



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Εκμάθηση προγραμματισμού μέσω της ανάπτυξης  
παιχνιδιού**

**Ευάγγελος Α. Αρβανιτόπουλος**

**Επιβλέπουσες: Μαρία Γρηγοριάδου, Ομότιμη Καθηγήτρια  
Αγορίτσα Γόγουλου, Ε.Δ.Ι.Π.**

**ΑΘΗΝΑ**

**ΝΟΕΜΒΡΙΟΣ 2018**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Εκμάθηση προγραμματισμού μέσω της ανάπτυξης παιχνιδιού

**Ευάγγελος Α. Αρβανιτόπουλος**

**A.M.: 1115201000165**

**ΕΠΙΒΛΕΠΟΥΣΕΣ:** **Μαρία Γρηγοριάδου, Ομότιμη Καθηγήτρια**  
**Αγορίτσα Γόγουλου, Ε.Δ.Ι.Π.**

## ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία αποσκοπεί στην εκμάθηση προγραμματιστικών εννοιών και εννοιών ανάπτυξης ενός παιχνιδιού μέσω της δημιουργίας ενός παιχνιδιού αξιοποιώντας το περιβάλλον Game Maker. Αξιοποιώντας αποτελέσματα της βιβλιογραφίας και αρχές της διδακτικής του προγραμματισμού, η προτεινόμενη προσέγγιση αφορά στον σχεδιασμό Φύλλων Εργασίας για την ανάπτυξη του παιχνιδιού. Τα Φύλλα Εργασίας ακολουθούν συγκεκριμένες αρχές όπως γνωστοποίηση μαθησιακών στόχων, διαγραμματική απεικόνιση της ακολουθίας των ενεργειών για την επίτευξη των στόχων, κινητοποίηση του μαθητή, καλλιέργεια της αναλυτικής και κριτικής του σκέψης και καλλιέργεια δεξιοτήτων στην ανάπτυξη προγραμμάτων και παιχνιδιών θέτοντας ερωτήσεις, παροχή μαθησιακού υλικού.

Στο πλαίσιο της εργασίας έγινε πιλοτική εφαρμογή των δύο από τα έξι φύλλα εργασίας, που σχεδιάστηκαν, σε φοιτητές του Τμήματος. Τα αποτελέσματα δείχνουν τη θετική αποδοχή του συγκεκριμένου σχεδιασμού των Φύλλων Εργασίας και επιβεβαιώνουν ότι η καθοδηγούμενη προσέγγιση στην εκμάθηση προγραμματισμού μπορεί να βοηθήσει τους αρχάριους προγραμματιστές.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:** Τεχνολογία - Πληροφορική

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Εκπαιδευτικά παιχνίδια, GameMaker Studio 2, εκμάθηση προγραμματισμού παιχνιδιών, βασικές προγραμματιστικές έννοιες, έννοιες προγραμματισμού παιχνιδιών

## **ABSTRACT**

This thesis focuses on the learning of programming concepts and on game development by creating a game using the Game Maker environment. Taking advantage on the results of the literature and the principles of didactics of programming, the proposed approach concerns the design of learning activities for the development of the game. The activities follow specific principles such as informing the learner about the learning objectives, diagrammatic representation of the sequence of actions needed to achieve the goals, motivating the student, cultivating analytical and critical thinking, and cultivating skills in developing programs and games and providing suitable learning material.

In the framework of the thesis, two of the six designed activities that were implemented, were used by the students of the Department. The results show the positive attitude towards the proposed design of the activities and confirm that a guided approach to learning programming can help novice programmers.

**SUBJECT AREA:** Technology - Informatics

**KEYWORDS:** Educational games, GameMaker Studio 2, learning game programming, basic programming concepts, concepts of programming games

# ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΠΡΟΛΟΓΟΣ</b> .....	<b>8</b>
<b>1. ΕΚΜΑΘΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕΣΩ ΠΑΙΧΝΙΔΙΩΝ</b> .....	<b>9</b>
1.1 Εισαγωγή .....	9
1.2 Κατηγοριοποίηση των Παιχνιδιών.....	9
1.3 Διαδικτυακές Πλατφόρμες για Εκμάθηση προγραμματισμού.....	10
1.4 Παραδείγματα εκμάθησης προγραμματισμού μέσω της ανάπτυξης παιχνιδιών .....	11
<b>2. ΑΝΑΠΤΥΣΣΟΝΤΑΣ ΤΟ ΠΑΙΧΝΙΔΙ EXPLORER</b> .....	<b>12</b>
2.1 Αρχές εκμάθησης προγραμματισμού μέσω της ανάπτυξης ενός παιχνιδιού .....	12
2.2 Τα Φύλλα Εργασίας .....	15
2.3 Μαθησιακοί Στόχοι.....	15
2.4 Οδηγίες και Προετοιμασία .....	17
2.5 Το παιχνίδι Explorer.....	17
2.6 Περιβάλλον GameMaker Studio .....	17
2.6.1 Δυνατότητες.....	17
2.6.2 Ιστορικό.....	18
2.6.3 Πώς χρησιμοποιείται .....	18
<b>3. ΠΙΛΟΤΙΚΗ ΕΦΑΡΜΟΓΗ ΣΕ ΦΟΙΤΗΤΕΣ</b> .....	<b>19</b>
3.1 Το ερωτηματολόγιο .....	19
3.2 Τα αποτελέσματα .....	20
<b>4. ΕΠΙΛΟΓΟΣ</b> .....	<b>27</b>
<b>ΠΑΡΑΡΤΗΜΑ</b> .....	<b>28</b>
<b>ΑΝΑΦΟΡΕΣ</b> .....	<b>151</b>

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Ερώτηση σχετικά με τον στόχο του 1 <sup>ου</sup> Φύλλου Εργασίας .....	21
Σχήμα 2: Ερώτηση σχετικά με τις οδηγίες του 1 <sup>ου</sup> Φύλλου Εργασίας .....	21
Σχήμα 3: Ερώτηση σχετικά με το επίπεδο δυσκολίας του 1 <sup>ου</sup> Φύλλου Εργασίας .....	22
Σχήμα 4: Ερώτηση σχετικά με το στόχο του 2 <sup>ου</sup> Φύλλου Εργασίας .....	22
Σχήμα 5: Ερώτηση σχετικά με τις οδηγίες του 2 <sup>ου</sup> Φύλλου Εργασίας .....	23
Σχήμα 6: Ερώτηση σχετικά με το επίπεδο δυσκολίας του 2 <sup>ου</sup> Φύλλου Εργασίας .....	23
Σχήμα 7: Ερώτηση σχετικά με το επεξηγηματικό υλικό .....	24
Σχήμα 8: Ερώτηση σχετικά με την οργάνωση των Φύλλων Εργασίας .....	24
Σχήμα 9: Ερώτηση σχετικά με την χρήση εικονιδίων .....	25
Σχήμα 10: Ερώτηση σχετικά με τη χρήση χρωμάτων στο κείμενο .....	25
Σχήμα 11: Ερώτηση σχετικά με το Game Maker ως περιβάλλον ανάπτυξης παιχνιδιών .....	26

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Ενδεικτικοί στόχοι φύλλου Εργασίας .....	12
Εικόνα 2: Σύνδεση των δραστηριοτήτων σε επίπεδο στόχων ανάπτυξης και προγραμματιστικών εννοιών εκμάθησης .....	12
Εικόνα 3: Ακολουθία ενεργειών για την επίτευξη των στόχων του Φύλλου Εργασίας ...	13
Εικόνα 4: Καθορισμός ενδιάμεσου στόχου .....	13
Εικόνα 5: Ερωτήσεις κινητοποίησης και προβληματισμού .....	13
Εικόνα 6: Μαθησιακό υλικό .....	14
Εικόνα 7: Ερωτήσεις προς απάντηση μετά την ολοκλήρωση του Φύλλου Εργασίας ....	14
Εικόνα 8: Νέοι στόχοι-προκλήσεις προς επίλυση .....	14

## **ΠΡΟΛΟΓΟΣ**

Η πτυχιακή αυτή εργασία διενεργήθηκε στο πλαίσιο των προπτυχιακών σπουδών του τμήματος Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ. Η δημιουργία του εκπαιδευτικού υλικού έγινε με χρήση του Google Documents και Microsoft Word και βασίζεται στην εκπαιδευτική πλατφόρμα GameMaker Studio 2. Ευχαριστώ πολύ την καθηγήτριά μου κ. Γόγουλου για τη βοήθεια και την στήριξή της.



## 1. ΕΚΜΑΘΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕΣΩ ΠΑΙΧΝΙΔΙΩΝ

### 1.1 Εισαγωγή

Τα παιδιά πάντα έπαιζαν παιχνίδια και μάθαιναν μέσα από αυτά. Όσο μεγαλώνουν επιλέγουν να παίζουν παιχνίδια με κανόνες παρά ελεύθερα. Παιχνίδια με κάρτες, επιτραπέζια και βιντεοπαιχνίδια κερδίζουν ολοένα και περισσότερο τον χρόνο των παιδιών. Το ίδιο το παιχνίδι μπορεί λοιπόν να αποτελέσει εργαλείο στην εκπαίδευση και ειδικά στον προγραμματισμό.

Σήμερα, οι άνθρωποι χρησιμοποιούν υπολογιστές και άλλες ψηφιακές συσκευές στην καθημερινή τους ζωή για δουλειά ή διασκέδαση. Τα παιδιά γεννιούνται σε ένα ψηφιακό κόσμο και μαθαίνουν από πολύ μικρή ηλικία να χειρίζονται ακόμα και προγράμματα πλοήγησης (web browsers). Ωστόσο συχνά δεν γνωρίζουν πως λειτουργούν τα προγράμματα που χρησιμοποιούν ή πως δημιουργούνται. Για να έχουν αυτή τη γνώση καθώς και για να προσεγγιστεί καλύτερα το αντικείμενο της πληροφορικής είναι σκόπιμο να αντιμετωπίσουν τον προγραμματισμό και να έρθουν σε επαφή με τις βασικές δομές αλλά και τον τρόπο σκέψης του.

Μέσα από τη δημιουργία παιχνιδιών λοιπόν τα παιδιά έχουν την ευκαιρία να μπουν στον κόσμο του προγραμματισμού παίζοντας, και μαθαίνοντας καλύτερα τον τρόπο με τον οποίο λειτουργούν οι εφαρμογές που αγαπούν και χρησιμοποιούν στην καθημερινή τους ζωή.

Η μάθηση βασισμένη στο παιχνίδι (game-based learning) αφορά στην αξιοποίηση των παιχνιδιών για εκπαιδευτικούς σκοπούς. Ερευνητές επισημαίνουν θετικά στοιχεία από την αξιοποίηση των παιχνιδιών ως μαθησιακά εργαλεία: κινητοποίηση, πρόκληση ενδιαφέροντος και εμπλοκή των μαθητών, απόκτηση εμπειριών και ανατροφοδότηση όσον αφορά την επίδοση [1], [2]. Πολλά εκπαιδευτικά περιβάλλοντα χρησιμοποιούν τεχνικές παιχνιδιών (gamification) που περιλαμβάνουν επίτευξη στόχων, ανατροφοδότηση επιβράβευση [3].

### 1.2 Κατηγοριοποίηση των Παιχνιδιών

Τα παιχνίδια μπορούν να κατηγοριοποιηθούν με βάση τα παρακάτω κριτήρια [1]:

**Στόχος-Application Area.** Μπορούμε να χωρίσουμε τα παιχνίδια με βάση τον στόχο τους στην εκπαίδευση. Για παράδειγμα εάν στοχεύουν στη μάθηση σε συγκεκριμένες έννοιες, εάν εστιάζουν στην κατάρτιση, στη διαπροσωπική επικοινωνία, σε θέματα φροντίδας/υγείας.

**Δυνατότητα-Modality.** Τα παιχνίδια χωρίζονται σε σχέση με το πως πραγματοποιούν την αλληλεπίδραση με τον χρήστη, π.χ. οπτικά, ακουστικά ή με επαφή.

**Μορφή Διεπαφής-Interaction Style.** Τα παιχνίδια μπορούν να χωριστούν με βάση το αν προσφέρουν διεπαφή μέσω πληκτρολογίου, ποντικιού, joystick ή συσκευές εικονικής πραγματικότητας.

**Περιβάλλον-Environment.** Ανάλογα με το περιβάλλον του παιχνιδιού, αυτό μπορεί να περιλαμβάνει: κοινωνική παρουσία, εικονικό περιβάλλον, 2D ή 3D αναπαράσταση, κινητικότητα και χρήση του διαδικτύου.

**Εκπαιδευτική Προσέγγιση-Learning Approach.** Αυτή η κατηγορία αφορά μόνο τα εκπαιδευτικά παιχνίδια και δηλώνει εάν το παιχνίδι παίζεται από έναν παίκτη, από πολλούς, εάν είναι μικτό, δηλ. κάποιο τμήμα του παιχνιδιού από έναν παίκτη και ένα

άλλον από πολλούς, εάν απαιτεί συνεργασία μεταξύ των παικτών ή υποστήριξη από τον εκπαιδευτικό.

### 1.3 Διαδικτυακές Πλατφόρμες για Εκμάθηση προγραμματισμού

Υπάρχουν πολλών ειδών Διαδικτυακές πλατφόρμες για εκμάθηση προγραμματισμού. Οι πιο συνηθισμένες εκτελούν αυτόματα τον κώδικα και δίνουν feedback [1]. Παρέχουν μαθήματα που μπορεί ο εκπαιδευόμενος να χρησιμοποιήσει και να αναπτύξει τις δεξιότητες του πάνω σε κάποιο αντικείμενο του προγραμματισμού. Όσον αφορά τις Διαδικτυακές πλατφόρμες που βασίζονται στα παιχνίδια μπορούν να ταξινομηθούν σε τρεις κατηγορίες ανάλογα με τον σκοπό τους: εκμάθηση κώδικα, εκμάθηση αλγοριθμικών εννοιών, και εκμάθηση προγραμματισμού παιχνιδιών [1].

**Εκμάθηση κώδικα.** Σε αυτήν την κατηγορία εντάσσονται οι πλατφόρμες που αποσκοπούν στην δυνατότητα των εκπαιδευόμενων να προγραμματίζουν βάση κάποιας συγκεκριμένης γλώσσας προγραμματισμού, και να μπορούν να λύνουν προβλήματα μέσω αυτής. Οι δραστηριότητες που υπάρχουν σε αυτές τις πλατφόρμες περιλαμβάνουν:

- Αποσφαλμάτωση. Δίνεται στον χρήστη ένα πρόγραμμα με λάθη για να διορθώσει.
- Συμπλήρωση. Δίνεται στον χρήστη ένα ημιτελές πρόγραμμα και καλείται να το συμπληρώσει.
- Κατασκευή προγράμματος. Σε αυτήν την περίπτωση ο χρήστης πρέπει να δημιουργήσει το δικό του ολοκληρωμένο πρόγραμμα ή συνάρτηση ή εντολή.
- Κατασκευή όμοιου προγράμματος. Δίνεται η εκτέλεση ενός προγράμματος και ο χρήστης προσπαθεί να κατασκευάσει ένα δικό του που να επιτελεί αυτήν την λειτουργία.

**Εκμάθηση αλγοριθμικών εννοιών.** Οι πλατφόρμες αυτές αποσκοπούν στην κατανόηση των αλγοριθμικών εννοιών καθώς και στη γενικότερη καλλιέργεια της αλγοριθμικής σκέψης στους μαθητές.

Ένα παράδειγμα είναι η πλατφόρμα LightBot (<http://lightbot.com>) που στοχεύει στην κατανόηση της αναδρομής μέσα από Puzzle. Για κάθε τέτοιο Puzzle ο μαθητής καλείται να κατασκευάσει ένα πρόγραμμα χρησιμοποιώντας Blocks, με σκοπό να καθοδηγήσει ένα Robot προς ένα συγκεκριμένο στόχο. Ένα άλλο παράδειγμα είναι το Initial Conditions όπου ο μαθητής καλείται να δώσει λύση σε κάποιο πρόβλημα αναζήτησης. Αυτού του είδους τα προβλήματα υπάρχουν στην τεχνητή νοημοσύνη και λύνονται με εξαντλητικούς αλγορίθμους. Οι βασικές έννοιες των αλγορίθμων και η ανάπτυξη αλγοριθμικής σκέψης είναι απαραίτητο συστατικό στην εκμάθηση του προγραμματισμού.

**Εκμάθηση προγραμματισμού παιχνιδιών.** Τέλος υπάρχει η κατηγορία στην οποία ανήκουν πλατφόρμες οι οποίες προσφέρουν στους χρήστες την δυνατότητα να προγραμματίζουν παιχνίδια. Πρόκειται για οπτικό προγραμματισμό και παρέχει εντολές σε μπλοκ τα οποία καθορίζουν την συμπεριφορά των Sprites.

Ένα παράδειγμα πλατφόρμας αυτής της κατηγορίας είναι το Scratch. Το βασικό παράθυρο του Scratch περιλαμβάνει τμήμα για το παιχνίδι και τμήμα με τις εντολές που υλοποιούν τη συμπεριφορά των sprites που χρησιμοποιούνται.

Σε αυτήν την κατηγορία εντάσσεται και το Game Maker που ουσιαστικά κάνει χρήση αντικειμένων, γεγονότων και ενεργειών. Γεγονότα όπως το πάτημα ενός κουμπιού, η σύγκρουση μεταξύ αντικειμένων ή το κλικ του ποντικιού προκαλούν την εκτέλεση

ενεργειών όπως η δημιουργία ενός αντικειμένου, η μετακίνηση του αντικειμένου μέσα στο χώρο του παιχνιδιού ή η δημιουργία ή αλλαγή μιας μεταβλητής.

Σε αυτές τις πλατφόρμες χρειάζεται ο μαθητής να μπορεί να προγραμματίσει αλλά όχι απαραίτητα να γράφει κώδικα. Στόχος είναι να αναπτύξει τη δημιουργικότητά του και ταυτόχρονα να γνωρίσει τον τρόπο με τον οποίο κατασκευάζονται τα παιχνίδια.

#### **1.4 Παραδείγματα εκμάθησης προγραμματισμού μέσω της ανάπτυξης παιχνιδιών**

Τα τελευταία χρόνια πλατφόρμες ανάπτυξης παιχνιδιών χρησιμοποιούνται για την εκμάθηση βασικών προγραμματιστικών εννοιών. Ανάλογα με τους επιδιωκόμενους μαθησιακούς στόχους και το επίπεδο των μαθητών μπορούν να χρησιμοποιηθούν διαφορετικές πλατφόρμες.

Στο [4] αναφέρεται η αξιοποίηση του περιβάλλοντος Alice για την εκμάθηση βασικών εννοιών του αντικειμενοστραφούς προγραμματισμού σε γλώσσα Java. Θεωρείται κατάλληλο για ηλικίες άνω των 8 χρόνων. Στο πλαίσιο της πρότασή τους, οι μαθητές αναπτύσσουν ένα παιχνίδι-ιστορία στο περιβάλλον Alice υπό την καθοδήγηση του εκπαιδευτικού. Τα αποτελέσματα της συγκεκριμένης παρέμβασης δείχνουν ότι οι μαθητές θεωρούν ότι το περιβάλλον βοήθησε στο να μάθουν βασικές έννοιες για το αντικειμενοστραφή προγραμματισμό μέσα σε ένα ευχάριστο και ευκολότερο προγραμματιστικό περιβάλλον.

Στο [5], η Johnson αξιοποίησε το προγραμματιστικό περιβάλλον Game Maker στο πλαίσιο της θεωρίας του κονστραξιονισμού σε μαθητές ηλικίας 13-14 χρόνων. Οι μαθητές εργάστηκαν σε ζευγάρια με στόχο την ανάπτυξη ενός παιχνιδιού. Με βάση τα αποτελέσματα της έρευνας, προέκυψε ότι οι μαθητές αντιμετώπισαν δυσκολίες στην ανάλυση ενός προβλήματος και στην αλγοριθμική σκέψη και παρανόησαν τη σημασία λέξεων όπως «γεγονός» (event), «ενέργεια» (action) ή «δωμάτιο» (room). Τα συμπεράσματα καταλήγουν στο ότι οι μαθητές χρειάζονται υποστήριξη όσον αφορά στους μηχανισμούς παιχνιδιού που απαιτούνται αλλά και στις προγραμματιστικές δομές που χρειάζονται για την υλοποίηση αυτών.

## 2. ΑΝΑΠΤΥΣΣΟΝΤΑΣ ΤΟ ΠΑΙΧΝΙΔΙ EXPLORER

### 2.1 Αρχές εκμάθησης προγραμματισμού μέσω της ανάπτυξης ενός παιχνιδιού

Στο πλαίσιο της παρούσας εργασίας επιχειρήθηκε η αξιοποίηση του περιβάλλοντος Game Maker για την εκμάθηση βασικών εννοιών προγραμματισμού καθώς και ανάπτυξης παιχνιδιών. Λαμβάνοντας υπόψη τα πορίσματα της βιβλιογραφίας [5] η πρότασή μας βασίστηκε στο σχεδιασμό μίας ακολουθίας Φύλλων Εργασίας τα οποία καθοδηγούν τους μαθητές στην ανάπτυξη του παιχνιδιού «Explorer» και στην εκμάθηση του προγραμματισμού.

Συγκεκριμένα, κάθε Φύλλο Εργασίας ...

- αναφέρει τους στόχους που σχετίζονται με προγραμματιστικές έννοιες και έννοιες ανάπτυξης παιχνιδιού (Εικόνα 1).



Αλγοριθμικές έννοιες:

Μεταβλητή (Αρχικοποίηση και Εμφάνιση), Απλή Δομή Επιλογής

Έννοιες Προγραμματισμού Παιχνιδιών:

Γεγονός Σύγκρουσης (Collision Event), Μεταβλητή Σκορ

Εικόνα 1: Ενδεικτικοί στόχοι φύλλου Εργασίας

- παρέχει σύνδεση με όσα ήδη ο μαθητής έχει δημιουργήσει και έχει μάθει (Εικόνα 2)



Τι έχουμε φτιάξει ως τώρα;

Στην 1<sup>η</sup> δραστηριότητα φτιάξαμε ένα απλό παιχνίδι στο οποίο ο χρήστης,



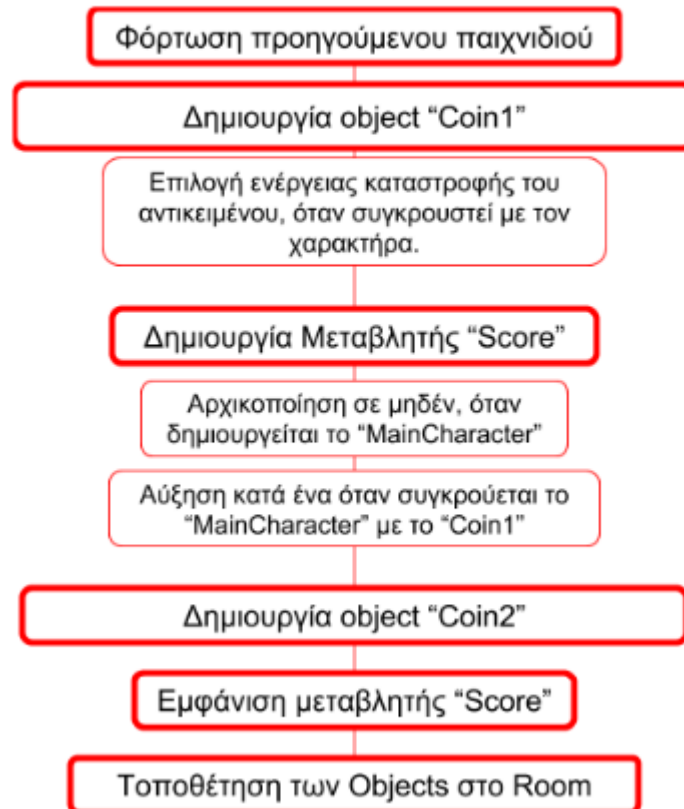
Τι θα μάθουμε;

Στο πλαίσιο της 2<sup>ης</sup> δραστηριότητας θα ασχοληθούμε με μεταβλητές καθώς και με τη δομή επιλογής.

Εικόνα 2: Σύνδεση των δραστηριοτήτων σε επίπεδο στόχων ανάπτυξης και προγραμματιστικών εννοιών εκμάθησης

- παρέχει διαγραμματική απεικόνιση της ακολουθίας των ενεργειών για την επίτευξη των στόχων του Φύλλου Εργασίας (Εικόνα 3)

## Διάγραμμα Ροής Δραστηριότητας



Εικόνα 3: Ακολουθία ενεργειών για την επίτευξη των στόχων του Φύλλου Εργασίας

- καθοδηγεί τον μαθητή καθορίζοντας σταδιακά στόχους τους οποίους καλείται να υλοποιήσει (Εικόνα 4)

## GOAL

Καθόρισε ένα Collision event με το "Coin1" στο object "MainCharacter" το οποίο όταν γίνεται trigger, δηλαδή όταν καλείται, να αυξάνει τη μεταβλητή "score" κατά ένα.


Εικόνα 4: Καθορισμός ενδιάμεσου στόχου

- κινητοποιεί τον μαθητή, καλλιεργεί την αναλυτική και κριτική του σκέψη και καλλιεργεί δεξιότητες στην ανάπτυξη προγραμμάτων και παιχνιδιών θέτοντας ερωτήσεις (Εικόνα 5)

? Πώς μπορούμε να ορίσουμε να γίνεται ενέργεια εξόδου από το παιχνίδι μόνο όταν το score είναι 10;

Εικόνα 5: Ερωτήσεις κινητοποίησης και προβληματισμού

- παρέχει υλικό για νέες έννοιες (Εικόνα 6)



Η δομή επιλογής χρησιμεύει όταν θέλουμε να γίνονται ενέργειες κάτω από μία συνθήκη, δηλαδή μόνο αν μία μεταβλητή έχει κάποια συγκεκριμένη τιμή. Σε αυτή την περίπτωση όταν καλείται το event μέσα στο οποίο υπάρχει η δομή επιλογής, ελέγχεται η συνθήκη και μόνο αν είναι αληθής εκτελούνται οι εντολές που βρίσκονται μέσα σε αυτήν.

Η δομή επιλογής διαβάζεται ως εξής: AN <Συνθήκη είναι αληθής> ΤΟΤΕ <Εκτέλεσε τις εντολές>

Στην περίπτωση που δεν είναι αληθής οι εντολές ΔΕΝ εκτελούνται.

Εικόνα 6: Μαθησιακό υλικό

- καλλιεργεί την αφομοίωση της νέας γνώσης μέσα από ερωτήσεις που θέτει (Εικόνα 7)



- Τι είναι μία μεταβλητή μέσα σε ένα object; Δώσε ένα παράδειγμα μιας άλλης μεταβλητής που θα μπορούσε να υπάρχει σε ένα παιχνίδι.
- Σε τι χρησιμεύει ένα Collision Event;
- Πώς θα μπορούσαμε να ελέγξουμε αν το score είναι μηδέν μέσα σε ένα Event;
- Τι θα συμβεί αν βάλουμε μία ενέργεια εμφάνισης μιας μεταβλητής στην οθόνη σε ένα Collision Event; Θα εμφανιστεί η μεταβλητή ή όχι, και γιατί;

Εικόνα 7: Ερωτήσεις προς απάντηση μετά την ολοκλήρωση του Φύλλου Εργασίας

- υποστηρίζει την εφαρμογή της γνώσης μέσα από την επίλυση νέων στόχων - προκλήσεων (Εικόνα 8)



- Άλλαξε το object "Fire" έτσι ώστε όταν έρχεται σε επαφή με το βασικό χαρακτήρα να καταστρέφεται.
- Τροποποίησε το παιχνίδι έτσι ώστε να ξεκινάει με 10 ζωές.
- Προγραμμάτισε το room έτσι ώστε να δημιουργεί στην εκκίνηση του παιχνιδιού μία ακόμα "Fire" στο πάνω μέρος της οθόνης σε τυχαίο σημείο.
- Άλλαξε το Creation Code έτσι ώστε να δημιουργεί 50 τοίχους μέσα στο room σε τυχαίες θέσεις.

Εικόνα 8: Νέοι στόχοι-προκλήσεις προς επίλυση

## 2.2 Τα Φύλλα Εργασίας

Τα φύλλα εργασίας που σχεδιάστηκαν για την ανάπτυξη του παιχνιδιού είναι έξι και αποσκοπούν στη δημιουργία ενός παιχνιδιού 2Δ. Στόχος είναι ο μαθητής να εξοικειωθεί με βασικές αλγοριθμικές έννοιες αλλά και έννοιες προγραμματισμού παιχνιδιών.

## 2.3 Μαθησιακοί Στόχοι

Οι μαθησιακοί στόχοι των φύλλων εργασίας είναι :

Ο μαθητής να είναι σε θέση να ...

- Δημιουργεί αντικείμενα με εικόνες και γεγονότα που θα αποσκοπούν σε συγκεκριμένες λειτουργίες
- Χρησιμοποιεί χρονοπρογραμματισμό για απλές εργασίες
- Χρησιμοποιεί απλή, σύνθετη και εμφωλευμένη δομή επιλογής
- Χρησιμοποιεί απλή δομή επανάληψης (loop)
- Αρχικοποιεί, να αλλάζει και να εμφανίζει την τιμή μιας μεταβλητής
- Χρησιμοποιεί τις συντεταγμένες της οθόνης για να τοποθετεί αντικείμενα σε συγκεκριμένες θέσεις
- Χρησιμοποιεί συναρτήσεις παραγωγής τυχαίων αριθμών
- Κατασκευάζει αυτόματες κινήσεις για τα αντικείμενα
- Φτιάχνει πίστες, οθόνη έναρξης και λήξης του παιχνιδιού καθώς και αντικείμενα κουμπιά

Οι αλγοριθμικές έννοιες που καλύπτονται είναι οι ακόλουθες:

- Μεταβλητή. Αρχικοποίηση, αλλαγή τιμής και εμφάνιση στην οθόνη.
- Δομή επιλογής. Απλή, σύνθετη και εμφωλευμένη
- Δομή επανάληψης. Απλή

Οι έννοιες προγραμματισμού παιχνιδιών που εξετάζονται είναι οι ακόλουθες:

- Αντικείμενα
- Γεγονότα
- Ενέργειες
- Μεταβλητές (score και lives)
- Δωμάτια (πίστες)

Στο 1<sup>ο</sup> Φύλλο Εργασίας ο μαθητής καλείται να κατασκευάσει τον βασικό χαρακτήρα καθώς και τις κινήσεις του.

Οι μαθησιακοί στόχοι του 1<sup>ου</sup> Φύλλου Εργασίας είναι ο μαθητής να είναι σε θέση να...

- Δημιουργεί αντικείμενα
- Χρησιμοποιεί γεγονότα πληκτρολογίου και δημιουργίας
- Χρησιμοποιεί ενέργειες κίνησης
- Δημιουργεί δωμάτια και να τοποθετεί αντικείμενα μέσα σε αυτά

Στο 2<sup>ο</sup> Φύλλο Εργασίας ο μαθητής καλείται να δημιουργήσει μεταβλητή για το σκορ καθώς και αντικείμενα νομίσματα που θα αυξάνουν αυτή τη μεταβλητή καθώς και αντικείμενο εξόδου από το παιχνίδι.

Οι μαθησιακοί στόχοι του 2<sup>ου</sup> Φύλου Εργασίας είναι ο μαθητής να είναι σε θέση να...

- Δημιουργεί, αλλάζει και εμφανίζει Μεταβλητή Σκορ
- Χρησιμοποιεί Δομή απλής Επιλογής
- Χρησιμοποιεί Γεγονός Σύγκρουσης

Στο 3<sup>ο</sup> Φύλλο Εργασίας ο μαθητής δημιουργεί τα αντικείμενα φωτιάς, τοίχου και μεταβλητή για τον υπολογισμό των ζωών.

Οι μαθησιακοί στόχοι του 3<sup>ου</sup> Φύλου Εργασίας είναι ο μαθητής να είναι σε θέση να...

- Χρησιμοποιεί Δομή Απλής Επανάληψης
- Δημιουργεί, αλλάζει και εμφανίζει Μεταβλητή Ζωής
- Χρησιμοποιεί ενέργεια Δημιουργίας Αντικειμένου

Στο 4<sup>ο</sup> Φύλλο Εργασίας ο μαθητής δημιουργεί αντικείμενο που αυξάνει τις ζωές, το εχθρικό αντικείμενο και το μονοπάτι κίνησής του.

Οι μαθησιακοί στόχοι του 4<sup>ου</sup> Φύλου Εργασίας είναι ο μαθητής να είναι σε θέση να...

- Χρησιμοποιεί Σύνθετη και Εμφωλευμένη Δομή Επιλογής
- Χρησιμοποιεί Γεγονός Βήματος
- Δημιουργεί και να εντάσει στη συμπεριφορά αντικειμένων Μονοπάτια
- Χρησιμοποιεί Γεγονός Ποντικιού
- Χρησιμοποιεί τα Δωμάτια έτσι ώστε να φτιάχνει πίστες δηλαδή επίπεδα δυσκολίας

Στο 5<sup>ο</sup> Φύλλο Εργασίας ο μαθητής βάζει ήχους στο παιχνίδι, χρησιμοποιεί Timeline για τον χρονοπρογραμματισμό και τοποθετεί υπόβαθρο στα δωμάτια.

Οι μαθησιακοί στόχοι του 5<sup>ου</sup> Φύλου Εργασίας είναι ο μαθητής να είναι σε θέση να...

- Δημιουργεί Timeline έτσι ώστε να κάνει Χρονοπρογραμματισμό

Στο 6<sup>ο</sup> Φύλλο Εργασίας ο μαθητής ολοκληρώνει το παιχνίδι με οθόνες έναρξης και λήξης με αντίστοιχα κουμπιά.

Οι μαθησιακοί στόχοι του 6<sup>ου</sup> Φύλου Εργασίας είναι ο μαθητής να είναι σε θέση να...

- Δημιουργεί Δωμάτιο εκκίνησης και τερματισμού με χρήση αντικειμένων-κουμπιών.



## 2.4 Οδηγίες και Προετοιμασία

Καταρχήν πρέπει σε όλους τους υπολογιστές που θα χρησιμοποιηθούν τα φύλλα εργασίας από τους μαθητές να εγκατασταθεί το λογισμικό από την ιστοσελίδα <https://www.yoyogames.com/>.

Κάθε μαθητής πρέπει να δημιουργήσει δικό του λογαριασμό έτσι ώστε να κάνει login καθώς ανοίγει την πλατφόρμα. Δεν μπορεί να χρησιμοποιηθεί ο ίδιος λογαριασμός πολλές φορές.

Σε κάθε υπολογιστή θα πρέπει να υπάρχει ο φάκελος Resources που δίνεται για τα φύλλα εργασίας καθώς περιλαμβάνει εικόνες και ήχους που χρειάζονται για την υλοποίηση του παιχνιδιού.

Παραπέμπουμε στο παράρτημα όπου βρίσκονται τα φύλλα εργασίας

## 2.5 Το παιχνίδι Explorer

Το παιχνίδι λέγεται Explorer, και μετά την ολοκλήρωση όλων των φύλλων εργασίας λειτουργεί ως εξής:

Υπάρχει ο βασικός Χαρακτήρας που κινείται με τα βέλη του πληκτρολογίου μέσα στα δωμάτια. Υπάρχουν κέρματα αξίας 1 και 2 που προσπαθεί να μαζέψει έτσι ώστε να αυξήσει το σκορ του με στόχο να φτάσει στο αντικείμενο εξόδου της πίστας.

Μέσα στα δωμάτια υπάρχουν ζώες που μπορεί να πάρει, φωτιές που αν αγγίξει επιστρέφει στην αρχική θέση του καθώς και εχθρικά αντικείμενα «Σκελετοί» που κινούνται αυτόματα σε κύκλο αλλά και κατακόρυφα. Τέλος, υπάρχουν τοίχοι που δεν μπορεί να περάσει μέσα από αυτούς και δημιουργούν το μονοπάτι που θα πρέπει να ακολουθήσει. Το παιχνίδι ολοκληρώνεται όταν ο παίκτης συγκεντρώσει αρκετό σκορ και φτάσει επιτυχώς στο αντικείμενο εξόδου.

## 2.6 Περιβάλλον GameMaker Studio

Το GameMaker Studio είναι μία πλατφόρμα ανάπτυξης παιχνιδιών που αναπτύσσεται από την YoYo Games. Χρησιμοποιείται κυρίως στην εκπαίδευση. Είναι περιβάλλον οπτικού προγραμματισμού και χρησιμοποιεί blocks για τον προγραμματισμό αντικειμένων (objects), τα οποία αλληλεπιδρούν μεταξύ τους. Πρόκειται για event-driven προγραμματισμό που σημαίνει ότι υπάρχουν γεγονότα, τα οποία όταν συμβαίνουν εκτελούνται κάποιες ενέργειες.

### 2.6.1 Δυνατότητες

Το GameMaker Studio παρέχει ένα πλήρες σύνολο εντολών για χειρισμό αντικειμένων ήχων, εικόνων και μεταβλητών, δομές επιλογής και επανάληψης, καθώς επίσης και έξυπνους τρόπους για χρονοπρογραμματισμό και δημιουργία επιπέδων δηλαδή δωματίων.

Μέσω αυτής της πλατφόρμας μπορεί κανείς να δημιουργήσει παιχνίδια 2D με εικόνες και ήχους καθώς και 3D με ειδικές εντολές που δεν είναι διαθέσιμες στην δωρεάν έκδοση.

## 2.6.2 Ιστορικό

Το περιβάλλον αυτό δημιουργήθηκε από τον Mark Overmars και η πρώτη έκδοση κυκλοφόρησε στις 15 Νοεμβρίου του 1999, με το όνομα Animo. Η πλατφόρμα αυτή αποσκοπούσε στη δημιουργία γραφικών και όχι παιχνιδιών. Στην πορεία άλλαξε σε Game Maker και υποστήριζε και τη δημιουργία 2D παιχνιδιών. Το 2004 κυκλοφόρησε η έκδοση 6 που είχε μεγάλη απήχηση και περιελάμβανε τη δυνατότητα κατασκευής 3D αντικειμένων. Το 2007 ξεκίνησε η συνεργασία με την YoYo Games, η οποία και ανέλαβε το παιχνίδι.

Σήμερα χρησιμοποιούμε την έκδοση GameMaker Studio 2, η οποία κυκλοφόρησε τον Μάρτιο του 2017, είναι γραμμένη σε C# και παρέχει ένα καινούργιο εξελιγμένο IDE με περισσότερες λεπτομέρειες και δυνατότητες σε ένα ευέλικτο περιβάλλον.

## 2.6.3 Πώς χρησιμοποιείται

Το περιβάλλον χρησιμοποιεί τους πόρους δηλαδή τα Resources. Αυτοί περιλαμβάνουν:

- Τις εικόνες που θα χρησιμοποιηθούν για αντικείμενα και υπόβαθρο στις πίστες (Sprites).
- Τα αντικείμενα που περιλαμβάνουν από ένα Sprite, γεγονότα (events) καθώς και ενέργειες (actions), που εκτελούνται όταν γίνεται trigger ένα γεγονός. Ο προγραμματιστής καλείται να επιλέξει Event και να καθορίσει ενέργειες που θα εκτελούνται. Με αυτόν τον τρόπο καθορίζεται η συμπεριφορά ενός αντικειμένου.
- Τα δωμάτια που είναι οι πίστες του παιχνιδιού και εκεί μέσα τοποθετούνται τα αντικείμενα. Τα δωμάτια μπορούν να χρησιμοποιηθούν και για την οθόνη έναρξης και λήξης του παιχνιδιού.
- Τους ήχους, που ακούγονται μέσω συγκεκριμένων ενεργειών μέσα στα Events.

Επίσης υπάρχουν και άλλα Resources, όπως για παράδειγμα Timelines, που χρησιμεύουν στον χρονοπρογραμματισμό, Paths, δηλαδή μονοπάτια που ακολουθούν τα αντικείμενα και κινούνται αυτόματα καθώς και Scripts όπου μπορεί ο προγραμματιστής να γράψει δικό του κώδικα σε μορφή κειμένου.

### 3. ΠΙΛΟΤΙΚΗ ΕΦΑΡΜΟΓΗ ΣΕ ΦΟΙΤΗΤΕΣ

#### 3.1 Το ερωτηματολόγιο

Προκειμένου να έχουμε ανατροφοδότηση σε ότι αφορά την ποιότητα και την αποτελεσματικότητα του 1<sup>ου</sup> και 2<sup>ου</sup> φύλλου εργασίας, καλέσαμε 22 φοιτητές να τα εκτελέσουν και να απαντήσουν σε ένα ερωτηματολόγιο. Πρέπει να αναφερθεί ότι κανείς από τους 22 φοιτητές δεν είχε ξαναχρησιμοποιήσει το περιβάλλον Game Maker. Ο χρόνος που δόθηκε στους φοιτητές για την εκπόνηση των δύο φύλλων εργασίας ήταν δύο διδακτικές ώρες. Το ερωτηματολόγιο περιλάμβανε ερωτήσεις σχετικά με:

- Τον στόχο των φύλλων εργασίας
- Την ποιότητα
- Το επίπεδο δυσκολίας
- Το επεξηγηματικό υλικό
- Την οργάνωση
- Την μορφοποίηση
- Το GameMaker Studio ως εργαλείο ανάπτυξης παιχνιδιών

Συγκεκριμένα οι ερωτήσεις και οι απαντήσεις ήταν οι εξής:

- Σχετικά με το 1ο φύλλο εργασίας:
  - Ο στόχος του 1ου Φύλλου Εργασίας ήταν σαφής;
    - Καθόλου έως πάρα πολύ
  - Οι οδηγίες του 1ου Φύλλου Εργασίας για τη δημιουργία του παιχνιδιού ήταν ...
    - Κατανοητές, Σαφείς, Καθοδηγητικές, Πολύ αναλυτικές
  - Το επίπεδο δυσκολίας του 1ου Φύλλου Εργασίας κρίνεται αποδεκτό για αρχάριους;
    - Καθόλου έως Πάρα πολύ
- Σχετικά με το 2ο Φύλλο Εργασίας:
  - Ο στόχος του 2ου Φύλλου Εργασίας ήταν σαφής;
    - Καθόλου έως Πάρα πολύ
  - Οι οδηγίες του 2ου Φύλλου Εργασίας για τη δημιουργία του παιχνιδιού ήταν ...
    - Κατανοητές, Σαφείς, Καθοδηγητικές, Πολύ αναλυτικές
  - Το επίπεδο δυσκολίας του 2ου Φύλλου Εργασίας κρίνεται αποδεκτό για αρχάριους;
    - Καθόλου έως Πάρα πολύ
- Σχετικά με τη οργάνωση και τη μορφοποίηση των Φύλλων Εργασίας:
  - Το επεξηγηματικό υλικό (π.χ. για την έννοια της μεταβλητής) πιστεύετε ότι βοηθά τον μαθητή στην κατανόηση της εκάστοτε έννοιας;
    - Καθόλου έως Πάρα πολύ

- Πώς κρίνετε την οργάνωση των Φύλλων Εργασίας (π.χ. εναλλαγή οδηγιών με θεωρία/εξηγήσεις, υλοποίηση σε διακριτά βήματα);
  - Βοηθητική Κουραστική
- Η χρήση εικονιδίων πιστεύετε ότι βοηθά τον μαθητή στον εντοπισμό της επιθυμητής πληροφορίας;
  - Καθόλου έως Πάρα πολύ
- Η χρήση χρωμάτων στο κείμενο πιστεύετε ότι βοηθά τον μαθητή στον εντοπισμό της επιθυμητής πληροφορίας;
  - Καθόλου έως Πάρα πολύ
- Πώς κρίνετε το GAME MAKER ως περιβάλλον ανάπτυξης παιχνιδιών;
  - Εύχρηστο, Κατάλληλο για μαθητές Γυμνασίου, Κατάλληλο για μαθητές Λυκείου, Κατάλληλο για μαθητές Τριτοβάθμιας

### 3.2 Τα αποτελέσματα

Όλοι οι φοιτητές κατάφεραν να ολοκληρώσουν τα δύο φύλλα εργασίας στον προγραμματισμένο χρόνο και μάλιστα κάποιοι σε λιγότερο χρόνο. Δεν ζητήθηκε κανενός είδους βοήθεια και επεξήγηση. Οι φοιτητές εργάστηκαν αποκλειστικά χρησιμοποιώντας τα Φύλλα Εργασίας.

Όσον αφορά τις απαντήσεις των φοιτητών στις ερωτήσεις, αυτές ήταν θετικές.

Συγκεκριμένα, και στα δύο φύλλα εργασίας, ο στόχος χαρακτηρίστηκε σαφής και οι οδηγίες πάρα πολύ αναλυτικές. Τα δύο τρίτα των φοιτητών χαρακτήρισαν το επίπεδο δυσκολίας πολύ ή πάρα πολύ αποδεκτό για αρχάριους.

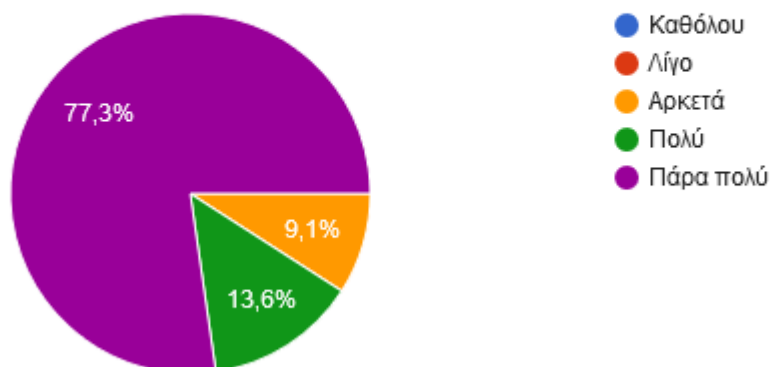
Σε ότι αφορά το επεξηγηματικό υλικό τα τρία τέταρτα των φοιτητών απάντησαν ότι βοηθά πολύ ή πάρα πολύ το μαθητή στην κατανόηση της εκάστοτε έννοιας. Η οργάνωση των φύλλων εργασίας χαρακτηρίστηκε από τους περισσότερους φοιτητές πολύ βοηθητική και καθόλου ή λίγο κουραστική. Από την συντριπτική πλειοψηφία των φοιτητών προκύπτει ότι η χρήση των εικονιδίων καθώς και χρωμάτων βοηθά πάρα πολύ τον μαθητή στον εντοπισμό της επιθυμητής πληροφορίας.

Τέλος, το GameMaker ως εργαλείο προγραμματισμού παιχνιδιών χαρακτηρίστηκε αρκετά έως πάρα πολύ εύχρηστο, κατά μεγάλη πλειοψηφία αρκετά και πολύ κατάλληλο για μαθητές γυμνασίου, πολύ κατάλληλο για μαθητές λυκείου, και πάρα πολύ κατάλληλο για μαθητές τριτοβάθμιας.

Συγκεκριμένα οι απαντήσεις που δόθηκαν φαίνονται στα παρακάτω σχήματα.

## Ο στόχος του 1ου Φύλλου Εργασίας ήταν σαφής;

22 απαντήσεις



Σχήμα 1: Ερώτηση σχετικά με τον στόχο του 1ου Φύλλου Εργασίας

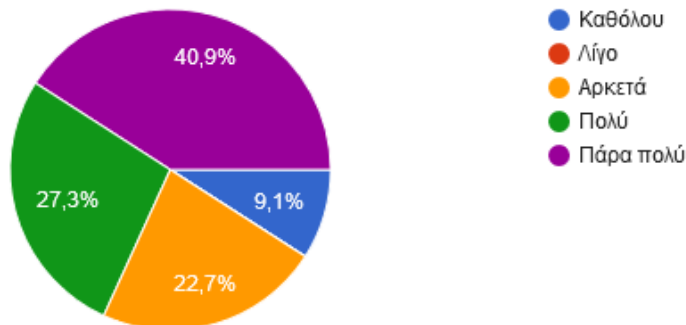
Οι οδηγίες του 1ου Φύλλου Εργασίας για τη δημιουργία του παιχνιδιού ήταν ...



Σχήμα 2: Ερώτηση σχετικά με τις οδηγίες του 1ου Φύλλου Εργασίας

### Το επίπεδο δυσκολίας του 1ου Φύλλου Εργασίας κρίνεται αποδεκτό για αρχάριους;

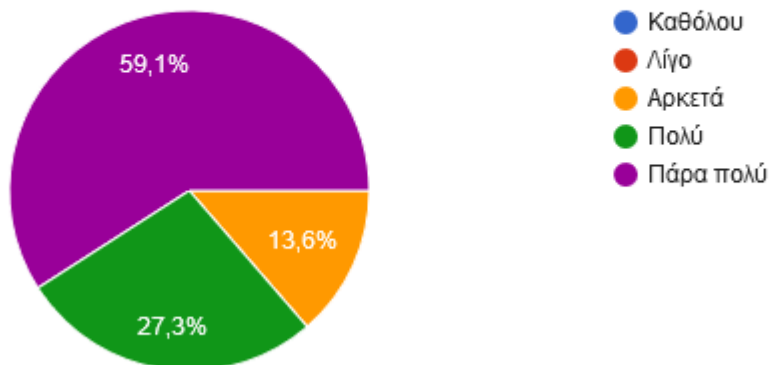
22 απαντήσεις



Σχήμα 3: Ερώτηση σχετικά με το επίπεδο δυσκολίας του 1<sup>ου</sup> Φύλλου Εργασίας

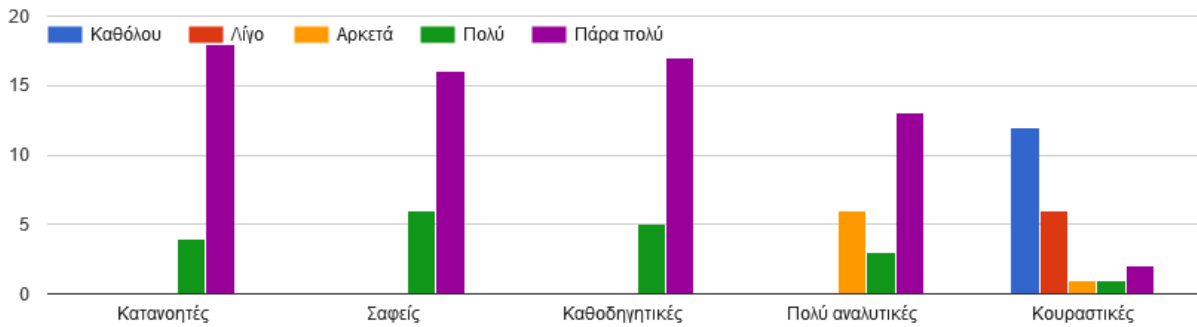
### Ο στόχος του 2ου Φύλλου Εργασίας ήταν σαφής;

22 απαντήσεις



Σχήμα 4: Ερώτηση σχετικά με το στόχο του 2<sup>ου</sup> Φύλλου Εργασίας

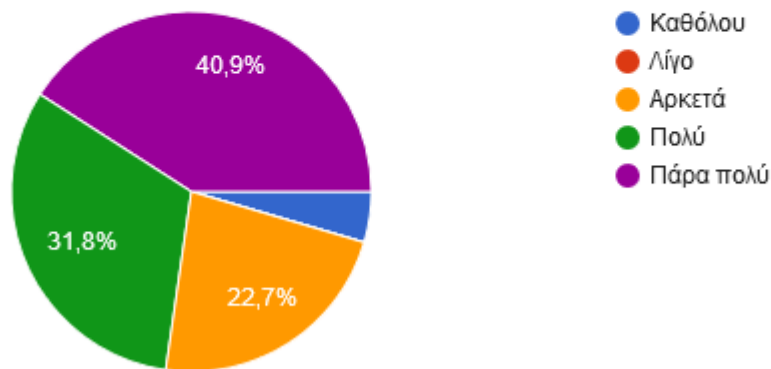
Οι οδηγίες του 2ου Φύλλου Εργασίας για τη δημιουργία του παιχνιδιού ήταν ...



Σχήμα 5: Ερώτηση σχετικά με τις οδηγίες του 2<sup>ου</sup> Φύλλου Εργασίας

Το επίπεδο δυσκολίας του 2ου Φύλλου Εργασίας κρίνεται αποδεκτό για αρχάριους;

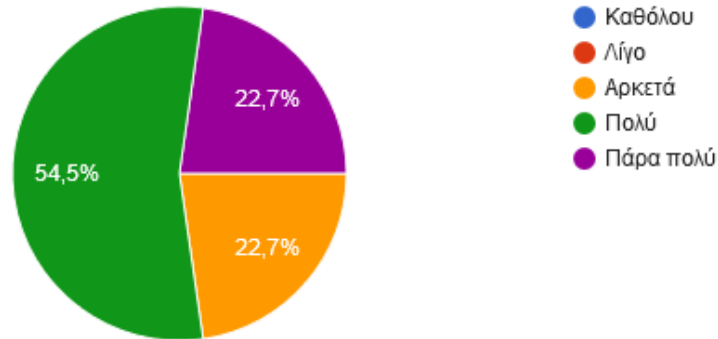
22 απαντήσεις



Σχήμα 6: Ερώτηση σχετικά με το επίπεδο δυσκολίας του 2<sup>ου</sup> Φύλλου Εργασίας

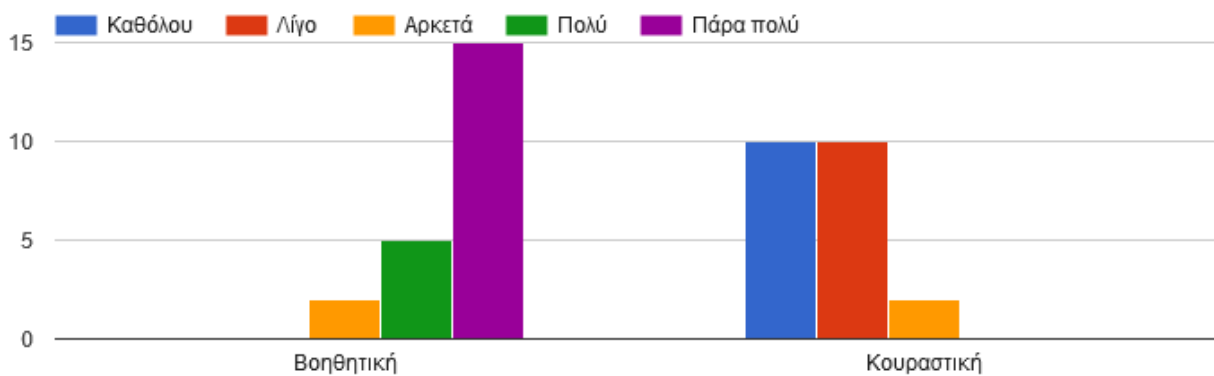
Το επεξηγηματικό υλικό (π.χ. για την έννοια της μεταβλητής) πιστεύετε ότι βοηθά τον μαθητή στην κατανόηση της εκάστοτε έννοιας;

22 απαντήσεις



Σχήμα 7: Ερώτηση σχετικά με το επεξηγηματικό υλικό

Πώς κρίνετε την οργάνωση των Φύλλων Εργασίας (π.χ. εναλλαγή οδηγιών με θεωρία/εξηγήσεις, υλοποίηση σε διακριτά βήματα);

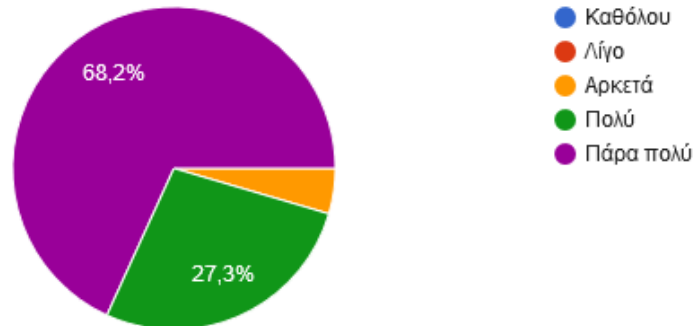


Σχήμα 8: Ερώτηση σχετικά με την οργάνωση των Φύλλων Εργασίας



## Η χρήση εικονιδίων πιστεύετε ότι βοηθά τον μαθητή στον εντοπισμό της επιθυμητής πληροφορίας;

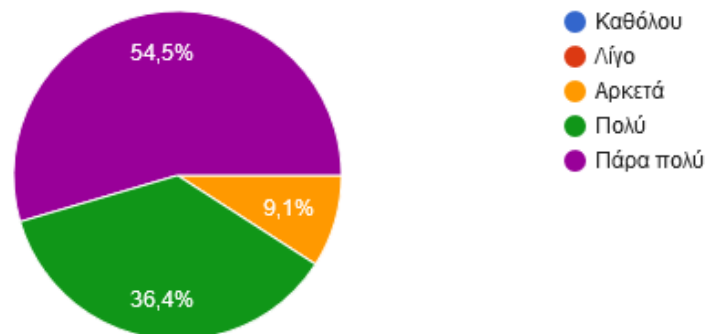
22 απαντήσεις



Σχήμα 9: Ερώτηση σχετικά με την χρήση εικονιδίων

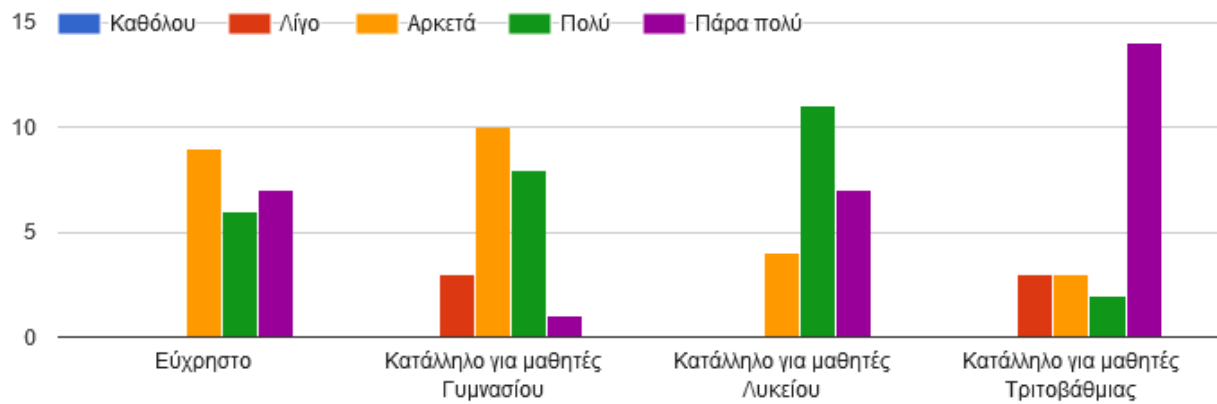
## Η χρήση χρωμάτων στο κείμενο πιστεύετε ότι βοηθά τον μαθητή στον εντοπισμό της επιθυμητής πληροφορίας;

22 απαντήσεις



Σχήμα 10: Ερώτηση σχετικά με τη χρήση χρωμάτων στο κείμενο

## Πώς κρίνετε το GAME MAKER ως περιβάλλον ανάπτυξης παιχνιδιών;



Σχήμα 11: Ερώτηση σχετικά με το Game Maker ως περιβάλλον ανάπτυξης παιχνιδιών

## 4. ΕΠΙΛΟΓΟΣ

Στο πλαίσιο της παρούσας εργασίας σχεδιάστηκαν κατάλληλα Φύλλα Εργασίας για την ανάπτυξη ενός παιχνιδιού με στόχο την εκμάθηση προγραμματισμού. Η πιλοτική εφαρμογή δύο εκ των Φύλλων Εργασίας, έδειξε ότι ο οπτικός προγραμματισμός στην κατασκευή παιχνιδιών όπως αυτός γίνεται με το GameMaker, μπορεί να βοηθήσει ιδιαίτερα στην κατανόηση αλγοριθμικών εννοιών αλλά και εννοιών που αφορούν στον προγραμματισμό παιχνιδιών.

Το οπτικό περιβάλλον δίνει τη δυνατότητα στον μαθητή να βλέπει ομαδοποιημένες εντολές όπως στην περίπτωση της δομής επιλογής και της δομής επανάληψης και να κατανοήσει τον τρόπο που αυτές εκτελούνται. Η κατασκευή παιχνιδιών, από την άλλη πλευρά, προκαλεί το ενδιαφέρον των μαθητών και σταδιακά τους εισάγει στον προγραμματισμό

Τα φύλλα εργασίας παρέχουν ένα καθοδηγούμενο πλαίσιο, ειδικά για αρχάριους μαθητές, που τους βοηθά στην κατανόηση και στην εφαρμογή της νέας γνώσης.

Ως μελλοντική ενέργεια της συγκεκριμένης προσπάθειας αποτελεί η εφαρμογή των συγκεκριμένης προσέγγισης σε πραγματικές συνθήκες τάξης και η αποτίμηση των αποτελεσμάτων.

## ΠΑΡΑΡΤΗΜΑ

### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1 GAME MAKER

Το πρώτο μας παιχνίδι !



#### Έννοιες Προγραμματισμού Παιχνιδιών:

Αντικείμενο (Object), Γεγονός Πληκτρολογίου (Keyboard Event), Γεγονός Δημιουργίας (Create Event), Δωμάτιο (Room), Ενέργειες Κίνησης

#### ? Τι είναι το Game Maker;

Το Game Maker είναι ένα λογισμικό προγραμματισμού με το οποίο κατασκευάζουμε παιχνίδια. Για να το κάνουμε αυτό, χρησιμοποιούμε **objects** (αντικείμενα) με τα οποία καθορίζουμε τις οντότητες του παιχνιδιού που θέλουμε να φτιάξουμε.

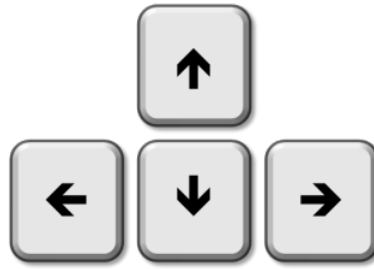
#### ? Τι είναι τα Objects;

Κάθε **object** που θα φτιάξουμε αντιστοιχεί σε μία οντότητα και πρέπει να έχει:

- ένα **sprite** (εικόνα) με το οποίο θα εμφανίζεται στον κόσμο μας, για παράδειγμα ένα ανθρωπάκι.
- ένα ή περισσότερα **events** (γεγονότα) και ενέργειες. Οι ενέργειες καθορίζουν το πως θα συμπεριφέρεται το **object** μας. Για παράδειγμα μία ενέργεια θα μπορούσε να είναι να κινηθεί το **object** προς κάποια κατεύθυνση στην οθόνη ή να αλλάξει η εικόνα του, δηλαδή το **sprite**. Τα **events** πυροδοτούν τις ενέργειες που θέλουμε να εκτελεί το **object**. Για παράδειγμα ένα **event** είναι το πάτημα ενός κουμπιού ή η σύγκρουση ενός **object** με ένα άλλο. Όταν συμβαίνει ένα **event** εκτελούνται συγκεκριμένες ενέργειες.

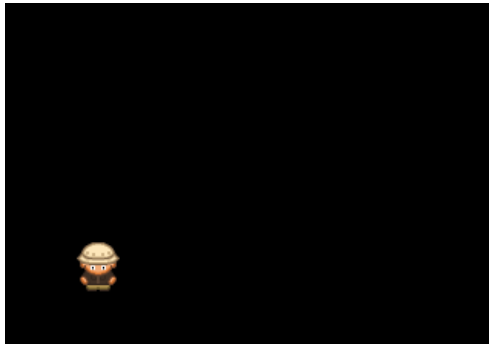
#### ? Τι θα φτιάξουμε;

Θα φτιάξουμε ένα παιχνίδι στο οποίο θα υπάρχει ένα **object**, ο βασικός χαρακτήρας. Αυτό το **object** θα έχει την εξής λειτουργία: Όταν ο χρήστης του παιχνιδιού πατάει ένα από τα βελάκια του πληκτρολογίου θα κινείται προς την αντίστοιχη κατεύθυνση μέσα στο κόσμο που θα φτιάξουμε.

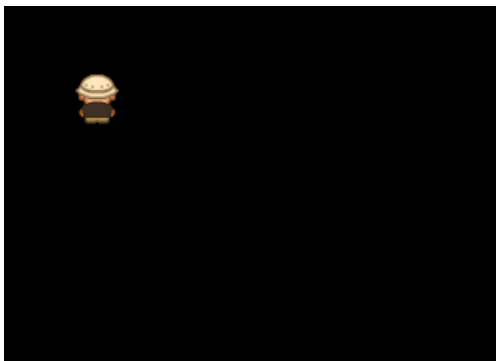


Εικόνα 1. Τα βέλη του πληκτρολογίου

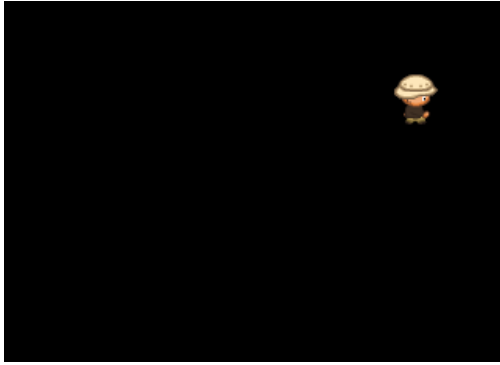
Δες στις εικόνες 2,3,4,5 και 6 πώς θα μετακινείται το **object** ανάλογα με το ποιο κουμπί πατάει ο χρήστης:



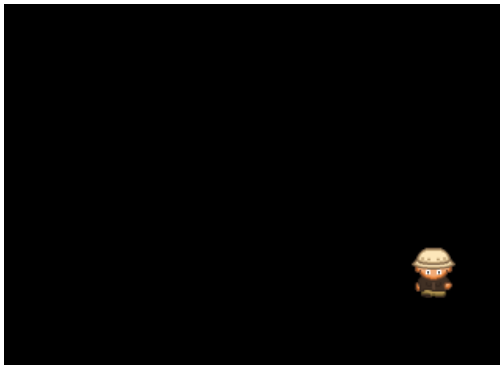
Εικόνα 2. Αρχική θέση object



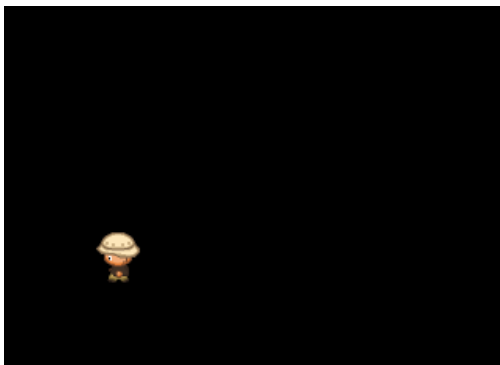
Εικόνα 3. Αφού πατηθεί το πάνω βελάκι



Εικόνα 4. Αφού πατηθεί το δεξί βελάκι.



Εικόνα 5. Αφού πατηθεί το κάτω βελάκι.



Εικόνα 6. Αφού πατηθεί το αριστερό βελάκι.

## Διάγραμμα Ροής Δραστηριότητας:





Η διαφορά ανάμεσα στο Sprite και στο Object είναι ότι το Sprite είναι η εικόνα με την οποία θα εμφανίζεται το Object. Το Object περιλαμβάνει όλες τις λειτουργίες δηλαδή τα events και τις ενέργειες που θα κάνει αυτή η οντότητα.

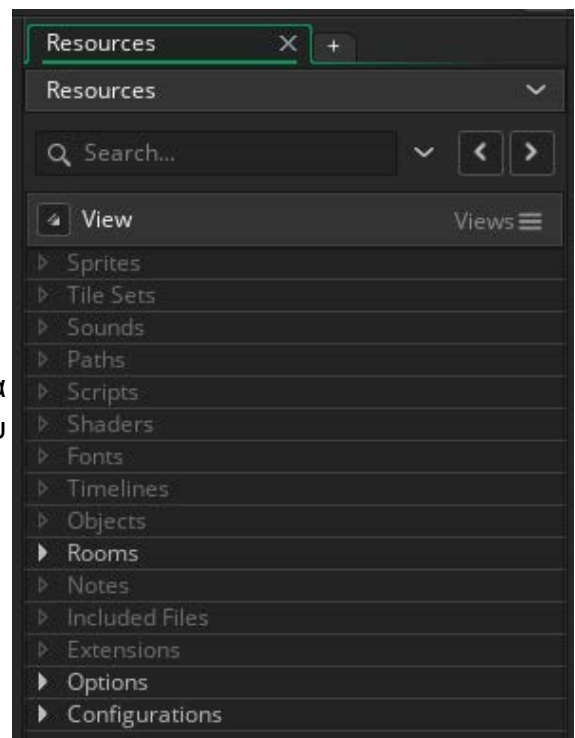
Ας ξεκινήσουμε λοιπόν!

Αρχικά άνοιξε το πρόγραμμα **Game Maker** και μετά ακολούθησε τα παρακάτω βήματα:

1. Κάνε Log In με τα στοιχεία του λογαριασμού σου, εάν ζητηθεί
2. Επίλεξε “New” και αμέσως “Drag and Drop”
3. Μετά ανοίγει ένα παράθυρο στο οποίο επέλεξε το όνομα καθώς και το φάκελο του αρχείου επεξεργασίας του παιχνιδιού σου.

Αμέσως θα οδηγηθείς στην αρχική σελίδα ενός κενού project.

Δεξιά υπάρχει η μπάρα **Resources**. Εκεί θα βρίσκονται όλοι οι πόροι του παιχνιδιού που θα δημιουργήσεις όπως **Sprites**, **Objects** με λειτουργίες, **Rooms** (δωμάτια) και άλλα (εικόνα 7).



Εικόνα 7. Η μπάρα των Resources



## ΒΗΜΑ ΠΡΩΤΟ: Δημιουργία Sprites

? Με ποιον τρόπο εισάγουμε εικόνες έτσι ώστε να τις χρησιμοποιήσουμε όταν φτιάξουμε τον βασικό μας ήρωα;



Τα sprites χρησιμοποιούνται ως αναπαράσταση των objects στο παιχνίδι. Δηλαδή κάθε τι που θέλουμε να εμφανίζεται πρέπει να έχει δικό του sprite. Προσοχή! Όταν φτιάχνουμε objects στον κόσμο μας με περισσότερες από μία λειτουργίες ίσως να χρειάζεται ένα sprite για κάθε μία από αυτές.

### GOAL

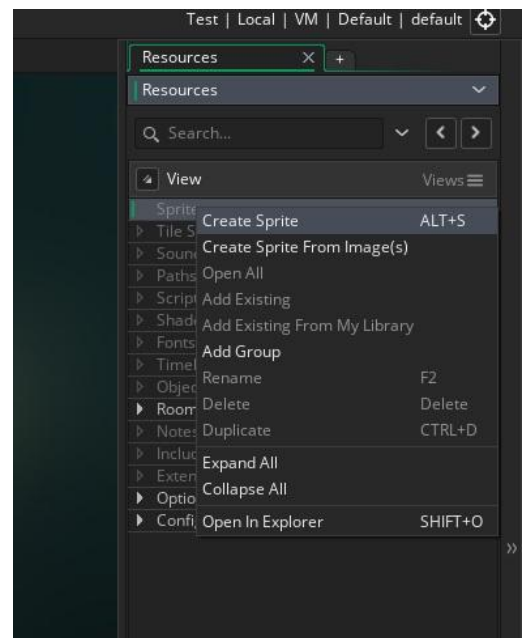
Δημιούργησε sprite για την κίνηση προς τα πάνω

Για το **object** που θα φτιάξεις, δηλαδή τον βασικό χαρακτήρα, θα χρειαστείς 5 συνολικά **sprites**. Ένα για κάθε κίνηση που θα κάνει (πάνω, κάτω, δεξιά, αριστερά) και ένα για τη στιγμή που θα είναι ακινητοποιημένος.

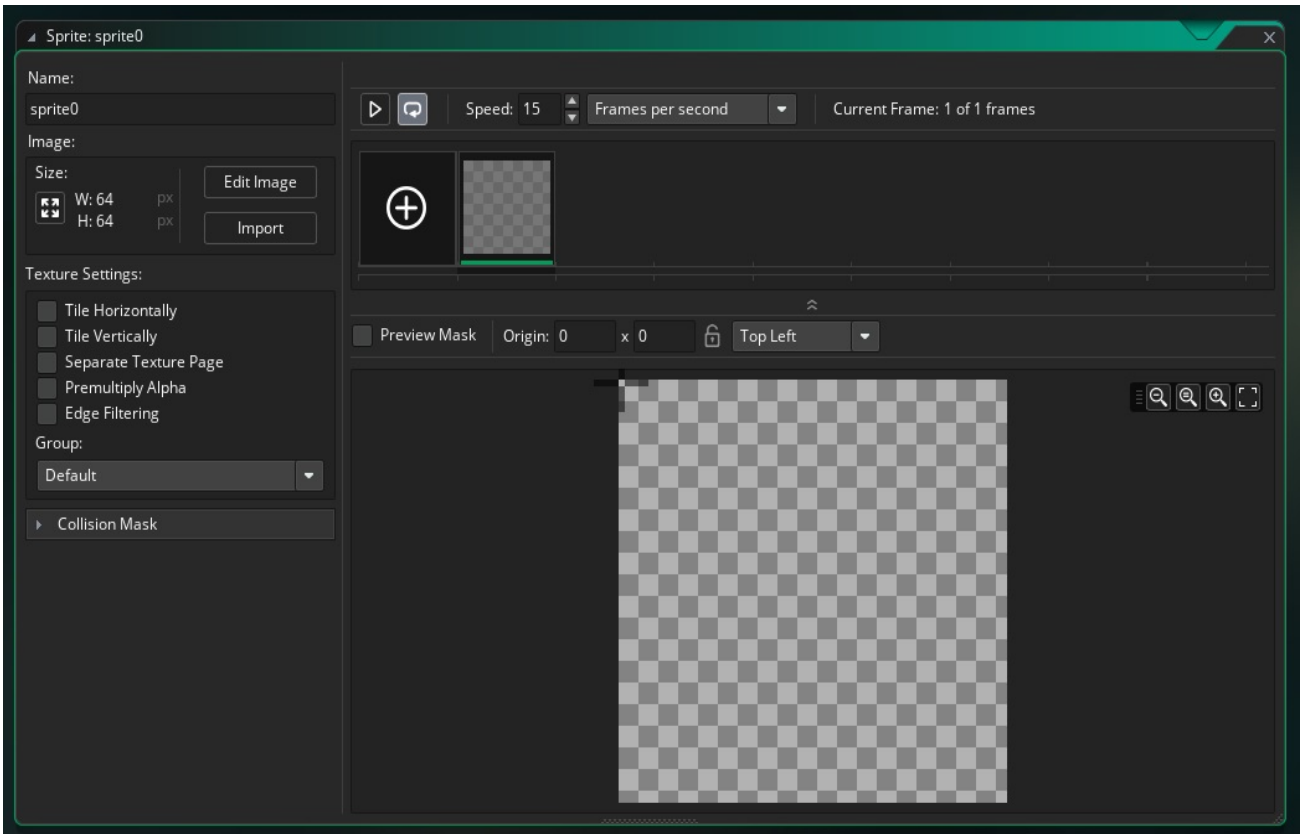
Πρέπει πρώτα να φτιάξεις τα **sprites** και μετά το **object**.

Για να δημιουργήσεις ένα **sprite** κάνε δεξί κλικ στο **Sprites** της μπάρας με τους πόρους και κλικ στην επιλογή “**Create Sprite**” (εικόνα 8).

Έτσι ανοίγει το παράθυρο επεξεργασίας **Sprite**, όπως φαίνεται στην εικόνα 9:



Εικόνα 8. Το μενού για δημιουργία Sprite



Εικόνα 9. Παράθυρο επεξεργασίας ενός κενού Sprite

Τώρα είσαι έτοιμος να δημιουργήσεις το **sprite** του **object** καθώς αυτό θα κινείται προς τα πάνω. Μέσα στο παραπάνω παράθυρο κάνε 3 αλλαγές (Εικόνα 10, κόκκινα πλαίσια).



Εικόνα 10. Δημιουργία Sprite με όνομα MainCharacterUp

- Αλλαξε το όνομα (**Name** πάνω αριστερά) από **“sprite0”** σε **“MainCharacterUp”**.
1. Εισήγαγε το κατάλληλο αρχείο ως εξής:
    - a. Πάτα **import**
    - b. Στο παράθυρο που θα εμφανιστεί, μπες στον φάκελο **Resources** και εκεί θα βρεις πολλές εικόνες για τον χαρακτήρα μας.
    - c. Επίλεξε **explorer\_up**
  2. Πάνω από τη μεγάλη εικόνα κάνε κλικ στο **“Top Left”** και επίλεξε **“Middle Centre”**.

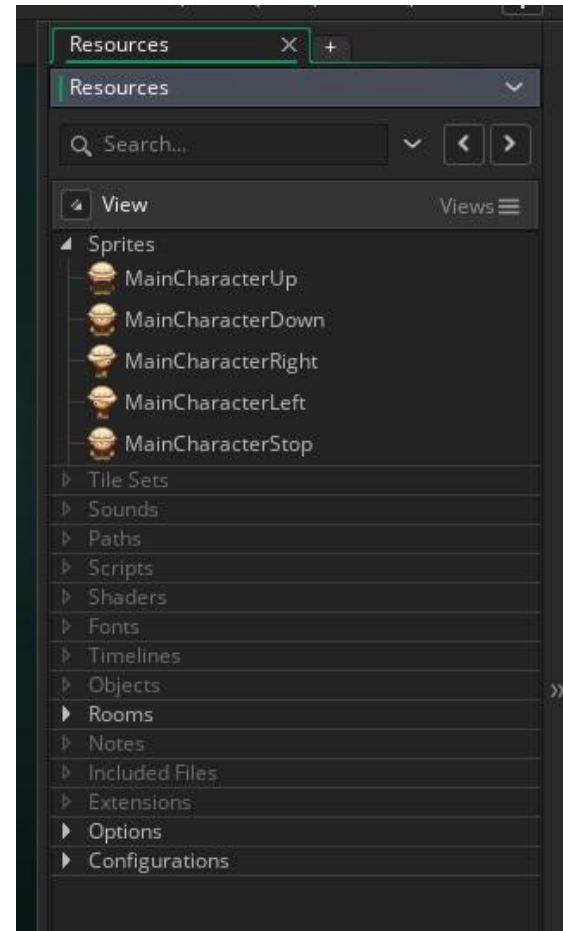
3. Για να δεις πως θα εμφανίζεται το **object** στο κόσμο μας όταν θα κινείται προς τα πάνω μπορείς να κάνεις κλικ στο “**Play/Pause animation**” όπως φαίνεται στην εικόνα 10.
4. Τέλος κλείσε το παράθυρο.

## GOAL

Φτιάξε 4 καινούργια sprites για τις υπόλοιπες κινήσεις, χρησιμοποιώντας κατάλληλα ονόματα και επιλέγοντας τα αντίστοιχα αρχεία από τον φάκελο που Resources.

Αφού φτιάξεις αυτά τα 4 καινούργια **sprites** το αποτέλεσμα θα πρέπει να είναι αντίστοιχο με αυτό της εικόνας 11. Δηλαδή στα **Resources** να υπάρχουν 5 **sprites**.

Εικόνα 11. Το αποτέλεσμα της δημιουργίας των 5 sprite



## ΒΗΜΑ ΔΕΥΤΕΡΟ: Δημιουργία Object

? Με ποιον τρόπο θα δημιουργήσουμε τον βασικό ήρωά μας έτσι ώστε να πηγαίνει πάνω κάτω με τα βελάκια και ταυτόχρονα να αλλάζει το sprite;



Τα **objects**(αντικείμενα) είναι οι οντότητες του παιχνιδιού. Τα **objects** συνδέονται με ένα αρχικό **sprite**, και ταυτόχρονα μπορούμε να τα προγραμματίσουμε να εκτελούν κάποιες ενέργειες όταν συμβαίνουν διάφορα **events** (γεγονότα).

Για παράδειγμα ένα **event** είναι να πατηθεί από τον χρήστη του προγράμματος το πάνω βελάκι. Σε αυτή την περίπτωση το **object** θέλουμε να:

- Αλλάζει το αρχικό του **sprite** με το αντίστοιχο που έχουμε ήδη εισάγει παραπάνω
- Μετακινείται προς τα πάνω με συγκεκριμένη ταχύτητα

## GOAL

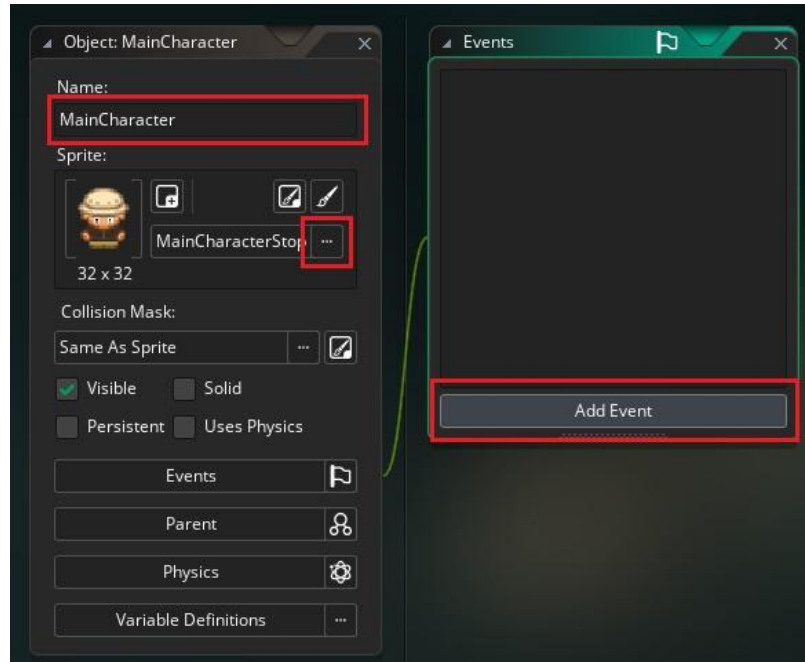
### Φτιάξε ένα Object

- με αρχικό **sprite** το “**MainCharacterStopped**”
- ένα **event** (πάτημα του πλήκτρου με το πάνω βελάκι) στο οποίο να
  - αλλάζει το **sprite** με το “**MainCharacterUp**”
  - και να οριοθετείται η κατεύθυνση προς τα πάνω και η ταχύτητα της κίνησής του.

Ακολουθήσε τα παρακάτω βήματα:

1. Στη μπάρα των πόρων κάνε δεξί κλικ στο **Objects** και επίλεξε **Create Object**.
2. Στο παράθυρο που εμφανίζεται μπορείς να επεξεργαστείς το καινούργιο **Object**. Κάνε τα εξής:
  - a. Άλλαξε το όνομα σε “**MainCharacter**”.
  - b. Κάνοντας κλικ στο κουμπί με τις τρεις τελείες όπως φαίνεται στην εικόνα 12, θα εμφανιστεί ένα παράθυρο για να διαλέξεις ποιο **Sprite** από αυτά που έχεις δημιουργήσει θέλεις για το καινούργιο **Object**.

c. Επίλεξε **MainCharacterStopped**.



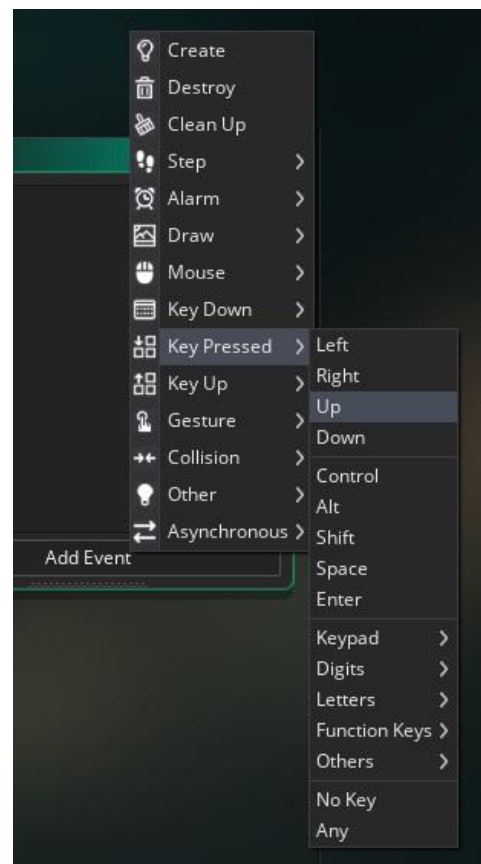
Εικόνα 12. Δημιουργία object “MainCharacter”

Τώρα είσαι έτοιμος να καθορίσεις την κίνηση που θα κάνει το **object** σου κάθε φορά που ο χρήστης του παιχνιδιού θα πατάει κάποιο πλήκτρο κίνησης (βελάκι):

a. Αρχικά κάνε κλικ στο **Add Event** όπως φαίνεται στην εικόνα 12.

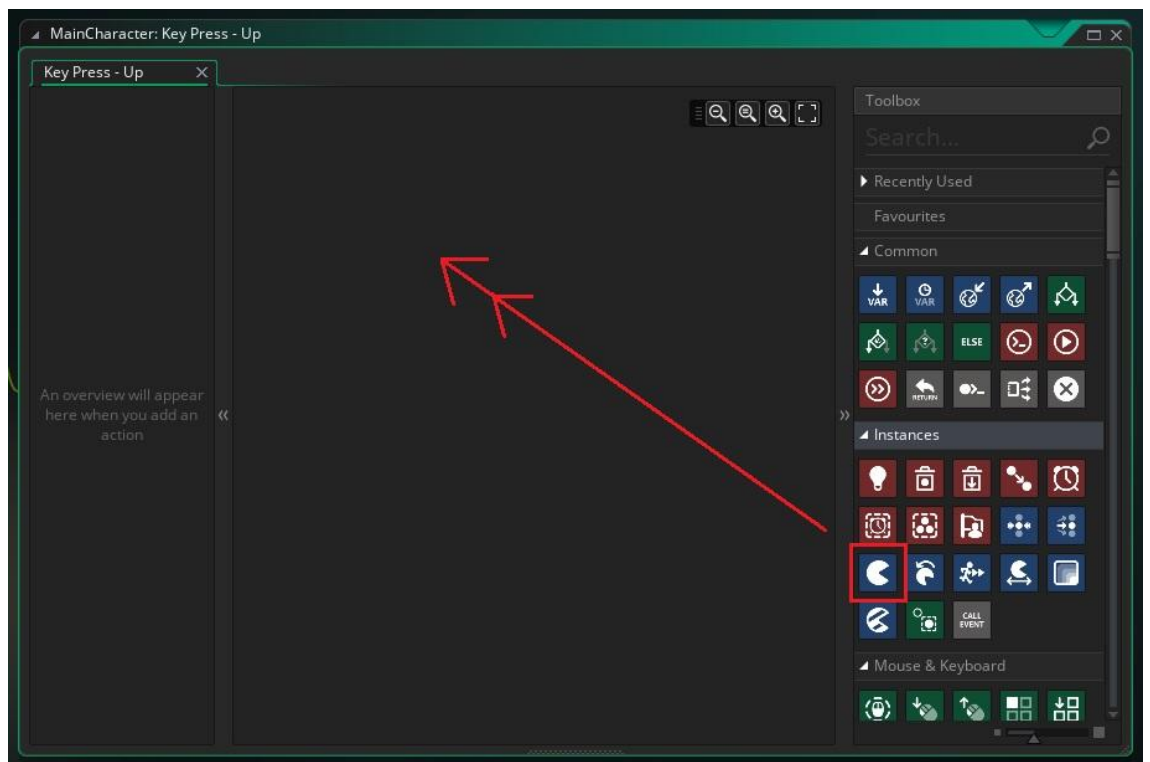
b. Στο μενού που θα εμφανιστεί, επίλεξε **KeyPressed** και μετά **Up** όπως στην εικόνα 13, για να καθορίσεις τι θα γίνεται στην περίπτωση που ο χρήστης πατήσει το πάνω βελάκι.

c. Στο παράθυρο επεξεργασίας λειτουργιών του συγκεκριμένου **Event**, στο αριστερό και στο κεντρικό τμήμα εμφανίζονται οι ενέργειες που θα γίνονται και στο δεξί τμήμα εμφανίζεται ένα **toolbox** με όλες τις ενέργειες σε μορφή εικονιδίων, που μπορείς να φέρεις στο κεντρικό τμήμα προκειμένου να γίνονται όταν συμβαίνει το συγκεκριμένο **Event**.



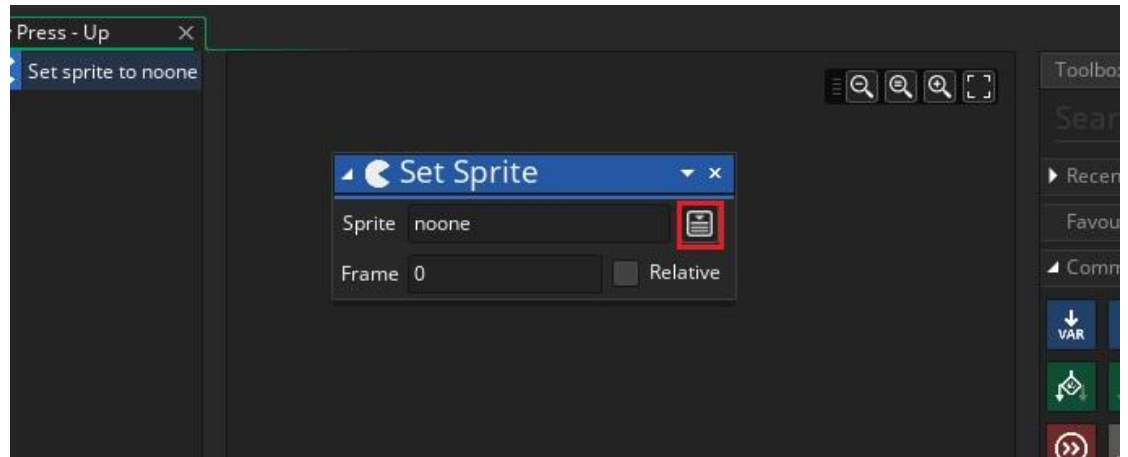
Εικόνα 13. Επιλογή Key Pressed Up Event

- d. Εμείς θέλουμε κάθε φορά που πατιέται το πάνω βελάκι του πληκτρολογίου να γίνονται τρεις ενέργειες:
- i. Αρχικά να αλλάζει το **sprite** του **object** με το κατάλληλο από αυτά που έχουμε δημιουργήσει. Για να το κάνεις αυτό, σύρε την ενέργεια **Set Sprite** (εικονίδιο pacman) μέσα στο κεντρικό τμήμα του παραθύρου, όπως φαίνεται στην εικόνα 14.



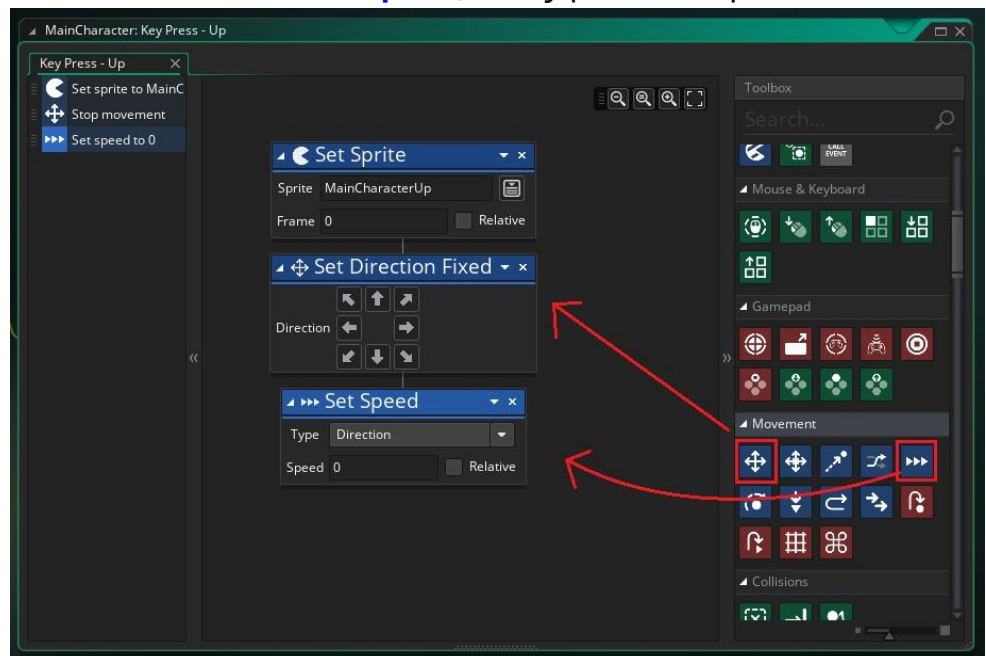
Εικόνα 14. Καθορισμός ενέργειας Set Sprite στο Key Pressed Up Event

- ii. Έτσι εισάγεται η λειτουργία **Set Sprite**. Επίλεξε το κατάλληλο **Sprite** από αυτά που έχεις φτιάξει πατώντας πάνω στο κουμπί όπως φαίνεται στην εικόνα 15.



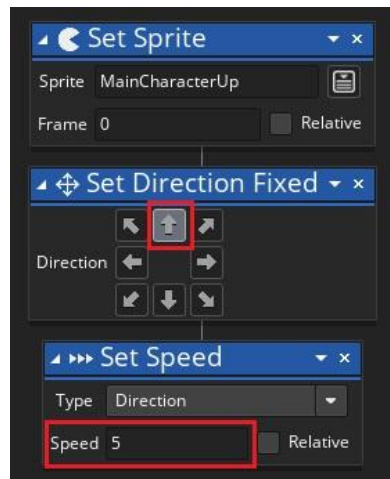
Εικόνα 15. Κουμπί καθορισμού Sprite στην ενέργεια Set Sprite

- iii. Τώρα εισήγαγε δύο ακόμα λειτουργίες από το **toolbox**. **Set Direction Fixed** και **Set Speed**, όπως φαίνεται στην εικόνα 16.



Εικόνα 16. Εισαγωγή ενεργειών Set Direction Fixed και Set Speed στο Key Press-Up Event

- iv. Για τη λειτουργία **Set Direction Fixed** κάνε κλικ στο πάνω βελάκι και για τη λειτουργία **Set Speed** άλλαξε την τιμή **speed** και κάνε την 5, όπως φαίνεται στην εικόνα 17.



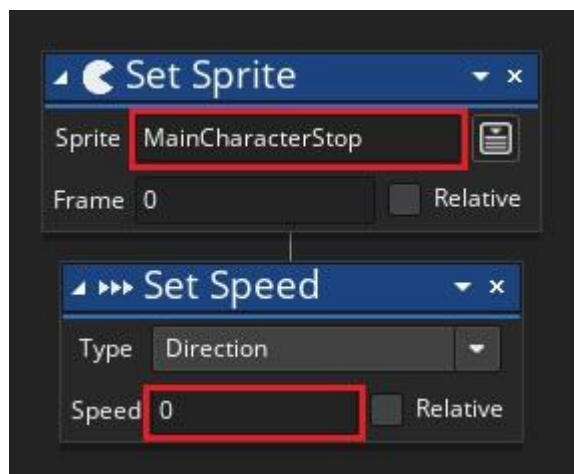
Εικόνα 17. Επιλογή παραμέτρων στις ενέργειες Set Direction Fixed και Set Speed

## GOAL

Όρισε στο ίδιο object τρία (3) ακόμα events για τις τρεις κινήσεις που πρέπει να κάνει ο χαρακτήρας μας ώστε να πηγαίνει κάτω, δεξιά και αριστερά, βάζοντας μέσα στο event τις ίδιες τρεις ενέργειες (αλλαγή sprite, καθορισμός κατεύθυνσης, καθορισμός ταχύτητας).

Αφού καθορίσεις όλες τις κινήσεις, πρέπει να καθορίσεις και την ακινησία σε περίπτωση που ο χρήστης του παιχνιδιού ΔΕΝ πατάει κανένα πλήκτρο.

1. Αρχικά δημιούργησε ένα καινούργιο **Event (Key Down -> No Key)**.
2. Μετά στο παράθυρο λειτουργιών του **Event** καθόρισε 2 λειτουργίες:
  - a. **Set Sprite**, όπου θα επιλέξεις το **MainCharacterStop** και
  - b. **Set Speed**, όπου θα βάλεις μηδενική ταχύτητα, έτσι ώστε να σταματάει, όπως φαίνεται στην εικόνα 18.
3. Τέλος, κλείσε το παράθυρο.



Εικόνα 18. Καθορισμός παραμέτρων στις ενέργειες Set Sprite και Set Speed



Συγχαρητήρια! Ολοκλήρωσες τη λειτουργία του **object** “MainCharacter”.

## ΒΗΜΑ ΤΡΙΤΟ: Δημιουργία room

? Πού θα κινείται όμως το Object;

Είναι ώρα να φτιάξουμε τον κόσμο μας, δηλαδή το room (δωμάτιο) στο οποίο θα υπάρχει και θα κινείται ο MainCharacter που έχουμε φτιάξει. Όμως, τι είναι το Room;

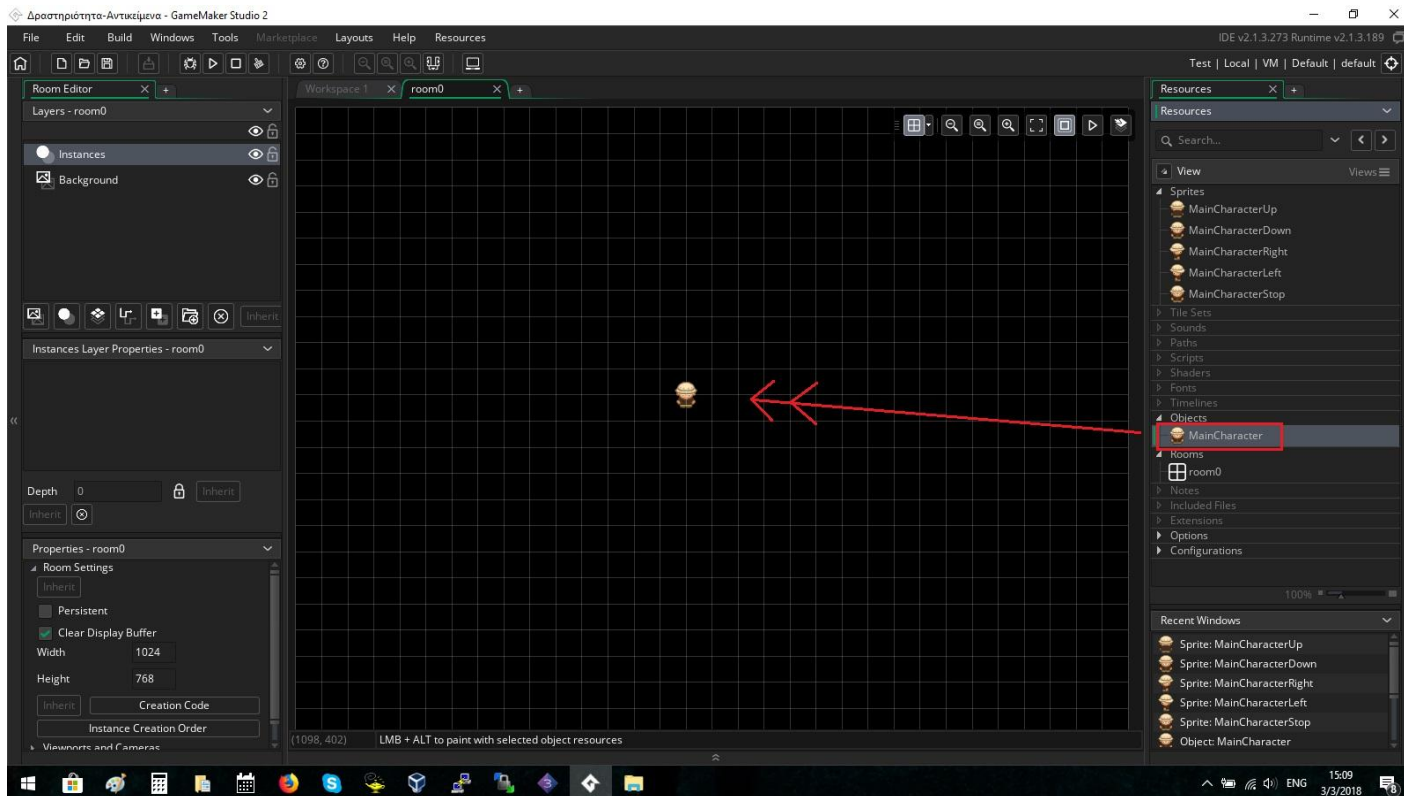


Σε ένα room (δωμάτιο) μπορούμε να τοποθετήσουμε τα objects που έχουμε φτιάξει και έτσι να δημιουργήσουμε τη σκηνή του παιχνιδιού, δηλαδή τον κόσμο μας. Ο τρόπος με τον οποίο θα τοποθετηθούν τα objects μέσα στο room θα είναι και αυτός με τον οποίο θα εμφανιστούν όταν θα ξεκινήσει να εκτελείται το πρόγραμμα που φτιάξαμε.

**ΠΡΟΣΟΧΗ:** Στο παράθυρο επεξεργασίας του room μπορούμε να καθορίσουμε πώς θα είναι ο κόσμος μας. Δεν είναι η εκτέλεση του προγράμματος. Δεν μπορούμε δηλαδή να παίξουμε ακόμα.

Για να φτιάξεις το **Room** κάνε τα παρακάτω:

1. Κάνε διπλό κλικ στο **room0**, που βρίσκεται στη μπάρα πόρων (**Resources**).
2. Ανοίγει το παράθυρο επεξεργασίας του **Room**.
3. Τώρα πρέπει να βάλεις τον ήρωά στο **Room**. Σύρε λοιπόν από τη μπάρα πόρων το **object** “MainCharacter”, μέσα στο μαύρο πλαίσιο με τα τετράγωνα, όπως φαίνεται στην εικόνα 19.

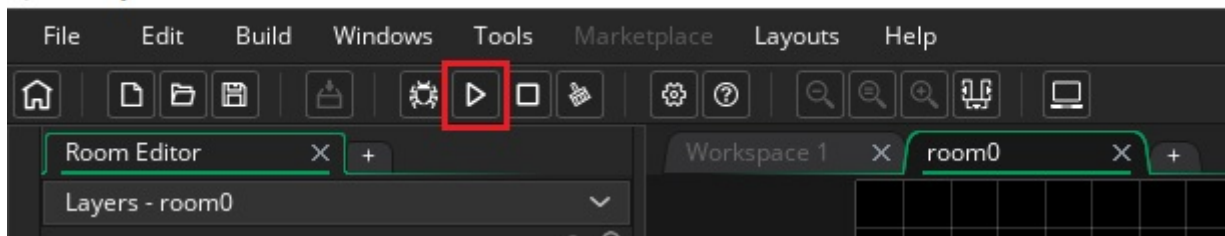


Εικόνα 19. Δημιουργία δωματίου και τοποθέτηση του object “MainCharacter” μέσα σε αυτό

Τώρα είσαι έτοιμος να παίξεις το παιχνίδι που φτιάξαμε.

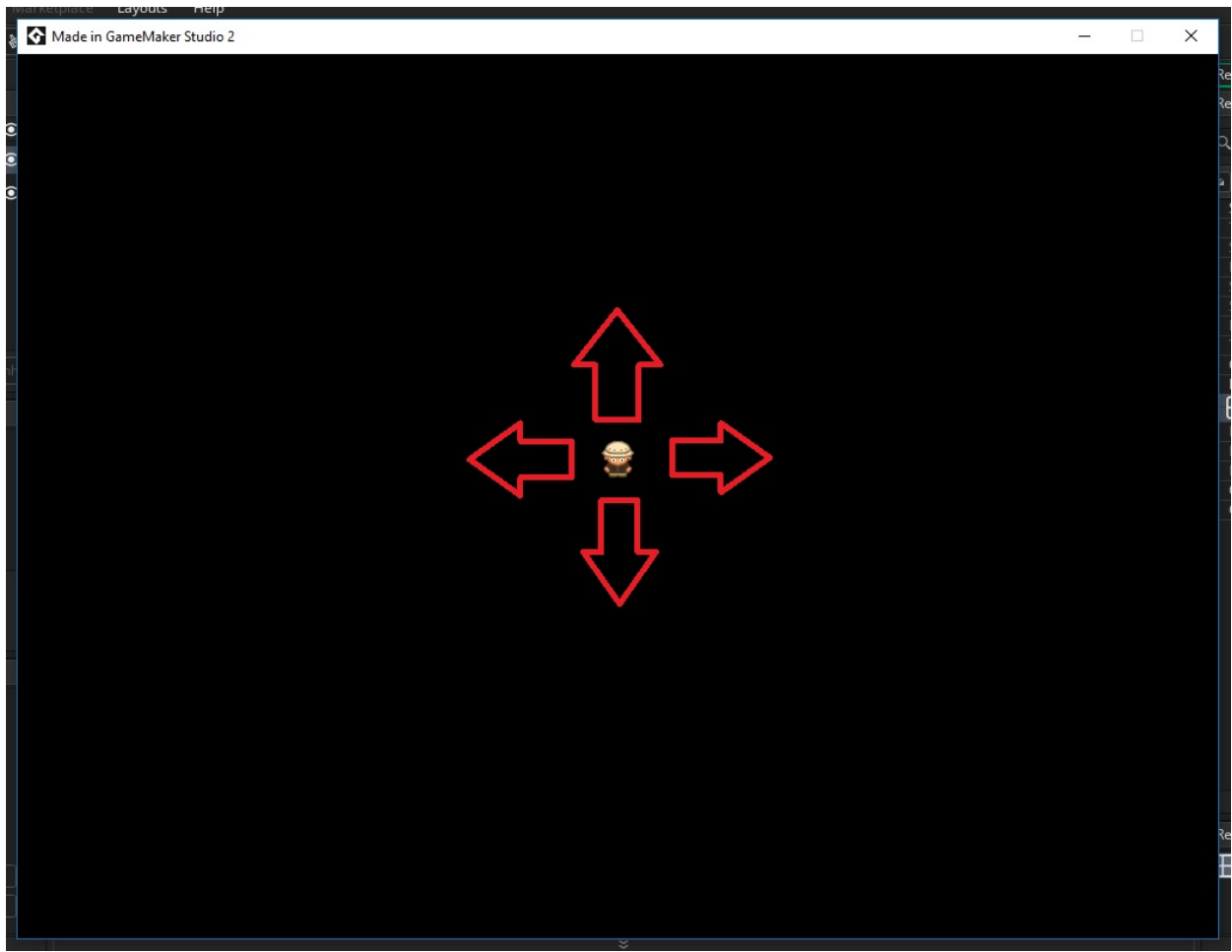
Κάνε κλικ στο κουμπί **Run** (τριγωνάκι) που βρίσκεται στο πάνω μέρος του προγράμματος, όπως φαίνεται στην εικόνα 20.

Activity - GameMaker Studio 2



Εικόνα 20. Toolbar στο πάνω μέρος του προγράμματος και το κουμπί Run

Στην εικόνα 21 εμφανίζεται το παράθυρο στο οποίο μπορείς να παίξεις.



Εικόνα 21. Το παράθυρο του παιχνιδιού

**Καλή Διασκέδαση!!**



- Πώς μπορούμε να αλλάξουμε την ταχύτητα του object μας σε 10 όταν ο χρήστης του προγράμματος πατάει το δεξί βελάκι;
- Τι θα χρειαζόμασταν αν θέλαμε να φτιάξουμε και έναν ακόμα χαρακτήρα ο οποίος να κινείται με τα πλήκτρα W,S,A και D;
- Τι θα συνέβαινε αν βάζαμε στο room ακόμα μία φορά τον χαρακτήρα που έχουμε φτιάξει σε άλλη θέση;

## Do!

- Διπλασιάστε την ταχύτητα του χαρακτήρα σε όλες τις κινήσεις.
- Δημιουργήστε άλλο ένα object "Diamond" έτσι ώστε όταν ο χαρακτήρας το αγγίξει (collision event) μέσα στο room, αυτό να εξαφανίζεται (DESTROY event).
- Αλλάξτε τη λειτουργία του βασικού object, δηλαδή του χαρακτήρα, έτσι ώστε όταν ο χρήστης κάνει κλικ με το ποντίκι πάνω του, να σταματάει το παιχνίδι.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2 GAME MAKER

Συνεχίζουμε το παιχνίδι μας με μεταβλητές και δομή επιλογής !



### Αλγοριθμικές έννοιες:

Μεταβλητή (Αρχικοποίηση και Εμφάνιση), Απλή Δομή Επιλογής

### Έννοιες Προγραμματισμού Παιχνιδιών:

Γεγονός Σύγκρουσης (Collision Event), Μεταβλητή Σκορ

## ? Τι έχουμε φτιάξει ως τώρα;

Στην 1<sup>η</sup> δραστηριότητα φτιάξαμε ένα απλό παιχνίδι στο οποίο ο χρήστης, χρησιμοποιώντας τα βελάκια του πληκτρολογίου, μετακινούσε στην οθόνη έναν χαρακτήρα. Μέσα από το πρώτο μας παιχνίδι μάθαμε:

- Τι είναι τα **objects** (αντικείμενα)
- Τι είναι τα **sprites** (εικόνες)
- Πως δημιουργούμε **events** (γεγονότα)
- Πως εντάσσουμε ενέργειες σε κάθε ένα από αυτά τα γεγονότα

## ? Τι θα μάθουμε;

Στο πλαίσιο της 2<sup>ης</sup> δραστηριότητας θα ασχοληθούμε με μεταβλητές καθώς και με τη δομή επιλογής.

Η μεταβλητή έχει διττή υπόσταση: φυσική ως χώρος αποθήκευσης και εννοιολογική ως σημασία/ρόλο που τις προσδίδουμε. Η μεταβλητή έχει μία τιμή. Μπορούμε να αλλάξουμε την τιμή της, δηλαδή το περιεχόμενο, να το εμφανίσουμε στην οθόνη, ή ακόμα όταν συμβεί κάποιο **event** να ελέγξουμε ποια τιμή έχει ώστε να κάνουμε ενέργειες όπως τερματισμός παιχνιδιού δηλαδή **exit**.

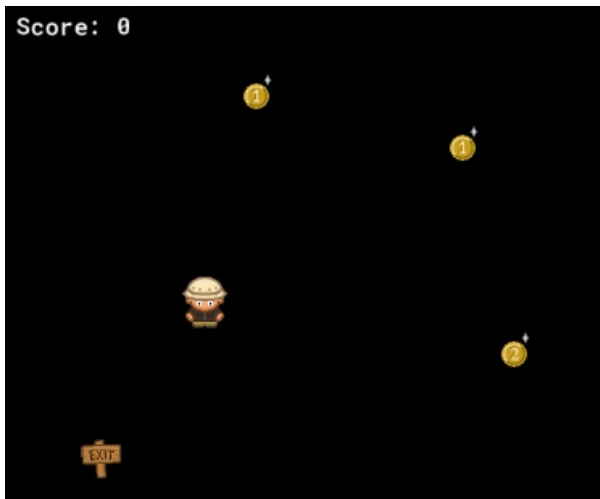
Επίσης σε αυτή τη δραστηριότητα θα μάθουμε πως μπορούμε να επιλέγουμε ποια ενέργεια θα γίνει με βάση μία συνθήκη. Δηλαδή να μην είναι προκαθορισμένες οι ενέργειες αλλά να επιλέγονται ανάλογα με την τιμή μιας μεταβλητής. Αυτή η δομή λέγεται δομή επιλογής.

## ? Τι θα φτιάξουμε;

Θα επεκτείνουμε το παιχνίδι μας ως εξής:

- Θα δημιουργήσουμε μια μεταβλητή η οποία θα δηλώνει το σκορ, και θα εμφανίζεται πάνω δεξιά στην οθόνη.
- Θα φτιάξουμε 2 καινούργια **objects** ένα για ένα νόμισμα αξίας 1 και για ένα νόμισμα αξίας 2, τα οποία θα βρίσκονται διάσπαρτα μέσα στο **room** και θα προσπαθεί ο παίκτης να τα μαζέψει, καταφέροντας έτσι να ανεβάσει το σκορ του.
- Θα δημιουργήσουμε ένα τελευταίο **object** τερματισμού, στο οποίο θα πηγαίνει ο παίκτης αφού έχει συγκεντρώσει score, για να τερματίσει.

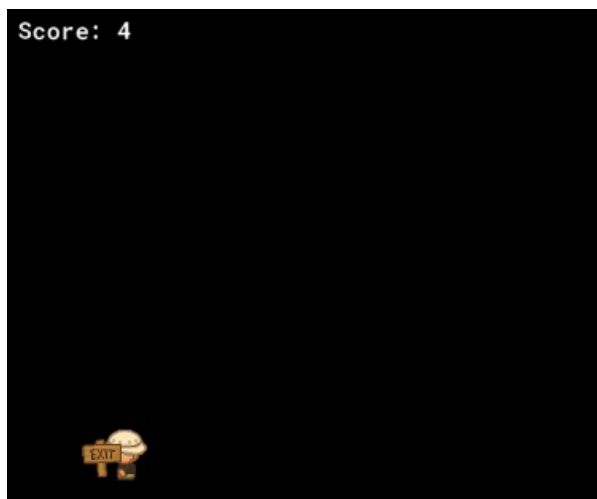
Η λειτουργία του παιχνιδιού φαίνεται στις εικόνες 1,2 και 3.



Εικόνα 1. Αρχική θέση παίκτη με σκορ στο μηδέν.

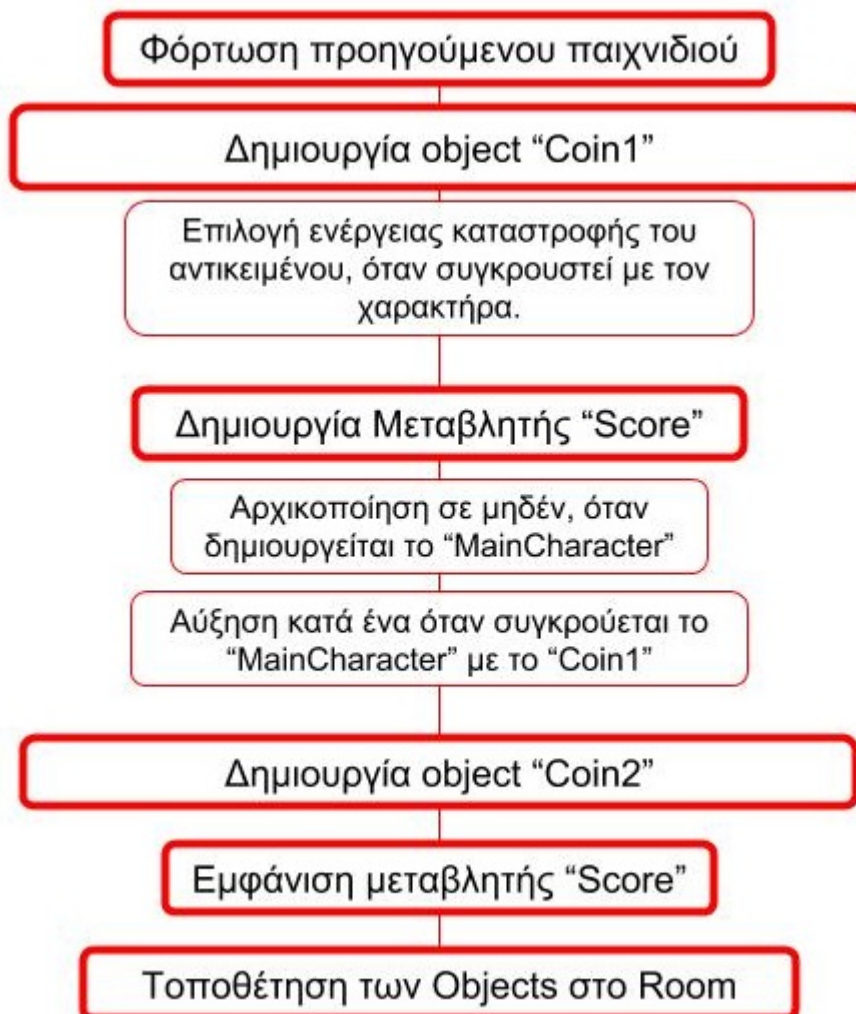


Εικόνα 2. Ο παίκτης έχει μαζέψει δύο νομίσματα αξίας 1 και έτσι το σκορ έχει ανέβει στο 2.



Εικόνα 3. Ο παίκτης έχει μαζέψει όλα τα νομίσματα και φτάνει στην έξοδο όπου εκεί τελειώνει το παιχνίδι.

## Διάγραμμα Ροής Δραστηριότητας





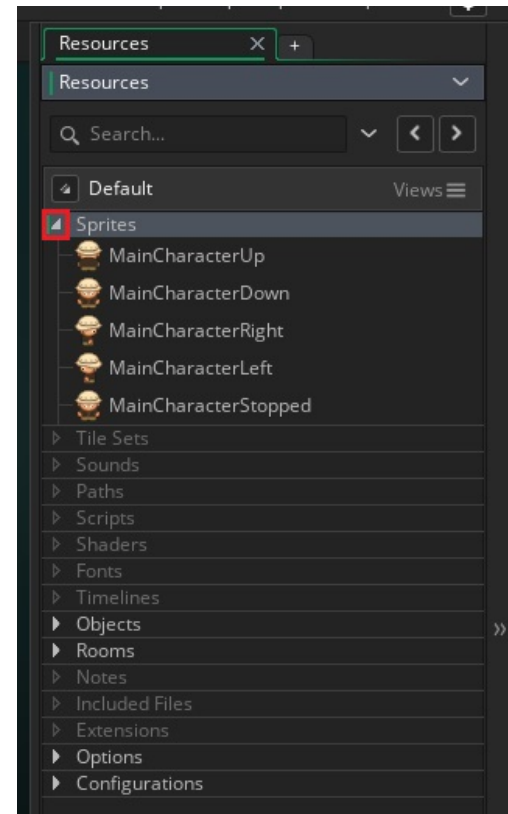
Εκμάθηση προγραμματισμού μέσω της ανάπτυξης παιχνιδιού

Ας ξεκινήσουμε λοιπόν!

Εκτέλεσε το **Game Maker**, στην αρχική οθόνη του προγράμματος, πάτησε **Open**, στο παράθυρο που εμφανίζεται βρες το προηγούμενο πρόγραμμα και κάνε διπλό κλικ για να ανοίξει.

Στην μπάρα των **Resources** (πόρων) δεξιά, υπάρχουν όλα τα **sprites**, **objects** και **rooms**, που έχεις δημιουργήσει (εικόνα 4).

Εικόνα 4. Η μπάρα Resources με τα Resources που έχουν δημιουργηθεί



## ΒΗΜΑ ΠΡΩΤΟ: Δημιουργία πρώτου νομίσματος με αξία 1

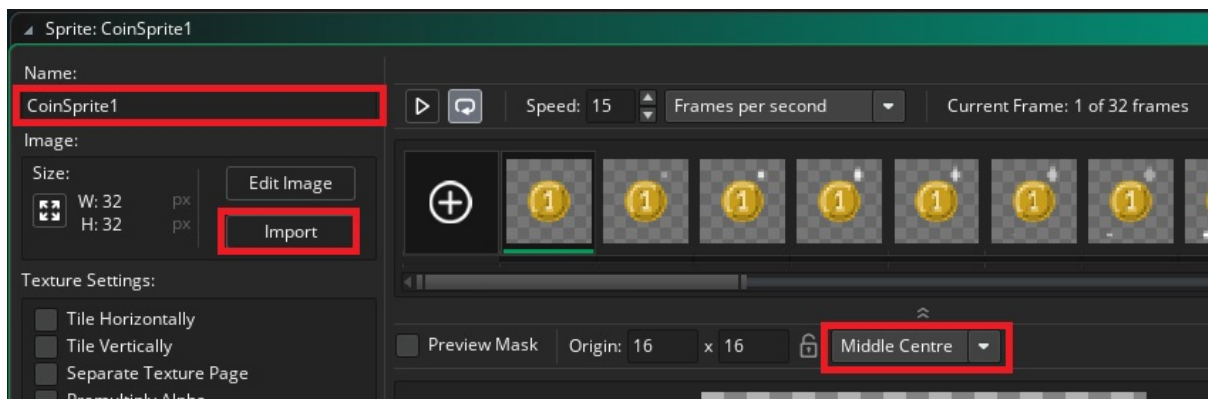


Δημιούργησε ένα καινούργιο **Sprite** και **Object** για το πρώτο νόμισμα.

Φτιάξε ένα καινούργιο **sprite**, όπως ακριβώς μάθαμε στην προηγούμενη δραστηριότητα.

1. Κάνε δεξί κλικ στα **sprites** της μπάρας πόρων και επίλεξε **Create Sprite**
2. Ονόμασέ το: "CoinSprite1"
3. Επίλεξε το αρχείο: coin1\_sparkle\_strip32.png από το φάκελο **Resources**, πατώντας στο κουμπί **import**.

- Τέλος, άλλαξε την επιλογή **Top\_Left** σε **Middle\_Centre**, όπως φαίνεται στην εικόνα 5.



Εικόνα 5. Δημιουργία Sprite "CoinSprt1"

- Τέλος, κλείσε το παράθυρο επεξεργασίας του **sprite**.

Τώρα φτιάξε ένα καινούργιο **object**, όπως μάθαμε στην προηγούμενη δραστηριότητα, ως εξής:

- Κάνε δεξί κλικ στα **objects**, και διάλεξε **Create Object**.
- Ονόμασέ το: "Coin1"
- Διάλεξε το **sprite** με όνομα "CoinSprite1", που μόλις δημιούργησες.

**?** Πώς μπορεί το **object** να εκτελεί συγκεκριμένες ενέργειες κάθε φορά που αγγίζει ένα άλλο;



Το **Collision Event**, είναι ένα event το οποίο ενεργοποιείται κάθε φορά που ένα **object A** αγγίζει ένα άλλο **object**. Αυτό σημαίνει ότι μέσα στο **room** όταν το **A** έρθει σε επαφή με το **B**, θα εκτελεστούν οι ενέργειες που έχουμε βάλει να γίνονται μέσα στο event αυτό.

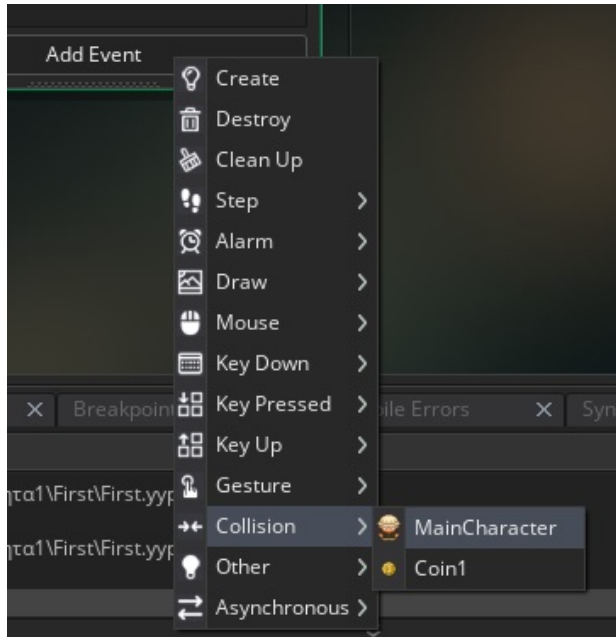
**ΠΡΟΣΟΧΗ:** Μπορεί να υπάρχει **Collision Event** και στα δύο **objects** που πρόκειται να συγκρουστούν. Όταν συγκρουστούν θα γίνουν όλες οι ενέργειες που έχουμε ορίσει και στα δύο **Objects!!**



Κατασκεύασε Event και ενέργειες στο object "Coin1", έτσι ώστε όταν συγκρούεται με το "MainCharacter", να εξαφανίζεται.

Στο παράθυρο επεξεργασίας, που βρίσκεσαι, δηλαδή του **object** "Coin1" και ΟΧΙ του "MainCharacter", ακολούθησε τα παρακάτω βήματα:

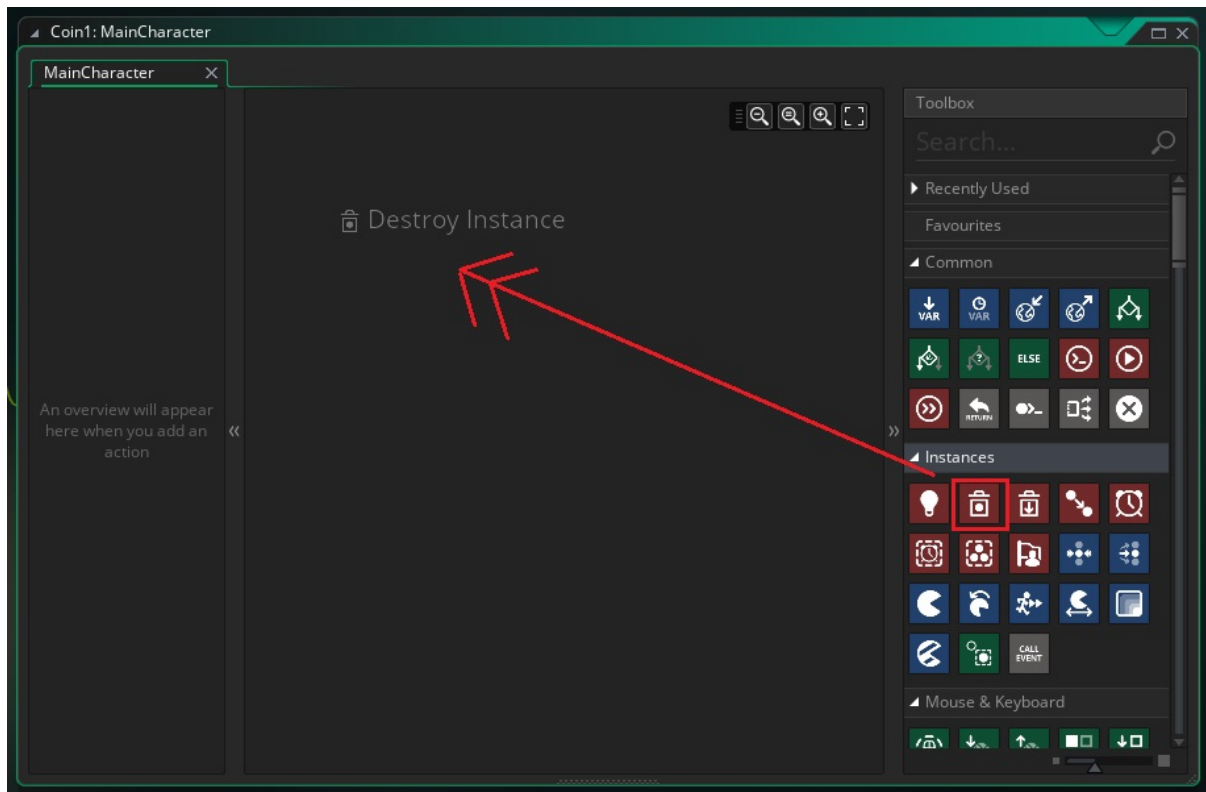
1. Κάνε κλικ στο κουμπί **Add Event**
2. Στο μενού που θα εμφανιστεί σύρε το ποντίκι στην επιλογή **Collision** και μετά κάνε κλικ στο "MainCharacter", όπως φαίνεται στην εικόνα 6.



Εικόνα 6. Δημιουργία Collision Event

Τώρα μπορείς να καθορίσεις τις ενέργειες που θα κάνει το πρώτο νόμισμα, μόλις το αγγίξει ο βασικός χαρακτήρας που έχουμε ήδη από την πρώτη δραστηριότητα.

3. Στο παράθυρο ενεργειών που εμφανίζεται, θέλουμε μία ενέργεια που να εξαφανίζει το "Coin1". Αυτό γίνεται με την ενέργεια **Destroy Instance**. Σύρε από την **μπάρα ενεργειών** το εικονίδιο με τον κάδο το οποίο αντιστοιχεί στην καταστροφή **object** και λέγεται **Destroy Instance**, όπως φαίνεται στην εικόνα 7.



Εικόνα 7. Εισαγωγή ενέργειας Destroy Instance στο Collision Event του Coin1 με το MainCharacter

4. Τέλος, κλείσε το παράθυρο.

Συγχαρητήρια! Μόλις κατασκεύασες το πρώτο νόμισμα.

Πριν φτιάξεις και το δεύτερο θα δημιουργήσεις τη μεταβλητή του "score".

## ΒΗΜΑ ΔΕΥΤΕΡΟ: Δημιουργία του Score

? Πώς μπορούμε να έχουμε score στο παιχνίδι μας;



Η μεταβλητή είναι ένας χώρος στη μνήμη του προγράμματος που φτιάχνουμε στον οποίο θα αποθηκεύεται ένας αριθμός. Αυτός ο αριθμός μπορεί να είναι οτιδήποτε χρειαζόμαστε μέσα σε ένα παιχνίδι. Κατά τη διάρκεια του προγράμματος μπορούμε να αλλάξουμε αυτόν τον αριθμό, να τον εμφανίσουμε στην οθόνη αλλά και να καθορίσουμε ενέργειες που μπορούν να γίνονται ανάλογα με το ποια είναι η τιμή του.

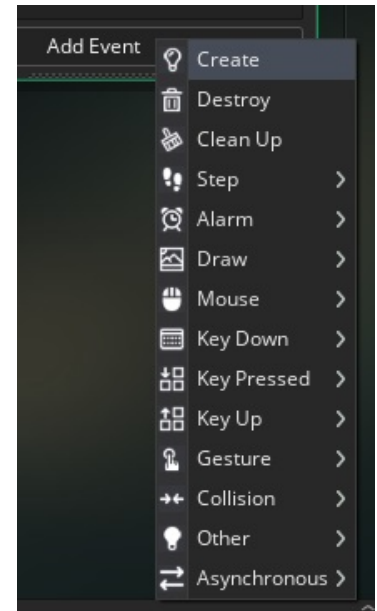
Στο παιχνίδι, θέλουμε η μεταβλητή του **score** αρχικά να έχει τιμή 0 και στη συνέχεια κάθε φορά που ο παίκτης θα μαζεύει τα νομίσματα, αυτή να αυξάνεται.

## GOAL

Αρχικοποίησε τη μεταβλητή "score" που βρίσκεται στο object "MainCharacter" σε 0 καθώς αυτό δημιουργείται.

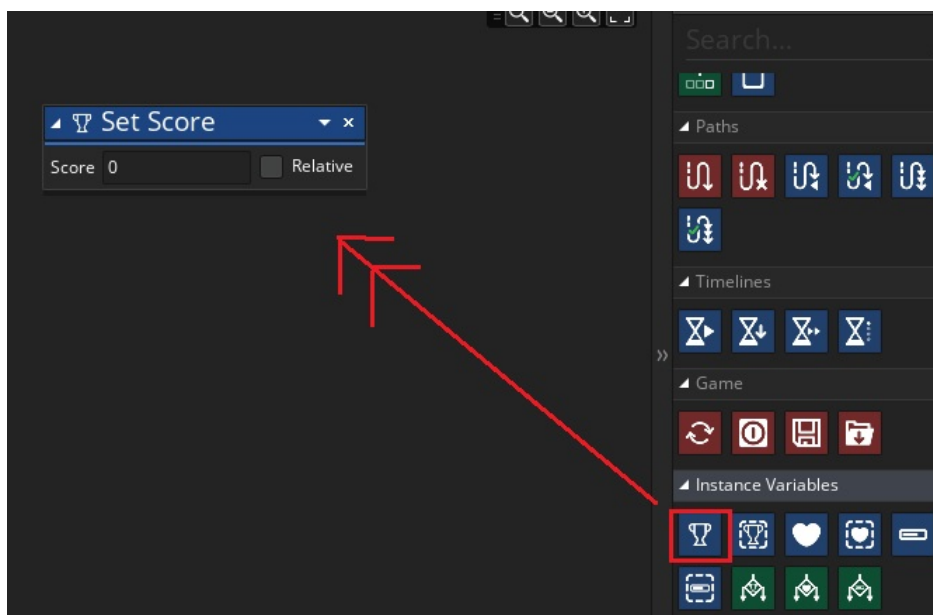
Αρχικά άνοιξε το παράθυρο επεξεργασίας του **object** "MainCharacter", κάνοντας διπλό κλικ σε αυτό στη **μπάρα** των πόρων. Τώρα ακολούθησε τα παρακάτω βήματα:

1. Κάνε κλικ στο κουμπί **Create Event** που βρίσκεται στο μικρό παραθυράκι με όλα τα **events** που έχει ήδη το **object** "MainCharacter".
2. Στο μενού που θα εμφανιστεί διάλεξε **Create**, όπως φαίνεται στην εικόνα 8.



Εικόνα 8. Επιλογή Create Event

3. Στο παράθυρο των ενεργειών που θα εμφανιστεί σύρε την ενέργεια **Set Score**, που έχει εικονίδιο ένα κύπελο, όπως φαίνεται στην εικόνα 9.



Εικόνα 9. Εισαγωγή ενέργειας Set Score

Η μεταβλητή "**score**" θα αρχικοποιείται σε 0, αμέσως μόλις δημιουργείται το **object** "MainCharacter".

Τέλεια! Τώρα είμαστε έτοιμοι να καθορίσουμε την αύξηση της μεταβλητής όταν αγγίζει ο "MainCharacter" το **object** "Coin1".

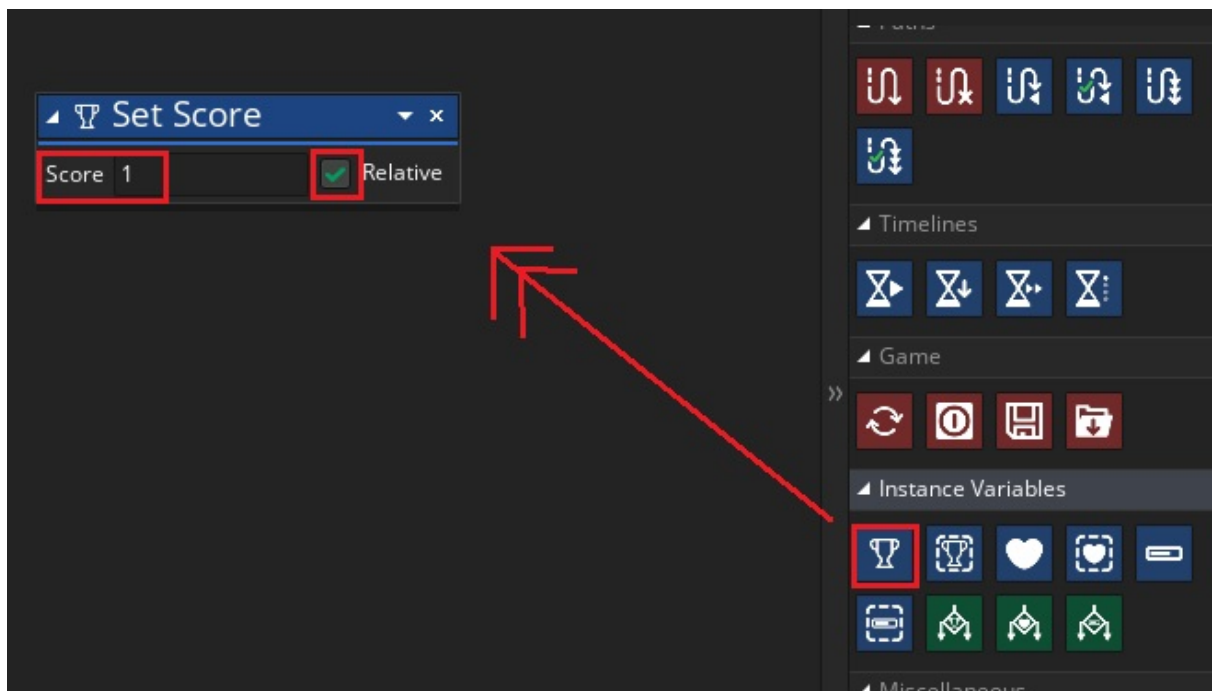
## GOAL

Καθόρισε ένα **Collision event** με το "Coin1" στο **object** "MainCharacter" το οποίο όταν γίνεται **trigger**, δηλαδή όταν καλείται, να αυξάνει τη μεταβλητή "score" κατά ένα.

Στο παράθυρο επεξεργασίας του "MainCharacter" κάνε τα παρακάτω:

1. Δημιούργησε ένα καινούργιο **Collision event** με το **object** "Coin1".
2. Σύρε την ενέργεια **Set Score** από την μπάρα μέσα στο παράθυρο ενεργειών.
3. Στο μικρό παραθυράκι άλλαξε τα εξής, όπως φαίνεται στην εικόνα 10.
  - a. Σβήσε το 0 που υπάρχει σαν τιμή και κάντο 1.
  - b. Τσέκαρε το κουτάκι **Relative**.

Αυτό σημαίνει πως όταν κληθεί αυτό το **event**, η τιμή της μεταβλητής "score" θα ανέβει κατά 1 και ΔΕΝ θα γίνει απλώς 1.



Εικόνα 10. Εισαγωγή της ενέργειας Set Score και καθορισμός των παραμέτρων της

4. Τέλος, κλείσε τα παράθυρα επεξεργασίας του **object** "MainCharacter" για να αποθηκευτούν οι αλλαγές.

Άφογα!! Ολοκλήρωσες όλη τη λειτουργικότητα του πρώτου νομίσματος. Αυτό σημαίνει ότι δημιούργησες **sprite**, **object**, και κάθε φορά που αγγίζει ο βασικός χαρακτήρας το **object** αυτό, θα αυξάνεται η μεταβλητή "score" κατά ένα και θα εξαφανίζεται το νόμισμα.

Τώρα θα φτιάξουμε το δεύτερο νόμισμα.

## ΒΗΜΑ ΤΡΙΤΟ: Δημιουργία δεύτερου νομίσματος αξίας

### 2

? Τι διαφορετικό θα έχει το δεύτερο νόμισμα σε σχέση με το πρώτο;

Το δεύτερο νόμισμα θα είναι ίδιο με το πρώτο μόνο που:

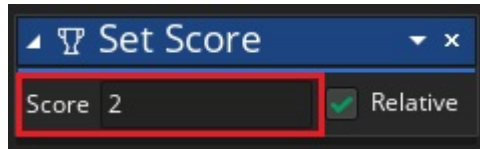
- Θα έχει διαφορετικό **sprite** και
- στο **Collision Event** που θα δημιουργήσουμε από το "MainCharacter" προς αυτό, το **score** θα ανεβαίνει κατά 2 και όχι κατά 1, γιατί η αξία του είναι μεγαλύτερη.



Φτιάξε ένα **object** για το δεύτερο νόμισμα, ακολουθώντας τα βήματα που ακολούθησες για το πρώτο, και όλη τη λειτουργικότητά του, δηλαδή το **collision event** του "MainCharacter" προς αυτό.

Αν δε θυμάσαι ακολούθησε τα παρακάτω βήματα:

1. Αρχικά δημιούργησε το κατάλληλο **sprite** με όνομα "CoinSprite2", και βάλε για εικόνα το αρχείο [coin2\\_sparkle\\_strip32.png](#) που βρίσκεται στο φάκελο Resources.
2. Μετά φτιάξε ένα **object** με όνομα "Coin2", και θέσε σαν **sprite** του, το **sprite** που μόλις δημιούργησες.
3. Στη συνέχεια, όρισε ένα **Collision Event**, από το "Coin2" προς το "MainCharacter", στο οποίο το **object** "Coin2" θα εξαφανίζεται.
4. Αφού τα κάνεις αυτά κλείσε το παράθυρο του "Coin2" και άνοιξε το παράθυρο του **object** "MainCharacter".
5. Τώρα δημιούργησε ένα καινούργιο **Collision Event** από το παράθυρο του "MainCharacter" προς το "Coin2".
6. Σύρε από την μπάρα ενεργειών την ενέργεια **Set Score**, όπως έκανες και για το "Coin1", βάλε στο παραθυράκι την τιμή 2 (και όχι 1), και τσέκαρε το **Relative**, όπως φαίνεται στην εικόνα 11.



Εικόνα 11. Καθορισμός παραμέτρων ενέργειας Set Score

7. Τέλος, κλείσε το παράθυρο επεξεργασίας του **object** "MainCharacter".

## ΒΗΜΑ ΤΕΤΑΡΤΟ: Δημιουργία Object στόχου του παιχνιδιού

Αφού ο παίκτης μαζέψει αρκετά νομίσματα και ανεβάσει το **score** του, θα πρέπει να υπάρχει ένα **object** το οποίο όταν αγγίζει ο βασικός χαρακτήρας θα τελειώνει το παιχνίδι.

? Τι πρέπει να κάνουμε για να τελειώνει το παιχνίδι κάθε φορά που φτάνει ο παίκτης σε ένα συγκεκριμένο score και αγγίζει το object εξόδου;

### GOAL

Φτιάξε αρχικά ένα object με όνομα "ExitObject" και sprite [exitsign.png](#), με ένα άδειο Collision Event.

Αν δε θυμάσαι ακολούθησε τα παρακάτω βήματα:

1. Δημιούργησε ένα **sprite** με όνομα "ExitSprite", και **sprite** το [exitsign.png](#) από το φάκελο Resources, όπως έκανες και στα προηγούμενα **sprites**.
2. Φτιάξε ένα **object** με όνομα "ExitObject" και επέλεξε για **sprite** το "ExitSprite".
3. Τέλος, κλείσε το παράθυρο.

? Πώς μπορούμε να ορίσουμε να γίνεται ενέργεια εξόδου από το παιχνίδι μόνο όταν το score είναι 10;



Η δομή επιλογής χρησιμεύει όταν θέλουμε να γίνονται ενέργειες κάτω από μία συνθήκη, δηλαδή μόνο αν μία μεταβλητή έχει κάποια συγκεκριμένη τιμή. Σε αυτή την περίπτωση όταν καλείται το event μέσα στο



οποίο υπάρχει η δομή επιλογής, ελέγχεται η συνθήκη και μόνο αν είναι αληθής εκτελούνται οι εντολές που βρίσκονται μέσα σε αυτήν.

Η δομή επιλογής διαβάζεται ως εξής: AN <Συνθήκη είναι αληθής> TOTE <Εκτέλεσε τις εντολές>

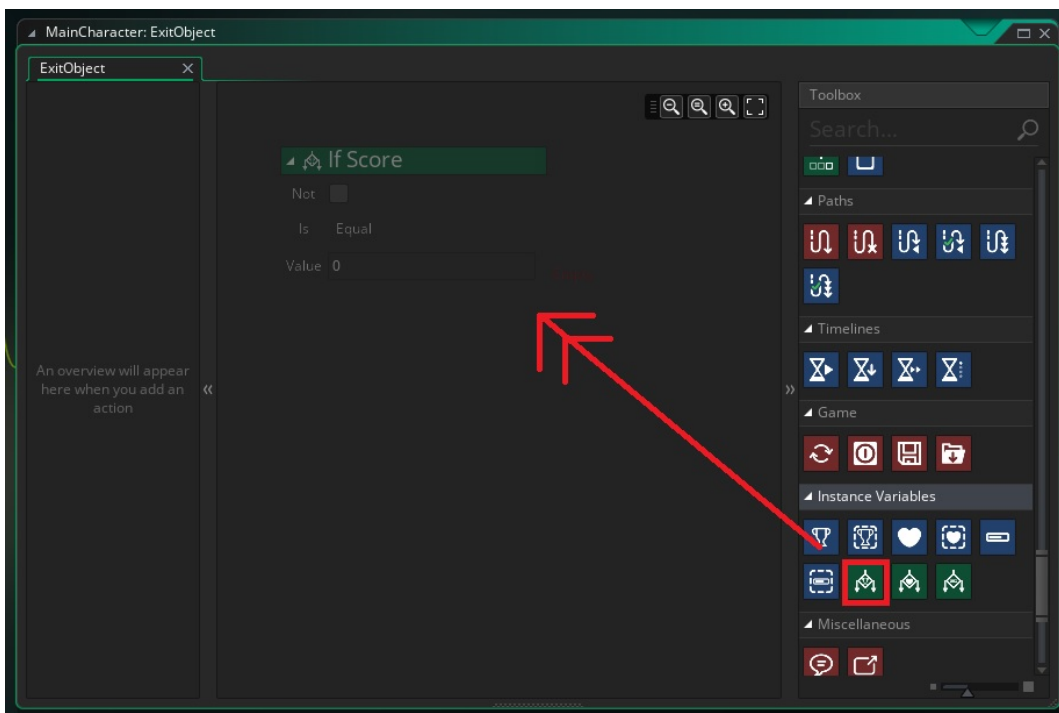
Στην περίπτωση που δεν είναι αληθής οι εντολές ΔΕΝ εκτελούνται.

## GOAL

Χρησιμοποίησε τη δομή επιλογής μέσα σε Collision Event στο object "MainCharacter", προκειμένου να τερματίζει το παιχνίδι μόνο όταν το score είναι μεγαλύτερο ή ίσο με 10.

Ακολούθησε τα παρακάτω βήματα:

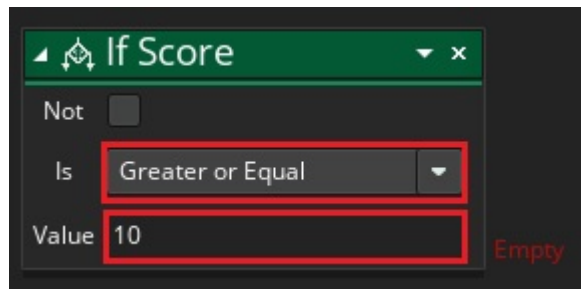
1. Άνοιξε το παράθυρο επεξεργασίας του "MainCharacter" και πρόσθεσε ένα τελευταίο **Collision Event** με το "ExitObject".
2. Σύρε από τη μπάρα ενεργειών την ενέργεια **If Score**, (Βρίσκεται αρκετά κάτω στη λίστα) όπως φαίνεται στην εικόνα 12.



Εικόνα 12. Εισαγωγή της ενέργειας ελέγχου If Score

3. Στο μικρό παάθυρο με τις επιλογές της συνθήκης που θα εμφανιστεί άλλαξε τα εξής:
  - a. Η επιλογή του **is** πρέπει να γίνει **Greater or Equal**, δηλαδή μεγαλύτερο ή ίσο.
  - b. Η τιμή, δηλαδή το **Value** πρέπει να γίνει 10.

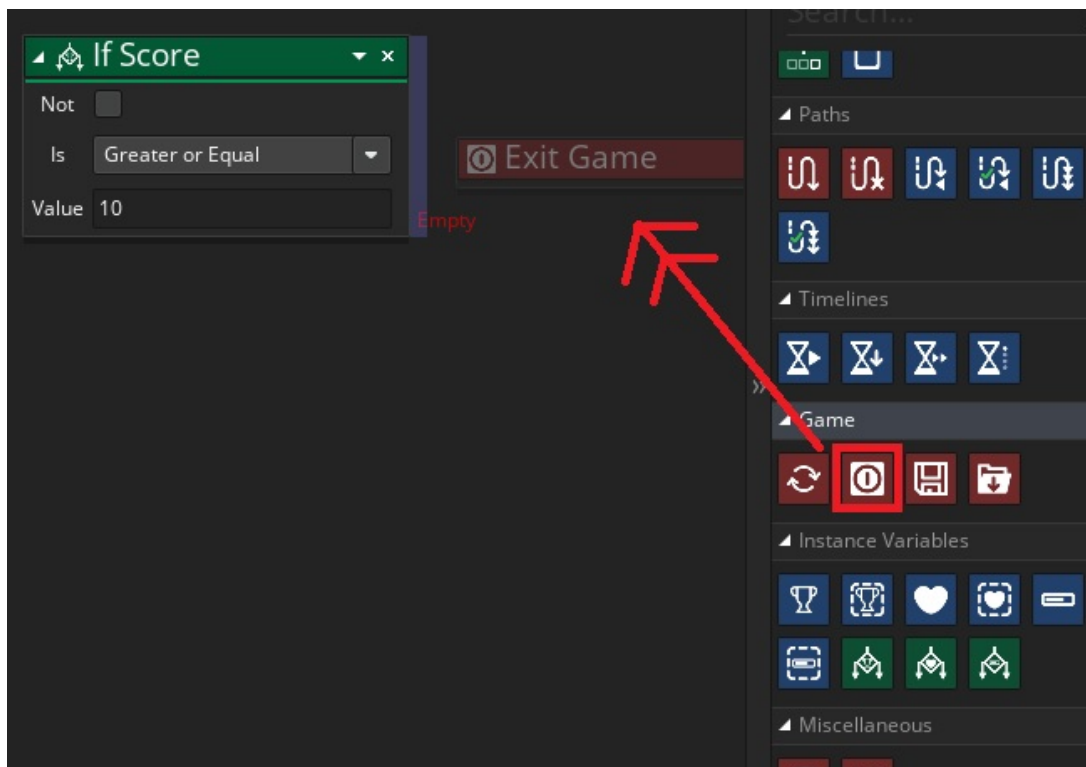
Αυτό σημαίνει πρακτικά ότι η συνθήκη θα διαβάζεται: “Αν το score είναι μεγαλύτερο ή ίσο του 10”.



Εικόνα 13. Καθορισμός παραμέτρων συνθήκης

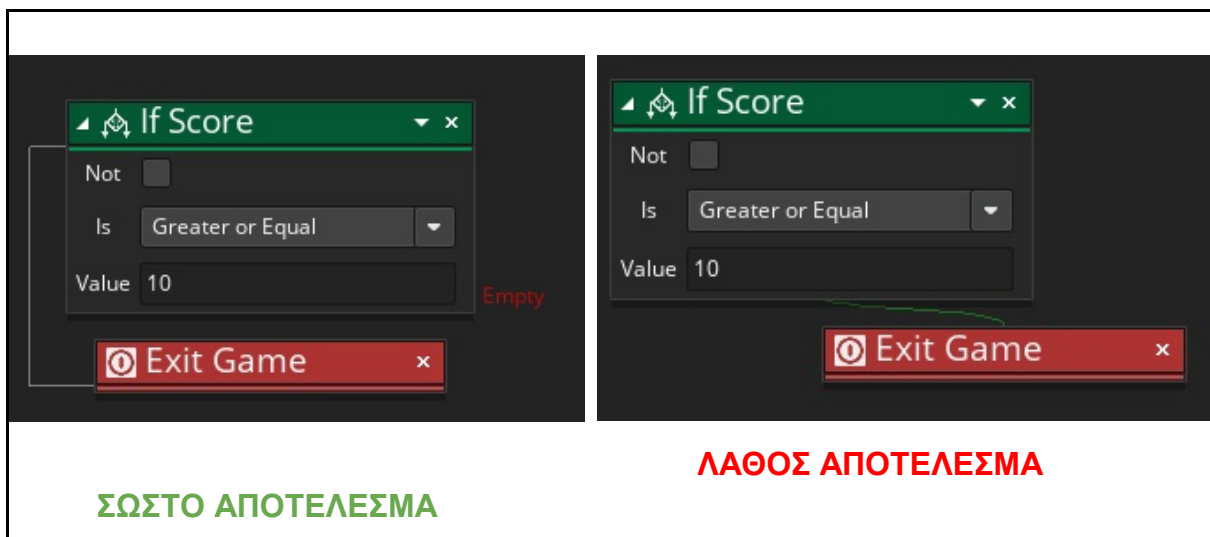
Παρατήρησε ότι δίπλα από το μικρό αυτό παράθυρο εμφανίζεται η λέξη “Empty”(άδειος). Αυτό σημαίνει ότι δεν έχουμε βάλει ενέργειες που θα εκτελεστούν αν ισχύει η συνθήκη.

4. Τώρα πρέπει να εντάξουμε την ενέργεια εξόδου από το παιχνίδι, **Exit Game**, μέσα στη δομή επιλογής. Σύρε την ενέργεια **Exit Game**, στα ΔΕΞΙΑ του μικρού αυτού παραθύρου **If Score**, όπως φαίνεται στην εικόνα 14 και ΟΧΙ κάτω από αυτό.



Εικόνα 14. Εισαγωγή ενέργειας Exit Game μέσα στη δομή επιλογής της ενέργειας If Score

Το αποτέλεσμα αυτού το βήματος πρέπει να είναι σύμφωνο με το δεξί μέρος της εικόνας 15.



Εικόνα 15. Το λάθος και το σωστό αποτέλεσμα της εισαγωγής της ενέργειας Exit Game

5. Τέλος, κλείσε το παράθυρο.

## ΒΗΜΑ ΠΕΜΠΤΟ: Εμφάνιση του score στην οθόνη

? Πώς μπορούμε να εμφανίσουμε την τιμή μιας μεταβλητής, όπως το score, στην οθόνη;



Το Draw GUI Event δεν είναι ένα event όπως όλα όσα έχουμε μάθει. Είναι ένα event μέσα στο οποίο μπορούμε να βάλουμε ενέργειες σχεδιασμού σχημάτων ή εκτύπωσης / εμφάνισης κειμένου στην οθόνη του χρήστη. Αν αυτές τις ενέργειες τις βάλουμε οπουδήποτε αλλού δεν θα λειτουργήσουν.

Στις ενέργειες αυτές χρειάζεται να προσδιοριστεί η θέση της οθόνης στην οποία θέλουμε να εμφανίζεται κάποια μεταβλητή ή κείμενο.

Αυτό γίνεται με τη χρήση συντεταγμένων.

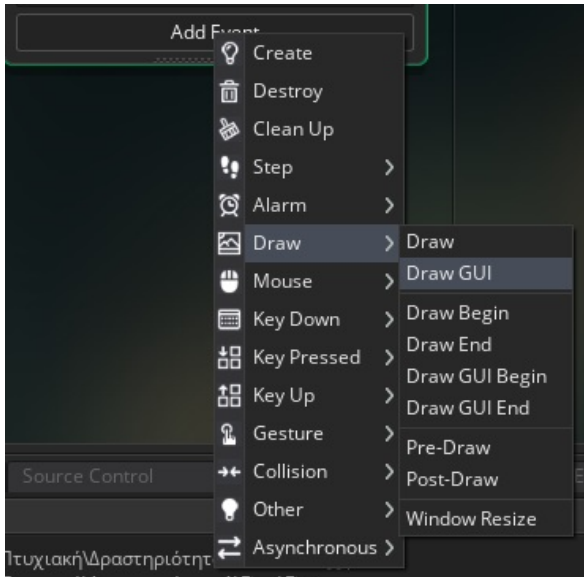
Κάθε σημείο στην οθόνη έχει ένα ζευγάρι συντεταγμένων (X,Y), που αναφέρονται στον οριζόντιο και στον κάθετο άξονα αντίστοιχα. Το σημείο (0,0) βρίσκεται στην πάνω αριστερή γωνία. Καθώς κινούμαστε προς τα δεξιά αυξάνεται η μεταβλητή X και καθώς κινούμαστε προς τα κάτω αυξάνεται η μεταβλητή Y.



Καθόρισε event και ενέργεια στο object "MainCharacter" που να εμφανίζει το score πάνω δεξιά.

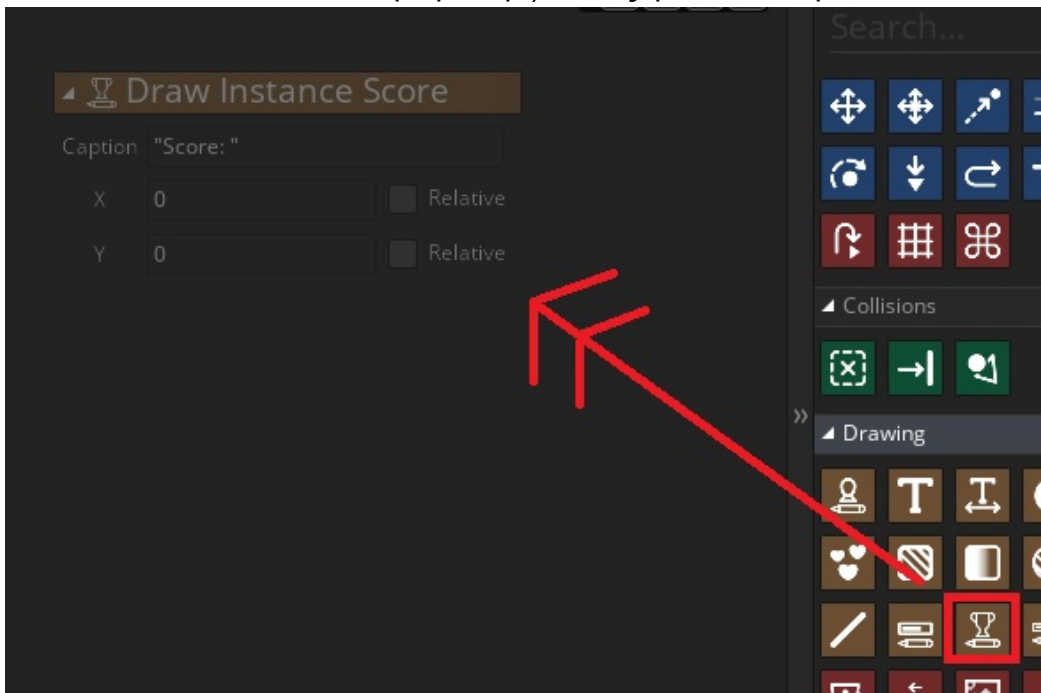
Για να το κάνεις αυτό ακολούθησε τα παρακάτω βήματα:

1. Άνοιξε το παράθυρο επεξεργασίας του MainCharacter
2. Δημιούργησε ένα **Draw GUI Event**, όπως φαίνεται στην εικόνα 16.



Εικόνα 16. Δημιουργία Draw GUI Event

3. Μέσα στο παράθυρο του **Event**, σύρε την ενέργεια **Draw Instance Score** (Το εικονίδιο είναι ένα κύπελλο με μολύβι), όπως φαίνεται στην εικόνα 17.

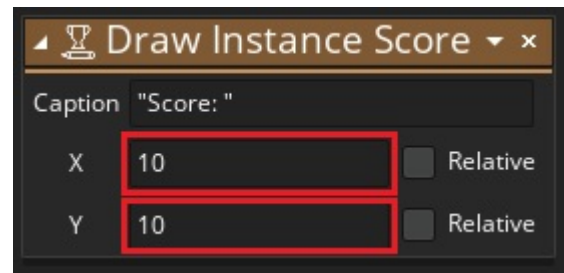


Εικόνα 17. Εισαγωγή Draw Instance Score

4. Στο παραθυράκι βάλε για X και για Y συντεταγμένες το 10, όπως φαίνεται στην εικόνα 18.

Αυτό σημαίνει ότι θα εμφανιστεί το κείμενο "Score:" λίγο πιο μέσα από την πάνω αριστερά γωνία.

5. Τέλος, κλείσε το παράθυρο.



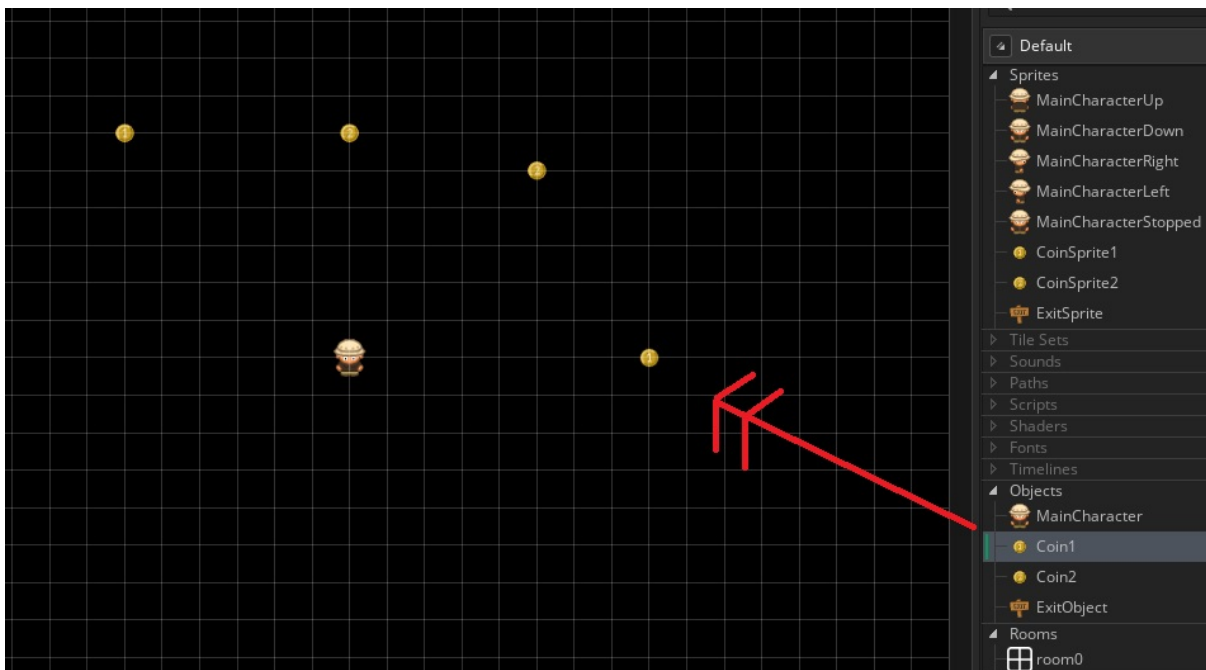
Εικόνα 18. Καθορισμός συντεταγμένων X,Y

Τώρα είμαστε έτοιμοι να βάλουμε τα **objects** μας στο **room** που θα παίζει ο παίκτης.

## ΒΗΜΑ ΕΚΤΟ: Δημιουργία του Room

### GOAL

Άνοιξε το Room, όπως μάθαμε στην προηγούμενη δραστηριότητα και πρόσθεσε μέσα σε αυτό ένα "ExitObject" και αρκετά νομίσματα και από τα δύο είδη, ούτως ώστε να επαρκούν για να μπορεί ο παίκτης να φτάσει στο επιθυμητό score. Μπορείς να βάλεις όσα θέλεις και όπου τα θέλεις εσύ!



Εικόνα 19. Τοποθέτηση αντικειμένων στο room0

Αφού κλείσεις το παράθυρο. Κάνε κλικ στο κουμπί Play (τριγωνάκι) που βρίσκεται στην εργαλειοθήκη στο πάνω μέρος του προγράμματος, για να παίξεις.

## Καλή Διασκέδαση!



- Τι είναι μία μεταβλητή μέσα σε ένα object; Δώσε ένα παράδειγμα μιας άλλης μεταβλητής που θα μπορούσε να υπάρχει σε ένα παιχνίδι.
- Σε τι χρησιμεύει ένα Collision Event;
- Πώς θα μπορούσαμε να ελέγξουμε αν το score είναι μηδέν μέσα σε ένα Event;
- Τι θα συμβεί αν βάλουμε μία ενέργεια εμφάνισης μιας μεταβλητής στην οθόνη σε ένα Collision Event; Θα εμφανιστεί η μεταβλητή ή όχι, και γιατί;

## Do!

- Τροποποίησε το πρόγραμμα έτσι ώστε από την αρχή το score να είναι 5.
- Άλλαξε την αξία του δεύτερου νομίσματος έτσι ώστε το score να ανεβαίνει κατά 4.
- Άλλαξε τη συνθήκη τερματισμού του παιχνιδιού έτσι ώστε να ισχύει μόνο όταν το score είναι ίσο με 20.
- Δημιούργησε ένα object το οποίο όταν έρχεται σε επαφή με τον χαρακτήρα να ξαναξεκινάει το παιχνίδι από την αρχή (Restart).

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3 GAME MAKER

Συνεχίζουμε με Δομή Επανάληψης και Τυχαιότητα !



Αλγοριθμικές έννοιες:

Δομή Επανάληψης (απλή)

Έννοιες Προγραμματισμού Παιχνιδιών:

Μεταβλητή Ζωής, Δημιουργία Αντικειμένου (Create Instance)

? Τι έχουμε φτιάξει ως τώρα;

Στη 2<sup>η</sup> Δραστηριότητα επεκτείναμε το παιχνίδι ως εξής: Ο βασικός χαρακτήρας μαζεύει νομίσματα αξίας 1 ή 2 με σκοπό να ανεβάσει το σκορ του. Τέλος, όταν ο παίκτης μαζέψει όλα τα νομίσματα κατευθύνεται προς το **object** εξόδου όπου και τερματίζει το παιχνίδι. Πιο συγκεκριμένα αντιμετωπίσαμε τις εξής έννοιες:

- Collision Event
- Μεταβλητή
- Δομή επιλογής
- Draw GUI Event

? Τι θα μάθουμε;

Στο πλαίσιο της 3<sup>ης</sup> δραστηριότητας θα ασχοληθούμε με τη δομή επανάληψης και την παραγωγή τυχαίων αριθμών.

Στον προγραμματισμό όταν θέλουμε να επαναλάβουμε την ίδια εντολή ή διαδικασία πολλές φορές αποφεύγουμε να την επαναλαμβάνουμε στον κώδικά μας και χρησιμοποιούμε έναν μηχανισμό που λέγεται δομή επανάληψης. Ορίζουμε πόσες φορές θέλουμε να εκτελεστεί η εντολή που μας ενδιαφέρει και ο υπολογιστής τις εκτελεί. Επίσης κάποιες φορές θέλουμε να δημιουργούνται μέσα στο **room** τα **objects** μας σε τυχαίες θέσεις κάθε φορά που ξεκινάει το παιχνίδι. Για αυτό το λόγο χρησιμοποιούμε κάποιες συναρτήσεις όπως η `random` για την επιστροφή ενός τυχαίου αριθμού.

? Τι θα φτιάξουμε;

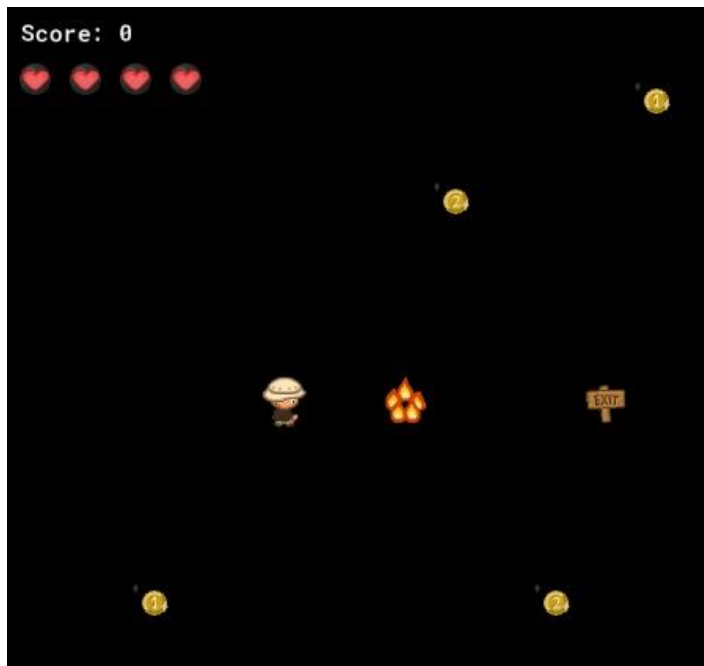
Σήμερα θα επεκτείνουμε το παιχνίδι ως εξής:

- Αρχικά θα υπάρχει μία ακόμα μεταβλητή που θα μετράει τις ζωές που θα έχει ο παίκτης καθώς παίζει, χρησιμοποιώντας τη μεταβλητή "Lives"
- Μετά θα δημιουργήσουμε ένα **object** "Fire". Όταν ο βασικός χαρακτήρας αγγίξει το **object** "Fire" θα επιστρέφει στην αρχική του θέση και ταυτόχρονα θα μειώνονται οι ζωές του.

Εκμάθηση προγραμματισμού μέσω της ανάπτυξης παιχνιδιού

- Έπειτα θα φτιάξουμε το **object** "Wall", το οποίο θα λειτουργεί όπως λέει το όνομά του. Δηλαδή δεν θα μπορεί ο χαρακτήρας να περνάει μέσα από τον τοίχο οπότε θα σταματά.
- Τέλος, θα διαχειριστούμε το **create event** του **room** στο οποίο θα καθορίσουμε πώς θα δημιουργούνται τυχαία τα **objects** μέσα στο χώρο (το **create event** του **room** θα το μάθουμε παρακάτω).

Η λειτουργία του παιχνιδιού φαίνεται στις εικόνες 1,2 και 3.



Εικόνα 1. Ο βασικός χαρακτήρας πλησιάζει το object "Fire" και οι ζωές που εμφανίζονται πάνω αριστερά είναι τέσσερις.



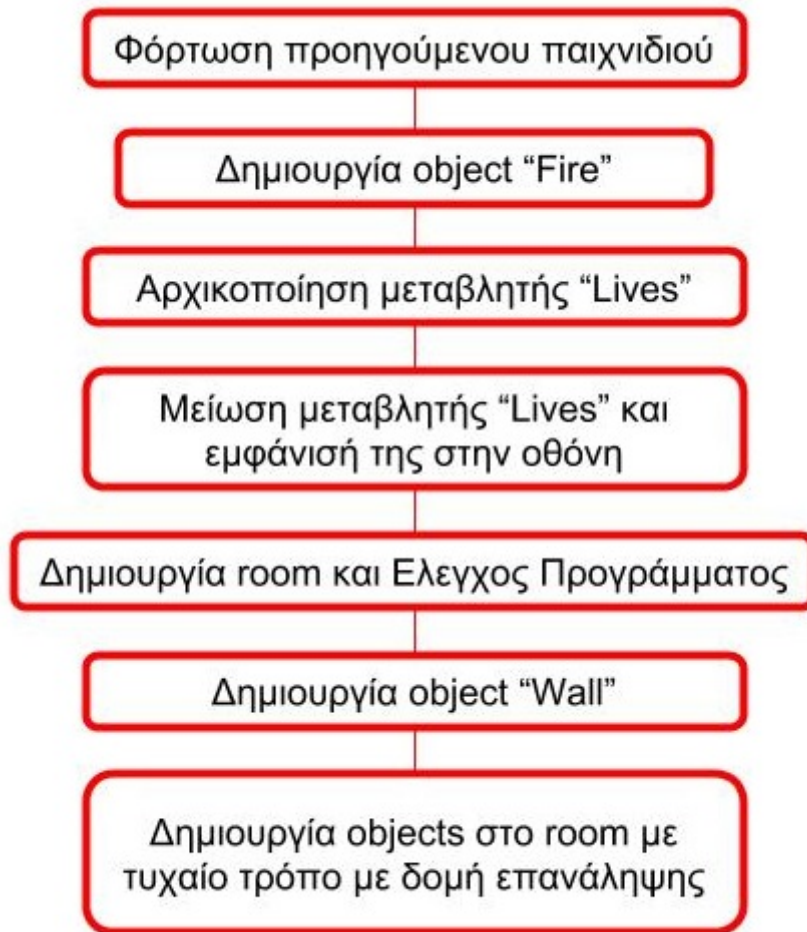


Εικόνα 2. Ο βασικός χαρακτήρας έχει ακουμπήσει το object "Fire" και έχει επιστρέψει στην αρχική του θέση - οι ζωές του είναι μειωμένες κατά μία.



Εικόνα 3. Το αποτέλεσμα της τυχαίας δημιουργίας των objects στο room.

## Διάγραμμα Ροής Δραστηριότητας



Ας ξεκινήσουμε λοιπόν!

Εκτέλεσε το **Game Maker**, στην αρχική οθόνη του προγράμματος, πάτησε **Open**, στο παράθυρο που εμφανίζεται βρες το προηγούμενο πρόγραμμα και κάνε διπλό κλικ για να ανοίξει.

Στην μπάρα των **Resources** (πόρων) δεξιά στην οθόνη, υπάρχουν όλα τα **sprites**, **objects** και **rooms**, που έχεις δημιουργήσει.

### ΒΗΜΑ ΠΡΩΤΟ: Δημιουργία Object "Fire"

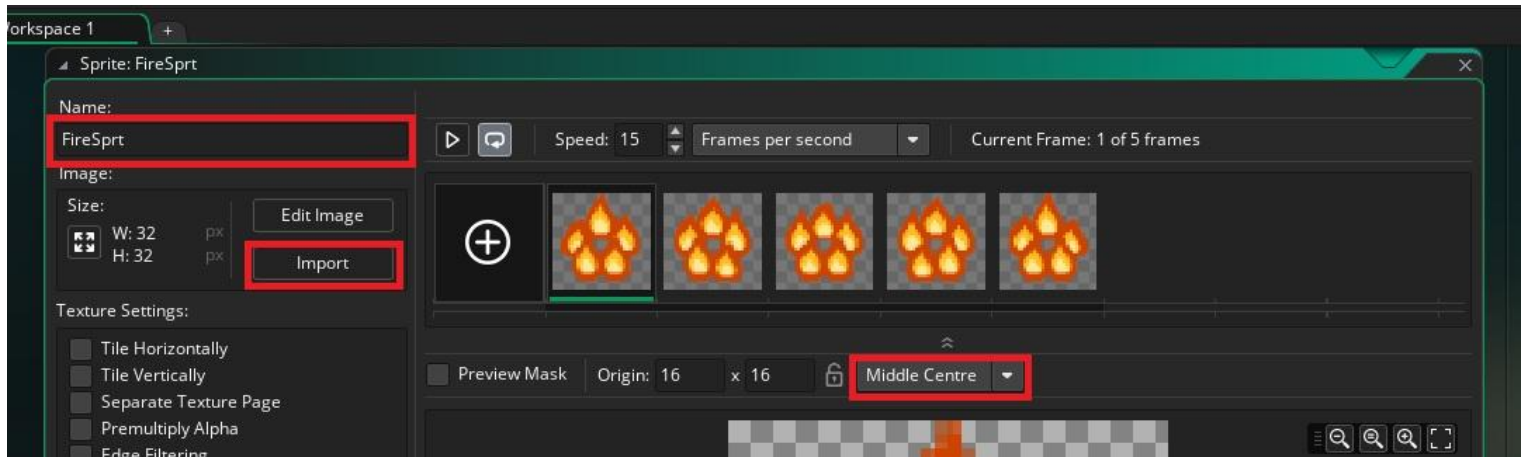


Δημιούργησε το object "Fire"

Για να δημιουργήσεις το **object** "Fire", αρχικά χρειάζεσαι το **sprite** της.

Δημιούργησε ένα καινούργιο **sprite**:

1. Για όνομα γράψε "FireSprt"
2. Για **sprite** διάλεξε το αρχείο "fire\_strip5.png" από τον φάκελο Resources
3. Μη ξεχάσεις να επιλέξεις Middle\_Centre, όπως φαίνεται στην Εικόνα 4:



Εικόνα 4: Δημιουργία sprite για το object "Fire"

Τώρα είσαι έτοιμος να φτιάξεις το **object** "Fire". Δημιούργησε ένα καινούργιο **object** με τα εξής χαρακτηριστικά:

1. Για όνομα διάλεξε "Fire"
2. Για **sprite** το "FireSprt", που μόλις δημιούργησες

Το **object** αυτό δεν χρειάζεται να έχει δικά του **events** και ενέργειες. Αντ' αυτού το **object** "MainCharacter" είναι αυτό που θα κάνει κάποιες ενέργειες όταν αγγίζει το **object** "Fire".

Πιο συγκεκριμένα θέλουμε το "MainCharacter", όταν αγγίζει το **object** "Fire":

- Να επιστρέφει στην αρχική του θέση και
- Να μειώνονται οι ζωές που θα έχει ο παίκτης

## ΒΗΜΑ ΔΕΥΤΕΡΟ: Δημιουργία μεταβλητής "Lives"

? Πώς πιστεύεις ότι θα αναπαριστούμε τις ζωές που θα έχει στη διάθεσή του ο παίκτης και θα μειώνονται κάθε φορά που αγγίζει το object "Fire";

Η απάντηση είναι ΜΕΤΑΒΛΗΤΗ. Θα χρησιμοποιήσουμε μία μεταβλητή η οποία θα αναπαριστά τις ζωές.

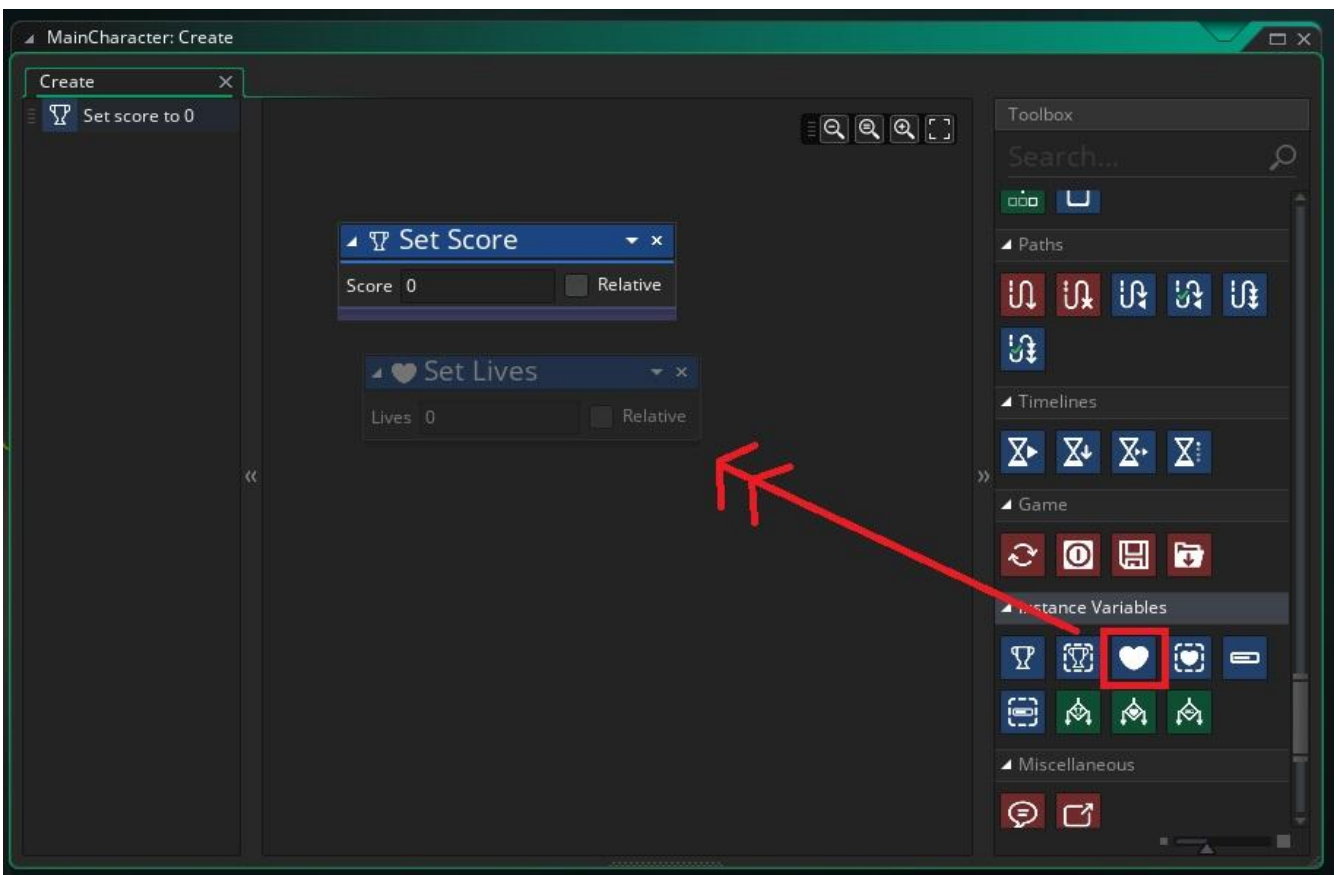
## GOAL

Φτιάξε μια καινούργια μεταβλητή με όνομα "Lives", στο object "MainCharacter", που θα αρχικοποιείται στην τιμή 4 (επειδή τόσες ζωές θέλουμε να έχει το παιχνίδι), θα μειώνεται κάθε φορά που το object "MainCharacter" αγγίζει το object "Fire" και θα εμφανίζεται με την ειδική ενέργεια εμφάνισης ζώων στην οθόνη.

Αρχικά θέλουμε να αρχικοποιήσουμε τη μεταβλητή των ζωών, "Lives". Αυτό θα γίνεται κάθε φορά που ξεκινάει το παιχνίδι, δηλαδή στο **Create Event** του object "MainCharacter".

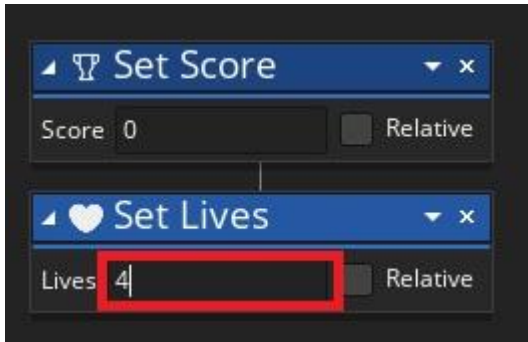
Για να το κάνεις αυτό:

1. Άνοιξε το **object** "MainCharacter"
2. Μπες στο **Create Event**
3. Σύρε την ενέργεια **Set Lives** μέσα στις ενέργειες του **event** αυτού (υπάρχει μόνο μία), όπως φαίνεται στην Εικόνα 5:



Εικόνα 5. Καθορισμός ενέργειας Set Lives στο object "MainCharacter"

4. Άλλαξε την τιμή στην οποία θα αρχικοποιηθεί τη συγκεκριμένη μεταβλητή από 0 σε 4, όπως φαίνεται στην εικόνα 6:

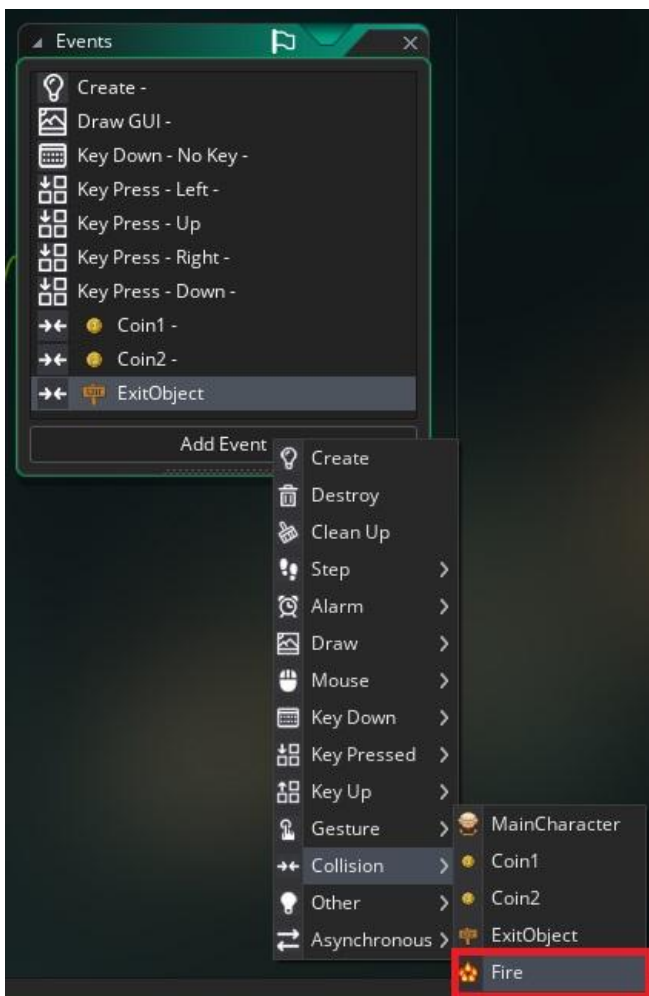


Εικόνα 6. Αρχικοποίηση μεταβλητής "Lives"

Στη συνέχεια επέκτεινε τις ενέργειες στα **events** του **object** "MainCharacter", έτσι ώστε να μειώνεται η μεταβλητή "Lives", καθώς και να ξεκινάει από την αρχική του θέση κάθε φορά που αγγίζει το **object** "Fire".

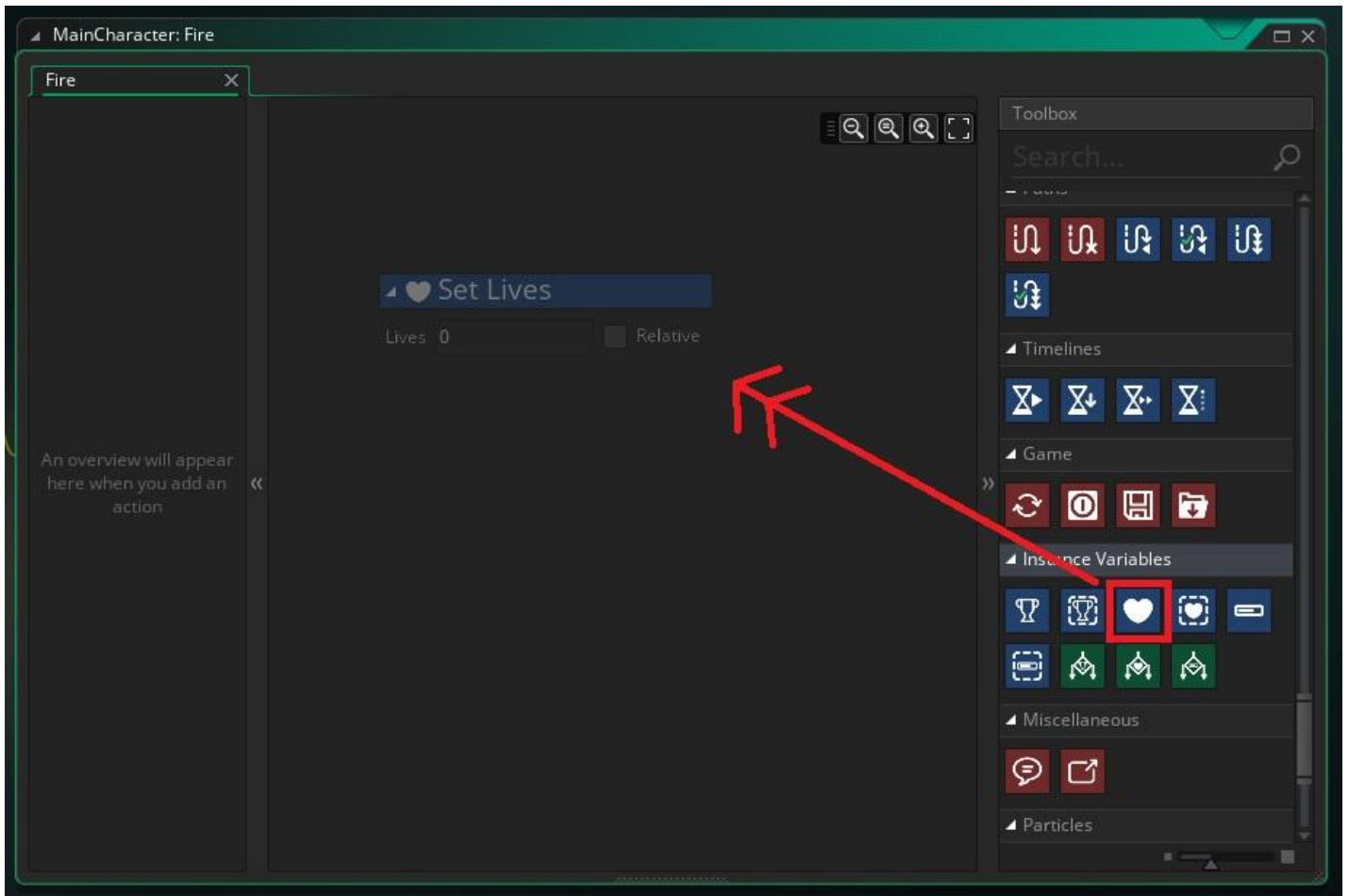
Ακολούθησε τα παρακάτω βήματα:

1. Δημιούργησε ένα καινούργιο **Collision Event** για το **object** "Fire", όπως φαίνεται στην εικόνα 7.



Εικόνα 7. Δημιουργία Collision Event για το object "Fire"

2. Στο παράθυρο που θα ανοίξει εισήγαγε 2 ενέργειες. Αρχικά την **Set Lives**, όπως φαίνεται στην Εικόνα 8.



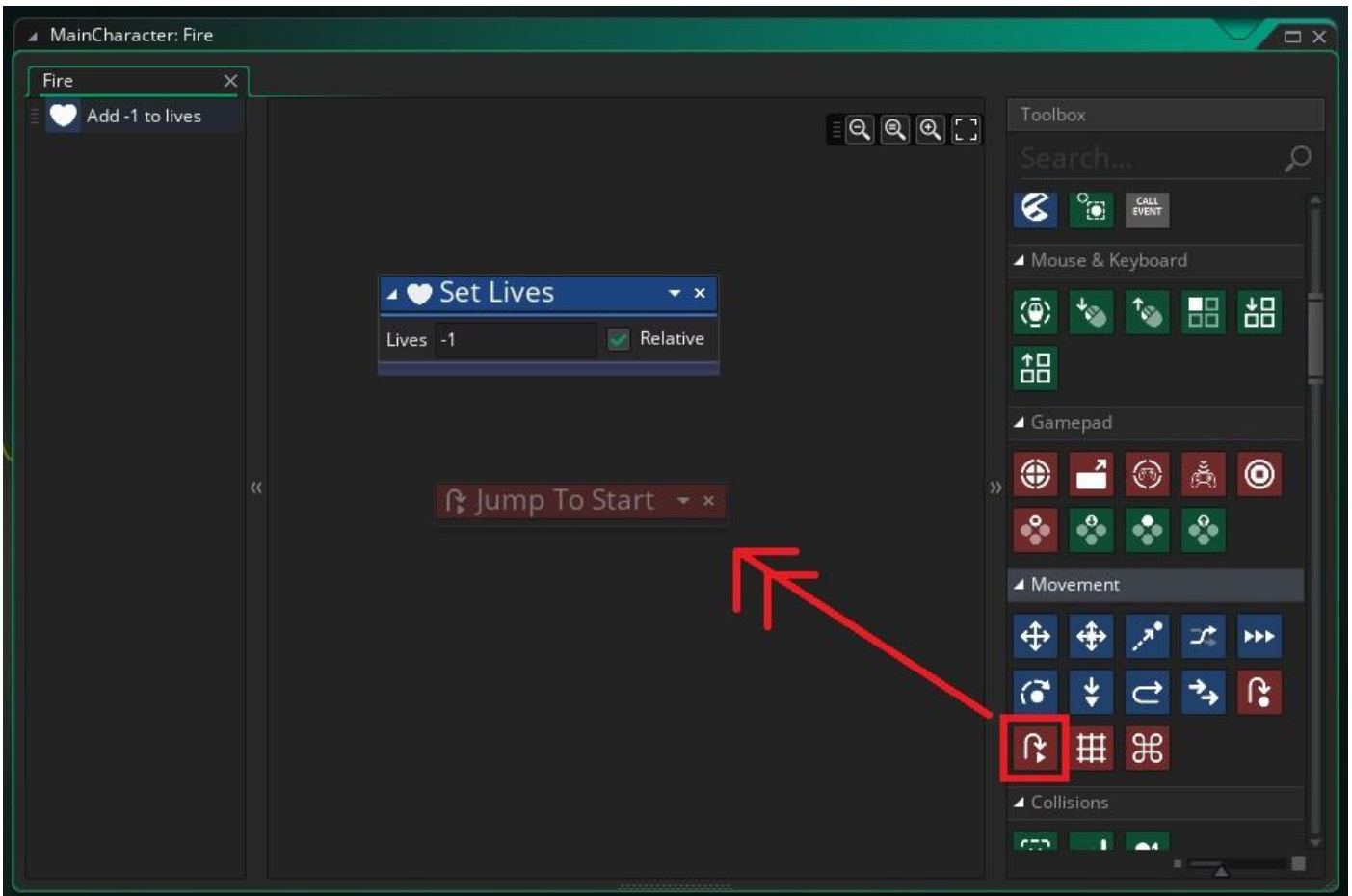
Εικόνα 8. Καθορισμός εντολής Set Lives στο event Collision Event

3. Θέσε την τιμή -1 (ώστε να μειώνονται οι ζωές κατά 1), τσεκάροντας ταυτόχρονα και το **RELATIVE** (ώστε η μείωση να γίνεται σχετικά με την προηγούμενη τιμή), όπως φαίνεται στην Εικόνα 9.



Εικόνα 9. Καθορισμός τιμής αλλαγής

4. Εισήγαγε την ενέργεια **Jump to Start**, όπως φαίνεται στην Εικόνα 10, έτσι ώστε το "MainCharacter" να γυρίσει στην αρχική του θέση.



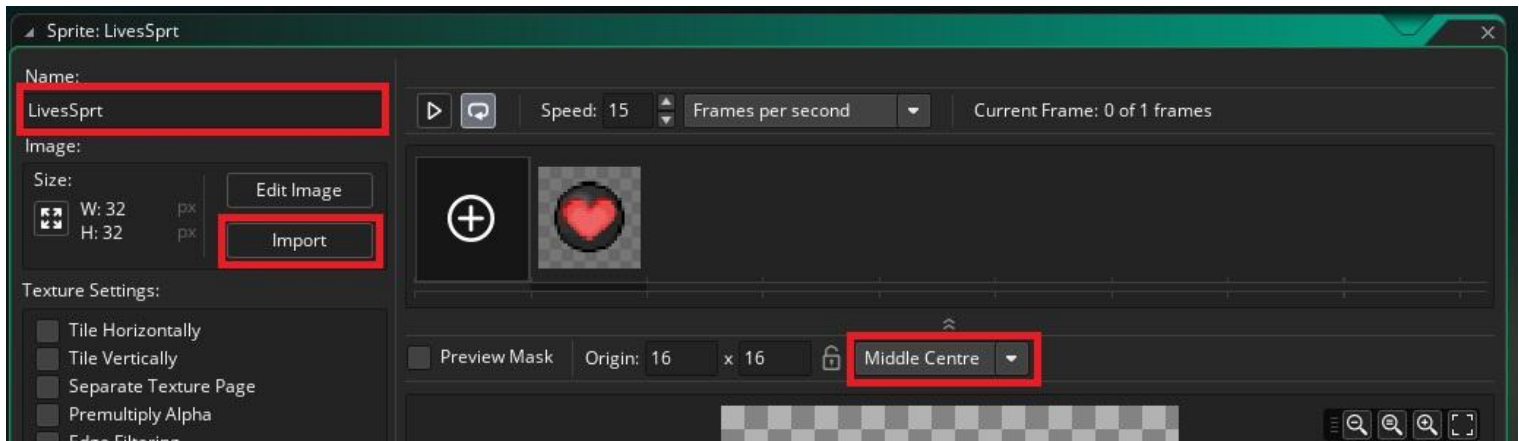
Εικόνα 10. Καθορισμός εντολής Jump To Start στο Collision Event

5. Κλείσε το παράθυρο επεξεργασίας του **object** "MainCharacter".

Τώρα αν ο "MainCharacter" ακουμπήσει το **object** "Fire" θα γυρίσει στην αρχική θέση και θα μειωθούν οι ζωές κατά μία.

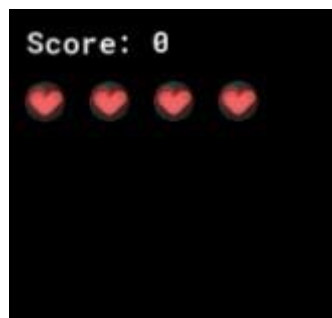
Τέλος, θέλουμε να εμφανίζονται πάνω αριστερά οι ζωές που έχει κάθε στιγμή ο παίκτης. Για να το κάνουμε αυτό θα χρειαστεί να δημιουργήσουμε ένα **sprite**, το οποίο θα χρησιμοποιηθεί ως σύμβολο για τη ζωή.

Φτιάξε λοιπόν ένα καινούργιο **sprite** με όνομα "LivesSprt". Για εικόνα διάλεξε το αρχείο "item\_life.png" και μην ξεχασεις να επιλέξεις Middle\_Centre, όπως φαίνεται στην Εικόνα 11:



Εικόνα 11. Δημιουργία sprite "LivesSprt" για τις ζωές

Για να εμφανίζονται οι ζωές όπως γίνεται συνήθως στα παιχνίδια (Εικόνα 12), θα πρέπει να βάλουμε ειδική ενέργεια στο **Draw GUI Event** του **object** "MainCharacter".

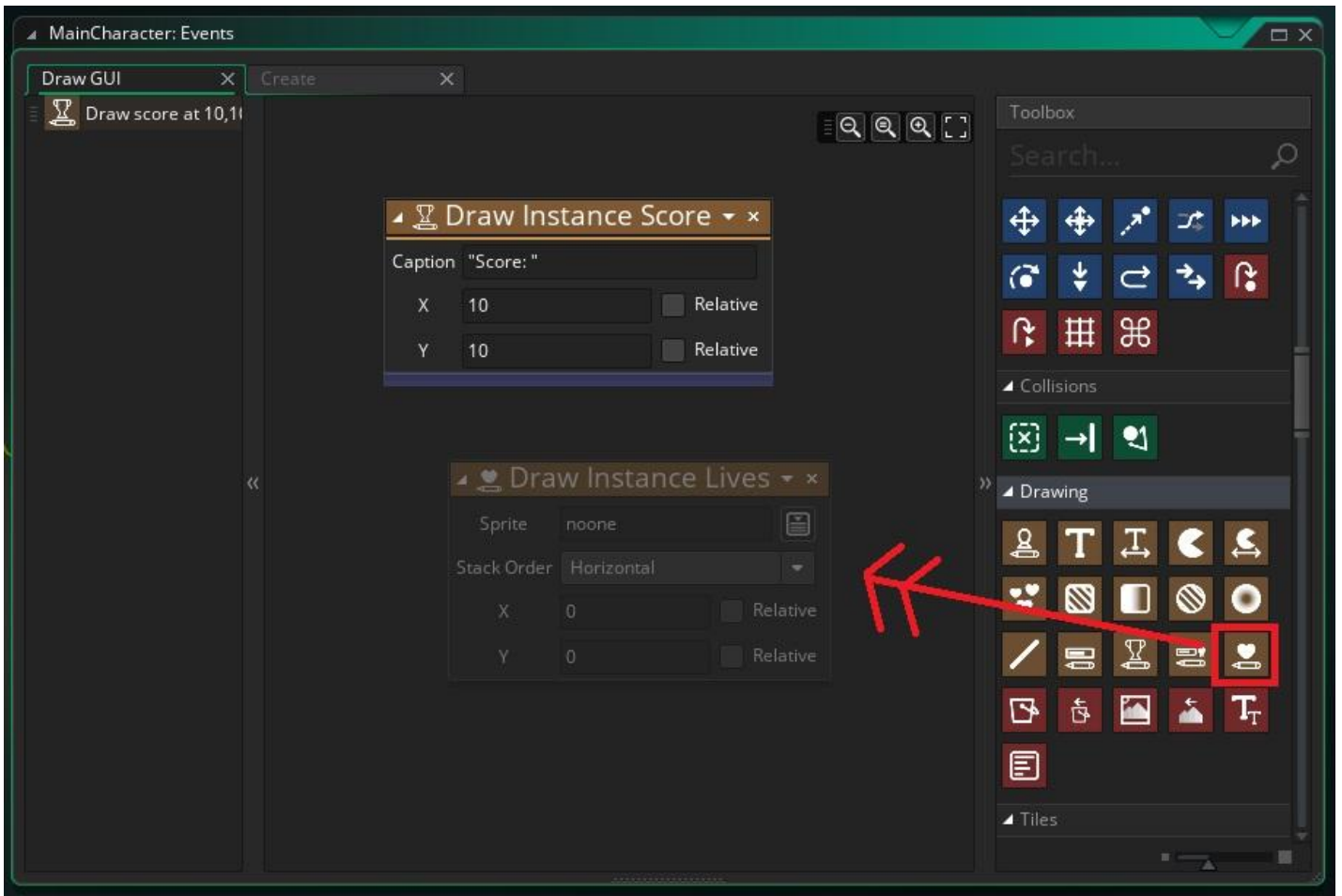


Εικόνα 12. Αρχικές ζωές: 4

Για να το κάνεις αυτό ακολούθησε τα παρακάτω βήματα:

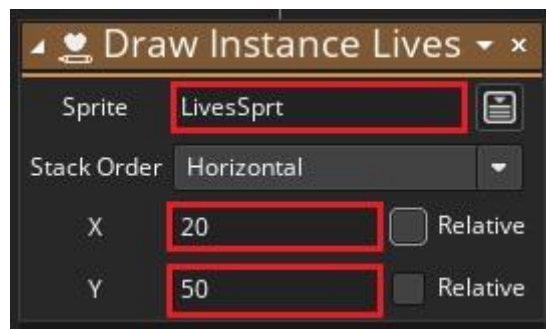
1. Άνοιξε το παράθυρο επεξεργασίας του **object** "MainCharacter".
2. Άνοιξε το ήδη υπάρχον **Draw GUI Event**
3. Σύρε μέσα στο παράθυρο την ενέργεια **Draw Instance Lives**, όπως φαίνεται στην Εικόνα 13:





Εικόνα 13. Καθορισμός προβολής ζωών

4. Διάλεξε το **sprite** "LivesSprt", που μόλις δημιούργησες και γράψε στις συντεταγμένες X και Y τις τιμές 20 και 50 (Εικόνα 14) , έτσι ώστε οι καρδιές που θα εμφανιστούν να βρίσκονται πάνω αριστερά ακριβώς κάτω από το σκορ. Περισσότερα για τις συντεταγμένες μπορείς να δεις στην 2<sup>η</sup> Δραστηριότητα.



Εικόνα 14. Καθορισμός συντεταγμένων

5. Τέλος, κλείσε το παράθυρο επεξεργασίας του **object** "MainCharacter".

## ΒΗΜΑ ΤΡΙΤΟ: Έλεγχος-Δοκιμή

? Πώς πιστεύεις ότι ελέγχουμε τη σωστή λειτουργία των προγραμμάτων που φτιάχνουμε;

Η απάντηση φυσικά είναι δοκιμάζοντας. Ο καλύτερος τρόπος για να βρούμε πιθανά λάθη στο κώδικά μας είναι σίγουρα να τρέξουμε το πρόγραμμά μας και να δούμε αν τα **objects** που έχουμε φτιάξει λειτουργούν όπως θα θέλαμε.

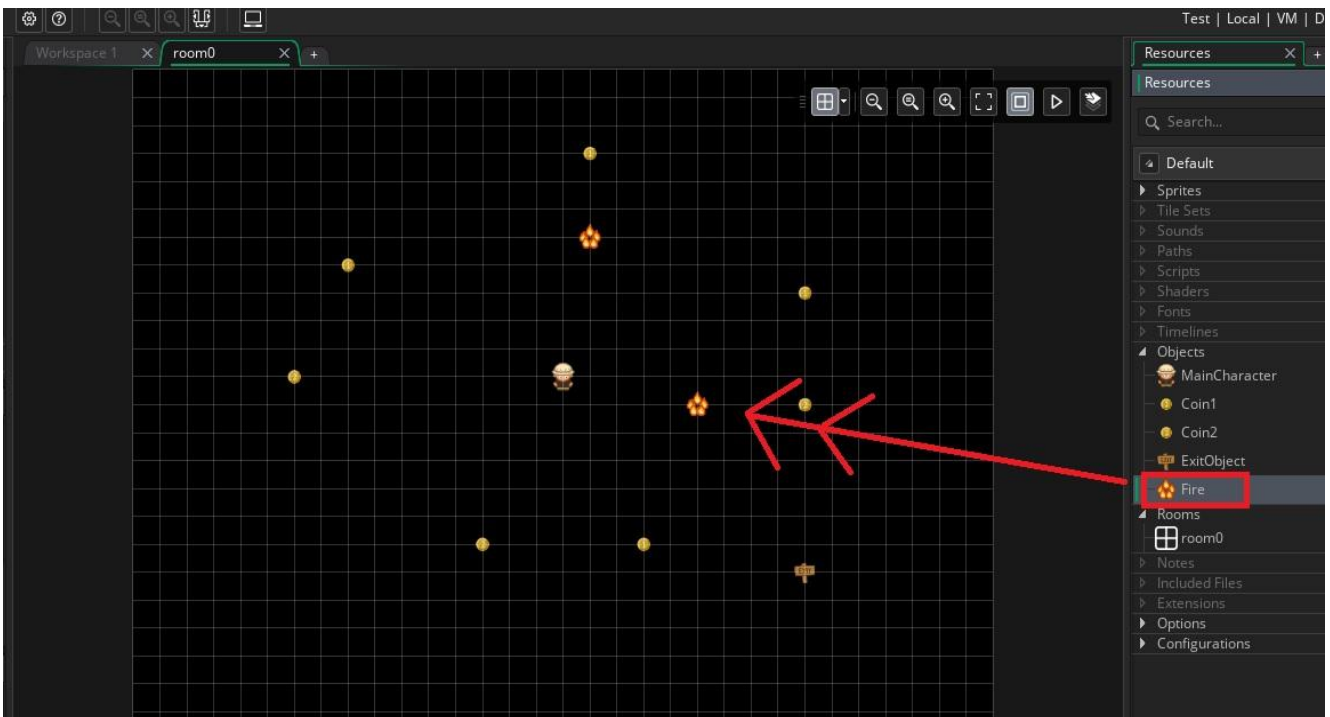
Αν συμβαίνει αυτό προχωράμε παρακάτω. Αν όμως όχι, γυρίζουμε πίσω και προσπαθούμε να βρούμε τι έχουμε κάνει λάθος και το διορθώνουμε.

Αυτό ακριβώς θα κάνουμε και σε αυτό το βήμα.

# GOAL

Τοποθέτησε μέσα στο Room0, objects "Fire", τρέξε το πρόγραμμα και δες αν δουλεύει σωστά.

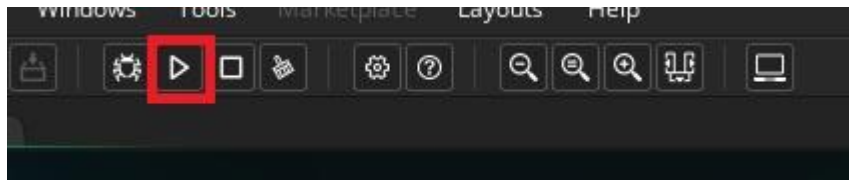
Άνοιξε το παράθυρο επεξεργασίας του **room** και πρόσθεσε φωτιές (Εικόνα 15).



Εικόνα 15. Παράθυρο επεξεργασίας room

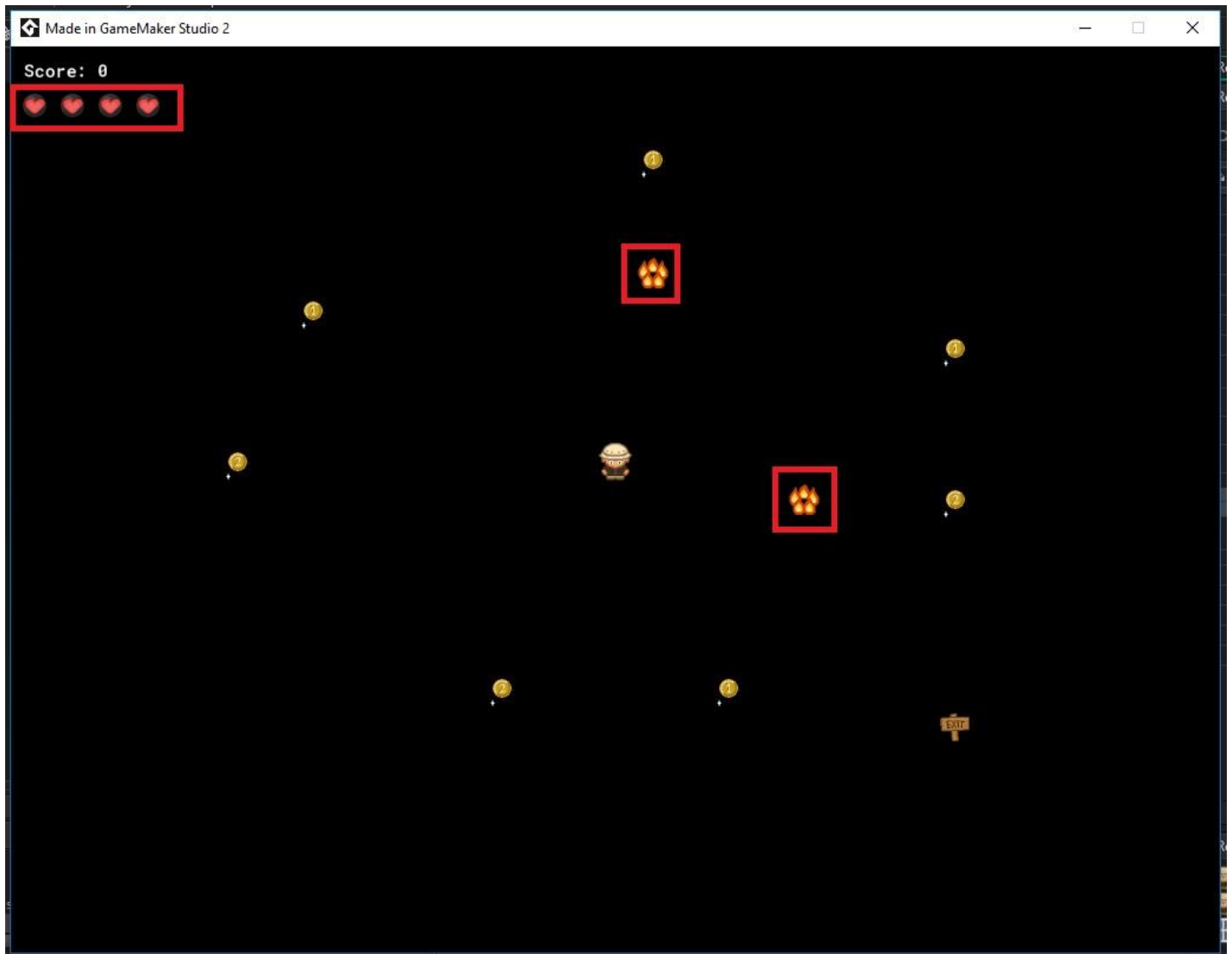
Αφού κλείσεις το παράθυρο επεξεργασίας του **room**, πάτησε το κουμπί Play, για να ξεκινήσει το παιχνίδι (Εικόνα 16)

Εκμάθηση προγραμματισμού μέσω της ανάπτυξης παιχνιδιού



Εικόνα 16. Έναρξη παιχνιδιού

Η αρχική οθόνη του παιχνιδιού πρέπει να μοιάζει όπως στην Εικόνα 17.



Εικόνα 17: Παράθυρο εκτέλεσης παιχνιδιού

Θα πρέπει τώρα να εξετάσουμε τέσσερα πράγματα:

1. Οι καρδιές που αναπαριστούν τις ζωές, εμφανίζονται σωστά πάνω αριστερά;
2. Στην αρχή του παιχνιδιού οι ζωές είναι 4;
3. Αν αγγίξει ο "MainCharacter" το **object** "Fire", ξαναγυρνά στην αρχική του θέση;

4. Αφού γυρίσει στην αρχική του θέση, οι ζωές μειώνονται κατά μία;

Αν η απάντηση στα παραπάνω ερωτήματα είναι ναι τότε μπορείς να προχωρήσεις στα επόμενα βήματα, αλλιώς πρέπει να γυρίσεις πίσω και να κάνεις τις απαραίτητες διορθώσεις.

## ΒΗΜΑ ΤΕΤΑΡΤΟ: Δημιουργία Τοίχου

? Πώς θα φτιάχναμε ένα Object το οποίο να λειτουργεί σαν τοίχος;  
Δηλαδή να μη μπορεί ο παίκτης να περάσει από μέσα του;



Ένα object που έχει την ιδιότητα **solid**, θεωρείται στερεό, δηλαδή δεν μπορούν τα άλλα objects να περάσουν από μέσα του.

Το ότι ένα object είναι **solid**, δεν σημαίνει ότι στα **Collision Event** των άλλων object δεν πρέπει να υπάρχει ενέργεια ακινητοποίησης, όταν συγκρούονται με αυτό.

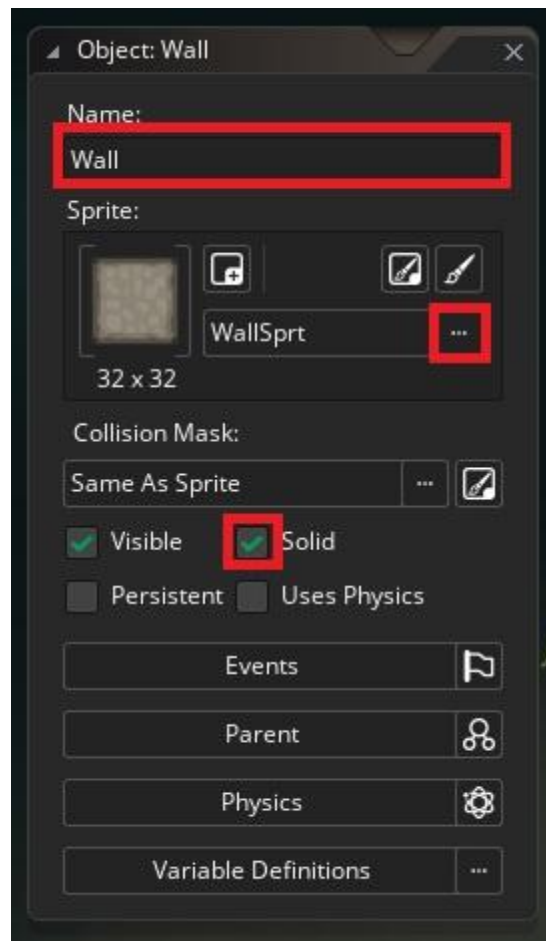
## GOAL

Φτιάξε ένα object με όνομα "Wall", το οποίο να έχει την ιδιότητα **solid** και δημιούργησε **Collision Event** από το object "MainCharacter" προς το object "Wall" με ενέργεια ακινητοποίησης

Για να το καταφέρεις αυτό ακολούθησε τα παρακάτω βήματα:

1. Αρχικά φτιάξε ένα **sprite** με όνομα "WallSprt" επιλέγοντας το αρχείο "wall\_block.png", από το φάκελο Resources
2. Έπειτα δημιούργησε ένα **object** με όνομα "Wall" και **sprite**, αυτό που μόλις έφτιαξες

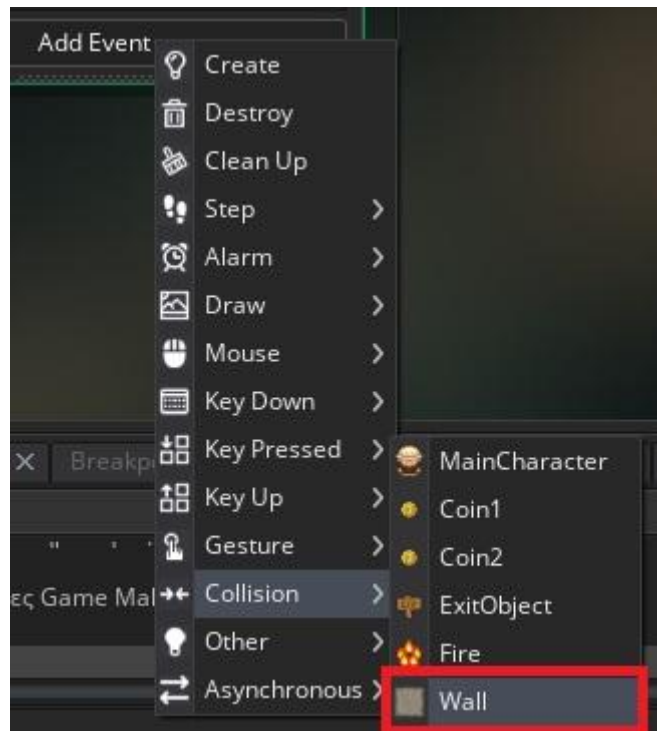
3. Μέσα στο παράθυρο επεξεργασίας του **object** τσέκαρε την επιλογή **solid** (Εικόνα 18)
4. Τέλος, κλείσε το παράθυρο



Εικόνα 18. Καθορισμός παραμέτρων object "Wall"

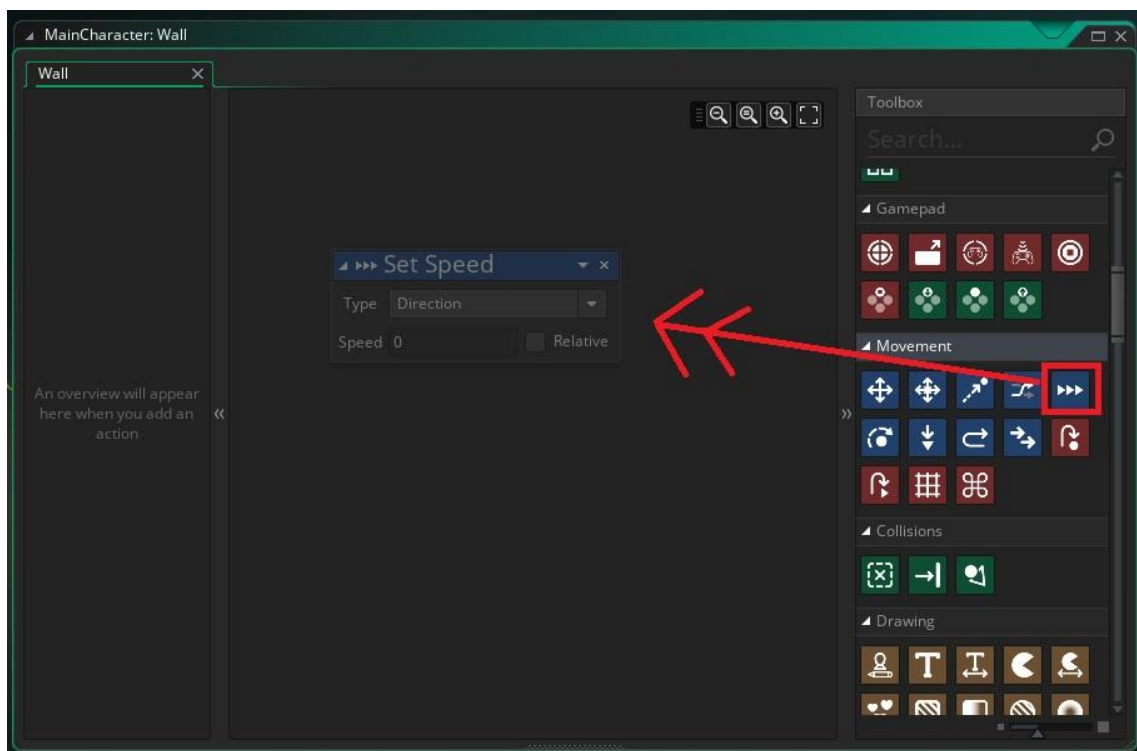
Τώρα πρέπει να καθορίσουμε στο **object** "MainCharacter" ένα **Collision Event** με το "Wall", στο οποίο θα υπάρχει ενέργεια ακινητοποίησης. Ακολούθησε τα παρακάτω βήματα (Εικόνα 19):

1. Άνοιξε το **object** "MainCharacter"
2. Δημιούργησε ένα **Collision Event** με το **object** "Wall"



Εικόνα 19. Δημιουργία Collision Event για το object "Wall"

3. Τώρα σύρε την ενέργεια **Set Speed** μέσα στο παράθυρο, όπως φαίνεται στην Εικόνα 20:



Εικόνα 20. Καθορισμός ενέργειας Set Speed

Ορίζοντας ταχύτητα 0, ακινητοποιούμε το **object**.

4. Χωρίς να αλλάξεις τίποτα στις επιλογές της ενέργειας κλείσε το παράθυρο.

## ΒΗΜΑ ΠΕΜΠΤΟ: Τυχαία Δημιουργία Objects στο Room

Στο βήμα αυτό θα μάθουμε πως να αυτοματοποιούμε τη δημιουργία **objects** στο **room** μέσω ενεργειών. Οι παρακάτω στόχοι αποσκοπούν στη δημιουργία:

- Ενός **object** "MainCharacter" στο αριστερό μέρος της οθόνης
- 5 **objects** "Fire" σε τυχαίες θέσεις
- 4 **objects** "Coin1" σε τυχαίες θέσεις
- 3 **objects** "Coin2" σε τυχαίες θέσεις
- 10 **objects** "Wall" σε τυχαίες θέσεις
- του **object** "ExitObject" στο δεξί μέρος της οθόνης

? Τι πιστεύεις ότι πρέπει να κάνουμε για να γίνονται ενέργειες κατά τη δημιουργία ενός room, δηλαδή στην αρχή του παιχνιδιού;

Η απάντηση είναι στο **Creation Code** του **room**. Μέχρι τώρα έχουμε δει **events** στα **objects**.



Το Creation Code ενός room είναι ένα Event όπως αυτά των object που ενεργοποιείται ακριβώς μόλις ξεκινάει να λειτουργεί το room. Ανήκει στο room και όχι σε object και μέσα σε αυτό το Event βάζουμε ενέργειες που θέλουμε να γίνονται κατά τη φάση της εκκίνησης του παιχνιδιού.

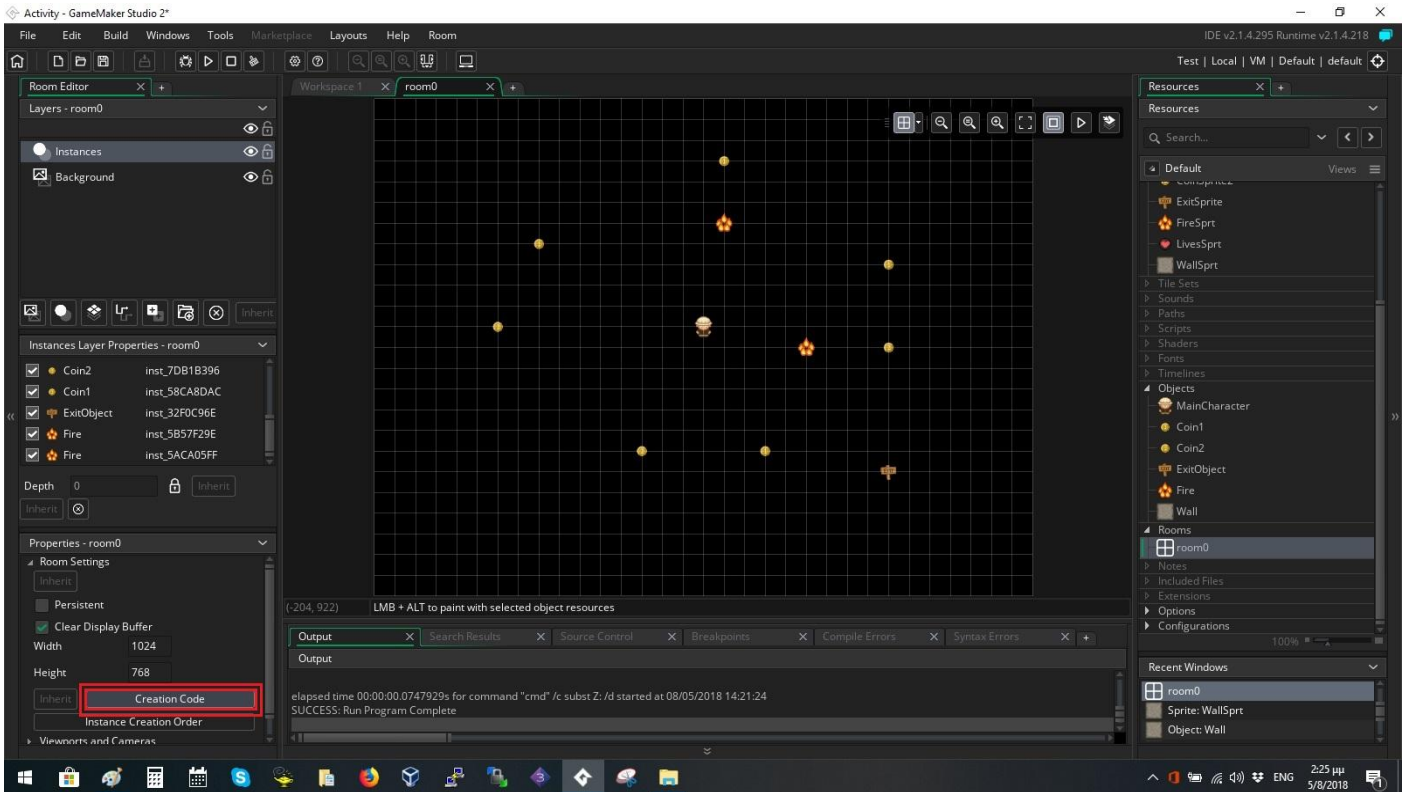
## GOAL

Άνοιξε το Creation Code του Room0 και τοποθέτησε μία ενέργεια δημιουργίας του object "MainCharacter".

Ακολούθησε τα παρακάτω βήματα:

1. Άνοιξε το **Room0**

2. Κάνε κλικ στο κουμπι **Creation Code** που βρίσκεται κάτω αριστερά (Εικόνα 21), για να ανοίξει το παράθυρο ενεργειών του Creation Code:



Εικόνα 21. Ενεργοποίηση Creation Code του Room0

Θέλουμε μία ενέργεια που να δημιουργεί **objects** μέσα στο **room** σε συγκεκριμένη θέση.



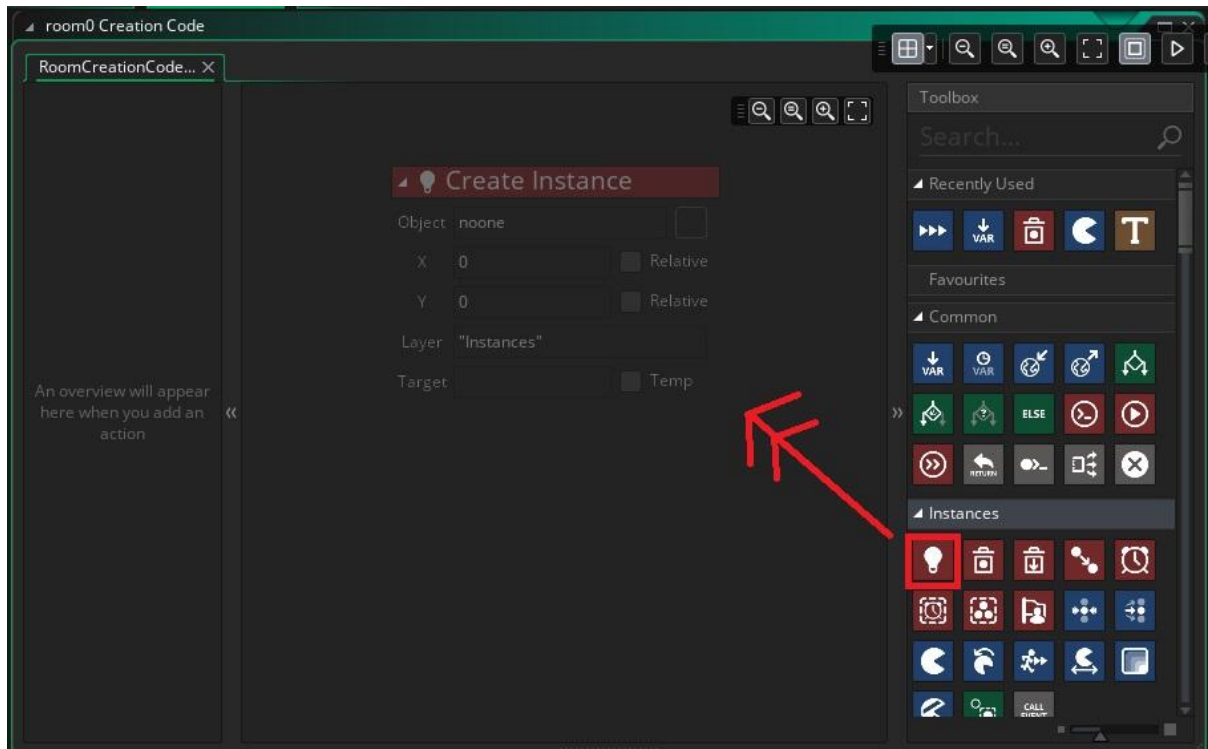
Η ενέργεια Create Instance που είναι διαθέσιμη σε όλα τα Events, είναι μία ενέργεια που όταν εκτελεστεί θα δημιουργήσει ένα object σε συγκεκριμένη θέση του room.

Οι παράμετροι που πρέπει να ορίσουμε είναι:

- Το object που θέλουμε να δημιουργηθεί
- Τις συντεταγμένες X και Y του καινούργιου object

3. Στο παράθυρο που βρισκόμαστε, σύρε την ενέργεια Create Instance όπως στην Εικόνα 22:





Εικόνα 22. Δημιουργία στιγμιότυπων object στο room (Create Instance)

? Πώς θα καθορίσουμε να δημιουργηθεί το object μας στη μέση του αριστερού τμήματος της οθόνης;

Για να αναφερθούμε στη μέση της οθόνης πρέπει να γνωρίζουμε το μέγεθος της οθόνης και να το διαιρέσουμε με το 2.

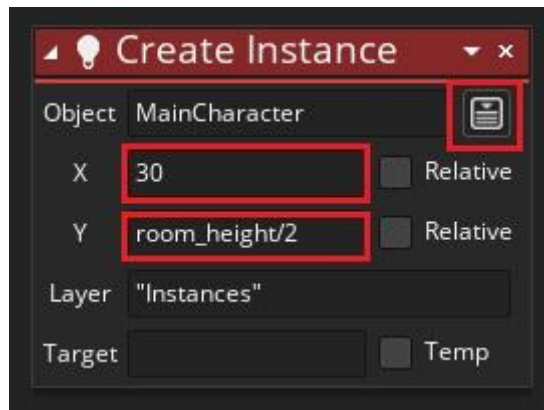


Το `room_width` και το `room_height` είναι δύο μεταβλητές του Game Maker τις οποίες εμείς ΔΕΝ μπορούμε να αλλάξουμε και έχουν την πληροφορία για το μέγεθος της οθόνης σε PIXEL.

Το `room_width` μας επιστρέφει ως τιμή το πλάτος της οθόνης και το `room_height` το ύψος της οθόνης.

Με αυτές τις δύο μεταβλητές μπορούμε να αναφερθούμε σε σχετικές θέσεις της οθόνης όπως αρχή, μέση, τέλος κάνοντας τις απαραίτητες πράξεις.

4. Για να τοποθετηθεί το **object** "MainCharacter" στη μέση της οθόνης πρέπει να οριστεί η συντεταγμένη Y σε  $room\_height/2$  και η συντεταγμένη X σε μία επιθυμητή τιμή κοντά στο αριστερό όριο (π.χ. 30 όπως φαίνεται στην Εικόνα 23).



Εικόνα 23. Καθορισμός θέσης object "MainCharacter"

? Πώς μπορώ να δημιουργήσω πολλά object χωρίς να βάλω την ενέργεια Create Instance πολλές φορές;

Με δομή ΕΠΑΝΑΛΗΨΗΣ. Τι είναι, όμως η δομή επανάληψης;



Η δομή επανάληψης μας επιτρέπει να κάνουμε μία ενέργεια πολλές φορές γράφοντας την ενέργεια αυτή μόνο μία φορά. Το Game Maker παρέχει πολλές διαφορετικές δομές επανάληψης.



Repeat.

Η εντολή Repeat επιτρέπει την επανάληψη ενός συνόλου εντολών συγκεκριμένο πλήθος φορές.



For.

Η εντολή For επιτρέπει την επανάληψη ενός συνόλου εντολών συγκεκριμένο πλήθος φορές, με τη διαφορά ότι έχουμε στη διάθεσή μας μία συγκεκριμένη μεταβλητή (συνήθως λέγεται i), η οποία δείχνει σε ποια επανάληψη βρισκόμαστε.



While.

Η εντολή While μας επιτρέπει να εκτελούμε ένα σύνολο εντολών όσο ισχύει

μία συγκεκριμένη συνθήκη για κάποια μεταβλητή, δηλαδή άγνωστο πλήθος φορών.



Loop με Break.

Η εντολή Loop εκτελεί ένα σύνολο εντολών για πάντα μέχρι να συναντήσει μέσα σε αυτό το σύνολο την εντολή Break.

Κάθε δομή επανάληψης έχει συγκεκριμένες δυνατότητες και για αυτό επιλέγουμε ποια θα χρησιμοποιήσουμε με βάση τις ανάγκες μας.

Στη συγκεκριμένη περίπτωση χρειαζόμαστε μία δομή επανάληψης που θα εκτελεί συγκεκριμένο πλήθος φορών, 5, την ενέργεια **Create Instance**.

? Ποια δομή επανάληψης πιστεύεις ότι χρειαζόμαστε σε αυτήν την περίπτωση;

## GOAL

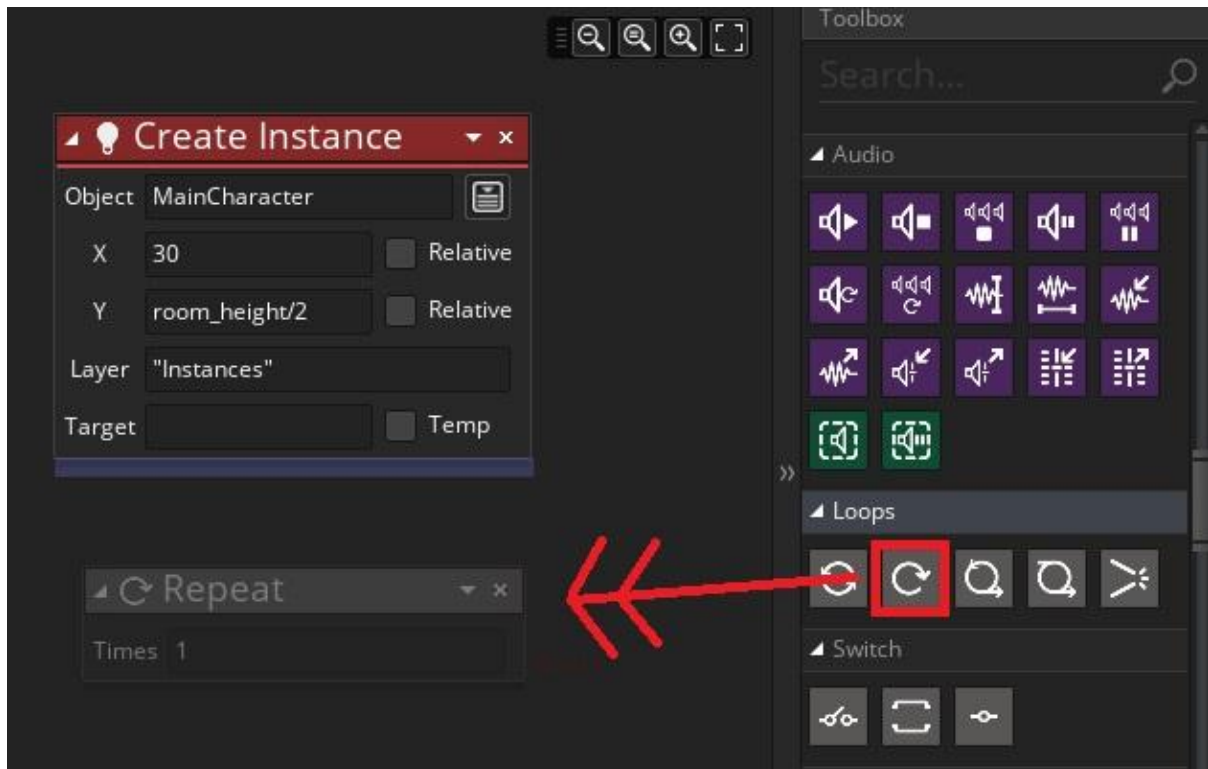
Τοποθέτησε μία εντολή Repeat με ενέργεια τη δημιουργία του object "Fire" σε τυχαίες θέσεις μέσα στην οθόνη.

Να θυμηθούμε ότι χρειάζεται να δημιουργήσουμε

- 5 **objects** "Fire" σε τυχαίες θέσεις
- 4 **objects** "Coin1" σε τυχαίες θέσεις
- 3 **objects** "Coin2" σε τυχαίες θέσεις
- 10 **objects** "Wall" σε τυχαίες θέσεις
- το **object** "ExitObject" στο δεξί μέρος της οθόνης

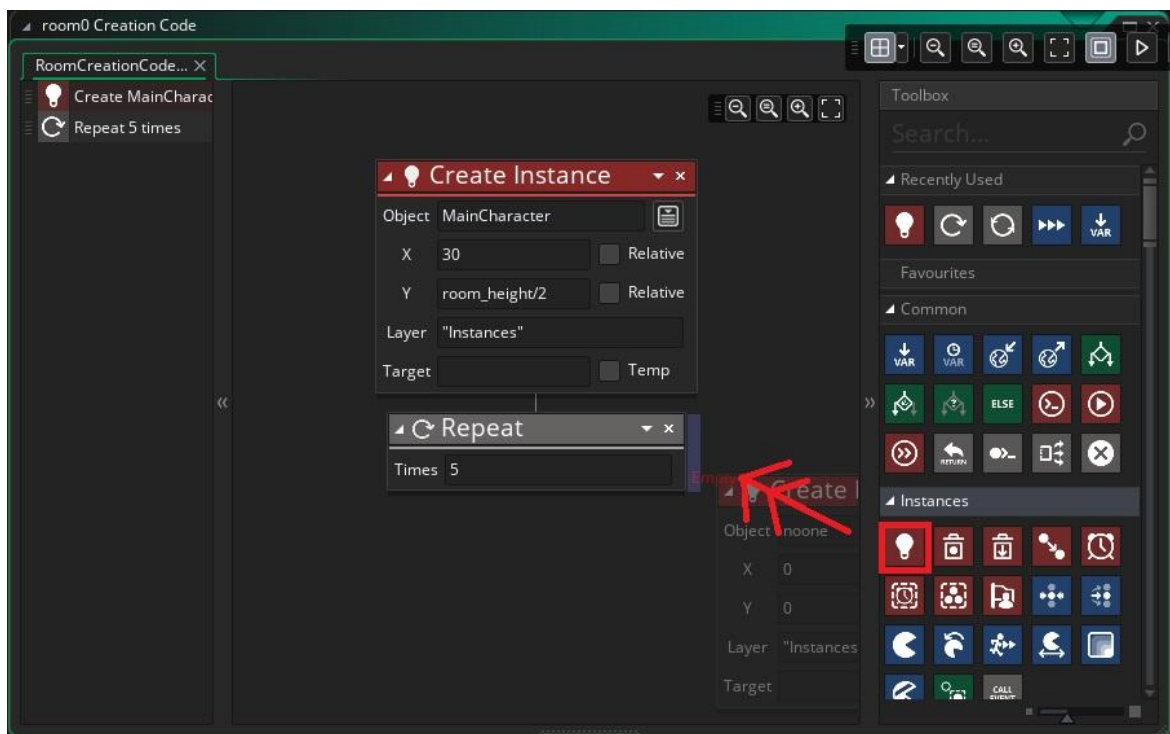
Ακολουθήσε τα παρακάτω βήματα:

1. Σύρε την ενέργεια (εντολή) **Repeat** μέσα στο παράθυρο με τις ενέργειες, όπως φαίνεται Εικόνα 24.



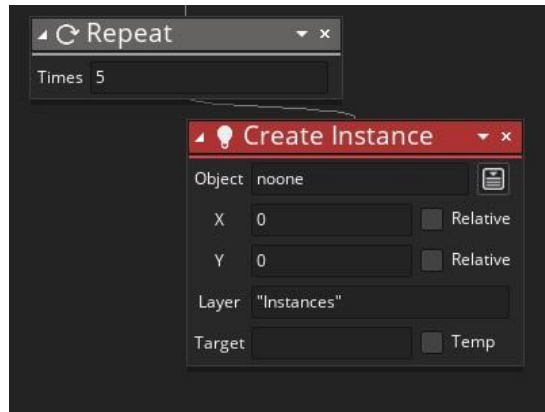
Εικόνα 24. Δημιουργία επανάληψης στο παράθυρο

2. Άλλαξε την τιμή του πεδίου Times σε 5 (για τη δημιουργία 5 **objects** "Fire")
3. Εισήγαγε την ενέργεια **Create Instance** δεξιά από το **Repeat** (εμφώλευση εντολής) και όχι από κάτω (διαδοχική ακολουθία) (εικόνα 25)



Εικόνα 25. Εισαγωγή ενέργειας Create Instance μέσα στο σύνολο εντολών της Repeat.

Το αποτέλεσμα αυτού του βήματος πρέπει να είναι σύμφωνο την εικόνα 26.



Εικόνα 26. Δημιουργία objects με την εντολή Repeat

? Πώς πιστεύεις ότι μπορούμε να δημιουργήσουμε έναν τυχαίο αριθμό από το 0 έως έναν αριθμό της επιλογής μας;

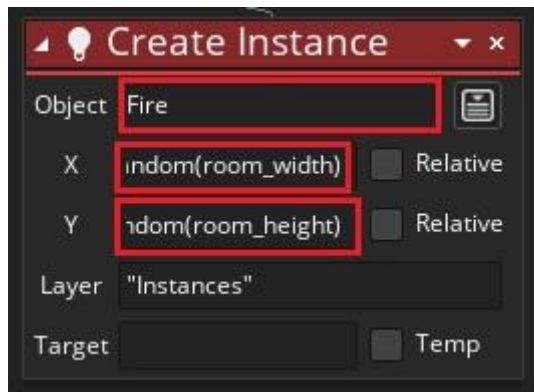


Η συνάρτηση `random` παρέχει τη δυνατότητα δημιουργίας τυχαίων αριθμών. Αντί δηλαδή να γράψουμε εμείς κάποιον αριθμό ή μεταβλητή σε κάποιο πεδίο, μπορούμε να δημιουργήσουμε έναν αριθμό τυχαία.

Η συνάρτηση `random` δέχεται ως είσοδο έναν αριθμό  $X$  και μας επιστρέφει έναν τυχαίο αριθμό από 0 έως  $X$ .

Αυτό που θέλουμε εμείς είναι αντί για συγκεκριμένες συντεταγμένες στα πεδία  $X$  και  $Y$  της ενέργειας `Create Instance` να δώσουμε `random(room_width)` και `random(room_height)` αντίστοιχα, έτσι ώστε τα **objects** να εμφανιστούν σε τυχαίες θέσεις μέσα στην οθόνη.

4. Για την ενέργεια **Create Instance** επέλεξε το **object** "Fire", και γράψε στις συντεταγμένες  $X$  και  $Y$ , `random(room_width)` και `random(room_height)`, όπως φαίνεται στην Εικόνα 27:



Εικόνα 27. Δημιουργία στιγμιότυπου object σε τυχαία θέση

## GOAL

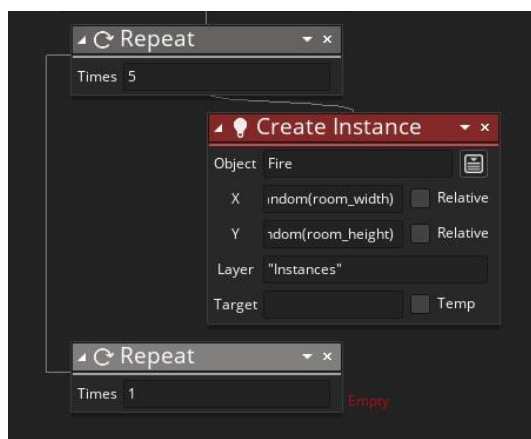
Συνέχισε τον κώδικα δημιουργώντας:

- 4 objects "Coin1"
- 3 objects "Coin2" και
- 10 objects "Wall"

σε τυχαίες θέσεις

**ΠΡΟΣΟΧΗ!**

Η εντολές Repeat πρέπει να είναι στοιχισμένες, η μία κάτω από την άλλη, για να δηλώνεται η ακολουθία των ενεργειών (Εικόνα 28).



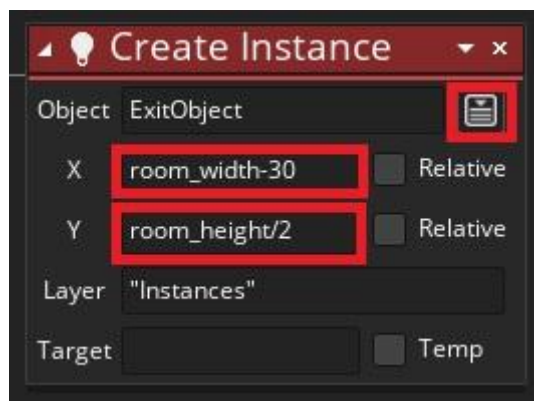
Εικόνα 28. Ακολουθία εντολών Repeat



Ολοκλήρωσε την επιθυμητή συμπεριφορά για το παιχνίδι, εισάγοντας το object "ExitObject" στα δεξιά στη μέση της οθόνης.

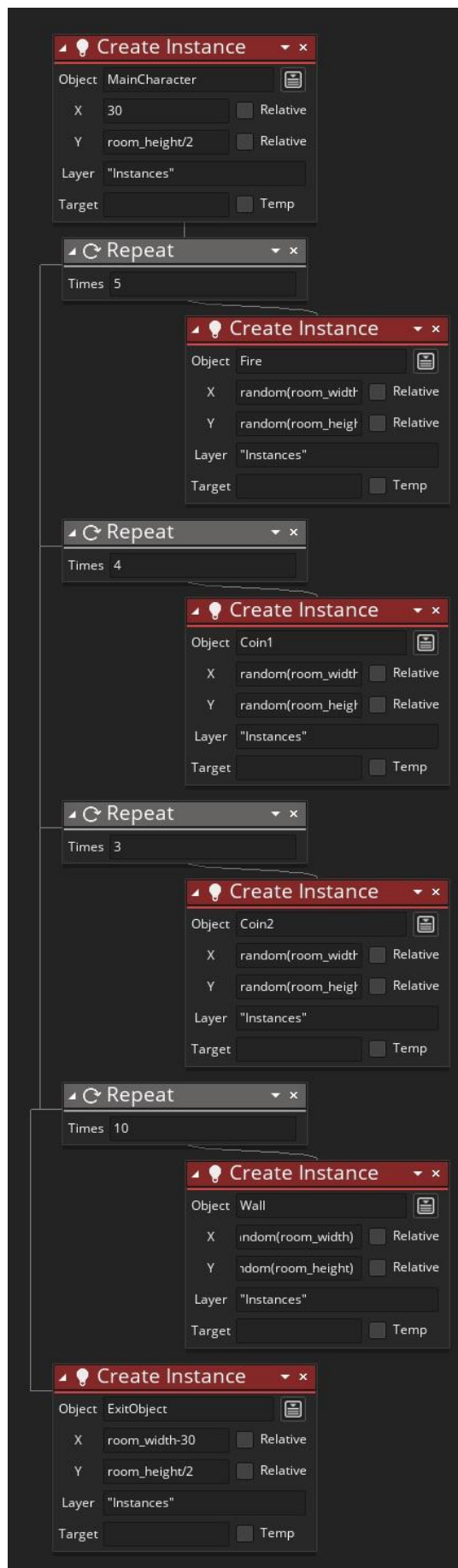
Ακολούθησε τα παρακάτω βήματα:

1. Σύρε την ενέργεια **Create Instance** στο τέλος του κώδικα ΚΑΤΩ από την εντολή Repeat
2. Επίλεξε το **object** "ExitObject"
3. Όρισε συντεταγμένες  $room\_width-30$  για X, έτσι ώστε να εμφανιστεί 30 pixel αριστερά από το δεξί όριο της οθόνης και  $room\_height/2$  για Y (Εικόνα 29).



Εικόνα 29. Δημιουργία object "ExitObject"

Το συνολικό αποτέλεσμα των παραπάνω στόχων πρέπει να σύμφωνο με την παρακάτω Εικόνα 30.



Εικόνα 30. Δημιουργία στιγμιότυπων objects σε τυχαίες θέσεις



Τώρα κλείσε το παράθυρο επεξεργασίας του **Creation Code** και γύρισε στο **room** όπως το είχες δημιουργήσει.

Πριν δοκιμάσουμε τελικά το παιχνίδι πρέπει να διαγραφούν όλα τα **objects** του **room** όπως υπήρχαν πριν τη χρήση της εντολής **Repeat** διότι διαφορετικά θα υπάρχουν 2 φορές.

Κλείσε το παράθυρο επεξεργασίας του **room** και δοκίμασε το παιχνίδι που έφτιαξες!

## Καλή Διασκέδαση!



- Σε τι χρησιμεύει η συνάρτηση random; Τι παίρνει σαν είσοδο και τι μας δίνει στην έξοδο;
- Γιατί μας είναι χρήσιμη η δομή επανάληψης; Ανέφερε μία εντολή επανάληψης που μας δίνει το Game Maker.
- Πώς μπορούμε να αναφερθούμε στις συντεταγμένες που ορίζουν το μέσο της οθόνης;
- Πού πρέπει να βάλω ενέργειες έτσι ώστε να εκτελεστούν όταν αρχίζει το παιχνίδι;



- Άλλαξε το object "Fire" έτσι ώστε όταν έρχεται σε επαφή με το βασικό χαρακτήρα να καταστρέφεται.
- Τροποποίησε το παιχνίδι έτσι ώστε να ξεκινάει με 10 ζωές.
- Προγραμμάτισε το room έτσι ώστε να δημιουργεί στην εκκίνηση του παιχνιδιού μία ακόμα "Fire" στο πάνω μέρος της οθόνης σε τυχαίο σημείο.
- Άλλαξε το Creation Code έτσι ώστε να δημιουργεί 50 τοίχους μέσα στο room σε τυχαίες θέσεις.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4 GAME MAKER

Συνεχίζουμε με Path και Διάδραση με το ποντίκι



### Αλγοριθμικές έννοιες:

Εμφωλευμένη Δομή Επιλογής, Σύνθετη Δομή Επιλογής

### Έννοιες Προγραμματισμού Παιχνιδιών:

Γεγονός Βήματος (Step Event), Μονοπάτι (Path), Γεγονός Αλληλεπίδρασης μέσω Ποντικιού (Mouse Event), Πίστα (Rooms)

### ? Τι έχουμε φτιάξει ως τώρα;

Στην 3<sup>η</sup> Δραστηριότητα επεκτείναμε το παιχνίδι μας έτσι ώστε:

- να υπάρχουν **objects** φωτιές τις οποίες αν αγγίξει ο παίκτης θα χάνει μια ζωή
- να υπάρχει **object** "Wall" που ο παίκτης δε θα μπορεί να περνάει μέσα από αυτόν, και τέλος
- να δημιουργούνται τα **objects** αυτόματα με το **Creation Code** του **room**.

### ? Τι θα μάθουμε;

Στο πλαίσιο της 4<sup>ης</sup> δραστηριότητας θα ασχοληθούμε με:

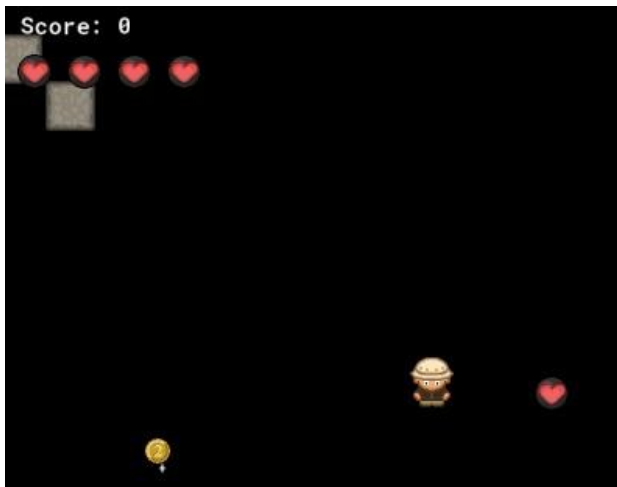
- το **Path** (μονοπάτι), προκειμένου να προγραμματίσουμε τα **objects** να κινούνται μόνο τους μέσα στο **room**
- το ποντίκι, δηλαδή πως ορίζουμε **event** με βάση το κλικ του ποντικιού και πως αναφερόμαστε στη θέση του δείκτη μέσα στην οθόνη

### ? Τι θα φτιάξουμε;

Σήμερα θα επεκτείνουμε το παιχνίδι μας ως εξής:

- Θα δημιουργήσουμε **object** ζωή το οποίο όταν αγγίξει ο παίκτης να αυξάνονται οι ζωές του
- Θα δημιουργήσουμε ένα εχθρικό **object** σκελετό, το οποίο θα κινείται μόνο του και όταν ο παίκτης το αγγίξει θα χάνει μία ζωή και θα γυρνά στην αρχική του θέση
- Θα φτιάξουμε το **object** πέτρα με το οποίο θα μπορεί να επιτίθεται ο παίκτης στα **objects** σκελετός – τα objects πέτρα θα εμφανίζονται μέσα στο **room** με κλικ του ποντικιού
- Τέλος, θα φτιάξουμε περισσότερα από ένα **rooms** προκειμένου να υπάρχουν πίστες στο παιχνίδι

Η λειτουργία των ζώων φαίνεται στις εικόνες 1 και 2.



Εικόνα 1. Ο παίκτης πλησιάζει το object ζωή και η μεταβλητή ζωές έχει την τιμή 4



Εικόνα 2. Ο παίκτης αγγίζει το object ζωή και η μεταβλητή "Lives" αυξάνεται

Η αυτόματη κίνηση του εχθρικού **object** σκελετός φαίνεται στην εικόνα 3.



Εικόνα 3. Η κίνηση του object "Skeleton"

Η λειτουργία της πέτρας σε σχέση με το σκελετό φαίνεται στις εικόνες 4 και 5.

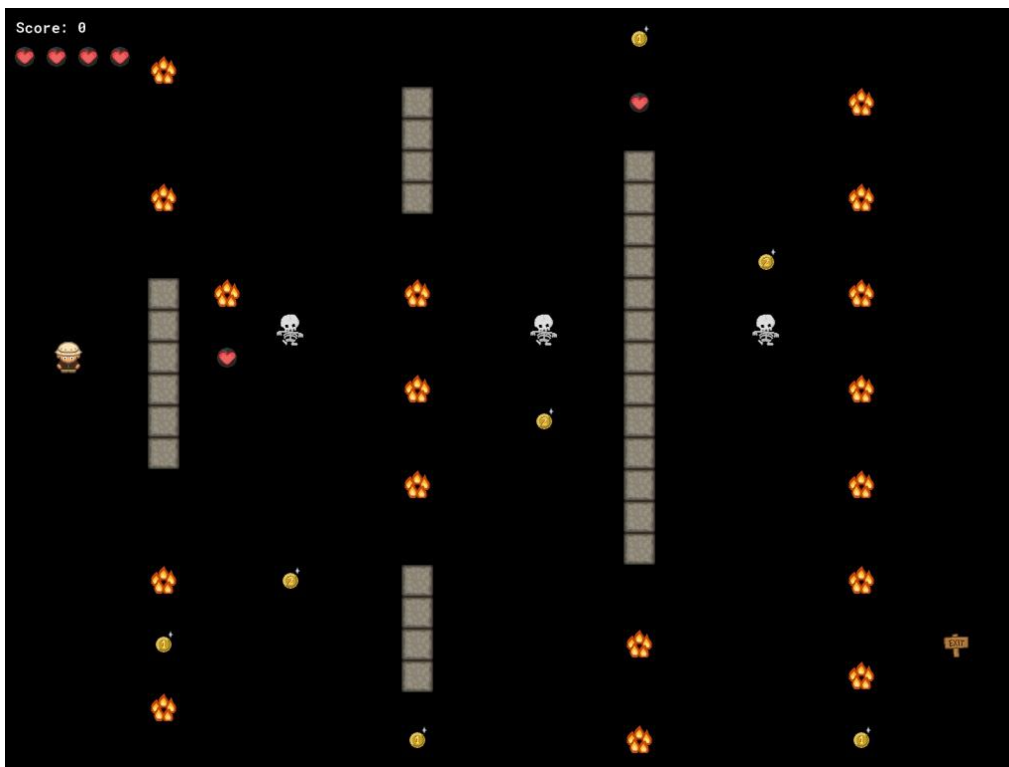


Εικόνα 4. Ο παίκτης πετάει την πέτρα προς το σκελετό



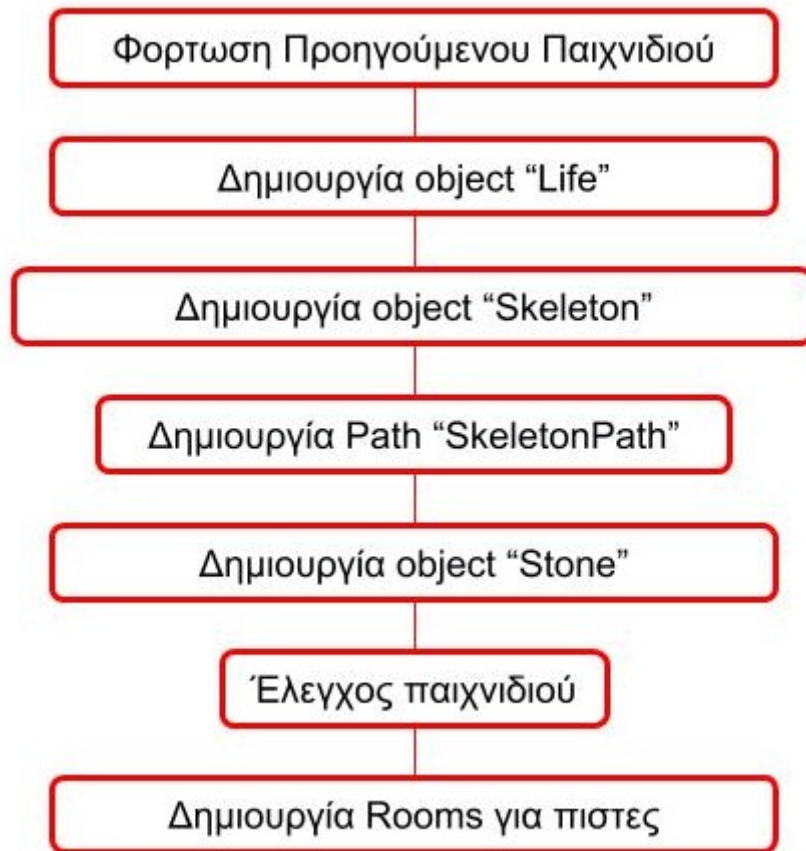
Εικόνα 5. Ο σκελετός εξαφανίζεται

Τέλος, θα φτιάξουμε διαφορετικές πίστες δηλαδή **Rooms** που θα μοιάζουν όπως στην εικόνα 6.



Εικόνα 6. Ενδεικτική πίστα-Room

## Διάγραμμα Ροής Δραστηριότητας



Ας ξεκινήσουμε λοιπόν!

Εκτέλεσε το **Game Maker**, στην αρχική οθόνη του προγράμματος, πάτησε **Open**, στο παράθυρο που εμφανίζεται βρες το προηγούμενο πρόγραμμα και κάνε διπλό κλικ για να ανοίξει.

Στην μπάρα των **Resources** (πόρων), υπάρχουν όλα τα **sprites**, **objects** και **rooms**, που έχεις δημιουργήσει.

### **ΒΗΜΑ ΠΡΩΤΟ: Δημιουργία object "Life"**

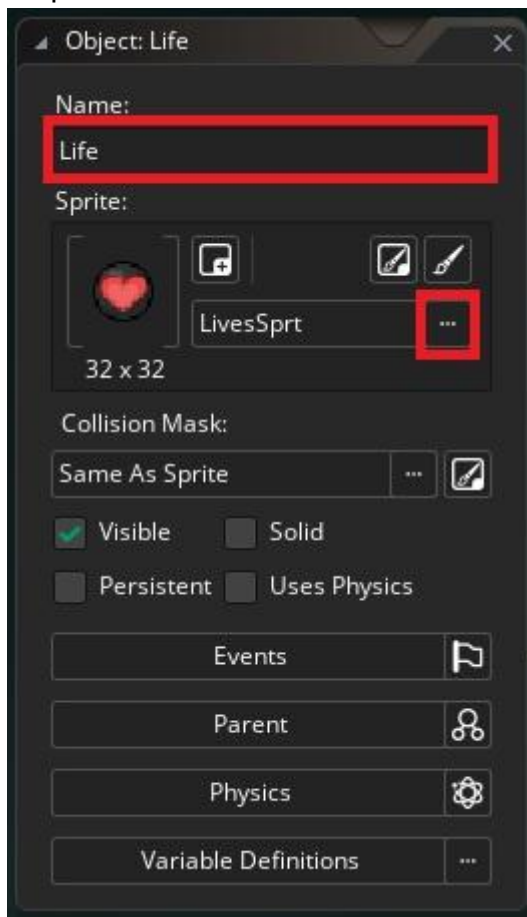
Όπως γνωρίζουμε από όλα τα παιχνίδια αυτού του τύπου, υπάρχουν μέσα στις πίστες **objects** τα οποία αναπαριστούν ζώες τα οποία αν αγγίξει ο παίκτης αυξάνει τη μεταβλητή των ζωών του. Αυτό θα κάνουμε και εμείς τώρα.



Δημιούργησε object "Life", με sprite "LifeSprt" έτσι ώστε να αυξάνεται η μεταβλητή "Lives" του object "MainCharacter" κατά ένα όταν το αγγίζει.

Ακολούθησε τα παρακάτω βήματα:

1. Αρχικά δημιούργησε το **object** "Life" με **sprite** το "LivesSprt", όπως φαίνεται στην εικόνα 7.

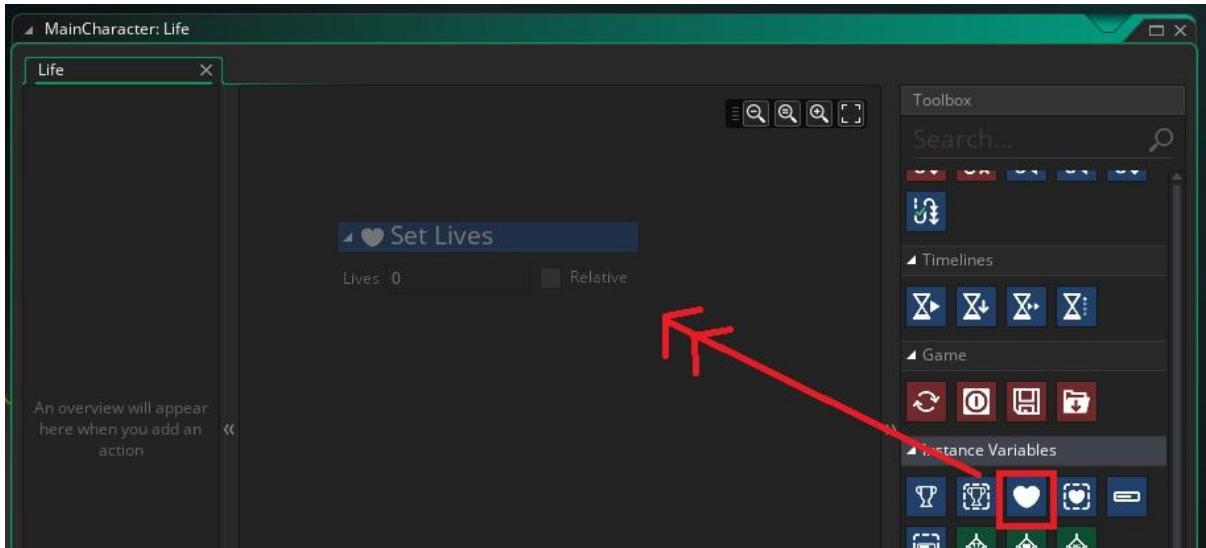


Εικόνα 7. Δημιουργία object "Life"

2. Έπειτα προγραμμάτισέ το **object** αυτό μέσω των **events** έτσι ώστε όταν αγγίζει το "MainCharacter" να εξαφανίζεται, όπως έμαθες από τις προηγούμενες δραστηριότητες.
3. Τέλος, κλείσε το παράθυρο επεξεργασίας του **object**.

Πρέπει στο **object** "MainCharacter" να αυξήσουμε τη μεταβλητή "Lives" όταν αγγίζει το καινούργιο **object** "Life".

4. Άνοιξε το "MainCharacter" και δημιούργησε ένα **Collision Event** με το "Life"
5. Μέσα σε αυτό σύρε την ενέργεια **Set Lives** από την μπάρα ενεργειών όπως φαίνεται στην εικόνα 8.



Εικόνα 8. Εισαγωγή ενέργειας Set Lives στο Collision Event του "MainCharacter" με το "Life"

6. Άλλαξε την τιμή της "Lives" σε 1 και τσέκαρα το relative έτσι ώστε να αυξάνεται η μεταβλητή κατά ένα και όχι να γίνεται σταθερά πάντοτε 1, όπως φαίνεται στην εικόνα 9.



Εικόνα 9. Καθορισμός παραμέτρων ενέργειας Set Life

7. Τέλος, κλείσε το παράθυρο επεξεργασίας του "MainCharacter".

**?** Με ποιό τρόπο όμως θα δώσουμε την εντολή να σταματάει το παιχνίδι κάθε φορά που οι ζωές δηλαδή η μεταβλητή "Lives" φτάνει στο 0;



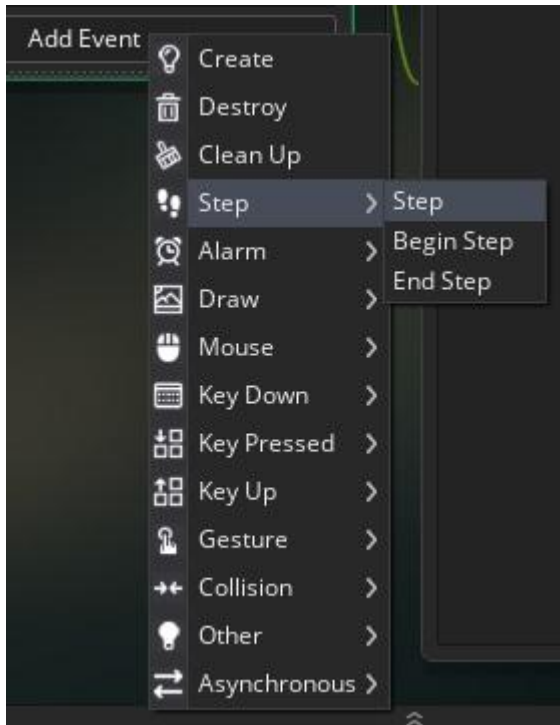
Το Step Event είναι ένα event το οποίο ενεργοποιείται κάθε στιγμή μέσα στο παιχνίδι. Είναι πάρα πολύ χρήσιμο καθώς υπάρχουν ενέργειες που θέλουμε να γίνονται κάθε στιγμή, όπως όταν μετράμε χρόνο, ή όταν θέλουμε να γίνεται έλεγχος μιας μεταβλητής συνεχώς.



Τροποποίησε το object "MainCharacter" έτσι ώστε σε κάθε step του παιχνιδιού να ελέγχεται η τιμή της μεταβλητής "Lives" και αν είναι 0 να σταματάει το παιχνίδι.

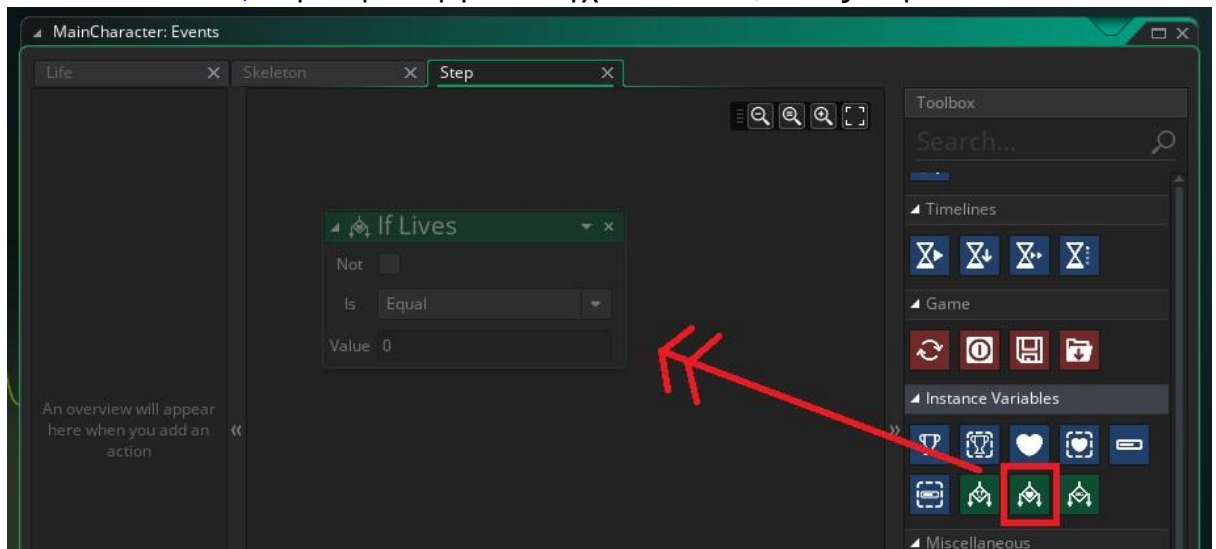
Ακολούθησε τα παρακάτω βήματα:

1. Αρχικά άνοιξε το **object** "MainCharacter"
2. Δημιούργησε ένα καινούργιο **Step Event** όπως φαίνεται στην εικόνα 10.



Εικόνα 10. Δημιουργία Step Event

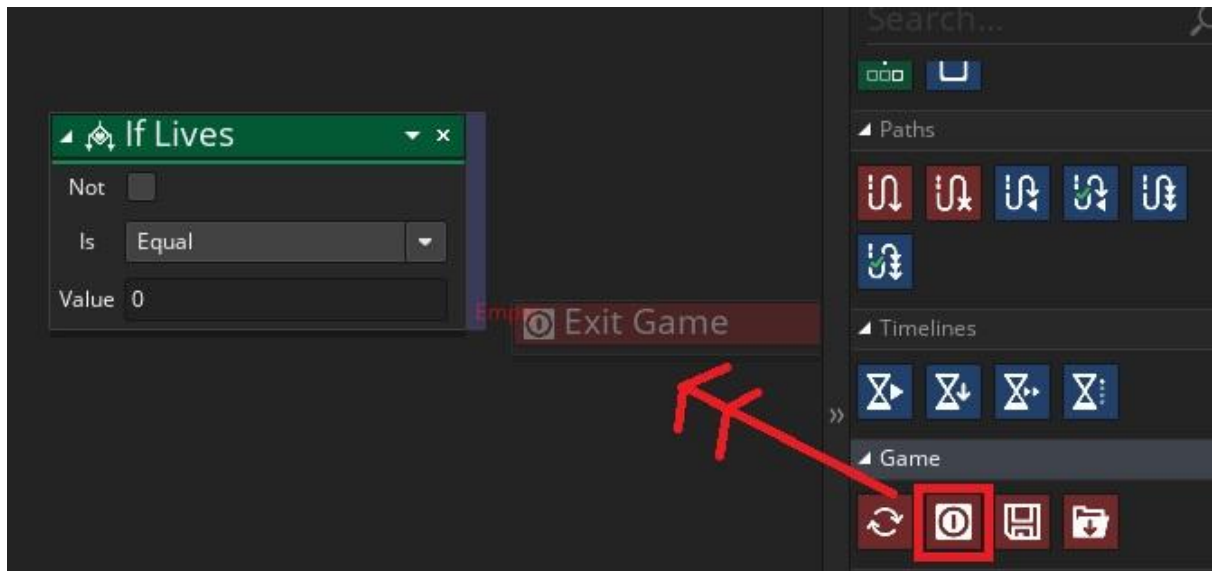
3. Μέσα στο **Event**, σύρε την ενέργεια ελέγχου **If Lives**, όπως στην εικόνα 11.



Εικόνα 11. Εισαγωγή ενέργειας Set Lives

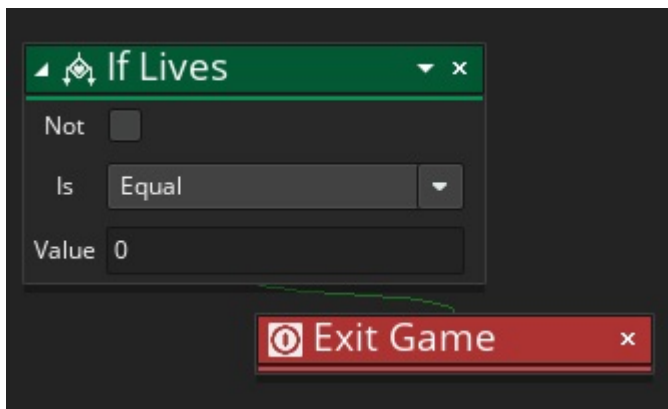
4. Τώρα σύρε την ενέργεια τερματισμού του παιχνιδιού **Exit Game**, στα αριστερά της **If Lives**, έτσι ώστε η ενέργεια να πραγματοποιηθεί αν ισχύει η συνθήκη ελέγχου, όπως φαίνεται στην εικόνα 12.





Εικόνα 12. Εισαγωγή της ενέργειας Exit Game μέσα στην ενέργεια If Lives

Το σωστό αποτέλεσμα είναι αυτό που φαίνεται στην εικόνα 13.



Εικόνα 13. Αποτέλεσμα εισαγωγή της ενέργειας Exit Game στο σύνολο εντολών της If Lives

5. Τέλος, κλείσε το παράθυρο.

Τώρα έχουμε δημιουργήσει το **object** ζωή και ταυτόχρονα έχουμε προγραμματίσει το παιχνίδι έτσι ώστε να τελειώνει όταν οι ζωές φτάσουν στο μηδέν και είμαστε έτοιμοι να προχωρήσουμε.

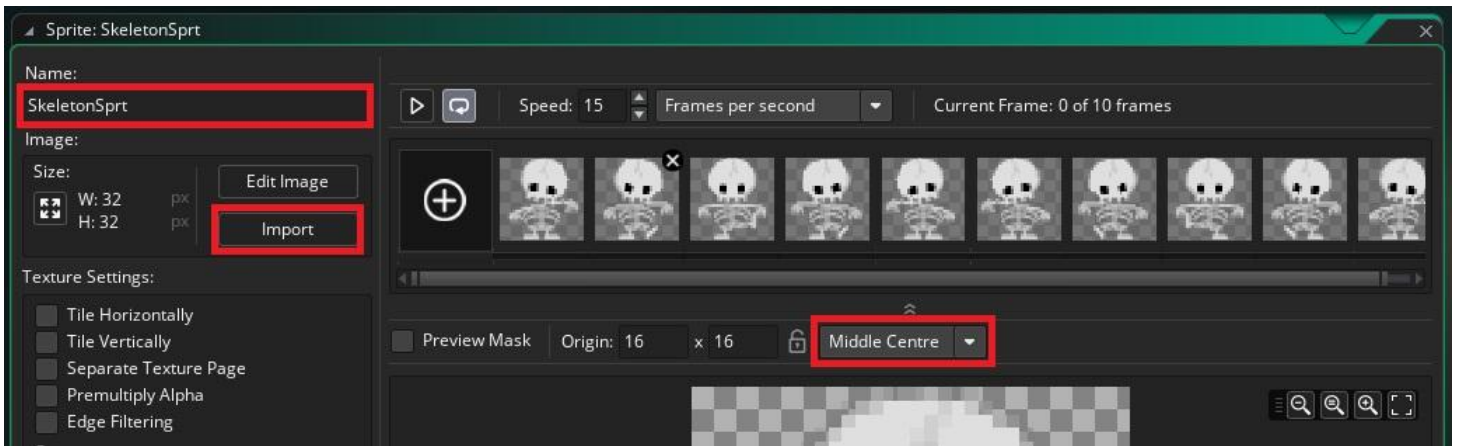
## ΒΗΜΑ ΔΕΥΤΕΡΟ: Δημιουργία του εχθρικού object και του Path



Δημιούργησε ένα καινούργιο object "Skeleton" το οποίο όταν αγγίζει το "MainCharacter", το δεύτερο να επιστρέφει στην αρχική του θέση και να μειώνονται οι ζωές κατά μία.

Για να το κάνεις αυτό ακολούθησε τα παρακάτω βήματα:

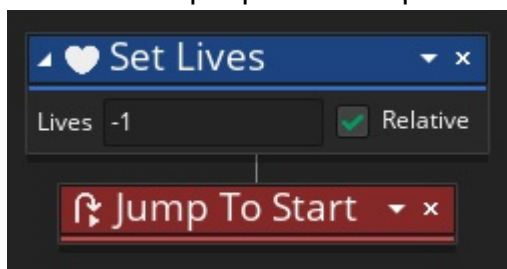
1. Αρχικά φτιάξε ένα καινούργιο **Sprite** με όνομα "SkeletonSprt" χρησιμοποιώντας το αρχείο "skeleton\_walking\_strip10.png", από τον φάκελο Resources, όπως φαίνεται στην εικόνα 14.



Εικόνα 14. Δημιουργία sprite "SkeletonSprt"

2. Κλείσε το παράθυρο επεξεργασίας του **sprite** και δημιούργησε ένα καινούργιο **object** με όνομα "Skeleton" με **sprite** αυτό που μόλις δημιούργησες.
3. Τώρα πήγαινε στο **object** "MainCharacter" και δημιούργησε ένα καινούργιο **Collision Event** με το "Skeleton", μέσα στο οποίο θα βάλεις εξής ενέργειες:
  - a. **Set Lives** με -1 και τσεκαρισμένο το **Relative** και
  - b. **Jump to start**, όπως ακριβώς είχες κάνει και με το "Fire".

Το αποτέλεσμα φαίνεται στην εικόνα 15.



Εικόνα 15. Μείωση της μεταβλητής "Lives" κατά 1 και επαναφορά στην αρχή

4. Τέλος κλείσε το παράθυρο επεξεργασίας του **object**.

Τώρα αυτό που θέλουμε να κάνουμε είναι το **object** "Skeleton" να κινείται από μόνο του σε μία ευθεία γραμμή.

**?** Με ποιο τρόπο κάνουμε τα **objects** να κινούνται από μόνο τους σε ένα συγκεκριμένο μονοπάτι;



Το Path είναι μια διαδρομή την οποία καθορίζουμε εμείς και στην οποία εισάγουμε στα objects έτσι ώστε να την ακολουθήσουν και να κινούνται μόνο τους χωρίς event πληκτρολογίου από τον χρήστη.

Το Path είναι ένα resource το οποίο δημιουργούμε στην μπάρα των πόρων όπως ακριβώς και τα Sprites, τα Objects και τα Rooms.

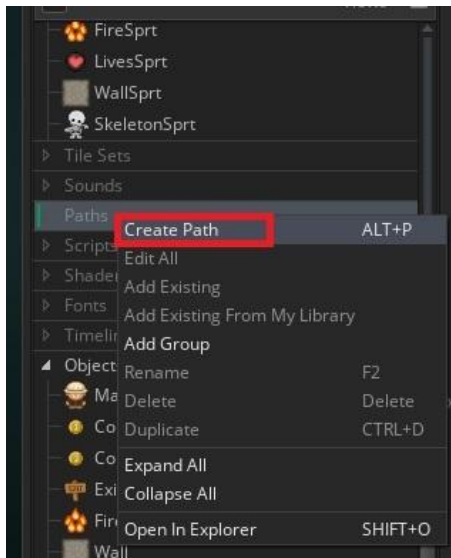
Εμείς προσδιορίζουμε τα σημεία της διαδρομής και η διαδρομή δημιουργείται αυτόματα.

## GOAL

Φτιάξε ένα Path με όνομα "SkeletonPath", το οποίο θα αποτελείται από 2 σημεία κατακόρυφα προκειμένου η κίνηση του αντικειμένου στη διαδρομή να είναι πάνω-κάτω.

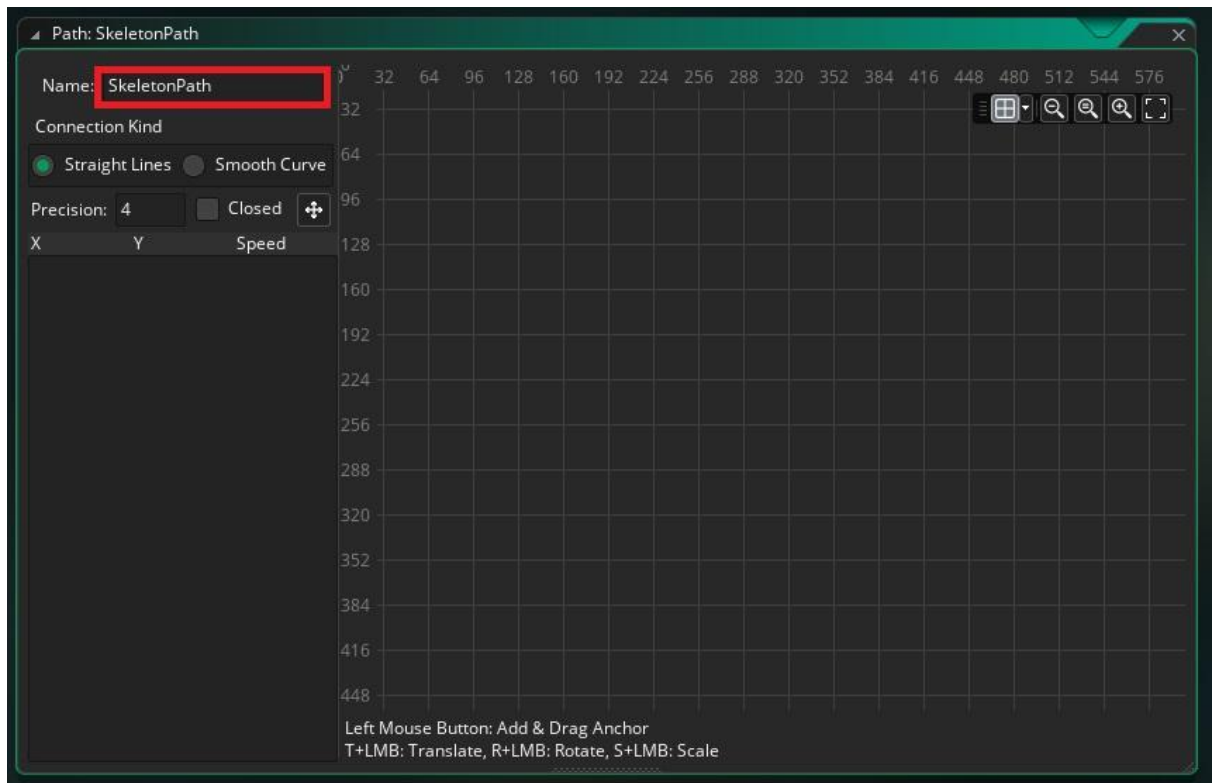
Ακολούθησε τα παρακάτω βήματα:

1. Στην μπάρα των πόρων κάνε δεξί κλικ στη λέξη **Paths** και επίλεξε **Create Path**, όπως φαίνεται στην εικόνα 16.



Εικόνα 16. Δημιουργία Path

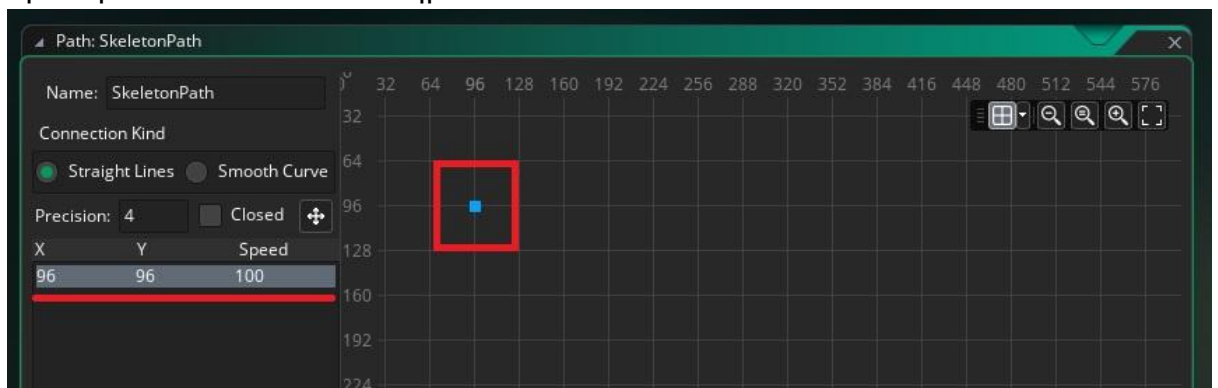
2. Στο παράθυρο επεξεργασίας του **Path** ορίζουμε αρχικά το όνομα του **Path** ("SkeletonPath"), όπως φαίνεται στην εικόνα 17.



Εικόνα 17. Παράθυρο επεξεργασίας Path

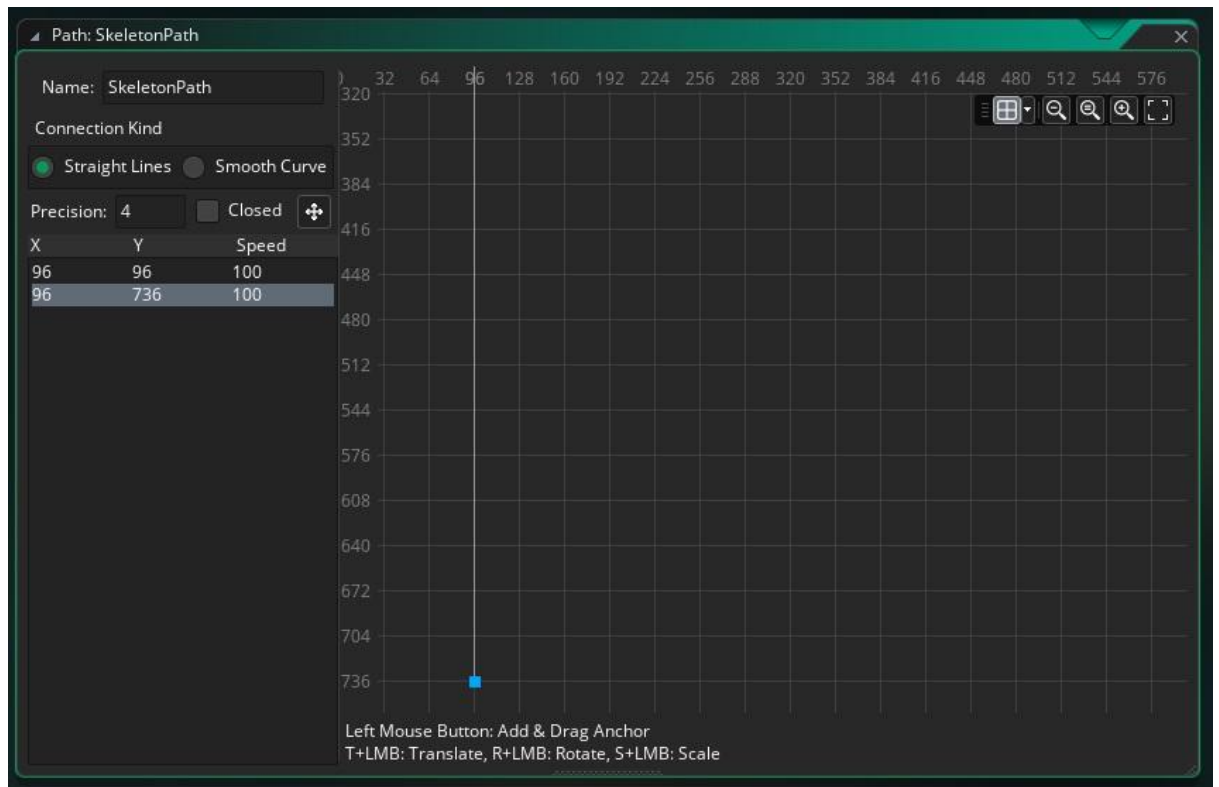
Στο πλέγμα που δίνεται μπορούμε να ορίσουμε τα σημεία της διαδρομής καθώς και εάν επιθυμούμε να είναι ευθεία (επιλογή «Straight Lines») ή καμπύλη (επιλογή «Smooth Curve»).

3. Κάνε κλικ στο σημείο (96,96) μέσα στο πλέγμα. Όπως φαίνεται και στη εικόνα 18, θα εμφανιστεί το πρώτο σημείο του **Path** μέσα στο πλέγμα καθώς και κάτω αριστερά στον πίνακα των σημείων.



Εικόνα 18. Επιλογή του πρώτου σημείου του Path

4. Τώρα κάνε κλικ στο σημείο (96,736) του πλέγματος για να δημιουργήσουμε το επόμενο σημείο. Όπως φαίνεται στην εικόνα 19, δημιουργείται άλλο ένα σημείο στο πλέγμα καθώς και στον πίνακα.



Εικόνα 19. Επιλογή του δεύτερου σημείου του Path

5. Τέλος, κλείσε το παράθυρο επεξεργασίας του **Path**.

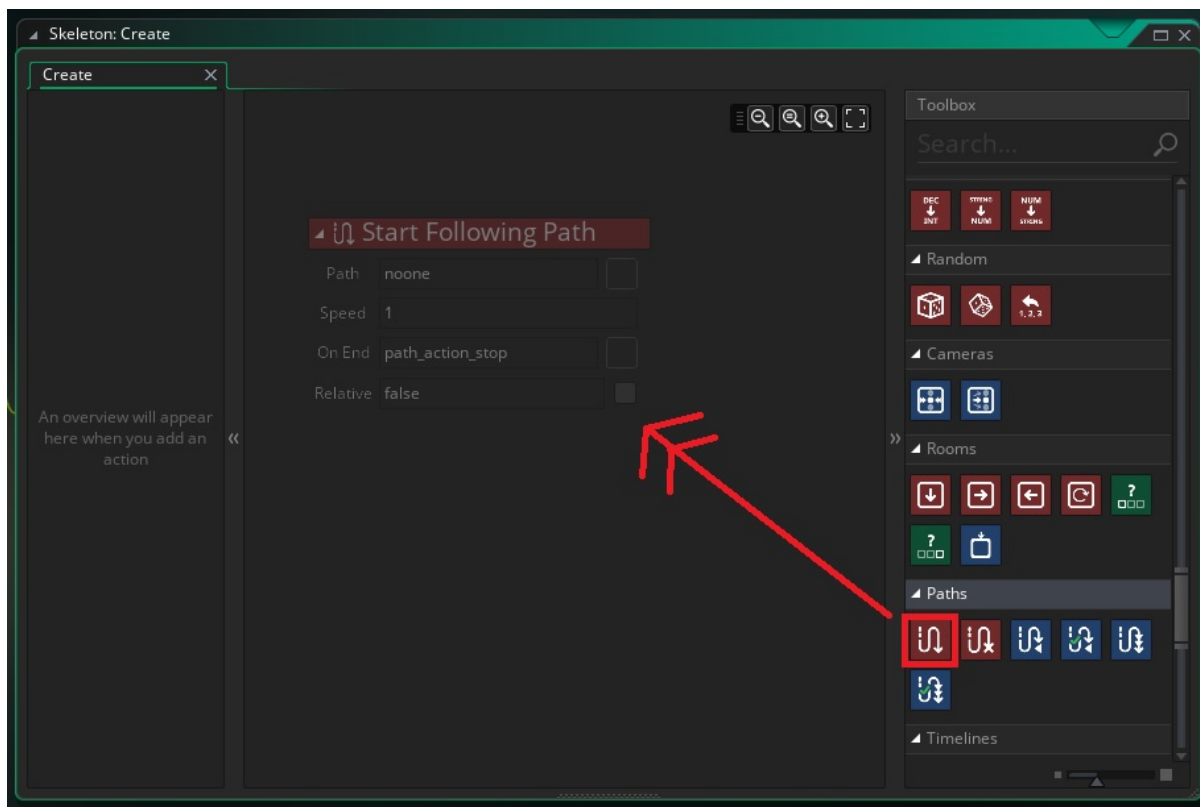
Τώρα έχουμε δημιουργήσει το **Path** και είμαστε έτοιμοι να το βάλουμε σε εφαρμογή στο **object** "Skeleton", έτσι ώστε να κινείται αυτόματα.

## GOAL

Κάνε το **object** "Skeleton" να ακολουθεί το "SkeletonPath" που μόλις έφτιαξες.

Ακολούθησε τα παρακάτω βήματα:

1. Αρχικά άνοιξε το **object** "Skeleton".
2. Μπες στο **Create Event** του **object**.
3. Σύρε την ενέργεια **Start Following Path**, από την μπάρα ενεργειών, όπως φαίνεται στην εικόνα 20.



Εικόνα 20. Εισαγωγή της ενέργειας Start Following Path στο Create Event του object "Skeleton"



Εικόνα 21. Προσδιορισμός παραμέτρων ενέργειας Start Following path

4. Τώρα πρέπει να προσδιορίσουμε τις παραμέτρους αυτής της ενέργειας, όπως φαίνεται στην εικόνα 21.
  - a. Αρχικά επέλεξε το **path**, "SkeletonPath", που δημιουργήσαμε
  - b. Όρισε την ταχύτητα με την οποία θα κινηθεί το **object** μέσω του **path** αυτού π.χ. 4.
  - c. Τέλος, στην επιλογή On End, πρέπει να προσδιορίσεις το τι θα κάνει το **object** μόλις φτάσει στο τέλος.  
Επίλεξε "path\_action\_reverse", προκειμένου να κάνει την αντίθετη κίνηση έτσι ώστε να γυρίσει πάλι στην αρχή.
5. Τέλος, κλείσε το παράθυρο επεξεργασίας του **object** "Skeleton".

## ΒΗΜΑ ΤΡΙΤΟ: Δημιουργία object "Stone"

Σε αυτό το βήμα θα δημιουργήσουμε ένα **object** πέτρα το οποίο θα είναι το όπλο του "MainCharacter" εναντίον των σκελετών. Πιο συγκεκριμένα:

- Θα εμφανίζεται με κλικ του παίκτη στην οθόνη και θα κινείται προς το σημείο που βρίσκεται ο κέρσορας και
- όταν αγγίξει το σκελετό, και ο σκελετός και η πέτρα θα εξαφανίζονται.

? Με ποιο τρόπο πιστεύεις ότι μπορούμε να αναφερθούμε στις συντεταγμένες της θέσης του κέρσορα, δηλαδή το ποντικιού;

Η απάντηση είναι μέσω των μεταβλητών **mouse\_x** και **mouse\_y**.



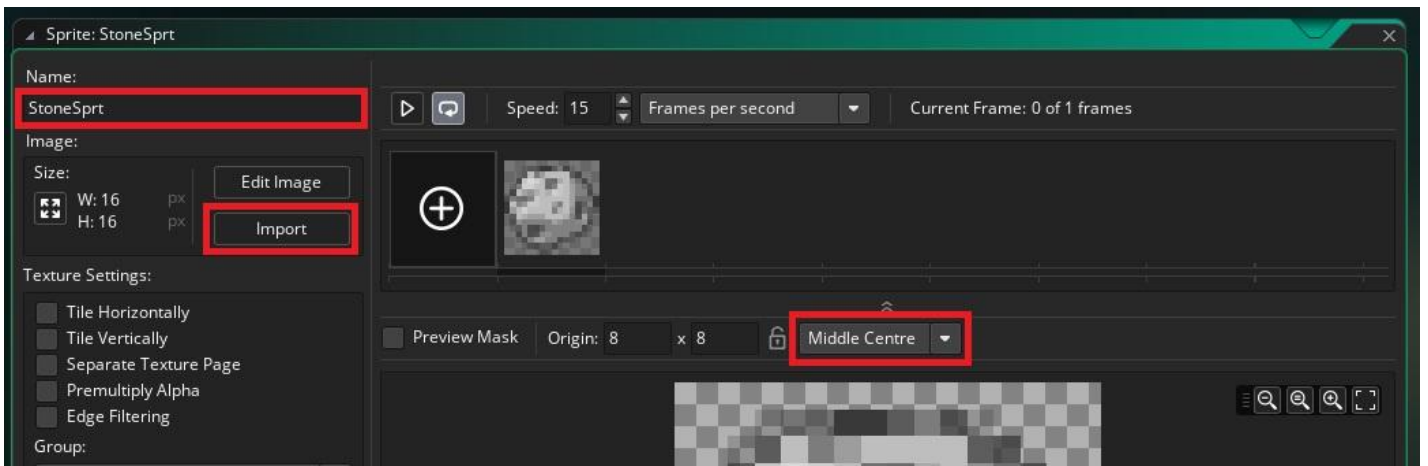
Οι μεταβλητές `mouse_x` και `mouse_y`, είναι δύο μεταβλητές της πλατφόρμας του Game Maker, που στο όνομά τους μας επιστρέφουν τις συντεταγμένες του ποντικιού στον οριζόντιο και στον κάθετο άξονα αντίστοιχα. Δεν μπορούμε εμείς να αλλάξουμε την τιμή τους γιατί η τιμή αυτή εξαρτάται από το που δείχνει ο κέρσορας κάθε στιγμή.

## GOAL

Δημιούργησε ένα object "Stone" και προγραμμάτισέ το έτσι ώστε με το που δημιουργηθεί να κινηθεί προς τη θέση του κέρσορα.

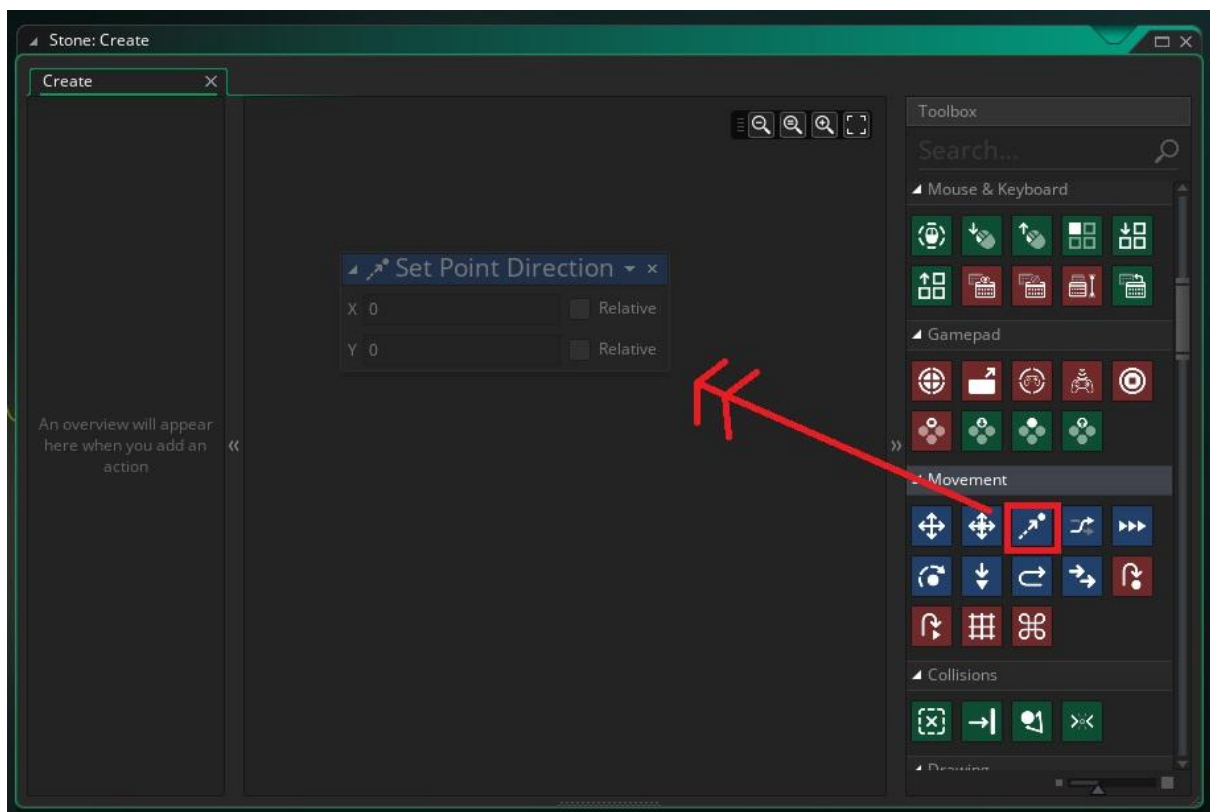
Ακολούθησε τα παρακάτω βήματα:

1. Αρχικά, δημιούργησε ένα καινούργιο **sprite** με όνομα "StoneSprt" με αρχείο "stone.png" από το φάκελο Resources, όπως φαίνεται στην εικόνα 22.



Εικόνα 22. Δημιουργία sprite "StoneSprt"

- Μετά, δημιούργησε ένα καινούργιο **Object**, με όνομα "Stone" και **sprite** το παραπάνω.
- Μπες στο **Create Event** και σύρε την ενέργεια καθορισμού κατεύθυνσης της κίνησης **Set Point Direction** από την μπάρα ενεργειών, όπως φαίνεται στην εικόνα 23.



Εικόνα 23. Εισαγωγή της ενέργειας Set Point Direction στο Create Event του object "Stone"

Οι παράμετροι της ενέργειας αυτής είναι οι συντεταγμένες του σημείου προς το οποίο θα οριστεί η κατεύθυνση. Εμείς θέλουμε το σημείο αυτό να είναι ο κέρσορας.

- Γράψε στα πεδία X και Y, *mouse\_x* και *mouse\_y* αντίστοιχα, όπως φαίνεται στην εικόνα 24.





Εικόνα 24. Προσδιορισμός συντεταγμένων κατεύθυνσης

Τώρα πρέπει να καθορίσουμε την ταχύτητα της κίνησης προς αυτήν την κατεύθυνση.

5. Σύρε την ενέργεια **Set Speed**, κάτω από την προηγούμενη και δώσε σαν τιμή speed το 10, όπως φαίνεται στην εικόνα 25.



Εικόνα 25. Καθορισμός ταχύτητας

6. Τέλος, κλείσε το παράθυρο επεξεργασίας του **object**.

Με αυτόν τον τρόπο το **object** "Stone" μόλις δημιουργηθεί θα κινηθεί προς το σημείο που βρίσκεται ο κέρσορας.

Τι θα γίνεται όμως όταν συγκρουστεί με το **object** "Skeleton";

Η απάντηση είναι ότι και το "Skeleton" και το "Stone" θέλουμε όταν αγγίζει το ένα το άλλο να αλληλοκαταστρέφονται.

## GOAL

Τροποποίησε τα δύο **objects** "Skeleton" και "Stone" έτσι ώστε όταν αγγίζει το ένα το άλλο να καταστρέφονται.

Για να καταστραφεί το "Skeleton" όταν αγγίζει τη πέτρα, ακολούθησε τα παρακάτω βήματα:

1. Άνοιξε το **object**
2. Μπες στο **Collision Event** με το "Stone"
3. Σύρε την ενέργεια **Destroy Instance** μέσα στο **Event**.
4. Κλείσε το παράθυρο

Τώρα κάνε το ίδιο με το **object** "Stone".

Με αυτόν το τρόπο τα δύο **objects** θα αλληλοκαταστραφούν αν αγγίζει το ένα το άλλο.

Πότε όμως και πού θα δημιουργείται το **object** "Stone". Θέλουμε να δημιουργείται με κλικ το ποντικιού στην οθόνη με αρχική θέση το "MainCharacter" έτσι ώστε να φαίνεται ότι ο παίκτης είναι αυτός που επιτίθεται στα εχθρικά **objects**.

? Με ποιό τρόπο όμως μπορώ να εκτελέσω ενέργειες με κλικ του ποντικιού σε ένα **object**;



Τα **Mouse Events** είναι μία οικογένεια από **events** που αναφέρονται στη δραστηριότητα του ποντικιού.

Περιλαμβάνει ενέργειες όπως δεξί ή αριστερό κλικ πάνω σε ένα **object**, κράτημα του δεξιού ή του αριστερού κλικ, κλικ σε περιοχή έξω από το **object**, πέρασμα του δείκτη του ποντικιού πάνω από ένα **object** χωρίς κλικ και άλλα.

Μας είναι ιδιαίτερα χρήσιμα καθώς με διάδραση με το ποντίκι το παιχνίδι εμπλουτίζεται και γίνεται πιο ενδιαφέρον.

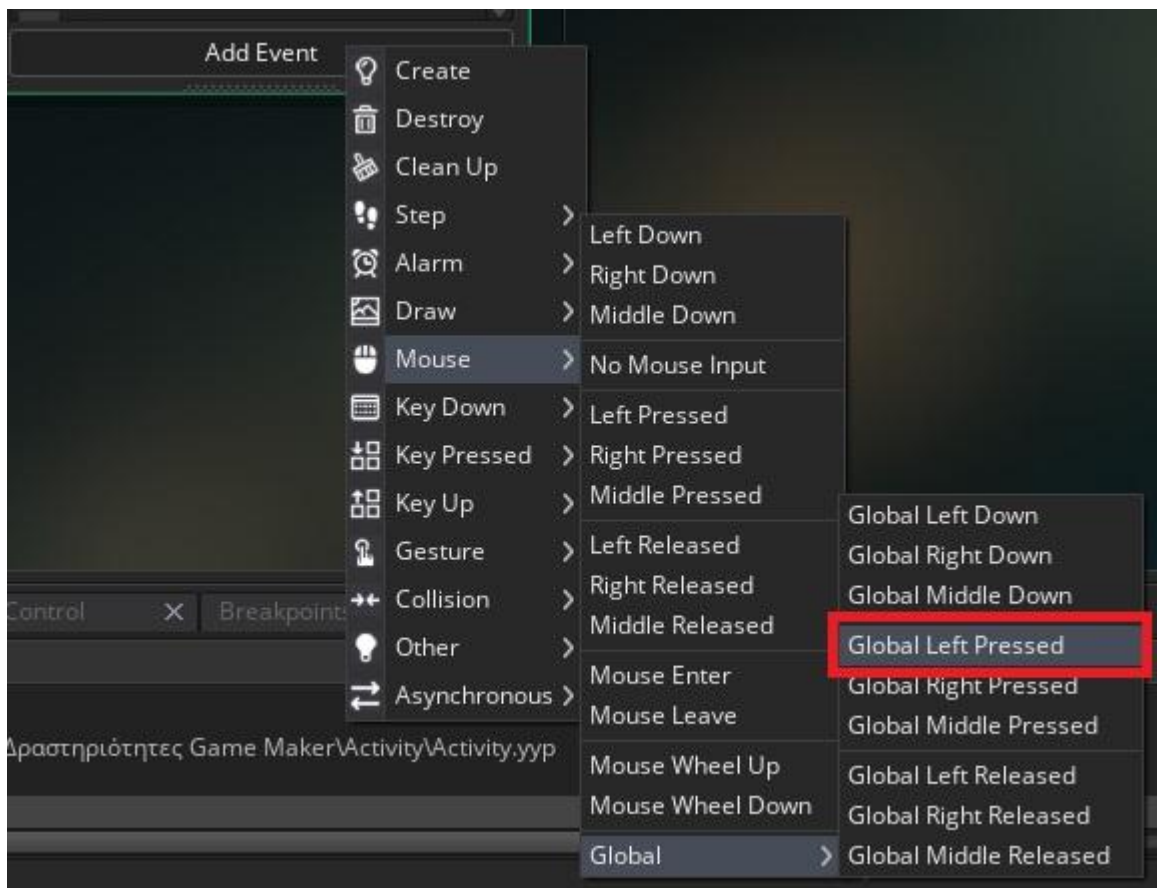
## GOAL

Τροποποίησε το **object** "MainCharacter" έτσι ώστε κάθε φορά που γίνεται κλικ οπουδήποτε στην οθόνη να δημιουργείται **object** "Stone" στη θέση του "MainCharacter".

Ακολούθησε τα παρακάτω βήματα:

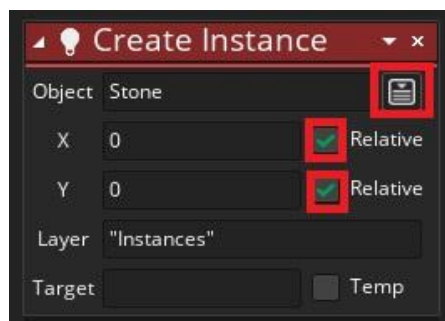
1. Άνοιξε το "MainCharacter"
2. Μπες στο **Global Left Pressed Event** που βρίσκεται στην οικογένεια των **Mouse Events**, όπως φαίνεται στην εικόνα 26.

Για την ακρίβεια η λέξη **Global** σημαίνει **Mouse Event** το οποίο να μην γίνεται πάνω στο **object** αλλά γενικά σε όλη την οθόνη και επίσης η λέξη **Pressed** σημαίνει **event** όταν πατηθεί το κουμπί του ποντικιού και όχι όταν κρατιέται πατημένο ή όταν σηκώνεται. Αυτές οι επιλογές υπάρχουν και για **Right** καθώς και για **Middle** κλικ, δηλαδή δεξί και μεσαίο.



Εικόνα 26. Δημιουργία Global Left Pressed Event

Μέσα σε αυτό το **event** σύρε μία ενέργεια **Create Instance** και ρύθμισε τις παραμέτρους ως εξής:



Εικόνα 27. Προσδιορισμός παραμέτρων Create Instance

- a. Επίλεξε το **object** "Stone"
- b. Σε ότι αφορά τη θέση στην οποία θα δημιουργηθεί δηλαδή το X και το Y του καινούργιου **object** θέλουμε (0,0) με **Relative** τσεκαρισμένο, δηλαδή στην αρχή των αξόνων σε σχέση με τις συντεταγμένες του "MainCharacter", όπως φαίνεται στην εικόνα 27.

3. Τέλος, κλείσε το παράθυρο.

Με αυτόν τον τρόπο το **object** "Stone" θα δημιουργείται στη θέση που βρίσκεται το "MainCharacter" κάθε φορά που θα γίνεται κλικ οπουδήποτε μέσα στην οθόνη και όπως προγραμματίσαμε νωρίτερα θα κατευθυνθεί προς της θέση του κέρσορα.

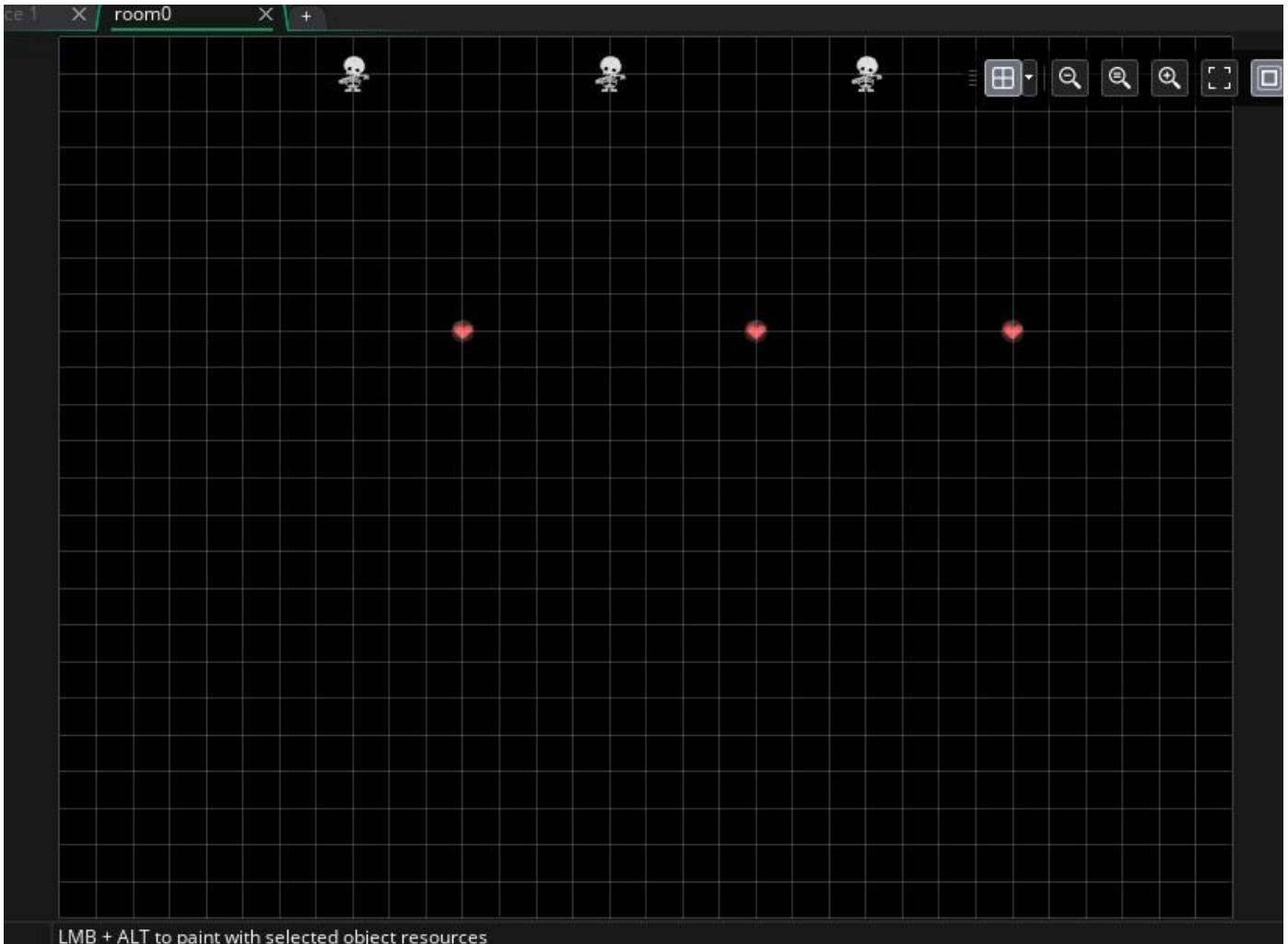
## ΒΗΜΑ ΤΕΤΑΡΤΟ: Έλεγχος

Για να ολοκληρώσουμε το παιχνίδι μας στη 4<sup>η</sup> Δραστηριότητα, θα πρέπει να ελέγξουμε αν τα **objects** λειτουργούν με τον τρόπο που θα θέλαμε.

Για να το κάνουμε αυτό πρώτα πρέπει να τοποθετήσουμε στο ήδη υπάρχον **Room** τα **objects** που μόλις δημιουργήσαμε.

Άνοιξε το Room0 και τοποθέτησε μέσα σε αυτό

- 3 **objects** "Skeleton" και
- 3 **objects** "Life", όπως φαίνεται στην εικόνα 28.



Εικόνα 28. Τοποθέτηση αντικειμένων μέσα στο Room0

Τώρα τρέξε το παιχνίδι και δοκίμασε όλες τις λειτουργίες των objects που έφτιαξες.

## ΒΗΜΑ ΠΕΜΠΤΟ: Δημιουργία Επιπέδων

Σε αυτό το βήμα θα δημιουργήσουμε πολλά επίπεδα δηλαδή πίστες τις οποίες πρέπει να περάσει ο παίκτης για να τερματίσει το παιχνίδι. Αυτό θα γίνει μέσω των **rooms**. Στο παιχνίδι μας έχουμε ήδη ένα **room** στο οποίο τα **objects** δημιουργούνται με τυχαίο τρόπο.

Στις καινούργιες όμως πίστες που θα φτιάξουμε θα τοποθετήσουμε τα **objects** μας εμείς χειροκίνητα προκειμένου να υπάρχει διαφορετική δυσκολία μεταξύ τους. Πριν δημιουργήσουμε τα **rooms** όπως τα θέλουμε θα πρέπει να κάνουμε μία τροποποίηση στον τρόπο τερματισμού του **room**.

Υπενθυμίζουμε ότι το παιχνίδι τερματίζεται όταν το "MainCharacter" αγγίξει το "ExitObject", μόνο όταν η μεταβλητή του σκορ είναι μεγαλύτερη ή ίση του 10.

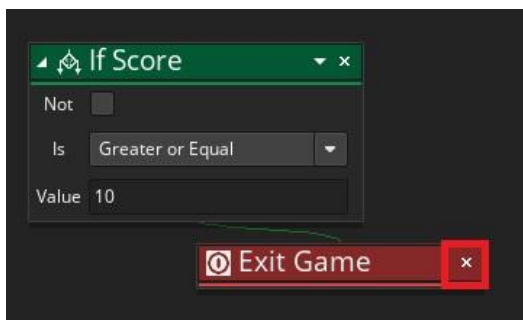
Επειδή θα έχουμε πολλές πίστες πρέπει να τροποποιήσουμε τον κώδικα ως εξής:  
Αν βρισκόμαστε στο τελευταίο **room** τότε ΕΞΟΔΟΣ ΑΠΟ ΤΟ ΠΑΙΧΝΙΔΙ  
Αλλιώς ΠΡΟΧΩΡΗΣΕ ΣΤΟ ΕΠΟΜΕΝΟ ROOM.

## GOAL

Τροποποίησε τον κώδικα τερματισμού του "MainCharacter" στο Collision Event με το "ExitObject" έτσι ώστε να τερματίζει μόνο όταν πρόκειται για το τελευταίο room.

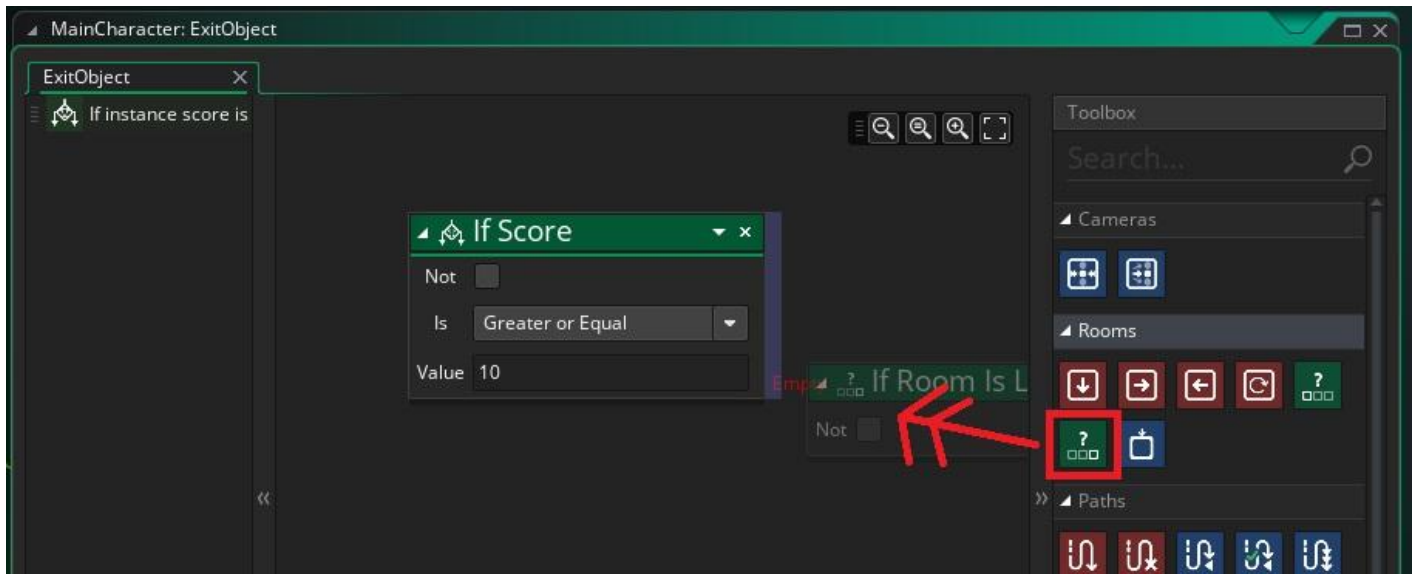
Για να το κάνεις αυτό ακολούθησε τα παρακάτω βήματα:

1. Άνοιξε το **object** "MainCharacter" και μπες στο **Collision Event** με το "ExitObject".
2. Κάνε κλικ στο X της ενέργειας **Exit Game** έτσι ώστε να διαγραφεί από τον κώδικα, όπως φαίνεται στην εικόνα 29.



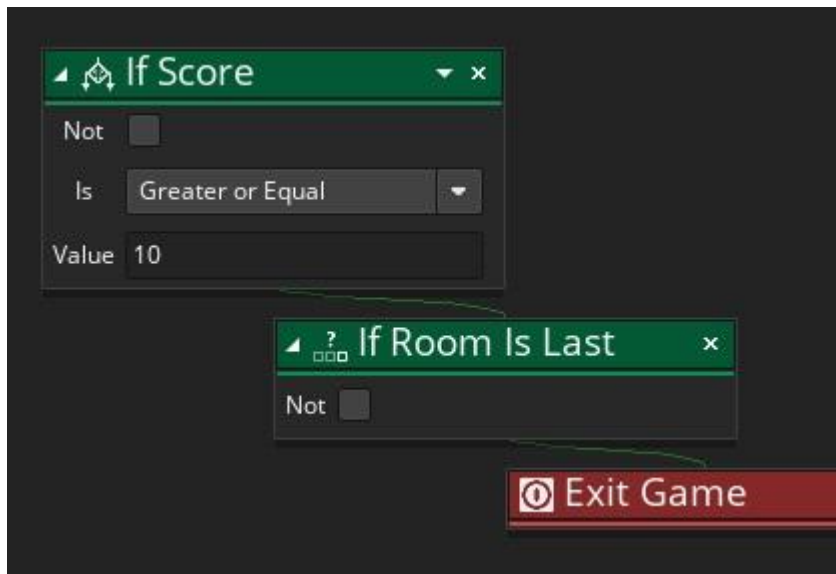
Εικόνα 29. Διαγραφή ενέργειας Exit Game

3. Σύρε την ενέργεια ελέγχου **If Room is Last** στην ίδια θέση που βρισκόταν και το **Exit Game**, όπως φαίνεται στην εικόνα 30.



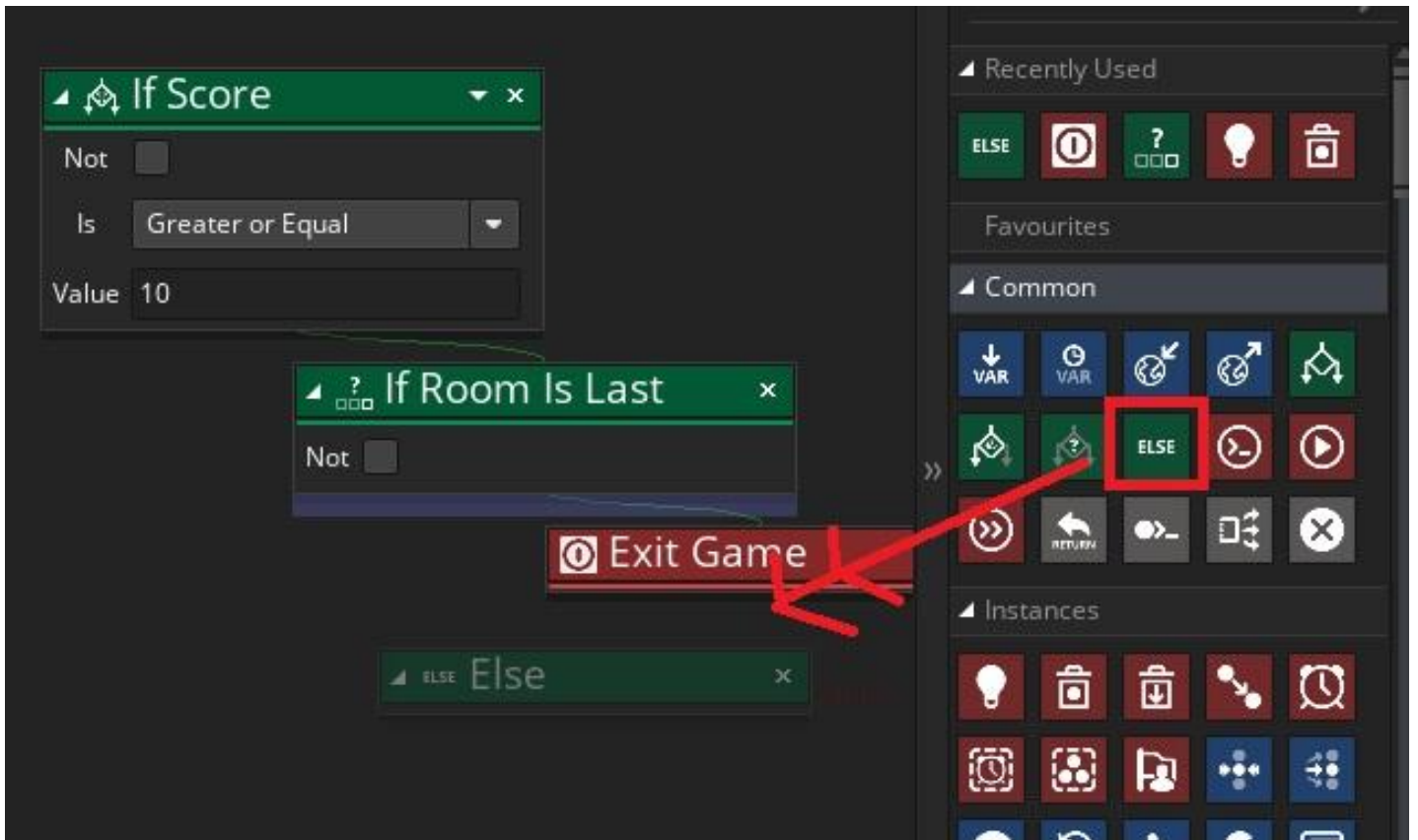
Εικόνα 30. Εισαγωγή της ενέργειας If Room Is Last μέσα στο σύνολο εντολών της If Score

4. Τώρα σύρε μέσα στον κώδικα της ενέργειας ελέγχου **If Room is Last** την ενέργεια **Exit Game**, έτσι ώστε να τερματίζεται το παιχνίδι μόνο όταν βρισκόμαστε στο τελευταίο **room**. Το αποτέλεσμα φαίνεται στην εικόνα 31.



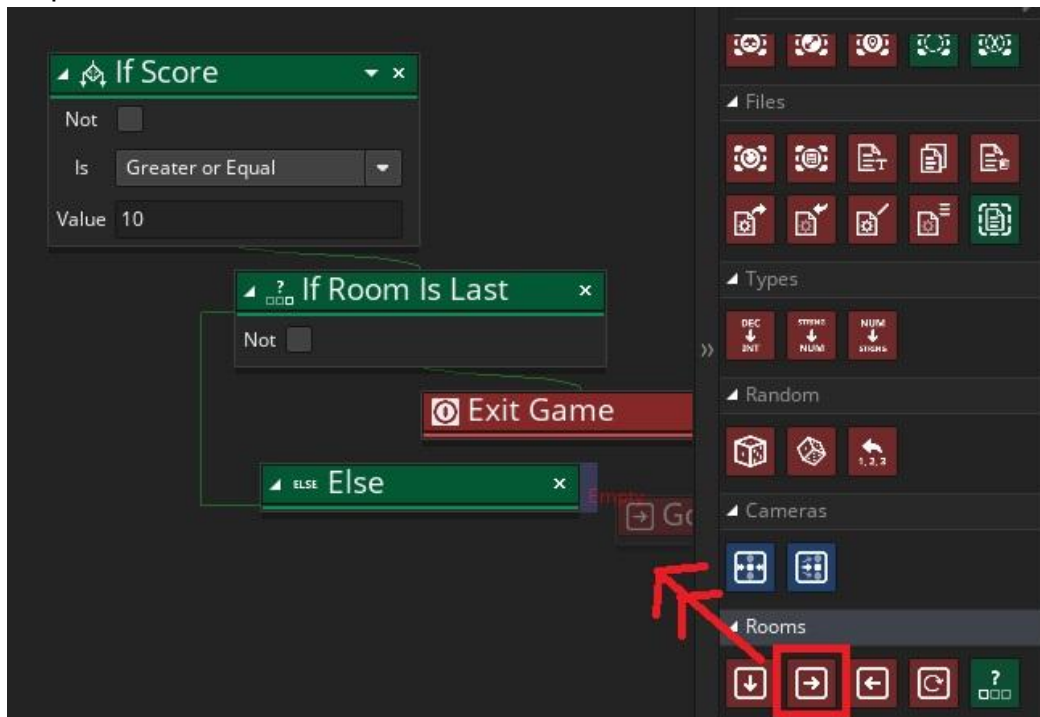
Εικόνα 31. Το αποτέλεσμα της εισαγωγής της ενέργειας Exit Game μέσα στο σύνολο εντολών της If Room Is Last

5. Τώρα σύρε την ενέργεια ελέγχου **Else** κάτω από το **If Room is Last** όπως φαίνεται στην εικόνα 32.



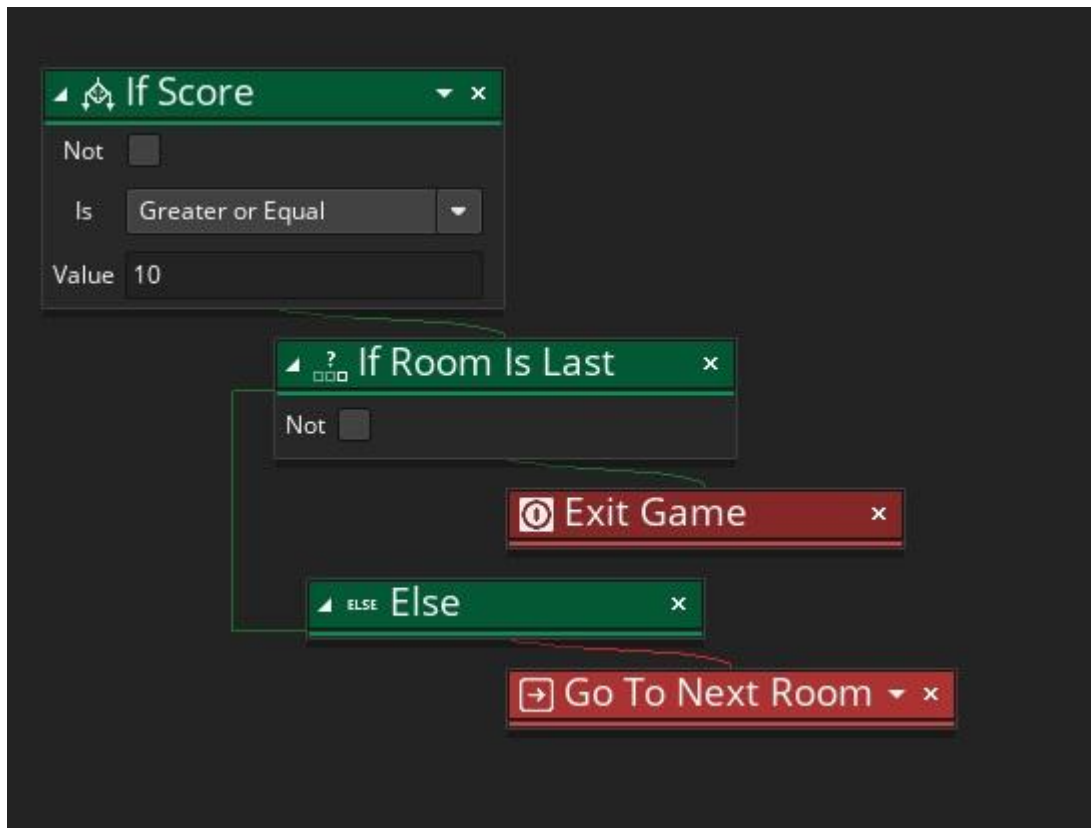
Εικόνα 32. Εισαγωγή της ενέργειας Else κάτω από την If Room Is Last

6. Μέσα στην ενέργεια **Else** σύρε την ενέργεια **Go To Next Room**, έτσι ώστε όταν δεν είμαστε στο τελευταίο **room**, να προχωράμε στο επόμενο, όπως φαίνεται στην εικόνα 33.



Εικόνα 33. Εισαγωγή της ενέργειας Go To Next Room μέσα στο σύνολο εντολών του Else

Το αποτέλεσμα του κώδικα που έχεις φτιάξει πρέπει να είναι σύμφωνο με την εικόνα 34.



Εικόνα 34. Κώδικας τερματισμού ενός Room

7. Τέλος κλείσε το παράθυρο επεξεργασίας του "MainCharacter".

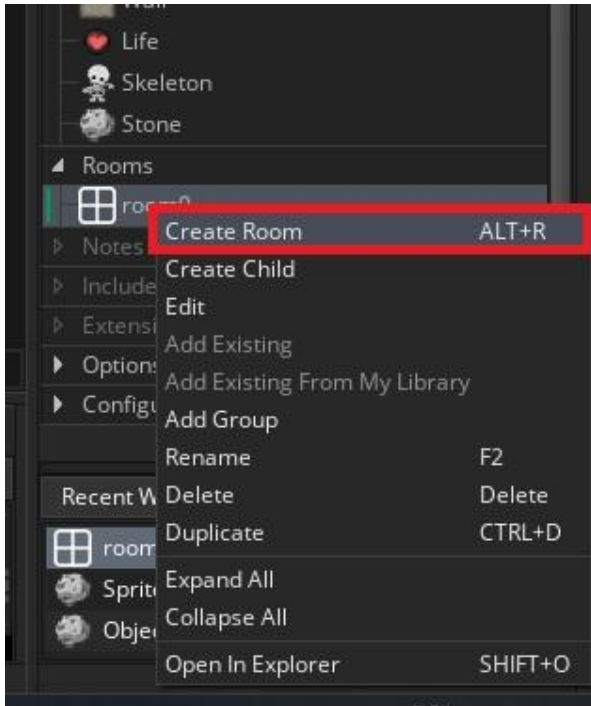
Τώρα είμαστε έτοιμοι να δημιουργήσουμε πολλά **rooms** δηλαδή πίστες.

## GOAL

Δημιούργησε 3 rooms με διαφορετική δυσκολία το καθένα και τοποθέτησε τα objects που έχεις φτιάξει όπου θέλεις μέσα σε αυτά.

Για να δημιουργήσεις ένα καινούργιο **room** κάνε δεξί κλικ και **Create Room**, πάνω στα **Rooms** στην μπάρα των πόρων δεξιά, όπως φαίνεται στην εικόνα 35.





Εικόνα 35. Δημιουργία Room

Τοποθέτησε τα **objects** μέσα σε αυτό το **room** όπως θέλεις εσύ. Ένα παράδειγμα **room**-πίστας φαίνεται στην εικόνα 36.



Εικόνα 36. Παράδειγμα τοποθέτησης objects στο room

Αφού φτιάξεις 3 καινούργια **rooms**, κλείσε όλα τα ανοιχτά παράθυρα επεξεργασίας και παίξε το παιχνίδι που έφτιαξες.

## Καλή Διασκέδαση!



- Τι είναι και πότε χρησιμοποιούμε το Path;
- Πώς μπορώ να έχω τις συντεταγμένες του θέσης του ποντικιού στην οθόνη;
- Τι είναι τα Mouse Event και σε τι χρησιμεύουν;
- Τι είδους Event θα χρησιμοποιήσω για να γίνονται ενέργειες κάθε στιγμή μέσα στο παιχνίδι;

### Do!

- Τροποποίησε τον κώδικα του "MainCharacter" έτσι ώστε να τερματίζει όταν οι ζωές γίνουν λιγότερες από 0 και όχι ίσες με 0.
- Τροποποίησε το πρόγραμμα έτσι ώστε να εμφανίζει τις συντεταγμένες του ποντικιού στην οθόνη.
- Δημιούργησε ένα καινούργιο Path έτσι ώστε να κάνει κύκλο το object στο οποίο εφαρμόζεται.
- Άλλαξε τον κώδικα του "Skeleton" έτσι ώστε να εξαφανίζεται όταν ο παίκτης κάνει κλικ πάνω του.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5 GAME MAKER

Συνεχίζουμε με Ήχους, Timelines και Backgrounds



Έννοιες Προγραμματισμού Παιχνιδιών:

Timeline (Χρονοπρογραμματισμός)

? Τι έχουμε φτιάξει ως τώρα;

Στη 4<sup>η</sup> Δραστηριότητα επεκτείναμε το παιχνίδι μας έτσι ώστε:

- Να υπάρχει **object** “Life” τέτοιο ώστε όταν ο παίκτης το αγγίζει να κερδίζει μία ζωή
- Να υπάρχει **object** “Skeleton” που κινείται μόνο του με **Path** και όταν ο παίκτης το αγγίζει ξαναγυρνά στην αρχική του θέση και χάνει μια ζωή
- Να μπορεί ο παίκτης να πετά **object** “Stone” προς τα “Skeleton” με σκοπό να τα εξαφανίσει
- Να υπάρχουν περισσότερες από μία πίστες με διαφορετικό επίπεδο δυσκολίας

? Τι θα μάθουμε;

Στο πλαίσιο της 5<sup>ης</sup> δραστηριότητας θα ασχοληθούμε με:

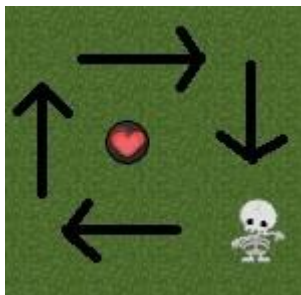
- **Sounds** (ήχους) που θα ακούγονται σε συγκεκριμένες στιγμές του παιχνιδιού έτσι ώστε να είναι πιο ρεαλιστικό
- **Timelines** (χρονοπρογραμματισμός), έτσι ώστε να εμφανίζονται αντικείμενα κατά τη διάρκεια του παιχνιδιού
- **Background** (υπόβαθρο στα Room), έτσι ώστε να εμπλουτίσουμε την συνολική εικόνα του παιχνιδιού

? Τι θα φτιάξουμε;

Σήμερα θα επεκτείνουμε το παιχνίδι μας ως εξής:

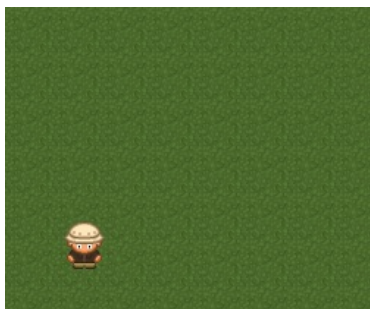
- Θα δημιουργήσουμε **sounds** που θα ακούγονται όταν ο παίκτης παίρνει ένα item (Coin ή Life), όταν κερδίζει το παιχνίδι και πάει στο επόμενο **room**, όταν χάνει μία ζωή ή σκοτώνει ένα **object** “Skeleton”, και όταν πετάει ένα **object** “Stone”
- Θα δημιουργήσουμε ένα αντίγραφο **object** “Skeleton2” που θα έχει την ίδια ακριβώς λειτουργία με το πρώτο και θα το αλλάξουμε έτσι ώστε να σχηματίζει η κίνησή του ένα τετράγωνο και όχι μία ευθεία γραμμή
- Θα δημιουργήσουμε **Timeline**, όπου θα δημιουργούνται τα **object** “Coin1” και “Coin2” κατά τη διάρκεια του παιχνιδιού και όχι στην αρχή
- Θα τοποθετήσουμε **Background** στα **Rooms** μέσω **Tile** (πλακίδιο)

Η λειτουργία του object “Skeleton2” φαίνεται στην εικόνα 1.

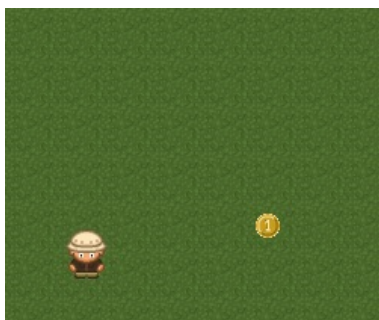


Εικόνα 1. Το Path που ακολουθεί το object “Skeleton2”

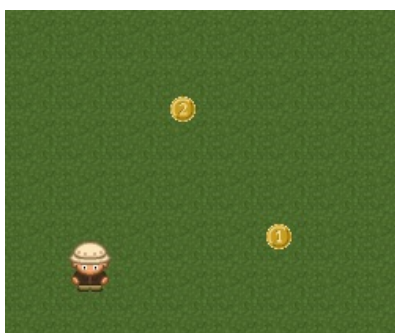
Η λειτουργία του Timeline φαίνεται στις εικόνες 2,3 και 4.



Εικόνα 2. Το Room κατά τη δημιουργία του δεν περιέχει καθόλου Coins

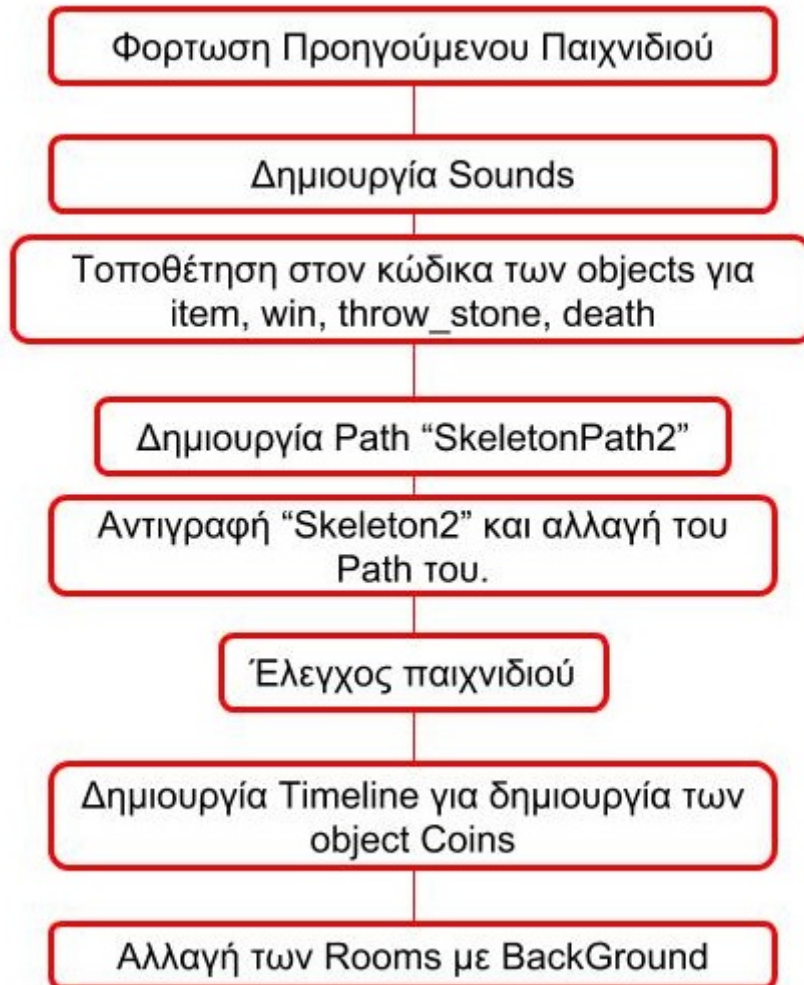


Εικόνα 3. Στο Room μετά από λίγα δευτερόλεπτα εμφανίζονται objects “Coin1”



Εικόνα 4. Στο Room τελικά εμφανίζονται και objects “Coin2”

## Διάγραμμα Ροής Δραστηριότητας



Ας ξεκινήσουμε λοιπόν!

Εκτέλεσε το **Game Maker**, στην αρχική οθόνη του προγράμματος, πάτησε **Open**, στο παράθυρο που εμφανίζεται βρες το προηγούμενο πρόγραμμα και κάνε διπλό κλικ για να ανοίξει.

Στην μπάρα των **Resources** (πόρων), υπάρχουν όλα τα **sprites**, **objects** και **rooms**, που έχεις δημιουργήσει.

## ΒΗΜΑ ΠΡΩΤΟ: Εισαγωγή Sounds

Ένα παιχνίδι για να είναι πιο ζωντανό και πιο διασκεδαστικό πρέπει οπωσδήποτε να συμπεριλαμβάνει και ήχους, οι οποίοι θα συνοδεύουν τη δράση. Για παράδειγμα όταν ο παίκτης κερδίζει ή χάνει, ή όταν αποκτά κάποιο αντικείμενο θα πρέπει να ακούγεται ένας ήχος που να σηματοδοτεί αυτή την εξέλιξη.

Εμείς στο παιχνίδι μας θα εισάγουμε 4 ήχους οι οποίοι θα ακούγονται όταν:

- Ο παίκτης αποκτά μια ζωή ή ένα νόμισμα
- Ο παίκτης περνά σε επόμενη πίστα
- Ο παίκτης επιτίθεται με “Stone”
- Ο παίκτης ή το αντικείμενο “Skeleton” σκοτώνεται

? Πώς όμως μπορούμε να εισάγουμε ήχους μέσα στο παιχνίδι και πώς τους χρησιμοποιούμε;



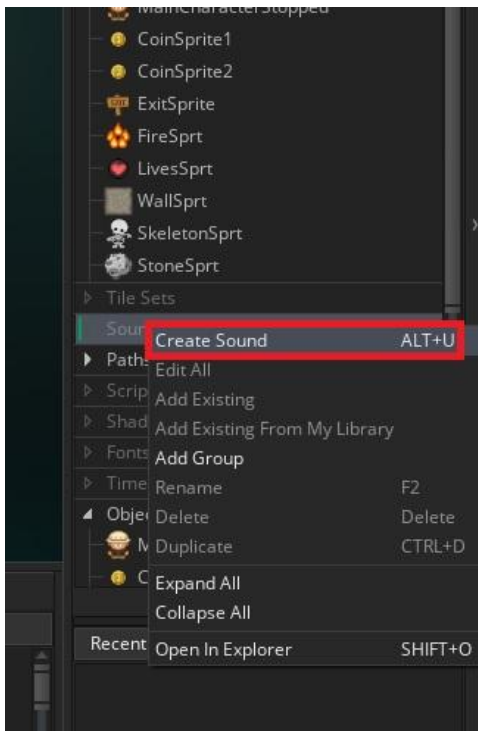
Τα Sounds είναι Resources που υπάρχουν στο Game Maker, όπως ακριβώς και τα Sprites, Objects, Paths και Rooms. Δημιουργούμε Sounds στην μπάρα Resources και εισάγουμε ενέργειες, όπως Play Audio για την αναπαραγωγή των ήχων αυτών στα Events των αντικειμένων. Όταν εκτελεστούν αυτές οι ενέργειες θα ακουστούν οι ήχοι που έχουμε επιλέξει.



Δημιούργησε ένα καινούργιο Sound με όνομα “item\_sound”.

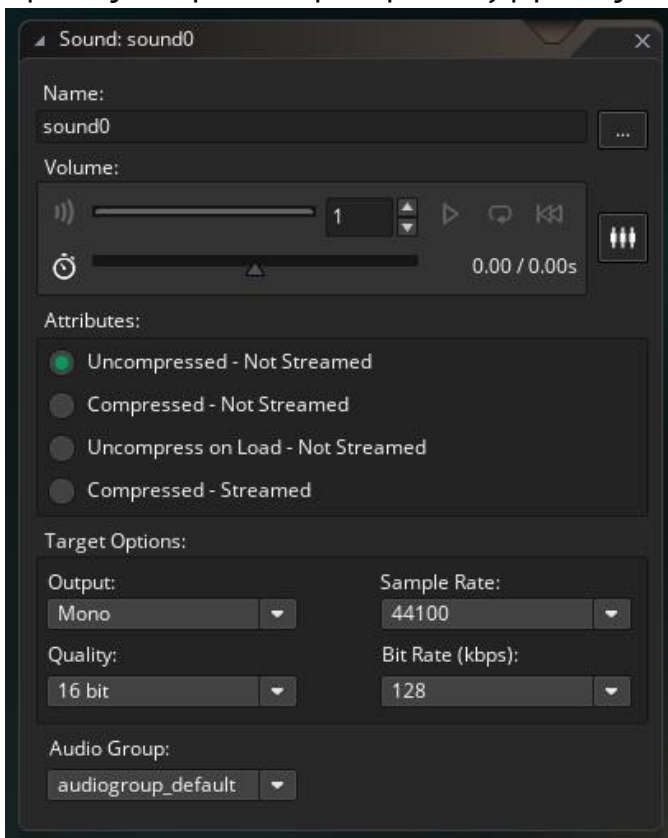
Ακολούθησε τα παρακάτω βήματα:

1. Κάνε δεξί κλικ στο **Sounds** της μπάρας **Resources**, όπως φαίνεται στην εικόνα 5.



Εικόνα 5. Δημιουργία Sound

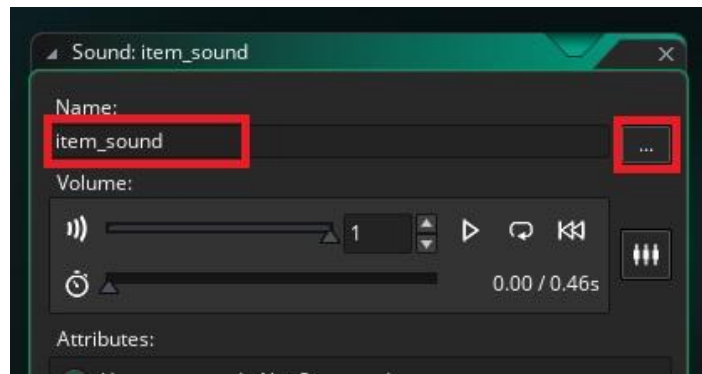
Αμέσως ανοίγει το παράθυρο επεξεργασίας **Sound** (εικόνα 6).



Εικόνα 6. Το παράθυρο επεξεργασίας Sound

2. Μέσα σε αυτό κάνε τις εξής αλλαγές, όπως φαίνεται στην εικόνα 7:
  - a. Άλλαξε το όνομα σε "item\_sound"

- b. Κάνε κλικ στο κουμπί με τις 3 τελείες και επίλεξε το αρχείο "item.wav" από τον φάκελο Resources
- c. Κλείσε το παράθυρο επεξεργασίας του **Sound**.



Εικόνα 7. Επιλογή ονόματος και αρχείου για το Sound "item\_sound"

Τώρα είμαστε έτοιμοι να εισάγουμε το **Sound** αυτό στο κώδικα των αντικειμένων που μας ενδιαφέρει.

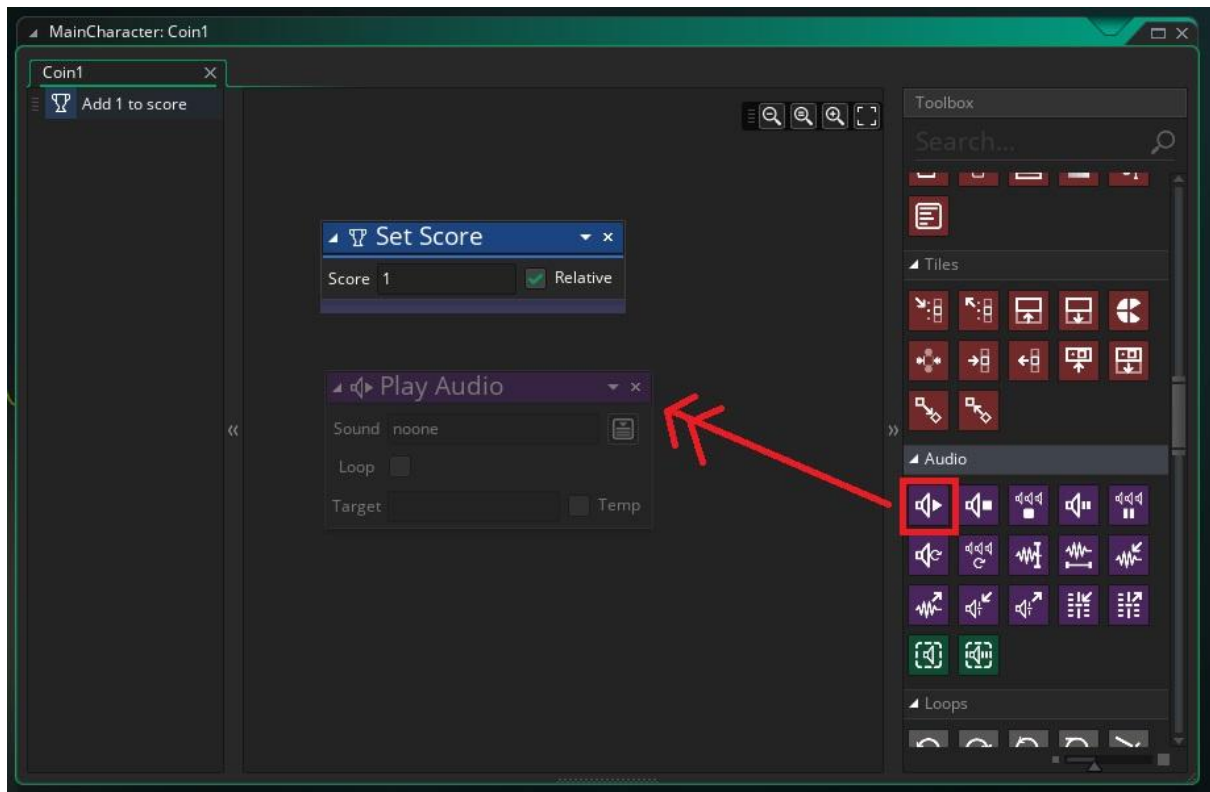
## GOAL

Τροποποίησε το object "MainCharacter" έτσι ώστε να ακούγεται ο ήχος "item\_sound" κάθε φορά που αγγίζει ένα από τα αντικείμενα "Coin1", "Coin2" και "Life".

Ακολούθησε τα παρακάτω βήματα:

1. Άνοιξε το **object** "MainCharacter", και μπες στο **Collision Event** με το "Coin1".
2. Σύρε την ενέργεια **Play Audio**, κάτω από την ήδη υπάρχουσα ενέργεια **Set Score**, όπως φαίνεται στην εικόνα 8.





Εικόνα 8. Εισαγωγή της ενέργειας Play Audio στο Collision Event του “MainCharacter” με το “Coin1”

3. Επίλεξε μέσα στο παραθυράκι της ενέργειας αυτής το **Sound** “item\_sound”, όπως φαίνεται στην εικόνα 9.



Εικόνα 9. Επιλογή του Sound “item\_sound”

Τώρα κάθε φορά που το “MainCharacter” θα αγγίζει το “Coin1” θα ακούγεται το sound “item\_sound”

4. Επανάλαβε τα βήματα 2 και 3 για τα **Collision Event** του “MainCharacter” με το “Coin2” και το “Life”, προκειμένου να ακούγεται ο ήχος “item\_sound” και στις περιπτώσεις των δύο αυτών **objects**.

Τέλεια! Το “MainCharacter” όταν αγγίζει ένα οποιοδήποτε item από τα παραπάνω μέσα στο παιχνίδι θα ακούγεται ο ήχος που επιλέξαμε.



Δημιούργησε ένα Sound με όνομα “win\_sound”, το οποίο θα ακούγεται, όταν ο παίκτης θα κερδίζει σε κάθε πίστα.

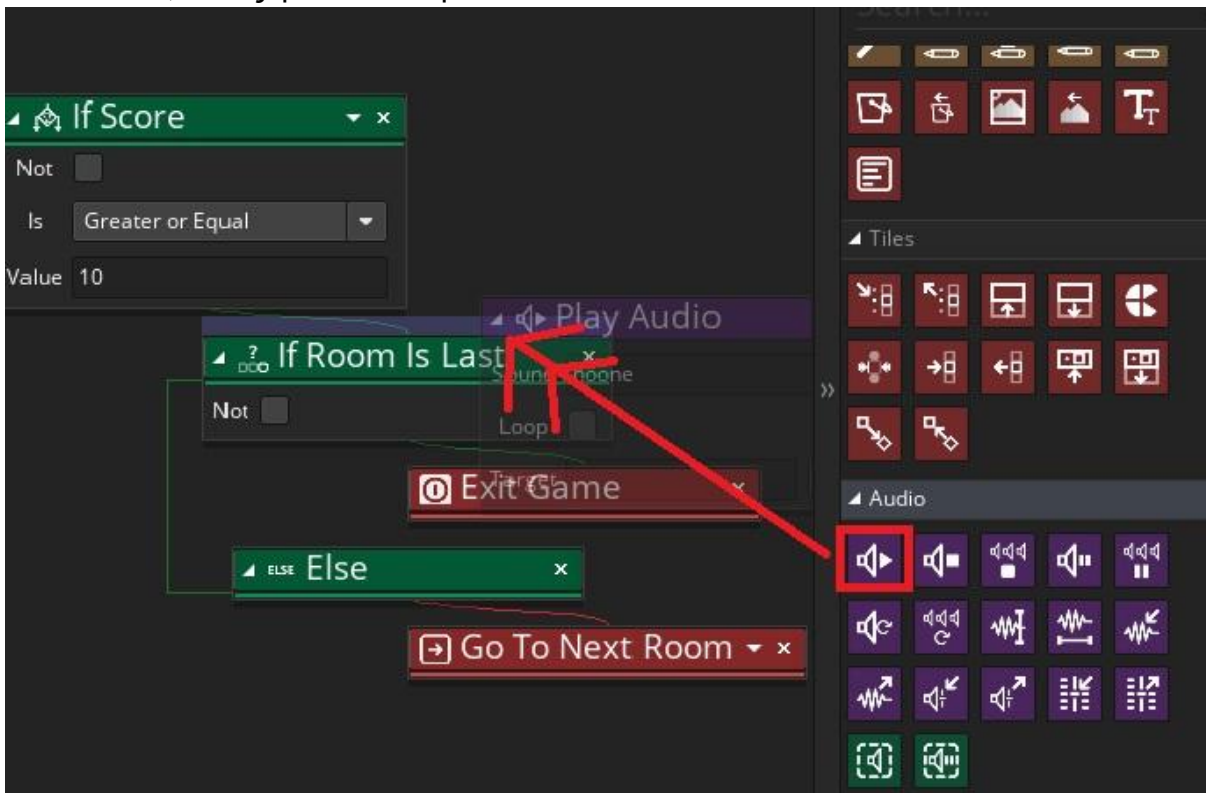
Ακολουθήσε τα παρακάτω βήματα:

1. Δημιούργησε ένα καινούργιο **sound** με όνομα “win\_sound” επιλέγοντας το αρχείο “win.wav” από τον φάκελο Resources.
2. Άνοιξε το **object** “MainCharacter” και μπες στο **Collision Event** με το “ExitObject”.

Εκεί υπάρχει ο κώδικας με τον οποίο επιλέγουμε αν θα τερματίσουμε το παιχνίδι ή αν θα πάμε στο επόμενο δωμάτιο, στην περίπτωση που η τιμή της μεταβλητής **Score** είναι μεγαλύτερη ή ίση του 10.

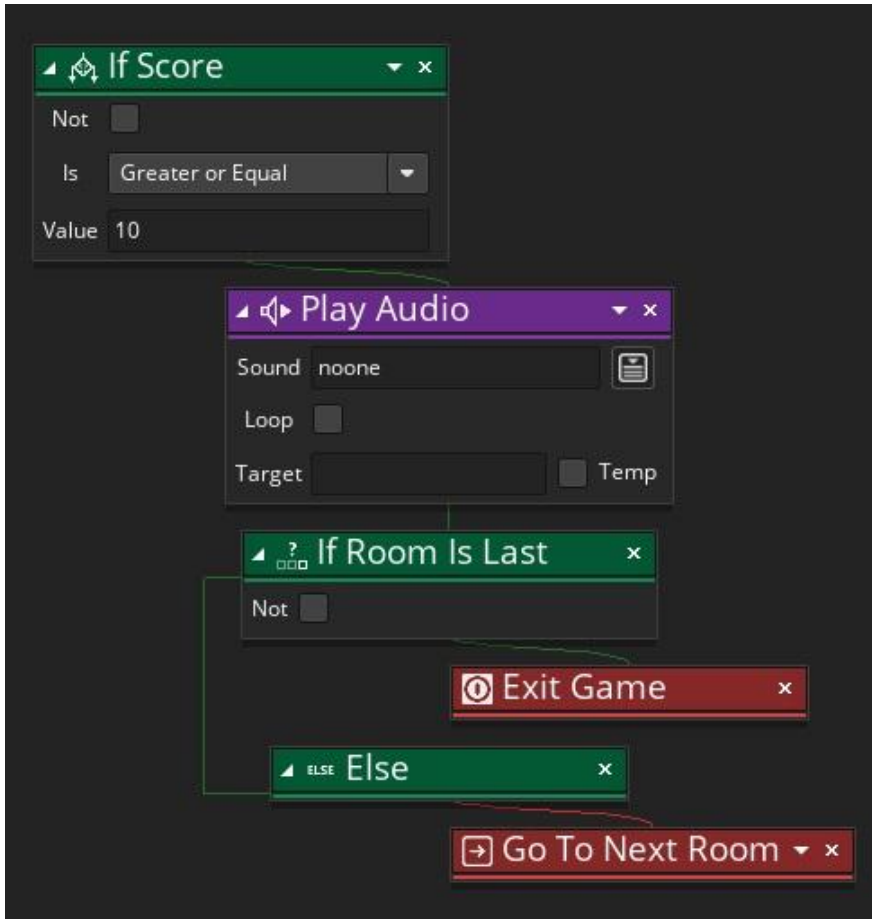
Εμείς θέλουμε να τοποθετήσουμε την ενέργεια **Play Audio** πριν από την εσωτερική δομή επιλογής, έτσι ώστε να ακούγεται το **Sound** και στις δύο περιπτώσεις. Πιο συγκεκριμένα:

3. Σύρε την ενέργεια **Play Audio** από την μπάρα ενεργειών μέσα στο σύνολο εντολών της ενέργειας **If Score** και ακριβώς πριν από την ενέργεια **If Room Is Last**, όπως φαίνεται στην εικόνα 10.



Εικόνα 10. Εισαγωγή ενέργειας Play Audio μέσα στο σύνολο εντολών της ενέργειας If Score

Το αποτέλεσμα αυτού του βήματος φαίνεται στην εικόνα 11.



Εικόνα 11. Η ενέργεια Play Audio μέσα στο σύνολο εντολών της If Score

4. Τώρα επέλεξε το **Sound** “win\_sound”, όπως φαίνεται στην εικόνα 12.



Εικόνα 12. Επιλογή του Sound “win\_sound”

5. Τέλος κλείσε το παράθυρο επεξεργασίας του **object** “MainCharacter”

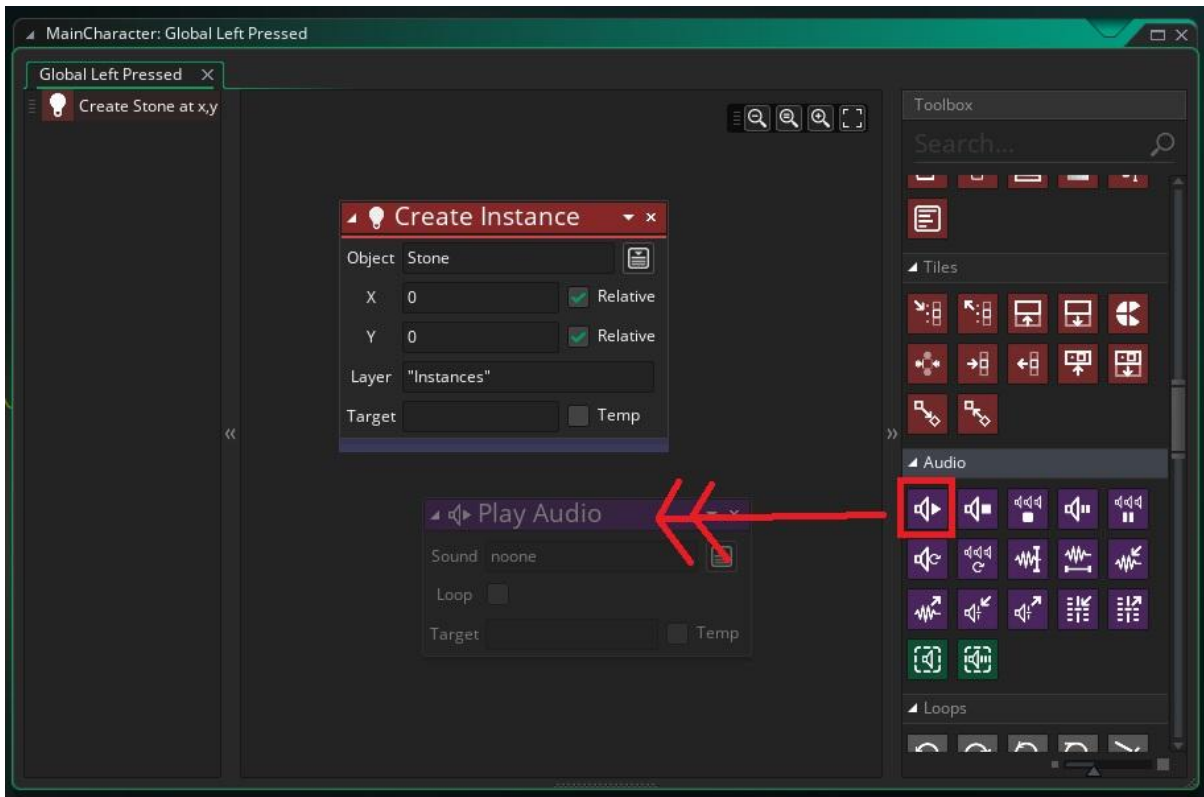
## GOAL

Δημιούργησε ένα Sound με όνομα “throw\_stone\_sound”, και κάντο να ακούγεται κάθε φορά που ο παίκτης κάνει κλικ στην οθόνη για να επιτεθεί με πέτρα.

Ακολούθησε τα παρακάτω βήματα:

1. Αρχικά δημιούργησε ένα καινούργιο **Sound** με όνομα “throw\_stone\_sound” εισάγοντας το αρχείο “throw\_stone.wav”.

2. Άνοιξε το **object** “MainCharacter” και μπες στο **Global Left Pressed Event** που υπάρχει ήδη ανάμεσα στα **Events** του object.
3. Σύρε την ενέργεια **Play Audio** από την μπάρα ενεργειών κάτω από την ενέργεια **Create Instance** που υπάρχει ήδη, όπως φαίνεται στην εικόνα 13.



Εικόνα 13. Εισαγωγή της ενέργειας Play Audio

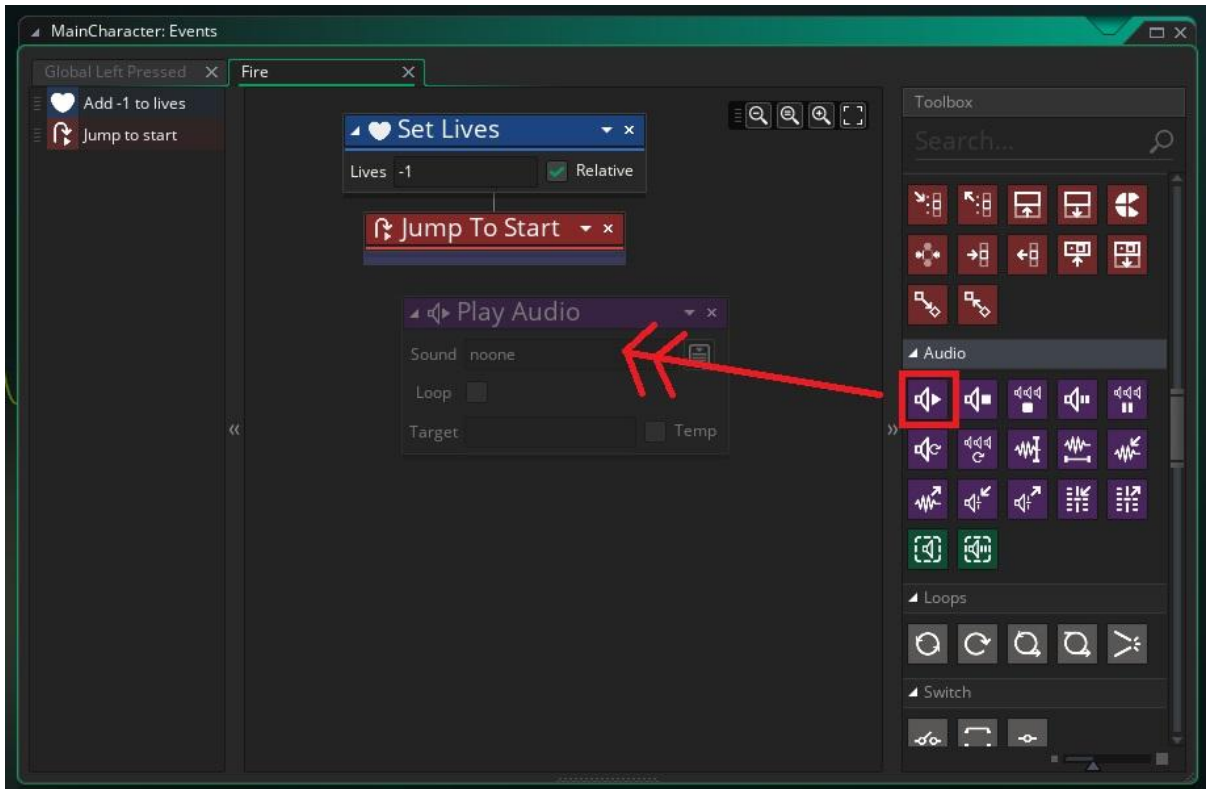
4. Στο παραθυράκι της ενέργειας επέλεξε το **sound** “throw\_stone\_sound”
5. Τέλος, κλείσε το παράθυρο επεξεργασίας του **Object**.

## GOAL

Δημιούργησε ένα τελευταίο Sound με όνομα “death\_sound”, και βάλε το να ακούγεται κάθε φορά που το “MainCharacter” αγγίζει το “Fire” ή το “Skeleton” και κάθε φορά που το “Skeleton” αγγίζει το “Stone”.

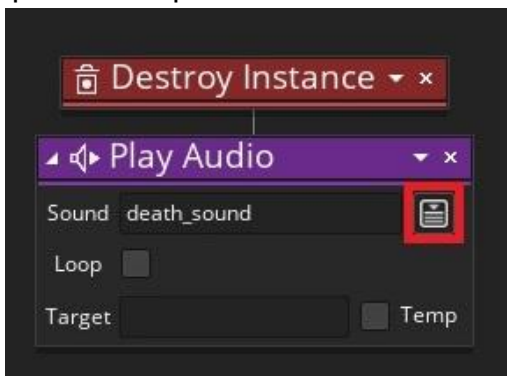
Ακολούθησε τα παρακάτω βήματα:

1. Δημιούργησε ένα καινούργιο **sound** με όνομα “death\_sound”, και επέλεξε το αρχείο “death.wav”
2. Τώρα, άνοιξε το **object** “MainCharacter” και μπες στο **Collision Event** με το “Fire”.
3. Στις ήδη υπάρχουσες ενέργειες σύρε την ενέργεια **Play Audio**, όπως φαίνεται στην εικόνα 14.



Εικόνα 14. Εισαγωγή της ενέργειας Play Audio

4. Στο παραθυράκι της ενέργειας **Play Audio** επέλεξε το **Sound** “death\_sound”
5. Τώρα κάνε το ίδιο και στο **Collision Event** με το **object** “Skeleton”
6. Κλείσε το παράθυρο επεξεργασίας του “MainCharacter”, άνοιξε αυτό του **object** “Skeleton” και μπες στο **Collision Event** με το **object** “Stone”.
7. Εισήγαγε την ενέργεια **Play Audio** και επέλεξε το **sound** “death\_sound”, όπως φαίνεται στην εικόνα 15.



Εικόνα 15. Επιλογή του Sound “death\_sound”

8. Τέλος κλείσε το παράθυρο επεξεργασίας του “Skeleton”  
Τώρα το παιχνίδι μας είναι εμπλουτισμένο με ήχους που θα ακούγονται και έτσι ο παίκτης θα το απολαμβάνει περισσότερο.

## ΒΗΜΑ ΔΕΥΤΕΡΟ: Δημιουργία αντιγράφου object Skeleton με κυκλικό Path

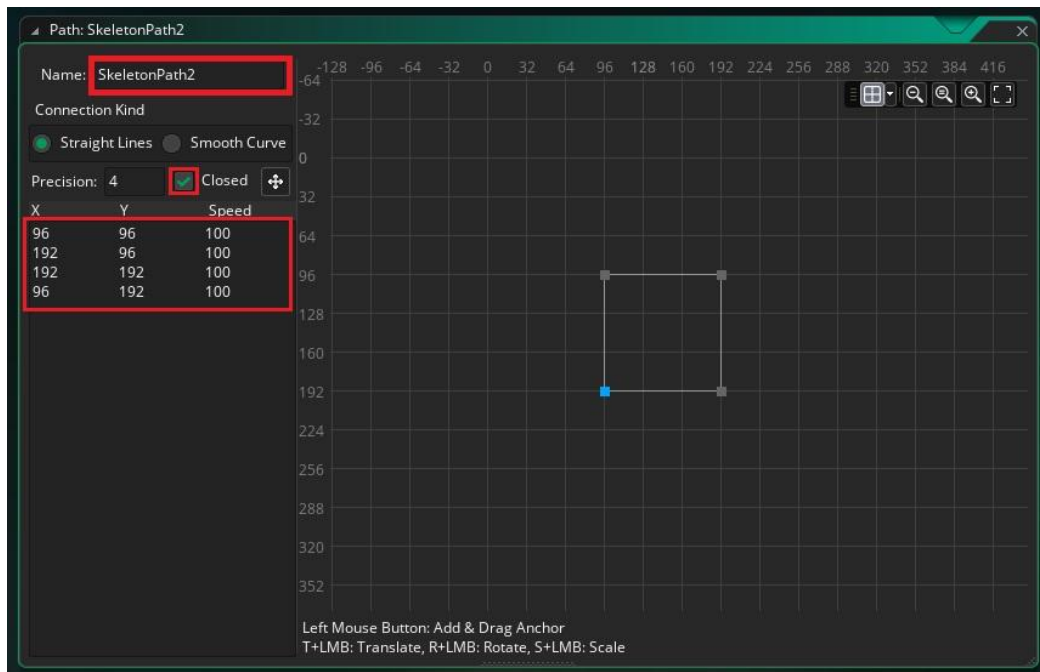
Σε αυτό το βήμα θα δημιουργήσουμε ένα καινούργιο **object** “Skeleton2” που θα έχει τις ίδιες λειτουργίες με το αρχικό, μέσω αντιγραφής, και θα του αλλάξουμε το **path** προκειμένου να κάνει κυκλική πορεία αντί για ευθεία γραμμή.



Δημιούργησε ένα καινούργιο **path** με όνομα “SkeletonPath2”, το οποίο θα σχηματίζει ένα τετράγωνο.

Ακολούθησε τα παρακάτω βήματα:

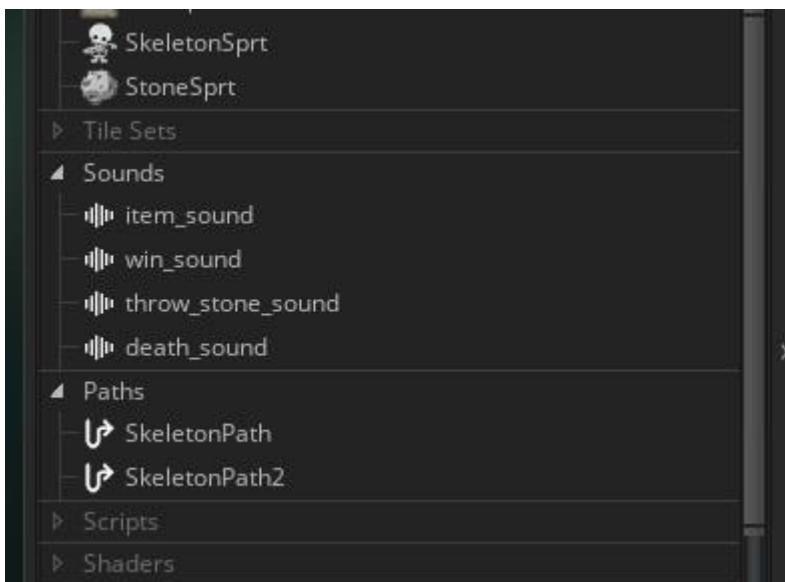
1. Δημιούργησε ένα καινούργιο **path** με δεξί κλικ στα **Paths** της μπάρας πόρων.
2. Στο παράθυρο που ανοίγει γράψε το όνομα “SkeletonPath2” και επίλεξε στο πλέγμα τα παρακάτω τέσσερα σημεία: (96,96), (192,96), (192,192), (96,192), έτσι ώστε το **object** που θα ακολουθήσει το **path** αυτό να σχηματίσει τετράγωνο. Οι αλλαγές στο παράθυρο επεξεργασίας του **path** φαίνονται στην παρακάτω εικόνα 16.



Εικόνα 16. Το path “SkeletonPath2”

3. Κλείσε το παράθυρο του “SkeletonPath2”

Το αποτέλεσμα της δημιουργίας του **path** καθώς και των **sounds** πρέπει σε αυτό το σημείο να είναι σύμφωνο με την εικόνα 17.



Εικόνα 17. Τα sounds και τα path στην μπάρα Resources

Τώρα είμαστε έτοιμοι να δημιουργήσουμε το **object** “Skeleton2” το οποίο θα ακολουθεί αυτό το **path**.

? Πώς πιστεύεις ότι μπορώ να δημιουργήσω ένα καινούργιο object με τις ίδιες ιδιότητες βάσει ενός υπάρχοντος;



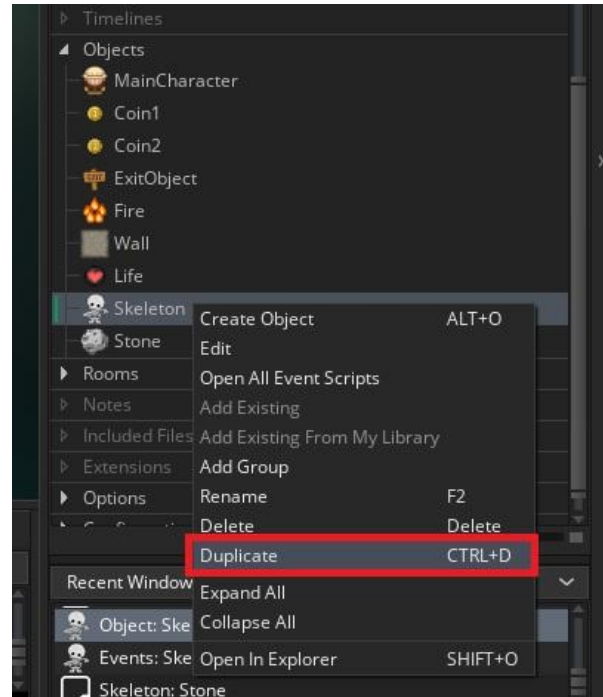
Η επιλογή Duplicate μας δίνει την δυνατότητα να αντιγράψουμε ένα ήδη υπάρχον object και να δημιουργηθεί ένα καινούργιο. Το καινούργιο θα έχει ακριβώς την ίδια λειτουργία με το αρχικό, δηλαδή το sprite, τα events καθώς και οι λειτουργίες του θα είναι οι ίδιες. Αυτή η επιλογή μας είναι χρήσιμη στην περίπτωση που θέλουμε να δημιουργήσουμε ένα ίδιο object και να το επεξεργαστούμε για να το διαφοροποιήσουμε ή να το επεκτείνουμε.

## GOAL

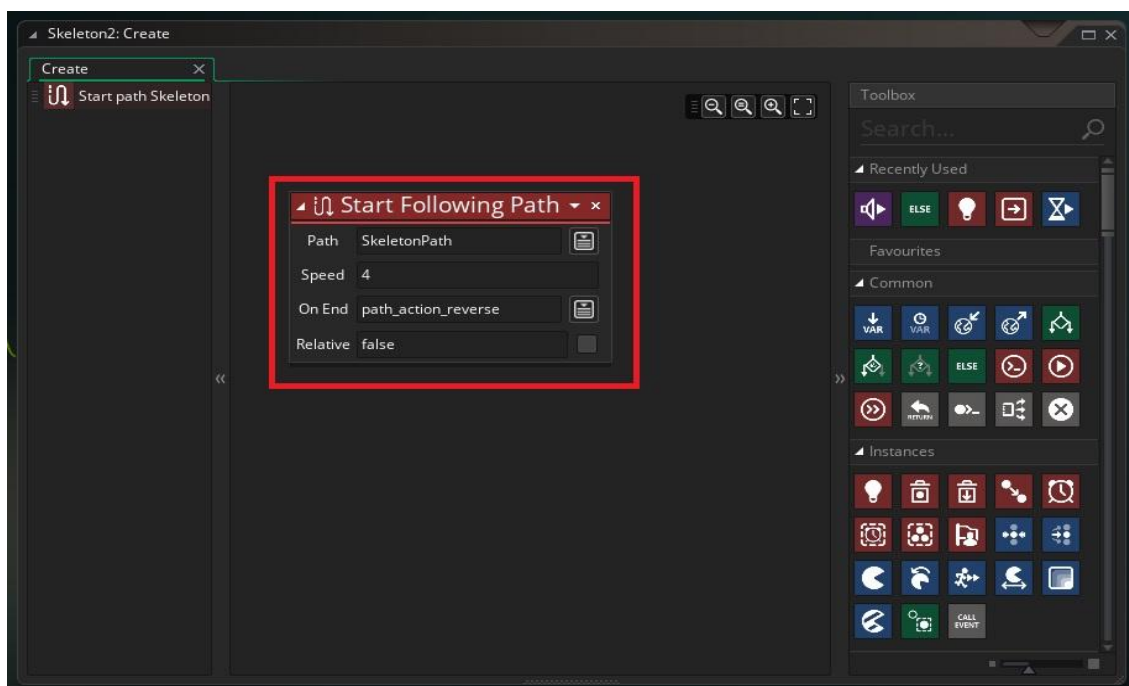
Δημιούργησε ένα αντικείμενο με όνομα “Skeleton2”, κάνοντας απλώς duplicate και άλλαξε τον τρόπο που θα κινείται το Path του.

Ακολούθησε τα παρακάτω βήματα:

1. Κάνε δεξί κλικ στο “SkeletonPath”, που βρίσκεται στην μπάρα **Resources**, και επίλεξε **Duplicate**, όπως φαίνεται στην εικόνα 18.
2. Στο παράθυρο επεξεργασίας του καινούργιου **object**, άλλαξε το όνομα και κάνε το “Skeleton2” και μπες στο **Create Event**. Παρατηρούμε ότι το καινούργιο object έχει τις ίδιες ιδιότητες με το αρχικό, δηλαδή υπάρχει ήδη η ενέργεια **Start Following Path**, όπως φαίνεται στην εικόνα 19.



Εικόνα 18. Αντιγραφή του object “Skeleton”



Εικόνα 19. Το ήδη υπάρχον Create Event του object “Skeleton2”

3. Άλλαξε τις παραμέτρους της **Start Following Path**, ως εξής:
  - a. Επίλεξε για **Path** το “SkeletonPath2”
  - b. Στο πεδίο **On End** επέλεξε **Loop**, στο μενού που θα εμφανιστεί με το πάτημα του κουμπιού, προκειμένου να εμφανιστεί path action continue, όπως φαίνεται στην εικόνα 20.





Εικόνα 20. Επιλογή παραμέτρων ενέργειας Start Following Path

4. Κλείσε το παράθυρο επεξεργασίας του **object** “Skeleton2”.

Τέλεια. Τώρα έχουμε φτιάξει ένα **object** “Skeleton2” με την ίδια λειτουργία με το “Skeleton”, με τη μόνη διαφορά ότι ακολουθεί διαφορετικό **Path**. Είναι η ώρα λοιπόν για έναν μικρό έλεγχο.

## ΒΗΜΑ ΤΡΙΤΟ: Έλεγχος

Ας ελέγξουμε εάν οι αλλαγές λειτουργούν ....

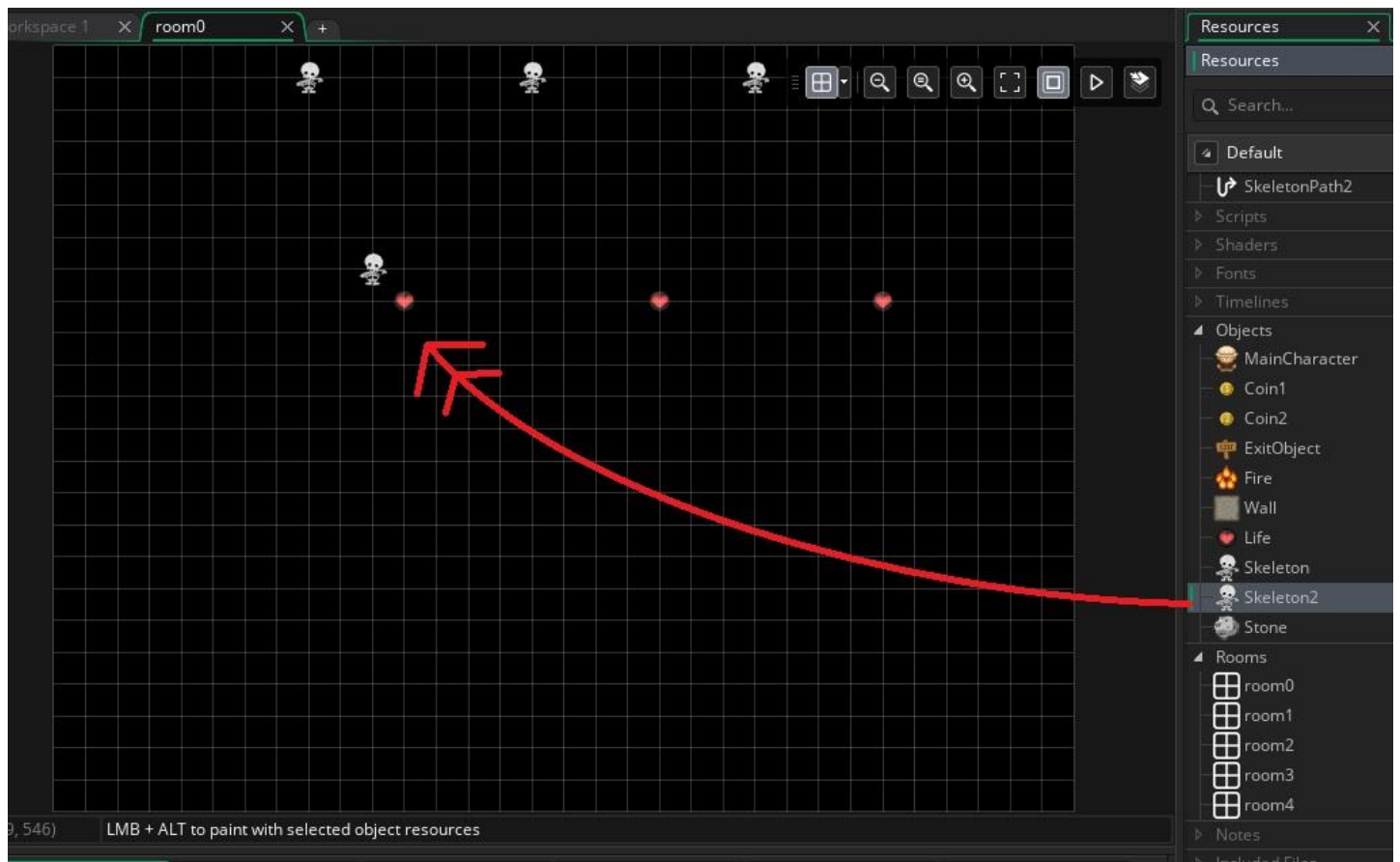
Για να το κάνουμε αυτό πρώτα πρέπει να τοποθετήσουμε στο ήδη υπάρχον **Room0** το **object** “Skeleton2” που μόλις δημιουργήσαμε.

Άνοιξε λοιπόν το **Room0** και τοποθέτησε μέσα το **object** αυτό όπως φαίνεται στην εικόνα 21.

Μετά πάτησε το κουμπί **Play** για να δοκιμάσεις το παιχνίδι.

Πρέπει παίζοντας να ελέγξεις τα εξής πράγματα:

1. Τα 4 **Sounds** (item\_sound, win\_sound, throw\_sound και death\_sound)
2. Το **path** που θα ακολουθεί το **object** “Skeleton2”



Εικόνα 21. Τοποθέτηση object “Skeleton2” στο Room0

## ΒΗΜΑ ΤΕΤΑΡΤΟ: Timeline

? Πώς μπορούμε να προγραμματίσουμε το παιχνίδι έτσι ώστε να εμφανίζονται objects κατά τη διάρκεια αυτού και όχι στην αρχή ή μέσω κάποιου event;



Το Timeline είναι μία δυνατότητα που μας δίνει το Game Maker προκειμένου να γίνονται ενέργειες κατά τη διάρκεια του παιχνιδιού σε προκαθορισμένο χρόνο. Ένα Timeline μοιάζει με μία σειρά εντολών που γίνονται σε συγκεκριμένες χρονικές στιγμές από την στιγμή που ξεκινάει.

Για παράδειγμα:

- Την στιγμή 0, εκτέλεσε την ενέργεια Α
- Την στιγμή 10, εκτέλεσε την ενέργεια Β
- Την στιγμή 50, εκτέλεσε την ενέργεια Γ

Μας είναι ιδιαίτερα χρήσιμο καθώς είμαστε αναγκασμένοι από τις απαιτήσεις των παιχνιδιών που φτιάχνουμε να δημιουργούνται objects, να ακούγονται sounds, ή να αλλάζουν οι μεταβλητές μας σε συγκεκριμένες χρονικές στιγμές. Ο χρόνος του Timeline μετριέται σε Steps.

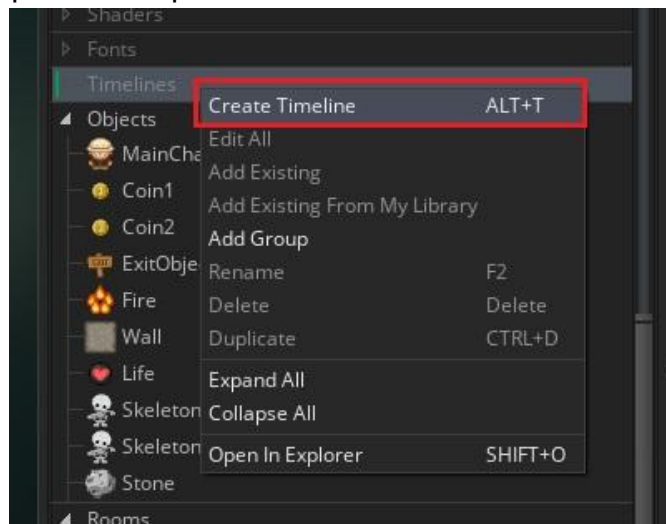
## GOAL

Δημιούργησε ένα Timeline με όνομα “coins\_timeline”, και προγραμμάτισέ το να δημιουργεί σε τυχαίες θέσεις

- 4 objects “Coin1” την στιγμή 300 και
- 3 objects “Coin2” την στιγμή 600

Ακολουθήσε τα παρακάτω βήματα:

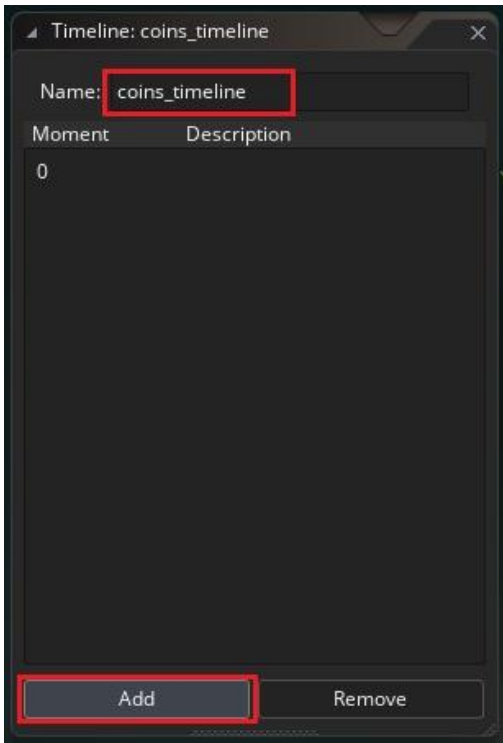
1. Αρχικά κάνε δεξί κλικ στο **Timelines** της μπάρας **Resources** και επίλεξε **Create Timeline**, όπως φαίνεται στην εικόνα 22.



Εικόνα 22. Δημιουργία νέου Timeline

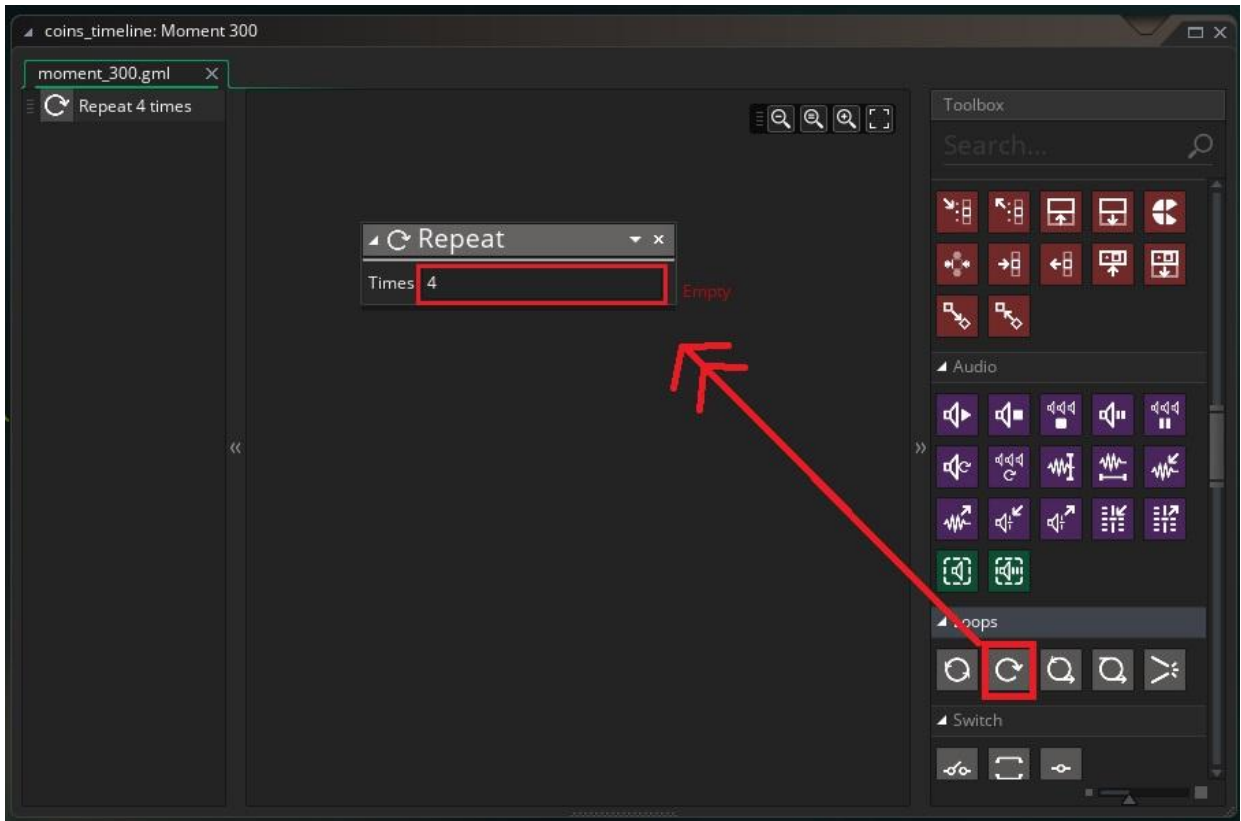
Αμέσως ανοίγει το παράθυρο επεξεργασίας του **Timeline**.

2. Τώρα άλλαξε το όνομά του σε “coins\_timeline” και επίλεξε **Add**, όπως φαίνεται στην εικόνα 23, προκειμένου να προσθέσουμε μία στιγμή στην οποία θα μπου οι ενέργειες που θέλουμε.



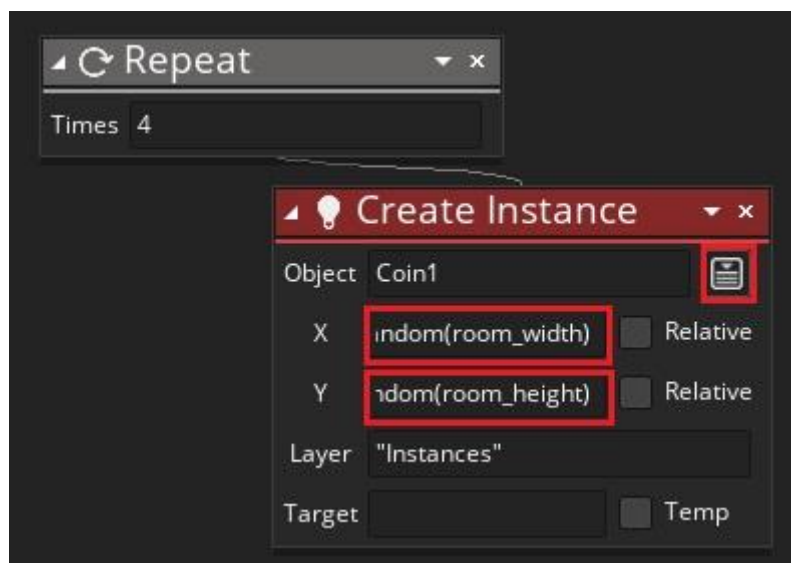
Εικόνα 23. Επιλογή ονόματος και προσθήκη χρονικής στιγμής

3. Αμέσως εμφανίζεται η στιγμή 0. Κάνε δεξί κλικ στο 0, επέλεξε **Change Moment**, και άλλαξε το 0 σε 300.
4. Κάνε διπλό κλικ πάνω στο 300 για να ανοίξει το παράθυρο που θα εισάγουμε τις ενέργειες που θέλουμε.
5. Σύρε την ενέργεια **Repeat** και καθόρισε αριθμό επαναλήψεων 4, όπως φαίνεται στην εικόνα 24, έτσι ώστε η εντολή δημιουργίας του **object** που θα βάλουμε να εκτελεστεί 4 φορές.



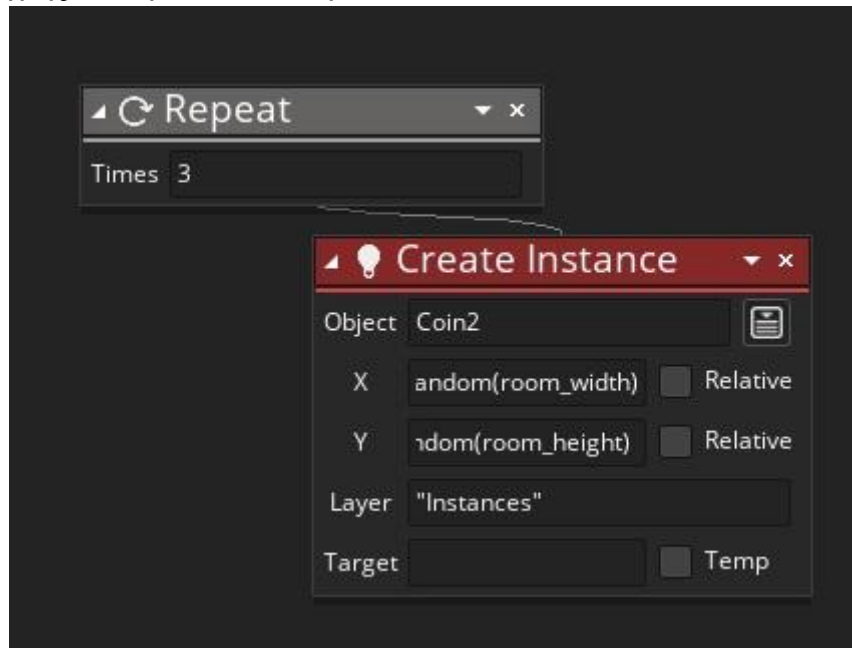
Εικόνα 24. Εισαγωγή ενέργειας Repeat στη στιγμή 300 του Timeline

6. Τώρα σύρε μέσα στο σύνολο εντολών της **Repeat** (εμφωλευμένα), την ενέργεια **Create Instance** και καθόρισε να δημιουργεί **object** "Coin1" σε τυχαία θέση μέσα στο δωμάτιο. Αυτό γίνεται χρησιμοποιώντας τη συνάρτηση **random** καθώς και τις μεταβλητές συστήματος **room\_width** και **room\_height**, ως εξής: για X όρισε  $random(room\_width)$  και για Y  $random(room\_height)$ , όπως φαίνεται στην εικόνα 25.



Εικόνα 25. Καθορισμός παραμέτρων ενέργειας Create Instance της στιγμής 300

7. Τώρα δημιούργησε μία ακόμα στιγμή, αλλάξέ την σε 600 και επανάλαβε τα βήματα 4 έως 6 προκειμένου να δημιουργήσεις 3 **objects** “Coin2”. Οι ενέργειες της στιγμής 600 φαίνονται στην εικόνα 26.



Εικόνα 26. Καθορισμός παραμέτρων ενέργειας Create Instance της στιγμής 600

8. Τέλος κλείσε το παράθυρο επεξεργασίας του **Timeline**.

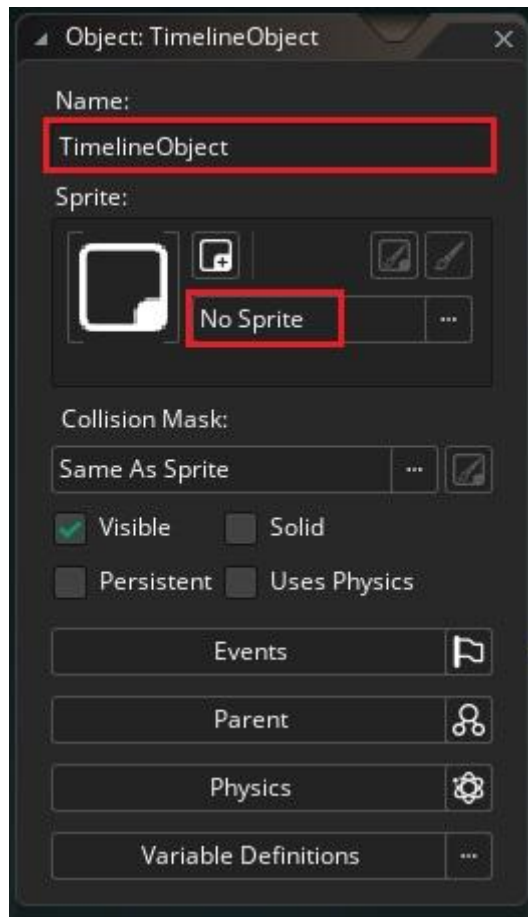
? Πώς όμως θα καλέσουμε το Timeline που μόλις δημιουργήσαμε προκειμένου να εκτελεστούν οι ενέργειες που καθορίσαμε;

## GOAL

Δημιούργησε ένα Object με όνομα “Timeline\_object” χωρίς Sprite, και προγραμμάτισέ το έτσι ώστε όταν δημιουργείται να καλεί το Timeline “coins\_timeline”.

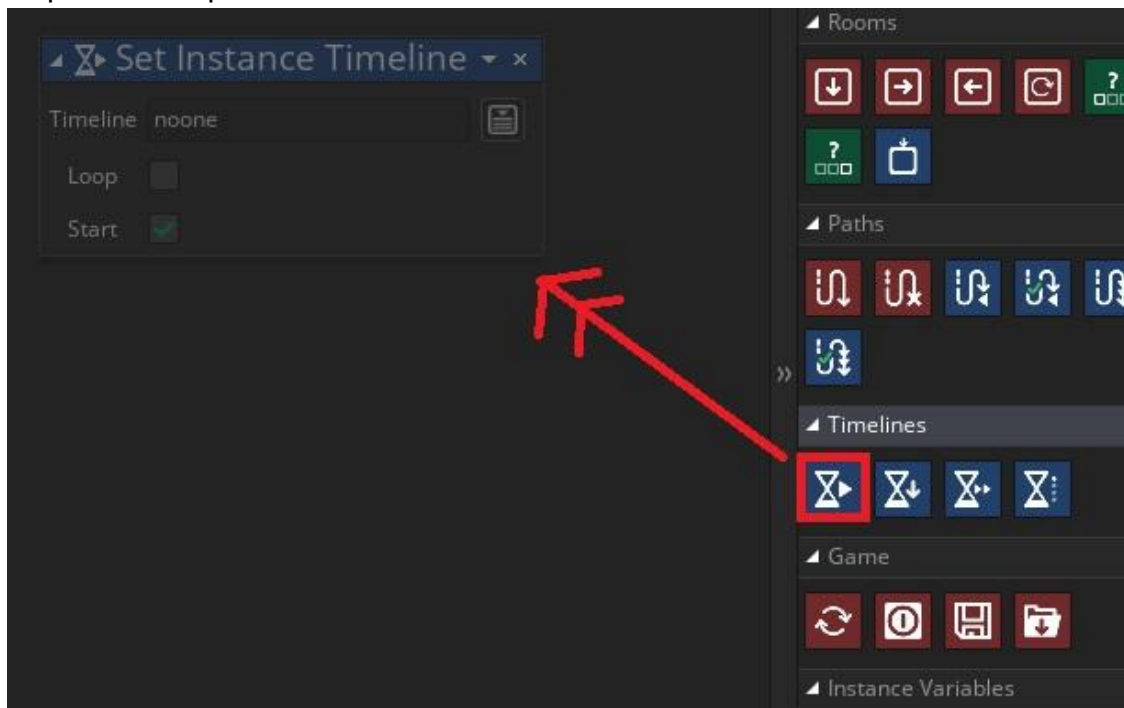
Ακολούθησε τα παρακάτω βήματα:

1. Αρχικά δημιούργησε ένα καινούργιο **object** με όνομα “TimelineObject” χωρίς να του δώσεις κάποιο **sprite**, όπως φαίνεται στην εικόνα 27.



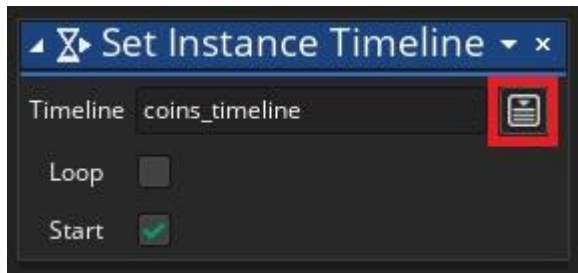
Εικόνα 27. Δημιουργία του TimelineObject

2. Άνοιξε το **Create Event** και σύρε την ενέργεια **Set Instance Timeline**, όπως φαίνεται στην εικόνα 28.



Εικόνα 28. Εισαγωγή της ενέργειας Set Instance Timeline

3. Τώρα, καθόρισε ποιο **Timeline** θα καλέσει το **object**, δηλαδή το “coins\_timeline” που έχουμε δημιουργήσει, κάνοντας κλικ στο κουμπί που φαίνεται στην εικόνα 29.



Εικόνα 29. Καθορισμός Timeline

4. Τέλος, κλείσε το παράθυρο επεξεργασίας του **object** “TimelineObject”.

Τέλεια! Τώρα όταν τοποθετήσουμε το **object** “TimelineObject” σε κάποιο **room**, αυτόματα την στιγμή 300 θα δημιουργηθούν 4 “Coin1” και την στιγμή 600, 3 “Coin2”, σε τυχαίες θέσεις μέσα στο **room**.

Στο επόμενο βήμα θα επεξεργαστούμε το **room** έτσι ώστε να τοποθετηθεί υπόβαθρο (background).

## ΒΗΜΑ ΠΕΜΠΤΟ: Background και Δημιουργία Room με το Object “TimelineObject”

? Πώς πιστεύεις ότι μπορούμε να τοποθετήσουμε φόντο, δηλαδή Background σε ένα Room;



Το Tile, όπως λέει και το όνομά του, σημαίνει πλακίδιο. Όταν ορίσουμε σε ένα Room, Background με Tile, τότε επιλέγουμε ένα μικρό Sprite και αυτό θα καλύψει όλη την οθόνη του παιχνιδιού οριζόντια και κάθετα. Πιο συγκεκριμένα η εικόνα του Sprite που θα επιλέξουμε, θα εμφανίζεται επαναλαμβανόμενα ξεκινώντας από την πάνω αριστερή γωνία και προσπαθώντας να καλύψει όλη την οθόνη. Με αυτόν τον τρόπο γεμίζει το μαύρο φόντο του παιχνιδιού.

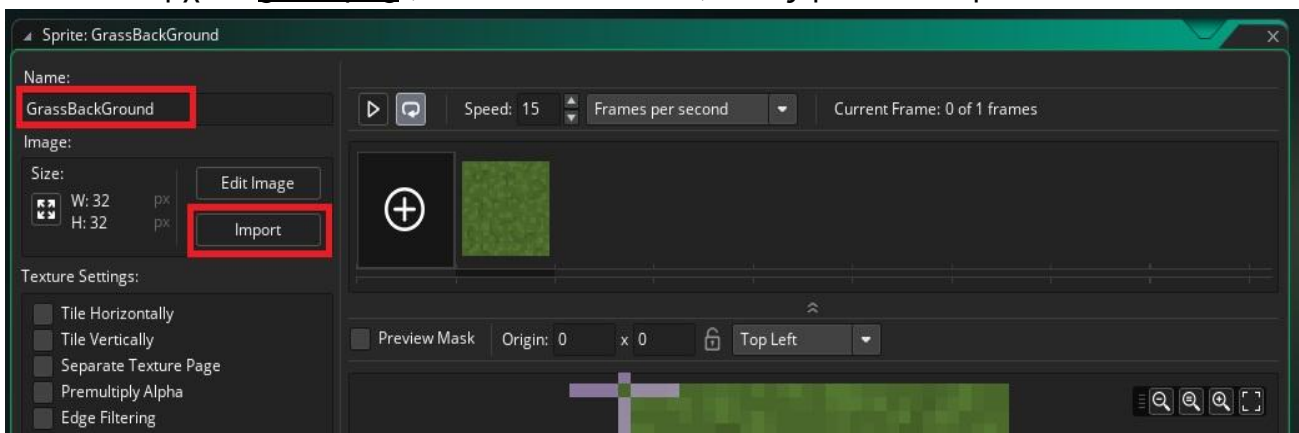




Δημιούργησε ένα Sprite με όνομα “GrassBackGround”, και καθόρισε να εμφανίζεται ως Tile, σε όλα τα Rooms.

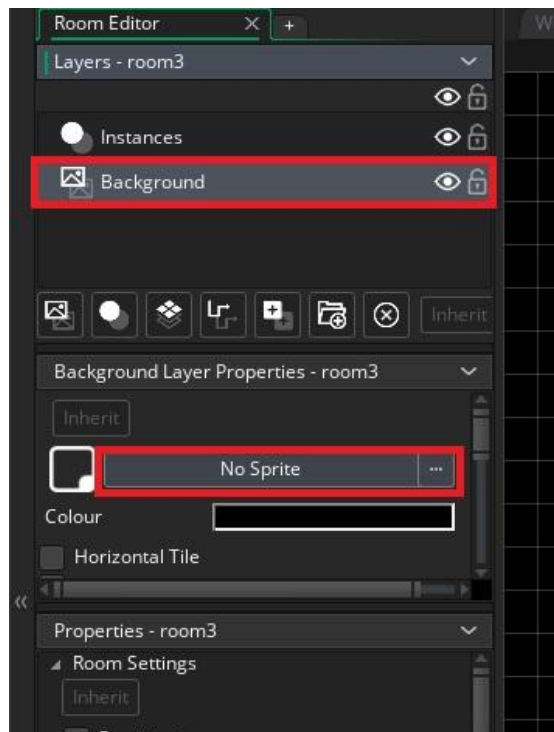
Ακολούθησε τα παρακάτω βήματα:

1. Αρχικά δημιούργησε ένα **Sprite** με όνομα “GrassBackGround” επιλέγοντας το αρχείο “*grass.png*”, από τα **Resources**, όπως φαίνεται στην εικόνα 30.



Εικόνα 30. Δημιουργία Sprite “GrassBackGround”

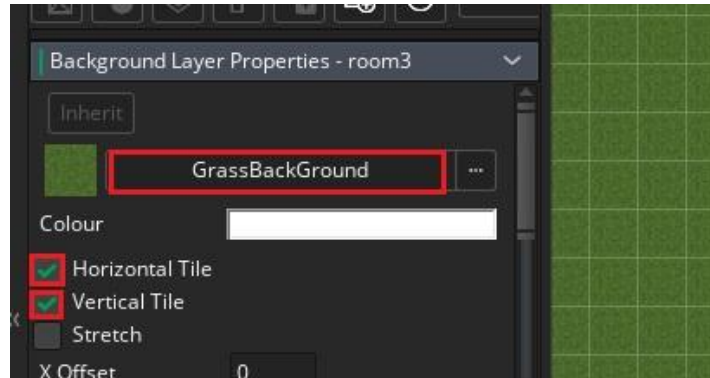
2. Άνοιξε το **Room0**, και επέλεξε από τα αριστερά της οθόνης **Background** και έπειτα **No Sprite**, για να επιλέξεις το **Sprite** που μόλις δημιούργησες, όπως φαίνεται στην εικόνα 31.



Εικόνα 31. Επιλογή Background

3. Αφού επιλέξεις το “GrassBackGround” για **Sprite**, τσέκαρε τις δύο επιλογές **Horizontal Tile** και **Vertical Tile**, όπως φαίνεται στην εικόνα 32, για να

καθορίσεις ότι το **Tile** αυτό θα τοποθετηθεί επαναληπτικά και οριζοντίως και καθέτως.



Εικόνα 32. Καθορισμός εμφάνισης του Tile οριζόντια και κάθετα

4. Τώρα κάνε κλικ στο **Instances** που βρίσκεται ακριβώς πάνω από την επιλογή **Background** πάνω αριστερά στην οθόνη και κλείσε το παράθυρο επεξεργασίας του **Room**.

Επανάλαβε τα βήματα 2 έως 4 για να καθορίσεις το **Sprite** ως **Background Tile** σε όλα τα **Rooms**.

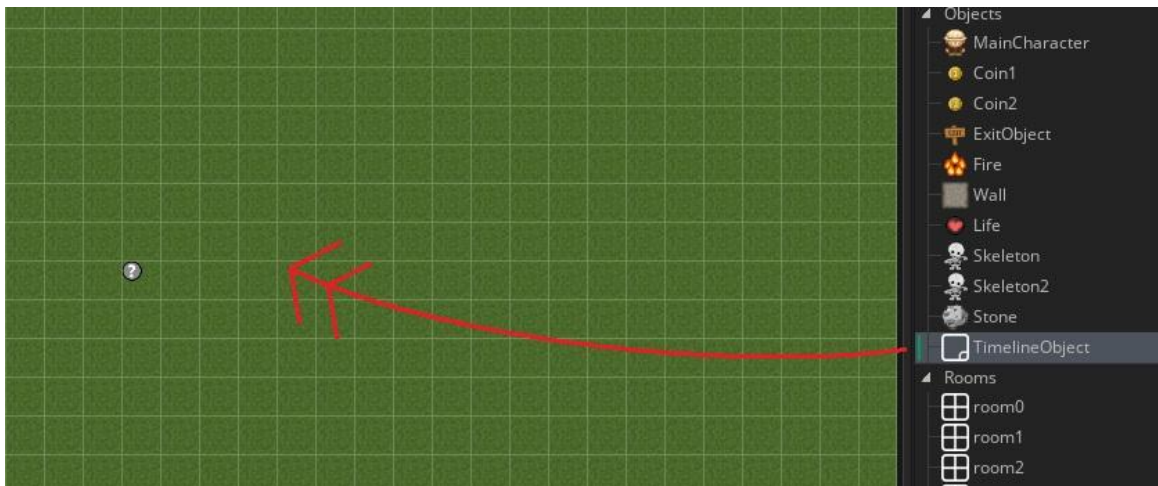
## GOAL

Δημιούργησε ένα τελευταίο **Room** και τοποθέτησε μέσα σε αυτό το **Object** “**TimelineObject**” καθώς και όλα τα αντικείμενα που χρειάζονται όπως εσύ επιθυμείς.

### ΠΡΟΣΟΧΗ!

Δεν πρέπει να τοποθετήσεις μέσα σε αυτό το **Room**, **Objects** “**Coin1**” και “**Coin2**” καθώς αυτά θα δημιουργηθούν αυτόματα με το **Timeline**. Αρκεί μόνο η εισαγωγή του “**TimelineObject**”.

Επίσης επειδή το συγκεκριμένο **Object** δεν έχει **Sprite**, δεν θα εμφανίζεται κατά τη διάρκεια εκτέλεσης του παιχνιδιού. Όμως μέσα στο παράθυρο επεξεργασίας του **Room**, θα εμφανίζεται με το “?”, όπως φαίνεται στην εικόνα 33.



Εικόνα 33. Εισαγωγή του Object “TimelineObject” στο Room4

Τέλεια! Τώρα είμαστε έτοιμοι να παίξουμε το παιχνίδι.  
Κάνε κλικ στο κουμπί **Play**.

## Καλή Διασκέδαση!



- Τι είναι τα Sounds και σε τι μας είναι χρήσιμα;
- Με ποια ενέργεια ακούγεται ένα Sound μέσα σε ένα Event;
- Πώς μπορούμε να προγραμματίσουμε ένα αντικείμενο έτσι ώστε αυτό να κινείται σε σχήμα πενταγώνου;
- Σε τι μας χρησιμεύει το Timeline και πώς το καλούμε από ένα Event;
- Από πού επιλέγουμε εικόνες για να τοποθετήσουμε ως Background σε ένα Room;



- Επέκτεινε το πρόγραμμα έτσι ώστε να ακούγεται ο ήχος “throw\_stone\_sound”, που έχεις φτιάξει, και στην περίπτωση που το object “MainCharacter”, αγγίζει το “Wall”
- Καθόρισε ακόμα μία στιγμή (900) στο Timeline “coins\_timeline”, έτσι ώστε να δημιουργούνται 3 objects “Fire” σε τυχαίες θέσεις στην οθόνη.
- Κάνε τα objects “Fire” να κινούνται σε τετραγωνικό σχήμα.
- Άλλαξε το Background του Room0 εισάγοντας μία δική σου εικόνα στο Game Maker.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6 GAME MAKER

Τελειοποιούμε το Παιχνίδι μας με Start και Exit Screen



### Έννοιες Προγραμματισμού Παιχνιδιών: Δωμάτιο Εκκίνησης και Τερματισμού

#### ? Τι έχουμε φτιάξει ως τώρα;

Στη 5<sup>η</sup> Δραστηριότητα επεκτείναμε το παιχνίδι μας έτσι ώστε:

- Να ακούγονται κάποια **Sounds**, κάθε φορά που ο παίκτης αποκτά ένα item, χάνει μια ζωή, πετάει μία πέτρα ή φτάνει στο στόχο της πίστας
- Να υπάρχει δεύτερο **Object** “Skeleton2” με βάση το αρχικό, του οποίου η κίνηση να σχηματίζει τετράγωνο
- Να δημιουργούνται **Objects** “Coin1” και “Coin2”, κατά τη διάρκεια του παιχνιδιού και όχι στην αρχή, μέσω **Timeline**
- Να υπάρχει **Background** (υπόβαθρο), στα **Rooms**, δηλαδή στις πίστες.

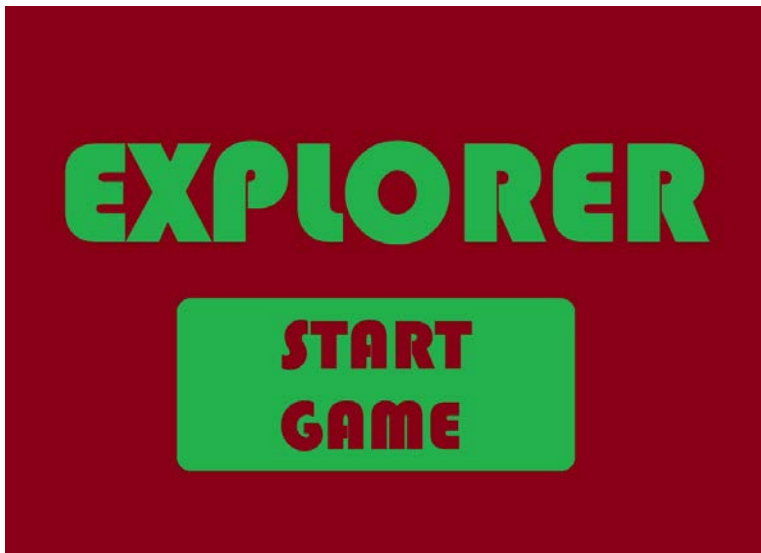
#### ? Τι θα φτιάξουμε;

Σήμερα θα τελειοποιήσουμε το παιχνίδι μας δημιουργώντας *StartScreen* και *ExitScreen*.

- Το *StartScreen* μπαίνει στην αρχή του παιχνιδιού και έχει το λογότυπο και ένα κουμπί εκκίνησης.
- Το *ExitScreen* μπαίνει στο τέλος του παιχνιδιού μέσα στο οποίο αναγράφεται “YOU WIN” και υπάρχει κουμπί εξόδου από το παιχνίδι.

Για να το κάνουμε αυτό θα δημιουργήσουμε 4 καινούργια **Sprite** εκ των οποίων τα δυο θα χρησιμοποιηθούν για τα **Backgrounds** και τα άλλα δύο για τα **Objects** κουμπιά που θα επιτελούν την παραπάνω λειτουργία.

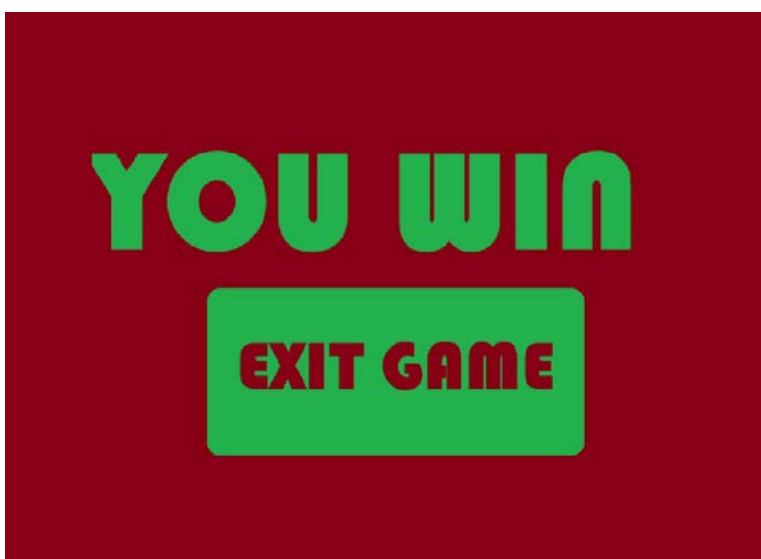
Η λειτουργία των *StartScreen* και *ExitScreen*, φαίνεται στις εικόνες 1, 2 και 3:



Εικόνα 1. Η αρχική οθόνη (StartScreen)

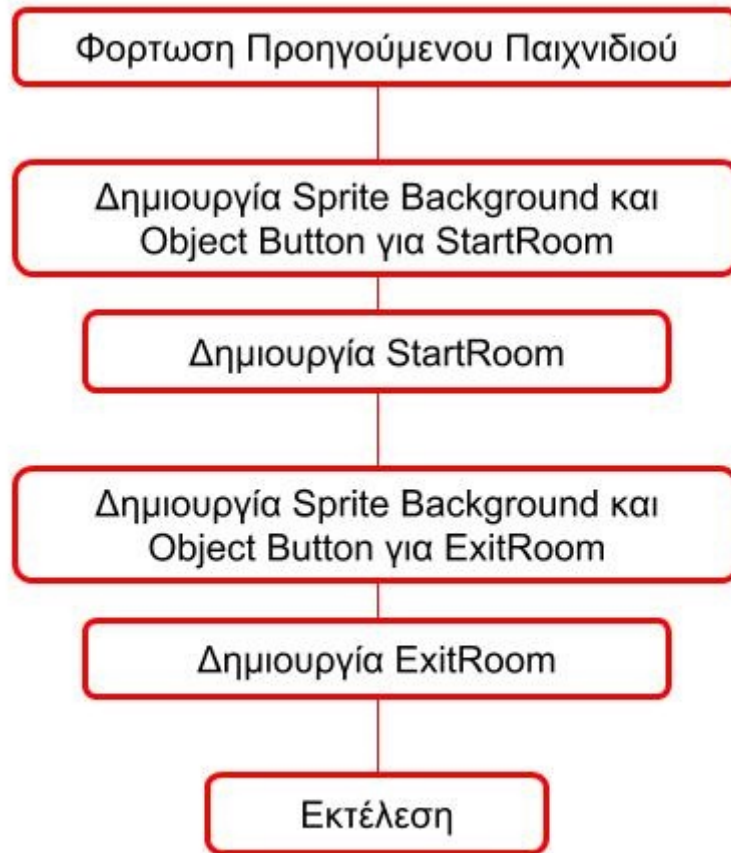


Εικόνα 2. Κατα τη διάρκεια του παιχνιδιού (ενδιάμεσα Rooms)



Εικόνα 3. Τελική οθόνη παιχνιδιού (EndRoom)

## Διάγραμμα Ροής Δραστηριότητας



Ας ξεκινήσουμε λοιπόν!

Εκτέλεσε το **Game Maker**, στην αρχική οθόνη του προγράμματος, πάτησε **Open**, στο παράθυρο που εμφανίζεται βρες το προηγούμενο πρόγραμμα και κάνε διπλό κλικ για να ανοίξει.

Στην μπάρα των **Resources** (πόρων), υπάρχουν όλα τα **sprites**, **objects** και **rooms**, που έχεις δημιουργήσει.

Ένα παιχνίδι πρέπει να αποτελείται από τα παρακάτω μέρη:

- Μία αρχική οθόνη (StartScreen), στην οποία θα εμφανίζονται το λογότυπο δηλαδή το όνομα του παιχνιδιού καθώς και επιλογές σχετικά με το παιχνίδι (συνήθως σε μορφή μενού) και οπωσδήποτε κάποια επιλογή εκκίνησης του παιχνιδιού (Start Button)
- Το κυρίως παιχνίδι που αποτελείται από τα **Rooms**-πίστες (Game Play)
- Μία τελική οθόνη (ExitScreen) η οποία εμφανίζεται μετά το τέλος το παιχνιδιού.

Εκμάθηση προγραμματισμού μέσω της ανάπτυξης παιχνιδιού

Στο δικό μας παιχνίδι θα έχουμε ένα **Room** που θα είναι το StartScreen, 3 **Rooms** κυρίως παιχνίδι και ένα τελευταίο **Room**, ως ExitScreen.

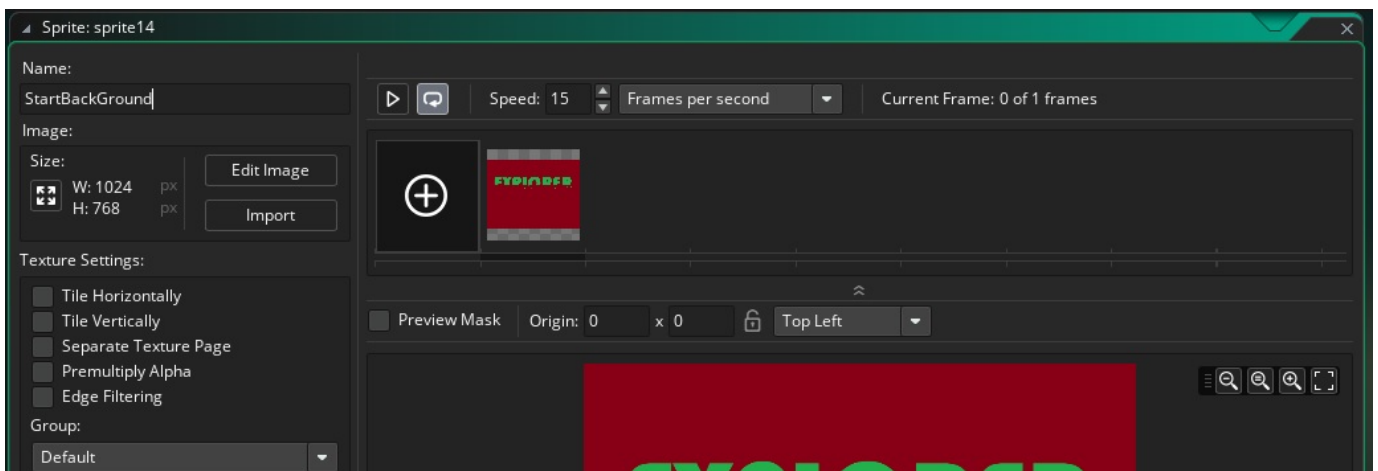
## ΒΗΜΑ ΠΡΩΤΟ: Δημιουργία StartScreen



Δημιούργησε ένα **Sprite** για **Background** του StartScreen καθώς και ένα **Object** που να επιτελεί τη λειτουργία του StartButton.

Ακολούθησε τα παρακάτω βήματα:

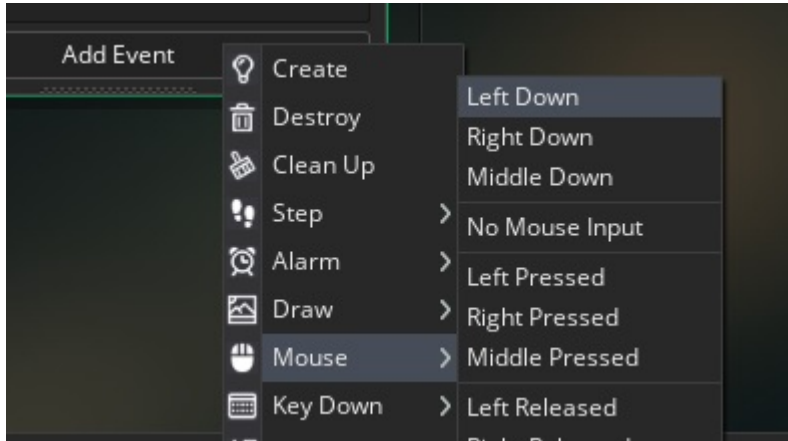
1. Δημιούργησε ένα **Sprite** με όνομα “StartBackGround” χρησιμοποιώντας το αρχείο “start\_screen.png”, όπως φαίνεται στην εικόνα 4.



Εικόνα 4. Δημιουργία Sprite “StartBackGround”

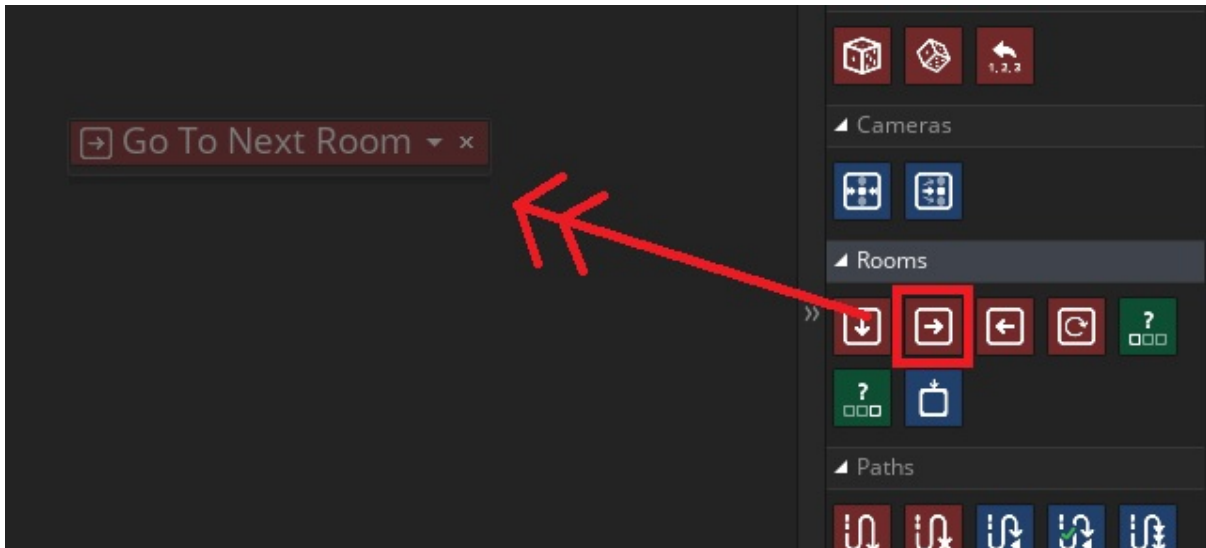
2. Τώρα δημιούργησε ένα **Sprite** με όνομα “StartButtonSprt”, χρησιμοποιώντας την εικόνα “start\_button.png”.
3. Δημιούργησε ένα καινούργιο **Object** με όνομα “StartButton”, με **Sprite** το “StartButtonSprt”, που μόλις δημιούργησες.

4. ΜΠΕΣ ΣΤΟ **Mouse Left Down Event**, όπως φαίνεται στην εικόνα 5.



Εικόνα 5. Δημιουργία Mouse Left Down Event

5. Σύρε την ενέργεια **Go To Next Room**, μέσα στο **Event**, όπως φαίνεται στην εικόνα 6.



Εικόνα 6. Εισαγωγή ενέργειας Go To Next Room.

6. Τέλος, κλείσε το παράθυρο επεξεργασίας του **Object**.

Τώρα είμαστε έτοιμοι να δημιουργήσουμε το StartScreen.

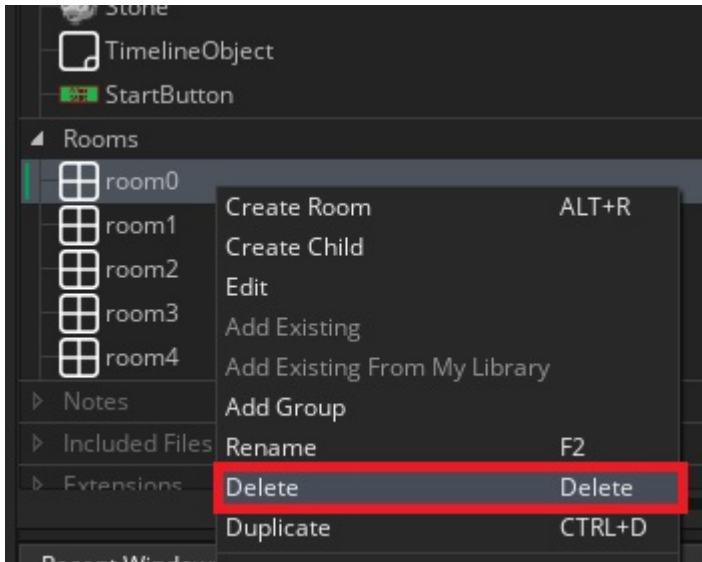
## GOAL

Αντικατέστησε το Room0 με ένα καινούργιο Room. Δώσε το όνομα StartScreen, τοποθέτησε Background το Sprite "StartBackGround", και εισήγαγε το Object "StartButton", που έχεις δημιουργήσει.



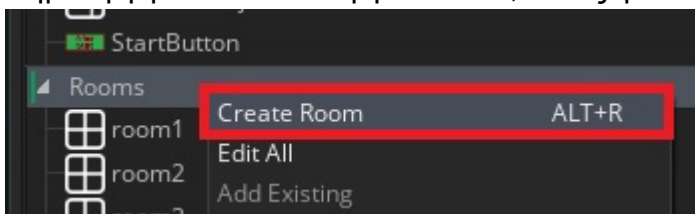
Ακολουθήσε τα παρακάτω βήματα:

1. Διέγραψε το **Room0**, κάνοντας δεξί κλικ και επιλέγοντας **Delete**, όπως φαίνεται στην εικόνα 7.



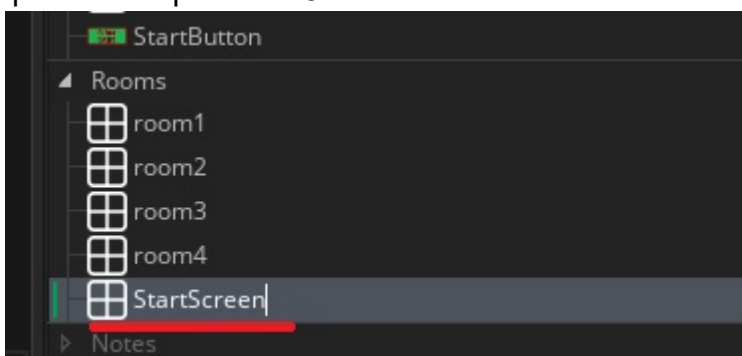
Εικόνα 7. Διαγραφή Room0

2. Δημιούργησε ένα καινούργιο **Room**, όπως φαίνεται στην εικόνα 8.



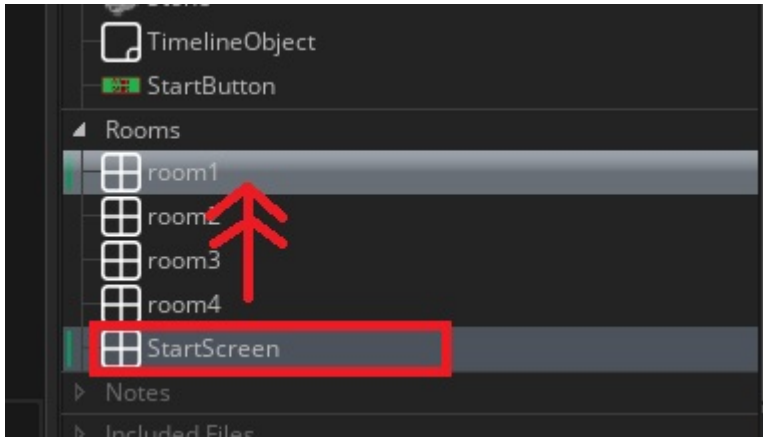
Εικόνα 8. Δημιουργία δωματίου

3. Ονόμασε αυτό το **Room**, "StartScreen" με διπλό κλικ πάνω στο όνομά του, όπως φαίνεται στην εικόνα 9.



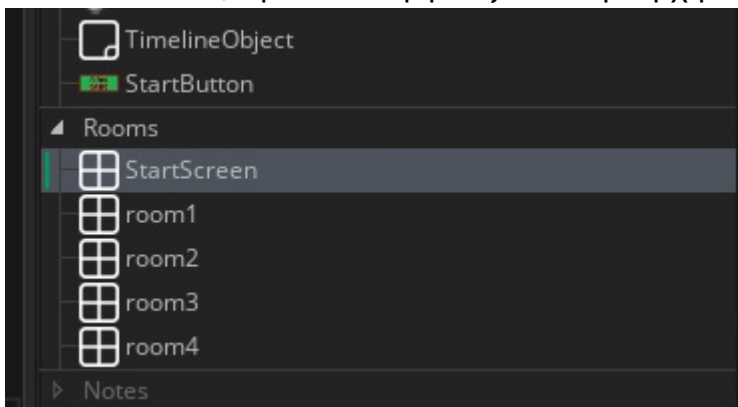
Εικόνα 9. Μετονομασία Δωματίου

4. Σύρε το με κρατημένο κλικ στην πρώτη θέση της λίστας των **Rooms**, έτσι ώστε να εμφανιστεί στην αρχή του παιχνιδιού, όπως φαίνεται στην εικόνα 10.



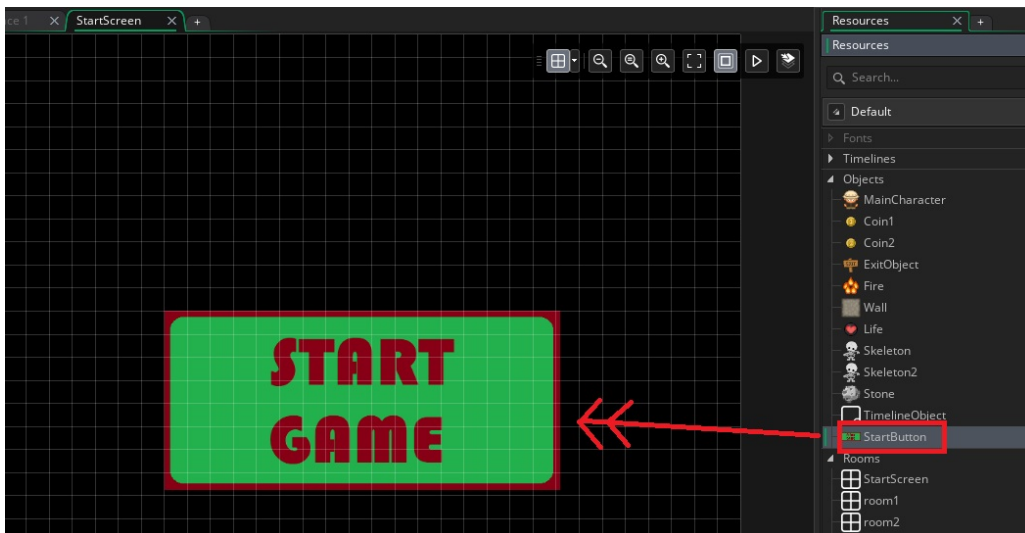
Εικόνα 10. Μεταφορά του StartScreen στην αρχή

Το StartScreen, πρέπει να εμφανίζεται στην αρχή όπως φαίνεται στην εικόνα 11.



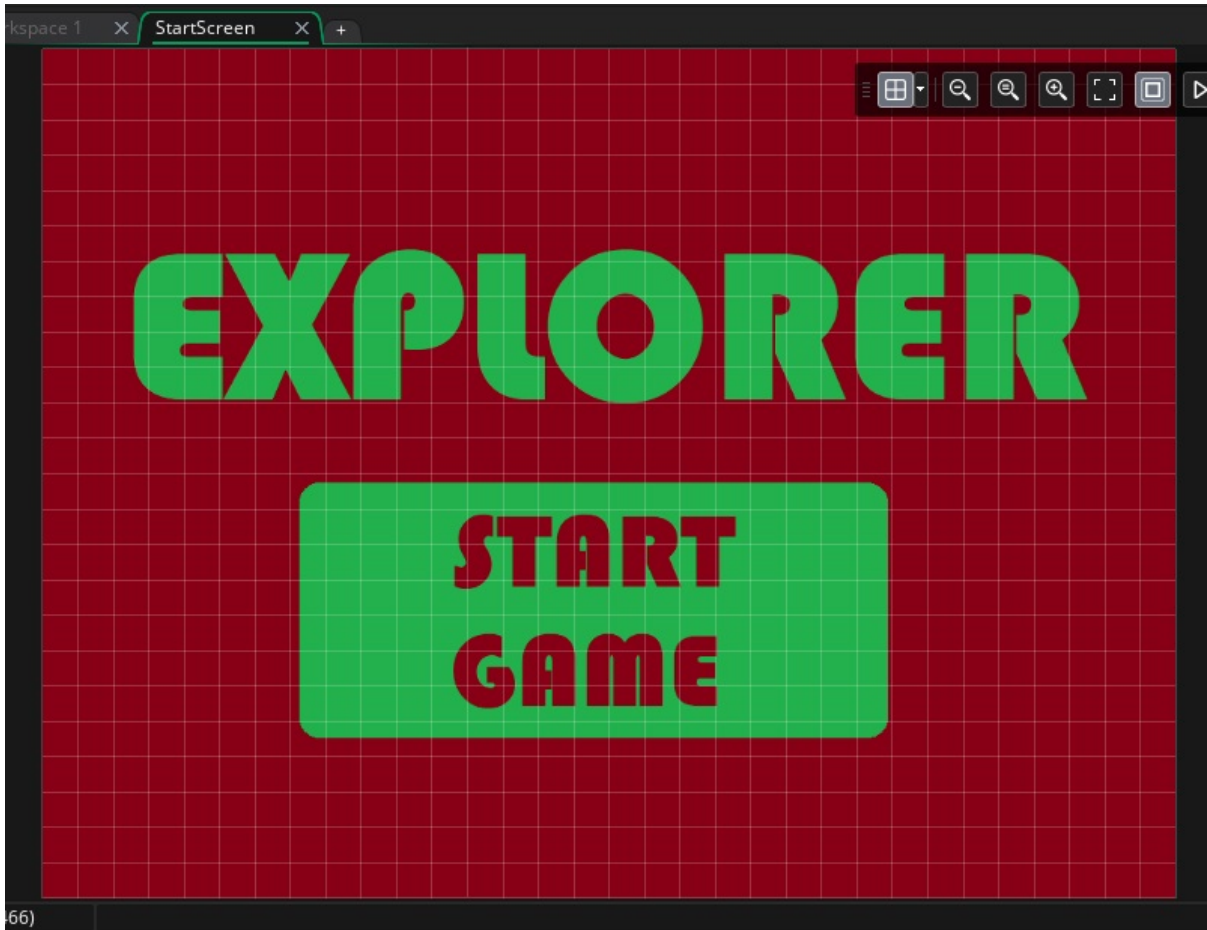
Εικόνα 11. Το StartScreen πρώτο στη λίστα των Rooms.

5. Τώρα σύρε το **object** “StartButton” μέσα στο δωμάτιο όπως φαίνεται στην εικόνα 12.



Εικόνα 12. Εισαγωγή “StartButton” στο “StartScreen”

6. Τώρα όρισε για **Background** του “StartScreen”, το **Sprite** “StartBackGround”, όπως έμαθες στην προηγούμενη δραστηριότητα. Το αποτέλεσμα αυτού του βήματος φαίνεται στην εικόνα 13.



Εικόνα 13. Το τελικό "StartScreen"

7. Τέλος, κλείσε το παράθυρο επεξεργασίας του "StartScreen".

Τέλεια! Τώρα είμαστε έτοιμοι να φτιάξουμε το "ExitScreen".

## ΒΗΜΑ ΔΕΥΤΕΡΟ: Δημιουργία ExitScreen

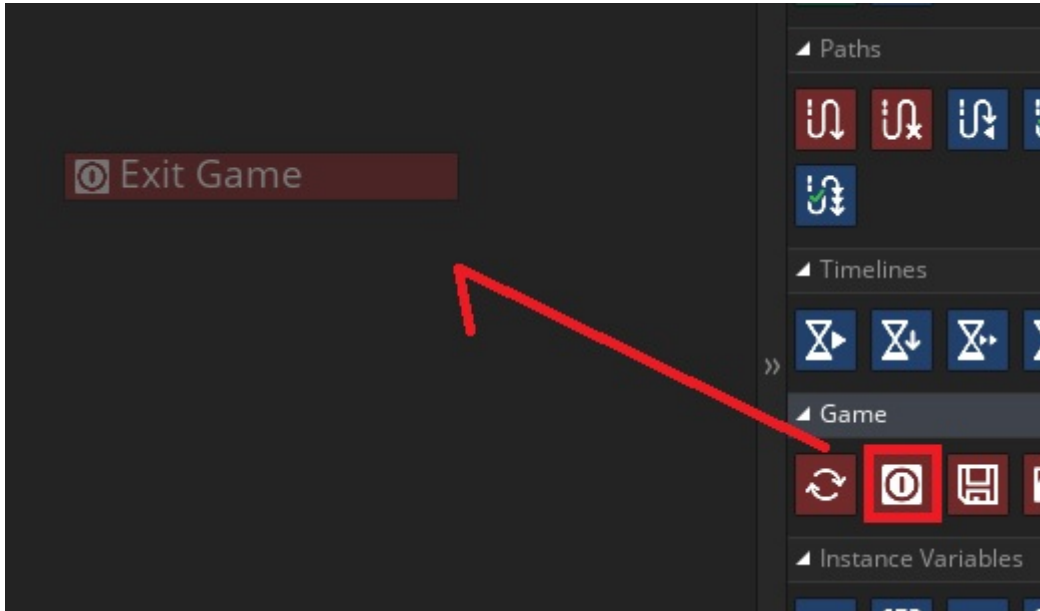
### GOAL

Δημιούργησε ένα Sprite για Background του ExitScreen καθώς και ένα Object που να επιτελεί τη λειτουργία του ExitButton.

Ακολούθησε τα παρακάτω βήματα:

1. Δημιούργησε ένα **Sprite** με όνομα "ExitBackGround" χρησιμοποιώντας το αρχείο "win\_screen.png".

2. Τώρα δημιούργησε ένα **Sprite** με όνομα “ExitButtonSprt”, χρησιμοποιώντας την εικόνα “exit\_button.png”.
3. Δημιούργησε ένα καινούργιο **Object** με όνομα “ExitButton”, με **Sprite** το “ExitButtonSprt”, που μόλις δημιούργησες.
4. Μπες στο **Mouse Left Down Event** και σύρε την ενέργεια **Exit Game**, μέσα στο **Event**, όπως φαίνεται στην εικόνα 14.



Εικόνα 14. Εισαγωγή ενέργειας Exit Game.

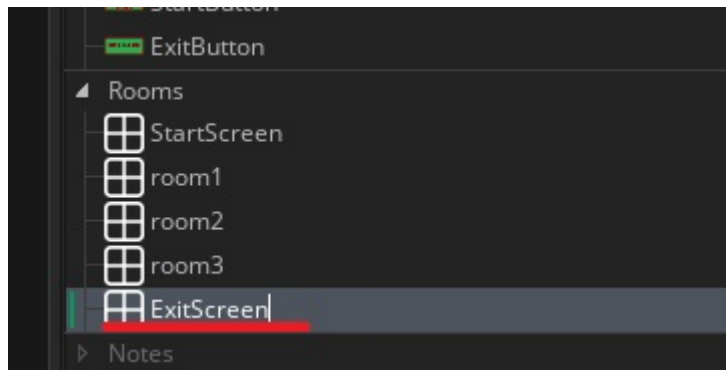
5. Τέλος, κλείσε το παράθυρο επεξεργασίας του **Object**.

## GOAL

Αντικατέστησε το Room4 με ένα καινούργιο Room. Δώσε το όνομα “ExitScreen”, τοποθέτησε Background το Sprite “ExitBackGround”, και εισήγαγε το Object “ExitButton”, που έχεις δημιουργήσει.

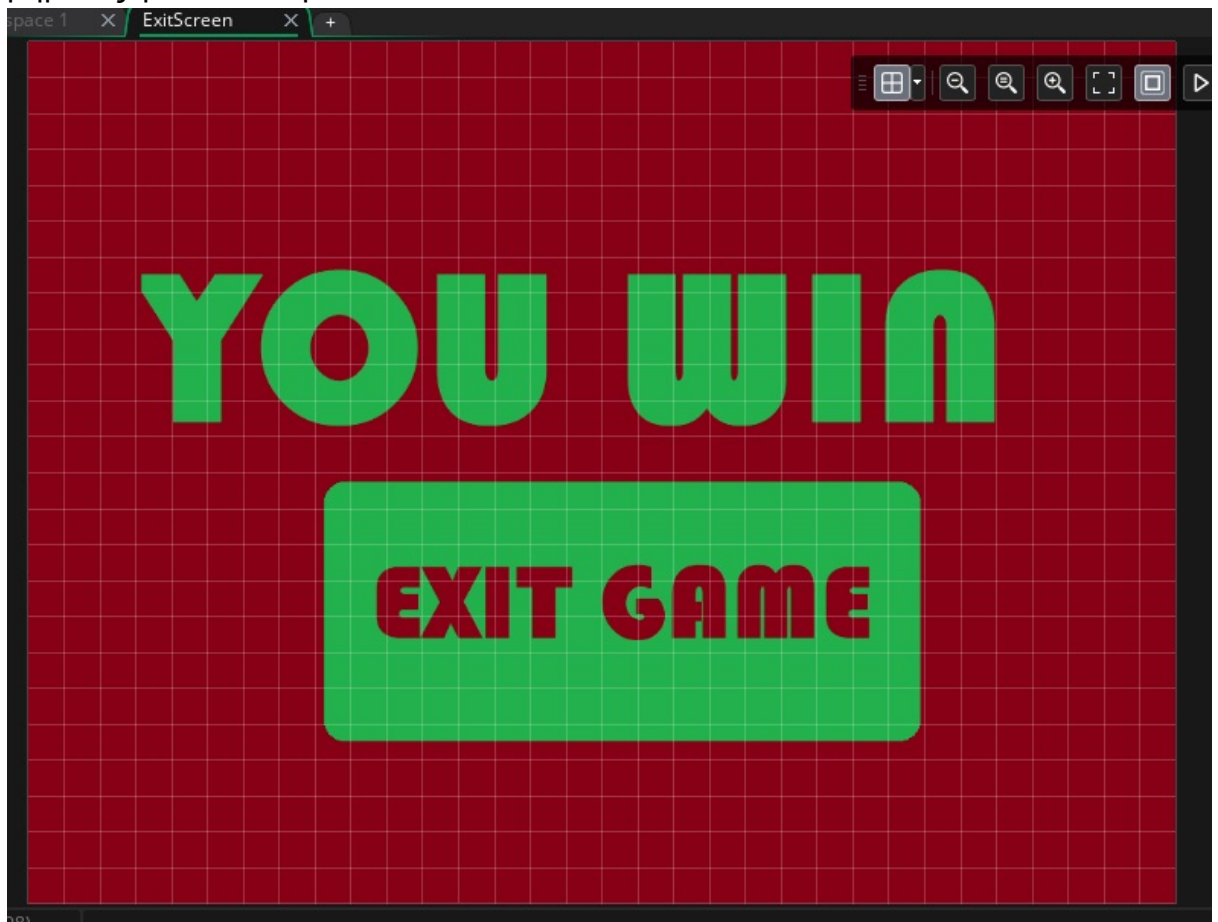
Ακολούθησε τα παρακάτω βήματα:

1. Διέγραψε το **Room4**, κάνοντας δεξί κλικ και επιλέγοντας **Delete**, όπως έκανες και με το **Room0**.
2. Δημιούργησε ένα καινούργιο **Room** και ονόμασέ το, “ExitScreen” με διπλό κλικ πάνω στο όνομά του, όπως φαίνεται στην εικόνα 15.



Εικόνα 15. Μετονομασία Δωματίου

3. Τώρα σύρε το **object** “ExitButton” μέσα στο δωμάτιο και όρισε για **Background** του “ExitScreen”, το **Sprite** “ExitBackGround”. Το αποτέλεσμα αυτού του βήματος φαίνεται στην εικόνα 16.



Εικόνα 16. Το τελικό “ExitScreen”

4. Τέλος, κλείσε το παράθυρο επεξεργασίας του “ExitScreen”.

Τέλεια! Τώρα το παιχνίδι μας είναι ολοκληρωμένο. Δοκίμασε την τελική του μορφή κάνοντας κλικ στο κουμπί **Play**.

## Καλή Διασκέδαση!



- Από πόσα μέρη πρέπει να αποτελείται ένα παιχνίδι και ποια είναι αυτά;
- Τι περιλαμβάνει συνήθως η αρχική οθόνη ενός παιχνιδιού και τι η τελική;
- Πώς μπορούμε να δημιουργήσουμε μία αρχική οθόνη;
- Πώς φτιάχνουμε κουμπιά με λειτουργίες;

### Do!

- Εισήγαγε το κουμπί ExitButton στην αρχική οθόνη του παιχνιδιού, έτσι ώστε να μπορεί ο παίκτης να φύγει από το παιχνίδι αμέσως μόλις μπήκε.
- Εισήγαγε το κουμπί StartButton στην οθόνη εξόδου του παιχνιδιού, έτσι ώστε ο παίκτης αφού τελειώσει να μπορεί να το ξαναπαιξει.
- Τροποποίησε το object "MainCharacter", έτσι ώστε όταν χάνει ο παίκτης όλες του τις ζωές να επιστρέφει στην αρχική οθόνη του παιχνιδιού (TIP: Χρησιμοποίησε την ενέργεια Go To Room).

## ΑΝΑΦΟΡΕΣ

- [1] S.Combefis, G. Beresnevicius and V. Dagiene, Learning Programming through Games and Contests: Overview, Characterisation and Discussion, *Olympiads in Informatics*, Vol. 10, 2016, pp. 39–60.
- [2] M. Prensky, Digital game-based learning. *Computers in Entertainment (CIE)*, 1(1), 2003, pp. 21.
- [3] J. Korpi. "Teaching programming to children through games", Master Thesis, Tampere University of Technology, 2014.
- [4] A. Florea, A. Gellert, D. Florea, A.C. Florea, Teaching Programming by Developing games in Alice, Proc. The 12th International Scientific Conference eLearning and Software for Education Bucharest, April 21-22, 2016, pp. 503-510.
- [5] C. Johnson, Learning basic programming concepts with Game Maker, *International Journal of Computer Science Education in Schools*, Vol. 1, No. 2, May 2017, DOI: 10.21585/ijcses.v1i2.5.