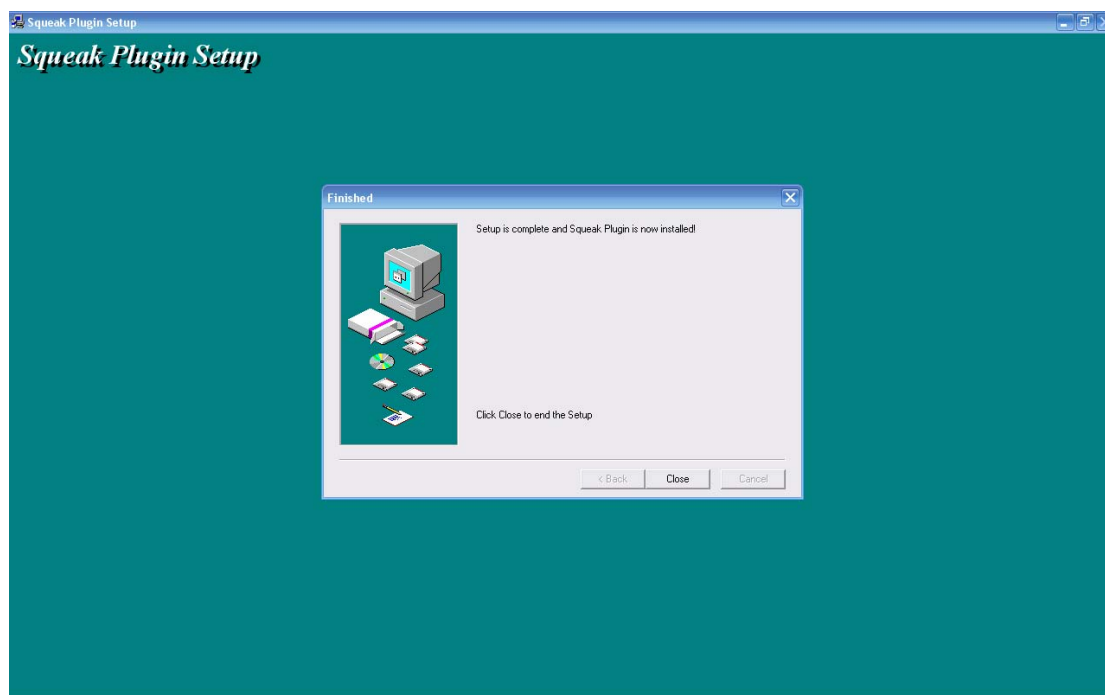
Στη συνέχεια αποθηκεύεται τοπικά στον υπολογιστή μας το αρχείο με το όνομα SqueakPluginInstaller.

Πατάμε διπλό κλικ πάνω στο αρχείο αυτό για να ξεκινήσουμε την εγκατάσταση και εμφανίζεται η εικόνα **1Error! Reference source not found..** Τότε πατάμε 'Next' και περιμένουμε.



**Εικόνα 1 Πρώτη εικόνα διαδικασίας εγκατάστασης**

Μόλις εμφανιστεί η εικόνα 2 πατάμε 'Close'. Εφόσον όλα έχουν πάει καλά, θα εμφανιστεί στην επιφάνεια εργασίας μας ένα εικονίδιο που λέγεται 'Squeak'. Κάνοντας διπλό κλικ στο εικονίδιο αυτό, ξεκινάμε το Etoys.



**Εικόνα 2: Τελική εικόνα εγκατάστασης**

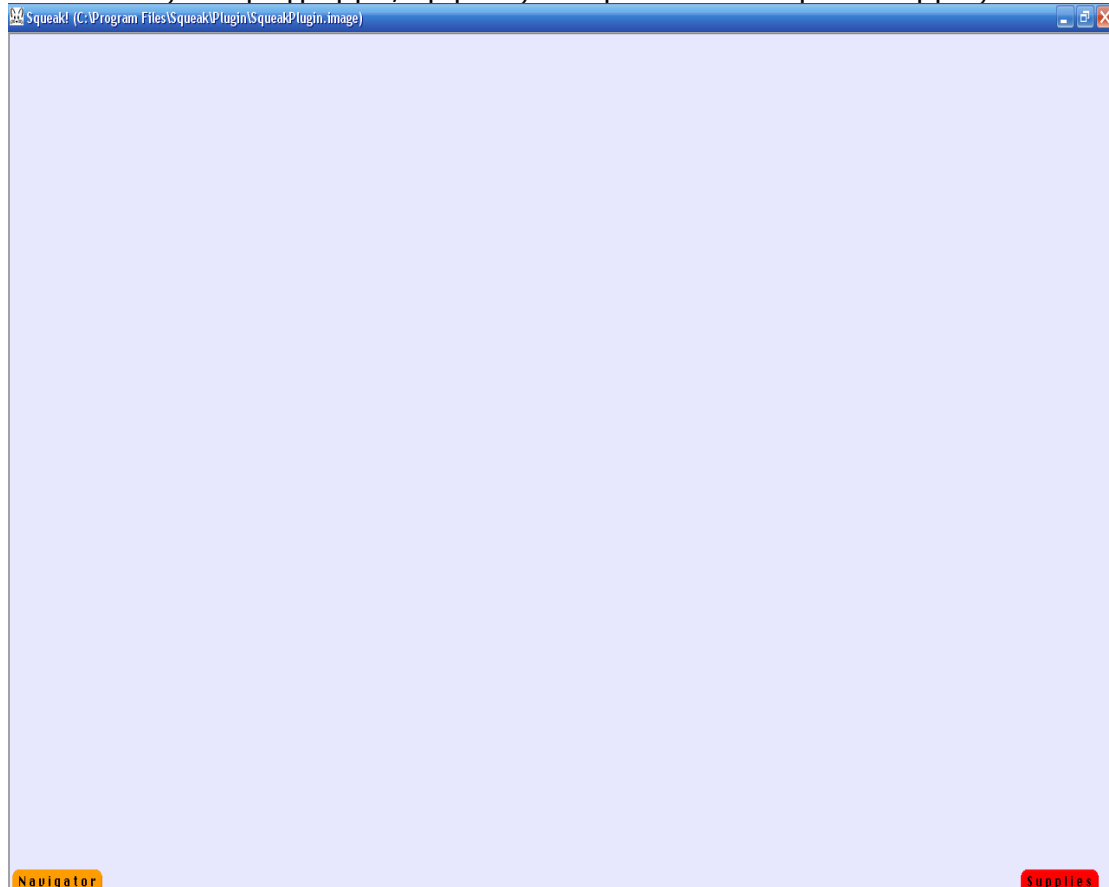
### 3. Εξοικείωση με το περιβάλλον

Θα ξεκινήσουμε με την περιγραφή του περιβάλλοντος του squeak. Πιο συγκεκριμένα θα γίνει επεξήγηση των εικονιδίων που υπάρχουν αλλά και των βασικών ενεργειών που μπορεί να κάνει ο χρήστης με αυτά.

#### 3.1. Βασικά Εργαλεία

Τα βασικά εργαλεία του squeak παρουσιάζονται στις παρακάτω εικόνες με την κατάλληλη επεξήγηση το καθένα.

Ξεκινώντας το πρόγραμμα, εμφανίζεται η εικόνα 3 στην οθόνη μας.



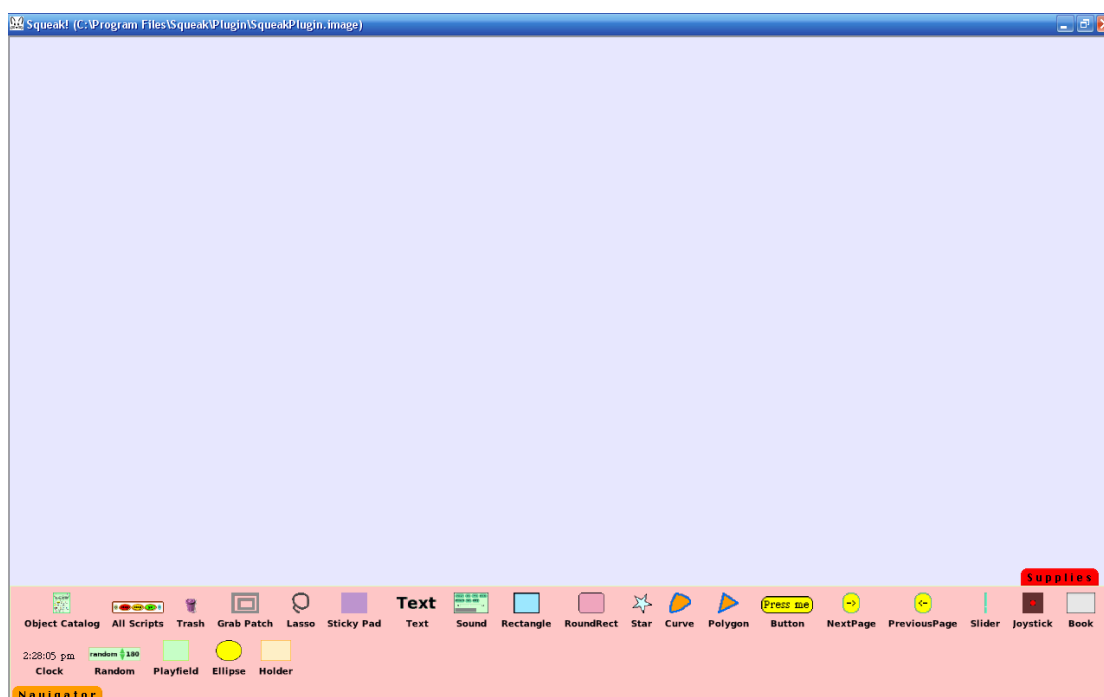
**Εικόνα 3: Αρχική εικόνα του περιβάλλοντος του Squeak**

Κάτω αριστερά και δεξιά υπάρχουν καρτέλες που πατώντας με απλό κλικ εμφανίζονται όπως φαίνονται στις εικόνες 4 και 5.

## Παραδείγματα Χρήσης για Αρχάριους



**Εικόνα 4 Καρτέλα Navigator**




















**Εικόνα 5 Καρτέλα Supplies**


















Οι δυνατότητες που μας παρέχουν τα εργαλεία αυτά, παρουσιάζονται σύντομα στις εικόνες 6 και 7.

**Supplies**

## Supplies Tab

 <small>Object Catalog</small>	<b>Object Catalog</b> - Περιήγηση στον κατάλογο με το υπάρχον περιεχόμενο	 <small>Star</small>	<b>Star</b> - Δημιουργία αντικειμένου με σχήμα άστρου
 <small>All Scripts</small>	<b>All Scripts</b> - Δείτε και ελέγξτε όλα τα scripts του project.	 <small>Curve</small>	<b>Curve</b> - Δημιουργία καμπυλόγραμμου σχήματος
 <small>Trash</small>	<b>Trash</b> - Διαγράψτε ένα αντικείμενο	 <small>Polygon</small>	<b>Polygon</b> - Δημιουργία πολυγώνου
 <small>Sticky Pad</small>	<b>Sticky Pad</b> - Διαφανή πολύγωνα μπορούν να αποκοπούν και να χρησιμοποιηθούν για μέτρηση	 <small>Button</small>	<b>Button</b> - Δημιουργία κουμπιού
 <small>Text</small>	<b>Text</b> - Κείμενο που μπορείτε να επεξεργαστείτε	 <small>NextPage</small>	<b>NextPage</b> - Μετακίνηση στην επόμενη σελίδα
 <small>Sound</small>	<b>Sound</b> - Εργαλείο για εγγραφή ήχου	 <small>PreviousPage</small>	<b>PreviousPage</b> - Μετακίνηση στην προηγούμενη σελίδα
 <small>Rectangle</small>	<b>Rectangle</b> - Δημιουργία ορθογωνίου σχήματος	 <small>Holder</small>	<b>Holder</b> - Χώρος για απεικόνιση εναλλακτικών εικόνων και κίνησης
 <small>RoundRect</small>	<b>RoundRect</b> - Δημιουργία τετράπλευρου με καμπύλα άκρα	 <small>Playfield</small>	<b>Playfield</b> - Χώρος για συναρμολόγηση τμημάτων ή για προβολή κίνησης (animation)
 <small>Ellipse</small>	<b>Ellipse</b> - Δημιουργία καμπυλόγραμμου σχήματος ή κύκλου		

**Supplies Flap**



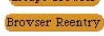






















**Supplies**

**Εικόνα 6: Επεξήγηση συμβόλων της περιοχής Supplies**

**Navigator**


## Navigator Tab

 <small>NEW</small>	<b>New</b> - Δημιουργία νέου project	 <small>FIND</small>	<b>Find</b> - Βρείτε ένα project. Κρατήστε το κουμπάκι πατημένο για περισσότερες επιλογές
 <small>Share</small>	<b>Share</b> - Μοιραστείτε το project με φίλους σας	 <small>Escape Browser</small>	<b>Escape Browser/Browser Reentry</b> - Αλλαγή μεταξύ Πλήρης Οθόνης και παραθύρου του φυλλομετρητή
 <small>PREV</small>	<b>Previous</b> - Ανοίξτε το προηγούμενο project	 <small>Browser Reentry</small>	<b>Hide Tabs</b> - Αφαιρέστε τις λωρίδες (tabs) από την οθόνη
 <small>NEXT</small>	<b>Next</b> - Ανοίξτε το επόμενο project	 <small>Painting</small>	<b>Painting</b> - Εμφάνιση υλικών σχεδιασμού
 <small>PUBLISH IT!</small>	<b>Publish it!</b> - Αποθήκευση του project στο σκληρό δίσκο ή σε server για μοίρασμά του	 <small>Sound</small>	<b>Sound</b> - Αλλαγή έντασης ήχου
 <small>Newer?</small>	<b>Newer?</b> - Έλεγχος για νεότερη έκδοση του project	 <small>Undo</small>	<b>Undo</b> - Ακύρωση τελευταίας εντολής
 <small>TELL</small>	<b>Tell!</b> - Πείτε σε ένα φίλο σας για το project	 <small>QUIT</small>	<b>Quit</b> - Έξοδος από το Squeak

**Navigator Flap**

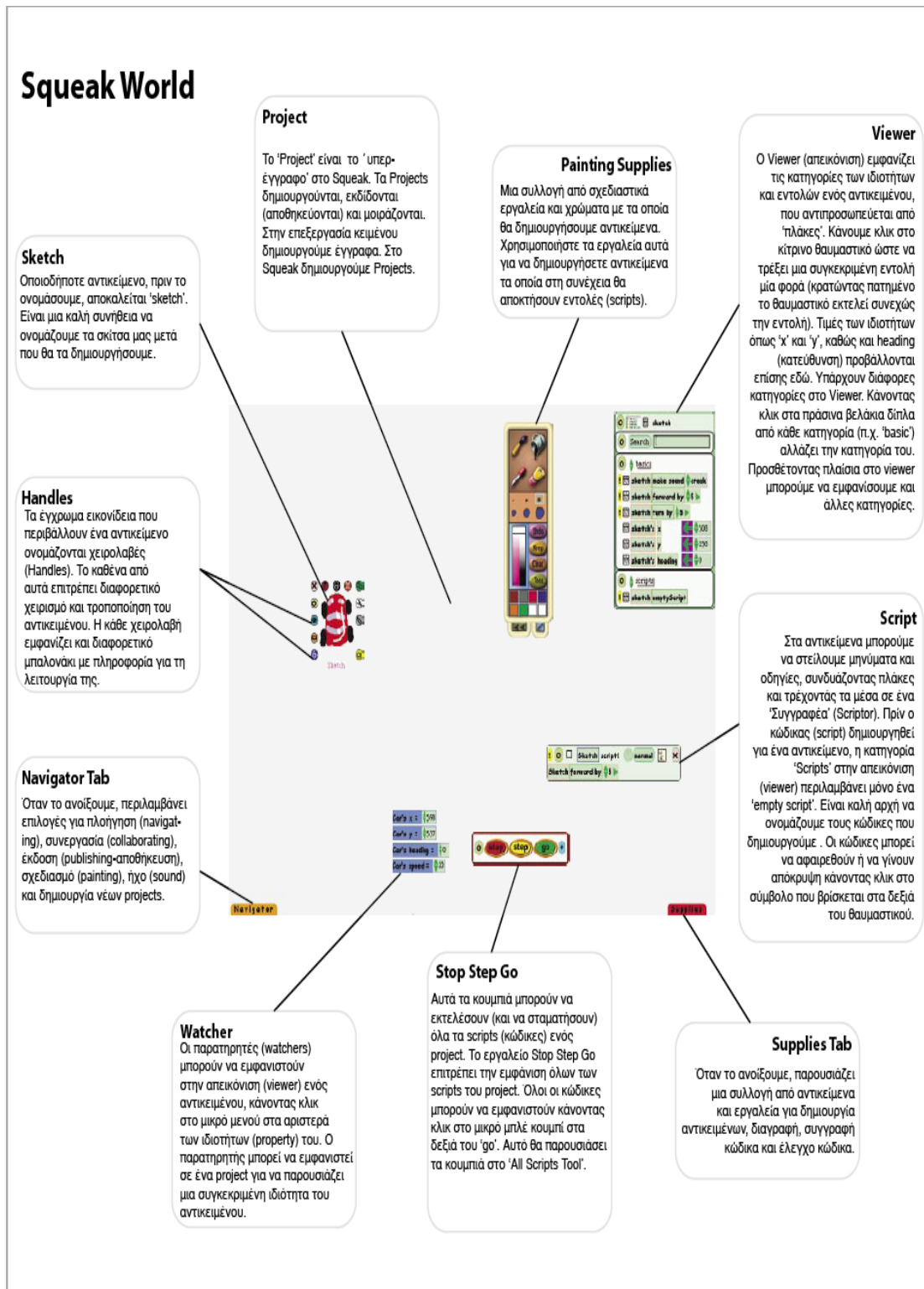
**Navigator**

NEW
Share
< PREV
NEXT >
PUBLISH IT!
Newer?
Tell!
FIND
Browser Reentry
Hide tabs


Undo
QUIT

**Εικόνα 7: Επεξήγηση συμβόλων της περιοχής Navigator**

Όταν δουλεύουμε στο squeak, θα χρησιμοποιούμε πιο συχνά τα εργαλεία που φαίνονται στην εικόνα 8.



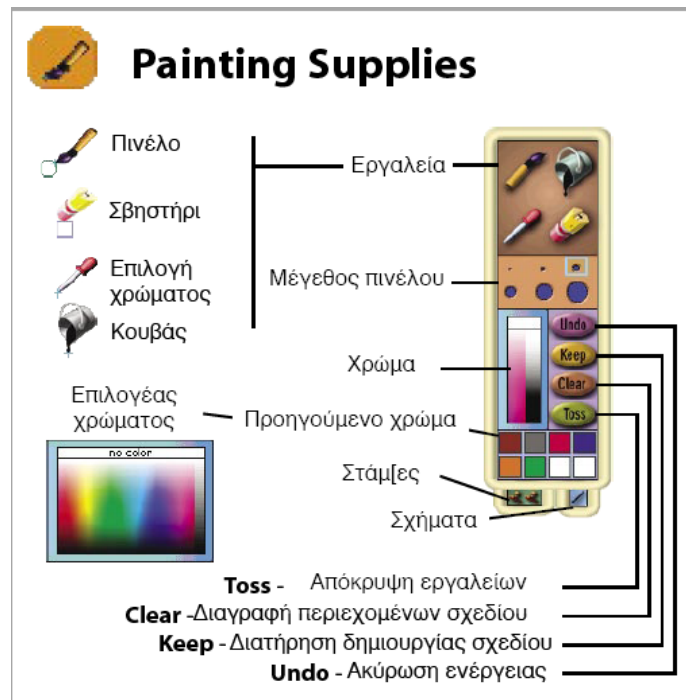
**Εικόνα 8: Ανάλυση των πιο συχνών εργαλείων του Squeak**

Πιο συγκεκριμένα, όταν θα φτιάχνουμε κάτι(αντικείμενο), όπως για παράδειγμα στην εικόνα 8 το αυτοκίνητο(Sketch), έχουμε τη δυνατότητα να παρέμβουμε και να αλλάξουμε τις ιδιότητες του χρησιμοποιώντας τα εργαλεία που το περιστοιχίζουν. Τα εργαλεία αυτά φαίνονται πιο καθαρά και επεξηγούνται στην εικόνα 9.

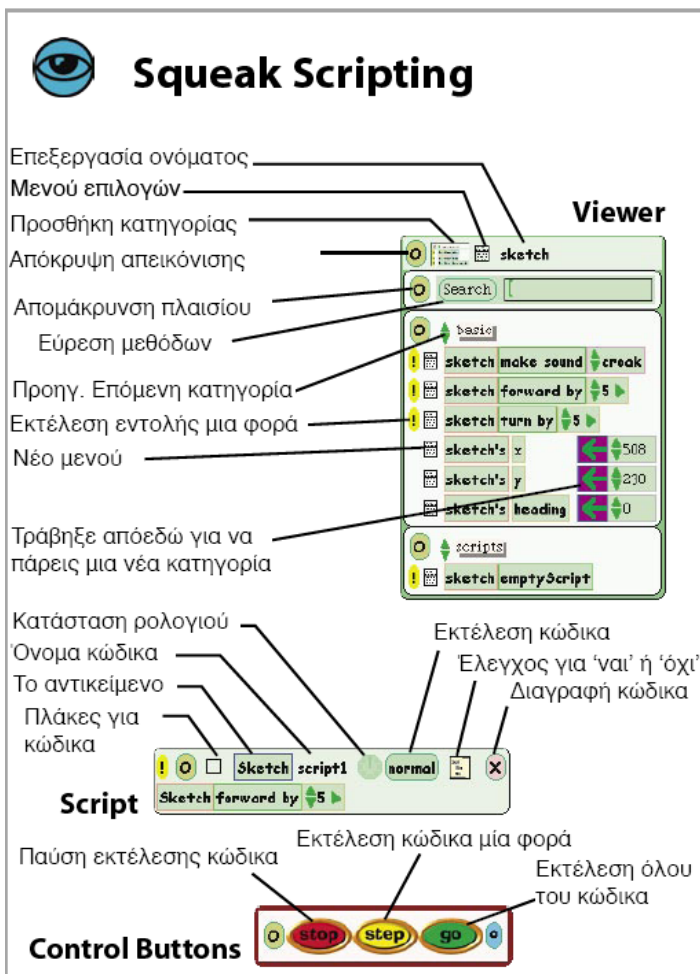


**Εικόνα 9: Επιλογές ενεργειών αντικειμένου**

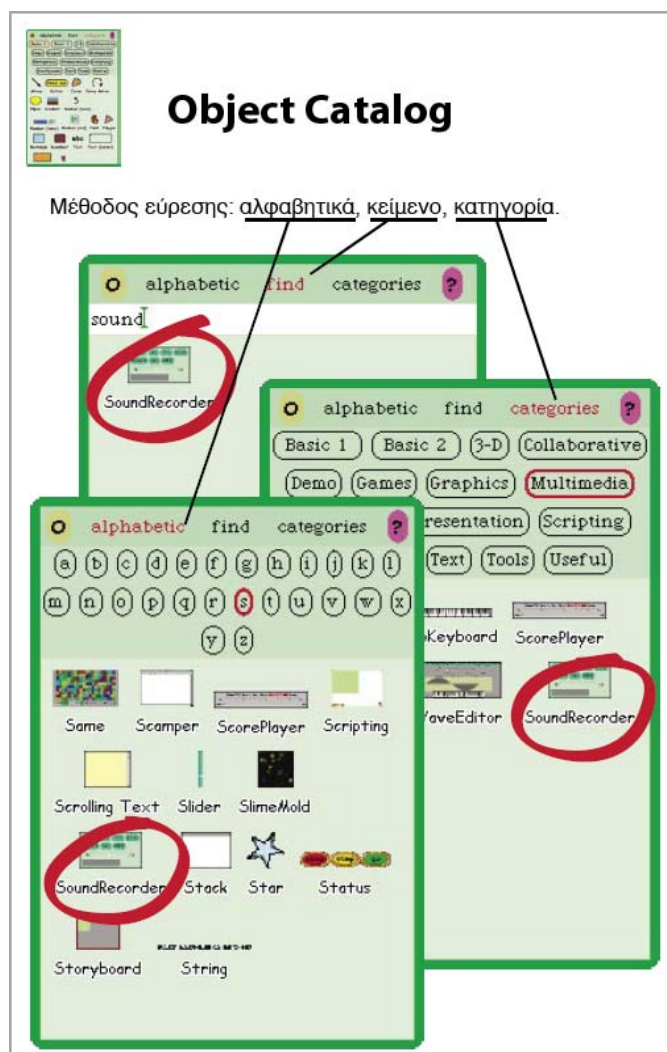
Όταν θα θελήσουμε να φτιάξουμε ένα αντικείμενο, συνήθως θα χρειαστεί να το ζωγραφίσουμε. Για τη λειτουργία αυτή, η οποία θα παρουσιαστεί παρακάτω, χρησιμοποιούμε την παλέτα ζωγραφικής, της οποίας η λειτουργία φαίνεται στην εικόνα 10. Τέλος παρουσιάζουμε τις εικόνες 11 και 12 οι οποίες θα μας χρησιμεύσουν στην απόφασή μας να βάλουμε για παράδειγμα σε κίνηση ένα αντικείμενο(Squeak Scripting) ή στην εύρεση ενός αντικειμένου(object catalog).



**Εικόνα 10: Παλέτα ζωγραφικής**



**Εικόνα 11: Εργαλεία κατασκευής κώδικα**



**Εικόνα 12: Κατάλογος αντικειμένων του Squeak**

### **3.2. Δημιουργία-Αποθήκευση Αρχείου**

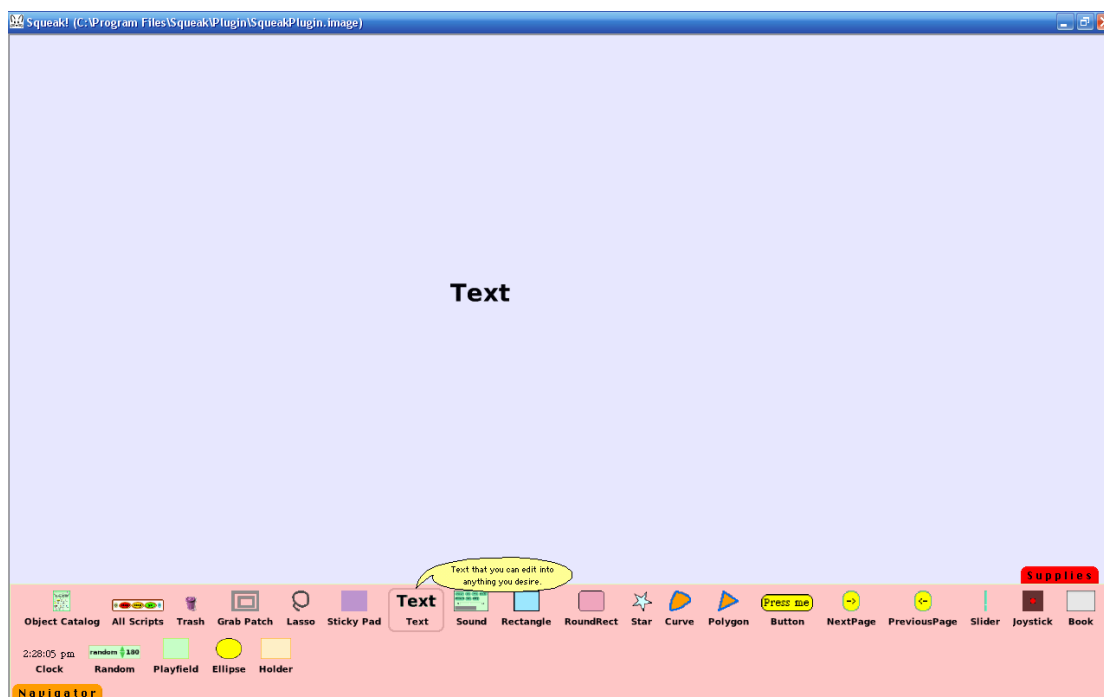
Αφού περιγράψαμε σύντομα τα βασικά εργαλεία του Squeak, στη συνέχεια θα δείξουμε τον τρόπο δημιουργίας ενός αρχείου. Θα ξεκινήσουμε το πρόγραμμα και θα μας εμφανιστεί η εικόνα 13.



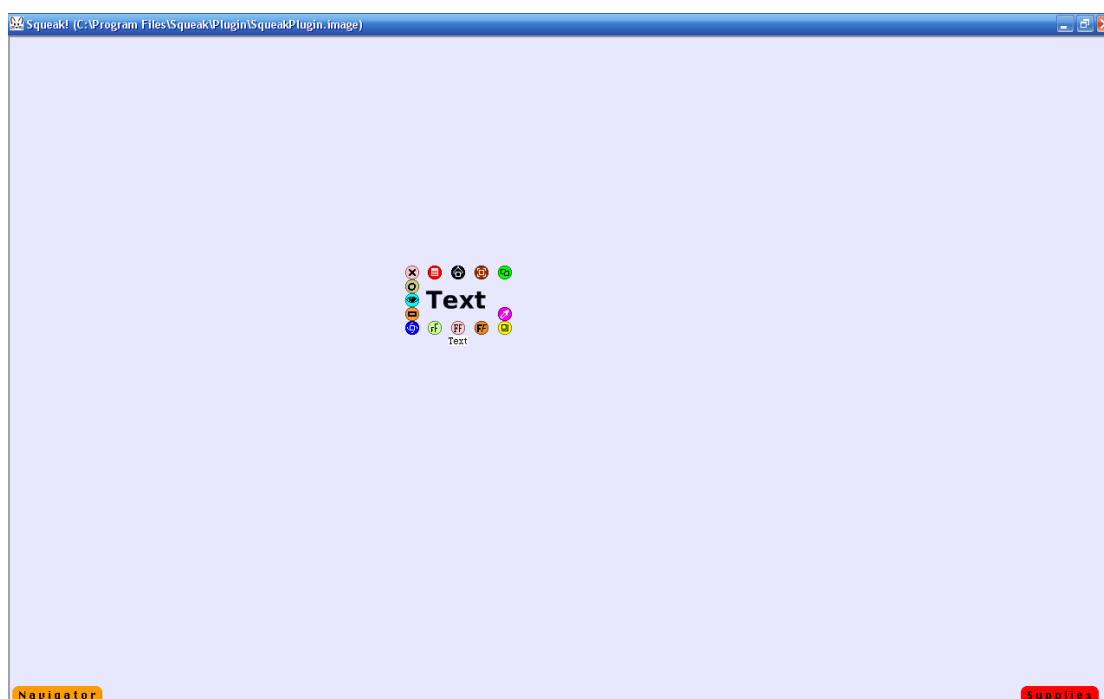
**Εικόνα 13: Κεντρική Σελίδα του Squeak**

Το συγκεκριμένο παράθυρο της εικόνας 13 το ονομάζουμε 'Squeak Desktop'. Στη συνέχεια ανοίγουμε την καρτέλα 'Supplies' και τραβάμε προς της επιφάνεια εργασίας το αντικείμενο(εργαλείο) 'Text'. Ο όρος 'τραβάμε' σημαίνει ότι πηγαίνουμε με το δείκτη του ποντικιού πάνω στο επιθυμητό αντικείμενο, πατάμε αριστερό κλικ πάνω του και κρατάμε πατημένο το πλήκτρο μέχρι να το τραβήξουμε στο επιθυμητό σημείο, στο οποίο αφήνουμε το αριστερό κλικ. Έτσι έχουμε το αποτέλεσμα της εικόνας 14. Κατόπιν πηγαίνουμε με το δείκτη του ποντικιού πάνω στο αντικείμενο που φτιάξαμε, και πατάμε το κουμπί 'Alt' μαζί με το αριστερό κλικ του ποντικού και παίρνουμε την εικόνα 15. Το αποτέλεσμα είναι να εμφανίζονται γύρω από το αντικείμενο 'Text' εργαλεία με τα οποία μπορούμε να το αλλάξουμε.

## Παραδείγματα Χρήσης για Αρχάριους

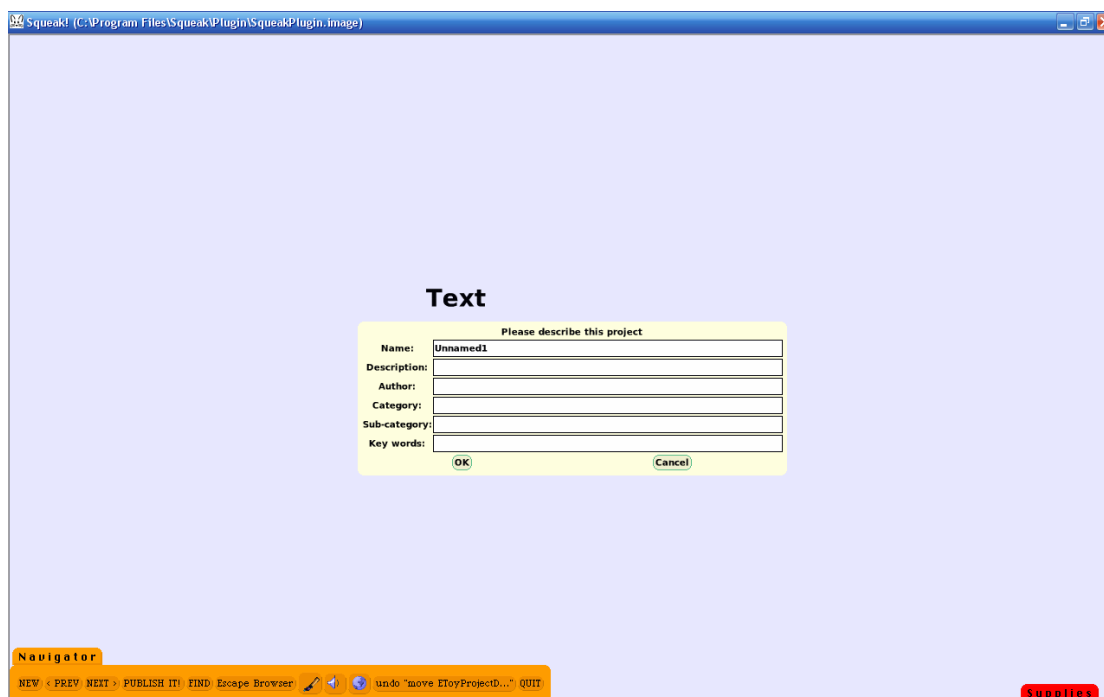


**Εικόνα 14: Δημιουργία αντικειμένου Text**



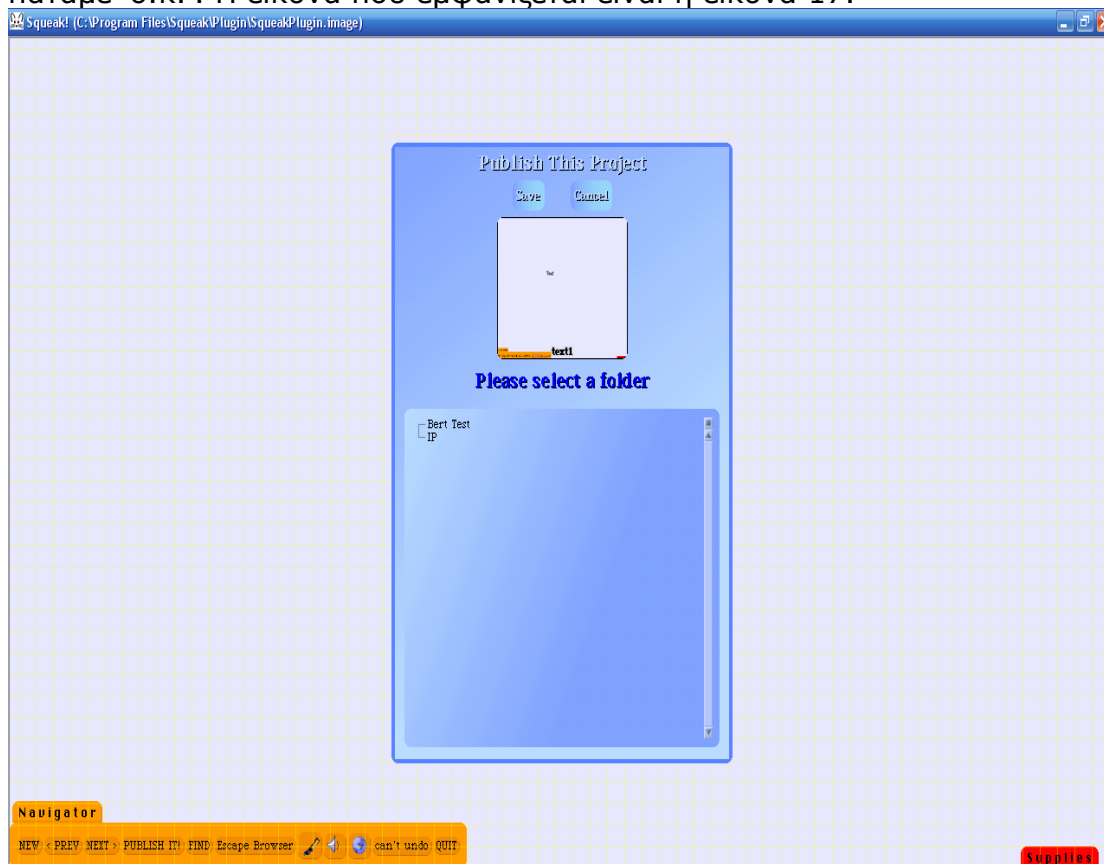
**Εικόνα 15: Εργαλεία επιλογών για το αντικείμενο Text**

Αφού φτιάξαμε κάτι στο Squeak, θα θέλαμε να το αποθηκεύσουμε. Για το λόγο αυτό, ανοίγουμε την καρτέλα Navigator και πατάμε εκεί που λέει 'PUBLISH IT' και μας εμφανίζεται η εικόνα 16.



### Εικόνα 16: Αποθήκευση αρχείου – βήμα 1

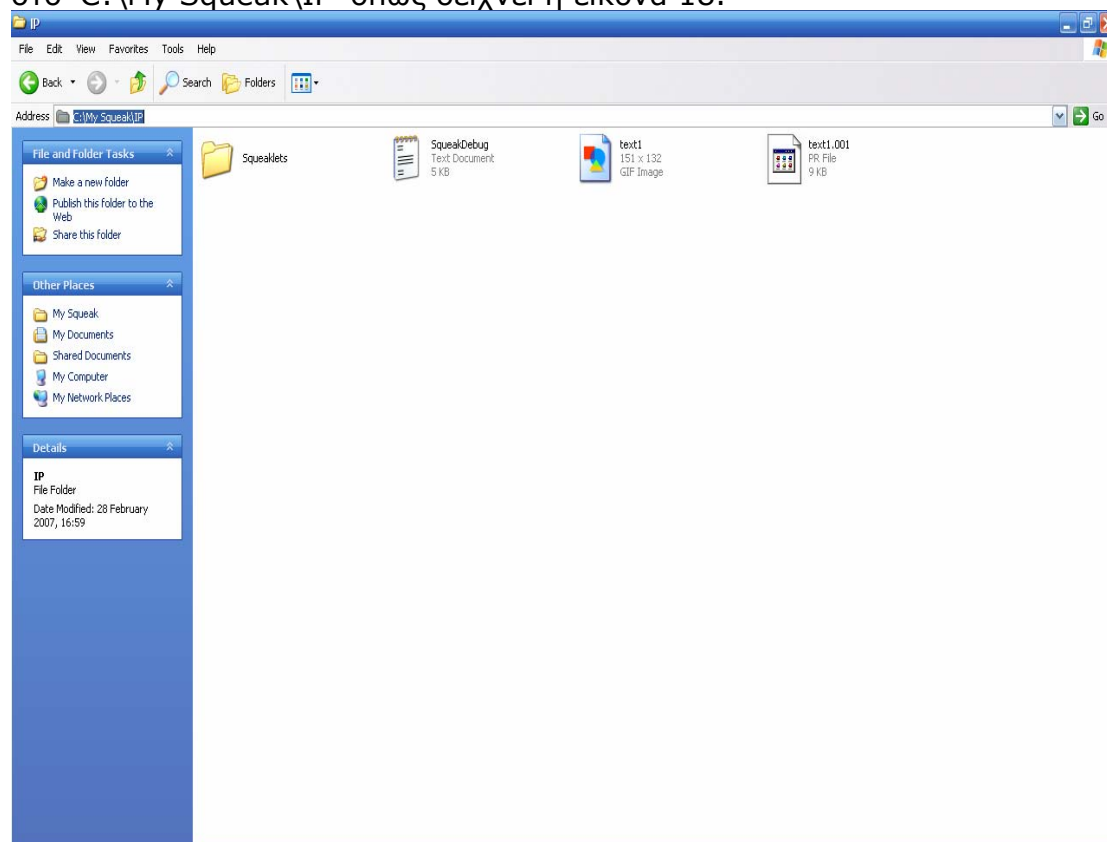
Εκεί που λέει 'Name' γράφουμε το όνομα(το οποίο θα είναι για παράδειγμα 'text1') το οποίο θέλουμε να πάρει η νέα μας κατασκευή(project) και πατάμε 'o.k.'. Η εικόνα που εμφανίζεται είναι η εικόνα 17.



### Εικόνα 17: Αποθήκευση αρχείου – βήμα 2

Στην εικόνα αυτή(εικόνα 17) το πρόγραμμα μας ρωτά να επιλέξουμε ένα φάκελο στον οποίο να αποθηκεύσουμε το αρχείο. Εμείς επιλέγουμε τον φάκελο 'IP' που αντιπροσωπεύει τον υπολογιστή μας, και η επιλογή του θα

μας οδηγήσει στην αποθήκευση του αρχείου τοπικά. Στη συνέχεια πατάμε 'Save' και το αρχείο μας αποθηκεύτηκε. Το αρχείο αυτόματα αποθηκεύεται στο 'C:\My Squeak\IP' όπως δείχνει η εικόνα 18.



**Εικόνα 18: Το αρχείο που δημιουργήσαμε αποθηκευμένο στο σκληρό δίσκο**

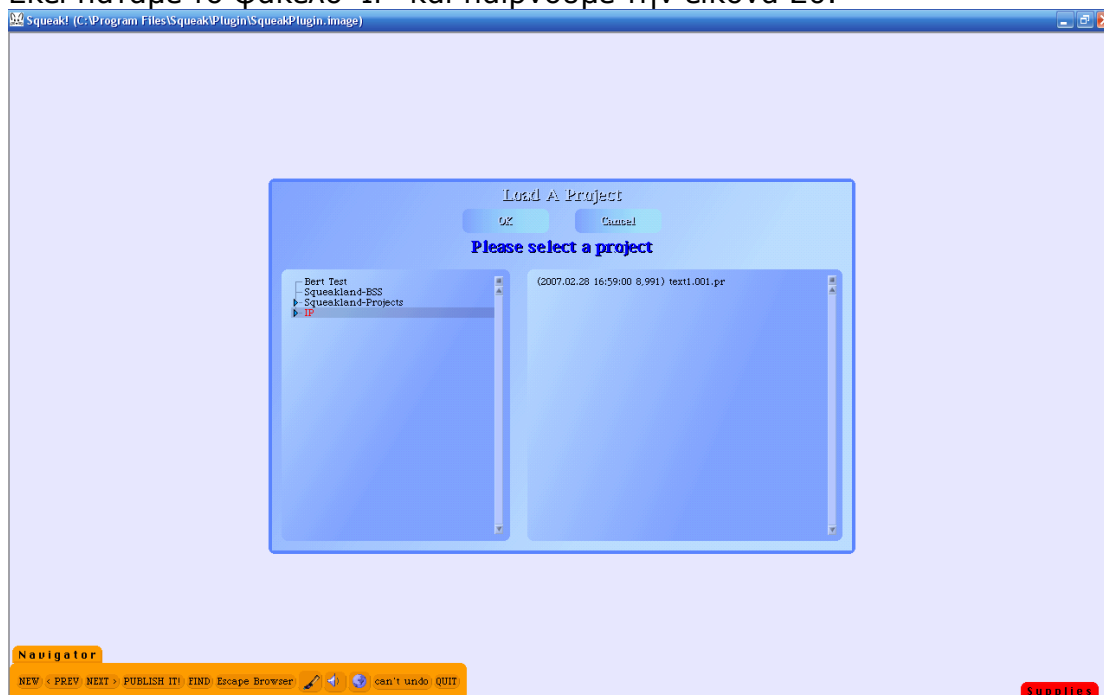
### **3.3. Άνοιγμα Αρχείου**

Θέλουμε τώρα να βρούμε και να ανοίξουμε το αρχείο που φτιάξαμε πριν. Για το λόγο αυτό ανοίγουμε το πρόγραμμα Squeak και πηγαίνουμε στην καρτέλα 'Navigator' εκεί που λέει 'FIND' και πατάμε απλό κλικ. Η εικόνα που μας εμφανίζεται είναι η εικόνα 19.



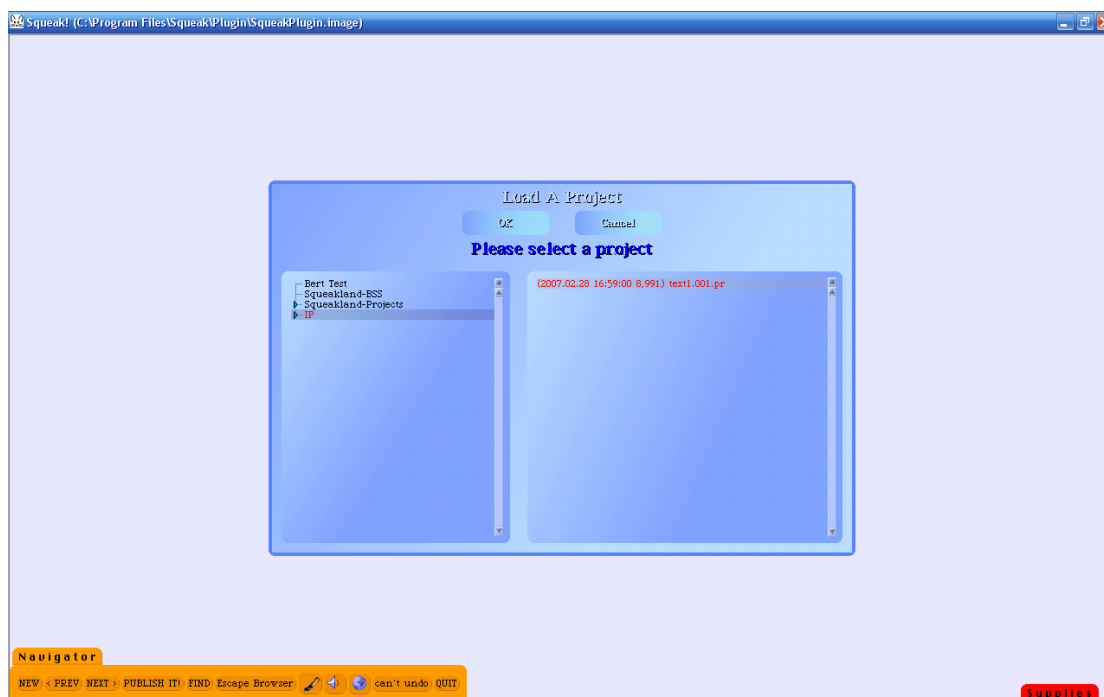
**Εικόνα 19: Άνοιγμα Αρχείου – βήμα 1**

Εκεί πατάμε το φάκελο 'IP' και παίρνουμε την εικόνα 20.

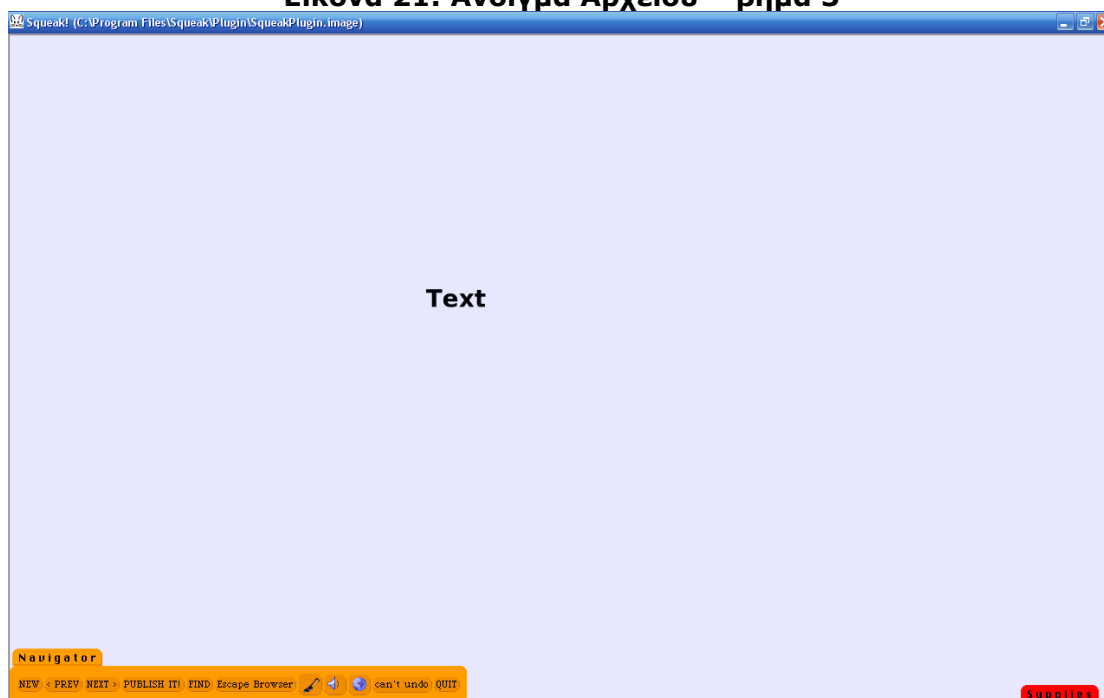


**Εικόνα 20: Άνοιγμα Αρχείου – βήμα 2**

Στη συνέχεια επιλέγουμε το αρχείο που επιθυμούμε να ανοίξουμε (εικόνα 21), στην περίπτωση μας το 'text1.001.pr' που φτιάξαμε πριν, και πατάμε 'OK'. Έτσι παίρνουμε την εικόνα 22 που είναι και η επιθυμητή.



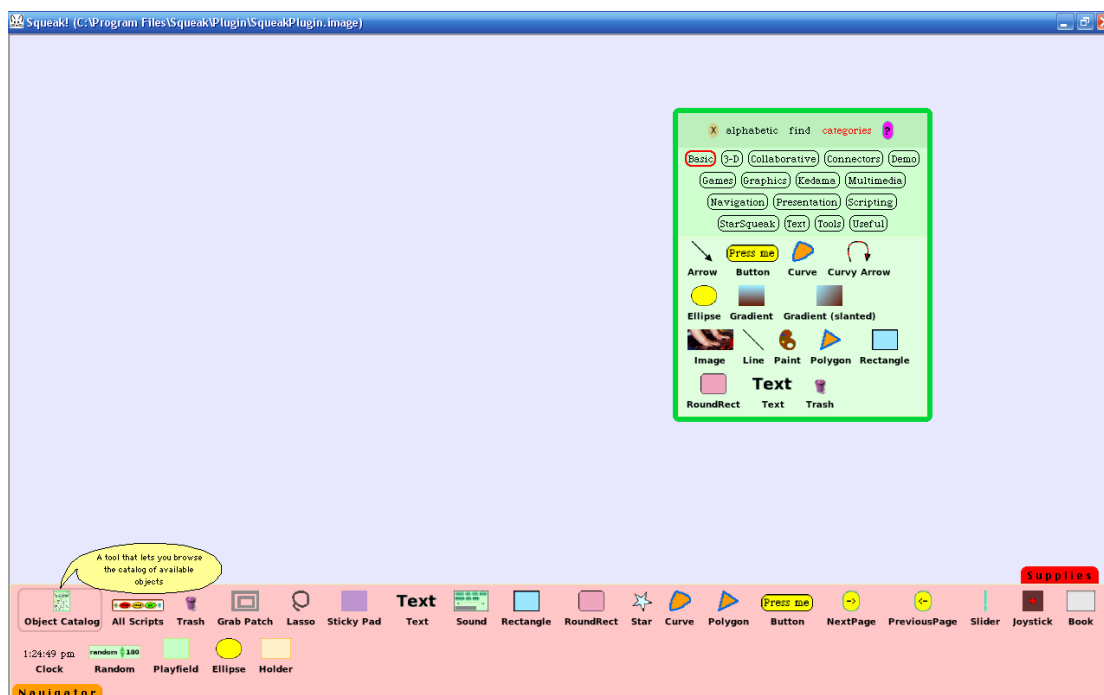
**Εικόνα 21: Άνοιγμα Αρχείου – βήμα 3**



**Εικόνα 22: Το αποθηκευμένο αρχείο ανοίγει στην επιφάνεια εργασίας**

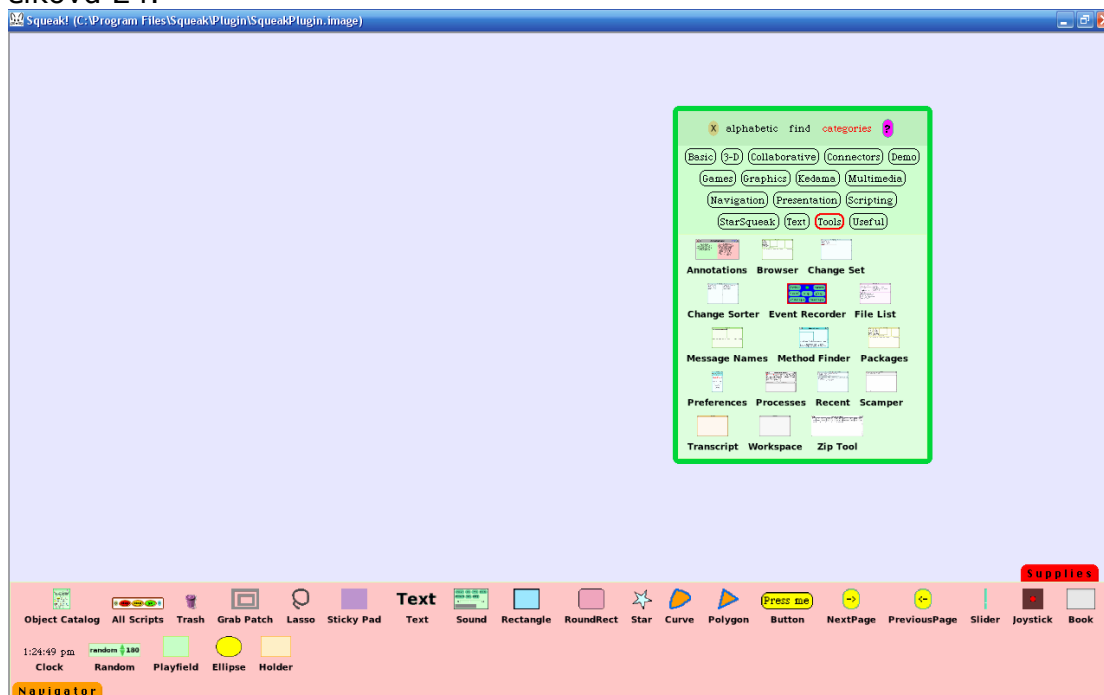
### **3.4. Workspace και Transcript**

Στη παράγραφο αυτή θα δούμε τι είναι το Workspace και το Transcript καθώς και τη διαφορά τους. Ξεκινάμε το πρόγραμμα και ανοίγουμε την καρτέλα 'Supplies'. Εκεί τραβάμε ένα 'Object Catalog' όπως φαίνεται στην εικόνα 23.



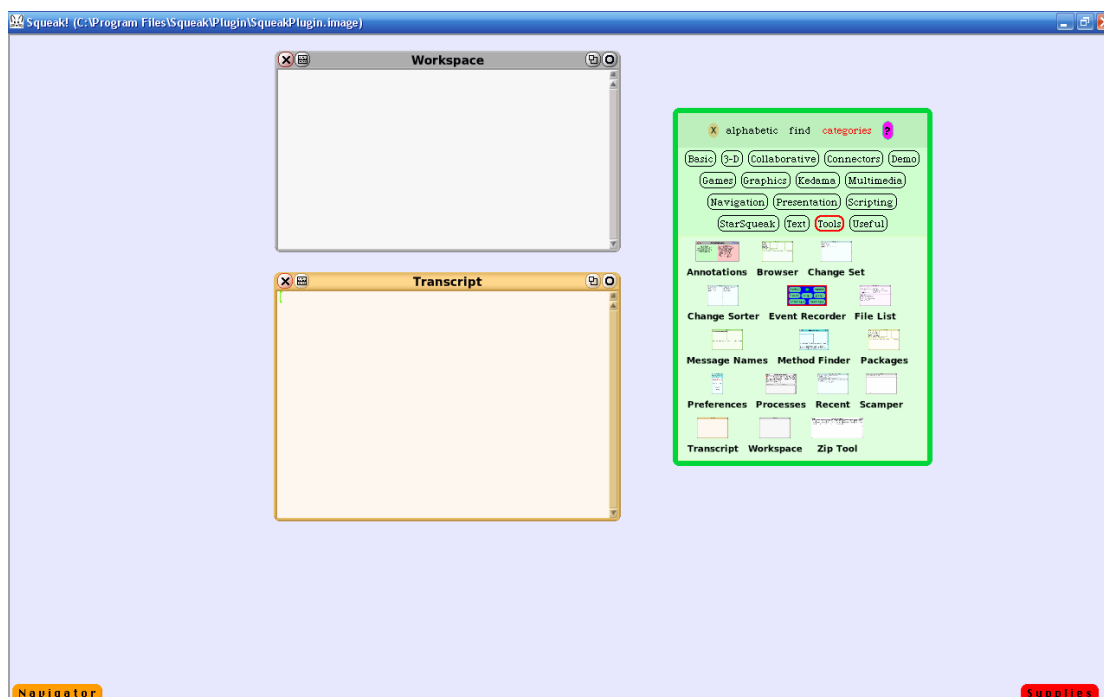
**Εικόνα 23: Η καρτέλα των αντικειμένων του squeak**

Στη πράσινη καρτέλα πατάμε το κουμπί που λέει 'Tools' και παίρνουμε την εικόνα 24.



**Εικόνα 24: Επιλέγοντας την καρτέλα εργαλείων από τα αντικείμενα**

Στη συνέχεια τραβάμε έξω ένα 'Workspace' και ένα 'Transcript'. Το αποτέλεσμα φαίνεται στην εικόνα 25.



**Εικόνα 25: Workspace και Transcript στην επιφάνεια εργασίας μας**

Το 'Workspace' μπορεί να τοποθετηθεί σε ένα project και θα λειτουργήσει σε 'χρόνο plugin' αλλά το 'Transcript' δουλεύει μόνο στην 'κατασκευή του χρόνου του project'. Για να γίνει κατανοητή η παραπάνω πρόταση, παραθέτουμε το παρακάτω παράδειγμα.

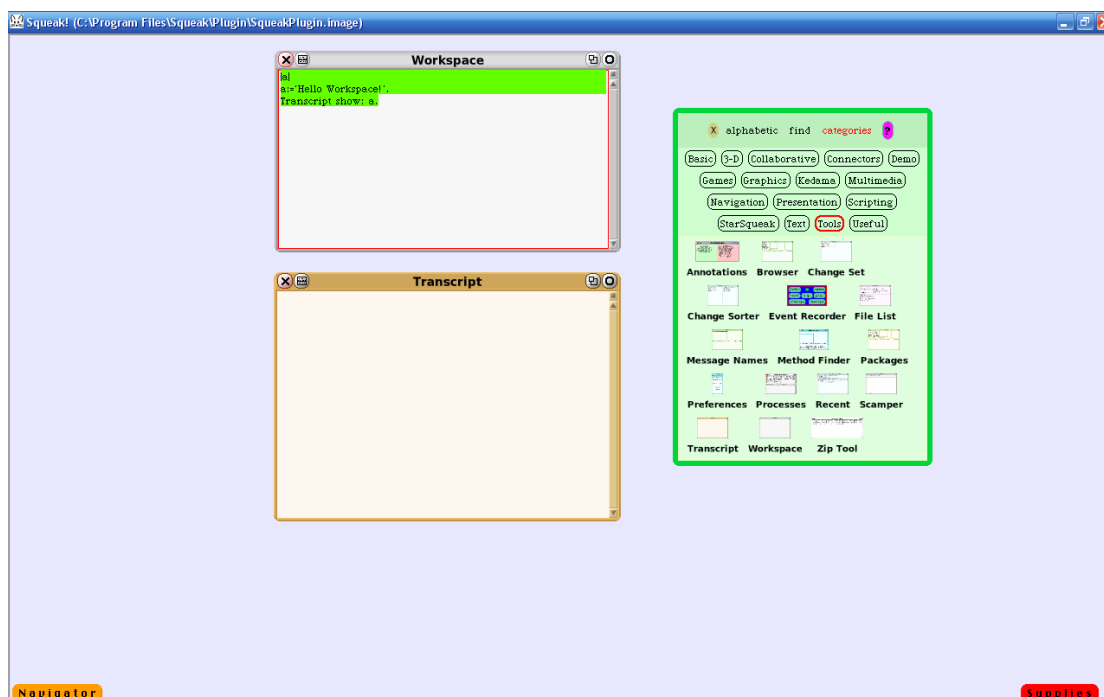
Γράφουμε στο 'Workspace' τα εξής:

```
|a|
```

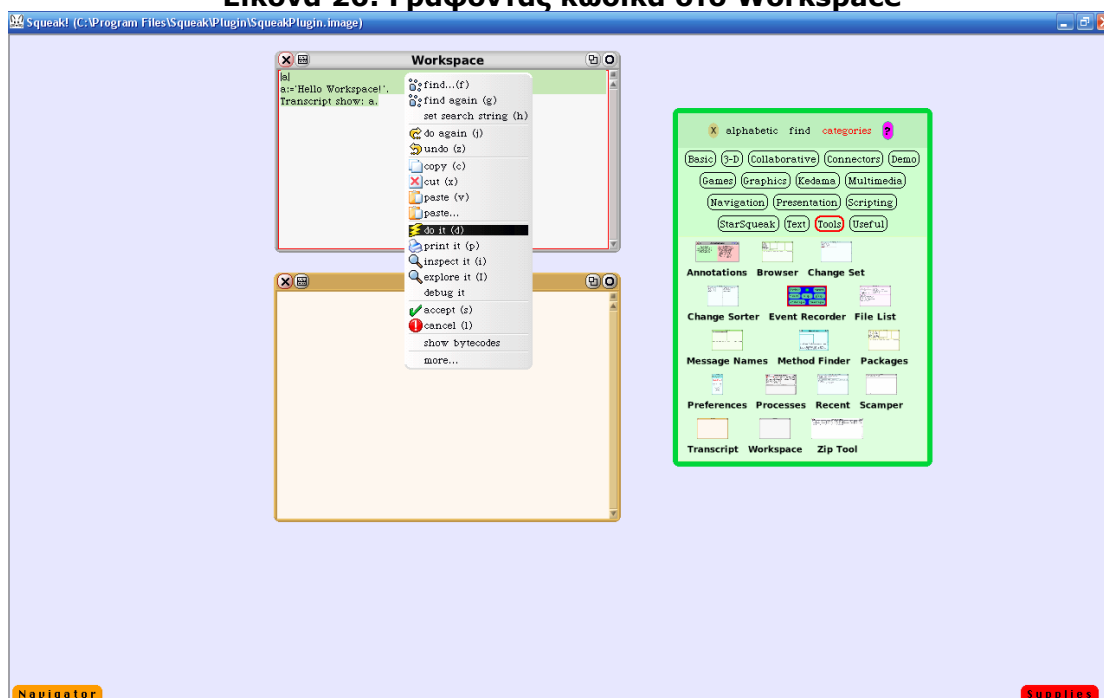
```
a:= 'Hello Workspace!'.
```

```
Transcript show: a.
```

Στη συνέχεια επιλέγουμε όλο το κείμενο με τον δείκτη του ποντικιού(εικόνα 26) και πατάμε στη συνέχεια δεξί κλικ. Προκύπτει η εικόνα 27 όπου επιλέγουμε το 'do it'.

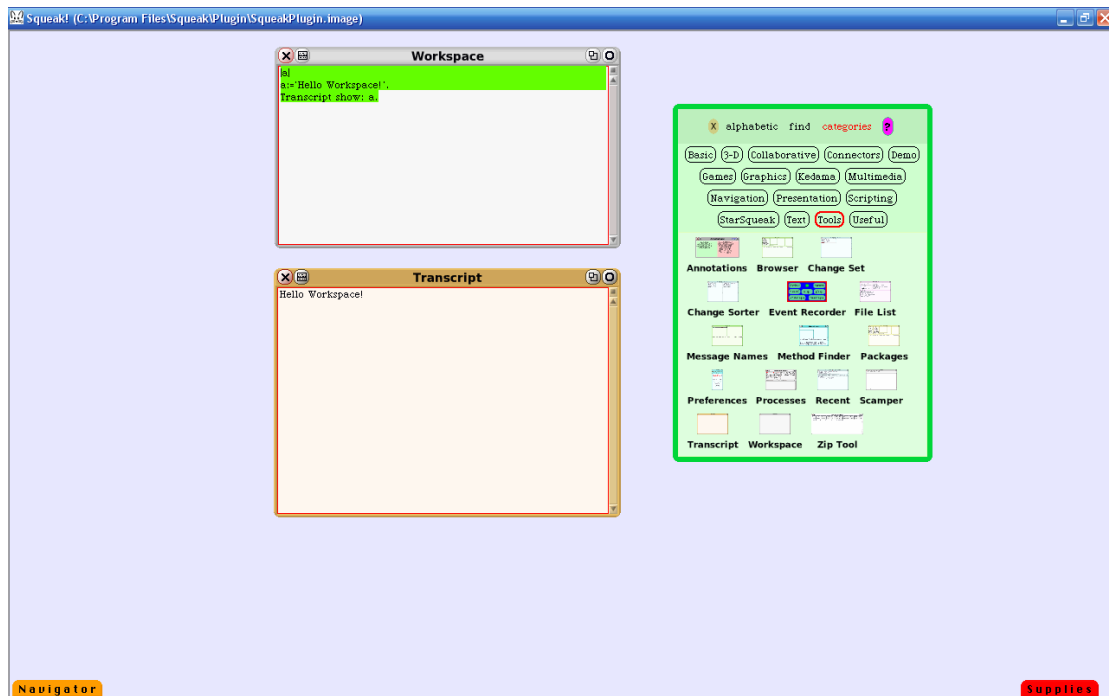


**Εικόνα 26: Γράφοντας κώδικα στο Workspace**



**Εικόνα 27: Εμφανίζοντας τα αποτελέσματα στο transcript**

Το αποτέλεσμα είναι η εικόνα 28. Οπότε καταλαβαίνουμε ότι άμα θέλουμε να εμφανίσουμε κάτι στο 'Transcript' γράφουμε στο 'Workspace' το εξής: 'Transcript show: something'



**Εικόνα 28: Το αποτέλεσμα της εκτέλεσης κώδικα από το workspace στο transcript**

### 3.5. Δημιουργώντας Σημειώσεις

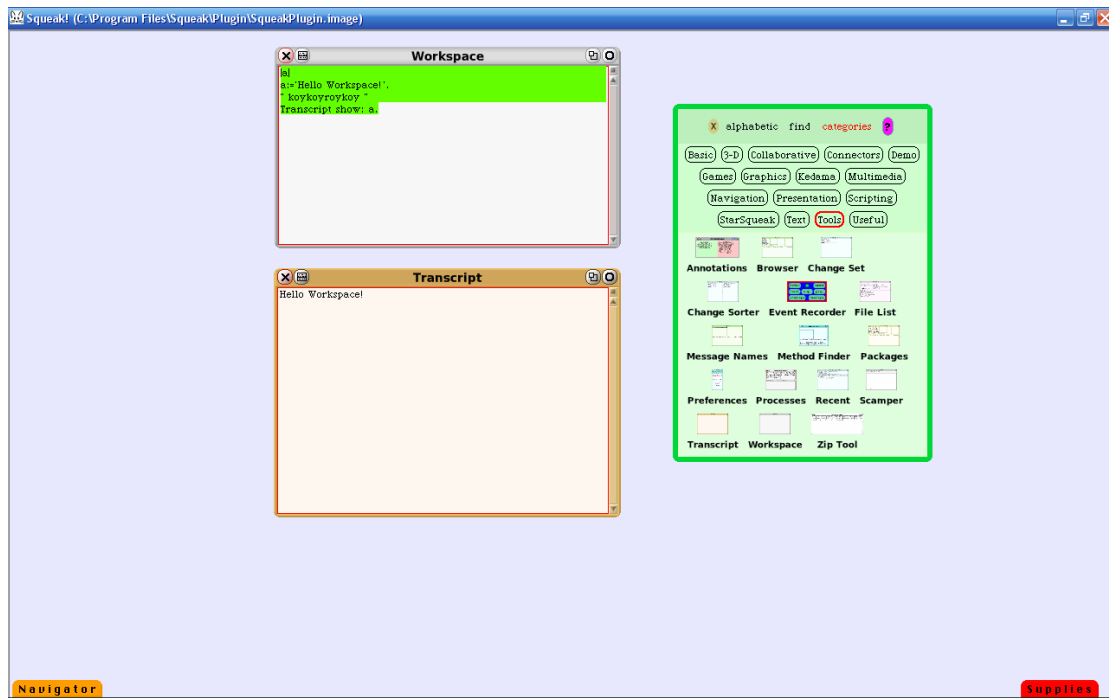
Στο προηγούμενο παράδειγμα, θέλουμε να προσθέσουμε σχόλια μέσα στο κώδικα που γράψαμε. Το μόνο που χρειάζεται είναι να συμπεριλάβουμε μέσα σε διπλά εισαγωγικά οτιδήποτε θέλουμε το πρόγραμμα να θεωρήσει ως σχόλια του συγγραφέα και όχι ως τμήμα του κώδικα που πρέπει να τρέξει.

Έτσι ο νέος μας κώδικας είναι ο εξής:

```
|a|  
a:= 'Hello Workspace!'.  
"this is the note of the code..."  
Transcript show: a.
```

Όπως βλέπουμε και στην εικόνα 29 το αποτέλεσμα είναι το ίδιο. Ο λόγος που θα βάλουμε σχόλια μέσα σε ένα τμήμα του κώδικα είναι γιατί άμα το τμήμα του κώδικα αυξηθεί σε γραμμές είναι δύσκολο να θυμάσαι για πιο λόγο φτιάχτηκε τι. Οπότε άμα θελήσεις αργότερα να δεις τον κώδικα που έχεις φτιάξει, με τα σχόλια μπορείς να τον καταλάβεις.

## Παραδείγματα Χρήσης για Αρχάριους



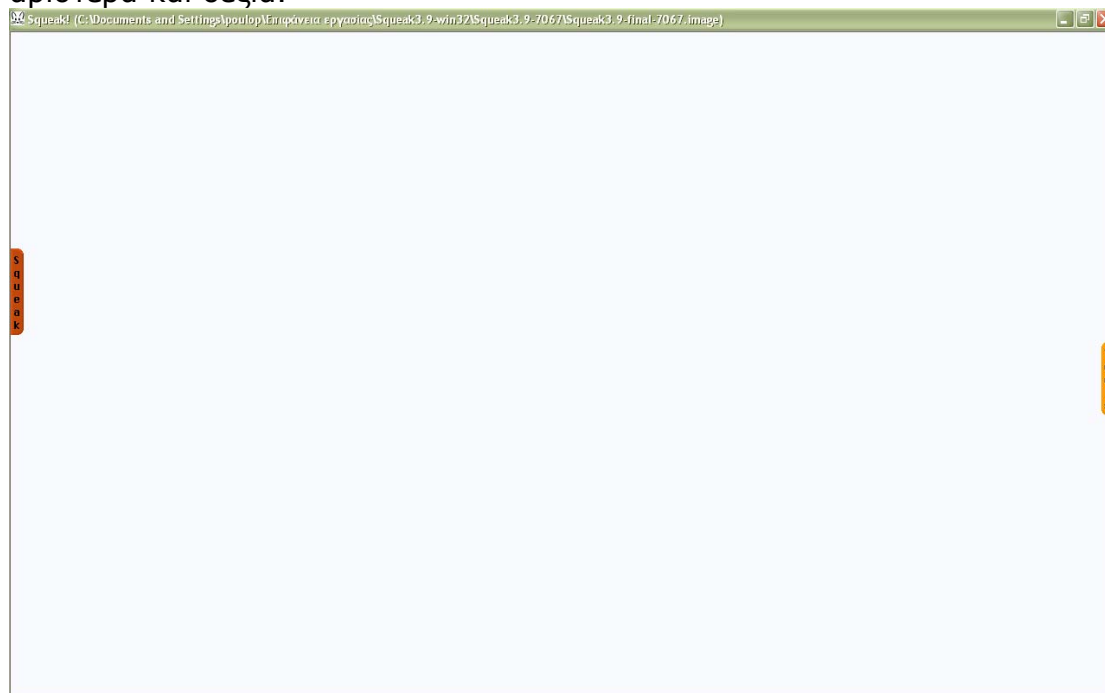
**Εικόνα 29: Το αποτέλεσμα της εκτέλεσης κώδικα από το workspace στο transcript (κώδικας με σχόλια)**

## 4. Παραδείγματα Χρήσης για Αρχάριους

Σε αυτή την ενότητα υπάρχουν παραδείγματα χρήσης για αρχάριους χρήστες

### 4.1. Το πρώτο μου πρόγραμμα

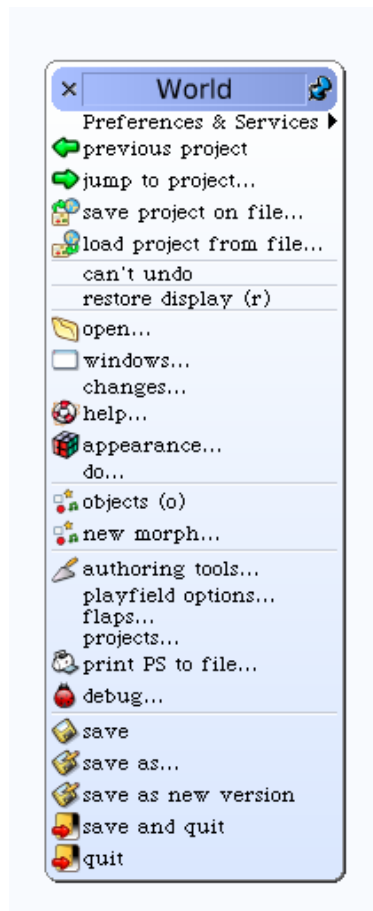
Αυτό είναι το πρώτο πρόγραμμα που φτιάχνει κάποιος μόλις ξεκινά να δουλεύει σε μία νέα γλώσσα προγραμματισμού (Hello World). Ανοίγοντας το squeak ερχόμαστε αντιμέτωποι με μία λευκή οθόνη και δύο επιλογές αριστερά και δεξιά.



**Εικόνα 30: Το κεντρικό γραφικό περιβάλλον του Squeak**

Αριστερά έχουμε το πλήκτρο που μας δίνει γενικές πληροφορίες για το squeak ενώ δεξιά, αν πατήσουμε το πλήκτρο tools μπορούμε να δούμε διάφορα εργαλεία για την ανάπτυξη προγραμμάτων και κώδικα.

Δύο επιπλέον βασικά στοιχεία είναι οι επιλογές που εμφανίζονται αν στην κεντρική οθόνη πατήσουμε αριστερό κλικ ή και δεξί κλικ. Συγκριτικά με το περιβάλλον Squeak Etoys η διαφορά εντοπίζεται πως σε αυτό το περιβάλλον δουλεύουμε κυρίως με δεξί και αριστερό κλικ στην επιφάνεια εργασίας για να δούμε τα διάφορα εργαλεία του Squeak ενώ τα εργαλεία (tools) που πριν είδαμε στην πλατφόρμα εργαλείων τα βρίσκουμε στη δεξιά πλευρά της οθόνης. Είτε χρησιμοποιήσουμε αυτό το περιβάλλον είτε το Squeak Etoys το αποτέλεσμα που λαμβάνουμε είναι το ίδιο. Το περιβάλλον etoys είναι ωστόσο αρκετά πιο εύκολο στη χρήση και γι' αυτό προτείνεται. Να σημειωθεί πως ότι βρίσκουμε στο περιβάλλον Squeak με δεξί και αριστερό κλικ το βρίσκουμε στο Squeak Etoys στα Navigator και Supplies.

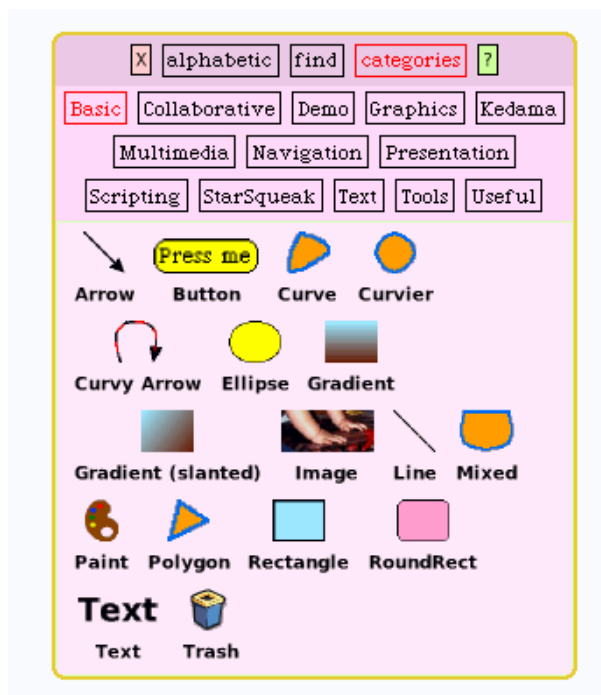


**Εικόνα 31: Επιλογές που εμφανίζονται με αριστερό κλικ**



**Εικόνα 32: Επιλογές που εμφανίζονται με δεξί κλικ**

Για να φτιάξουμε το πρώτο απλοϊκό μας πρόγραμμα κάνουμε αριστερό κλικ στο κέντρο της επιφάνειας εργασίας του squeak και επιλέγουμε την επιλογή object. Στην οθόνη εμφανίζεται το μενού που φαίνεται στην παρακάτω εικόνα.



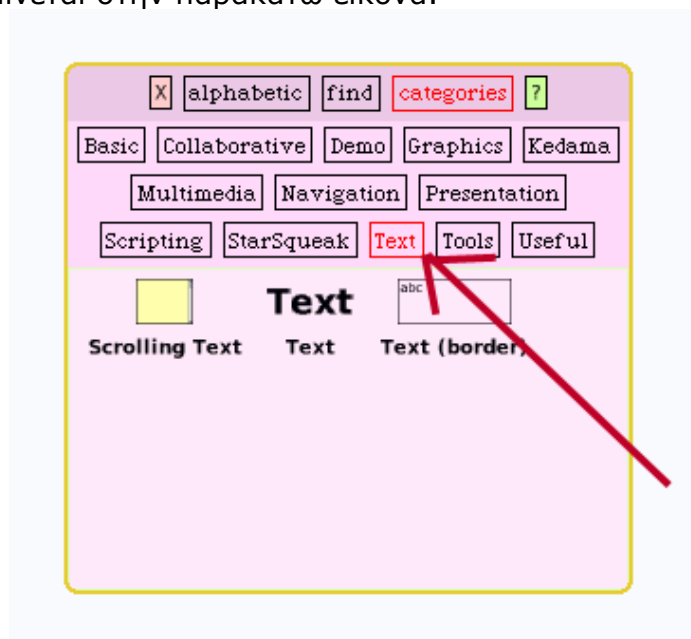
**Εικόνα 33: Τα αντικείμενα (objects) του Squeak**

**Σημείωση**

Μπορούμε να ανοίξουμε τα objects χρησιμοποιώντας τα πλήκτρα Alt+o ( κρατώντας πατημένο το Alt πατούμε το όμικρον)

**Etoys:** Μπορούμε να ανοίξουμε τα objects από το supplies -> object catalog

Αφού ανοίξουμε τα αντικείμενα επιλέγουμε από το πάνω μέρος την επιλογή Text όπως φαίνεται στην παρακάτω εικόνα.

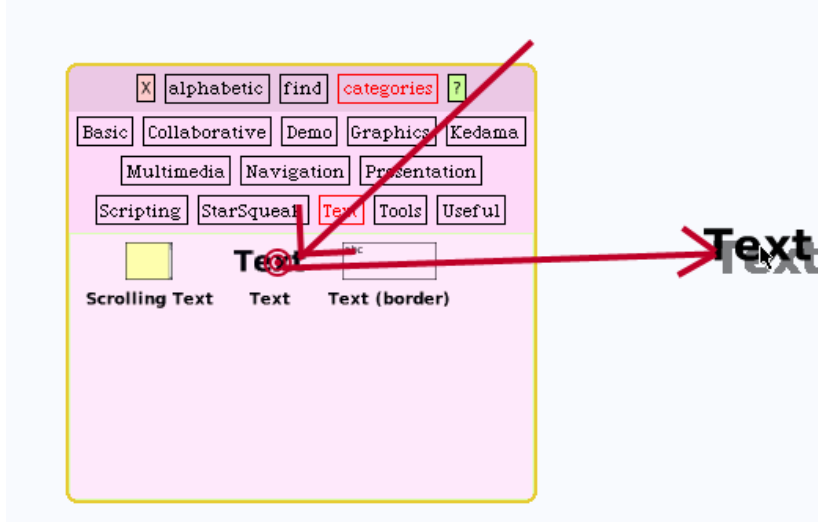


**Εικόνα 34: Επιλέγοντας τα αντικείμενα τύπου Text (κείμενο)**

Βλέπουμε πως εμφανίζονται τρία αντικείμενα τύπου κειμένου (text). Το πρώτο αντικείμενο δημιουργεί κινούμενο κείμενο (scrolling text), το

δεύτερο δημιουργεί απλό κείμενο, ενώ το τρίτο δημιουργεί κείμενο με περίγραμμα (Text (border)).

Επιλέγουμε το αντικείμενο «κείμενο» (Text) και αφού κρατήσουμε πατημένο το αριστερό πλήκτρο του ποντικιού σύρουμε το αντικείμενο στο λευκό χώρο της επιφάνειας εργασίας του Squeak όπως φαίνεται παρακάτω.



**Εικόνα 35: Επιλέγουμε το αντικείμενο Text και το σύρουμε στην επιφάνεια εργασίας**

Αφήνοντας το πλήκτρο του ποντικιού θα δούμε το αντικείμενο text να τοποθετείται στην επιφάνεια εργασίας. Εν συνεχεία πατούμε με το πλήκτρο του ποντικιού ακριβώς στα αριστερά του "t" της λέξης Text και βλέπουμε την παρακάτω εικόνα.



**Εικόνα 36: Εμφανίζοντας τον κέρσορα γραψίματος**

Αν πατήσουμε το πλήκτρο backspace θα δούμε ότι η λέξη TEXT σβήνεται και στη θέση της μπορούμε να γράψουμε οποιαδήποτε λέξη (στα αγγλικά). Εν προκειμένω γράφουμε το κείμενο Hello World! (Γεια σου κόσμε!) και προκύπτει το παρακάτω αποτέλεσμα.



**Εικόνα 37: Γεια σου κόσμε!**

Προκειμένου να βάλουμε τις λέξεις σε μία σειρά και όχι «κομμένες» όπως φαίνεται στην εικόνα κρατούμε το πλήκτρο Alt πατημένο και πατούμε αριστερό κλικ στο χώρο που είναι γραμμένο το "Hello World!". Παρατηρούμε πως εμφανίζονται τα παρακάτω στοιχεία τριγύρω από το κείμενο "Hello World!".

### Σημείωση

Από εδώ και στο εξής θα αναφερόμαστε σε αυτά τα τριγύρω αντικείμενα ως επιλογές οποιουδήποτε αντικειμένου φτιάχνουμε. Θα γράφουμε τη φράση: «Ανοίγουμε τις επιλογές του αντικειμένου»



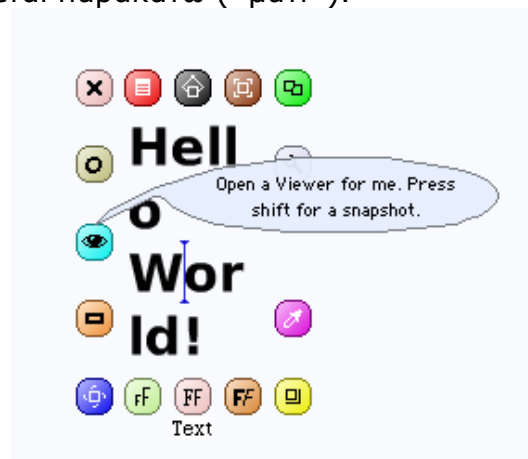
**Εικόνα 38: Στοιχεία που αφορούν το αντικείμενο κειμένου "hello world!"**

Κρατώντας πατημένο το αριστερό πλήκτρο του ποντικιού πατούμε το κάτω αριστερό εικονίδιο από αυτά που εμφανίστηκαν και σύρουμε το ποντίκι προς τα δεξιά όπως φαίνεται παρακάτω.



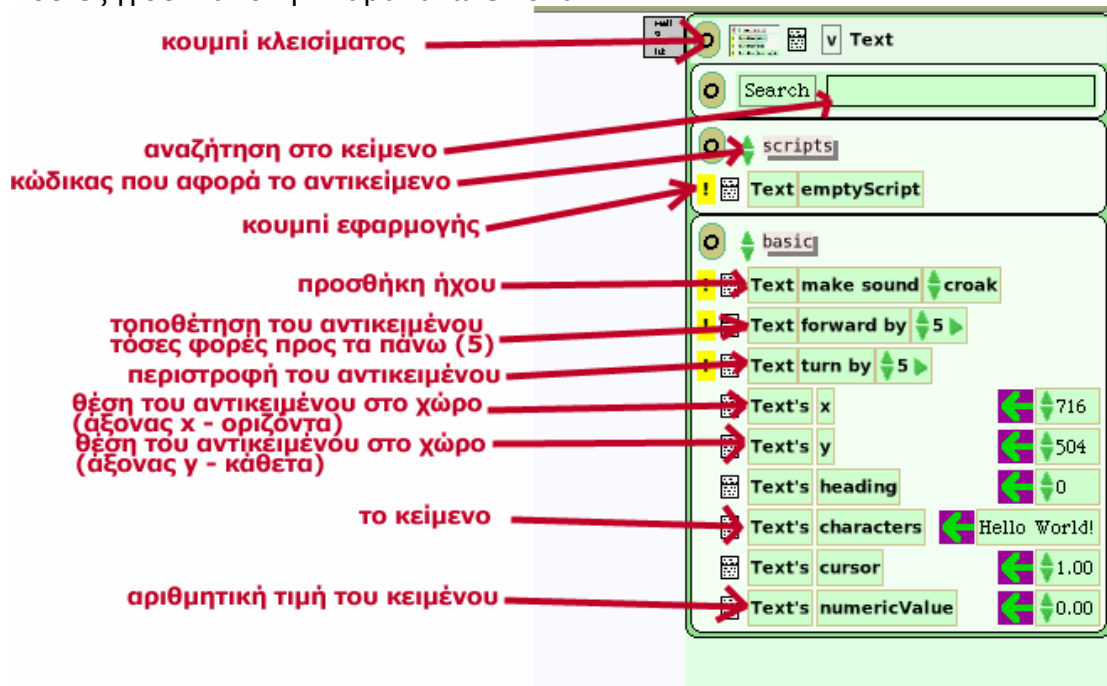
**Εικόνα 39: Σύροντας το κουμπι size από τα attributes ενός αντικειμένου**

Μπορούμε να εμφανίσουμε ένα παράθυρο με όλες τις πληροφορίες για το αντικείμενο (attributes) εμφανίζοντας τα attributes και επιλέγοντας το εικονίδιο που φαίνεται παρακάτω («μάτι»).



**Εικόνα 40: το attribute «μάτι»**

Πατώντας το «μάτι» βλέπουμε μία σειρά στοιχείων για το αντικείμενο μας που εξηγούνται στην παρακάτω εικόνα.



**Εικόνα 41: Επεξήγηση attributes ενός αντικειμένου**

#### Σημείωση

Στο εξής θα αναφερόμαστε στην παραπάνω ενέργεια ως: ανοίγουμε τα attributes ενός αντικειμένου. Η διαδικασία θα είναι να ανοίξουμε τις επιλογές του αντικειμένου και να πατήσουμε το «μάτι».

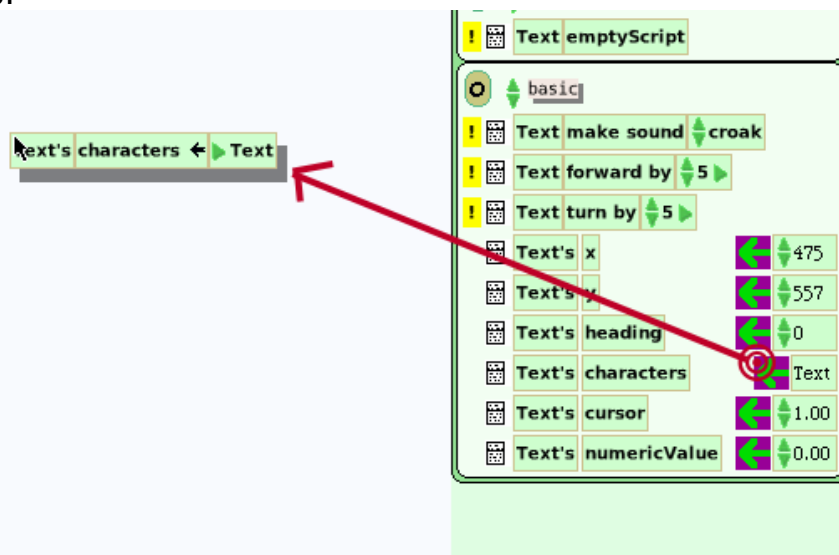
Μπορούμε να πειραματιστούμε με τα παραπάνω attributes. Για παράδειγμα μπορούμε να αλλάξουμε το ίδιο το κείμενο. Πατούμε μέσα στο Hello World (στα attributes), γράφουμε ένα καινούριο κείμενο (π.χ. "Hi") και πατούμε Enter. Βλέπουμε αμέσως πως το κείμενο αλλάζει από Hello World! σε "Hi". Μπορούμε να αλλάξουμε τη θέση του αντικειμένου στο χώρο. Για παράδειγμα στο πεδίο Text's X στη θέση του 716 (η τιμή που θα φαίνεται σε εσάς θα είναι διαφορετική) γράφουμε 500 και πατούμε Enter. Το ίδιο κάνουμε και με το Text's y και αλλάζει η κάθετη τοποθέτηση του αντικειμένου στο χώρο. Σε όσα πεδία αριστερά υπάρχει το θαυμαστικό σε κίτρινο κυκλάκι (κουμπι εφαρμογής) θα πρέπει αφού επιλέξουμε την τιμή που θέλουμε να πατήσουμε το κυκλάκι για να δούμε ότι εφαρμόζεται. Για παράδειγμα στην επιλογή Text turn by, επιλέγουμε από τα βελιάκια πάνω-κάτω που βρίσκονται αριστερά από την αναγραφόμενη αριθμητική τιμή να αλλάξουμε αυτή την τιμή θέτοντας το πόσο θέλουμε να περιστραφεί το αντικείμενο. Πατώντας δύο φορές το βελιάκι πάνω το 5 γίνεται 7. Ενώ αλλάζει από 5 σε επτά η τιμή δε βλέπουμε κάποια αλλαγή στο κείμενό μας, ωστόσο αν πατήσουμε το κουμπι εφαρμογής που βρίσκεται στα αριστερά του Text turn by θα δούμε το κείμενο να περιστρέφεται δεξιόστροφα. Για να «κλείσουμε» το παράθυρο των attributes πατούμε το πάνω κουμπι κλεισίματος.

Συνεχίστε τον πειραματισμό με τα attributes ενός αντικειμένου. Σε περίπτωση που νιώθετε ότι κάνατε κάτι λάθος και έχετε χαλάσει τα αντικείμενα που είχατε στην επιφάνεια εργασίας του squeak, μπορείτε να

σβήσετε οποιοδήποτε αντικείμενο ανοίγοντας τις επιλογές του και πατώντας το X που βρίσκεται πάνω αριστερά στις επιλογές.

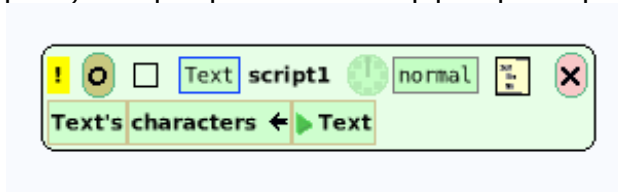
#### 4.1.1. Κάνοντας το ίδιο με κώδικα!

Ανοίγουμε το κενό περιβάλλον του squeak και εμφανίζουμε τα αντικείμενα (objects). Επιλέγουμε το κείμενο και τοποθετούμε ένα Text στην επιφάνεια εργασίας. Εν συνεχεία ανοίγουμε τα attributes του αντικειμένου και επιλέγουμε από το Text characters το βελάκι που δείχνει προς τα αριστερά και το σύρουμε προς την επιφάνεια εργασίας του squeak όπως φαίνεται παρακάτω.



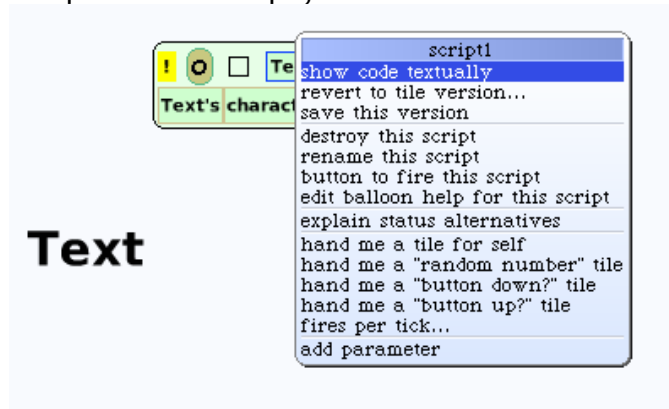
**Εικόνα 42: Εμφανίζοντας το attribute Text's character στην επιφάνεια εργασίας του squeak**

Βλέπουμε να εμφανίζεται η παρακάτω επιλογή στην επιφάνεια εργασίας.



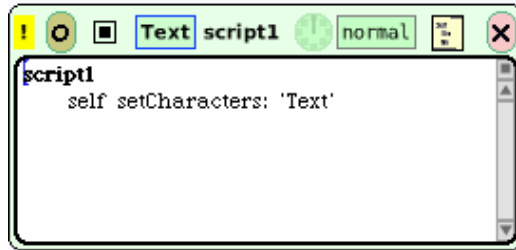
**Εικόνα 43: Το text's character στην επιφάνεια εργασίας**

Εν συνεχεία κάνουμε αριστερό κλικ στο κουμπι Text και βλέπουμε να εμφανίζονται οι παρακάτω επιλογές.



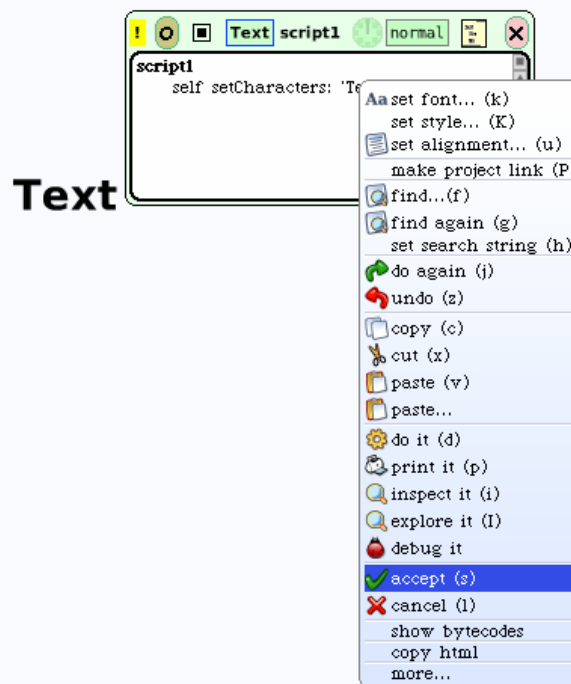
**Εικόνα 44: τα στοιχεία του κουμπιού Text**

Επιλέγουμε την πρώτη επιλογή «show code textually» και βλέπουμε την παρακάτω εικόνα.



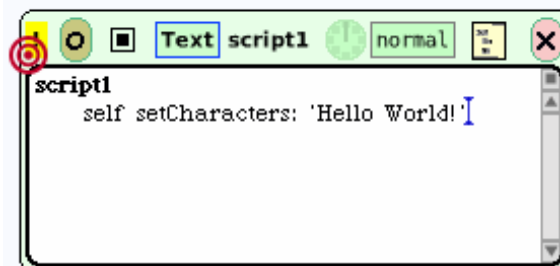
**Εικόνα 45: Επιλογή show code textually**

Σε αυτό που εμφανίζεται αντικαθιστούμε τη λέξη Text, με τη λέξη Hello World!, με προσοχή ώστε να μη σβήσουμε τα «αυτάκια» ('). Για να εφαρμοστεί αυτό που αλλάξαμε πατούμε δεξί κλικ μέσα στη λευκή επιφάνεια που μας έχει εμφανιστεί και επιλέγουμε Accept όπως φαίνεται στο παρακάτω σχήμα.



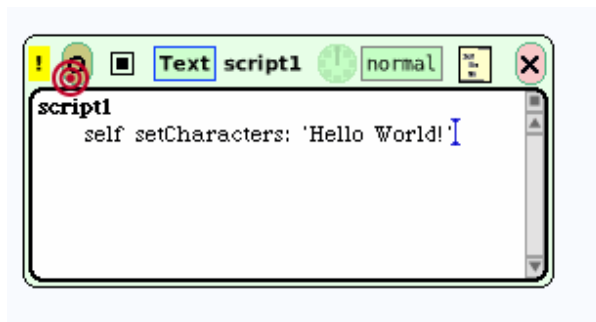
**Εικόνα 46: Επιλέγοντας την αποδοχή για το καινούριο κείμενο**

Στο παράθυρο που μας εμφανίζεται γράφουμε τα αρχικά μας γράμμα και πατούμε Accept. Εν συνεχεία πατούμε το κουμπί αποδοχής που βρίσκεται πάνω αριστερά στο ανοικτό παράθυρο.



**Εικόνα 47: Αποδοχή και εφαρμογή της αλλαγής που κάναμε**

Βλέπουμε πως το κείμενο αλλάζει από Text σε Hello World!. Για να κλείσουμε το παράθυρο που έχει ανοίξει πατούμε το κουμπί κλεισίματός του.



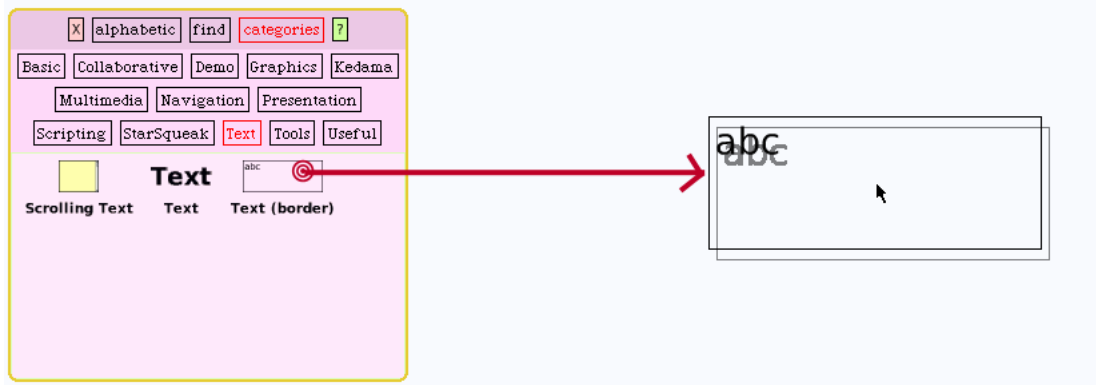
**Εικόνα 48: Κλείσιμο του παραθύρου**

Μπορούμε να πειραματιστούμε επαναλαμβάνοντας την ίδια διαδικασία για τα ίδια attributes που δοκιμάσαμε παραπάνω.

#### **4.2. Αντικείμενα, Οπτικός Προγραμματισμός**

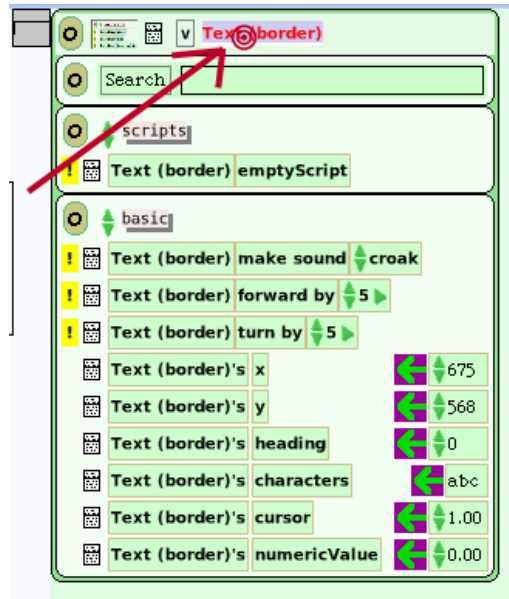
Σε αυτή την ενότητα γίνεται εισαγωγή στις βασικές έννοιες του squeak που είναι τα αντικείμενα και ο προγραμματισμός πάνω στα ιδεατά αντικείμενα. Στο συγκεκριμένο παράδειγμα, θα δημιουργήσουμε δύο αντικείμενα text (border) και θα αντιγράψουμε το περιεχόμενο του ενός στο άλλο.

Από εδώ και στο εξής θα ονομάζουμε την επιφάνεια εργασία του Squeak Κόσμο (World). Ανοίγουμε ένα νέο project και στον νέο κόσμο δημιουργούμε ένα αντικείμενο κειμένου με πλαίσιο (text border).



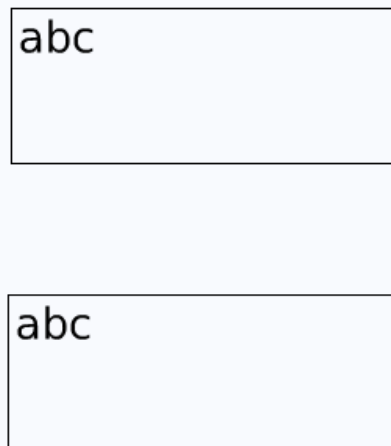
**Εικόνα 49: Δημιουργώντας ένα αντικείμενο κειμένου με πλαίσιο**

Είναι σημαντικό πολλές φορές για να γνωρίζουμε σε ποια αντικείμενα αναφερόμαστε να τους αλλάζουμε το όνομα. Για να γίνει αυτό, ανοίγουμε τα attributes του αντικειμένου και επιλέγουμε όπως φαίνεται στην παρακάτω εικόνα.



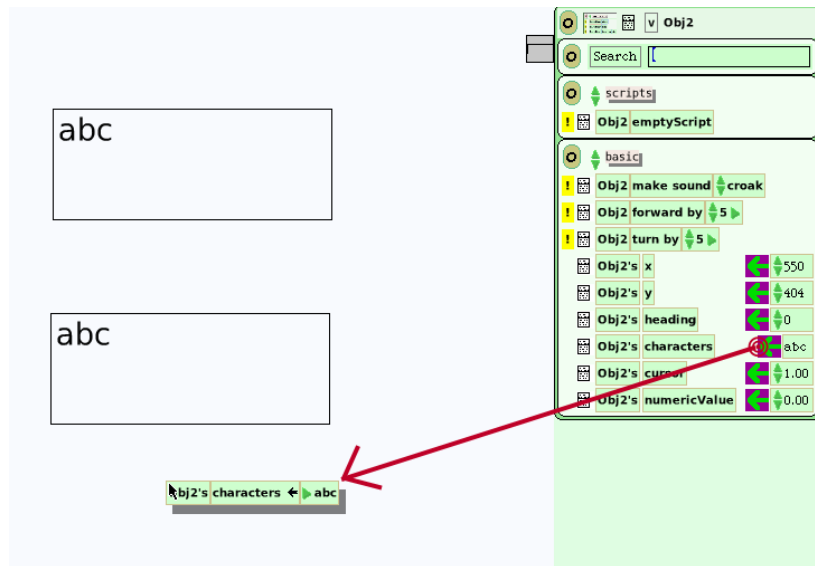
**Εικόνα 50: Αλλάζοντας το όνομα του αντικειμένου**

Επιλέγουμε το Text (border), γράφουμε Obj1 και πατούμε Enter. Το αντικείμενό μας έχει διαφορετική ονομασία τώρα. Για να πραγματοποιήσουμε τη διαδικασία θα πρέπει να φτιάξουμε ένα ακόμα αντικείμενο στο οποίο θα αντιγράψουμε τα περιεχόμενα του Obj1. Δημιουργούμε άλλο ένα αντικείμενο τύπου text (border) και το τοποθετούμε στην επιφάνεια όπως φαίνεται στην παρακάτω εικόνα.

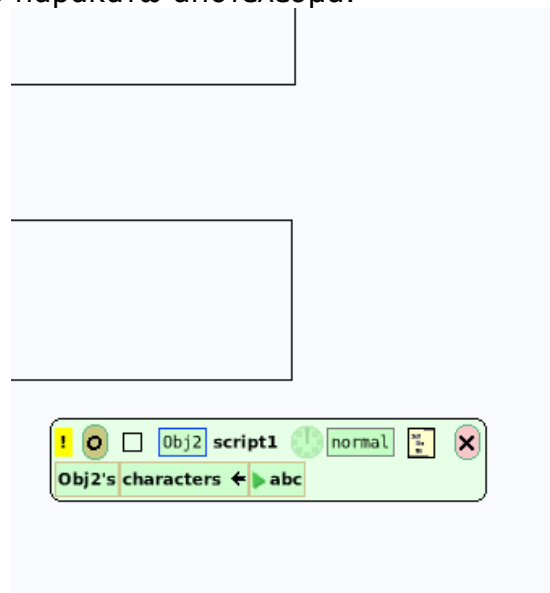


**Εικόνα 51: Αντικείμενα τύπου text (border) τοποθετημένα στον κόσμο**

Εν συνεχεία μετονομάζουμε το δεύτερο αντικείμενο σε Obj2 (όπως κάναμε παραπάνω) και θα αναφερόμαστε στα αντικείμενα σαν Obj1 και Obj2. Τώρα θα δούμε με ποιόν τρόπο μπορούμε να πραγματοποιήσουμε την αντιγραφή από ένα αντικείμενο σε ένα άλλο χρησιμοποιώντας μόνο επιλογές με το ποντίκι χωρίς να γράψουμε ούτε μία γραμμή κώδικα. Ανοίγουμε τα attributes του Obj2 και τοποθετούμε το Text characters στον κόσμο όπως φαίνεται στην παρακάτω εικόνα.

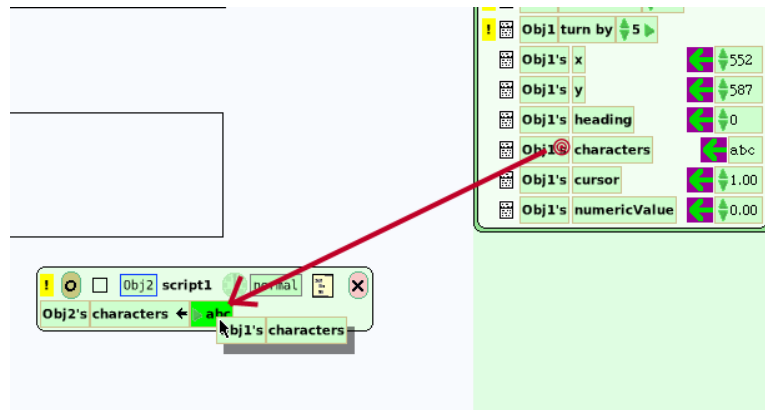


**Εικόνα 52: Εμφανίζοντας το Obj2 characters στον κόσμο (α)**  
...και παίρνουμε το παρακάτω αποτέλεσμα.



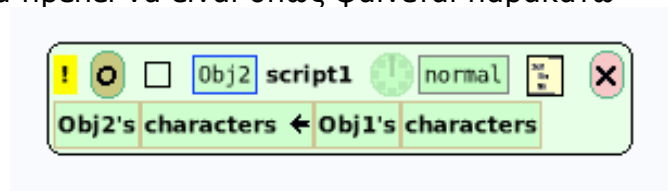
**Εικόνα 53: Εμφανίζοντας το Obj2 characters στον κόσμο (β)**

Εν συνεχεία ανοίγουμε τα attributes του αντικειμένου Obj1 και με τον ίδιο τρόπο σύρουμε το Obj1's characters πάνω στην επιλογή abc όπως φαίνεται στο παρακάτω σχήμα.



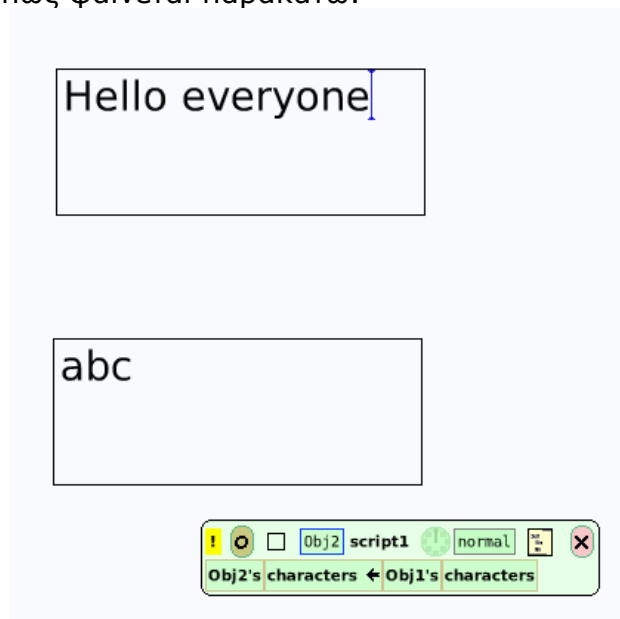
**Εικόνα 54: Τοποθετώντας το Obj1's characters στο abc του Obj2's characters**

Το αποτέλεσμα πρέπει να είναι όπως φαίνεται παρακάτω



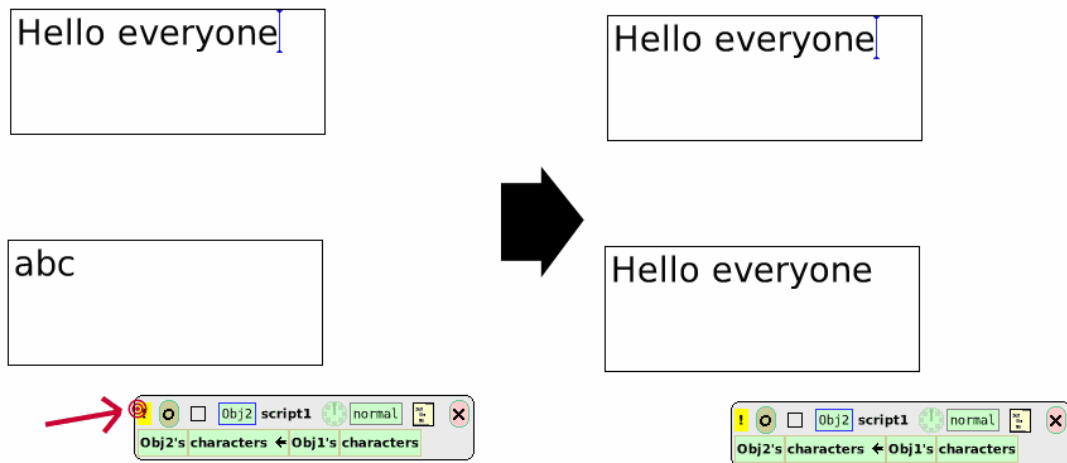
**Εικόνα 55: Αποτέλεσμα τοποθέτησης Obj1's characters στο abc του Obj2's characters**

Εν συνεχεία πατούμε αριστερό κλικ μέσα στο χώρο του Obj1 και γράφουμε μια δυο λέξεις όπως φαίνεται παρακάτω.



**Εικόνα 56: Γράφουμε Hello everyone στο Obj1**

Για να εκτελέσουμε το script που έχουμε δημιουργήσει πατούμε το κουμπί εκτέλεσης όπως φαίνεται παρακάτω. Το αποτέλεσμα που παίρνουμε φαίνεται παρακάτω.

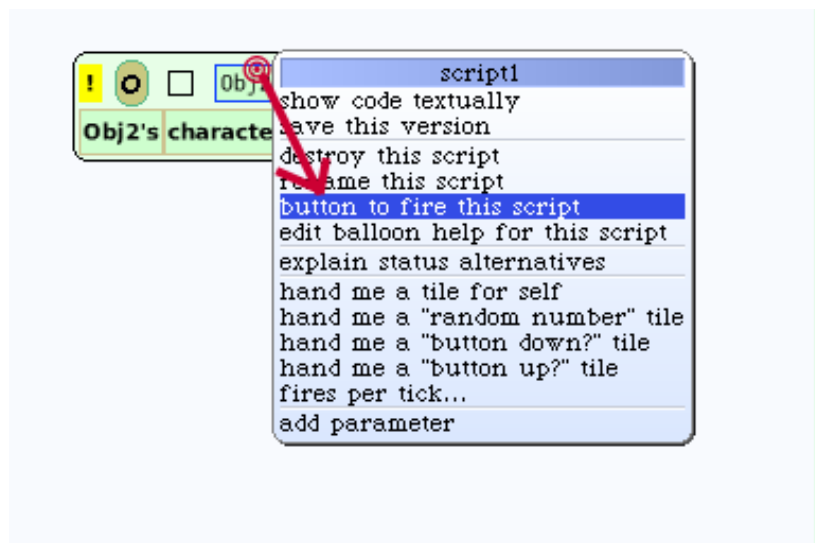


**Εικόνα 57: Αντιγραφή των χαρακτήρων του Obj1 στο Obj2**

Προκειμένου να γίνει η διαδικασία πιο φαντασμαγορική θα πραγματοποιήσουμε δύο βασικά στοιχεία. Θα προσθέσουμε ένα κουμπι στον κόσμο που θα πραγματοποιεί την εκτέλεση του script και κατά την αντιγραφή θα αντιγράψουμε το περιεχόμενο από το Obj1 στο Obj2.

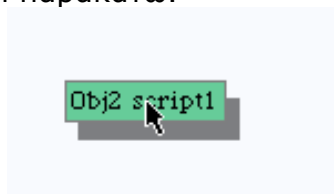
#### 4.2.1. Εκτέλεση με τη χρήση κουμπιού

Ενώ έχουμε ανοιχτά τα παραπάνω κάνουμε αριστερό κλικ στο κουμπι Obj2 και επιλέγουμε button to fire the script όπως φαίνεται στο παρακάτω σχήμα.



**Εικόνα 58: Δημιουργώντας ένα κουμπι για την εκτέλεση του script**

Συνέχεια εμφανίζεται ένα κουμπι και το τοποθετούμε όπου θέλουμε μέσα στον κόσμο όπως φαίνεται παρακάτω.



**Εικόνα 59: Το κουμπι για την εκτέλεση του κώδικα**

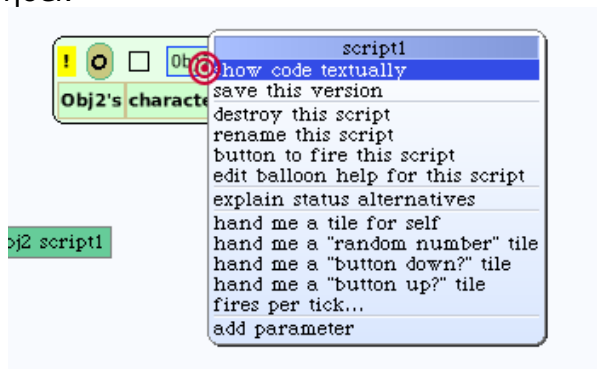
Πατώντας κλικ οπουδήποτε μέσα στον κόσμο το κουμπι τοποθετείται. Εν συνέχεια γράφουμε κάτι διαφορετικό μέσα στο Obj1 και πατούμε αυτό το

κουμπί. Παρατηρούμε πως πράγματι το περιεχόμενο αντιγράφεται από το Obj1 στο Obj2.

#### 4.2.2. Αντιγραφή και σβήσιμο του περιεχομένου του Obj1

Για να κάνουμε ακόμα πιο εντυπωσιακό το πρόγραμμά μας θα προσθέσουμε ένα ακόμα στοιχείο. Μόλις αντιγράφονται τα στοιχεία από το Obj1 στο Obj2 θα σβήνεται τα περιεχόμενα του Obj1.

Για να γίνει αυτό θα πρέπει να προσθέσουμε με το χέρι ένα μικρό κομμάτι κώδικα. Για το λόγο αυτό ανοίγουμε την προβολή κώδικα για το script που έχουμε δημιουργήσει.



**Εικόνα 60: Ανοίγοντας τον κώδικα του Script1**

Αφού πατήσουμε την προβολή κώδικα βλέπουμε πως ήδη γράφει τα παρακάτω στοιχεία:

```
script1
```

```
self setCharacters: Obj3 getCharacters
```

Προσθέτουμε μία ακόμα γραμμή στον κώδικα:

```
Obj3 setCharacters: `.`
```

Έτσι ο κώδικας να γίνει όπως παρακάτω:

```
script1
```

```
self setCharacters: Obj3 getCharacters.
```

```
Obj3 setCharacters: `.`
```

Προσέχουμε ιδιαίτερα την τελεία στο τέλος της δεύτερης πρότασης και πιο συγκεκριμένα μετά το `getCharacters`. Ουσιαστικά πρέπει να πάρουμε το αντικείμενο πριν το `getCharacters`, να το γράψουμε στην από κάτω γραμμή με την εντολή `setCharacters: `.`, δηλαδή αφού αντιγράψουμε τους χαρακτήρες του Obj3 στο δεύτερο αντικείμενο θα πρέπει να θέσουμε τους χαρακτήρες του Obj3 σε κενό.

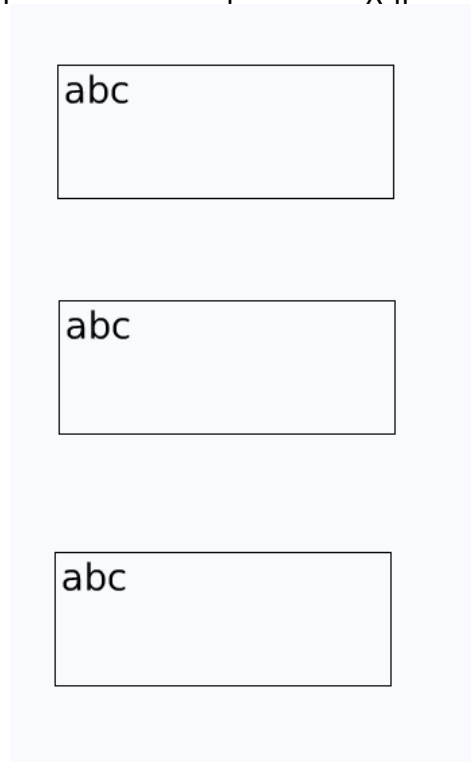
#### Σημείωση

Κάθε φορά που αλλάζει μία γραμμή στον κώδικα θα πρέπει να προστίθεται η τελεία!

### 4.3. Φτιάχνοντας μία αριθμομηχανή

Πρώτο «σύνθετο» παράδειγμα. Κατασκευή μίας αριθμομηχανής που εκτελεί τις 4 βασικές πράξεις μεταξύ δύο αριθμών. Σε πρώτη φάση θα φτιάξουμε τρία `texts` (`border`) και θα τα συνδέσουμε μεταξύ τους. Προκειμένου να μπορέσουμε να συνδέσουμε στοιχεία μεταξύ τους θα πρέπει να φτιάξουμε ένα `script` για το αντικείμενο στο οποίο θα εμφανίζονται τα αποτελέσματα και μέσα του να προσθέσουμε «ανώδυνα» στοιχεία για τα υπόλοιπα αντικείμενα.

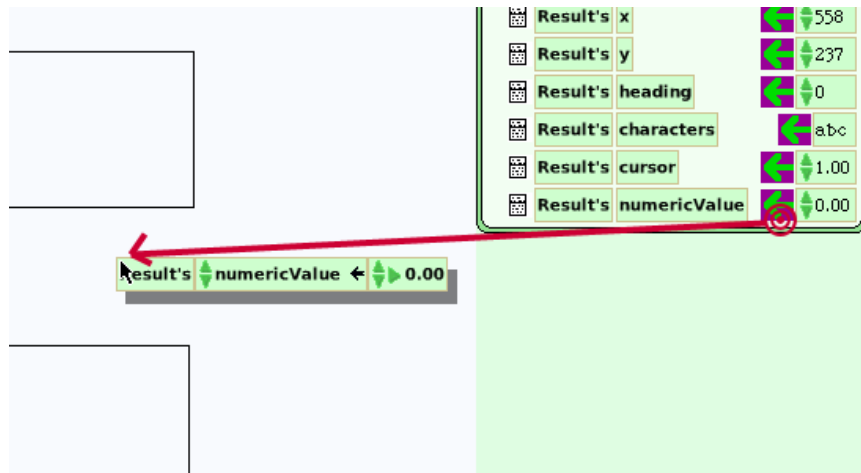
Ας τα πάρουμε όμως όλα από την αρχή. Αρχικά κατασκευάζουμε τα τρία `texts` (`border`) όπως φαίνεται στο παρακάτω σχήμα.



**Εικόνα 61: Κατασκευή τριών `text` (`border`)**

Εν συνεχεία ονομάζουμε τα τρία αντικείμενα. Για να ονομάσουμε τα αντικείμενα ακολουθούμε τις ίδιες οδηγίες με το προηγούμενο κεφάλαιο. Δίνουμε τις ονομασίες `Number1`, `Number2` και `Result` με τη σειρά που φαίνονται τα αντικείμενα από πάνω προς τα κάτω.

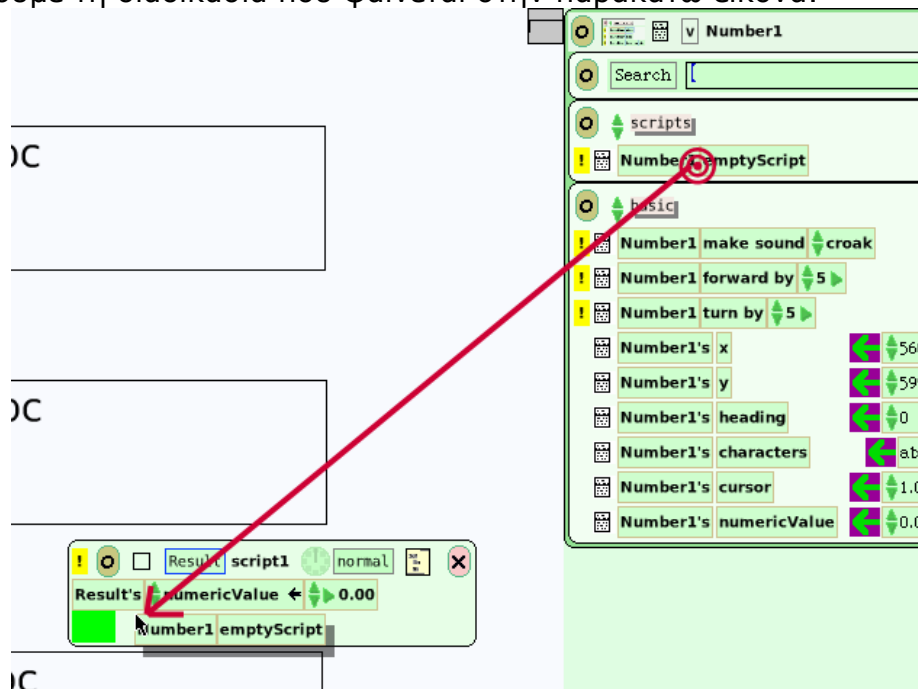
Στη συνέχεια ανοίγουμε τα `attributes` του `Result` και τοποθετούμε στον κόσμο όπως φαίνεται παρακάτω το `Result's numericValue`.



**Εικόνα 62: Τοποθέτηση του attribute numericValue στον κόσμο**

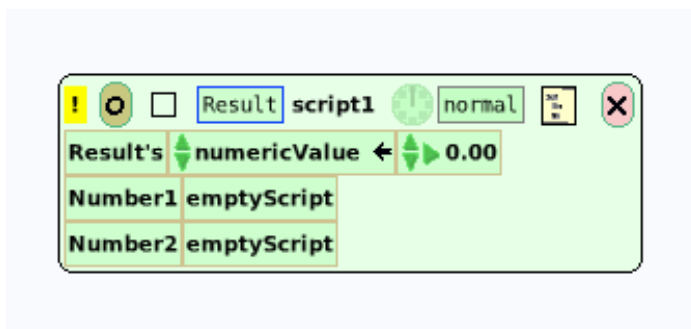
Σε αυτή τη φάση θα πρέπει να συνδέσουμε τα Number1 και Number2 με το Result. Για να το κάνουμε αυτό θα πρέπει να πάρουμε ένα ανώδυνο στοιχείο από το καθένα και να το τοποθετήσουμε μαζί με τα στοιχεία του Result στο script που φτιάχνουμε. Το πιο εύκολο είναι να χρησιμοποιήσουμε το attribute emptyScript.

Αφού ανοίξουμε τα attributes καθενός από τα Number1 και Number2 εκτελούμε τη διαδικασία που φαίνεται στην παρακάτω εικόνα.



**Εικόνα 63: Σύνδεση Number1 με το Result**

Όπως φαίνεται για να συνδέσουμε τα Number1 και 2 με το Result τοποθετούμε το attribute emptyScript μέσα στο script του Result. Αφού προσθέσουμε και τα δύο θα έχουμε.



**Εικόνα 64: Συνδεδεμένα Result, Number1 και Number2**

Εν συνεχεία θα αλλάξουμε τον κώδικα του script με τον ίδιο τρόπο με προηγουμένως.

Ο κώδικας που βλέπουμε μοιάζει με το παρακάτω κομμάτι:

```
script1
    self setNumericValue: 0.
    Number3 emptyScript.
    Number4 emptyScript
```

Τον αλλάζουμε ώστε να πραγματοποιεί πράξεις και τον μετατρέπουμε όπως φαίνεται παρακάτω:

```
script1
    self setNumericValue: Number3 getNumericValue + Number4
    getNumericValue.
```

Στα στοιχεία δεν προσθέτουμε τα πραγματικά ονόματα των αντικειμένων αλλά αυτά που φαίνονταν μέσα στο πρώτο script, δηλαδή εν προκειμένω Number3 και Number4.

#### Σημείωση

Δεν ξεχνούμε πως αφού τελειώσουμε τον κώδικα θα πρέπει να πατήσουμε δεξί κλικ και Accept!

Αφού τελειώσουμε μπορούμε να πατήσουμε το κουμπί εκτέλεσης ή να δημιουργήσουμε ένα κουμπί για εκτέλεση για να δούμε το αποτέλεσμα.

### 4.3.1. Επέκταση αριθμομηχανής με περισσότερες πράξεις

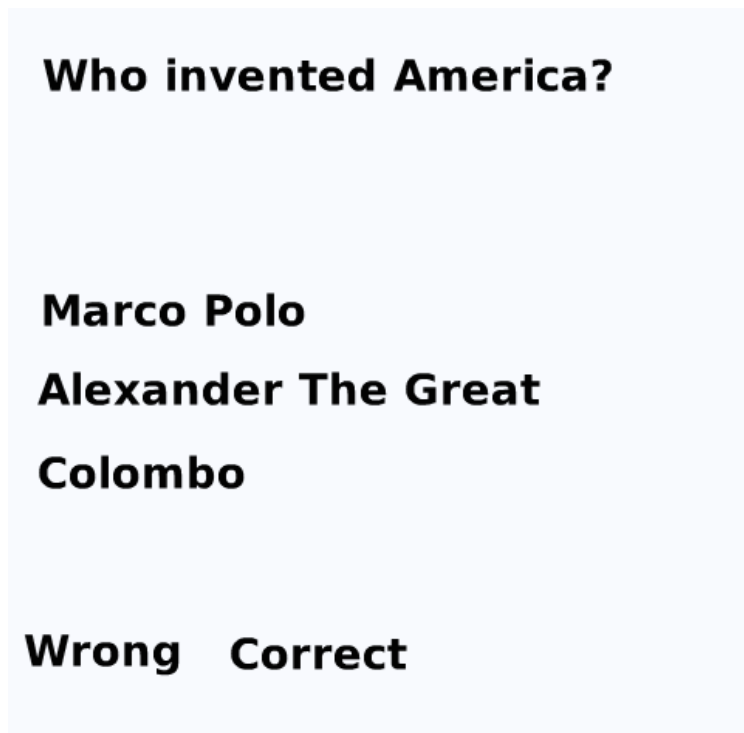
Προκειμένου να επεκτείνουμε την αριθμομηχανή προκειμένου να κάνει περισσότερες πράξεις θα πρέπει να επαναλάβουμε από την αρχή την παραπάνω διαδικασία από το σημείο που λαμβάνουμε το numericValue του Result και ξεκινάμε τη διαδικασία σύνδεσης με τα Result1 και Result2. Αυτό έχει σαν συνέπεια να δημιουργηθεί ένα επιπλέον script το οποίο θα εκτελεί κάποια διαφορετική πράξη όπως αφαίρεση πολλαπλασιασμό κλπ.

### 4.4. Φτιάχνοντας ένα κουίζ

Κατασκευή ενός κουίζ ερωτήσεων. Στην οθόνη εμφανίζεται ερώτηση με πιθανές απαντήσεις και ο χρήστης καλείται να επιλέξει τη σωστή απάντηση. Προκειμένου να δημιουργήσουμε το Κουίζ θα πρέπει να δημιουργήσουμε πρώτα τα πιθανά αντικείμενα που θα χρησιμοποιηθούν. Για το λόγο αυτό

θα χρησιμοποιηθούν απλά Text αντικείμενα. Τα αντικείμενα που θα χρειαστούμε είναι ένα αντικείμενο Text που θα είναι η ερώτηση, τρία αντικείμενα Text που θα είναι οι πιθανές απαντήσεις και δύο επιπλέον αντικείμενα Text που θα είναι η ανταπόκριση στην απάντηση των χρηστών και θα είναι τα ενδεχόμενα Σωστό (Correct) ή Λάθος (Wrong).

Αρχικά δημιουργούμε όλα τα αντικείμενα όπως φαίνεται στον παρακάτω κόσμο.

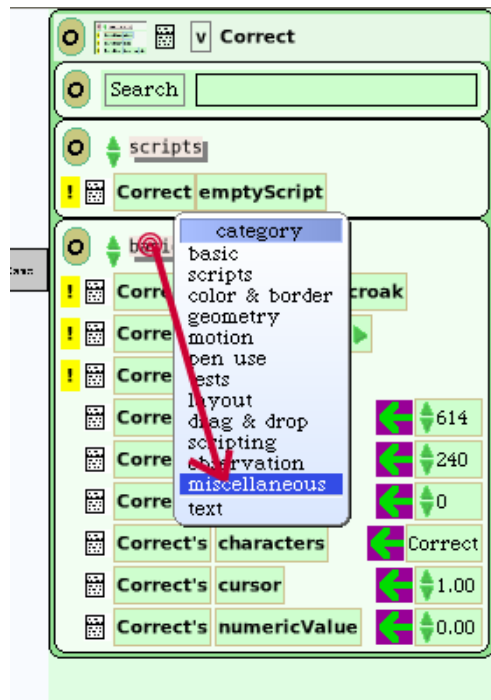


**Εικόνα 65: Έξι αντικείμενα Text για τη δημιουργία quiz**

Σκεφτόμαστε τη διαδικασία την οποία θα ακολουθήσουμε. Η διαδικασία θα είναι η εξής. Αρχικά, θα εμφανίζεται μόνο η ερώτηση και ένα κουμπί δίπλα στην ερώτηση. Εν συνεχεία, με το πάτημα του κουμπιού θα ξεκινά το τεστ και θα εμφανίζονται οι απαντήσεις και κουμπιά δίπλα σε κάθε απάντηση. Τέλος, ανάλογα με το κουμπί πιθανής απάντησης που θα επιλέγει ο χρήστης θα εμφανίζεται αν η απάντηση ήταν σωστή ή λάθος.

Αρχικά θα πρέπει να δώσουμε ένα όνομα σε κάθε αντικείμενο. Την ερώτηση θα την ονομάσουμε Question, τις απαντήσεις θα τις ονομάσουμε διαδοχικά Answer1, Answer2 και Answer3. Τέλος, τις αποκρίσεις θα τις ονομάσουμε Wrong και Correct αντίστοιχα με τα ονόματά τους.

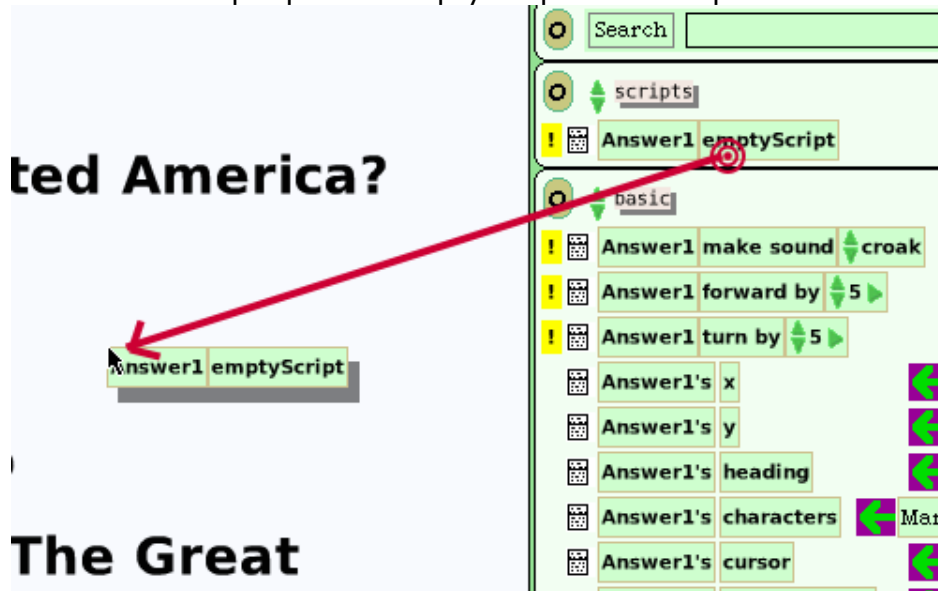
Αρχικά δημιουργούμε τα κουμπιά των απαντήσεων. Κάθε κουμπί θα πρέπει να έχει ως αποτέλεσμα την εμφάνιση των Wrong ή Correct. Η εμφάνιση και το σβήσιμο του text γίνεται με τις μεθόδους Show και Hide που βρίσκονται στην περιοχή miscellaneous των attributes. Για να εμφανίσουμε τα miscellaneous ανοίγουμε τα attributes του αντικειμένου Wrong και επιλέγουμε με αριστερό κλικ όπως φαίνεται στην παρακάτω εικόνα.



**Εικόνα 66: επιλέγοντας τα miscellaneous attributes**

Ας δημιουργήσουμε όμως τα κουμπιά για κάθε ερώτηση. Η λογική είναι η εξής, μόλις πατάμε το κουμπί της λάθος απάντησης θα πρέπει να εμφανίζεται το wrong ενώ μόλις πατάμε το κουμπί της σωστής ερώτησης θα πρέπει να εμφανίζεται το correct.

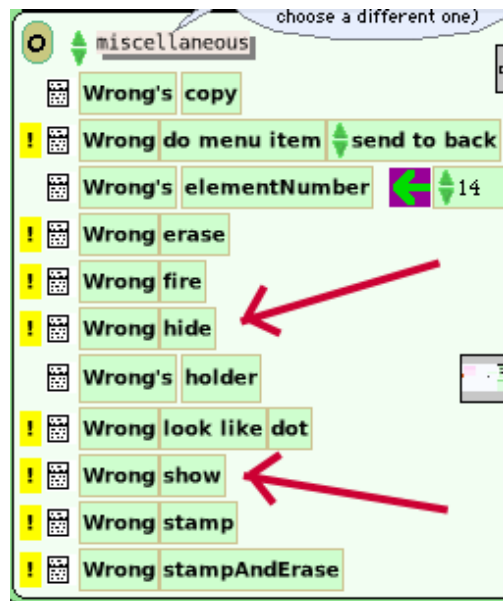
Ξεκινούμε με το αντικείμενο Answer1 (που οδηγεί σε λάθος). Ανοίγουμε τα attributes του και σύρουμε ένα emptyScript στον κόσμο.



**Εικόνα 67: Δημιουργώντας ένα νέο Script**

Όταν πατήσει ο χρήστης αυτή την απάντηση θα πρέπει να εμφανίζεται το Wrong και να σβήνει το Correct.

Ανοίγουμε λοιπόν πρώτα τα miscellaneous attributes του wrong.



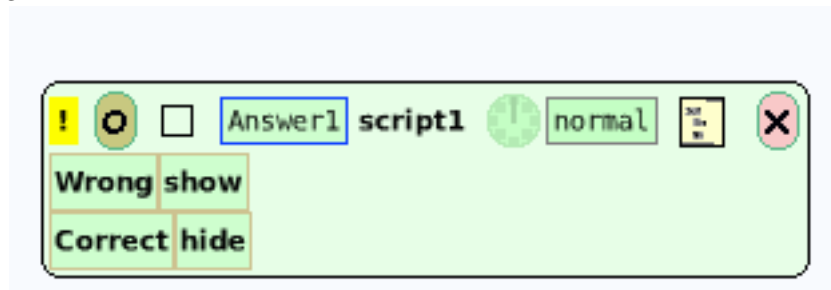
**Εικόνα 68: Τα miscellaneous attributes του αντικειμένου Wrong**

Αυτό που θέλουμε είναι πως όταν κάποιος πατά το κουμπί του script που έχουμε ανοιχτό στον κόσμο να εμφανίζεται το Wrong. Πιάνουμε λοιπόν το Wrong Show και το σύρουμε μέσα στο ανοιχτό script στον κόσμο.



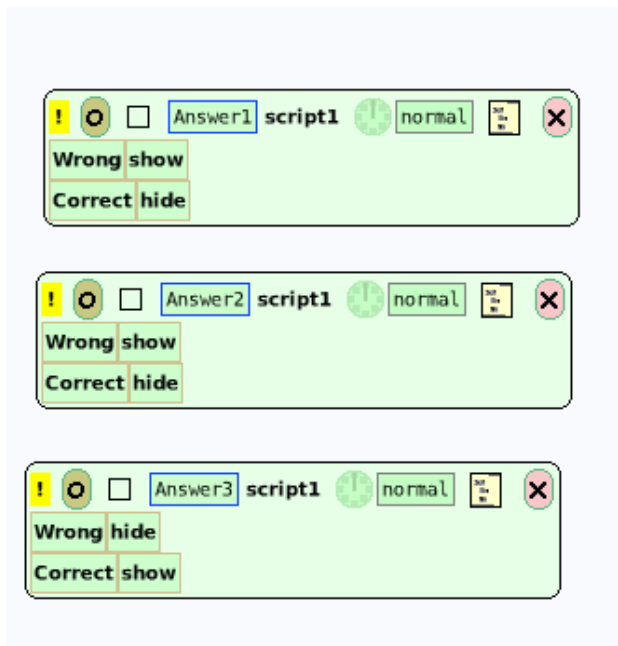
**Εικόνα 69: προσθέτοντας ένα attribute μέσα στο script**

Την ίδια διαδικασία πραγματοποιούμε και για το αντικείμενο correct, μόνο που αυτή τη φορά παίρνουμε το attribute hide και το προσθέτουμε μέσα στο script.



**Εικόνα 70: Το script μιας εκ των απαντήσεων (λάθος απάντηση)**

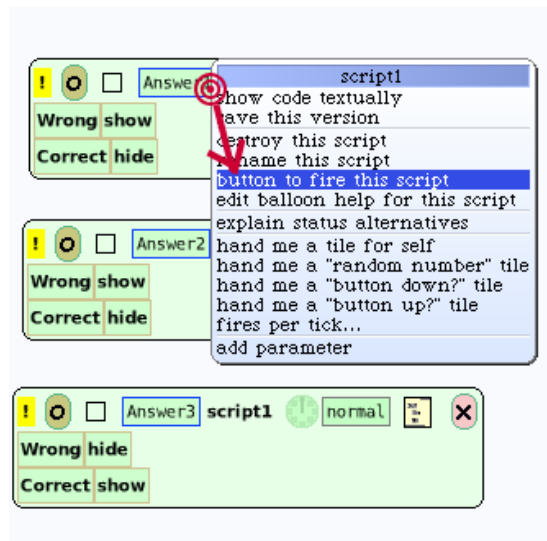
Επαναλαμβάνουμε την ίδια διαδικασία για όλες τις πιθανές απαντήσεις και στο τέλος θα έχουμε τρία scripts.



**Εικόνα 71: Τα scripts όλων των απαντήσεων**

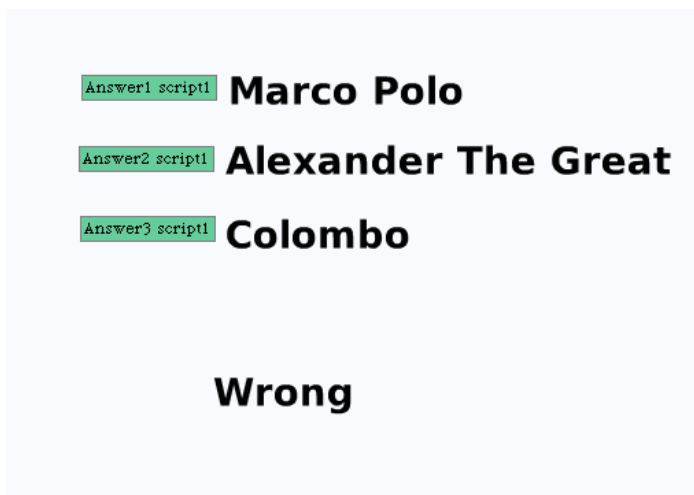
Όπως φαίνεται και από την παραπάνω εικόνα, η σωστή απάντηση θα έχει Show στο Correct και Hide στο Wrong.

Για καθένα από τα scripts θα πρέπει να δημιουργήσουμε ένα κουμπί όπως έχουμε μάθει και να το τοποθετήσουμε δίπλα στην απάντηση την οποία αφορά.



**Εικόνα 72: Δημιουργία κουμπιού για το script**

Μετακινούμε το κουμπί δίπλα στην απάντηση που αφορά για κάθε κουμπί που έχουμε.



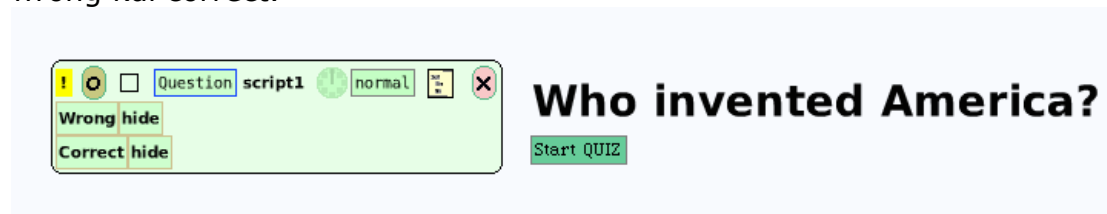
**Εικόνα 73: Ερωτήσεις με τις απαντήσεις τους**

Έχουμε δημιουργήσει όλες τις ερωτήσεις με τα κουμπιά ενεργοποίησής τους αλλά θέλουμε να τους αλλάξουμε το κείμενο προκειμένου να μη λένε: Answer1 script1.

Κάνουμε δεξί κλικ πάνω στο κάθε κουμπί και επιλέγουμε change label. Εκεί γράφουμε ότι θέλουμε να λέει το κουμπί.

Τέλος προκειμένου να τελειοποιήσουμε το τεστ θα δημιουργήσουμε ένα κουμπί αρχικοποίησης το οποίο θα εμφανίζει τα κουμπιά των πιθανών απαντήσεων και θα εξαφανίζει εντελώς τα στοιχεία Wrong και Correct πριν δώσουμε μία απάντηση.

Φτιάχνουμε λοιπόν ένα script για το αντικείμενο Question. Μέσα θα βάλουμε τα attributes HIDE για καθένα από τα αντικείμενα απαντήσεων wrong και correct.



**Εικόνα 74: Το script και το κουμπί του script για το αντικείμενο question.**

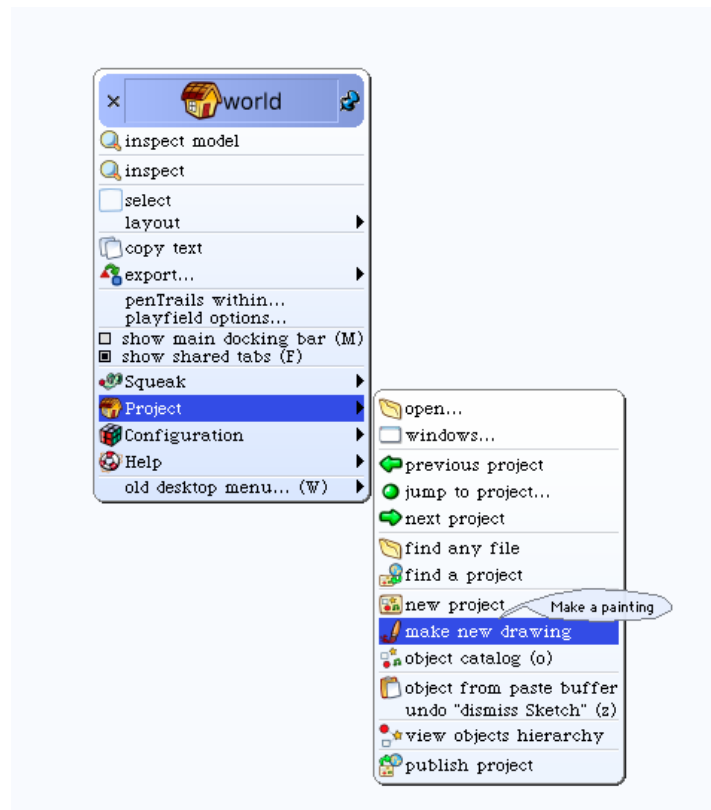
Το τεστ μας είναι έτοιμο και τα αποτελέσματα από τη χρήση του απεικονίζονται παρακάτω.



Εικόνα 75: Εκτέλεση του test

#### 4.5. Σχεδιάζοντας Δυσδιάστατα Αντικείμενα

Σε αυτό το παράδειγμα αναλύεται ο τρόπος με τον οποίο δημιουργούνται αντικείμενα δύο διαστάσεων. Ουσιαστικά πρόκειται για τον τρόπο με τον οποίο πραγματοποιείται η «ζωγραφική» μέσα στον κόσμο του squeak. Για να μπορέσουμε να ζωγραφίσουμε μέσα στο squeak θα πρέπει να ανοίξουμε το παράθυρο ζωγραφικής. Για να γίνει αυτό πατούμε δεξί κλικ στην επιφάνεια εργασίας και επιλέγουμε project -> make new drawing.



**Εικόνα 76: Ανοίγοντας το παράθυρο ζωγραφικής**

#### Σημείωση

**Etoys:** Για να ανοίξουμε τα εργαλεία ζωγραφικής από το Squeak Etoys θα πρέπει να επιλέξουμε από το Navigator το κουμπί που απεικονίζει ένα πινέλο.

Αυτό που εμφανίζεται πατώντας στο παράθυρο ζωγραφικής είναι το μενού που φαίνεται παρακάτω. Παραθέτονται και επεξηγήσεις των συμβόλων.

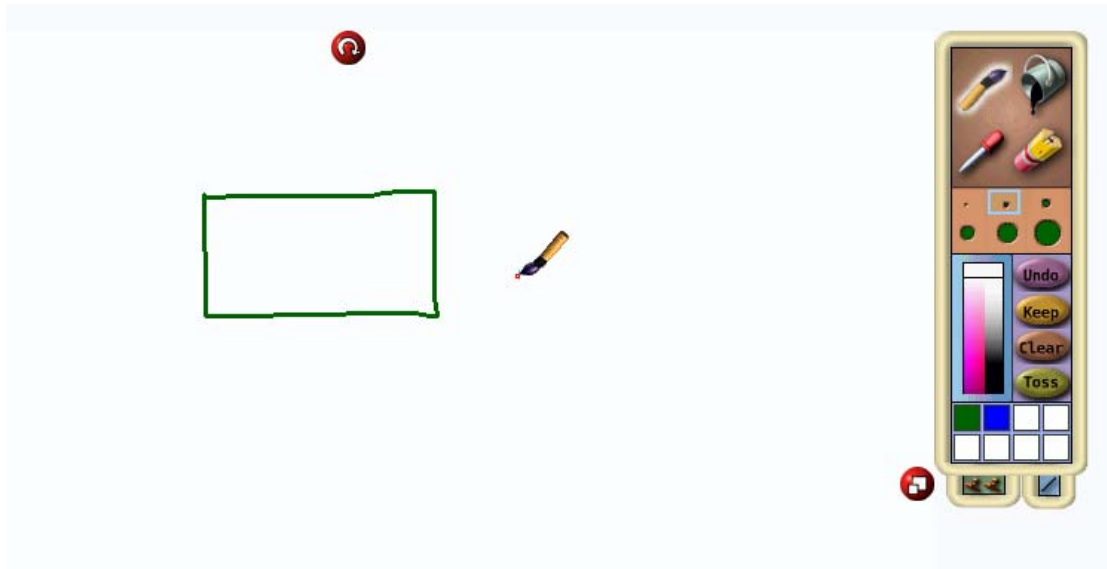


**Εικόνα 77: Παράθυρο ζωγραφικής**

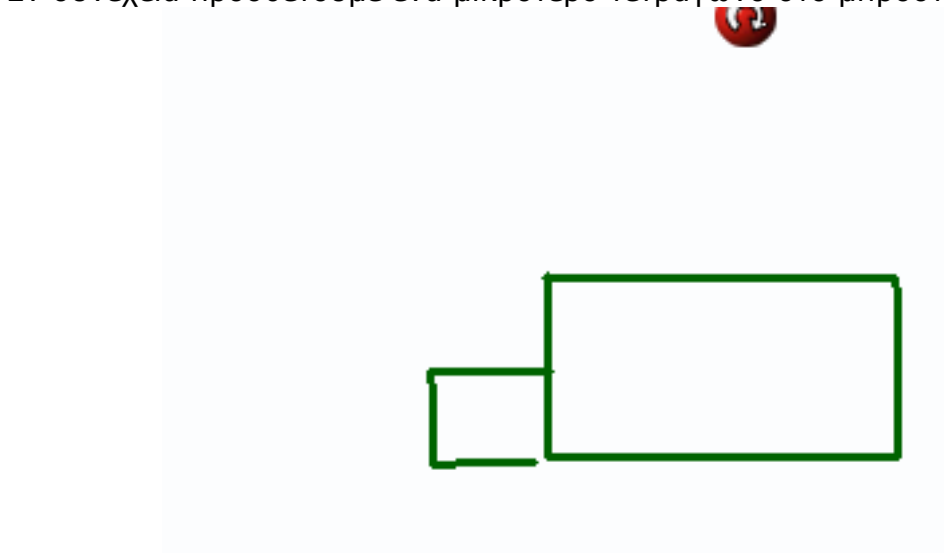
Προκειμένου να ζωγραφίσουμε επιλέγουμε ένα χρώμα από την επιλογή χρώματος και παίρνουμε το μολύβι. Επιλέγουμε το δεύτερο μέγεθος κέρσορα και δοκιμάζουμε να σχεδιάσουμε σχήματα στην επιφάνεια κάτω από την περιστροφή σχεδίου. Σε πρώτη φάση καλό είναι να πειραματιστούμε με όλες τις επιλογές που μας δίνει το παράθυρο ζωγραφικής για καλύτερη εξοικείωση. Σε κάθε περίπτωση μόλις είναι έτοιμο το σχέδιό μας επιλέγουμε Keep από τα μικρά κουμπιά του παραθύρου σχεδιασμού και το σχέδιο μας γίνεται αντικείμενο στην επιφάνεια εργασίας.

#### 4.5.1. Φτιάχνοντας το πρώτο μου αυτοκίνητο

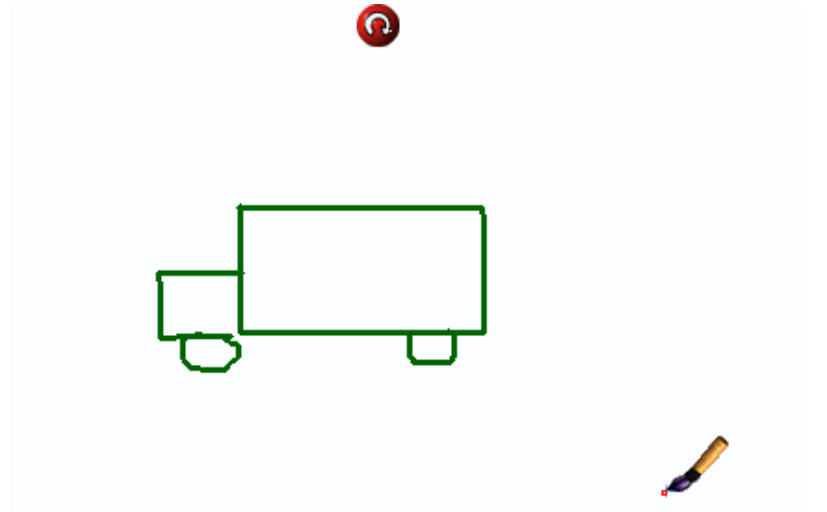
Ως πρώτο παράδειγμα για αντικείμενα δύο διαστάσεων επιλέγεται η κατασκευή ενός αυτοκινήτου. Ανοίγουμε το παράθυρο ζωγραφικής. Προκειμένου να ζωγραφίσουμε επιλέγουμε ένα χρώμα από την επιλογή χρώματος και παίρνουμε το μολύβι. Επιλέγουμε λεπτό κέρσορα για να μη βγει χοντροκομμένο το σχέδιο και σχεδιάζουμε στην αρχή ένα τετράγωνο.



**Εικόνα 78: Σχεδιάζοντας το πρώτο μου αυτοκίνητο βήμα 1**  
Εν συνεχεία προσθέτουμε ένα μικρότερο τετράγωνο στο μπροστά μέρος.



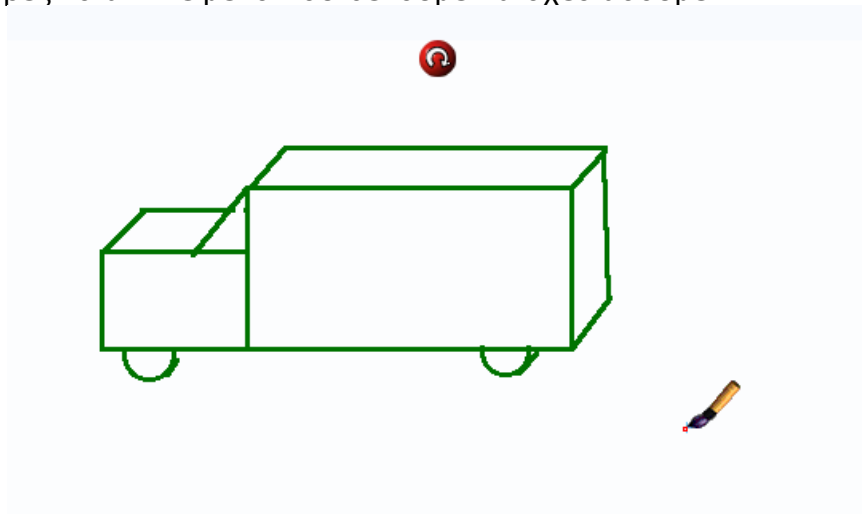
**Εικόνα 79: Σχεδιάζοντας το πρώτο μου αυτοκίνητο βήμα 2**  
Τέλος σχεδιάζουμε δύο κύκλους για ρόδες



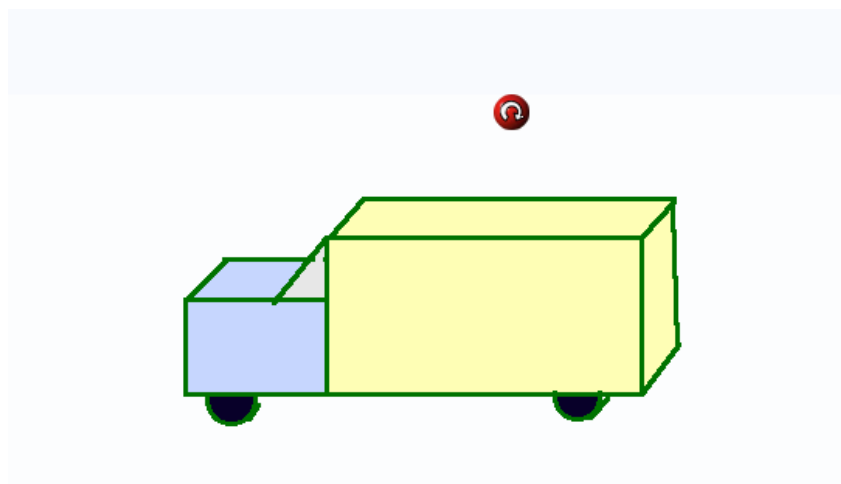
**Εικόνα 80: Σχεδιάζοντας το πρώτο μου αυτοκίνητο βήμα 3**

Ύστερα επιλέγουμε από το μενού δεξιά την επιλογή Keep. Το παράθυρο ζωγραφικής κλείνει και επιστρέφουμε στον κόσμο του squeak με το αντικείμενο-αυτοκίνητο έτοιμο.

Μπορούμε να πειραματιστούμε ακόμα περισσότερο με τα εργαλεία ζωγραφικής και μάλιστα να χρησιμοποιήσουμε έτοιμα σχήματα για να είναι πιο ακριβές το αντικείμενο που θέλουμε να σχεδιάσουμε.



**Εικόνα 81: αυτοκίνητο σχεδιασμένο με έτοιμα σχήματα**



**Εικόνα 82: ολοκλήρωση σχεδίου με προσθήκη χρώματος (γέμισμα)**

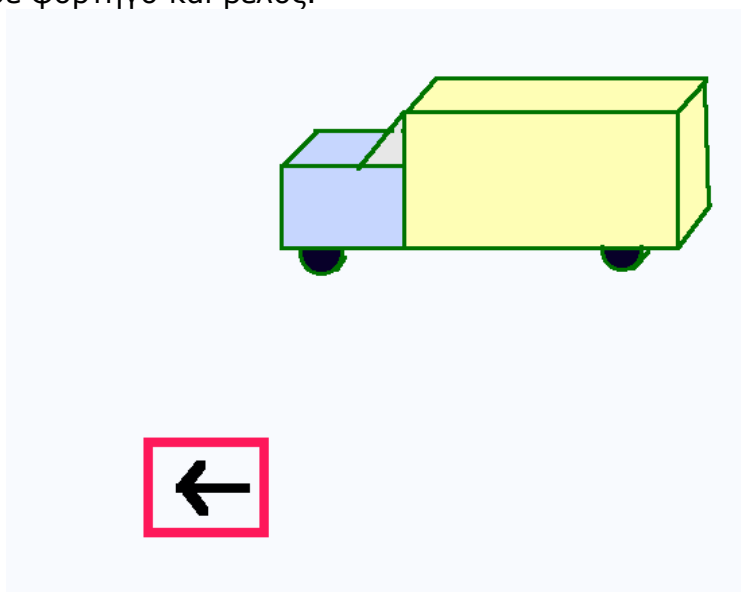
#### **4.6. Βάζοντας Κίνηση σε Αντικείμενα**

Σε συνδυασμό με το παραπάνω παράδειγμα αυτό το παράδειγμα έχει σαν σκοπό να δείξει πως βάζουμε κίνηση σε αντικείμενα. Αφού μάθαμε πώς να σχεδιάζουμε αντικείμενα ήρθε η ώρα να βάλουμε το αντικείμενό μας σε κίνηση. Πιο συγκεκριμένα μέσα από το συγκεκριμένο παράδειγμα θα μάθουμε πώς να κάνουμε το αυτοκίνητό μας να κινείται.

Αρχικά και αφού έχουμε σχεδιάσει το αυτοκίνητο πατούμε το πλήκτρο Keer για να κλείσει το παράθυρο ζωγραφικής και να επιστρέψουμε στον κόσμο του squeak με το αυτοκίνητο-αντικείμενο.

Προκειμένου να δώσουμε κίνηση στο φορτηγό το σκεπτικό είναι να δημιουργήσουμε ένα κουμπί στο πάτημα του οποίου θα αλλάζουμε τη συνισταμένη X του αντικειμένου του φορτηγού. Σε πρώτη φάση δημιουργούμε ένα αντικείμενο-κουμπί με το πάτημα του οποίου θα πραγματοποιείται κίνηση.

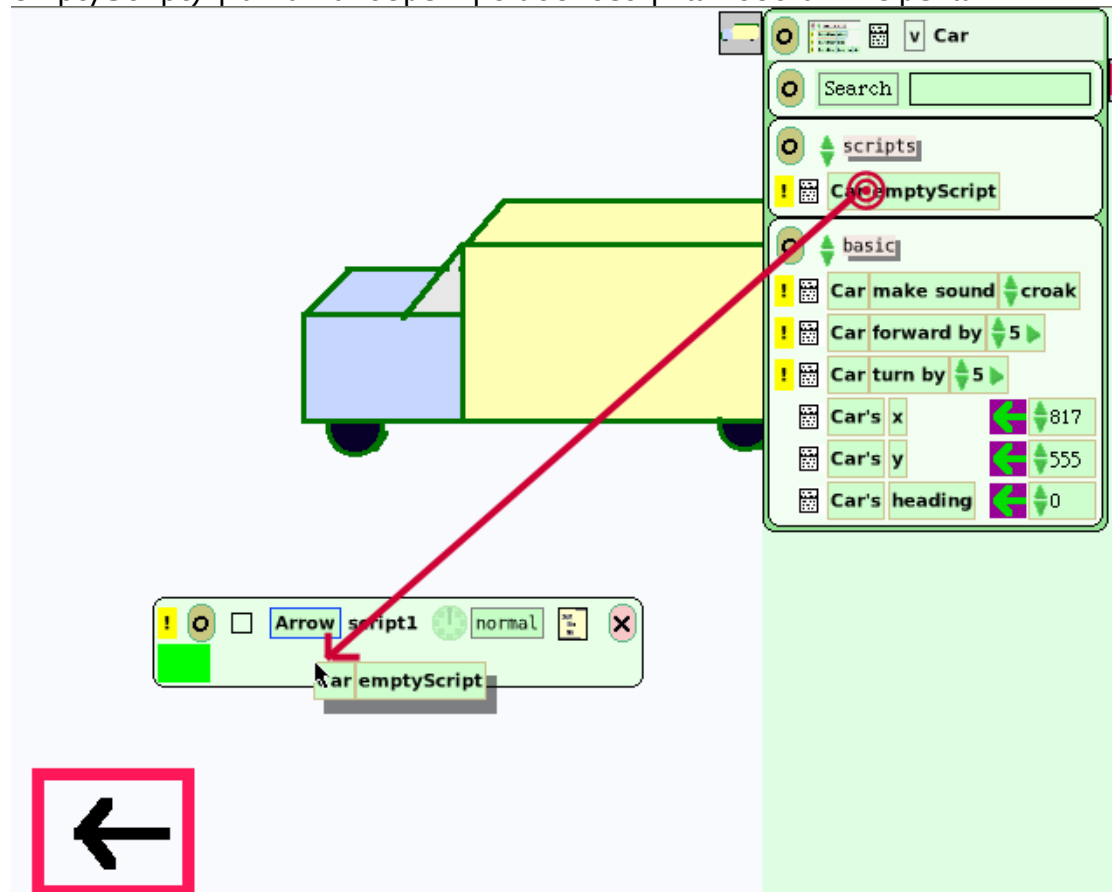
Στην επιφάνεια εργασίας έχουμε τώρα το φορτηγό μας και το σχέδιο που υποδεικνύει πώς να γίνεται η κίνηση. Δίνουμε τα ονόματα Car και Arrow αντίστοιχα σε φορτηγό και βέλος.



**Εικόνα 83: Αυτοκίνητο και σχέδιο/πλήκτρο μετακίνησης**

Για να μπορούμε να συνδυάσουμε τα δύο στοιχεία θα πρέπει να ανοίξουμε ένα script που θα αφορά το βέλος και να «βάλουμε» το στοιχείο φορητό μέσα στο script. Μάλιστα αυτό που μας ενδιαφέρει είναι το attribute Car X. Ανοίγουμε τα attributes του βέλους και δημιουργούμε ένα νέο script στον κόσμο. (βλέπε Εικόνα 67: Δημιουργώντας ένα νέο Script).

Εν συνεχεία ανοίγουμε τα attributes του Car και προσθέτουμε ένα ανώδυνο attribute μέσα στο script του Arrow (εν προκειμένω το Car emptyScript) για να κάνουμε τη διασύνδεση των δύο αντικειμένων.



**Εικόνα 84: Διασυνδέοντας αυτοκίνητο και βέλος**

Εν συνεχεία ανοίγουμε την επεξεργασία κώδικα για το script. Σβήνουμε τα πάντα εκτός από την αρχική φράση

Script1

Γράφουμε ώστε ο κώδικας να γίνει

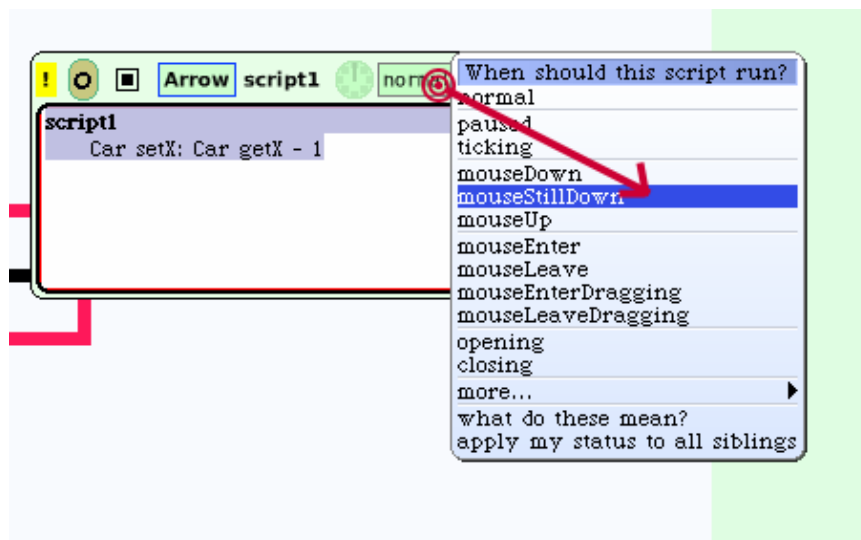
script1

Car setX: Car getX - 1

Προσοχή

Δεν ξεχνούμε να πατήσουμε αποδοχή ΚΑΘΕ φορά που γράφουμε ή αλλάζουμε τον κώδικα.

Εν συνεχεία πρέπει να επιλέξουμε πως θα εκτελείται ο κώδικας για το λόγο αυτό ακολουθούμε το παρακάτω σχήμα.



**Εικόνα 85: Επιλέγοντας πως θα εκτελείται το script.**

Αυτό που πρέπει να κάνουμε είναι αριστερό κλικ πάνω στη λέξη Normal και να επιλέξουμε MouseStillDown. Αυτό σημαίνει πως όσο κρατάμε πατημένο το αριστερό πλήκτρο του ποντικιού πάνω στο βέλος που έχουμε κατασκευάσει θα αλλάζει η θέση του αντικειμένου Car κατά 1pixel (1 μονάδα οθόνης) προς τα αριστερά. Την αριστερή φορά δείχνει το αρνητικό σύμβολο.

Ας πειραματιστούμε λίγο με αυτό που κατασκευάσαμε.

Αλλάζουμε τον κώδικα και αντί για -1 γράφουμε -5. Δεν ξεχνούμε να κάνουμε Accept μετά τις αλλαγές στον κώδικα και πριν εκτελέσουμε το Script. Παρατηρούμε πως μόλις πατήσουμε και κρατήσουμε πατημένο το αριστερό κλικ πάνω στο βέλος το αυτοκίνητο κινείται πιο γρήγορα. Αυτό είναι επόμενο αφού τώρα «λέμε» στο αυτοκίνητο να κινείται 5 μονάδες οθόνης προς τα αριστερά.

Ας πειραματιστούμε λίγο ακόμα.

Αλλάζουμε τον κώδικα και αντί για μείον βάζουμε συν. Δηλαδή ο κώδικας που έχουμε είναι:

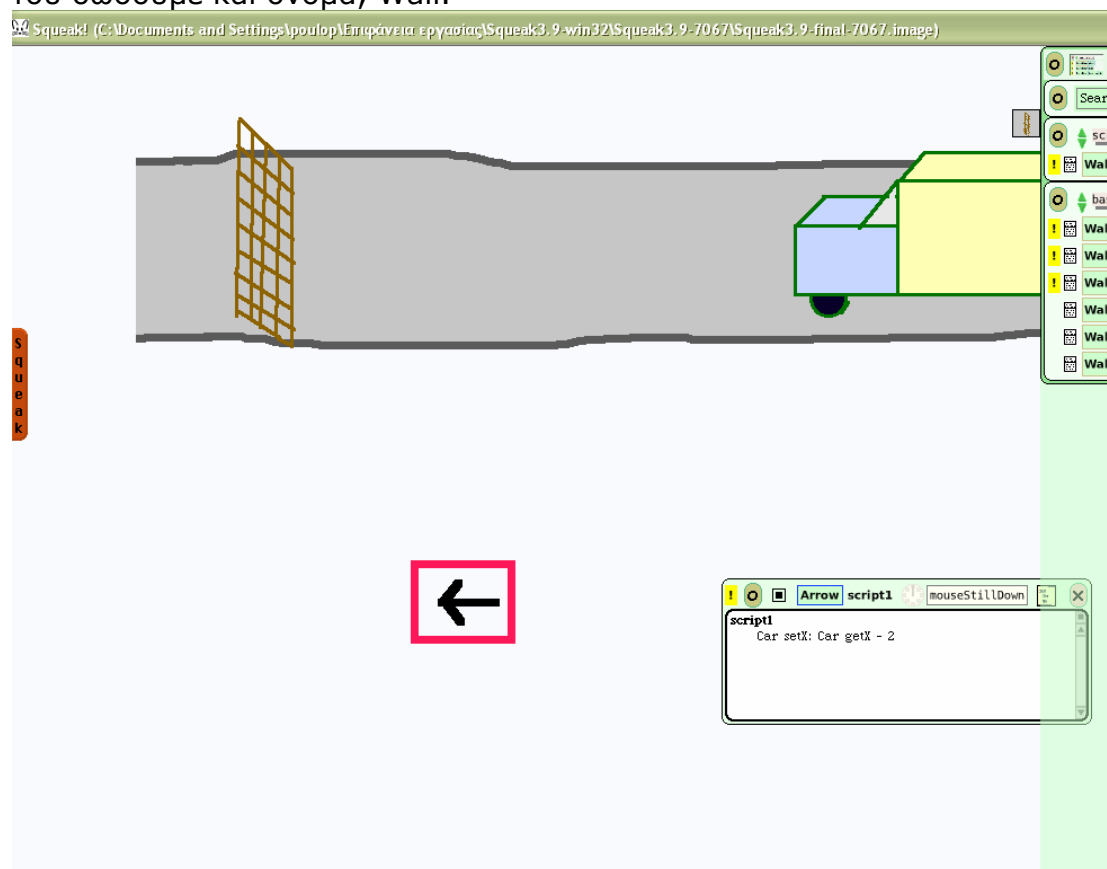
```
script1
  Car setX: Car getX + 1
```

Βλέπουμε πως το αυτοκίνητό μας πηγαίνει... όπισθεν. Αυτό είναι φυσικό αφού τώρα είπαμε στο αντικείμενο αυτοκίνητο να πηγαίνει δεξιά (πρόσημο συν [+]).

#### **4.7. Δημιουργώντας «συγκρούσεις»**

Σε αυτό το παράδειγμα θα αναλυθεί ο τρόπος με τον οποίο μπορεί να συνδυαστεί η κίνηση ενός αντικειμένου με μία σύγκρουση με εμπόδιο που βρίσκεται στο δρόμο του. Σε αυτό το σημείο θα δούμε πως μπορούμε να εισάγουμε τη λογική των συγκρούσεων μέσα στο πρόγραμμά μας. Για να το καταφέρουμε αυτό θα πρέπει να έχουμε υπόψη μας την ύπαρξη του attribute override. Όταν ένα αντικείμενο αγγίζει ένα άλλο, τότε η μεταβλητή αυτή παίρνει την τιμή ένα. Θα εκμεταλλευτούμε αυτό το γεγονός προκειμένου να «καταλάβουμε» πότε το αντικείμενό μας θα ακουμπήσει το αντικείμενο τείχος.

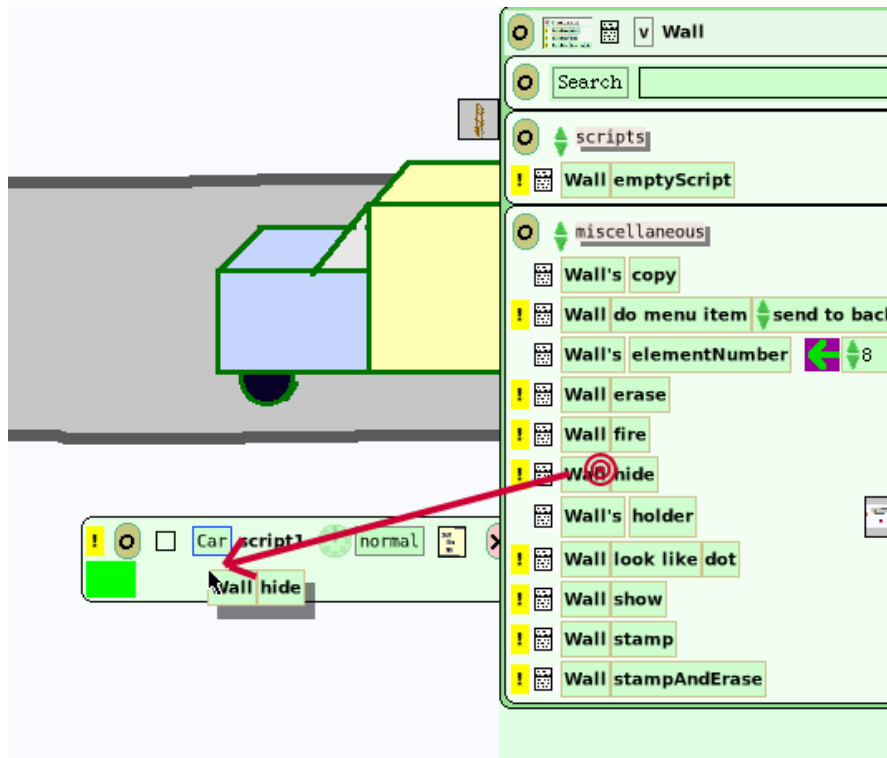
Για το λόγο αυτό στα δύο αντικείμενα που έχουμε από το προηγούμενο παράδειγμα, προσθέσουμε ένα δρόμο και έναν τοίχο όπως φαίνεται στο πιο κάτω σχήμα. Μάλιστα αφού θα χρησιμοποιήσουμε το αντικείμενο/τοιχος θα του δώσουμε και όνομα, Wall.



**Εικόνα 86: Αυτοκίνητο, βέλος, δρόμος και τοίχος**

Αυτό που κάνουμε είναι να προσθέσουμε ένα script στο φορηγό που άλλωστε είναι το κινητό αντικείμενο και θα το διασυνδέσουμε με τον τοίχο. Αυτό που θέλουμε να συμβαίνει σε πρώτη φάση είναι να καταλάβουμε πότε υπάρχει σύγκρουση και με τη σύγκρουση να σπάσει ο τοίχος.

Ανοίγουμε τα attributes του Car και δημιουργούμε ένα νέο script. Για να το διασυνδέσουμε με τον αντικείμενο Wall θα πρέπει να προσθέσουμε ένα attribute του τοίχου μέσα στο script του car. Συγκεκριμένα αυτό που θέλουμε είναι να σπάει/εξαφανίζεται ο τοίχος, άρα μία εύκολη λύση είναι μόλις το αυτοκίνητο ακουμπά στον τοίχο αυτός να εξαφανίζεται. Άρα αφού ανοίξουμε τα attributes του wall βρίσκουμε το wall Hide από τα miscellaneous attributes του wall και το σύρουμε μέσα στο script του Car.



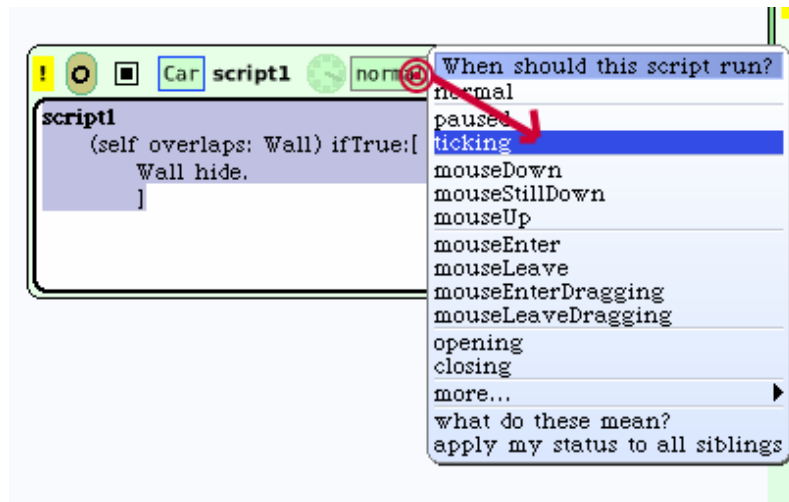
**Εικόνα 87: διασυνδέοντας το Car με το Wall**

Τώρα θα πρέπει να προσθέσουμε τον κώδικα που πραγματοποιεί τον έλεγχο της σύγκρουσης. Ανοίγουμε την επεξεργασία κώδικα και προσθέτουμε το παρακάτω μέσα στον κώδικα ώστε ο κώδικας να γίνει:

```
script1
```

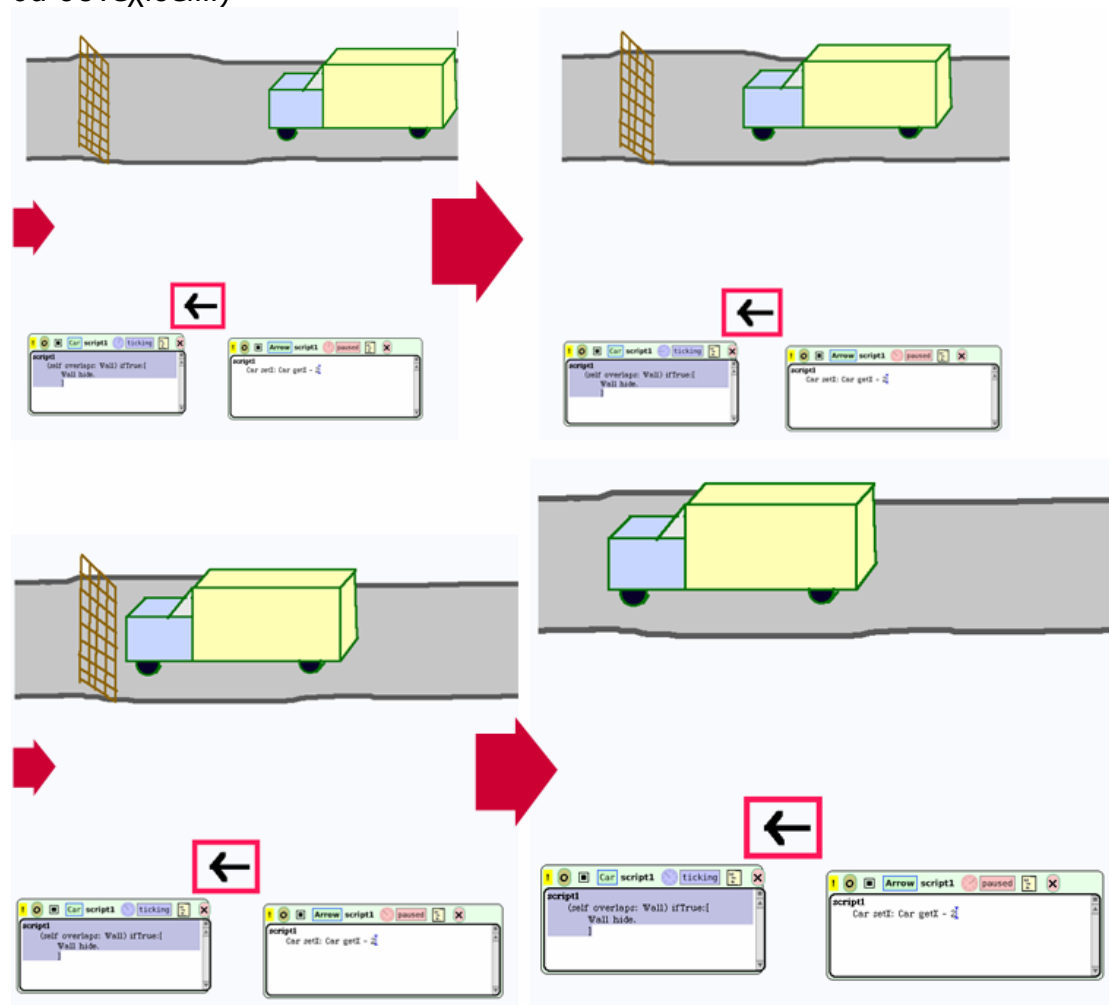
```
(self overlaps: Wall) ifTrue:[
    Wall hide.
]
```

Ο κώδικας αυτός πρέπει να είναι σε θέση να καταλάβει ανά πάσα στιγμή την κατάσταση μεταξύ φορτηγού και τοίχου. Όμως έχουμε μάθει πως ένα script εκτελείται είτε αν φτιάξουμε ένα κουμπί είτε αν του δώσουμε εντολή εκτέλεσης. Σε αυτή την περίπτωση, θέλουμε ένα βρόγχο που θα ελέγχει κάθε στιγμή την κατάσταση αυτών των αντικειμένων. Άρα θέλουμε το script να τρέχει ασταμάτητα χωρίς την παρέμβασή μας. Για να επιτευχθεί αυτό πρέπει να το βάλουμε σε λειτουργία ticking.



**Εικόνα 88: βάζοντας το script σε loop**

Πατούμε με αριστερό κλικ το Normal και επιλέγουμε το ticking. Με αυτό τον τρόπο όπως φαίνεται και από το ρολόι αριστερά του normal το script ξεκινά να εκτελείται χωρίς την παρέμβασή μας. Τώρα είμαστε έτοιμοι να δούμε τη σύγκρουση. (Το φορτηγό ως βαρύτερο θα σπάσει τον τοίχο και θα συνεχίσει...)



**Εικόνα 89: το φορτηγό συγκρούεται με τον τοίχο και τον καταστρέφει**

Σημείωση

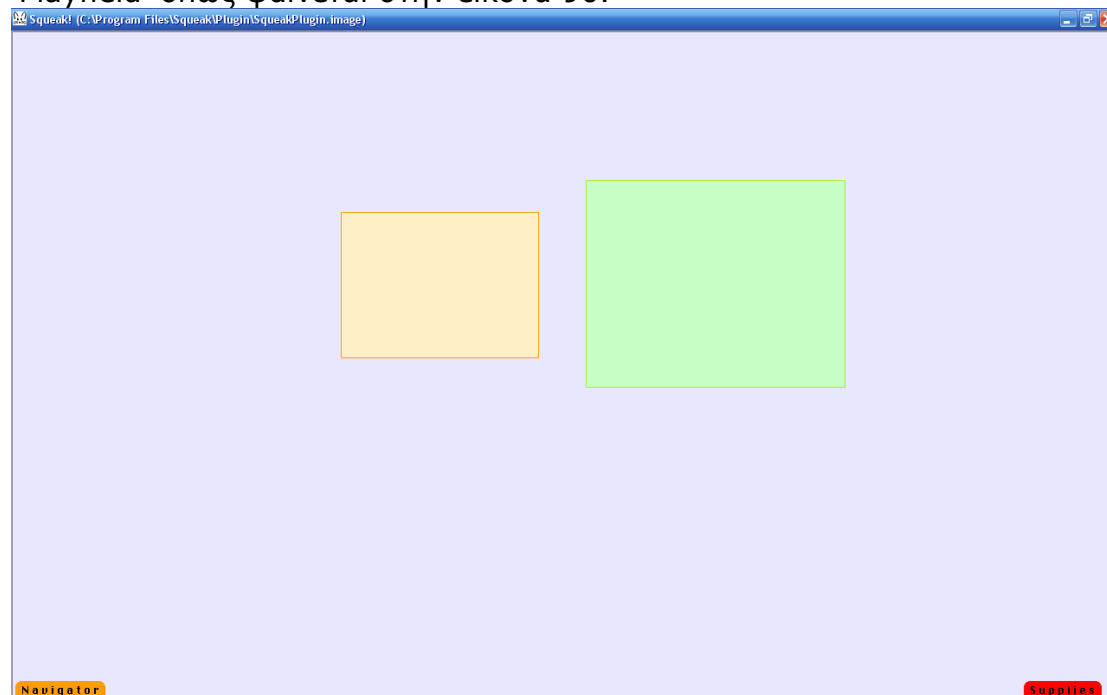
Αν εξαφανιστεί το αντικείμενο τοίχος είναι δυνατόν να το εμφανίσουμε πάλι εφόσον έχουμε κρατήσει ανοιχτά τα attributes του. Πηγαίνουμε στα miscellaneous και επιλέγουμε εκτέλεση του Wall Show.

Αν έχουμε κλείσει αυτό το παράθυρο τότε πατούμε δεξί κλικ στον κόσμο του squeak και επιλέγουμε Project -> View Project hierarchy. Εμφανίζονται όλα τα αντικείμενα του κόσμου και πατώντας στο Wall θα δούμε τις επιλογές του αντικειμένου. Από εκεί ανοίγουμε τα attributes και επιλέγουμε να εκτελέσουμε το attribute Wall Show

### 4.8. Κινούμενα σχέδια

Στο συγκεκριμένο σημείο θα φτιάξουμε ένα κινούμενο σχέδιο. Για να το πετύχουμε αυτό, θα σχεδιάσουμε μια εικόνα (ένα σχέδιο) για κάθε frame, αλλάζοντας λίγο την εικόνα για κάθε frame ξεχωριστά. Όταν θα τρέχει η καινούργια μας ταινία, θα έχουμε την ψευδαίσθηση ότι αυτό που φτιάξαμε είναι αληθινό.

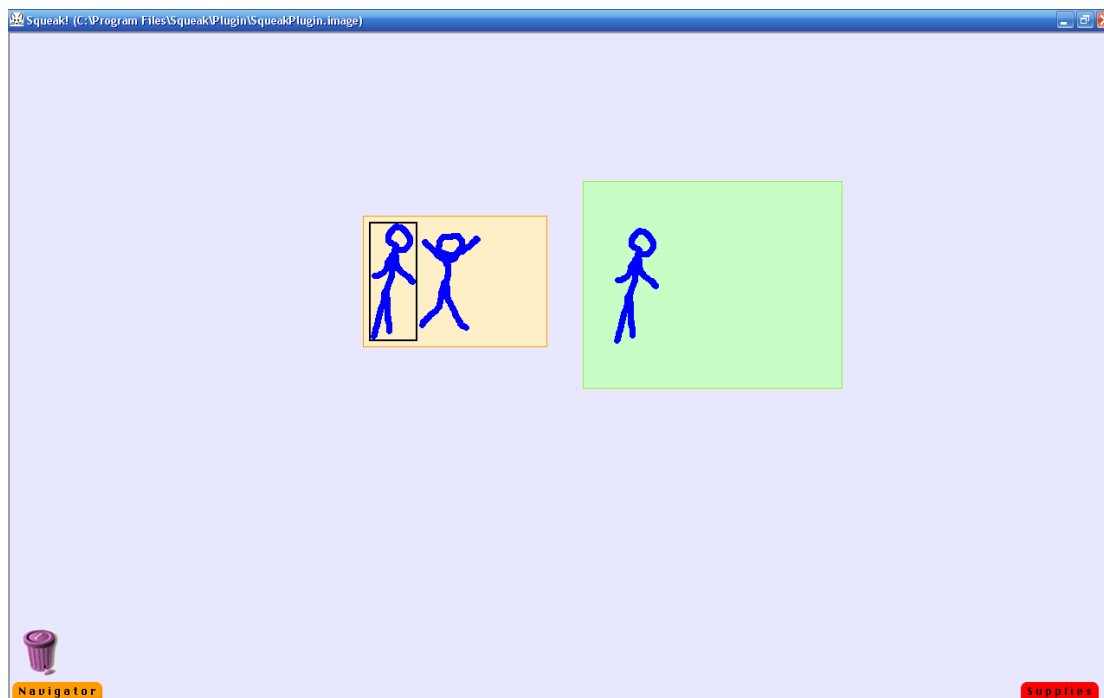
Το παραπάνω μπορούμε να το φτιάξουμε στο Squeak. Ξεκινάμε το πρόγραμμα και ανοίγουμε την καρτέρα 'Supplies'. Στη συνέχεια τραβάμε στο 'Squeak Desktop' δύο αντικείμενα, τα αντικείμενα 'Holder' και 'Playfield' όπως φαίνεται στην εικόνα 90.



**Εικόνα 90: Ο κόσμος με δύο αντικείμενα: Holder και Playfield**

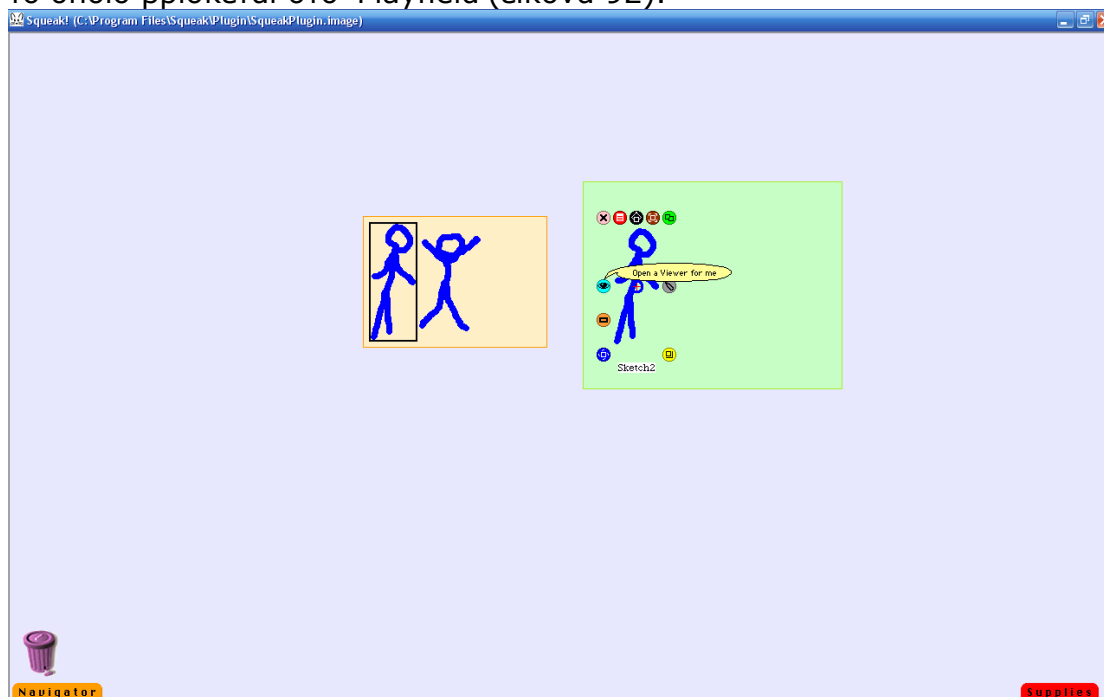
Στο αντικείμενο 'Holder' θα έχουμε τα frames της ταινίας μας και το αντικείμενο 'Playfield' θα είναι η 'οθόνη' όπου θα φαίνεται το κινούμενο σχέδιό μας.

Πρώτα από όλα, θα φτιάξουμε ένα νέο σχέδιο(αντικείμενο ζωγραφικής) για κάθε frame και θα το βάλουμε στο αντικείμενο 'Holder'. Στη συνέχεια θα αντιγράψουμε το πρώτο σχέδιο και θα το βάλουμε στο 'Playfield'. Το αποτέλεσμα φαίνεται στην εικόνα 91.



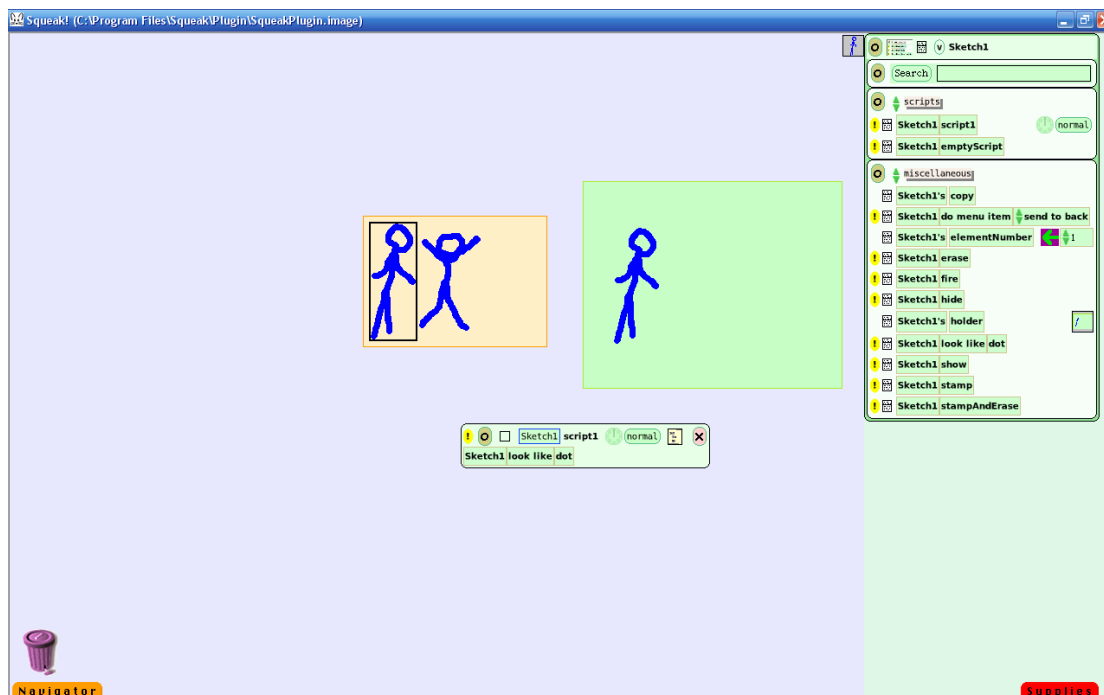
**Εικόνα 91: Τα δύο αντικείμενα στο Holder και το αντιγραμμένο αντικείμενο στο playfield**

Το επόμενο μας βήμα είναι να ανοίξουμε τον 'Viewer' του πρώτου σχεδίου, το οποίο βρίσκεται στο 'Playfield'(εικόνα 92).



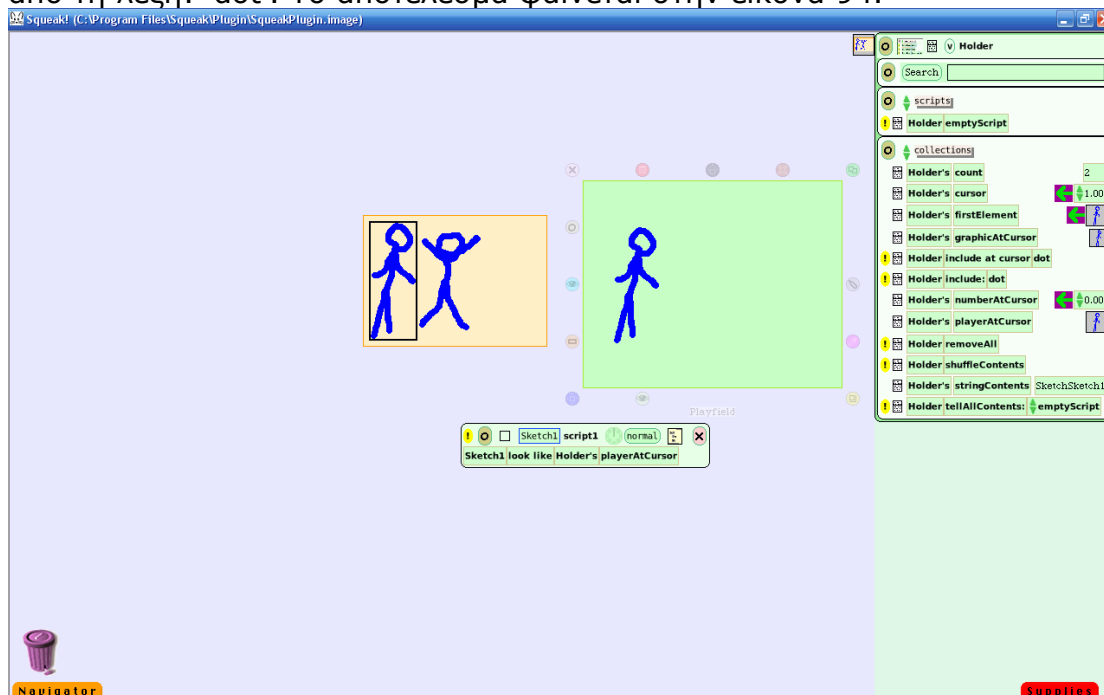
**Εικόνα 92: Άνοιγμα επιλογών του αντικειμένου στο playfield**

Αφού του δώσουμε το όνομα 'Sketch1', αλλάζουμε στον 'Viewer' κατηγορία και από 'basic' την κάνουμε 'miscellaneous'. Στη κατηγορία αυτή τραβάμε την γραμμή που λέει 'look like:dot' για να δημιουργήσουμε ένα νέο script. Το αποτέλεσμα φαίνεται στην εικόνα 93.



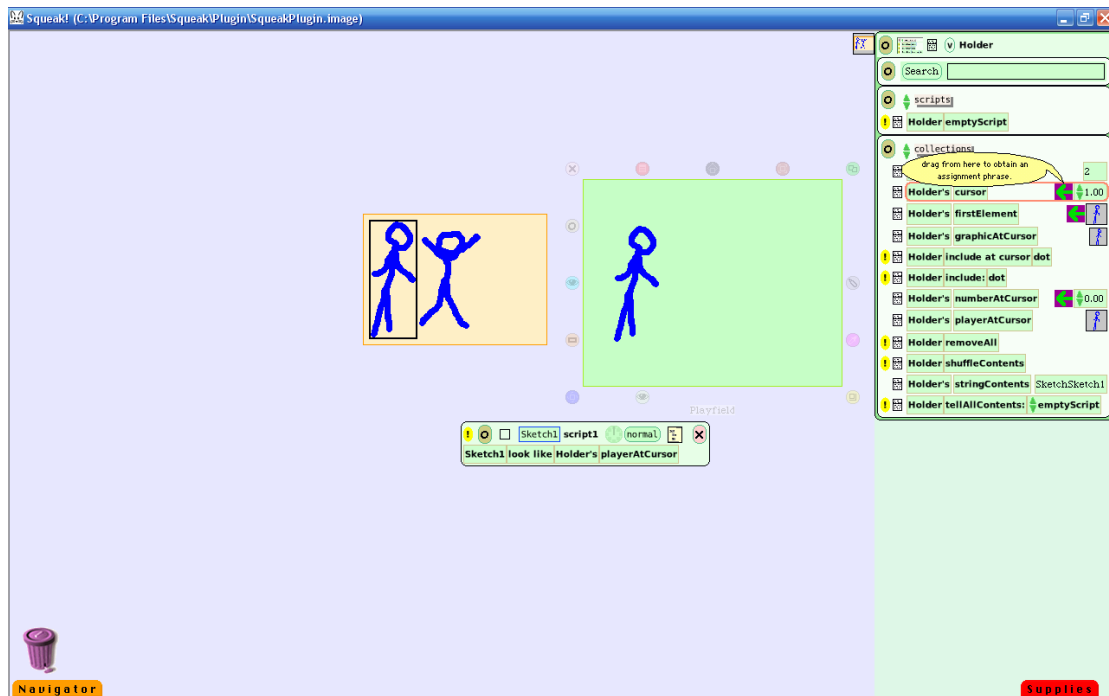
**Εικόνα 93: Δημιουργία νέου script για τα κινούμενα σχέδια**

Στη συνέχεια ανοίγουμε τον 'Viewer' του αντικειμένου 'Holder' και τραβάμε από τον κατάλογο 'collections' τη γραμμή 'Holder's PlayerAtCursor' πάνω από τη λέξη: 'dot'. Το αποτέλεσμα φαίνεται στην εικόνα 94.



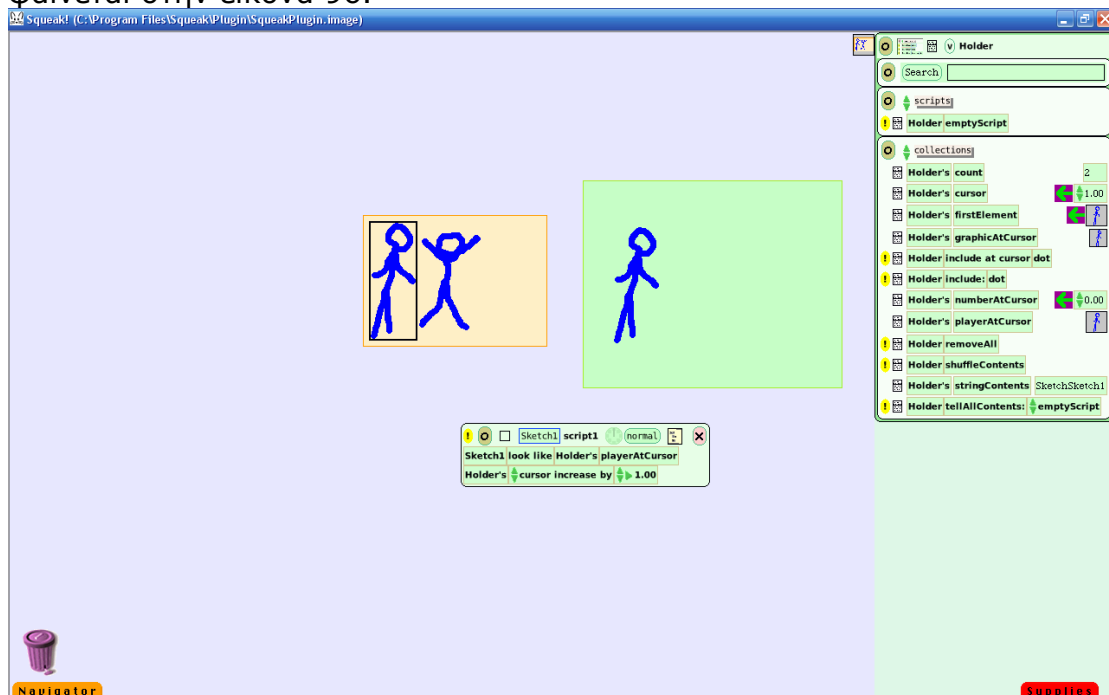
**Εικόνα 94: Βάζοντας το player at cursor μέσα στο script**

Το επόμενο που πρέπει να κάνουμε είναι να προσθέσουμε μια δεύτερη γραμμή στο script. Στη γραμμή αυτή θα τραβήξουμε από τον 'Viewer' τη γραμμή 'Holder's cursor: 1.00' όπως φαίνεται στην επόμενη εικόνα.



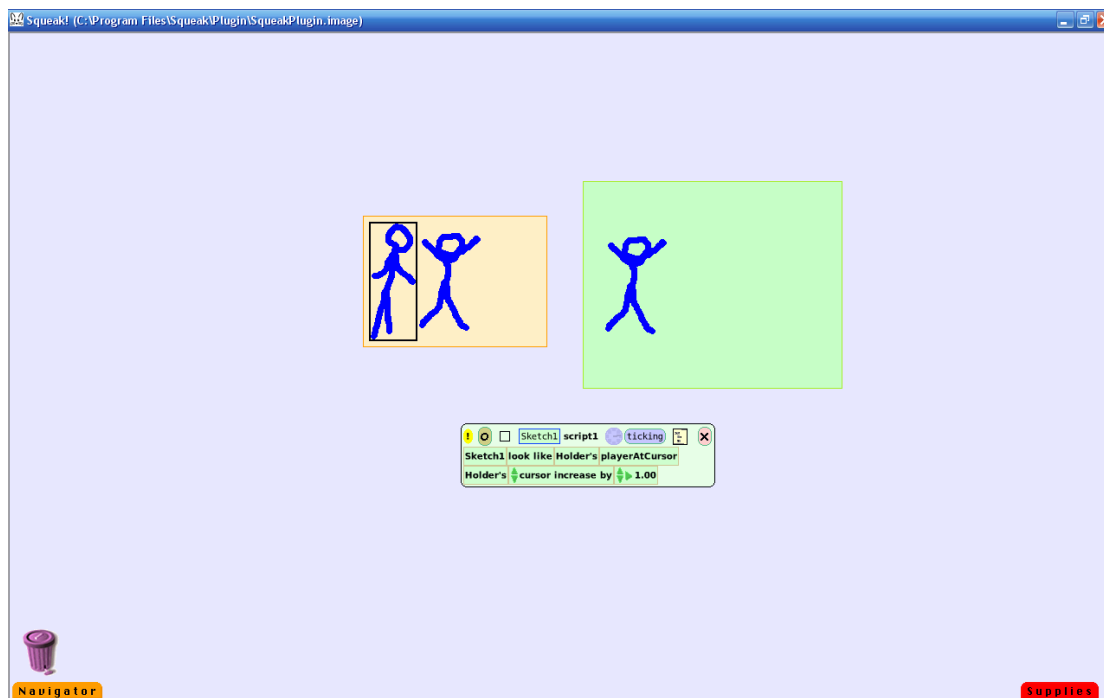
**Εικόνα 95: Επιλέγοντας το attribute cursor**

Στη γραμμή αυτή πατώντας το πρώτο μικρό πράσινο βελάκι αλλάζουμε τη μέθοδο και σταματάμε εκεί που λέει 'cursor increase by: 1'. Το αποτέλεσμα φαίνεται στην εικόνα 96.



**Εικόνα 96: Προσθέτοντας το attribute cursor increase by μέσα στο script**

Τέλος για να τρέξει το script, αλλάζουμε στο script εκεί που λέει 'normal' το αλλάζουμε σε 'ticking' (εικόνα 97). Έτσι φτιάξαμε το πρώτο μας κινούμενο σχέδιο.



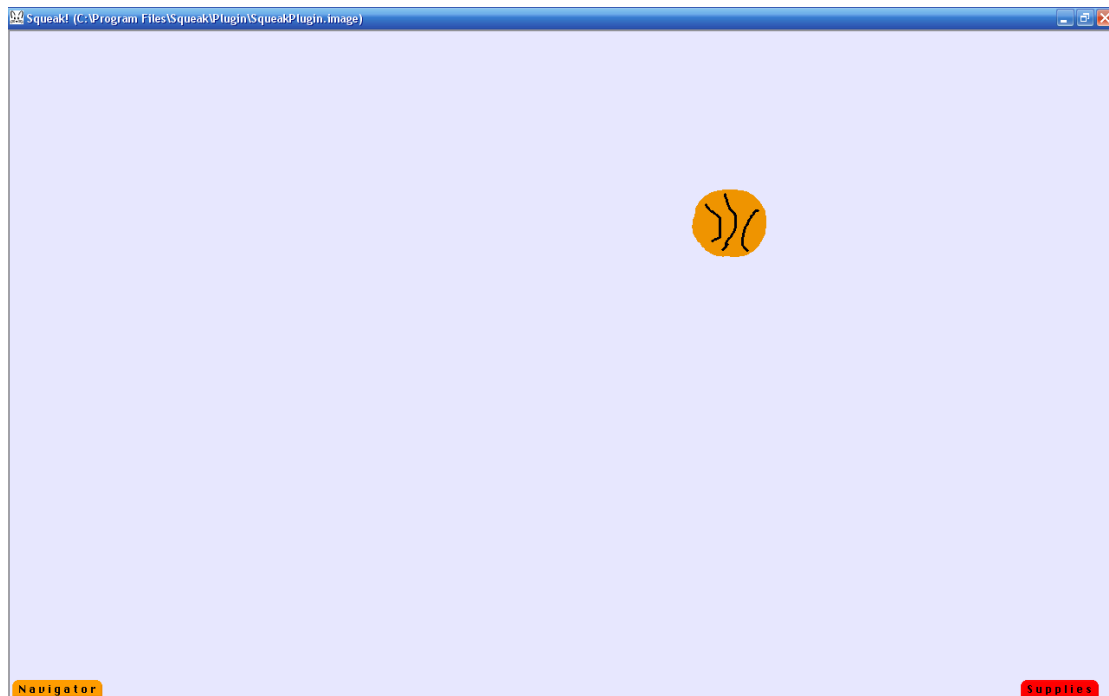
**Εικόνα 97: Το κινούμενο σχέδιο εν δράσει...**

#### 4.8.1. Ζωγραφίζοντας την κίνηση μιας μπάλας μπάσκετ

Στο παράδειγμα αυτό θα φτιάξουμε μια μπάλα μπάσκετ και θα την βάλουμε σε κίνηση. Αφού ξεκινήσουμε το πρόγραμμα, ανοίγουμε την παλέτα ζωγραφικής από την καρτέλα 'Navigator' πατώντας το εικονίδιο με το πιναλάκι (εικόνα 98). Έτσι φτιάξαμε την μπάλα μπάσκετ που φαίνεται στην εικόνα 99.

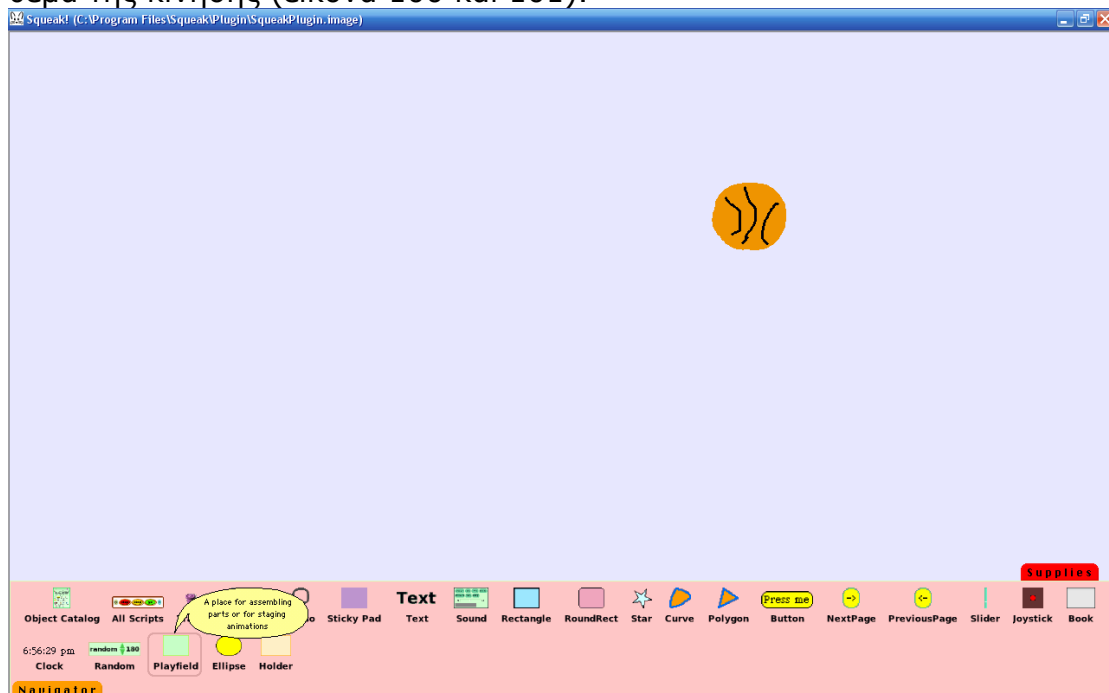


**Εικόνα 98: Επιλέγοντας τα εργαλεία ζωγραφικής από το Squeak Etoys**

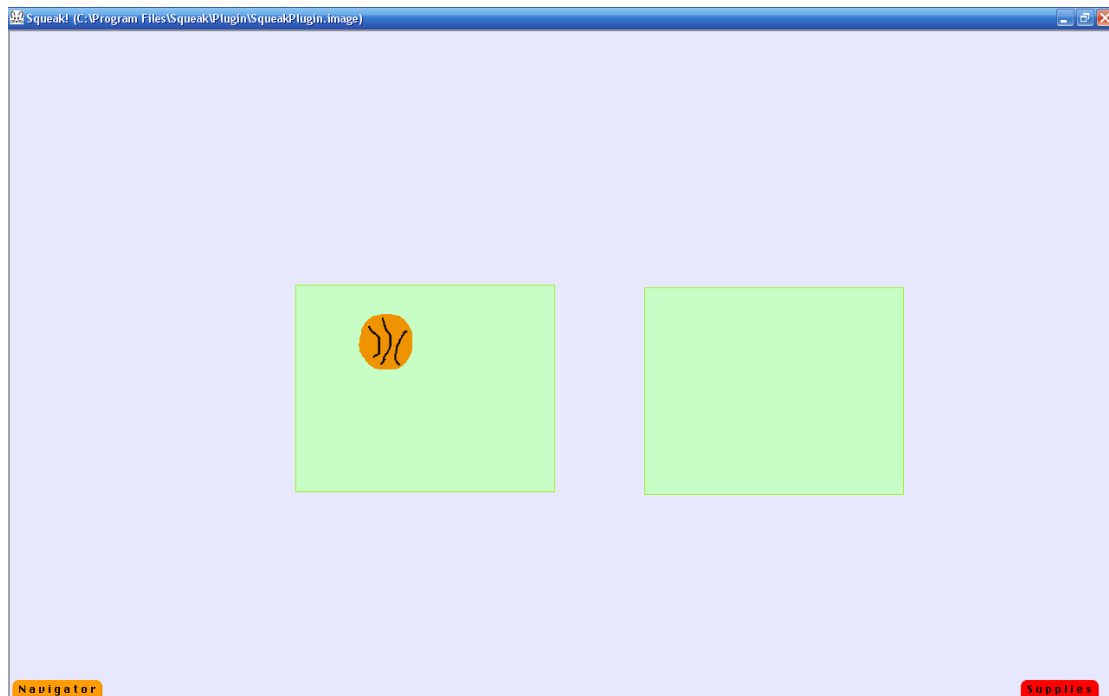


**Εικόνα 99: Σχεδιασμός μίας μπάλας μπάσκετ**

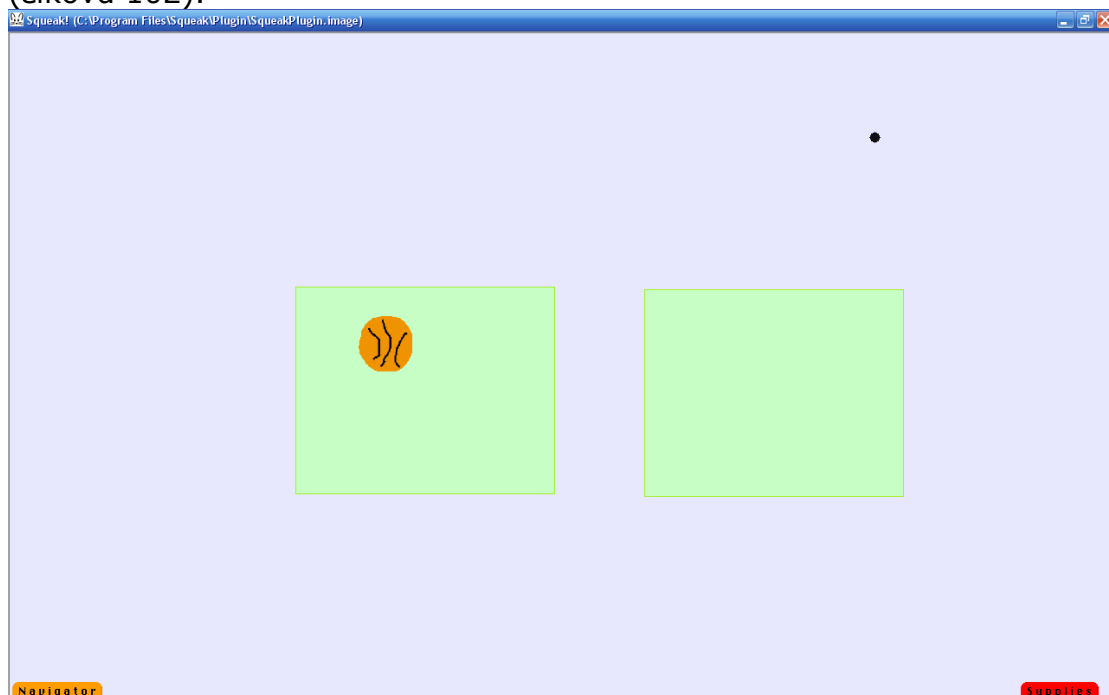
Το αντικείμενο που φτιάξαμε θα το ονομάσουμε 'basketball'. Στη συνέχεια θα τραβήξουμε στο 'Squeak Desktop' δύο αντικείμενα 'Playfield' από την καρτέλα 'Supplies', τα οποία θα μας βοηθήσουν για το θέμα της κίνησης (εικόνα 100 και 101).



**Εικόνα 100: Επιλέγοντας τα αντικείμενα playfield από το supplies**

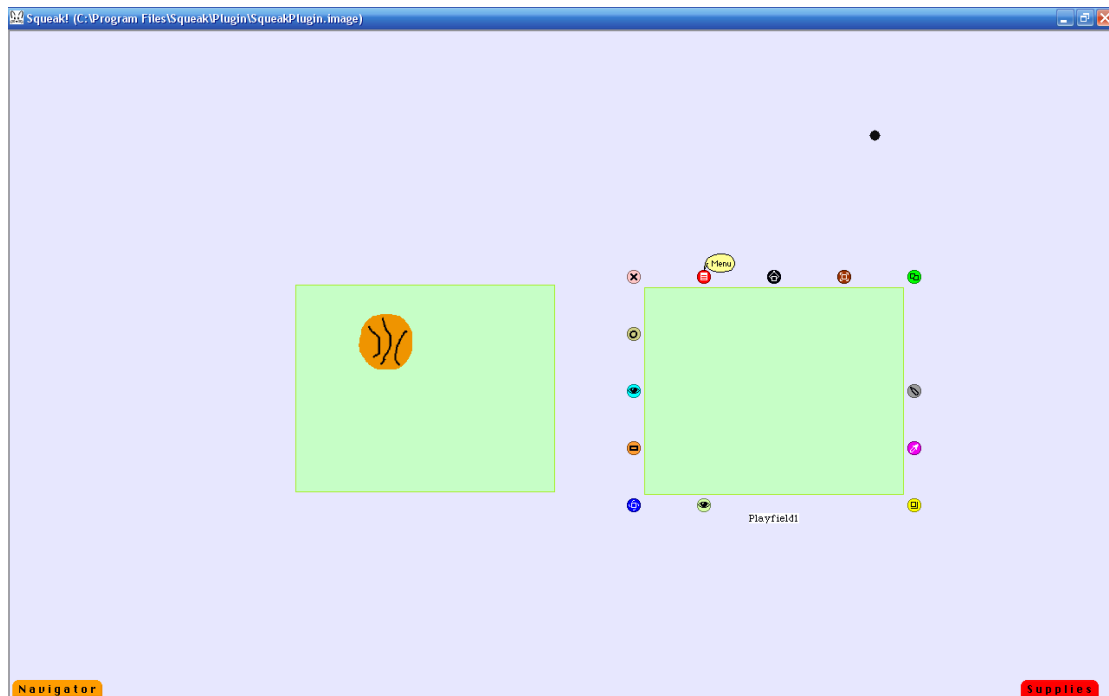


**Εικόνα 101: Τοποθετώντας τη μπάλα στο ένα αντικείμενο playfield**  
Στη συνέχεια φτιάχνουμε μια απλή μαύρη τελεία [μέσα από τη ζωγραφική] (εικόνα 102).

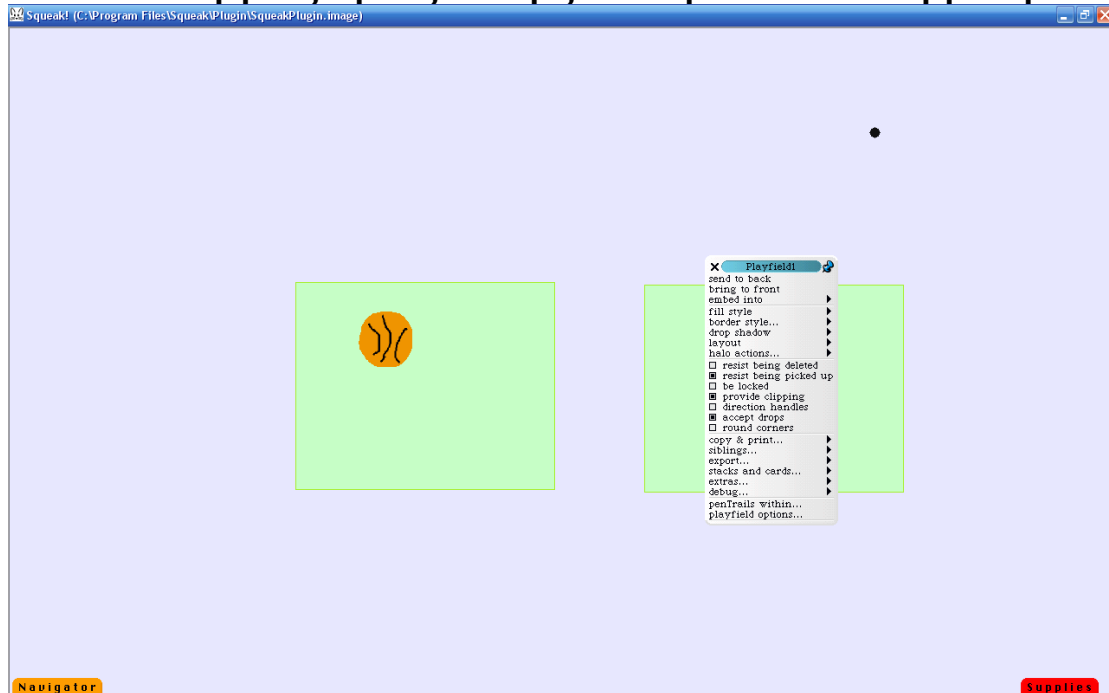


**Εικόνα 102: Δημιουργία νέου αντικειμένου ζωγραφικής: δημιουργία τελείας**

Αφού σχεδιάσαμε την μαύρη τελεία πηγαίνουμε στο αντικείμενο 'Playfield' που δεν έχει την μπάλα και πατάμε το εργαλείο 'Menu'(εικόνα 103). Το αποτέλεσμα φαίνεται στην εικόνα 104.

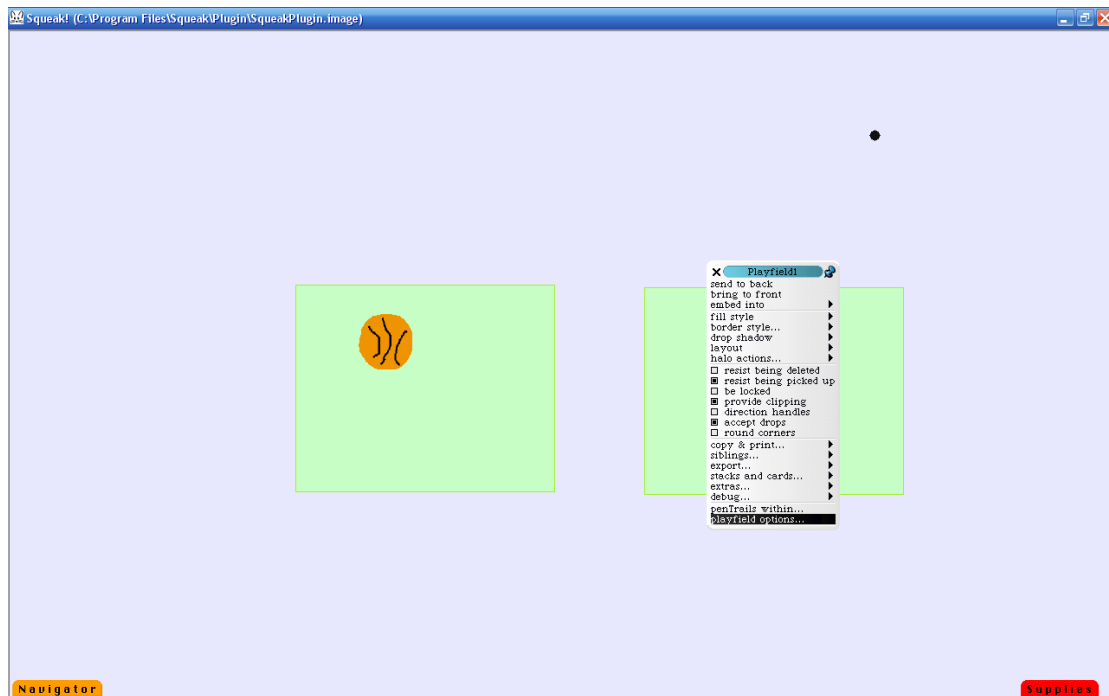


**Εικόνα 103: Εμφανίζουμε τις επιλογές αντικειμένου και επιλογή του μενού**

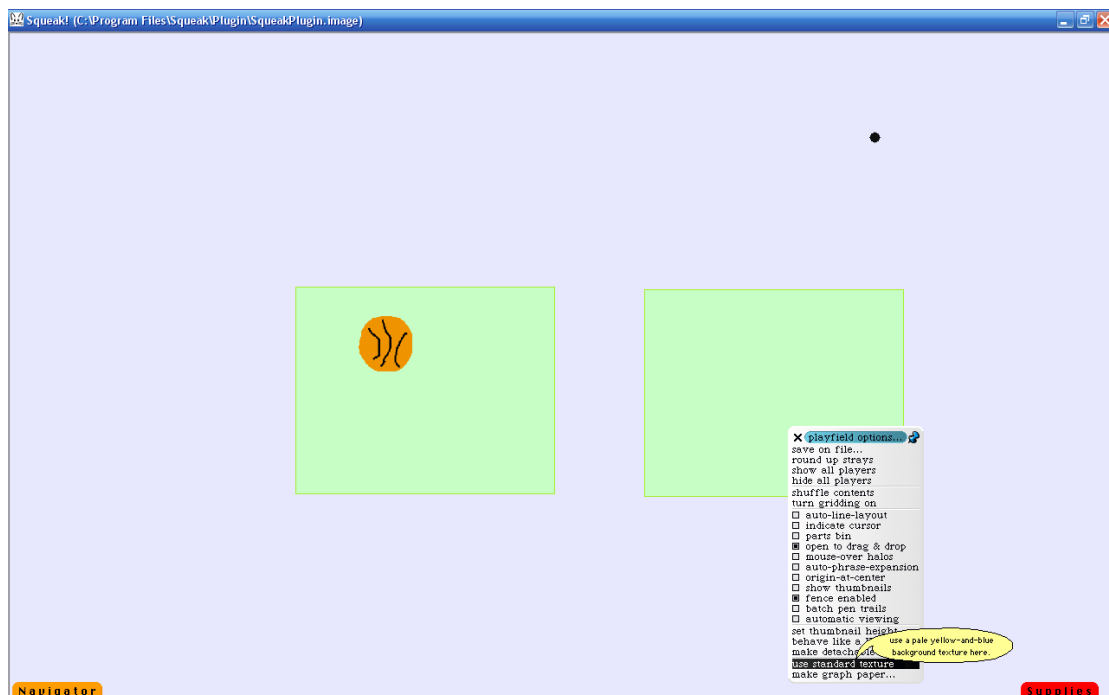


**Εικόνα 104: Το μενού του αντικειμένου playfield**

Εκεί επιλέγουμε το 'playfield options' (εικόνα 105), και από το καινούργιο μενού που προκύπτει επιλέγουμε 'use standard texture' (εικόνα 106).

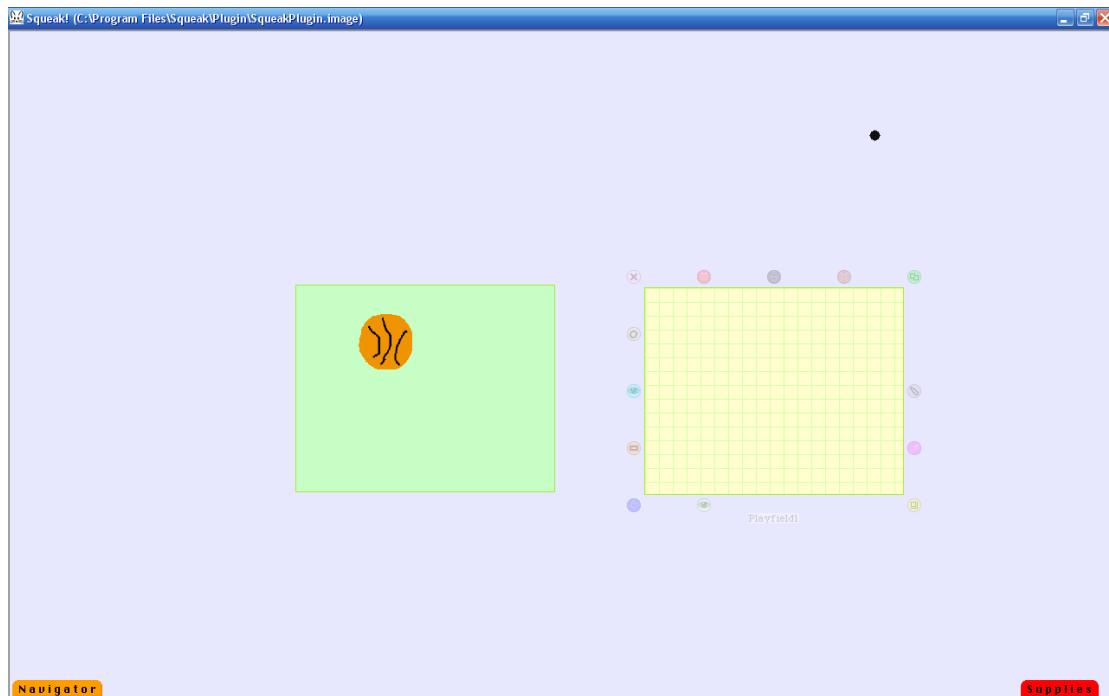


**Εικόνα 105: Επιλογή του playfield options**

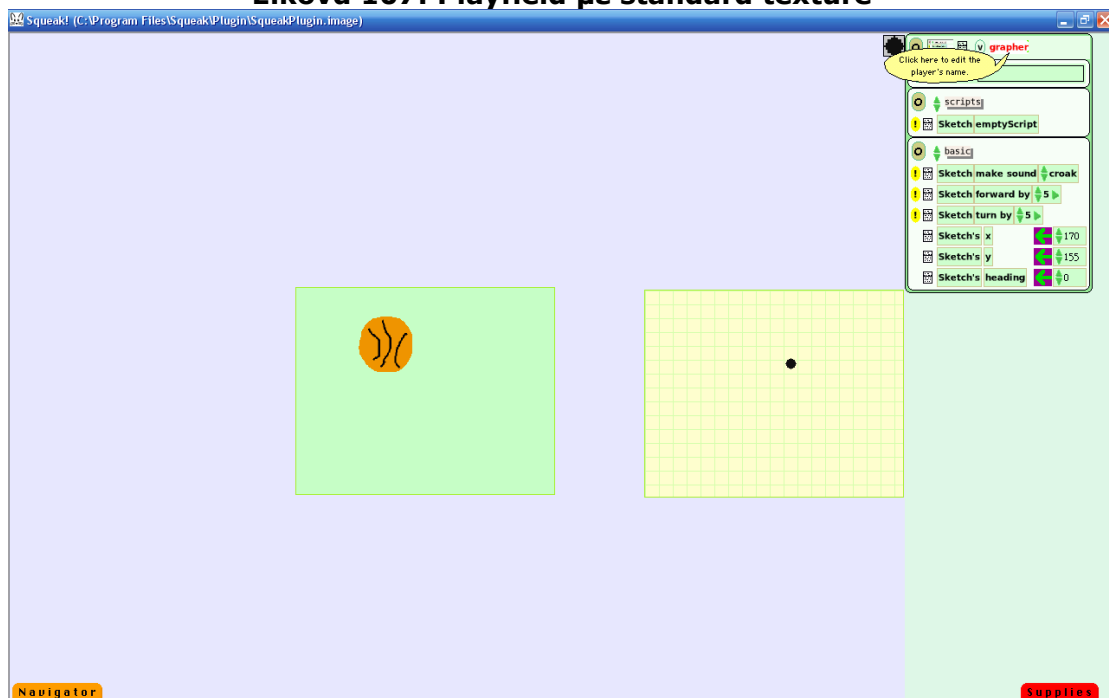


**Εικόνα 106: Επιλογή του Use Standard Texture**

Το αποτέλεσμα φαίνεται στην εικόνα 107. Κατόπιν παίρνουμε την μαύρη τελεία και την βάζουμε στο κίτρινο 'Playfield' ενώ παράλληλα της δίνουμε το όνομα 'grapher' (εικόνα 108).

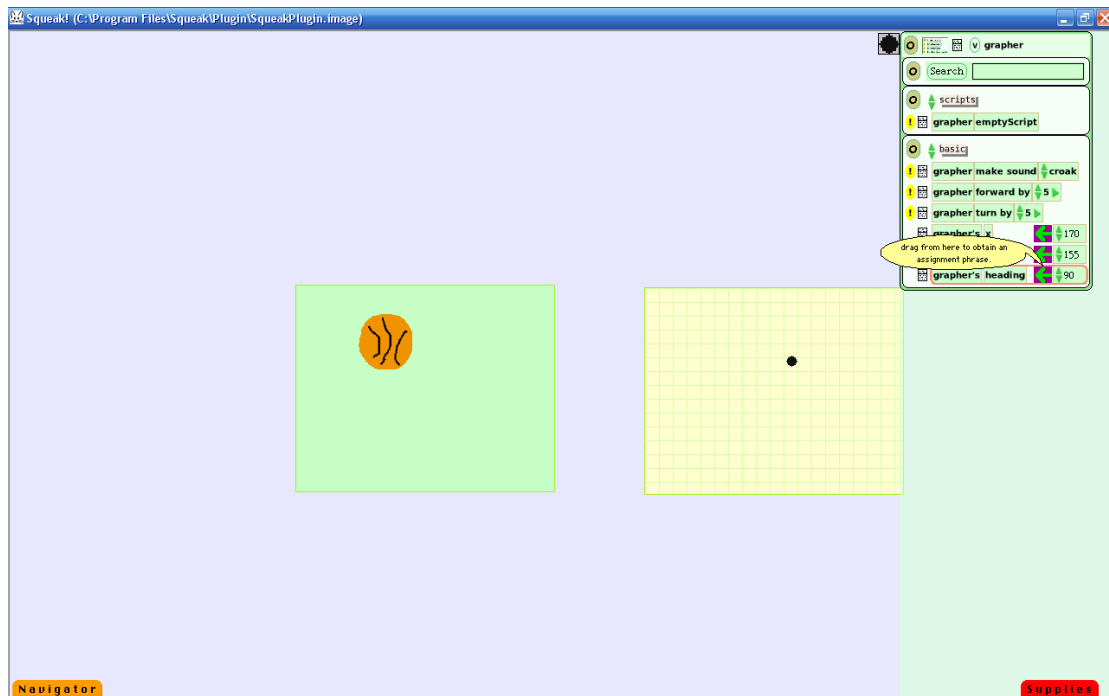


**Εικόνα 107: Playfield με standard texture**



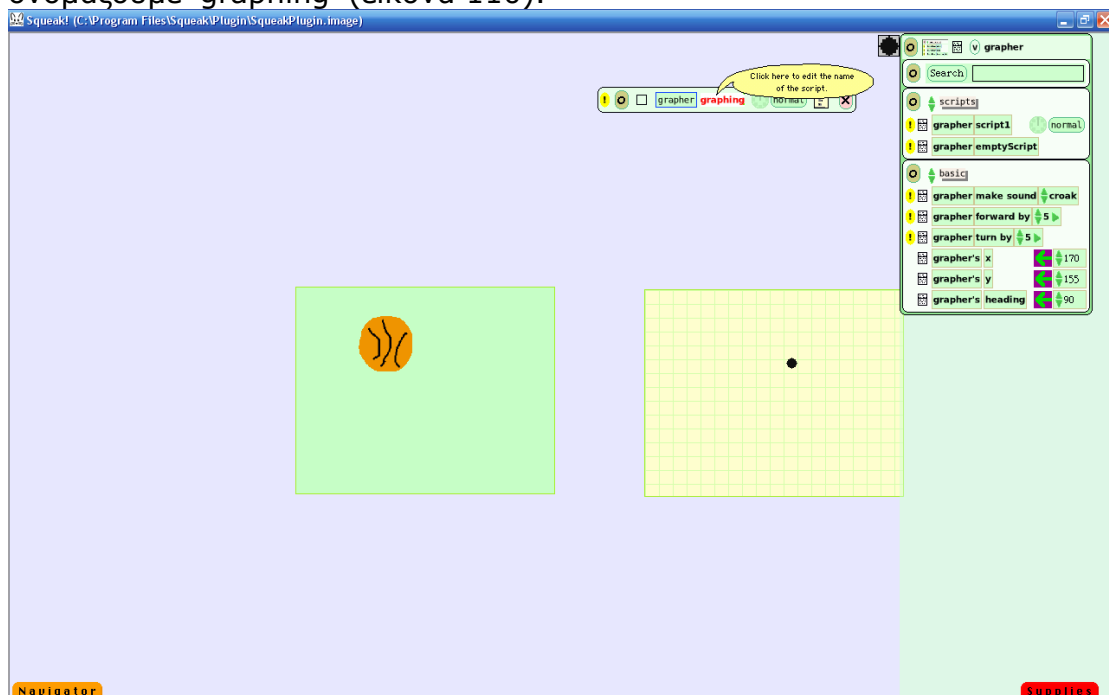
**Εικόνα 108: Τοποθέτηση του αντικειμένου (τελεία) μέσα στο playfield και μετονομασία του αντικειμένου**

Στο εργαλείο 'Viewer' του αντικειμένου 'grapher' που είναι ήδη ανοιχτό, πάμε και βάζουμε τον αριθμό '90' εκεί που λέει 'grapher's heading' όπως φαίνεται στην εικόνα 109.



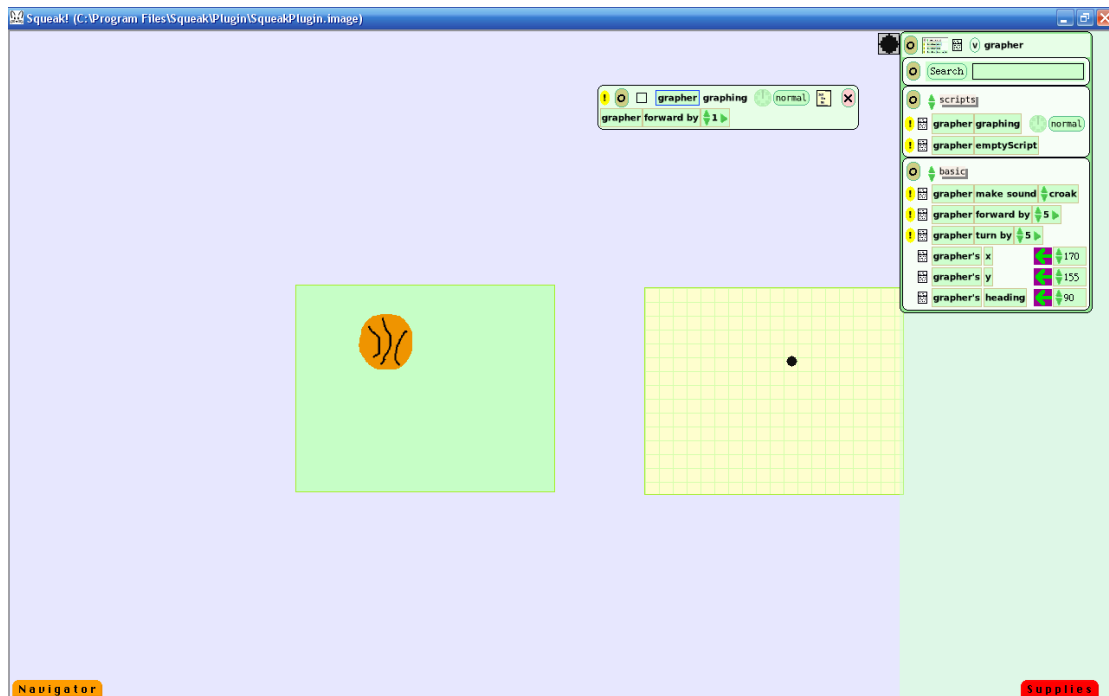
**Εικόνα 109: Αλλαγή του grapher's heading**

Επίσης πάμε και τραβάμε ένα 'emptyScript' στο 'Squeak Desktop', και το ονομάζουμε 'graphing' (εικόνα 110).

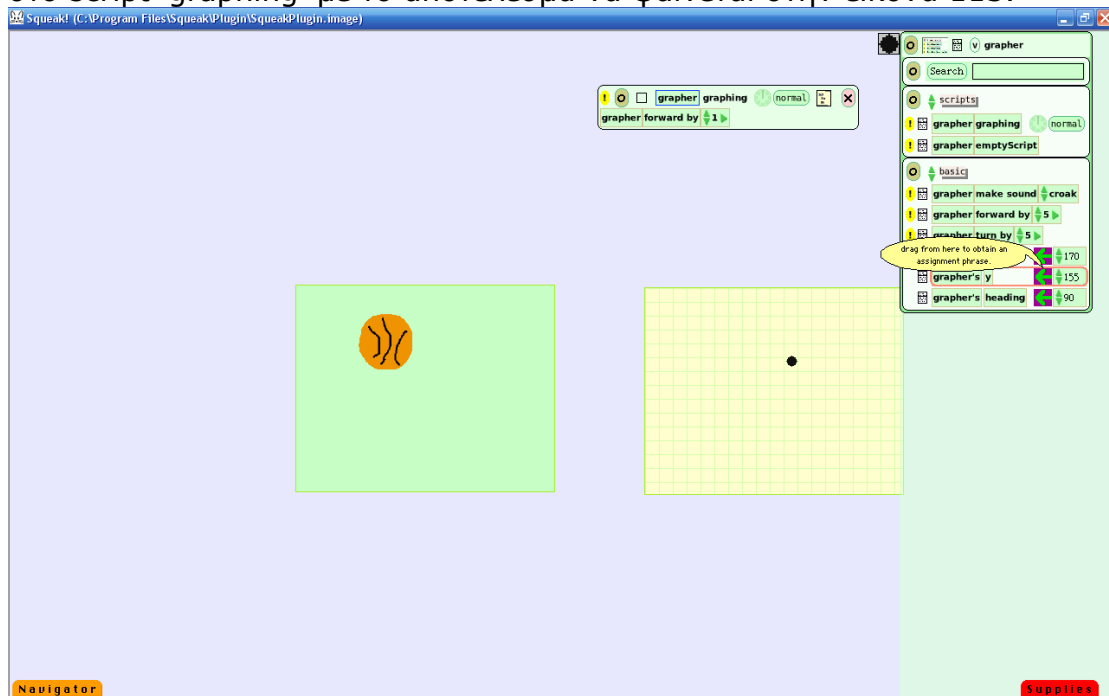


**Εικόνα 110: Δημιουργία script και μετονομασία του**

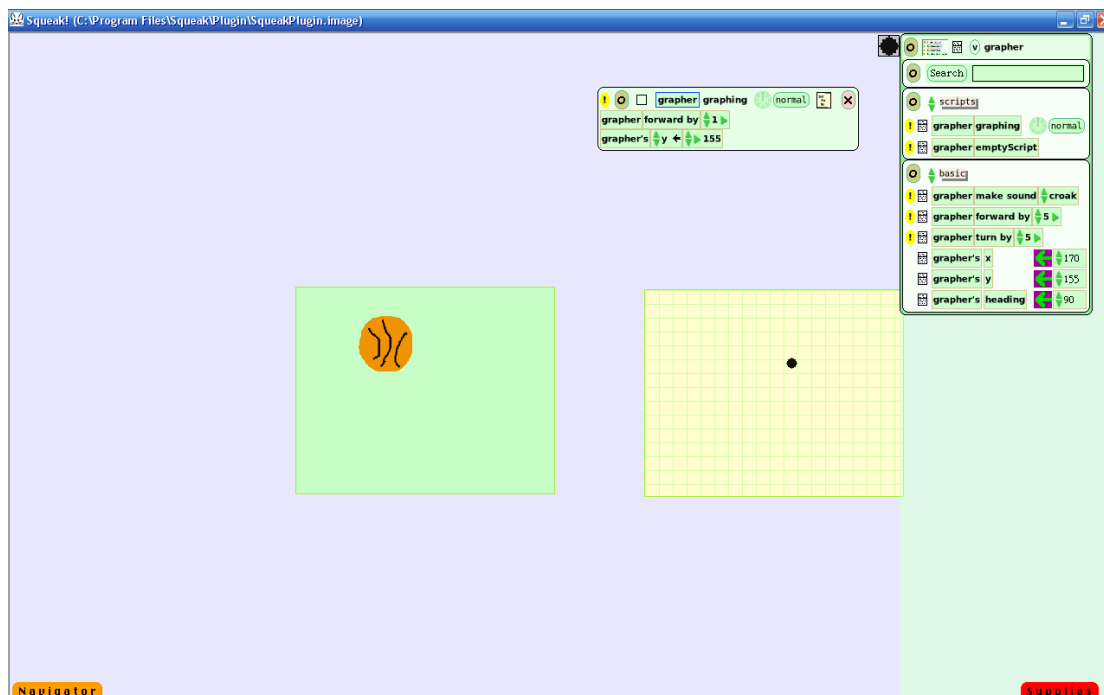
Πάμε μετά και τραβάμε ένα 'grapher forward by' οποίο το αφήνουμε μέσα στο script 'graphing', όπως φαίνεται στην εικόνα 111. Η τιμή που θα πάρει το 'grapher forward by' θα είναι '1'.



**Εικόνα 111: Τοποθετώντας ένα attribute forward by μέσα στο script**  
 Μετά τραβάμε ένα 'grapher's y <=' (εικόνα 112), και το αφήνουμε μέσα στο script 'graphing' με το αποτέλεσμα να φαίνεται στην εικόνα 113.

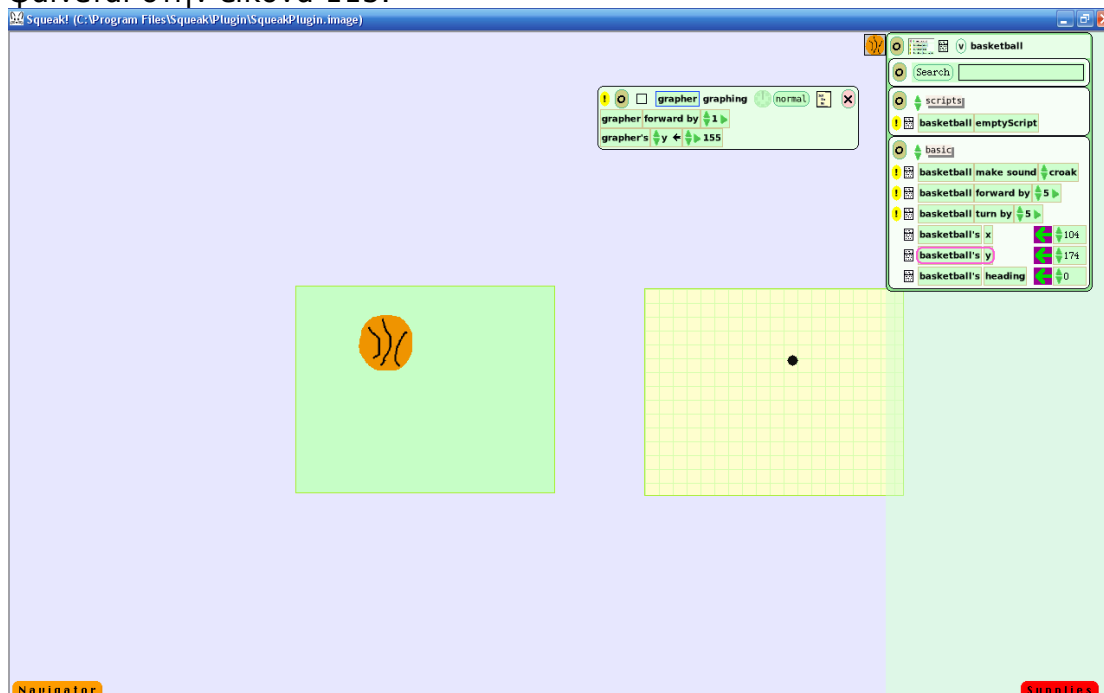


**Εικόνα 112: Επιλογή του grapher's y**

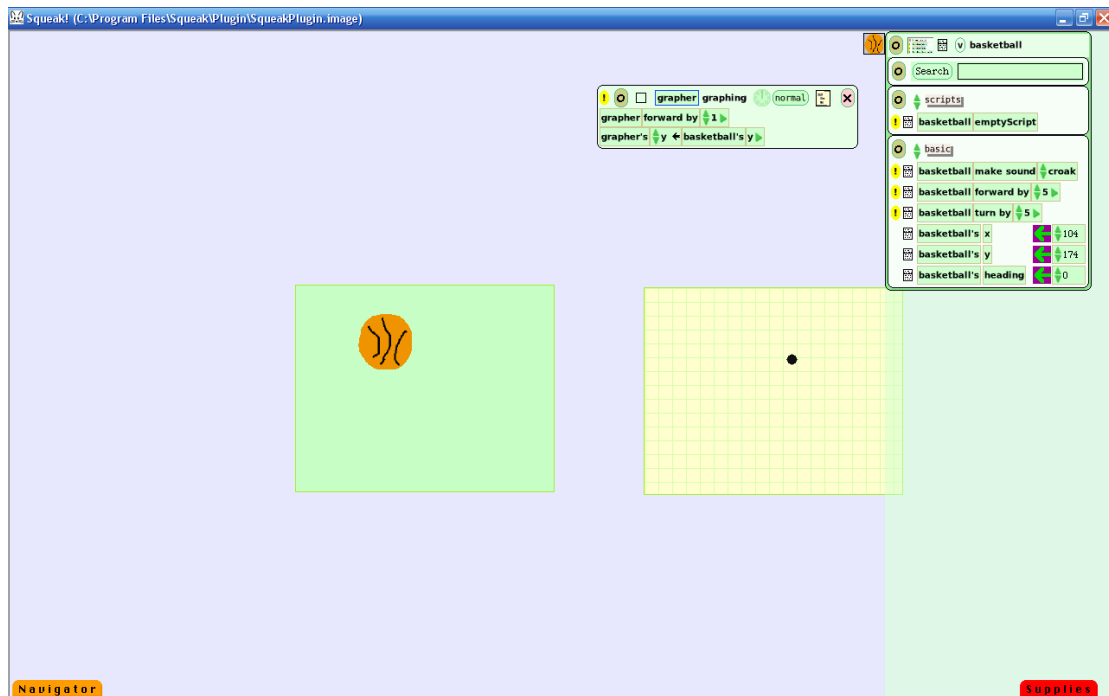


**Εικόνα 113: Τοποθέτηση του grapher's y μέσα στο script**

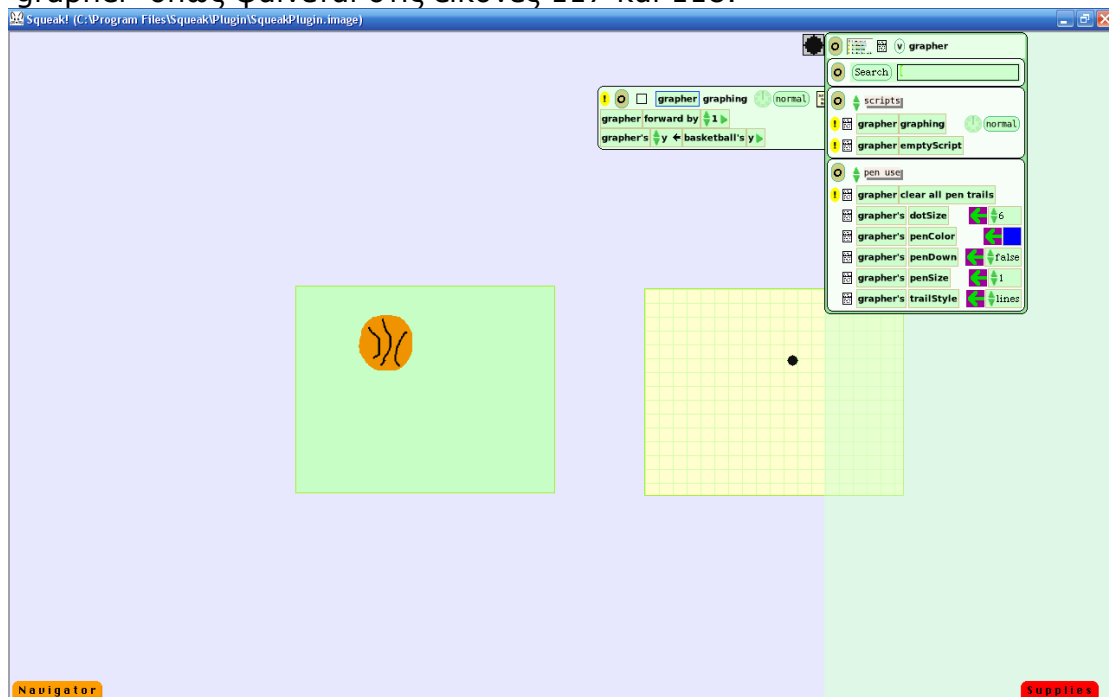
Κατόπιν πάμε στον 'Viewer' του αντικειμένου 'basketball' και τραβάμε ένα 'basketball's y' (εικόνα 114) και το αφήνουμε στην τιμή του 'grapher's y', που βρίσκεται στο script 'graphing'. Το αποτέλεσμα της πράξης αυτής φαίνεται στην εικόνα 115.



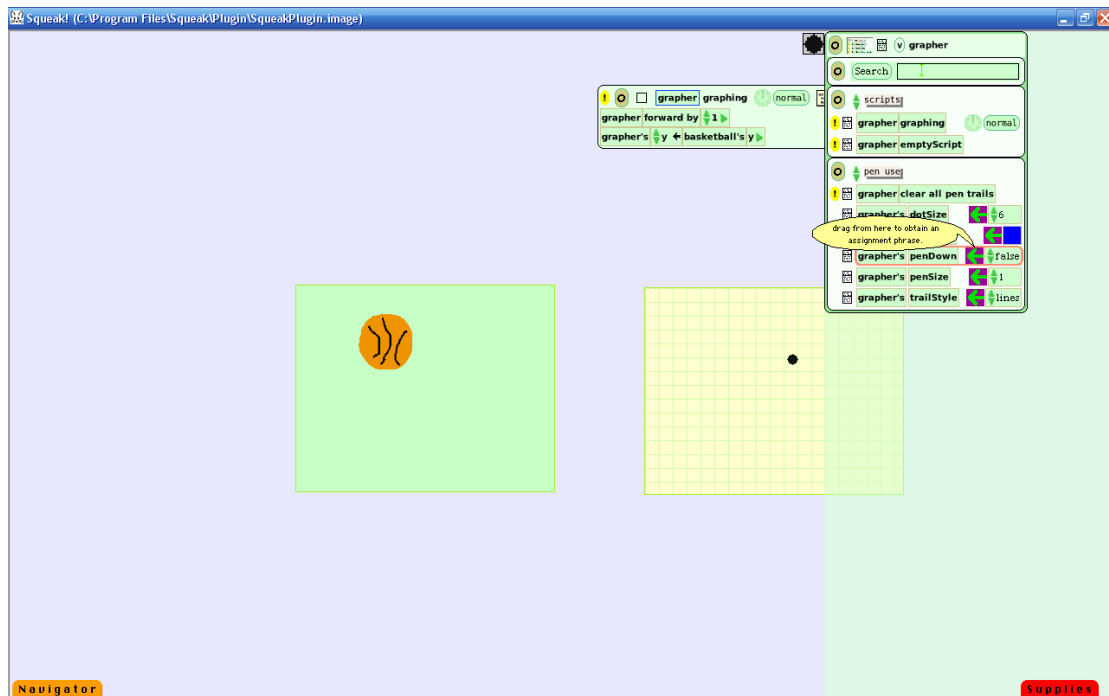
**Εικόνα 114: Επιλογή του basketball y**



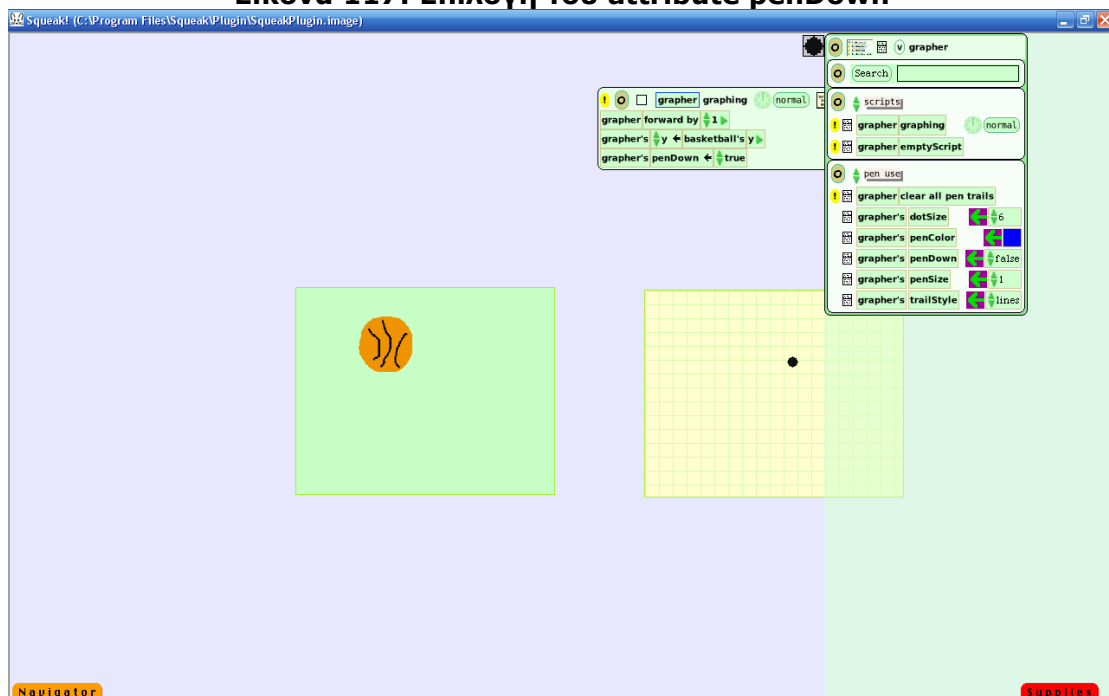
**Εικόνα 115: Τοποθέτηση του basketball y μέσα στην τιμή του grapher's y**  
 Με την τελευταία μας κίνηση συνδέσαμε τα αντικείμενα 'grapher' και 'basketball' μέσα από το script 'graphing'. Γυρίζουμε τώρα στον 'Viewer' του 'grapher' και αλλάζοντας κατηγορία από 'basic' σε 'pen use' (εικόνα 116) τραβάμε ένα 'grapher's penDown' και το αφήνουμε στο script 'grapher' όπως φαίνεται στις εικόνες 117 και 118.



**Εικόνα 116: Κατηγορία pen use από τα attributes του grapher**



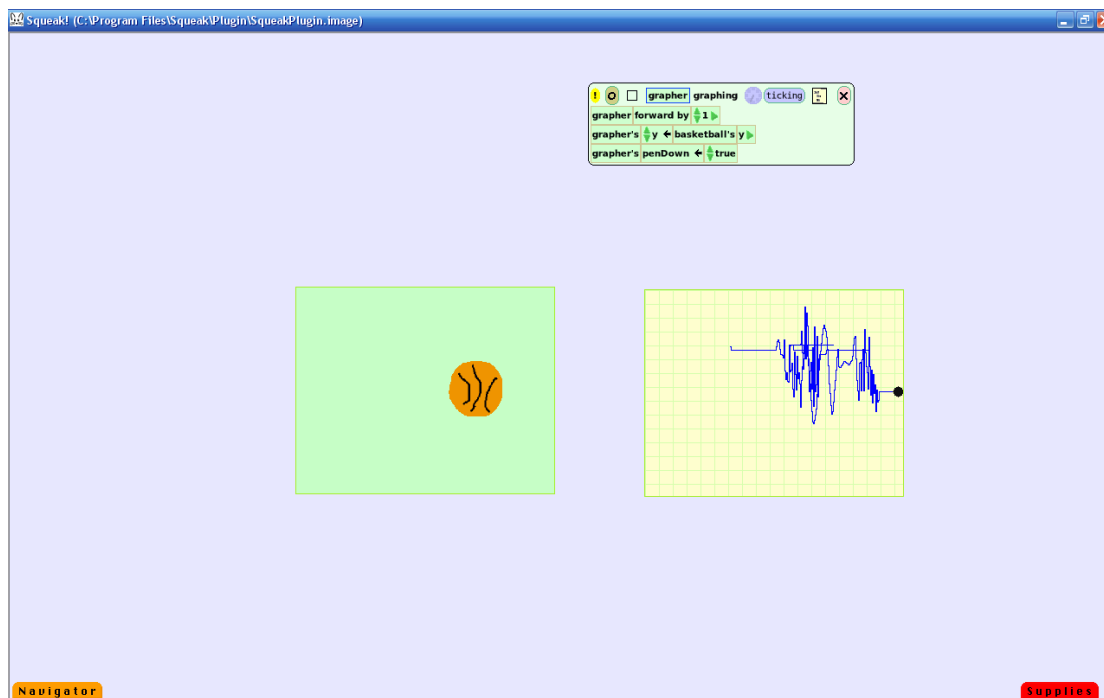
**Εικόνα 117: Επιλογή του attribute penDown**



**Εικόνα 118: Τοποθέτηση του attribute penDown μέσα στο script**

Πρέπει η τιμή του 'grapher's penDown' να είναι 'true'.

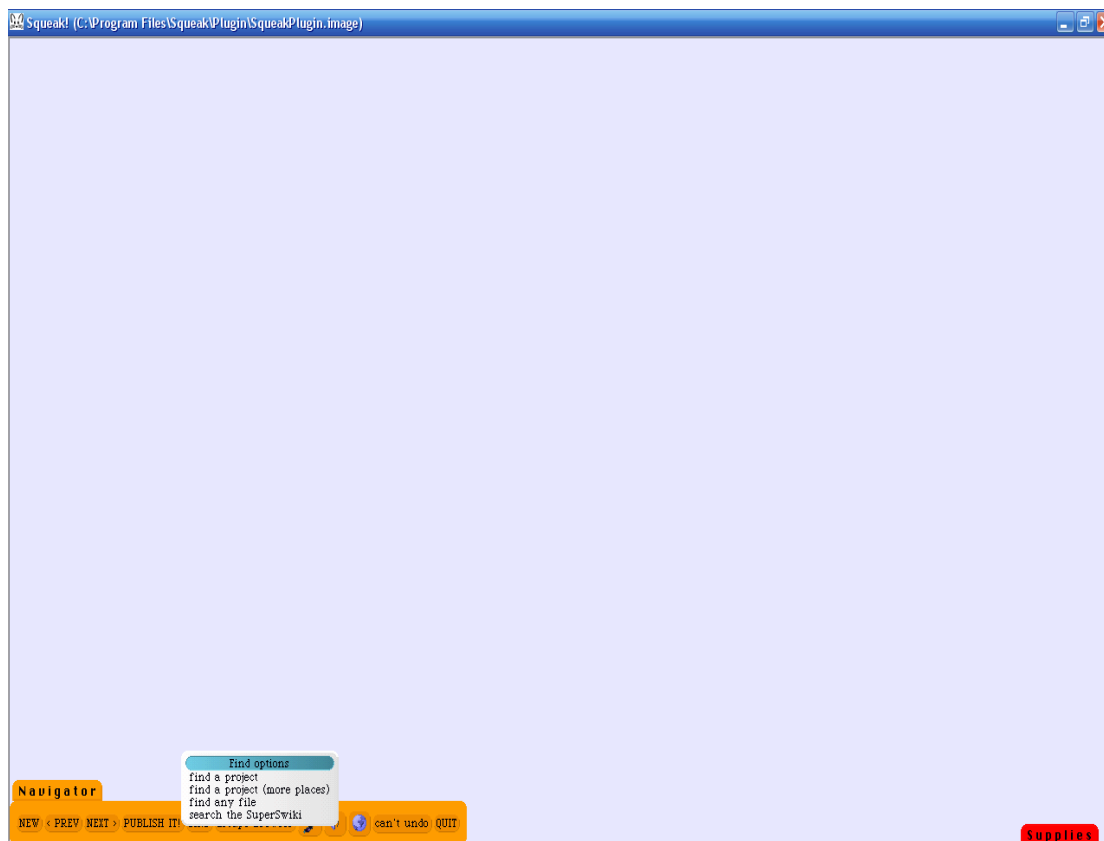
Τελειώνοντας, πατάμε το ρολόι που έχει το script 'grapher' ενώ παράλληλα αρχίζουμε και κουνάμε όπως θέλουμε την μπάλα. Βλέπουμε ότι με τον τρόπο αυτό σχεδιάζουμε τη κίνησή της (εικόνα 119) στον άξονα των 'y' καθώς περνάει ο χρόνος.



**Εικόνα 119: Αναπαράσταση της κίνησης της μπάλας**

#### **4.9. Εισαγωγή έτοιμων εικόνων στο πρόγραμμα**

Καμιά φορά η δυνατότητα ζωγραφικής που μας παρέχει το πρόγραμμα Squeak, δεν είναι αρκετή για τα πράγματα που θέλουμε να κάνουμε. Για το λόγο αυτό, μας δίνεται η δυνατότητα να εισάγουμε πολλούς διαφορετικούς τύπους εικόνων στο 'Squeak Desktop'. Ξεκινάμε το πρόγραμμα, ανοίγουμε την καρτέλα 'Navigator' και κρατάμε για μερικά δευτερόλεπτα πατημένο το αριστερό κλικ του ποντικιού πάνω στο κουμπί 'Find'. Τότε μας εμφανίζεται η εικόνα 120

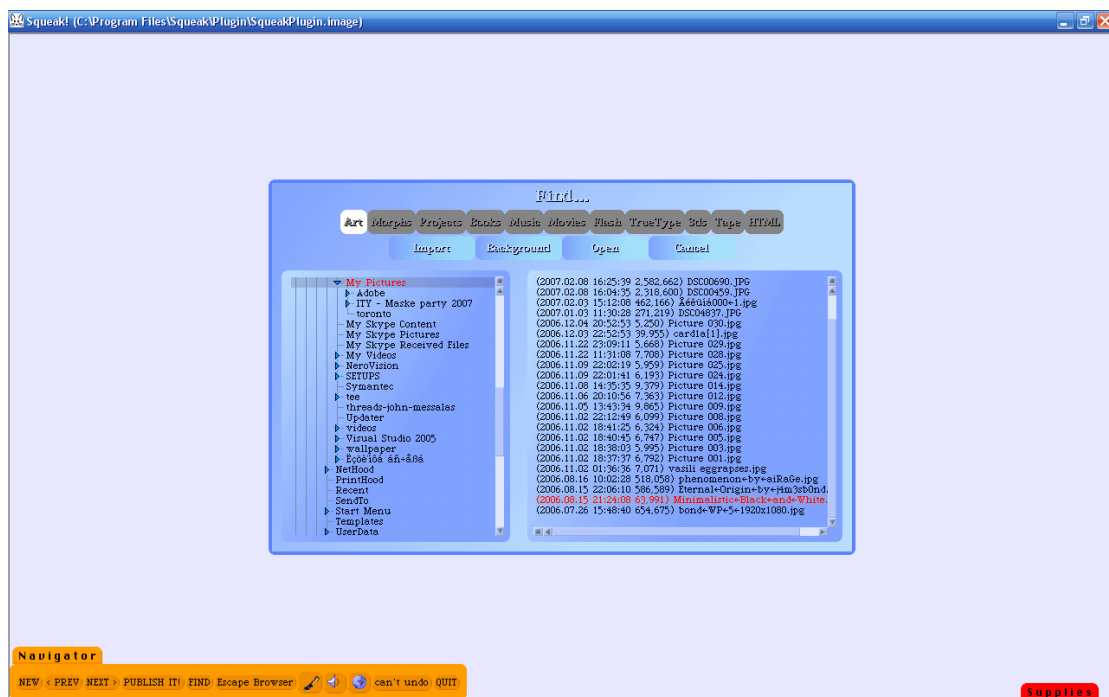


### **Εικόνα 120: Κρατώντας πατημένο το αριστερό κλικ πάνω στο Find**

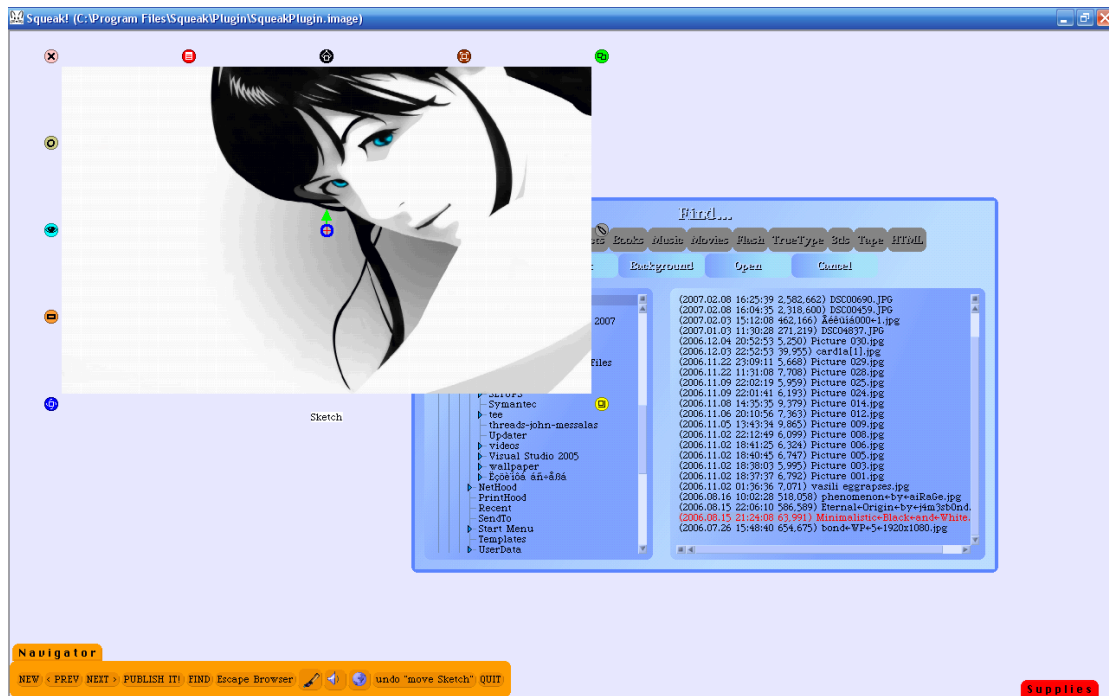
Στο μενού αυτό που εμφανίστηκε, πατάμε το 'find any file'. Τότε εμφανίζεται η εικόνα 121, η οποία ποικίλει ανάλογα με τα αρχεία του εκάστοτε προσωπικού σας υπολογιστή. Αναζητώντας το αρχείο που θέλουμε να φέρουμε στο 'Squeak Desktop', καταλήγουμε στην εικόνα 122. Στην εικόνα αυτή επιλέγουμε την επιθυμητή εικόνα και πατάμε 'Open'. Έτσι παίρνουμε το αποτέλεσμα της εικόνας 123. Έτσι η εικόνα θα εισέλθει στο 'Squeak Desktop'. Η εικόνα αυτή είναι ένα αντικείμενο σαν όλα τα άλλα: μπορεί να μετακινηθεί, να του αλλάξεις μέγεθος καθώς και όλες τις άλλες ιδιότητες των αντικειμένων.



Εικόνα 121: Καρτέλα επιλογής αρχείου για άνοιγμα



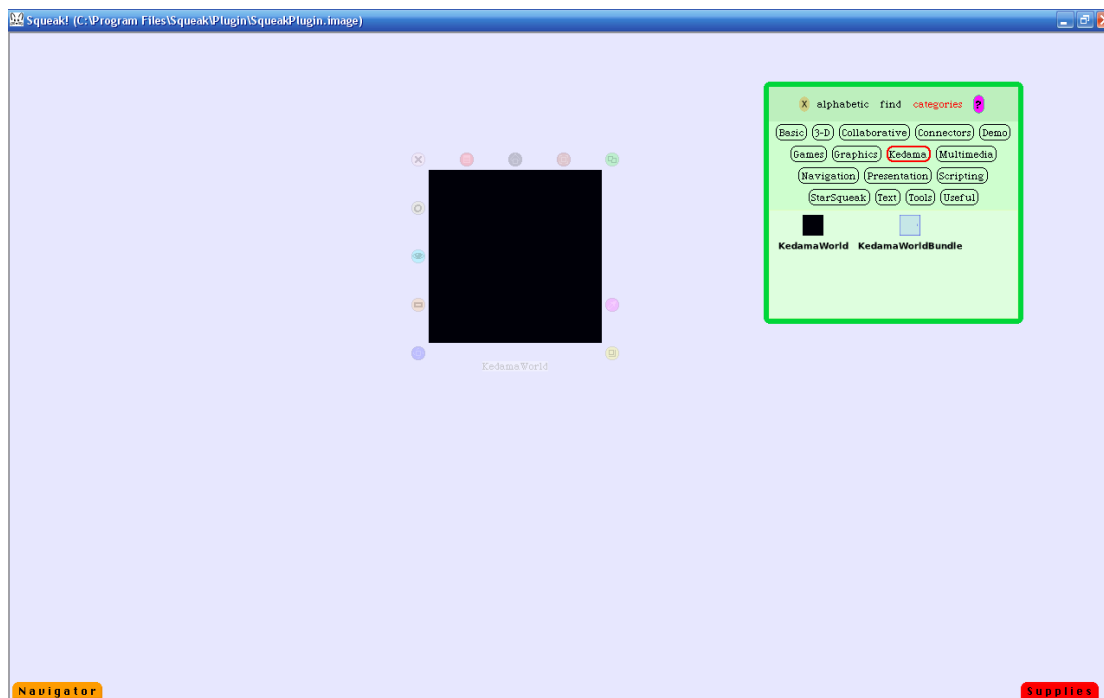
Εικόνα 122: Επιλογή εικόνας για άνοιγμα



Εικόνα 123: Η εικόνα που επιλέξαμε μέσα στο περιβάλλον του squeak

#### 4.10. Kedama-εξομοίωση κίνησης ατόμων

Στο τμήμα αυτό θα περιγράψουμε την κατασκευή μιας εφαρμογής με τη χρήση του 'Kedama', ενός scriptable τμήματος του προγράμματος Squeak. Το 'Kedama' μπορεί να χρησιμοποιηθεί για να δημιουργήσει εξομοιώσεις που περιλαμβάνουν αντικείμενα που αντιπροσωπεύουν άτομα, μυρμήγκια, βακτήρια ή συστήματα της ανθρώπινης διαδραστικότητας που μας επιτρέπουν να μελετήσουμε τυχαία ή/και οργανωμένη συμπεριφορά. Ξεκινάμε ανοίγοντας το πρόγραμμα και πηγαίνοντας στην καρτέλα 'Supplies'. Εκεί τραβάμε ένα αντικείμενο 'Object Catalog'. Εκεί υπάρχει ένα κουμπί που λέγεται 'Kedama' και το πατάμε. Στη συνέχεια τραβάμε στο 'Squeak Desktop' ένα αντικείμενο KedamaWorld όπως φαίνεται στην εικόνα 124.



**Εικόνα 124: Δημιουργώντας ένα Kedama-World αντικείμενο**

Κατόπιν, ανοίγουμε ένα 'Viewer' του αντικειμένου 'KedamaWorld' και βλέπουμε ότι υπάρχει μια νέα κατηγορία που λέγεται 'kedama'.

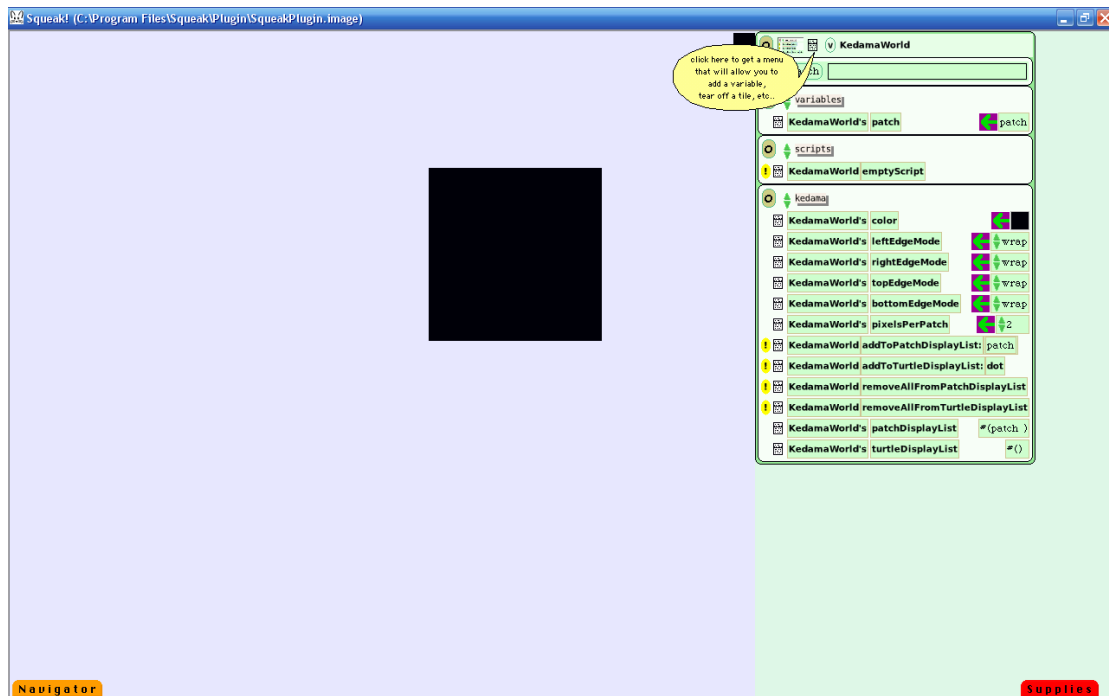
Όσον αφορά τους χρήστες με υψηλή ανάλυση στην οθόνη τους, υπάρχει η εντολή `pixelsPerPatch` στην παραπάνω κατηγορία με την οποία μπορείς να αλλάξεις τον αριθμό των pixel για να φαίνεται πιο καθαρά, οτιδήποτε φτιάχνεται στο αντικείμενο 'KedamaWorld'.

Θα ξεκινήσουμε γράφοντας ένα απλό script. Πρόκειται να φτιάξουμε πολλές χελώνες(turtles) που θα αναπαριστούν τα άτομα μέσα σε ένα σύστημα, και θα τα κάνουμε να κινηθούν.

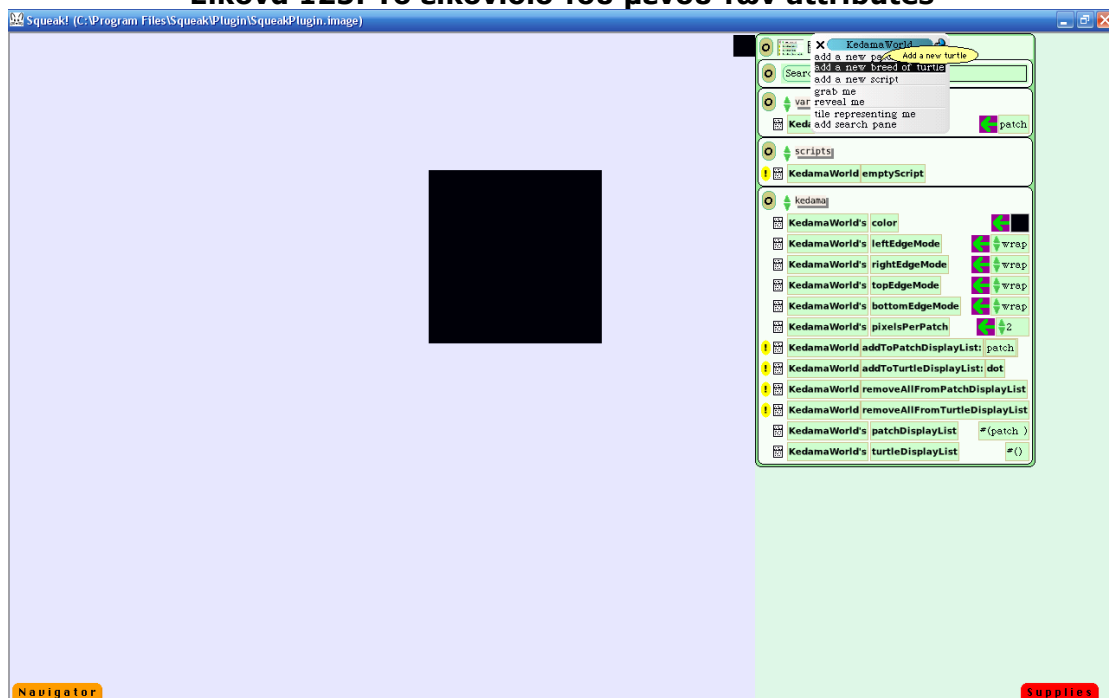
Τα βήματα είναι τα εξής:

- Φτιάχνουμε μια νέα ομάδα(breed) από χελώνες
- Ορίζουμε τον αριθμό των χελωνών στην ομάδα αυτή
- Κάνουμε τις χελώνες να κινηθούν

Για να φτιάξεις μια νέα ομάδα από χελώνες, πατάς πάνω στο εικονίδιο του μενού(menu)(εικόνα 125) που βρίσκεται στην κορυφή του 'Viewer' και επιλέγεις 'add a new breed of turtle'(εικόνα 126).

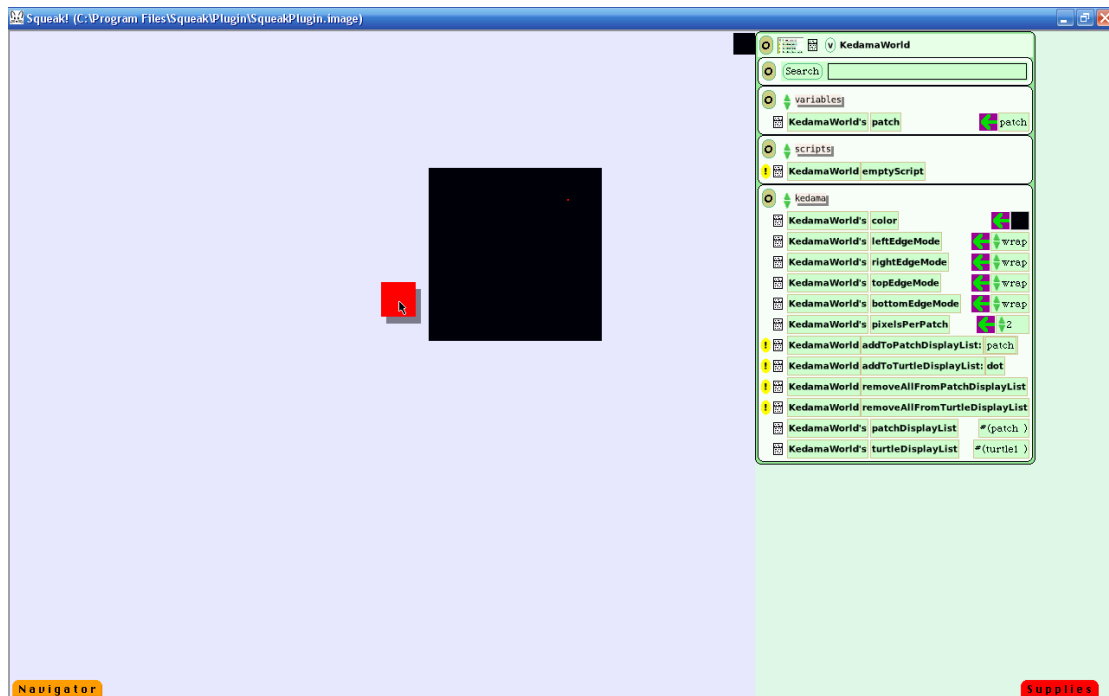


**Εικόνα 125: Το εικονίδιο του μενού των attributes**



**Εικόνα 126: Επιλογή του add a new breed of turtle**

Το αποτέλεσμα είναι να πάρεις ένα χρωματιστό κουτί στο δείκτη του ποντικιού όπως φαίνεται στην εικόνα 127.

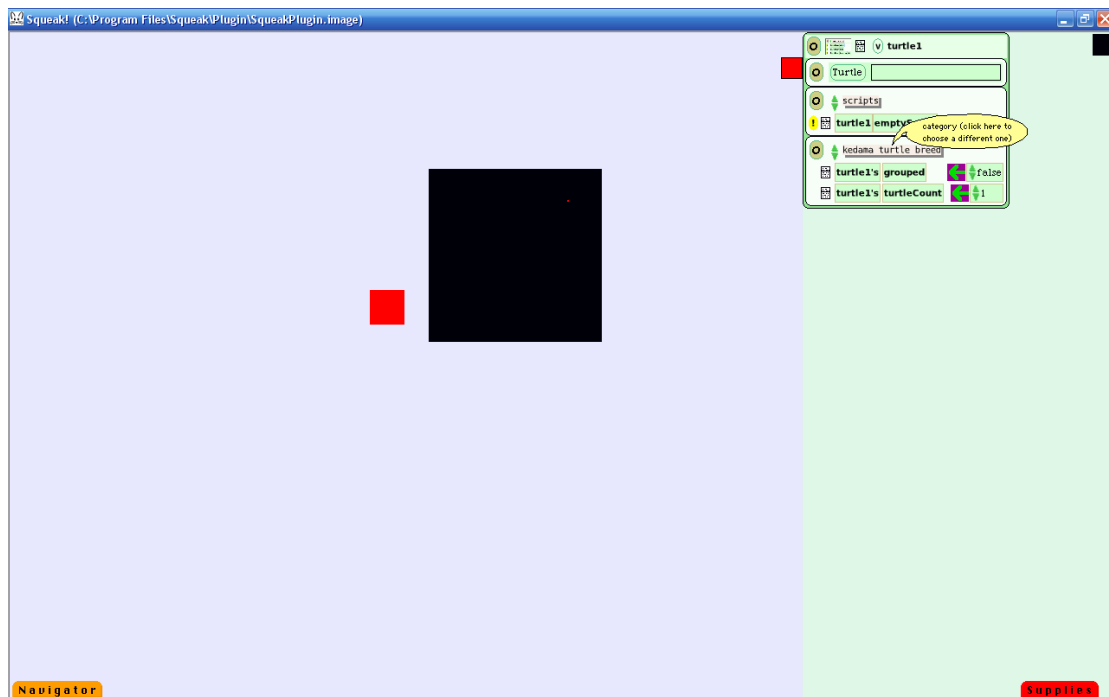


**Εικόνα 127: Το «χρωματιστό κουτί»**

Το κουτί αυτό είναι η αναπαράσταση της ομάδα των χελωνών στον κόσμο. Παίζει το ρόλο του μοντέλου(model) για όλες τις χελώνες που δημιουργείς από αυτή την ομάδα. Μια μικρή, χρωματιστή τελεία(μια χελώνα) θα εμφανιστεί ,παράλληλα, στο 'KedamaWorld'. Πάντα η πρώτη ομάδα από χελώνες είναι κόκκινη. Εάν δημιουργήσουμε περισσότερες από μια ομάδες χελωνών, τα επόμενα χρώματα αναθέτονται τυχαία.

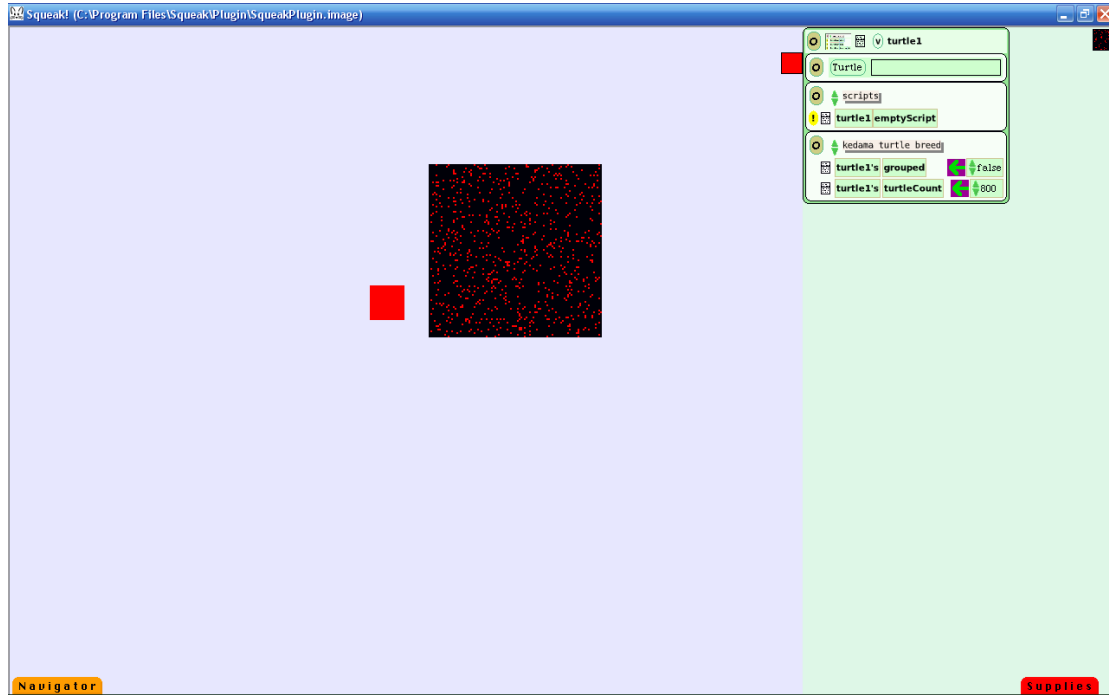
Στη συνέχεια, ορίζουμε τον αριθμό των χελωνών. Για να πάρουμε καινούργιες χελώνες υπάρχουν δύο τρόποι, ανάλογα με το τι θέλεις να κάνουν οι χελώνες.

Εάν θέλεις όλες οι χελώνες να συμπεριφερθούν με τον ίδιο τρόπο και να ακολουθήσουν το ίδιο script, τότε θέλεις να προσθέσεις χελώνες από την ίδια ομάδα όπως στην αρχή. Για το λόγο αυτό ανοίγουμε έναν 'Viewer' του κόκκινου τετραγώνου που αναπαριστά την ομάδα και επιλέγουμε την κατηγορία 'kedama turtle breed'(εικόνα 128)



**Εικόνα 128: Επιλέγοντας το Kedama Turtle Breed**

Στην κατηγορία αυτή, στην γραμμή που λέει 'turtle1's turtleCount' μπορούμε να ορίσουμε πόσες χελώνες θέλουμε. Όλες οι νέες μας χελώνες θα έχουν το ίδιο χρώμα με την αρχική. Επίσης, παρόλο που υπάρχουν πολλές χελώνες, μοιράζονται το ίδιο όνομα αφού είναι από την ίδια ομάδα. Εάν θέλουμε οι νέες χελώνες να συμπεριφέρονται διαφορετικά και να ακολουθήσουν ένα διαφορετικό script, πρέπει να προσθέσεις μια νέα ομάδα από χελώνες. Ακολουθώντας την παραπάνω διαδικασία, μπορείς να φτιάξεις μια νέα ομάδα, και ένα διαφορετικό χρώμα από τελείες θα εμφανιστεί. Για την εξομοίωσή μας, θέλουμε όλες οι χελώνες να συμπεριφέρονται με τον ίδιο τρόπο, σαν άτομα, για το λόγο αυτό θα χρησιμοποιήσουμε την πρώτη μέθοδο. Όταν τελειώσουμε, θα δούμε στο αντικείμενο 'KedamaWorld' γεμάτο με πολλές κόκκινες χελώνες μέσα του, όλες από την ίδια ομάδα(εικόνα 129).



**Εικόνα 129: Πολλές κόκκινες χελώνες μέσα στο Kedama World**

Τώρα θα κάνουμε τις χελώνες μας να κινηθούν. Είναι παρόμοιο με το να κάνεις άλλα αντικείμενα στο Squeak να κινηθούν. Έτσι, στον 'Viewer' της ομάδας των χελωνών (κόκκινο κουτί) επιλέγουμε την κατηγορία 'kedama turtle'. Εκεί υπάρχει η γραμμή 'turtle forward by'. Τραβάμε την γραμμή αυτή και φτιάχνουμε ένα νέο script. Στη συνέχεια ορίζουμε πόσο μακριά θέλουμε οι χελώνες να κινηθούν κάθε φορά, βάζουμε το script σε κατάσταση 'running' και βλέπουμε τις χελώνες μας να κινούνται.

Έτσι λοιπόν ολοκληρώσαμε με τη χρήση του kedama την κατασκευή μιας εξομοίωσης ατόμων.

## 5. Αναφορές

- [1] <http://www.squeak.org/>
- [2] <http://www.squeakland.org/>
- [3] <http://www.squeak.org/Documentation/> squeak documentation
- [4] <http://www.dmu.com/squeak/> οδηγός Squeak
- [5] <http://www.whysmalltalk.com/tutorials/squeak.htm> σύνδεσμοι για το Squeak
- [6] [http://www.objectsroot.com/squeak/squeak\\_tutorial.html](http://www.objectsroot.com/squeak/squeak_tutorial.html) Smalltalk για Java programmers
- [7] <http://www.bitwisemag.com/copy/programming/smalltalk/smalltalk1.html>  
SMALLTALK -squeak: A BEGINNER'S GUIDE
- [8] <http://www.cosc.canterbury.ac.nz/wolfgang.kreutzer/cosc205/smalltalk1.html>  
Basic Aspects of Squeak
- [9] <http://www.fit.vutbr.cz/study/courses/OMP/public/software/sqcdrom2/Tutorials/index.html> Squeak and Smalltalk Tutorials and Teaching Materials
- [10] <http://www.consultar.com/Squeak/tutorial/> tutorial squeak
- [11] <http://wilkesjoiner.com/UsingSqueak.html> using squeak
- [12] <http://static.squeak.org/tutorials/BankAccount.html> making a bank account
- [13] <http://wiki.squeak.org/squeak/792> squeak tutorials
- [14] <http://wiki.squeak.org/squeak/377> introduction to squeak
- [15] <http://www.smalltalk-central.com/showDocument.ssp?section=Articles&name=Tutorials> smalltalk central
- [16] <http://www.mars.dti.ne.jp/~umejava/smalltalk/squeak/sqlinks.html> squeak link page
- [17] <http://squeak.pbwiki.com/Connector%20Tutorial> squeak connector tutorial
- [18] [http://classic.sidwell.edu/academics/lower\\_school/LS\\_Tech\\_Projects/Squeak/SolarSys.pdf](http://classic.sidwell.edu/academics/lower_school/LS_Tech_Projects/Squeak/SolarSys.pdf) squeak tutorial solar system
- [19] <http://jacquesricher.com/rescued/SqueakTutorialEverything.pdf> παρουσίαση Squeak
- [20] <http://www.goodstart.com/tutorials.shtml> smalltalk links
- [21] <http://www.squeakcmi.org/staticpages/index.php?page=tutorials> tutorials για squeak
- [22] <http://www.fit.vutbr.cz/study/courses/OMP/public/software/sqcdrom2/Tutorials/Morphic%20Intro/Morphic.html> morphic-the squeak user interface
- [23] <http://wiki.squeak.org/squeak/4> basic squeak development tools
- [24] <http://www.muco.w.com/squeak-qref.html> squeak smalltalk: language reference
- [25] <http://static.squeak.org/tutorials/morphic-tutorial-1.html> tutorial with morphic graphics system
- [26] <http://coweb.cc.gatech.edu:8888/squeakbook/>
- [27] <http://coweb.cc.gatech.edu:8888/squeakbook/uploads/steinmetz.pdf>
- [28] <http://www.squeaksource.com/>
- [29] <http://dmoz.org/Computers/Programming/Languages/Smalltalk/Squeak/Software/>
- [30] <http://www.dmu.com/squeak/>
- [31] <http://www.dmu.com/squeak/sq0.html>
- [32] [http://www.squeakland.org/school/drive\\_a\\_car/html/Drivecar12.html](http://www.squeakland.org/school/drive_a_car/html/Drivecar12.html)
- [33] <http://squeakland.org/whatis/tutorials.html>

## Αναφορές

- [34] [http://classic.sidwell.edu/academics/lower\\_school/LS\\_Tech\\_Projects/squeak!.htm](http://classic.sidwell.edu/academics/lower_school/LS_Tech_Projects/squeak!.htm)
- [35] <http://www.squeakland.org/sqmedia/projects/etoyposters.htm>