



Εγχειρίδιο Εκμάθησης  
Squeak / Squeak Etoys  
για προχωρημένους χρήστες

Ομάδα Εργασίας  
Βασίλειος Πουλόπουλος  
Ιωάννης Παππάς

**Περιεχόμενα**

1.	Εισαγωγή .....	3
1.1.	Χαρακτηριστικά του Squeak .....	6
1.2.	Smalltalk .....	7
1.3.	Squeak και olpc .....	8
2.	Παραδείγματα Χρήσης για προχωρημένους χρήστες.....	9
2.1.	Κατασκευή του ηλιακού συστήματος.....	9
2.2.	Προσγείωση στη σελήνη .....	22
3.	Σχολικά παραδείγματα .....	34
3.1.	Κατάσταση της ύλης.....	34
3.2.	Θερμοκρασία και Πίεση .....	42
3.3.	Ζυγαριά κλασμάτων.....	63
3.4.	Αριθμητικό σταυρόλεξο .....	74
3.5.	Εμβαδόν επίπεδων σχημάτων .....	78
4.	Αναφορές .....	86

## Λίστα Σχημάτων

Εικόνα 1: Κατασκευή του πλανήτη ήλιου .....	9
Εικόνα 2: Κατασκευή τριών ακόμα πλανητών .....	10
Εικόνα 3: Γέμισμα φόντου με μαύρο χρώμα .....	10
Εικόνα 4: Μείωση του μεγέθους του φόντου .....	11
Εικόνα 5: Τοποθετώντας το αντικείμενο ‘eclipse’ πάνω στο μαύρο φόντο .....	11
Εικόνα 6: Αλλαγή μεγέθους του αντικειμένου ‘eclipse’ .....	12
Εικόνα 7: Χρήση του ροζ εργαλείου ‘Property Sheet’ .....	12
Εικόνα 8: Κατασκευάζοντας τις τροχιές των πλανητών .....	13
Εικόνα 9: Ολοκλήρωση των τροχιών των πλανητών .....	13
Εικόνα 10: Ο ήλιος στο κέντρο του πλανητικού συστήματος .....	14
Εικόνα 11: Η επιλογή ‘μάτι’ του αντικειμένου ‘planet3’ (μπλε πλανήτης) .....	14
Εικόνα 12: Επιλέγοντας ένα ‘empty script’ για το αντικείμενο ‘planet3’ .....	15
Εικόνα 13: Δίνοντας το όνομα ‘revolving planet3’ στο ‘empty script’ του αντικειμένου ‘planet3’ .....	15
Εικόνα 14: το κουμπί του ‘test’ του script ‘revolving planet3’ .....	16
Εικόνα 15: Δημιουργία ‘test’ δυνατότητας στο script-βήμα 1 .....	16
Εικόνα 16: Δημιουργία ‘test’ δυνατότητας στο script-βήμα 2 .....	17
Εικόνα 17: Αλλάζοντας κατηγορία από ‘basic’ σε ‘tests’ .....	17
Εικόνα 18: Κατηγορία ‘tests’ .....	18
Εικόνα 19: Τοποθέτηση της γραμμής ‘color sees’ στο script ‘revolvingplanet3’ βήμα 1 .....	18
Εικόνα 20: Τοποθέτηση της γραμμής ‘color sees’ στο script ‘revolvingplanet3’ βήμα 2 .....	19
Εικόνα 21: Χρήση του εργαλείου σταγονόμετρο βήμα 1 .....	19
Εικόνα 22: Χρήση του εργαλείου σταγονόμετρο βήμα 2 .....	20
Εικόνα 23: Χρήση του εργαλείου σταγονόμετρο βήμα 3 .....	20
Εικόνα 24: Τοποθέτηση της γραμμής ‘forward by’ στο script .....	21
Εικόνα 25: Δημιουργία δευτέρου test στο script .....	21
Εικόνα 26: Δημιουργία τρίτου test στο script .....	22
Εικόνα 27: Αντικείμενα ‘playfield’, ‘joystick’ και ‘stop/step/go’ .....	23
Εικόνα 28: Κατασκευή του διαστημοπλοίου .....	23
Εικόνα 29: Άνοιγμα του εργαλείου ‘μάτι’ του αντικειμένου-διαστημοπλοίου .....	24
Εικόνα 30: Αλλαγή του ονόματος του αντικειμένου-διαστημοπλοίου .....	24
Εικόνα 31: Δημιουργία καινούργιας μεταβλητής για το αντικείμενο-διαστημόπλοιο .....	25
Εικόνα 32: Η κατηγορία ‘variables’ του αντικειμένου-διαστημόπλοιο .....	25
Εικόνα 33: Το script ‘gravity’ .....	26
Εικόνα 34: Κατασκευάζοντας το script ‘gravity’ βήμα 1 .....	26
Εικόνα 35: Κατασκευάζοντας το script ‘gravity’ βήμα 2 .....	27
Εικόνα 36: Κατασκευάζοντας το script ‘gravity’ βήμα 3 .....	27
Εικόνα 37: Κατασκευάζοντας το script ‘gravity’ βήμα 4 .....	28
Εικόνα 38: Κατασκευάζοντας το script ‘gravity’ βήμα 5 .....	28
Εικόνα 39: Κατασκευάζοντας το script ‘gravity’ βήμα 6 .....	29
Εικόνα 40: Το script ‘motor’ .....	29
Εικόνα 41: Κατασκευή του χώρου προσγείωσης του διαστημοπλοίου .....	30
Εικόνα 42: Το script ‘land’ .....	30
Εικόνα 43: Το script ‘allprocesspause’ .....	31
Εικόνα 44: Επεκτείνοντας τη λειτουργικότητα του script ‘land’ .....	32
Εικόνα 45: Η επιλογή ‘button to fire this script’ στο menu ενός script .....	32

Εικόνα 46: Το τελικό πρόγραμμα .....	33
Εικόνα 47: όλα τα αντικείμενα που θα χρειαστούν .....	34
Εικόνα 48: τοποθετώντας μία επιπλέον μεταβλητή σε ένα αντικείμενο .....	36
Εικόνα 49: όλα τα scripts του προγράμματος.....	37
Εικόνα 50: Τα στοιχεία του RESET .....	37
Εικόνα 51: Τα στοιχεία του World script .....	38
Εικόνα 52: Scripts για τα κουμπιά επιλογής του χρήστη .....	38
Εικόνα 53: Στοιχεία που πρέπει να μπουν μέσα στο script της ένωσης.....	39
Εικόνα 54: το script για το κουμπί start.....	41
Εικόνα 55: Δημιουργώντας ένα Kedama-World αντικείμενο .....	42
Εικόνα 56: Το εικονίδιο του μενού των attributes.....	43
Εικόνα 57: Επιλογή του add a new breed of turtle .....	43
Εικόνα 58: Το «χρωματιστό κουτί».....	44
Εικόνα 59: Επιλέγοντας το Kedama Turtle Breed.....	44
Εικόνα 60: Πολλές κόκκινες χελώνες μέσα στο Kedama World .....	45
Εικόνα 61:Το αντικείμενο ‘Text’ στην επιφάνεια εργασίας.....	45
Εικόνα 62: Συμπλήρωση τίτλου του προγράμματος .....	46
Εικόνα 63: Το ‘Object Catalog’ .....	46
Εικόνα 64: Το αντικείμενο ‘Text (border)’ .....	47
Εικόνα 65: Συμπληρώνοντας το αντικείμενο ‘Text (border)’ .....	47
Εικόνα 66: Βάζοντας τις τιμές του παραδείγματός μας.....	48
Εικόνα 67: Αλλαγή ονόματος αντικειμένου .....	48
Εικόνα 68: Αντικείμενο world.....	49
Εικόνα 69: Μενού αντικειμένου ‘world’ .....	50
Εικόνα 70: Δημιουργώντας μια μεταβλητή του αντικειμένου ‘world’ .....	50
Εικόνα 71: Η μεταβλητή ‘global’ του αντικειμένου ‘world’ .....	51
Εικόνα 72: Η μεταβλητή ‘var20’ του αντικειμένου ‘20’ .....	51
Εικόνα 73: Δημιουργία script για το αντικείμενο ‘20’ .....	52
Εικόνα 74: Γραμμή ‘temp’s characters’ της κατηγορίας basic.....	52
Εικόνα 75:Script του αντικειμένου ‘20’ .....	53
Εικόνα 76: γραμμή ‘20’s characters’ του αντικειμένου ‘20’ βήμα 1 .....	53
Εικόνα 77: γραμμή ‘20’s characters’ του αντικειμένου ‘20’ βήμα 2 .....	54
Εικόνα 78: Γραμμή ‘press’s characters’ της κατηγορίας basic .....	54
Εικόνα 79:Script του αντικειμένου ‘20’ .....	55
Εικόνα 80: γραμμή ‘20’s characters’ του αντικειμένου ‘20’ βήμα 1 .....	55
Εικόνα 81: γραμμή ‘20’s characters’ του αντικειμένου ‘20’ βήμα 2 .....	56
Εικόνα 82: Χρήση της μεταβλητής ‘world’s global’ του αντικειμένου ‘world’ .....	56
Εικόνα 83: νέα γραμμή για το script του αντικειμένου ‘20’ .....	57
Εικόνα 84: Η μεταβλητή ‘world’s global’ παίρνει την τιμή της μεταβλητής ‘var20’ .....	57
Εικόνα 85: Τα script της θερμοκρασίας-πίεσης.....	58
Εικόνα 86: Μενού επιλογών του script.....	58
Εικόνα 87: Το κουμπί εκτέλεσης του πρώτου script .....	59
Εικόνα 88: Αλλαγή ετικέτας κουμπιού εκτέλεσης script .....	59
Εικόνα 89: Ολοκλήρωση της δημιουργίας των κουμπιών εκτέλεσης script .....	60
Εικόνα 90: Αλλαγή μεγέθους του αντικειμένου ‘Kedama World’ .....	60
Εικόνα 91: Κατασκευή δοχείου για το αέριο.....	61
Εικόνα 92: Αλλαγή χρώματος του αντικειμένου ‘Kedama World’ .....	61
Εικόνα 93: Δημιουργία script για την κίνηση των μορίων.....	62
Εικόνα 94: Ολοκλήρωση script ‘kinisiMorian’ .....	62
Εικόνα 95: Χρήση αντικειμένου ‘All Scripts’ .....	63

Εικόνα 96: Ο σχεδιασμός της ζυγαριάς .....	64
Εικόνα 97: Το εργαλείο ‘μάτι’ του αντικειμένου ‘world’ .....	64
Εικόνα 98: Οι μεταβλητές ‘var1’ και ‘var2’ του αντικειμένου ‘world’ .....	65
Εικόνα 99: Δημιουργία του πρώτου κλάσματος της ζυγαριάς .....	66
Εικόνα 100: Όλα τα κλάσματα του προγράμματος .....	66
Εικόνα 101: Η μεταβλητή ‘world’s var1’ στο script ‘world script1’ .....	67
Εικόνα 102: script επιλογής κλάσματος .....	68
Εικόνα 103: Κουμπιά επιλογής των κλασμάτων .....	68
Εικόνα 104: Τρία αντικείμενα ‘Arrow’ .....	69
Εικόνα 105: Περιστροφή των βελών της ζυγαριάς .....	69
Εικόνα 106: Εξαφάνιση των script από την οθόνη του τελικού χρήστη .....	70
Εικόνα 107: δημιουργία ‘Test’ μέσα στο script .....	70
Εικόνα 108: Δημιουργία τριών ‘Test’ στο script count.....	71
Εικόνα 109: Πρώτη σύγκριση για την κίνηση των βελών της ζυγαριάς .....	71
Εικόνα 110: Κατασκευή των υπόλοιπων συγκρίσεων του script ‘count’ .....	72
Εικόνα 111: Η τελική μορφή του script ‘count’ .....	72
Εικόνα 112: Κουμπί εκτέλεσης του script count .....	73
Εικόνα 113: Δημιουργία script ‘reset’ .....	74
Εικόνα 114: Αρχική κατάσταση του προγράμματος ζυγαριάς .....	74
Εικόνα 115: το αριθμητικό σταυρόλεξο .....	75
Εικόνα 116: όλα τα script για τα scrolling texts .....	75
Εικόνα 117: όλα τα scripts των scrolling texts μαζί με την αντίστοιχη τιμή των κενών κελιών .....	76
Εικόνα 118: Το script για το κουμπί εκκίνησης .....	77
Εικόνα 119: Τα scripts των κουμπιών βοήθειας .....	77
Εικόνα 120: Το σύστημά μας ολοκληρωμένο .....	78
Εικόνα 121: κατασκευή κύκλου με την ακτίνα του .....	78
Εικόνα 122: Εμβαδό που θέλουμε να υπολογίσουμε στον κύκλο .....	79
Εικόνα 123: χρήση του εργαλείου lasso βήμα 1 .....	79
Εικόνα 124: χρήση του εργαλείου lasso βήμα 2 .....	80
Εικόνα 125: Ελαχιστοποίηση του αντικειμένου ‘kyklos’ .....	80
Εικόνα 126: Δημιουργία του αντικειμένου ‘tetragono’ .....	81
Εικόνα 127: Δημιουργία του αντικειμένου ‘orthogonio’ .....	81
Εικόνα 128: Δημιουργία του αντικειμένου ‘parallilogramo’ .....	82
Εικόνα 129: Δημιουργία του αντικειμένου ‘trigono’ .....	82
Εικόνα 130: ονομασία script εμφάνισης αντικειμένων .....	83
Εικόνα 131: Δημιουργία script εμφάνισης των αντικειμένων .....	84
Εικόνα 132: Δημιουργία κουμπιών εκτέλεσης των script .....	84
Εικόνα 133: script reset .....	85
Εικόνα 134: Τελική μορφή του προγράμματος .....	85

## 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 περάσματος 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 NNTP(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 παίζει 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.



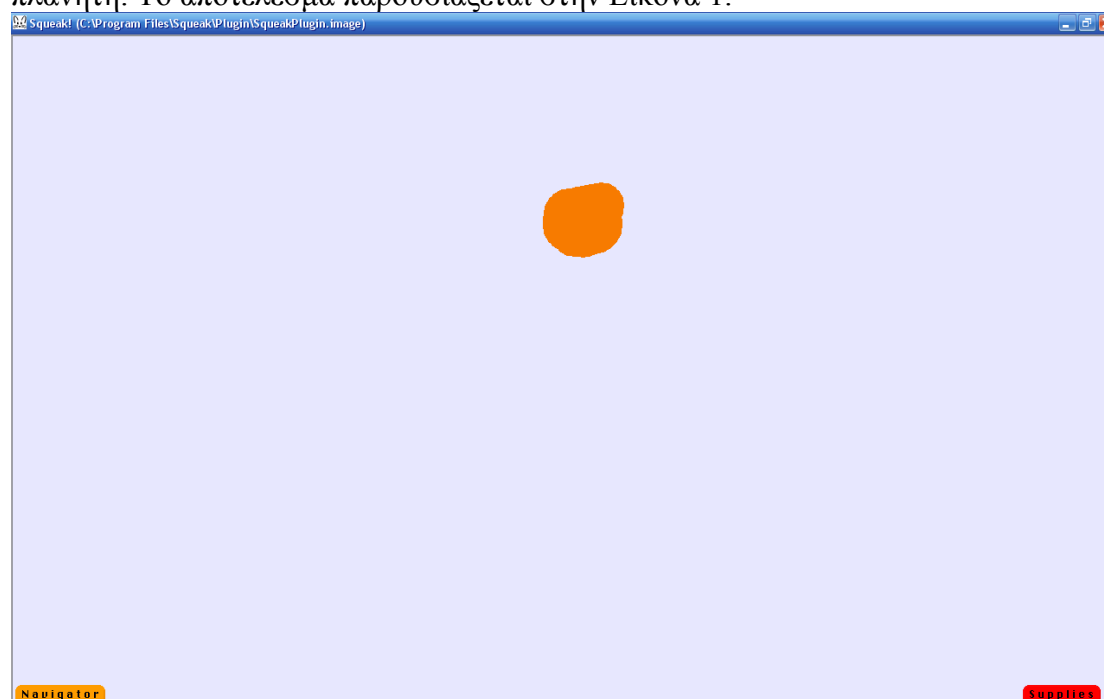
## 2. Παραδείγματα Χρήσης για Προχωρημένους Χρήστες

Σε αυτή την ενότητα γίνεται παρουσίαση δύο παραδειγμάτων που δίνουν τη δυνατότητα στους χρήστες που είναι ήδη εξοικειωμένοι με το περιβάλλον Etoys, να μελετήσουν πιο προχωρημένες λειτουργίες του.

### 2.1. Κατασκευή του ηλιακού συστήματος

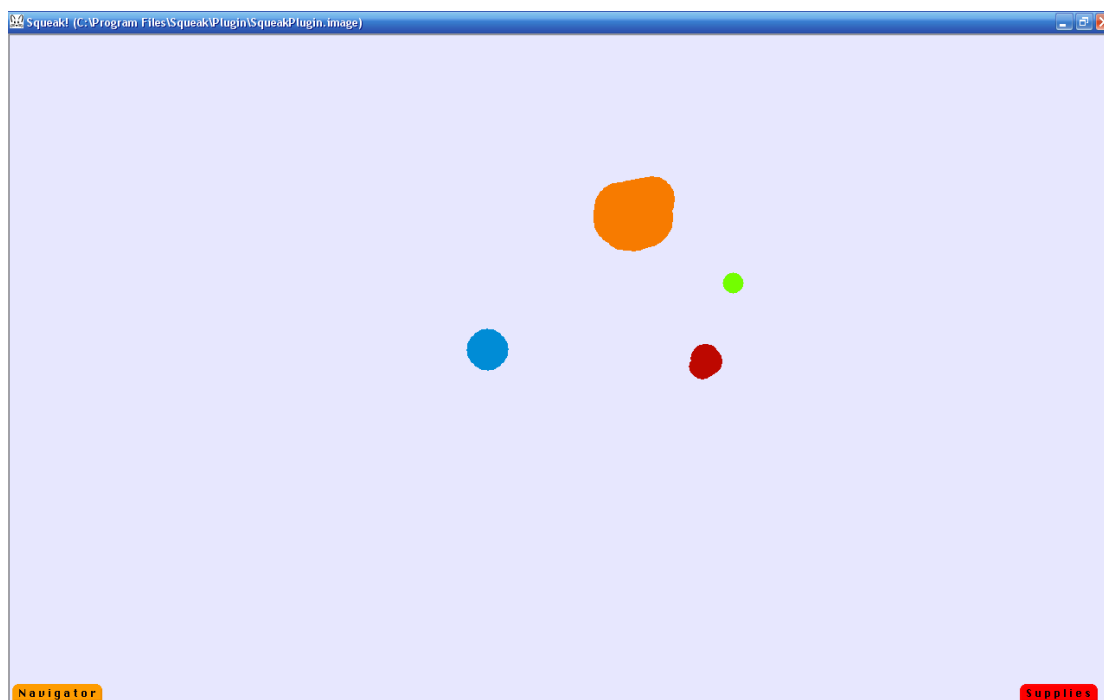
Το πρώτο παράδειγμα αφορά την κατασκευή του ηλιακού συστήματος. Ο στόχος είναι η αναδημιουργία του ηλιακού συστήματος, με την κατασκευή ενός χώρου που περιλαμβάνει πλανήτες, κομήτες, ή μετεωρίτες που περιστρέφονται γύρω από τον ήλιο. Οι συναρτήσεις του Etoys που θα χρησιμοποιηθούν είναι: forward by, turn by και color sees.

Πρώτα θα πρέπει να γίνει η κατασκευή των πλανητών. Ξεκινάμε ανοίγοντας την παλέτα ζωγραφικής από την καρτέλα 'Navigator' και κατασκευάζοντας τον πρώτο πλανήτη. Το αποτέλεσμα παρουσιάζεται στην Εικόνα 1.



Εικόνα 1: Κατασκευή του πλανήτη ήλιου

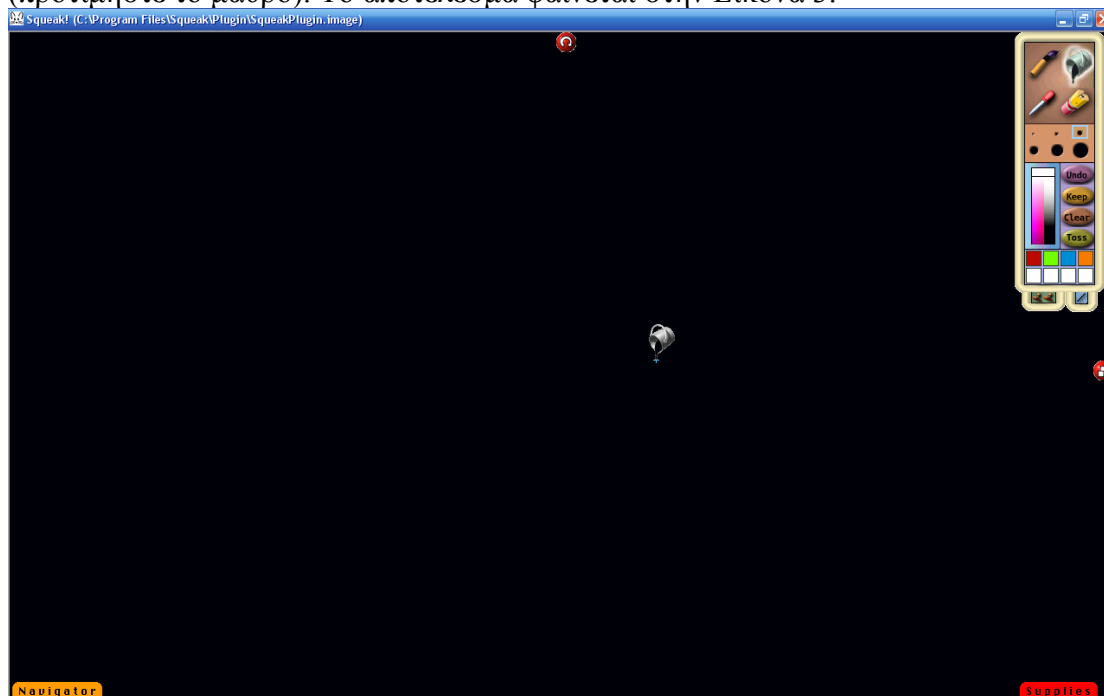
Οι πλανήτες μας μπορούν να κατασκευαστούν σε οποιοδήποτε μέγεθος και χρώμα θέλουμε. Μπορούμε να χρωματίσουμε του κύκλους με διαφορετικές σκιές χρωμάτων για να γίνει πιο ενδιαφέρον. Μετά την κατασκευή τριών ακόμα πλανητών, το αποτέλεσμα φαίνεται στην Εικόνα 2.



**Εικόνα 2:Κατασκευή τριών ακόμα πλανητών**

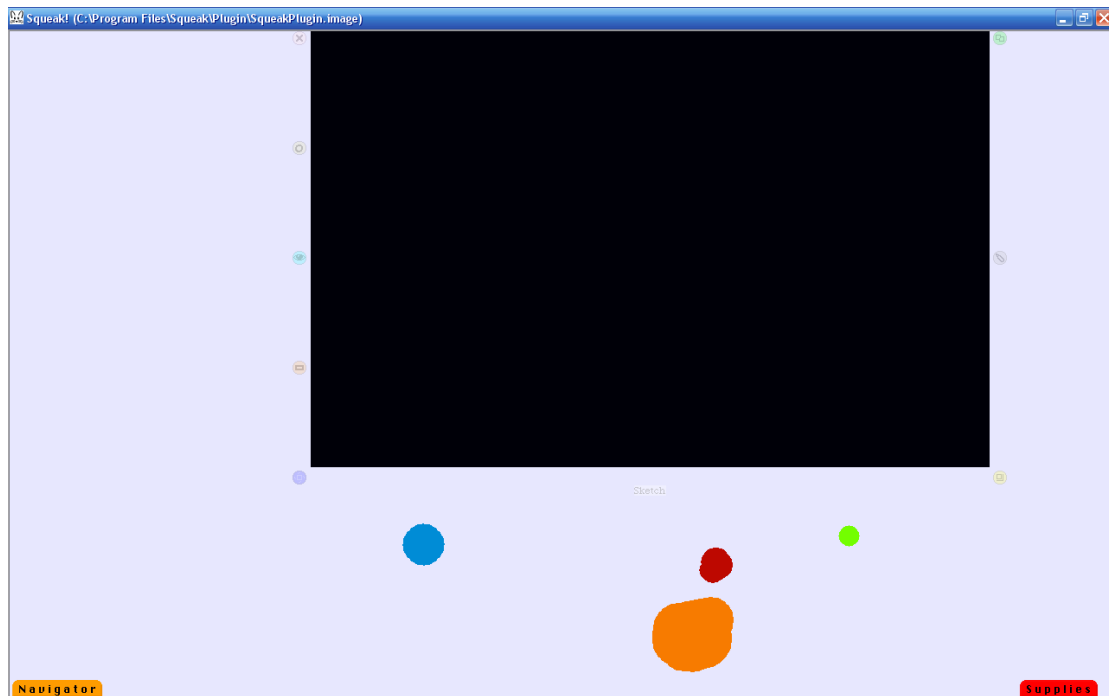
Στην συνέχεια δίνουμε σε κάθε αντικείμενο-πλανήτη το όνομα που επιθυμούμε. Για τις ανάγκες του παραδείγματος θα περιοριστούμε στους τέσσερις αυτούς πλανήτες, αλλά δεν υπάρχει κάποιος περιορισμός στον αριθμό των πλανητών που μπορούμε να κατασκευάσουμε.

Το επόμενο βήμα είναι η κατασκευή του χώρου κίνησης των πλανητών. Ανοίγουμε την παλέτα ζωγραφικής και γεμίζουμε το φόντο με ένα οποιοδήποτε σκούρο χρώμα (προτιμήστε το μαύρο). Το αποτέλεσμα φαίνεται στην Εικόνα 3.



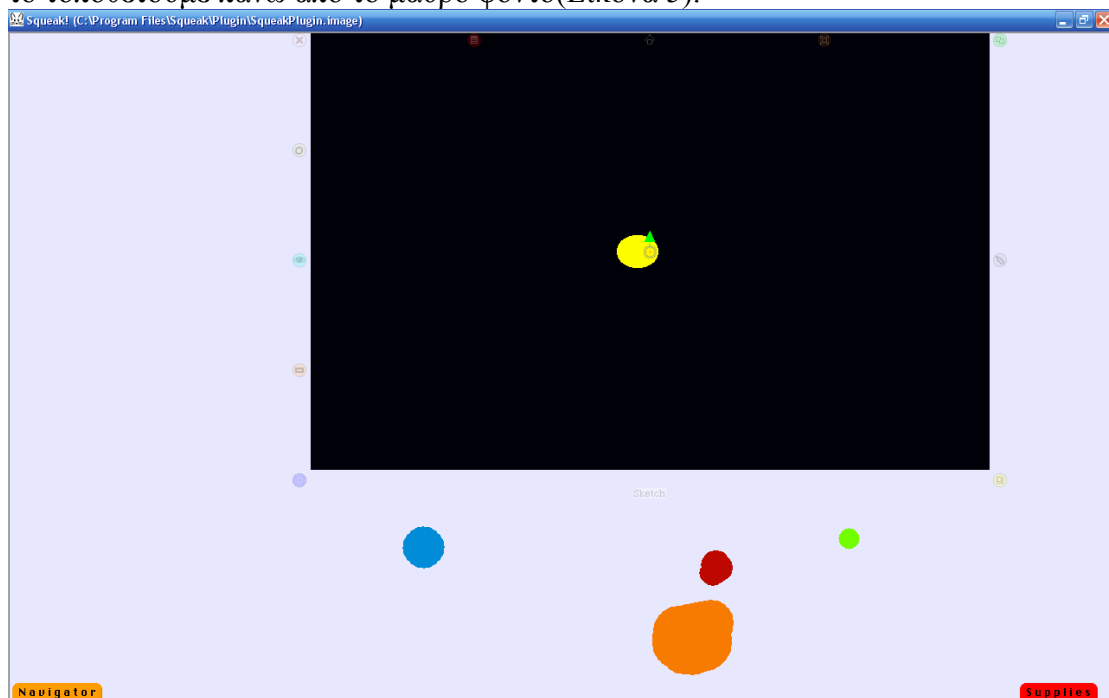
**Εικόνα 3: Γέμισμα φόντου με μαύρο χρώμα**

Αφού πατήσουμε το κουμπί 'keep',στη συνέχεια μειώνουμε λίγο το μέγεθος του φόντου παίρνοντας το αποτέλεσμα της Εικόνα 4.



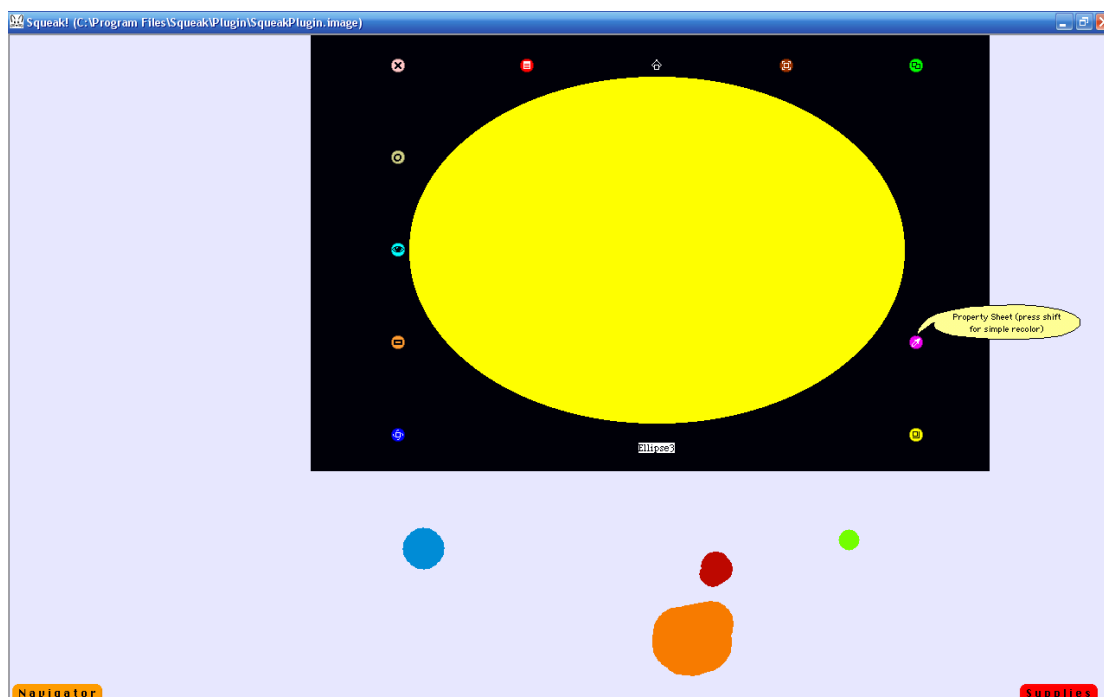
**Εικόνα 4: Μείωση του μεγέθους του φόντου**

Κατόπιν ανοίγουμε την καρτέλα 'Supplies' και τραβάμε ένα αντικείμενο 'eclipse' και το τοποθετούμε πάνω από το μαύρο φόντο(Εικόνα 5).



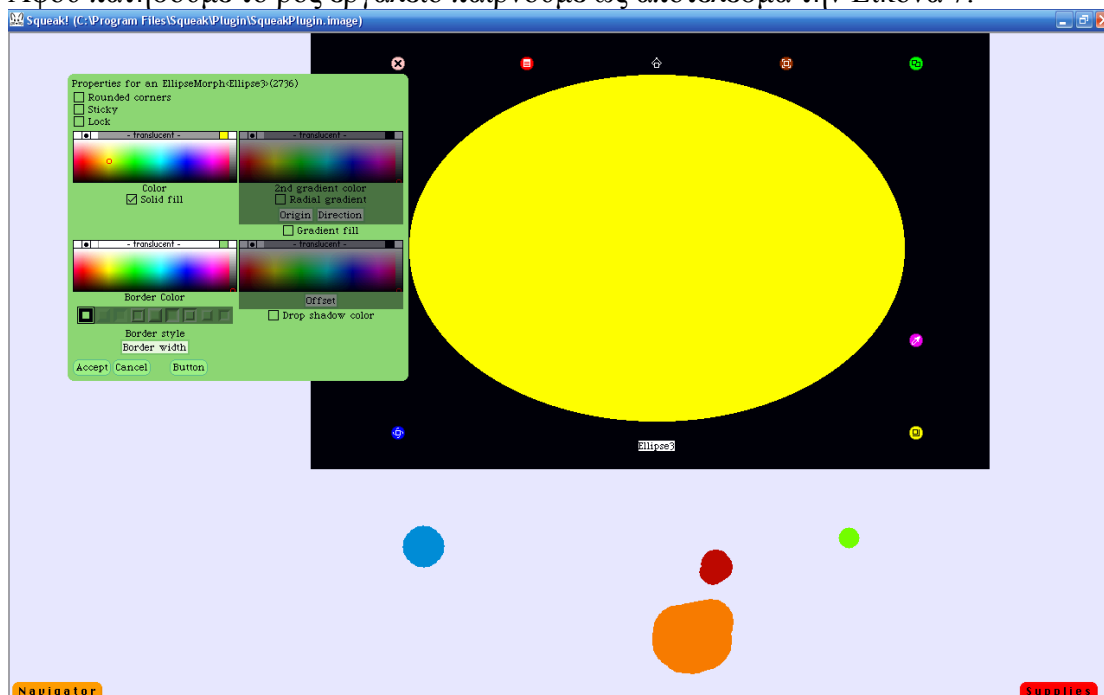
**Εικόνα 5: Τοποθετώντας το αντικείμενο 'eclipse' πάνω στο μαύρο φόντο**

Το επόμενο βήμα είναι να αλλάξουμε το μέγεθος του αντικειμένου 'eclipse' και να πατήσουμε το ροζ εργαλείο που ως αντικείμενο(Εικόνα 6).



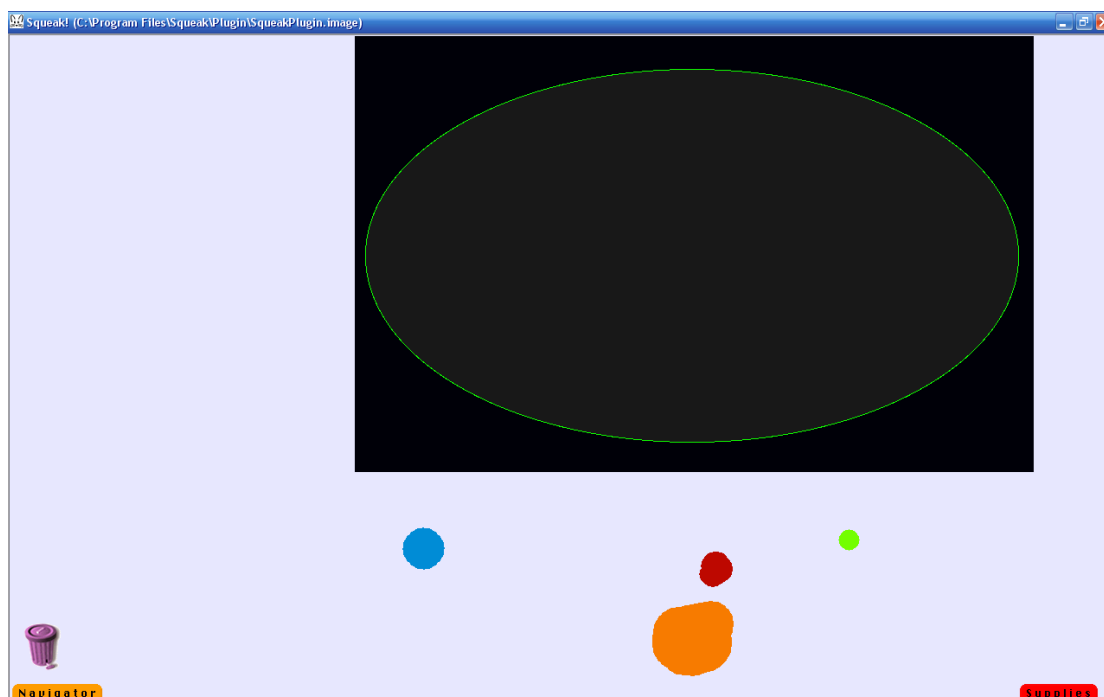
Εικόνα 6: Αλλαγή μεγέθους του αντικειμένου 'eclipse'

Αφού πατήσουμε το ροζ εργαλείο παίρνουμε ως αποτέλεσμα την Εικόνα 7.



Εικόνα 7: Χρήση του ροζ εργαλείου 'Property Sheet'

Με τη χρήση του πάνω κουτιού αλλάζει το χρώμα του αντικειμένου 'eclipse' ενώ το κάτω κουτί χρησιμεύει για την επιλογή της εξωτερικής γραμμής του αντικειμένου 'eclipse'. Τη γραμμή αυτή θα ακολουθήσει ο εκάστοτε πλανήτης μας. Μετά την επιλογή των χρωμάτων πατάμε το κουμπί 'Accept'. Το αποτέλεσμα φαίνεται στην Εικόνα 8.



**Εικόνα 8: Κατασκευάζοντας τις τροχιές των πλανητών**

Με τον ίδιο τρόπο κατασκευάζονται και οι υπόλοιπες τροχιές με το αποτέλεσμα να φαίνεται στην Εικόνα 9.



**Εικόνα 9: Ολοκλήρωση των τροχιών των πλανητών**

Στη συνέχεια ο πλανήτης ήλιος τοποθετείται στο κέντρο των τροχιών (Εικόνα 10), ενώ οι υπόλοιποι τρεις πλανήτες θα πρέπει να τεθούν σε τροχιά γύρω από τον ήλιο.



**Εικόνα 10: Ο ήλιος στο κέντρο του πλανητικού συστήματος**

Θα γίνει η αρχή με τον πλανήτη ο οποίος θα βρίσκεται στη πιο μακρινή τροχιά και είναι ο μπλε πλανήτης με το όνομα 'planet3'. Ανοίγουμε τις επιλογές του αντικειμένου και πατάμε το 'μάτι' (Εικόνα 11).



**Εικόνα 11: Η επιλογή 'μάτι' του αντικειμένου 'planet3' (μπλε πλανήτης)**

Τραβάμε ένα 'empty script' (Εικόνα 12) και του δίνουμε ένα όνομα όπως για παράδειγμα 'revolving planet3' (Εικόνα 13).

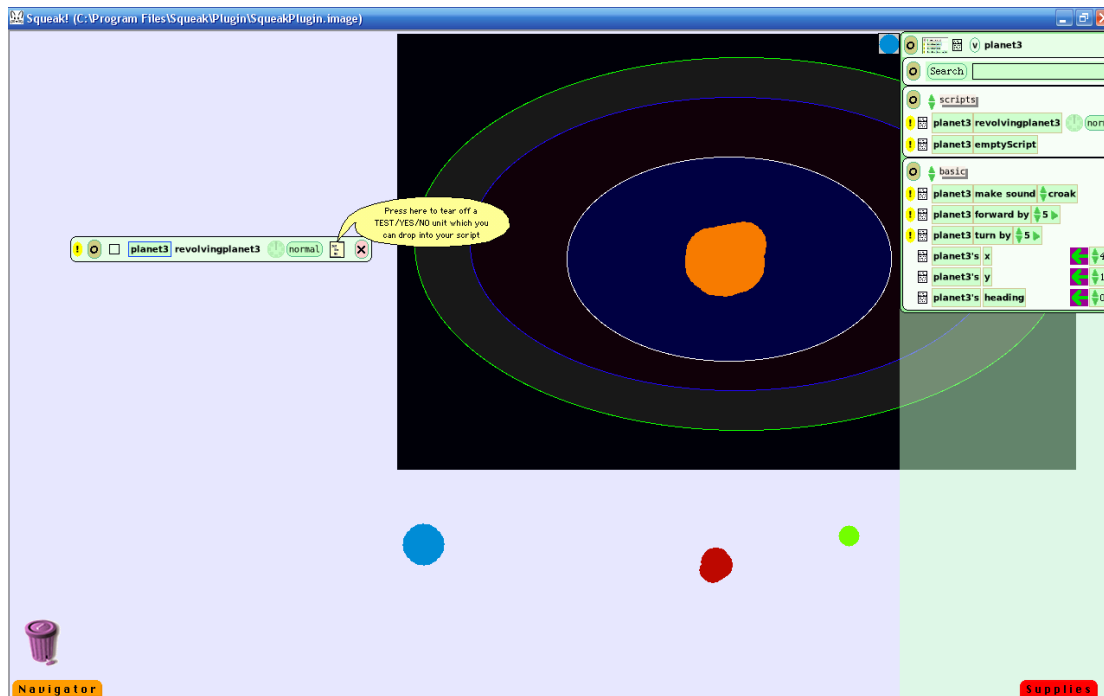


Εικόνα 12: Επιλέγοντας ένα 'empty script' για το αντικείμενο 'planet3'



Εικόνα 13: Δίνοντας το όνομα 'revolving planet3' στο 'empty script' του αντικείμενου 'planet3'. Στην πάνω δεξιά γωνία του 'script' υπάρχει ένα κίτρινο κουμπί που είναι σαν ένα κομμάτι χαρτί και λέγεται 'test' (Εικόνα 14).

## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες



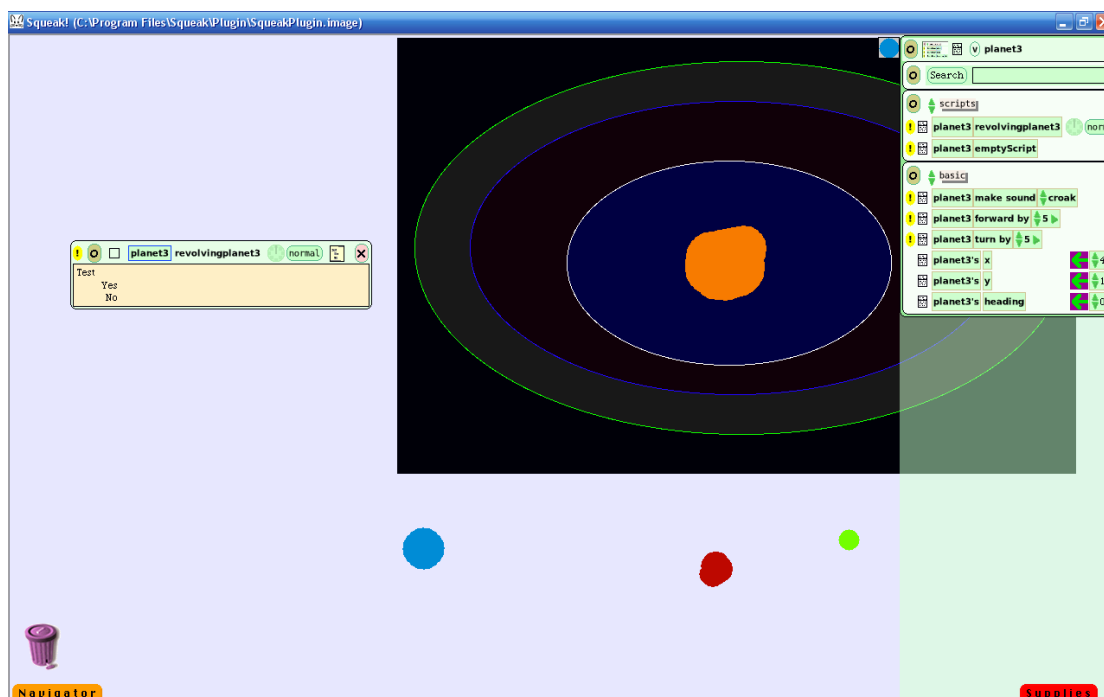
Εικόνα 14: το κουμπί του 'test' του script 'revolving planet3'

Το κουμπί 'test' το τραβάμε προς τα κάτω μέχρι ένα πράσινο κουτί να εμφανιστεί(Εικόνα 15) και στη συνέχεια το αφήνουμε εκεί(Εικόνα 16).



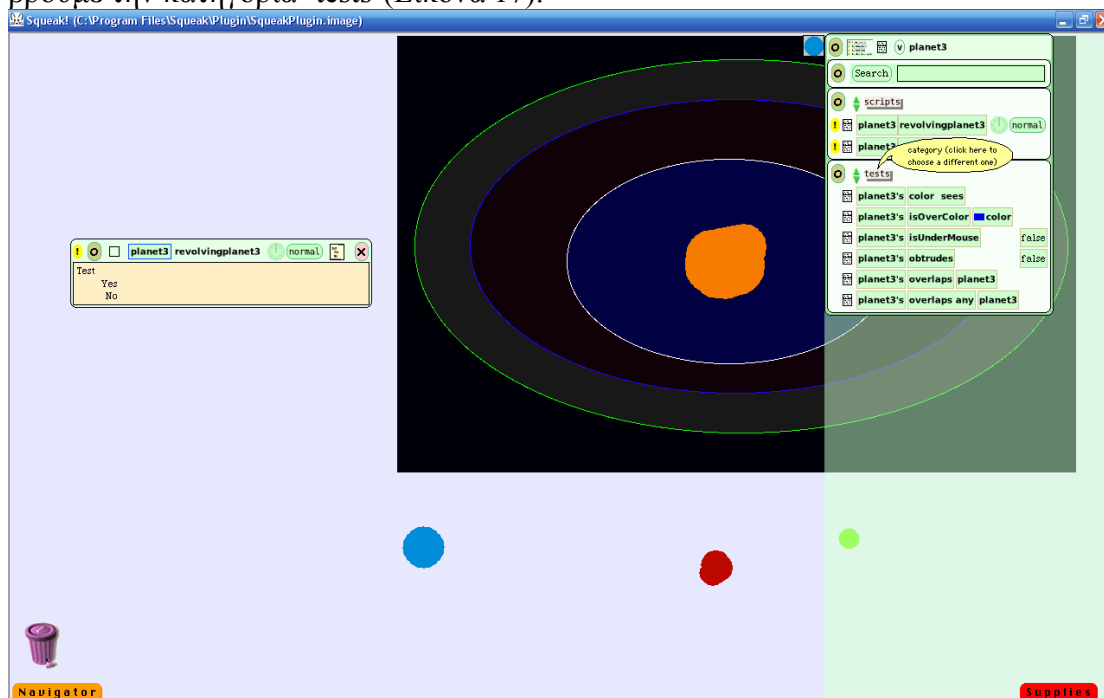
Εικόνα 15: Δημιουργία 'test' δυνατότητας στο script-βήμα 1





Εικόνα 16: Δημιουργία 'test' δυνατότητας στο script-βήμα 2

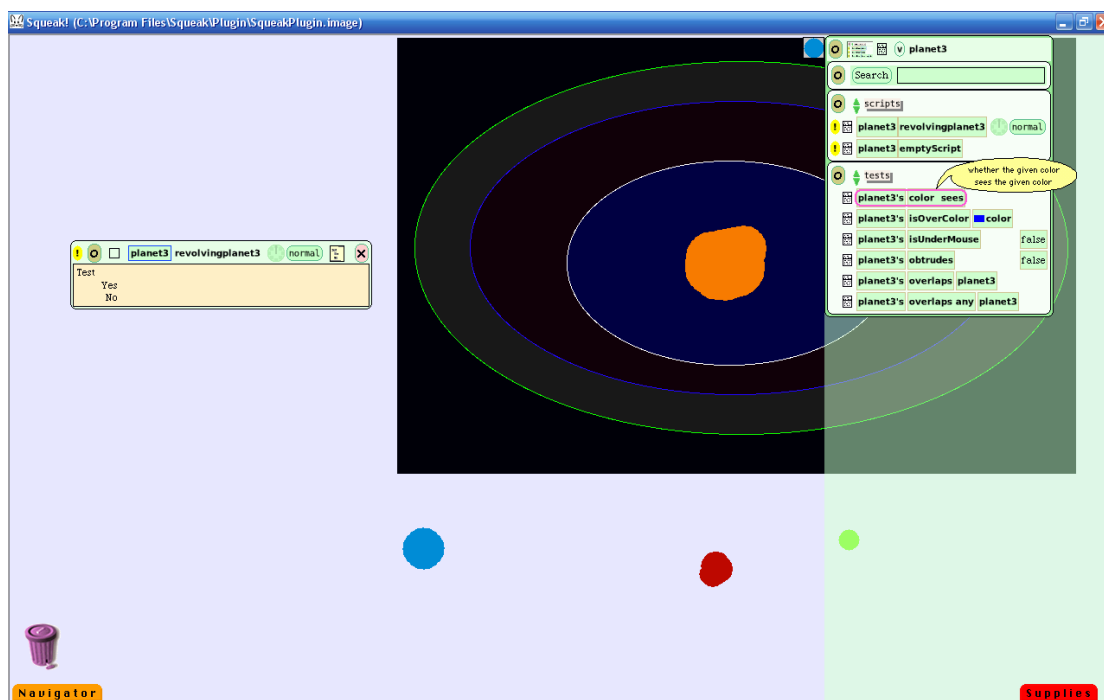
Πηγαίνουμε τώρα πίσω στο εργαλείο 'μάτι' για να αλλάξουμε κατηγορία και να βρούμε την κατηγορία 'tests'(Εικόνα 17).



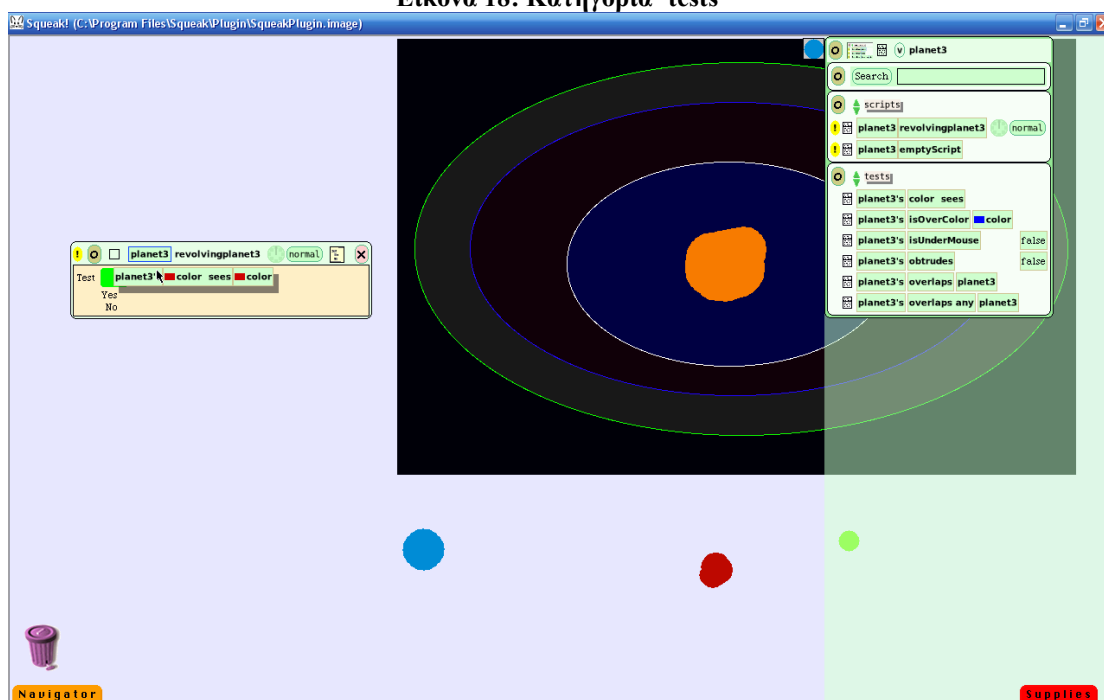
Εικόνα 17: Αλλάζοντας κατηγορία από 'basic' σε 'tests'

Στην κατηγορία αυτή τραβάμε τη γραμμή που λέει 'color sees'(Εικόνα 18) και τη τοποθετούμε δίπλα στη λέξη 'Test' του script 'revolvingplanet3'(Εικόνα 19)(Εικόνα 20).

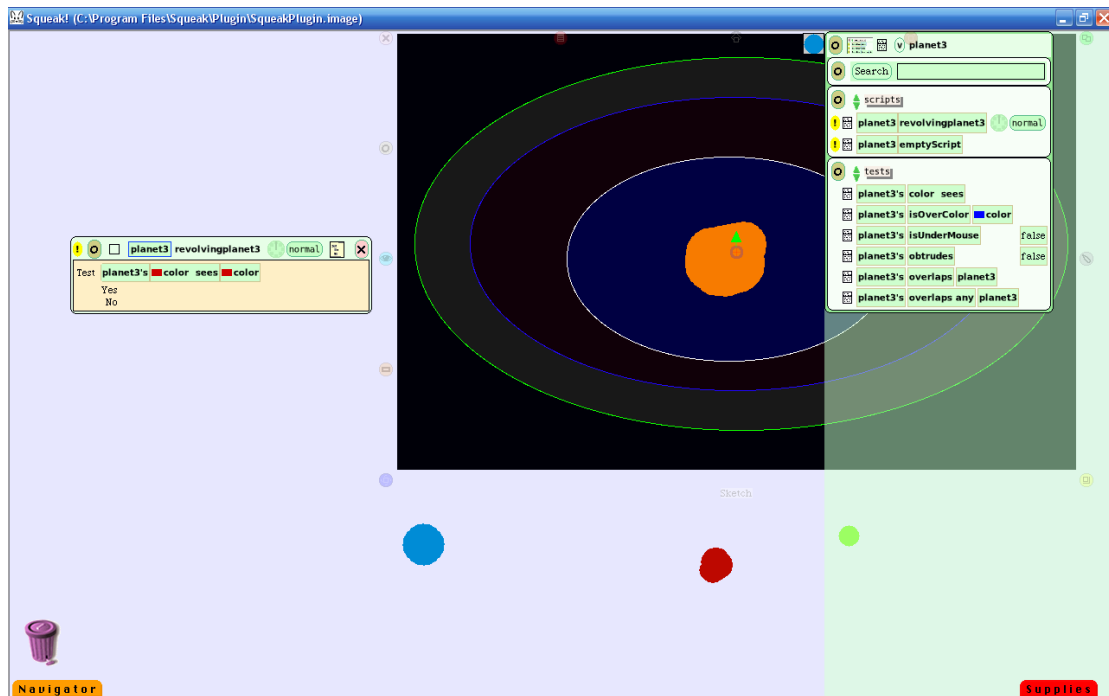
## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες



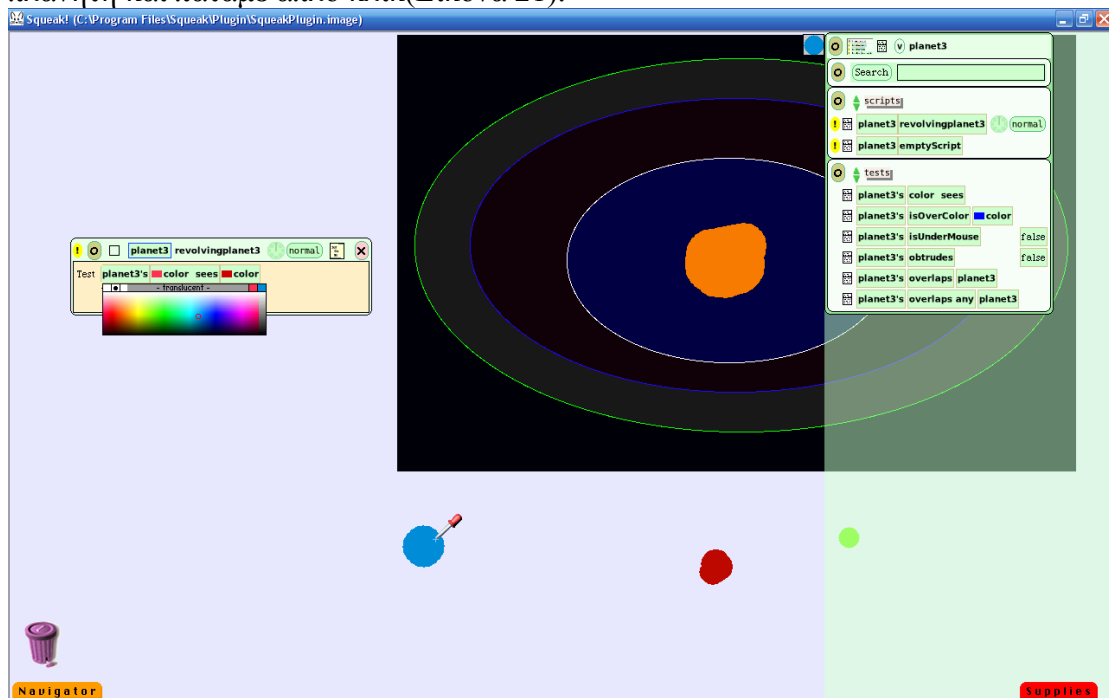
Εικόνα 18: Κατηγορία 'tests'



Εικόνα 19: Τοποθέτηση της γραμμής 'color sees' στο script 'revolvingplanet3' βήμα 1

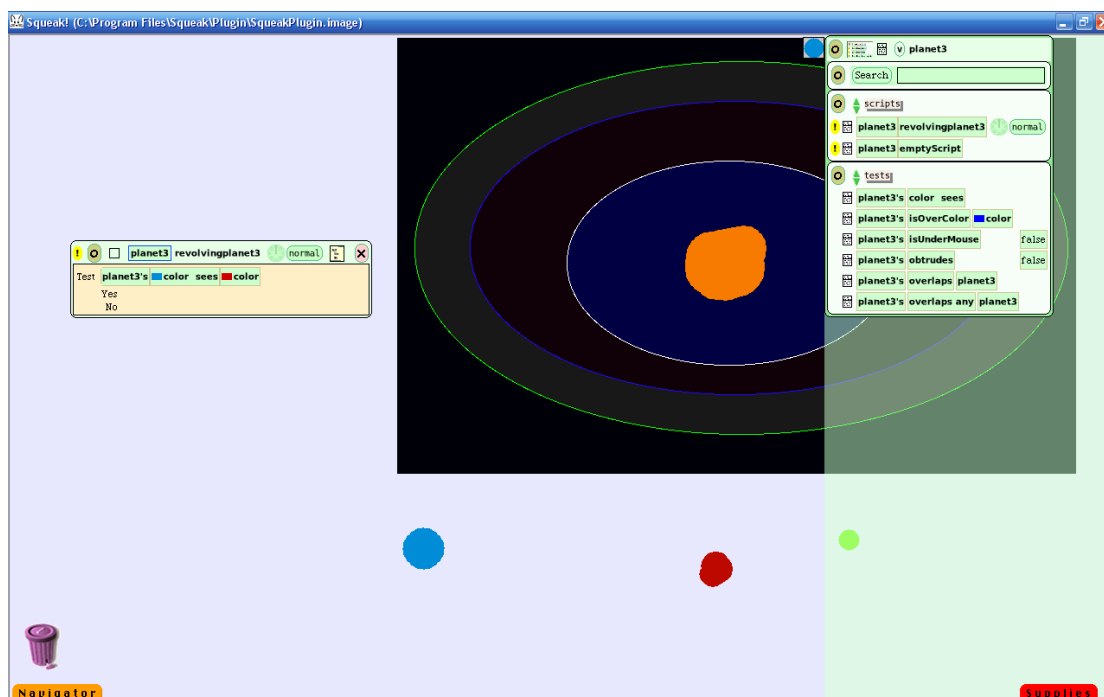


**Εικόνα 20: Τοποθέτηση της γραμμής 'color sees' στο script 'revolvingplanet3' βήμα 2**  
Κατόπιν πατάμε το πρώτο κόκκινο κουτί της γραμμής 'Test' του script και μας εμφανίζεται ένας αριθμός από χρώματα. Πηγαίνουμε με το σταγονόμετρο στον μπλε πλανήτη και πατάμε απλό κλικ(Εικόνα 21).



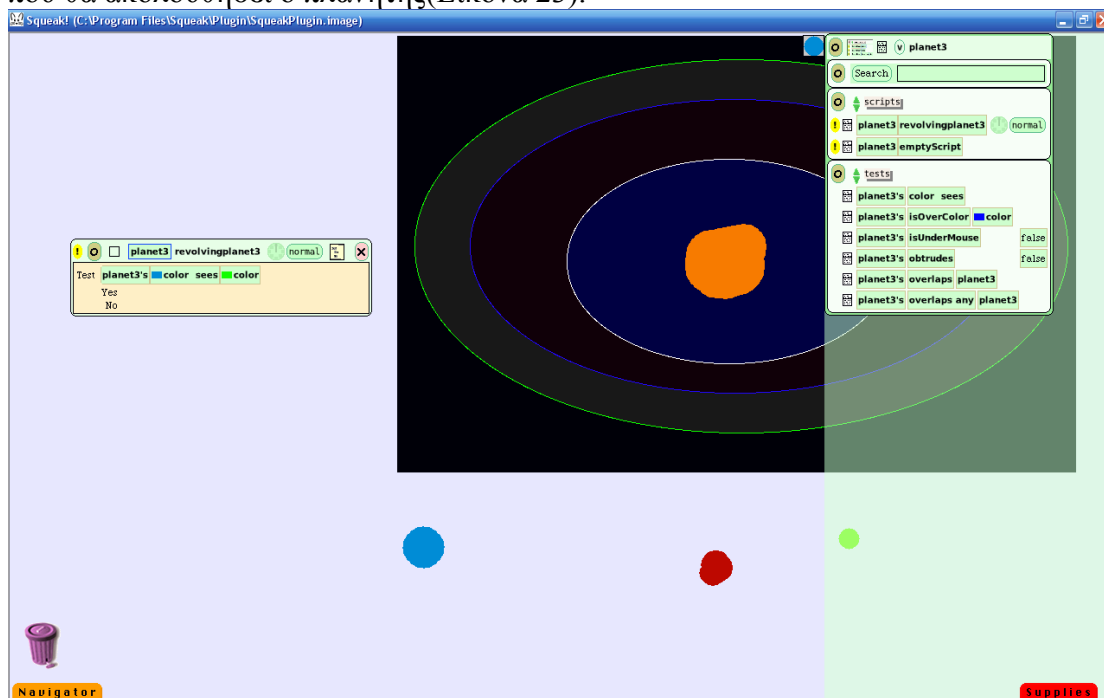
**Εικόνα 21: Χρήση του εργαλείου σταγονόμετρο βήμα 1**  
Το αποτέλεσμα φαίνεται στην Εικόνα 22 και είναι φανερό ότι του κουτί πήρε το χρώμα του πλανήτη.

## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες



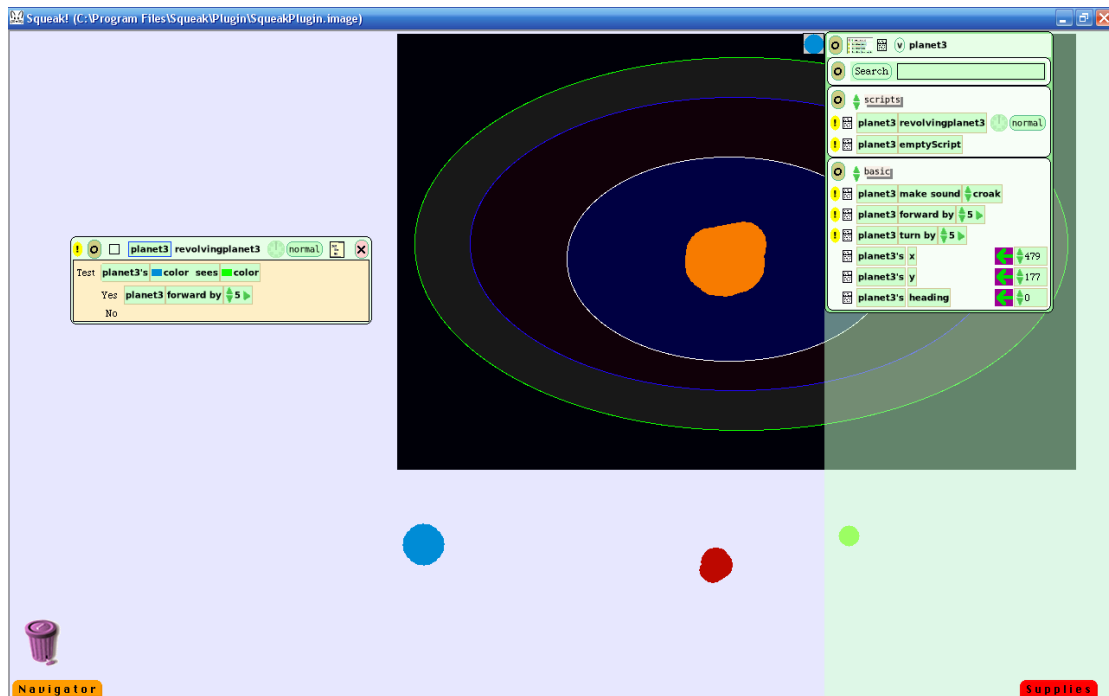
Εικόνα 22: Χρήση του εργαλείου σταγονόμετρο βήμα 2

Στο επόμενο κόκκινο κουτί της γραμμής επιλέγουμε ως χρώμα το χρώμα της τροχιάς που θα ακολουθήσει ο πλανήτης(Εικόνα 23).



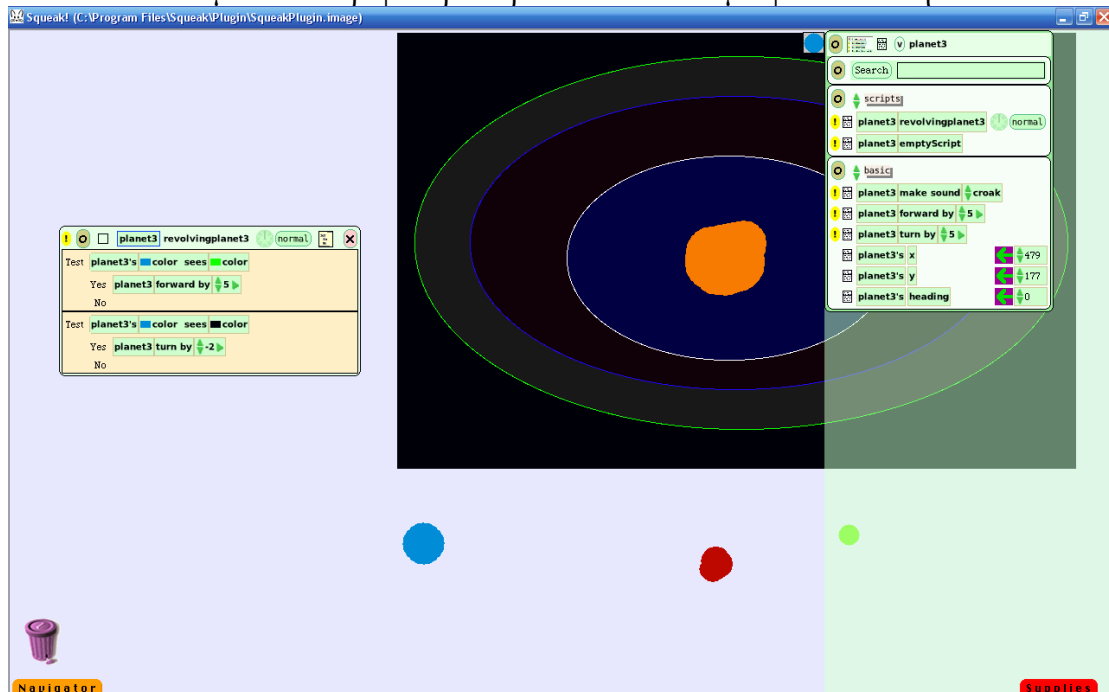
Εικόνα 23: Χρήση του εργαλείου σταγονόμετρο βήμα 3

Στο εργαλείο 'μάτι' αλλάζουμε κατηγορία και πηγαίνουμε πάλι στην κατηγορία 'basic'. Εκεί τραβάμε τη γραμμή 'forward by' και την τοποθετούμε δίπλα στη λέξη 'Yes' που υπάρχει στο script(Εικόνα 24).



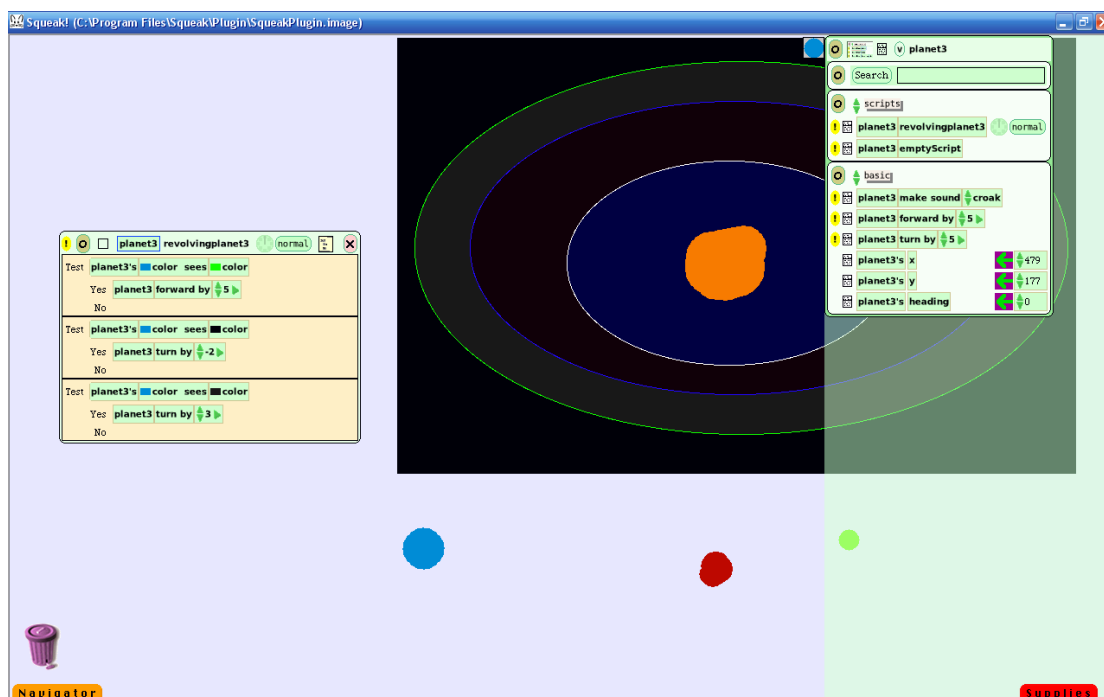
**Εικόνα 24: Τοποθέτηση της γραμμής 'forward by' στο script**

Τώρα τραβάμε άλλο ένα 'test' και το τοποθετούμε ακριβώς από κάτω από το πρώτο 'test' του script. Κάνουμε ακριβώς ότι και πριν μόνο που στο δεύτερο κουτί του χρώματος θα βάλουμε ως χρώμα το χρώμα του φόντου. Επίσης, αντί να τραβήξουμε τη γραμμή 'forward by' τραβάμε τη γραμμή 'turn by'. Την τοποθετούμε δίπλα από τη λέξη 'Yes' και αλλάζουμε τον αριθμό ώστε να είναι αρνητικός, διότι ο αρνητικός κάνει το αντικείμενο να στρίψει αριστερά. Το αποτέλεσμα φαίνεται στην Εικόνα 25.



**Εικόνα 25: Δημιουργία δεύτερου test στο script**

Τέλος προσθέτουμε ένα ακόμη 'test' στο script κάνοντας ακριβώς ότι και πριν. Η διαφορά είναι ότι στο δεύτερο κουτί χρώματος θα πρέπει να βάλουμε το χρώμα του αντικειμένου 'eclipse' στην τροχιά του οποίου θα κινείται ο μπλε πλανήτης. Επίσης, θα προσθέτουμε ακόμη μια φορά τη γραμμή 'turn by' και θα έχουμε θετικό αριθμό. Το αποτέλεσμα φαίνεται στην Εικόνα 26.



Εικόνα 26: Δημιουργία τρίτου test στο script

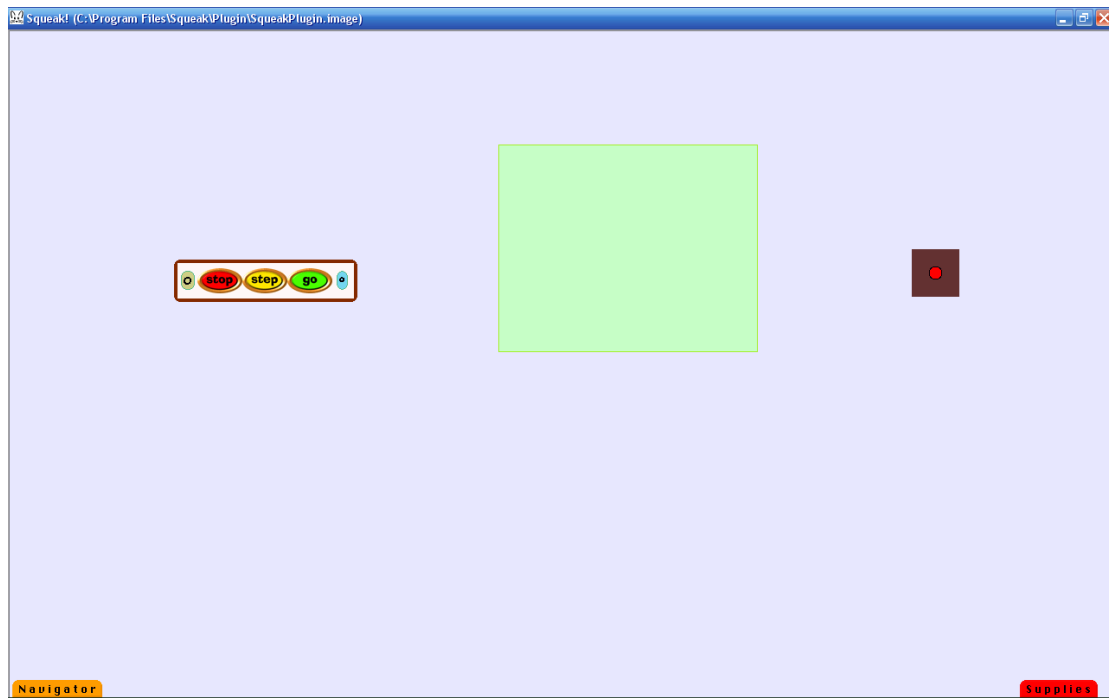
Έτσι αλλάζοντας την επιλογή του script από normal σε ticking θα αρχίσει ο πλανήτης να περιστρέφεται. Κάνουμε ακριβώς το ίδιο και για τους άλλους τους πλανήτες με τη διαφορά ότι στο δεύτερο 'test' στο δεύτερο χρωματικό κουτί δεν θα μπει το χρώμα του φόντου αλλά το χρώμα που υπάρχει από την εξωτερική πλευρά της τροχιάς. Τέλος μια καλή πρόταση είναι σε κάθε πλανήτη που φτιάχνεται να περιέχεται και μια μικρή κουκίδα σε μια άκρη το χρώμα της οποίας θα μπαίνει ως χρώμα πλανήτη στο πρώτο κουτί χρώματος του εκάστοτε script.

### 2.2. Προσγείωση στη σελήνη

Στο παράδειγμα που ακολουθεί θα γίνει κατασκευή δύο διαστημοπλοίων, θα σχεδιαστεί ο χώρος της σελήνης και θα γίνει χρήση του αντικειμένου 'joystick' για την κίνησή τους.

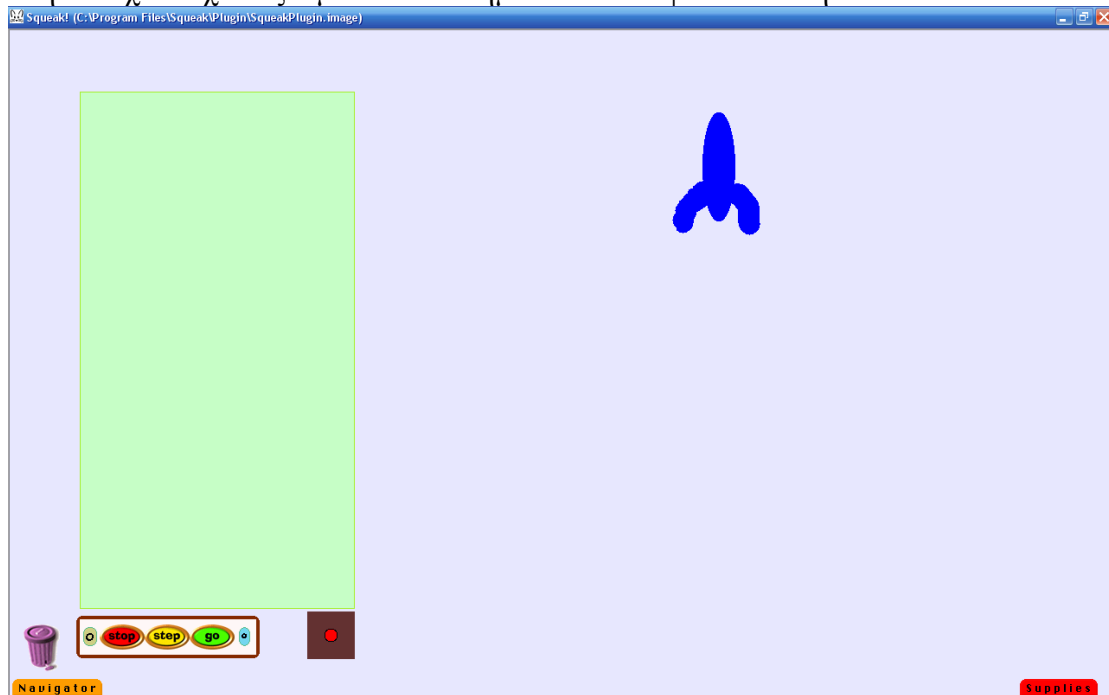
Ξεκινώντας ανοίγουμε την καρτέλα 'Supplies' και τραβάμε ένα αντικείμενο 'Playfield', ένα αντικείμενο 'joystick' και ένα αντικείμενο 'Stop/Step/Go' όπως φαίνεται στην Εικόνα 27.

## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες



**Εικόνα 27: Αντικείμενα 'playfield', 'joystick' και 'stop/step/go'**

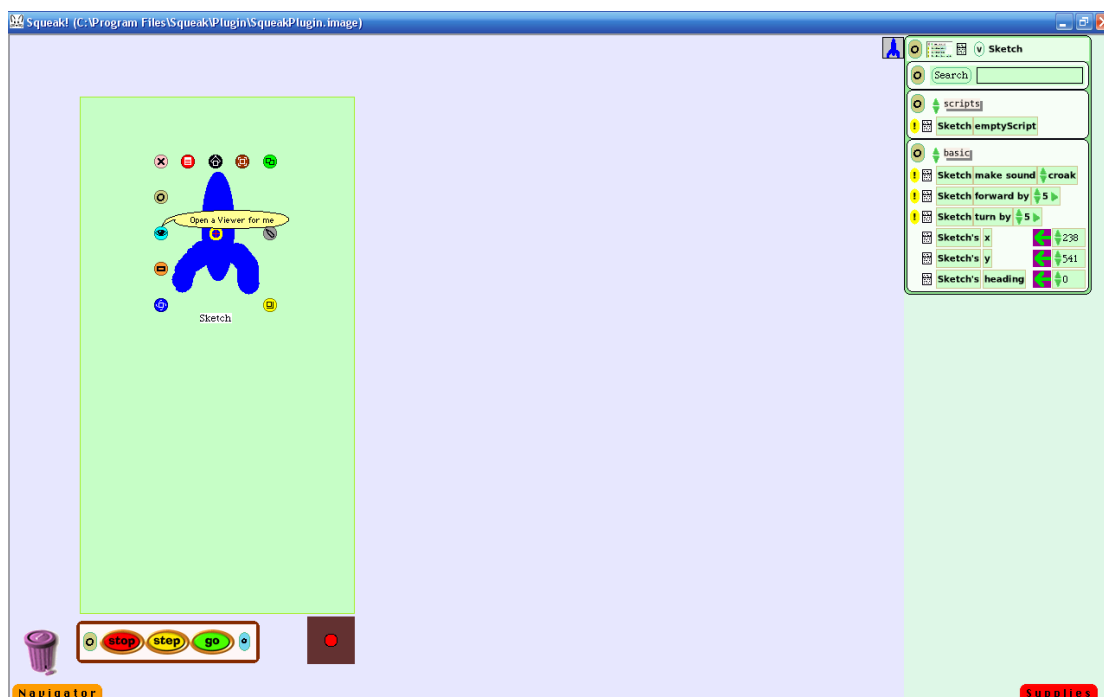
Στη συνέχεια σχεδιάζουμε ένα διαστημόπλοιο που φαίνεται στην Εικόνα 28.



**Εικόνα 28: Κατασκευή του διαστημοπλοίου**

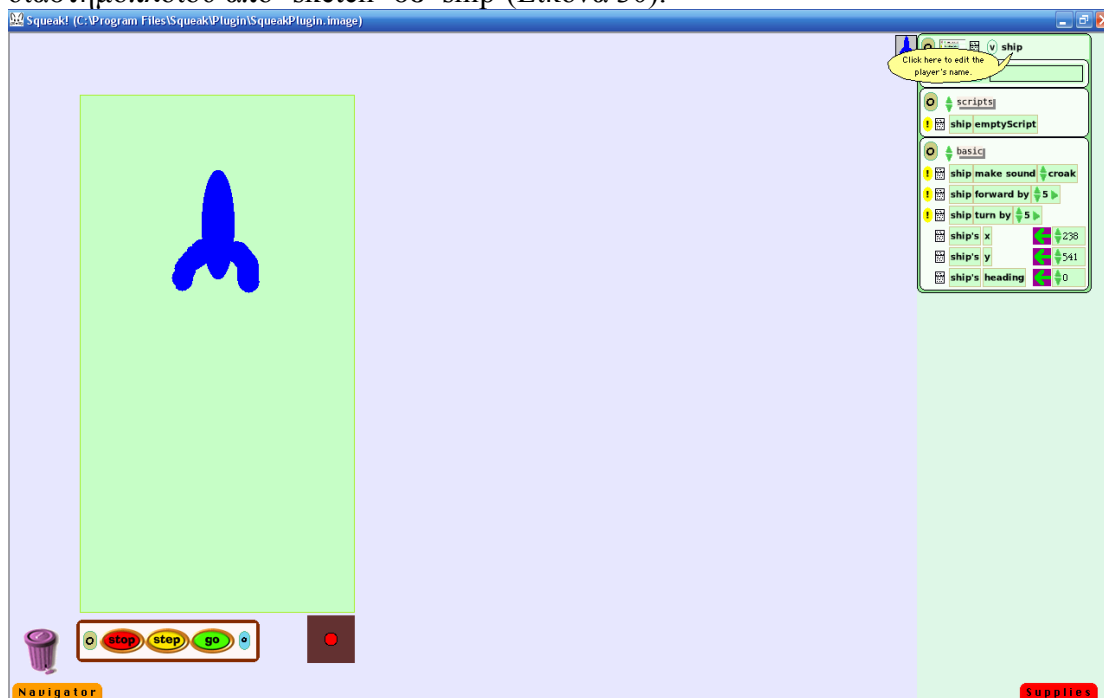
Μετά την κατασκευή, τοποθετούμε το διαστημόπλοιο στο 'playfield'. Το επόμενο βήμα είναι η χρήση του εργαλείου 'μάτι' (Εικόνα 29).

## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες



Εικόνα 29: Άνοιγμα του εργαλείου ‘μάτι’ του αντικειμένου-διαστημοπλοίου

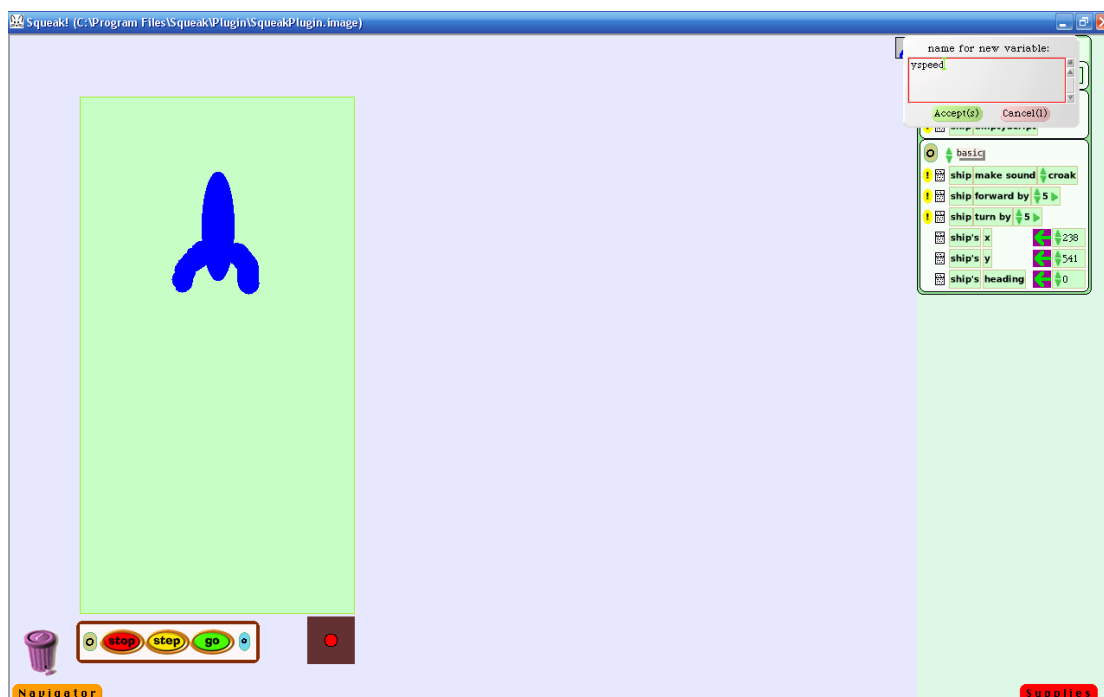
Το πρώτο που κάνουμε είναι να αλλάξουμε το όνομα του αντικειμένου-διαστημοπλοίου από ‘sketch’ σε ‘ship’ (Εικόνα 30).



Εικόνα 30: Αλλαγή του ονόματος του αντικειμένου-διαστημοπλοίου

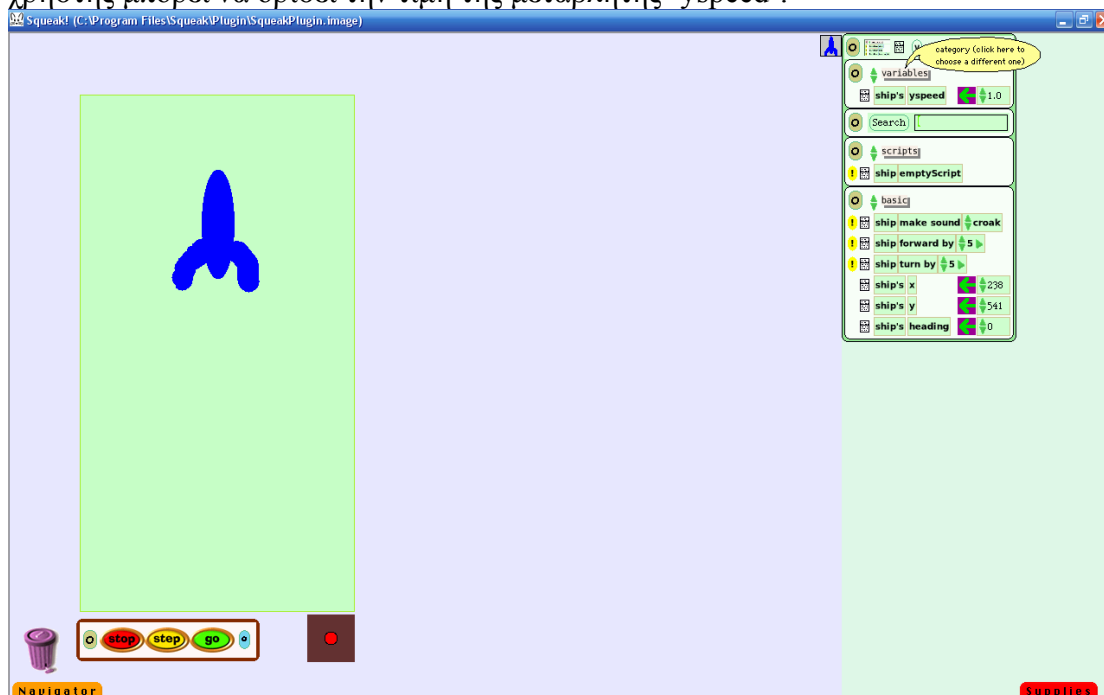
Επίσης πρέπει να δημιουργηθεί μια μεταβλητή(variable) για την ταχύτητα του διαστημοπλοίου κατά μήκος του Y άξονα. Πηγαίνουμε στην ήδη ανοιγμένη καρτέλα του εργαλείου ‘μάτι’, βρίσκουμε το κουμπί ‘v’ που βρίσκεται στην κορυφή της καρτέλας και το πατάμε. Στο κουτί που εμφανίζεται, γράφουμε τη λέξη ‘yspeed’ και πατάμε ‘Accept(s)’ (Εικόνα 31).





**Εικόνα 31: Δημιουργία καινούργιας μεταβλητής για το αντικείμενο-διαστημόπλοιο**

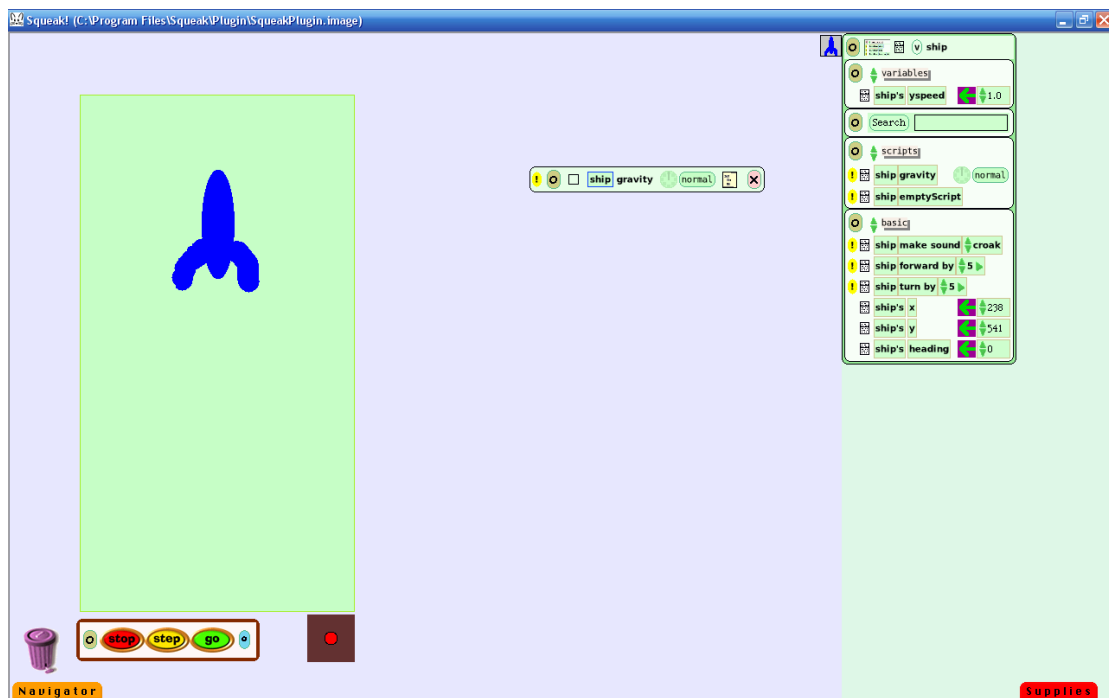
Με τον τρόπο αυτό δημιουργούμε μια καινούργια μεταβλητή με το όνομα 'yspeed' του αντικειμένου-διαστημόπλοιο. Αυτό φαίνεται στην κατηγορία 'variables' του εργαλείου 'μάτι' στην οποία υπάρχει η γραμμή 'ship's yspeed'. Στην γραμμή αυτή, ο χρήστης μπορεί να ορίσει την τιμή της μεταβλητής 'yspeed'.



**Εικόνα 32: Η κατηγορία 'variables' του αντικειμένου-διαστημόπλοιο**

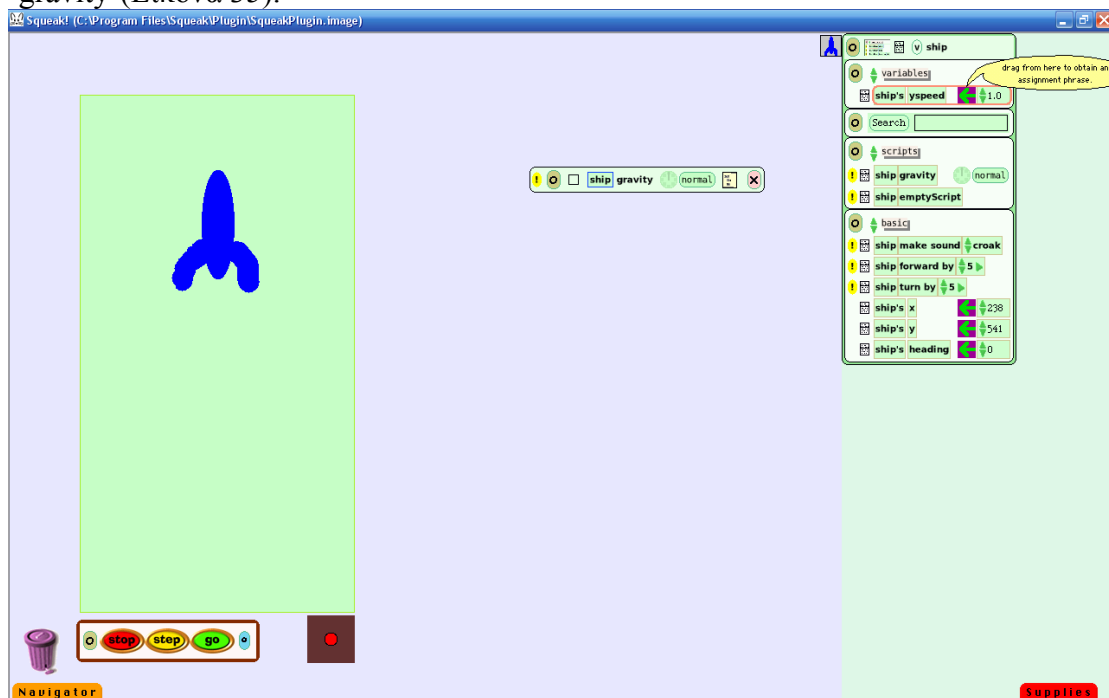
Το επόμενο βήμα είναι η κατασκευή ορισμένων script για να μπει το διαστημόπλοιο σε κίνηση. Θα ξεκινήσουμε με το script 'gravity'. Βγάζουμε ένα 'empty script' από το εργαλείο 'μάτι' και το ονομάζουμε 'gravity' (Εικόνα 33).

## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες



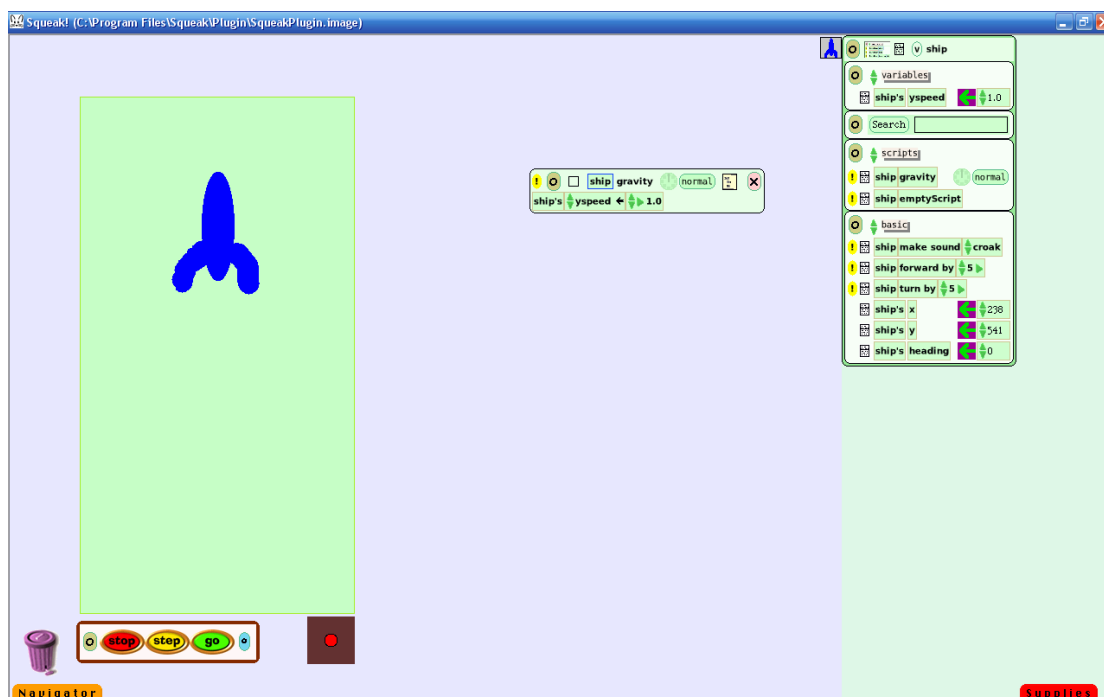
Εικόνα 33: Το script 'gravity'

Στη συνέχεια βγάζουμε μια 'yspeed' γραμμή(Εικόνα 34) και την αφήνουμε στο script 'gravity'(Εικόνα 35).



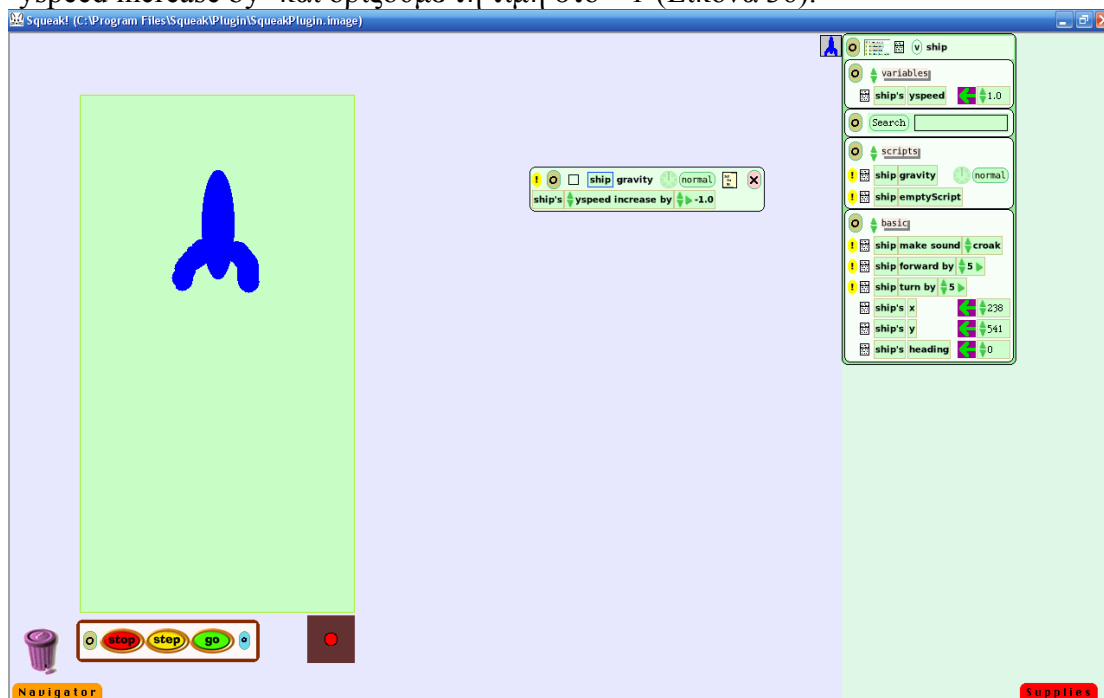
Εικόνα 34: Κατασκευάζοντας το script 'gravity' βήμα 1

## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες



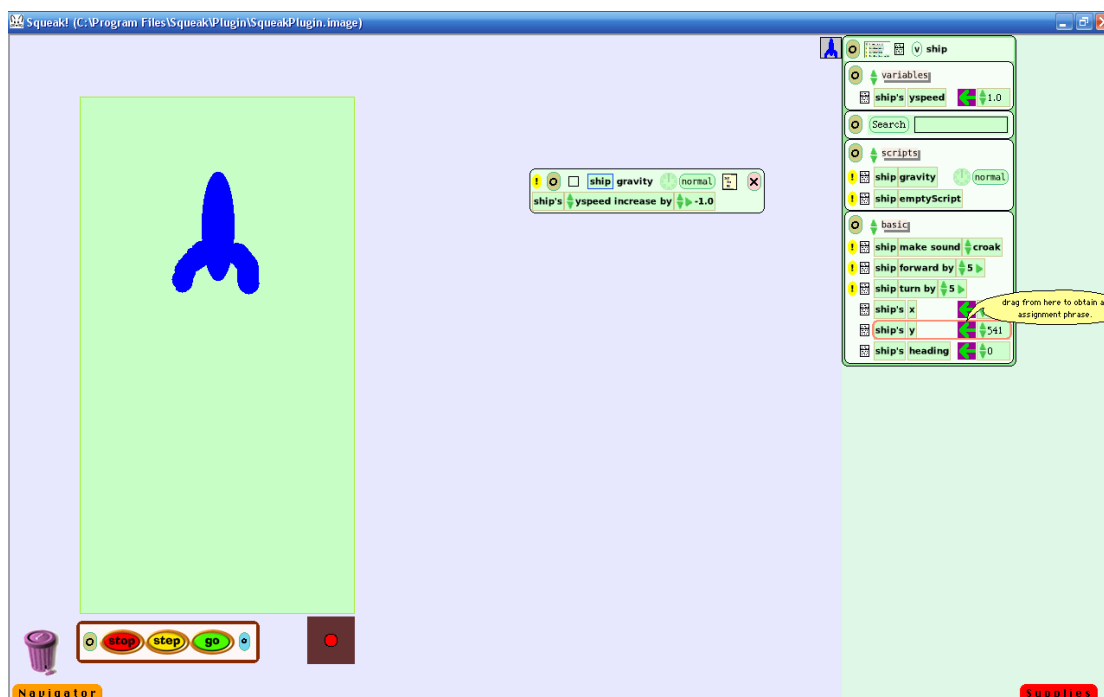
Εικόνα 35: Κατασκευάζοντας το script 'gravity' βήμα 2

Χρησιμοποιώντας τα πράσινα up/down βελάκια στο 'yspeed' επιλέγουμε τη γραμμή 'yspeed increase by' και ορίζουμε τη τιμή στο '-1' (Εικόνα 36).

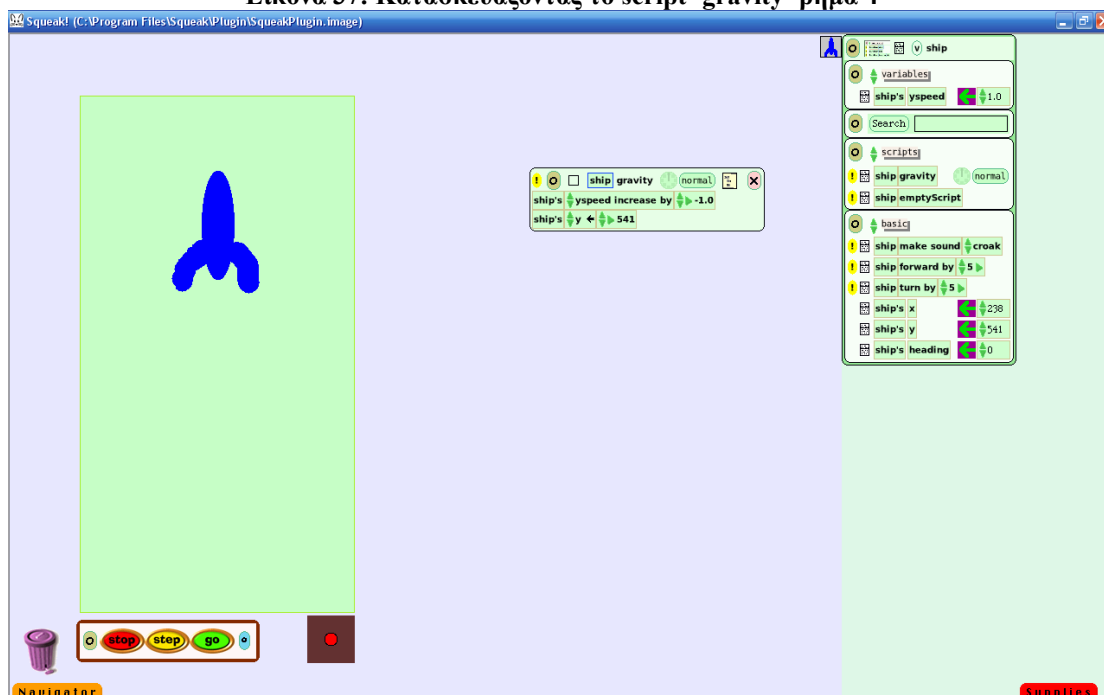


Εικόνα 36: Κατασκευάζοντας το script 'gravity' βήμα 3

Πηγαίνουμε τώρα και τραβάμε μια γραμμή 'ship's y' (ολόκληρη) (Εικόνα 37) και την αφήνουμε στη δεύτερη γραμμή του script 'gravity' (Εικόνα 38).

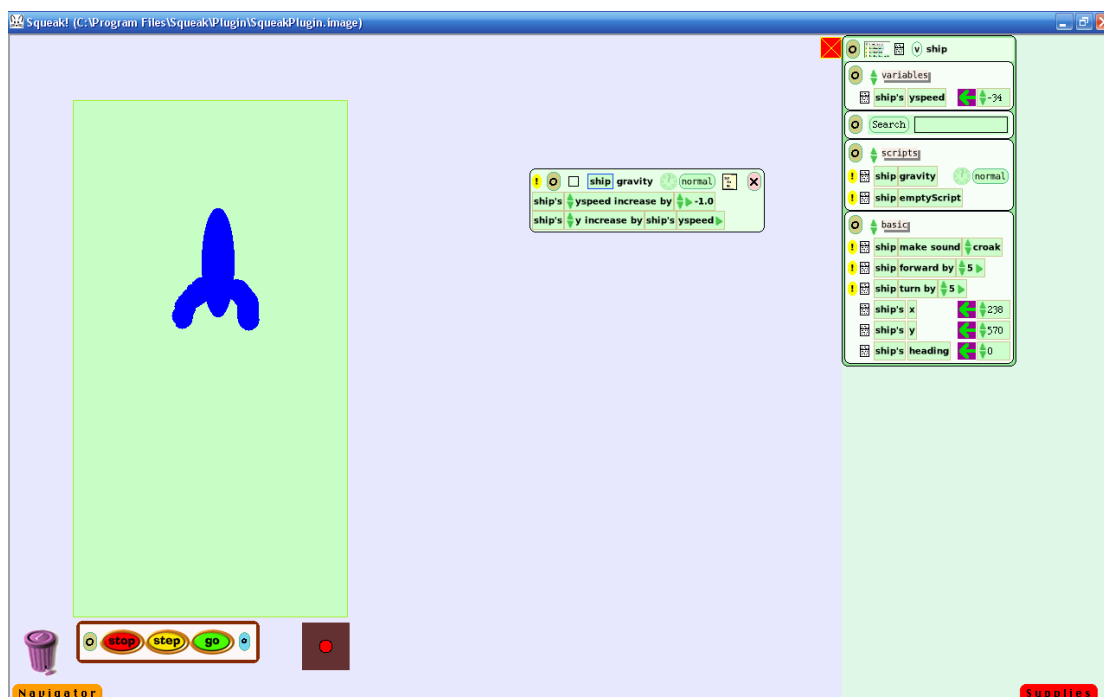


Εικόνα 37: Κατασκευάζοντας το script 'gravity' βήμα 4



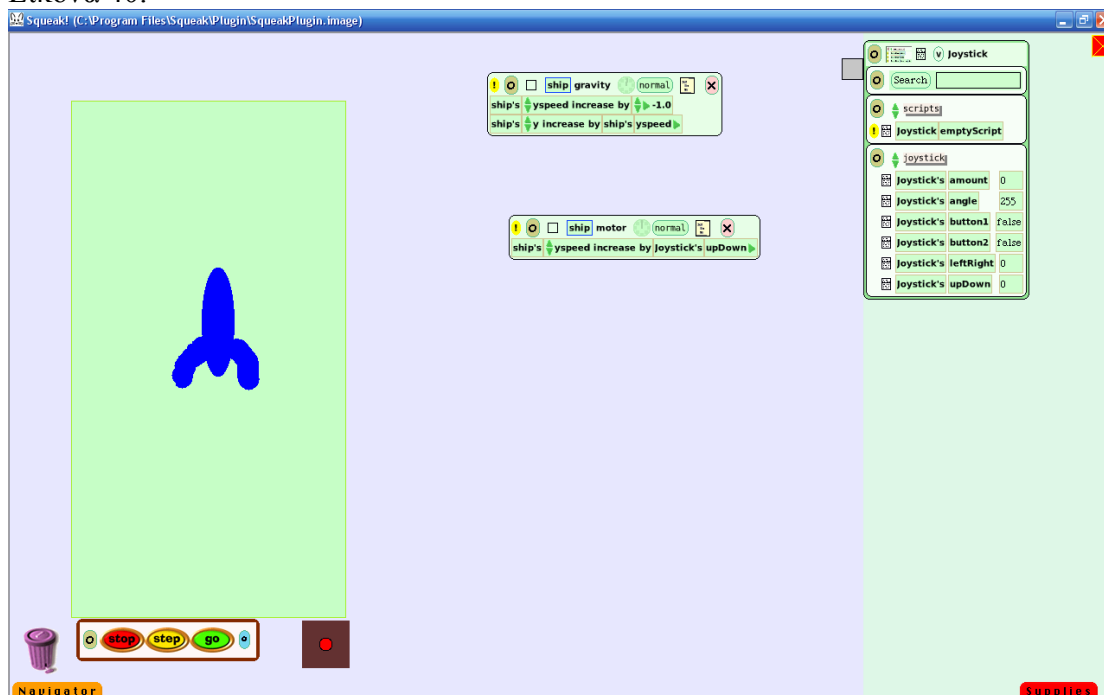
Εικόνα 38: Κατασκευάζοντας το script 'gravity' βήμα 5

Τη δεύτερη αυτή γραμμή την αλλάζουμε σε 'y increase by'. Στη συνέχεια πηγαίνουμε στο εργαλείο 'μάτι' και παίρνουμε μια γραμμή 'ship's yspeed', αυτή τη φορά όχι ολόκληρη τη γραμμή, και την αφήνουμε μέσα στη τιμή του 'yspeed increase by'. Το αποτέλεσμα φαίνεται στην Εικόνα 39.



**Εικόνα 39: Κατασκευάζοντας το script 'gravity' βήμα 6**

Τώρα θα φτιάξουμε ένα νέο script που θα το ονομάσουμε 'motor'. Πηγαίνουμε στο εργαλείο 'μάτι' του διαστημοπλοίου και βγάζουμε ένα 'empty script'. Στη συνέχεια βγάζουμε μια ολόκληρη γραμμή 'ship's yspeed' και την αφήνουμε στο script 'motor'. Τη γραμμή αυτή την αλλάζουμε σε 'yspeed increase by'. Στη συνέχεια ανοίγουμε ένα εργαλείο 'μάτι' για το αντικείμενο 'joystick' και επιλέγουμε την κατηγορία 'joystick'. Από την κατηγορία αυτή, τραβάμε τη γραμμή 'Joystick's upDown' και την αφήνουμε στο κουτάκι της τιμής του yspeed στο script 'joystick'. Το αποτέλεσμα φαίνεται στην Εικόνα 40.

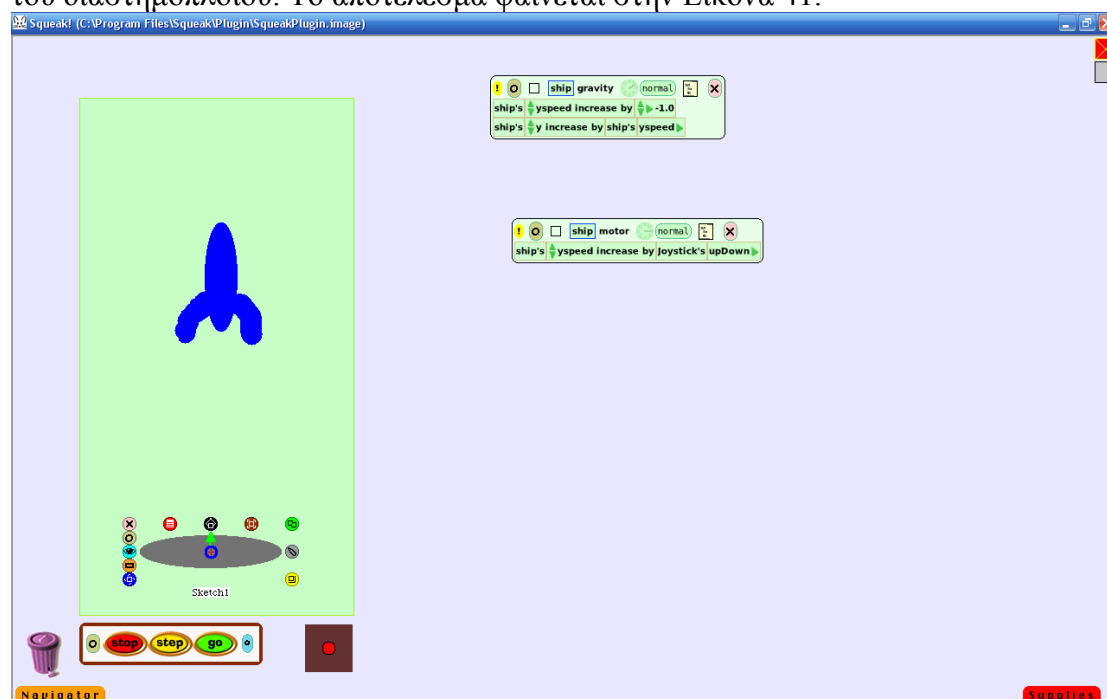


**Εικόνα 40: Το script 'motor'**

Έτσι με το script 'motor' το διαστημόπλοιο απέκτησε μια μηχανή(motor) που μπορεί να δουλέψει ενάντια στη βαρύτητα(gravity), να ελέγξει την ταχύτητα της πτώσης του διαστημοπλοίου αλλά και ακόμα να το προωθήσει στο διάστημα.

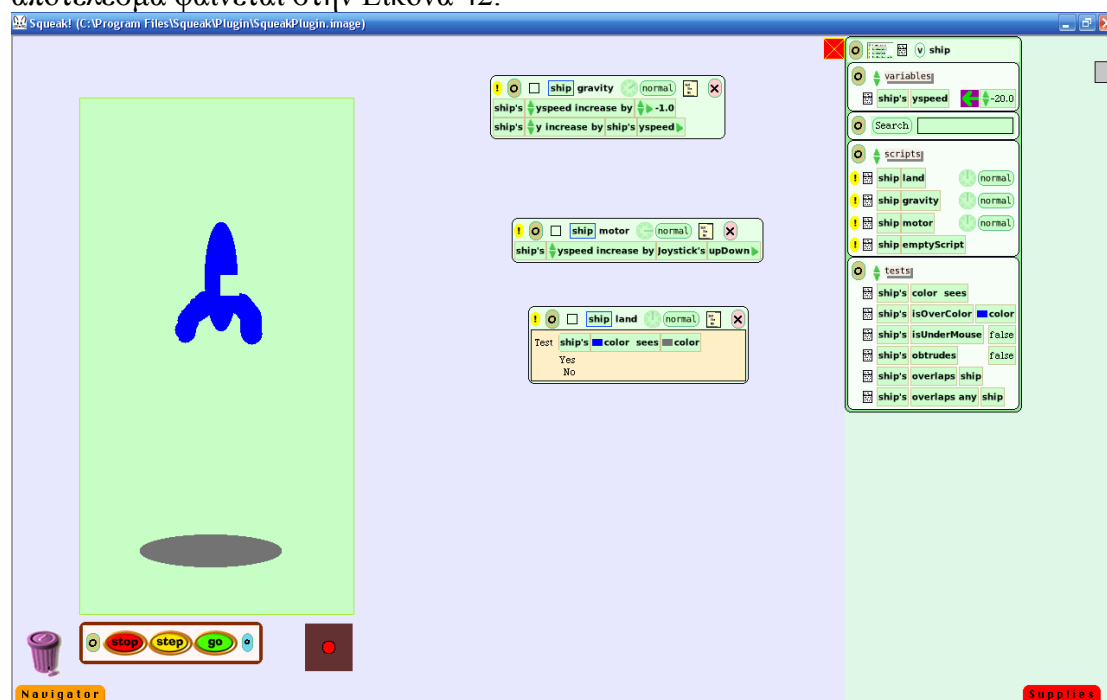
## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες

Στη συνέχεια θα φτιάξουμε ένα σχήμα το οποίο θα αποτελεί το χώρο προσγείωσης του διαστημοπλοίου. Το αποτέλεσμα φαίνεται στην Εικόνα 41.



**Εικόνα 41: Κατασκευή του χώρου προσγείωσης του διαστημοπλοίου**

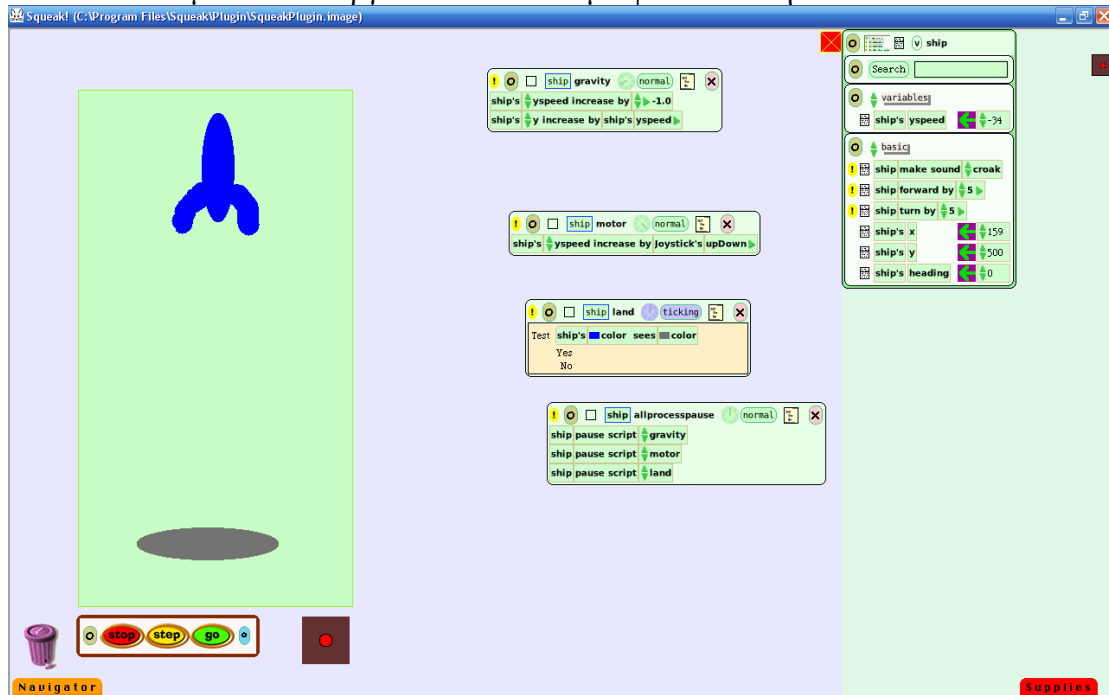
Κατόπιν θα φτιάξουμε ένα script το οποίο θα το ονομάσουμε 'land' (προσγείωση). Το 'empty script' αυτό το παίρνουμε από το εργαλείο 'μάτι' του διαστημοπλοίου. Στη συνέχεια βγάζουμε από το πάνω μέρος του script, ένα test και το αφήνουμε μέσα στο script. Πηγαίνουμε μετά πίσω στο εργαλείο 'μάτι' του διαστημοπλοίου και από την κατηγορία 'tests' παίρνουμε μια γραμμή 'ship's color sees' και τη ρίχνουμε δίπλα στη λέξη 'Test'. Στη συνέχεια βάζουμε στο πρώτο κουτί χρώματος το χρώμα του διαστημοπλοίου και στο δεύτερο κουτί το χρώμα του χώρου προσγείωσης. Το αποτέλεσμα φαίνεται στην Εικόνα 42.



**Εικόνα 42: Το script 'land'**

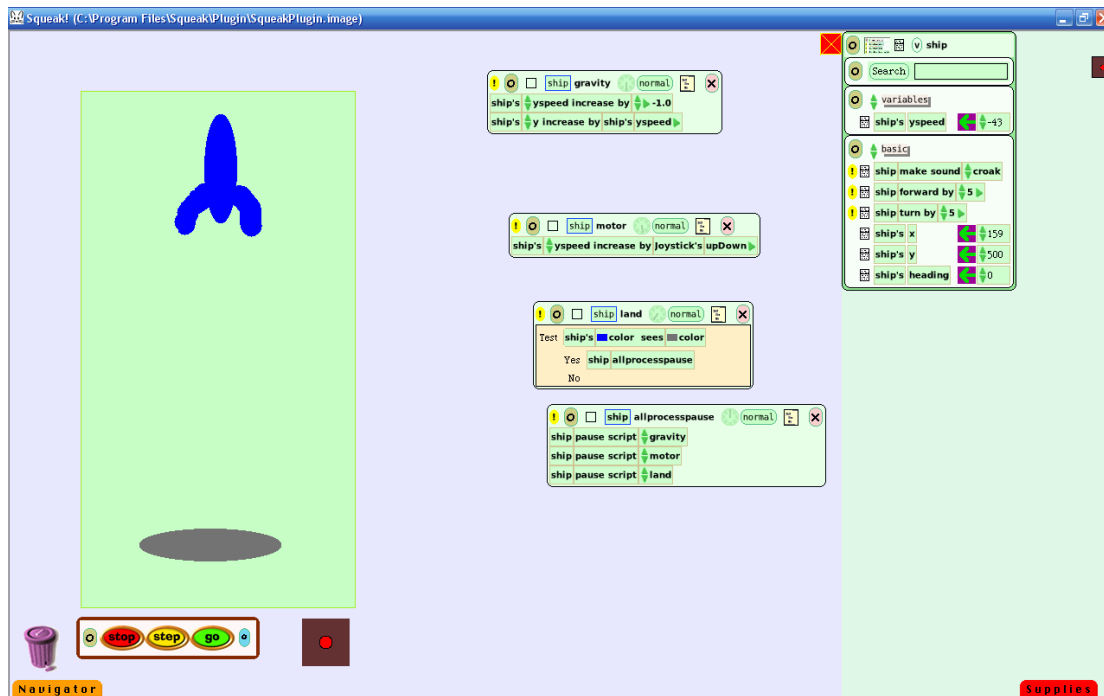
## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες

Όταν προσγειωθεί το διαστημόπλοιο, θα πρέπει να κλείσουν τα πάντα. Πρέπει να κατασκευαστεί ένα script για να γίνει αυτό. Παίρνουμε ένα άδειο script από το εργαλείο ‘μάτι’ του διαστημοπλοίου και το ονομάζουμε ‘allprocesspause’. Στη συνέχεια αλλάζουμε κατηγορία στο εργαλείο ‘μάτι’ και πηγαίνουμε στην κατηγορία ‘scripting’. Από εκεί τραβάμε 3 ‘pause script’ και τα τοποθετούμε στο script ‘allprocesspause’. Χρησιμοποιώντας τα πράσινα βελάκια των γραμμών, βάζουμε στη μια γραμμή τη λέξη ‘gravity’, στην επόμενη γραμμή τη λέξη ‘motor’ και στη τελευταία γραμμή τη λέξη ‘land’. Το script αυτό θα σταματήσει τα τρία βασικά script, όταν το θέσουμε σε λειτουργία. Το αποτέλεσμα φαίνεται στην Εικόνα 43.



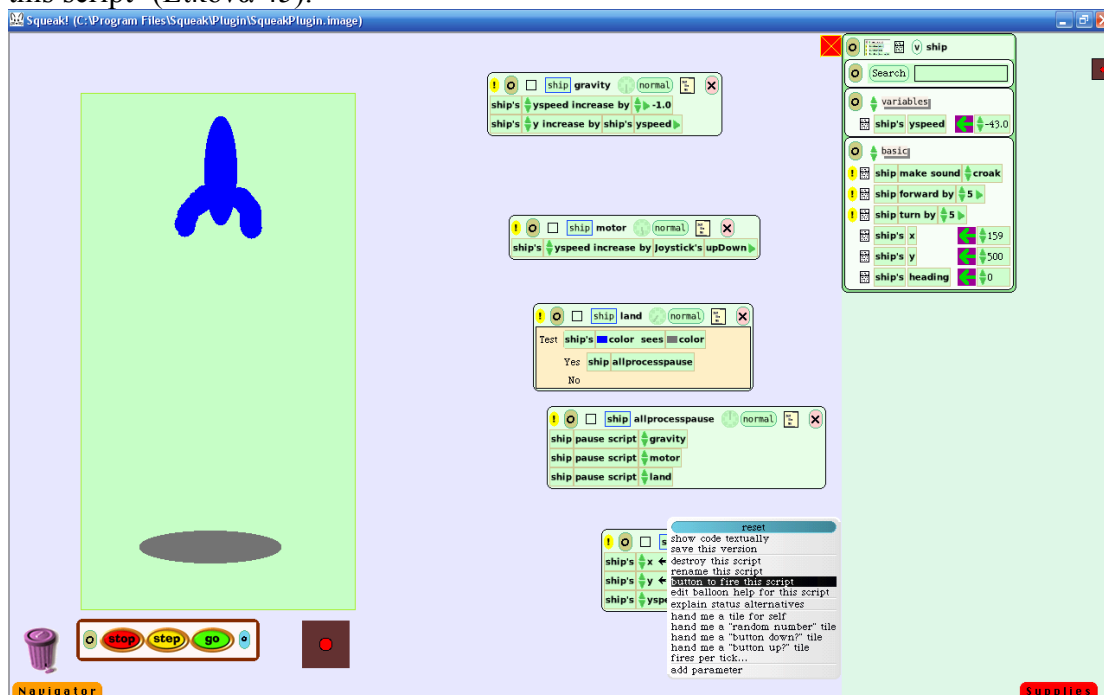
Εικόνα 43: Το script ‘allprocesspause’

Θα γυρίσουμε για λίγο στο script ‘land’. Πηγαίνουμε στο εργαλείο ‘μάτι’ του διαστημοπλοίου και από την κατηγορία ‘scripts’ παίρνουμε μια γραμμή ‘allprocesspause’. Τη γραμμή αυτή την αφήνουμε δίπλα στη λέξη ‘Yes’ του script ‘land’. Το αποτέλεσμα φαίνεται στην Εικόνα 44. Με τον τρόπο αυτό, όταν το διαστημόπλοιο αγγίζει τον χώρο προσγείωσης(γκρι έλλειψη), όλα τα βασικά scripts θα σταματήσουν.



Εικόνα 44: Επεκτείνοντας τη λειτουργικότητα του script 'land'

Στη συνέχεια θα κατασκευάσουμε ένα script που θα το ονομάσουμε 'reset'. Πηγαίνουμε στο εργαλείο 'μάτι' και ανοίγουμε την κατηγορία 'scripts'. Από εκεί τραβάμε ένα 'empty script' και το ονομάζουμε 'reset'. Στη συνέχεια πηγαίνουμε στην κατηγορία 'basic' και τραβάμε ολόκληρες τις γραμμές 'ship's x' και 'ship's y' και τις τοποθετούμε στο script 'reset'. Κατόπιν πηγαίνουμε στην κατηγορία 'variables' και παίρνουμε ολόκληρη τη γραμμή 'ship's yspeed' την οποία στη συνέχεια τοποθετούμε στο script 'reset'. Αυτή τη μεταβλητή, τη μεταβλητή 'yspeed' θα την ορίσουμε με τιμή '0'. Το επόμενο βήμα είναι να πατήσουμε στο script 'reset' τη λέξη 'ship' και να πάρουμε ένα μενού. Στο μενού που θα εμφανιστεί πατάμε τη γραμμή 'button to fire this script' (Εικόνα 45).

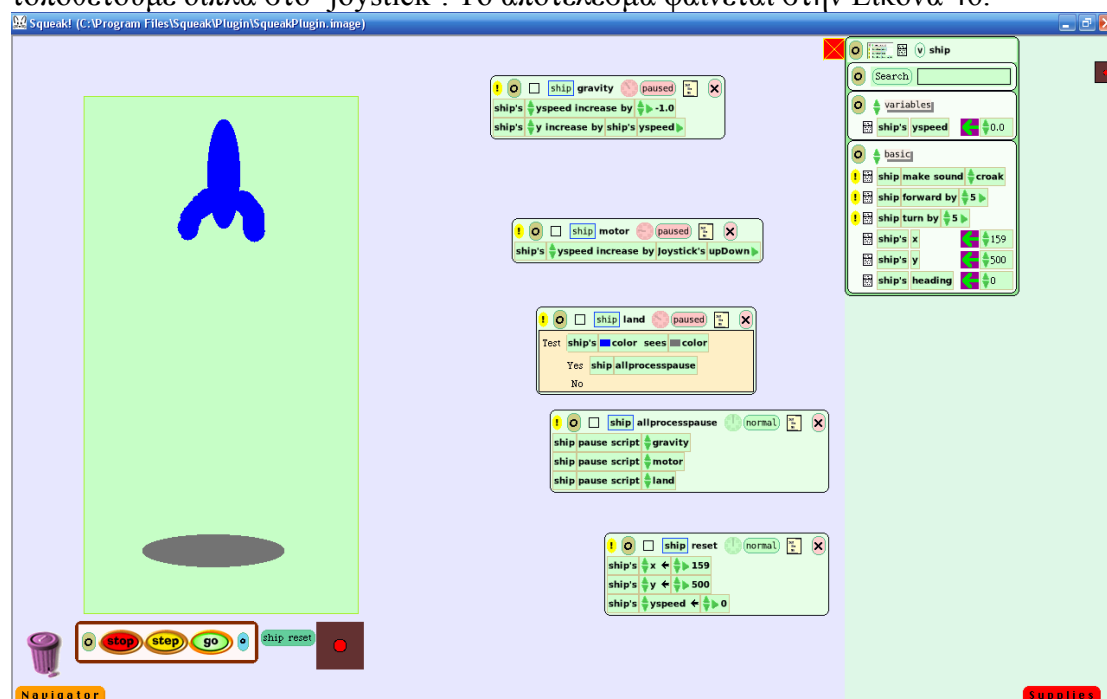


Εικόνα 45: Η επιλογή 'button to fire this script' στο menu ενός script



## Παραδείγματα Χρήσης για Προχωρημένους Χρήστες

Το αποτέλεσμα είναι ένα κουμπί το οποίο ονομάζεται 'ship reset' και το οποίο το τοποθετούμε δίπλα στο 'joystick'. Το αποτέλεσμα φαίνεται στην Εικόνα 46.



Εικόνα 46: Το τελικό πρόγραμμα

Τέλος να αναφέρουμε ότι όταν πατάμε το κουμπί 'go', το διαστημόπλοιο θα ξεκινήσει να πέφτει γρήγορα αυξάνοντας την ταχύτητά του. Μπορούμε να χρησιμοποιήσουμε το joystick για να ελέγξουμε την ταχύτητα της πτώσης. Όταν το διαστημόπλοιο αγγίζει το χώρο προσγείωσης, τα πάντα θα σταματήσουν.

### 3. Σχολικά παραδείγματα

#### 3.1. Κατάσταση της ύλης

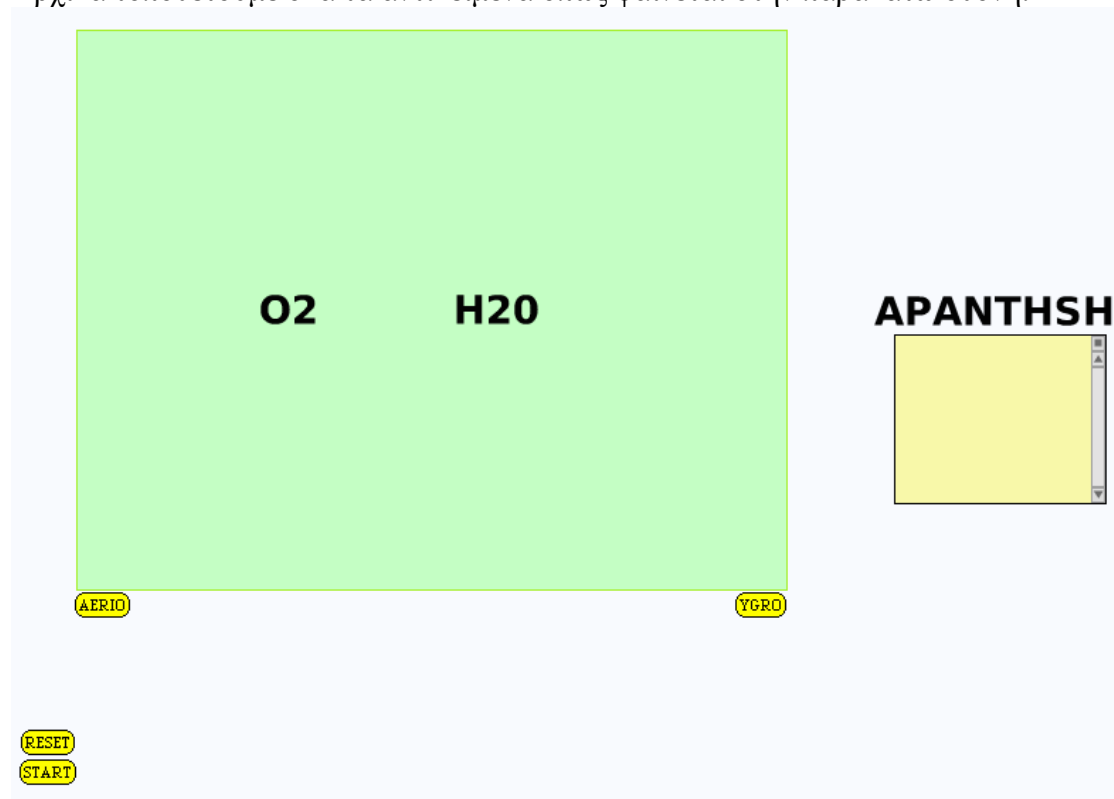
Στο συγκεκριμένο παράδειγμα θα δημιουργήσουμε ένα παιχνίδι σαν το γνωστό TETRIS. Στο παράδειγμα θα δημιουργήσουμε ένα περιβάλλον για να «παίζει» το παιχνίδι. Η διαδικασία είναι η εξής: Από το πάνω μέρος του περιβάλλοντος θα πέφτουν αντικείμενα (χημικές ενώσεις) και ο χρήστης θα πρέπει να επιλέξει αν η χημική ένωση που «πέφτει» είναι υγρή ή αέρια σε ΚΣ. Η εφαρμογή θα διαθέτει και ένα παράθυρο όπου θα εμφανίζονται οι απαντήσεις στις επιλογές του. Αν το χημικό στοιχείο φτάσει στο κάτω μέρος του περιβάλλοντος χωρίς να έχει δοθεί απάντηση, τότε θα πρέπει να του εμφανίζεται η σωστή απάντηση. Στο συγκεκριμένο παράδειγμα θα δούμε δύο χημικές ενώσεις  $H_2O$  (νερό) και  $O_2$  (οξυγόνο).

Σύμφωνα με τα παραπάνω τα «υλικά» που θα χρειαστούμε είναι

- Ένα playfield
- Τέσσερα buttons
- Τρία αντικείμενα τύπου TEXT
- Ένα scrolling text

Ας δούμε αναλυτικά σε τι θα χρησιμεύσει το κάθε αντικείμενο. Το playfield θα είναι ο χώρος που θα εκτελείται όλη η διαδικασία (θα πέφτουν από το πάνω της μέρος τα αντικείμενα...). Τα τέσσερα κουμπιά θα είναι τα εξής: δύο θα χρησιμεύσουν για να επιλέγει ο χρήστης αν το αντικείμενο που «πέφτει» είναι υγρό ή αέριο και άλλα δύο θα χρησιμεύσουν στο να γίνεται αρχικοποίηση όλων των τιμών και έναρξη του παιχνιδιού. Τα αντικείμενα τύπου text θα περιέχουν το όνομα της ένωσης και μία πινακίδα με την ένδειξη ΑΠΑΝΤΗΣΗ για να γνωρίζουν οι χρήστες που θα περιμένουν τις απαντήσεις τους. Τέλος μέσα στο scrolling text θα εμφανίζονται οι απαντήσεις του προγράμματος στις επιλογές του χρήστη.

Αρχικά τοποθετούμε όλα τα αντικείμενα όπως φαίνεται στην παρακάτω οθόνη.

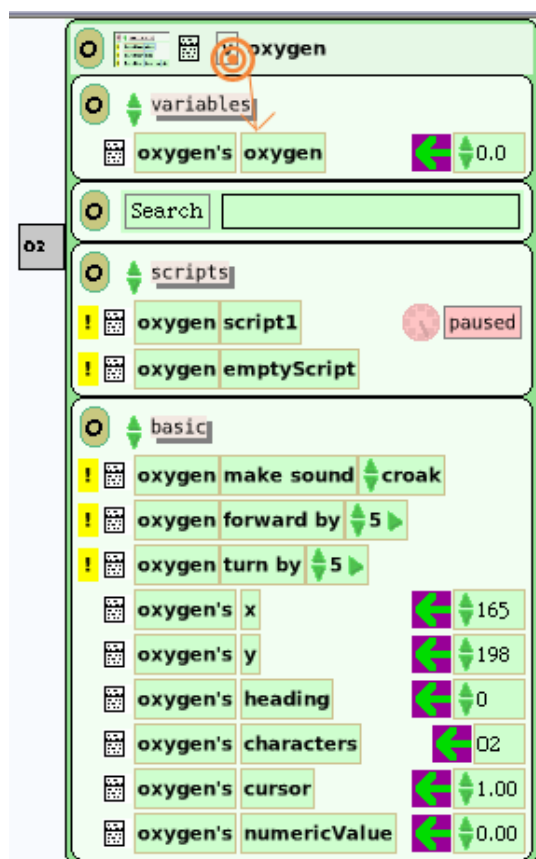


Εικόνα 47: όλα τα αντικείμενα που θα χρειαστούν

Στην εικόνα βλέπουμε τα τρία αντικείμενα τύπου TEXT ( $O_2$  = Οξυγόνο,  $H_2O$  = Νερό και APANTHSH), το playfield (με το πράσινο background, τα τέσσερα κουμπιά (AERIO, YGRO, RESET και START) καθώς και το scrolling text στα δεξιά. Αρχικά θα πρέπει να εξετάσουμε το σενάριο μας προκειμένου να είναι πιο εύκολη η υλοποίηση του συστήματος. Αυτό που θέλουμε να πραγματοποιήσουμε είναι τα αντικείμενα  $O_2$  και  $H_2O$  να φαίνονται σαν να «πέφτουν» από το πάνω μέρος του playfield ένα-ένα με τη σειρά και ο χρήστης κατά τη διάρκεια που ένα αντικείμενο πέφτει να μπορεί να επιλέξει αν είναι αέριο ή υγρό. Ανάλογα με την απάντηση που δίνει ο χρήστης θα πρέπει να εμφανίζεται το αντίστοιχο μήνυμα μέσα στο scrolling text για το αν η απάντηση που έχει δοθεί είναι σωστή ή λανθασμένη. Αν πάλι ο χρήστης δεν προλάβει να πατήσει κάποιο από τα κουμπιά και το αντικείμενο φτάσει στο κάτω μέρος του playfield θα πρέπει να πηγαίνουμε στο επόμενο αντικείμενο και να εμφανίζεται και πάλι στο scrolling text η σωστή απάντηση. Από αυτή την περιγραφή του παιχνιδιού προκύπτουν τα εξής.

- Για κάθε text-χημική ένωση θα πρέπει να ορίσουμε μία αρχική θέση (πάνω μέρος του playfield) και μία ταχύτητα καθόδου(y - άξονας). Συνεπάγεται ένα script για κάθε χημική ένωση
- Μόλις ο χρήστης δώσει μία απάντηση θα πρέπει να την καταγράφουμε και να εφαρμόζουμε την εξής διαδικασία
  - Εμφάνιση αποτελέσματος στο scrolling text
  - Σταμάτημα script του συγκεκριμένου αντικειμένου
  - Εξαφάνιση αντικειμένου
  - Εκτέλεση script επόμενου αντικειμένου
- Μόλις το αντικείμενο φτάσει στο κάτω μέρος θα πρέπει
  - Να εμφανίζεται το αποτέλεσμα στο scrolling text
  - Εκτέλεση script επόμενου αντικειμένου
- Μόλις ο χρήστης πατήσει reset θα πρέπει όλα τα αντικείμενα να επανέλθουν σε αρχική κατάσταση
- Μόλις ο χρήστης πατήσει start θα πρέπει να γίνεται αρχικοποίηση όλων των μεταβλητών και επαναφορά αντικειμένων σε σημείο που να μην είναι ορατά στο playfield.

Πριν συνεχίσουμε την ανάλυση ας δούμε με ποιον τρόπο μπορούμε να ελέγχουμε την ορθότητα των απαντήσεων που δίνουμε ανά πάσα στιγμή. Δημιουργούμε μεταβλητές για τις χημικές ενώσεις που τις ονομάζουμε ίδια με το όνομα της χημικής ένωσης. (Για να δημιουργήσουμε μία μεταβλητή επιλέγουμε από τα στοιχεία του αντικειμένου το κουμπί V που βρίσκεται ακριβώς αριστερά στο όνομα του αντικειμένου). Έτσι έχουμε τόσες τέτοιες μεταβλητές για κάθε αντικείμενο όσα είναι και τα αντικείμενα.



Εικόνα 48: τοποθετώντας μία επιπλέον μεταβλητή σε ένα αντικείμενο

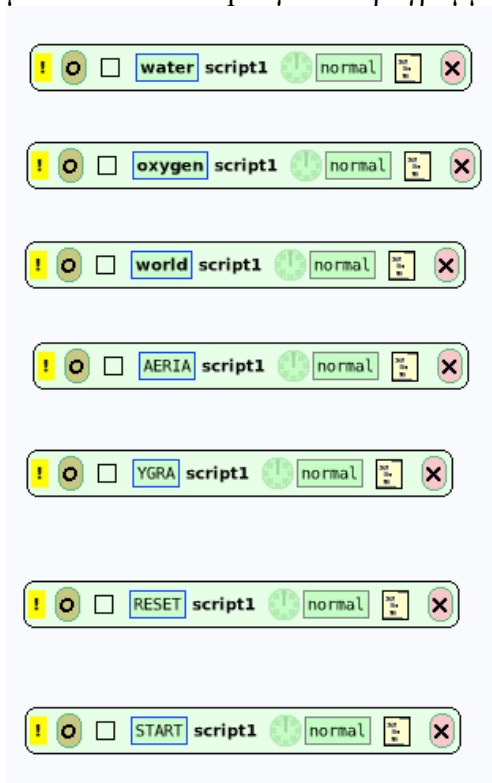
Κάθε φορά που ο χρήστης κάνει μία επιλογή (ΑΕΡΙΟ ή ΥΓΡΟ), τότε θέτουμε τιμές σε αυτή τη μεταβλητή που ταιριάζουν με την επιλογή του χρήστη. Έτσι, αν έχουμε 4 χημικές ενώσεις θα έχουμε και 4 αντικείμενα. Όταν ο χρήστης επιλέγει ΑΕΡΙΟ ή ΥΓΡΟ, ΑΝΕΞΑΡΤΗΤΑ ΑΠΟ ΤΟ ΠΟΙΟ ΑΝΤΙΚΕΙΜΕΝΟ ΠΕΦΤΕΙ ΤΗ ΔΕΔΟΜΕΝΗ ΣΤΙΓΜΗ θα πρέπει να ανατεθούν τιμές σε αυτές τις μεταβλητές βάση της απάντησης. Έτσι αν έχουμε τρία αέρα και 1 υγρό και ο χρήστης πατήσει ΑΕΡΙΟ τότε θα πρέπει να δώσουμε την τιμή 1 στις μεταβλητές των ενώσεων που είναι ΑΕΡΙΑ και την τιμή 2 στη μεταβλητή των ενώσεων που είναι υγρό. Αν πατήσουμε ΥΓΡΟ θα πρέπει να δώσουμε την τιμή 2 στις μεταβλητές των ενώσεων που είναι ΑΕΡΙΑ και την τιμή 1 στη μεταβλητή των ενώσεων που είναι υγρό. Με αυτό τον τρόπο θα μπορούμε να γνωρίζουμε τη σωστή απάντηση του χρήστη. Δεδομένου ότι για κάθε αντικείμενο που πέφτει θα πρέπει να αφήνουμε το χρήστη να επιλέξει αν είναι υγρό ή αέριο κάθε φορά που θα ξεκινά ένα καινούριο script οι τιμές αυτές θα πρέπει να μηδενίζονται.

Ας ξεκινήσουμε λοιπόν με αυτά τα στοιχεία να χτίζουμε το πρόγραμμά μας. Θα κάνουμε αρχή από τα απλά στοιχεία του προγράμματος. Θεωρούμε πως μέχρι τώρα έχουμε τα αρχικά στοιχεία του προγράμματος όπως δείξαμε στην αρχή καθώς και μία μεταβλητή για κάθε χημική ένωση η οποία έχει το όνομα της χημικής ένωσης. Στο συγκεκριμένο παράδειγμα λοιπόν θα πρέπει να έχουμε τις επιπλέον μεταβλητές oxygen μέσα στο αντικείμενο oxygen και water μέσα στο αντικείμενο water.

Αρχικά θα φτιάξουμε τα scripts για όλα τα αντικείμενα. Θα χρειαστούμε ένα script για κάθε κουμπί, ένα script για κάθε χημικό στοιχείο και ένα επιπλέον script που θα έχει ρόλο να μηδενίσει τις τιμές των επιπρόσθετων μεταβλητών που φτιάξαμε όταν θα γίνεται η μετάβαση από ένα αντικείμενο σε άλλο. Αυτό το τελευταίο script δε μας απασχολεί σε ποιο αντικείμενο θα ανήκει γι' αυτό θα το κάνουμε ένα script του κόσμου. Για να εμφανίσουμε τα στοιχεία του κόσμου πατάμε Alt+Click σε ένα κενό

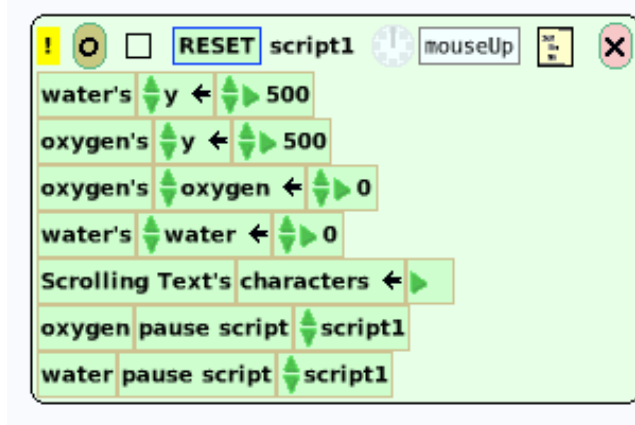
## Σχολικά Παραδείγματα

μέρος της οθόνης και επιλέγουμε το μάτι. Από εδώ θα φτιάξουμε αυτό το γενικό script. Έτσι δημιουργούμε επτά κενά scripts για το πρόγραμμά μας.



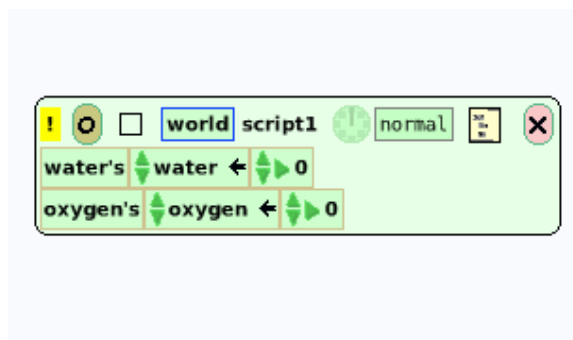
Εικόνα 49: όλα τα scripts του προγράμματος

Ας ξεκινήσουμε από τα εύκολα στοιχεία των scripts. Για παράδειγμα το script του Reset θέλουμε να μηδενίζει όλες τις τιμές που θα χρησιμοποιηθούν και να τοποθετεί τα αντικείμενα σε τέτοιο χώρο ώστε να μην είναι ορατά μόλις ξεκινά το παιχνίδι. Άρα, τοποθετούμε μέσα του όλα αυτά τα attributes σύροντάς τα από τις ιδιότητες των αντικειμένων. Έτσι έχουμε τα στοιχεία όπως φαίνονται στην επόμενη εικόνα.



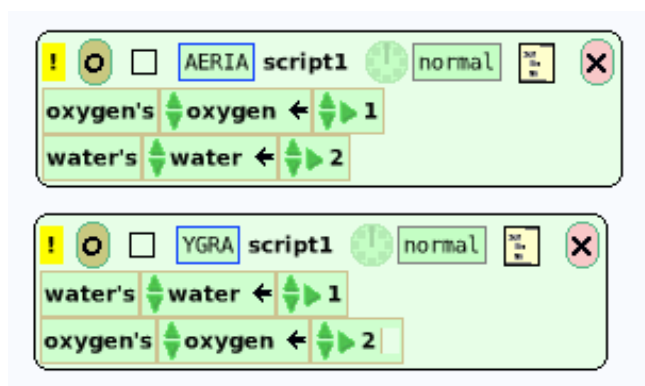
Εικόνα 50: Τα στοιχεία του RESET

Έπειτα περνάμε στο World Script το οποίο μηδενίζει τις επιπρόσθετες μεταβλητές κατά τη μετάβαση από ένα script σε άλλο.



Εικόνα 51: Τα στοιχεία του World script

Στη συνέχεια κατασκευάζουμε τα scripts που αφορούν τα κουμπιά απαντήσεων των χρηστών.



Εικόνα 52: Scripts για τα κουμπιά επιλογής του χρήστη

Όπως φαίνεται από αυτά τα scripts λέμε στο πρόγραμμα να θέσει την τιμή 1 στο οξυγόνο όταν ο χρήστης πατήσει ΑΕΡΙΑ και ταυτόχρονα θέτουμε την τιμή του νερού σε 2. Αντίστροφα συμβαίνει όταν ο χρήστης επιλέξει υγρά. Αυτό το κάνουμε για τον εξής λόγο. Όταν κατεβαίνει ένα αντικείμενο στην οθόνη (εκτελείται το script του) και ο χρήστης πατήσει ένα εκ των αέρια ή υγρά τότε ανατίθενται τιμές στις επιπρόσθετες μεταβλητές και ένας απλός έλεγχος μέσα στο script του αντικειμένου μπορεί να μας δώσει την απάντηση για το αν έχει πατηθεί το σωστό κουμπί ή όχι.

Για αυτά τα τρία scripts προκειμένου να εκτελούνται μόλις πατήσουμε το κουμπί θα πρέπει να το δηλώσουμε. Έτσι πατούμε αριστερό κλικ στη λέξη Normal (που βρίσκεται μέσα στο script) και την αλλάζουμε σε MouseUp.

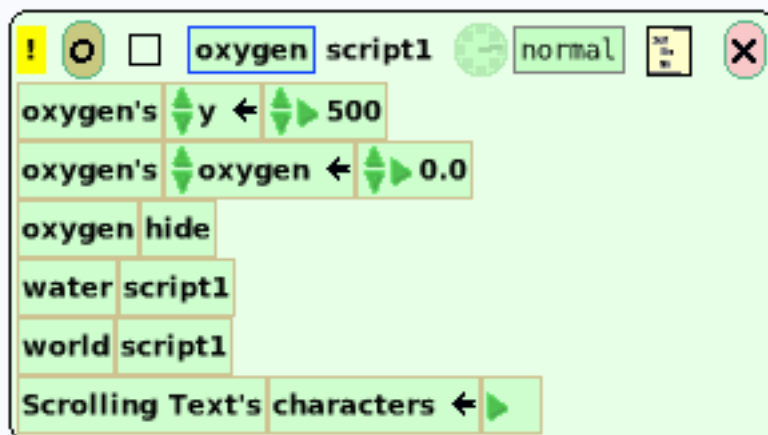
Ας περάσουμε όμως τώρα στα πιο σύνθετα scripts. Αυτά που αφορούν τα αντικείμενα που πέφτουν.

Αφού γράψουμε μέσα στο text τη λέξη οξυγόνο ανοίγουμε τα στοιχεία του αντικειμένου (εργαλείο μάτι μετά από Alt+Click στο αντικείμενο). Ονομάζουμε το αντικείμενο oxygen για να μπορούμε να το ξεχωρίζουμε. Για το συγκεκριμένο αντικείμενο δημιουργούμε ένα νέο script. Με τα δεδομένα που έχουμε γνωρίζουμε ότι με αυτό το script θα πρέπει

- να κάνουμε το αντικείμενο να πέφτει,
- να αλλάζουμε στοιχεία μέσα στο scrolling text,
- να τερματίζουμε το ίδιο το script όταν τελειώσει η διαδικασία του συγκεκριμένου αντικειμένου,
- να κάνουμε έλεγχο για την ορθότητα της απάντησης όταν δοθεί.
- να ξεκινούμε το επόμενο script,
- να κρύβουμε το αντικείμενο-χημική ένωση όταν τελειώσει η διαδικασία του συγκεκριμένου αντικειμένου και τέλος
- να μηδενίζουμε τις απαντήσεις όταν δοθεί μία απάντηση για το συγκεκριμένο αντικείμενο.

## Σχολικά Παραδείγματα

Βάση των παραπάνω βάζουμε μέσα στο script τις παρακάτω ιδιότητες όπως φαίνονται στην εικόνα.



Εικόνα 53: Στοιχεία που πρέπει να μπουν μέσα στο script της ένωσης

Ας δούμε όμως που θα μας φανεί χρήσιμο το καθένα. Αρχικά θα χρειαστούμε το oxygen's y προκειμένου να κάνουμε το αντικείμενο να πέφτει. Θα λέμε στο script να μειώνει το y κατά κάποιες μονάδες προκειμένου να απεικονίζεται η πτώση. Θα χρειαστούμε το oxygen's oxygen (επιπρόσθετη μεταβλητή μας) προκειμένου να ελέγχουμε τις απαντήσεις του χρήστη. Θα χρειαστούμε το oxygen's hide προκειμένου να κρύβουμε το αντικείμενο μόλις δε μας είναι πλέον χρήσιμο. Θα χρειαστούμε το water script1 (είναι το script για το αμέσως επόμενο αντικείμενο που θα πέσει όταν τελειώσει η διαδικασία του συγκεκριμένου). Θα χρειαστούμε το world script προκειμένου να μηδενίσουμε τις μεταβλητές απαντήσεων όταν περάσουμε από ένα script σε άλλο και τέλος θα χρειαστούμε το scrolling text για να γράψουμε την απάντηση. Εν συνεχεία γυρίζουμε σε code mode (show code textually) και γράφουμε τον παρακάτω κώδικα.

```
script1

self show.
self setY: self getY - 5.
(self getY < 100)
ifTrue: [
    self hide.
    self stopScript: #script1.
    ScrollingText setCharacters: 'To O2 einai AERIO'.
    Water startScript: #script1.
].
(self getOxygen = 1)
ifTrue: [
    self hide.
```

```

        self stopScript: #script1.
        ScrollingText setCharacters: 'SWSTA - To O2 einai AERIO'.
        START doScript: #smallReset.
        Water startScript: #script1.

    ].
    (self getOxygen = 2)
    ifTrue: [
        self hide.
        self stopScript: #script1.
        ScrollingText setCharacters: 'LATHOS - To O2 einai AERIO'.
        START doScript: #smallReset.
        Water startScript: #script1.
    ]

```

Ας δούμε ένα-ένα τα κομμάτια του κώδικα. Αρχικά εμφανίζουμε το αντικείμενο, αφού είναι σε κατάσταση hide. Εν συνεχεία, θέτουμε το Y του αντικειμένου να μειώνεται κατά 5. έπειτα ελέγχουμε μήπως το αντικείμενο έχει κατέβει πολύ χαμηλά οπότε πρέπει να διακοπεί η διαδικασία (ο χρήστης δεν πρόλαβε να πατήσει κάποιο κουμπί εκτέλεσης). Τέλος κάνουμε έλεγχο για να δούμε τι έχει πατήσει ο χρήστης από τα κουπιά εκτέλεσης. Να θυμηθούμε πως όταν ο χρήστης πατήσει κάποιο κουμπί εκτέλεσης τότε θέτουμε τιμές στις βοηθητικές μεταβλητές. Συνεπώς αρκεί να κάνουμε έναν έλεγχο για να δούμε τι τιμή έχει πάρει η συγκεκριμένη βοηθητική τιμή. Αν ο χρήστης πατήσει κάποιο από αυτά, τότε ο έλεγχος θα περάσει μέσα στις εντολές οι οποίες είναι με τη σειρά:

- απόκρυψη του συγκεκριμένου αντικειμένου
- σταμάτημα της επαναλαμβανομένης εκτέλεσης του συγκεκριμένου script
- Εγγραφή στο Scrolling text της απάντησης
- Καθάρισμα των επιπρόσθετων μεταβλητών
- Εκτέλεση του επόμενου script

Ακριβώς το ίδιο χρειαζόμαστε και για το δεύτερο αντικείμενο που περιμένουμε να αρχίσει να πέφτει στην οθόνη αφού τελειώσουμε τη διαδικασία με το πρώτο. Αφού λοιπόν μεταφέρουμε μέσα στο δεύτερο script όσα αντικείμενα θα χρησιμοποιήσουμε περνάμε σε κώδικα πάλι και γράφουμε παραπλήσιο κώδικα με το πρώτο αντικείμενο όπως φαίνεται παρακάτω.

```

script1

    self show.
    self setY: self getY - 5.
    (self getY < 100)

```



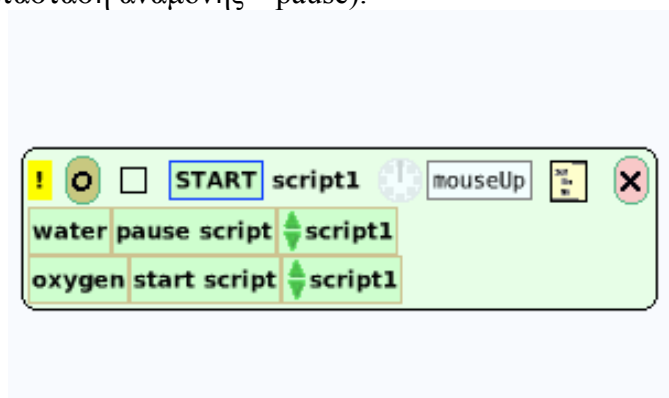
```

ifTrue: [
    self hide.
    self stopScript: #script1.
    ScrollingText setCharacters: 'To H2O einai YGRO'.
].
(self getWater = 1)
ifTrue: [
    self hide.
    self stopScript: #script1.
    ScrollingText setCharacters: 'SWSTA - To H2O einai YGRO'.
].
(self getWater = 2)
ifTrue: [
    self hide.
    self stopScript: #script1.
    ScrollingText setCharacters: 'LATHOS - To H2O einai YGRO'.
]

```

Οι διαφορές που παρατηρούνται εντοπίζονται στο γεγονός ότι αυτό είναι το τελευταίο αντικείμενο του προγράμματός μας συνεπώς δεν είναι απαραίτητο μετά από αυτό να εκτελέσουμε άλλο script.

Αφού τελειώσαμε και με αυτό το script απομένει να κατασκευάσουμε το script για το κουμπί εκκίνησης. Το κουμπί εκκίνησης θα πρέπει να θέτει σε κατάσταση start το πρώτο αντικείμενο που πρέπει να ξεκινήσει να πέφτει ενώ θα πρέπει να παγώσει όλα τα υπόλοιπα (κατάσταση αναμονής – pause).



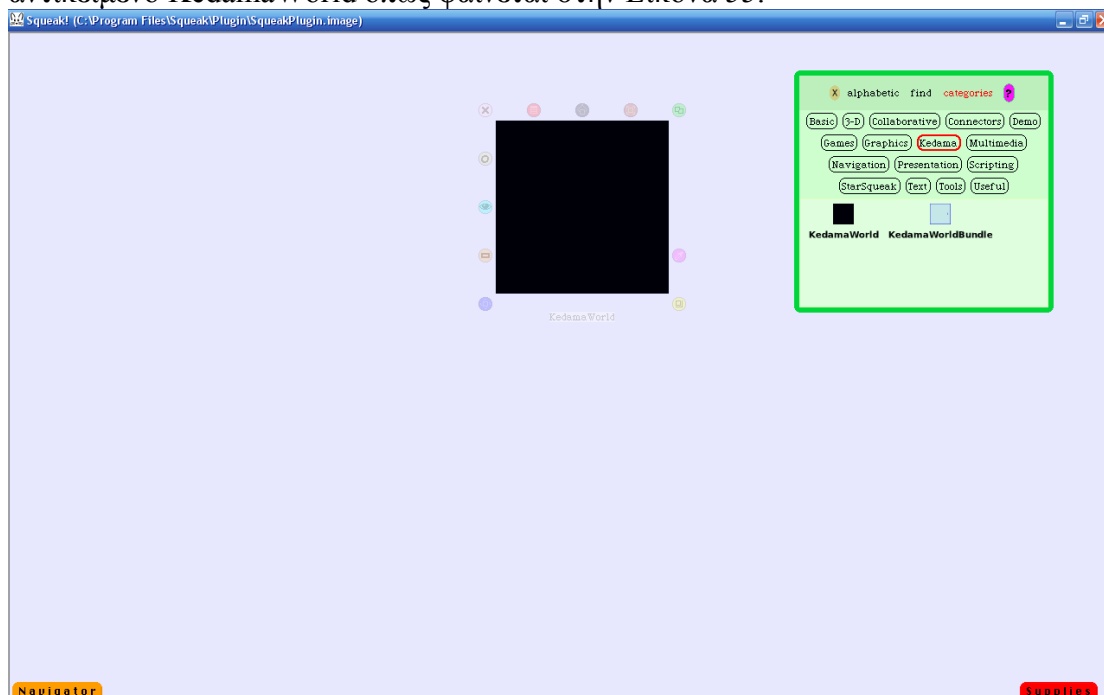
Εικόνα 54: το script για το κουμπί start

Για να τρέξει το πρόγραμμά μας πατούμε το κουμπί reset και εν συνεχεία το κουμπί start. Κάθε φορά που θέλουμε να εκκινήσουμε το πρόγραμμά μας θα πρέπει πρώτα να το κάνουμε reset και έπειτα start. Σε αυτό το πρόγραμμα είναι πολύ εύκολο να

προσθέσουμε τόσο άλλα κουμπιά εκτέλεσης (π.χ. στερεά) και φυσικά να προσθέσουμε και άλλες χημικές ενώσεις.

### 3.2. Θερμοκρασία και Πίεση

Στο παράδειγμα που ακολουθεί θα γίνει η κατασκευή ενός κλειστού δοχείου στο οποίο θα φαίνεται η κίνηση των μορίων ενός αερίου σε σχέση με τη θερμοκρασία. 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 55.



Εικόνα 55: Δημιουργώντας ένα Kedama-World αντικείμενο

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

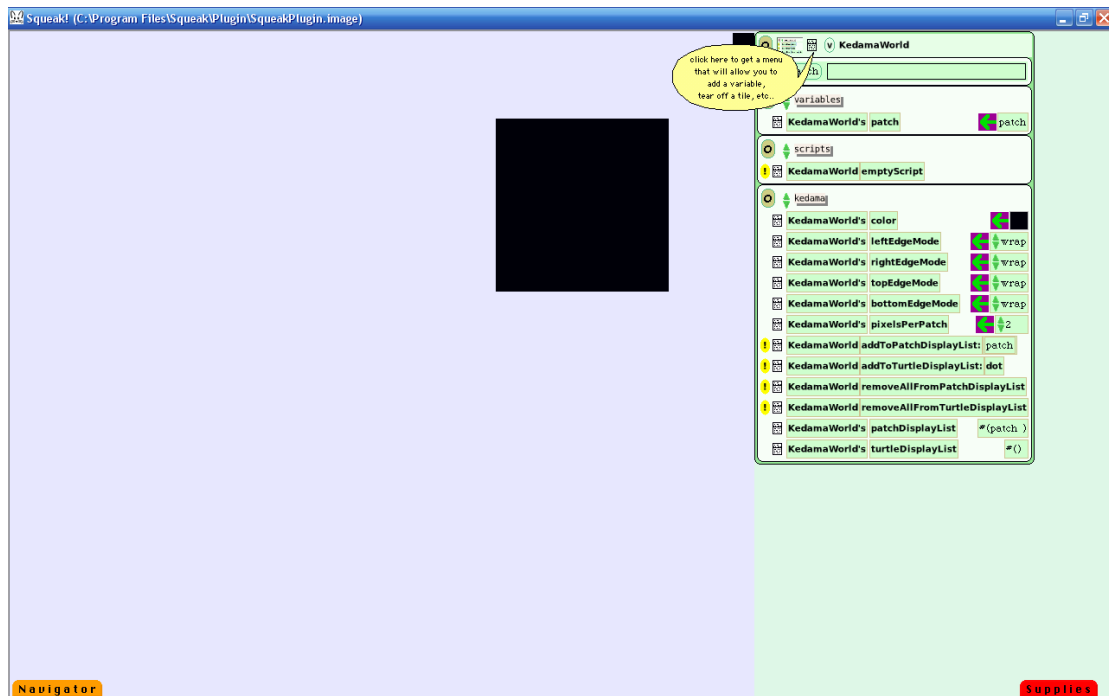
Όσον αφορά τους χρήστες με υψηλή ανάλυση στην οθόνη τους, υπάρχει η εντολή pixelsPerPatch στην παραπάνω κατηγορία με την οποία μπορούμε να αλλάξουμε τον αριθμό των pixel για να φαίνεται πιο καθαρά, οτιδήποτε φτιάχνεται στο αντικείμενο ‘KedamaWorld’.

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

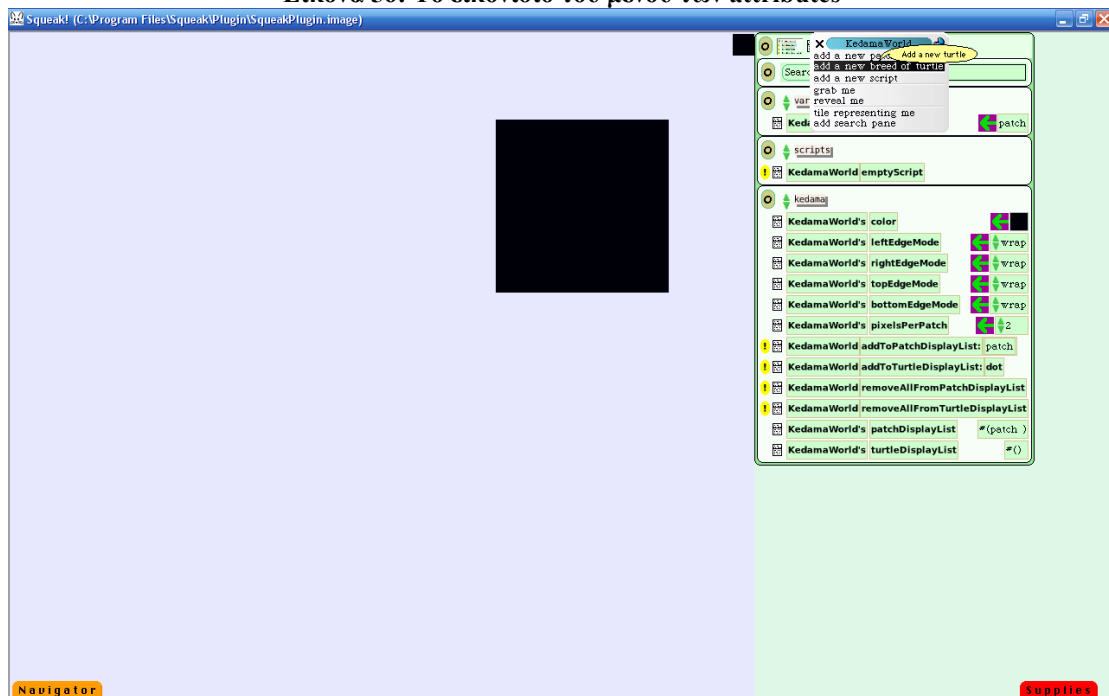
Τα βήματα είναι τα εξής:

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

Για να φτιάξουμε μια νέα ομάδα από χελώνες, πατάμε πάνω στο εικονίδιο του μενού(menu)(Εικόνα 56) που βρίσκεται στην κορυφή του ‘Viewer’ και επιλέγουμε ‘add a new breed of turtle’(Εικόνα 57).

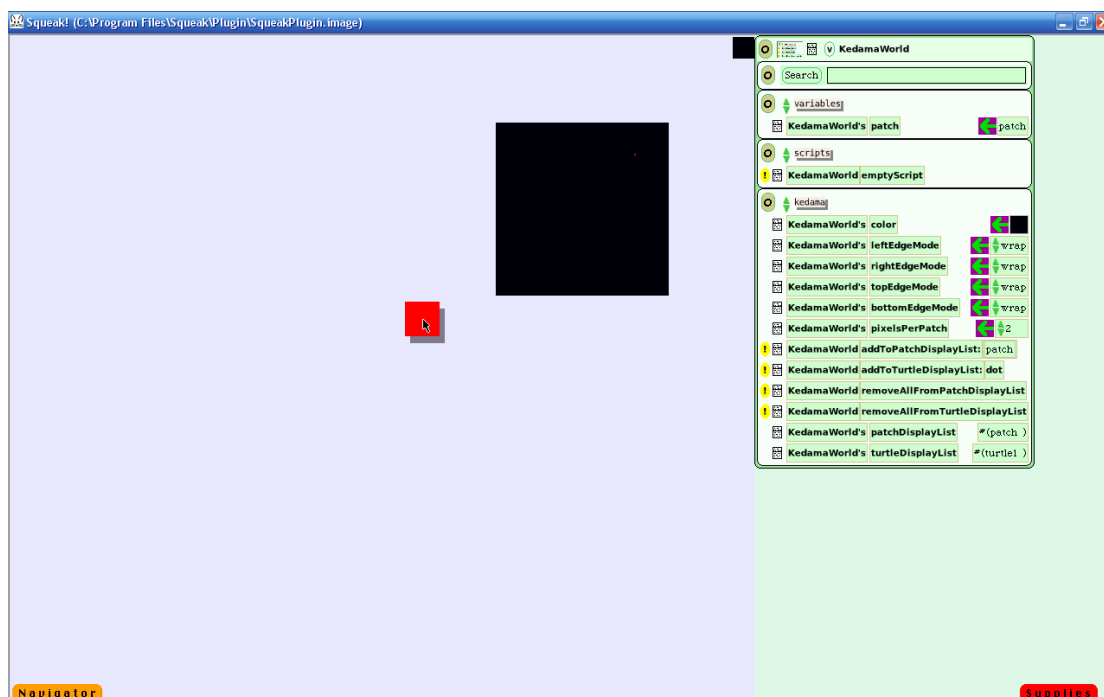


**Εικόνα 56: Το εικονίδιο του μενού των attributes**



**Εικόνα 57: Επιλογή του add a new breed of turtle**

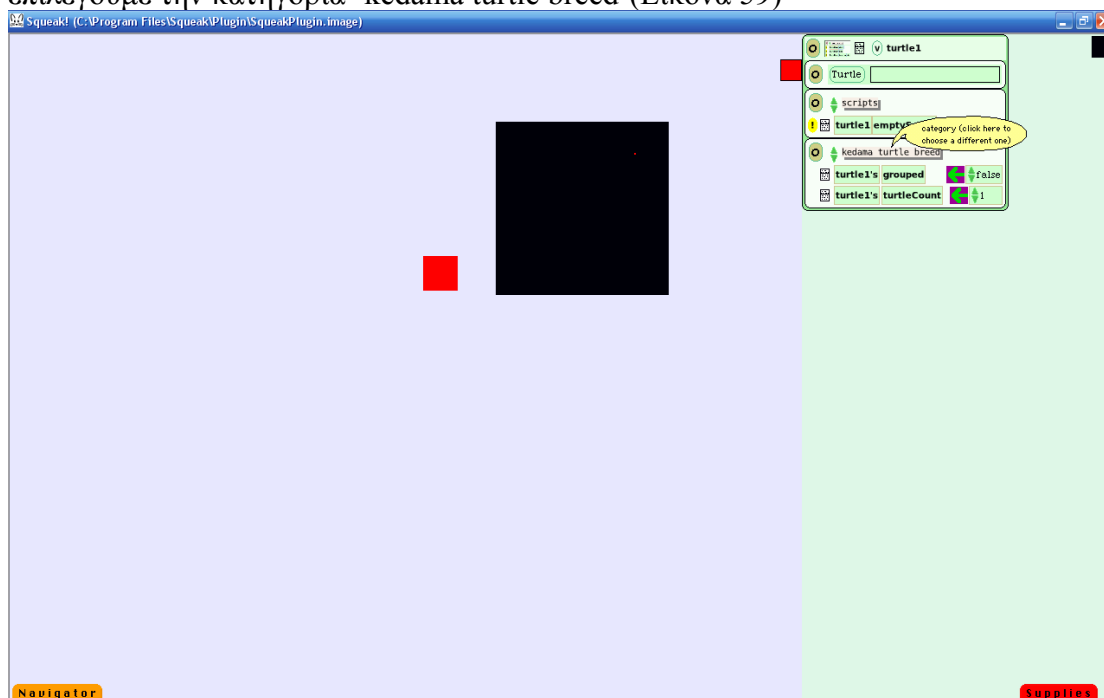
Το αποτέλεσμα που παίρνουμε είναι ένα χρωματιστό κουτί στο δείκτη του ποντικιού όπως φαίνεται στην Εικόνα 58.



Εικόνα 58: Το «χρωματιστό κουτί»

Το κουτί αυτό είναι η αναπαράσταση της ομάδας των χελωνών στον κόσμο. Παίζει το ρόλο του μοντέλου(model) για όλες τις χελώνες που δημιουργούμε 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.

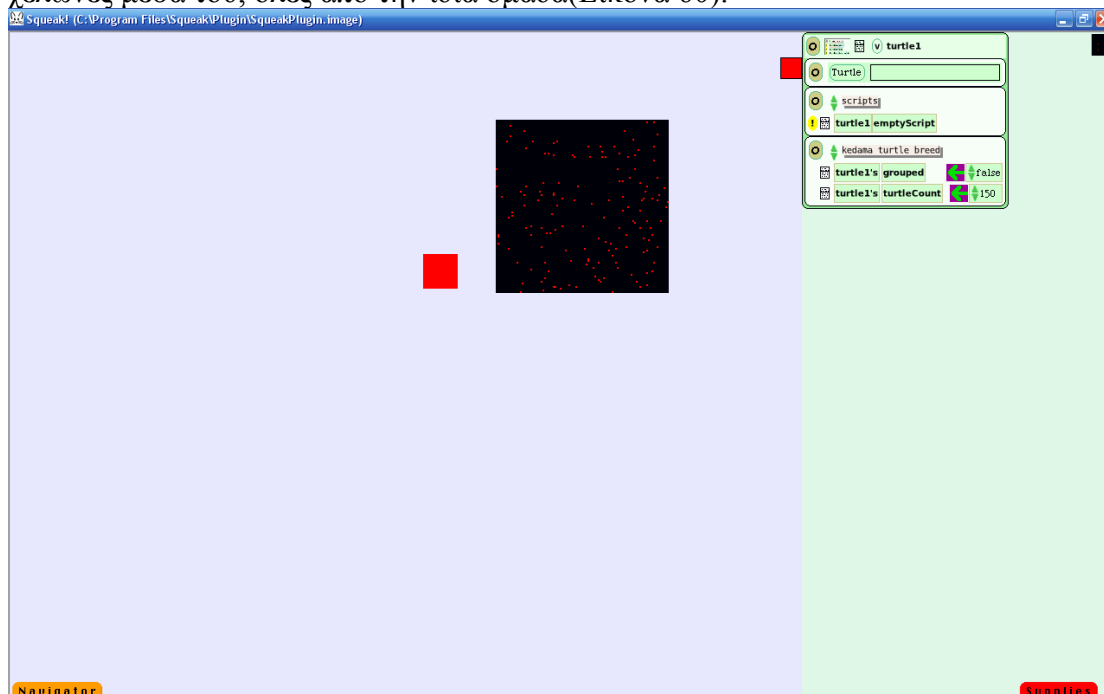
Στη συνέχεια, ορίζουμε τον αριθμό των χελωνών. Επειδή θέλουμε όλες οι χελώνες να συμπεριφερθούν με τον ίδιο τρόπο, και να ακολουθήσουν το ίδιο script, τότε θέλουμε να προσθέσουμε χελώνες από την ίδια ομάδα όπως στην αρχή. Για το λόγο αυτό ανοίγουμε έναν 'Viewer' του κόκκινου τετραγώνου που αναπαριστά την ομάδα και επιλέγουμε την κατηγορία 'kedama turtle breed'(Εικόνα 59)



Εικόνα 59: Επιλέγοντας το Kedama Turtle Breed

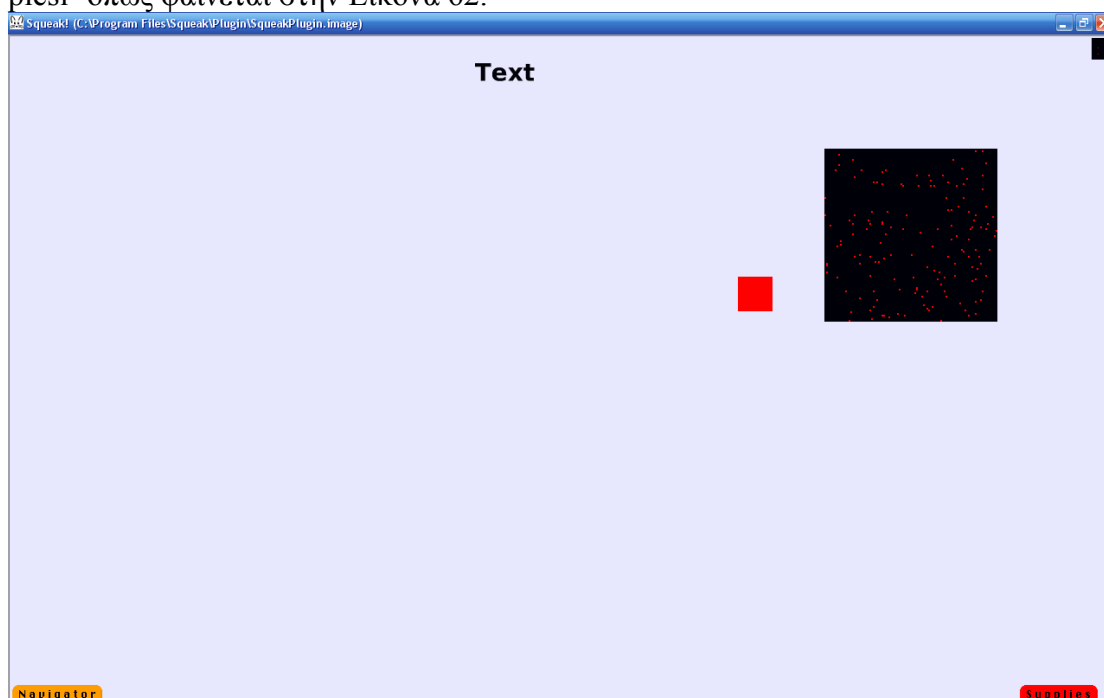
## Σχολικά Παραδείγματα

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. Όταν βάλουμε τον αριθμό των χελωνών που θέλουμε, θα δούμε στο αντικείμενο 'KedamaWorld' γεμάτο με πολλές κόκκινες χελώνες μέσα του, όλες από την ίδια ομάδα(Εικόνα 60).

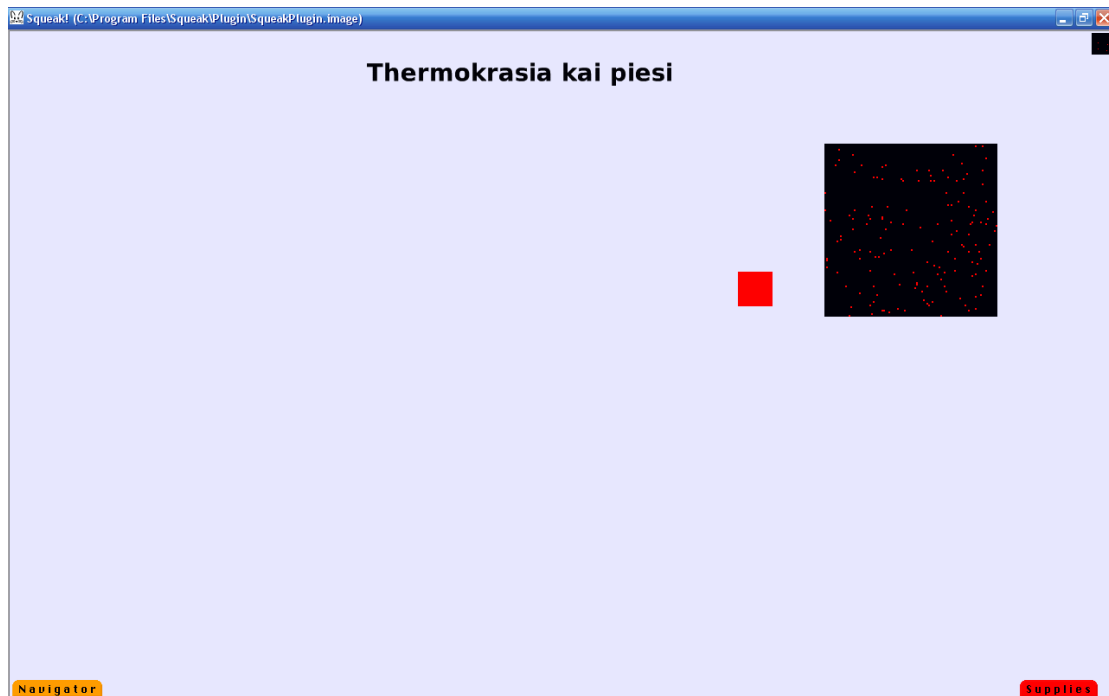


**Εικόνα 60: Πολλές κόκκινες χελώνες μέσα στο Kedama World**

Το επόμενο μας βήμα είναι να βάλουμε το τίτλο του προγράμματος. Από την καρτέλα 'Supplies' τραβάμε ένα 'Text' αντικείμενο και το τοποθετούμε στο πάνω μέρος του περιβάλλοντος(Εικόνα 61). Στη συνέχεια του δίνουμε το όνομα 'Thermokrasia kai piesi' όπως φαίνεται στην Εικόνα 62.

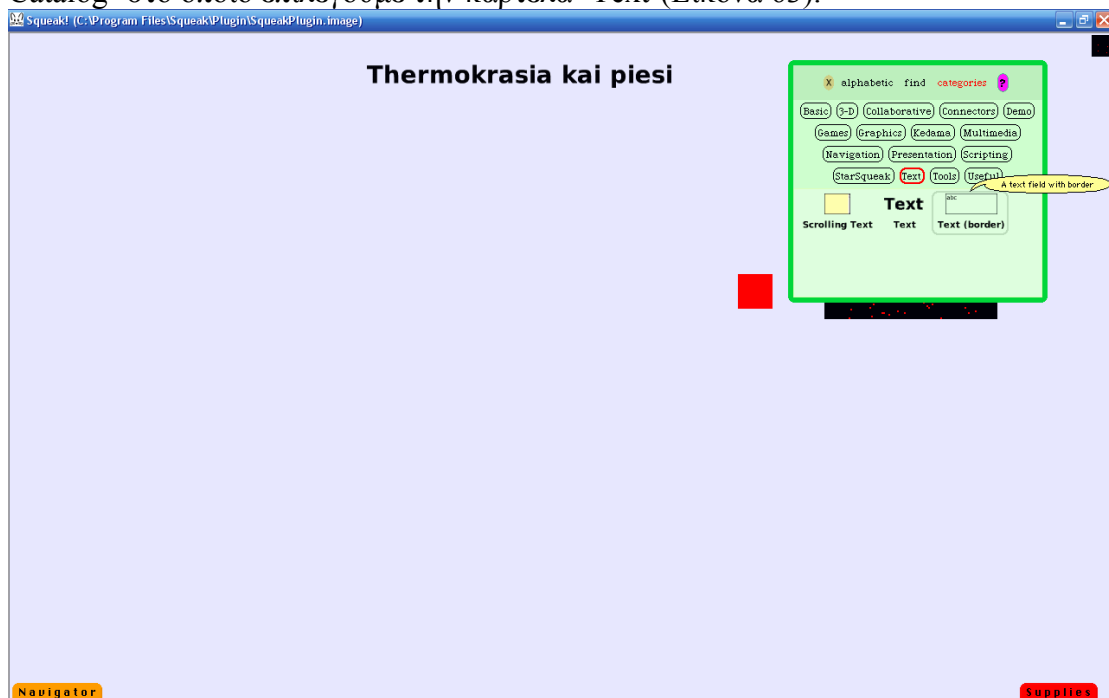


**Εικόνα 61: Το αντικείμενο 'Text' στην επιφάνεια εργασίας**



**Εικόνα 62: Συμπλήρωση τίτλου του προγράμματος**

Στη συνέχεια ανοίγουμε την καρτέλα 'Supplies' και σύρουμε ένα αντικείμενο 'Object Catalog' στο οποίο επιλέγουμε την καρτέλα 'Text' (Εικόνα 63).



**Εικόνα 63: Το 'Object Catalog'**

Στην καρτέλα αυτή τραβάμε ένα αντικείμενο 'Text (border)' όπως φαίνεται στην Εικόνα 64 και γράφουμε την τιμή '20 Celcius' (Εικόνα 65).

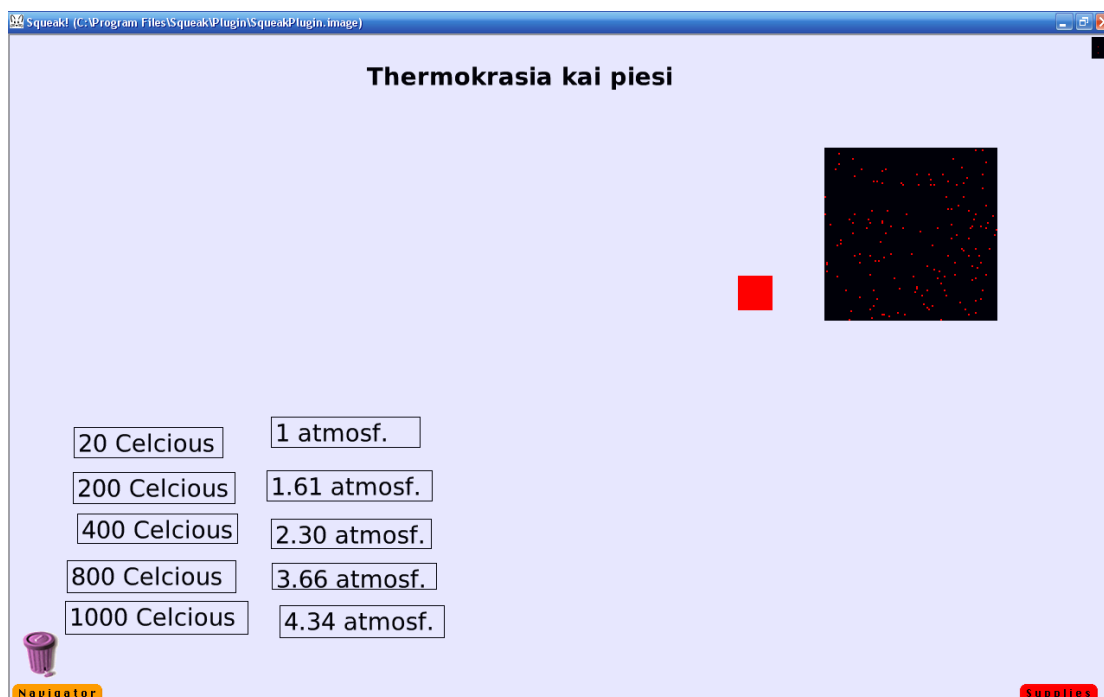


**Εικόνα 64: Το αντικείμενο 'Text (border)'**



**Εικόνα 65: Συμπληρώνοντας το αντικείμενο 'Text (border)'**

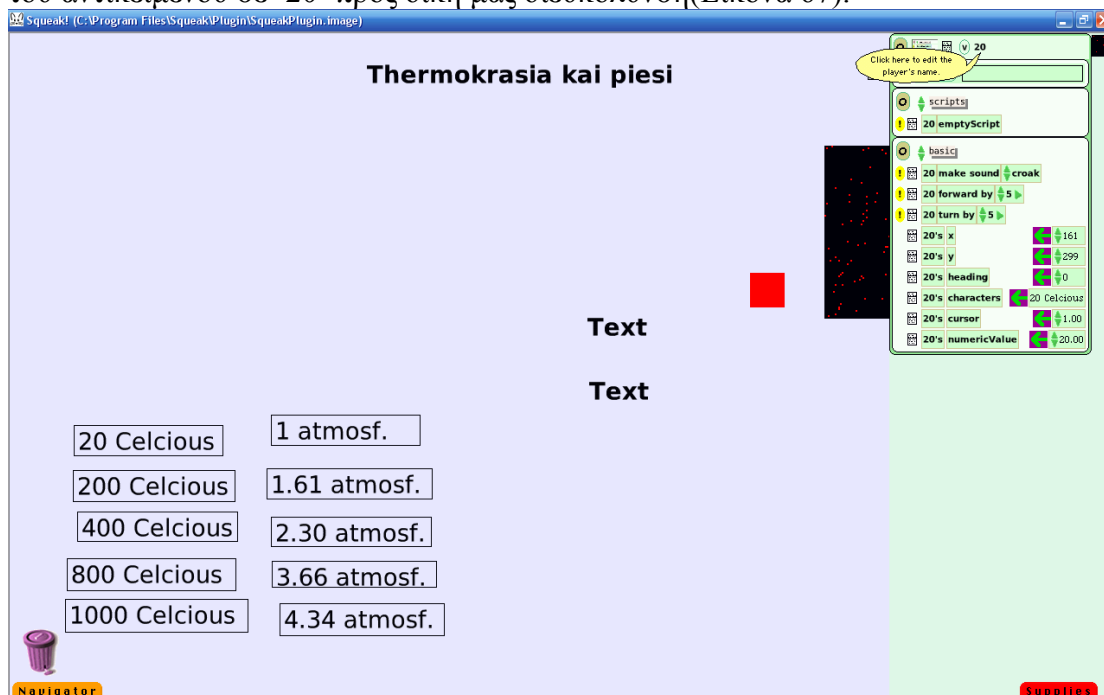
Πατώντας 'Alt + κλικ' στο αντικείμενο και πατώντας το πράσινο εργαλείο του και σύροντας λίγο προς οποιαδήποτε κατεύθυνση, τότε το αντικείμενο μας αντιγράφεται. Κάνοντας λοιπόν το ίδιο και για τις τιμές 20, 200, 400, 800, 1000 βαθμούς κελσίου, με αντίστοιχες πιέσεις τις '1', '1.61', '2.30', '3.66', '4.34' του παραδείγματός μας, παίρνουμε ως αποτέλεσμα την Εικόνα 66.



**Εικόνα 66: Βάζοντας τις τιμές του παραδείγματός μας.**

Στη συνέχεια ανοίγουμε την καρτέλα 'Supplies' και τραβάμε δύο αντικείμενα 'Text' στα οποία θα εμφανίζονται οι τιμές της θερμοκρασίας και της πίεσης αντίστοιχα. Τα ονόματα που έχουμε δώσει στα δύο αυτά αντικείμενα είναι 'temp' και 'press'.

Κατόπιν θα φτιάξουμε τα script όπου για κάθε θερμοκρασία θα αντιστοιχεί μια ατμόσφαιρα και θα απεικονίζεται με την κίνηση των μορίων. Πηγαίνουμε στο αντικείμενο 'Text (border)' που έχει τη θερμοκρασία '20 Celcius'. Στο αντικείμενο αυτό πατάμε το εργαλείο 'μάτι'. Στο μενού που προκύπτει αλλάζουμε την ονομασία του αντικειμένου σε '20' προς δική μας διευκόλυνση (Εικόνα 67).



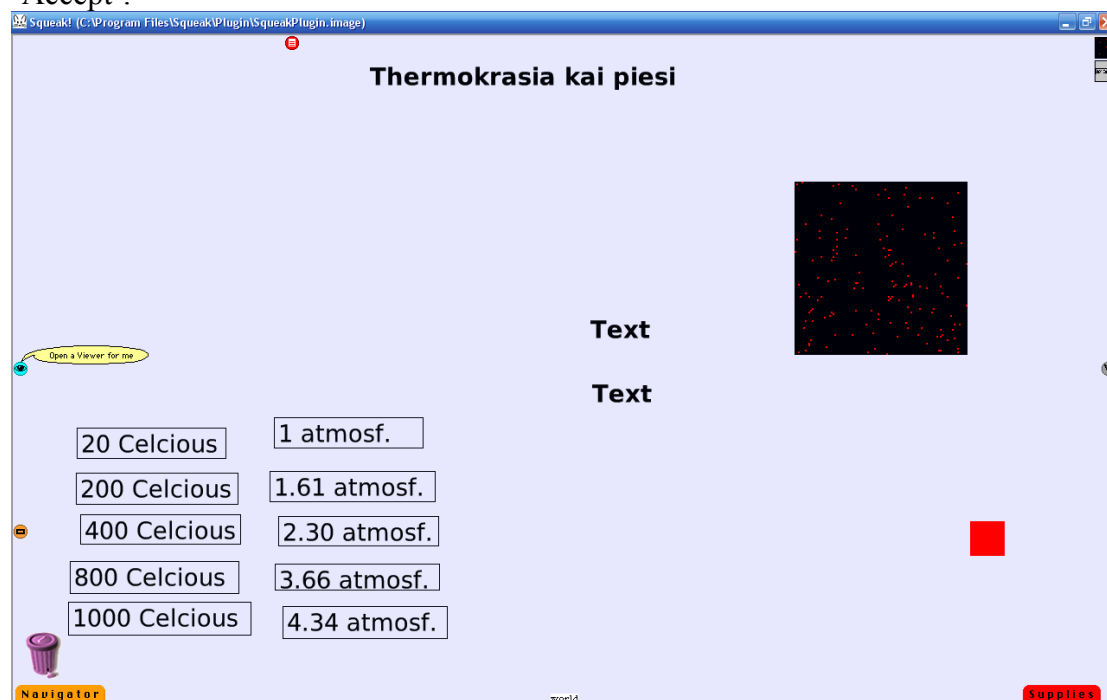
**Εικόνα 67: Αλλαγή ονόματος αντικειμένου**

Αντίστοιχα κάνουμε τις αλλαγές και στα υπόλοιπα ονόματα των αντικειμένων που περιέχουν τις τιμές των θερμοκρασιών και της πίεσης. Όσον αφορά τα ονόματα που θα δώσουμε στα αντικείμενα που περιέχουν τη πίεση, θα έχουν το format 'atm(τιμή



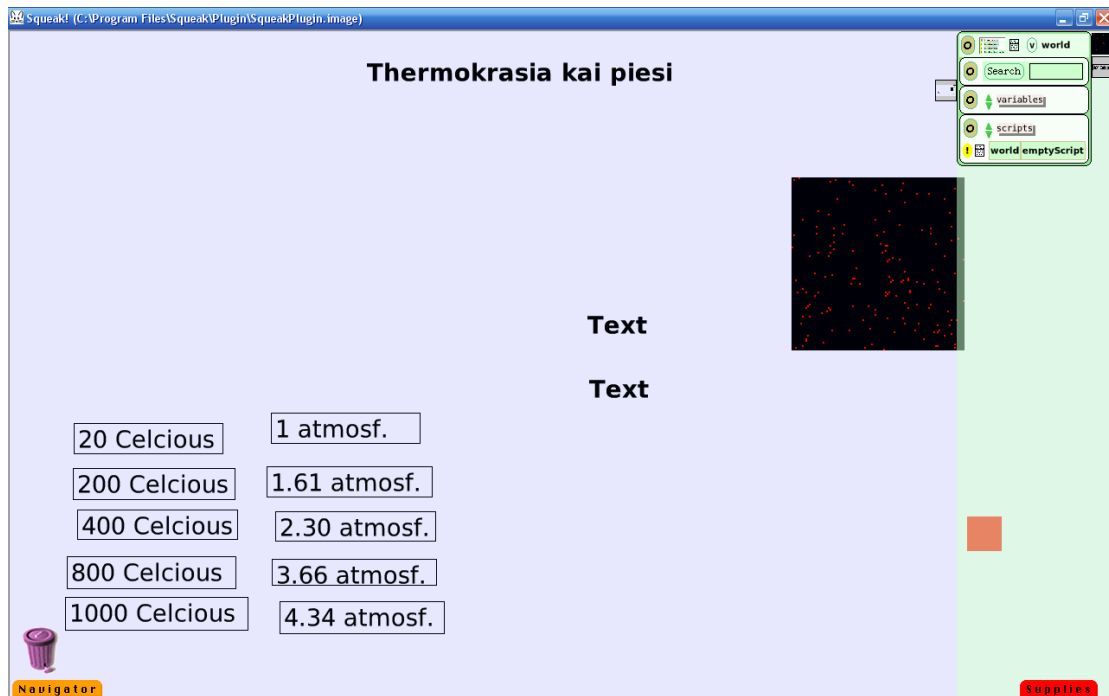
πίεσης)’. Για παράδειγμα το αντικείμενο που περιέχει τη τιμή ‘1atmosf’ θα ονομάζεται ‘atm1’.

Κατόπιν θα δημιουργήσουμε μια μεταβλητή του αντικειμένου ‘world’. Το αντικείμενο ‘world’ είναι ουσιαστικά όλο το πρόγραμμά μας, δηλαδή είναι το αντικείμενο το οποίο περικλείει όλα τα αντικείμενα που έχουμε δημιουργήσει ήδη και όσα πρόκειται να δημιουργήσουμε. Η μεταβλητή αυτή θα ευθύνεται για τις όποιες αλλαγές στην ταχύτητα των μορίων ανάλογα με τη δεδομένη τιμή θερμοκρασίας. Για να το επιτύχουμε αυτό, πατάμε ‘alt’ και ‘αριστερό κλικ’ στην γκρίζα περιοχή του προγράμματος. Στην Εικόνα 68 που εμφανίζεται πατάμε το εργαλείο ‘μάτι’ του αντικειμένου ‘world’ (Εικόνα 69). Στο μενού αυτό πατάμε στο κουμπί ‘ν’ για να δημιουργήσουμε μια μεταβλητή (Εικόνα 70). Την μεταβλητή αυτή την ονομάζουμε ‘global’ (Εικόνα 71). Για να ολοκληρώσουμε την δημιουργία της, απλά πατάμε ‘Accept’.

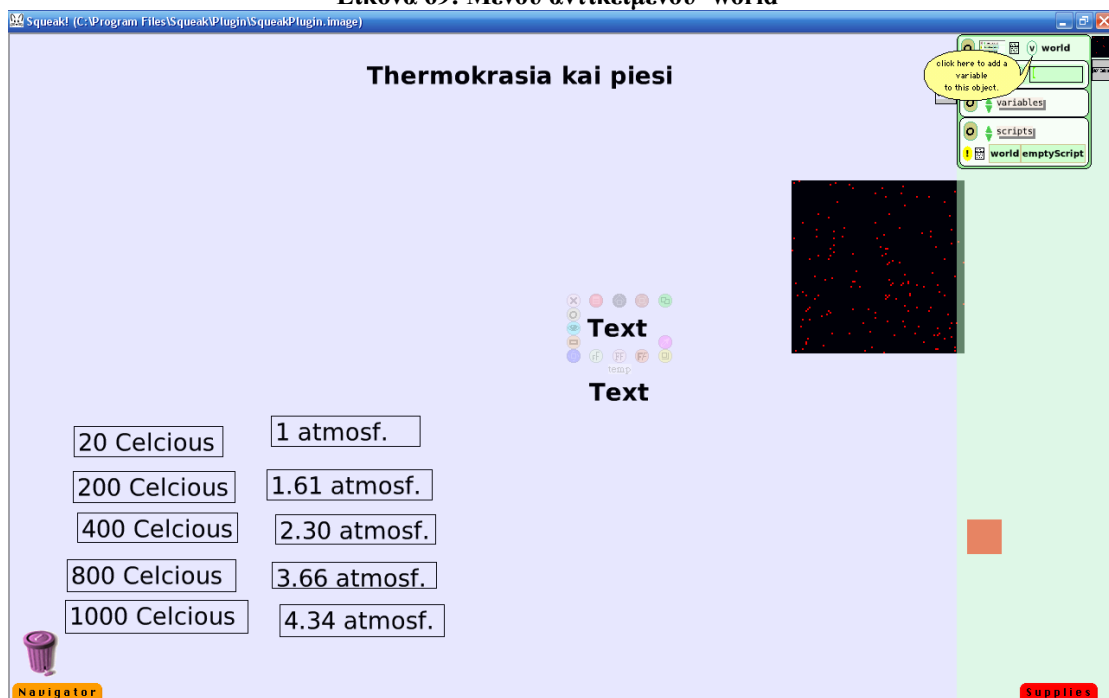


Εικόνα 68: Αντικείμενο world

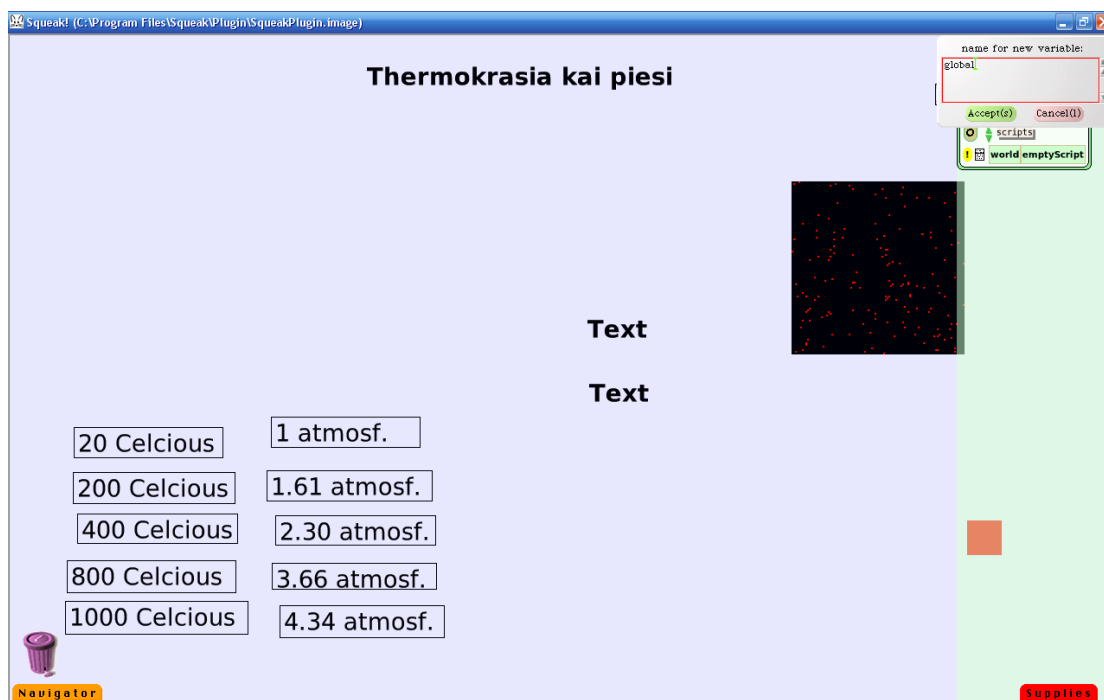
## Σχολικά Παραδείγματα



Εικόνα 69: Μενού αντικειμένου 'world'

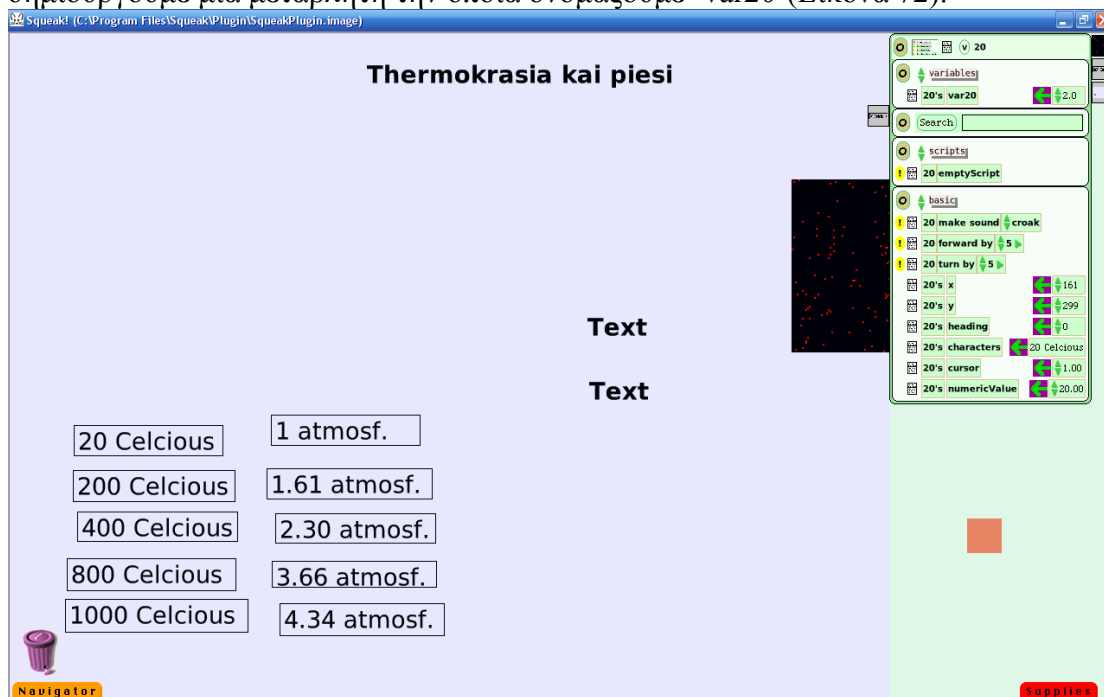


Εικόνα 70: Δημιουργώντας μια μεταβλητή του αντικειμένου 'world'



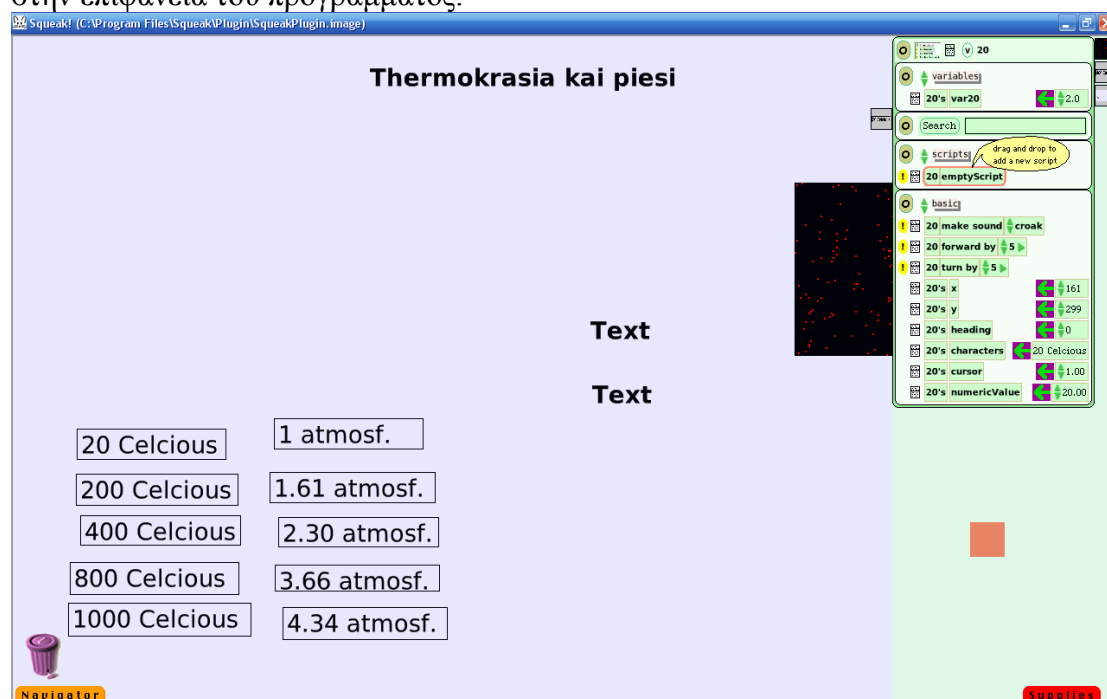
Εικόνα 71: Η μεταβλητή 'global' του αντικειμένου 'world'

Στη συνέχεια θα πρέπει να δημιουργήσουμε τα script, τα οποία θα ανανεώνουν τα τις τιμές της θερμοκρασίας και της πίεσης, που θα απεικονίζονται στο πρόγραμμα. Πηγαίνουμε στο αντικείμενο '20' και ανοίγουμε το εργαλείο 'μάτι'. Εκεί δημιουργούμε μια μεταβλητή την οποία ονομάζουμε 'var20' (Εικόνα 72).



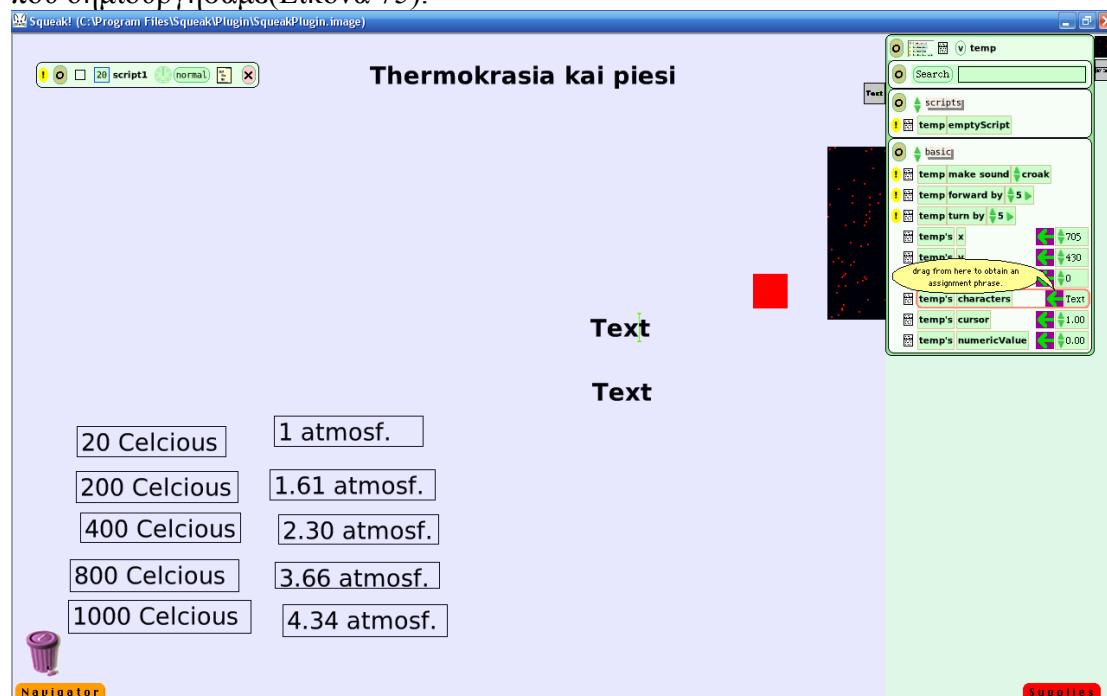
Εικόνα 72: Η μεταβλητή 'var20' του αντικειμένου '20'

Από την κατηγορία ‘script’ παίρνουμε ένα ‘empty script’(Εικόνα 73) και το σύρουμε στην επιφάνεια του προγράμματος.

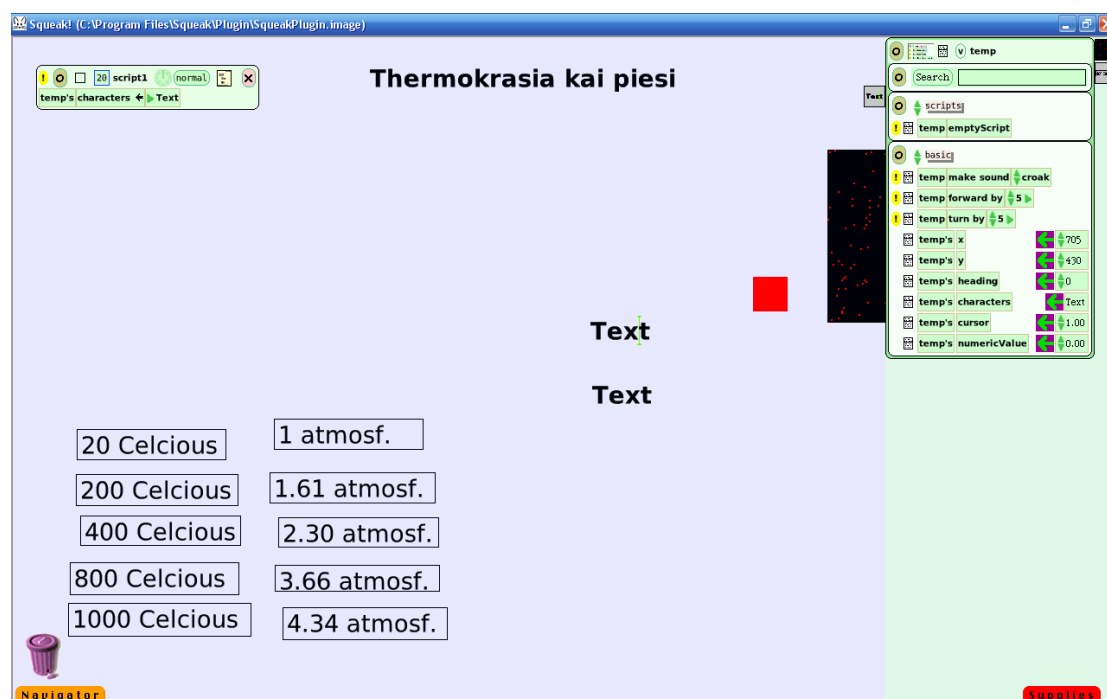


Εικόνα 73: Δημιουργία script για το αντικείμενο ‘20’

Στη συνέχεια πηγαίνουμε στο αντικείμενο ‘temp’ και χρησιμοποιούμε το εργαλείο ‘μάτι’. Πηγαίνουμε στη κατηγορία ‘basic’ και επιλέγουμε από το ‘temp’s characters’ το βελάκι που δείχνει προς τα αριστερά(Εικόνα 74) και το σύρουμε μέσα στο script που δημιουργήσαμε(Εικόνα 75).

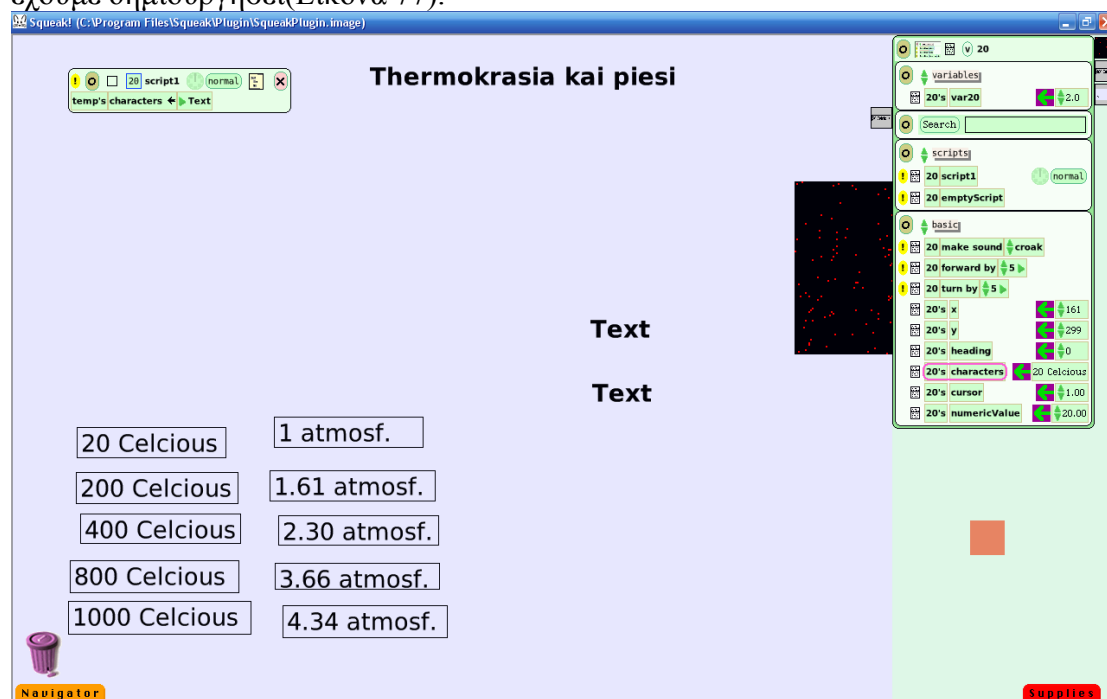


Εικόνα 74: Γραμμή ‘temp’s characters’ της κατηγορίας basic

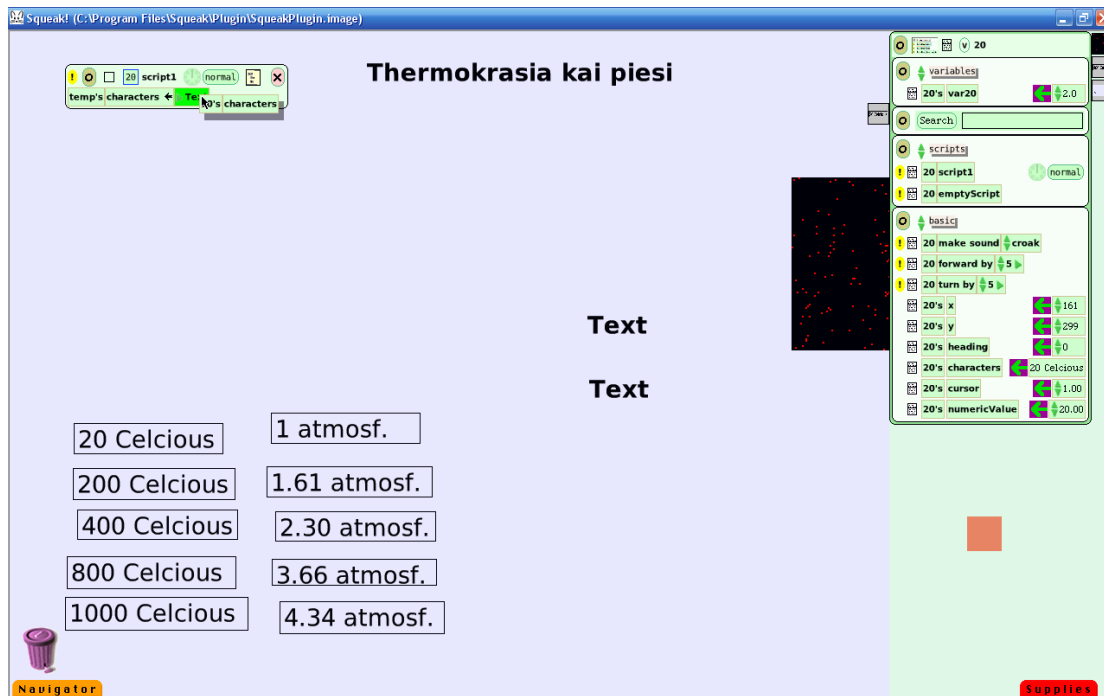


Εικόνα 75:Script του αντικειμένου '20'

Στη συνέχεια ανοίγουμε πάλι το εργαλείο 'μάτι' του αντικειμένου '20' και πηγαίνουμε στην κατηγορία 'basic'. Εκεί θα σύρουμε το πεδίο '20's characters'(Εικόνα 76) πάνω στη λέξη 'Text' του script του αντικειμένου '20' που έχουμε δημιουργήσει(Εικόνα 77).

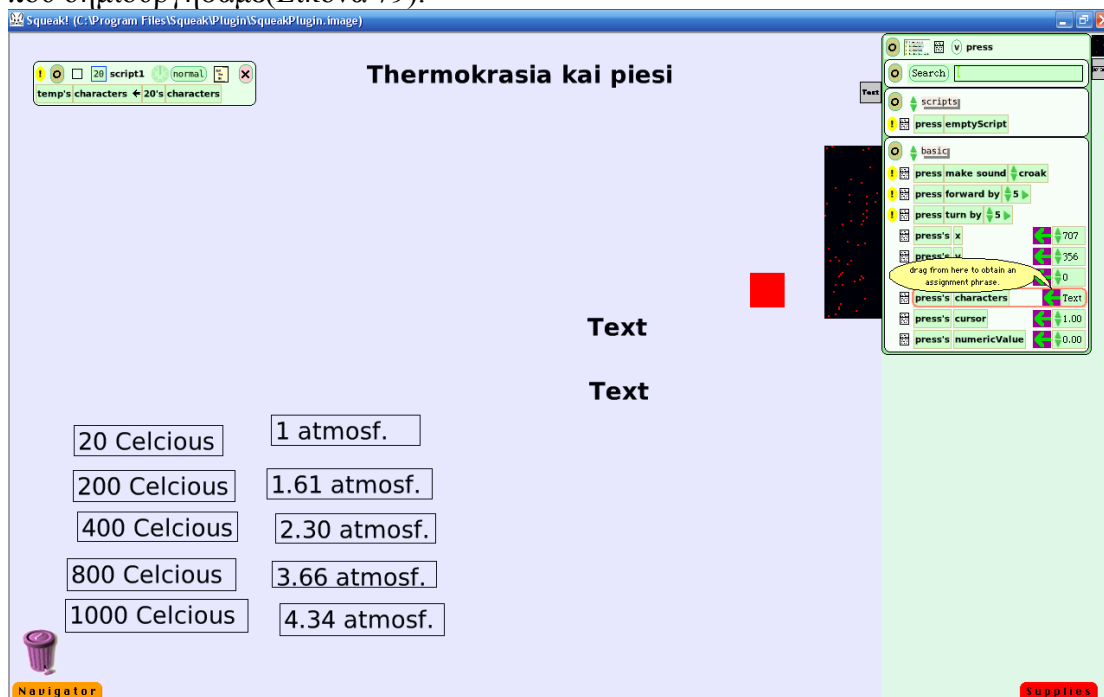


Εικόνα 76: γραμμή '20's characters' του αντικειμένου '20' βήμα 1

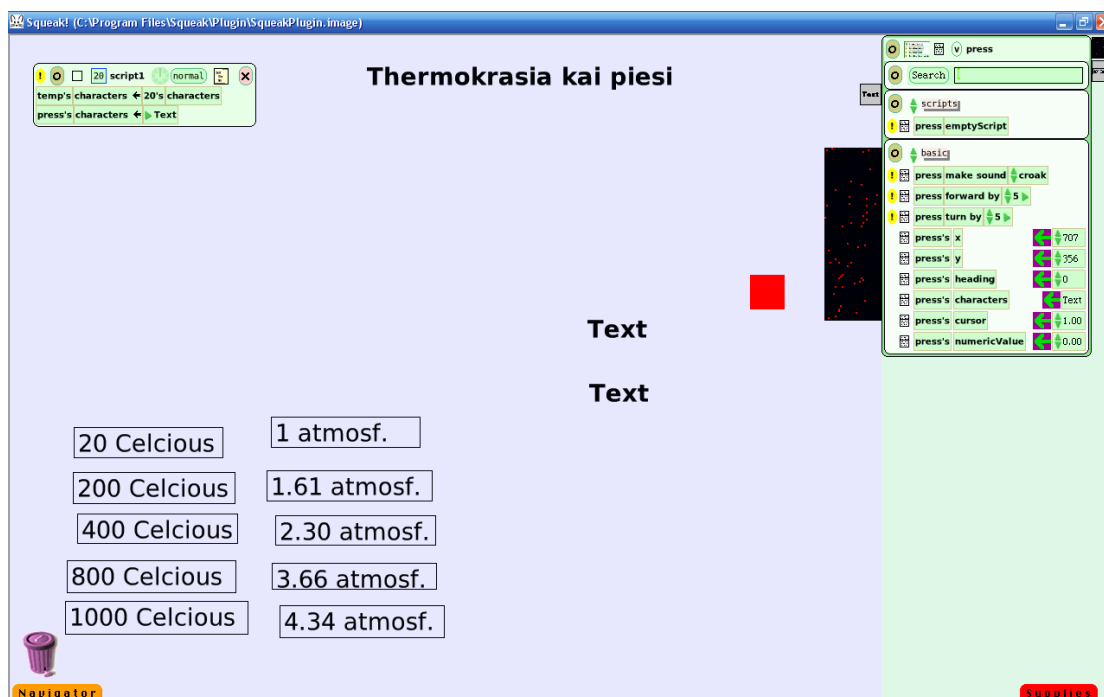


Εικόνα 77: γραμμή '20's characters' του αντικειμένου '20' βήμα 2

Στη συνέχεια πηγαίνουμε στο αντικείμενο 'press' και χρησιμοποιούμε το εργαλείο 'μάτι'. Πηγαίνουμε στη κατηγορία 'basic' και επιλέγουμε από το 'press's characters' το βελάκι που δείχνει προς τα αριστερά(Εικόνα 78) και το σύρουμε μέσα στο script που δημιουργήσαμε(Εικόνα 79).

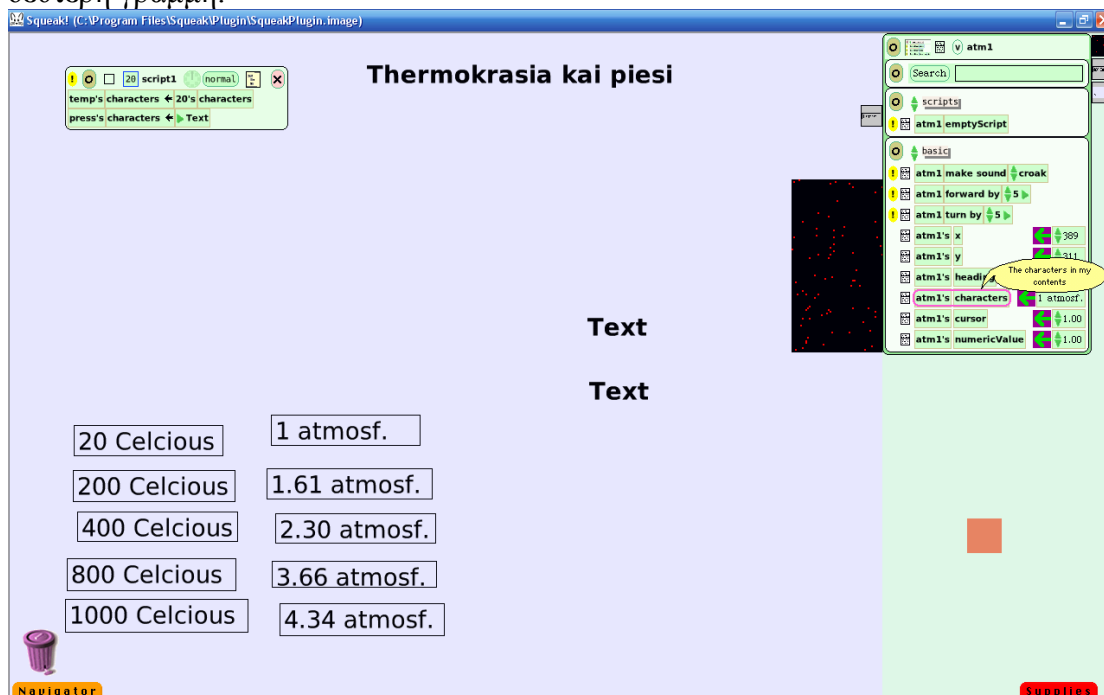


Εικόνα 78: Γραμμή 'press's characters' της κατηγορίας basic

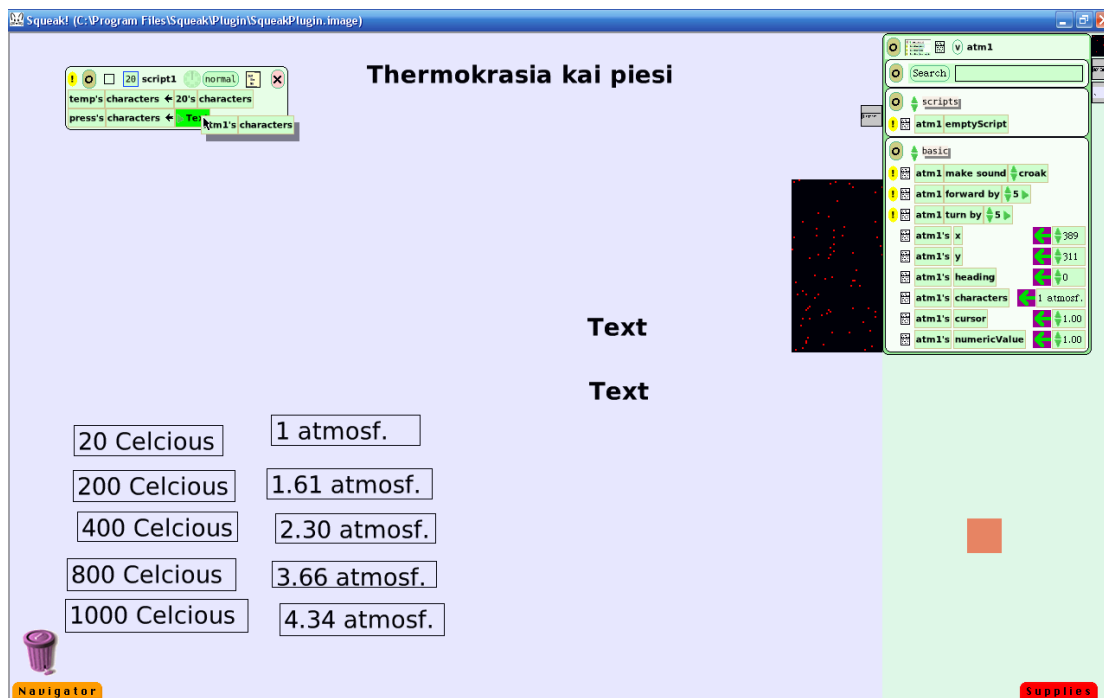


Εικόνα 79:Script του αντικειμένου '20'

Στη συνέχεια ανοίγουμε το εργαλείο 'μάτι' του αντικειμένου 'atm1' και πηγαίνουμε στην κατηγορία 'basic'. Εκεί θα σύρουμε το πεδίο 'atm1's characters' (Εικόνα 80) πάνω στη λέξη 'Text' του script του αντικειμένου '20' που έχουμε δημιουργήσει στη δεύτερη γραμμή.

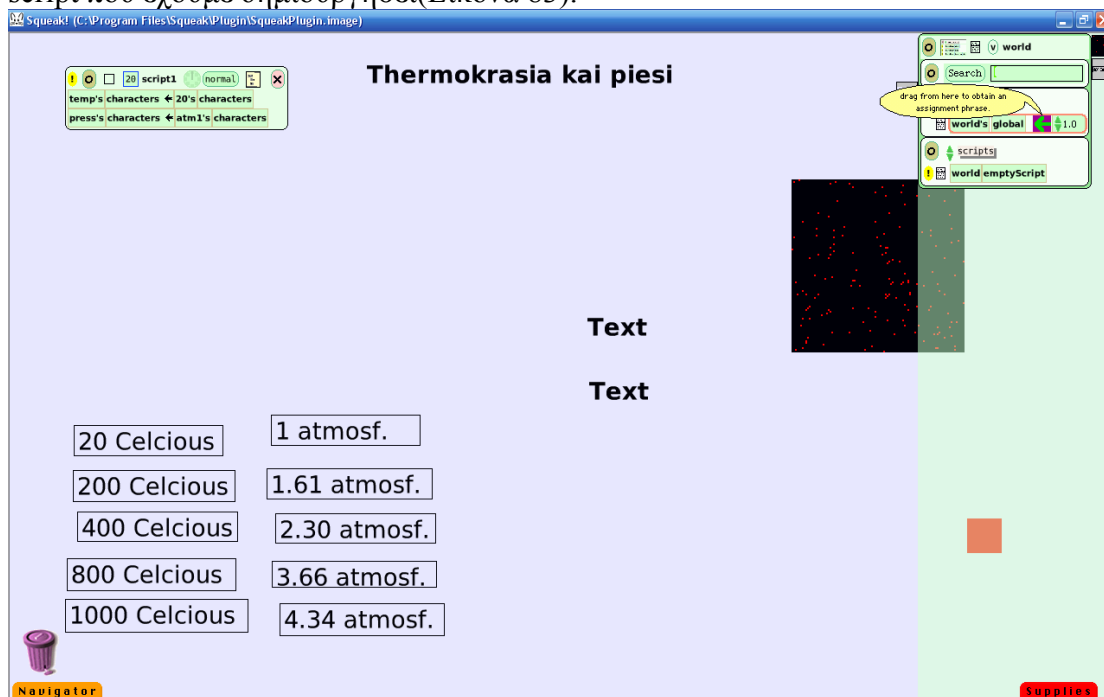


Εικόνα 80: γραμμή '20's characters' του αντικειμένου '20' βήμα 1



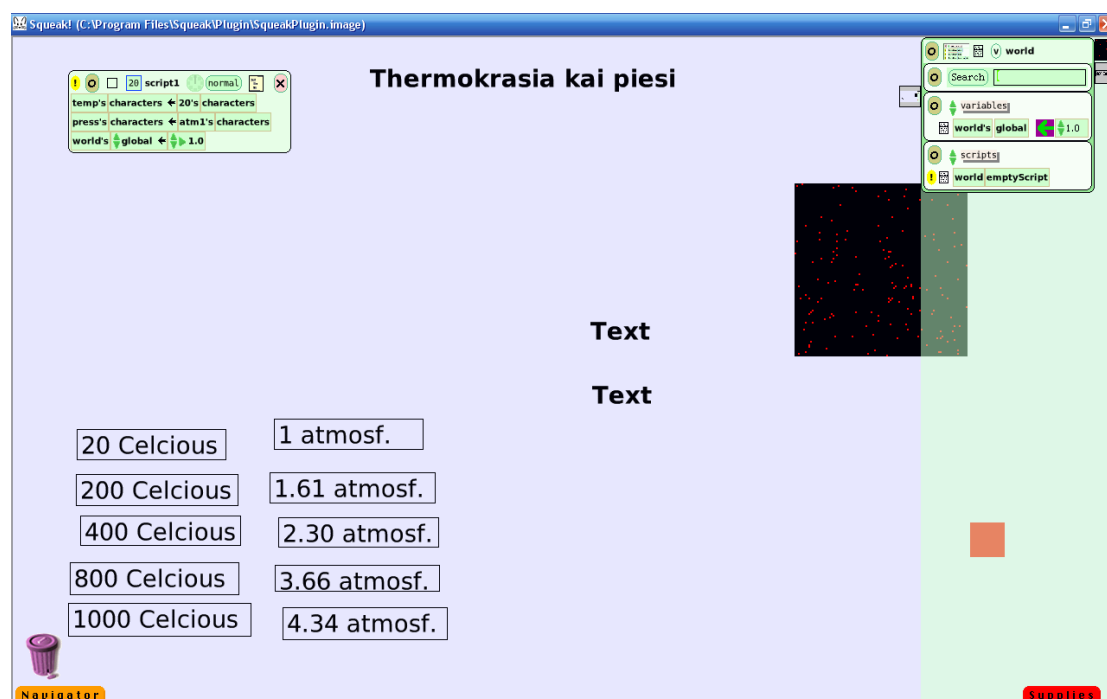
Εικόνα 81: γραμμή '20's characters' του αντικειμένου '20' βήμα 2

Ανοίγουμε στη συνέχεια το εργαλείο 'μάτι' του αντικειμένου 'world'. Εκεί πηγαίνουμε στην κατηγορία 'variables' και επιλέγουμε από τη γραμμή 'world's global' το βελάκι που δείχνει προς τα αριστερά (Εικόνα 82) και το σύρουμε μέσα στο script που έχουμε δημιουργήσει (Εικόνα 83).



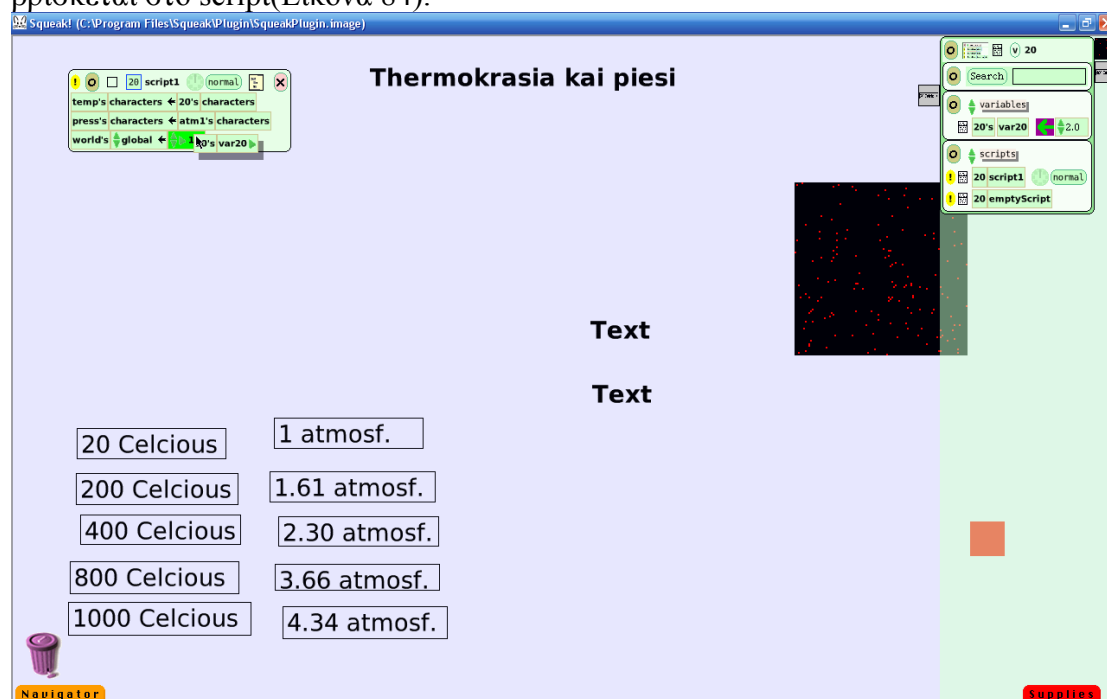
Εικόνα 82: Χρήση της μεταβλητής 'world's global' του αντικειμένου 'world'





Εικόνα 83: νέα γραμμή για το script του αντικειμένου '20'

Το επόμενο βήμα είναι να ενημερώνεται η μεταβλητή 'global' από την τιμή της μεταβλητής 'var20'. Αυτό γίνεται με το να ανοίξουμε το εργαλείο 'μάτι' του αντικειμένου '20' και να σύρουμε από την κατηγορία variables τη γραμμή '20's var20' και το τοποθετούμε πάνω στη τιμή '1.0' της μεταβλητής 'world's global' που βρίσκεται στο script (Εικόνα 84).

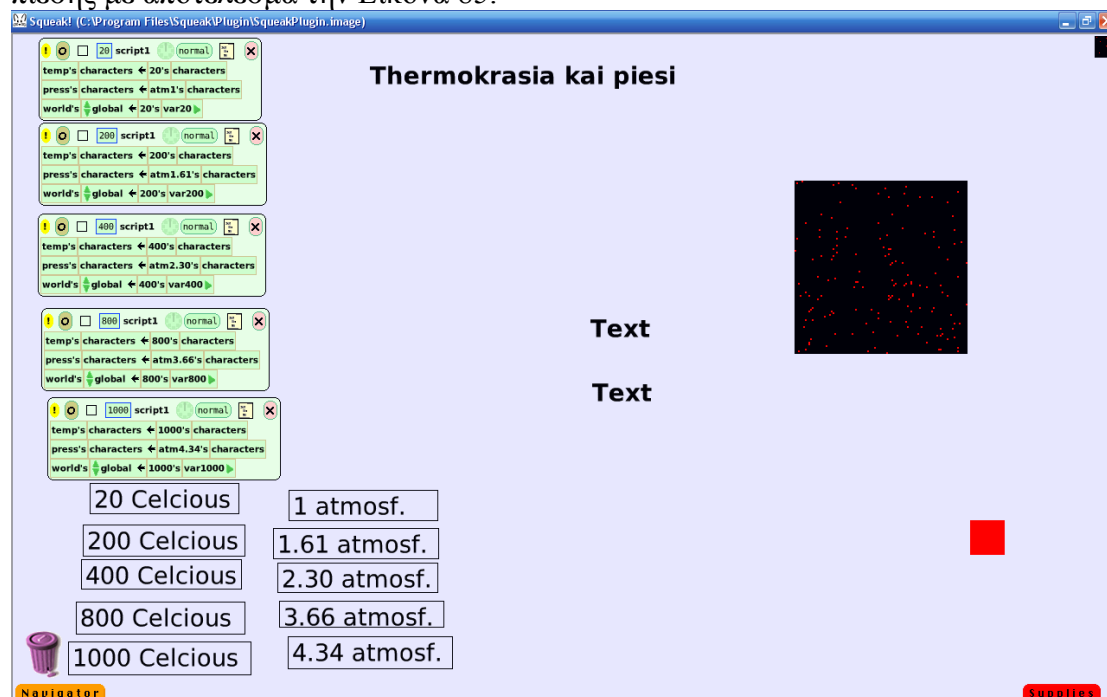


Εικόνα 84: Η μεταβλητή 'world's global' παίρνει την τιμή της μεταβλητής 'var20'

Στη συνέχεια με τον ίδιο τρόπο θα δημιουργήσουμε άλλες τέσσερις μεταβλητές μια για καθένα από τα αντικείμενα '200' '400' '800' '1000', οι οποίες για λόγους ευκολίας θα ονομαστούν 'var200', 'var400', 'var800', 'var1000' αντίστοιχα. Οι τιμές που θα δώσουμε στις μεταβλητές είναι οι εξής: var20=1, var200=2, var400=3, var800=4, var1000=5.

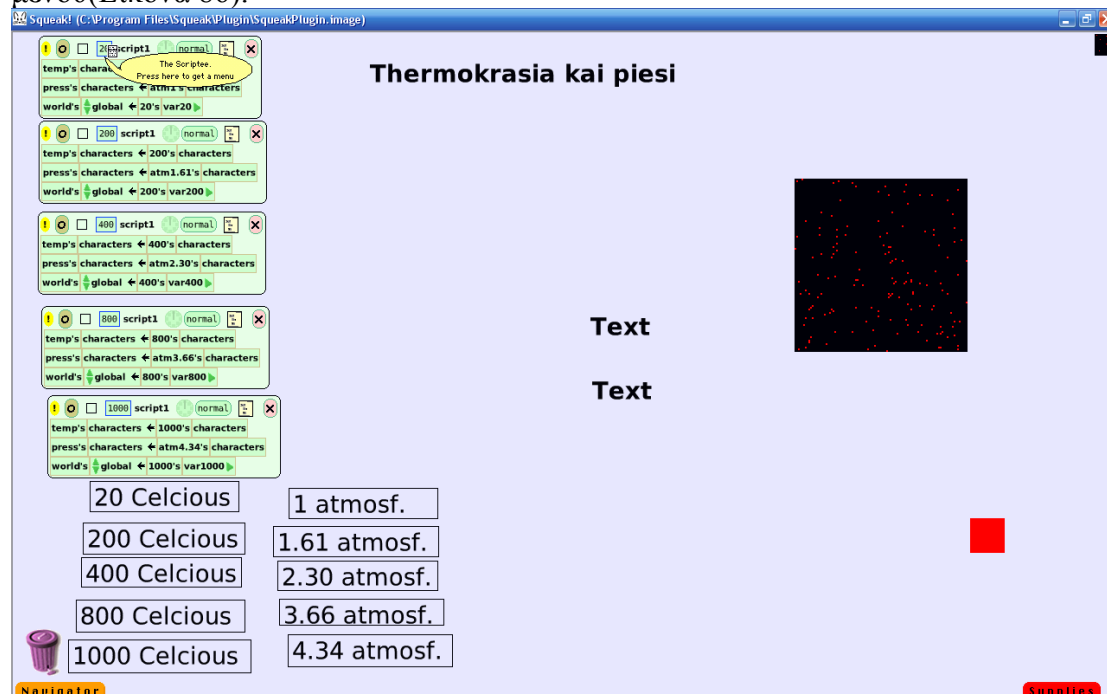
## Σχολικά Παραδείγματα

Με τον ίδιο τρόπο δημιουργούμε τα script και για τα υπόλοιπα ζεύγη θερμοκρασίας-πίεσης με αποτέλεσμα την Εικόνα 85.



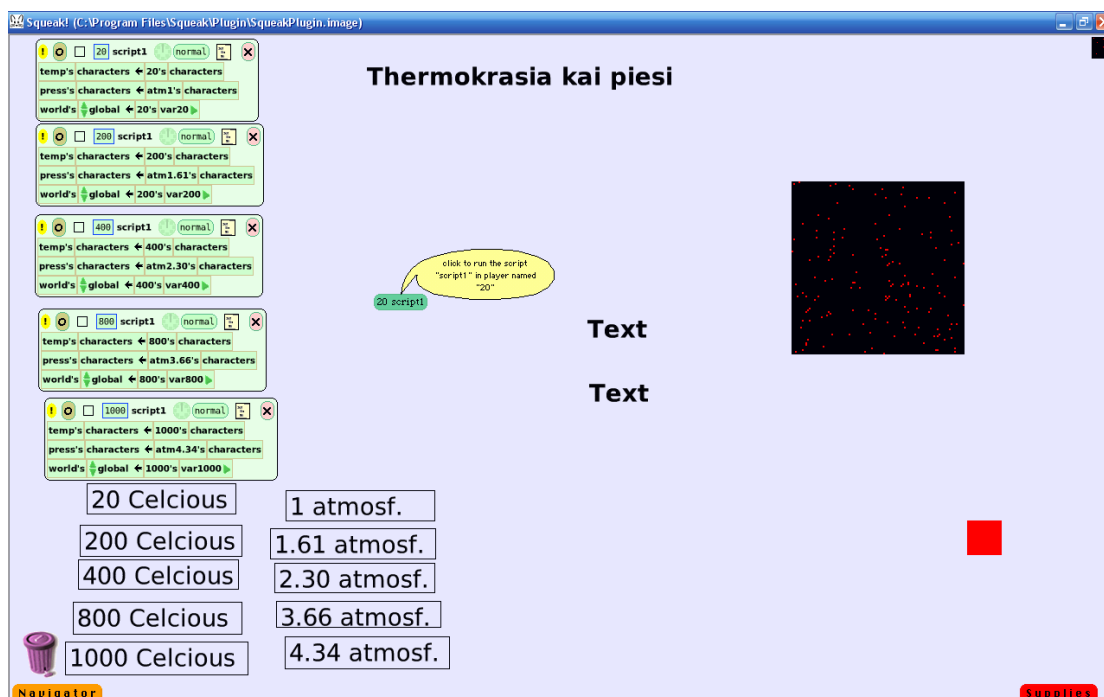
Εικόνα 85: Τα script της θερμοκρασίας-πίεσης

Κατόπιν κατασκευάζουμε τα κουμπιά από τα οποία θα τρέχουν τα script και θα εμφανίζεται η αντίστοιχη θερμοκρασία και η κίνηση των μορίων. Πηγαίνουμε στο πρώτο script που στη συγκεκριμένη περίπτωση ανήκει στο αντικείμενο '20' και κάνουμε κλικ στο όνομα του αντικειμένου(20) από το οποίο θα εμφανιστεί ένα μενού(Εικόνα 86).



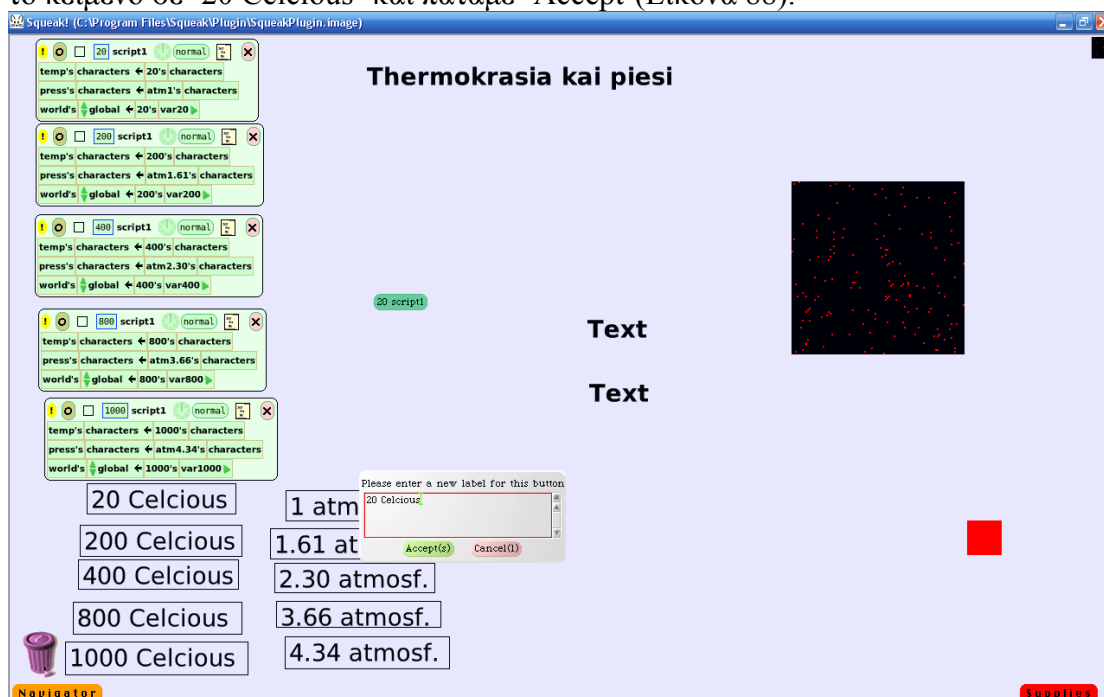
Εικόνα 86: Μενού επιλογών του script

Στο μενού αυτό επιλέγουμε τη γραμμή 'button to fire this script' και εμφανίζεται ένα κουμπί με το οποίο πατώντας το τρέχει το script(Εικόνα 87).



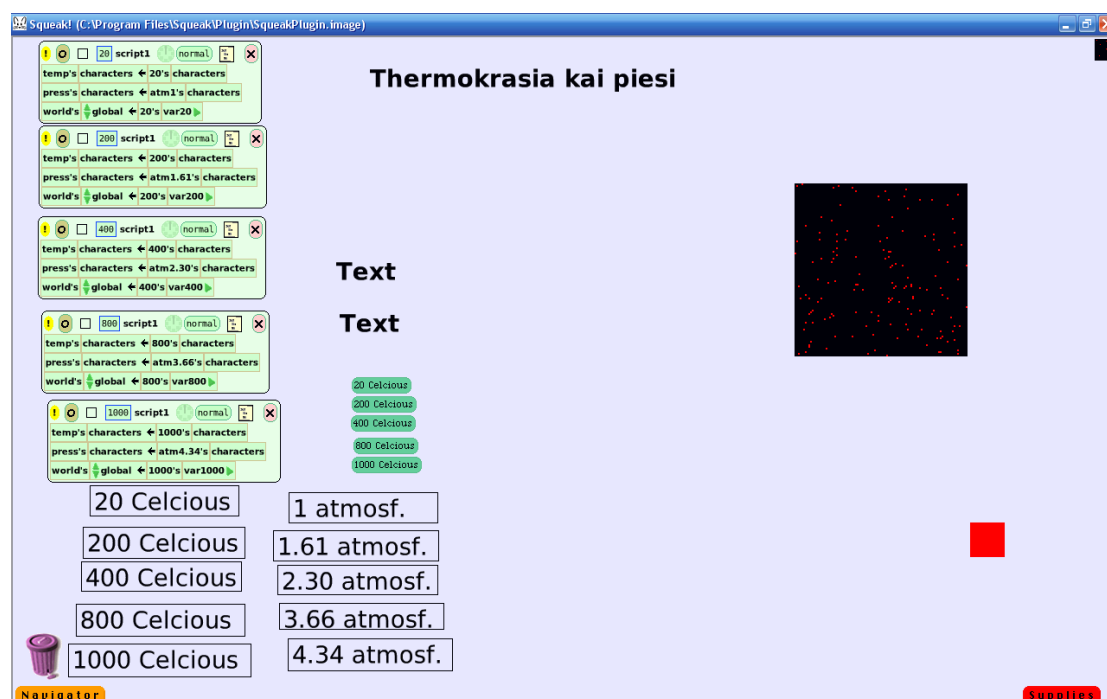
**Εικόνα 87: Το κουμπί εκτέλεσης του πρώτου script**

Ανοίγοντας το εργαλείο μενού(κόκκινο κουμπί) του κουμπιού που μόλις δημιουργήσαμε και επιλέγοντας τη γραμμή 'change label', μπορούμε να αλλάξουμε το κείμενο που εμφανίζεται μέσα στο κουμπί. Στη προκειμένη περίπτωση αλλάζουμε το κείμενο σε '20 Celcius' και πατάμε 'Accept'(Εικόνα 88).



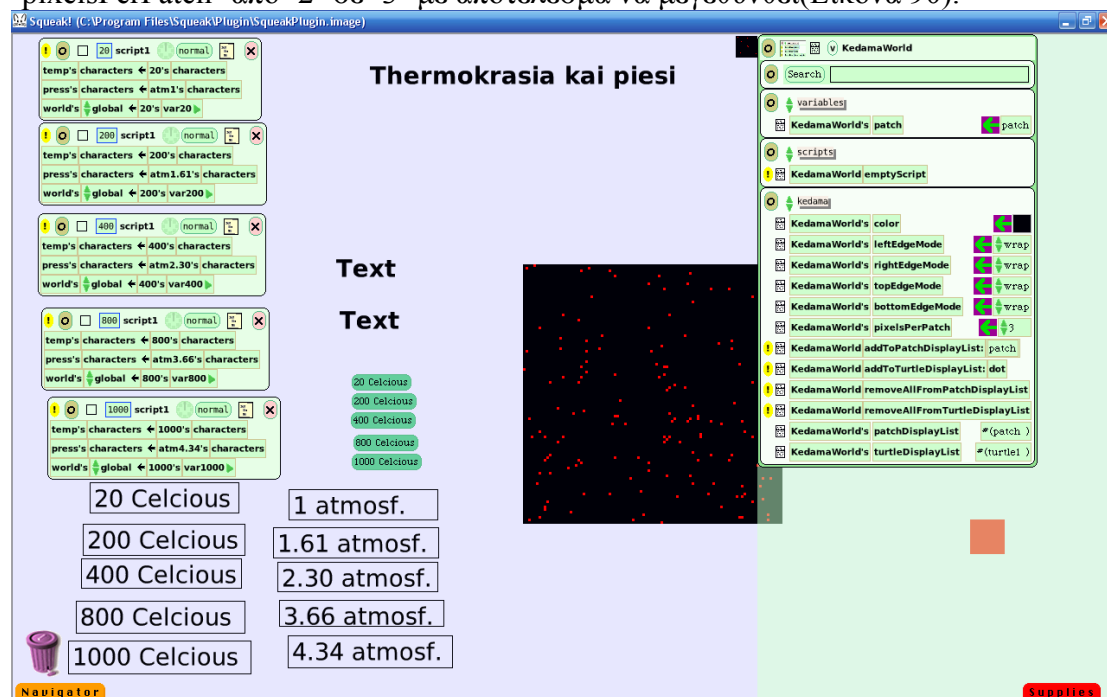
**Εικόνα 88: Αλλαγή ετικέτας κουμπιού εκτέλεσης script**

Με τον ίδιο τρόπο δημιουργούμε κουμπιά εκτέλεσης και αλλάζουμε τις ετικέτες για τα υπόλοιπα script(Εικόνα 89).



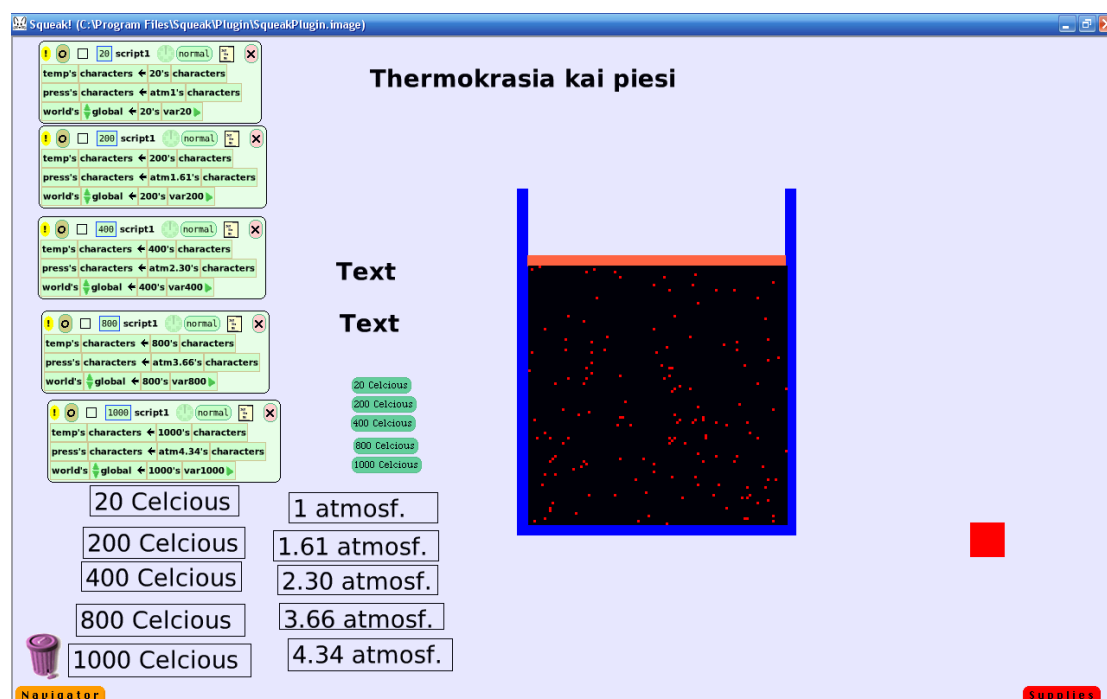
**Εικόνα 89: Ολοκλήρωση της δημιουργίας των κουμπιών εκτέλεσης script**

Στη συνέχεια θα ασχοληθούμε με την οπτική βελτίωση της κίνησης των μορίων. Θα γίνει μεγέθυνση του kedama καθώς και κατασκευή περιγράμματος έτσι ώστε να φαίνεται σαν δοχείο που περιέχει μόρια αερίου. Ανοίγουμε το εργαλείο μάτι του αντικειμένου 'Kedama World' και στην κατηγορία 'kedama' αλλάζουμε το 'pixelsPerPatch' από '2' σε '3' με αποτέλεσμα να μεγεθυνθεί (Εικόνα 90).



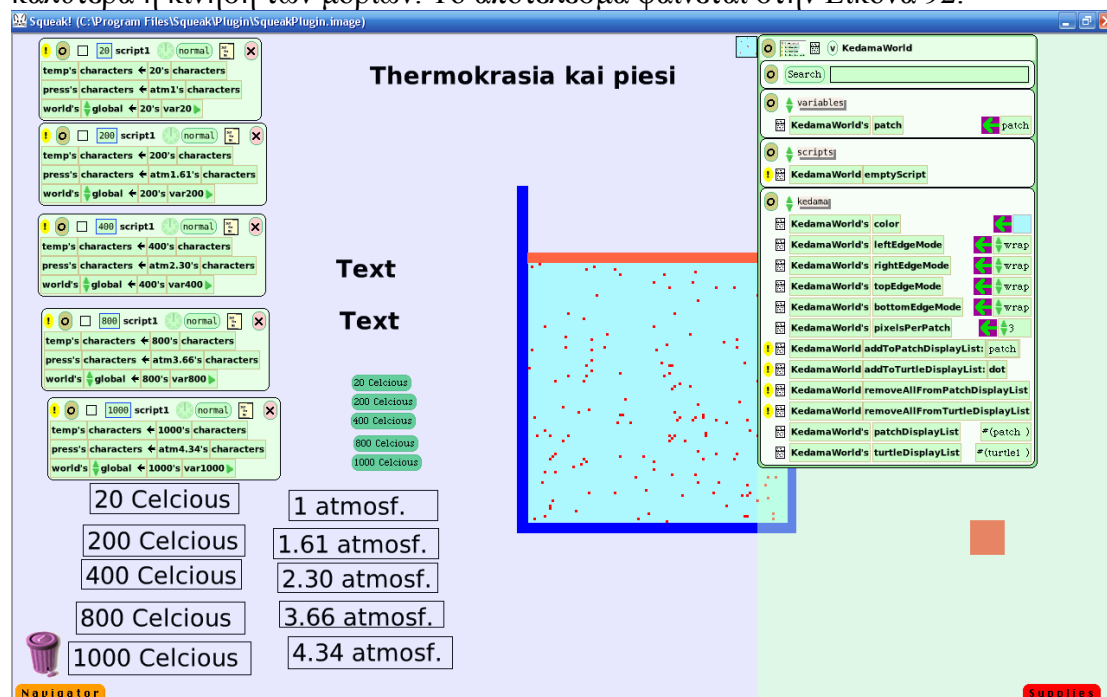
**Εικόνα 90: Αλλαγή μεγέθους του αντικειμένου 'Kedama World'**

Στη συνέχεια με το εργαλείο της ζωγραφικής κατασκευάζουμε ένα δοχείο το οποίο φαίνεται στην Εικόνα 91. Η κατασκευή έγινε ως εξής: αρχικά με το εργαλείο τετράγωνο της ζωγραφικής σχεδιάσαμε ένα ορθογώνιο το οποίο αντιστοιχούσε στις διαστάσεις του kedama. Με τη γόμα σβήσαμε την πάνω πλευρά και με το εργαλείο γραμμή καλύψαμε την πάνω πλευρά του kedama έτσι ώστε να φαίνεται ότι τα μόρια βρίσκονται σε κλειστό αέριο.



**Εικόνα 91: Κατασκευή δοχείου για το αέριο**

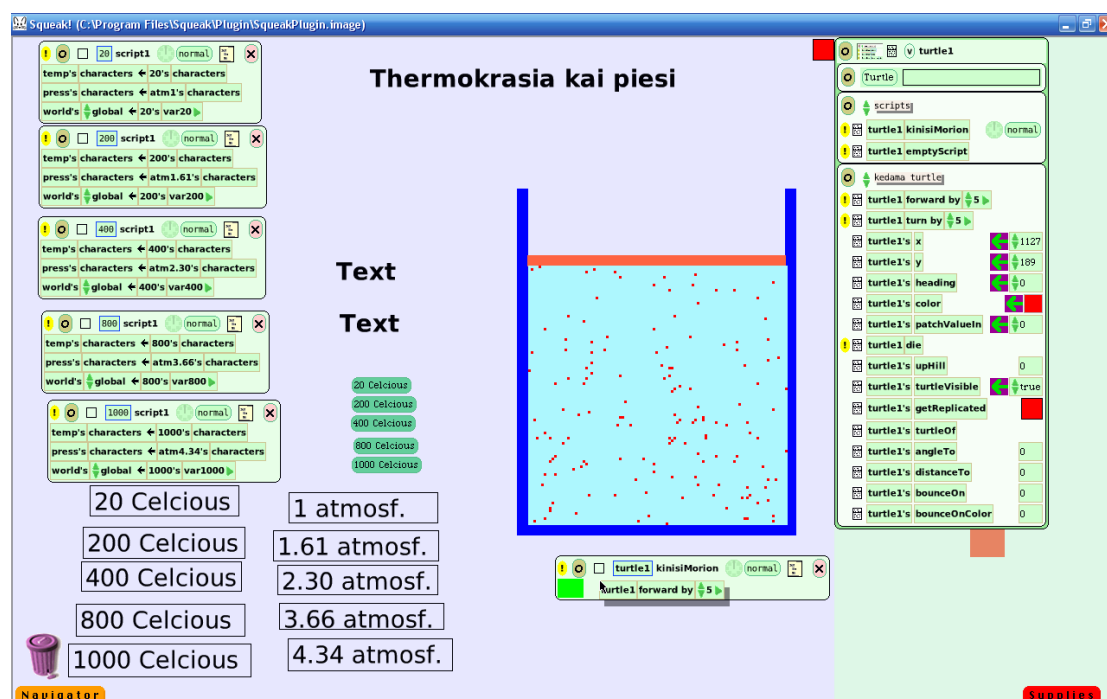
Επιλέγοντας το εργαλείο μάτι του αντικειμένου 'Kedama World' στην κατηγορία 'kedama' αλλάζουμε το χρώμα και επιλέγουμε ένα πιο ανοιχτό έτσι ώστε να φαίνεται καλύτερα η κίνηση των μορίων. Το αποτέλεσμα φαίνεται στην Εικόνα 92.



**Εικόνα 92: Αλλαγή χρώματος του αντικειμένου 'Kedama World'**

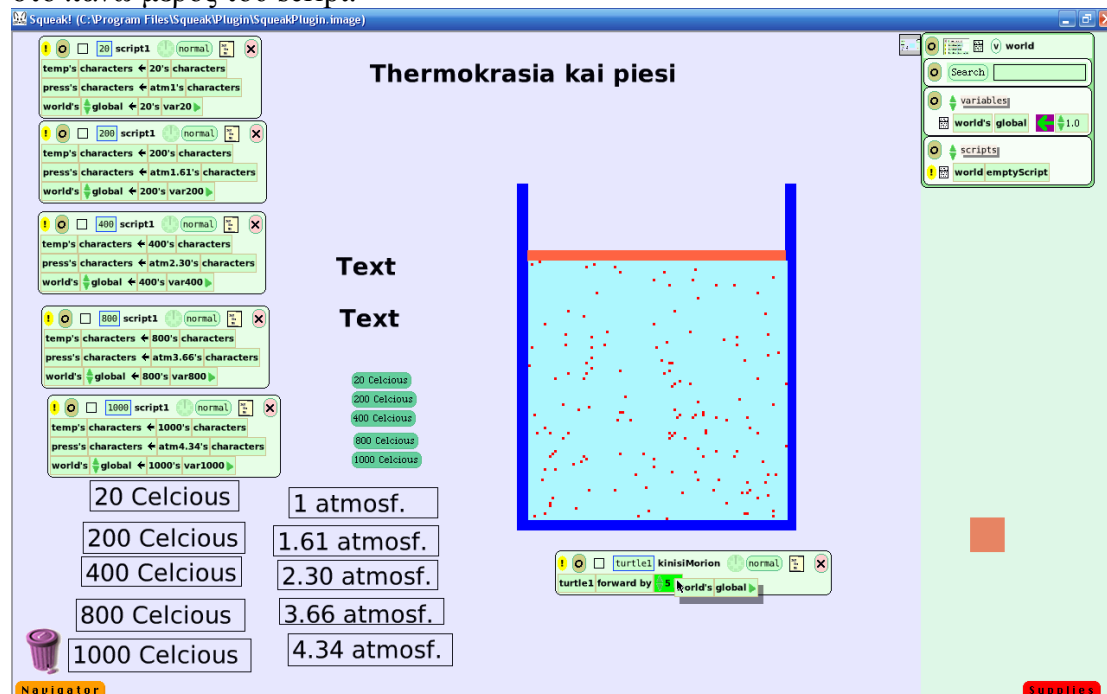
Ένα από τα τελικά στάδια είναι η δημιουργία του script για την κίνηση των μορίων, το οποίο θα ενεργοποιεί και θα ελέγχει την κίνηση των μορίων, με βάση την επιθυμητή θερμοκρασία. Ανοίγουμε το εργαλείο 'μάτι' του αντικειμένου 'turtle1' και στην κατηγορία 'scripts' τραβάμε ένα 'emptyScript' στην επιφάνεια του προγράμματος το οποίο το ονομάζουμε 'kinisiMorion'.

Στη συνέχεια στην κατηγορία 'kedamaturtle' του αντικειμένου 'turtle1' στην γραμμή 'turtle1 forward by' κάνουμε κλικ και το σύρουμε μέσα στο script που μόλις δημιουργήσαμε (Εικόνα 93).



Εικόνα 93: Δημιουργία script για την κίνηση των μορίων

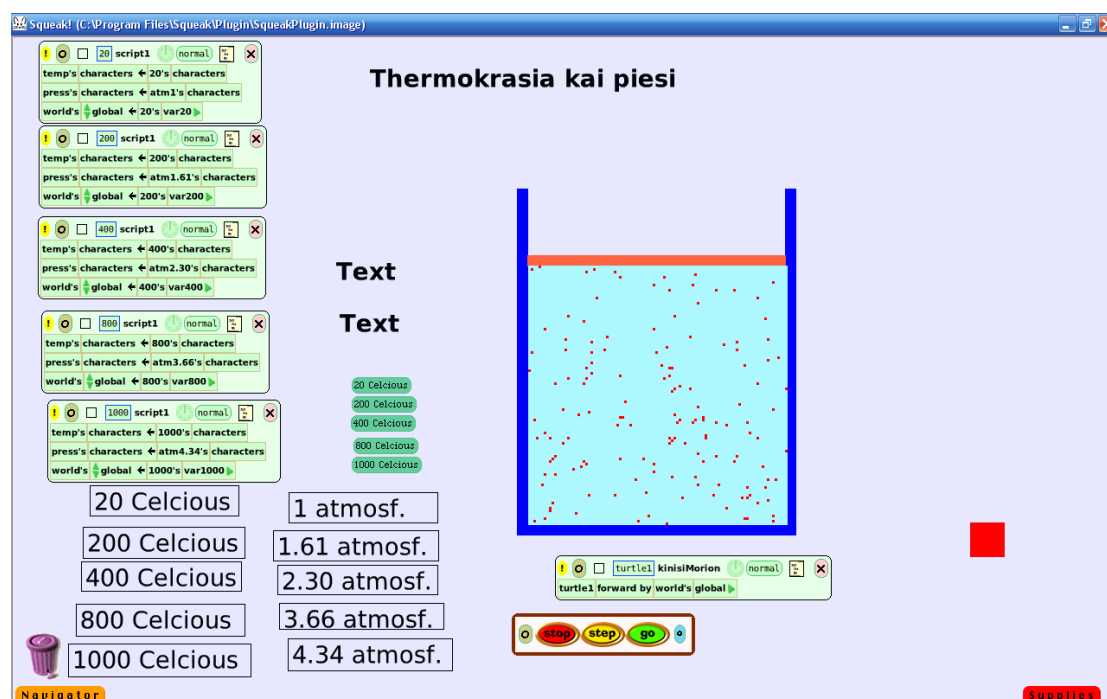
Στη συνέχεια ανοίγουμε το εργαλείο ‘μάτι’ του αντικειμένου ‘world’ και πηγαίνουμε στην κατηγορία ‘variables’ στην γραμμή ‘world’s global’, αλλάζουμε την τιμή της σε ‘0’ και σύρουμε όλη τη γραμμή μέσα στο script ‘kinisiMorion’ και το αφήνουμε στο σημείο που γράφει ‘5’ (Εικόνα 94). Στη συνέχεια αλλάζουμε τον τρόπο εκτέλεσης του script από ‘normal’ σε ‘paused’. Αυτό γίνεται κάνοντας κλικ στο ρολόι που βρίσκεται στο πάνω μέρος του script.



Εικόνα 94: Ολοκλήρωση script ‘kinisiMorion’

Στη καρτέλα ‘Supplies’ επιλέγουμε ένα αντικείμενο ‘All Scripts’ και το σύρουμε στην επιφάνεια του προγράμματος. Το αποτέλεσμα φαίνεται στην Εικόνα 95.





Εικόνα 95: Χρήση αντικειμένου 'All Scripts'

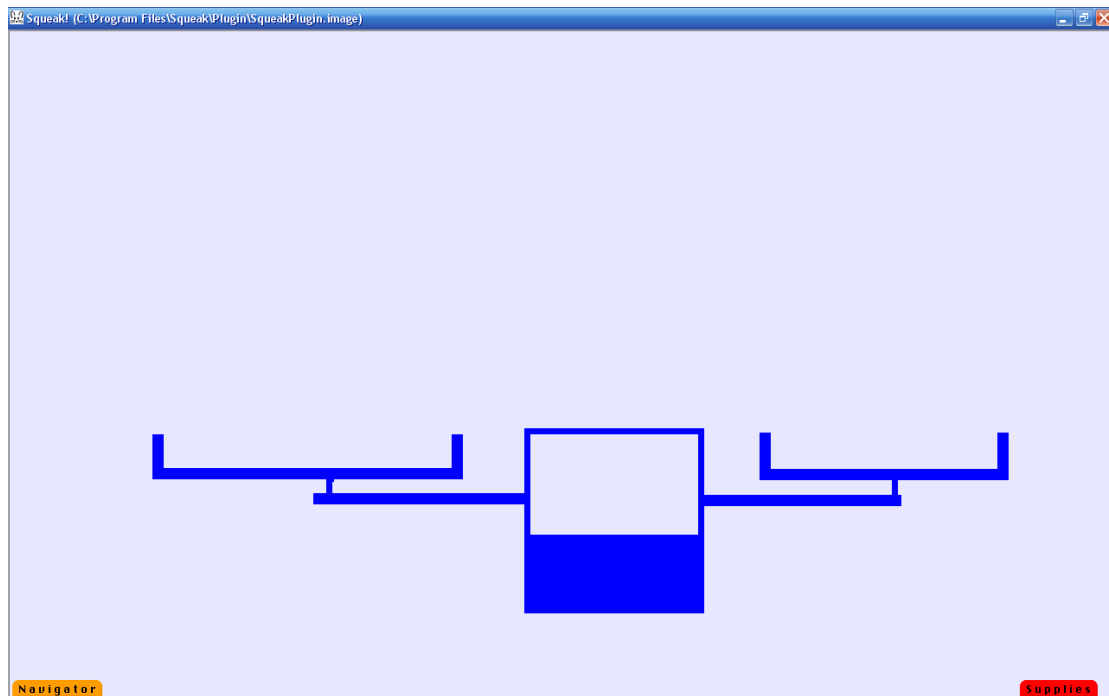
Το επόμενο βήμα είναι να εξαφανίσουμε από την οθόνη του προγράμματος όλα τα αντικείμενα τα οποία δεν είναι απαραίτητο να είναι εμφανή στον τελικό χρήστη. Για το λόγο αυτό επιλέγουμε το εργαλείο 'κύκλος(collapse)' στα αντικείμενα που αντιπροσωπεύουν τη θερμοκρασία και την πίεση. Με αυτό τον τρόπο ελαχιστοποιούνται στο πάνω αριστερό μέρος της οθόνης. Με τον ίδιο τρόπο πατάμε το κουμπί 'κύκλος' που βρίσκεται στο πάνω αριστερό μέρος του script και το script χάνεται από την οθόνη. Εάν θέλουμε να εμφανίσουμε κάποιο αντικείμενο κάνουμε κλικ στο κύκλο που βρίσκεται στο δεξί μέρος της ονομασίας του ελαχιστοποιημένου αντικειμένου ενώ αν θέλουμε να εμφανίσουμε κάποιο script, επιλέγουμε το εργαλείο μάτι του αντικειμένου στο οποίο ανήκει, και πηγαίνουμε στην κατηγορία scripts και το σύρουμε στην επιφάνεια. Το επιθυμητό αποτέλεσμα φαίνεται στην .

Το πρόγραμμα είναι έτοιμο. Για να το τρέξουμε πατάμε 'go', διαλέγουμε κάποια θερμοκρασία, αυτόματα εμφανίζεται η αντίστοιχη πίεση και απεικονίζεται παράλληλα και η κίνηση των μορίων.

### 3.3. Ζυγαριά κλασμάτων

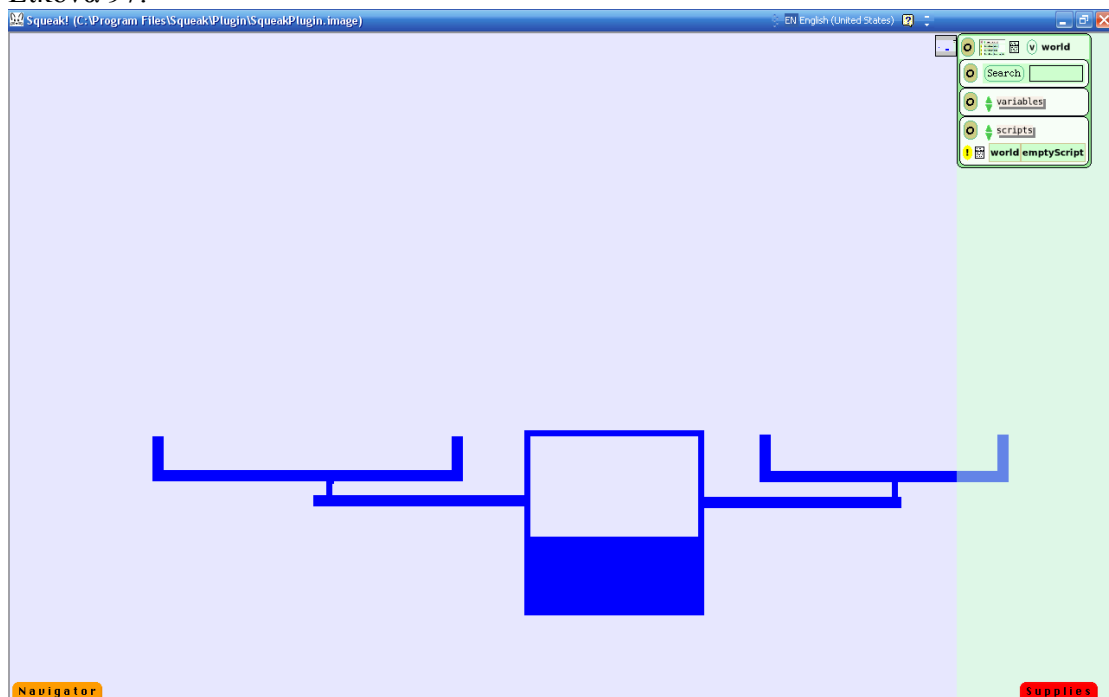
Σκοπός του παραδείγματος αυτού είναι να κατανοήσουν τα παιδιά το αριθμητικό μέγεθος των κλασμάτων μέσω της σύγκρισης που προσφέρει το όργανο της ζυγαριάς. Τα παιδιά θα τοποθετούν τα κλάσματα που επιθυμούν πάνω στη ζυγαριά, θα σκέφτονται το πιθανό αποτέλεσμα και πατώντας το κουμπί εκτέλεσης του προγράμματος θα απεικονίζεται η κλίση της ζυγαριάς. Με τον τρόπο αυτό ο μαθητής θα παίρνει τη σωστή απάντηση και με αντίστοιχο κουμπί θα έχει τη δυνατότητα να αρχικοποιήσει την κατάσταση του προγράμματος ούτως ώστε να το ξαναεκτελέσει αργότερα.

Ξεκινάμε με τη σχεδίαση της ζυγαριάς του προγράμματος. Χρησιμοποιώντας το εργαλείο της ζωγραφικής από την καρτέλα 'Navigator' σχεδιάζουμε μια ζυγαριά το αποτέλεσμα της οποίας φαίνεται στην Εικόνα 96.



**Εικόνα 96: Ο σχεδιασμός της ζυγαριάς**

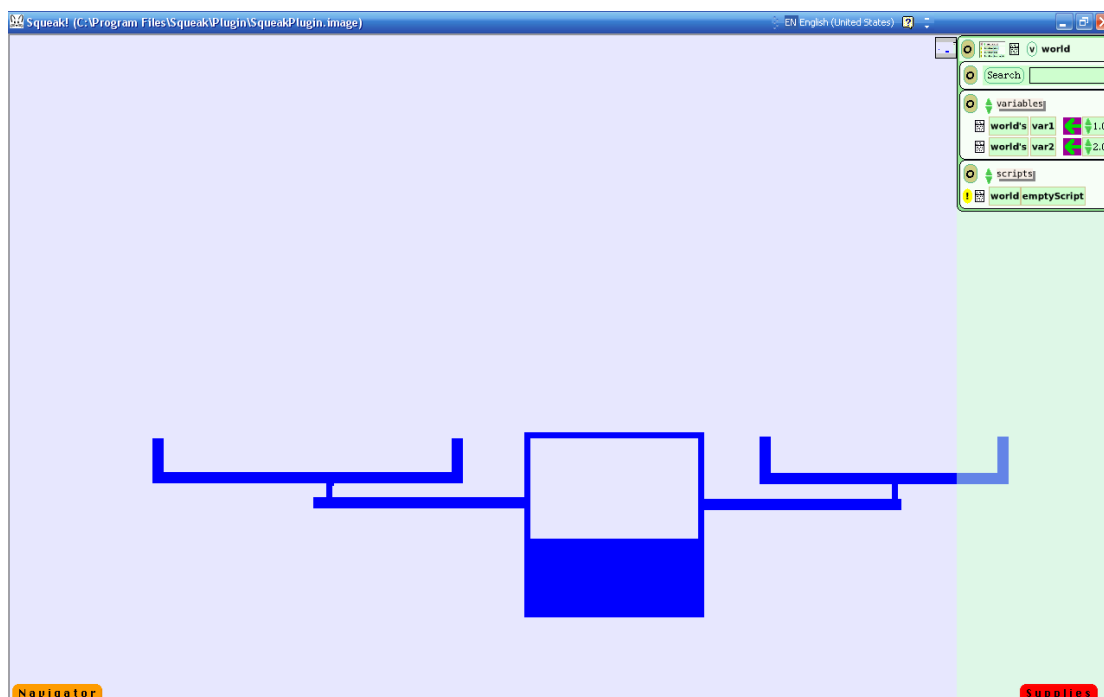
Στη συνέχεια πατάμε 'alt' και 'κλικ' στην επιφάνεια του προγράμματος σε περιοχή που δεν περιλαμβάνει άλλο αντικείμενο και επιλέγουμε το εργαλείο 'μάτι' με σκοπό να εμφανίσουμε το μενού του αντικειμένου 'world'. Το αποτέλεσμα φαίνεται στην Εικόνα 97.



**Εικόνα 97: Το εργαλείο 'μάτι' του αντικειμένου 'world'**

Πηγαίνουμε στο κουμπί 'v' που βρίσκεται στο πάνω μέρος του μενού, και κάνουμε 'κλικ'. Στο παράθυρο που προκύπτει γράφουμε το όνομα που θέλουμε να δώσουμε στην νέα μεταβλητή. Το όνομα που δίνουμε είναι 'var1'. Με τον ίδιο τρόπο δημιουργούμε και άλλη μια μεταβλητή 'var2' και το τελικό αποτέλεσμα φαίνεται στην Εικόνα 98.

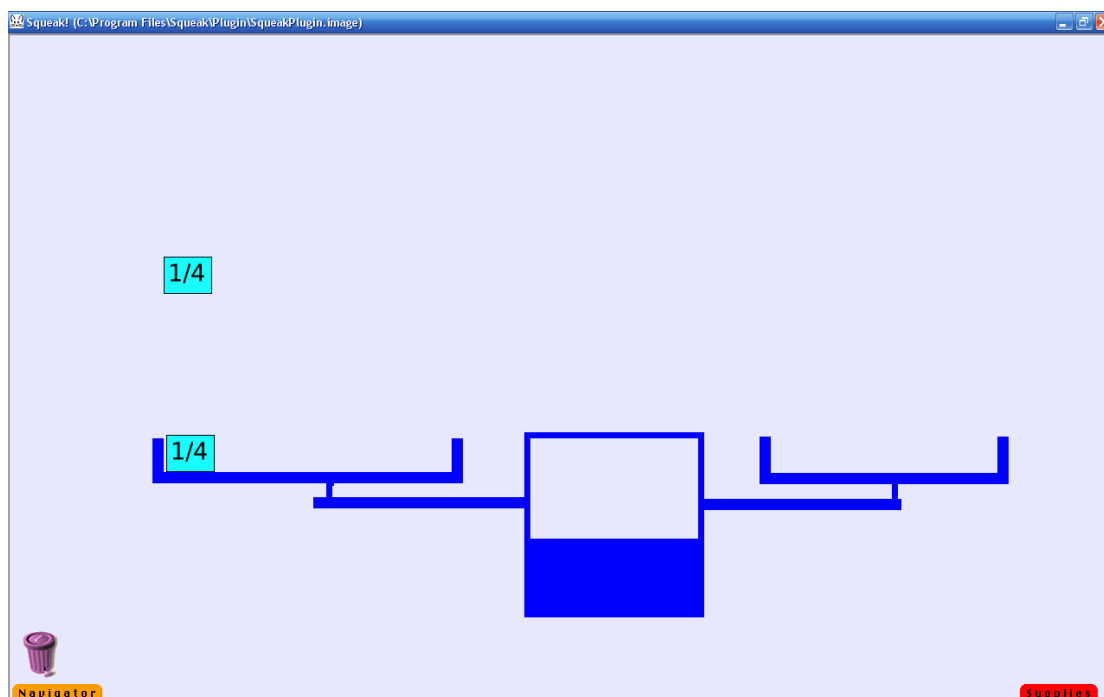




**Εικόνα 98: Οι μεταβλητές 'var1' και 'var2' του αντικειμένου 'world'**

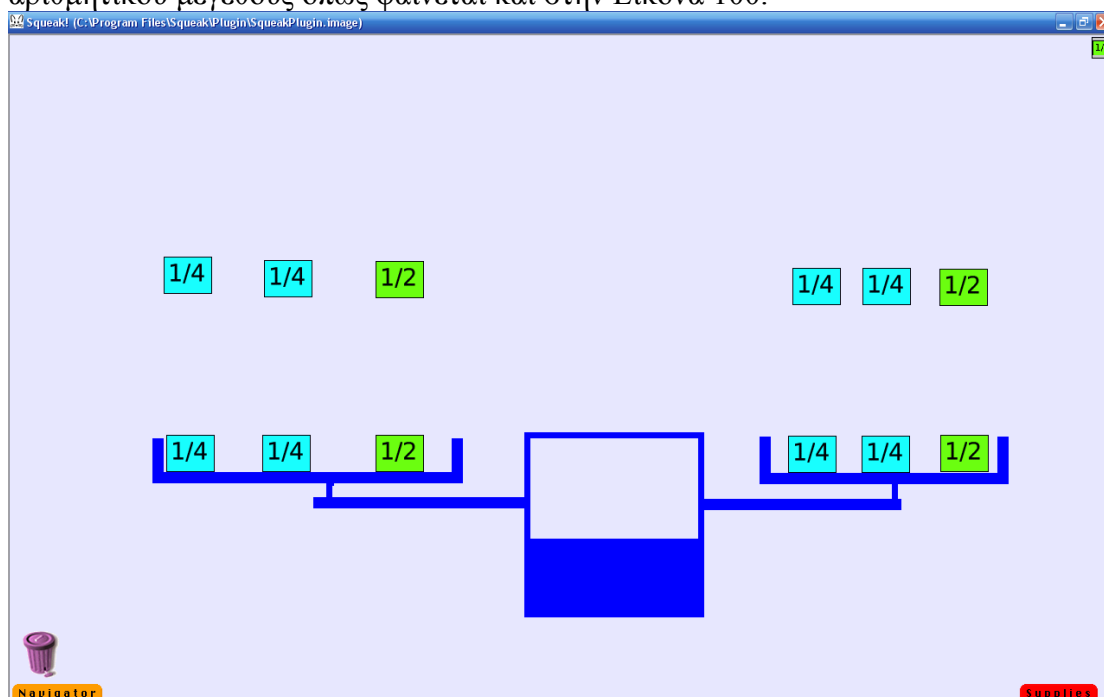
Σκοπός δημιουργίας των μεταβλητών αυτών είναι να συμμετέχουν στην σύγκριση μεταξύ της αριστερής και της δεξιάς πλευράς της ζυγαριάς και να κατευθύνουν το βέλος της, στη κατάλληλη κατεύθυνση.

Το επόμενο βήμα είναι η δημιουργία των κλασμάτων που θα τοποθετούνται πάνω στη ζυγαριά. Σε κάθε πλευρά της ζυγαριάς θα έχουμε δύο κλάσματα του '1/4' και ένα του '1/2'. Τα κλάσματα θα είναι αντικείμενα 'Text', θα έχουν ταμπέλα που θα γράφουν το κλάσμα και αντικείμενα ίδιου κλάσματος θα έχουν το ίδιο χρώμα. Επιλέγουμε από την καρτέλα 'Supplies' το αντικείμενο 'Object Catalog'. Εκεί ανοίγουμε την καρτέλα 'Text' και σύρουμε ένα αντικείμενο 'Text (border)'. Μέσα στο αντικείμενο αυτό, αλλάζουμε το κείμενο από 'abc' σε '1/4'. Στη συνέχεια πατάμε το ροζ εργαλείο του αντικειμένου αυτού και επιλέγουμε το χρώμα της αρεσκείας μας από την πάνω αριστερή παλέτα χρωμάτων. Μόλις ολοκληρώσουμε την επιλογή του χρώματος, πατάμε το κουμπί 'Accept' και στη συνέχεια με το πράσινο εργαλείο δημιουργούμε ένα νέο όμοιο αντικείμενο. Το αποτέλεσμα φαίνεται στην Εικόνα 99.



**Εικόνα 99: Δημιουργία του πρώτου κλάσματος της ζυγαριάς**

Τα δύο αυτά αντικείμενα θα εναλλάσσουν την εμφάνισή τους δημιουργώντας στο μαθητή την εντύπωση ότι διαλέγοντας το κλάσμα, το κλάσμα αυτό πέφτει πάνω στη ζυγαριά. Με το ίδιο τρόπο δημιουργούμε και άλλα κλάσματα ίδιου ή διαφορετικού αριθμητικού μεγέθους όπως φαίνεται και στην Εικόνα 100.

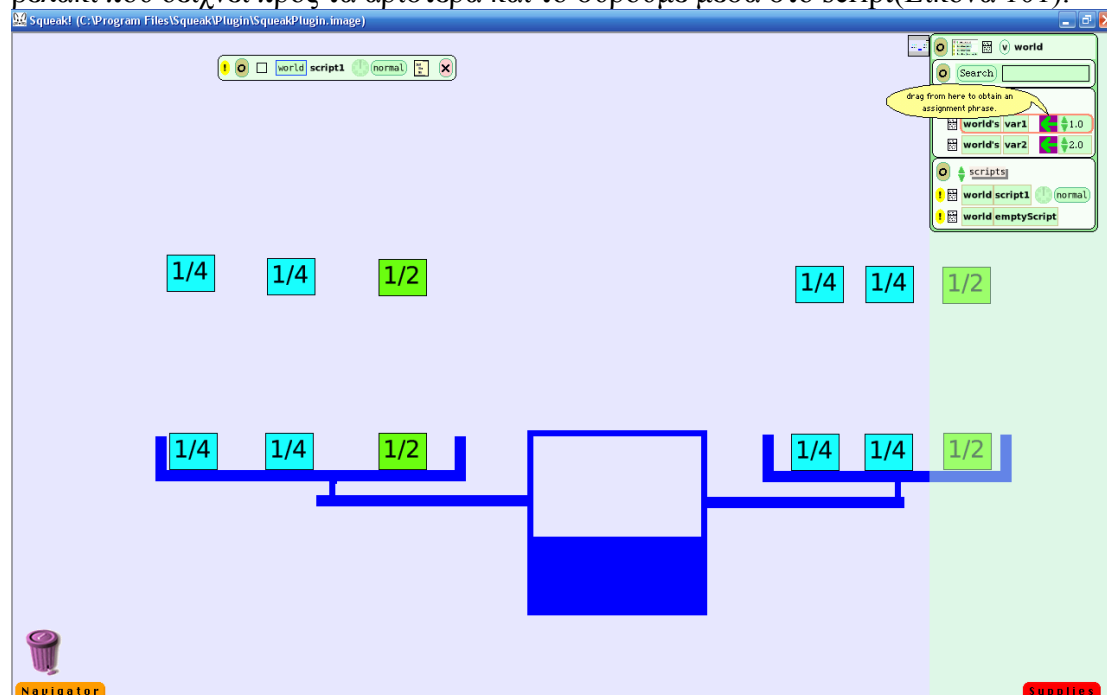


**Εικόνα 100: Όλα τα κλάσματα του προγράμματος**

Στη συνέχεια θα δώσουμε σε κάθε ένα αντικείμενο, ένα συγκεκριμένο όνομα έτσι ώστε να μας βοηθήσει στη δημιουργία των script καθώς και μας βοηθήσει στη δημιουργία όλου του υπόλοιπου προγράμματος. Για ευκολία ονομάζουμε τα αντικείμενα ως εξής: τα αντικείμενα της πάνω σειράς παίρνουν τα ονόματα '1' ως '6' από αριστερά προς τα δεξιά ενώ τα αντικείμενα της κάτω σειράς παίρνουν τα ονόματα '1.1' ως '1.6'.

## Σχολικά Παραδείγματα

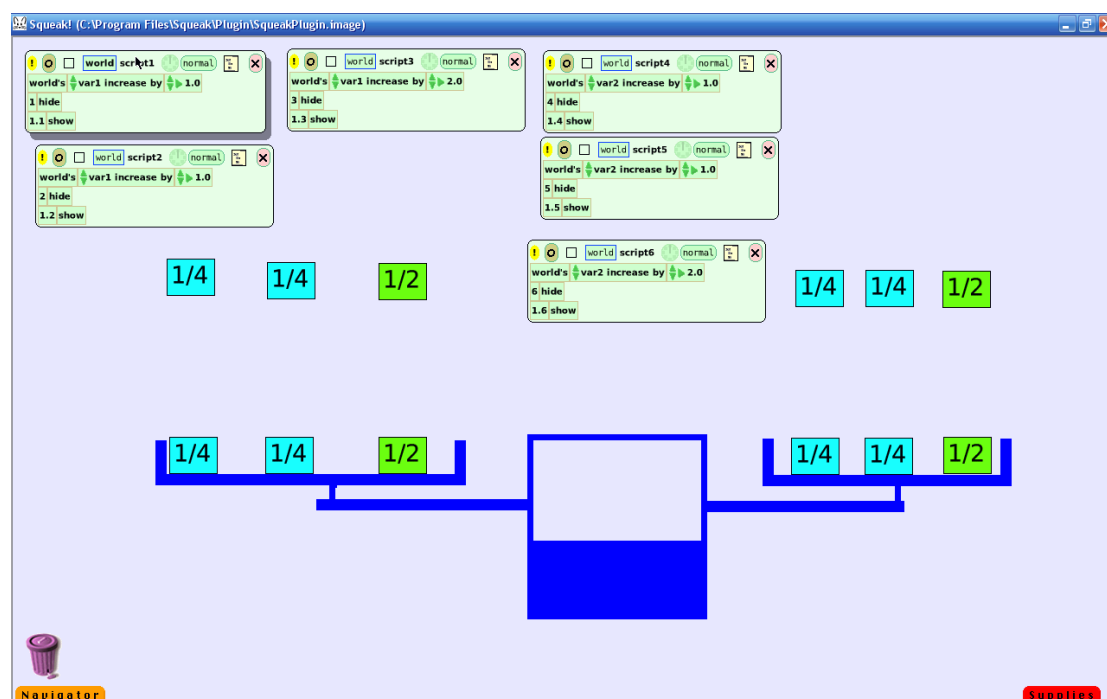
Στη συνέχεια πατάμε ‘alt’ και ‘κλικ’ και στον ελεύθερο χώρο του προγράμματος και ανοίγουμε το εργαλείο ‘μάτι’ του αντικειμένου ‘world’. Από το μενού που προκύπτει, τραβάμε ένα ‘worldemptyScript’. Το script αυτό ονομάζεται ‘world script1’. Κατόπιν πηγαίνουμε στην κατηγορία ‘variables’ και τραβάμε τη γραμμή ‘world’s var1’ από το βελάκι που δείχνει προς τα αριστερά και το σύρουμε μέσα στο script (Εικόνα 101).



Εικόνα 101: Η μεταβλητή ‘world’s var1’ στο script ‘world script1’

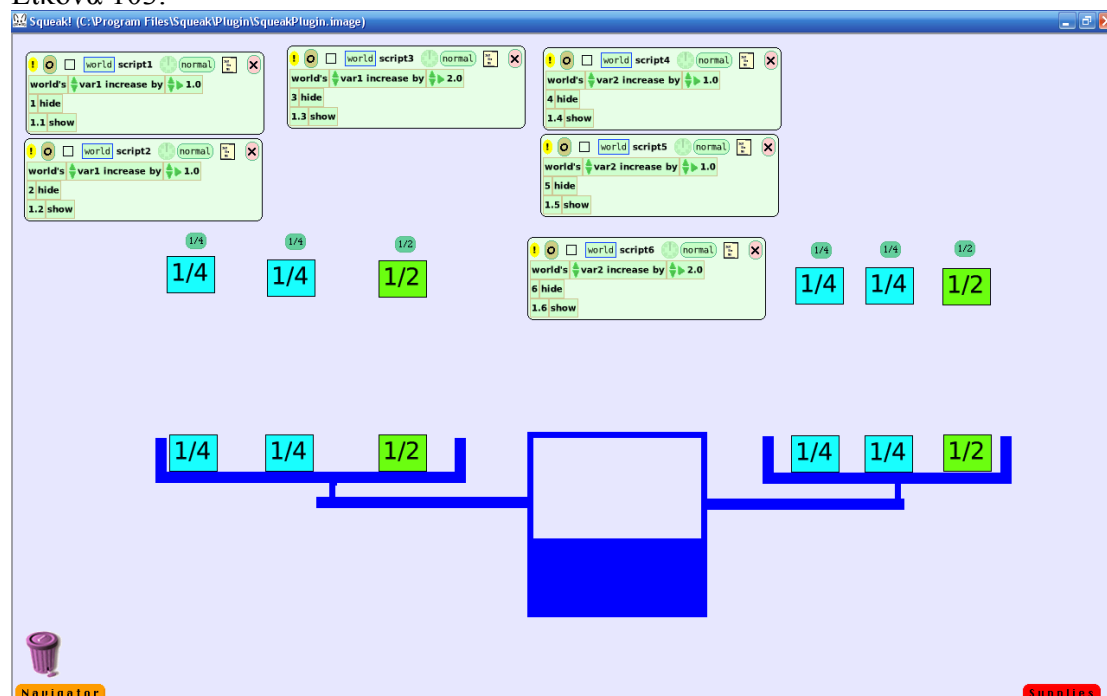
Τη γραμμή αυτή του script την αλλάζουμε σε ‘world’s var1 increase by 1’. Αυτό που επιθυμούμε είναι ότι όταν ο μαθητής επιλέγει ένα κλάσμα, να σβήνεται το κλάσμα που βρίσκεται στην πάνω γραμμή και να εμφανίζεται το αντίστοιχο κλάσμα που βρίσκεται πάνω στη ζυγαριά. Για να το επιτύχουμε αυτό ανοίγουμε το εργαλείο ‘μάτι’ του αντικειμένου ‘1’ και πηγαίνουμε στη κατηγορία ‘miscellaneous’. Από κει τραβάμε τη γραμμή ‘1hide’ και τη σύρουμε μέσα στο script. Αντίστοιχα ανοίγουμε το αντικείμενο ‘1.1’ και πηγαίνουμε στη κατηγορία ‘miscellaneous’. Εκεί τραβάμε τη γραμμή ‘1.1 show’ και τη βάζουμε στο script. Έτσι όταν εκτελέσουμε το script αυτό, θα κρυφθεί το πάνω κλάσμα, θα εμφανιστεί το κάτω και επίσης θα αυξηθεί η τιμή της μεταβλητής ‘var1’ κατά ‘1’. Με τον ίδιο τρόπο κατασκευάζουμε τα αντίστοιχα script για τα υπόλοιπα κλάσματα του αριστερού μέρους της ζυγαριάς.

Παρομοίως, κατασκευάζουμε και τα κλάσματα της δεξιάς πλευράς της ζυγαριάς με τη διαφορά ότι αντί να χρησιμοποιήσουμε τη μεταβλητή ‘world’s var1’ θα χρησιμοποιήσουμε τη μεταβλητή ‘world’s var2’. Ένα στοιχείο που πρέπει να προσέξουμε είναι στα μεν ‘1/4’ η αύξηση της μεταβλητής πρέπει να είναι ‘1’ ενώ στα ‘1/2’ θα πρέπει να είναι ‘2’. Το τελικό αποτέλεσμα φαίνεται στην Εικόνα 102.



**Εικόνα 102: script επιλογής κλάσματος**

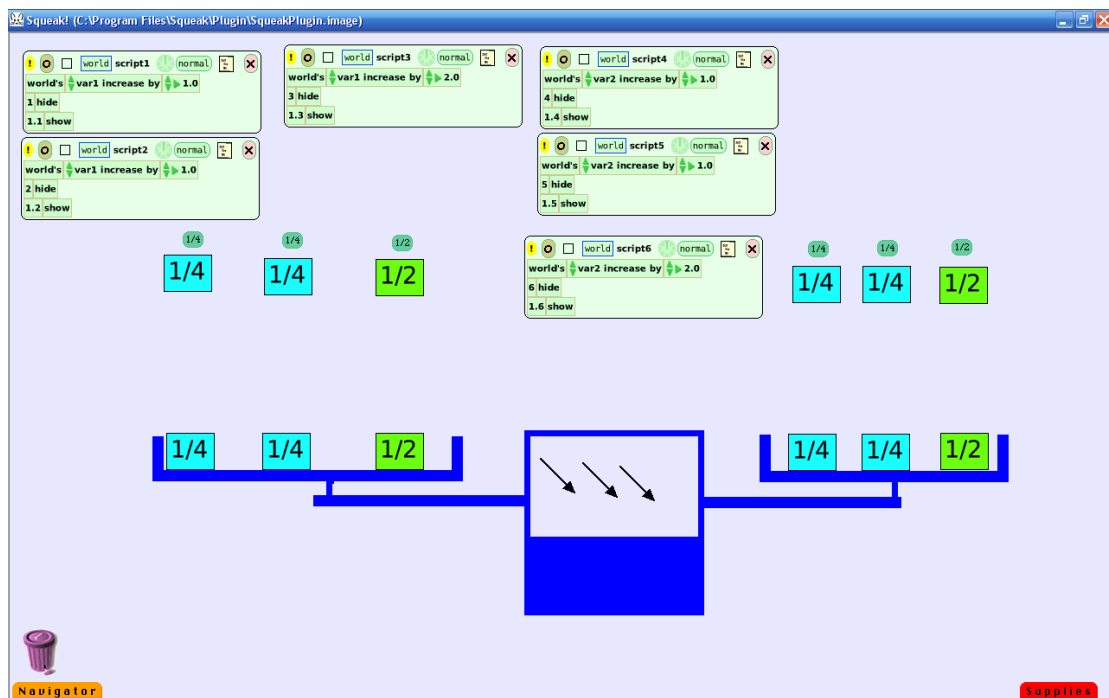
Κατόπιν θα δημιουργήσουμε τα κουμπιά για τη λειτουργία αυτών των script. Πηγαίνουμε στο script και πατάμε πάνω στο όνομα του αντικειμένου στο οποίο ανήκει το script και από το μενού που προκύπτει επιλέγουμε τη γραμμή 'button to fire this script'. Στη συνέχεια επιλέγοντας το κόκκινο εργαλείο του κουμπιού που μόλις δημιουργήσαμε, επιλέγουμε τη γραμμή 'change label' και ανάλογα με το κλάσμα γράφουμε την τιμή. Το αποτέλεσμα της πράξης αυτής για όλα τα script φαίνεται στην Εικόνα 103.



**Εικόνα 103: Κουμπιά επιλογής των κλασμάτων**

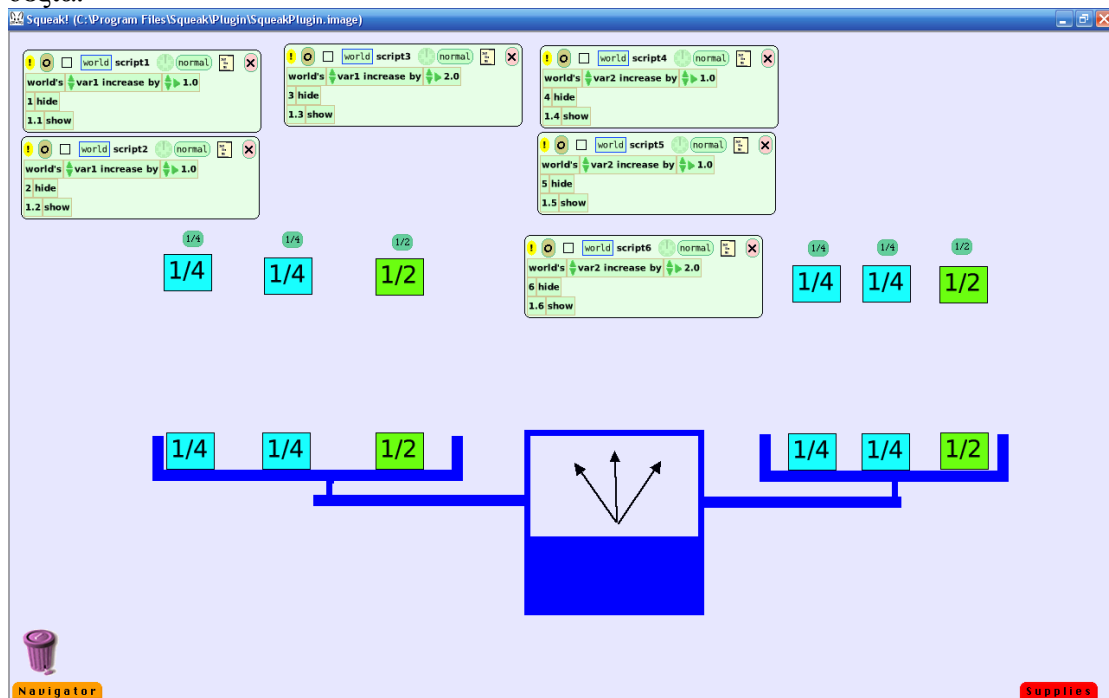
Το επόμενο βήμα είναι κατασκευή του βέλους της ζυγαριάς. Ουσιαστικά θα γίνει κατασκευή 3 βελών από τα οποία θα εμφανίζεται μόνο το κατάλληλο, ανάλογα με την αναμενόμενη κλίση της ζυγαριάς. Ανοίγουμε το εργαλείο 'object catalog' από τη καρτέλα 'Supplies' και τραβάμε τρία βέλη (Εικόνα 104).

## Σχολικά Παραδείγματα



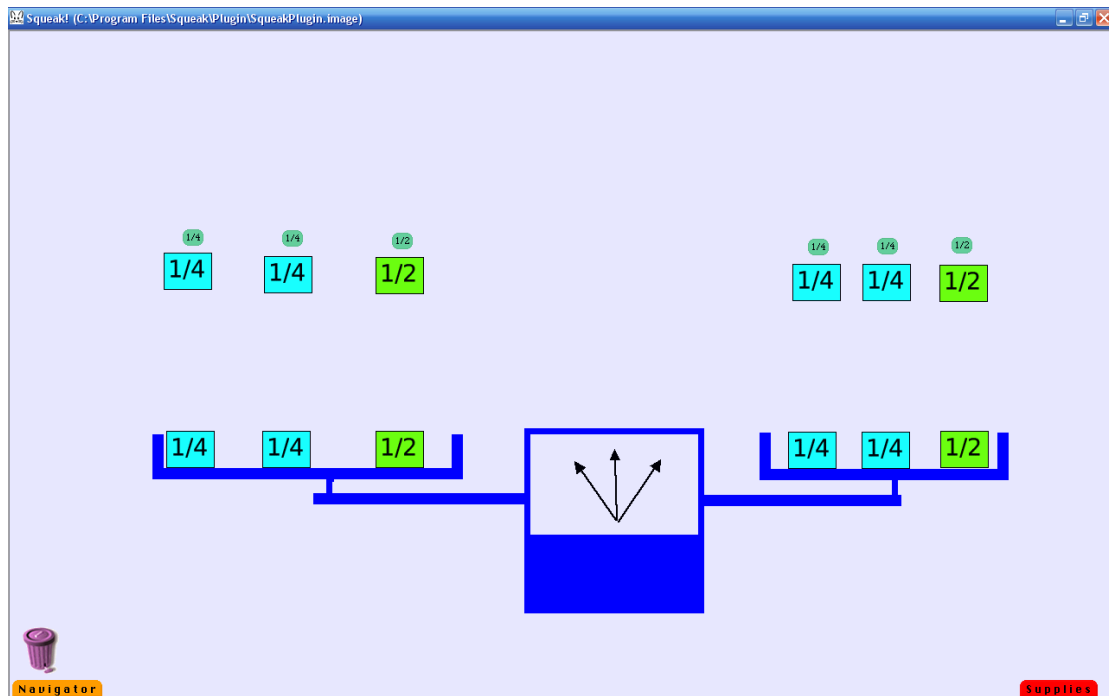
Εικόνα 104: Τρία αντικείμενα 'Arrow'

Στη συνέχεια τα περιστρέφουμε με το μπλε εργαλείο (rotate) (Εικόνα 105) και τους δίνουμε τα ονόματα 'left', 'middle' και 'right' αντίστοιχα από αριστερά προς τα δεξιά.



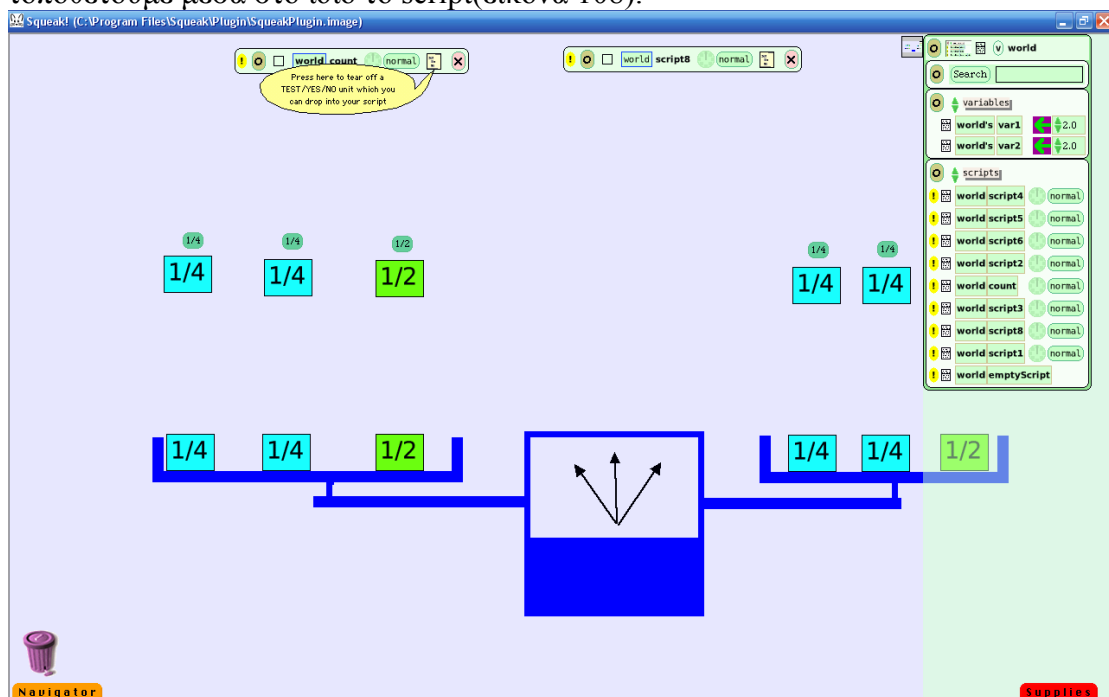
Εικόνα 105: Περιστροφή των βελών της ζυγαριάς

Στη συνέχεια πατάμε στα script που έχουμε δημιουργήσει, το κουμπί κύκλος που έχουν στο πάνω μέρος τους και παίρνουμε την Εικόνα 106.

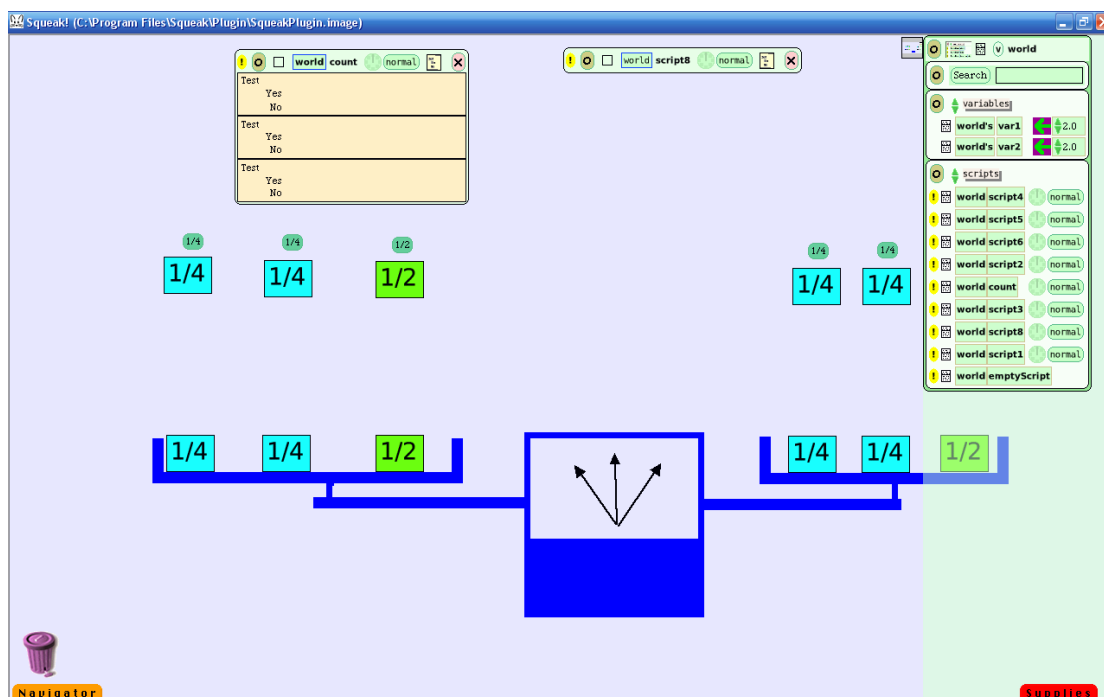


**Εικόνα 106: Εξαφάνιση των script από την οθόνη του τελικού χρήστη**

Ανοίγουμε τώρα το αντικείμενο 'world' και τραβάμε δύο 'world empty script'. Το πρώτο το ονομάζουμε 'count' και θα είναι υπεύθυνο για τις συγκρίσεις των κλασμάτων και την εμφάνιση του κατάλληλου βέλους. Στο script αυτό πηγαίνουμε στο πάνω δεξί μέρος του (εικόνα 107) και τραβάμε τρία αρχεία Test τα οποία τοποθετούμε μέσα στο ίδιο το script(εικόνα 108).

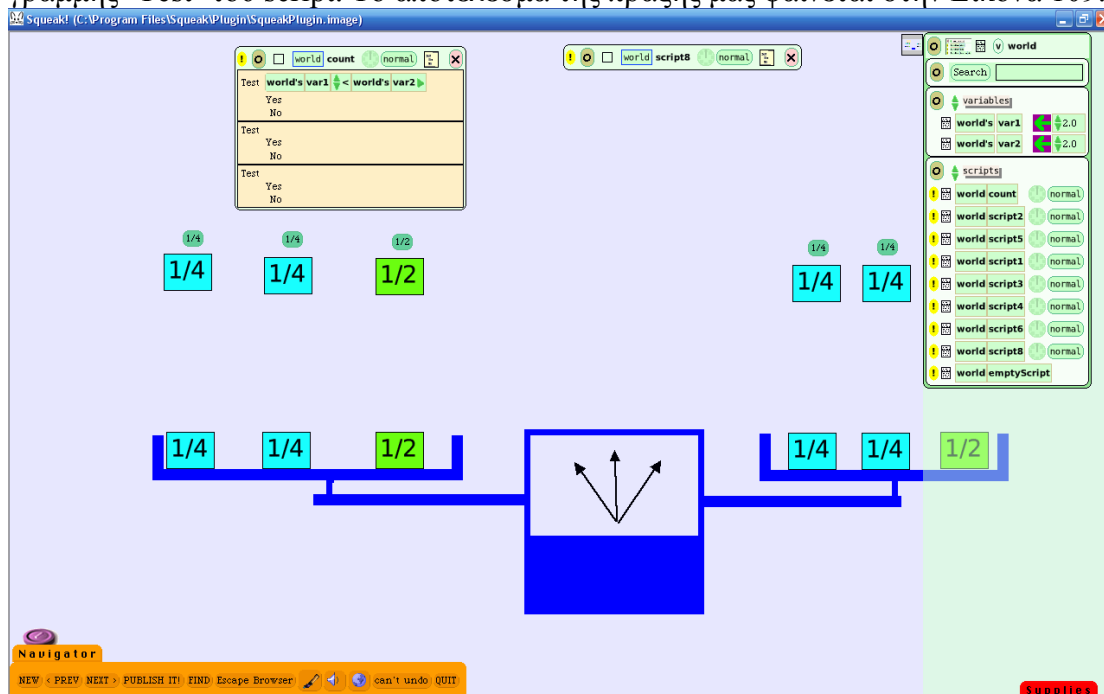


**εικόνα 107: δημιουργία 'Test' μέσα στο script**



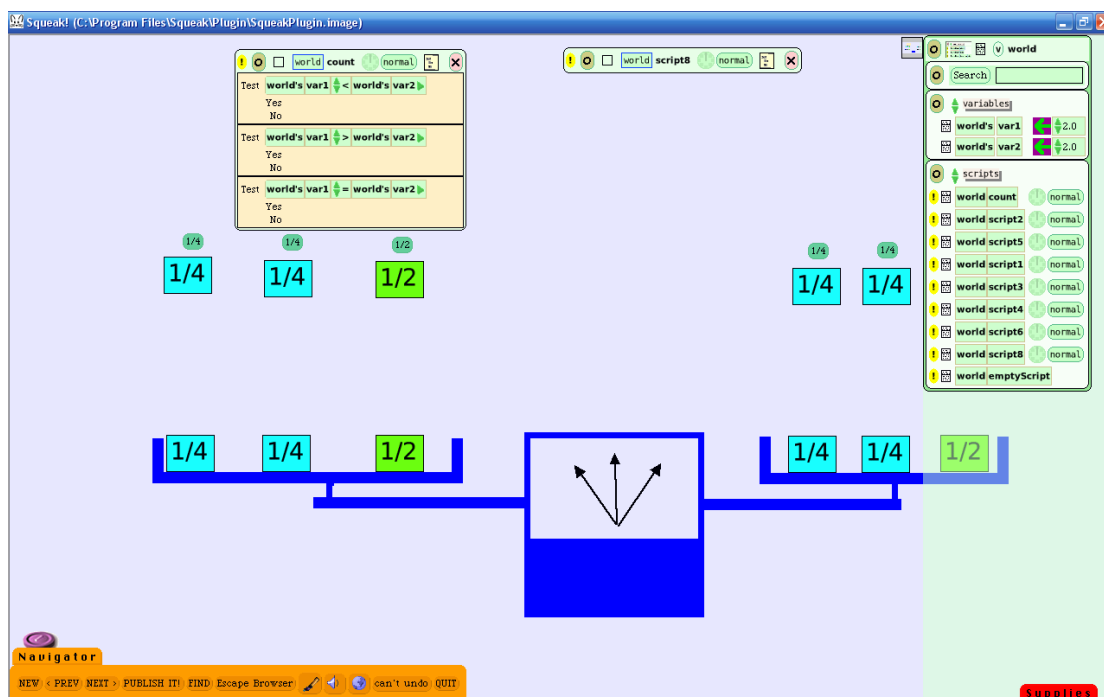
εικόνα 108: Δημιουργία τριών 'Test' στο script count

Και στα τρία 'Test' συγκρίνουμε τις μεταβλητές 'var1' και 'var2' και ανάλογα με το πια συνθήκη επαληθεύεται (>, <, =), τρέχει το αντίστοιχο 'Test'. Στην κατηγορία 'variables' του αντικειμένου 'world' σύρουμε τη γραμμή 'world's var1' και την αφήνουμε στο πρώτο 'Test' δεξιά από τη λέξη 'Test'. Στη συνέχεια, στην ίδια κατηγορία τραβάμε τη γραμμή 'world's var2' στο τελευταίο κουτί από δεξιά της γραμμής 'Test' του script. Το αποτέλεσμα της πράξης μας φαίνεται στην Εικόνα 109.



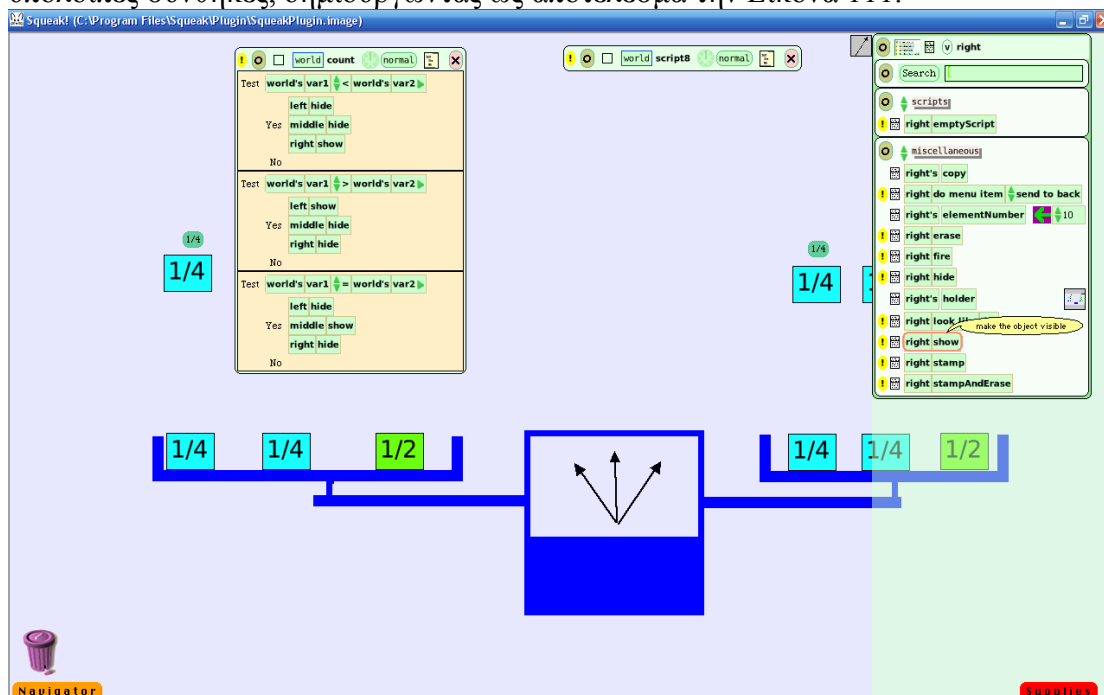
Εικόνα 109: Πρώτη σύγκριση για την κίνηση των βελών της ζυγαριάς

Με τον ίδιο τρόπο, στα άλλα δύο 'Test' δημιουργούμε και τις υπόλοιπες συγκρίσεις '>' και '=', όπως βλέπουμε στην Εικόνα 110.



**Εικόνα 110: Κατασκευή των υπόλοιπων συγκρίσεων του script 'count'**

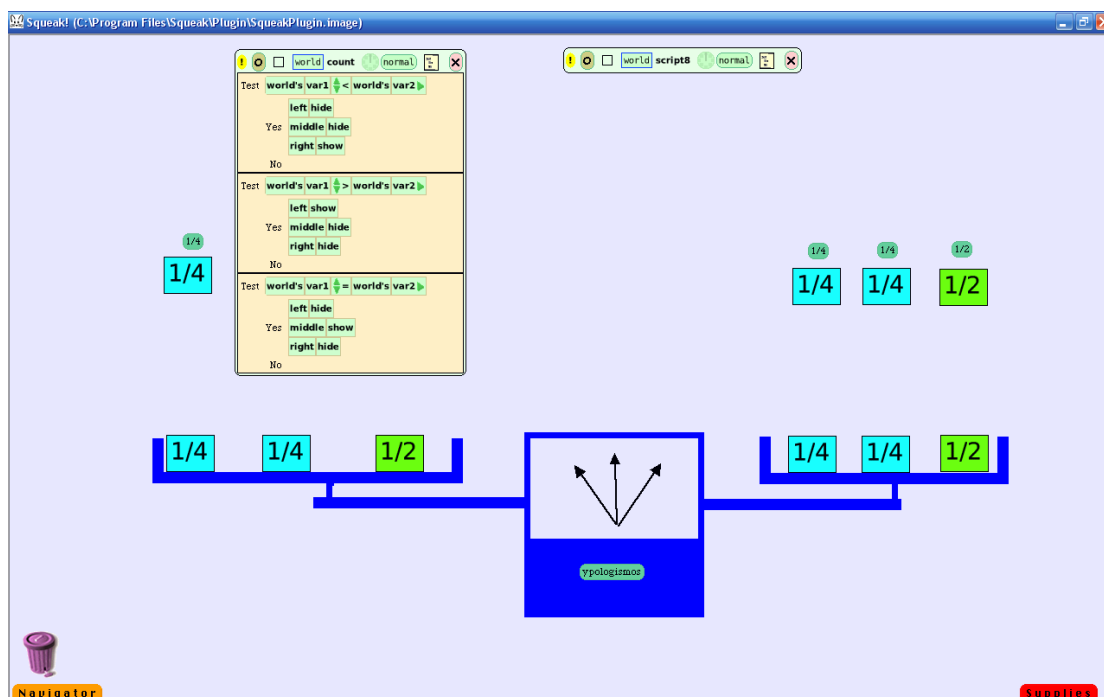
Ανάλογα με τα αν επαληθεύεται η συνθήκη των 'Test' ή όχι, θα εμφανίζεται το αντίστοιχο βέλος. Έτσι αν το 'var1' '<' 'var2', θα πρέπει να εμφανίζεται το δεξί βέλος και να μην εμφανίζονται τα άλλα δύο. Με τον ίδιο τρόπο συντάσσονται οι υπόλοιπες συνθήκες, δημιουργώντας ως αποτέλεσμα την Εικόνα 111.



**Εικόνα 111: Η τελική μορφή του script 'count'**

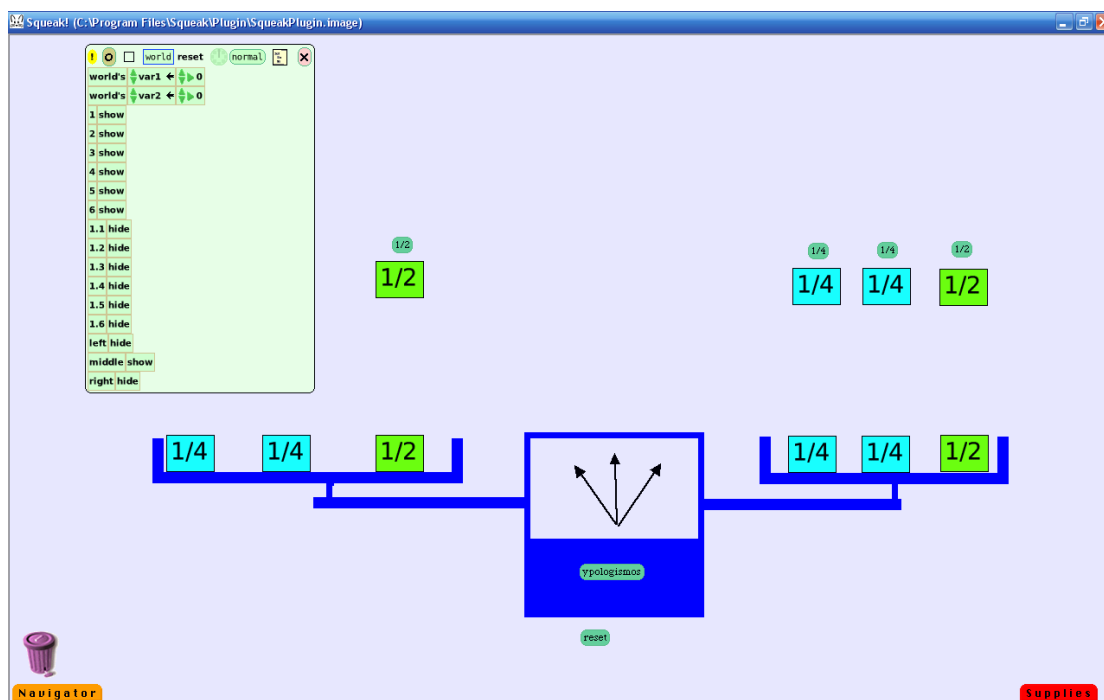
Για το script αυτό, δημιουργούμε ένα κουμπί εκτέλεσης με τον τρόπο που έχουμε ήδη παρουσιάσει, και αλλάζουμε το όνομα της ταμπέλας του σε 'γρολογισμός' (Εικόνα 112).





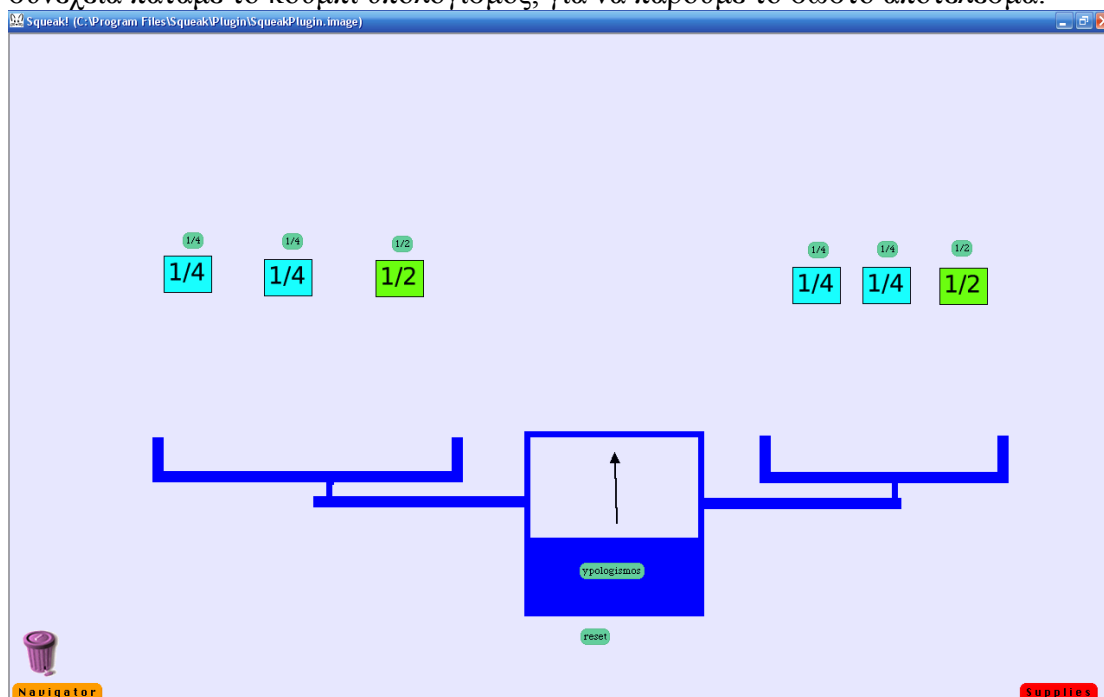
**Εικόνα 112: Κουμπί εκτέλεσης του script count**

Το τελικό στάδιο είναι η δημιουργία ενός script το οποίο θα ονομάζεται 'reset' και θα είναι υπεύθυνο για την επαναφορά του προγράμματος στην αρχική του κατάσταση. Η αρχική κατάσταση έχει ως εξής: δεν θα υπάρχουν τοποθετημένα αντικείμενα πάνω στη ζυγαριά όπως επίσης δεν θα φαίνονται το δεξί και το αριστερό βέλος. Αντίθετα εμφανή θα είναι τα αντικείμενα που βρίσκονται στην πάνω σειρά όπως επίσης και το μεσαίο βέλος. Τέλος θα πρέπει η μεταβλητές 'var1' και 'var2' να έχουν τη τιμή '0'. Αυτό έχει σαν αποτέλεσμα να τοποθετηθούν στο script εντολές show που βρίσκονται στην κατηγορία miscellaneous του κάθε αντικειμένου που πρέπει να είναι εμφανές και αντίστοιχα εντολές hide, που βρίσκονται στην ίδια κατηγορία για αυτά που δεν πρέπει να φαίνονται. Όσον αφορά το μηδενισμό των μεταβλητών, στην καρτέλα 'variables' του αντικειμένου 'world' θα σύρουμε και τις δύο γραμμές από τα βέλη που δείχνουν προς τα αριστερά, μέσα στο script. Επίσης φτιάχνουμε ένα κουμπί εκτέλεσης του script που το ονομάζουμε 'reset' Οπότε το τελικό αποτέλεσμα φαίνεται στην Εικόνα 113.



Εικόνα 113: Δημιουργία script 'reset'

Τέλος, να αναφέρουμε τον τρόπο εκτέλεσης του προγράμματος. Αρχικά πατάμε το 'reset' για να αρχικοποιηθεί η κατάσταση (Εικόνα 114). Στη συνέχεια διαλέγουμε οποιοδήποτε κλάσμα θέλουμε πατώντας τα αντίστοιχα κουμπιά που βρίσκονται πάνω από τα κλάσματα, αφήνουμε το μαθητή να υποθέσει τη κλίση της ζυγαριάς και στη συνέχεια πατάμε το κουμπί υπολογισμός, για να πάρουμε το σωστό αποτέλεσμα.



Εικόνα 114: Αρχική κατάσταση του προγράμματος ζυγαριάς

### 3.4. Αριθμητικό σταυρόλεξο

Σκοπός της συγκεκριμένης εργασίας είναι η κατασκευή ενός αριθμητικού σταυρόλεξου. Το αριθμητικό σταυρόλεξο περιέχει αριθμούς και πράξεις ενώ υπάρχουν κενά που πρέπει να συμπληρωθούν. Προκειμένου να φτιάξουμε το περιβάλλον εργασίας, χρησιμοποιούμε τη ζωγραφική για να φτιάξουμε το

## Σχολικά Παραδείγματα

σταυρόλεξο. Αρχικά φτιάχνουμε ένα τετράγωνο και γραμμές μέσα του για να κατασκευάσουμε όλα τα τετράγωνα. Τέλος τα γεμίζουμε με χρώμα τα μαύρα τετράγωνα που πρέπει να μείνουν κενά. Αν συνεχίσει με τη βοήθεια Text αντικειμένων τοποθετούμε όλα τα στοιχεία μέσα στο σταυρόλεξο καθώς και γράμματα και αριθμούς οριζόντια και κάθετα για να ξεχωρίζουμε τα στοιχεία μεταξύ τους. Τέλος, προσθέτουμε όλες τις αριθμητικές πράξεις που θέλουμε να υπάρχουν μέσα στο σταυρόλεξο και όλους τους αριθμούς που αφορούν τις πράξεις.

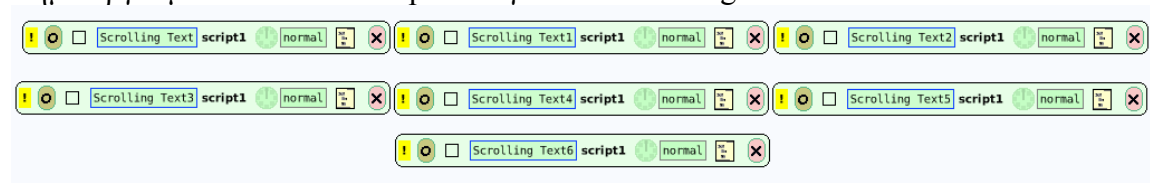
	A	B	C	D	E	F	G
1	2	.		+	5	=	11
2	.		.		.		.
3		.		+		=	22
4	+		+		+		+
5		.	2	+	4	=	
6	=		=		=		=
7	13	.	17	+	39	=	

Εικόνα 115: το αριθμητικό σταυρόλεξο

Αυτή τη στιγμή έχουμε κατασκευάσει ένα σταυρόλεξο το οποίο περιέχει 7 κενά στοιχεία. Πιο συγκεκριμένα πρόκειται για τα A3, A5, C1, C3, E3, G5 και G7. Αυτά τα κελιά είναι αντικείμενα Text (border). Στη διαδικασία του προγράμματος ο χρήστης καλείται να συμπληρώσει τα κενά και το πρόγραμμα θα πρέπει να απαντά στο χρήστη για κάθε πράξη αν δίνεται η σωστή απάντηση ή όχι. Τέλος θα πρέπει να υπάρχουν επιπλέον πλήκτρα τα οποία θα δίνουν βοήθεια στο χρήστη και πιο συγκεκριμένα θα συμπληρώνουν απαντήσει σε κελιά αν υπάρχει δυσκολία επίλυσης. Για το λόγο αυτό, κατασκευάζουμε 7 scrolling texts και 7 κουμπιά. Στα scrolling texts θα φαίνεται η απάντηση στα όσα γράφει ο χρήστης μέσα στα κενά ενώ με τα κουμπιά θα συμπληρώνεται αυτόματα η σωστή απάντηση.

Δεδομένου ότι ήδη γνωρίζουμε τις λύσεις στο συγκεκριμένο πρόβλημα, είναι πολύ απλό να κάνουμε το εξής. Να δημιουργήσουμε ένα script για κάθε scrolling text το οποίο να ελέγχει τι απάντηση έχει δοθεί στο αντίστοιχο κελί και να εμφανίζουμε ένα αποτέλεσμα. Αντίστοιχα για τα κουμπιά αυτό που πρέπει να κάνουμε είναι μόλις πατούμε ένα κουμπί τότε να καταγράφεται η απάντηση μέσα στο σωστό κελί.

Δημιουργούμε λοιπόν τα 7 scripts ένα για κάθε scrolling text.

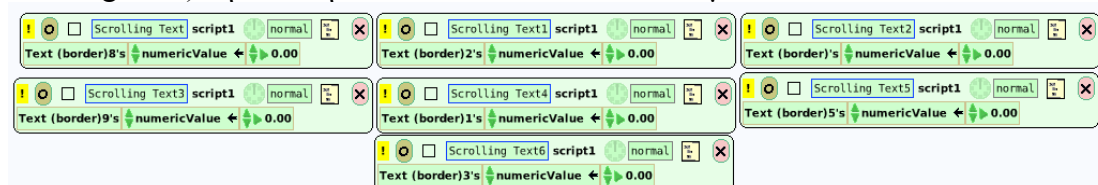


Εικόνα 116: όλα τα script για τα scrolling texts

Καθένα από αυτά τα scripts συνδέεται άμεσα με την τιμή που έχουν τα κενά κελιά κειμένου που βρίσκονται μέσα στο σταυρόλεξο. Άρα σε κάθε script πρέπει να

## Σχολικά Παραδείγματα

προσθέσουμε την τιμή του αντίστοιχου αντικειμένου. Συνεπώς ανοίγουμε τα στοιχεία όλων των αντικειμένων Text (border) και σύρουμε μέσα στο σωστό script (των scrolling texts) την ιδιότητα Numericvalue των αντικειμένου.



Εικόνα 117: όλα τα scripts των scrolling texts μαζί με την αντίστοιχη τιμή των κενών κελιών

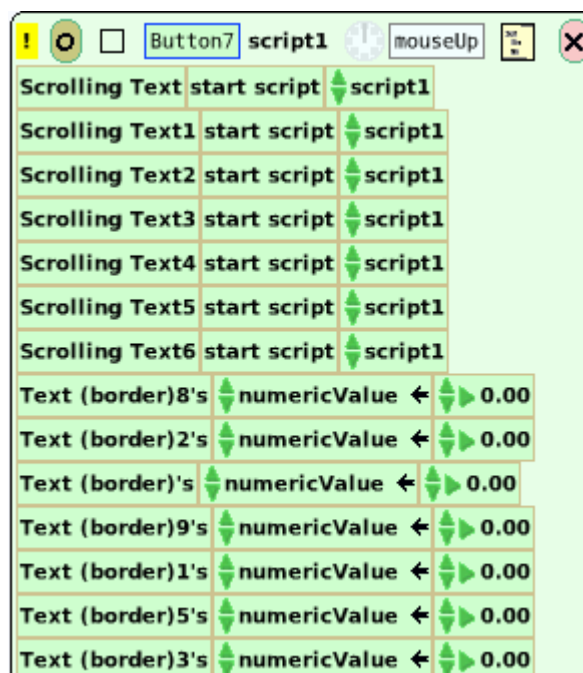
Αυτό που θέλουμε να κάνουμε είναι το εξής: να κάνουμε έναν έλεγχο μέσα στο script για την τιμή που έχει το αντίστοιχο κενό κελί. Αν η τιμή είναι η σωστή να εμφανίζουμε μέσα στο scrolling text ότι έχουμε σωστή απάντηση, ειδικά να εμφανίζουμε ότι έχουμε λάθος απάντηση.

Για το λόγο αυτό δημιουργούμε τον παρακάτω κώδικα σε κάθε Scrolling Text.

```
script1
  (Textborder8 getNumericValue = 3)
  ifTrue: [
    self setCharacters: 'SWSTO'.
  ]
  ifFalse: [
    self setCharacters: 'LATHOS'.
  ]
```

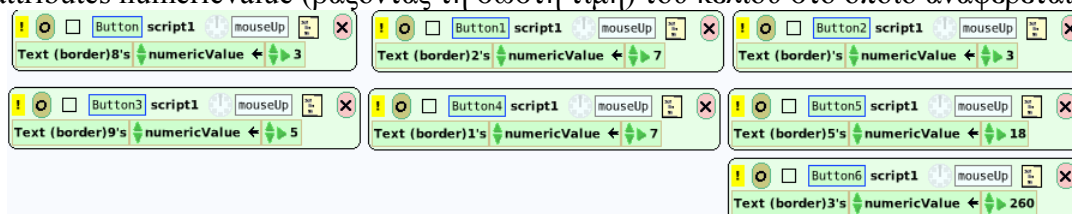
Αυτό το κομμάτι κώδικα μας ενημερώνει για το εξής. Αν η αριθμητική τιμή του Textborder8 το οποίο στην περίπτωση μας αντιστοιχεί στο κελί A3 είναι ίση με 3 τότε γράφουμε μέσα στο αντίστοιχο scrolling text τη λέξη ΣΩΣΤΟ, ειδικά γράφουμε τη λέξη ΛΑΘΟΣ. Αυτό θα πρέπει να γίνει για κάθε scrolling text και σε κάθε περίπτωση θα πρέπει να προσέξουμε να έχουμε ορίσει σωστά το Textborder στο οποίο αναφερόμαστε αλλά και την τιμή η οποία είναι η σωστή.

Προκειμένου να γίνεται συνεχώς ο έλεγχος για τη σωστή απάντηση και όχι μόνο μία φορά το script θα πρέπει να αλλάξει execution type από normal σε ticking. Για το λόγο αυτό θα δημιουργήσουμε ένα κουμπί το οποίο θα βάζει όλα τα script σε ticking. Δημιουργούμε ένα κουμπί και φτιάχνουμε γι' αυτό ένα καινούριο script. Ρίχνουμε μέσα σε αυτό για κάθε αντικείμενο scrolling text αλλά και για κάθε αντικείμενο Text (border) για τα μεν πρώτα από τα attributes scripting το startscript και για τα μεν Text (border) το setNumericValue = 0 όπως φαίνεται παρακάτω.



Εικόνα 118: Το script για το κουμπί εκκίνησης

Δεν ξεχνούμε πως επειδή πρόκειται για κουμπί θα πρέπει να ορίσουμε την εκτέλεση του script σε mouse up και όχι normal. Πατώντας αυτό το κουμπί το πρόγραμμά μας ξεκινά και είναι έτοιμο για εκτέλεση. Αυτό που μας έχει απομείνει είναι τα βοηθητικά κουμπιά που μας δίνουν απαντήσεις. Θα πρέπει να συσχετίσουμε κάθε κουμπί με το αντίστοιχο Text (border) το οποίο αντιπροσωπεύει και να δώσουμε τη σωστή τιμή. Έτσι φτιάχνουμε για κάθε κουμπί ένα script και μέσα στο script προσθέτουμε τα attributes numericvalue (βάζοντας τη σωστή τιμή) του κελιού στο οποίο αναφέρεται.



Εικόνα 119: Τα scripts των κουμπιών βοήθειας

Δεν ξεχνούμε για ακόμα μία φορά πως θα πρέπει το script από normal να αλλάξει σε mouseup για την εκτέλεσή του γιατί πρόκειται για κουμπιά.

Το πρόγραμμά μας είναι έτοιμο. Μπορούμε να κάνουμε δοκιμές αν λειτουργεί σωστά αλλά και βασισμένοι σε αυτό να δημιουργήσουμε πολλά αριθμητικά σταυρόλεξα με όσες διαφορετικές πράξεις θελήσουμε.

	A	B	C	D	E	F	G
1	2	.	0	+	5	=	11
2	.		.		.		.
3	0	.	0	+	0	=	22
4	+		+		+		+
5	0	.	2	+	4	=	0
6	=		=		=		=
7	13	.	17	+	39	=	0

**A3** LATHOS

**A5** LATHOS

**C1** LATHOS

**C3** LATHOS

**E3** LATHOS

**G5** LATHOS

**G7** LATHOS

LYSH A3

LYSH A5

LYSH C1

LYSH C3

LYSH E3

LYSH G5

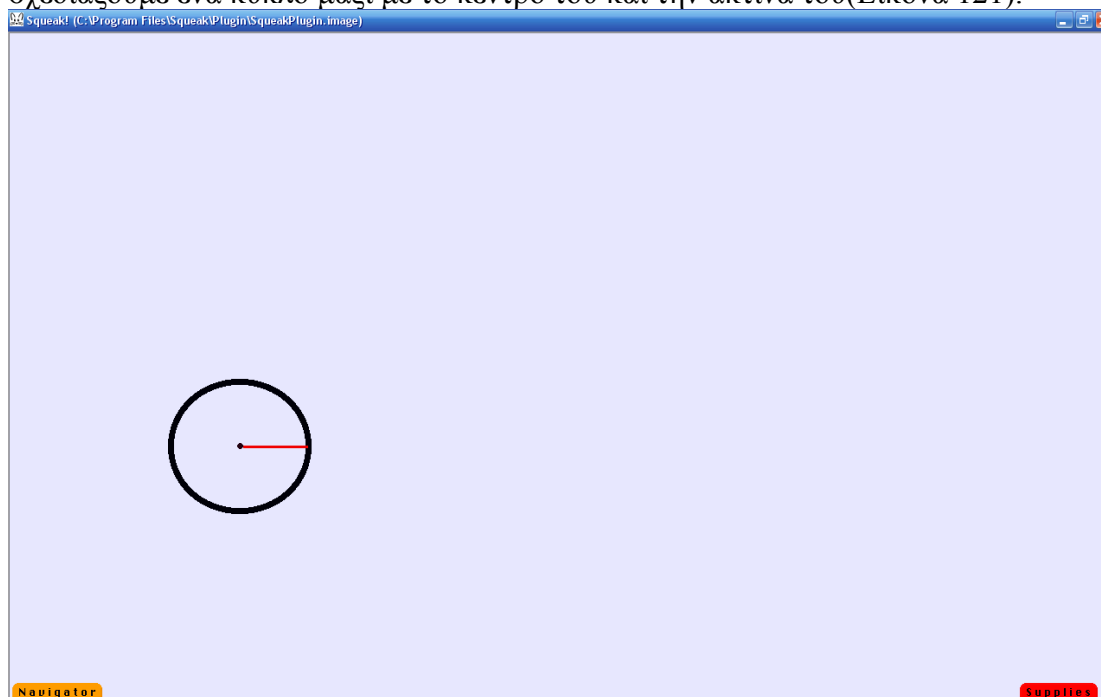
LYSH G7

RESET AND START

Εικόνα 120: Το σύστημά μας ολοκληρωμένο

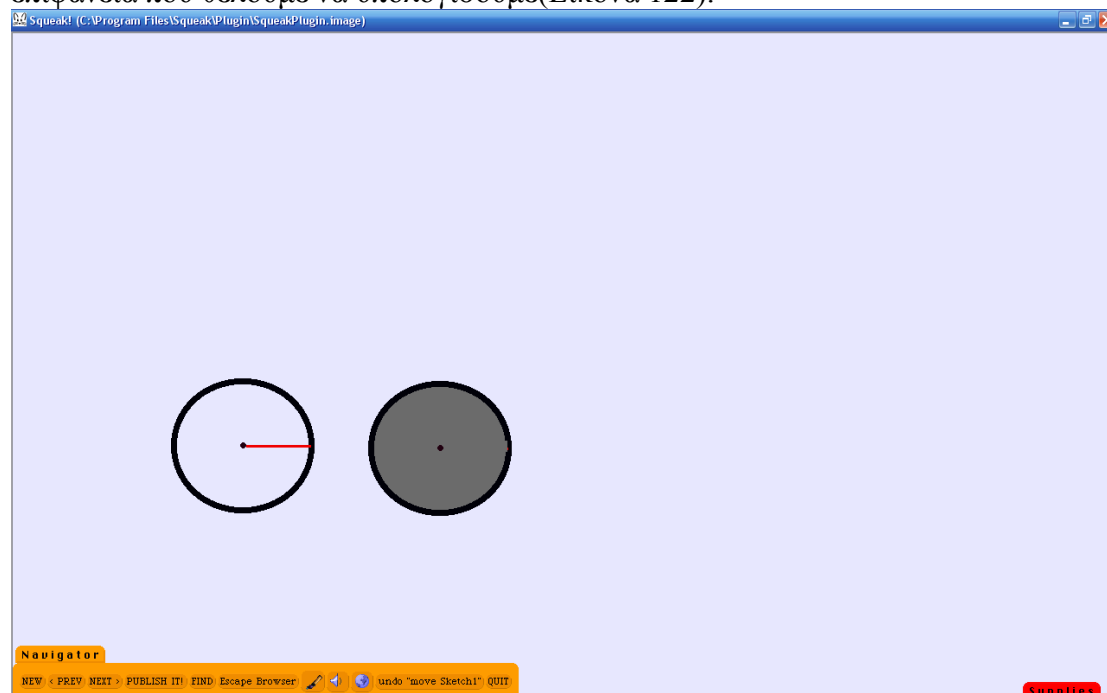
### 3.5. Εμβαδόν επίπεδων σχημάτων

Στο παράδειγμα αυτό θα εμφανίζουμε τον τρόπο υπολογισμού του εμβαδού των βασικών επίπεδων σχημάτων. Τα επίπεδα σχήματα που θα παρουσιάσουμε θα είναι ο κύκλος, το τετράγωνο, το τρίγωνο και το παραλληλόγραμμο. Τα παιδιά θα επιλέγουν το σχήμα της αρεσκείας τους και θα τους εμφανίζονται οι κατάλληλες πληροφορίες. Ξεκινάμε με τη δημιουργία των αντικειμένων του προγράμματος τα οποία θα παρουσιάζουν στα παιδιά τον τρόπο υπολογισμού του εμβαδού. Το πρώτο αντικείμενο αφορά τον κύκλο. Ανοίγουμε το εργαλείο της ζωγραφικής και σχεδιάζουμε ένα κύκλο μαζί με το κέντρο του και την ακτίνα του (Εικόνα 121).



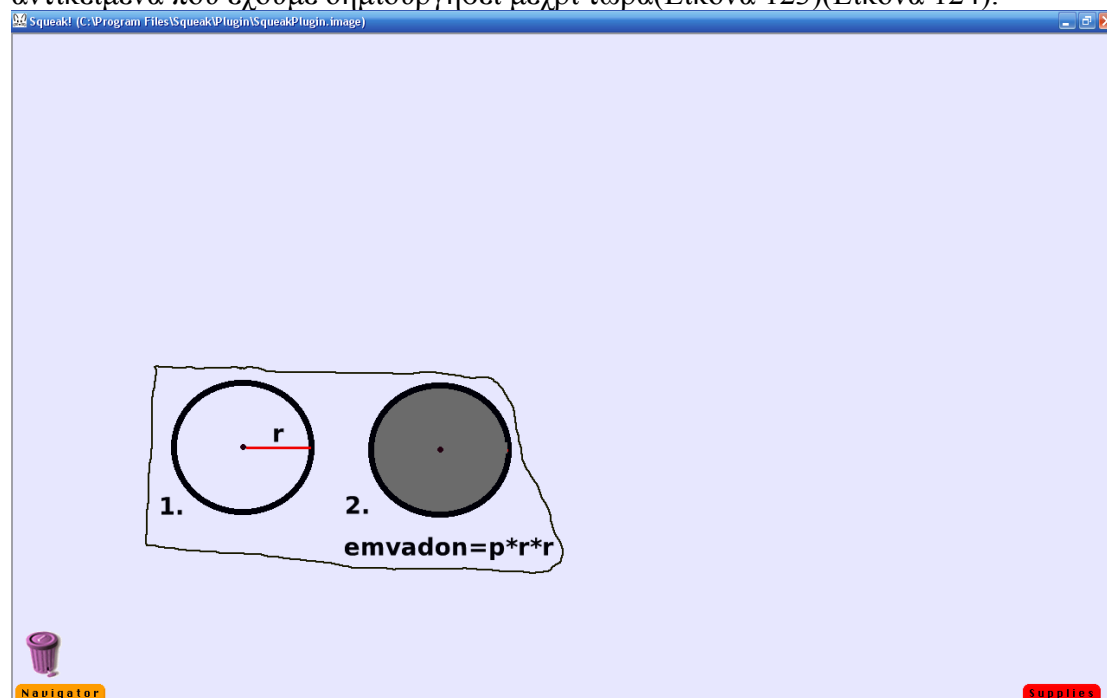
Εικόνα 121: κατασκευή κύκλου με την ακτίνα του

Στη συνέχεια κατασκευάζουμε έναν καινούργιο κύκλο στο μέγεθος του αρχικού του οποίο το εσωτερικό θα είναι γεμισμένο με διαφορετικό χρώμα για να μας δείξει την επιφάνεια που θέλουμε να υπολογίσουμε (Εικόνα 122).

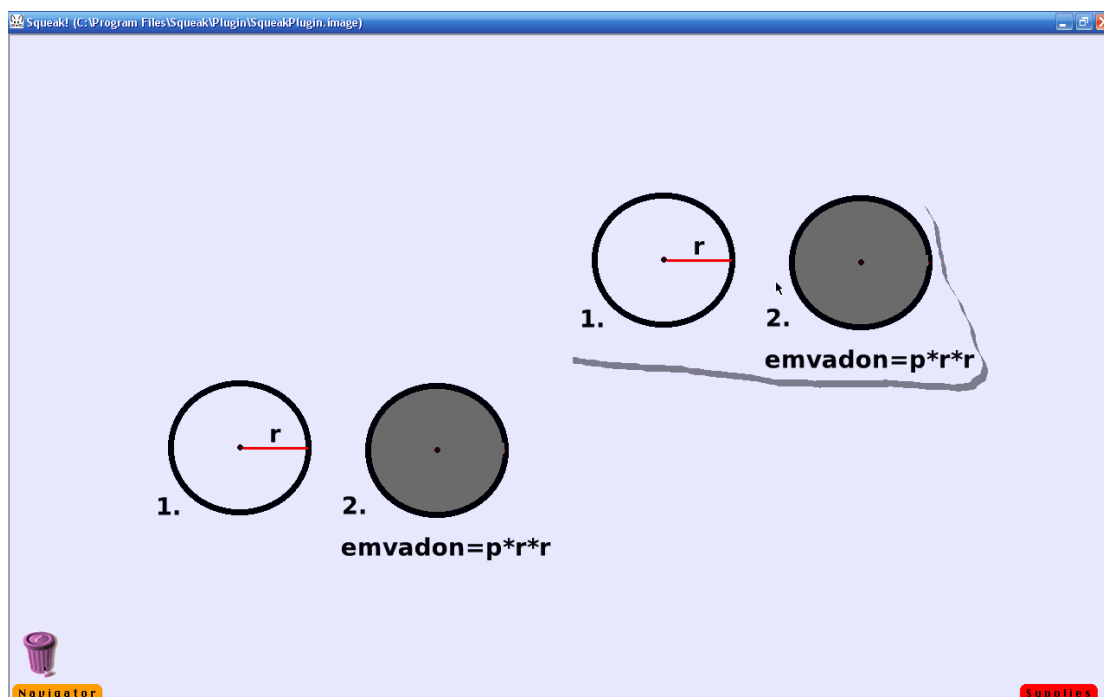


**Εικόνα 122: Εμβαδό που θέλουμε να υπολογίσουμε στον κύκλο**

Στη συνέχεια χρησιμοποιούμε αντικείμενα 'text' για να εισάγουμε κάποιες περαιτέρω πληροφορίες όσον αφορά την αλληλουχία των εικόνων καθώς και τον τύπο υπολογισμού του εμβαδού. Θα προσπαθήσουμε να αποφύγουμε τη σύγχυση που μπορεί να προκύψει από τη δημιουργία πολλών αντικειμένων. Για το λόγο αυτό, σύρουμε το εργαλείο 'Lasso' από την καρτέλα 'Supplies' στη επιφάνεια του προγράμματος και αυτόματα ο δείκτης του ποντικιού παίρνει τη μορφή σταυρού. Με το 'Lasso' κόβουμε την επιθυμητή επιφάνεια έτσι ώστε να περικλείει όλα τα αντικείμενα που έχουμε δημιουργήσει μέχρι τώρα (Εικόνα 123) (Εικόνα 124).

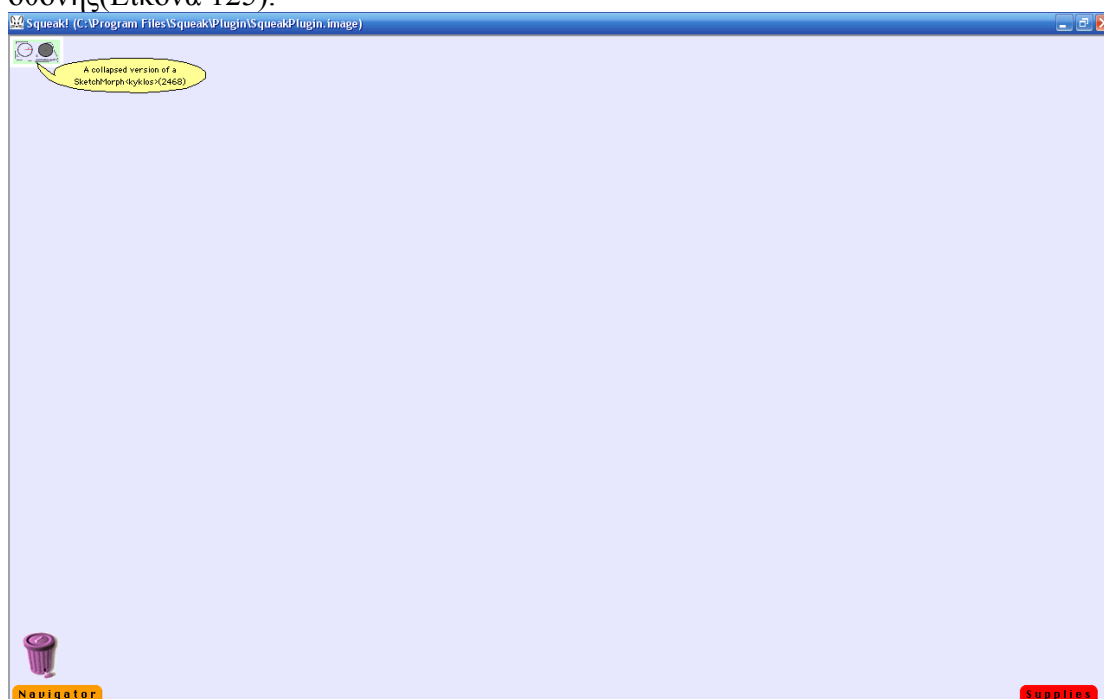


**Εικόνα 123: χρήση του εργαλείου lasso βήμα 1**



**Εικόνα 124: χρήση του εργαλείου lasso βήμα 2**

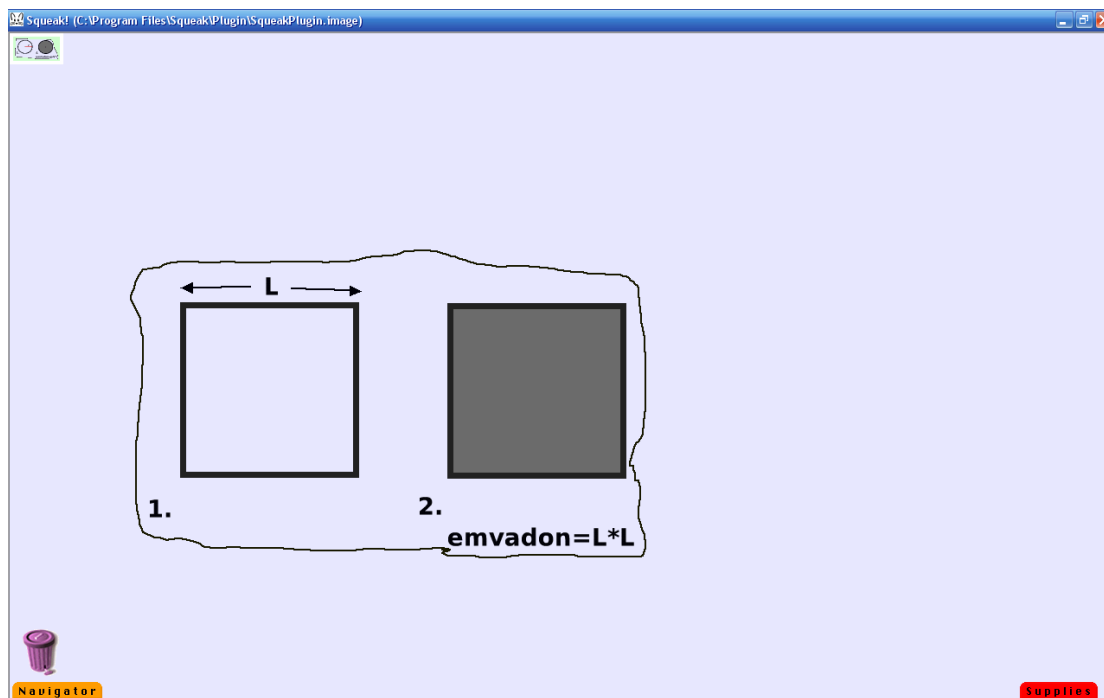
Αυτό που προκύπτει είναι ουσιαστικά μια ομαδοποίηση όλων των αντικειμένων σε ένα καινούργιο το οποίο το ονομάζουμε 'kyklos'. Οπότε τώρα μπορούμε να σβήσουμε τα αρχικά αντικείμενα. Για να μην μας εμποδίζει αυτό λόγω του μεγέθους του, με το εργαλείο κύκλος, το ελαχιστοποιούμε στο πάνω αριστερό μέρος της οθόνης(Εικόνα 125).



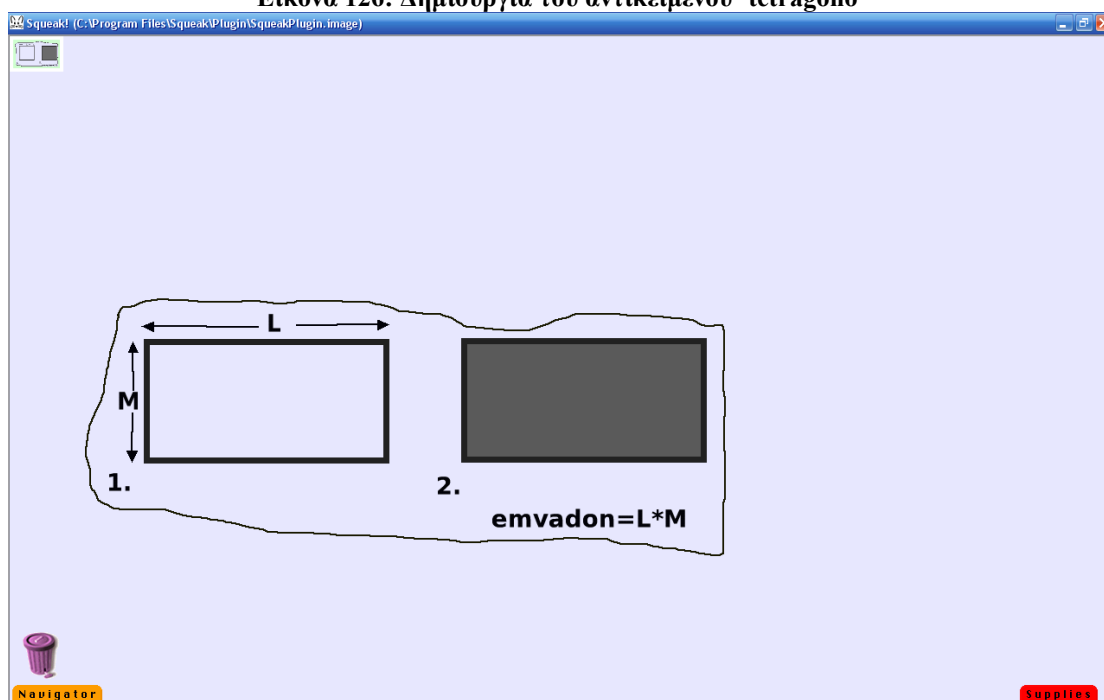
**Εικόνα 125: Ελαχιστοποίηση του αντικειμένου 'kyklos'**

Με τον ίδιο τρόπο κατασκευάζουμε και τα αντικείμενα των άλλων σχημάτων όπως φαίνεται και στις παρακάτω εικόνες: Εικόνα 126, Εικόνα 127, Εικόνα 128, Εικόνα 129.

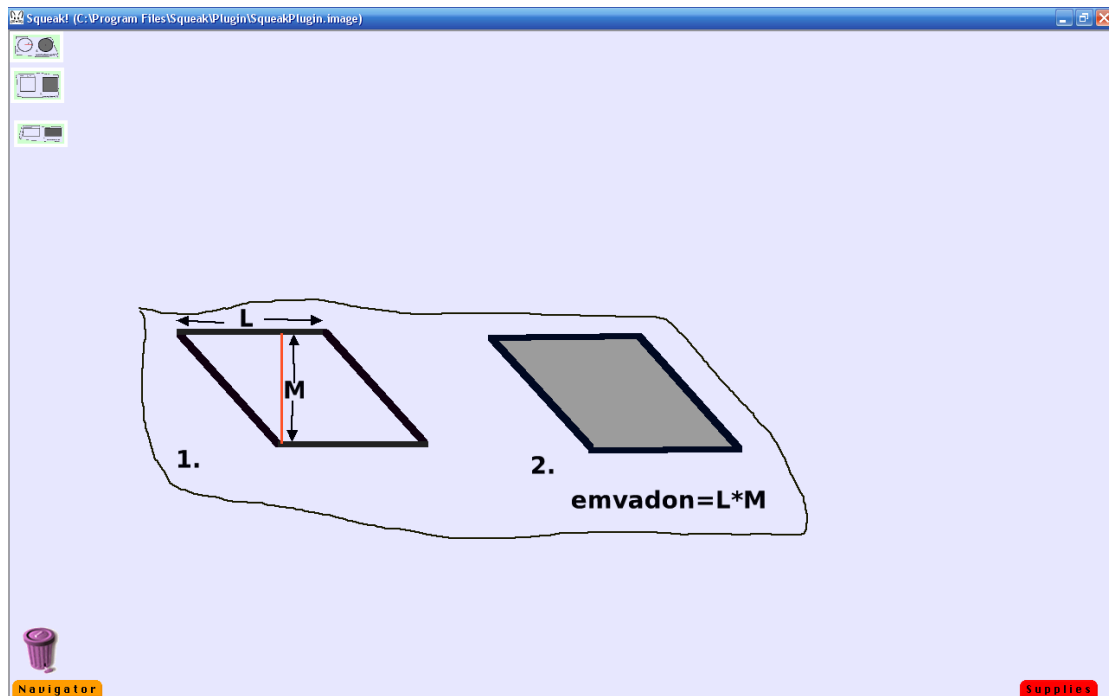




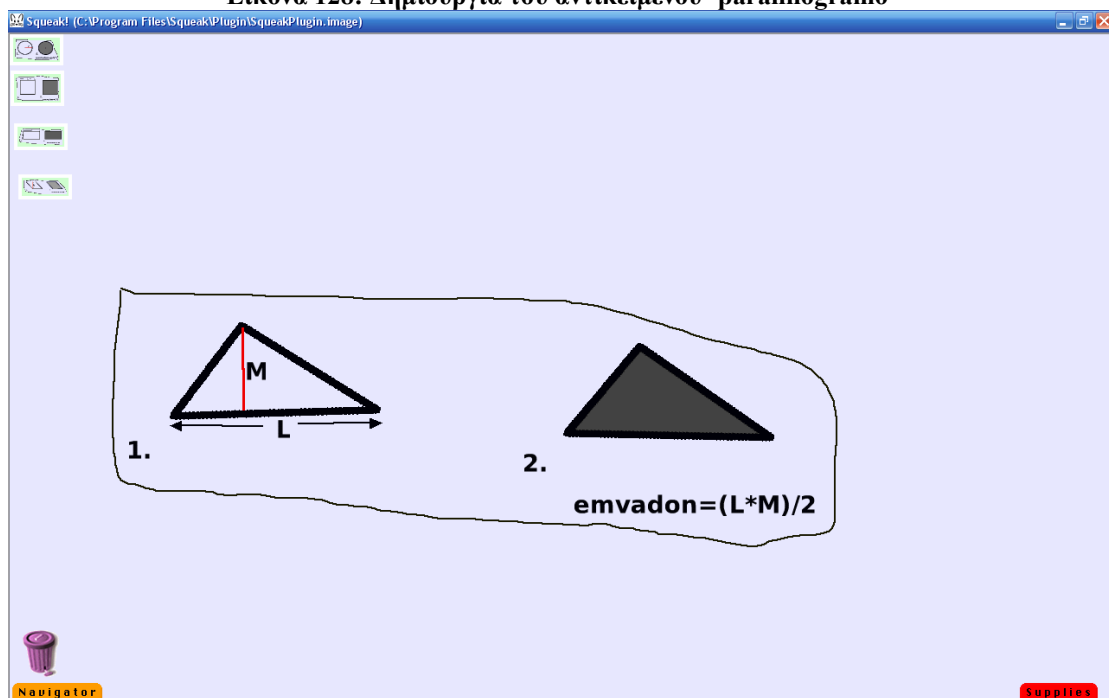
Εικόνα 126: Δημιουργία του αντικειμένου 'tetragono'



Εικόνα 127: Δημιουργία του αντικειμένου 'orthogonio'

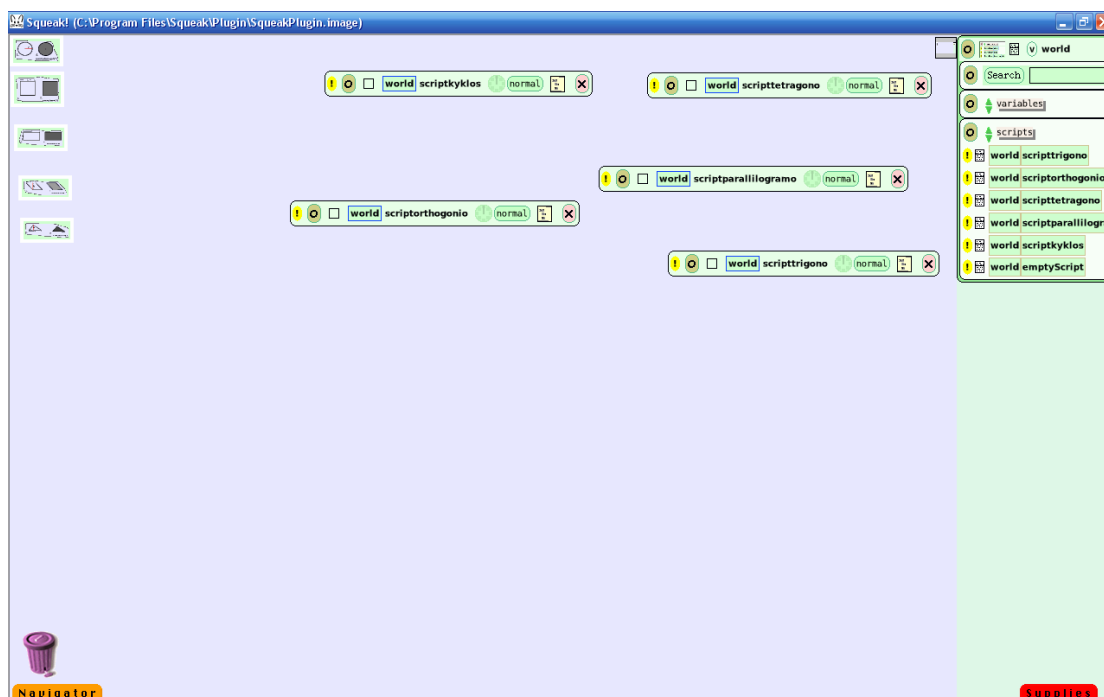


Εικόνα 128: Δημιουργία του αντικειμένου 'parallilogramo'



Εικόνα 129: Δημιουργία του αντικειμένου 'trigono'

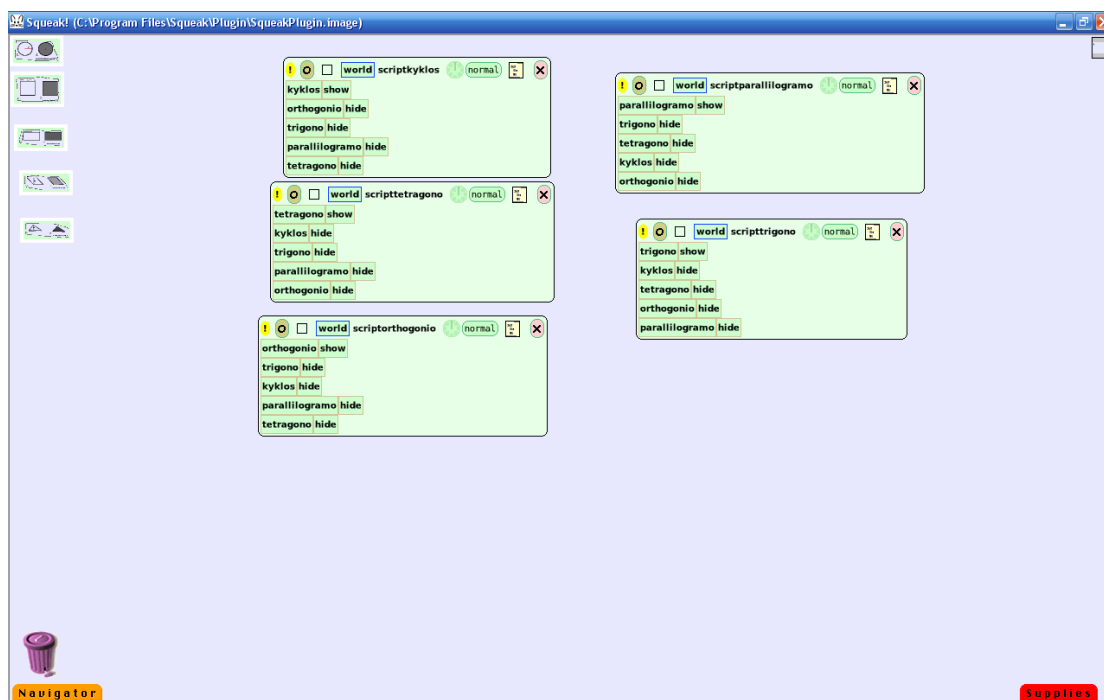
Στη συνέχεια θα δημιουργήσουμε τα script με τα οποία θα εμφανίσουμε τα αντικείμενα που μόλις δημιουργήσαμε. Ανοίγουμε το εργαλείο μάτι του αντικειμένου 'world' και σύρουμε πέντε 'worldemptyscript' τα οποία τα ονομάζουμε 'script\*\*\*' όπου στα αστεράκια συμπληρώνουμε το όνομα του αντικειμένου που θα εμφανίζεται (Εικόνα 130).



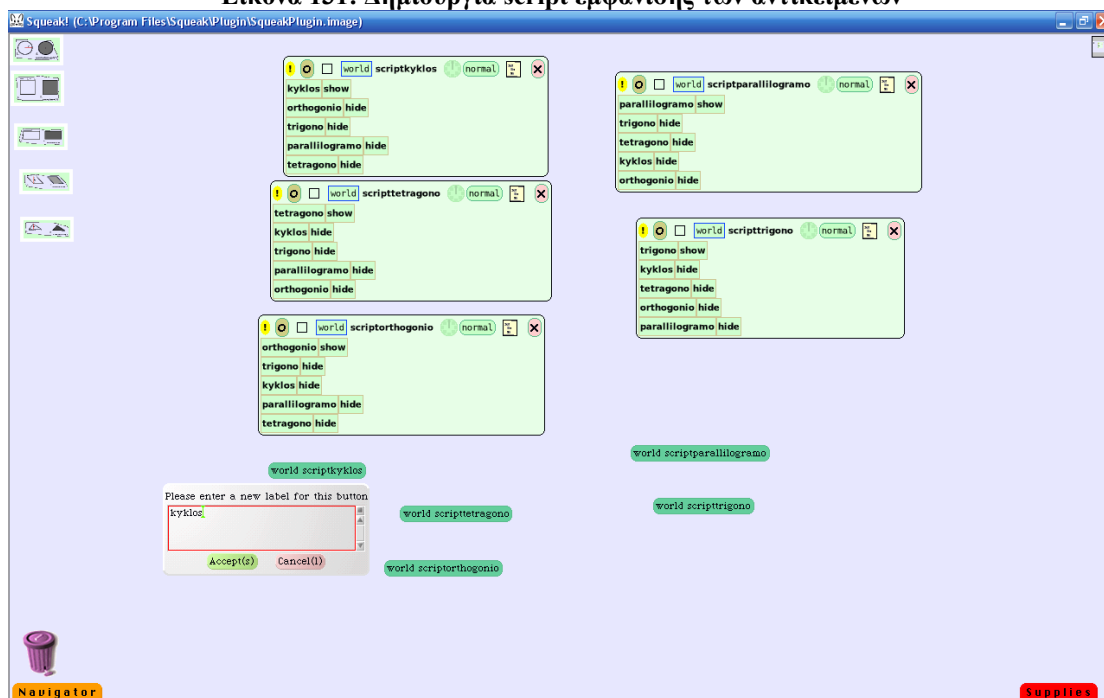
**Εικόνα 130: ονομασία script εμφάνισης αντικειμένων**

Το επόμενο στάδιο είναι ο εμπλουτισμός των script με εντολές 'show' και 'hide' οι οποίες βρίσκονται στην καρτέλα 'miscellaneous' του κάθε αντικειμένου. Για το λόγο αυτό εάν θέλουμε να εμφανίσουμε τον υπολογισμό του εμβαδού του τετραγώνου, επαναφέρουμε από την ελαχιστοποιημένη του μορφή το αντικείμενο, με απλό κλικ πάνω σε αυτό, και στη συνέχεια με το εργαλείο μάτι και πηγαίνοντας στη κατηγορία 'miscellaneous' τραβάμε τη γραμμή 'tetragono show' και την τοποθετούμε μέσα στο script 'scripttetragono'. Στη συνέχεια ανοίγουμε και το αντικείμενο 'kyklos' και στην κατηγορία 'miscellaneous' τραβάμε τη γραμμή 'kyklos hide' και την τοποθετούμε μέσα στο script 'scripttetragono'. Οι αντίστοιχες εντολές 'hide' τοποθετούνται και για τα υπόλοιπα τρία αντικείμενα ('orthogonio', 'trigono', 'parallilogramo') στο scripttetragono. Με τον ίδιο τρόπο συμπληρώνουμε και τα υπόλοιπα script ('scriptkyklos', 'scriptorthogonio', 'scriptparallilogramo', 'scripttrigono'). Αυτό σημαίνει ότι τοποθετούμε την εντολή show στο αντικείμενο που θέλουμε να εμφανίζεται και εντολές hide για τα υπόλοιπα αντικείμενα. Κατόπιν για κάθε ένα script δημιουργούμε ένα κουμπί εκτέλεσης. Το αποτέλεσμα φαίνεται στις Εικόνα 131, Εικόνα 132.

## Σχολικά Παραδείγματα



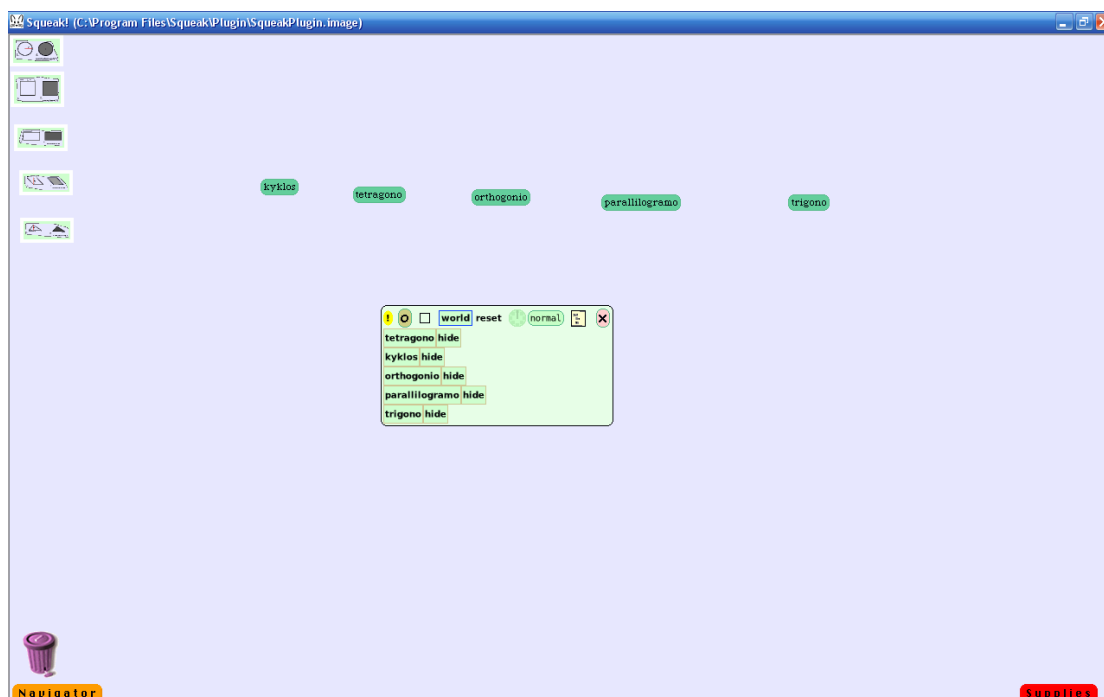
Εικόνα 131: Δημιουργία script εμφάνισης των αντικειμένων



Εικόνα 132: Δημιουργία κουμπιών εκτέλεσης των script

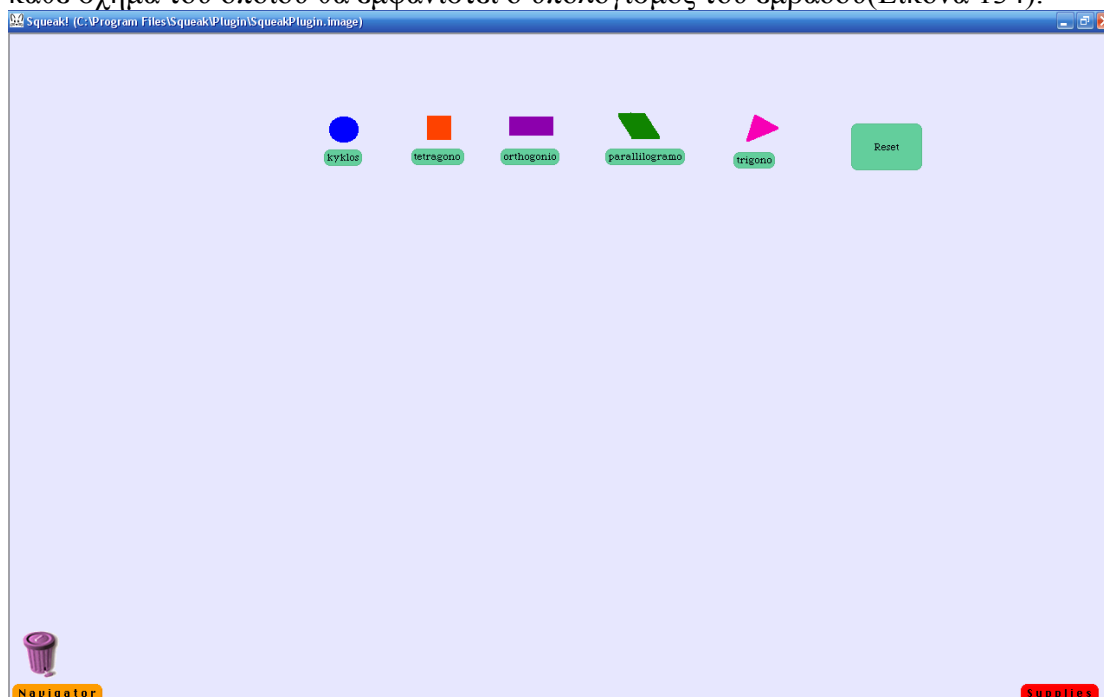
Θα ήταν χρήσιμο ακόμα ένα script το οποίο θα το ονομάσουμε reset και το οποίο μετά την εκτέλεση του προγράμματος θα επαναφέρει το πρόγραμμα στην αρχική του κατάσταση. Για να το επιτύχουμε αυτό, ανοίγουμε το εργαλείο μάτι του αντικειμένου 'world', τραβάμε ένα 'world emptyscript' και το ονομάζουμε reset. Στο script βάζουμε πέντε εντολές 'hide', η κάθε μία από τις οποίες προέρχεται από την κατηγορία miscellaneous του κάθε αντικειμένου που αντιπροσωπεύει τα σχήματα(Εικόνα 133), στο οποίο προσθέτουμε και κουμπί εκτέλεσης.

## Σχολικά Παραδείγματα



Εικόνα 133: script reset

Τέλος για οπτικούς λόγους σχεδιάζουμε πέντε μικρά σχήματα τα οποία θα τοποθετηθούν πάνω από τα κουμπιά εκτέλεσης των script και θα αντιπροσωπεύουν το κάθε σχήμα του οποίου θα εμφανιστεί ο υπολογισμός του εμβαδού(Εικόνα 134).



Εικόνα 134: Τελική μορφή του προγράμματος

Η λειτουργία του προγράμματος έχει ως εξής: τα παιδιά μπορούν να κάνουν κλικ στα κουμπιά που βρίσκονται κάτω από τα σχήματα και να εμφανίζεται ο τρόπος υπολογισμού του συγκεκριμένου σχήματος αντίστοιχα. Τέλος για να επανέρθει το πρόγραμμα στην αρχική του κατάσταση, πατάμε το κουμπί 'reset'.

## 4. 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.mucow.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>