



**ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ**  
UNIVERSITY OF PATRAS

**ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΔΙΟΙΚΗΣΗΣ  
ΕΠΙΧΕΙΡΗΣΕΩΝ  
ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΘΕΜΑ: ΔΗΜΙΟΥΡΓΙΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ  
ΜΕ ΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΕΡΓΑΛΕΙΟ GAME MAKER STUDIO**

**Βλάχος Ιωάννης**

Επιβλέπων καθηγητής

Καρούσος Διονύσιος

**ΜΕΣΟΛΟΓΓΙ 2019**

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Διοίκησης Επιχειρήσεων/Μεσολογγίου του ΤΕΙ Δυτικής Ελλάδας δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία δημιουργήθηκε με σκοπό την κατανόηση μερικών βασικών λειτουργιών του προγράμματος Game Maker Studio της Εταιρίας YoYo Game και έχει ως θέμα την δημιουργία παιχνιδιών. Στόχος τη πτυχιακής αυτής είναι να δώσει μια ομαλή εισαγωγή σε αυτούς που επιθυμούν να κάνουν τα πρώτα τους βήματα στο χώρο του gaming development .

Όπως αναφέραμε και παραπάνω η πτυχιακή αυτή επικεντρώνετε γύρο από το πρόγραμμα Game Maker Studio , ένα πρόγραμμα με μεγάλη ιστορία και πολλές δυνατότητες . θα επικεντρωθούμε κυρίως στις δυνατότητες του προγράμματος για Development (2D) δισδιάστατα ηλεκτρονικά παιχνίδια με μια μικρή αναφορά στο χώρο του (3D) τρισδιάστατα ηλεκτρονικά παιχνίδια. Θα αναλύσουμε τα εργαλεία που μας παρέχονται και διαθέτει το πρόγραμμα και της δυνατότητές τους. Ακόμη θα ερευνήσουμε και θα κατανοήσουμε ως έναν βαθμό την προγραμματιστική γλώσσα που χρησιμοποιεί το Game Maker Studio δηλαδή την GM-Language. Θα δούμε το συντακτικό της, συναρτήσεις, μεταβλητές και της βιβλιοθήκες της.

Αφού έχουμε κατανοήσει ως έναν βαθμό τα παραπάνω θα τελειώσουμε την πτυχιακή με την δημιουργία ενός ηλεκτρονικού παιχνιδιού (DEMO), που θα το ονομάσουμε " CATCH THE CLOWN " και θα αναλύσουμε κάποιες από της αυτοματοποιημένες λειτουργίες της μηχανής μας, που αυτό σημαίνει ότι θα δούμε πως μπορούμε να φτιάξουμε ένα ηλεκτρονικό παιχνίδι χωρίς να χρησιμοποιήσουμε καθόλου κώδικα κάνοντας το ιδανικό παράδειγμα για τον απλό χρήστη.

Ακόμη να αναφέρουμε πως παρά το γεγονός ότι Game Maker Studio είναι ένα πολύ αξιόλογο και πάρα πολύ διαδεδομένο πρόγραμμα, σίγουρα δεν είναι το μοναδικό. Εξύψου πολύ διαδεδομένο και αξιόλογο προγράμματα είναι το Unity που σε αντίθεση με το G.M.S. μπορεί ο χρήστης με αυτό να δημιουργήσει τόσο (3D) τρισδιάστατα παιχνίδια όσο και (2D) δυσδιάστατα. Είναι σημαντικό να αναφέρουμε ότι το Unity χρησιμοποιεί παραπάνω από μια

γλώσσες προγραμματισμού δίνοντάς του παραπάνω δυνατότητες και ελευθερίες, αλλά προσθέτει το αρνητικό της επιπλέον δυσκολίας κάνοντάς το δύσκολο σας πρόταση για αρχάριους στον χώρο.



## Περιεχόμενα

<b>ΠΕΡΙΛΗΨΗ</b> .....	2
<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ</b> .....	5
<b>ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ</b> .....	6
<b>ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ</b> .....	6
<b>ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ</b> .....	8
<b>ΕΓΧΕΙΡΙΔΙΟ ΕΚΜΑΘΗΣΗΣ ΤΟΥ GAME MAKER STUDIO</b> .....	8
<b>ΕΙΣΑΓΩΓΗ</b> .....	8
<b>2.1 ΟΙ ΟΝΤΟΤΗΤΕΣ ΤΟΥ GAME MAKER</b> .....	9
<b>2.1.1 SPRITES</b> .....	10
<b>2.1.2 SOUNDS</b> .....	15
<b>2.1.3 BACKGROUNDS</b> .....	17
<b>2.1.4 PATHS</b> .....	18
<b>2.1.5 SCRIPTS</b> .....	20
<b>2.1.6 FONTS</b> .....	21
<b>2.1.7 TIME LINES</b> .....	22
<b>2.1.8 OBJECTS</b> .....	24
<b>2.1.9 ROOMS</b> .....	25
<b>ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ</b> .....	27
<b>ΔΗΜΙΟΥΡΓΙΑ ΠΑΙΧΝΙΔΙΩΝ / EVENTS ΚΑΙ ACTIONS</b> .....	27
<b>3.1 EVENTS</b> .....	33
<b>3.1.2 ACTIONS</b> .....	43
<b>ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ</b> .....	122
<b>GML</b> .....	122
<b>4.1 LOOPS &amp; IF STATEMENT</b> .....	128
<b>4.2 ΜΕΤΑΒΛΗΤΕΣ</b> .....	132
<b>ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ</b> .....	134
<b>ΥΛΟΠΟΙΗΣΗ ΠΑΙΧΝΙΔΙΟΥ - CATCH THE CLOWN</b> .....	134
<b>ΕΠΙΛΟΓΟΣ</b> .....	144
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΕΣ ΠΗΓΕΣ</b> .....	145

## **ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ**

GMS: Game Maker Studio

GML: Game Maker Language

Web ή www: Παγκόσμιος Ιστός

# ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

## ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Είναι γνωστό, πως σε οτιδήποτε κι αν θέλουμε να κάνουμε, και κυρίως όταν έχει σχέση με τον προγραμματισμό και την υλοποίηση κάποιου έργου, το πρώτο πράγμα θα που πρέπει να κάνουμε, είναι ένας καλός, σωστός και αντάξιος των απαιτήσεών μας αρχικός σχεδιασμός. Αυτό ισχύει και όταν θέλουμε να δημιουργήσουμε ένα ηλεκτρονικό παιχνίδι, άσχετα με την πλατφόρμα που θα χρησιμοποιήσουμε. Υπάρχει μια σχετική «διαφωνία» για το ποιο πιστοποιείται ως το πρώτο ηλεκτρονικό (ψηφιακό) παιχνίδι που δημιουργήθηκε για τους υπολογιστές. Αυτό βέβαια μπορεί να απαντηθεί λαμβάνοντας υπόψη τον ορισμό που θα δώσει ο καθένας για τα «βιντεοπαιχνίδια». Αυτό το λέμε διότι παραδείγματος χάρη η λέξη «βίντεο» στη σύνθετη λέξη «βιντεοπαιχνίδι» αναφέρεται στη συσκευή απεικόνισης που χρησιμοποιείται για την χρήση του παιχνιδιού. Πλέον, με τη λέξη «βιντεοπαιχνίδι», λαμβάνουμε υπόψη μας και άλλους όρους και παραμέτρους, όπως τον τύπο της οθόνης που χρησιμοποιούμε, την πλατφόρμα και τα λοιπά. Ως πρώτο βιντεοπαιχνίδι στην ιστορία των ηλεκτρονικών υπολογιστών πάντως έχει επικρατήσει το γνωστό ως «Pong» το οποίο εμφανίστηκε την δεκαετία του 60.

Το «Pong» υπήρξε μια πολύ απλή εφαρμογή, καθώς αποτελούνταν μόνο από κινούμενα τετραγωνάκια, τα γνωστά σε όλους μας pixels. Είναι ένα παιχνίδι, που πολλοί από εμάς ίσως να έχουν παίξει, στην πιο εξελιγμένη μορφή του φυσικά, όπου ο παίχτης κινεί με το ποντίκι του ( όσο αφορά την έκδοσή του για ηλεκτρονικό υπολογιστή) ένα παραλληλόγραμμο αντικείμενο πάνω κάτω, και προσπαθεί έτσι να απωθήσει ένα μπαλάκι ώστε να μην περάσει την νοητή γραμμή πίσω από αυτό το αντικείμενο. Αλλά αντιθέτως προσπαθεί να στείλει αυτό το μπαλάκι ακριβώς απέναντί του, στην νοητή γραμμή του αντιπάλου, που στην προκειμένη περίπτωση αποτελεί μια πρώιμη μορφή Τεχνητής Νοημοσύνης, αφού ο αντίπαλός του είναι ο υπολογιστής.

- Να σημειωθεί πως το συγκεκριμένο ηλεκτρονικό παιχνίδι, γνώρισε τεράστια επιτυχία, σε παγκόσμιο επίπεδο. Δεν άργησαν να εμφανιστούν στις καφετέριες τα γνωστά κουτιά ηλεκτρονικών παιχνιδιών που λειτουργούν με κέρματα, με το όνομα Atari-Pong, αφού η Atari ήταν και η πρώτη εταιρία που το έβγαλε σε αυτήν την μορφή. Επομένως είναι λάθος να πιστεύουμε πως το Pacman, για το οποίο μας μιλάνε οι γονείς μας που έπαιζαν στις καφετέριες, είναι και το πιο παλιό ηλεκτρονικό παιχνίδι που υπήρχε σε αυτή τη μορφή. Michael Rohde (2014) Game Maker: Studio for Dummies

Όπως ειπώθηκε παραπάνω , υπάρχουν και αυτοί οι οποίοι αμφισβητούν την πρωτιά του Pong, και θεωρούν ως «Αρχηγούς» των ηλεκτρονικών παιχνιδιών, κάποια άλλα δημιουργήματα.

Ένα από αυτά είναι και το «Cathode Ray Tube Amusement Device» , δημιούργημα του 1947.

Το παιχνίδι αυτό, αποτελούσε έναν προσομοιωτή πυραύλων, κάτι που εμπνεύστηκαν οι σχεδιαστές του από τις οθόνες ραντάρ του Δεύτερου Παγκοσμίου Πολέμου. Το παιχνίδι χρησιμοποιούσε κάποια αναλογικά κυκλώματα και όχι ψηφιακά, και μπορούσε κάποιος να ελέγξει την πορεία και την θέση της κουκίδας στην οθόνη.

Το ίδιο έτος, έγινε επίσης μια προσπάθεια για τη δημιουργία του πρώτου ηλεκτρονικού προσομοιωτή για παιχνίδια σκακιού (Chess). Η προσπάθεια αυτή, αν και όχι εκατό της εκατό επιτυχής, έγινε από τους Alan Turing και Dietrich Prinz . Παρόλα αυτά ήταν ένα ακόμη μεγάλο βήμα του ανθρώπου όσο αφορά την νοημοσύνη των μηχανών.

Το 1951 γράφεται στην ιστορία ως το έτος που δημιουργήθηκε ο πρώτος υπολογιστής που φτιάχτηκε αποκλειστικά και μόνο για να παίζει ένα παιχνίδι. Ήταν στην ουσία ας πούμε η πρώτη κονσόλα ηλεκτρονικού παιχνιδιού, εάν μπορεί βέβαια να ονομαστεί έτσι λόγω των τεραστίων διαστάσεων που είχε. Το νόημα αυτής, NIMROD, και ο κατασκευαστής της ο Ferranti. Για την ιστορία, το παιχνίδι αυτό ονομαζόταν Nim.

Το 1952 έχουμε την εμφάνιση του OXO, ή αλλιώς Noughts and Crosses( Tic-Tac-Toe) γνωστό στην Ελλάδα ως Τρίλιζα. Ο Alexander S. Douglas δημιούργησε για πρώτη φορά, μια ηλεκτρονική μορφή του παιχνιδιού «Τρίλιζα», εμφανίζοντας έτσι στη «σκηνή των ηλεκτρονικών παιχνιδιών» το πρώτο παιχνίδι στον υπολογιστή που χρησιμοποιούσε μια ψηφιακή οθόνη γραφικών. Επίσης η εφαρμογή αυτή πρωτοπορεί στο ότι μπορούσε να αποθηκευτεί. Για να παίζει κάποιος αυτό το παιχνίδι, χρησιμοποιούσε ένα περιστρεφόμενο μοχλό.

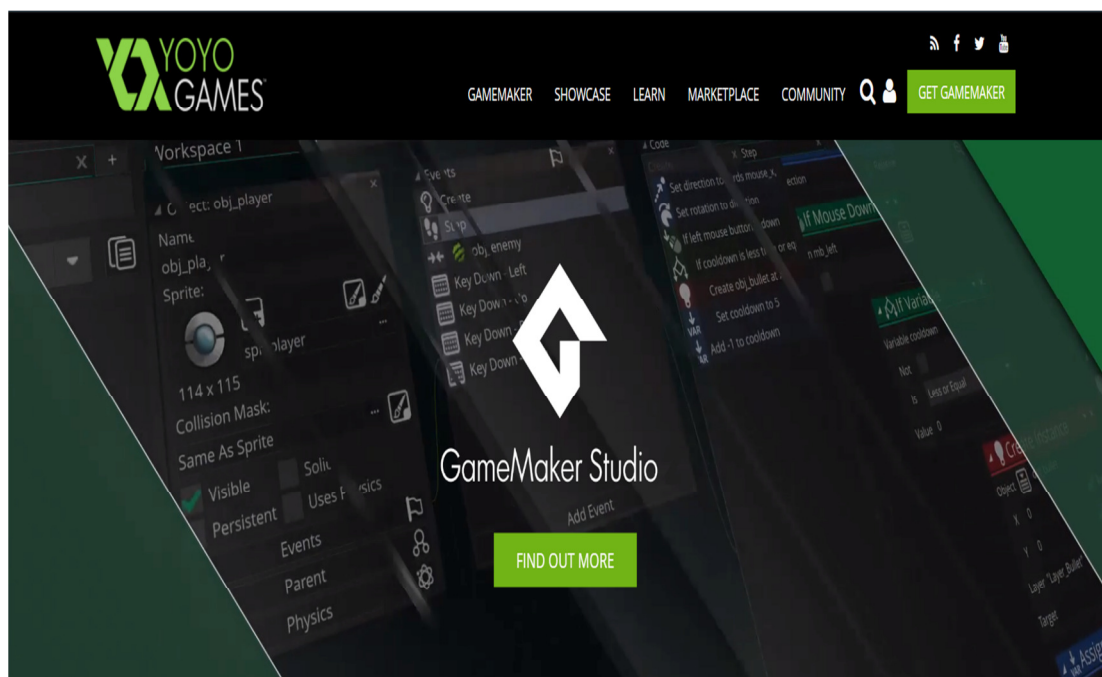
Εξίσου σημαντική εφεύρεση επιτεύχθηκε το 1958, από τον William Higinbotham, ο οποίος κατασκεύασε έναν εξομοιωτή τένις, ή σύμφωνα με κάποιους έναν εξομοιωτή Ping-Pong, το «Tennis for two».

## ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

# ΕΓΧΕΙΡΙΔΙΟ ΕΚΜΑΘΗΣΗΣ ΤΟΥ GAME MAKER STUDIO

### ΕΙΣΑΓΩΓΗ

Το GAME MAKER είναι από τα πιο διαδεδομένα εργαλεία, και κύριο στόχο έχουν την δημιουργία παιχνιδιών για διάφορες πλατφόρμες. Δημιουργός και εκδότης αυτής της εφαρμογής είναι ο Mark Overmars <http://www.yoyogames.com/> όπου είναι η επίσημη ιστοσελίδα της εταιρίας.



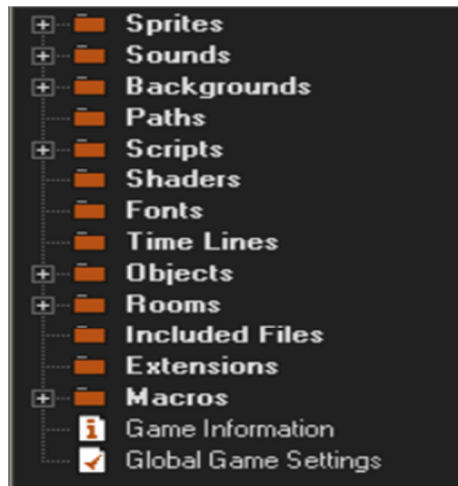
Ηλεκτρονική πλατφόρμα Game Maker Studio

Φυσικά το GAME MAKER STUDIO δεν είναι μόνο για Windows, αλλά και για τα υπόλοιπα γνωστά λειτουργικά συστήματα. Σκοπός του GAME MAKER

STUDIO είναι να παρέχει ένα φιλικό στον αρχάριο χρήστη πρόγραμμα ανάπτυξης δισδιάστατων παιχνιδιών, χωρίς όμως να ξεχνάει τους επαγγελματίες. Μπορεί ο οποιοσδήποτε να το χρησιμοποιήσει με την εκμάθηση κάποιων μόνο βασικών λειτουργιών της εφαρμογής. Το πρόγραμμα μπορεί να τρέξει στα περισσότερα λειτουργικά συστήματα, ακόμη και σε παλαιότερες εκδόσεις όπως τα WINDOWS XP. Με το GAME MAKER είναι δυνατόν κάποιος να δημιουργήσει επίσης και τρισδιάστατα παιχνίδια μαζί με μια πληθώρα άλλων επιλογών στην Full έκδοσή του όπως παιχνίδια για Android, για τον web ή ακόμη και για κονσόλες. Σημαντικό και αξιοσημείωτο είναι ότι το GAME MAKER παρέχει ένα περιβάλλον εύκολα χρησιμοποιούμενο ακόμα και από αρχάριους χρήστες, ενώ ταυτόχρονα, παρέχει τεράστιες δυνατότητες για ευπαρουσίαστα και εξελιγμένα παιχνίδια. Michael Rohde (2014) *Game Maker: Studio for Dummies*

## 2.1 ΟΙ ΟΝΤΟΤΗΤΕΣ ΤΟΥ GAME MAKER

- Οι «οντότητες» αποτελούν τον κορμό αλλά και το πρώτο κομμάτι γνώσης που πρέπει να κατανοήσουμε προκειμένου να κάνουμε τα πρώτα μας βήματα στην εφαρμογή. Αφού έχουμε ένα σχέδιο στο μυαλό μας για την βασική ιδέα του παιχνιδιού που θέλουμε να δημιουργήσουμε, τότε ξεκινάμε την εισαγωγή και την δημιουργία των ανάλογων οντοτήτων που θα χρειαστούμε, τις οποίες και θα δούμε εκτενέστερα στη συνέχεια. Όταν ανοίξουμε το GAME MAKER, θα δούμε μια λίστα όλων των διαφορετικών τύπων οντοτήτων που μπορούμε να χρησιμοποιήσουμε, στην αριστερή πλευρά της οθόνης. Οι οντότητες αυτές είναι τοποθετημένες σε μια στήλη που έχει δένδρως μορφή, όπως ακριβώς παρατηρούμε και στο παράθυρο εξερεύνησης αρχείων των Windows, που μπορούν να ανοίξουν και να κλείσουν. Αυτό το μέρος της οθόνης το ονομάζουμε : εξερευνητή των οντοτήτων. Michael Rohde (2014) *Game Maker: Studio for Dummies*



*Οντότητες GMS*

Ονομαστικά οι τύποι οντοτήτων είναι οι εξής:

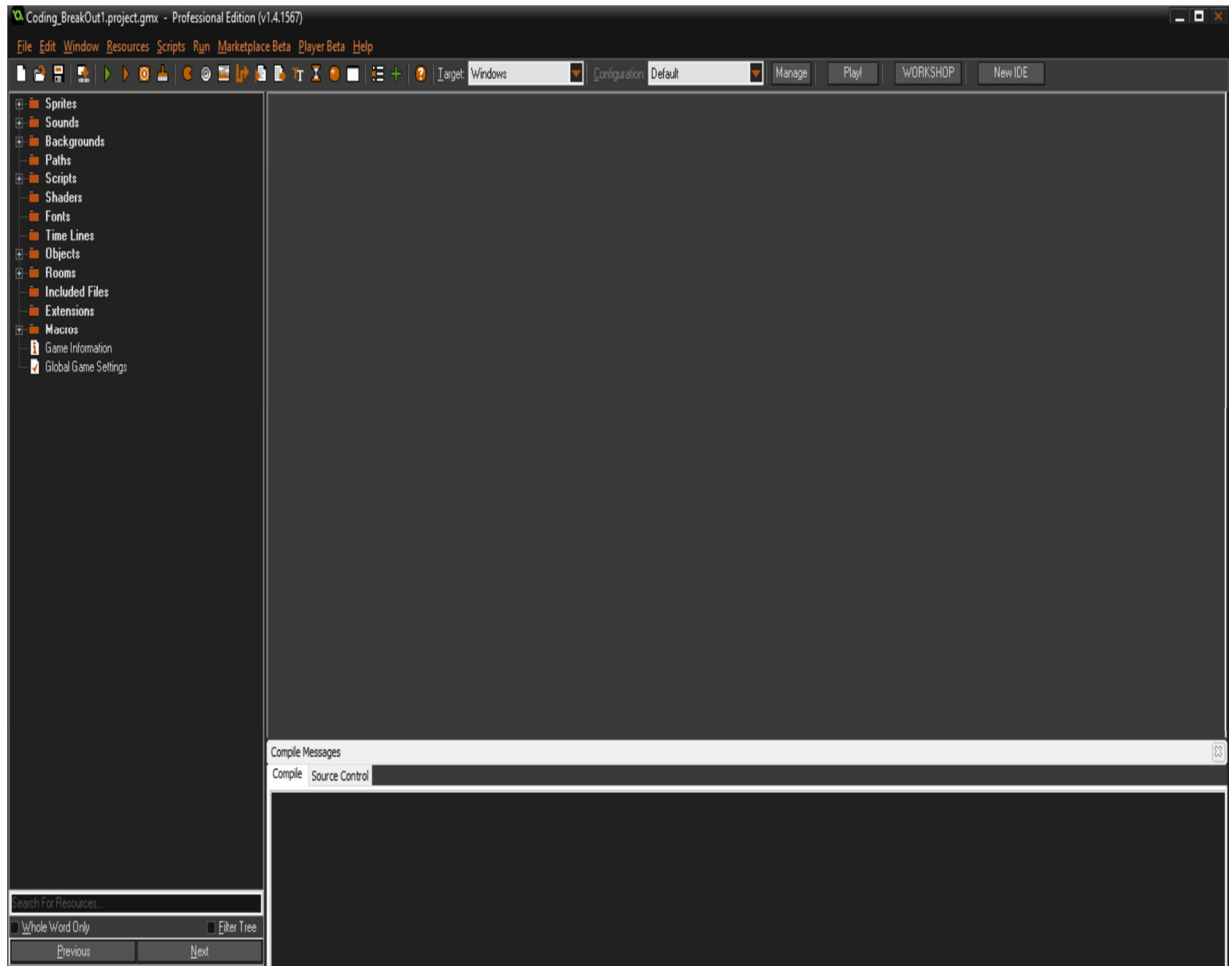
1. Sprites Sounds
2. Backgrounds
3. Paths Scripts
4. Fonts
5. Time Lines
6. Objects Rooms

Ας δούμε αναλυτικότερα τους τύπους οντοτήτων και την λειτουργία τους.

### **2.1.1 SPRITES**

Τα Sprites είναι ουσιαστικά είναι μία εικόνα ή μία σειρά εικόνων, επειδή εκτός από τα δεδομένα για μια εικόνα περιέχει και κάποιες ακόμα ιδιότητες που αφορούν το πλάτος, ύψος ,διαφάνεια, κ.ά. Το κάθε sprite μπορούμε να το φορτώσουμε στο παιχνίδι είτε με τη φόρτωση αρχείων εικόνας που είναι ήδη αποθηκευμένες στον υπολογιστή , είτε με τη δημιουργία νέων εικόνων. Να σημειωθεί πως ο τύπος των εικόνων που μπορούμε να φορτώσουμε είναι .ico ή .gif . Πρώτα θα δούμε πως γίνεται η δημιουργία μιας νέας οντότητας sprite . Αυτό μπορεί να γίνει με τρεις διαφορετικούς τρόπους:

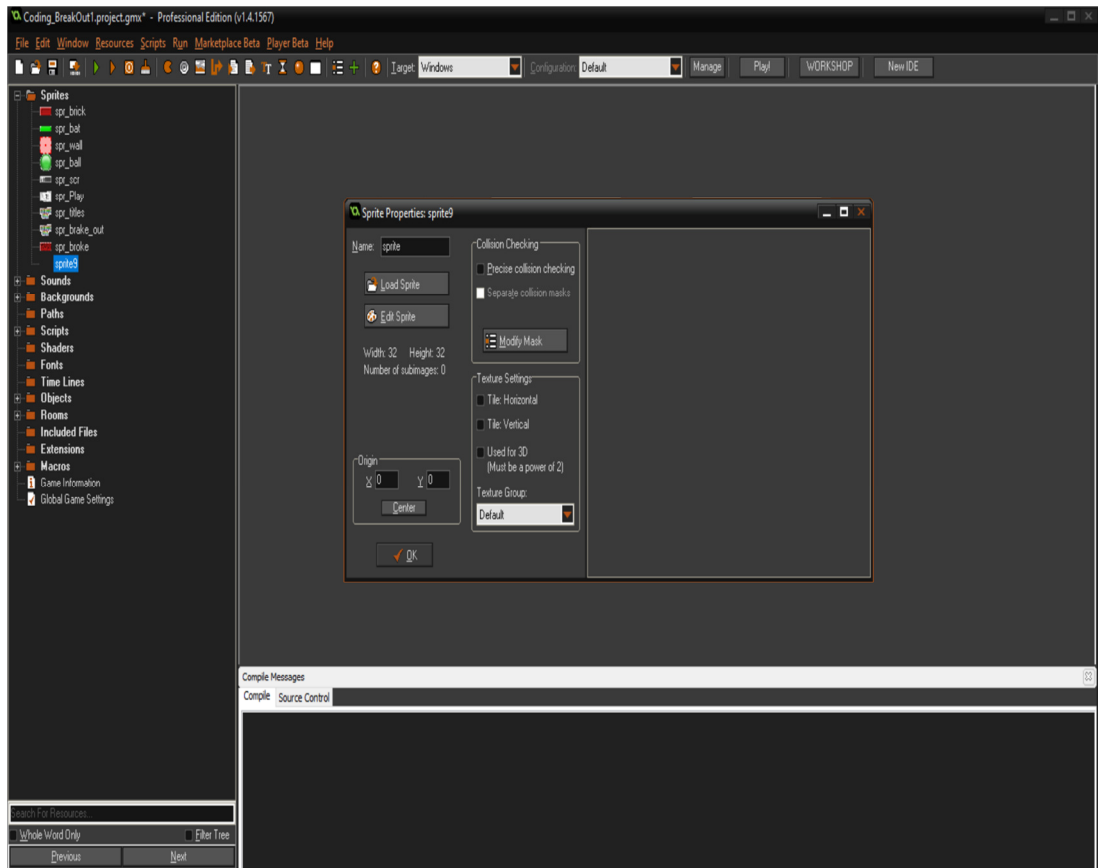
1. Από το βασικό μενού στην κορυφή της εφαρμογής Resources -> Create Sprite
2. Με τη βοήθεια του πληκτρολογίου πατώντας ταυτόχρονα CTRL-SHIFT-S
3. Με δεξί κλικ στο Sprite στο μενού και επιλογή Create Sprite



*Περιβάλλον GMS*

- Συνεχίζουμε σε ένα παράθυρο στο οποίο θέτουμε τις ιδιότητες που θέλουμε να έχει το sprite που θα δημιουργήσουμε. Habgood J. Overmars M. (2006) The Game Maker's Apprentice: Game Development for Beginners



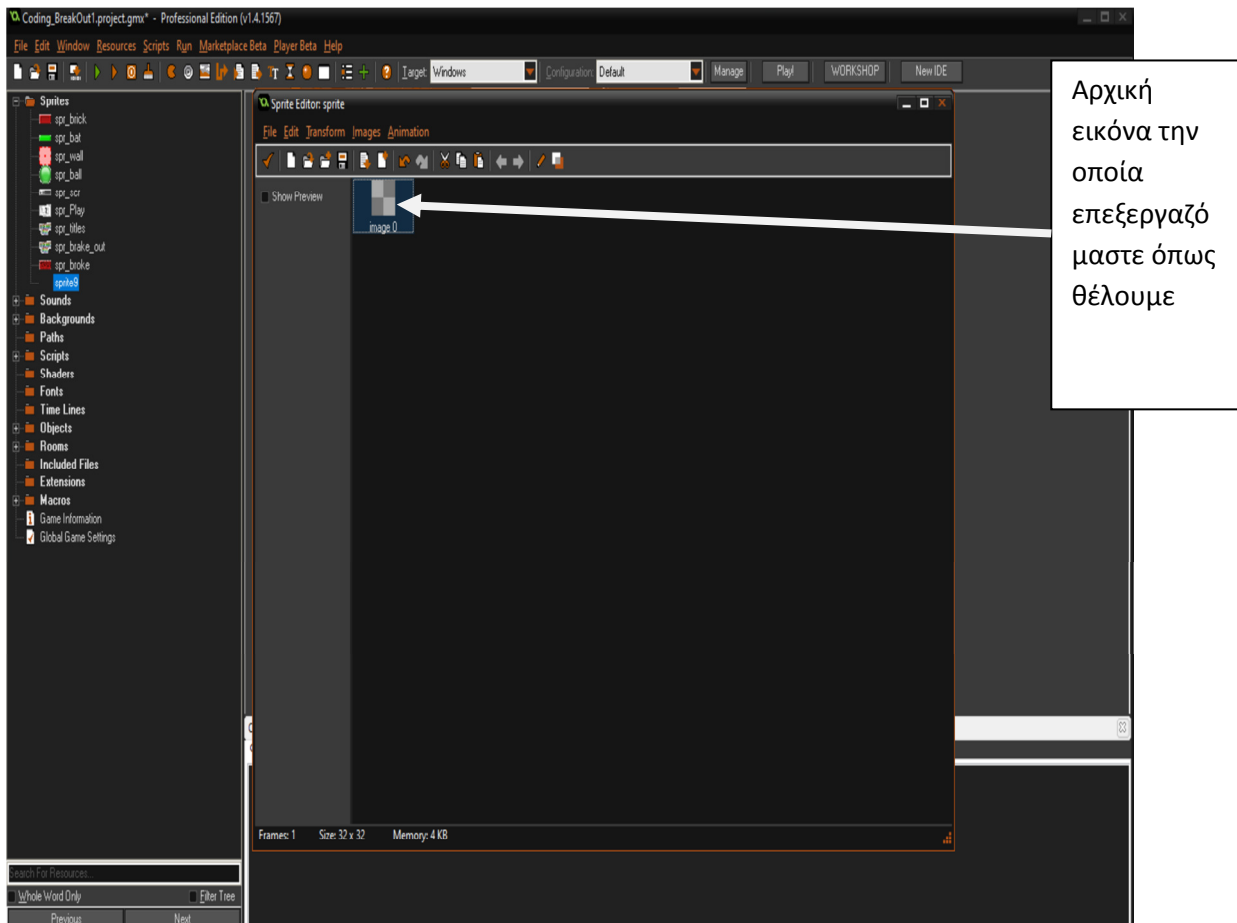


### *Ιδιότητες Sprite*

Το όνομα του sprite μπορεί μόνο να αποτελείται από λατινικούς χαρακτήρες μικρά ή κεφαλαία χωρίς καθόλου σύμβολα με εξαίρεση την κάτω παύλα και από αριθμούς. Το όνομα του sprite δεν μπορεί να ξεκινάει από αριθμό, αλλά μόνο από γράμμα της αλφαβήτου. Και να ξαναπώ πως τα το μόνο σύμβολο που μπορεί να χρησιμοποιηθεί στο όνομα είναι η κάτω παύλα και τίποτα άλλο. Το όνομα του sprite είναι πάρα πολύ σημαντικό διότι οποιαδήποτε στιγμή θελήσουμε να κάνουμε αλλαγές στο παιχνίδι, θα πρέπει να αναγνωρίζουμε και να ξεχωρίζουμε την κάθε οντότητά μας. Φανταστείτε μια εφαρμογή με 500 ή 1000 διαφορετικά sprites και ακόμη τόσα objects. Συνήθως τα sprites, για μεγαλύτερη ευκολία, ονομάζονται `spr_XXXX`, όπου `XXXX` κείμενο που μας παραπέμπει στο τι είναι το κάθε sprite. Σε αυτό το σημείο που πλέον η οντότητα (object) μας έχει κάποιο όνομα, σειρά έχει να της δώσουμε μορφή, όπου αυτό θα γίνει με την προσθήκη σε αυτήν μιας εικόνα. Αυτό μπορεί να γίνει με δύο διαφορετικούς τρόπους. Habgood J. Overmars M. (2006) *The Game Maker's Apprentice: Game Development for Beginners*

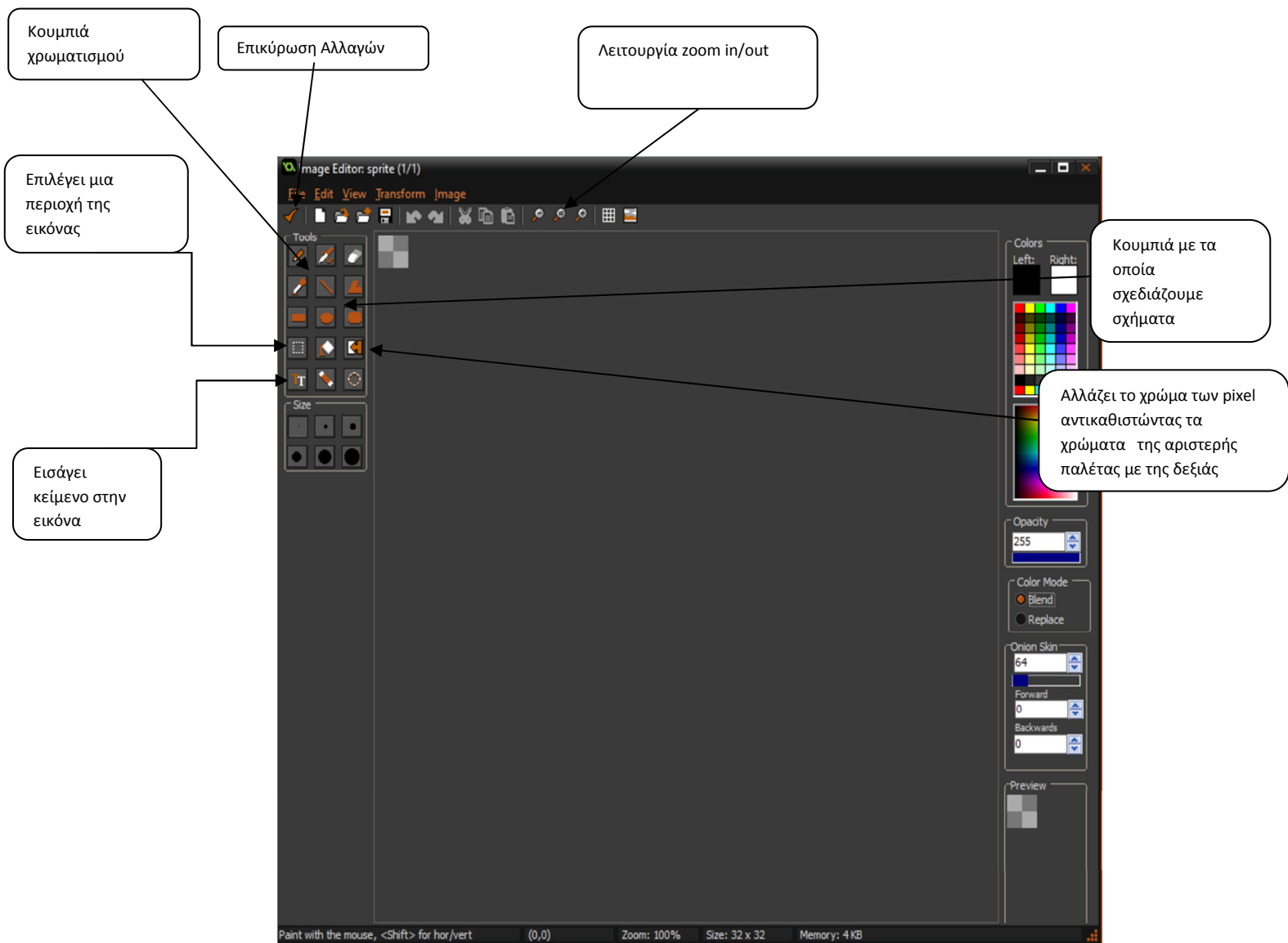
1. Ο πρώτος τρόπος είναι η φόρτωση μίας έτυμης εικόνας που έχουμε ήδη στον υπολογιστή μέσω του κουμπιού **load sprite**
2. Ο δεύτερος τρόπος είναι να δημιουργήσουμε μια καινούργια εικόνα μέσω του Game Maker Studio μέσω του κουμπιού **Edit Sprite**

Στην δεύτερη περίπτωση, όταν κάνουμε αριστερό κλικ στο Edit Sprite θα εμφανιστεί το παρακάτω παράθυρο:



### *Επεξεργασία Sprite*

Για να συνεχίσουμε επιλέγουμε στην κορυφή του μενού : Edit ->Edit ή κάνουμε διπλό κλικ πάνω στην εικόνα, κι έτσι προκύπτει το: Image Editor, στο οποίο θα επεξεργαστούμε την εικόνα μας.



### *Εργαλεία Επεξεργασίας Sprite*

Εδώ είναι ένα καλό σημείο να κάνουμε μία αναφορά στα **Backgrounds**. Από εδώ μπορούμε να προσθέσουμε ένα **Backgrounds** ή και ακόμη να το δημιουργήσουμε.

### 2.1.2 SOUNDS

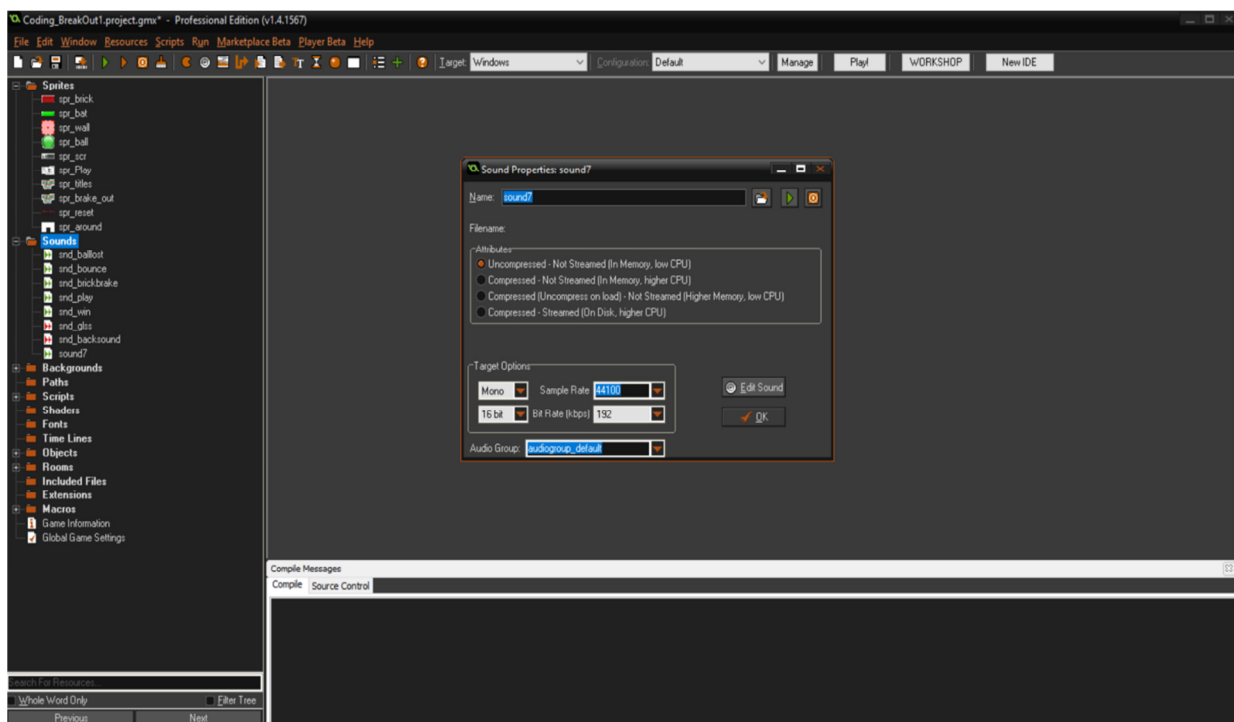
Ήχοι και η μουσική έχουν πάρα πολύ σημαντικό ρόλο για τη υλοποίηση ενός παιχνιδιού. Φανταστείτε ένα παιχνίδι το οποίο δεν έχει καθόλου μουσική και δεν ακούγεται κανένας ήχος σε σχέση με αυτά που γίνονται στην οθόνη. Επίσης, σημασία δεν έχει απλά να ακούμε κάποιον ή κάποιους ήχους και να είναι τελείως άσχετοι με το παιχνίδι. Οι ήχοι οι οποίοι θα επιλέξουμε πρέπει να είναι οι κατάλληλοι των περιστάσεων με βάση το παιχνίδι και το είδος παιχνιδιού που έχουμε σκοπό να δημιουργήσουμε. Δεν γίνεται να παίζει κάποιος ένα παιχνίδι πολέμου και όταν σκοτώνει έναν εχθρό του παιχνιδιού να ακούγεται κάποιος χαρούμενος ήχος, ή σαν μουσική υποβάθρου να ακούγεται ένα χαρούμενο τραγούδι.

Μπορούμε αν θέλουμε να εισάγουμε δικούς μας ήχους ή κάποια δικά μας τραγούδια που έχουμε αποθηκευμένα στον υπολογιστή. Αν θέλουμε όμως, έχουμε και την δυνατότητα να επιλέξουμε κάποιους από τους ήχους που υπάρχουν στη επίσημη διαδικτυακή σελίδα του GAME MAKER, ή και σε άλλες ιστοσελίδες ( δίνεται η δυνατότητα της ανάκτησης ολόκληρων φακέλων με ήχους, ώστε να επιλέξουμε τους κατάλληλους για το εκάστοτε παιχνίδι που δημιουργούμε ).

Για να προσθέσουμε έναν ήχο στο project μας ακολουθούμε έναν από τους τρόπους που βλέπουμε στην συνέχεια:

1. Από το μενού στην κορυφή της εφαρμογής Resources -> Create Sound
2. Με το πληκτρολόγιο πατώντας ταυτόχρονα CTRL-SHIFT-U
3. Με δεξί κλικ του ποντικιού στο Sounds στο μενού αριστερά και επιλογή Create Sound

Όταν εισάγουμε έναν ήχο υπάρχουν παράμετροι που μπορούμε να αλλάξουμε ανάλογος με τις απαιτήσεις του παιχνιδιού που δημιουργούμε. Πολύ σημαντική παρατήρηση, είναι η επιλογή που μας δίνεται, ώστε να ορίσουμε, για τον εκάστοτε ήχο, να φορτώνεται στο παιχνίδι μας την χρονική στιγμή που έχουμε ορίσει, και να μην φορτώνεται από την αρχή, κατά την εκκίνηση του παιχνιδιού. Αυτό βοηθάει στην μνήμη του υπολογιστή στο οποίο πρόκειται να τρέξει το παιχνίδι, στους πόρους του συστήματος που θα χρειαστεί να διατεθούν, και στο μέγεθος του αποθηκευτικού χώρου στο δίσκο που θα χρειαστεί να δαπανηθεί. Habgood J. Overmars M. (2006) The Game Maker's Apprentice: Game Development for Beginners



### Εισαγωγή Ήχου

Name: Σε αυτό το πεδίο θα εισάγουμε το όνομα. Έχει πάρα πολύ μεγάλη σημασία το όνομα που θα δώσουμε στον κάθε ήχο για τους ίδιους λόγους που αναφέραμε και για τα Sprites. Στο Game Maker, δεν υπάρχει κάποιος Editor για την δημιουργία ήχων. Μπορούμε όμως να χρησιμοποιήσουμε άλλους επεξεργαστές ήχων ξεχωριστά ή μέσω plug ins και να δημιουργήσουμε τους ήχους που χρειαζόμαστε και να τους φορτώσουμε στην εφαρμογή. Όταν ανεβάσουμε έναν ήχο, μπορούμε να τον αποθηκεύσουμε στο Game Maker, ώστε σε περίπτωση που χάσουμε το πρωτότυπο αρχείο ήχου, να μην έχουμε κάποιο πρόβλημα. Δίνεται η δυνατότητα μεταφόρτωσης

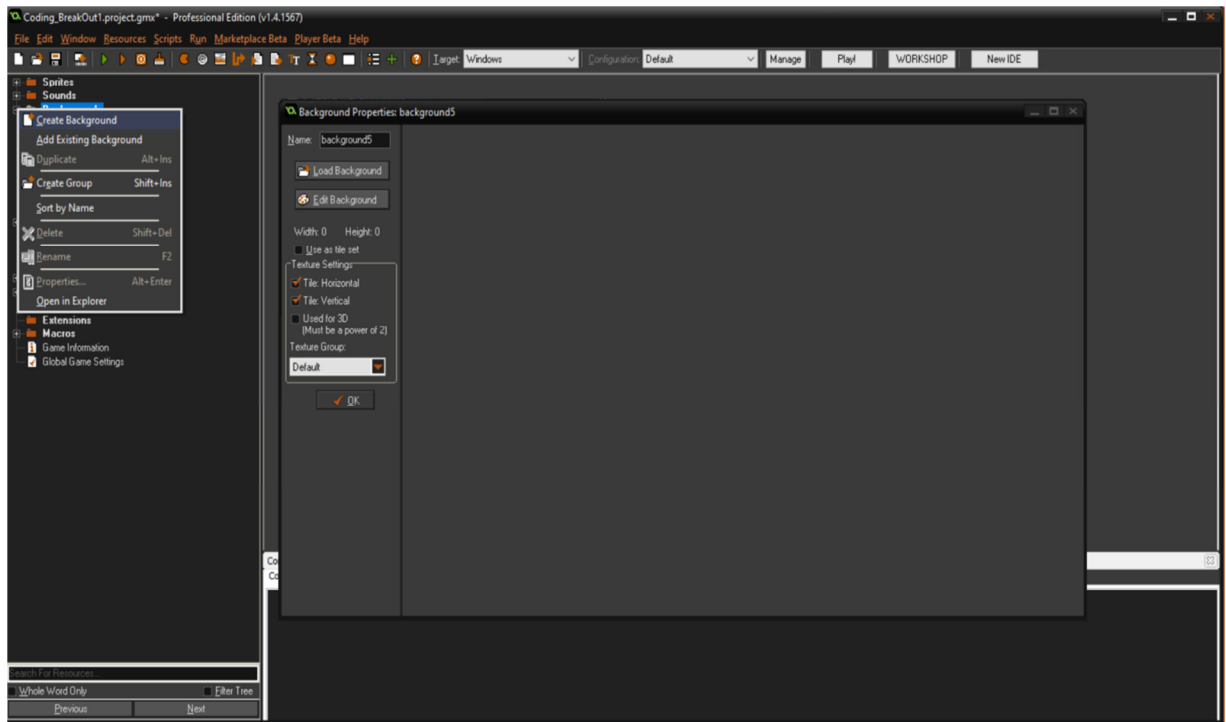
τριών (3) διαφορετικών τύπων αρχείων και είναι οι εξής : MP3 , Midi , Wave . Υπάρχουν δύο buttons , ένα πράσινο ( play ), με το οποίο μπορούμε να τεστάρουμε τον ήχο που φορτώσαμε, κι ένα κόκκινο ( stop ), με το οποίο σταματάμε την αναπαραγωγή του συγκεκριμένου ήχου. Εάν δεν πατήσουμε το stop, ο ήχος συνεχίζει να παίζει ασταμάτητα και επαναλαμβανόμενα. Habgood J. Overmars M. (2006) The Game Maker's Apprentice: Game Development for Beginners

### 2.1.3 BACKGROUNDS

Το κάθε δωμάτιο η πίστα του παιχνιδιού θα πρέπει να έχει και ένα φόντο, ένα Background. Το φόντο αυτό είναι μια εικόνα η οποία είναι μονίμως πίσω από τους χαρακτήρες και τα αντικείμενα του παιχνιδιού μας. Αλλά έχουμε και τη δυνατότητα να επιλέξουμε εάν η εικόνα αυτή να έχει κάποια διαφάνεια ή όχι. Το Game Maker, μας δίνει τη δυνατότητα να επιλέξουμε ως φόντο μία από τις εικόνες που έχει ήδη στις βιβλιοθήκες του. Επίσης, στο διαδίκτυο υπάρχουν πάρα πολλές , δωρεάν ή μη, βιβλιοθήκες, από επίσημες ιστοσελίδες του παιχνιδιού είτε από αλλού, όπου μπορούμε να αντλήσουμε εικόνες και να της χρησιμοποιήσουμε για το εκάστοτε project αφού πρώτα τις περάσουμε στην εφαρμογή. Τέλος, μας δίνεται η δυνατότητα να δημιουργήσουμε δικιές μας εικόνες, ώστε να τις χρησιμοποιήσουμε σε όποιο δωμάτιο ( Room ) χρειαστούμε του παιχνιδιού που σχεδιάζουμε. Αυτό μπορεί να γίνει είτε στην ειδική πλατφόρμα που μας παρέχει το Game Maker, είτε απλά μεταφορτώνοντας κάποιες από τις εικόνες που έχουμε ήδη υπάρχουν αποθηκευμένες στον υπολογιστή μας. Για να εισάγουμε ένα φόντο ( Background ) στο παιχνίδι μας ώστε να μπορέσουμε να το χρησιμοποιήσουμε σε κάποιο δωμάτιο, πρέπει να ακολουθήσουμε έναν από τους παρακάτω τρόπους:

1. Από το βασικό μενού της εφαρμογής Resources -> Create Background
2. . Με το πληκτρολόγιο πατώντας ταυτόχρονα CTRL-SHIFT-B
3. Με δεξί κλικ στο Backgrounds στο μενού και επιλογή Create Background

Για να τροποποιήσουμε ένα φόντο ή για να δημιουργήσουμε ένα νέο : Edit Background στη φόρμα “Background Properties”

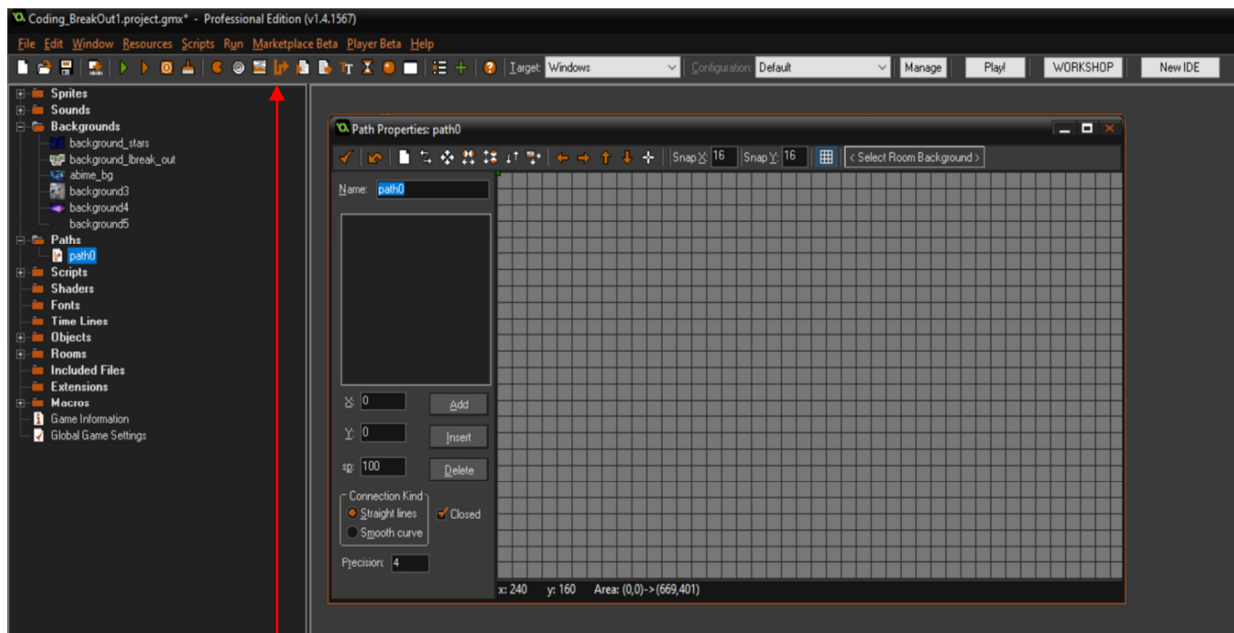


#### *Δημιουργία Background*

**Name:** Σε αυτό το πεδίο θα εισαχθεί το όνομα του Background που δημιουργήσαμε, εισάγαγέ ή επιλέξαμε. Η ονομασία που θα δοθεί έχει μεγάλη σημασία για τους λόγους που είπαμε και για τα Sprites . Habgood J. Overmars M. (2006) The Game Maker's Apprentice: Game Development for Beginners

### **2.1.4 PATHS**

Για να ακολουθούν τα βασικά μέλη του παιχνιδιού ένα βασικό μονοπάτι αυτό που έχουμε να κάνουμε είναι: Δημιουργούμε ένα μονοπάτι σχεδιάζοντάς το και στην συνέχεια τοποθετούμε και εφαρμόζουμε τα ανάλογα events που θέλουμε να λάβουν χώρα στην πορεία που θα ακολουθήσει κάποιο αντικείμενό μας.



### Δημιουργία Μονοπατιών

Για νέο Path στο παιχνίδι θα πρέπει να χρησιμοποιήσουμε έναν από τους παρακάτω τρόπους :

1. Κλικ στο κουμπί «Create a Path»
2. Κλικ Resources -> Create Path από το μενού πάνω
3. Πληκτρολογώντας στο πληκτρολόγιο CTRL- SHIFT -P
4. Δεξί κλικ στο Paths από το μενού και επιλογή Create Path

Habgood J. Overmars M. (2006) The Game Maker's Apprentice: Game Development for Beginners

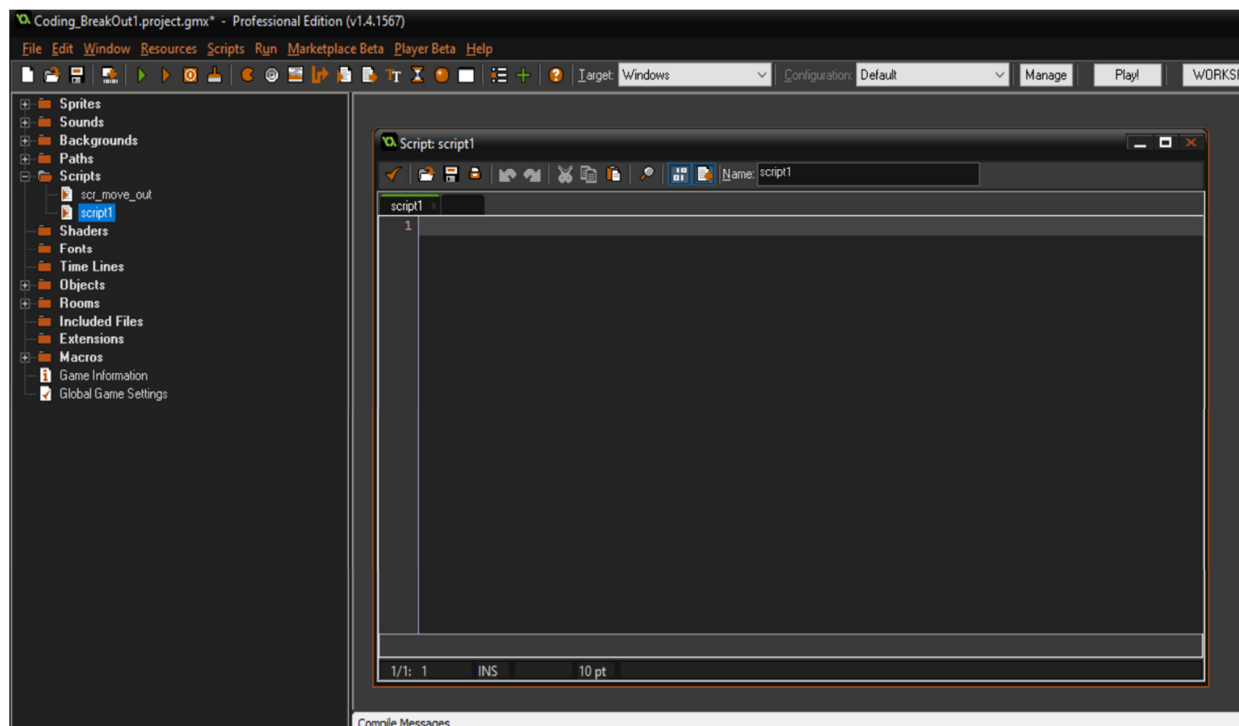


## 2.1.5 SCRIPTS

Εδώ μπορούμε να γράψουμε ολόκληρα κομμάτια κώδικα από μόνοι μας και να τα αποθηκεύσουμε και να τα χρησιμοποιήσουμε όποτε και όπου θέλουμε στο παιχνίδι που φτιάχνουμε. Η δημιουργία δικών μας κομματιών κώδικα πέρα από αυτά που έχει, βοηθάει στην στη λεπτομέρεια και στην επίλυση πόλων θεμάτων που εμφανίζονται κατά το development του παιχνιδιού . Αυτό φυσικά μας δίνει πολύ καλύτερα αποτελέσματα. Συνήθως αποθηκεύουμε κομμάτια κώδικα σε Scripts τα πρόκειται να επαναλάβουμε πολλές φορές στο παιχνίδι μας. Η γλώσσα που χρησιμοποιείται στο Game Maker είναι η GML ( Game Maker Language ) . Με τον τρόπο σύνταξης της GML θα ασχοληθούμε εκτενέστερα παρακάτω.

Για να εισάγουμε Script.

1. Επιλογή Resources -> Create Script από το μενού πάνω
2. Πληκτρολογώντας στο πληκτρολόγιο CTRL- SHIFT –C
3. Δεξί κλικ στο Scripts από το μενού αριστερά και επιλογή Create Script



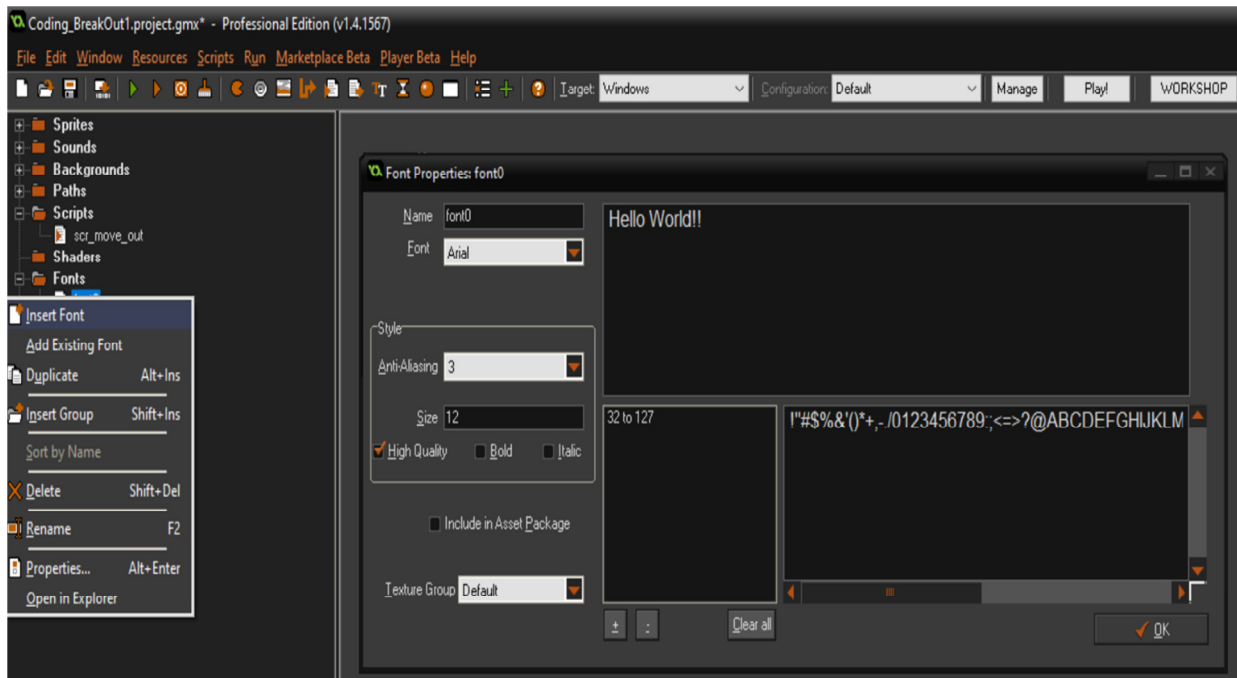
*Δημιουργία Script*

Name: Στο πεδίο βάζουμε το όνομα του Script. Lee Ford J. (2009) Getting Started with Game maker Studio

### 2.1.6 FONTS

Όπως προΐδεάζει και το όνομα της οντότητας εδώ εισάγουμε της γραμματοσειρές που αργότερα θα χρησιμοποιήσουμε στο παιχνίδι μας συμπεριλαμβάνοντας όλες τις παραμέτρους , όπως για παράδειγμα το Bolt , το Style , και το μέγεθος της γραμματοσειράς. Έτσι, εάν σε κάποιο σημείο του θέλουμε να αναγράφεται κάτι με συγκεκριμένη γραμματοσειρά την έχουμε ήδη έτοιμη και την χρησιμοποιούμε πάνω στο κείμενο που θέλουμε. Αυτό γίνεται κυρίως και για λόγους ταχύτητας, διότι μια αποθηκευμένη γραμματοσειρά μπορούμε να την χρησιμοποιήσουμε σε περισσότερα από ένα σημεία στο παιχνίδι μας, κι έτσι δεν χρειάζεται να επαναλαμβάνουμε την διαδικασία αναζήτησης επιλογής και γενικότερα ρύθμισης της γραμματοσειράς για κάθε φορά που την χρειαζόμαστε. Φανταστείτε πόσο χρόνο θα έπαιρνε αυτό σε ένα παιχνίδι text adventure. Για να εισάγουμε μια γραμματοσειρά ( Font ) ώστε να μπορέσουμε στη συνέχεια να τη χρησιμοποιήσουμε σε κάποιο ή κάποια σημεία του παιχνιδιού , πρέπει να ακολουθήσουμε έναν από τους παρακάτω τρόπους :

1. Resources -> Create Font από το μενού πάνω
2. Πληκτρολογώντας στο το πληκτρολόγιο CTRL- SHIFT –F
3. Δεξί κλικ στο Fonts από το μενού και επιλογή Create Font

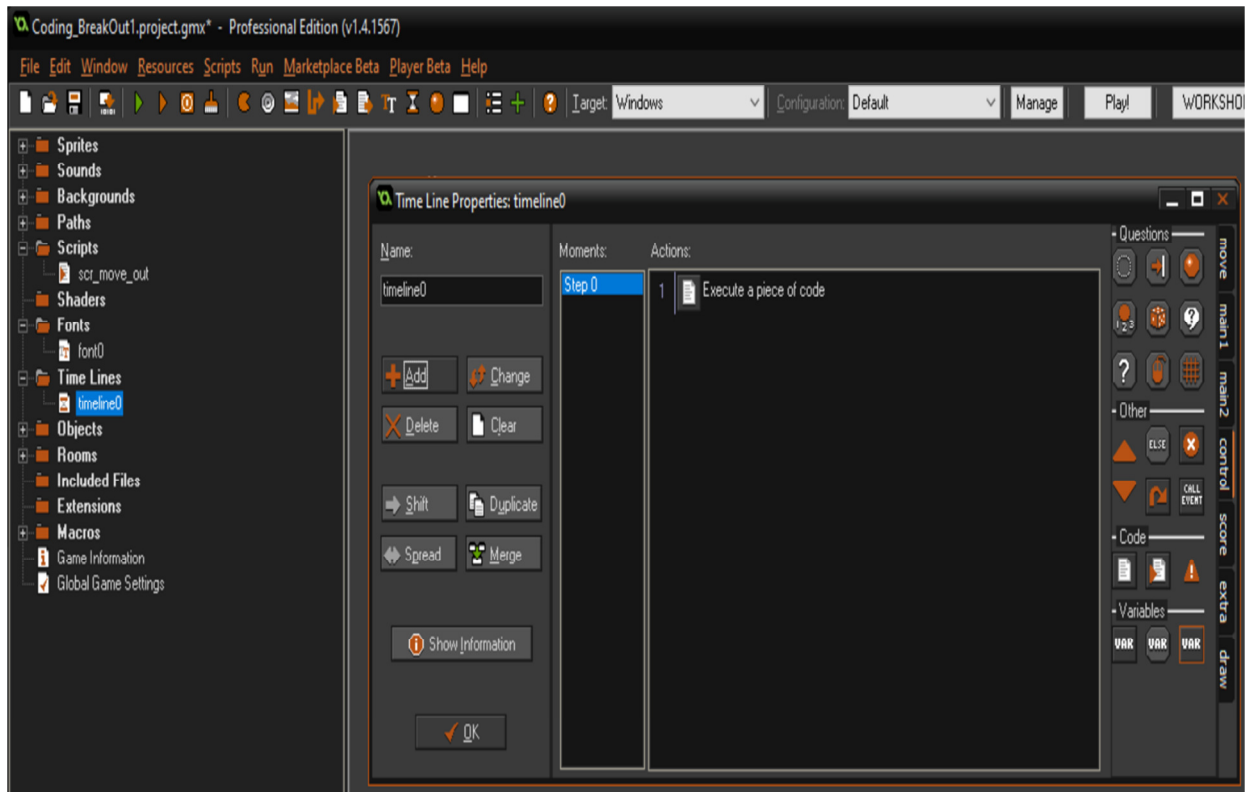


### Δημιουργία Fonts

Name: Εισάγουμε το όνομα του Font που δημιουργήσαμε. Lee Ford J. (2009)  
Getting Started with Game maker Studio

### 2.1.7 TIME LINES

Είναι πολύ συνηθισμένο σε ένα παιχνίδι κάποια πράγματα να συμβούν σε συγκεκριμένες χρονικές στιγμές. Αυτό γίνεται με το Alarm για το οποίο θα μιλήσουμε εκτενέστερα στην αργότερα. Όταν όμως το project είναι περίπλοκο, η μέθοδος του Alarm περιπλέκει τα πράγματα και δεν έχουμε το επιθυμητό αποτέλεσμα.



#### Δημιουργία Timeline

Για να εισάγουμε Time Line ώστε να μπορέσουμε στη συνέχεια να το χρησιμοποιήσουμε σε κάποιο ή κάποια σημεία του παιχνιδιού , πρέπει να ακολουθήσουμε έναν από τους παρακάτω τρόπους :

1. Resources ->Create Time Line από το μενού πάνω

2. Πληκτρολογώντας στο πληκτρολόγιο CTRL- SHIFT –T

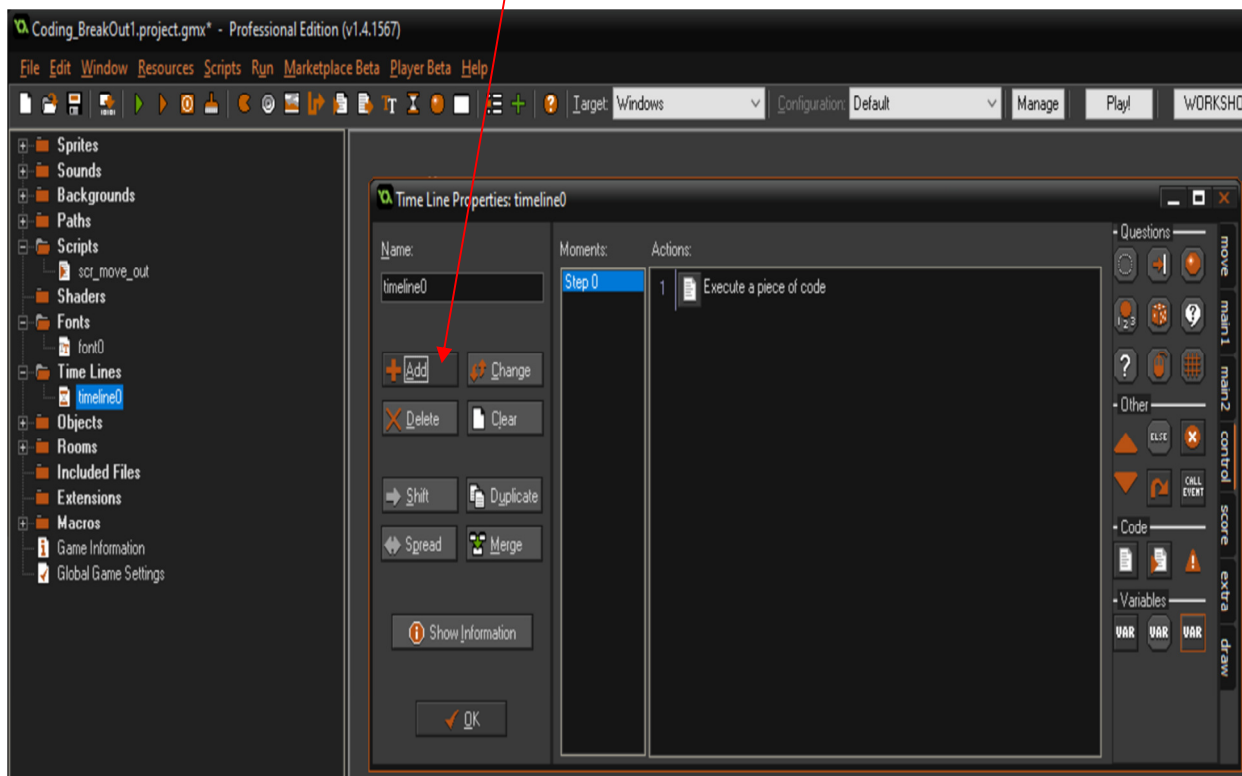
3. Δεξί κλικ στο Time Lines από το μενού και Create Time line. Lee Ford J.

(2009) Getting Started with Game maker Studio

## 2.1.8 OBJECTS

Τα αντικείμενα (Objects) αποτελούν τη πιο σημαντική οντότητα του Game Maker Studio. Είναι τα ενεργά αντικείμενα του κάθε παιχνιδιού που αλληλοεπιδρούν με το περιβάλλον και τον χρήστη και είναι αυτό που δίνει ζωντάνια στα παιχνίδια που θα δημιουργήσουμε. Το κάθε object συνήθως αντιπροσωπεύεται και από ένα sprite, αλλά όχι πάντα. Για να εισάγουμε ένα object στο παιχνίδι μας, πρέπει να ακολουθήσουμε έναν από τους παρακάτω τρόπους :

### 1. Κλικ το κουμπί Add an Object



*Δημιουργία Αντικειμένου*

2. Resources ->Create Object

3. Πληκτρολογώντας στο πληκτρολόγιο CTRL- SHIFT -O

4. Δεξί κλικ στο Objects και επιλογή Create Object

Αφού δημιουργήσουμε ένα object εμφανίζεται ένα παράθυρο όπου θα επιλέξουμε τις ιδιότητες που θέλουμε να έχει το αντικείμενό μας:

Name: Στο πεδίο αυτό μπαίνει το όνομα του Object. Ξεκινάμε την ονομασία των Objects συνήθως με obj\_XXX.

Sprite: Επιλέγουμε κάποιο από τα Sprite που έχουμε ήδη. Ένα Sprite δεν είναι τίποτα απολύτως από μόνο του και δεν μας χρησιμεύει πουθενά στο παιχνίδι που έχουμε ξεκινήσει να φτιάχνουμε, παρά μόνο όταν συνδεθεί με κάποιο.

Visible: Όταν είναι επιλεγμένο το Sprite είναι ορατό στο παιχνίδι ή το αντίθετο. Υπάρχουν και περιπτώσεις όπου θέλουμε σε κάποιο σημείο του παιχνιδιού να τοποθετούνται αντικείμενα τα οποία θα αλληλοεπιδρούν με τα υπόλοιπα, αλλά δεν θέλουμε φαίνονται όπως ένας αόρατος τοίχος, τον οποίο κανένα από τα άλλα objects δεν μπορεί να τον διαπεράσει

Solid: Όταν είναι επιλεγμένο όπως λέει και το ίδιο το όνομα το sprite είναι συμπαγές και επομένως αδιαπέραστο από οποιοδήποτε αντικείμενο που κινείται μέσα στο παιχνίδι, για παράδειγμα ένας τοίχος ή το δάπεδο ή κάποιο εμπόδιο. Lee Ford J. (2009) Getting Started with Game maker Studio

### **2.1.9 ROOMS**

Τα παιχνίδια που δημιουργούμε παίρνουν μέρος σε ένα ή σε περισσότερα δωμάτια τα οποία ονομάζονται rooms. Τα δωμάτια (rooms) αυτά αποτελούν τα διάφορα επίπεδα του παιχνιδιού μας, τα οποία είναι μη τρισδιάστατα, αλλά μπορούν να περιέχουν γραφικά που μοιάζουν με 3D μια παλιά τεχνική που περιλαμβάνει έξυπνη χρήση της γωνίας της κάμερας. Τα rooms αντιστοιχούν σε πίστες ή επίπεδα των ηλεκτρονικών παιχνιδιών.

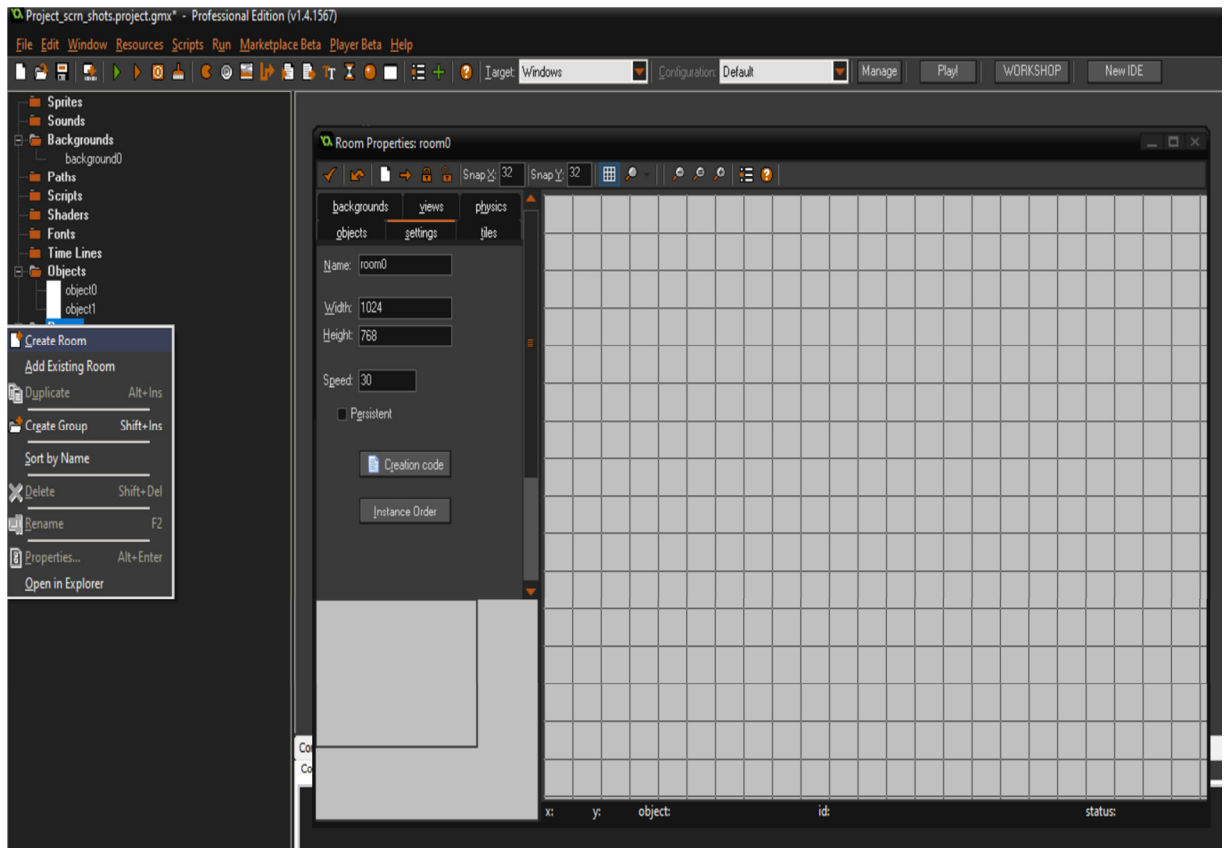
Για να εισάγουμε ένα δωμάτιο στο παιχνίδι μας, πρέπει να ακολουθήσουμε έναν από τους παρακάτω τρόπους :

#### **1. Resources -> Create Room**

2. Πληκτρολογώντας στο πληκτρολόγιο CTRL- SHIFT –R

3. Δεξί κλικ στο Rooms από το μενού αριστερά και επιλογή Create Room

Δημιουργώντας ένα νέο δίνεται η δυνατότητα να επεξεργαστούμε κάποιες ιδιότητες.



### *Δημιουργία Δωματίου*

Name: Εισάγουμε το όνομα του δωματίου και ξεκινάμε την ονομασία των Rooms με rm\_XXX

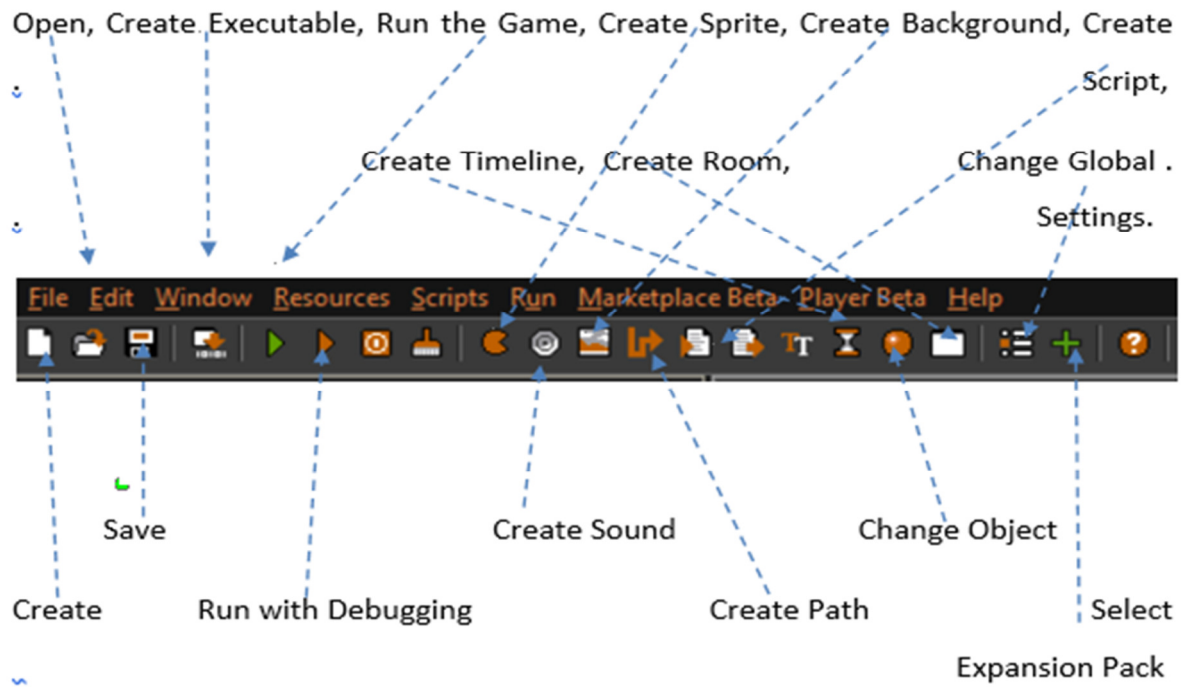
## ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

### ΔΗΜΙΟΥΡΓΙΑ ΠΑΙΧΝΙΔΙΩΝ / EVENTS ΚΑΙ ACTIONS

Αφού για αρχή δημιουργήσουμε δωμάτια (room) και τους έχουμε βάλει κάποιο φόντο (background), ξεκινάμε να τοποθετήσουμε αντικείμενα (objects) Drag And Drop αφού πρώτα επιλέξουμε το ανάλογο object στο tab objects του δωματίου. Έτσι χτίζουμε κομμάτι την πίστα του παιχνιδιού μας. Θα βάλουμε τα κατάλληλα αντικείμενα, με το παιχνίδι που δημιουργούμε, αφού φυσικά όπως είπαμε τα έχουμε αντιστοιχίσει με κάποιο Sprite όπου και όταν αυτό χρειάζεται. Συνήθως βάζουμε πρώτα τους τοίχους μαζί με κάποια αντικείμενα και φυσικά κάποιον ή κάποιους χαρακτήρες, τα οποία στην πορεία θα έχουν και την ανάλογη αλληλεπίδραση μεταξύ τους που θέλουμε. Είναι πολύ συνηθισμένο να χρειάζεται να τοποθετήσουμε και κάποια κείμενα όπως τα score ή τις ζωές με κόπιες τιμές οι οποίες μεταβάλλονται ανάλογος με την έκβαση του παιχνιδιού. Αυτό μπορεί ακόμη να επηρεάζει την πορεία του παιχνιδιού. Για να γίνει εφικτό αυτό θα πρέπει να ορίσουμε και κάποιες μεταβλητές κατά την εκκίνηση του παιχνιδιού και οι οποίες θα αλλάζουν τιμές ανάλογα με τις διάφορες αλληλεπιδράσεις που θα υπάρξουν στο παιχνίδι, και για τις οποίες θα μιλήσουμε στην συνέχεια. Πολλές φορές υπάρχουν αντικείμενα (objects) τα οποία τοποθετούμε, αλλά είναι αόρατα, δηλαδή δεν φαίνονται κατά την διάρκεια που κάποιος παίζει το παιχνίδι. Για παράδειγμα, αυτά μπορεί να είναι ήχοι, μουσικές υποκρούσεις που θέλουμε να εκτελούνται, αλλά χωρίς να φαίνονται.

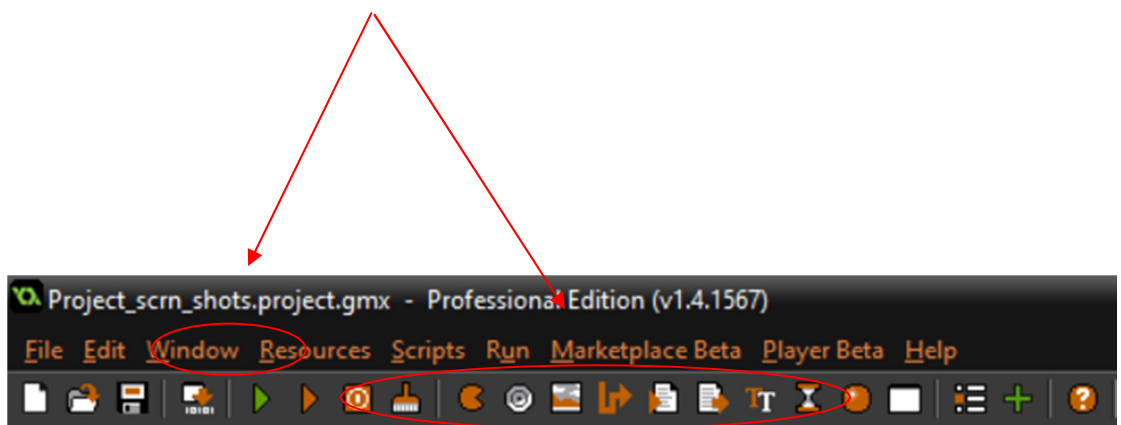
Συνεχίζουμε με βασικές γνώσεις που κάποιος πρέπει να γνωρίζει για να δημιουργήσει ένα παιχνίδι στο Game Maker Studio. Ας πούμε λοιπόν ότι έχουμε στήσει το δωμάτιο στο οποίο θέλουμε να δουλέψουμε με όλα αυτά που είπαμε παραπάνω. Κι έτσι ξεκινάμε να δημιουργούμε τις διάφορες αλληλεπιδράσεις μεταξύ τους. Να τονίσουμε πως το Game Maker μας δίνει την δυνατότητα είτε του προγραμματισμού των εντολών που θέλουμε να δώσουμε με την εξ' αρχής γραφή κώδικα στη γλώσσα του Game Maker, για την οποία θα μιλήσουμε στην συνέχεια, είτε με την χρήση έτοιμων στην ουσία εντολών, με την μέθοδο του Drag And Drop, χάρη στο γραφικό περιβάλλον που μας δίνει η εφαρμογή. Αυτή τη στιγμή λοιπόν θα ασχοληθούμε με την δεύτερη μέθοδο. Θα εξηγήσουμε αρχικά το Toolbar που υπάρχει σε οριζόντια θέση, στην κορυφή της εφαρμογής.





Και φυσικά για λόγους ευκολίας όπως σε όλα τα προγράμματα υπάρχει toolbar για γρήγορη πρόσβαση σε εργαλεία, έτσι και στο Game Maker Studio.

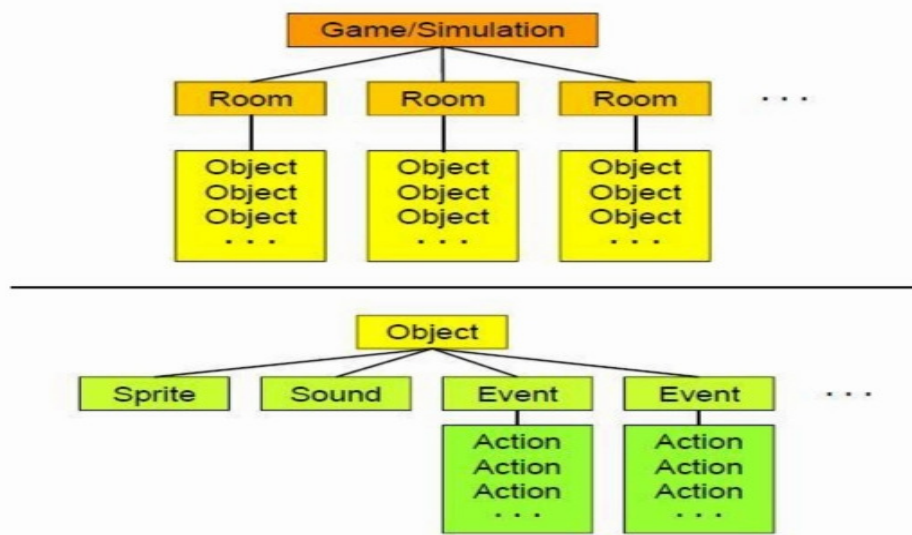
### Αυτά τα Βρίσκουμε εδώ



Μπάρα Εργαλείων GMS

Ήρθε η ώρα να εξηγήσουμε τη βασική δομή με την οποία γίνεται η δημιουργία ενός παιχνιδιού στη εφαρμογή αυτή. Όπως είπαμε, μεταξύ των Objects που έχουμε εισάγει σε ένα Room, υπάρχουν αναπόφευκτα αλληλεπιδράσεις, τις οποίες αποκαλούμε Events. Για παράδειγμα αλληλεπίδραση υπάρχει πχ ανάμεσα στο έδαφος και σε ένα αντικείμενο, όταν αυτά έρθουν σε επαφή. Όταν λοιπόν έχουμε κάποιο Event, τότε γίνονται και οι ανάλογες αντιδράσεις όπως εμείς θα τις ορίσουμε, για παράδειγμα αύξηση του score ή αφαίρεση Ζωής κτλ. . Οι αντιδράσεις αυτές καλούνται στο Game Maker, Actions. Actions μπορούμε να δηλώσουμε κατά την εκκίνηση του παιχνιδιού, κατά την είσοδο σε ένα Room και τα λοιπά, όπου όπως είναι κατανοητό, και αυτά θεωρούνται Events.

Η δόμηση ενός παιχνιδιού στην πλατφόρμα μας παρουσιάζεται κάπως έτσι



*Δενδροειδής διαδικασία ανάπτυξης παιχνιδιού*

Για την υλοποίηση και την δημιουργία ενός παιχνιδιού στο Game Maker, πρέπει να ακολουθήσουμε μια συγκεκριμένη σειρά από ενέργειες που πρέπει να κάνουμε. Τα βήματα λοιπόν που ακολουθούμε βασίζονται κυρίως στην εξής σειρά με δυνατότητα κάποιων αλλαγών στη σειρά εκτέλεσής τους, όπου αυτό φυσικά μπορεί να γίνει:

1. Περιγραφή του παιχνιδιού που σχεδιάζουμε να υλοποιήσουμε. Τι είναι αυτό που θέλουμε να δημιουργήσουμε, με τι έχει σχέση και πως το φανταζόμαστε
2. Ορίζουμε τα Sprites του παιχνιδιού μας
3. Ορίζουμε τους ήχους (Sounds) που θα εισάγουμε κατά το «τρέξιμο» του παιχνιδιού
4. Ορίζουμε όλα τα Objects, αλλά όχι ακόμη τα Events και τα Actions που θα λάβουν χώρα στην εκτέλεση του παιχνιδιού
5. Στη συνέχεια λοιπόν καθορίζουμε τα Events και τα Actions του κάθε αντικειμένου
6. Ορίζουμε και δημιουργούμε το δωμάτιο ή τα δωμάτια που θα χρησιμοποιήσουμε για τιςπίστες του παιχνιδιού
7. Τοποθετούμε τα διάφορα αντικείμενα που έχουμε ήδη δημιουργήσει, στα δωμάτια.

Εδώ να τονίσουμε πως ότι το κάθε αντικείμενο μπορεί να εισαχθεί περισσότερες από μία φορές στο παιχνίδι. Προσοχή τα Actions που έχουμε επιλέξει θα εκτελούνται για όλα τα «ίδια» αντικείμενα. Εάν δεν θέλουμε να συμβαίνει κάτι τέτοιο, αυτό που μπορούμε να κάνουμε είναι να

δημιουργήσουμε διαφορετικά αντικείμενα, με διαφορετική ονομασία φυσικά,

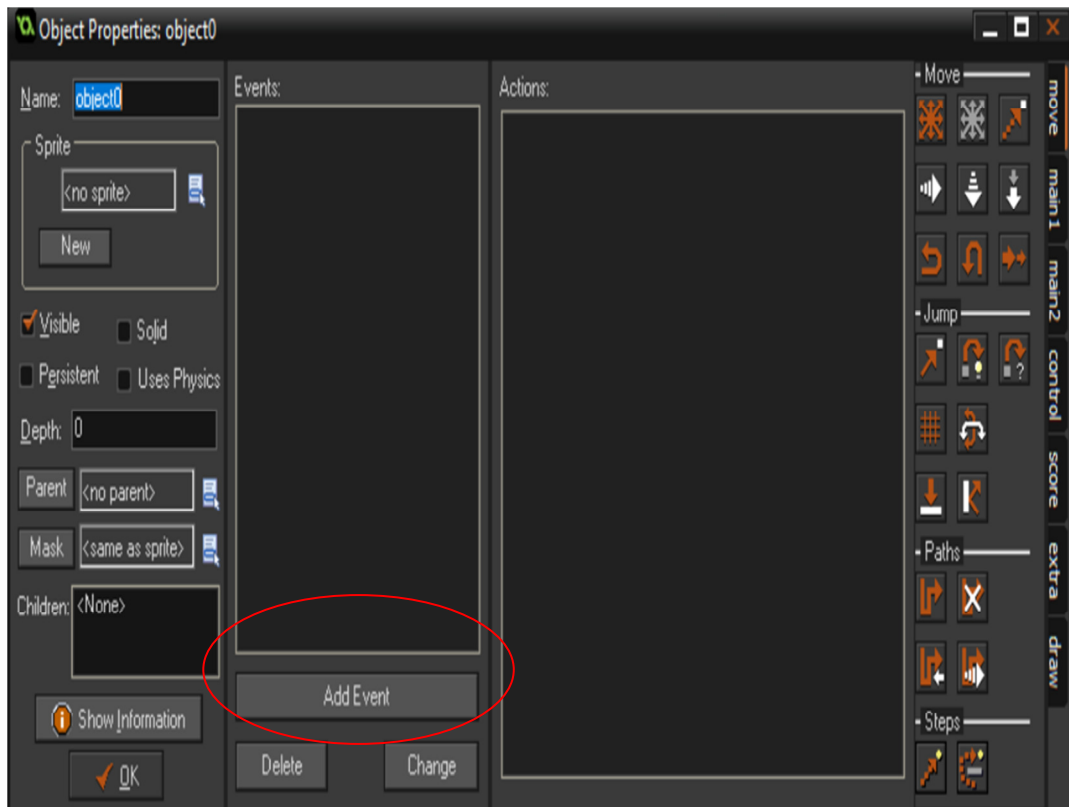
αντιστοιχώντας τα όμως με το ίδιο Sprite, ώστε να έχουν την ίδια μορφή ( για

παράδειγμα, ο ίδιος χαρακτήρας, σε διαφορετικά δωμάτια, να έχει διαφορετικές αλληλεπιδράσεις με παρόμοια αντικείμενα).

Εννοείται πως εάν κατά τη διάρκεια της δημιουργίας μας ή μετά το πέρας της θέλουμε να προσθέσουμε και κάτι άλλο ή να τροποποιήσουμε κάτι, για παράδειγμα ένα αντικείμενο, φυσικά και δεν μας το απαγορεύει κανείς, μπορούμε.

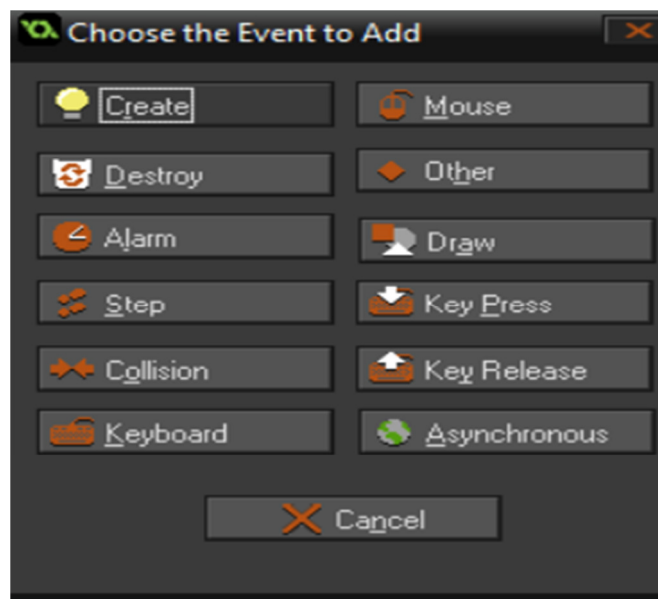
Επομένως ας ξεκινήσουμε με τη δημιουργία των Events. Όταν ανοίξουμε την καρτέλα επιλογών ενός αντικειμένου κάνοντας διπλό κλικ πάνω του, επιλέγουμε το

Add Events .



Προσθήκη Event στο αντικείμενο

Αφού κάνουμε κλικ επάνω στο Add Event θα ανοίξει ένα καινούργιο παράθυρο με τα Events που μπορούμε να προσθέσουμε.



*Επιλογή Event*

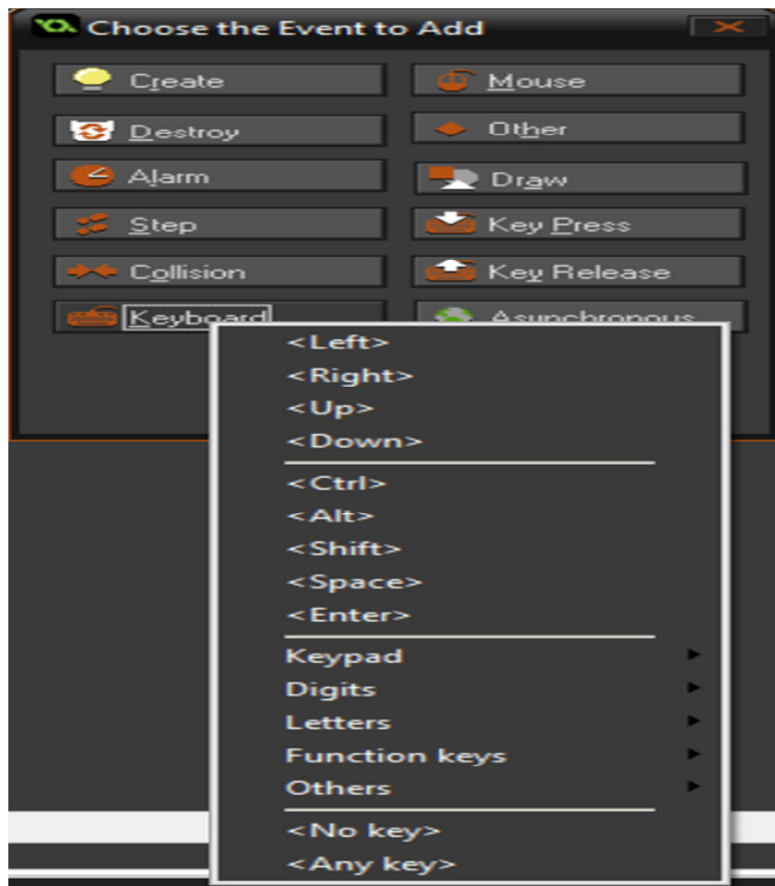
### 3.1 EVENTS

**Create** : Με αυτό το Event μας δίνεται η δυνατότητα να επιλέξουμε κάποια

Actions τα οποία θα ενεργοποιούνται κατά την δημιουργία του αντικειμένου, δηλαδή όταν το παιχνίδι τρέχει για πρώτη φορά ή κατά την είσοδο σε μια καινούργια πιστά και πιο συγκεκριμένα τη χρονική στιγμή που θα δημιουργηθεί και θα εμφανιστεί το αντικείμενο στο παιχνίδι.

**Mouse** : Με αυτό το Event μας δίνεται η δυνατότητα να επιλέξουμε κάποια

Actions, να οποία θα πραγματοποιηθούν ανάλογα με την επιλογή ή τις επιλογές που θα κάνουμε, από τις διάφορες ενέργειες που μπορούν να γίνουν από το ποντίκι ( Mouse ) του υπολογιστή μας, κατά τη διάρκεια του παιχνιδιού. Οι επιλογές αυτές, οι οποίες αποτελούν διαφορετικά μεμονωμένα Events, φαίνονται παρακάτω



*Επιλογή Event Keyboard*

**Κάποιες από τις ενέργειες που θα βρούμε είναι:**

1. **Left button:** Όταν κάνουμε αριστερό κλικ
2. **Right button:** Όταν κάνουμε δεξί κλικ
3. **Middle button:** Όταν κάνουμε κλικ με το μεσαίο κουμπί του ποντικιού
4. **No button:** Όταν δεν πατήσουμε κανένα κουμπί του ποντικιού

5. **Left Pressed:** Όταν πατήσουμε το αριστερό κουμπί, οι ενέργειες που θα γίνουν πριν το αφήσουμε
6. **Right pressed:** Όταν πατήσουμε το δεξί κουμπί, οι ενέργειες που θα γίνουν πριν το αφήσουμε
7. **Middle pressed:** Όταν πατήσουμε το μεσαίο κουμπί, οι ενέργειες που θα γίνουν πριν το αφήσουμε
8. **Left released:** Όταν αφήσουμε το αριστερό κουμπί του ποντικιού, που έχουμε ήδη κάνει κλικ
9. **Right released:** Όταν αφήσουμε το δεξί κουμπί του ποντικιού, που έχουμε ήδη κάνει κλικ
10. **Middle released:** Όταν αφήσουμε το μεσαίο κουμπί του ποντικιού, που έχουμε ήδη κάνει κλικ
11. **Mouse enter:** Όταν το ποντίκι μας περάσει και βρίσκεται πάνω από το αντικείμενό μας
12. **Mouse leave:** Όταν το ποντίκι φύγει από πάνω από το αντικείμενό μας
13. **Mouse wheel up:** Όταν κάνουμε scroll την ρόδα του ποντικιού μας προς τα πάνω
14. **Mouse wheel down:** Όταν κάνουμε scroll την ρόδα του ποντικιού μας προς τα κάτω

**Destroy:** Εδώ επιλέγουμε τις ενέργειες που θα γίνουν, όταν καταστραφεί το αντικείμενό μας κατά την διάρκεια του παιχνιδιού. Η καταστροφή ενός αντικειμένου αποτελεί ένα από τα προκατασκευασμένα Actions που μας δίνει η εφαρμογή, το οποίο και θα συναντήσουμε στην συνέχεια

**Alarm:** Με αυτό το Event, ορίζουμε ενέργειες που θέλουμε να γίνουν (Actions), κάποια συγκεκριμένη χρονική στιγμή του παιχνιδιού. Για παράδειγμα, μπορούμε να ορίσουμε ότι σε πέντε λεπτά (5') αφού ξεκινήσει το παιχνίδι άλλαξε το background από μέρα σε νύχτα. Επίσης, στον προγραμματισμό έχουμε μάθει να χρησιμοποιούμε κάποιες μεταβλητές οι οποίες ανάλογα με κάποιες ενέργειες που γίνονται, αλλάζουν τιμή. Έτσι, με το Alarm Event, ορίζουμε και τα Actions που θα τρέξουν όταν μια μεταβλητή

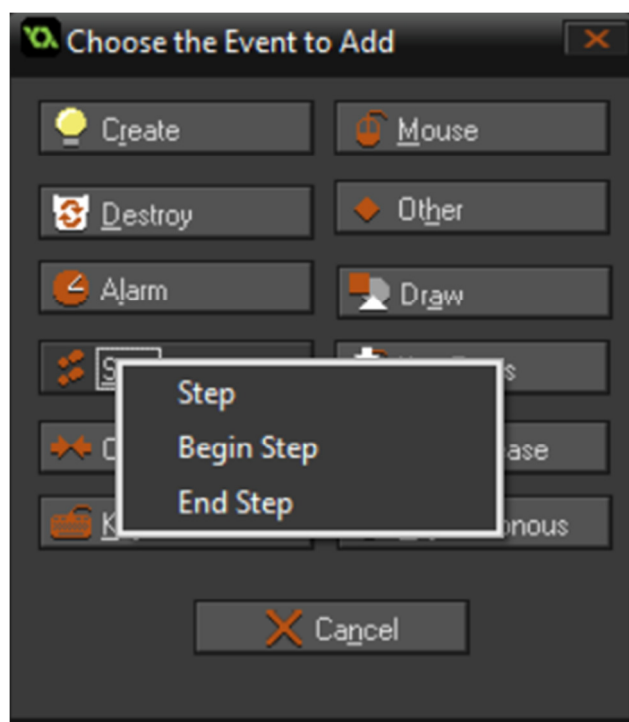


πάρει κάποια συγκεκριμένη τιμή. Για παράδειγμα όταν η μεταβλητή που έχουμε ορίσει για τις ζωές του χαρακτήρα μας πάρει την τιμή 0, τότε να εμφανίζεται το Game Over και να τελειώνει το παιχνίδι εκεί.

**Draw:** Στο παιχνίδι, μπορεί κάποια στιγμή να ορίσουμε μια ενέργεια η οποία

θα εμφανίζει κάτι καινούριο στο δωμάτιό μας, για παράδειγμα μια εικόνα, ή θα αλλάζει στο σκορ, τις ζωές π[υ μας έχουν απομείνει και τα λοιπά. Έτσι με αυτό το Event, αφού γίνει κάτι από τα παραπάνω που είπαμε, τότε ζητάμε την εκτέλεση κάποιων Actions.

**Step:** Σε πολλά παιχνίδια, όλοι μας έχουμε συναντήσει κάποιον παίχτη, συνήθως κάποιο ανθρωπάκι, το οποίο μπορούμε να ελέγξουμε εμείς, συνήθως με το πληκτρολόγιο. Επομένως κινούμε τον χαρακτήρα αυτόν, με κάποια βήματα, δεξιά αριστερά και τα λοιπά. Μπορούμε λοιπόν να ορίσουμε κάποια Actions, τα οποία θα συμβούν όταν κινούμε αυτό το Object, όταν ξεκινάει την κίνησή του ή όταν την τελειώνει.

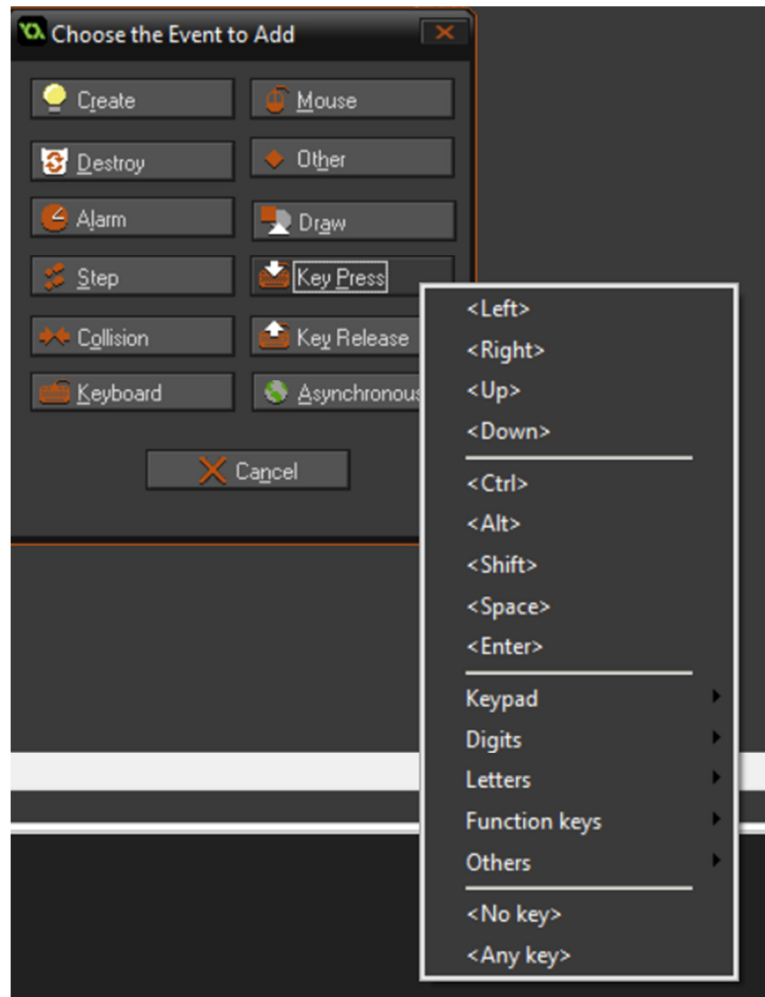


Επιλογή Event Step

**Key Press :** Φυσικά, σχεδόν σε όλα τα παιχνίδια, υπάρχουν Actions τα οποία

πραγματοποιούνται όταν πατήσουμε κάποιο ή κάποια πλήκτρα στο πληκτρολόγιο.

- Δηλαδή, μπορεί πατώντας κάποιο κουμπί μπορεί να κινείτε ο χαρακτήρας στην οθόνη , αλλά μπορεί φυσικά να γίνονται και άλλου είδους ενέργειες, όπως παύση του παιχνιδιού η κλείσιμο. Swamy Nanu, Swamy Naveena (2006) Basic Game Design & Creation for Fun & Learning



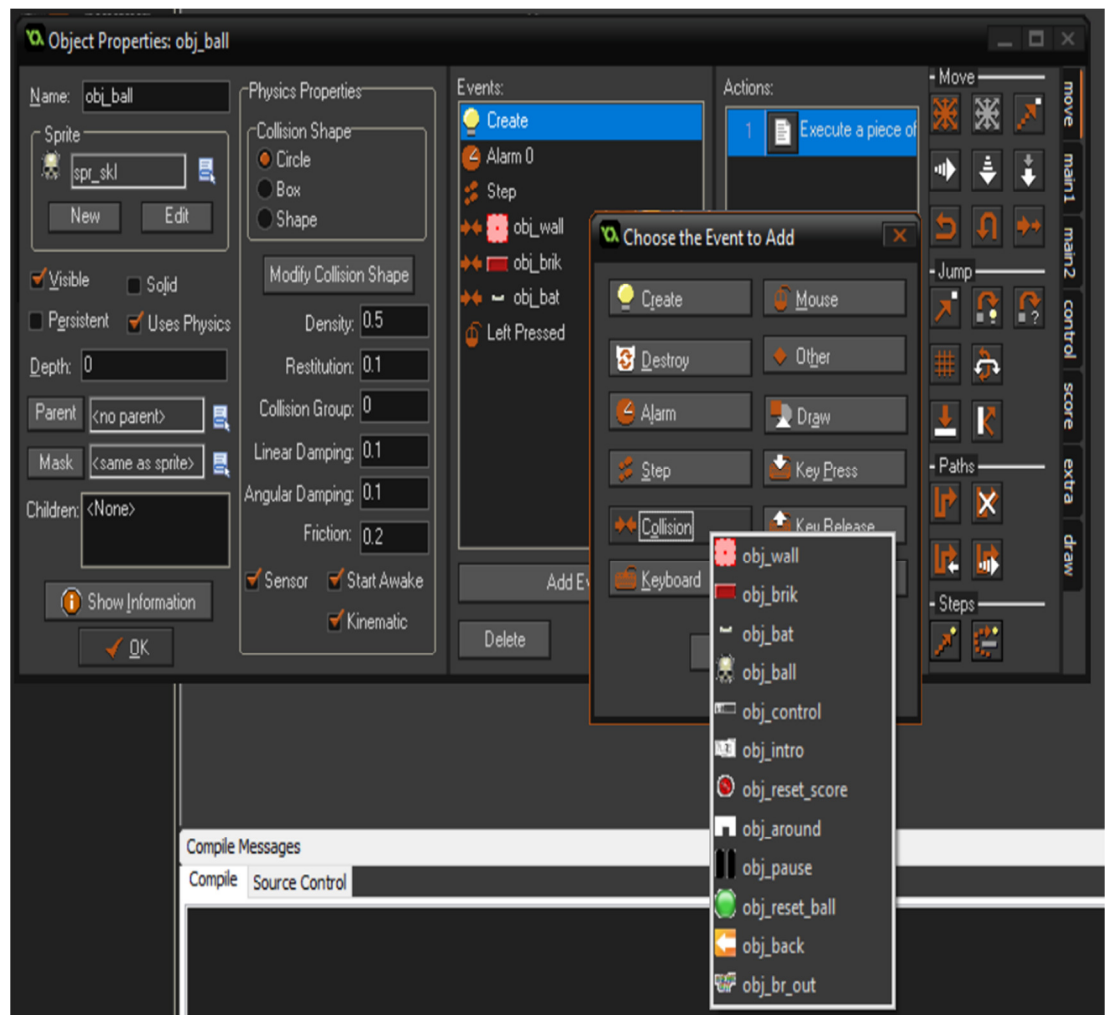
Επιλογή Event Key Press

Όπως είναι φυσικό μας δίνεται η δυνατότητα να επιλέξουμε το πλήκτρο που θέλουμε να γίνουν οι ανάλογες ενέργειες με το που το πατήσουμε, από όλα τα διαθέσιμα πλήκτρα του πληκτρολογίου μας. Από τα γράμματα, τους αριθμούς και τα λοιπά, όπως βλέπουμε και στην εικόνα. Η πιο κλασική περίπτωση χρήσης αυτού του Event, είναι η χρήση των Arrow Keys , για την καθοδήγηση κάποιου χαρακτήρα ή γενικά κάποιου αντικειμένου.

Michael Rohde (2014) Game Maker: Studio for Dummies

**Collision:** Με αυτό το Event, δημιουργούμε Actions, στην περίπτωση που το

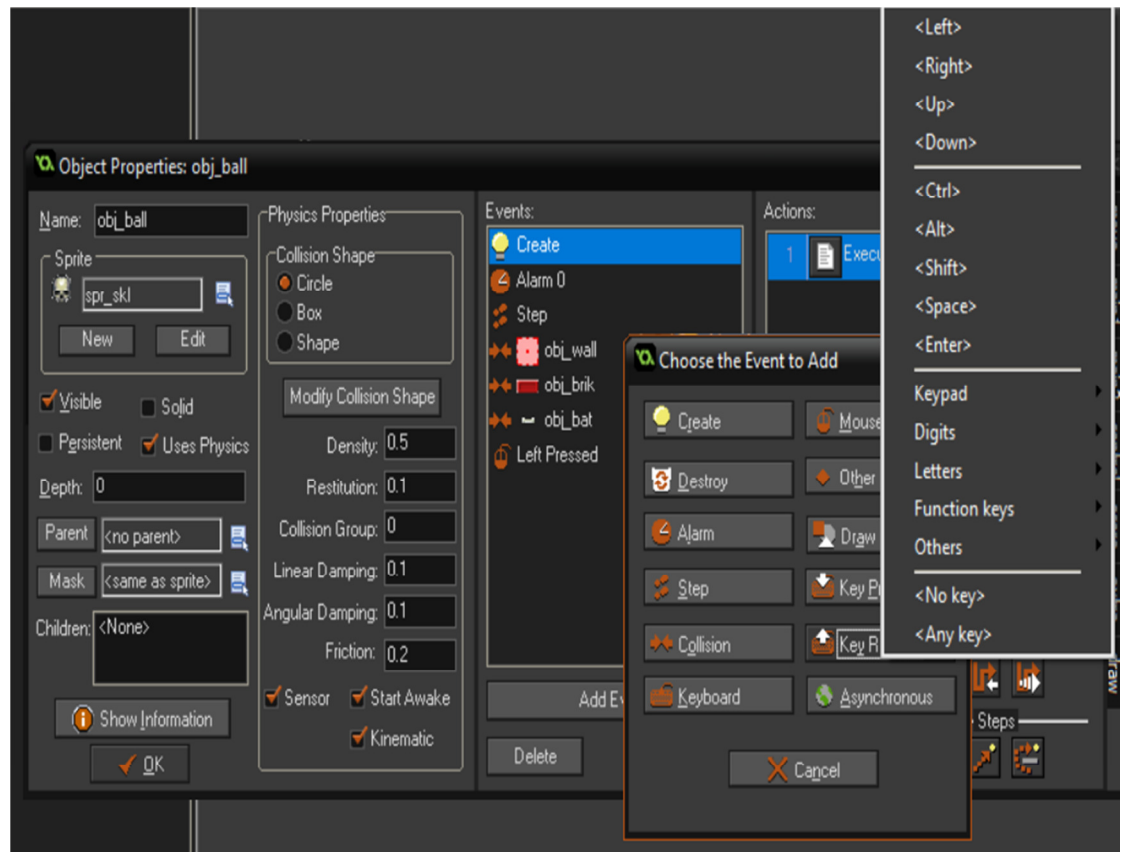
αντικείμενο ακουμπήσει με κάποιο άλλο αντικείμενο του παιχνιδιού που βρίσκεται και αυτό μέσα στο δωμάτιο. Κλασικό παράδειγμα είναι όταν ο κύριος χαρακτήρας του παιχνιδιού έρθει σε επαφή με κάποιο συγκεκριμένο αντικείμενο, χάνει μία ζωή. Το πιο βασικό παράδειγμα που μπορούμε να δώσουμε είναι αυτό που όταν ο χαρακτήρας μας έρθει σε επαφή με κάποιον τοίχο τότε ο χαρακτήρας δεν μπορεί να προχωρήσει κι άλλο προς αυτήν την κατεύθυνση. Με το πατήσουμε το κουμπί «Collision» από το μενού των Events, μας ανοίγει μια λίστα με όλα τα αντικείμενα που έχουμε ήδη δημιουργήσει, ώστε να επιλέξουμε αυτό που θέλουμε. Michael Rohde (2014) Game Maker: Studio for Dummies



Επιλογή Event για αντικείμενο

Εδώ βλέπουμε ένα παράδειγμα με τη λίστα που ανοίγει, στο Brake Out.

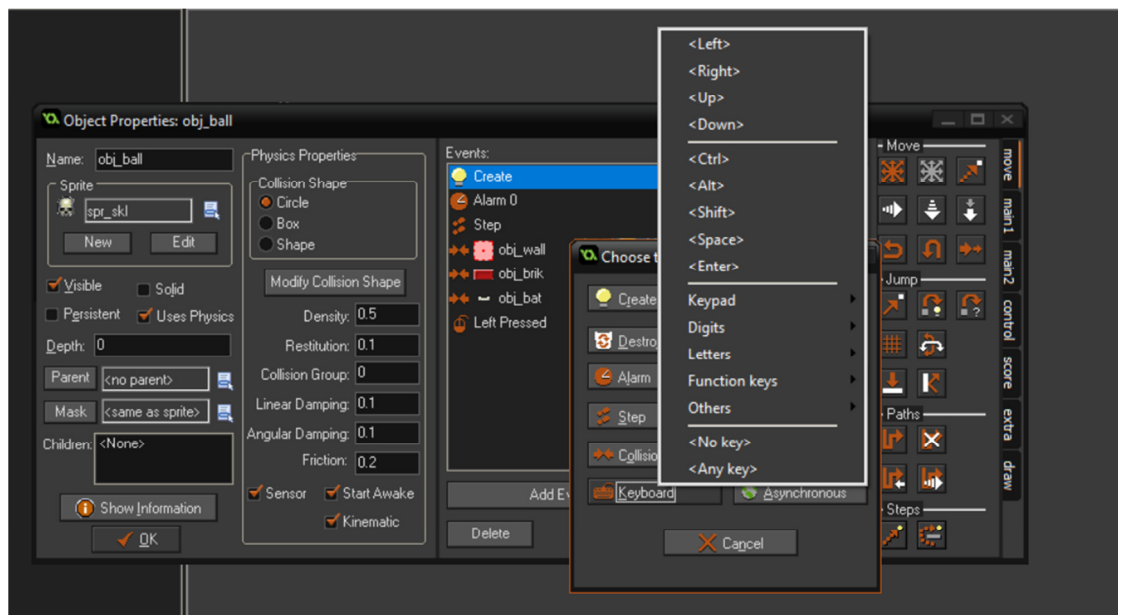
**Key Release** : Με αυτό το Event, δηλώνουμε τα Actions που θέλουμε να συμβούν όταν, αφού έχουμε πατήσει κάποιο κουμπί του πληκτρολογίου ή το έχουμε πατημένο και το ελευθερώνουμε . Για παράδειγμα, μπορεί να θέλουμε ο χαρακτήρα να κάνει κάποιο άλμα την στιγμή που θα αφήσουμε το κουμπί. Αυτό θα το πραγματοποιήσουμε ορίζοντας αυτό το Event.



Επιλογή Event για αντικείμενο 1

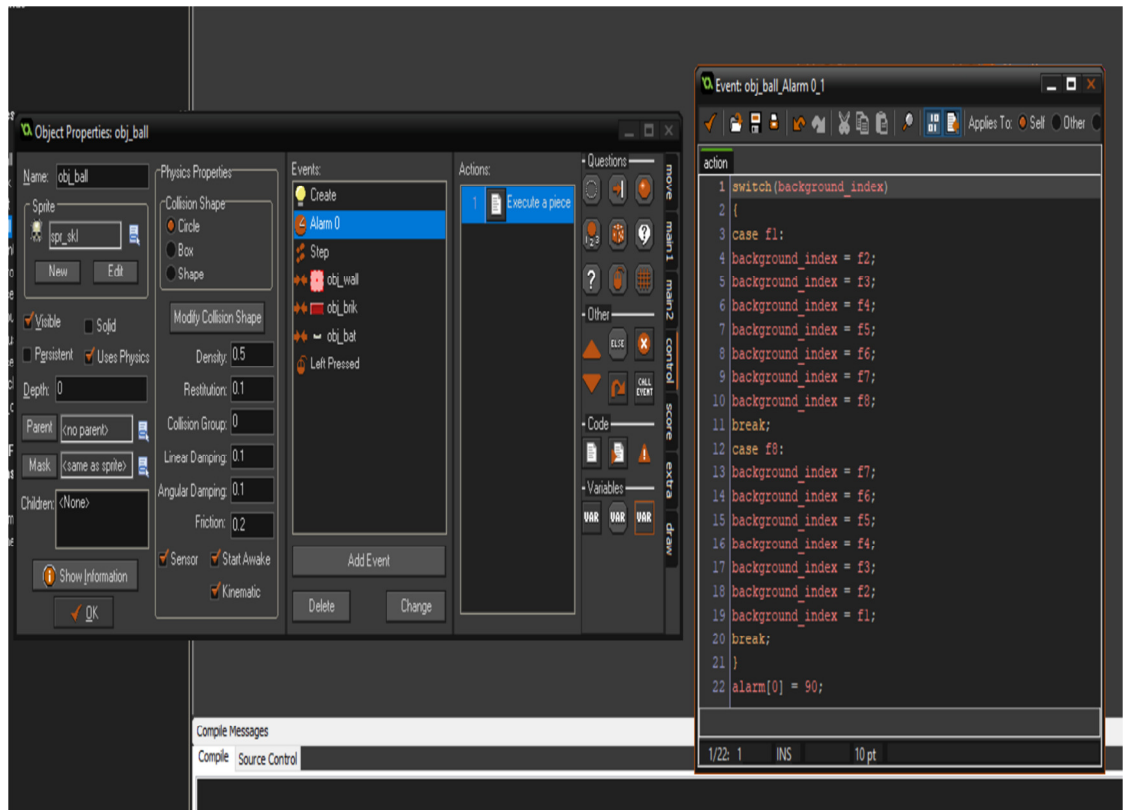
Όπως και με το Event «Key Press», έτσι κι εδώ υπάρχει το ανάλογο μενού.

**Keyboard** : Με αυτό το Event, επιλέγουμε τα Actions που θα ενεργοποιηθούν όταν πατήσουμε κάποιο κουμπί του πληκτρολογίου, γενικά. Η διαφορά του με το Key Press Event, είναι ότι στο δεύτερο, ενεργοποιούνται κάποιες ενέργειες, όταν απλά πατήσουμε μία φορά κάποιο πλήκτρο, και μόνο για τη στιγμή που το πατάμε. Όπως είναι λογικό, η λίστα με τις επιλογές που μας δίνει είναι η ίδια ακριβώς, όπως φαίνεται παρακάτω.



*Επιλογή Event για αντικείμενο 2*

Εδώ έφτασε η ώρα να μιλήσουμε και λίγο για τον κώδικα. Όπως έχω αναφέρει πολλές φορές παραπάνω το Game Maker Studio έχει την δική του γλώσσα την Game Maker LANGUAGE και αναπόφευκτά θα χρηστεί κάποιά στιγμή στο project. Το Game Maker μας δίνει την δυνατότητα σε κάθε EVENT να γράψουμε το δικό μας κώδικα με την χρήση ενός πολύ απλού Editor.



#### Εισαγωγή Κώδικα σε Event

Other : Αυτό το Event, μας δίνει μια λίστα από διάφορες επιλογές.

1. **Outside room** : Εδώ δίνουμε Actions για το αντικείμενό μας, όταν αυτό βρίσκεται εκτός του δωματίου
2. **Intersect boundary** : Εδώ δίνουμε Actions όταν το αντικείμενό μας φτάσει στο τέλος του δωματίου.
3. **Views** : Σε περίπτωση που στο παιχνίδι μας χρησιμοποιούμε κάποιες οπτικές γωνίες δηλαδή μπορούμε να ορίσουμε κάποια Actions, που θα πραγματοποιηθούν όταν το αντικείμενο βγει εκτός κάποιου View
4. **Game start** : Εδώ ορίζουμε κάποια Actions που θα συμβούν μόλις το παιχνίδι ξεκινήσει.
5. **Game end** : Εδώ ορίζουμε τα Actions που θα συμβούν μόλις το παιχνίδι μας φτάσει στο τέλος του.

6. **Room start** : Εδώ ορίζουμε τα Actions που θα θέσουμε, όταν ξεκινάει κάποια συγκεκριμένη πίστα, κάποιο Room, και όχι όταν ξεκινάει γενικά το παιχνίδι.

7. **Room end** : Και εδώ, αυτά τα Actions που θέλουμε να ορίσουμε όταν τελειώνει κάποιο συγκεκριμένο Room.

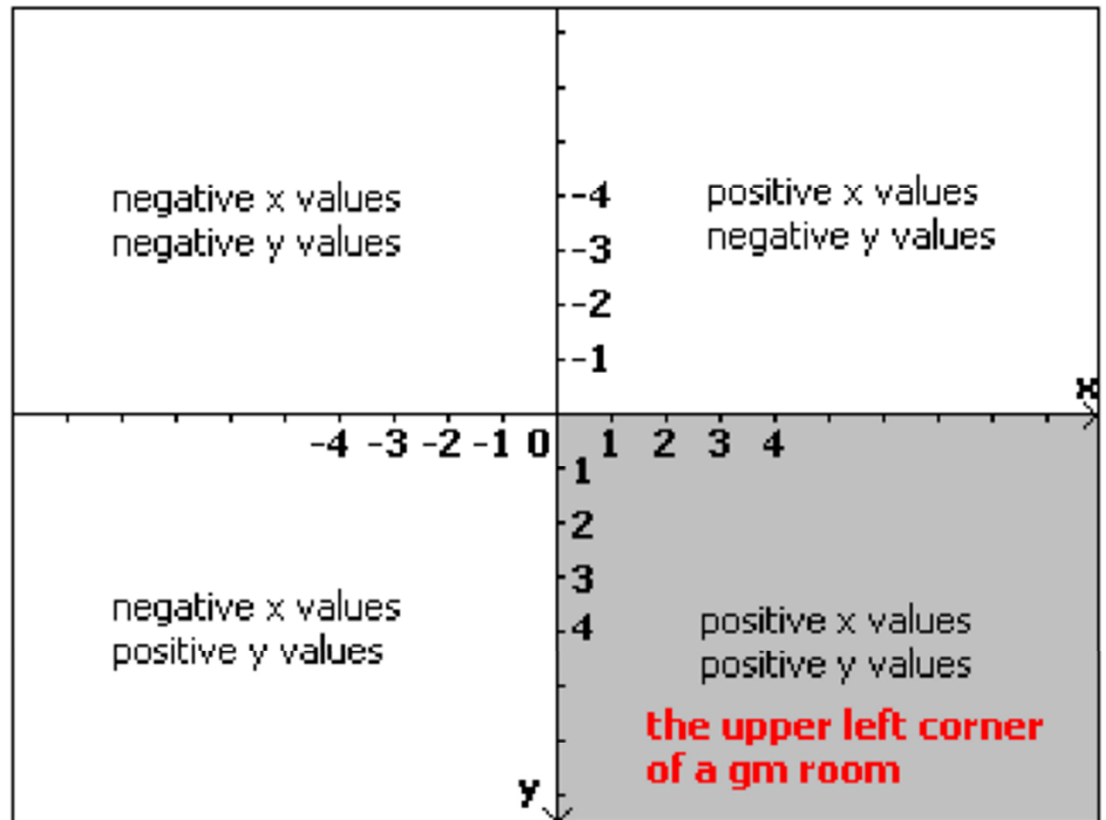
8. **No more lives** : όταν τελειώσουν οι ζωές που έχουμε ορίσει τη ενέργεια θα γίνετε όπως να εμφανίζει GAME OVER.

Swamy Nanu, Swamy Naveena (2006) Basic Game Design & Creation for Fun & Learning

### 3.1.2 ACTIONS

Αφού έχουμε επιλέξει το αντικείμενό επιλέξαμε και το Event που θέλουμε, προχωράμε επόμενο βήμα, που είναι η εισαγωγή των Actions. Τα Actions, είναι χωρισμένα σε κατηγορίες για λόγους ευκολίας του χρήστη. Οι κατηγορίες αυτές ονομαστικά, είναι: Move , main1, main2, control, score, extra και draw αρχικά θα δούμε έναν πίνακα ο οποίος προσδιορίζει ξεκαθαρίζει τις θέσεις που ορίζουμε να παίρνει το αντικείμενό μας, όταν χρησιμοποιούμε κυρίως τα Move Actions, τα οποία βασίζονται στον γνωστό σε όλους μας άξονα X και Y.





#### Actions

Παρακάτω θα εξηγήσουμε ένα προς ένα όλα τα Actions που το Game Maker Studio δίνει.

#### 3.1.2.1 MOVE ACTIONS

Move Fixed : Με το Action αυτό καθορίζουμε την κίνηση ενός αντικειμένου

προς μία συγκεκριμένη κατεύθυνση

Συνήθως το ορίζουμε σε κάποιο από βέλη του πληκτρολογίου. Ακόμη ορίζουμε την ταχύτητα με την οποία θέλουμε να κινείται το αντικείμενό και η ταχύτητα η οποία θα ορίσουμε, είναι σε Pixels.

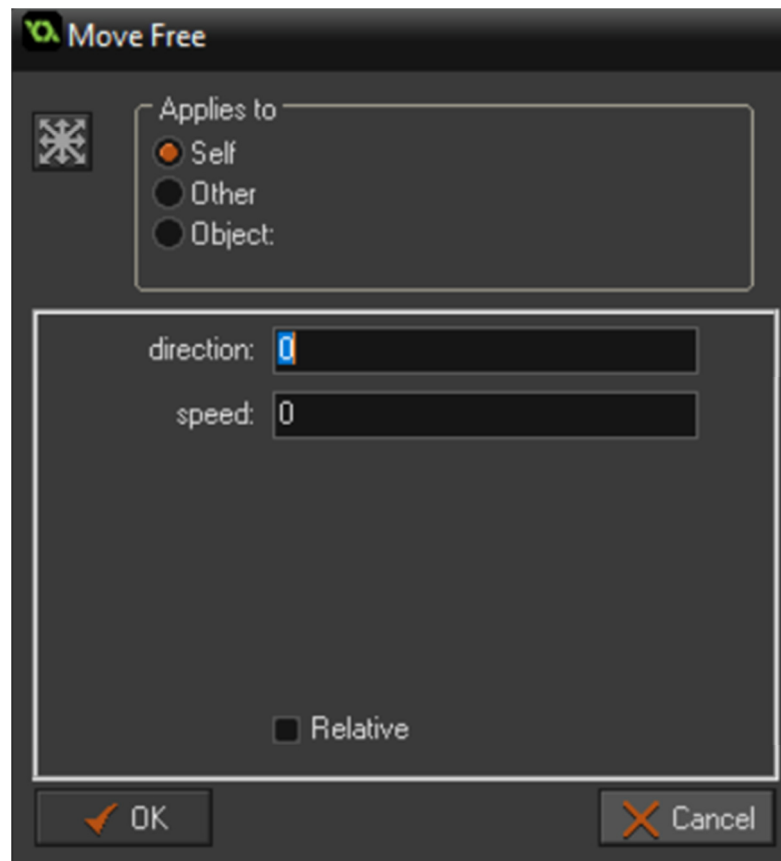


*Move Actions*

Σε πολλά Actions θα παρατηρήσετε ένα Check Box, με την ονομασία Relative. Εάν το τσεκάρουμε, τότε η αριθμητική τιμή που δίνουμε παραπάνω, θα προστίθεται κάθε φορά στην προηγούμενη.

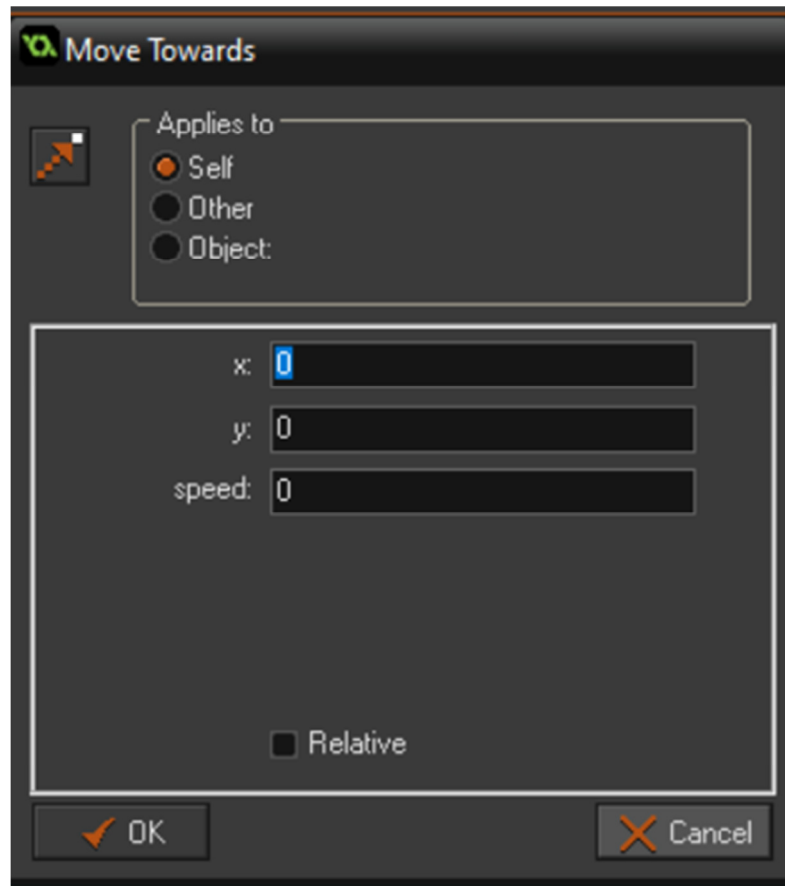
Move Free : Με αυτό το Action, δίνουμε κατεύθυνση στο αντικείμενό μας, με την βοήθεια των μοιρών. Στην ουσία, του δίνουμε μια γωνία, μεταξύ 0

και 360 μοιρών. Εάν δώσουμε 0, τότε κατευθύνεται ευθύγραμμα προς τα δεξιά. Οι 90 μοίρες, του δίνουν κάθετη ανοδική πορεία. Εάν θέλουμε να πάρει μια τυχαία πορεία, τότε πληκτρολογούμε στο πλαίσιο : random (360) . Αυτή η εντολή, επιλέγει έναν τυχαίο αριθμό, μικρότερο ή ίσο του αριθμού που δίνουμε, που σε αυτήν την περίπτωση είναι το 360.



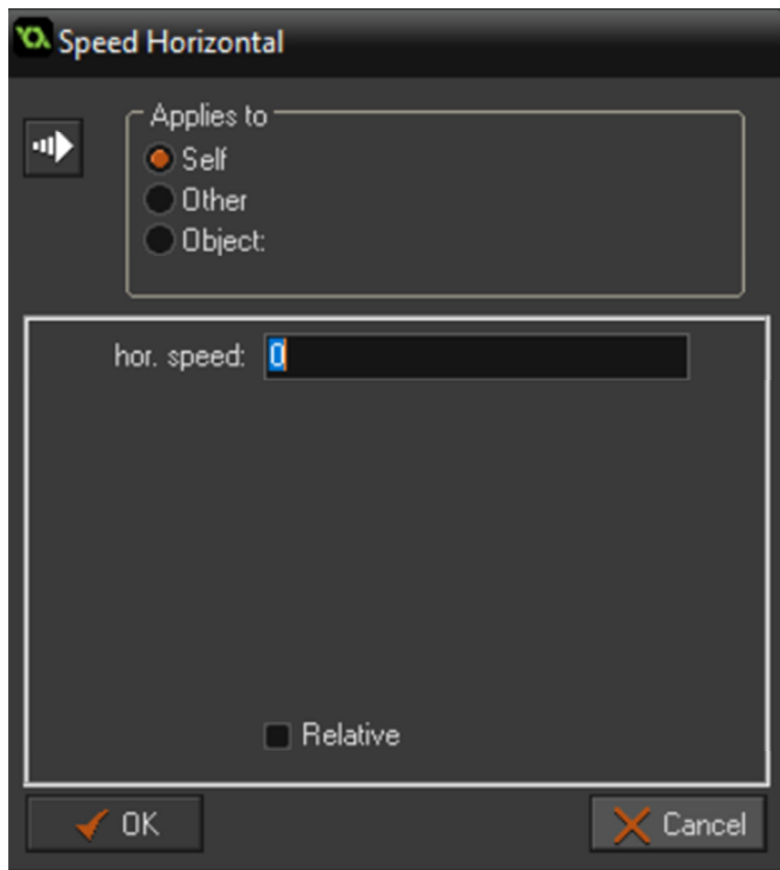
*Move Free Action*

**Move Towards** : Εδώ ορίσουμε μια συγκεκριμένη θέση του δωματίου προς την οποία θέλουμε να κατευθυνθεί το αντικείμενό μας.



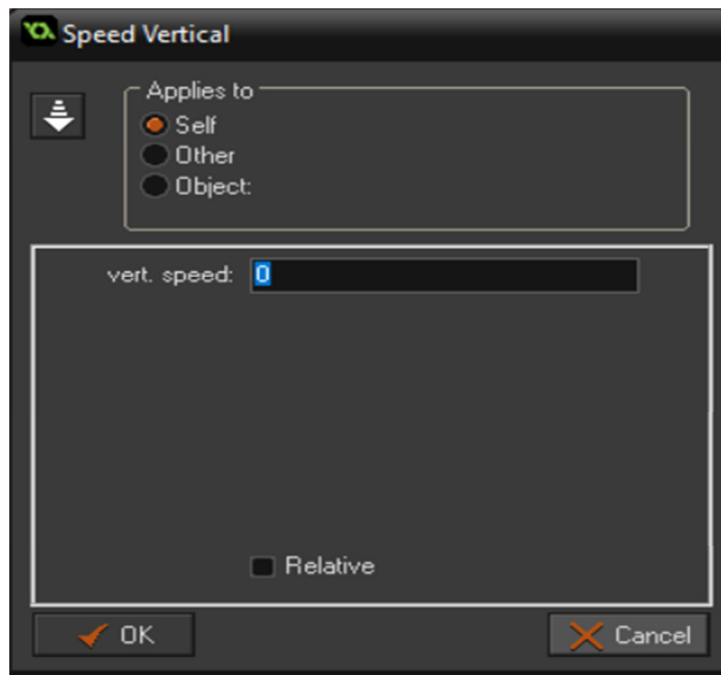
*Move Towards Actions*

Speed Horizontal : Με αυτό το Action, μας δίνεται η δυνατότητα να αλλάξουμε την οριζόντια ταχύτητα του αντικειμένου. Εάν αφήσουμε την τιμή στο 0, τότε η ταχύτητα δεν αλλάζει. Εάν βάλουμε θετικό αριθμό, τότε το αντικείμενο παίρνει ταχύτητα προς τα δεξιά. Εάν βάλουμε αρνητικό, τότε, παίρνει ταχύτητα τα αριστερά.



*Move Speed Horizontal Action*

Speed Vertical : Το ίδιο ακριβώς με το Speed Horizontal, μόνο που εδώ έχουμε να κάνουμε με την κάθετη ταχύτητα.



*Move Speed Vertical Action*

Set Gravity : Με αυτό το Action καθορίζουμε το πώς ένα αντικείμενο επηρεάζεται από την βαρύτητα, όπως πχ ένα χαρακτήρας που έκανε κάποιο άλμα και τώρα θα επιτρέψει στο έδαφος με κάποια ταχύτητα.



*Move Set Gravity Action*

Reverse Horizontal : Με αυτό το Action, αντιστρέφουμε την οριζόντια κίνηση του αντικειμένου μας.



*Move Reverse Horizontal Action*

Υπάρχουν κάποιοι παράμετροι να ορίσουμε πέρα από το για ποιο αντικείμενο θέλουμε να ενεργοποιηθεί το Aston.

Η ταχύτητα είναι αντιστρόφως ανάλογη με αυτήν που είχαμε ήδη θέσει με το ανάλογο Aston.

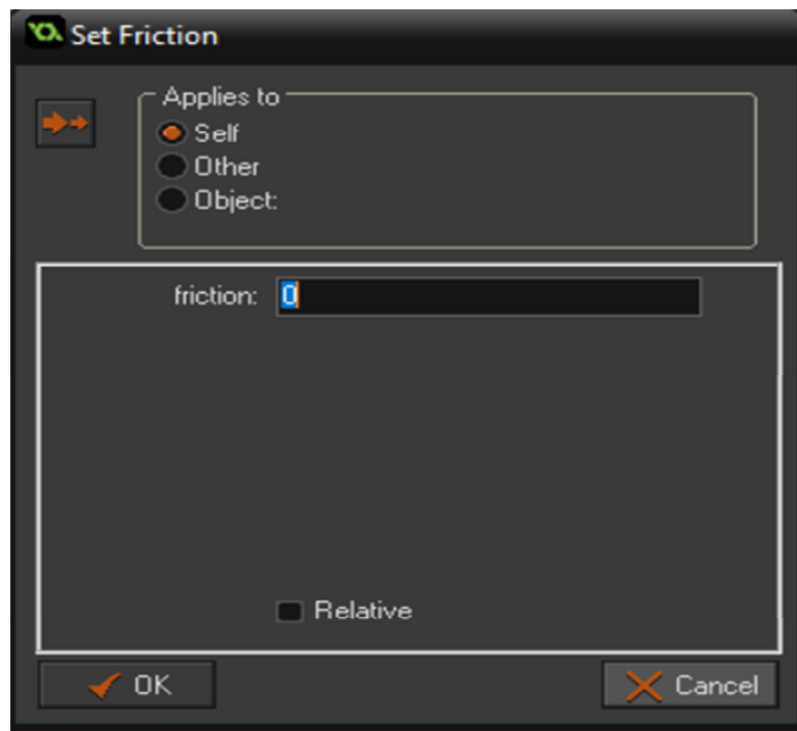


Reverse Vertical : Το ίδιο ακριβώς όπως και με το Reverse Horizontal, μόνο που σε αυτήν την περίπτωση αλλάζει κατεύθυνση στον κάθετο άξονα.



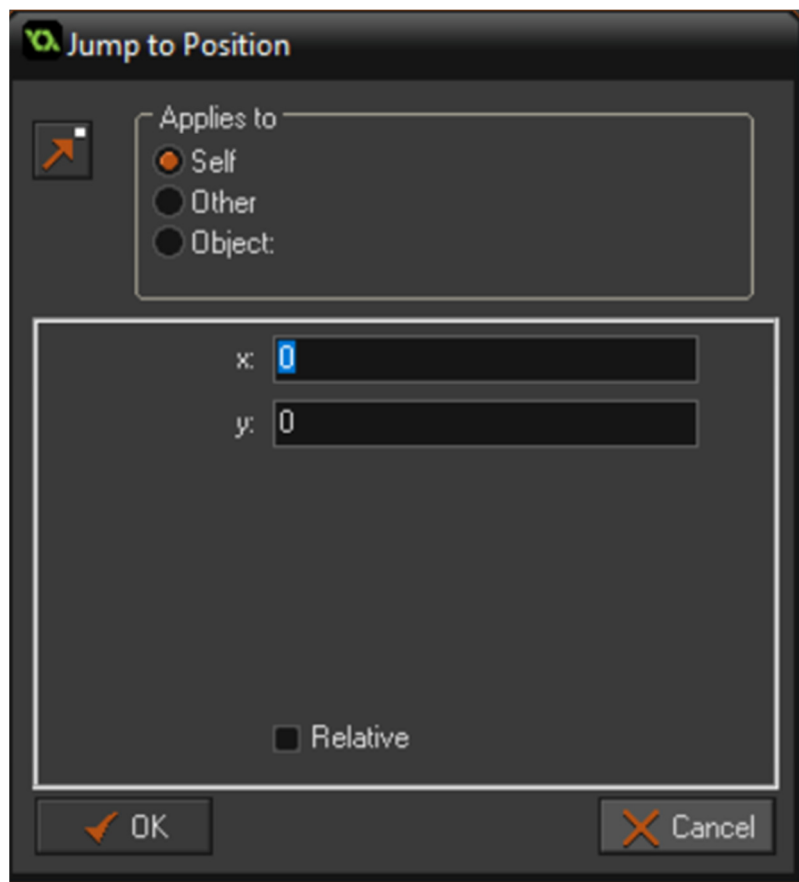
*Move Reverse Vertical*

Set Friction : Εδώ έχουμε ένα Action που έχει να κάνει με την τριβή του αντικειμένου. Εάν το αντικείμενο μας έχει μια κίνηση με κάποια τιμή  $X$  και δώσουμε μια τιμή στο Friction  $Y$  τότε το  $X$  μειώνεται συνεχώς κατά  $Y$  μέχρι η τιμή του να γίνει 0. Που θα έχει σαν αποτέλεσμα το αντικείμενό μας να σταματήσει. Εάν το  $X$  είναι αρνητικός αριθμός αυτό προστίθεται κάθε φορά στο  $Y$  μέχρι ξανά όπως πριν να γίνει 0.



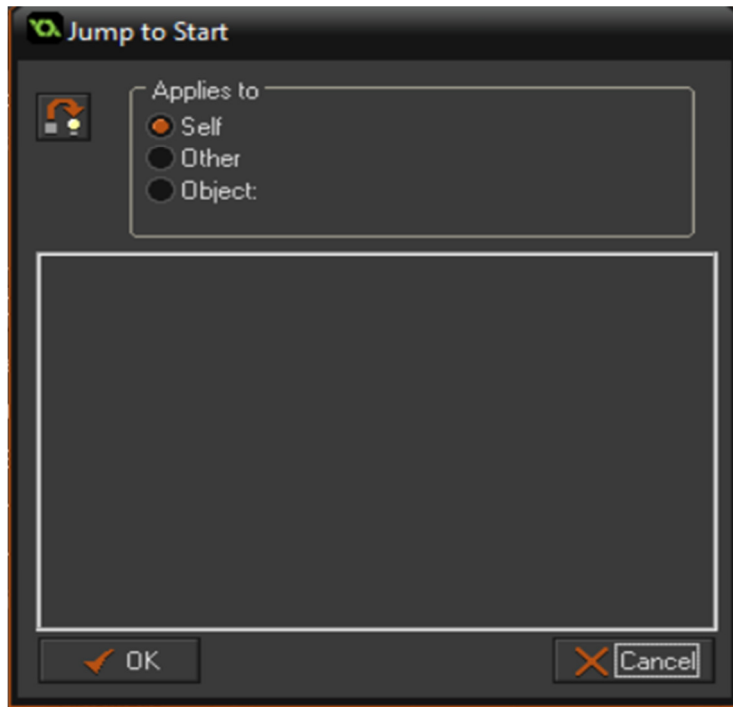
*Move Set Friction Action*

Jump to Position : Με αυτό το Action ορίζουμε ένα οποιοδήποτε σημείο στο δωμάτιο στο οποίο θέλουμε να πάει το αντικείμενό μας με τη χρήση συντεταγμένων x και y.



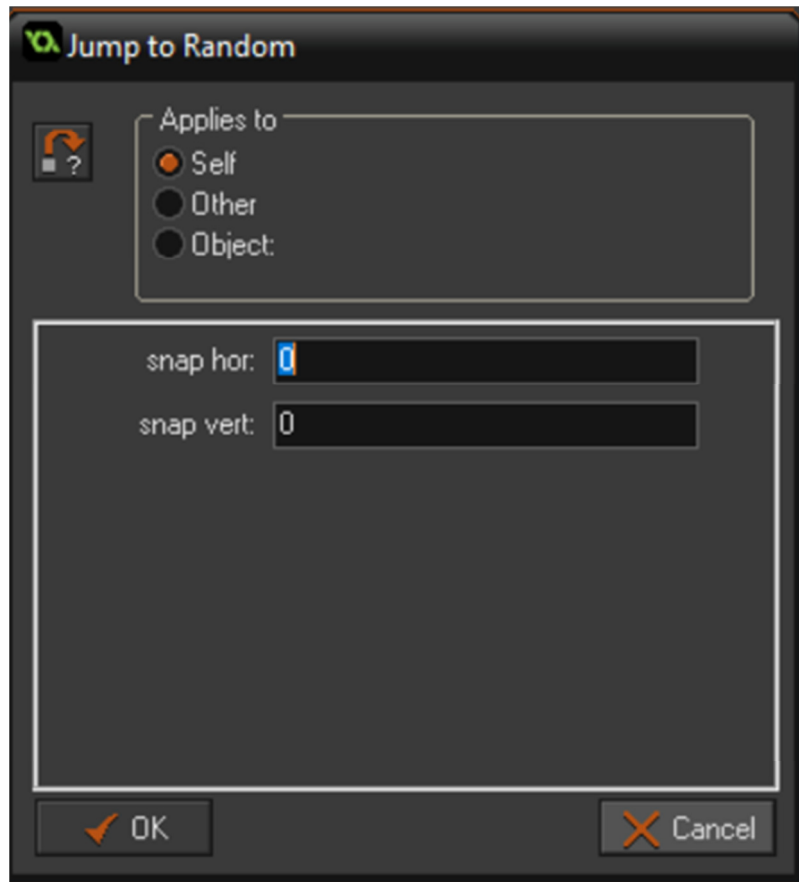
*Move Jump to Position Action*

Jump To Start : Αυτό το Action κάνει ότι κάνει και το Jump to Position αλλά εδώ δεν χρειάζεται να ορίσουμε x και y, διότι το αντικείμενο γυρίζει μόνο του στην θέση του δωματίου στην οποία τοποθετήθηκε αρχικά.



*Move Jump to Start Action*

Jump To Random: Με αυτό το Action γίνεται περίπου ότι και στα προηγούμενα 2 αλλά με την διαφορά του ότι το αντικείμενο δεν θα μεταφερθεί σε κάποιο σημείο του δωματίου που υπάρχει και κάποιο άλλο αντικείμενο του δωματίου που μπορεί να αλληλεπιδράσεις.



*Move Jump to Random Action*

Align To Grid : Με αυτό το Action ορίζουμε ένα πλέγμα με το μέγεθος κελιών να το προσδιορίζουμε εμείς όπου μέσα στο οποίο θα βρίσκεται το αντικείμενό μας .



*Move Align to Grid Action*

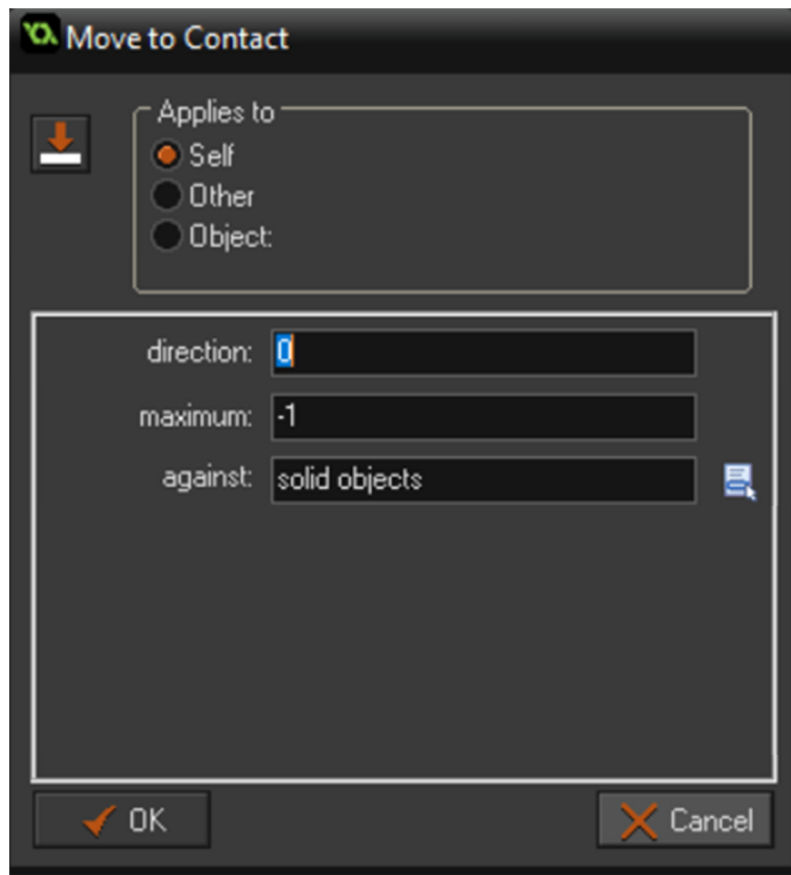
Wrap to Screen: Με αυτό το action το αντικείμενο θα βγει από την μία πλευρά και θα εμφανιστεί στην ακριβώς αντίθετη. Χαρακτηριστικό παράδειγμα είναι το παιχνίδι Pac-Man.



*Move Wrap Screen Action*

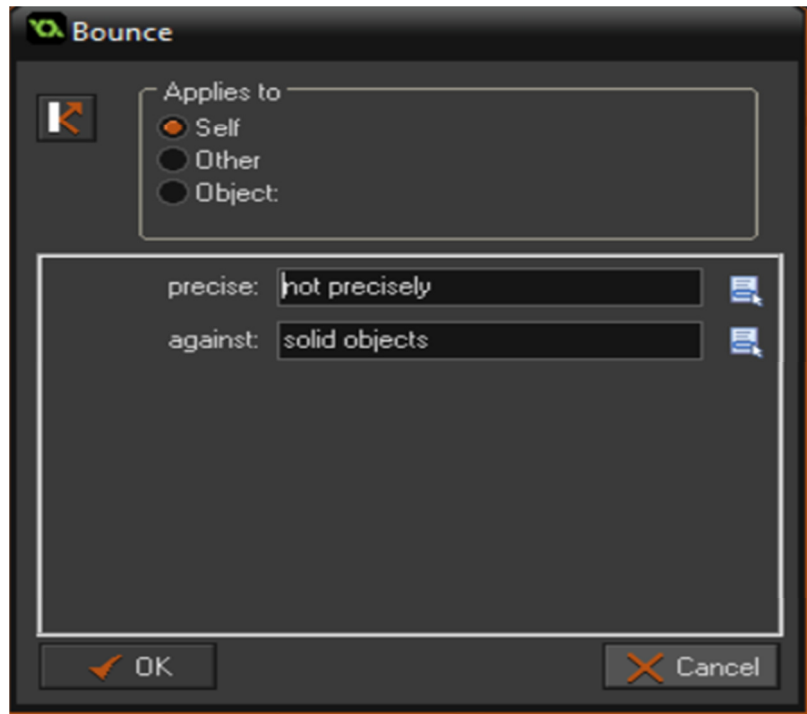
Move to Contact: Με αυτό το Action λέμε σε ένα αντικείμενο να κάνει  
μια  
Απόσταση μέχρι να συναντήσει ένα αντικείμενο του δωματίου.





*Move, Move to Contact Action*

Bounce : Αυτό το Action το συνδέουμε με ένα Collision Event με ένα άλλο αντικείμενο. Όταν συναντήσει λοιπόν αυτό το άλλο αντικείμενο θα υπάρξει αναπήδηση.

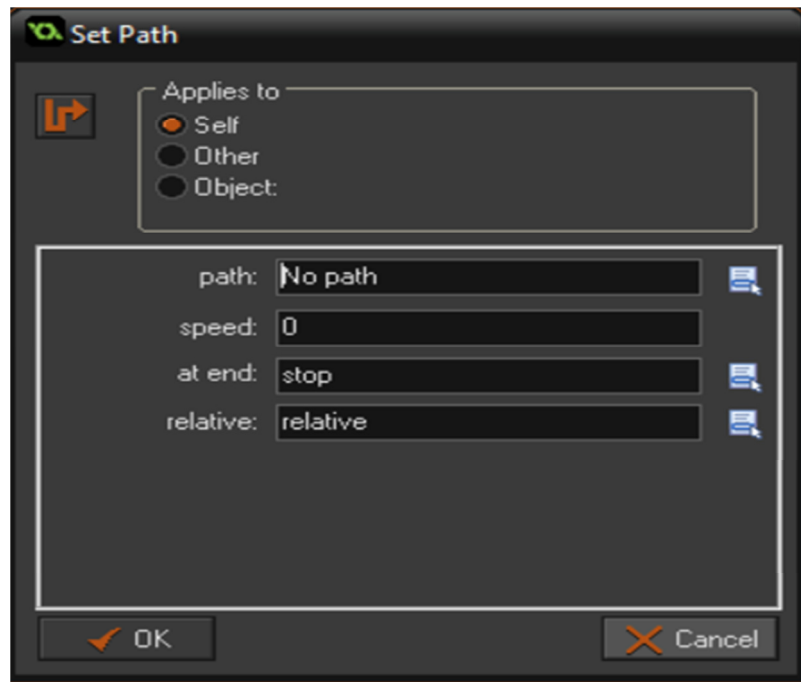


*Move Bounce Action*

Paths : Υπάρχουν υπάρχουν τέσσερα Actions που αφορούν τα Paths (μονοπάτια).

1. Την δημιουργία τους,
2. Το τέλος τους
3. Την θέση τους
4. Την ταχύτητά τους

Εδώ να σημειωθεί ότι σπάνια είναι χρήσιμα αλλά καλό είναι να τα ξέρουμε.

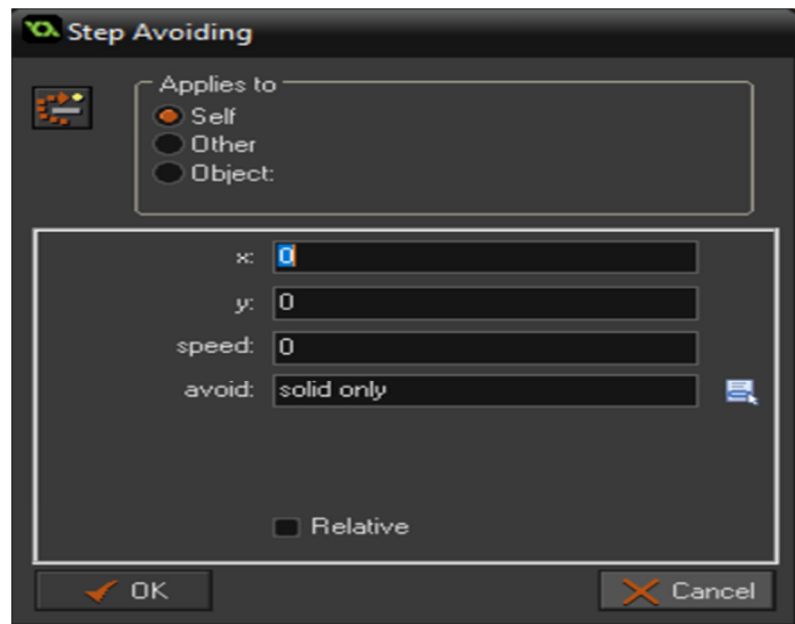


*Move Set Path Action*

Steps : Τέλος υπάρχουν ακόμη δύο Actions που αφορούν τα Steps (βήματα).

1. Το πρώτο αφορά την δημιουργία κάποιου βήματος που θα εκτελεστεί από το αντικείμενό προς μία κατεύθυνση
2. Το δεύτερο μας βοηθάει στην περίπτωση που το αντικείμενό συναντήσει κάποιο άλλο συγκεκριμένο αντικείμενο για να το αποφύγει και να συνεχίσει την πορεία του.

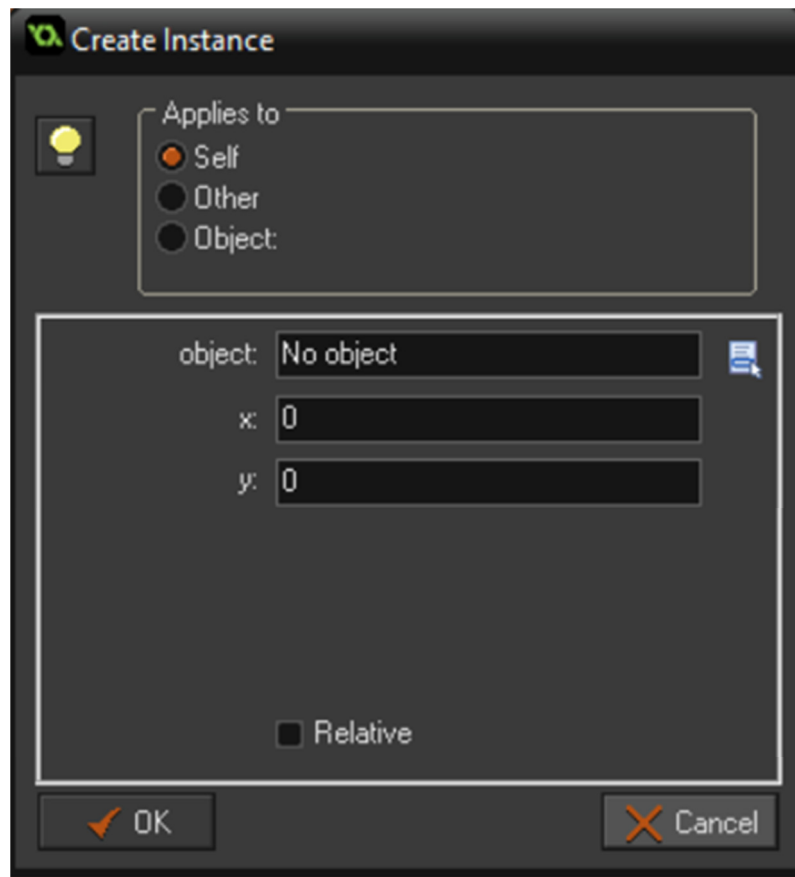
Swamy Nanu, Swamy Naveena (2006) Basic Game Design & Creation for Fun & Learning



*Move Step Avoiding Action*

### **3.1.2.2 MAIN 1 ACTIONS**

Create Instance : Με αυτό το Action μπορούμε να δημιουργήσουμε και να εμφανίσουμε ένα αντικείμενο μέσα στο δωμάτιο. Πρέπει να επιλέξουμε αντικείμενο φυσικά και της συντεταγμένες χ και y οπού το αντικείμενό μας θα εμφανίζεται .



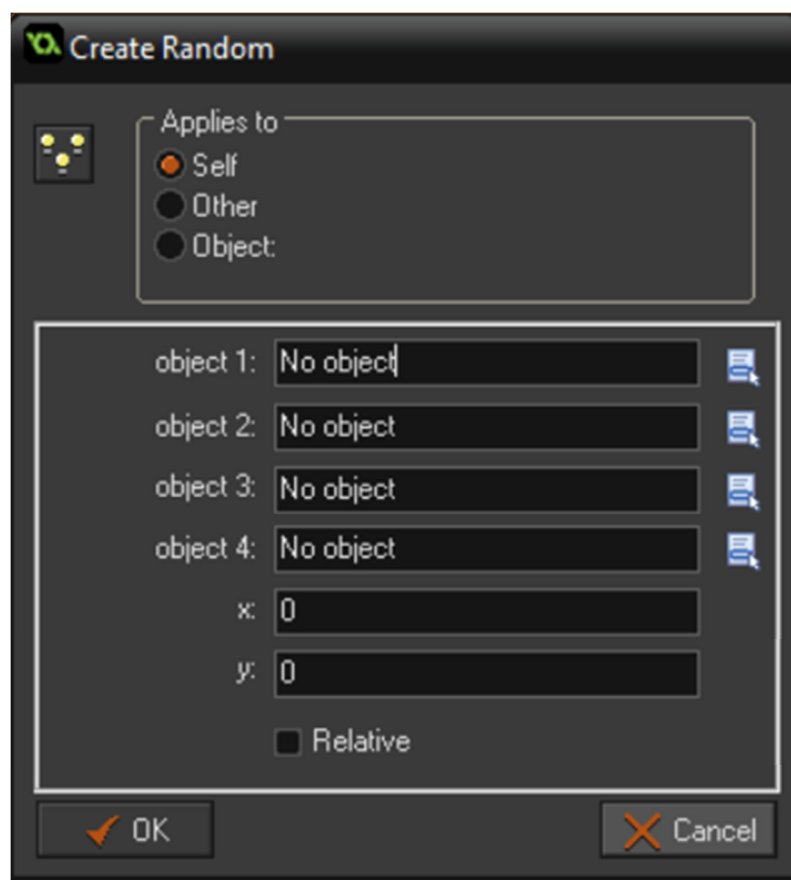
*Move Create Instance Action*

Create Moving : Εδώ δίνεται η δυνατότητα να δημιουργήσουμε ένα αντικείμενο, με κάποια κίνηση προς μια συγκεκριμένη κατεύθυνση. Όπως και πριν πρέπει να ορίσουμε συντεταγμένες  $x$  και  $y$  , ταχύτητα και κατεύθυνση.



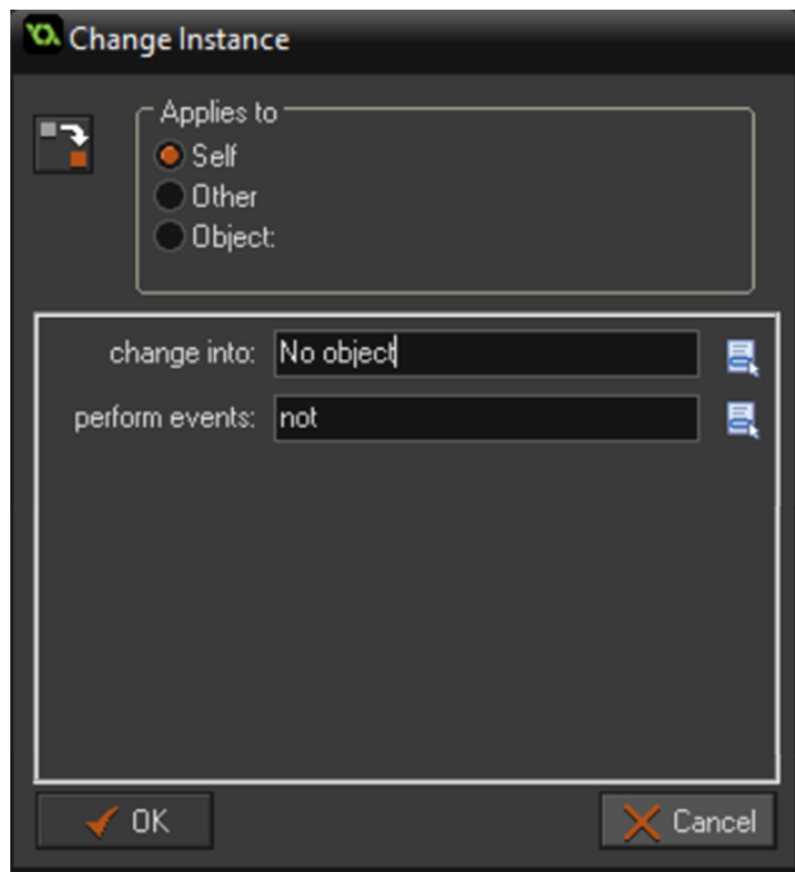
*Move Create Moving Action*

Create Random : Ισχύει ότι ισχύει και στο Create Instance την διαφορά ότι επιλέγουμε τέσσερα αντικείμενα. Ξανά όπως και πριν πρέπει να προσδιορίσουμε συντεταγμένες X και Y.



*Move create Random Action*

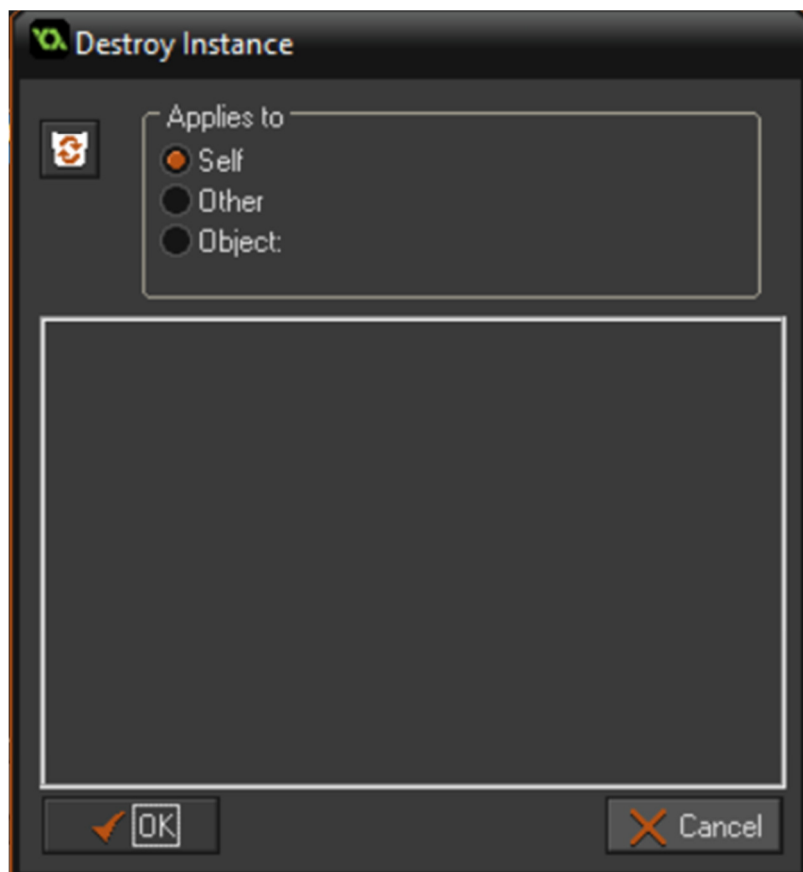
Change Instance : Με αυτό το Action, αντικαθιστάμε το αντικείμενό μας με κάποιο άλλο και το πότε θέλουμε να γίνει αυτό, αφού όλες οι ζωές τελειώσουν πχ το final boss να εκκρίνετε. Αυτό θα γίνει κατά αυτό τον τρόπο.



*Move Change Instance Action*

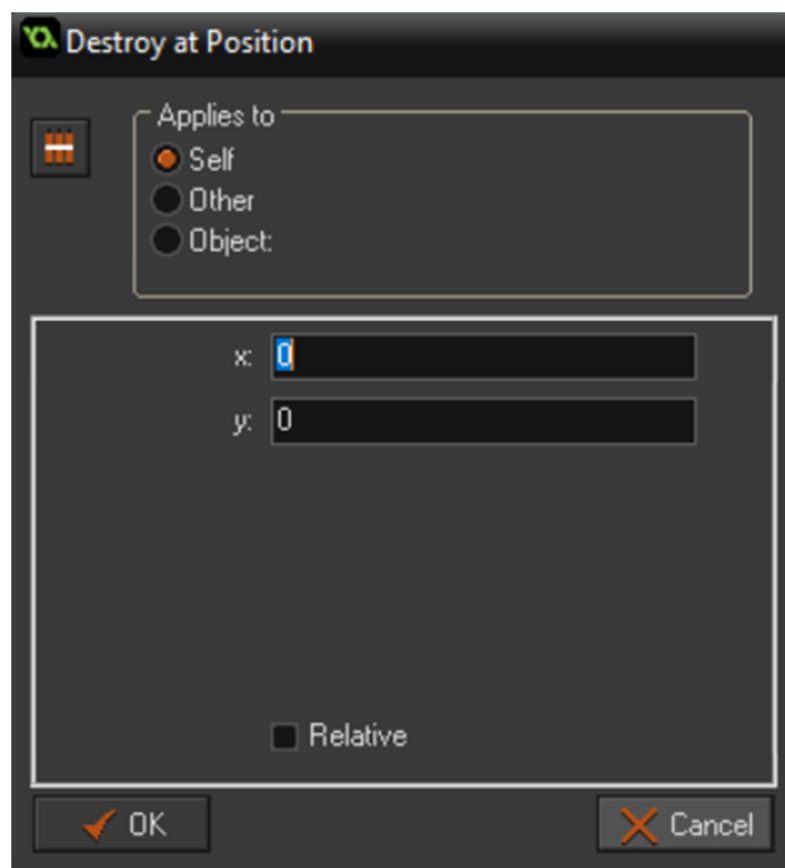


Destroy Instance : Αυτό το Action μας δίνει την δυνατότητα να "καταστρέψουμε" ένα αντικείμενο, δηλαδή να το αφαιρέσουμε από το δωμάτιο.



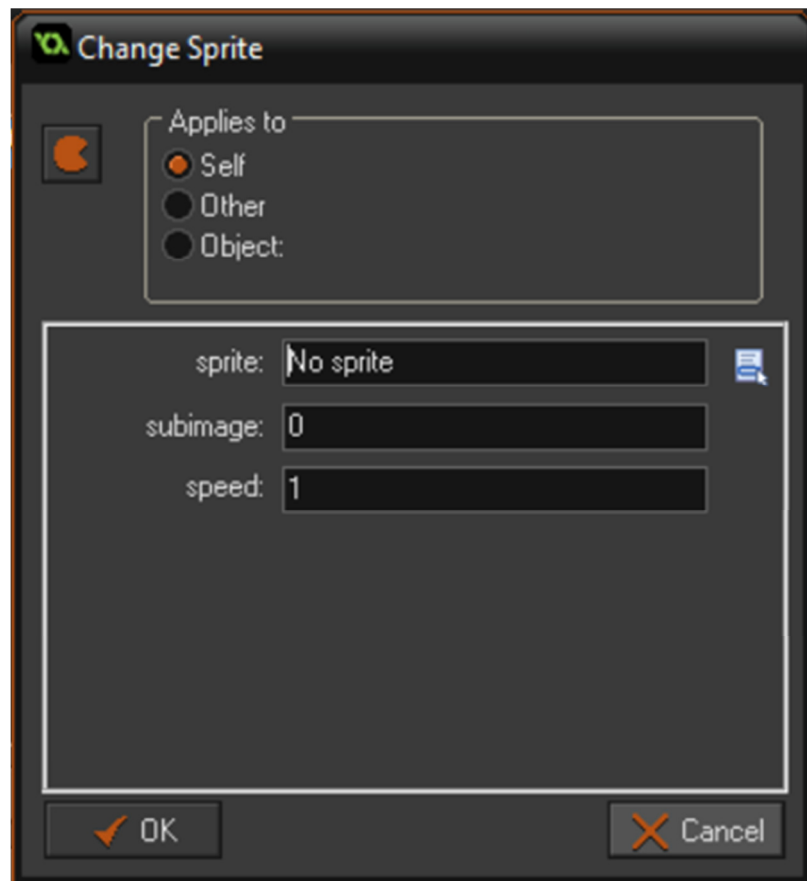
*Move Destroy Instance Action*

Destroy At Position : Αφαίρει ένα αντικείμενο της επιλογής μας με συγκεκριμένες συντεταγμένες X και Y.



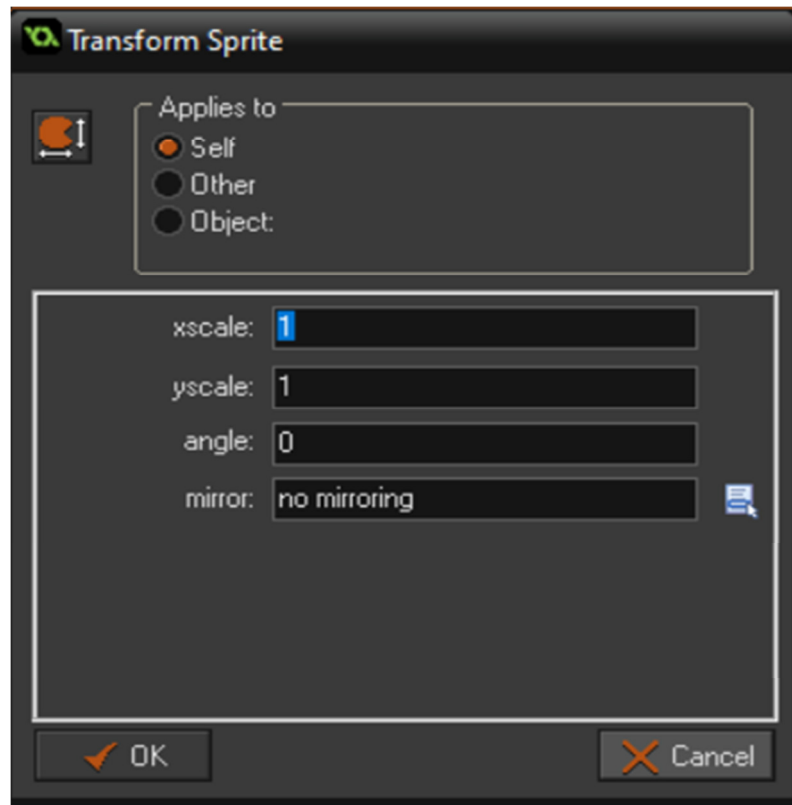
*Move Destroy at Position Action*

Change Sprite : Με αυτό το Action αντικαθιστούμε το Sprite της επιλογής μας κάποιο άλλο.



*Move Change Sprite Action*

Transform Sprite : Με αυτό το Action, μπορούμε να αλλάξουμε το μέγεθος που έχει το Sprite δηλαδή να πειράξουμε το ύψος και το πλάτος ακόμη μπορούμε να αλλάξουμε την φορά που θα έχει και την κατεύθυνσή του.



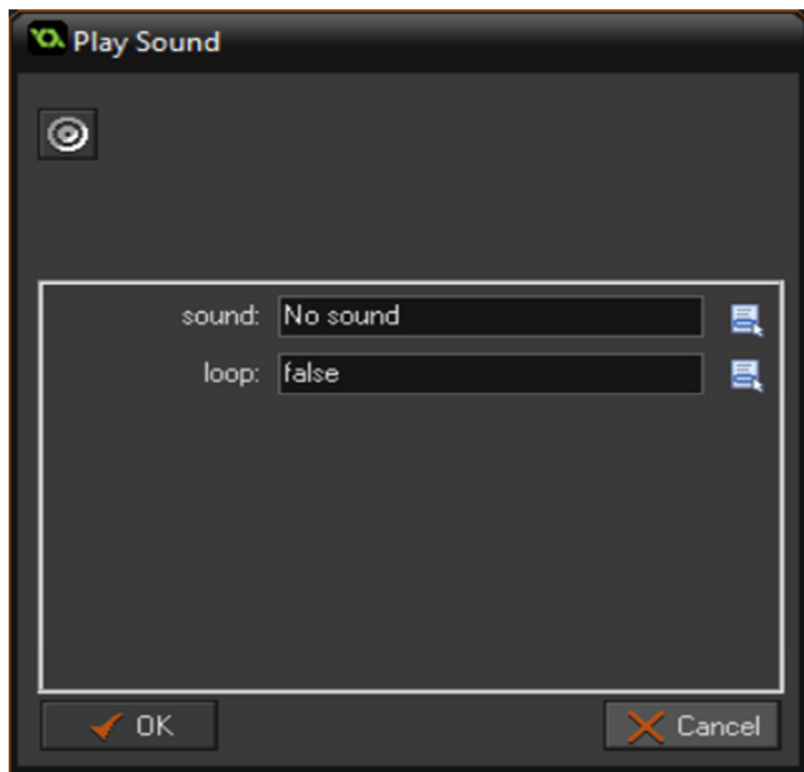
*Move Transform Sprite Action*

Ο πίνακας παρακάτω είναι μια χρήσιμη βοήθεια για τις αλλαγές που θα γίνουν στο Sprite σε σχέση με την τιμή του Scale που θα δώσουμε.

Scale	Width/Height change	Mirrored
Less than -1	Larger	Yes
-1	Normal	Yes
0 to -1	Smaller	Yes
0	None	No
0 to 1	Smaller	No
1	Normal	No
More than 1	Larger	No

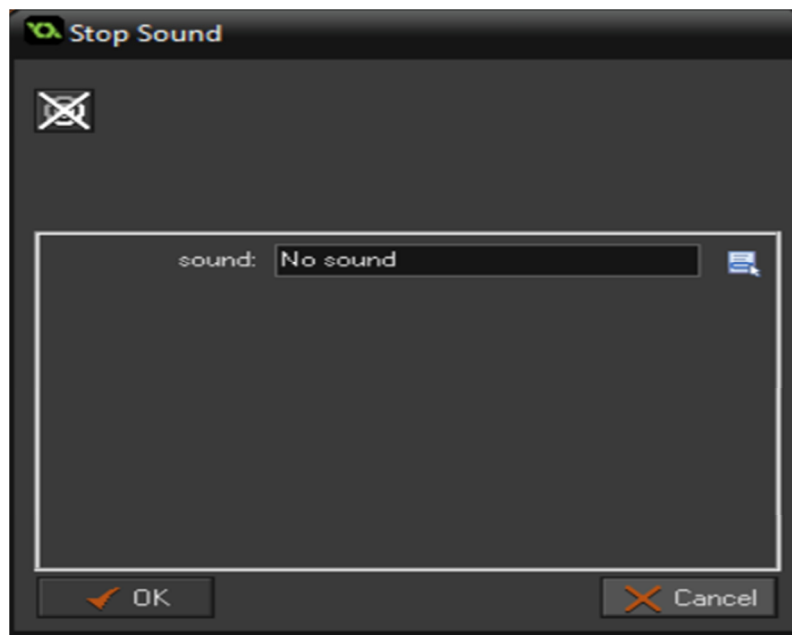
*Αλλαγές σε Sprite στην Τιμή του Scale*

Play Sound : Με αυτό το Action επιλέγουμε έναν για αναπαραγωγή και επιλέγουμε αν θα παίζει σε λούπα για πάντα ή μόνο μία φορά.



*Play Sound Action*

Stop Sound : Αντίστοιχα, με αυτό το Action δίνουμε εντολή να σταματήσει η αναπαραγωγή κάποιου ήχου.

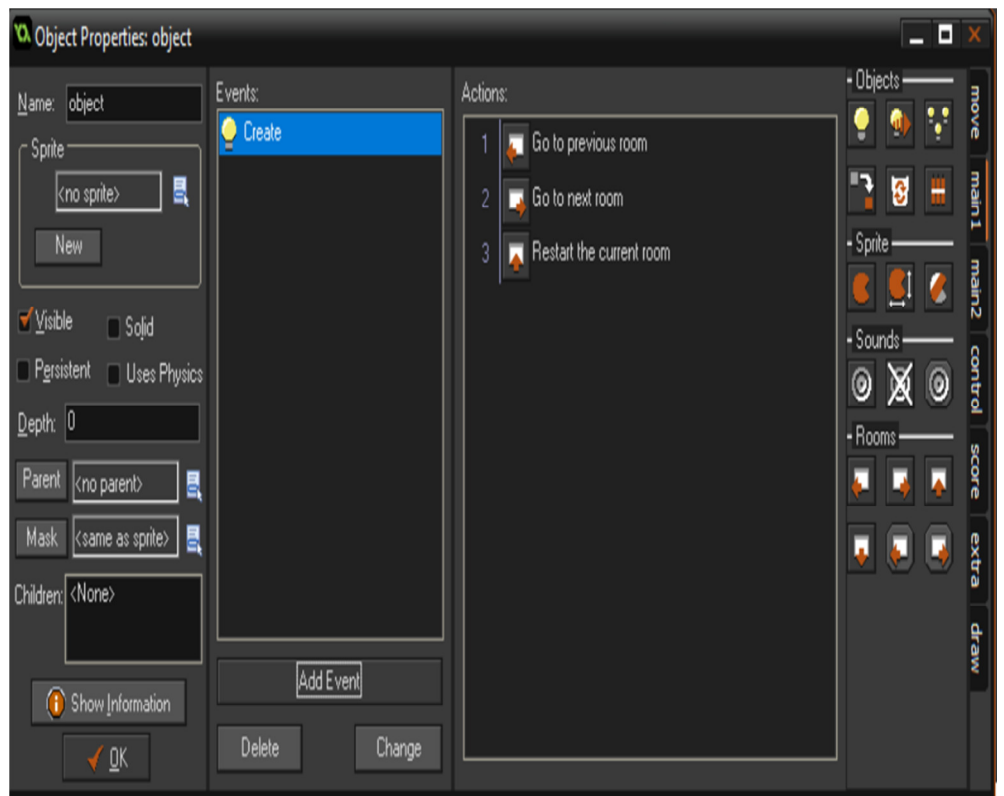


*Stop Sound Action*

Previous Room : Με αυτό το Action θα μεταφερθούμε στο προηγούμενο δωμάτιο παιχνιδιού μας ακόμη επιλέγουμε τον τρόπο εμφάνισης ή αλλιώς το εφέ αυτής της μεταφοράς

Next Room : Με αυτό το Action θα μεταφερθούμε στο επόμενο δωμάτιο.

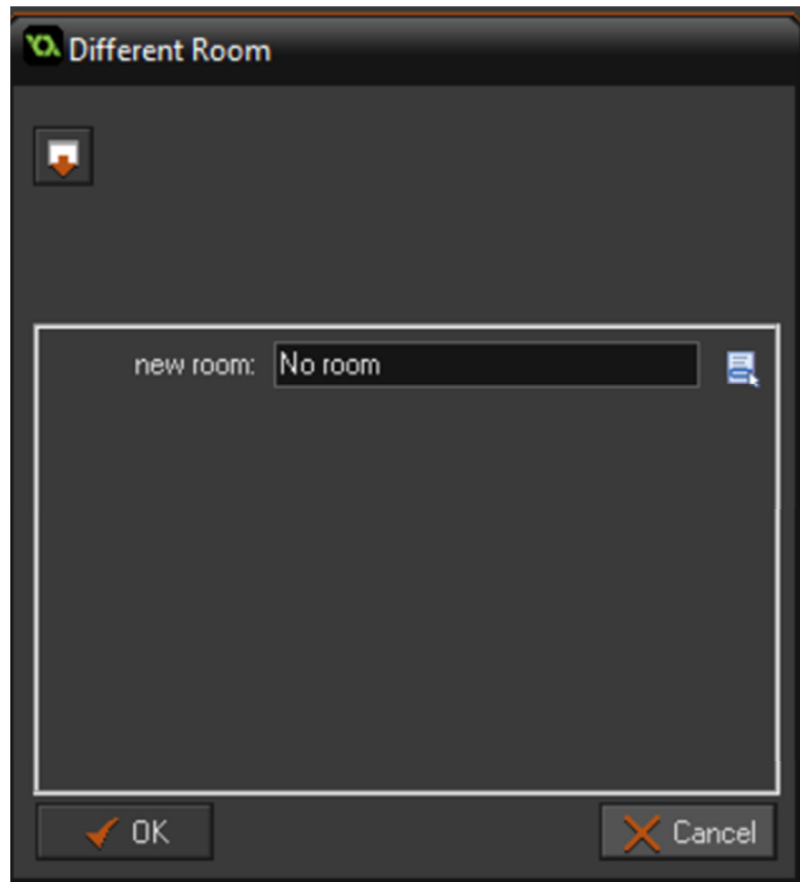
Restart Room : Με αυτό το Action ζητάμε να γίνει επανεκκίνηση του δωματίου στο οποίο βρισκόμαστε ήδη .



*Room Actions*

Different Room : Με αυτό το Action θα μεταφερθούμε σε κάποιο συγκεκριμένο δωμάτιο το οποίο το έχουμε επιλέξουμε εμείς ασχέτως της σειράς που βρίσκετε μέσα στο παιχνίδι.



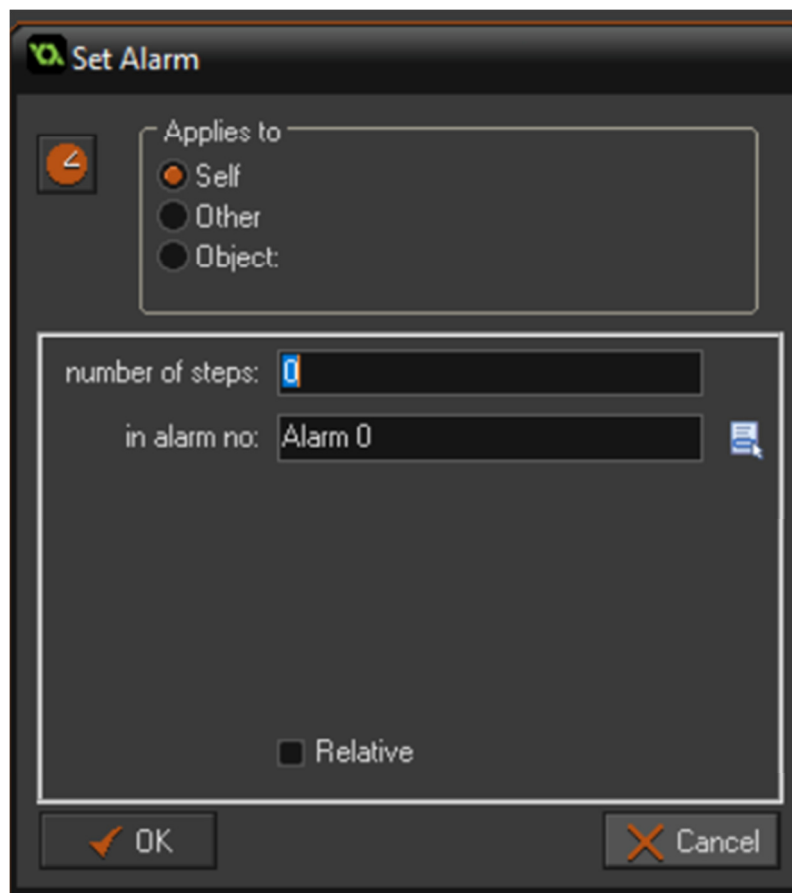


*Different Room Action*

Ακόμη να αναφέρουμε το Check Previous & Check Next τα οποία ελέγχουν πριν και μετά από το δωμάτιο στο οποίο βρισκόμαστε εκείνη την συγκεκριμένη χρονική στιγμή.

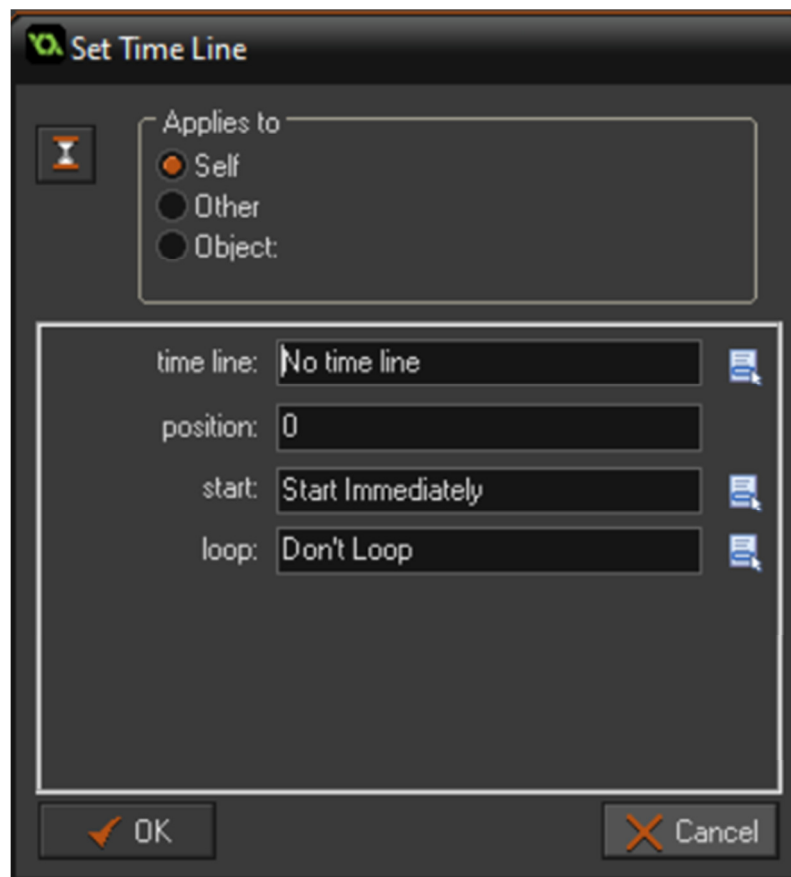
### 3.1.2.3 MAIN 2 ACTIONS

Set Alarm : Με αυτό το Action ορίζουμε τα Alarm Clocks που θα ενεργοποιούνται αφότου έχει ολοκληρωθεί ένας αριθμός steps που έχουμε ορίσει. Έτσι ενεργοποιείτε το Alarm την στιγμή που το θέλουμε και αυτό με την σειρά του ενεργοποιεί το Action που θέλουμε να γίνει μια συγκεκριμένη χρονική στιγμή στο παιχνίδι μας.



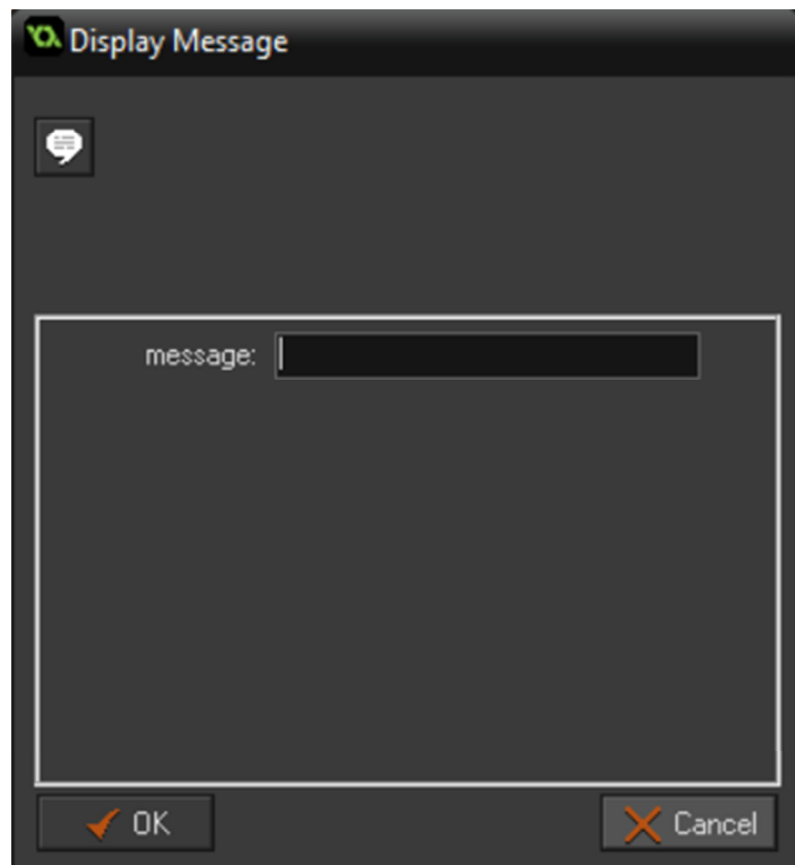
*Set Alarm Action*

Set Time Line: Με αυτό το Action ορίζουμε το Time Line όπως και άλλες παραμέτρους για το σημείο στο οποίο θέλουμε να ξεκινήσει και αν θα κάνει loop.



*Set Time Line Action*

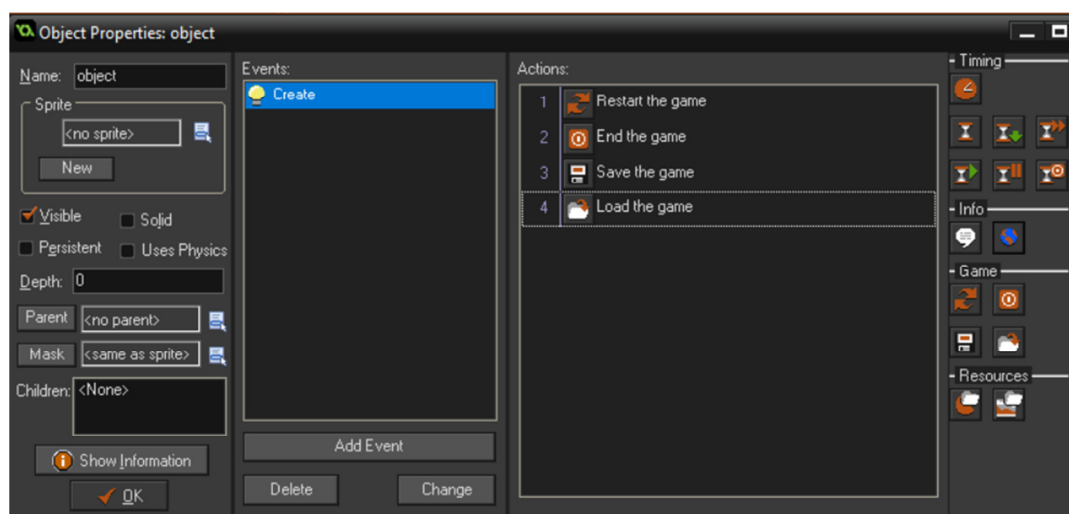
Display Message : Αυτό το Action έχουμε την δυνατότητα να εμφανίσουμε κάποιο μήνυμα κατά την διάρκεια του παιχνιδιού.



*Display Message Action*

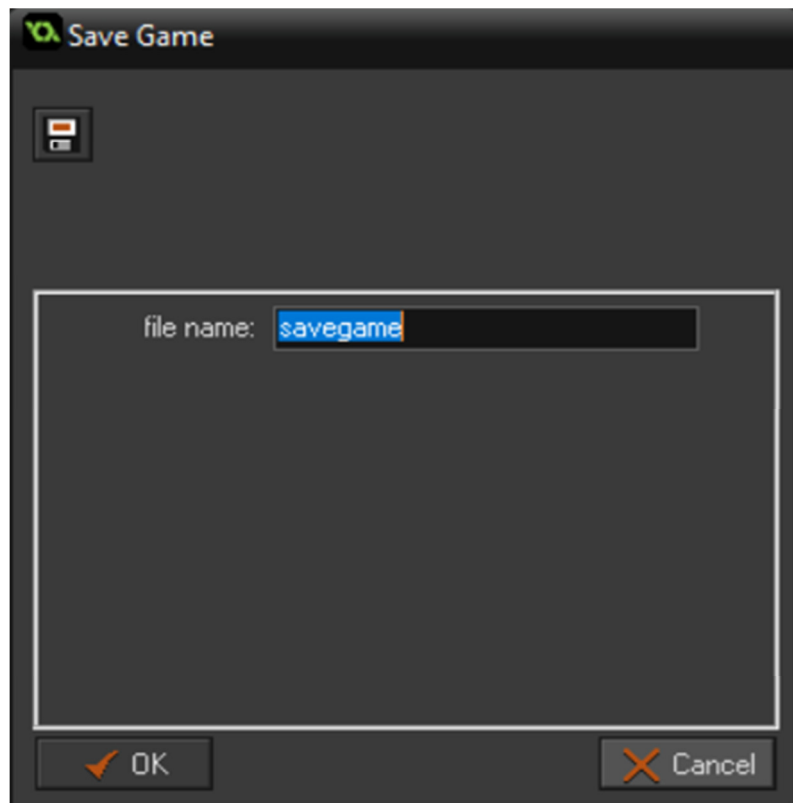
Restart Game : Με αυτό το Action δίνουμε την εντολή να ξανά αρχίσει το παιχνίδι από την αρχή.

And Game : Και προφανώς εδώ τερματίζουμε, σταματάμε το παιχνίδι, απλά.



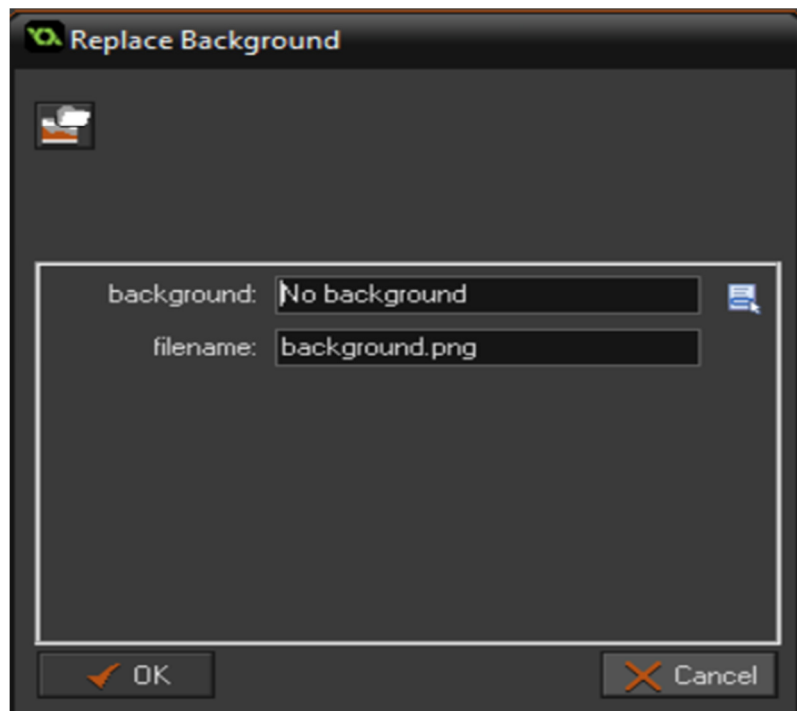
*Restarts and End Game*

Αντίστοιχα και με τα Save και Load Game, όπου στο σημείο στο οποίο τα τοποθετούμε, αποθηκεύουμε το παιχνίδι, ή φορτώνουμε κάποιο άλλο.



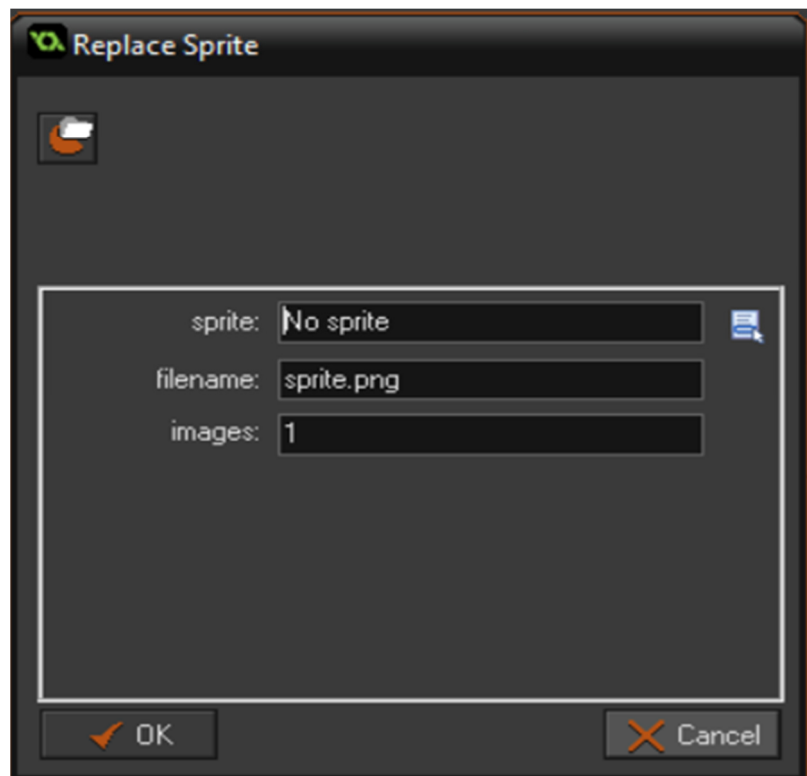
*Save Game Action*

Replace Background: Αλλάζει το background του δωματίου



*Replace background Action*

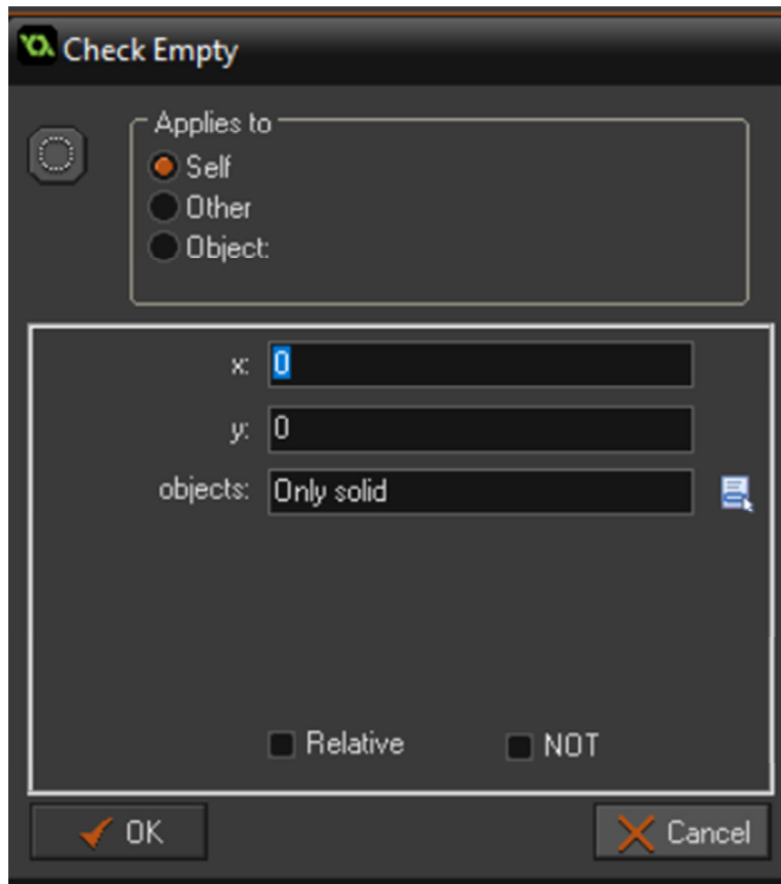
Replace Sprite: Αλλάζει το sprite σε ένα object.



*Replace Sprite Action*

#### **3.1.2.4 CONTROL ACTIONS**

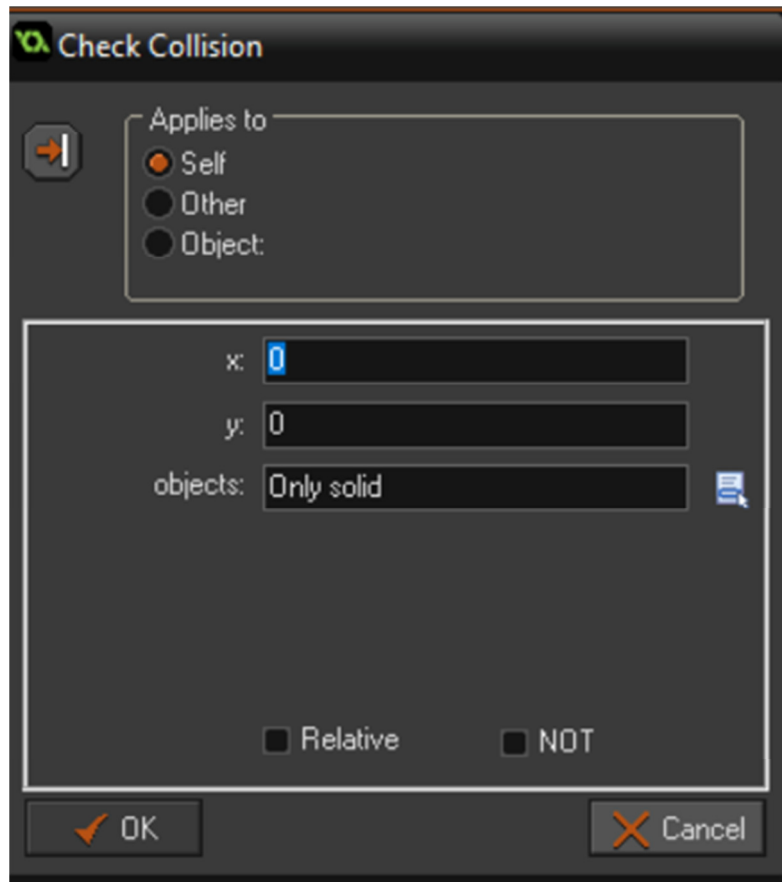
Check Empty : Με αυτό το Action ελέγχουμε εάν σε κάποιο σημείο του δωματίου που θέλουμε να κινηθεί το αντικείμενό μας υπάρχει κάποιο άλλο αντικείμενο.



*Check Empty Action*

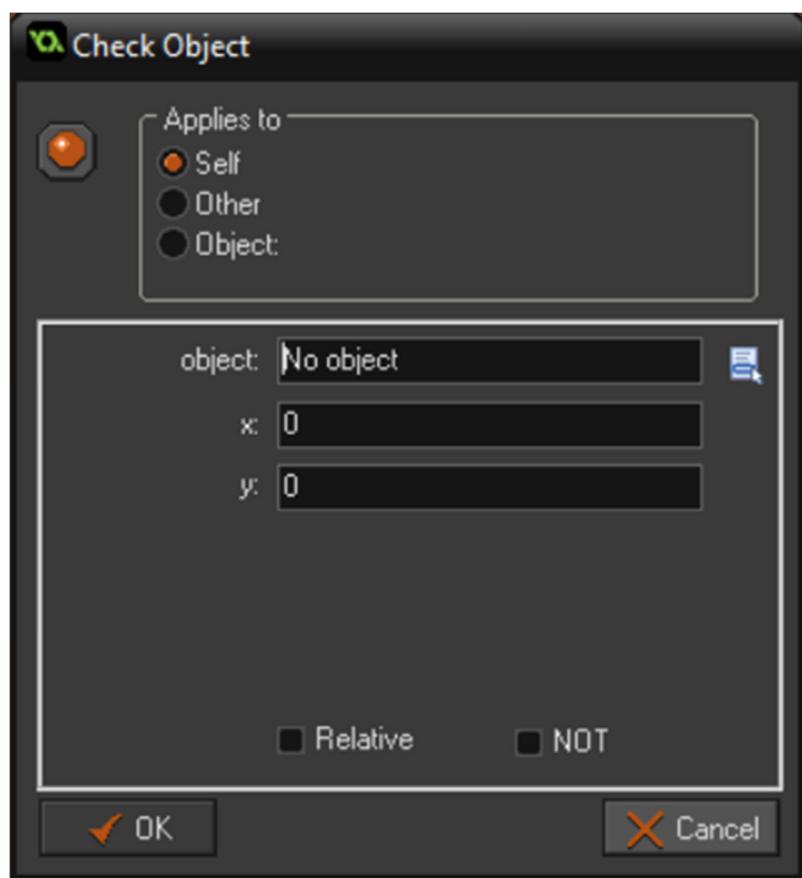
Check Collision : Αυτό το Action είναι το αντίστροφο με το Check Empty. Μας επιστρέφει «true» εάν στις συντεταγμένες που δίνουμε δημιουργείται κάποια αλληλεπίδραση (collision) με κάποιο άλλο αντικείμενο , δηλαδή το σημείο αυτό δεν είναι κενό.





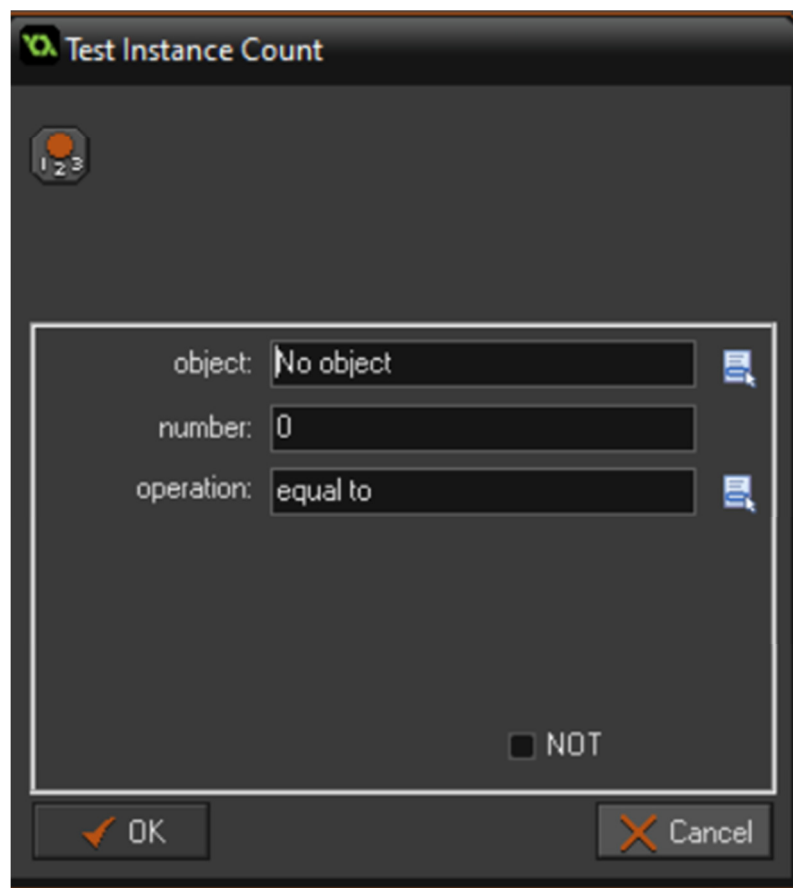
*Check Collision Action*

Check Object : Εδώ ελέγχουμε εάν σε κάποιο συγκεκριμένο σημείο του δωματίου υπάρχει κάποιο συγκεκριμένο αντικείμενο και ενεργούμε ανάλογος.



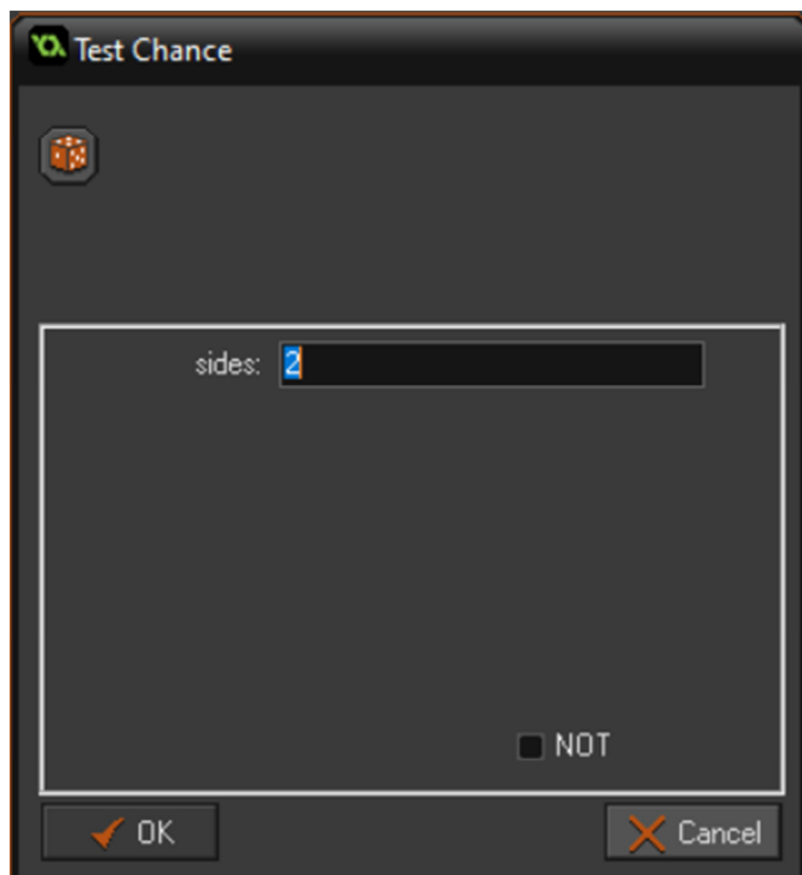
*Check Object Action*

Test Instance Count : Εδώ ελέγχουμε εάν κάποιο συγκεκριμένο αντικείμενο εμφανίζεται λιγότερες φορές, περισσότερες από τον συγκεκριμένο αριθμό που έχουμε ορίσει εμείς.



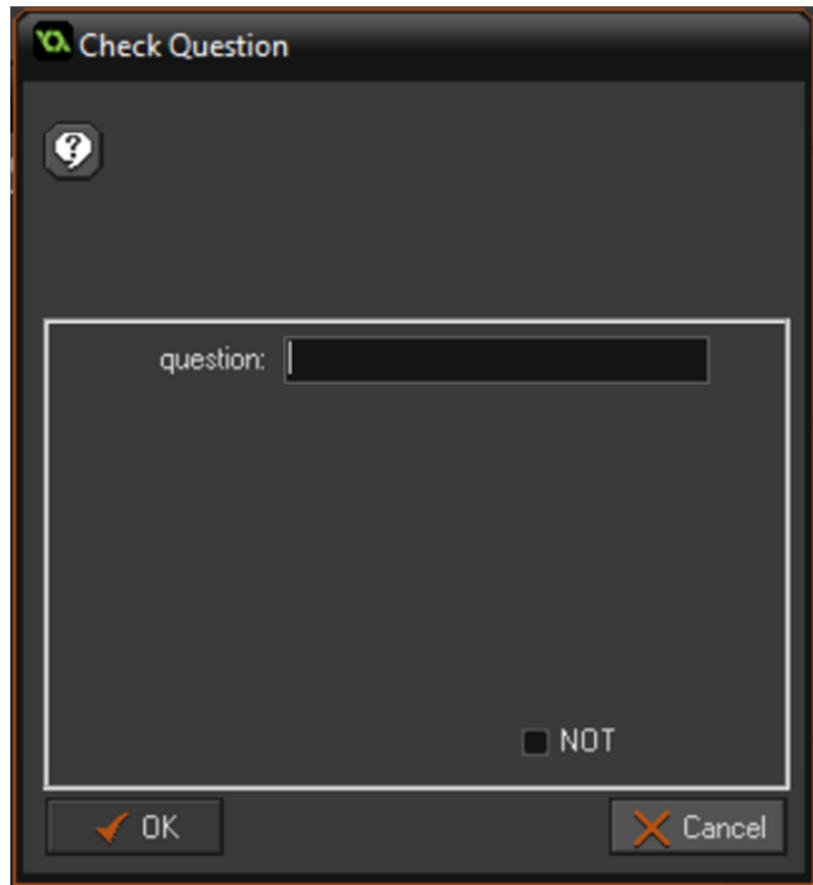
*Test Instance Count Action*

Test Chance : Με αυτό το Action φτιάχνουμε ένα ζάρι το οποίο έχει όσες πλευρές ορίσουμε εμείς. Αν έρθει 1 τότε εκτελούνται οι εντολές που θα δώσουμε εάν όχι τότε δεν εκτελούνται. Όσο πιο πολλές είναι οι πλευρές του ζαριού τόσο λιγότερες είναι και οι πιθανότητες να φέρουμε 1 άρα και να εκτελεστούν οι εντολές.



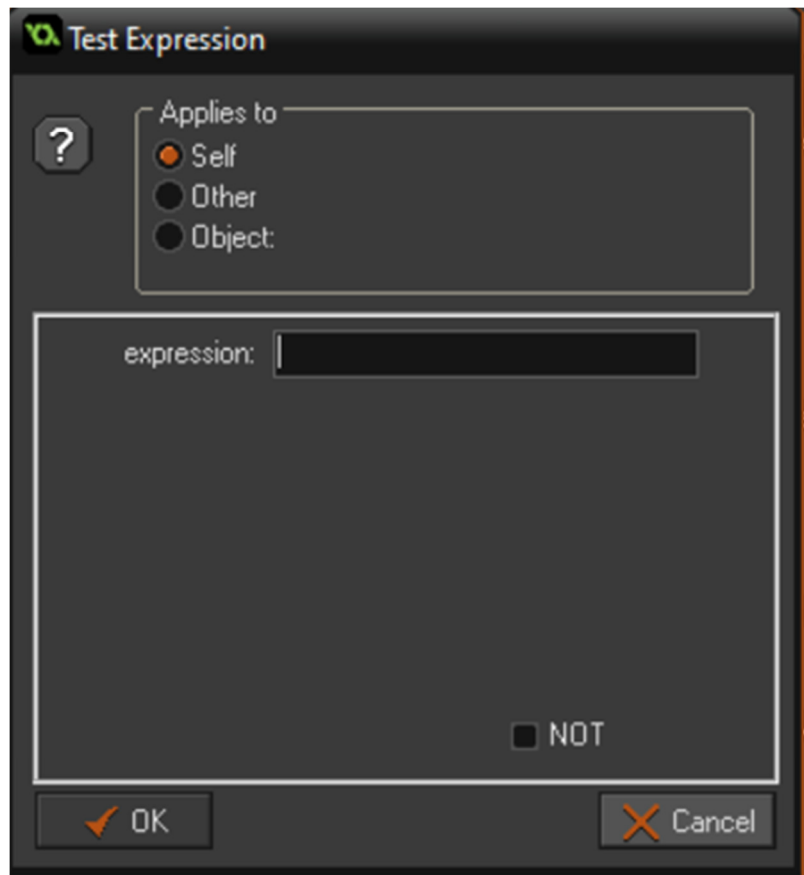
*Test Chance Action*

Check Question : Με αυτό το Action εμφανίζετε μια ερώτηση στον παίκτη του παιχνιδιού την οποία έχουμε συντάξει εμείς και περιέχει της επιλογές ναι και όχι. Σε κάθε επιλογή ορίζουμε τα Action που θα τρέχουνε.



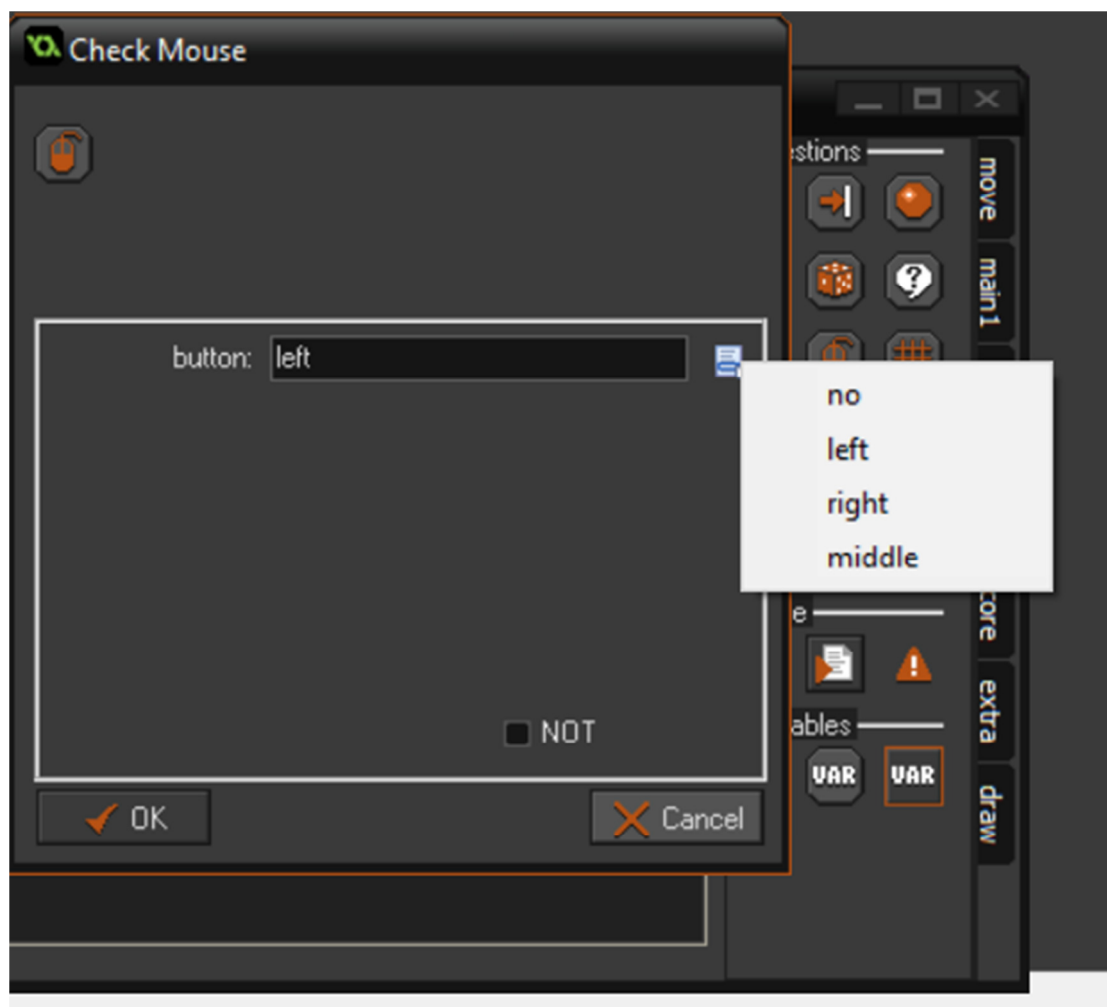
*Check Question Action*

Test Expression : Εδώ εισάγουμε μια έκφραση όπου ελέγχουμε εάν είναι σωστή, ώστε να εκτελεστούν οι παρακάτω εντολές που θα εισάγουμε, αλλιώς δεν εκτελούνται.



*Test Expression Action*

Check Mouse : Αυτό το Action επιστρέφει true όταν το πλήκτρο του ποντικιού που έχουμε επιλέξει έχει πατηθεί.



*Check Mouse Action*

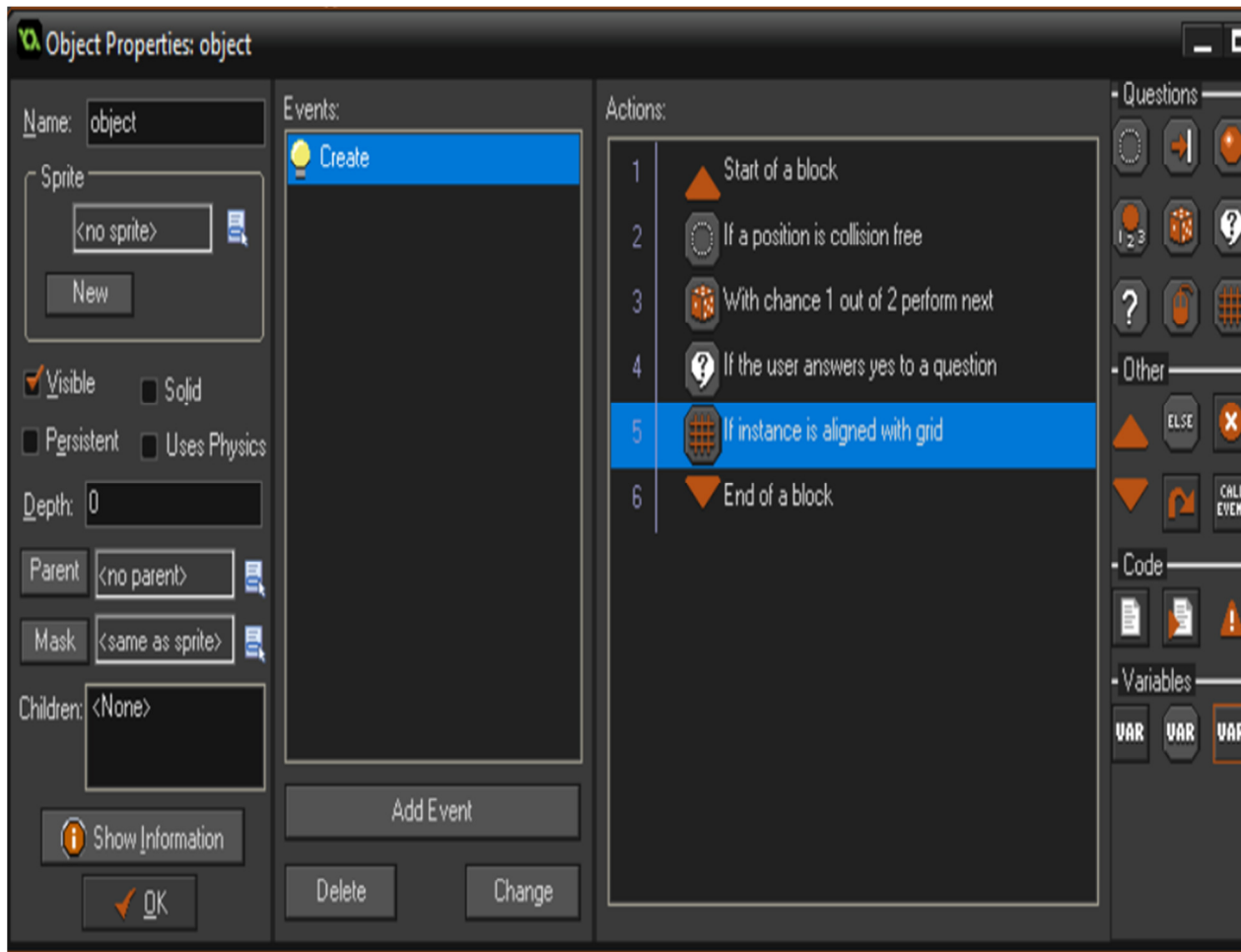
Check Grid : Εδώ επιλέγουμε μια περιοχή του δωματίου χρησιμοποιώντας συντεταγμένες X και Y και όταν το αντικείμενο που έχουμε ορίσει ικανοποιήσει αυτές τις συντεταγμένες τότε το Action επιστρέφει True.



*Check Grid Action*

Start και Stop Block : Με αυτά τα Action μπορούμε να δέσουμε μια σειρά εντολών μαζί οπού θέλουμε να εκτελούνται σαν ένα πακέτο.

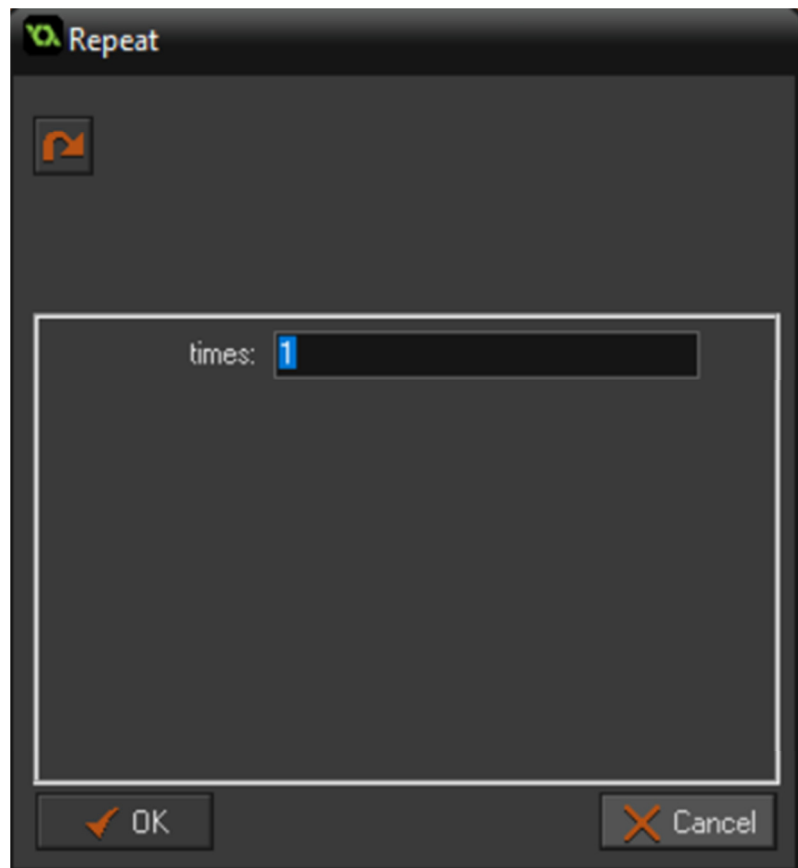




*Start And Stop Block Action*

Else : Εάν κάνουμε κάποια ερώτηση μέσω κάποιου action και μας δώσει ως απάντηση false, δεν θα εκτελεστούν οι εντολές που είχαμε ορίσει, αλλά οι εντολές που θα ορίσουμε κάτω από το Else.

Repeat : Με το Repeat, επαναλαμβάνεται το Action που ακολουθεί όσες φορές το ορίσουμε εμείς .

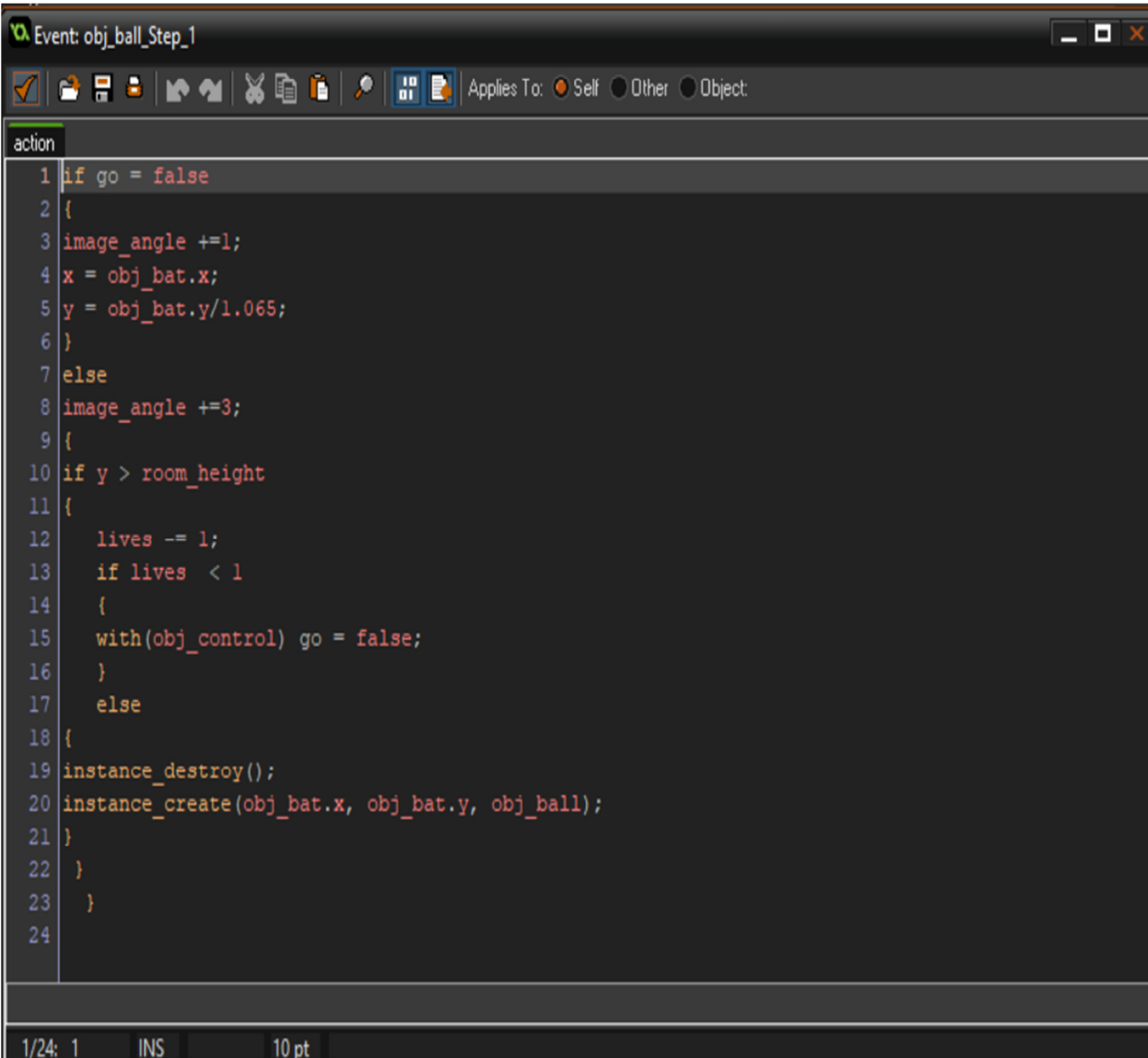


*Repeat Action*

Exit Event : Με αυτό το Action δίνουμε εντολή να βγούμε από το τρέχων Event. Για παράδειγμα μπορεί να έχουμε φτιάξει ένα Bonus υπό επίπεδο που να δίνει στον παίχτη extra πόντους και να είναι άσχετο με το main story και ο παίχτης να αποφασίσει να το παραλείψει.

Call Event : Με αυτό το Action καλούμε από την αρχή το Event

Execute Code: Με αυτό το Action ανοίγει ένα not pad όπου ο χρήστης μπορεί να γράψει το δικό του κώδικα.



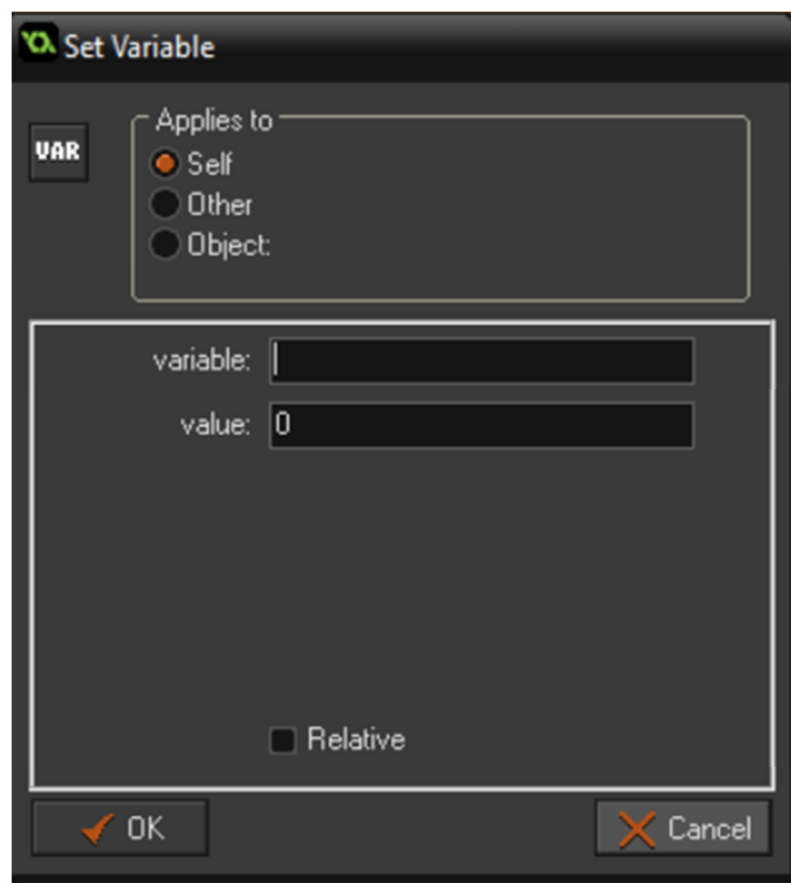
```
1 if go = false
2 {
3 image_angle +=1;
4 x = obj_bat.x;
5 y = obj_bat.y/1.065;
6 }
7 else
8 image_angle +=3;
9 {
10 if y > room_height
11 {
12 lives -= 1;
13 if lives < 1
14 {
15 with(obj_control) go = false;
16 }
17 else
18 {
19 instance_destroy();
20 instance_create(obj_bat.x, obj_bat.y, obj_ball);
21 }
22 }
23 }
24
```

1/24: 1    INS    10 pt

*Execute Code Action*

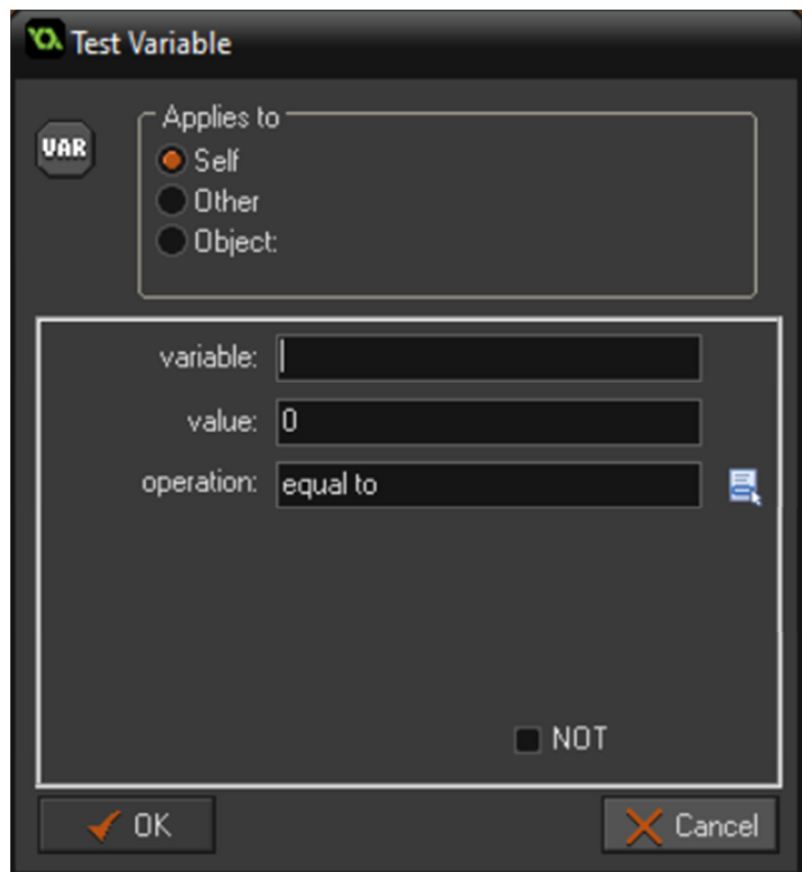
Προσοχή για πολύ μεγάλα κομμάτια κώδικα χρησιμοποιούμε SCRIPTS.

Set Variable : Με αυτό το Action θέτουμε την αρχική της τιμή σε μία μεταβλητή.



*Set Variable Action*

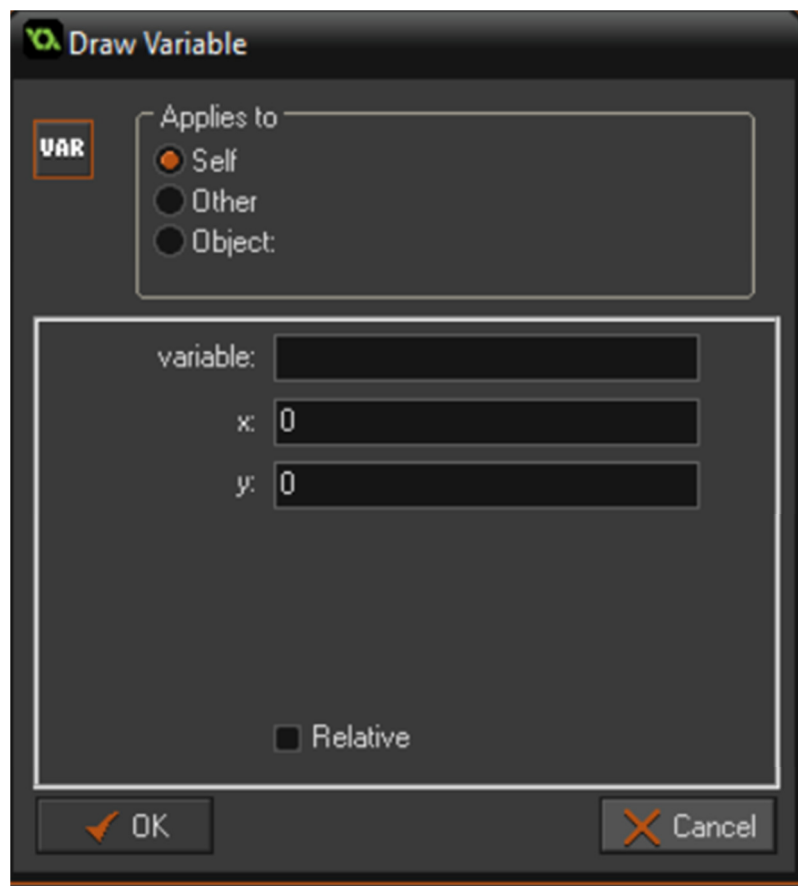
Test Variable : Εδώ συγκρίνουμε την τιμή μιας μεταβλητής εάν είναι ίση με κάποιον βαθμό ή εάν είναι μικρότερη ή μεγαλύτερη από κάποιον βαθμό.



The image shows a dialog box titled "Test Variable" with a dark grey background. At the top left, there is a green icon and the text "Test Variable". Below this, there is a section labeled "VAR" with a radio button icon. The "Applies to" section contains three radio buttons: "Self" (selected), "Other", and "Object". Below this, there are three input fields: "variable:" (empty), "value:" (containing "0"), and "operation:" (containing "equal to"). A small blue icon is visible to the right of the "operation:" field. At the bottom right of the main area, there is a checkbox labeled "NOT". At the bottom of the dialog, there are two buttons: "OK" with a checkmark icon and "Cancel" with an X icon.

*Test Variable Action*

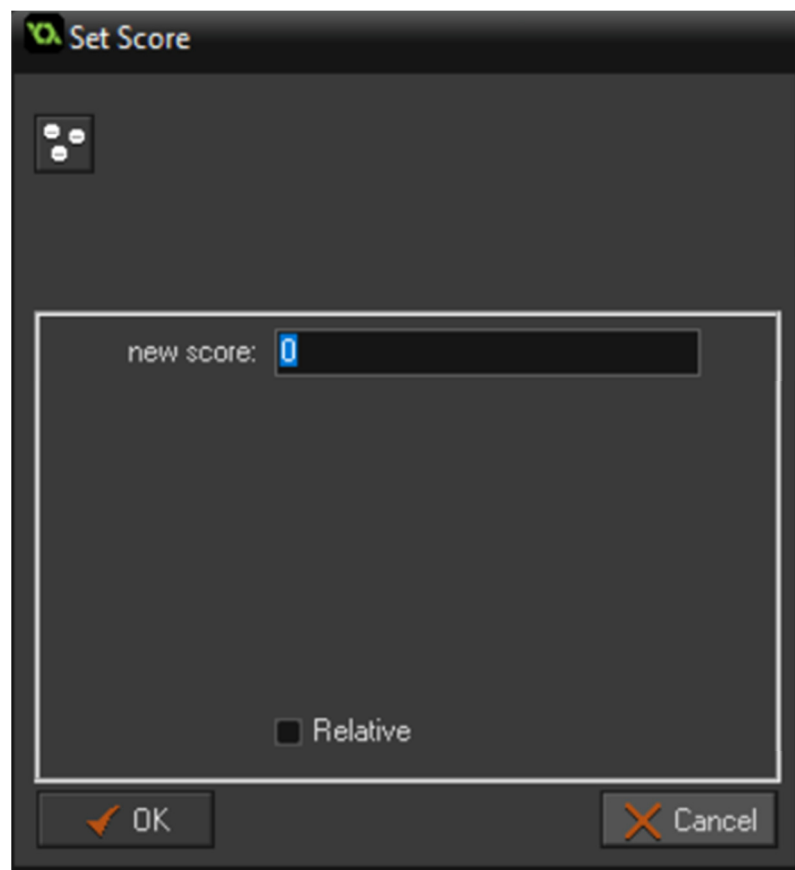
Draw Variable : Ορίζουμε να εμφανίζετε μία μεταβλητή όπως το σκορ μέσα σε κάποιο δωμάτιο.



*Draw Variable Action*

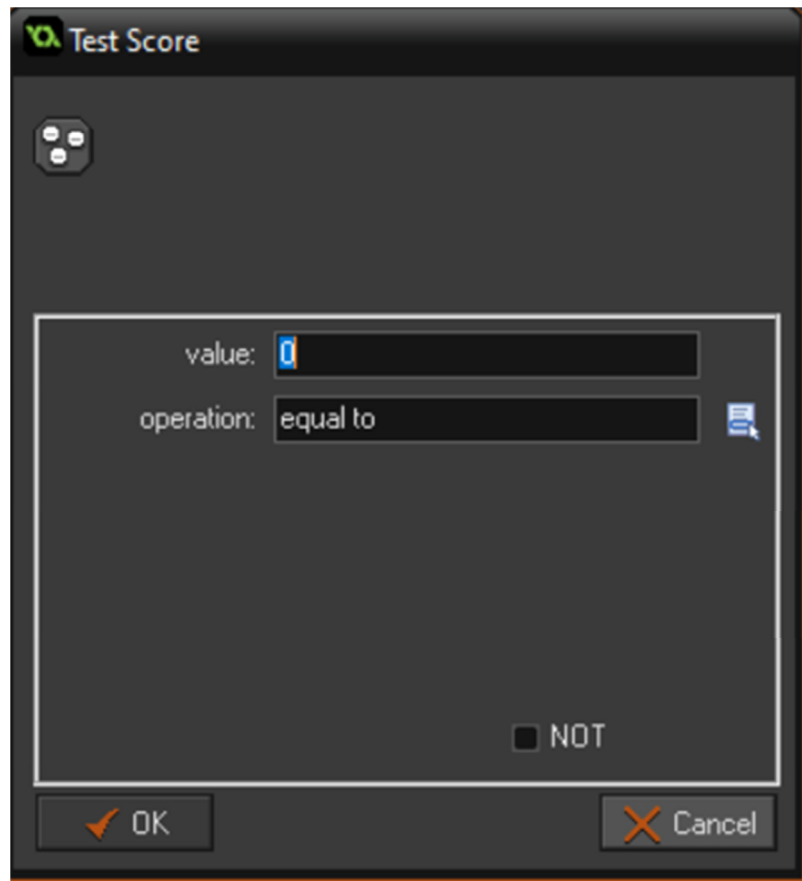
### 3.1.2.5 SCORE ACTIONS

Set Score: Ένας από του τρόπους που το Game Maker Studio κάνει την ζωή μας πιο εύκολη είναι ότι μας δίνει προ εγκατεστημένη μια μεταβλητή που κρατάει το Score. Με αυτό το Action μπορούμε να μεταβάλλουμε την τιμή της.



*Set Score Action*

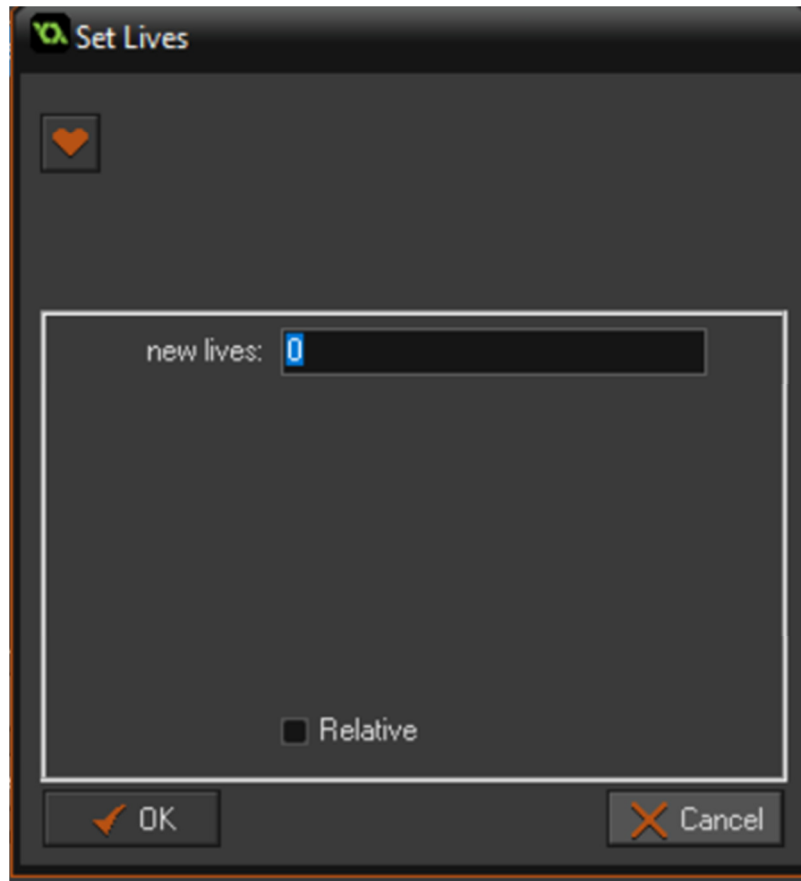
Test Score : Ελέγχουμε την τιμή του Score αν είναι ίση με την τιμή που δίνουμε , μεγαλύτερη ή μικρότερη ώστε να ορίσουμε τι ενέργειες θα γίνονται σε κάθε περίπτωση.



*Test Score Action*

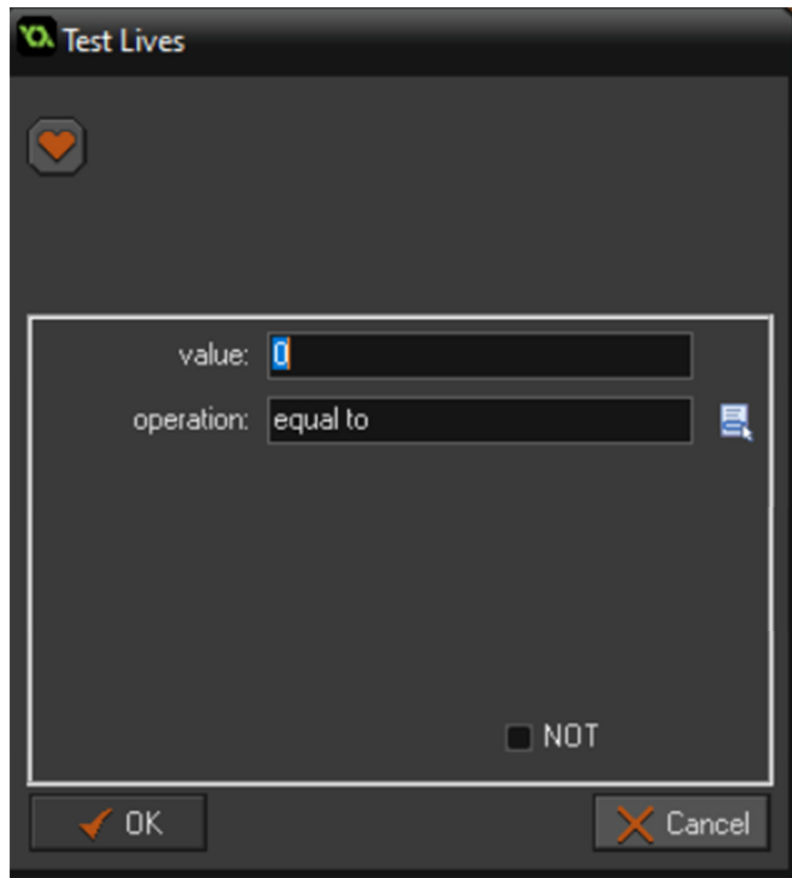


Set Lives: Αυτό το Action ελέγχει το έτυμο σύστημα ζωών που μας παρέχει το Game Maker Studio, ορίζουμε της ζωές που θέλουμε και της αυξομειώνουμε ανάλογα με τα Actions που έχουμε ορίσει.



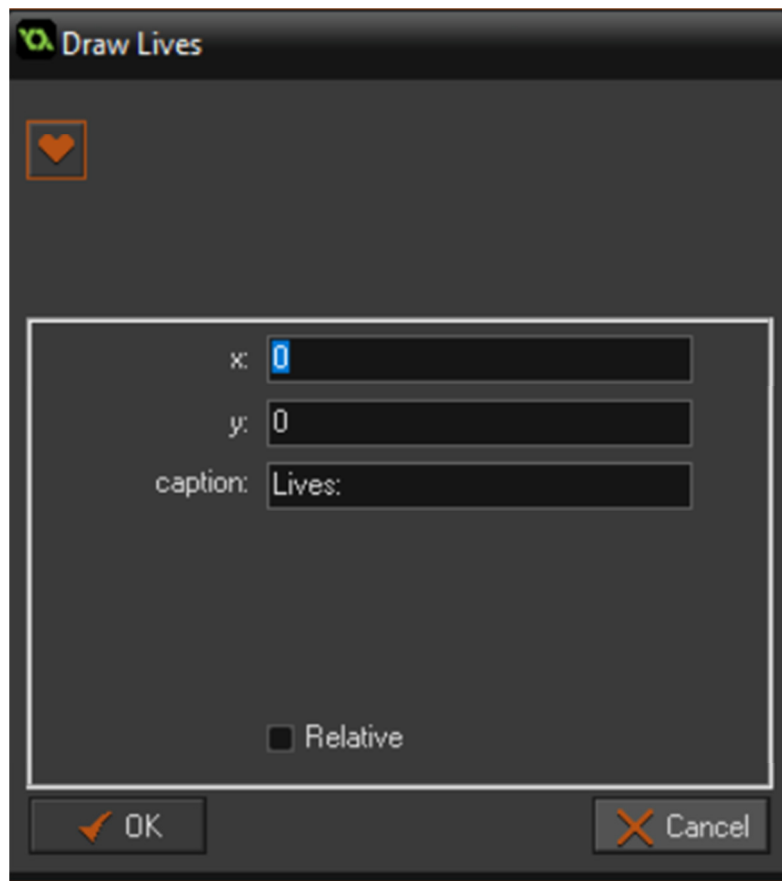
*Set Lives Action*

Test Lives :Με αυτό το Action επιλέγουμε τι ενέργειες που θα πραγματοποιούνται όταν ο αριθμός των ζωών είναι συγκεκριμένος. Η πιο συνηθισμένη χρήση είναι να ορίσουμε να εμφανίζετε Game Over όταν οι ζωές είναι μηδέν.



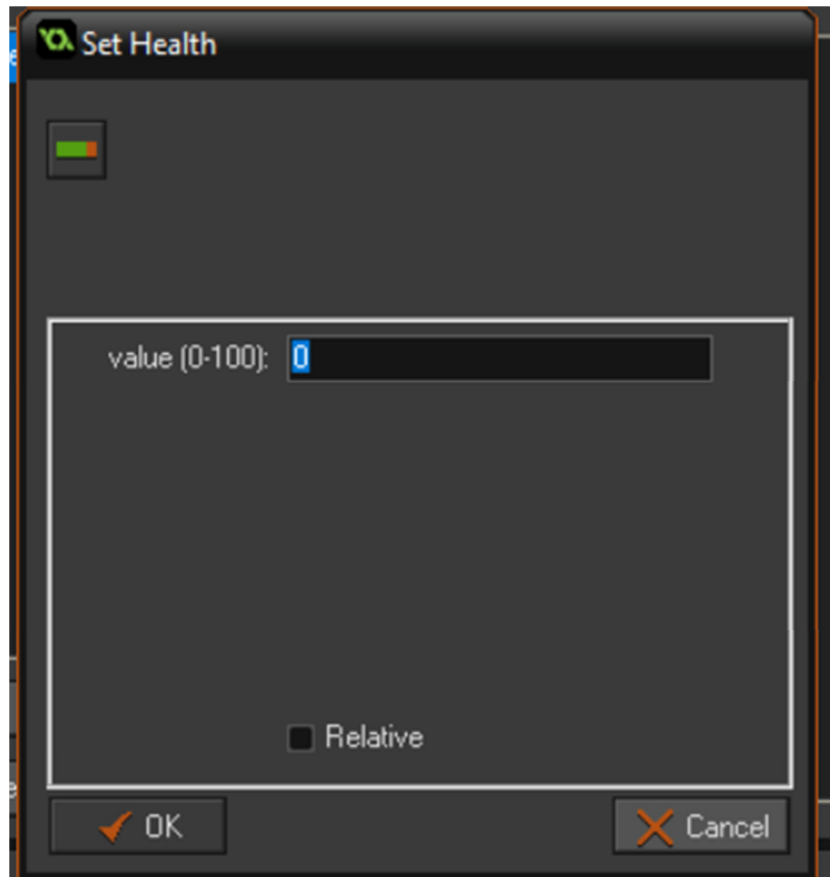
*Test Lives Action*

Draw life images : Με αυτό το Action μπορούμε να εμφανίζουμε στο παιχνίδι τον αριθμό των ζωών με την χρήση εικόνων όπως για παράδειγμα καρδούλες αντί για αριθμούς.



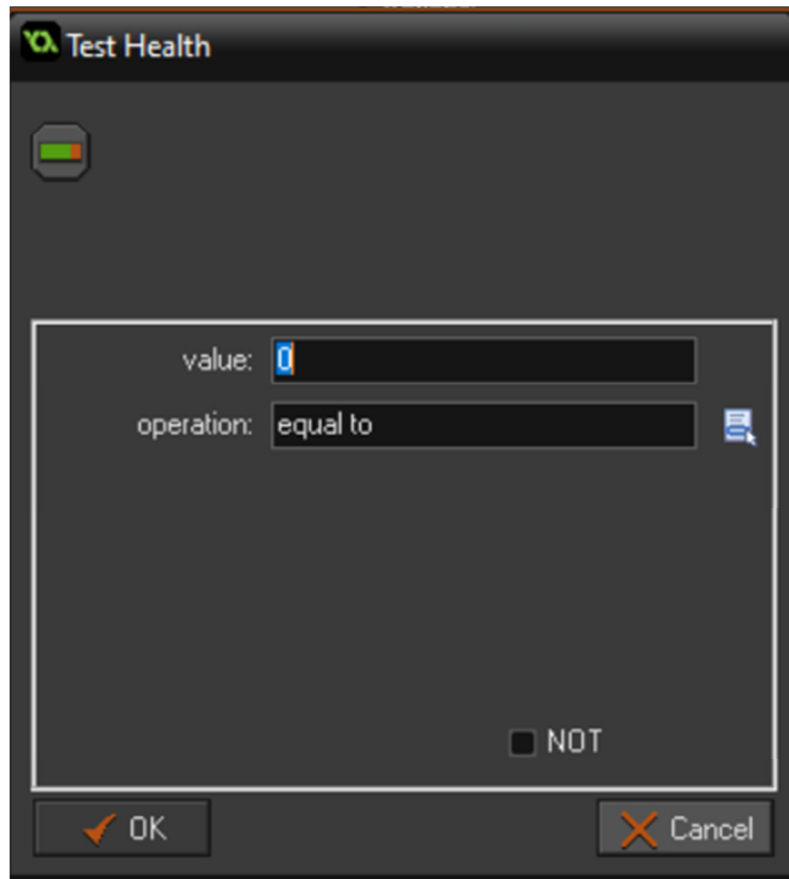
*Draw Lives Action*

Set Health : Το Game Maker Studio έχει ορισμένες βασικές μεταβλητές αποθηκευμένες στις βιβλιοθήκες του. Μία από αυτές είναι η μεταβλητή Health. Με το Action αυτό ορίζουμε την τιμή της όπου 100 είναι το μέγιστο και 0 το ελάχιστο. Με την τιμή να γίνετε 0 τότε θα ενεργοποιηθεί και το ανάλογο Event όπως το να αφαιρεθεί μία από της ζωές.



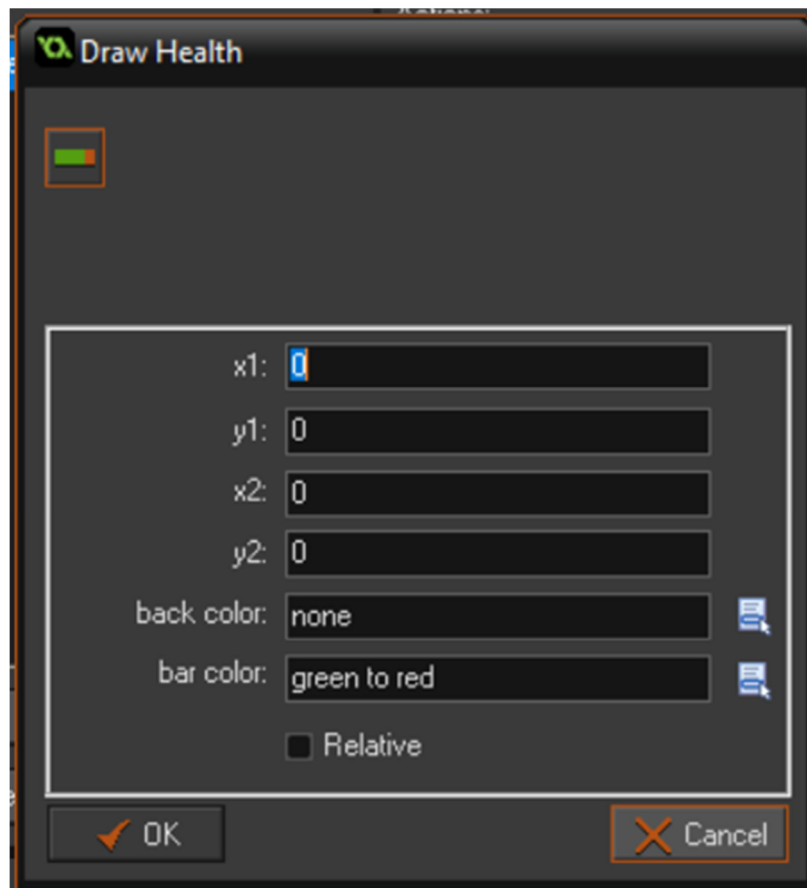
*Set Health Action*

Test Health Set Health : Ίδια λειτουργία με το Test Lives , όταν η ζωή φτάσει στο 0 το ανάλογο Action ενεργοποιείτε που του έχουμε ορίσει.



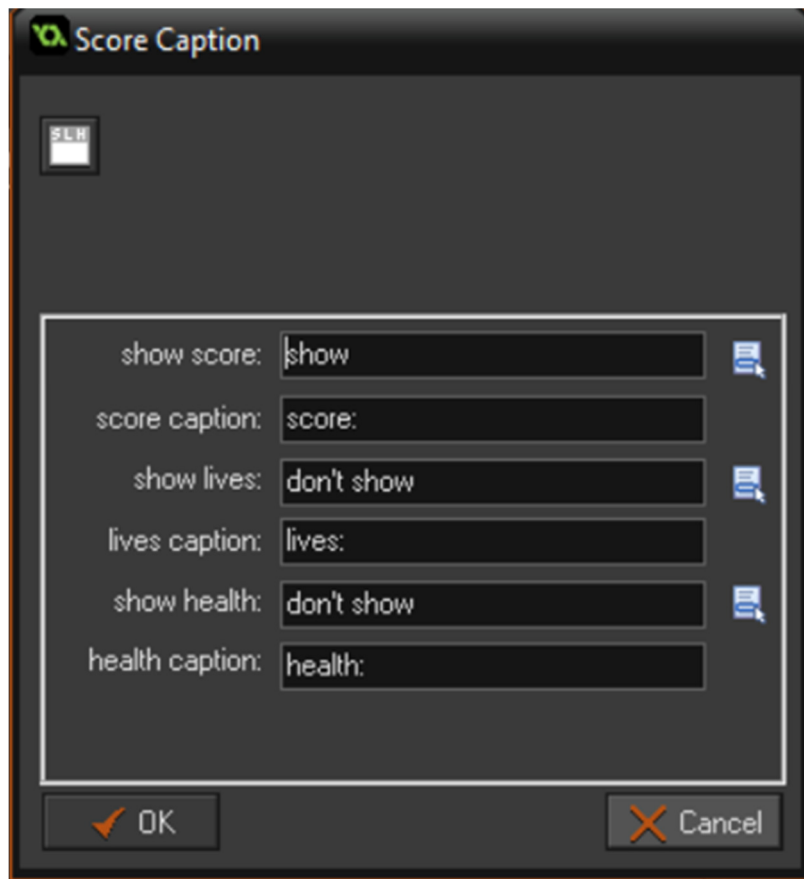
*Test Health Action*

Draw health : Η ζωή του παίχτη εμφανίζεται ως μία μπάρα αντί για καρδούλες σε κάποιο σημείο του του δωματίου που ορίζουμε εμείς όπως και τα χρώματα αυτής.



*Draw Health Action*

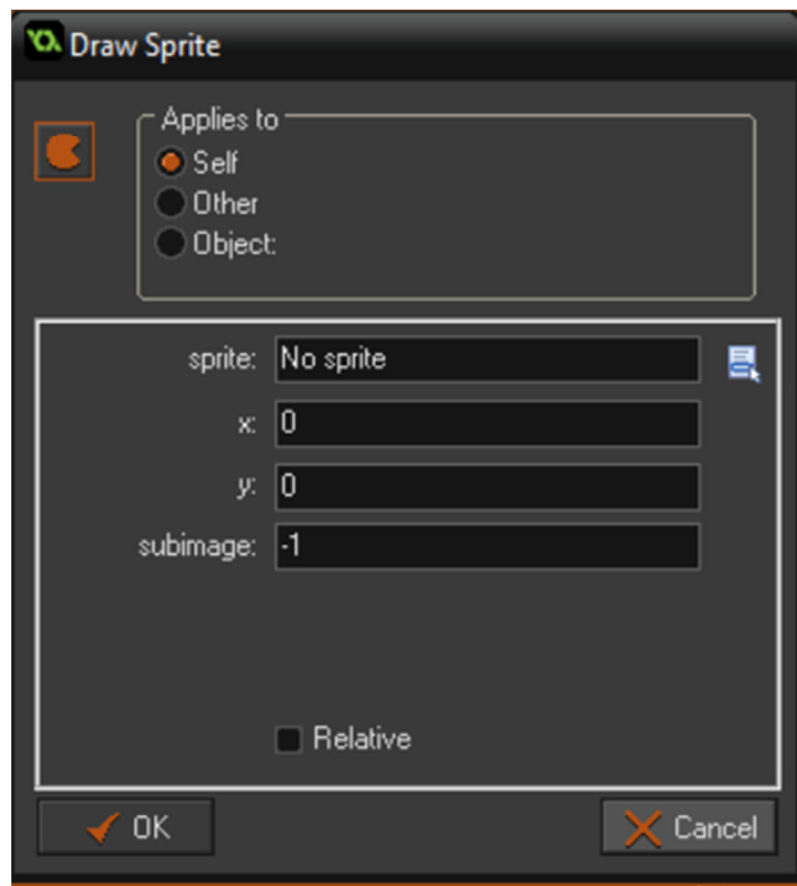
Score caption : Με αυτό το Action εμφανίζονται σε όποιο σημείο του δωματίου θέλουμε εμείς το όνομα του δωματίου το Score ζωές , η ενέργεια κτλ.



*Score Caption Action*

### 3.1.2.5 DRAW ACTIONS

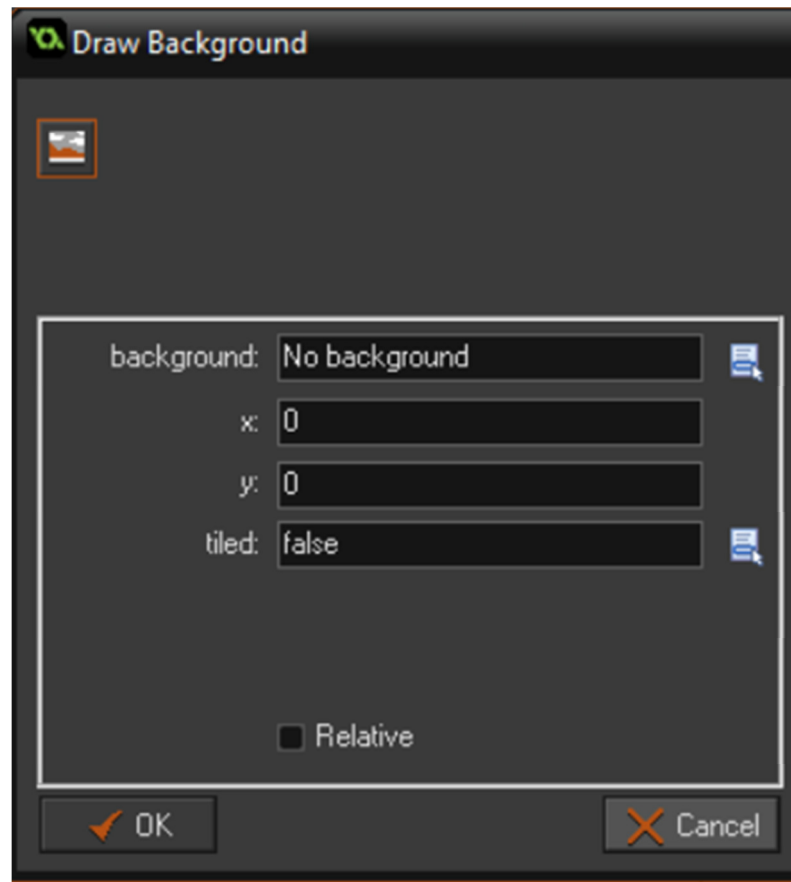
Draw Sprite : Εδώ απεικονίζουμε κάποιο Sprite δηλαδή μια εικόνα στο παιχνίδι μας ή αν το Sprite αποτελείται πολλές εικόνες, δηλαδή είναι μια κινούμενη εικόνα, για να τις εμφανίσει όλες σειριακά, βάζουμε την τιμή -1 στο subimage.



*Draw Sprites Action*

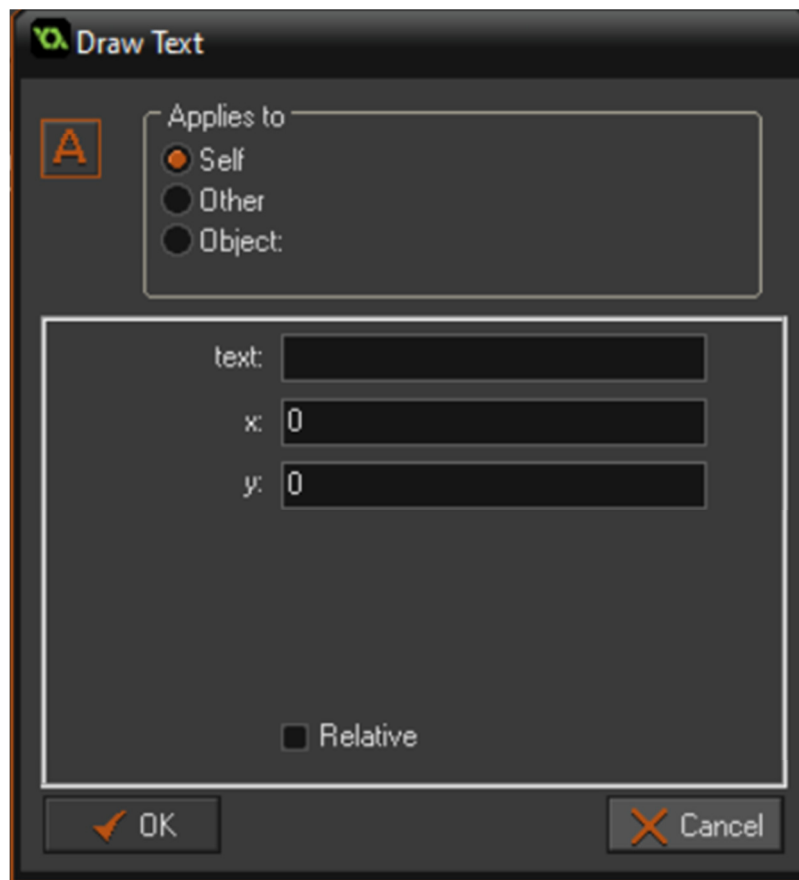
Draw Background : Επιλέγουμε εικόνα που ως background για το δωμάτιο του παιχνιδιού και αν θα την «τεντώσουμε» για να ταιριάζει ακριβώς στο δωμάτιο ή όχι.





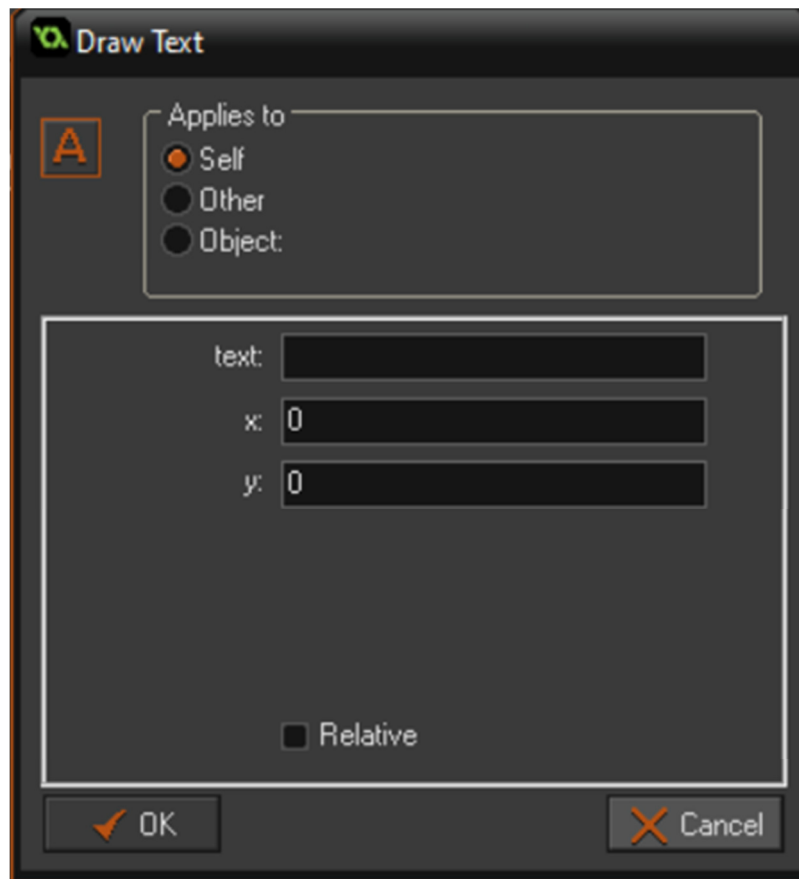
*Draw Background Action*

Draw Text : Εδώ εμφανίσουμε ένα κείμενο σε κάποιο σημείο του δωματίου της επιλογής μας. Μπορεί να είναι οτιδήποτε κείμενο χρειάζετε το παιχνίδι μας, από ένα κείμενο με οδηγίες μέχρι score ή κάποια άλλη μεταβλητή και τα λοιπά. Χρησιμοποιώντας το σύμβολο # αλλάζουμε γραμμή στο κείμενο που θέλουμε να απεικονίσουμε, επομένως με ένα μόνο Action, μπορούμε να εμφανίσουμε κείμενο πολλών γραμμών. Εάν το κείμενο που εισάγουμε το περικλύσουμε μέσα σε ‘ ‘ ή “ “ τότε το game maker δεν το εμφανίζει σαν απλό κείμενο αλλά το βλέπει ως έκφραση και εμφανίζει τα ανάλογα αποτελέσματα, για παράδειγμα όταν θέλουμε να εμφανίσουμε την τιμή κάποιας μεταβλητής.



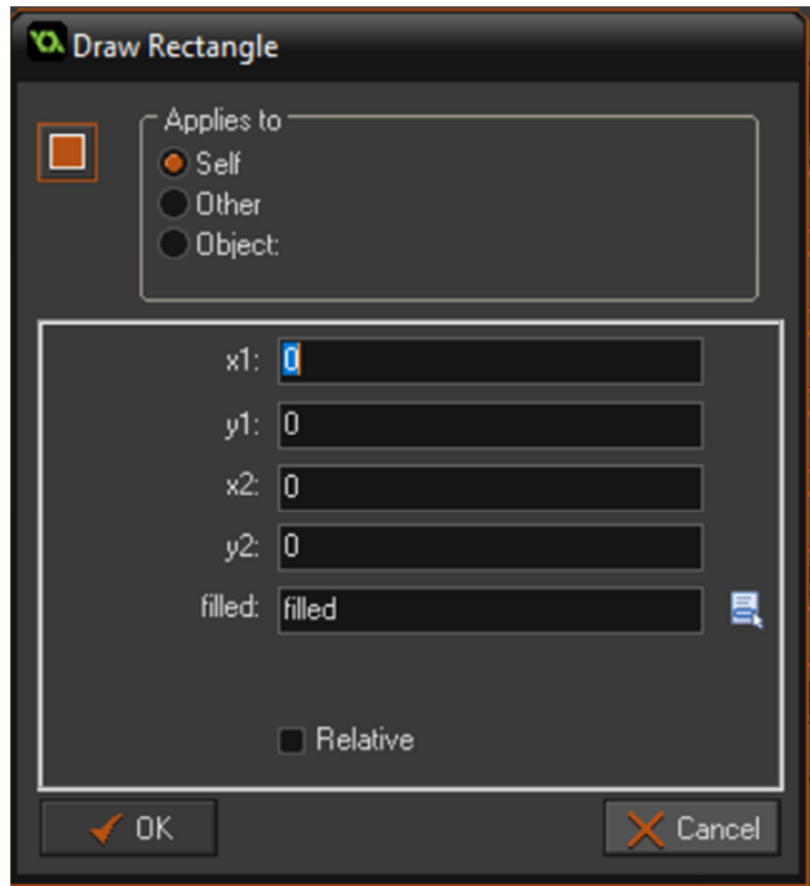
*Draw Text Action*

Draw Scaled Text : Είναι το ίδιο με το Draw με τη βασική διαφορά ότι έχουμε την δυνατότητα να αλλάζουμε και να επιλέξουμε το μέγεθος και τη μορφή του κειμένου που θέλουμε να εμφανίσουμε.



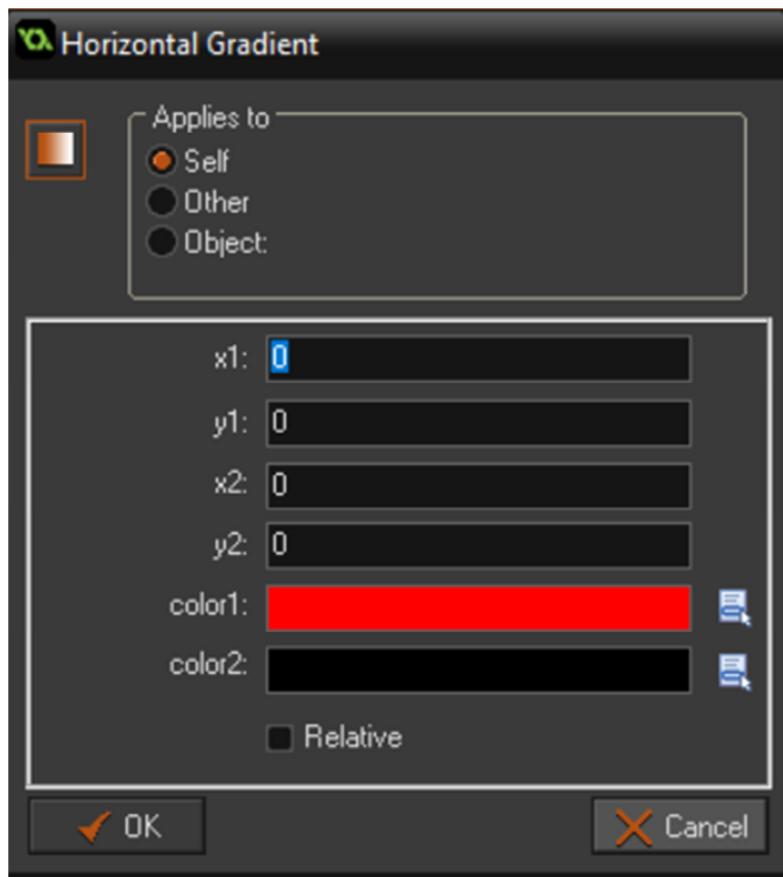
*Draw Scale Text Action*

Draw Rectangle : Απεικονίζει ένα απλό τετράγωνο στους άξονες X και Y που ορίσουμε εμείς. Αν κλικ άρουμε στο relative τότε οι συντεταγμένες μετράνε από το σημείο που βρίσκεται το αντικείμενό μας.



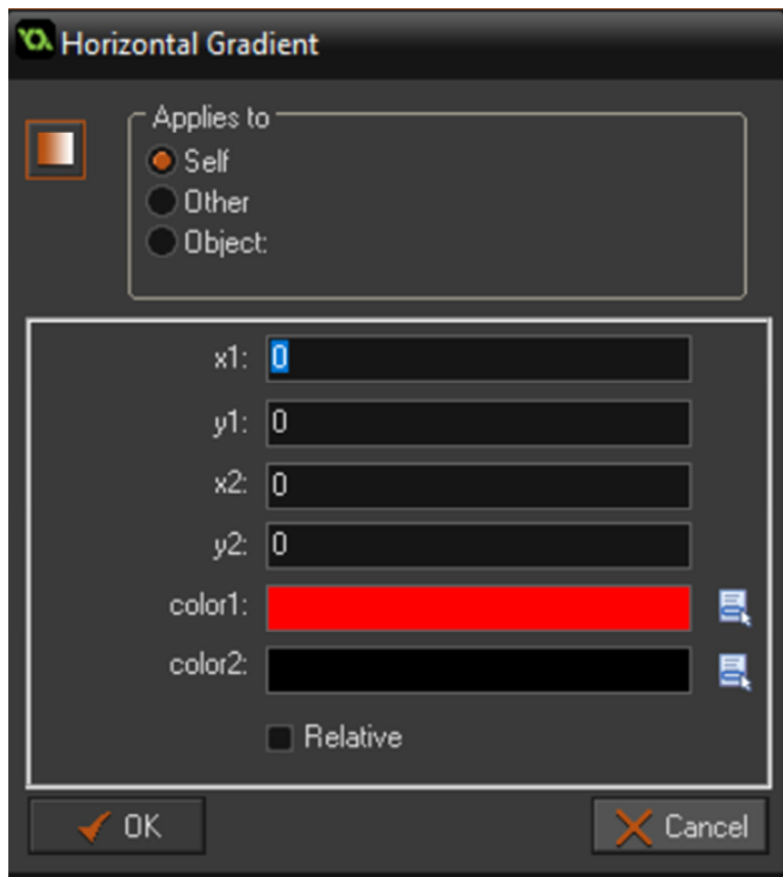
*Draw Rectangle Action*

Horizontal Gradient: Είναι το ίδιο με το Draw Rectangle όμως το τετράγωνο θα έχει δύο διαφορετικά χρώματα τα οποία και η εναλλαγή θα γίνετε οριζόντια από τα αριστερά προς τα δεξιά.

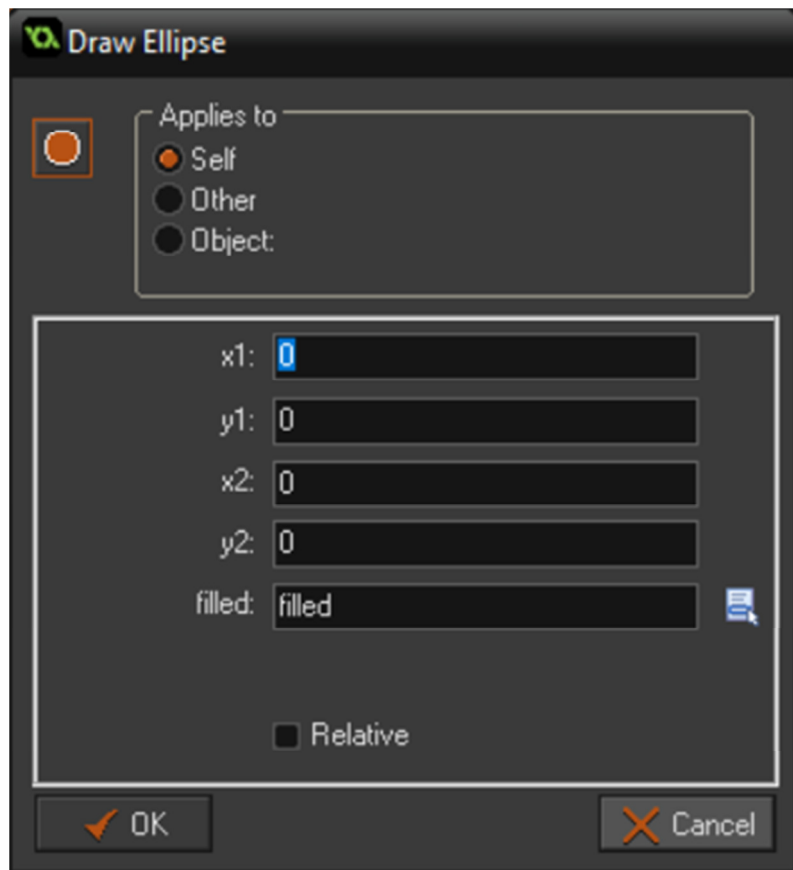


*Horizontal Gradient Action*

Vertical Gradient: Ακριβώς το ίδιο με το Horizontal Gradient αλλά εδώ η εναλλαγή των χρωμάτων γίνεται κάθετα.

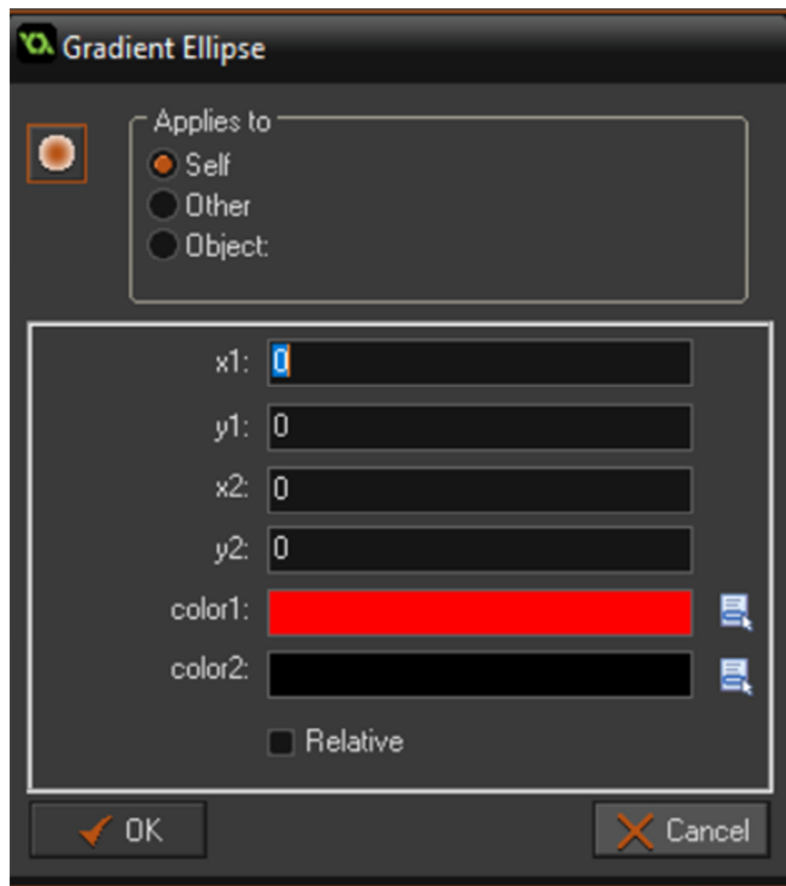


Draw Ellipse : Με αυτό το Action απεικονίσουμε μια έλλειψη , στο σημείο που εμείς θα προσδιορίσουμε.



*Draw Ellipse Action*

Gradient Ellipse : Με αυτό το Action απεικονίσουμε μια έλλειψη , στο σημείο που εμείς θα προσδιορίσουμε, οποία θα έχει άλλο χρώμα στο κέντρο του, στο εσωτερικό του, και διαφορετικό χρώμα εξωτερικά, στο περίγραμμά του.



*Gradient Ellipse Action*

Draw Line : Εδώ δηλώνουμε τις συντεταγμένες σε δύο σημεία ( $x$  ,  $x$ ) για οριζόντια και ( $Y$  ,  $Y$  ) για κάθετα, που θέλουμε να έχει μια ευθεία γραμμή και απεικονίζετε στο δωμάτιο του παιχνιδιού μας.





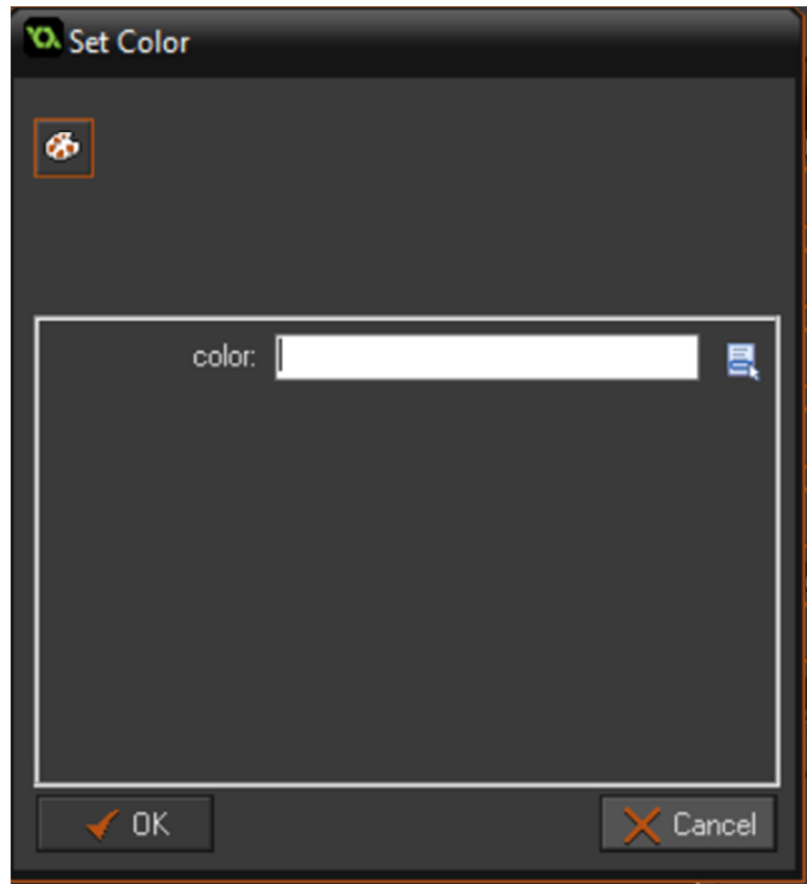
*Draw Line Action*

Draw Arrow : Είναι ακριβός το ίδιο με το Draw Line με την διαφορά ότι αντί για γραμμή φτιάχνουμε βέλος.



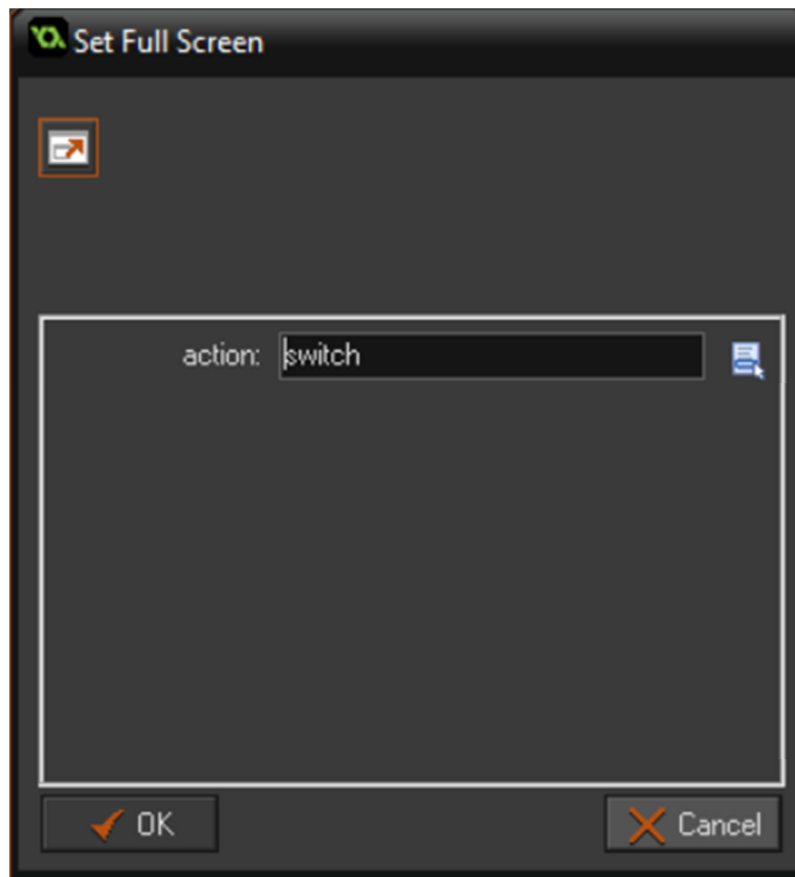
*Draw Line Action*

Set Color : Με αυτό το Action επιλέγουμε μόνο το χρώμα που θα έχουν να γεωμετρικά σχήματα.



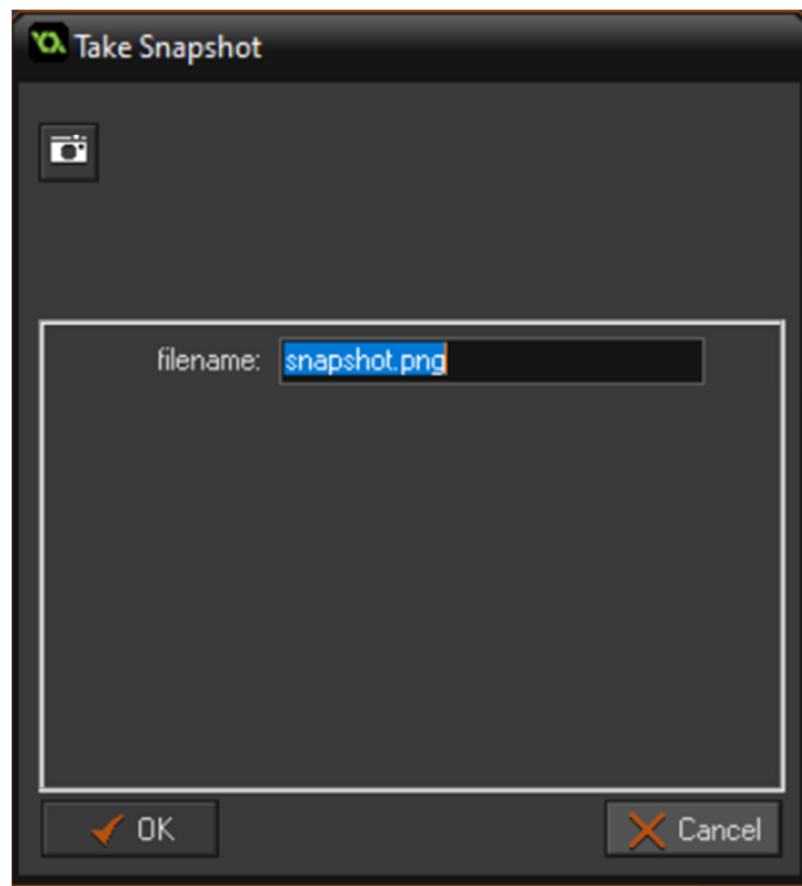
*Set Color Action*

Set Full Screen : Ορίζει αν το παιχνίδι θα τρέχει σε πλήρης οθόνη η σε window mod.



*Set Full Screen Action*

Take Snapshot: Βγάζει ScreenShot.



*Take Snapshot Action*

- Create Effect : Με αυτό το Action μπορούμε να προσθέσουμε εφέ κατά τη διάρκεια του παιχνιδιού μας όπως για μία έκρηξη, βροχή και τα λοιπά. Εμείς απλά επιλέγουμε τον τύπο του εφέ και το σημείο όπως ακόμη, το μέγεθός του, το χρώμα που θέλουμε να έχει και εάν θέλουμε να εμφανιστεί πίσω ή μπροστά από το αντικείμενό στο οποίο ενεργεί αυτό το Action.

Αγγελιδάκης Ν. (2012) Δημιουργία παιχνιδιού με το Game Maker 8.1 Lite

## ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

### GML

- Στο Game Maker Studio έχουμε την δυνατότητα να ορίσουμε τα δικά μας Actions με την χρήση τη γλώσσας του Game Maker Studio ου λέγεται Game Maker Language. Η GML είναι μία γλώσσα προγραμματισμού με αρκετά κοινά στοιχεία με την C++ και άλλες διάσημες γλώσσες. Παρακάτω θα δούμε τα βασικά για την σύνταξή της. Για αρχή ότι γράφουμε στην GML πρέπει να είναι μέσα σε {} και στο τέλος κάθε πρότασης πρέπει να βάζουμε ελληνικό ερωτηματικό. Trollsplatterer (2008) Game Maker Language

```
}  
<statement>;  
<statement>;  
...  
}
```

#### *Ορισμός GML*

Το Game Maker Studio έχει είδη εγκατεστημένες μέσα στις βιβλιοθήκες του μερικές βασικές Functions, φυσικά όμως μας δίνει και την δυνατότητα να δημιουργήσουμε και της δικές μας. Υπάρχουν δύο ειδών προ εγκατεστημένες Functions

1. 2D-Fuctions
2. 3D-Fuctions

Αλφαβητικά οι είδη εγκατεστημένες 2D Functions μέσα στις βιβλιοθήκες του είναι:

**A**

- Abs
  - Arccos
  - Arcsin
  - Arctan
  - Arctan2
  - Audio emitter
- falloff
- Audio falloff set
- model
- Audio play
- sound at

**B**

- Background
- create gradient
- Background
- get texture

**C**

- Ceil
  - Choose
  - Chr
  - Clipboard has
- text
- Collision point
  - Collision
- rectangle
- Cos

**D**

- D3d draw
- block
- D3d draw cone
  - D3d draw
- cylinder
- D3d draw
- ellipsoid
- D3d draw floor
  - D3d draw wall
  - D3d end
  - D3d light

**D cont.**

- D3d
- transform set identity
- D3d
- transform set rotation
- x
- D3d
- transform set rotation
- y
- D3d
- transform set rotation
- z
- D3d
- transform set scaling
- D3d
- transform set
- translation
- D3d vertex
  - D3d vertex
- color
- D3d vertex
- normal
- D3d vertex
- normal color
- D3d vertex
- normal texture
- D3d vertex
- normal texture color
- D3d vertex
- texture
- D3d vertex
- texture color
- Degtograd
  - Display
- mouse get x
- Display
- mouse get y
- Display
- mouse set
- Distance to
- object
- Distance to
- point

**D cont.**

- Draw text
- transformed color
- Draw
- triangle
- Draw
- triangle color
- Draw vertex
  - Draw vertex
- color
- Ds list
- create
- Ds list
- destroy

**E**

- Effect clear
  - Effect create
- above
- Effect create
- below
- Event
- perform
- Event
- perform object
- Event user
  - Execute
- program
- Execute
- shell
- Exp

**F**

- File
- attributes
- File bin
- close
- File bin
- open
- File bin
- position
- File bin read





<ul style="list-style-type: none"> <li>• D3d model</li> </ul>	valign	path
vertex texture color	<ul style="list-style-type: none"> <li>• Draw sprite</li> </ul>	<ul style="list-style-type: none"> <li>• Floor</li> <li>• Frac</li> </ul>
<ul style="list-style-type: none"> <li>• D3d model wall</li> <li>• D3d primitive</li> </ul>	ext	<b>G</b>
begin	<ul style="list-style-type: none"> <li>• Draw sprite</li> </ul>	<ul style="list-style-type: none"> <li>• GML</li> </ul>
<ul style="list-style-type: none"> <li>• D3d primitive</li> </ul>	part	Functions: Particle
begin texture	<ul style="list-style-type: none"> <li>• Draw sprite</li> </ul>	Destroyers
<ul style="list-style-type: none"> <li>• D3d primitive</li> </ul>	stretched	<ul style="list-style-type: none"> <li>• Game end</li> <li>• Game load</li> <li>• Game</li> </ul>
end	<ul style="list-style-type: none"> <li>• Draw sprite</li> </ul>	restart
<ul style="list-style-type: none"> <li>• D3d set culling</li> <li>• D3d set depth</li> <li>• D3d set fog</li> <li>• D3d set hidden</li> <li>• D3d set</li> </ul>	stretched ext	<ul style="list-style-type: none"> <li>• Game save</li> <li>• Get integer</li> <li>• Get string</li> </ul>
lighting	<ul style="list-style-type: none"> <li>• Draw sprite</li> </ul>	
<ul style="list-style-type: none"> <li>• D3d set</li> </ul>	tilted ext	<b>I</b>
perspective	surface	<ul style="list-style-type: none"> <li>• Ini close</li> <li>• Ini key</li> </ul>
<ul style="list-style-type: none"> <li>• D3d set</li> </ul>	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	delete
projection ext	surface ext	<ul style="list-style-type: none"> <li>• Ini key</li> </ul>
<ul style="list-style-type: none"> <li>• D3d set</li> </ul>	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	exists
projection ortho	surface general	<ul style="list-style-type: none"> <li>• Ini open</li> <li>• Ini read real</li> <li>• Ini read</li> </ul>
<ul style="list-style-type: none"> <li>• D3d set</li> </ul>	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	string
projection perspective	surface part	<ul style="list-style-type: none"> <li>• Ini section</li> </ul>
<ul style="list-style-type: none"> <li>• D3d set</li> </ul>	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	delete
shading	surface part ext	<ul style="list-style-type: none"> <li>• Ini section</li> </ul>
<ul style="list-style-type: none"> <li>• D3d start</li> <li>• D3d transform</li> </ul>	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	exists
add rotation x	surface stretched	<ul style="list-style-type: none"> <li>• Ini write real</li> <li>• Ini write</li> </ul>
<ul style="list-style-type: none"> <li>• D3d transform</li> </ul>	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	string
add scaling	surface stretched ext	
<ul style="list-style-type: none"> <li>• D3d transform</li> </ul>	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	
add translation	surface tiled	
	<ul style="list-style-type: none"> <li>• Draw</li> </ul>	
	surface tiled ext	
	<ul style="list-style-type: none"> <li>• Draw text</li> </ul>	
	color	
	<ul style="list-style-type: none"> <li>• Draw text</li> </ul>	
	ext	
	<ul style="list-style-type: none"> <li>• Draw text</li> </ul>	
	ext color	
	<ul style="list-style-type: none"> <li>• Draw text</li> </ul>	
	ext transformed	
	<ul style="list-style-type: none"> <li>• Draw text</li> </ul>	
	ext transformed color	
	<ul style="list-style-type: none"> <li>• Draw text</li> </ul>	
	transformed	



write	• Mplay data	• Round
	• Mplay end	
	<b>S</b>	
redraw	• Screen	
application title	• Set	
message	• Show debug	
	• Show error	
	• Show menu	

*2D Functions GMS*

Trollspalterer (2008) Game Maker Language

Αλφαβητικά οι είδη εγκατεστημένες 3D Functions μέσα στις βιβλιοθήκες του είναι:

<b>D</b>		<b>M cont.</b>		<b>S cont.</b>
block	• D3d draw	draw	• D3d model	• D3d set
cylinder	• D3d draw	floor	• D3d model	• D3d set
floor	• D3d draw	load	• D3d model	perspective
wall	• D3d draw	primitive begin	• D3d model	• D3d start
projection ext	• D3d set	primitive end	• D3d model	<b>T</b>
projection ortho	• D3d set	save	• D3d model	• D3d transform add rotation
projection perspective	• D3d set	vertex	• D3d model	x
shading	• D3d set	vertex color	• D3d model	• D3d transform add scaling
cone	• D3d draw	vertex normal	• D3d model	• D3d transform add
				translation
				• D3d transform set identity

*3D Functions GMS*

Trollspalterer (2008) Game Maker Language

• D3d draw	• D3d model	• D3d
ellipsoid	• D3d model	transform set rotation x
ellipsoid		• D3d
<b>E</b>		transform set rotation y
• D3d end		• D3d
		transform set rotation z
<b>L</b>		• D3d
• D3d light	• D3d model	transform set scaling
define direction	• D3d model	• D3d
• D3d light	vertex texture	transform set
define point	• D3d model	translation
• D3d light	vertex texture color	
enable	• D3d model	<b>V</b>
	wall	• D3d vertex
<b>M</b>		• D3d vertex
• D3d model	<b>P</b>	color
block	begin	• D3d vertex
• D3d model	• D3d primitive	normal
clear	begin texture	• D3d vertex
• D3d model	• D3d primitive	normal color
cone	end	• D3d vertex
• D3d model		normal texture
create	<b>S</b>	• D3d vertex
• D3d model	• D3d set culling	normal texture color
cylinder	• D3d set depth	• D3d vertex
• D3d model	• D3d set fog	texture
destroy	• D3d set	• D3d vertex
	hidden	texture color

3D Functions GMS 1

#### 4.1 LOOPS & IF STATEMENT

Η σύνταξη του IF είναι παρόμοια με άλλες γνωστές γλώσσες προγραμματισμού όπως η C, για παραδείγματα :

```
if (<expression>)
```

```
{
```

```
<statement>
```

```
}
```

```
else
```

```
{
```

```
<statement>
```

```
}
```

**REPEAT STATEMENT:**

Με το repeat εκτελείται συνεχόμενα η έκφραση μέσα στην repeat statement όσες φορές επιλέξουμε.

```
{  
  
repeat (3) instance_create (obj_bat.x, obj_bat.y, obj_ball);  
  
}
```

### **WHILE STATEMENT**

Όσο συμβαίνει και ισχύει κάτι, εκτελείται η Λούπα που περιλαμβάνει την έκφραση που θέλουμε.

```
{  
  
while (!place_free(x,y))  
  
{  
  
x = random(room_width);  
  
y = random(room_height);  
  
}  
  
}
```

### **DO STATEMENT:**

Η έκφραση αυτή μετά από κάθε λούπα ελέγχει να δει αν είναι ακόμα σε ισχύει

```
{  
  
do  
  
{  
  
x = random(room_width);  
  
y = random(room_height);
```

```
}  
  
until (place_free(x,y))  
  
}
```

### **FOR STATEMENT:**

Εδώ δίνουμε κάποια μεταβλητή ορίζουμε τη θα γίνει όταν η τιμή της μεταβλητής πάρει μια συγκεκριμένη τιμή. Κάθε φορά που τρέχει η λούπα η τιμή της μεταβλητής αλλάζει προς την κατεύθυνση που έχουμε ορίσει την ενέργεια που θέλουμε να γίνει.

```
{  
  
for (i=0; i<=19; i+=1) a[i] = i+1;  
  
}
```

### **SWITCH STATEMENT:**

Εδώ φτιάχνουμε περίπτωσης που θα τρέχουν όταν ικανοποιούνται συγκεκριμένα κριτήρια.

Το παρακάτω παράδειγμα είναι μια λούπα εναλλαγή εικόνων background.

```
switch(background_index)  
  
{  
  
case f1:  
  
background_index = f2;  
  
break;
```

```
case f2:

background_index = f3;

break;

case f3:

background_index = f4;

break;

case f4:

background_index = f5;

break;

case f5:

background_index = f6;

break;

case f6:

background_index = f7;

break;

case f7:

background_index = f8;

break;

case f8:

background_index = f1;

break;
```



Κάθε περίπτωση πριν προχωρήσουμε στην επόμενη πρέπει να την τελειώσουμε με `brake`; Trollsplatterer (2008) Game Maker Language

#### 4.2 ΜΕΤΑΒΛΗΤΕΣ

Η GML χρησιμοποιεί όπως και οι άλλες γλώσσες προγραμματισμού μεταβλητές. Υπάρχουν 2 ειδών μεταβλητές. Η τοπικές (Local) και οι Global. Για να ορίσουμε μία τοπική μεταβλητή απλά γράφουμε στον κώδικα μας το όνομα της μεταβλητής. Για παράδειγμα.

```
Variable = 1 ;
```

ή

```
Variable = random(10) ;
```

Η κάθε μεταβλητή που δημιουργούμε ισχύει για το αντικείμενο στο οποίο την δημιουργήσαμε, αν όμως θέλουμε να πειράξουμε μία μεταβλητή που έχουμε φτιάξει σε ένα αντικείμενο μέσα από ένα άλλο αντικείμενο, δηλαδή να την χειριστούμε σαν global μεταβλητή τότε κάνουμε το εξής.

```
Ball.speed = 12;
```

όπου ball είναι το αντικείμενο και speed η μεταβλητή.

Τέλος για να δημιουργήσουμε μία Global μεταβλητή την δηλώνουμε ως εξής.

```
Global.speed=1
```

Αυτό σημαίνει ότι όπου κι αν βρισκόμαστε, θα μπορούμε να καλούμε την μεταβλητή speed.

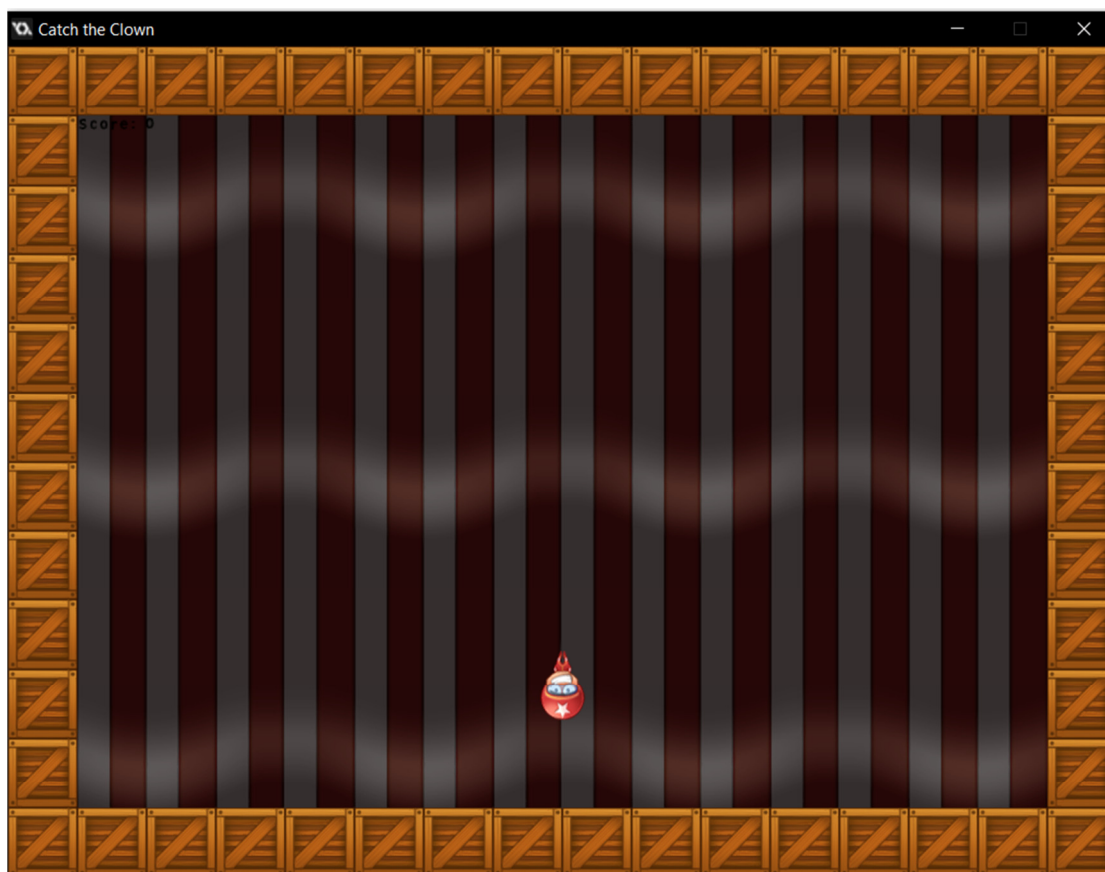
Στο Game Maker υπάρχουν ήδη έτυμες Global μεταβλητές.

X, y, hspeed, vspeed direction, speed, visible, image\_index, image\_speed,  
score, lives, health, mouse\_x, mouse\_y

## ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ

### ΥΛΟΠΟΙΗΣΗ ΠΑΙΧΝΙΔΙΟΥ - CATCH THE CLOWN

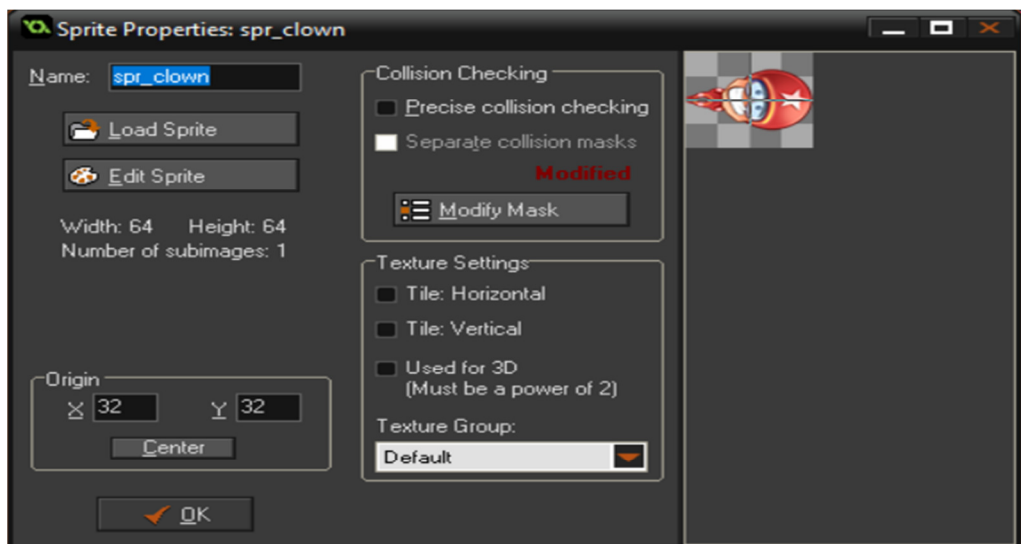
Το συγκεκριμένο παιχνίδι είναι ένα πάρα πολύ μικρό δῆγμα του τι μπορεί να κάνει κάποιος με το Game Maker Studio. Σκοπός του παιχνιδιού είναι να ο παίχτης να κάνει κλικ με το ποντίκι επάνω στον απρόβλεπτα κινούμε Κλόουν και να μαζέψει πόντους. Κάθε φορά που ο παίχτης καταφέρνει να κάνει κλικ επάνω στον κλόουν η ταχύτητά του αυξάνετε και κατά συνέπεια γίνεται πιο δύσκολο το επόμενο κλικ.



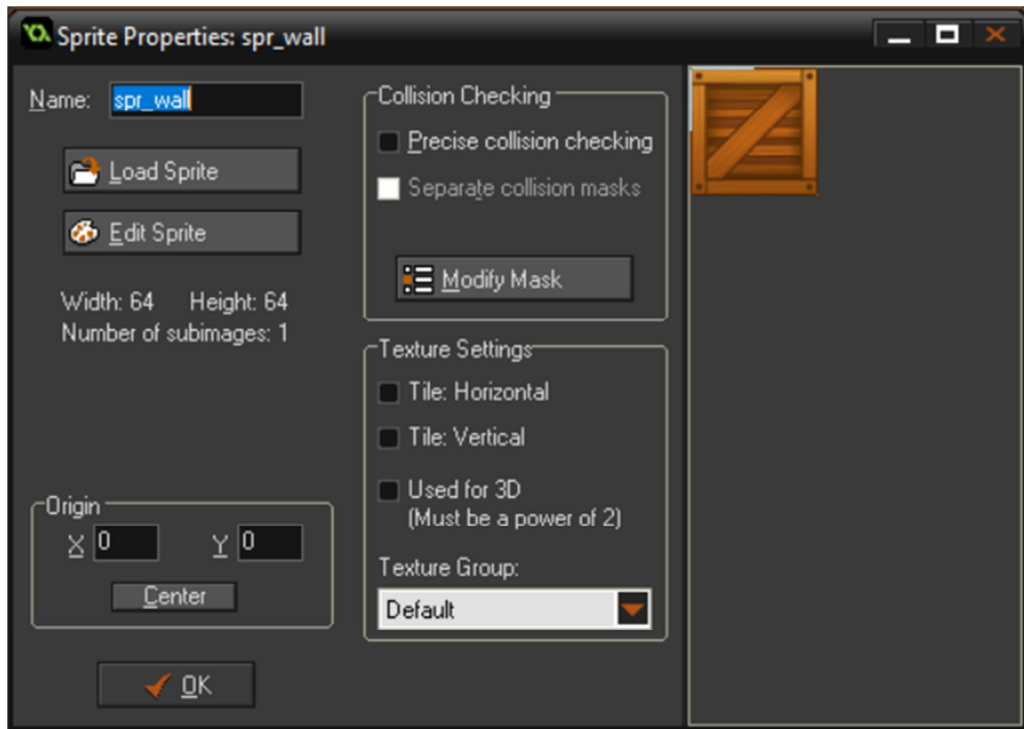
*Catch the Clown Game Screenshot*

Για αρχή φορτώνουμε τα Sprites που θα χρειαστούμε.

1. Το Sprite του κλόουν.
2. Του τείχους του δωματίου



*Sprites Properties For Clown Game*



*Sprites Properties For Clown Game (wall)*

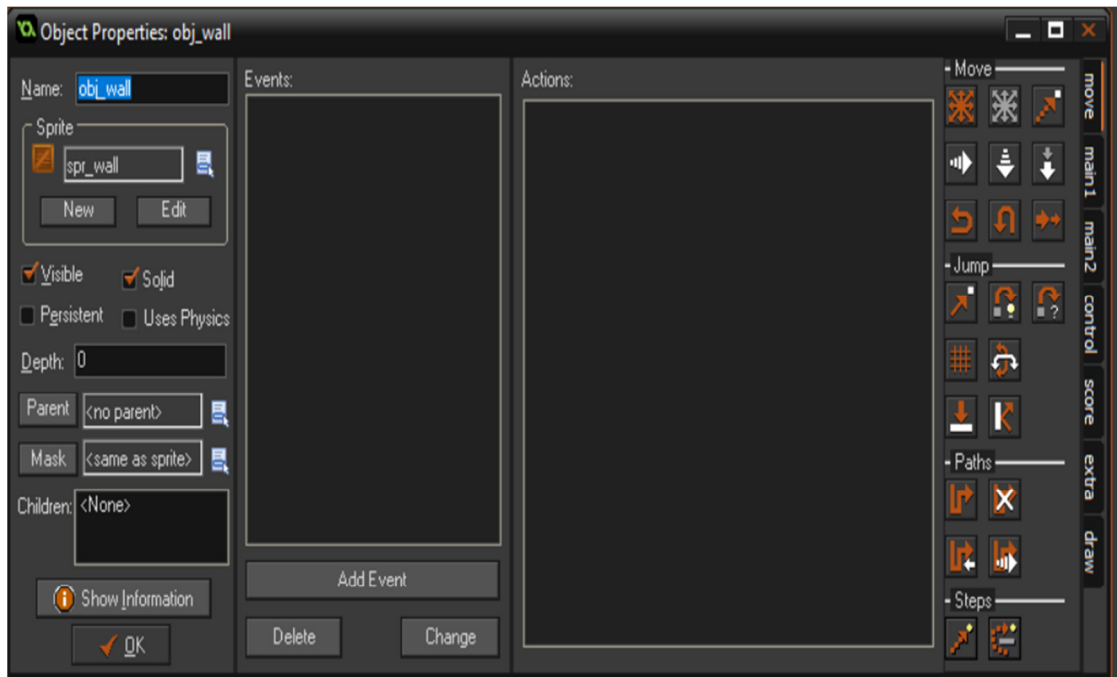
Στην συνέχεια φορτώνουμε ήχους και background.



*Ήχοι Background*

Ακόμα δεν έχουμε δημιουργήσει τίποτα όσον αναφορά το παιχνίδι μας, η πραγματική δημιουργία έρχεται με τα objects και τα Rooms.

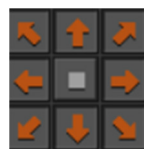
Για αρχή θα φτιάχνουμε το πρώτο object τον τοίχο που θα περιβάλλει το δωμάτιο που θα διαδραματίζετε όλη η δράση.



*Object Wall*

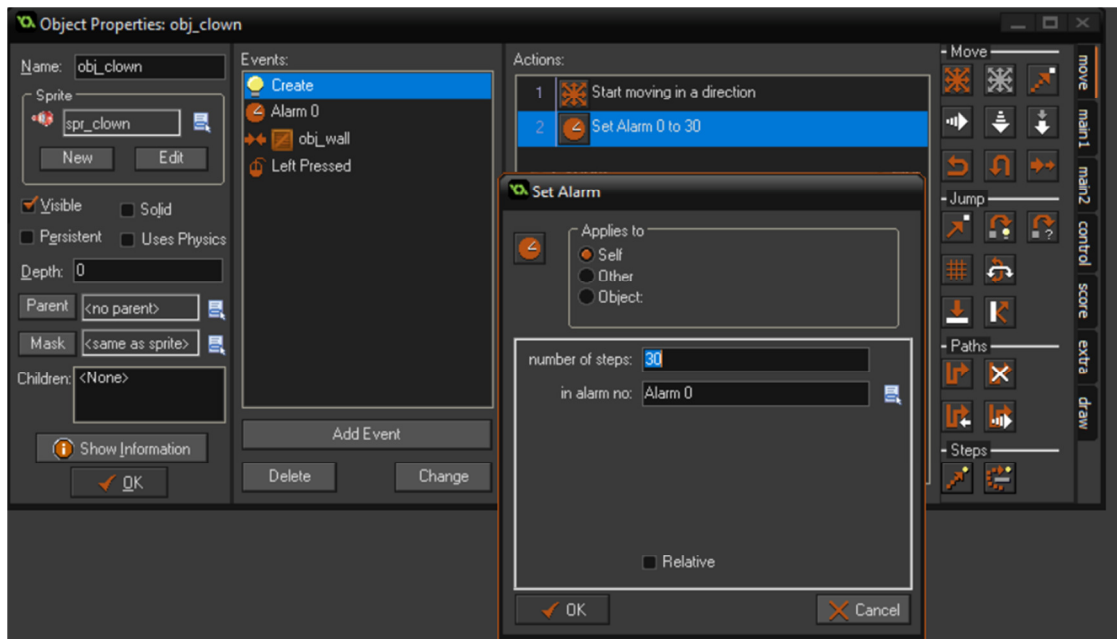
Εδώ δεν χρειάζεται να κάνουμε κάτι περαιτέρω πέραν του να επιλέξουμε της επιλογές Visible ώστε το αντικείμενο να εμφανίζεται μέσα στο δωμάτιο και το Solid ώστε το αντικείμενο να είναι συμπαγές και να μην μπορούν άλλα αντικείμενα να το διαπεράσουν.

Στην συνέχεια θα δημιουργήσουμε τον κλόουν. Πρώτα φτιάχνουμε ένα Create Event και το



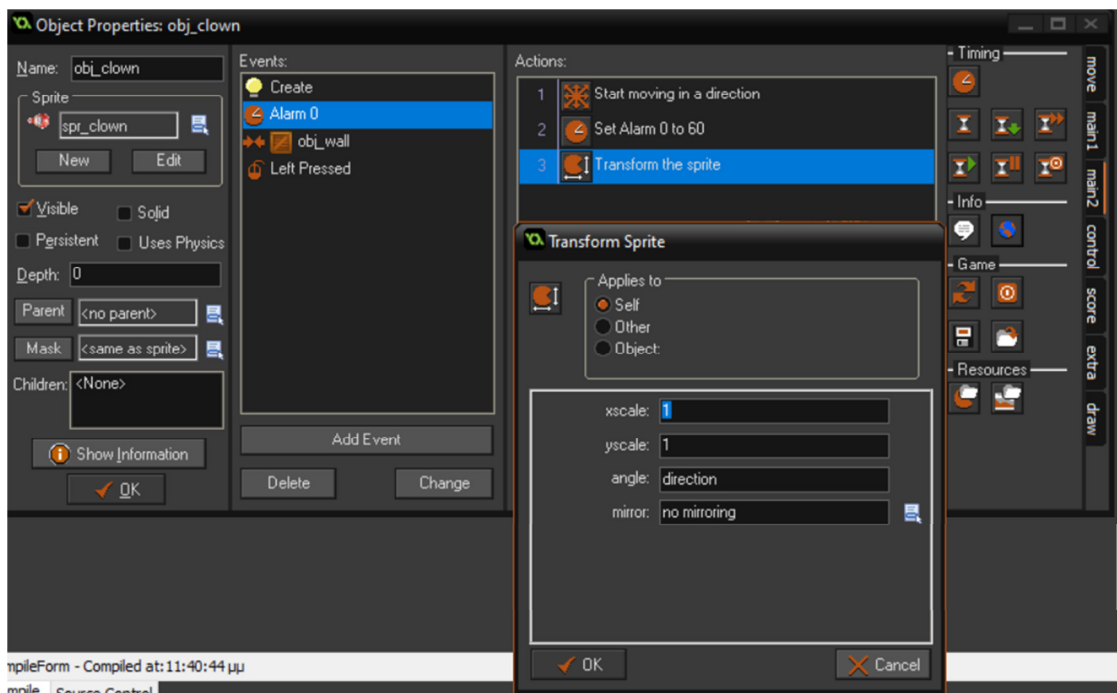
*Create Event for Clown*

«γεμίζουμε» με Alarm, με αριθμώ βημάτων 30 και Start moving in direction όλες της κατεύθυνσης.



#### *Alarm for Clown*

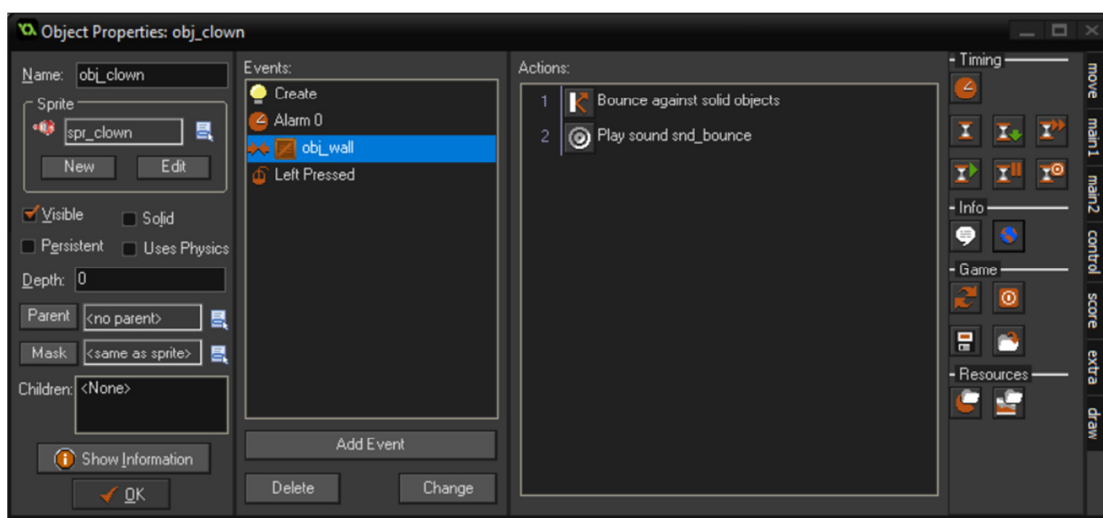
Στην συνέχεια φτιάχνουμε ένα Event Alarm όπου επαναλαμβάνουμε την παραπάνω διαδικασία με την διαφορά ότι ορίζουμε το δεύτερο Alarm στα 60 βήματα, ακόμη προσθέτουμε ένα ακόμη event transform sprite.



#### *Alarm Επανάληψης Διαδικασίας*

Αν αναρωτιέσαι γιατί υπάρχουν δύο Alarm και δυο Start moving in a direction η απάντηση κρύβεται στο Transform the sprite Action. Στην ουσία στο πρώτο λέμε στον κλόουν να κινείται μέσα στο δωμάτιο προς όλες της κατεύθυνσης. Ενώ στο δεύτερο λέμε στο ίδιο το object να αλλάξει κατεύθυνση σε σχέση με των εαυτό του.

Προχωρώντας το έργο δημιουργούμε ένα collision event όπου λέμε στον κλόουν , όταν θα έρχεται σε επαφή με τον τοίχο θα αναπηδάει και θα αναπαράγεται και ο κατάλληλος ήχος. Στο Action play sound να πούμε ότι στην επιλογή loop πρέπει να επιλέγουμε False.

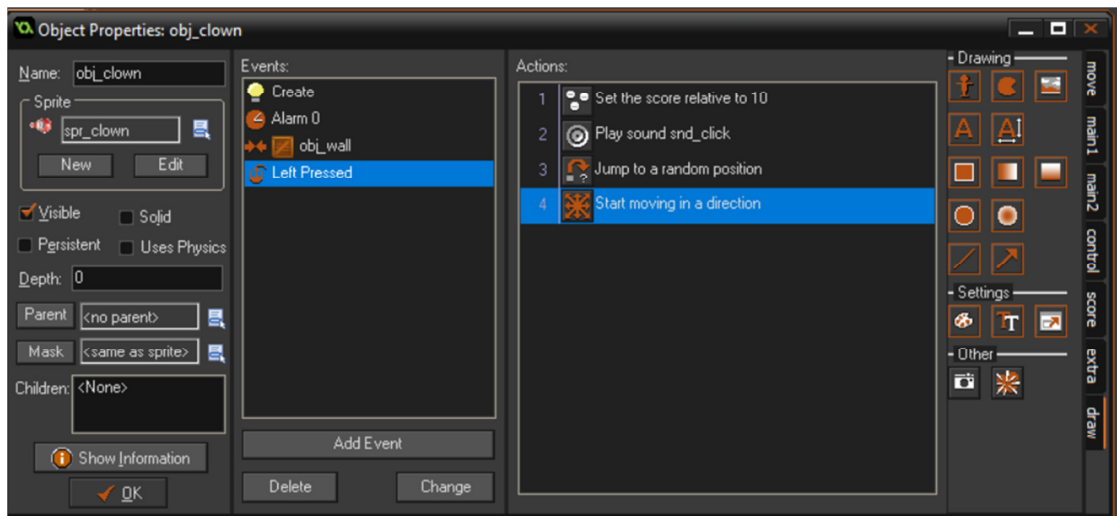


*Επιλογή Loop στο Play sound*

Τέλος «προγραμματίζουμε» της ενέργειες που γίνονται όταν κάνουμε κλικ με το ποντίκι στον κλόουν. Φτιάχνουμε ένα Left pressed Mouse Event και του φορτώνουμε τα παρακάτω Actions.

1. Set score όπου επιλέγουμε το relative
2. Plat sound και επιλέγουμε τον αντίστοιχο ήχο για το κλικ
3. Jump to random petition
4. Start moving in a direction πάλι επιλέγονται όλες της κατεύθυνσης

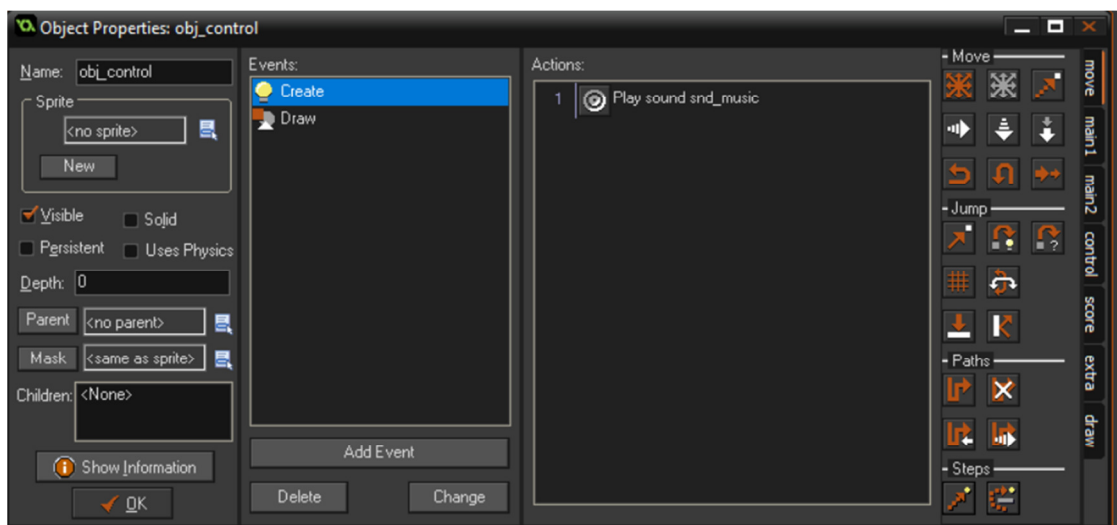




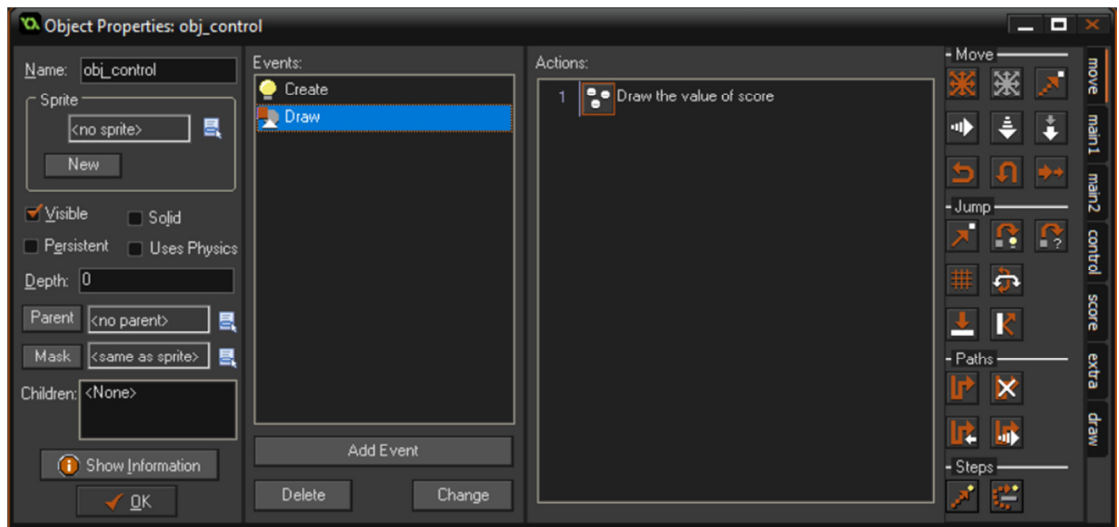
*Επιλογή Κατεύθυνσης Αντικειμένου*

Το τελευταίο object που θα δημιουργήσουμε είναι το obj\_control όπου και θα βάλουμε μουσική υπόκρουση και τρόπο το score να εμφανίζεται στο δωμάτιο σε real time.

1. Πρώτα με Create event και action – play sound και επιλογή του κατάλληλου με loop true
2. Δεύτερο Draw event και action – draw the value of score με της συντεταγμένες που θέλουμε να εμφανίζετε στο δωμάτιο.



*Δημιουργία Event-Action Play Sound και loop true*



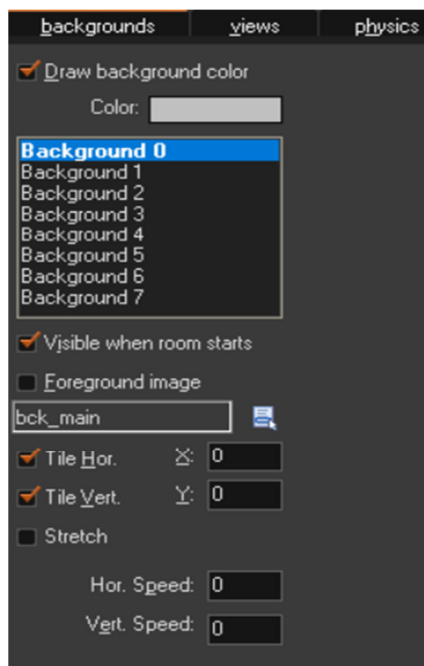
#### *Συντεταγμένες Δωματίου*

Το τελευταίο που θα χρειαστούμε να δημιουργήσουμε πριν ολοκληρώσουμε το παιχνίδι είναι φυσικά το δωμάτιο.



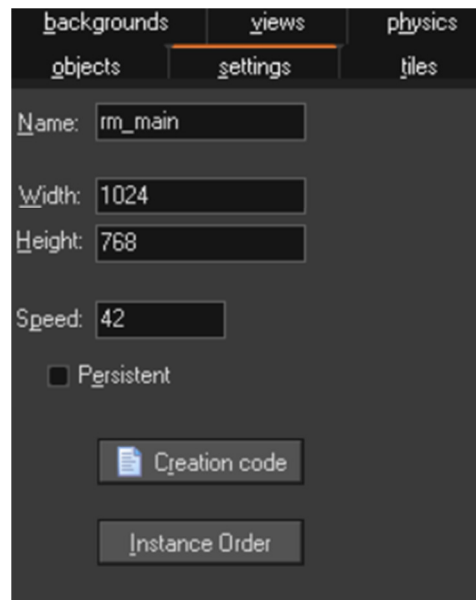
#### *Δημιουργία Δωματίου*

Πρώτα θα επιλέξουμε ένα Background



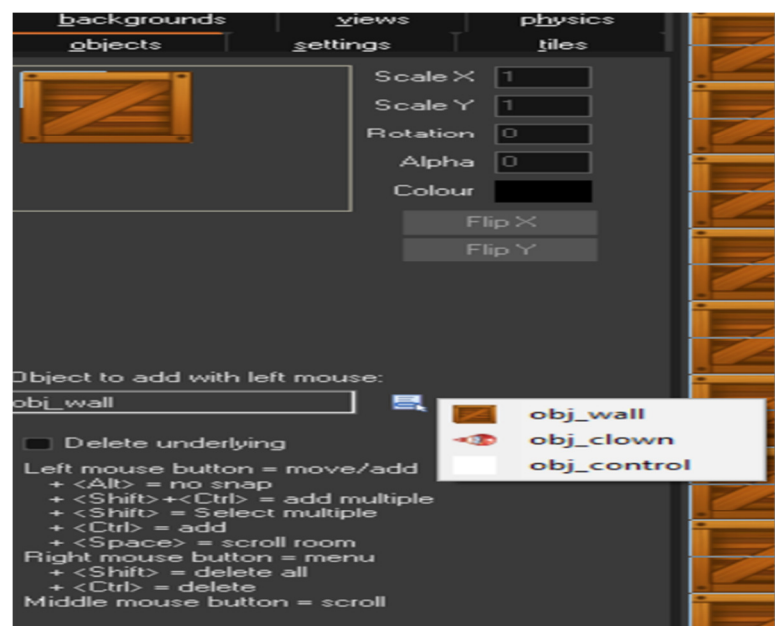
#### *Επιλογή Background*

Συνεχίζουμε με της κατάλληλες ρυθμίσει του δωματίου όπως η ταχύτητα του δωματίου, πλάτος και ύψος .

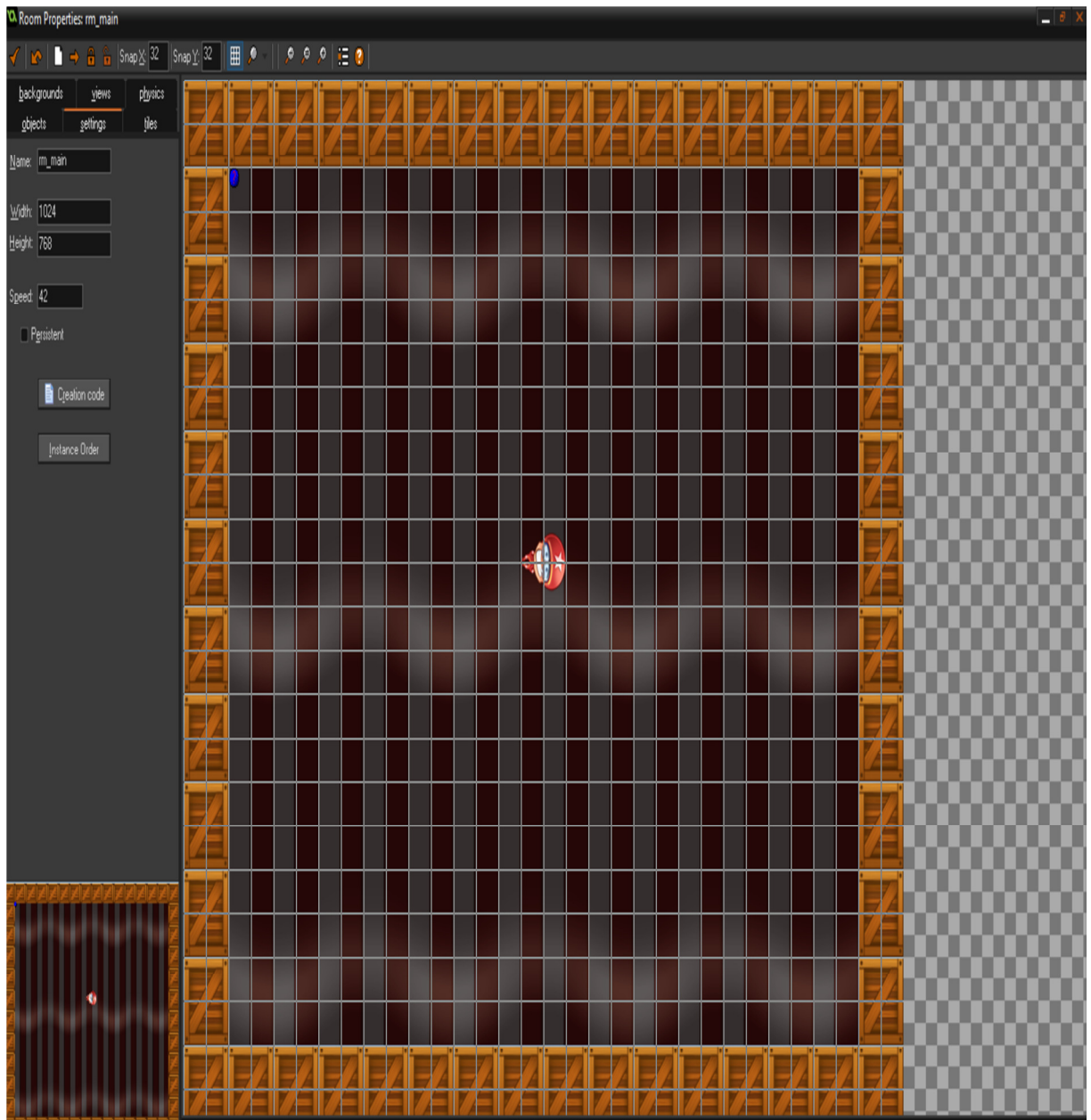


*Ρυθμίσεις Δωματίου*

Ολοκληρώνουμε τοποθετώντας τα αντικείμενα ένα στο δωμάτιο από το Tab Object



*Τοποθέτηση Αντικειμένου στο Δωμάτιο*



#### *Ολοκλήρωση Δωματίου*

Φτάνοντας σε αυτό το σημείο το παιχνίδι μας είναι έτομο για να το τρέξουμε και να δούμε αν λειτουργεί όπως θα έπρεπε η όχι και να κάνουμε και το ανάλογο debugging.

## ΕΠΙΛΟΓΟΣ

Κλείνοντας την παρούσα πτυχιακή εργασία, θεωρώ πως τα στοιχεία και οι λεπτομέρειες που χρειάζεται να γνωρίζει κάποιος ο οποίος θέλει να εμφυσήσει στα ηλεκτρονικά παιχνίδια και κυρίως στη δημιουργία αυτών μέσω του Game Maker Studio έχουν αναλυθεί επαρκώς. Οι δυνατότητες του συγκεκριμένου προγράμματος δεν είναι ιδιαίτερα μεγάλες στο χώρο του 3D gaming, οι δυνατότητές μου όμως στο 2D gaming είναι αρκετά μεγάλες.

Κατά την άποψή μου οι πληροφορίες που αποκτήθηκαν είναι αρκετές ούτως ώστε να γίνει η πρώτη επαφή με τον κόσμο των βιντεοπαιχνιδιών ενός ανθρώπου ο οποίος επιθυμεί να ασχοληθεί μαζί τους ως game developer κυρίως εάν ο ενδιαφερόμενος είναι αρχάριος στον συγκεκριμένο χώρο.

Εν κατακλείδι, το Game Maker Studio, είναι ένα πολύ χρήσιμο και ενδιαφέρον πρόγραμμα ανάπτυξης παιχνιδιών και είναι ένα πολύ καλό πρώτο βήμα για όποιον ενδιαφέρεται για τον συγκεκριμένο τομέα.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΕΣ ΠΗΓΕΣ**

### **ΒΙΒΛΙΑ**

- Michael Rohde (2014) Game Maker: Studio for Dummies
- Habgood J. Overmars M. (2006) The Game Maker's Apprentice: Game Development for Beginners
- Lee Ford J. (2009) Getting Started with Game maker Studio
- Swamy Nanu, Swamy Naveena (2006) Basic Game Design & Creation for Fun & Learning

### **E-BOOKS**

- Αγγελιδάκης N. (2012) Δημιουργία παιχνιδιού με το Game Maker 8.1 Lite
- Bailey M. (2011) Using Game Maker 8
- Overmars M. (2004) Designing Games with Game Maker version 5.3
- Trollsplatterer (2008) Game Maker Language

### **WEB PAGES**

- <https://docs2.yoyogames.com/>
- <https://docs.yoyogames.com/source/dadiospice/index.html>
- <https://www.youtube.com/user/letslearngamemaker>

**Copyright © ΤΕΙ Δυτικής Ελλάδας. Με επιφύλαξη παντός δικαιώματος.  
All rights reserved.**

**Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1988 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον.**