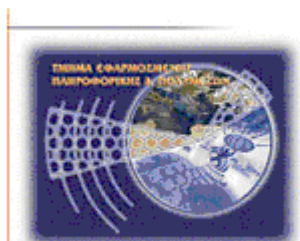




---

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης  
Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή Εργασία

Τίτλος :

ΑΝΑΛΥΣΗ ΣΧΕΔΙΑΣΗ ΚΑΙ ΠΙΛΟΤΙΚΗ ΥΛΟΠΟΙΗΣΗ ΕΩΣ SERIOUS GAMES ΓΙΑ  
ΑΥΤΙΣΤΙΚΑ ΠΑΙΔΙΑ

Κάββαλου Μαρία (2853)

Επιβλέπων καθηγητής : Βιδάκης Νικόλαος

ΗΡΑΚΛΕΙΟ

2016

---

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Βιδάκη Νικόλαο για τη συνολική του βοήθεια καθώς και τους συμφοιτητές μου με τους οποίους συνεργάστηκα.

---

## **Abstract**

Internet platform for online creation of educational games with four categories of users: a) special psychologists b) elementary teachers c) students d) administrators. Everyone's role is defined by the rights under the system. Due to the fact that the platform supports NUI devices games are easy to interact thus enabling children with learning disabilities or children with autism to use the platform and play specially designed for them games. The language in which it is written is c # and the program that was implemented at is unity game engine.

---

## Σύνοψη

Διαδικτυακή πλατφόρμα για on-line κατασκευή εκπαιδευτικών παιχνιδιών με 4 κατηγορίες χρηστών: α) ειδικοί ψυχολόγοι β) δάσκαλοι δημοτικού γ) μαθητές και δ) διαχειριστές. Ο ρόλος του καθενός ορίζεται από τα δικαιώματα που του παρέχει το σύστημα. Λόγω του ότι η πλατφόρμα υποστηρίζει NUI devices τα παιχνίδια είναι εύκολα στη διάδραση δίνοντας έτσι τη δυνατότητα σε παιδιά με μαθησιακές δυσκολίες ή παιδιά με αυτισμό να μπορούν να χρησιμοποιούν την πλατφόρμα και να παίζουν τα ειδικά σχεδιασμένα για αυτά παιχνίδια. Η γλώσσα στην οποία γράφτηκε είναι η c# και το πρόγραμμα στο οποίο υλοποιήθηκε είναι το unity game engine.

---

## Περιεχόμενα

<b>ΕΥΧΑΡΙΣΤΙΕΣ</b> .....	<b>I</b>
<b>ABSTRACT</b> .....	<b>II</b>
<b>ΣΥΝΟΨΗ</b> .....	<b>III</b>
<b>1. ΕΙΣΑΓΩΓΗ</b> .....	<b>1</b>
1.1. ΠΕΡΙΛΗΨΗ.....	1
1.2. ΚΙΝΗΤΡΟ ΓΙΑ ΤΗ ΔΙΕΞΑΓΩΓΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	1
1.3. ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	2
1.4. ΑΡΧΙΚΗ ΚΑΤΑΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ.....	2
1.5. ΔΟΜΗ ΕΡΓΑΣΙΑΣ.....	3
<b>2. ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ</b> .....	<b>4</b>
2.1. ΜΕΘΟΔΟΣ ΑΝΑΛΥΣΗΣ & ΑΝΑΠΤΥΞΗΣ ΠΤΥΧΙΑΚΗΣ.....	4
<b>3. ΣΧΕΔΙΟ ΔΡΑΣΗΣ ΓΙΑ ΤΗΝ ΕΚΠΟΝΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ</b> .....	<b>5</b>
3.1. STATE OF THE ART.....	5
<b>4. ΑΝΑΛΥΣΗ</b> .....	<b>12</b>
4.1. SERIOUS GAMES.....	12
4.2. INCLUSIVE EDUCATION.....	13
4.3. INCLUSIVE EDUCATION AND ASSISTIVE TECHNOLOGY.....	14
4.4. HCI & NUJ.....	16
<b>5. ΚΥΡΙΟ ΜΕΡΟΣ ΠΤΥΧΙΑΚΗΣ</b> .....	<b>18</b>
5.1. ΑΝΑΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ.....	18
5.1.1. <i>Ανάλυση απαιτήσεων</i> .....	18
5.2. ΣΧΕΔΙΑΣΜΟΣ ΥΛΟΠΟΙΗΣΗΣ.....	20
5.2.1. <i>Σχεδιασμός συστήματος</i> .....	20
5.3. ΣΧΕΔΙΑΣΜΟΣ ΔΙΕΠΑΦΗΣ ΧΡΗΣΤΗ (DESIGN).....	21
5.3.1. <i>Αλληλεπίδραση χρήστη</i> .....	22
5.3.2. <i>Ανάλυση χρηστών</i> .....	23
5.3.3. <i>Platform environment</i> .....	
5.4. ΥΛΟΠΟΙΗΣΗ.....	24
<i>Register</i> .....	24
<i>Log in</i> .....	24
<i>Expert Menu</i> .....	25
<i>Educator</i> .....	31
<i>Administrator</i> .....	40
5.4.1. <i>Hardware &amp; Software requirements</i> .....	41
<b>6. ΑΠΟΤΕΛΕΣΜΑΤΑ</b> .....	<b>42</b>
6.1. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	42
6.2. ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	42
<b>ΠΑΡΑΡΤΗΜΑ</b> .....	<b>43</b>

---

## Πίνακας εικόνων

Figure 1 Κύκλος ζωής του λογισμικού ή μοντέλο καταρράκτη	4
Figure 2 Scratch Interface	5
Figure 3 Alice 3D Interface	6
Figure 4 Activity diagram	20
Figure 5 Διαδικασία σχεδιασμού της διεπαφής χρήστη	22
Figure 6 Διαγραμμα use case	23
Figure 7 Οθόνη Register	24
Figure 8 Οθόνη Log in	24
Figure 9 Expert: Start screen	25
Figure 10 Expert: Template creation	26
Figure 11 Expert: Template Creation - Basic Info	26
Figure 12 Expert: Template creation - Style info	27
Figure 13 Expert: Template creation – Target group	27
Figure 14 Expert: Template creation - Scenery Basics	28
Figure 15 Expert: Template creation - Audio & Motion	28
Figure 16 Expert: Template creation - Play environment	29
Figure 17 Expert: Template creation - Rewarding	29
Figure 18 Educator: Start Screen	31
Figure 19 Educator: Template selection	32
Figure 20 Educator: Template customisation	32
Figure 21 Educator: Game description	33
Figure 22 Game Creation : First screen	34
Figure 23 Code: Adding scenes - "+" button	34
Figure 24 Educator: Game creation - Background selection	35
Figure 25 Educator: Game creation - Artifacts Selection	35
Figure 26 Educator: Game creation - Scripts Selection	36
Figure 27 Add scripts: Drag and Drop	36
Figure 28 Code: Drop script on artifact	36
Figure 29 Game editor options: Artifacts' Details	37
Figure 30 Game editor options: Delete Artifact's script	37
Figure 31 Code: Delete artifact's script	37
Figure 32 Game editor options: Toolbars – Rotate / Resize artifact	38
Figure 33 Game editor options: Toolbars - Delete artifact	38
Figure 34 Code: Rotate Artifact	39
Figure 35 Second toolbar	39
Figure 36 Grid Option	39
Figure 37 Administrator: Start screen	40
Figure 38 Administrator: Menu	40

## Πίνακας πινάκων

Table 1 Comparative table - Scratch, Alice, Greenfoot: General	8
Table 2 Comparative table - Scratch, Alice, Greenfoot: Game Creation	10

---

# 1. Εισαγωγή

## 1.1. Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία μιας διαδικτυακής πλατφόρμας με στόχο τη δημιουργία εκπαιδευτικών παιχνιδιών για παιδιά προσχολικής και σχολικής ηλικίας από τους δασκάλους τους. Ως εκ τούτου οι εκπαιδευτικοί έχουν τη δυνατότητα να δημιουργούν εύκολα εκπαιδευτικά παιχνίδια βασιζόμενοι σε προσχεδιασμένα πρότυπα παιχνιδιών που τους παρέχει το σύστημα και τα οποία μπορούν να τροποποιήσουν ανάλογα με την ελευθερία κινήσεων που τους παρέχει το σύστημα. Οι ειδικοί ψυχολόγοι κατασκευάζουν τα πρότυπα των παιχνιδιών με κριτήρια που αντιστοιχούν στην κατηγορία των μαθητών για την οποία προορίζεται το παιχνίδι. Καθορίζουν, ακόμη, το βαθμό στον οποίο μπορεί να επέμβει ο δάσκαλος κατά τη δημιουργία του. Τελικός χρήστης της πλατφόρμας είναι ο μαθητής ο οποίος παίζει το παιχνίδι.

## 1.2. Κίνητρο για τη διεξαγωγή της Εργασίας

Το παιχνίδι ανέκαθεν αποτελούσε μια εύκολη και ευχάριστη προσέγγιση των παιδιών μικρής ηλικίας με σκοπό να προκληθεί η ενθουσιώδης ανταπόκριση τους για τη μάθηση. Πολλές τεχνικές έχουν χρησιμοποιηθεί ανά τα χρόνια που στόχο έχουν να συνδυαστεί το παιχνίδι με τη μάθηση<sup>1</sup>.

Έτσι οι δάσκαλοι και οι νηπιαγωγοί καλούνται να οργανώνουν ομαδικά ή ατομικά παιχνίδια και άλλες εκπαιδευτικές δραστηριότητες ώστε να προωθήσουν τη φυσική, διανοητική και κοινωνική ανάπτυξη των μαθητών. Η πνευματική άσκηση και η ψυχαγωγία, η ατομική δημιουργία και το ομαδικό παιχνίδι, η διήγηση, το θέατρο, η ζωγραφική και η χειροτεχνία είναι τα σημαντικότερα μέσα που χρησιμοποιούνται από τους δασκάλους στην εκπαιδευτική δραστηριότητα<sup>2</sup>.

Από την άλλη, η είσοδος της τεχνολογίας στη ζωή των παιδιών προσχολικής ηλικίας αυξάνεται με ραγδαίο ρυθμό στις μέρες μας καλλιεργώντας το ενδιαφέρον τους γι' αυτήν με αποτέλεσμα να κερδίζει χρήστες όλο και μικρότερης ηλικίας. Το παιχνίδι μέσω του ηλεκτρονικού υπολογιστή αποτελεί, δηλαδή, πολύ αποτελεσματική μέθοδο για τη δημιουργική εκπαίδευση των παιδιών. Με αυτόν τον τρόπο, ενθαρρύνονται να χρησιμοποιούν εφαρμογές πολυμέσων εκπαιδευτικού περιεχομένου και να κατακτούν τις έννοιες της πλοήγησης και της αλληλεπίδρασης. Αποκτούν, επιπλέον στοιχειώδεις δεξιότητες και γνώσεις χειρισμού λογισμικού γενικής χρήσης. Απομυθοποιούν, δηλαδή, τον υπολογιστή και τον χρησιμοποιούν ως εργαλείο ανακάλυψης, δημιουργίας, έκφρασης αλλά και ως νοητικό εργαλείο και εργαλείο ανάπτυξης της σκέψης<sup>3</sup>.

---

### 1.3. Σκοπός και στόχοι της εργασίας

Σκοπός αυτής της εργασίας είναι να αποτελέσει ένα εναλλακτικό τρόπο διδασκαλίας μέσω του παιχνιδιού στον υπολογιστή. Να καλύψει τις υπάρχουσες ανεπάρκειες στον τρόπο που διεξάγεται η εκπαίδευση αλλά και να προωθήσει την προσαρμογή της στην εξέλιξη της τεχνολογίας. Σημαντικό στόχο συνιστά οι μαθητές προσχολικής και σχολικής ηλικίας να λάβουν την εκπαίδευση ως παιχνίδι μέσω δραστηριοτήτων στον υπολογιστή που τους εξάπτουν το ενδιαφέρον και τη φαντασία.

Η παρούσα πτυχιακή εργασία δίνει τη δυνατότητα στον εκπαιδευτικό να κατασκευάσει παιχνίδια στον υπολογιστή του, ελκυστικά ως προς το θέμα, με σαφήνεια, τα οποία ανταποκρίνονται στα ενδιαφέροντα των μαθητών του. Έχοντας αυτούς ως γνώμονες μπορεί να δημιουργήσει παιχνίδια που να περιλαμβάνουν όσο γίνεται μεγαλύτερο μέρος του ψυχισμού τους και επιπλέον να επιβραβεύσει την προσπάθειά τους προσφέροντας ξεκάθαρες απαντήσεις και ανακοινώνοντας την επιτυχία του.

Σκοπός αυτής της διαδικασίας είναι να προάγει τη δημιουργικότητα του παιδιού, επιτρέποντάς του να δοκιμάζει και να πειραματίζεται, εφόσον εμπεριέχει το στοιχείο του νέου, του καινούριου, του απροσδόκητου.

### 1.4. Αρχική καταγραφή προβλήματος

Είναι κοινά παραδεκτό ότι μια από τις βασικότερες προκλήσεις που αντιμετωπίζει η εκπαίδευση είναι η ενσωμάτωση των νέων τεχνολογιών στο σχολικό πρόγραμμα κατά τρόπο όσον το δυνατό πιο ωφέλιμο.

Κύριος στόχος είναι οι νέες τεχνολογίες να ενσωματωθούν πλήρως στο σχολικό πρόγραμμα και τη σχολική πραγματικότητα, να αποτελέσουν οργανικό κομμάτι της και εφιαλτήριο για πρόοδο. Οι νέες τεχνολογίες δε θα πρέπει σε καμία περίπτωση να θεωρούνται υποκατάστατο της συμβατικής διδασκαλίας. Αντίθετα θα πρέπει να εκλαμβάνονται ως ένα μέσο αναβάθμισης και βελτίωσης της διδασκαλίας καθώς και της εν γένει σχολικής πραγματικότητας. Επομένως, για να επιτευχθεί κάτι τέτοιο είναι πολύ σημαντικό η εισαγωγή της τεχνολογίας στην εκπαίδευση να ξεκινήσει από τα πρώτα κιόλας στάδια της εκπαιδευτικής διαδικασίας

Αναμφίβολα, ο καταλληλότερος τρόπος για να γνωρίσουν οι μαθητές προσχολικής και σχολικής ηλικίας την τεχνολογία και να εξοικειωθούν μ' αυτήν είναι τα παιχνίδια στον υπολογιστή. Το παιχνίδι δε θα πρέπει να περιορίζεται στα στενά πλαίσια της ύλης αλλά να επεκτείνεται περισσότερο, δίνοντας έτσι την ευκαιρία στους μαθητές να προσδιορίσουν εκείνοι το εύρος γνώσεων που επιθυμούν να αποκτήσουν. Οι μαθητές που συμμετέχουν σε τέτοια μαθήματα τείνουν να είναι περισσότερο αυτοεξυπηρετούμενοι και αυτοϋποστηριζόμενοι.

Επιπροσθέτως ένα παιχνίδι στον υπολογιστή πρέπει να υποστηρίζει την εξατομικευμένη μάθηση. Για παράδειγμα αν κάποιος αντιμετωπίζει δυσκολίες στην ανάγνωση θα πρέπει στο παιχνίδι να υπάρχουν ενσωματωμένες βελτιώσεις (άμεση επεξήγηση όρων, απλούστερη και περιεκτικότερη γλώσσα, σύντομο κείμενο) που βοηθούν ακόμα περισσότερο αυτούς τους μαθητές.

Γίνεται επομένως εύκολα αντιληπτό ότι είναι επιτακτική ανάγκη το εκπαιδευτικό σύστημα να προσαρμοστεί στις νέες καταστάσεις και να διαμορφώσει ένα νέο χαρακτήρα έτσι ώστε η τεχνολογία να μπορέσει να συμβάλει στο έργο της εκπαίδευσης.



---

## **1.5. Δομή εργασίας**

Η εργασία ξεκινάει με μια σύντομη περιγραφή της πτυχιακής όπου παρουσιάζεται ένα περίγραμμα του πως λειτουργεί και τι σκοπό έχει. Στη συνέχεια παρουσιάζονται τα κίνητρα που μας οδήγησαν στο να την κατασκευάσουμε, καθώς και τι προβλήματα σκοπεύει να επιλύσει. Εν συνεχεία αναλύεται η διαδικασία που ακολουθήθηκε για την κατασκευή της αλλά και η έρευνα που διεξήχθη πριν από αυτή. Έπειτα παρουσιάζονται όλες οι οθόνες της εφαρμογής και τέλος τα συμπεράσματα και η μελλοντική εργασία που προτείνεται.

## 2. Μεθοδολογία Υλοποίησης

Το μοντέλο διαδικασίας παραγωγής λογισμικού που χρησιμοποιήθηκε είναι το μοντέλο καταρράκτη <sup>4</sup>. Το μοντέλο αυτό παίρνει τις θεμελιώδεις δραστηριότητες της διαδικασίας, δηλαδή την εξαγωγή προδιαγραφών, την ανάπτυξη, την επικύρωση και την εξέλιξη και τις αναπαριστά ως ξεχωριστές φάσεις, όπως ο καθορισμός των απαιτήσεων, ο σχεδιασμός λογισμικού, η υλοποίηση, οι δοκιμές κ.λπ.

### 2.1. Μέθοδος ανάλυσης & ανάπτυξης πτυχιακής

Τα κύρια στάδια αυτού του μοντέλου αντιστοιχούν στις θεμελιώδεις δραστηριότητες ανάπτυξης :

1. Ανάλυση και καθορισμός απαιτήσεων.
2. Σχεδιασμός συστήματος και λογισμικού.
3. Υλοποίηση και δοκιμές υπομονάδων.
4. Ενοποίηση και δοκιμές συστήματος.
5. Λειτουργία και συντήρηση.

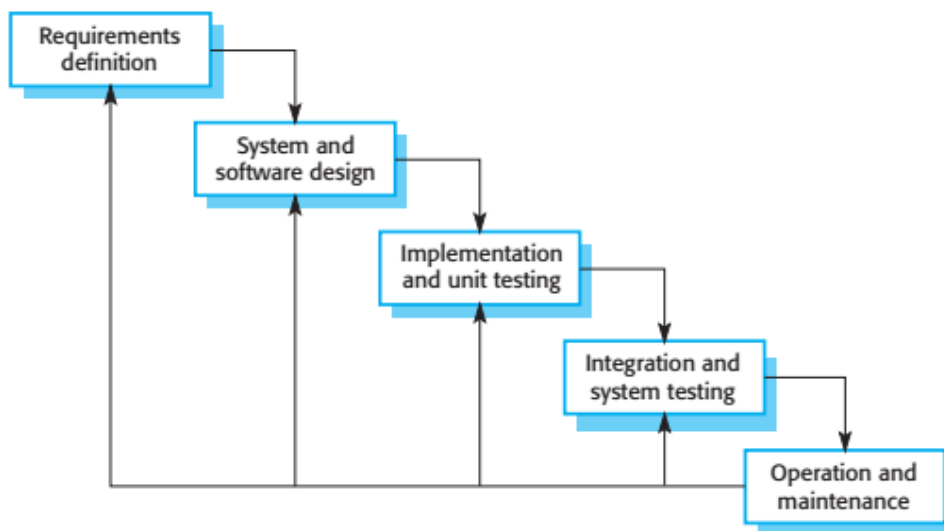


Figure 1 Κύκλος ζωής του λογισμικού ή μοντέλο καταρράκτη

### 3. Σχέδιο δράσης για την εκπόνηση της εργασίας

#### 3.1.State of the art

Ο κύριος άξονας γύρω από τον οποίο διεξήχθη η έρευνα για αυτήν την πτυχιακή εργασία ήταν τα προβλήματα που αντιμετωπίζουν οι μαθητές προσχολικής και σχολικής ηλικίας όσον αφορά τις μαθησιακές δυσκολίες και τον αυτισμό. Στα πλαίσια αυτής της έρευνας μελετήθηκαν μαθησιακές θεωρίες ώστε να κατανοήσουμε τα διάφορα μαθησιακά στυλ και να μπορέσουμε να προσαρμόσουμε την πλατφόρμα με τρόπο που να περιλαμβάνει και να παρέχει όσο το δυνατόν περισσότερα εργαλεία για κάθε τύπο μαθητή. Κυριότερος σκοπός είναι να υπάρχει η δυνατότητα να προσεγγιστούν μαθητές των οποίων το ενδιαφέρον διεγείρεται με διαφορετικούς τρόπους (π.χ οπτικά, ακουστικά ή κινητικά).

Τα σχετικά μοντέλα που ερευνήθηκαν στο πρώτο στάδιο της παρούσας πτυχιακής εργασίας ήταν τα Scratch, Alice και Greenfoot. Τα 3 αυτά συστήματα χρησιμοποιούνται για την εκμάθηση προγραμματισμού σε παιδιά διαφορετικών ηλικιακών group το καθένα.

#### Scratch<sup>5</sup>

Το scratch είναι μια οπτική γλώσσα προγραμματισμού σχεδιασμένη από το MIT media lab για την εκμάθηση του προγραμματισμού σε παιδιά και εφήβους. Είναι μια διαδικτυακή κοινότητα όπου οι χρήστες μπορούν να δημιουργήσουν το δικό τους διαδραστικό υλικό όπως ιστορίες, κάρτες, παιχνίδια αλλά και να τα μοιραστούν με τους υπόλοιπους χρήστες του συστήματος. Ο online scratch editor είναι όπως φαίνεται στο screenshot παρακάτω. Στη μέση του editor υπάρχει η βιβλιοθήκη με μια πληθώρα από objects, εντολές προγραμματισμού, ήχους και άλλα στοιχεία που συντελούν στη δημιουργία του διαδραστικού υλικού. Αριστερά υπάρχει η σκηνή μέσα στην οποία μπαίνουν τα objects/artifacts στα οποία θα προστεθεί η διάδραση. Δεξιά στην οθόνη είναι το πεδίο εκείνο στο οποίο ο χρήστης προσθέτει τη συμπεριφορά του artifact. Ο χρήστης το πραγματοποιεί “τραβώντας” (drag and drop) blocks εντολών στο πεδίο αυτό.

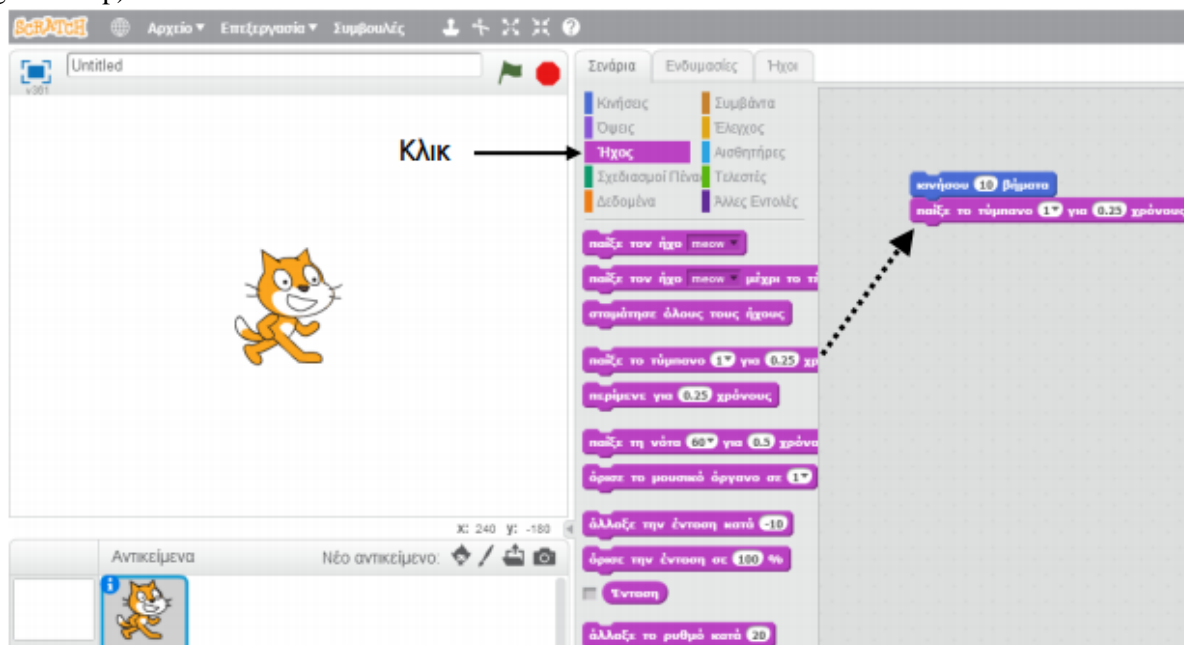


Figure 2 Scratch Interface

Το scratch επιτρέπει την online αλλά ακόμα και την offline χρήση του με εγκατάσταση του scratch editor. Απευθύνεται σε ηλικίες από 8 ως 16 ετών αλλά και σε μικρότερες υπό την επίβλεψη ενός γονέα ή εκπαιδευτικού.

## Alice<sup>6</sup>

Το Alice είναι ένα 3D περιβάλλον προγραμματισμού και παράλληλα διδακτικό εργαλείο σχεδιασμένο για την πρώτη επαφή των μαθητών με τον αντικειμενοστραφή προγραμματισμό. Επιτρέπει την εκμάθηση βασικών εννοιών στα πλαίσια δημιουργίας κινούμενων σχεδίων και απλών διαδραστικών παιχνιδιών. Παρακάτω φαίνεται το interface του Alice. Στη μέση βρίσκεται η σκηνή όπου ο χρήστης προσθέτει τα objects ενώ αριστερά στο πάνω σημείο ο χρήστης βλέπει τον κόσμο που έχει δημιουργήσει και τα αντικείμενα που έχει βάλει μέσα. Επιλέγοντας κάποιο από αυτά τα αντικείμενα ενεργοποιείται το πεδίο αριστερά χαμηλά το οποίο περιέχει τις διαθέσιμες ενέργειες που μπορεί να εκτελέσει το εκάστοτε αντικείμενο. Ο χρήστης επιλέγει από αυτά τα blocks και τα κάνει drag and drop στο ανάλογο πεδίο (μέση χαμηλά) προκειμένου να προσδώσει διάδραση στο αντικείμενο.

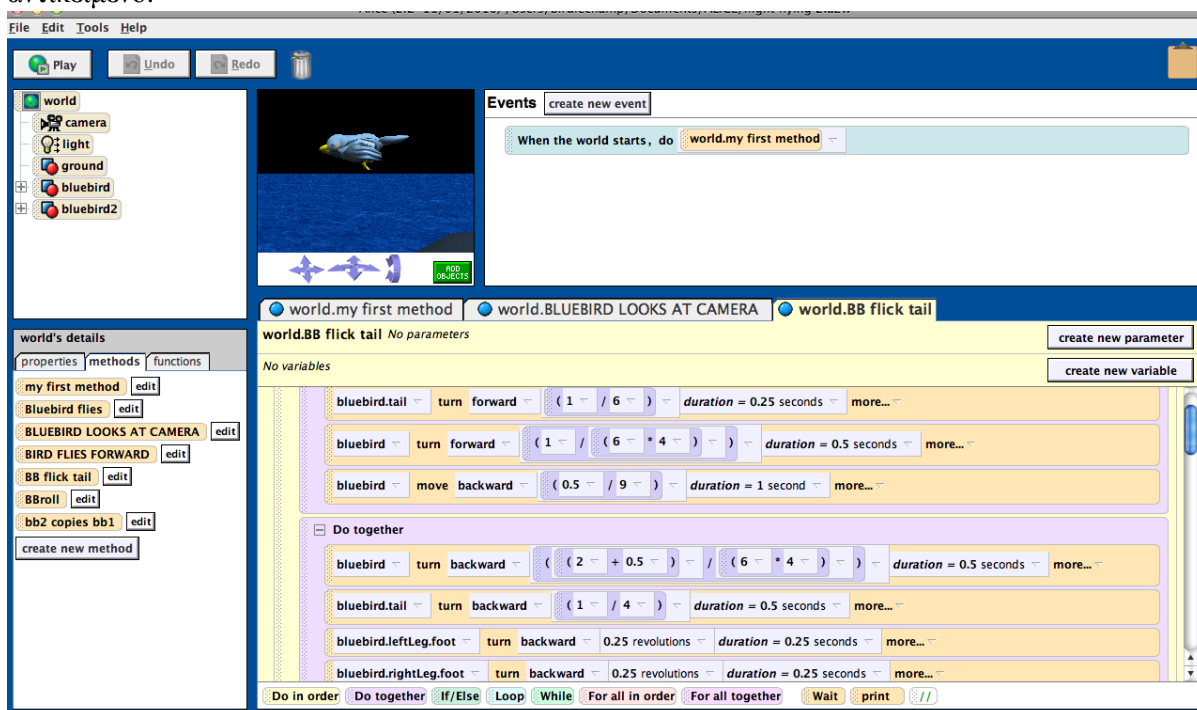
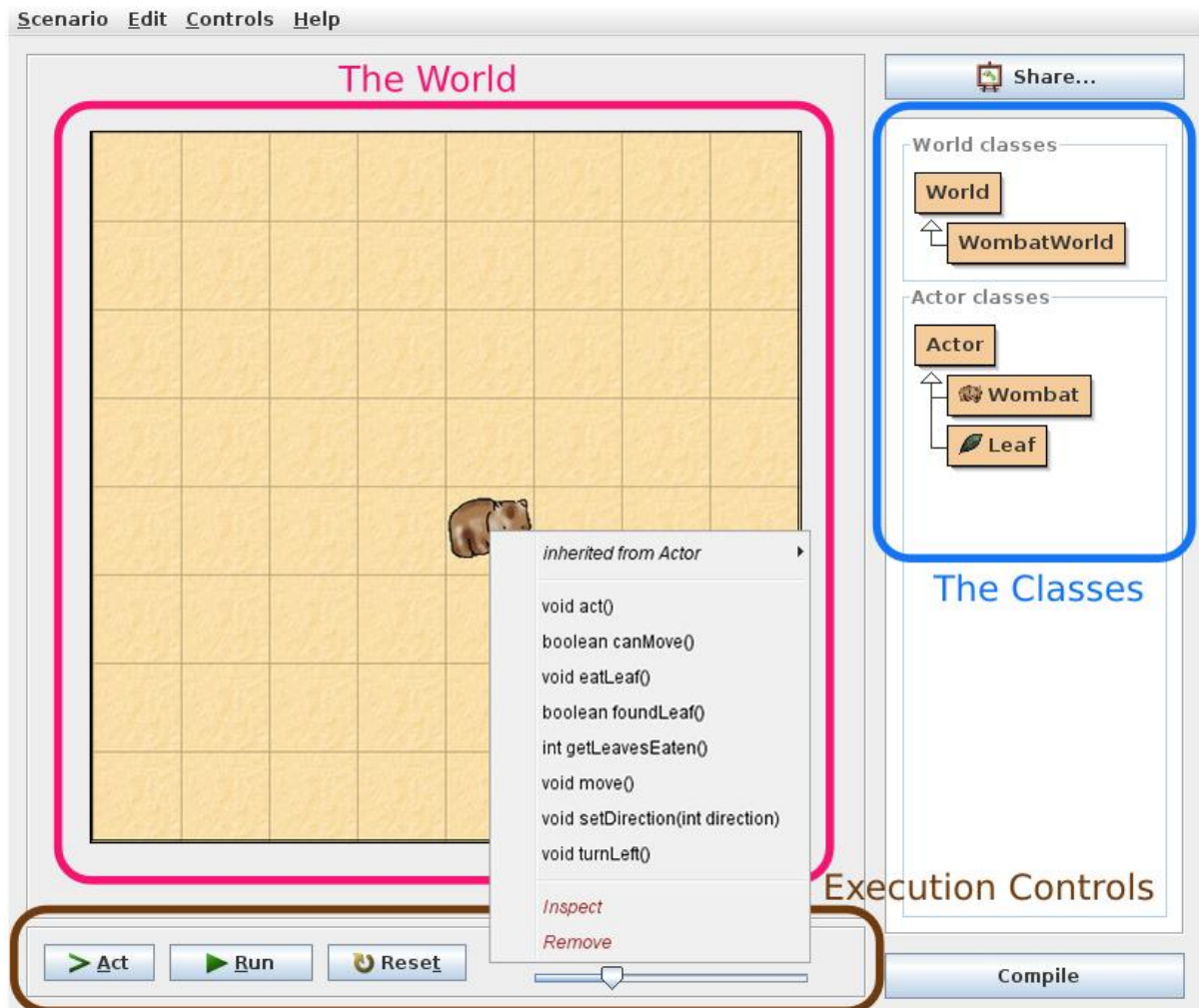


Figure 3 Alice 3D Interface

Για να χρησιμοποιήσει το Alice ο χρήστης, πρέπει να κατεβάσει και να εγκαταστήσει το πρόγραμμα στον υπολογιστή του. Μπορεί παρόλα αυτά να μοιραστεί τις δημιουργίες του στο διαδίκτυο. Το Alice απευθύνεται σε μαθητές ηλικίας από 8-12.

## Greenfoot<sup>7</sup>

Το Greenfoot διδάσκει αντικειμενοστρέφια με Java. Δίνει τη δυνατότητα στο χρήστη να φτιάχνει παιχνίδια, προσομοιώσεις και άλλα προγράμματα γραφικών. Τα αντικείμενα που χρησιμοποιεί ο χρήστης (actors) προγραμματίζονται με κανονικό κώδικα Java σε οπτική εκτέλεση. Στο screenshot φαίνεται η οπτική αναπαράσταση του κώδικα, παρέχεται όμως στο χρήστη η δυνατότητα να δει και τον ακριβή κώδικα που έχει δημιουργηθεί όπως σε ένα κανονικό IDE (ολοκληρωμένο περιβάλλον ανάπτυξης).



Το greenfoot απευθύνεται σε ηλικίες 14+ και απαιτεί τη λήψη και εγκατάστασή του στον υπολογιστή του χρήστη.

Ο λόγος για τον οποίο μελετήθηκαν αυτά τα συστήματα ήταν κυρίως το ότι χρησιμοποιούν το παιχνίδι με σκοπό την εκμάθηση ενός αντικειμένου (προγραμματισμού) σε παιδιά μικρής ηλικίας. Το επίκεντρο του ενδιαφέροντός μας σε αυτά τα μοντέλα αποτέλεσε κυρίως η διεπαφή, η διάδραση του χρήστη με την εφαρμογή και ο τρόπος με τον οποίο προσεγγίζεται ο νεαρός χρήστης.

Για την καλύτερη κατανόηση των διαφορών και ομοιοτήτων των προγραμμάτων αυτών αλλά και για να αποσαφηνίσουμε τα στοιχεία που μας ενδιαφέρουν σ' αυτά δημιουργήσαμε τους παρακάτω πίνακες :

## General

Features	Scratch	Alice	Greenfoot
Age Group	8 – 16	12 - 18	+14
Works on browser	Yes	No	No
Download	Yes	Yes	Yes
Profile Creation	Yes	No	No
Step by step guide	Yes	Yes	No
<u>Getting Started guide (PDF)</u>	Yes	Yes	Yes
Programming Language	Scratch Language	Alice ML (Java, C++, and C#)	Java
Programming with blocks(drag and drop)	Yes	Yes	No
2D	Yes	No	Yes
3D	No	Yes	Limited <sup>1</sup>
Animation(Interactive)	Yes	Yes	Yes
Art	Yes	Yes	Yes
Games	Yes	Yes	Yes
Music	Yes	Yes	Yes
Stories	Yes	Yes	Yes
View/Play with members' projects	Yes	-	Yes
Copy/Change member's projects	Yes	-	Yes
Help section	Yes	Yes	Yes
Forum	Yes	Yes	Yes

Table 1 Comparative table - Scratch, Alice, Greenfoot: General

<sup>1</sup>Limited only in graphics

---

Στα γενικά χαρακτηριστικά τους τα 3 αυτά συστήματα εμφανίζουν πολλές ομοιότητες. Διαφέρουν ωστόσο σε κάποια σημεία. Τα σημαντικότερα από αυτά είναι τα ηλικιακά group στα οποία απευθύνεται το καθένα, η δυνατότητά τους να χρησιμοποιηθούν χωρίς καμία εγκατάσταση αλλά κατευθείαν από τον browser καθώς και στον τρόπο με τον οποίο γίνεται η διάδραση ανάμεσα στο πρόγραμμα και το χρήστη ώστε να δημιουργήσει το διαδραστικό υλικό του.

## Game Creation

Features	Scratch	Alice	Greenfoot
Add background (from library )	Yes	Yes	Yes
Add sprites (from library)	Yes	Yes	Yes
Add your own sprite (including webcam)	Yes	Yes	Yes (No Webcam)
Add Custom Fonts		Yes	-
Moves	Choose from existing blocks <sup>2</sup>	Choose from existing blocks Path/Scripts	Scripting (Java)
Views	Choose from existing blocks	Choose from existing blocks Path/Scripts	Scripting (Java)
Sounds	Choose from existing blocks	Custom Sounds	Custom Path/Scripts
Controls	Choose from existing blocks	Scripting	Scripting (Java)
Events	Choose from existing blocks	Scripted (ML)	Scripted (Java)
Sensors	Choose from existing blocks	Custom sensors	Scripted events (Java)
Create commands	Build your own blocks	Custom Scripts	Custom Scripts (Java)
Supporting platforms	Windows, Mac OS, Linux	Windows, Mac OS, Linux	Windows, Mac OS, Android-Linux-HTML5 <sup>3</sup>
Debugger	No	No	Yes

Table 2 Comparative table - Scratch, Alice, Greenfoot: Game Creation

<sup>2</sup> Blocks can be slightly modified

<sup>3</sup> Supported at full versions (Paid)

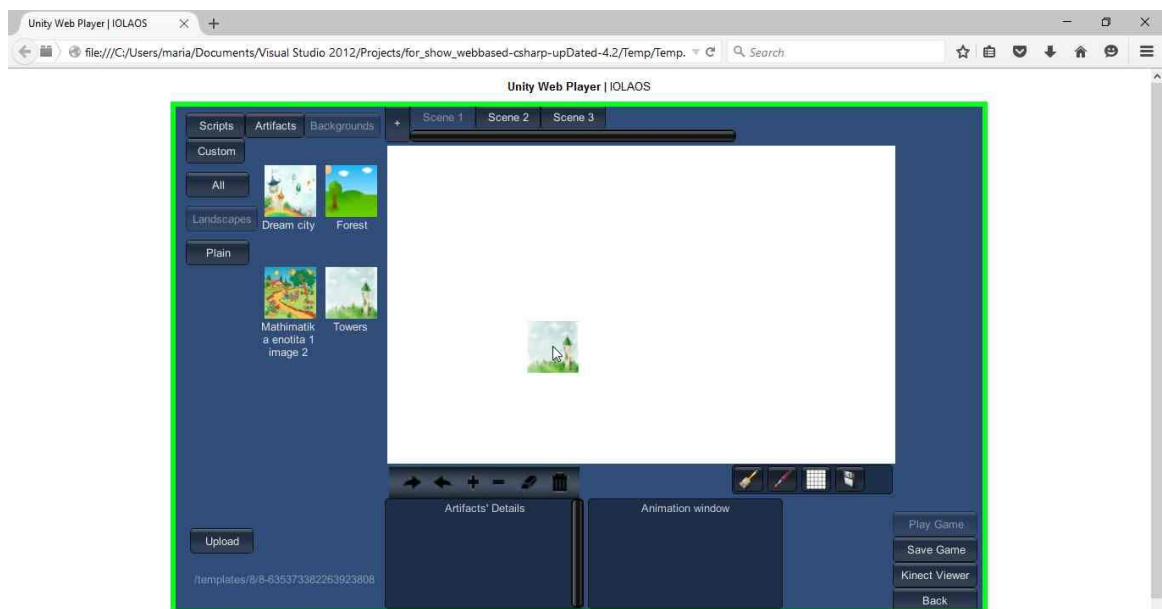


Κατά τη δημιουργία του διαδραστικού υλικού παρατηρήθηκαν αρκετές διαφορές στον τρόπο προσέγγισης. Ο τρόπος εισαγωγής των αντικειμένων που πρόκειται να επεξεργαστούν και να τους δοθεί διάδραση, δε διαφέρει κατά πολύ στα 3 αυτά συστήματα. Εκεί όμως που συναντώνται οι πιο αξιοσημείωτες διαφορές είναι στον τρόπο που προστίθεται η διάδραση στα αντικείμενα. Έτσι για παράδειγμα στο Scratch οι κινήσεις, οι ήχοι, οι εντολές κ.α , βρίσκονται έτοιμα στη βιβλιοθήκη με τη μορφή blocks που ο χρήστης διαχειρίζεται με drag and drop. Στο Alice η διαδικασία αυτή μπορεί να γίνει είτε επιλέγοντας τα ήδη υπάρχοντα blocks είτε κάνοντας δεξιά κλικ πάνω στο αντικείμενο όπου ο χρήστης βλέπει τις διαθέσιμες γι' αυτό ενέργειες.

Συγκρίνοντας τα 3 αυτά σχετικά μοντέλα έγινε μια προσπάθεια να κατανοήσουμε το πώς λειτουργούν και ποια στοιχεία χρησιμοποιούν ώστε να είναι όσο το δυνατόν ευκολότερα στη χρήση τους. Το ενδιαφέρον μας επικεντρώθηκε ακόμα και στον τρόπο που καθλώνουν το χρήστη ώστε να ασχοληθεί μ'αυτά και παράλληλα να αποκτήσει γνώσεις πάνω σε ένα αντικείμενο που θα του φανεί χρήσιμο στο μέλλον.

Ως εκ τούτου το δικό μας σύστημα, διαφέρει αρκετά από τα 3 προηγούμενα που μελετήθηκαν. Στη δική μας περίπτωση ο μαθητής δεν εμπλέκεται καθόλου με τη δημιουργία του διαδραστικού υλικού. Ο μαθητής καλείται μόνο να παίξει και να εκπαιδευτεί μέσω του παιχνιδιού. Τη δημιουργία του διαδραστικού υλικού αναλαμβάνει κάποιος εκπαιδευτικός ο οποίος κατασκευάζει το παιχνίδι στα μέτρα του μαθητή.

Παρόλα αυτά ένας από τους κύριους στόχους μας είναι η διεπαφή να εξυπηρετεί το χρήστη κατανοώντας την πιθανή απουσία εμπειρίας του σε αντίστοιχα συστήματα. Για το λόγο αυτό ο Ιόλαος όσον αφορά στο interface δεν απομακρύνεται πολύ από τα μοντέλα που μελετήθηκαν παραπάνω. Όπως βλέπουμε και στο screenshot παρακάτω, στο κέντρο της οθόνης βρίσκεται η σκηνή στην οποία θα δημιουργηθεί το διαδραστικό υλικό. Έτσι ο χρήστης καλείται να επιλέξει τα αντικείμενα που θέλει να χρησιμοποιήσει από τη βιβλιοθήκη στο αριστερό σημείο της οθόνης και “τραβώντας” τα μέσα στη σκηνή δημιουργεί το παιχνίδι του. Με τον ίδιο τρόπο προσθέτει και τη διάδραση στα αντικείμενα που έχει επιλέξει, “τραβώντας” δηλαδή blocks κώδικα πάνω σ'αυτά.



Ο Ιόλαος επιτρέπει τη δημιουργία προφίλ των χρηστών και την online χρήση του χωρίς να απαιτεί μεγάλη διαδικασία και εγκατάσταση βαρύ λογισμικού. Οι ηλικίες στις οποίες απευθύνεται καθορίζονται από το δημιουργό του διαδραστικού υλικού.

---

## 4. Ανάλυση

### 4.1. Serious games

Τα Serious games<sup>8</sup> (σοβαρά παιχνίδια) είναι παιχνίδια τα οποία σχεδιάζονται με σκοπό την εκπαίδευση πάνω σε ένα αντικείμενο. Παρόλο που χρησιμοποιούνται για ψυχαγωγία ο κύριος σκοπός τους είναι να εξασκήσουν ή να εκπαιδεύσουν το χρήστη. Αποτελούν προσομοιώσεις αληθινών γεγονότων σχεδιασμένων για την επίλυση προβλημάτων. Τα Serious games παίζονται στον υπολογιστή ακολουθώντας κάποιους κανόνες που έχουν προκαθοριστεί<sup>9</sup>. Υπάρχουν παρόλα αυτά και μη ψηφιακά serious games όπως είναι τα παιχνίδια που σχετίζονται με μαθηματικά, που χρησιμοποιούνται στα σχολεία<sup>10</sup>.

#### Ιστορική αναδρομή.

Η πρώτη χρήση του όρου “Serious Game” με την κοντινότερη έννοια στη σημερινή φαίνεται να είναι σε ένα βιβλίο του Clark Abt (1970) με τίτλο “Serious Games”. Ο Abt είναι ένας ερευνητής που εργάστηκε σε ένα ερευνητικό εργαστήριο στην Αμερική κατά τη διάρκεια του ψυχρού πολέμου. Ένας από τους στόχους του ήταν να χρησιμοποιήσει το παιχνίδι για εξάσκηση και εκπαίδευση. Πράγματι κατασκεύασε πολλά τέτοια παιχνίδια για υπολογιστή όπως είναι το T.E.M.P.E.R. Αυτό το παιχνίδι είχε χρησιμοποιηθεί από αξιωματικούς του στρατού για να μελετήσει τη σύγκρουση του Ψυχρού Πολέμου σε παγκόσμια κλίμακα. Ο Abt δίνει ένα καθαρό ορισμό για τα “Serious games”: «Τα παιχνίδια μπορεί να παίζονται σοβαρά ή αδιάφορα. Εμείς ασχολούμαστε με σοβαρά παιχνίδια με την έννοια ότι αυτά τα παιχνίδια έχουν ένα σαφή και προσεκτικά μελετημένο εκπαιδευτικό σκοπό και δεν προορίζονται να παίζονται κυρίως για διασκέδαση. Αυτό δεν σημαίνει ότι τα “Serious games” δεν είναι, ή δεν θα πρέπει να είναι, διασκεδαστικά.»

“Games may be played seriously or casually. We are concerned with serious games in the sense that these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement. This does not mean that serious games are not, or should not be, entertaining.”

Τα serious games βρίσκουν εφαρμογή σε πολλούς τομείς όπως είναι η εκπαίδευση, οι επιστήμες, η υγεία, η πολιτική, οι τέχνες και οι πολιτισμοί.

#### Τα serious games στην εκπαίδευση.

Τα εκπαιδευτικά παιχνίδια δεν έρχονται σε ευρεία χρήση μέχρι τη δεκαετία του 1990 με την αύξηση<sup>11</sup> της χρήσης των ηλεκτρονικών υπολογιστών, παρόλο που τέτοια παιχνίδια είχαν δημιουργηθεί και χρησιμοποιηθεί πολύ πριν. Το ενδιαφέρον όμως για τέτοια παιχνίδια ελαττώθηκε αφενός λόγω της κακής τους ποιότητας και αφετέρου λόγω της αύξησης του ενδιαφέροντος για το διαδίκτυο (Michael & Chen, 2006). Τα εκπαιδευτικά παιχνίδια ήρθαν επίσης αντιμέτωπα με την πρόκληση να αποδείξουν με ερευνητικά στοιχεία τα πλεονεκτήματα που επικαλούνταν. Λόγω όμως του ότι η μελέτη των παιχνιδιών σχετίζεται με πολλούς και διαφορετικούς τομείς, είναι δύσκολο να διεξαχθεί μια προσανατολισμένη έρευνα. Ως αποτέλεσμα της πολυπλοκότητας και της διαφορετικότητας των παιχνιδιών υπάρχουν μόνο λίγα και γρήγορα ευρήματα στη βιβλιογραφία. Παρόλα αυτά όμως οι έρευνες δείχνουν θετικά αποτελέσματα για τα παιχνίδια ως εκπαιδευτικά εργαλεία. Τα σοβαρά παιχνίδια μπορούν να υποστηρίξουν την ανάπτυξη ενός αριθμού διαφόρων δεξιοτήτων όπως είναι: η στρατηγική σκέψη, ο σχεδιασμός, η επικοινωνία, η συνεργασία, η λήψη ομαδικών αποφάσεων και οι διαπραγματευτικές ικανότητες.

Ωστόσο υπάρχουν αρκετά ζητήματα που πρέπει να ληφθούν υπόψη προκειμένου να αξιοποιηθεί πλήρως το δυναμικό των παιχνιδιών ως εκπαιδευτικά εργαλεία. Κάποια από αυτά είναι: οι πόροι (πολλά σχολεία δεν έχουν υπολογιστές ή είναι πολύ παλιοί για καινούργια παιχνίδια, η τεχνική υποστήριξη, ο χρόνος για τους εκπαιδευτικούς να εξοικειωθούν με το παιχνίδι, κ.λπ.), πώς να προσαρμοστεί το παιχνίδι στο πρόγραμμα σπουδών, δυσκολία στο να πεισθούν τα ενδιαφερόμενα μέρη για τα πιθανά οφέλη των ηλεκτρονικών παιχνιδιών.

---

## 4.2. Inclusive Education

Τα τελευταία χρόνια έχουν αναπτυχθεί ποικίλα μοντέλα που εστιάζουν στις στρατηγικές μάθησης, προκειμένου να στηρίξουν τη συμμετοχική εκπαίδευση (inclusive education) των μαθητών με ειδικές ανάγκες. Τέτοιου είδους προγράμματα έχουν σκοπό να διδάξουν στους μαθητές πώς να μαθαίνουν και πώς να επιλύουν προβλήματα. Επιπλέον μπορεί να υποστηριχθεί ότι η ανάληψη μεγαλύτερης ευθύνης από την πλευρά των μαθητών για τη δική τους μάθηση μπορεί να συμβάλει στην επιτυχία της συμμετοχικής εκπαίδευσης στα σχολεία της δευτεροβάθμιας εκπαίδευσης<sup>12</sup>.

Inclusive education (Συμμετοχική εκπαίδευση)

Με τον όρο inclusive education<sup>13</sup> εννοείται η εκπαίδευση κατά την οποία όλα τα παιδιά με ή χωρίς ειδικές εκπαιδευτικές ανάγκες είναι ευπρόσδεκτα από το σχολείο της γειτονιάς τους, για να παρακολουθήσουν μαθήματα σε κανονικές τάξεις και να δεχτούν την κατάλληλη υποστήριξη ώστε να μάθουν, να συνεισφέρουν και να συμμετέχουν σε όλα τα θέματα που αφορούν τη ζωή του σχολείου.

Για μεγάλο χρονικό διάστημα οι μαθητές με ειδικές εκπαιδευτικές ανάγκες εκπαιδεύονταν σε διαφορετικές τάξεις ή σε διαφορετικά σχολεία. Ο κόσμος συνήθισε στην ιδέα ότι ειδική εκπαίδευση σημαίνει και ξεχωριστή εκπαίδευση. Είναι γνωστό όμως ότι όταν οι μαθητές εκπαιδεύονται από κοινού, προκύπτουν θετικά ακαδημαϊκά και κοινωνικά αποτελέσματα για όλα τα παιδιά που έχουν εμπλακεί. Είναι επίσης, γνωστό ότι η απλή τοποθέτηση των παιδιών με και χωρίς ειδικές εκπαιδευτικές ανάγκες δεν παράγουν θετικά αποτελέσματα. Η συμμετοχική εκπαίδευση είναι αποτελεσματική όταν υπάρχει συνεχής υπεράσπιση, σχεδιασμός, υποστήριξη και δέσμευση<sup>14</sup>.

Η συμμετοχική εκπαίδευση επιτρέπει στους συμμετέχοντες μαθητές<sup>15</sup>:

- Την ανάπτυξη προσωπικής δύναμης, με υψηλές και κατάλληλες προσδοκίες για τον καθένα
- Την εργασία πάνω σε ατομικούς στόχους με την παράλληλη συμμετοχή στην τάξη με άλλους μαθητές της ηλικίας τους.
- Την εμπλοκή των γονέων στην εκπαίδευση και σε άλλες δραστηριότητες του σχολείου
- Την προώθηση μιας σχολικής κουλτούρας σεβασμού και αισθήματος ότι ανήκουν σε μια ομάδα. Η συμμετοχική εκπαίδευση δίνει την ευκαιρία στο μαθητή να μάθει και να δεχτεί τις ατομικές διαφορές, καθώς και τη μείωση των επιπτώσεων της παρενόχλησης και του εκφοβισμού.
- Την ανάπτυξη φιλίας ανάμεσα σε μια ευρεία ποικιλία από άλλα παιδιά, το καθένα με τις δικές του ανάγκες και ικανότητες.
- Την θετική επιρροή τόσο στο σχολείο όσο και στην κοινωνία να εκτιμούν τη διαφορετικότητα και την ένταξη σε ένα ευρύτερο πεδίο.

---

Ευρύτερη προσέγγιση :

Όπως διατυπώθηκε από την UNESCO<sup>16</sup> η έννοια και η εφαρμογή της συμμετοχικής εκπαίδευσης έχει αποκτήσει μεγαλύτερη σημασία τα τελευταία χρόνια. Διεθνώς ο όρος νοείται όλο και ευρύτερα ως μεταρρύθμιση που υποστηρίζει και υποδέχεται την ποικιλομορφία μεταξύ όλων των μαθητών.

Η συμμετοχική εκπαίδευση είναι μια διαδικασία που περιλαμβάνει τη μετατροπή των σχολείων και άλλων κέντρων μάθησης με τέτοιο τρόπο ώστε να ανταποκρίνονται στις ανάγκες όλων των παιδιών – συμπεριλαμβανομένων αγοριών και κοριτσιών, μαθητών από εθνικές και γλωσσικές μειονότητες, των αγροτικών πληθυσμών, των πληγέντων από HIV και AIDS , αυτών που αντιμετωπίζουν δυσκολίες στη μάθηση, καθώς και να παρέχουν μαθησιακές ευκαιρίες για όλη τη νεολαία και τους έφηβους.

Στόχος της είναι να εξαλείψει τον αποκλεισμό που είναι συνέπεια των αρνητικών στάσεων και της έλλειψης κατανόησης στη διαφορετικότητα της φυλής, στην οικονομική κατάσταση, στην κοινωνική τάξη, τη γλώσσα, τη θρησκεία, το φύλο, τον σεξουαλικό προσανατολισμό και στις ικανότητες. Η εκπαίδευση λαμβάνει χώρα σε πολλά περιβάλλοντα, ακόμα και μέσα στην οικογένεια αλλά και στην ευρύτερη κοινωνία. Κατά συνέπεια, η συμμετοχική εκπαίδευση δεν είναι ένα περιθωριακό ζήτημα αλλά είναι καίριας σημασίας για την επίτευξη υψηλής ποιότητας στην εκπαίδευση για όλους τους μαθητές και στην ανάπτυξη πιο συμμετοχικών κοινωνιών χωρίς αποκλεισμούς. Η ενιαία εκπαίδευση είναι απαραίτητη για να επιτευχθεί κοινωνική ισότητα και αποτελεί συστατικό στοιχείο της δια βίου μάθησης.

### **4.3. Inclusive education and assistive technology**

Ο όρος assistive technology (AT) ή βοηθητική/υποστηρικτική τεχνολογία είναι οποιοδήποτε αντικείμενο, κομμάτι εξοπλισμού, λογισμικό ή προϊόν το οποίο χρησιμοποιείται για να αυξήσει, να διατηρήσει ή να βελτιώσει τις λειτουργικές ικανότητες των ατόμων με ειδικές ανάγκες<sup>17</sup>.

Κατά τη διάρκεια των δύο τελευταίων δεκαετιών γίνεται όλο και περισσότερο αντιληπτή η ανάγκη για εξέταση της υποστηρικτικής τεχνολογίας (YT) ως μέσο για να παρέχει στα άτομα με ειδικές ανάγκες τη βελτίωση της πρόσβασης, της συμμετοχής και της προόδου στο συνολικό πρόγραμμα γενικής εκπαίδευσης. Η ανάγκη αυτή βασίζεται σε ένα συνεχώς αυξανόμενο αριθμό στοιχείων που δείχνουν ότι η YT μπορεί να χρησιμεύσει ως μια γνωσιακή πρόσθεση για τους μαθητές με ιδιαιτερότητες υποστηρίζοντας τις ικανότητές τους στην κατανόηση, ανάλυση, και σύνθεση των πληροφοριών που θα ήταν διαφορετικά απρόσιτες<sup>18</sup>.

Φυσικά όταν μιλάμε για υποστηρικτική τεχνολογία στο σχολείο για άτομα με ειδικές ανάγκες αναφερόμαστε και στα άτομα με ειδικές εκπαιδευτικές ανάγκες. Η χρήση της υποστηρικτικής εκπαίδευσης ως μέσο για την προώθηση της μάθησης για τους μαθητές με ειδικές εκπαιδευτικές ανάγκες στις τάξεις που εφαρμόζεται η συμμετοχική εκπαίδευση αποτελεί μια ριζική αλλαγή στην εκπαιδευτική διαδικασία. Οι πρακτικές αποκλεισμού στην εκπαίδευση που ήταν συχνό φαινόμενο στις προηγούμενες δεκαετίες αντικαταστάθηκαν με περιβάλλοντα τάξης όπου η ποικιλομορφία και η μάθηση θα μπορούσε να ενισχυθεί μέσω της χρήσης της τεχνολογίας.

Η Υποστηρικτική τεχνολογία παραδοσιακά απευθυνόταν σε άτομα με ειδικές ανάγκες που χρειάζονται βοήθεια με τις λειτουργικές δυνατότητές τους (π.χ., την ικανότητά τους να επικοινωνούν). Για παράδειγμα, ένα άτομο με προβλήματα ακοής μπορεί να χρησιμοποιήσει ένα σύστημα ενίσχυσης της ακοής του στην τάξη ή άτομα με σοβαρούς σωματικούς περιορισμούς μπορούν να χρησιμοποιήσουν ένα διακόπτη για να απαντούν σε ερωτήσεις με ναι ή όχι . Για τα άτομα όμως με μαθησιακές δυσκολίες πρέπει να χρησιμοποιηθεί διαφορετική προσέγγιση της υποστηρικτικής τεχνολογίας.

---

Ένα μεγάλο μέρος του πληθυσμού των μαθητών που φαίνεται να χρειάζονται βοήθεια, δεδομένου του ότι συχνά δεν είναι εύκολο να χωρέσουν σε ένα διαγνωστικό προφίλ, πολλές φορές στερούνται βοήθειας. Η Υποστηρικτική τεχνολογία εξυπηρετεί στη γεφύρωση αυτού του χάσματος με υποστήριξη στην πρακτική εκπαίδευση των παιδιών στην ίδια τάξη, συμπεριλαμβανομένων των παιδιών με σωματικές, νοητικές και αναπτυξιακές αναπηρίες βοηθώντας τους να μάθουν το υλικό με τρόπο που να μπορούν να καταλάβουν, εξαλείφοντας τα προβλήματα που τα εμποδίζουν από το να είναι στο ίδιο επίπεδο με τους συνομηλίκους τους<sup>19</sup>.

Στόχος της ΥΤ για τους μαθητές με εκπαιδευτικές δυσκολίες είναι να παρέχει την απαραίτητη υποστήριξη στους μαθητές έτσι ώστε να έχουν πρόσβαση, να συμμετέχουν και να μαθαίνουν από το γενικό πρόγραμμα σπουδών.

Για τους μαθητές που έχουν ειδικές μαθησιακές δυσκολίες, ένα εκπαιδευτικό λογισμικό μπορεί να βοηθήσει στην ανάπτυξη δεξιοτήτων, προσφέροντας πολυαισθητηριακή εμπειρία, θετική ενίσχυση, εξατομικευμένη διδασκαλία, και επανάληψη. Οι μαθητές που έχουν δυσκολία στην επεξεργασία γραπτών πληροφοριών μπορούν να ολοκληρώσουν τις γραπτές εργασίες με τη βοήθεια ηλεκτρονικών υπολογιστών, όπως το πρότυπο πρόγραμμα επεξεργασίας κειμένου μπορεί να αποδειχθεί ένα πολύτιμο εργαλείο για τους μαθητές με δυσγραφία, αδυναμία δηλαδή να γράφουν ευανάγνωστα κείμενα. Οι ήσυχοι χώροι εργασίας και οι ωτασπίδες μπορούν να κάνουν την είσοδο του υπολογιστή ευκολότερη για τους μαθητές που παρουσιάζουν υπερευαισθησία στο θόρυβο ή που πάσχουν από διάσπαση προσοχής. Προσαρμοσμένες συσκευές όπως μεγάλα εικονίδια εκτύπωσης, εναλλακτικά χρώματα στην οθόνη του υπολογιστή, και φωνητικές εντολές μπορούν να βοηθήσουν τα άτομα με προβλήματα ανάγνωσης.

Για παράδειγμα οι ηλεκτρονικές μαθηματικές ασκήσεις θα μπορούσαν να παρέχουν τη δυνατότητα οι αριθμοί που αναγράφονται να διαβάζονται δυνατά πατώντας ένα κουμπί βοηθώντας τους μαθητές που αντιμετωπίζουν δυσκολίες όσον αφορά την ευθυγράμμιση μαθηματικά προβλήματα χρησιμοποιώντας μολύβι και ένα χαρτί.

Με την εμφάνιση του κοινωνικού μοντέλου της αναπηρίας,

Αυξάνεται συνεχώς ο ισχυρισμός ότι τα μεγαλύτερα εμπόδια για την ένταξη των παιδιών με ειδικές εκπαιδευτικές ή μη ανάγκες είναι αποτέλεσμα των απρόσιτων περιβαλλόντων (Gal et al., 2010). Εκτός αυτού, εξαρτάται ακόμα από τη στάση των καθηγητών και σπουδαστών, το επίπεδο της πρόσβασης στη χρήση της τεχνολογίας, το επίπεδο εμπειρίας και κατάρτισης των εκπαιδευτικών σχετικά με τη χρήση της τεχνολογίας και την εφαρμογή της. Η αντίληψη των μαθητών, η κατάρτιση και αποδοχή, η προσαρμογή των προγραμμάτων σπουδών και η ενσωμάτωση της τεχνολογίας στις τάξεις συμμετοχικής εκπαίδευσης είναι μερικές από τις σημαντικότερες προκλήσεις και καθοριστικοί παράγοντες για την αποτελεσματική χρήση της υποστηρικτικής τεχνολογίας στην εκπαίδευση χωρίς αποκλεισμούς(συμμετοχική εκπαίδευση) (Lang, 2001, Petty, 2012? Reed και Bowser, 2005).

Η προσβασιμότητα αποτελεί πυλώνα στην επίτευξη της διαφορετικότητας, και ένα κρίσιμο παράγοντα για τη διασφάλιση της συμμετοχής των μαθητών στη μαθησιακή διαδικασία. Η πρόσβαση στην πληροφόρηση, η ευαισθητοποίηση, το γενικευμένο εκπαιδευτικό πρόγραμμα σπουδών, το εκπαιδευτικό υλικό, οι βοηθητικές συσκευές και οι απαραίτητες υπηρεσίες υποστήριξης μπορούν να βοηθήσουν τους μαθητές με ειδικές ανάγκες στην εκμάθηση, σπάζοντας όλους τους φραγμούς που τους εμποδίζουν να έχουν την ίδια πρόσβαση σε ποιοτική εκπαίδευση. Τα κανονικά σχολεία με συνολικό προσανατολισμό αποδείχθηκαν πιο αποτελεσματικά στην καταπολέμηση της συμπεριφοράς διακρίσεων, στην οικοδόμηση μιας κοινωνίας χωρίς αποκλεισμούς και στην επίτευξη μιας εκπαίδευσης για όλους (UNESCO, 1994).

---

## 4.4.HCI & NUI

Η αλληλεπίδραση χρήστη υπολογιστή (Human Computer Interaction) αποτελεί το επιστημονικό γνωστικό πεδίο που ασχολείται και μελετά την επικοινωνία ανθρώπου μηχανής<sup>20</sup>. Σύμφωνα με τον ορισμό του οργανισμού ACM(Association for Computing Machinery) «Η αλληλεπίδραση χρήστη – υπολογιστή ασχολείται με το σχεδιασμό, την αξιολόγηση και την υλοποίηση διαδραστικών υπολογιστικών συστημάτων για ανθρώπινη χρήση και με τη μελέτη των μειζόνων φαινομένων που τα περιβάλλουν»<sup>21</sup>.

Κατ' επέκταση, ο όρος διεπαφή χρήστη – υπολογιστή αποτελεί το τμήμα εκείνο ενός υπολογιστικού συστήματος μέσω του οποίου ο χρήστης αλληλεπιδρά με τα λειτουργικά μέρη του συστήματος προκειμένου να εκτελέσει συγκεκριμένα καθήκοντα. Αναλυτικότερα, ως διεπαφή χρήστη θεωρείται το σύνολο των τμημάτων/μερών του υπολογιστή (είτε υλικό είτε λογισμικό) μέσω των οποίων έρχονται σε επαφή και αλληλεπιδρούν ο χρήστης με το υπολογιστικό σύστημα. Συνιστά, δηλαδή, το σημείο επαφής χρήστη και μηχανής ή τη διαχωριστική γραμμή πίσω από την οποία από τη μια μεριά βρίσκεται το περιβάλλον επεξεργασίας του υπολογιστή και από την άλλη ο άνθρωπος.

Ο προσεκτικός σχεδιασμός της διεπαφής χρήστη αποτελεί ουσιαστικό μέρος της συνολικής διαδικασίας του σχεδιασμού λογισμικού. Για να αξιοποιηθούν πλήρως οι δυνατότητες ενός συστήματος λογισμικού, η διεπαφή χρήστη του πρέπει να σχεδιαστεί έτσι ώστε να ανταποκρίνεται στις ικανότητες, την πείρα και τις προσδοκίες των μελλοντικών χρηστών του. Ο καλός σχεδιασμός διεπαφών χρήστη είναι κρίσιμος παράγοντας για τη φερεγγυότητα ενός συστήματος. Πολλά από τα αποκαλούμενα «σφάλματα χρήστη» προκαλούνται από το γεγονός ότι οι διεπαφές δε λαμβάνουν υπόψη τις ικανότητες των πραγματικών χρηστών και το περιβάλλον στο οποίο εργάζονται. Μια κακοσχεδιασμένη διεπαφή σημαίνει ότι οι χρήστες μπορεί να μην είναι σε θέση να προσπελάσουν κάποιες από τις λειτουργίες του συστήματος, να κάνουν λάθη και να αισθάνονται ότι το σύστημα τους εμποδίζει αντί να τους βοηθά στην επίτευξη των στόχων για τους οποίους το χρησιμοποιούν.

Για το σωστό σχεδιασμό μιας διεπαφής οι παράγοντες που πρέπει να ληφθούν υπόψη και τα βήματα που πρέπει να ακολουθηθούν είναι τα εξής<sup>22</sup>:

- Μελέτη των χρηστών και των καθηκόντων τους.  
Θα πρέπει δηλαδή να μελετηθούν τα χαρακτηριστικά των χρηστών για τους οποίους προορίζεται το σύστημα και η χρήση του ίδιου του συστήματος.
- Μελέτη ανταγωνιστικών συστημάτων.  
Είναι πολύ σημαντικό να μελετώνται προϊόντα της ίδιας κατηγορίας όσον αφορά τη χρηστικότητα και τη λειτουργικότητά τους και να λαμβάνονται υπόψη οι κριτικές αυτών.
- Συμμετοχή των χρηστών.  
Οι χρήστες χρειάζεται να συμμετέχουν στη διαδικασία του σχεδιασμού μέσω τακτικών συναντήσεων γεγονός το οποίο βοηθάει πολύ στη μελέτη των αντιδράσεων τους έτσι ώστε να εκτιμηθεί η διεπαφή και να επέλθουν οι απαραίτητες αλλαγές.
- Συντονισμός της συνολικής διεπαφής έτσι ώστε να τηρείται μια συνοχή
- Αξιολόγηση της διεπαφής σύμφωνα με τις κατευθυντήριες γραμμές.
- Κατασκευή πρωτοτύπου ή μερική και γρήγορη υλοποίηση του συστήματος.
- Επαναληπτικός σχεδιασμός.  
Επανασχεδιασμός της διασύνδεσης με βάση την ανατροφοδότηση από την αξιολόγηση.

### NUI (Natural User Interface)

Η φυσική διασύνδεση χρήστη(NUI) είναι ένα σύστημα αλληλεπίδρασης χρήστη-υπολογιστή όπου ο χρήστης λειτουργεί με διαισθητικές ενέργειες που σχετίζονται με την καθημερινή ανθρώπινη συμπεριφορά<sup>23</sup>.

Ένα NUI μπορεί να λειτουργεί με πολλούς διαφορετικούς τρόπους, ανάλογα με το στόχο και του χρήστη και τις απαιτήσεις του. Μερικά NUIs βασίζονται σε ενδιάμεσες συσκευές για την επίτευξη της αλληλεπίδρασης ενώ άλλα πιο προηγμένα NUIs είναι είτε αόρατα είτε τόσο διακριτικά που γρήγορα φαίνονται αόρατα.

---

## Εφαρμογές και παραδείγματα.

Τα συστήματα αναγνώρισης κινήσεων και χειρονομιών (gesture recognition) ανιχνεύουν τις κινήσεις των χρηστών και τις μεταφράζουν σύμφωνα με τις οδηγίες. Τα συστήματα παιχνιδιών κίνησης στο Nintendo Wii και στο PlayStation δουλεύουν με επιταχυνσιόμετρα και γυροσκόπια για να αποδώσουν αίσθηση της κλίσης, της περιστροφής και της επιτάχυνσης.

Ένα πιο καινοτόμο είδος NUI είναι εφοδιασμένο με φωτογραφική μηχανή και λογισμικό στη συσκευή που αναγνωρίζει συγκεκριμένες χειρονομίες και τις μεταφράζει σε ενέργειες. Το Kinect της Microsoft, για παράδειγμα, είναι ένας αισθητήρας κίνησης για την κονσόλα παιχνιδιών Xbox 360 που επιτρέπει στους χρήστες να αλληλεπιδρούν με κινήσεις του σώματος, χειρονομίες και φωνητικές εντολές. Το Kinect αναγνωρίζει και διαχωρίζει τα σώματα και τις φωνές μεμονωμένων παικτών. Η αναγνώριση των κινήσεων του σώματος και των χεριών μπορεί επίσης να χρησιμοποιηθεί για την αλληλεπίδραση με τους υπολογιστές.

Η αναγνώριση ομιλίας (Speech recognition) επιτρέπει στους χρήστες να αλληλεπιδρούν με το σύστημα μέσω προφορικών εντολών. Το σύστημα αναγνωρίζει λέξεις και φράσεις και τις μετατρέπει σε αναγνώσιμη μορφή ώστε να πραγματοποιηθεί η αλληλεπίδραση. Οι εφαρμογές αναγνώρισης ομιλίας περιλαμβάνουν την δρομολόγηση των κλήσεων, μετατροπή ομιλίας σε κείμενο (speech-to-text) και hands-free λειτουργία του υπολογιστή και του κινητού τηλεφώνου.

## Πλεονεκτήματα χρήσης NUI συσκευών<sup>24</sup>:

- Παρέχουν μια πολυ-αισθητηριακή εμπειρία. Δεν υπάρχει μεσολαβητής στην αλληλεπίδραση η οποία πραγματοποιείται με χειρονομίες, κινήσεις του σώματος και φωνητικές εντολές
- Ενισχύει τον έλεγχο του χρήστη πάνω στη διασύνδεση. Δεν υπάρχει καμία τεχνητή συσκευή ελέγχου με τροχούς, κουμπιά ή πλήκτρα
- Οι χρήστες δεν χρειάζεται να διαθέτουν εξειδικευμένες γνώσεις ηλεκτρονικού υπολογιστή, προκειμένου να χειριστούν το περιβάλλον καθώς είναι σχεδιασμένα για «φυσική» αλληλεπίδραση.
- Υποστηρίζουν ταυτόχρονη διάδραση από περισσότερους από ένα χρήστη
- Μπορεί να είναι διασκεδαστική και ψυχαγωγική και να ενθαρρύνει το χρήστη για εξερεύνηση.

## Μειονεκτήματα χρήσης NUI συσκευών:

- Δεν υπάρχουν καθιερωμένες συμβάσεις. Εκτός από κάποια σχεδιαστικά πρότυπα σε συσκευές πολλαπλής αφής, οι άλλες διασυνδέσεις δεν έχουν ένα σύνολο κατευθυντήριων γραμμών ακόμα. Αυτό μπορεί να προκαλέσει κάποια σύγχυση μεταξύ των χρηστών
- Υπάρχει έλλειψη ανατροφοδότησης. Για παράδειγμα, δεν υπάρχει πάντα κάποια ένδειξη που να υποδεικνύει προς τα που δείχνει το δάκτυλο του χρήστη όπως συμβαίνει με τον δείκτη του ποντικιού στα GUIs
- Είναι πολύ δύσκολο να χρησιμοποιούνται πολύπλοκα συστήματα αποκλειστικά και μόνο με gesture-based διεπαφές.
- Ίδιες χειρονομίες σημαίνουν διαφορετικά πράγματα σε διαφορετικούς πολιτισμούς. Ως εκ τούτου, υπάρχει μια ανάγκη για διαπολιτισμικές συμβάσεις
- Για όλους τους παραπάνω λόγους, οι χρήστες μπορεί εύκολα να ενοχληθούν, να βαρεθούν ή να κουραστούν, αν δεν μπορούν να καταλάβουν πώς να αποκτήσουν τον έλεγχο πάνω στο γραφικό περιβάλλον.

---

## 5. Κύριο μέρος πτυχιακής

### 5.1.Ανάλυση Προβλήματος

Τα παιδιά μιας σχολικής τάξης έχουν ασφαλώς πολλά κοινά χαρακτηριστικά, αλλά έχουν και αξιοσημείωτες διαφορές. Το θέμα των ιδιαίτερων χαρακτηριστικών του παιδιού μπορεί να θεωρηθεί ως το κρισιμότερο στοιχείο που θα καθορίσει σε μεγάλο βαθμό την αποτελεσματικότητα στη χρήση της πληροφορικής στην εκπαίδευση.

Σημαντικό παράγοντα στη διεξαγωγή μιας ολοκληρωμένης και αποτελεσματικής εκπαιδευτικής διαδικασίας αποτελεί το κατά πόσο οι εκπαιδευτές διαθέτουν την κατάλληλη κατάρτιση ώστε να είναι σε θέση να αναγνωρίσουν ένα παιδί με μαθησιακές ή άλλες δυσκολίες και να το αντιμετωπίσουν κατάλληλα παρέχοντάς του την απαιτούμενη βοήθεια.

Ο δάσκαλος θα πρέπει σε τέτοιες περιπτώσεις να μπορεί να επιφέρει όλες εκείνες τις τροποποιήσεις στη διδακτική μέθοδο, με σκοπό να γίνεται κατανοητό το αντικείμενο από τους μαθητές και να προχωρούν όλοι με τον ίδιο ρυθμό .

Λαμβάνοντας υπόψη τα παραπάνω συμπεραίνουμε ότι η εκπαίδευση πρέπει να ανταποκρίνεται όσο γίνεται στις ανάγκες, τις δυνατότητες και τις αδυναμίες κάθε μαθητή. Αυτό μπορεί να επιτευχθεί με εκσυγχρονισμό στις μεθόδους διδασκαλίας, τη νοοτροπία και τις παρεχόμενες γνώσεις. Τα νέα δεδομένα προωθούν τη δημιουργία μιας εκπαίδευσης ευέλικτης με πολλές εναλλακτικές δυνατότητες<sup>25</sup>.

Το ερώτημα λοιπόν που προκύπτει είναι πως μπορεί η τεχνολογία να ενσωματωθεί στο εκπαιδευτικό σύστημα και να βοηθήσει το δάσκαλο και τους μαθητές στην επίτευξη της αποτελεσματικότερης και πιο ευχάριστης διδασκαλίας.

#### 5.1.1. Ανάλυση απαιτήσεων

##### Λειτουργικές απαιτήσεις

Οι χρήστες της πλατφόρμας του ΙΟΛΑΟΣ είναι οι εξής : 1) Διαχειριστές, 2) Ειδικοί ψυχολόγοι, 3) Εκπαιδευτικοί, 4)Παίκτες. Οι παίκτες μπορεί να είναι :

- παιδιά προσχολικής και σχολικής ηλικίας,
- παιδιά προσχολικής και σχολικής ηλικίας με μαθησιακές ή άλλες δυσκολίες,
- παιδιά προσχολικής και σχολικής ηλικίας με αυτισμό ή σύνδρομο Asperger.

Το σύστημα :

- λειτουργεί ως web εφαρμογή από τον browser του χρήστη
- απαιτεί ταυτοποίηση χρήστη.
- παρέχει τη δυνατότητα δημιουργίας προσωπικού προφίλ σε όλους τους χρήστες
- κρατάει το ιστορικό ενός παιχνιδιού(όπως στατιστικά ή σκόρ) σε κάθε προφίλ
- παρέχει προκαθορισμένα παιχνίδια για διάφορες κατηγορίες παικτών
- ενσωματώνει τη δυνατότητα χρήσης του με διαφορετικές συσκευές εισόδου όπως ποντίκι, πληκτρολόγιο και NUI devices (kinect,leap) .

Οι ειδικοί ψυχολόγοι μπορούν :

- να κατασκευάζουν πρότυπα παιχνιδιών



- 
- να θέτουν τους όρους και την ελευθερία στη δυνατότητα τροποποίησής τους.

Οι εκπαιδευτικοί μπορούν :

- να επιλέξουν και να τροποποιήσουν ένα πρότυπο παιχνιδιού ανάλογα με την ελευθερία που τους παρέχει το σύστημα
- να αποθηκεύσουν ένα τροποποιημένο παιχνίδι στο προφίλ τους.

Οι παίκτες :

- παίζουν τα ειδικά γι' αυτούς σχεδιασμένα παιχνίδια.

Οι διαχειριστές :

- έχουν το δικαίωμα να εκτελούν τις ενέργειες όλων των χρηστών.

### **Μη λειτουργικές απαιτήσεις.**

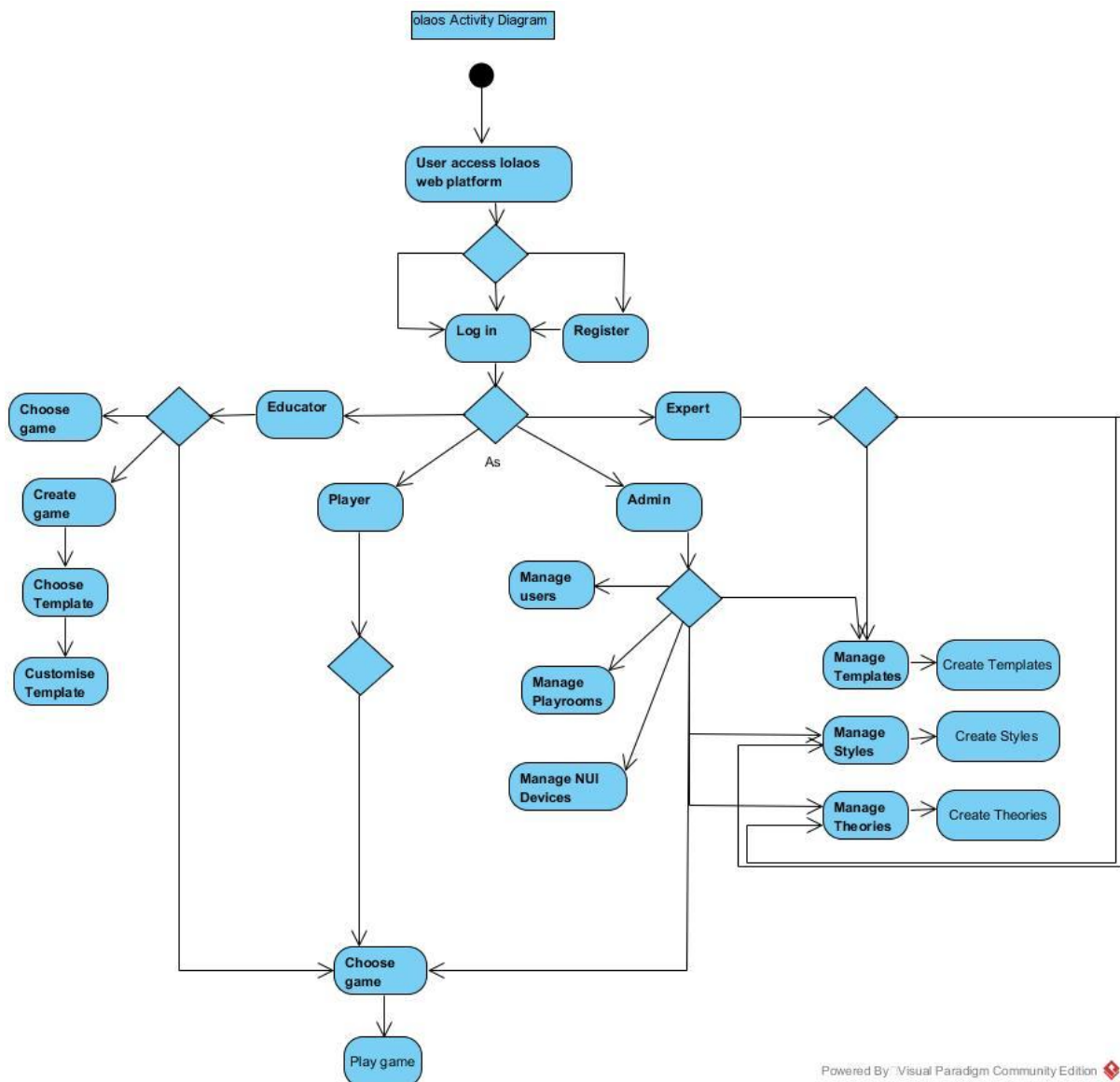
- Το σύστημα δε θα πρέπει να επιτρέπει καμία ενέργεια αν ο χρήστης δεν έχει πραγματοποιήσει είσοδο.
- Το σύστημα δε θα πρέπει να επιτρέπει καμία λειτουργία αν ο χρήστης δεν έχει ταυτοποιηθεί.
- Οποιοσδήποτε χρήστης θα πρέπει να μπορεί εύκολα και γρήγορα να κάνει εγγραφή ή είσοδο στο σύστημα μέσω των κατάλληλων μορφών και της καθοδήγησης που παρέχεται.
- Κατά τη διάρκεια οποιασδήποτε ενέργειας (όπως game creation, play, save ή load ) το σύστημα δεν πρέπει να έχει μη αποδεκτές/προσδοκώμενες καθυστερήσεις.
- Για όλες τις καθυστερήσεις που διαρκούν πάνω από 5 δευτερόλεπτα θα εμφανίζεται ένα dialog box που θα λέει “παρακαλώ περιμένετε”
- Σε περίπτωση μη ανταποκρινουσας κατάστασης μακράς διάρκειας, το σύστημα θα κάνει επανεκκίνηση.
- Ο μεσαίος χρήστης (ειδικός ψυχολόγος - εκπαιδευτικός) θα πρέπει να μπορεί εύκολα και γρήγορα να δημιουργεί νέο παιχνίδι αφότου έχει λάβει υπόψιν τις υπάρχουσες πηγές βοήθειας.
- Παρέχεται η δυνατότητα στο χρήστη να πραγματοποιεί οποιαδήποτε λειτουργία του συστήματος από κάθε παράθυρο σε λιγότερα από 4 κλικ.
- Μόνο ο ψυχολόγος και ο διαχειριστής έχουν τη δυνατότητα να κατασκευάζουν πρότυπα παιχνιδιών.
- Το σύστημα πρέπει να είναι εύκολο στη χρήση τόσο για τους χρήστες με εμπειρία σε σχετικά συστήματα όσο και για τους αρχάριους χρήστες.
- Όλες οι οθόνες θα πρέπει να έχουν κουμπί βοήθειας.
- Κάθε κουμπί βοήθειας σε μια δεδομένη οθόνη θα πρέπει να παρέχει τουλάχιστον ένα θέμα για κάθε λειτουργία (control) στην οθόνη.
- Η διεπαφή χρήστη πρέπει να είναι όσο πιο απλή και ελκυστική γίνεται ώστε να διευκολύνεται η χρήση και η διάδραση με αυτήν (σχήματα, χρώματα, ήχοι).
- Το γραφικό περιβάλλον πρέπει να προσελκύει τον παίκτη και να κρατά το ενδιαφέρον του σε υψηλά επίπεδα.

- Το λογισμικό πρέπει να είναι επεκτάσιμο και επαναχρησιμοποιήσιμο για μελλοντική επεξεργασία.
- Το σύστημα θα πρέπει να είναι συμβατό με τα Windows.
- Όλοι οι browsers επιτρέπουν τη χρήση του συστήματος χωρίς την εγκατάσταση προγραμμάτων.
- Το σύστημα μπορεί να χρησιμοποιηθεί από οποιαδήποτε μηχανή που διαθέτει browser.
- Η γλώσσα προγραμματισμού στην οποία θα υλοποιηθεί το σύστημα είναι η c#

## 5.2. Σχεδιασμός υλοποίησης

### 5.2.1. Σχεδιασμός συστήματος

Activity diagram



Powered By Visual Paradigm Community Edition

Figure 4 Activity diagram

---

Αναλυτικότερα, ο χρήστης κάνει Register αν δεν είναι ήδη εγγεγραμμένος και στη συνέχεια κάνει Log in. Ανάλογα με το είδος του χρήστη παρέχονται διαφορετικές ενέργειες. Αν ο χρήστης που πραγματοποιεί είσοδο στο σύστημα είναι:

- Εκπαιδευτής/Educator, μπορεί να επιλέξει κάποιο από τα υπάρχοντα παιχνίδια ή να δημιουργήσει ένα καινούριο παιχνίδι ακολουθώντας τα προκαθορισμένα βήματα. Αυτά είναι, επιλογή προσχεδίου παιχνιδιού και παραμετροποίησή του ως το βαθμό που του επιτρέπεται. Η διαδικασία αυτή θα πρέπει να είναι εύκολη και κατανοητή από τον χρήστη είτε εκείνος έχει εμπειρία με αντίστοιχα συστήματα είτε όχι. Θα πρέπει ακόμα να μπορεί να δημιουργήσει ένα παιχνίδι με λίγες κινήσεις και σε μικρό χρονικό διάστημα.
- Ειδικός ψυχολόγος/Expert, μπορεί να διαχειρίζεται τα προσχέδια των παιχνιδιών, δηλαδή να δημιουργεί καινούρια, να τροποποιεί ή να διαγράφει κάποια άλλα. Μπορεί ακόμα να διαχειρίζεται τις θεωρίες μάθησης που υπάρχουν στο σύστημα και να δημιουργεί καινούριες. Τέλος μπορεί να διαχειρίζεται τα στυλ που παρέχει το σύστημα για τα παιχνίδια και να δημιουργεί καινούρια. Τα στυλ αυτά μπορεί να αφορούν τις οθόνες ή τα χρώματα που μπορούν να χρησιμοποιηθούν στα παιχνίδια και άλλα τέτοια χαρακτηριστικά.
- Παίκτης/Player μπορεί να επιλέξει κάποιο από τα διαθέσιμα παιχνίδια και στη συνέχεια να τα παίζει.
- Διαχειριστής/Administrator έχει τη δυνατότητα να εκτελεί τις ενέργειες όλων των χρηστών και επιπλέον να διαχειρίζεται τους χρήστες, τα playrooms και τις συσκευές που μπορούν να χρησιμοποιηθούν

### 5.3 Σχεδιασμός διεπαφής Χρήστη (Design)

Η συνολική διαδικασία σχεδιασμού της διασύνδεσης χρήστη είναι η εξής :

#### 1. Ανάλυση χρηστών.

Στη διαδικασία ανάλυσης των χρηστών αναπτύσσεται κάποια κατανόηση για τις εργασίες που κάνουν οι χρήστες για το εργασιακό τους περιβάλλον, για τα άλλα συστήματα που χρησιμοποιούν, για το πώς αλληλεπιδρούν με άλλα άτομα στην εργασία τους, κ.ο.κ.

#### 2. Κατασκευή πρωτοτύπου του συστήματος.

Ο σχεδιασμός και η ανάπτυξη της διασύνδεσης χρήστη είναι μια επαναληπτική διαδικασία. Αν και οι χρήστες μπορούν να εκφέρουν γνώμη σχετικά με τις λειτουργίες που χρειάζονται από μια διασύνδεση, τους είναι πολύ δύσκολο να είναι συγκεκριμένοι μέχρι να δούν κάτι απτό. Χρειάζεται λοιπόν να αναπτυχθούν πρότυπα του συστήματος και να παρουσιαστούν στους χρήστες, οι οποίοι στη συνέχεια μπορούν να καθοδηγήσουν την εξέλιξη της διασύνδεσης.

#### 3. Αξιολόγηση διασύνδεσης.

Αν και κατά τη διαδικασία ανάπτυξης πρωτοτύπου, οι χρήστες παίζουν σημαντικό ρόλο στην ανατροφοδότηση, πρέπει να υπάρχει μια πιο τυποποιημένη δραστηριότητα αξιολόγησης, στην οποία συλλέγονται πληροφορίες για την πραγματική εμπειρία των χρηστών με τη διασύνδεση.

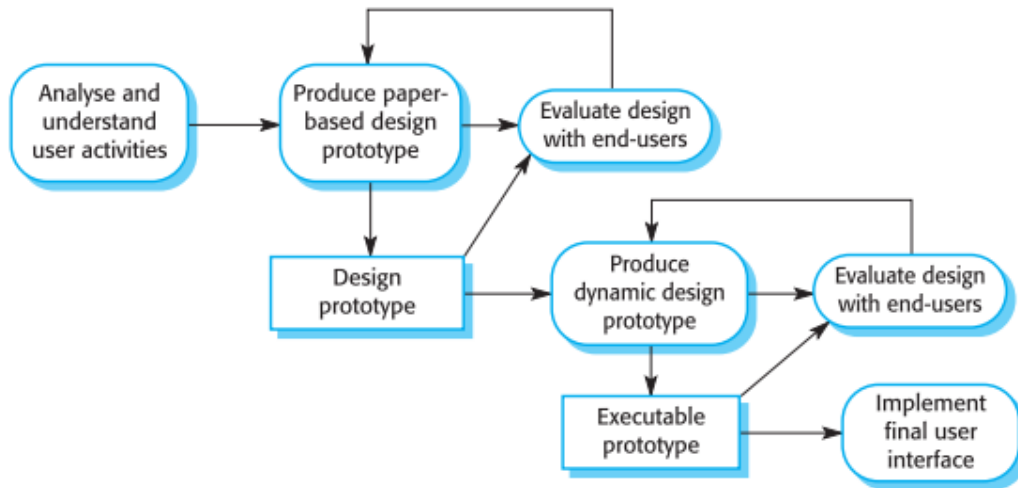


Figure 5 Διαδικασία σχεδιασμού της διεπαφής χρήστη

### 5.3.1 Αλληλεπίδραση χρήστη

Ο όρος αλληλεπίδραση χρήστη αναφέρεται στην τροφοδότηση του υπολογιστικού συστήματος με εντολές και σχετικά δεδομένα από το χρήστη. Στον τρόπο δηλαδή που μπορεί αυτός να χρησιμοποιήσει μια εφαρμογή για να εκτελέσει τις ενέργειες που επιθυμεί. Υπάρχουν διάφορες μορφές αλληλεπίδρασης και η πλατφόρμα μας χρησιμοποιεί τις παρακάτω :

#### Άμεσος χειρισμός.

Ο χρήστης αλληλεπιδρά άμεσα με αντικείμενα στην οθόνη. Ο χρήστης έχει τη δυνατότητα να πλοηγηθεί και να χρησιμοποιήσει την πλατφόρμα χρησιμοποιώντας είτε ποντίκι, είτε πληκτρολόγιο είτε άλλες NUI συσκευές.

#### Επιλογή μενού.

Ο χρήστης επιλέγει μια εντολή από ένα κατάλογο δυνατοτήτων(μενού). Η πλατφόρμα παρέχει τη δυνατότητα εκτέλεσης διαφόρων ενεργειών μέσα από επιλογές που δίνει στο μενού.

### 5.3.2 Ανάλυση χρηστών

Use case diagram

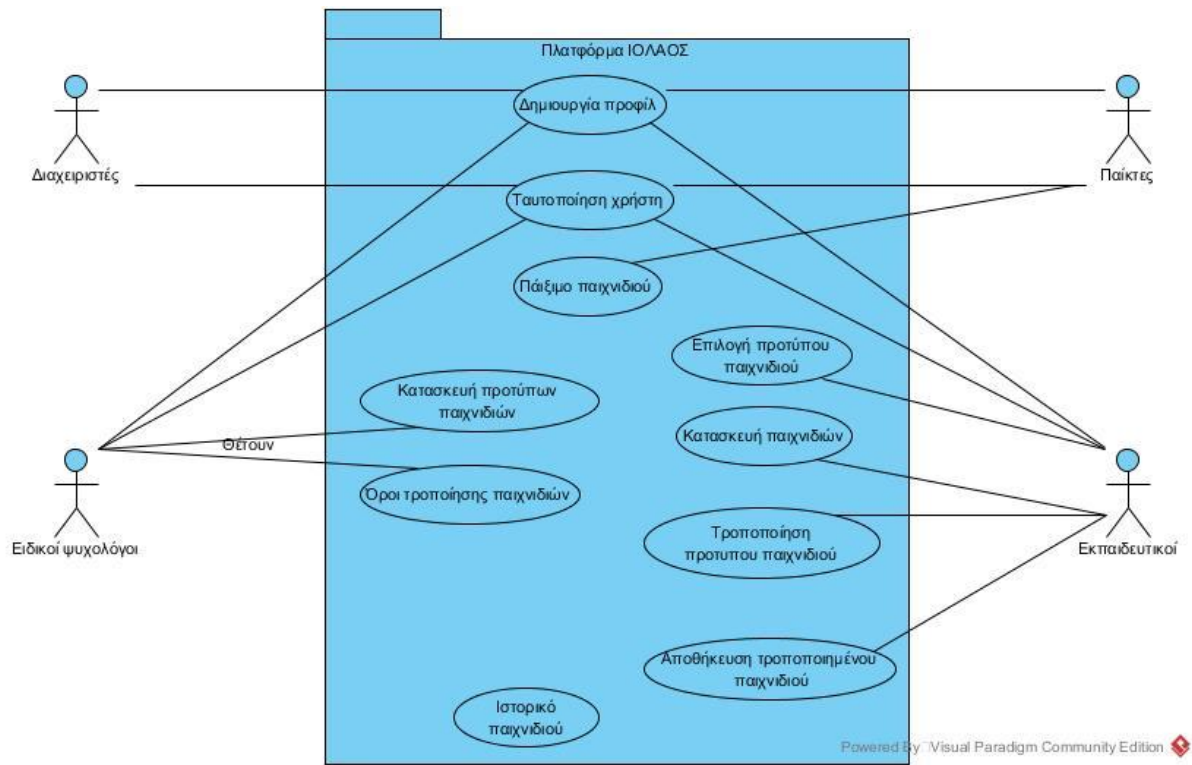


Figure 6 Διαγράμμα use case

## 5.4 Υλοποίηση

Η διαδικασία χρήσης της πλατφόρμας είναι η εξής :

Εφόσον ο χρήστης μπει στη σελίδα της πλατφόρμας μέσω του browser του πρέπει να πραγματοποιήσει εγγραφή στο σύστημα αν δεν είναι χρήστης ή είσοδο στο σύστημα αν είναι.

### Register

Κατά τη διαδικασία εγγραφής, η πλατφόρμα απαιτεί από το χρήστη να συμπληρώσει κάποια στοιχεία όπως φαίνεται παρακάτω.

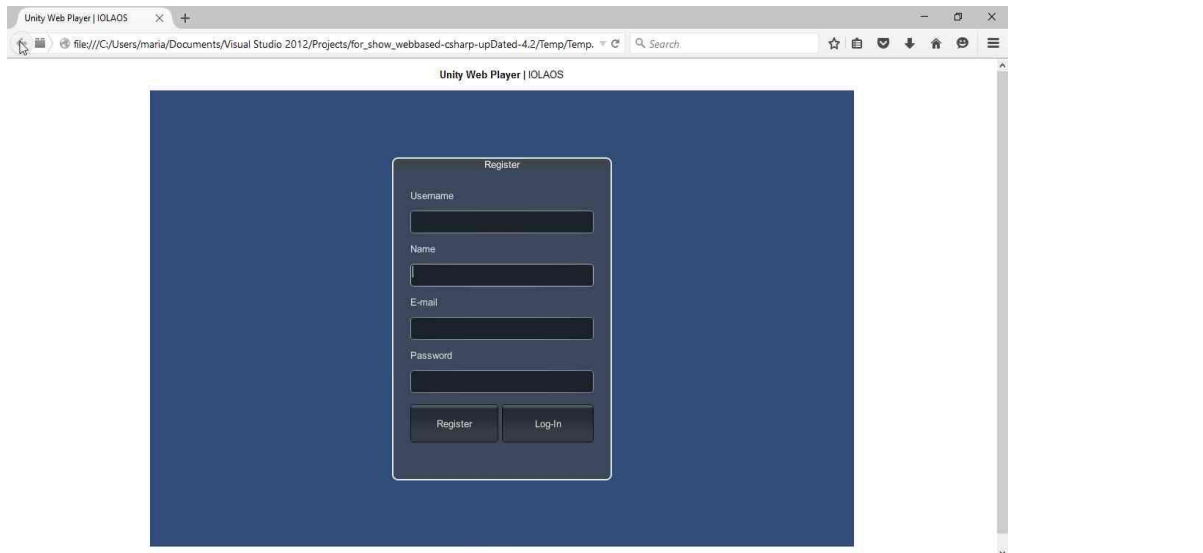


Figure 7 Οθόνη Register

### Log in

Κατά το log in ο χρήστης συμπληρώνει το user name και το password του από το προφίλ που έχει ήδη δημιουργήσει.

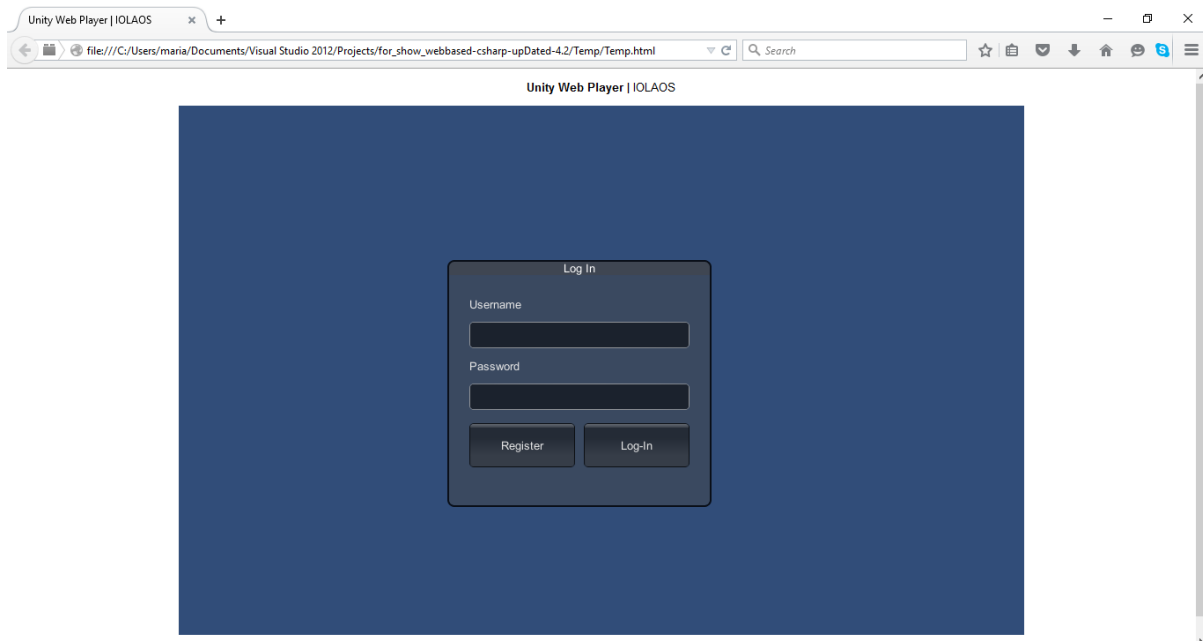


Figure 8 Οθόνη Log in

Σε αυτό το σημείο, αναλόγως με το είδος του χρήστη το σύστημα προσφέρει διαφορετική λειτουργικότητα. Έτσι έχουμε:

## Expert Menu

### *Manage templates:*

Όπου ο Expert user μπορεί να διαχειρίζεται τα διαθέσιμα προσχέδια παιχνιδιών(edit, delete, view) ή να δημιουργήσει ένα καινούριο. Του δίνεται ακόμα η δυνατότητα να διαχειριστεί τα διάφορα στυλ και τις θεωρίες πάνω στα οποία βασίζονται τα πρότυπα των παιχνιδιών που κατασκευάζει.

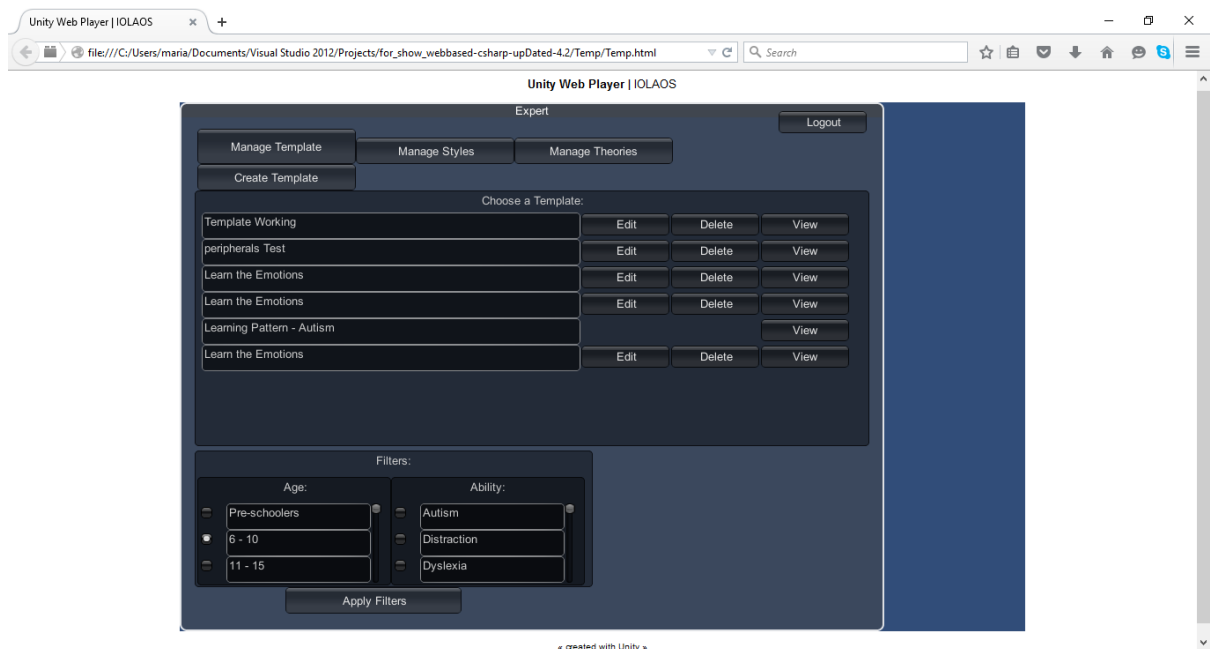


Figure 9 Expert: Start screen

Όπως φαίνεται και στο screenshot παραπάνω το σύστημα διευκολύνει το χρήστη στην πλοήγησή του δίνοντας του τη δυνατότητα να εκτελεί όλες τις ενέργειες επιλέγοντας από λίστες με απλά buttons ή radiobuttons.

## Template Creation

Η δημιουργία ενός προσχέδιου παιχνιδιού, είναι μια διαδικασία που αποτελείται από πολλά διαδοχικά βήματα όπου ο χρήστης ολοκληρώνοντας το κάθε ένα από αυτά προχωράει στο επόμενο όπως τον καθοδηγεί το σύστημα.

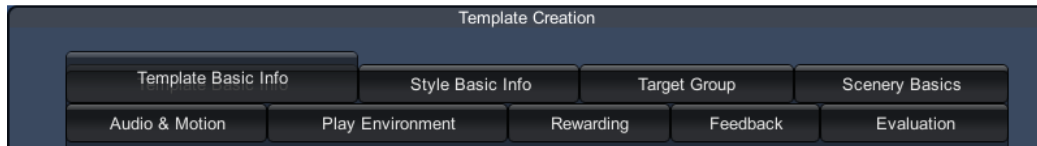


Figure 10 Expert: Template creation

### 1. Template basic info

Στο πρώτο στάδιο ο χρήστης καλείται να δώσει τις βασικές πληροφορίες του προσχέδιου που ετοιμάζει. Αυτές είναι τίτλος του template, μια περιγραφή του, ημερομηνία δημιουργίας, το όνομα του χρήστη καθώς και ποιες θεωρίες χρησιμοποιεί.

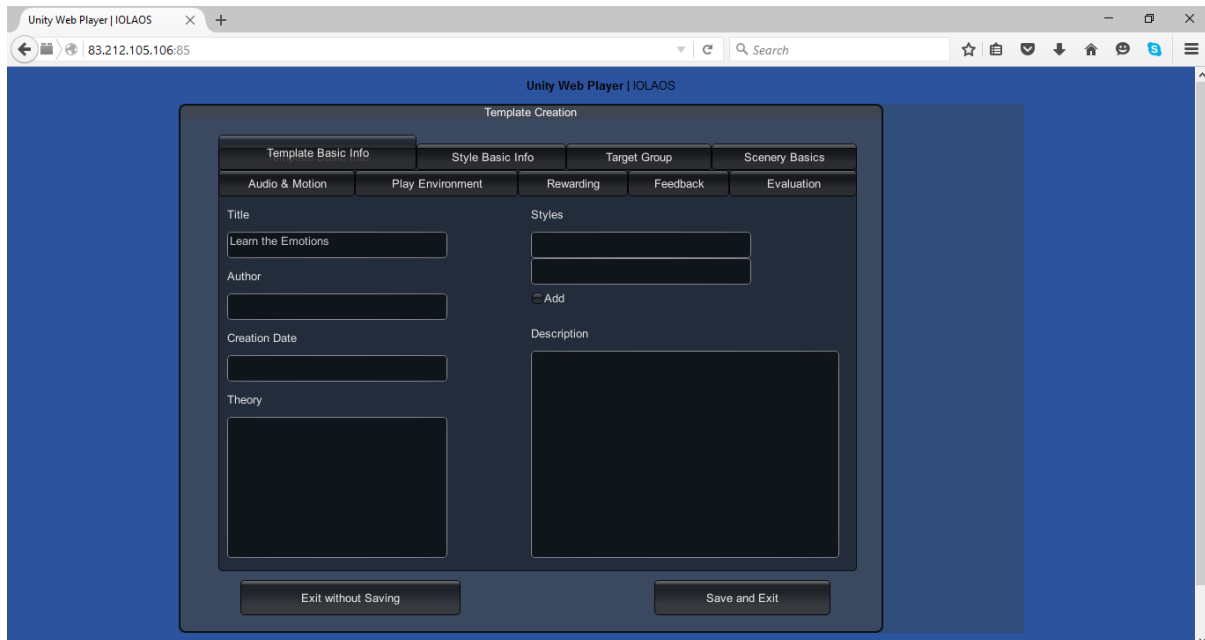


Figure 11 Expert: Template Creation - Basic Info



## 2. Style basic info

Προχωρώντας στο επόμενο βήμα ο χρήστης πρέπει να εισάγει τις διάφορες πληροφορίες για τα στυλ που θα χρησιμοποιήσει στην κατασκευή του προσχεδίου. Αυτές οι πληροφορίες είναι τίτλος, περιγραφή και άλλες λεπτομέρειες.

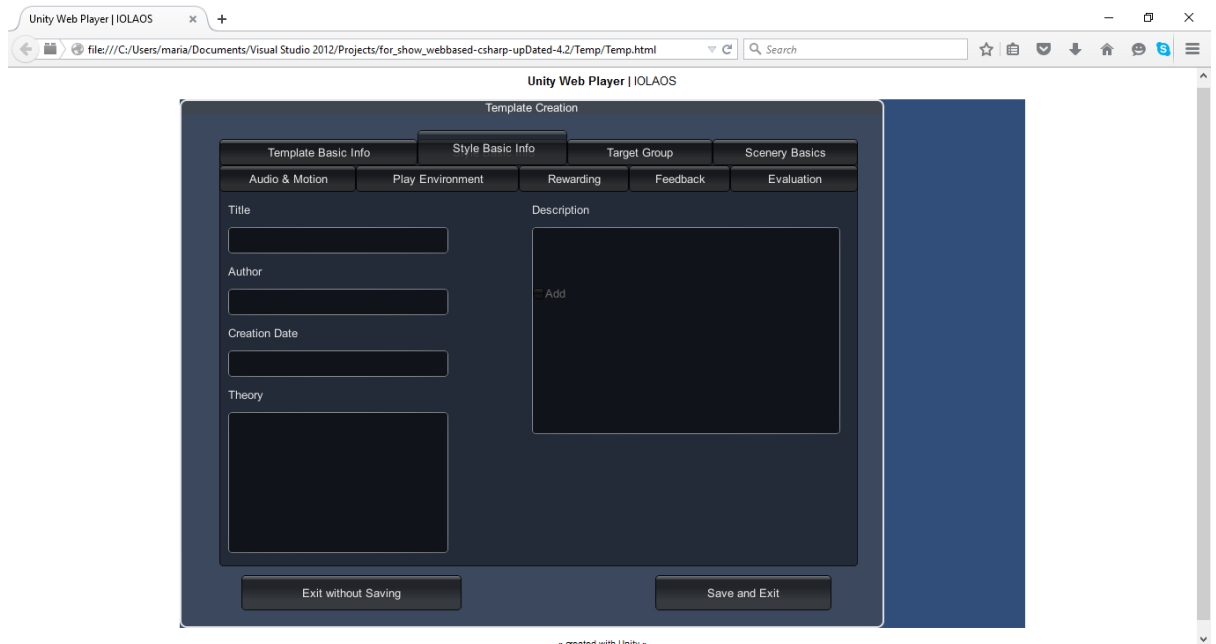


Figure 12 Expert: Template creation - Style info

## 3. Target Group

Στο τρίτο βήμα ο χρήστης καλείται να επιλέξει την ηλικιακή ομάδα ή την τάξη του σχολείου στην οποία απευθύνεται το template που δημιουργεί. Καλείται ακόμα να επιλέξει το αντικείμενο εκμάθησης (π.χ εκμάθηση συναισθημάτων) αλλά και ειδικές ανάγκες ή μαθησιακές δυσκολίες που μπορεί να έχει ο τελικός χρήστης του παιχνιδιού.

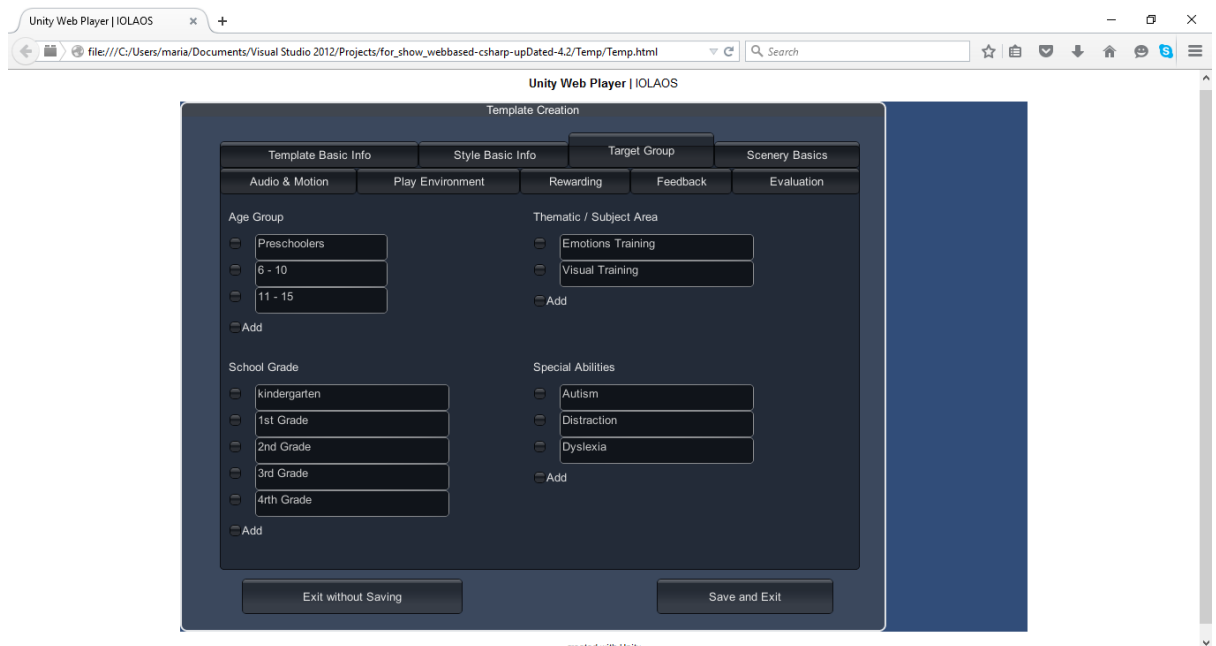


Figure 13 Expert: Template creation – Target group

#### 4. Scenery Basics

Στη συνέχεια γίνονται οι ρυθμίσεις των σκηνών/οθονών του παιχνιδιού. Αυτές αφορούν τον αριθμό των σκηνών που θα χρησιμοποιηθούν, τι χρώματα επιτρέπονται και περιορισμούς/κανόνες στη ροή του παιχνιδιού.

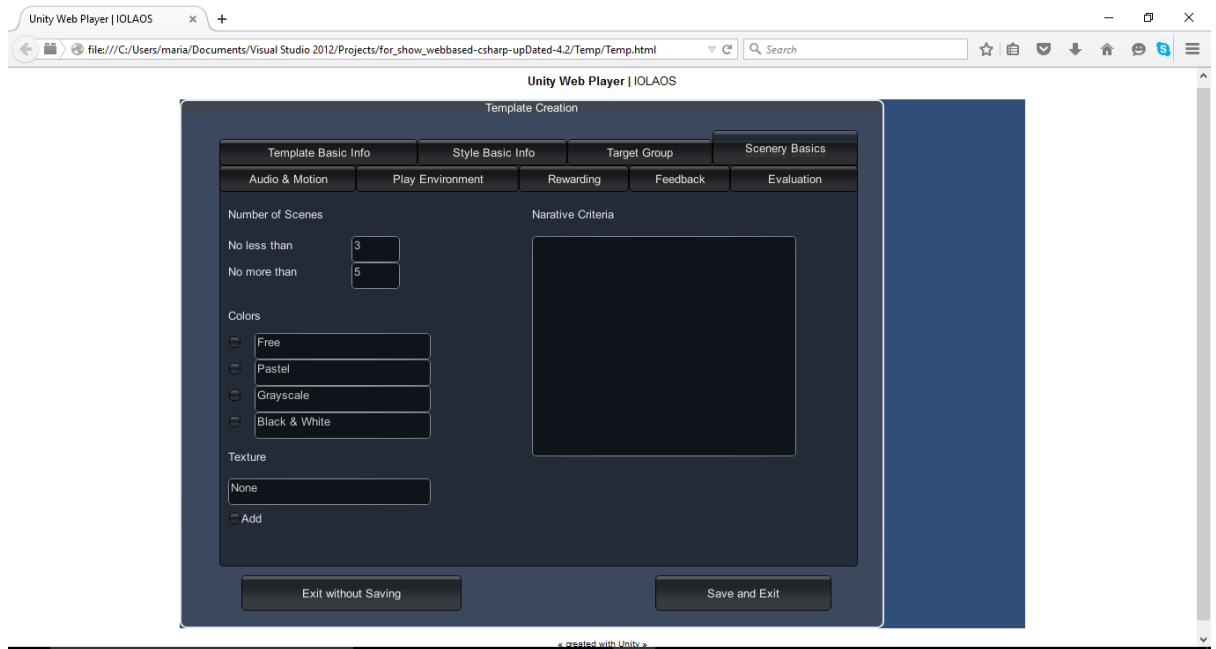


Figure 14 Expert: Template creation - Scenery Basics

#### 5. Audio & Motion

Στο σημείο αυτό ο expert θέτει τους περιορισμούς που έχουν να κάνουν με τους ήχους και τα κινούμενα αντικείμενα που μπορούν να χρησιμοποιηθούν στο παιχνίδι.

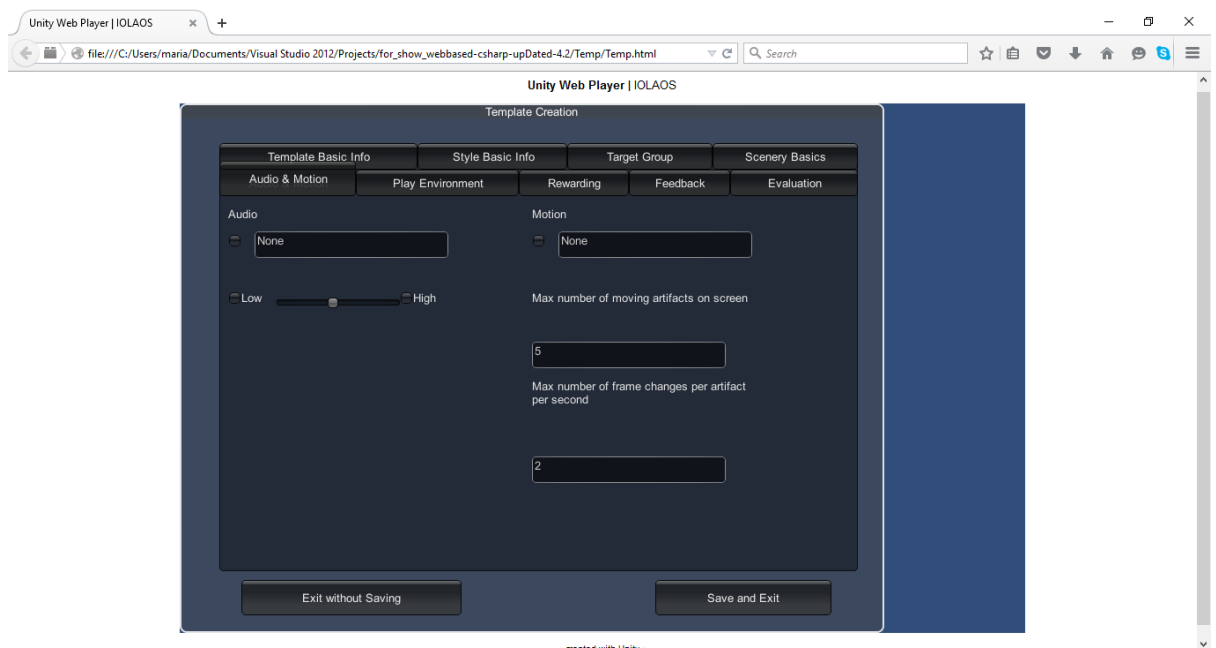


Figure 15 Expert: Template creation - Audio & Motion

## 6. Play environment

Σ' αυτό το στάδιο γίνονται οι ρυθμίσεις που αφορούν στο παίξιμο του παιχνιδιού. Πιο συγκεκριμένα, αν δηλαδή ο παίκτης θα παίξει μόνος του ή καθοδηγούμενος από κάποιον. Άλλες ρυθμίσεις που γίνονται σ' αυτό το στάδιο αφορούν στις περιφερειακές συσκευές που επιτρέπεται να χρησιμοποιηθούν όπως είναι το πληκτρολόγιο, το ποντίκι ή κάποιο άλλο NUI device.

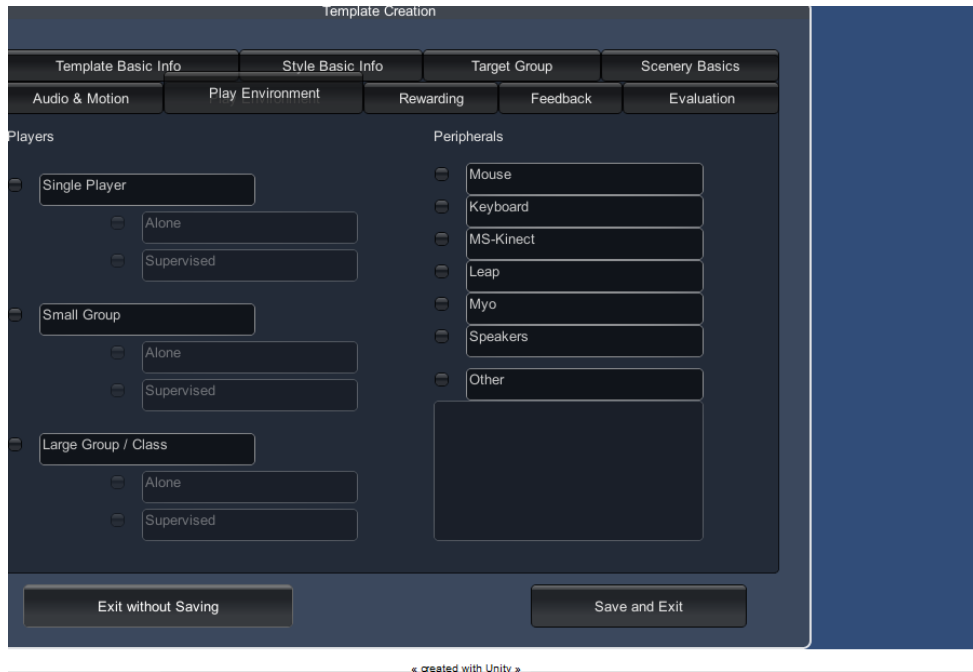


Figure 16 Expert: Template creation - Play environment

## 7. Rewarding

Πλησιάζοντας προς το τέλος, ο ειδικός ψυχολόγος ( Expert ) έχει σχεδόν ολοκληρώσει το προσχέδιο του παιχνιδιού και πρέπει να επιλέξει τον τρόπο με τον οποίο θα ανταμειφθεί ο παίκτης. Η επιλογή πρέπει να γίνει λαμβάνοντας υπόψη την ηλικία του παίκτη και τα ιδιαίτερα χαρακτηριστικά του.

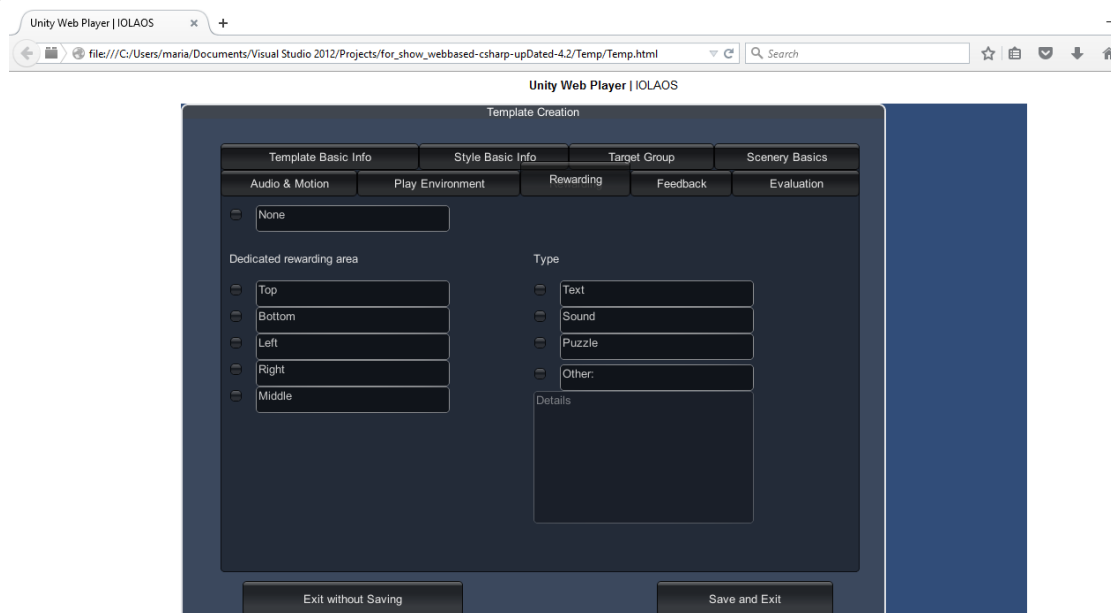


Figure 17 Expert: Template creation - Rewarding

## 8. Feedback

Στο προτελευταίο στάδιο της δημιουργίας του προσχεδίου ο expert επιλέγει τον τρόπο με τον οποίο ο παίκτης θα λαμβάνει ανατροφοδότηση από το παιχνίδι. Το πώς δηλαδή το παιχνίδι θα ανταποκρίνεται ανάλογα στις κινήσεις του παίκτη. Αυτή η ανατροφοδότηση μπορεί να έχει να κάνει είτε με αναδύμενα παράθυρα βοήθειας είτε με προβολή σκορ είτε με ήχους και οπτικά αντικείμενα. Οι επιλογές αυτές παρέχονται από μια λίστα μέσω της οποίας επιλέγει ο expert, παράλληλα όμως του δίνεται η δυνατότητα να δημιουργήσει ένα δικό του τρόπο ανατροφοδότησης.

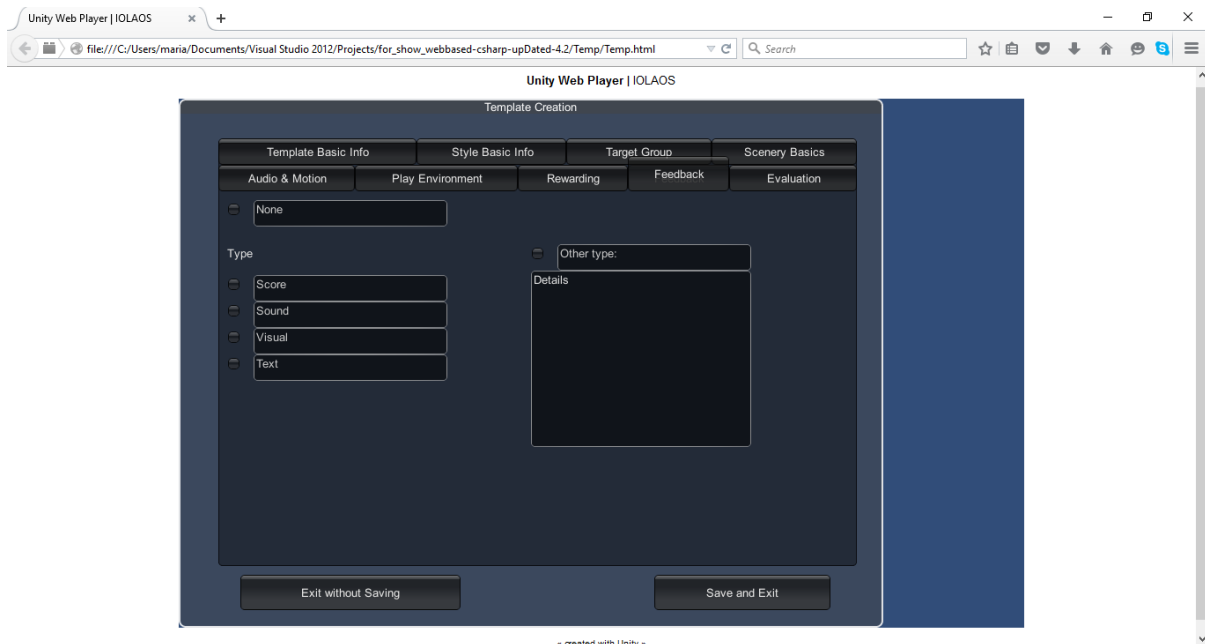
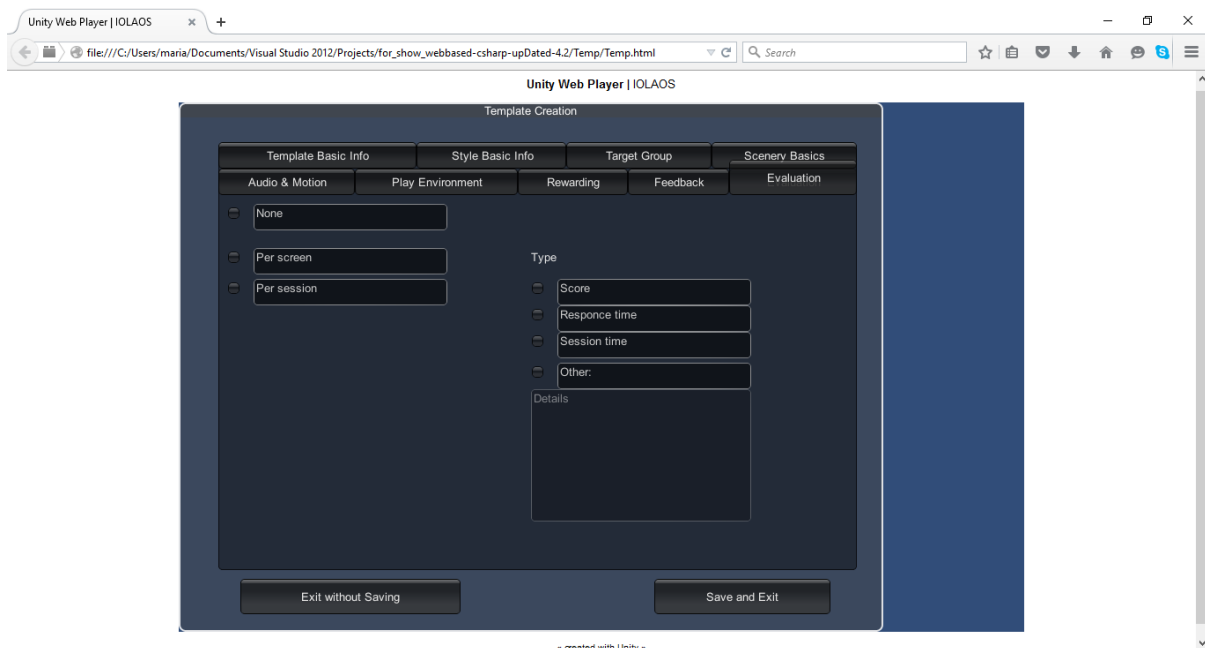


Figure 18 Expert: Template creation - Feedback

## 9. Evaluation

Στο 9<sup>ο</sup> και τελευταίο βήμα ο expert επιλέγει τον τρόπο με τον οποίο θα αξιολογηθεί ο παίκτης. Οι επιλογές που του παρέχει το σύστημα μέσω των διαθέσιμων λιστών είναι αξιολόγηση με σκορ ή χρόνο ανά οθόνη (επίπεδο) του παιχνιδιού ή για τη συνολική συνεδρία. Εκτός όμως από αυτές τις επιλογές ο expert μπορεί να δημιουργήσει ένα δικό του τρόπο για αξιολόγηση του παίκτη.



## Educator

Η παρακάτω εικόνα είναι η οθόνη που βλέπει ο educator όταν κάνει είσοδο στο σύστημα. Σ' αυτήν εμφανίζεται μια λίστα με τα διαθέσιμα παιχνίδια τα οποία μπορεί να τροποποιήσει, να παίξει ή να διαγράψει. Κάτω από τη λίστα με τα παιχνίδια όπως φαίνεται και στο screenshot υπάρχουν radiobuttons, επιλέγοντας τα οποία φιλτράρεται η λίστα με τα διαθέσιμα παιχνίδια.

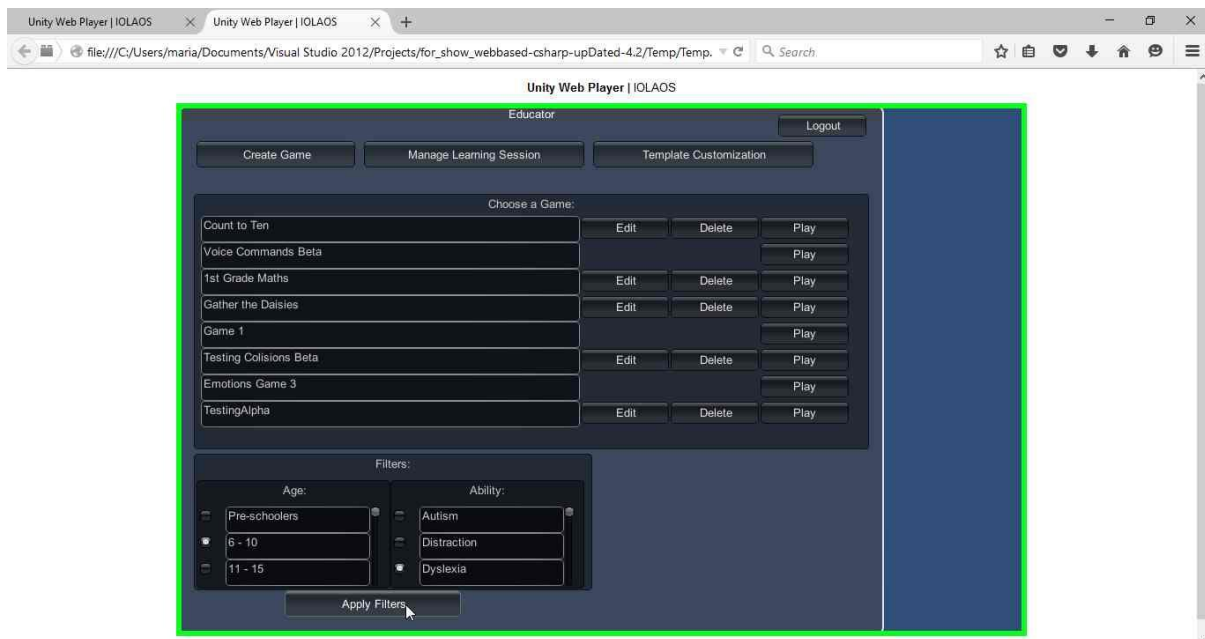


Figure 19 Educator: Start Screen

Οι διαθέσιμες για τον educator ενέργειες είναι οι εξής :

### Game initialisation

#### 1. Επιλογή template.

Ο εκπαιδευτικός επιλέγει ένα από τα διαθέσιμα προσχέδια παιχνιδιών που έχουν κατασκευάσει οι ειδικοί ψυχολόγοι (experts) λαμβάνοντας υπόψη την κατηγορία του παίκτη.

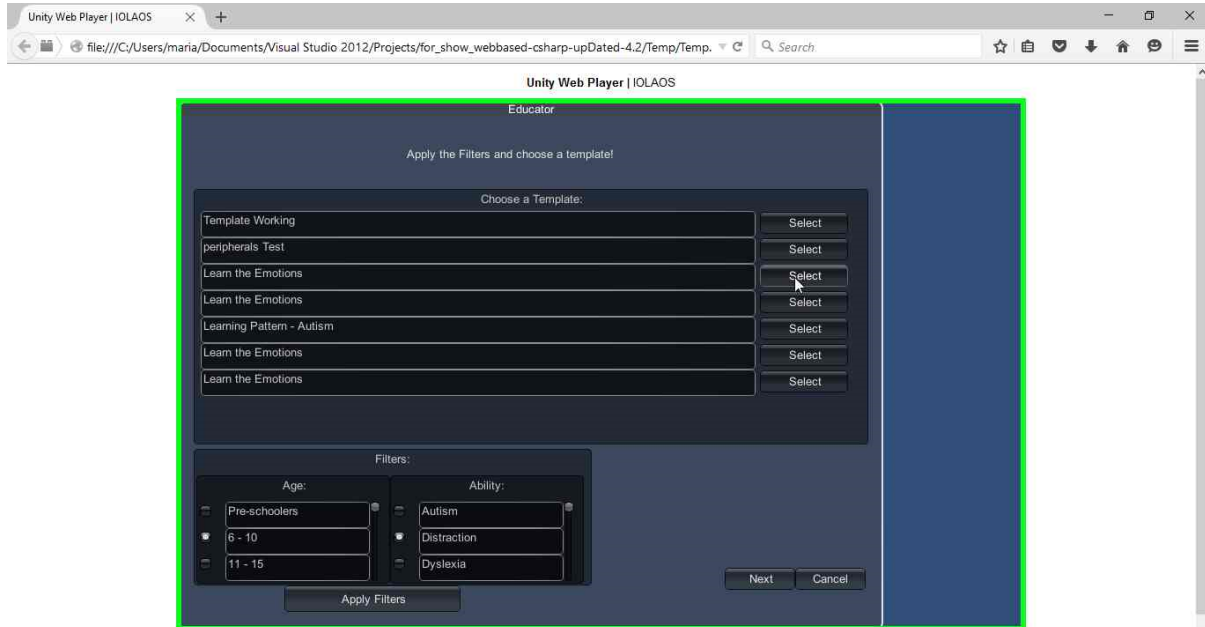


Figure 20 Educator: Template selection

#### 2. Παραμετροποίηση template

Αφού επιλέξει το template ο εκπαιδευτικός παραμετροποιεί τις ρυθμίσεις του στο βαθμό που του επιτρέπει το σύστημα, όπως έχει ορίσει δηλαδή ο expert. Έτσι όπως βλέπουμε μπορεί να αυξήσει ή να μειώσει τον αριθμό των οθονών στα όρια που έχει θέσει ο expert.

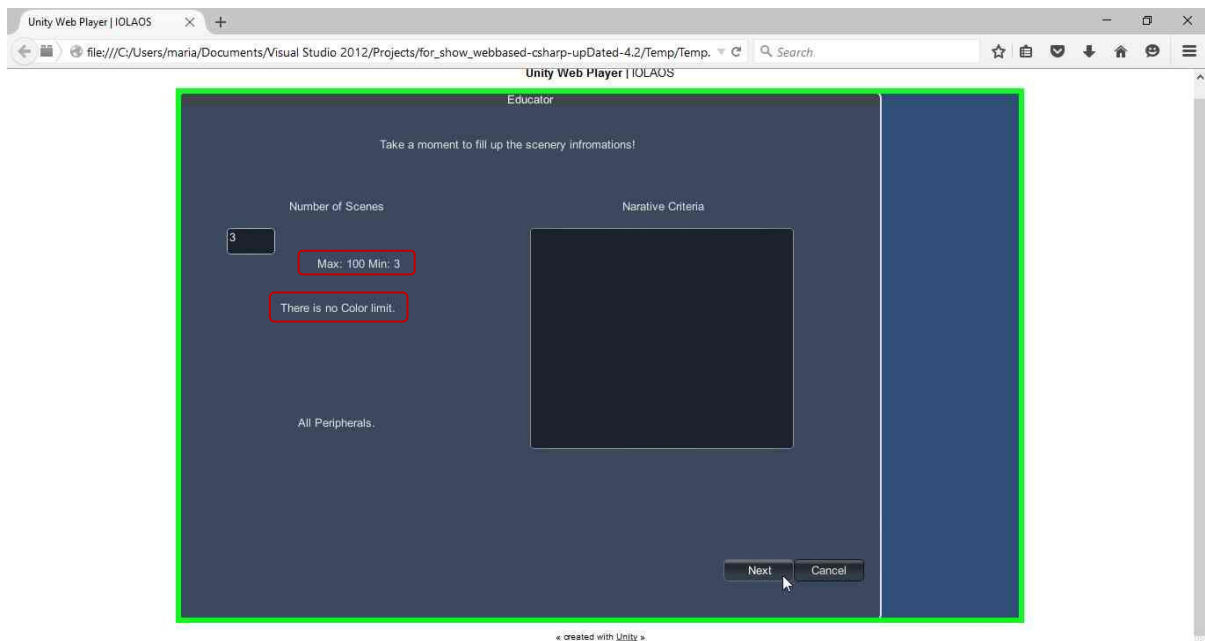


Figure 21 Educator: Template customization

Στο παραπάνω screenshot μπορούμε να παρατηρήσουμε ότι το σύστημα αναφέρει ότι ο μέγιστος αριθμός σκηνών που μπορεί να εισάγει ο educator είναι 100 ενώ ο ελάχιστος 3. Ακόμα ότι δεν υπάρχει όριο στα χρώματα που μπορούν να χρησιμοποιηθούν.

Επιλέγοντας το κουμπί “next” ο educator περνάει στο επόμενο στάδιο.

### 3. Καθορισμός ονόματος και περιγραφής παιχνιδιού.

Στο σημείο αυτό ο educator πραγματοποιεί τις τελικές ρυθμίσεις πριν προχωρήσει στη δημιουργία του παιχνιδιού. Έτσι ο educator δίνει το όνομα και την περιγραφή του παιχνιδιού και θέτει περιορισμούς στο ποιι μπορούν να έχουν πρόσβαση (visibility) στο παιχνίδι του.

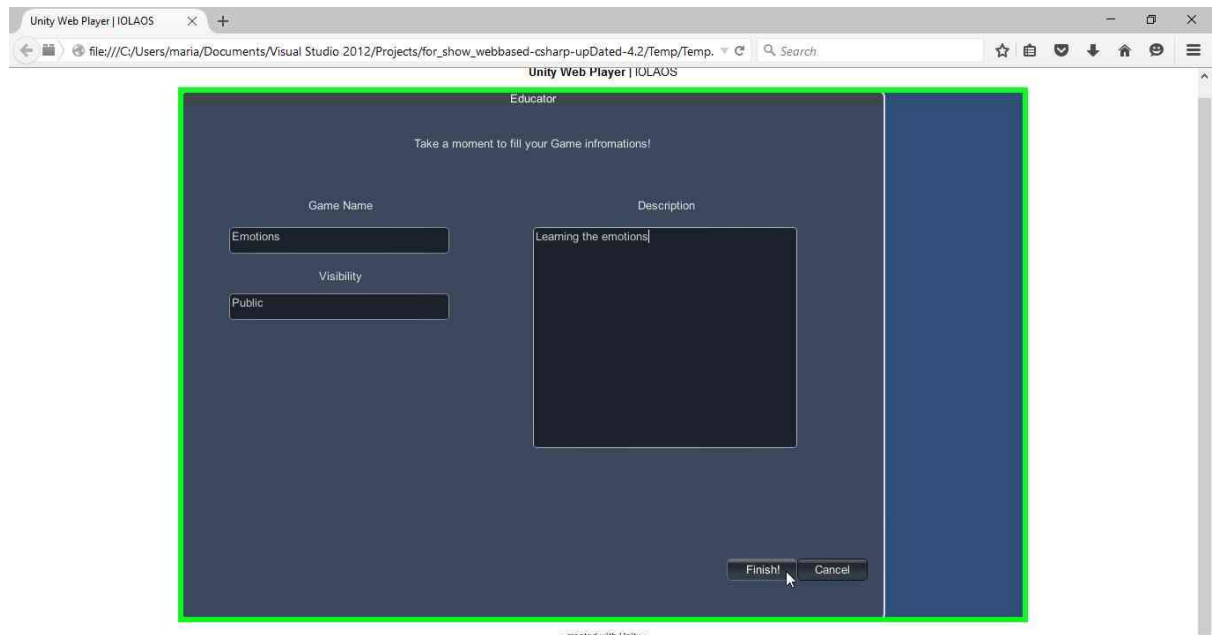


Figure 22 Educator: Game description

## Game Creation

Και αφού ολοκληρώσει τις αρχικές ρυθμίσεις του παιχνιδιού περνάει στην κατασκευή του. Πατώντας το κουμπί “Finish” ο χρήστης μεταφέρεται στην οθόνη κατασκευής του παιχνιδιού δηλαδή στον game editor.

Στην εικόνα 22 βλέπουμε την αρχική οθόνη της κατασκευής του παιχνιδιού. Στο κέντρο βλέπουμε τη σκηνή όπου εκεί θα κατασκευαστεί το παιχνίδι, αριστερά βρίσκονται οι βιβλιοθήκες και κάτω από τη σκηνή βρίσκονται κάποιες εργαλειοθήκες και παράθυρα που παρέχουν πληροφορίες. Εκείνο που αξίζει να τονιστεί είναι ότι η δημιουργία του παιχνιδιού ξεκινάει με 3 σκηνές, όσες δηλαδή ορίσαμε στα προηγούμενα βήματα. Αυτό όμως δεν εμποδίζει το δάσκαλο να προσθέσει τόσες σκηνές όσες του επιτρέπει το συγκεκριμένο template που έχει επιλέξει. Ο τρόπος για να το κάνει αυτό είναι πατώντας το κουμπί “+” που βρίσκεται πάνω από τη σκηνή.

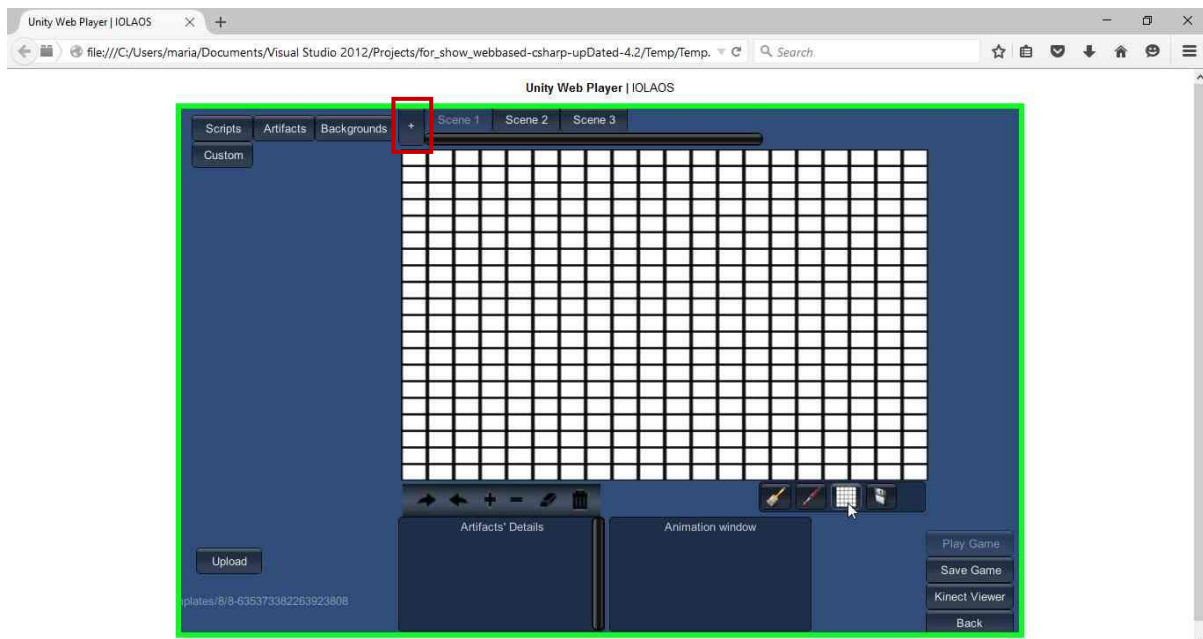


Figure 23 Game Creation : First screen

Ο κώδικας πίσω από αυτό το κουμπί είναι ο εξής:

```
if (GUI.Button(new Rect(0, 35, 30, 45), "+"))
{
    if (sceneCounter < loadedData.getMaxScenes())
    {
        sceneGo = (GameObject)Instantiate(scene, scene.transform.position, scene.transform.rotation);
        sceneCounter++;
    }
}
```

Figure 24 Code: Adding scenes - "+" button

Στην πρώτη σειρά είναι η δήλωση του button και οι διαστάσεις του ενώ στη συνέχεια γίνεται έλεγχος για το αν ο αριθμός των σκηνών δεν ξεπερνάει το επιτρεπόμενο όριο και εφόσον ισχύει η συνθήκη προχωράει στη δημιουργία της νέας σκηνής.

Ακολουθώντας απλά διαδοχικά βήματα ο educator κατασκευάζει το παιχνίδι:

### 1. Επιλογή background:

Επιλέγοντας την καρτέλα background παρατίθενται στο χρήστη οι διαθέσιμες εικόνες για background χωρισμένες σε κατηγορίες. Όπως φαίνεται παρακάτω, οι κατηγορίες αυτές είναι



“Landscapes” και “Plain”. Ο χρήστης επιλέγει και τοποθετεί το background στη σκηνή με drag and drop.

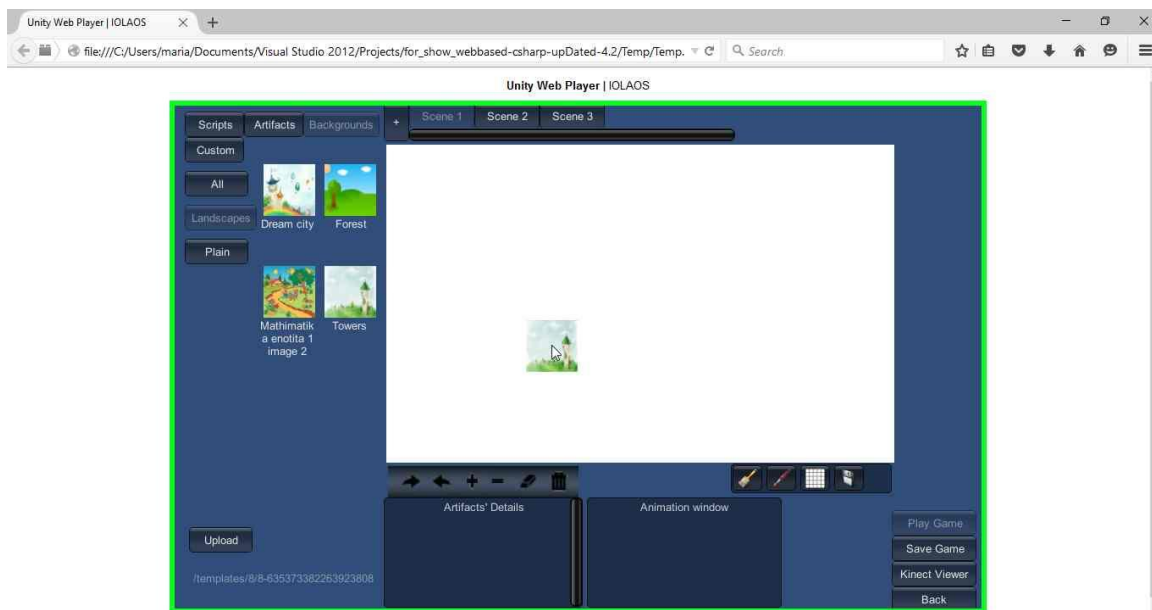


Figure 25 Educator: Game creation - Background selection

## 2. Προσθήκη των Artifacts.

Στο σημείο αυτό και αφού έχει επιλεγεί το background ο χρήστης προσθετει τα artifacts από την καρτέλα “Artifacts”. Τα αντικείμενα δηλαδή στα οποία θα προστεθεί η διάδραση. Τα Artifacts είναι καταναμημένα σε κατηγορίες, μερικές από τις οποίες είναι αυτές “People”, “Numbers”, “Shapes”. Με τον ίδιο τρόπο που περιγράψαμε παραπάνω δηλαδή με drag and drop ο χρήστης βάζει τα αντικείμενα στη σκηνή του παιχνιδιού.

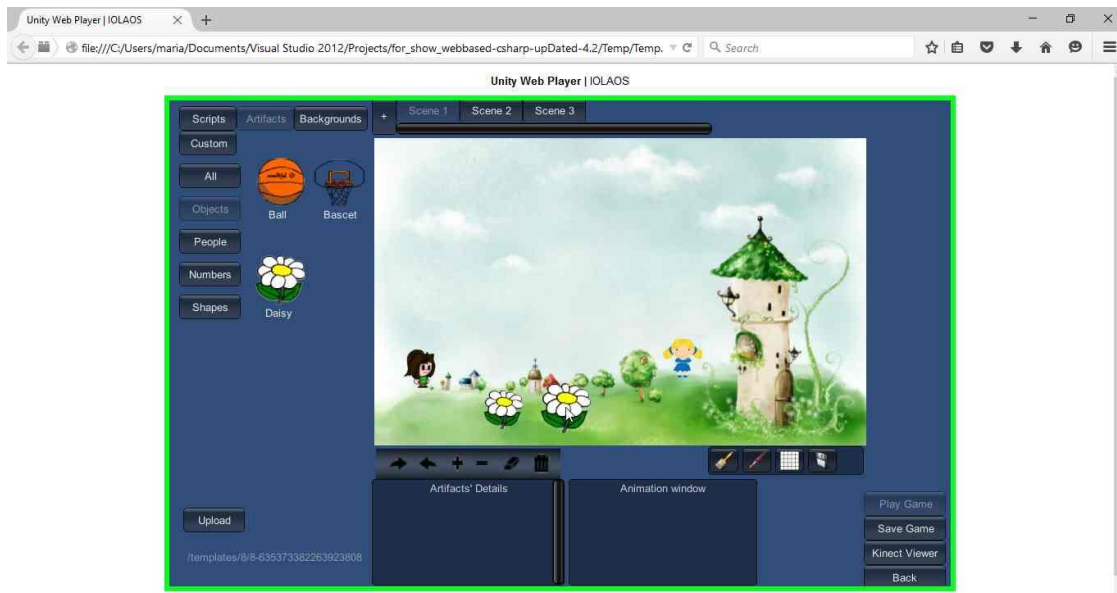


Figure 26 Educator: Game creation - Artifacts Selection

## 3. Προσθήκη των scripts.

Αφού ολοκληρώσει το σχεδιασμό του παιχνιδιού ο χρήστης προσθέτει τα scripts για να δώσει διάδραση στο παιχνίδι. Τα Scripts τοποθετούνται με τον ίδιο τρόπο με τα αντικείμενα που

περιγράψαμε ήδη. Δηλαδή από την καρτέλα Scripts ο χρήστης επιλέγει την επιθυμητή διάδραση για το κάθε artifact και με drag and drop εφαρμόζεται πάνω του.

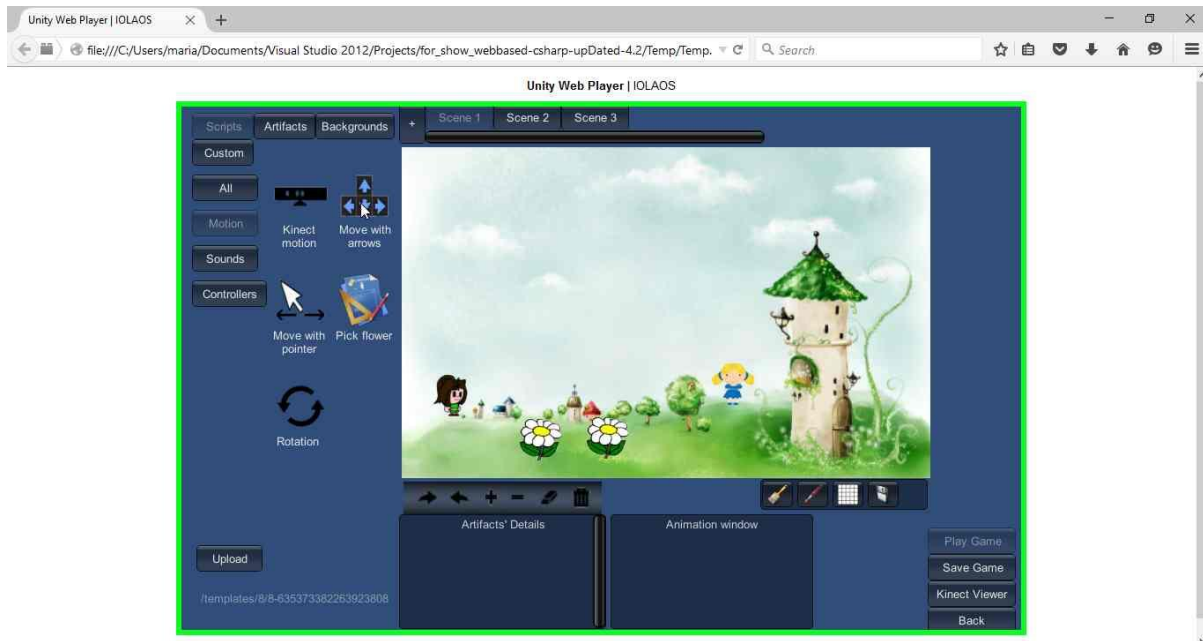


Figure 27 Educator: Game creation - Scripts Selection

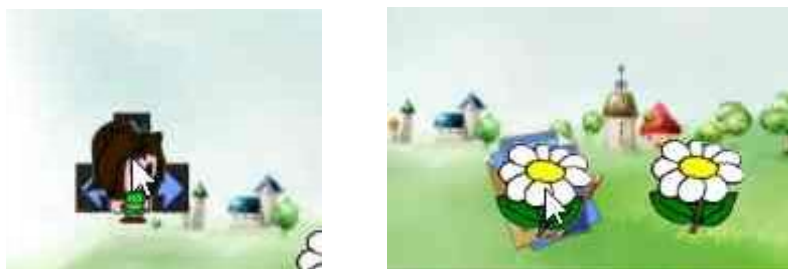


Figure 28 Add scripts: Drag and Drop

Στον κώδικα παρακάτω βλέπουμε πως προστίθεται το εκάστοτε script στο artifact. Στη συγκεκριμένη περίπτωση πρόκειται για το script “Move girl” και όπως φαίνεται, γίνεται με τον εξής τρόπο: στην πρώτη γραμμή γίνεται έλεγχος για τον αν το object που περιέχει το script βρίσκεται εντός των ορίων της θέσης του artifact. Αν η συνθήκη ικανοποιείται τότε το script προστίθεται στο artifact και στη συνέχεια αποθηκεύεται στη λίστα που περιέχει τις πληροφορίες του artifact.

```
if (Vector3.Distance(final_pos, sprite.transform.position) < 0.8)
{
    Debug.Log(sprite.name);
    GameObject.Find("Main Camera").GetComponent<game_editor>().scene_objects.getSingleObject(sprite.name).setComponent("MoveGirl");
    GameObject.Find("Main Camera").GetComponent<game_editor>().scene_objects.getSingleObject(sprite.name).setTag("main_actor");
}
```

Figure 29 Code: Drop script on artifact

## Artifacts Details:

Όταν ο χρήστης επιλέξει κάποιο από τα αντικείμενα στην οθόνη τότε το αντικείμενο γίνεται highlighted και ενεργοποιείται το πεδίο “Artifacts’ Details”. Στο πεδίο αυτό ο χρήστης βλέπει τις ιδιότητες των αντικειμένων “Position”, “Name”, “Rotation” καθώς και τα scripts που φέρουν.



Figure 30 Game editor options: Artifacts' Details

Οι ιδιότητες αυτές μπορούν να τροποποιηθούν από αυτό το πεδίο. Στα screenshots παρακάτω βλέπουμε ότι υπάρχει ένα button “x” με το οποίο ο χρήστης μπορεί εύκολα και γρήγορα να διαγράψει τα scripts που έχουν τα αντικείμενα.

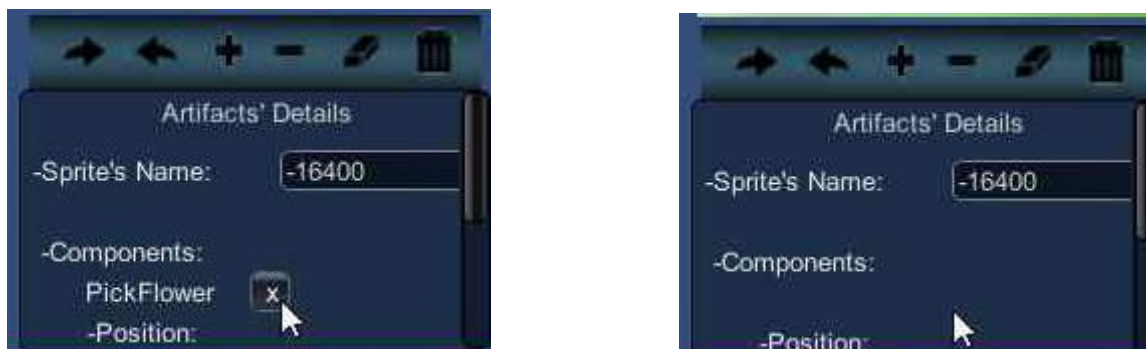


Figure 31 Game editor options: Delete Artifact's script

Ακολουθεί το screenshot με τον κώδικα για το button “x”. Στην πρώτη γραμμή γίνεται η δημιουργία του button ενώ στην δεύτερη γίνεται η ανάκληση του script από το artifact και η διαγραφή του.

```
if (GUI.Button(new Rect(115, (90 + (j * 20)), 20, 20), "x"))  
{  
    scene_objects.getSingleObject(InteractWithGo.clickedSprite).removeComponent(InteractWithGo.components[j]);  
}
```

Figure 32 Code: Delete artifact's script

## Toolbars

Ο game editor παρέχει δυο ειδών toolbars. Η μια είναι αυτή που φαίνεται στα ακριβώς από κάτω screenshots. Η εργαλειοθήκη αυτή περιλαμβάνει 6 λειτουργίες. Έχοντας επιλεγμένο ένα artifact που βρίσκεται στη σκηνή μπορούμε με τη βοήθεια της εργαλειοθήκης να εκτελέσουμε τις παρακάτω ενέργειες :

- Rotate Right: περιστροφή του artifact προς τα δεξιά
- Rotate Left: περιστροφή του artifact προς τα αριστερά
- Resize (maximize): μεγέθυνση του artifact
- Resize (minimize): σμίκρυνση του artifact
- Default settings: επαναφορά του artifact στις αρχικές του ρυθμίσεις
- Delete: διαγραφή του artifact.

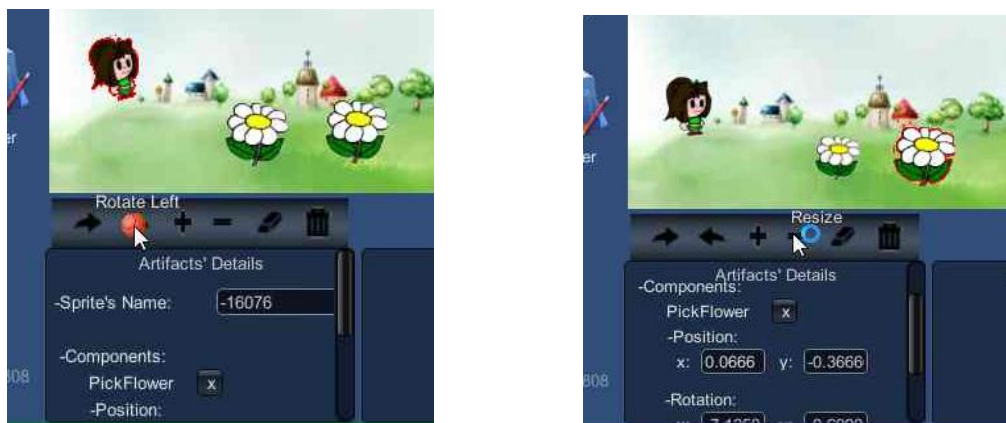


Figure 33 Game editor options: Toolbars – Rotate / Resize artifact



Figure 34 Game editor options: Toolbars - Delete artifact

Ακολουθεί ενδεικτικό κομμάτι κώδικα για να δούμε πως γίνονται κάποιες από αυτές τις λειτουργίες:

```
case "rotateLeft":
    current.transform.Rotate(Vector3.up, -150 * Time.deltaTime);
    try
    {
        current.GetComponent<SceneObj>().temp_obj.set_Rotation(current.transform.rotation);
        GameObject.Find("Main Camera").GetComponent<game_editor>().saveScene();
    }
}
```

Figure 35 Code: Rotate Artifact

Στην πρώτη γραμμή γίνεται έλεγχος για το ποιο κουμπί έχει πατηθεί και εφόσον ισχύει το rotate left, στη δεύτερη σειρά το επιλεγμένο artifact περιστρέφεται. Στις επόμενες γραμμές κώδικα αποθηκεύεται η νέα θέση του artifact στη λίστα και στη σκηνή.

Η δεύτερη εργαλειοθήκη δεν έχει να κάνει με την επεξεργασία των artifacts αλλά με ρυθμίσεις που αφορούν τη σκηνή.

Πατώντας το κουμπί που μοιάζει με πλέγμα στη σκηνή δημιουργείται ένα πλέγμα.



Figure 36 Second toolbar

Κάθε τετραγωνάκι στο πλέγμα είναι μια ενεργή περιοχή. Επιλέγοντας μερικά από αυτά ο χρήστης μπορεί να προσδώσει extra διαδραστικότητα στο background. Για παράδειγμα μπορεί να δημιουργήσει ενεργές ή μη περιοχές που όταν το artifact πέσει πάνω αλληλεπιδρούν με τον τρόπο που επιλέγει εκείνος.

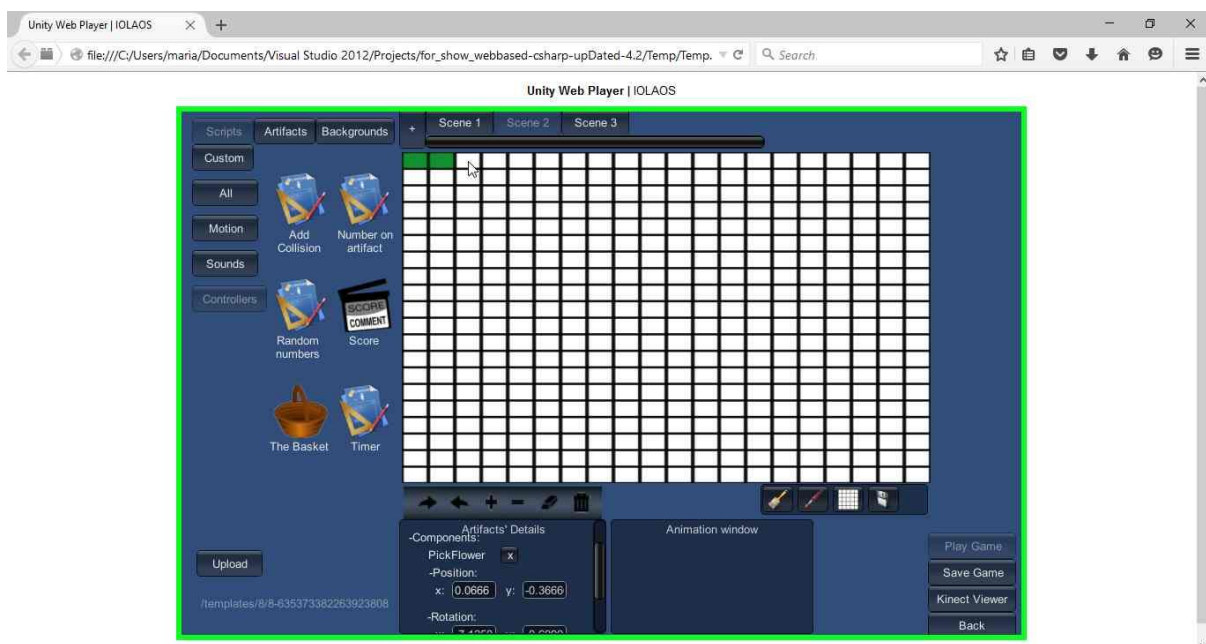


Figure 37 Grid Option

## Administrator

Οι administrators έχουν τη δυνατότητα να πραγματοποιούν όλες τις ενέργειες όλων των χρηστών. Επιπλέον όμως ο administrator μπορεί να επιλέξει ανάμεσα σε μια πληθώρα άλλων ενεργειών. Παρακάτω θα δούμε μόνο τις ενέργειες που δεν μπορούν να κάνουν οι υπόλοιποι χρήστες.

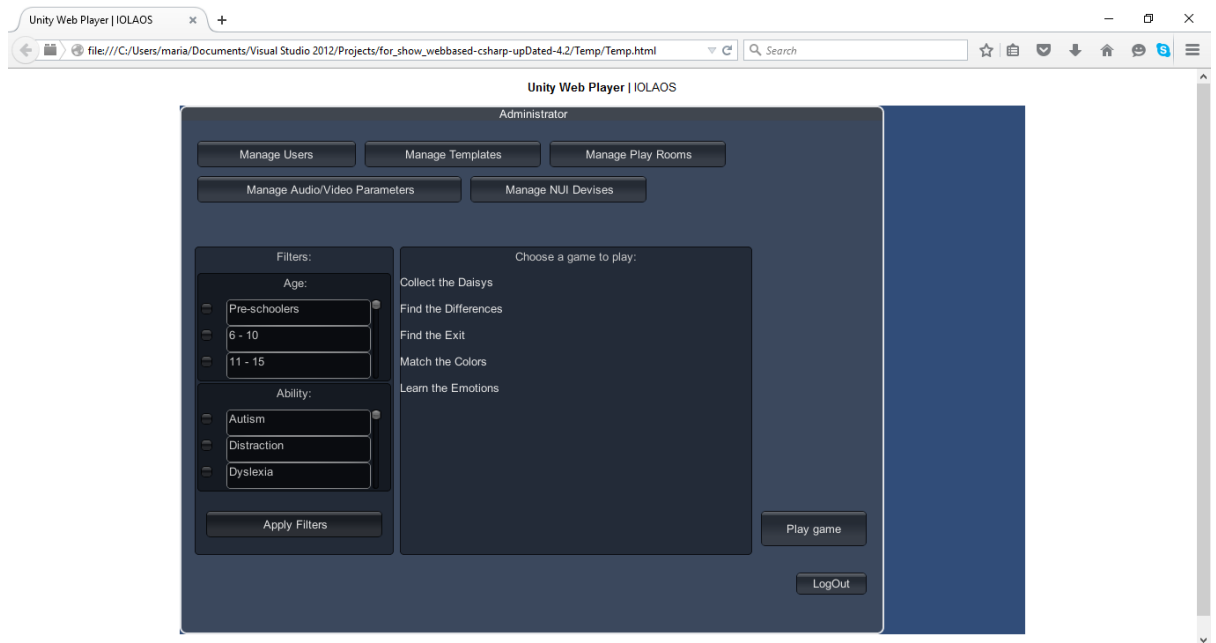


Figure 38 Administrator: Start screen

Όπως φαίνεται και στην εικόνα , κάποιες από αυτές τις ενέργειες είναι : “Manage users” όπου διαχειρίζεται τους χρήστες .Όταν δηλαδή ένας χρήστης κάνει register, πρέπει ο administrator να τον αποδεχτεί προκειμένου να μπορέσει να πραγματοποιήσει είσοδο στο σύστημα. Μπορεί επίσης να διαγράψει χρήστες.

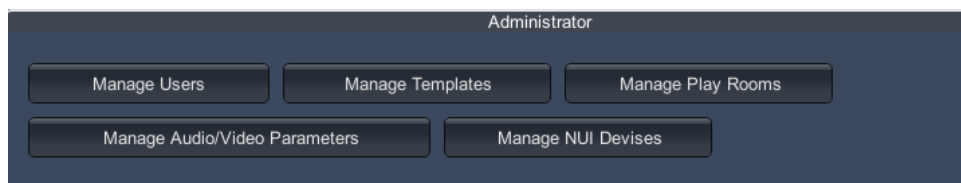


Figure 39 Administrator: Menu

---

### 5.4.1 Hardware & Software requirements

- Για να παίξει κάποιος το παιχνίδι μας χρειάζεται τα εξής<sup>26</sup>:

Desktop:

- OS: Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+
- Κάρτα γραφικών: DX9 (shader model 2.0) capabilities; γενικά ό,τι κατασκευάστηκε από το 2004 και μετά μπορεί να δουλέψει.
- CPU: SSE2 instruction set support.
- Web Player (deprecated): απαιτεί browser που υποστηρίζει plugins, όπως ο IE, ο Safari και μερικές εκδόσεις του Firefox
- iOS: απαιτεί iOS 6.0 ή νεότερο.

Αν χρησιμοποιηθεί Kinect για το παιχνίδι απαιτείται:

- 32-bit (x86) or 64-bit (x64) processors
- Dual-core, 2.66-GHz or faster processor
- USB 2.0 bus αφιερωμένη στο Kinect
- 2 GB RAM
- Graphics card that supports DirectX 9.0c
- Microsoft Kinect for Windows Sensor

- Για την ανάπτυξη αντίστοιχων παιχνιδιών απαιτούνται τα εξής:

Unity game engine

- OS: Windows 7 SP1+, 8, 10; Mac OS X 10.8+.  
Windows XP & δεν υποστηρίζονται και οι εκδόσεις server των Windows & OS X δεν έχουν δοκιμαστεί.
- GPU: Κάρτα γραφικών DX9 (shader model 2.0) capabilities.

Additional platform development requirements:

iOS: Mac computer running minimum OS X 10.9.4 version and Xcode 6.x.

- Android: Android SDK and Java Development Kit (JDK).
- Windows 8.1 Store Apps / Windows Phone 8.1: 64 bit Windows 8.1 Pro and Visual Studio 2013 Update 2+.
- WebGL: Mac OS X 10.8+ or Windows 7 SP1+ (64-bit editor only)



---

## 6. Αποτελέσματα

### 6.1. Συμπεράσματα

Από όσα αναλύθηκαν παραπάνω γίνεται εύκολα αντιληπτό ότι είναι επιτακτική ανάγκη να εισέλθουν νέοι τρόποι διδασκαλίας στον τρόπο με τον οποίο διεξάγεται το μάθημα μέσα σε μια σχολική τάξη ώστε αφενός να εξάψουν το ενδιαφέρον του μαθητή και αφετέρου να οδηγήσουν στην πραγματοποίηση ενός συμμετοχικού σχολείου.

Η εργασία μας επιχειρεί να πραγματοποιήσει τους δυο παραπάνω στόχους με την ενσωμάτωση του παιχνιδιού στην εκπαιδευτική διαδικασία. Το παιχνίδι αποτελεί για τα παιδιά έναν τρόπο δόμησης της σκέψης τους, ανάπτυξης της φαντασίας τους και διεύρυνσης των ορίων τους. Παρόλα αυτά η πληθώρα των παιχνιδιών που προσφέρονται ειδικά με τη χρήση νέων τεχνολογιών, συχνά δυσκολεύει τον εκπαιδευτικό όσον αφορά στην επιλογή του καταλληλότερου.

Η παρούσα εργασία προσφέρει ένα συνδυασμό της εκμάθησης με το παιχνίδι μέσω του ηλεκτρονικού υπολογιστή παρέχοντας την κατάλληλη υποστήριξη τόσο στον εκπαιδευτικό που κατασκευάζει τα παιχνίδια όσο και στα παιδιά που τα παίζουν. Ένας από τους πρωταρχικούς στόχους της εργασίας είναι να παρέχει εύχρηστο περιβάλλον στο χρήστη και να τον διευκολύνει στην πλοήγηση. Ταυτόχρονα επιχειρείται η εξατομικευμένη δημιουργία παιχνιδιού λαμβάνοντας υπόψη τον κάθε τύπο μαθητή, τις ιδιαιτερότητές και τις ειδικές ανάγκες που αντιμετωπίζει κάνοντας έτσι ένα βήμα πιο κοντά στην επίτευξη της συμμετοχικής εκπαίδευσης.

### 6.2. Μελλοντική εργασία και αποτελέσματα

Βλέποντας συνολικά την πορεία της εργασίας, μας ήταν ξεκάθαρο ότι το unity μας προσέφερε πολλά πλεονεκτήματα στη δημιουργία αυτού που είχαμε ως στόχο αλλά και πολλά μειονεκτήματα. Λόγω του ότι η εργασία κατασκευάστηκε εξολοκλήρου στο unity, με τις περισσότερες οθόνες να μην έχουν σχέση με το παιχνίδι αλλά να είναι απλές διεπαφές, καταλήξαμε στο συμπέρασμα ότι το unity δεν ενδείκνυται για κατασκευή διεπαφών, τουλάχιστον όχι για αυτό που επιχειρήσαμε εμείς να κάνουμε. Η διαδικασία αυτή στο unity είναι πολύ πιο περίπλοκη από ότι θα ήταν αν χρησιμοποιούσαμε διαφορετικές τεχνολογίες. Έτσι συνοψίζοντας θα μπορούσαμε να αναφέρουμε τα εξής ως μελλοντική εξέλιξη-διόρθωση της εργασίας:

- Εκτέλεση των διεπαφών με διαφορετική τεχνολογία (πχ html5) και το unity να χρησιμοποιείται μόνο κατά τη διαδικασία που ο χρήστης παίζει το παιχνίδι.
- Επέκταση του συστήματος ώστε να λαμβάνει υπόψη περισσότερες μαθησιακές και μη δυσκολίες πχ άλλα είδη αυτισμού ή ειδικές ανάγκες.
- Χρήση περισσότερων NUI συσκευών εκτός από το Kinect πχ leap, myo.
- Επέκταση του συστήματος ώστε να μπορεί να παίζει και σε tablets ή smartphones.



---

## Παράρτημα

### Ενδεικτικό κομμάτι Κώδικα

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using System.Xml;
using System.Xml.Serialization;
using System.IO;
using System.Text;
using System.Data;
using System;
using System.Net;

public class game_editor : MonoBehaviour
{
    string path;
    public GUISkin thisOrangeGUISkin;
    TabProperties btn;
    private List<List<TabProperties>> tabs_list = new List<List<TabProperties>>();
    // lista pou periexei oles tis listes me tabs tou screen
    private List<List<TabProperties>> subTabsList = new List<List<TabProperties>>();
    // lista pou periexei oles tis listes me tabs tou screen
    private List<int> disabled_buttons = new List<int>();
    // lista pou periexei ta disapled buttons tis ka8es listas me tabs
    private List<int> subDisabled_buttons = new List<int>();
    // lista pou periexei ta disapled buttons tis ka8es listas me tabs
    int c, y, k = 0, z, left, top, sceneCounter = 0;
    private int selectedTab, selectedSubTab, selectedLibraryTabid;
    string gameName = "Trial";
    // Tabs kai subtabs
    Dictionary<String, List<String>> objectCategories = new Dictionary<String,
    List<String>>() { { "Scripts", new List<String>() { "All", "Motion", "Sounds",
    "Controllers" } }, { "Artifacts", new List<String>() { "All", "Objects", "People",
    "Numbers", "Shapes" } }, { "Backgrounds", new List<String>() { "All",
    "Landscapes", "Plain" } }, { "Custom", new List<String>() { "Artifacts",
    "Backgrounds" } } };
    // an 8eloume h lista na exei proka8orismena tabs ta bazoume se mia lista string
    ExpertXML loadedData;
    //sth me8odo loadData fortwnei ola ta dedomena me tis odhgies pou exei dwsei o expert
    // Save work
    public ObjectsContainer scene_objects = new ObjectsContainer();
    string _FileLocation, _FileName, _TemplateFile;
    // Listes twv artifacts kai plhroforiwn tous
    private List<string> texturePaths = new List<string>();
    public Dictionary<string, string> artifactInfo, scriptInfo, backgroundInfo,
    shapeInfo = new Dictionary<string, string>();
    public List<LibraryObject> libraryObjects = new List<LibraryObject>();
    public List<string> artifactsNames = new List<string>();
    public bool showLabel = false;
    public bool nextline = false;
    public GameObject mother_cube;
    Texture2D temptexture;
    public GameObject sceneGo;
    public Dictionary<String, sceneProp> allScenes = new Dictionary<String,
    sceneProp>();
    //dictionary pou sth mia 8esh tou exei to onoma ths sknhs kai sthn allh th
    sknh(gameobject) kai kapoies plhrofories gi auth.
    public Dictionary<String, List<gameEditorObject>> goPerScene = new
    Dictionary<String, List<gameEditorObject>>();
```

---

```

//dictionary pou sth mia 8esh tou exei to onoma ths skhnhs kai sthn allh ta
gameobjects pou exoume valei mesa s'auth th sknhh

public List<gameEditorObject> goObject = new List<gameEditorObject>();
public sceneProp sceneprop = new sceneProp();
public GameObject scene;
public GameObject goByname;

//GUI vectors
Rect left_box = new Rect(20, 100, Screen.width / 3, Screen.height - 150);
private List<Rect> btnRects = new List<Rect>();
private Rect windowRect = new Rect(0, 0, Screen.width, Screen.height);
private Rect leftGroupRect, rightGroupRect, leftBoxR, rightBoxr;
private int alignSubTabs = 0;
bool loadedDat = false;

//DEBUGGING OPTIONS
private static bool DEBUG_MODE = true; //For debugging make false before build
private string DEBUG_LOG;

public string activeScene;
private string selectedTabName, selectedLibraryTab;
private bool showSubTabs = false;
bool playButtonEnabled = false;

bool showPopUp = false, firstTime = true;
private string gameName2 = "";
//periexei to input string pou tha dwsei o xristis kata to save game
//bool for kinect viewer state toggle on GUI
private bool toggleTxt = false;

//-----Image upload -----
string s1;
byte[] basearray;
Texture2D tex;
string imageName;
string imageVisibility;
string imageCategory;
List<Texture2D> CustomTextures = new List<Texture2D>();
CustomImageList CustomArtifacts = new CustomImageList();
//-----Image Upload-----

//-----TRIAL AREA-----
public int alignInspector = 0;
private static Dictionary<string, List<string>> spriteComponents = new
Dictionary<string, List<string>>();

private string xPos = "0.0", yPos = "0.0", xRot = "0.0", yRot = "0.0";
private Rect rightArea, toolbox;
private Color startcolor;

public Vector2 scrollPosition = Vector2.zero;
public Vector2 tabScrollPosition = Vector2.zero;
private int tabScrollbarSize = 386;
private float detailScrollbarSize = 0.01f;
//-----TRIAL AREA-----

```

---

```

// Use this for initialization
void Start()
{
    //Save filepath and name
    //_FileLocation = Application.dataPath + "/temp/";
    _FileLocation = "http://83.212.105.106:85";
    _FileName = "GameInfo.xml";
//to xml apo to expert menu (oi "odhgies" pou exei dwsei o expert gia ta paixnidia)
    _TemplateFile = "templateInfo.xml";
    if (DEBUG_MODE)
    {

        StartCoroutine(GetFileLocation());
    }
    else
    {
        loadedData = SettingsHandler.getGameTemplate();
    }
//tabs start here
    tabs_list.Add(new List<TabProperties>());
//pros8etoume oses listes me tabs 8eloume (deksia me scenes)
    tabs_list.Add(new List<TabProperties>());
//pros8etoume oses listes me tabs 8eloume (aristera me libraries)
    disabled_buttons.Add(0);
// arxikopoioume ta disabled buttons
    disabled_buttons.Add(0);
    scene = GameObject.Find("Scene");
//Vriskei th sknh pou uparxei hdh (cube)
    goByname = new GameObject();

    for (int i = 0; i < objectCategories.Count; i++)
    {
        subTabsList.Add(new List<TabProperties>());
        subDisabled_buttons.Add(0);
    }

    foreach (var pair in objectCategories)
    {
        TabProperties button = new TabProperties(c, true, pair.Key);
        tabs_list[1].Add(button);
        for (int i = 0; i < pair.Value.Count; i++)
        {
            TabProperties button1 = new TabProperties(c, true, pair.Value[i]);
            subTabsList[c].Add(button1);
        }
        c++;
    }

//rects of groups
    leftGroupRect = new Rect(windowRect.x + 15, windowRect.y + 10,
(windowRect.width / 3), windowRect.height - 100);
    //float rbcwidth = windowRect.width - (leftGroupRect.x + leftGroupRect.width)
- 35;
    rightGroupRect = new Rect(250, -35, 520, 300);//height 30
//rects of boxes
    leftBoxR = new Rect(0, 0, leftGroupRect.width, leftGroupRect.height);
    rightBoxr = new Rect(0, 0, rightGroupRect.width, rightGroupRect.height);
    SettingsHandler.generateUniqueTime(); //Generating the unique time.

```

---

```

if (!DEBUG_MODE)
{
    loadData();
}
//Diavasma vivlio8hkhs
texturePaths.Add("Artifacts/Objects");
texturePaths.Add("Artifacts/People");
texturePaths.Add("Artifacts/Numbers");
texturePaths.Add("Artifacts/Shapes");
artifactInfo = loadTextures(texturePaths);
texturePaths.Clear();

texturePaths.Add("Scripts/Controllers");
texturePaths.Add("Scripts/Motion");
texturePaths.Add("Scripts/Sounds");
scriptInfo = loadTextures(texturePaths);
texturePaths.Clear();

texturePaths.Add("Backgrounds/Landscapes");
texturePaths.Add("Backgrounds/Plain");
backgroundInfo = loadTextures(texturePaths);
texturePaths.Clear();

foreach (var pair in backgroundInfo)
{
    // Debug.Log("prwth..." + pair.Key);
}

if (tabs_list[0].Count > 5)
{
    for (int i = 5; i < tabs_list[0].Count; i++)
    {
        tabScrollbarSize = tabScrollbarSize + 80;    //sxedon
    }
}

IEnumerator GetFileLocation()
{
//creating form to for the upload.php (server-side)
WWWForm form = new WWWForm();

//form elements with data we need to send to php
form.AddField("action", "query");
form.AddField("request", "template_loc");
form.AddField("SearchValue", SettingsHandler.getGameTemplateID());
Debug.Log("Form Created");
DEBUG_LOG = "Form Created";
//applying the form to the php script
WWW w = new WWW("http://83.212.105.106:85/getData.php", form);
yield return w;

//checking for errors
if (w.error != null)
{
    Debug.Log(w.error);
    DEBUG_LOG = w.error;
}
else
{
    if (w.text == "-1")

```

```

        {
            Debug.Log("File not found");
            DEBUG_LOG = w.text;
        }
        else
        {
            Debug.Log("File Location aquired successful!");
            DEBUG_LOG = w.text;
            path = _FileLocation + w.text + ".xml";
            Debug.Log(path);
            StartCoroutine(loadXML());
        }
    }
}

public void checkToEnablePlay()
{
    if (playButtonEnabled)
    {
        GUI.enabled = true;
    }
    else
    {
        GUI.enabled = false;
    }
}

// Update is called once per frame
void Update()
{
}

private void OnGUI()
{
    rightArea = new Rect(rightGroupRect.x, rightGroupRect.y + rightGroupRect.height
+ 200, 470, 130); //rename inspectoRect to rightArea
    GUI.BeginGroup(rightArea);
        GUI.Box(new Rect(0, 0, rightArea.width / 2, rightArea.height),
"Artifacts' Details");
    scrollPosition = GUI.BeginScrollView(new Rect(0, 0, rightArea.width / 2,
rightArea.height * 9.9f / 10), scrollPosition, new Rect(0, 0, 0,
rightArea.height * 9.9f / 10 + (detailScrollbarSize)));
    if (InteractWithGo.clickedSprite != null)
    {
        detailScrollbarSize = 130;
        alignInspector = 0;
        xPos =
scene_objects.getSingleObject(InteractWithGo.clickedSprite).getPosition(
).x.ToString();
        yPos =
scene_objects.getSingleObject(InteractWithGo.clickedSprite).getPosition(
).y.ToString();
        xRot =
scene_objects.getSingleObject(InteractWithGo.clickedSprite).get_Rotation(
).x.ToString();
        yRot =
scene_objects.getSingleObject(InteractWithGo.clickedSprite).get_Rotation(
).y.ToString();
        GUI.Label(new Rect(0, 30, 100, 50), "-Sprite's Name: ");
        GUI.Label(new Rect(0, 70, 100, 50), "-Components: ");
        string tempComponent;
        GUI.TextField(new Rect(130, 30, 100, 20), InteractWithGo.clickedSprite);
    }
}

```

```

if (InteractWithGo.spriteComponents[InteractWithGo.clickedSprite].Count
!=0)
{
    for (int j = 0; j <
InteractWithGo.spriteComponents[InteractWithGo.clickedSprite].Count;
j++)
    {
        tempComponent = InteractWithGo.components[j];
        GUI.Label(new Rect(15, (90 + (j * 20)), 100, 20),
InteractWithGo.components[j]);
        if (GUI.Button(new Rect(115, (90 + (j * 20)), 20, 20), "x"))
        {
            scene_objects.getSingleObject(InteractWithGo.clickedSprite).
removeComponent(InteractWithGo.components[j]);
        }
        alignInspector = j * 20;
        detailScrollbarSize += 20;
    }
}

GUI.Label(new Rect(0, 110 + alignInspector, 120, 30), "-Position: ");
GUI.Label(new Rect(20, 130 + alignInspector, 50, 20), "x: ");
xPos = GUI.TextArea(new Rect(60, 130 + alignInspector, 50, 20), xPos);
GUI.Label(new Rect(100, 130 + alignInspector, 50, 20), "y: ");
yPos = GUI.TextField(new Rect(140, 130 + alignInspector, 50, 20), yPos);
GUI.Label(new Rect(0, 160 + alignInspector, 120, 30), "-Rotation: ");
GUI.Label(new Rect(20, 180 + alignInspector, 50, 20), "x: ");
xRot = GUI.TextField(new Rect(60, 180 + alignInspector, 50, 20), xRot);
GUI.Label(new Rect(100, 180 + alignInspector, 50, 30), "y: ");
yRot = GUI.TextField(new Rect(140, 180 + alignInspector, 50, 20), yRot);
if (GUI.Button(new Rect(100, 220 + alignInspector, 50, 30), "Apply"))
{
    Debug.Log("xpos" + xPos);
}
}

GUI.EndScrollView();
GUI.Box(new Rect(rightArea.width / 2 + 5, 0, rightArea.width / 2 - 5,
rightArea.height), "Animation window");
GUI.EndGroup();
if (showPopUp)
{
    GUI.Window(0, new Rect((Screen.width / 2) - 150, (Screen.height / 2) - 75
, 300, 190), showPopUpWindow, "Save Game");
}
//end- to pop up widow
int align = 12;
int x = 0;
if (showLabel)
{
    GUI.skin.label.alignment = TextAnchor.UpperCenter;
    for (int i = 0; i < artifactsNames.Count; i++)
    {
        if (i % 2 == 0)
        {
            x = 1;
            align += 120;
            GUI.Label(new Rect(100, align, 70, 100), artifactsNames[i]);
        }
        else
        {
            GUI.Label(new Rect(100 + (x * 75), align, 70,

```

```

        100),artifactsNames[i]);
        x++;
    }
}
}
Color col = GUI.backgroundColor;
GUI.backgroundColor = new Color(0, 0, 0, 0);
btnRects.Add(new Rect(10, 10, 160, 30));
for (int i = 1; i < 4; i++)
{
    btnRects.Add(new Rect(btnRects[i - 1].x + btnRects[i - 1].width + 10,
        btnRects[i - 1].y, btnRects[i - 1].width, btnRects[i - 1].height));
}

GUI.backgroundColor = col;
GUI.BeginGroup(rightGroupRect);

tabScrollPosition = GUI.BeginScrollView(new Rect(30, 30, 385, 50),
tabScrollPosition, new Rect(0, 0, tabScrollbarSize, 0));

if (loadedDat)
{
    for (int i = 0; i < tabs_list[0].Count; i++)
    {
        btnCreate(i, 0 + i * (388 / 5), 5, 388 / 5, 30, "Scene " + (i + 1),0);
    }
}
//-----
-----

GUI.enabled = true;

GUI.EndScrollView();
if (GUI.Button(new Rect(0, 35, 30, 45), "+"))
//otan path8ei to + vazei epipleon skhnes. Tou epitrepei mono oses exei pei o
expert(loadedData.getMaxScenes())
{
    if (sceneCounter < loadedData.getMaxScenes())
    {
        sceneGo = (GameObject)Instantiate(scene,
scene.transform.position,scene.transform.rotation);
        sceneCounter++;
        sceneGo.name = "Scene " + sceneCounter;
        sceneProp sceneprop1 = new
sceneProp(sceneGo.name,GetInstanceID().ToString(), sceneGo);
        allScenes.Add("Scene " + sceneCounter, sceneprop1);
        goPerScene.Add("Scene " + sceneCounter, new List<gameEditorObject>());
        foreach (KeyValuePair<string, sceneProp> pair in allScenes)
        {
            Debug.Log(pair.Key);
        }
        TabProperties button = new TabProperties(c, true, "Scene " + (c + 1));
        tabs_list[0].Add(button);

        if (sceneCounter > 5)
        {
            tabScrollbarSize = tabScrollbarSize + 80;
        }
    }
    else
        Debug.Log("More scenes are not allowed");
}

```

---

```

}

GUI.EndGroup();
GUI.BeginGroup(leftGroupRect);
z = 0;
left = 0;
top = 0;
if (objectCategories.Count % 3 > 0)
{
    alignSubTabs = objectCategories.Count / 3 * 30;
}
foreach (var pair in objectCategories)
{
    if (z % 3 == 0 && z > 0)
    {
        left = 0;
        top = 1;
        if (pair.Key.Length > 9)
        {
            btnCreate(z, left, top * 30, 90, 30, pair.Key, 1);
        }
        else
        {
            btnCreate(z, left, top * 30, 70, 30, pair.Key, 1);
        }
    }
    else
    {
        if (pair.Key.Length > 9)
        {
            btnCreate(z, left * (210 / 3), top * 30, 90, 30, pair.Key, 1);
        }
        else
        {
            btnCreate(z, left * (210 / 3), top * 30, 70, 30, pair.Key, 1);
        }
    }
    z++;
    left++;
}

```



---

```

if (showSubTabs)
{
    selectedSubTab = 0;
    for (int i = 0; i < subTabsList[selectedLibraryTabId].Count; i++)
    {
        if (subTabsList[selectedLibraryTabId][i].getName().Length > 9)
        {
            btnCreateSubTabs(i, 0, (40 + (i * 40)) + alignSubTabs, 85, 30,
                subTabsList[selectedLibraryTabId][i].getName(),
                selectedLibraryTabId);
        }
        else
        {
            btnCreateSubTabs(i, 0, (40 + (i * 40)) + alignSubTabs, 75, 30,
                subTabsList[selectedLibraryTabId][i].getName(),
                selectedLibraryTabId);
        }
    }
}
GUI.enabled = true;
GUI.EndGroup();

if (GUI.Button(new Rect(20, 500, 75, 30), "Upload"))
{
    Application.ExternalCall("uploadWindow", UserInformationHandler.getUserID());
}

if (GUI.Button(new Rect(Screen.width-110,Screen.height - 30, 100, 30),"Back"))
{
    switch (UserInformationHandler.getUserAccess())
    {
        case 0:
            Application.LoadLevel("admin_menu");
            break;
        case 1:
            Application.LoadLevel("expert_menu");
            break;
        case 2:
            Application.LoadLevel("educator_menu");
            break;
    }
}

if (GUI.Button(new Rect(Screen.width - 110, Screen.height - 60, 100, 30),
    "Kinect Viewer"))
{
    Application.LoadLevel("kinectUpperSkeleton");
}

if (GUI.Button(new Rect(Screen.width - 110, Screen.height - 90, 100, 30),
    "Save Game"))
{
    if (playButtonEnabled == false)
    {
        showPopUp = true;
    }
}
checkToEnablePlay();

```

```

if (GUI.Button(new Rect(Screen.width - 110, Screen.height - 120, 100, 30),
"Play Game"))
{
    if (playButtonEnabled)
    {
        saveData();
    }
    System.Threading.Thread.Sleep(2000);
    Application.LoadLevel("play_scene");
}

if (DEBUG_MODE)
{
    GUI.Label(new Rect(-20, 550, 300, 100), DEBUG_LOG);
}
}

//Creates the tabs(buttons)
public void btnCreate(int i, float left, float top, float width, float height,
string name, int tab_id)
{
    GUI.enabled = tabs_list[tab_id][i].getState();
    if (GUI.Button(new Rect(left, top, width, height), name))
    {
        if (allScenes.ContainsKey(name))
        //an to button pou patietai exei name pou iparxei mesa sto allScenes(dld an path8ei
scene tab) tote kalei autis tis me8odous
        {
            saveScene();
            placeScene(name);
        }
        Else
        //alliws an path8ei ena apo ta tabs aristera(sprites,scripts klp) kalei auth th me8odo
        {
            showLabel = false;
            for (int x = 0; x < subTabsList.Count; x++)
            {
                for (int y = 0; y < subTabsList[x].Count; y++)
                {
                    subTabsList[x][y].setState(true);
                }
            }
            displaceAll();
            showSubTabs = true;
            selectedLibraryTab = name;
            selectedLibraryTabid = i;
        }
        selectedTabName = name;
        selectedTab = i;
        try
        {
            tabs_list[tab_id][disabled_buttons[tab_id]].setState(true);
        }
        catch { }
        tabs_list[tab_id][selectedTab].setState(false);
        disabled_buttons[tab_id] = selectedTab;
        GUI.enabled = true;
    }
}
}

```

---

```

public void btnCreateSubTabs(int i, float left, float top, float width, float height,
string name, int tab_id)
{
    selectedSubTab = 0;
    GUI.enabled = subTabsList[tab_id][i].getState();

    if (GUI.Button(new Rect(left, top, width, height), name))
    {
        showArtifacts(name);
        showLabels(selectedLibraryTab + "/" + name);
        selectedSubTab = i;
        try
        {
            subTabsList[tab_id][subDisabled_buttons[tab_id]].setState(true);
        }
        catch { }
        subTabsList[tab_id][selectedSubTab].setState(false);
        subDisabled_buttons[tab_id] = selectedSubTab;
        GUI.enabled = true;
    }
}

public void showLabels(string path)
{
    List<string> names = new List<string>();
    string[] filename = path.Split('/');
    for (int i = 0; i < libraryObjects.Count; i++)
    {
        if (libraryObjects[i].getCategory() == filename[1])
        {
            names.Add(libraryObjects[i].getName());
        }
    }

    if (filename[1] == "All")
    {
        for (int i = 0; i < libraryObjects.Count; i++)
        {
            if (libraryObjects[i].getKind() == filename[0])
            {
                names.Add(libraryObjects[i].getName());
            }
        }
    }
    artifactsNames = names;
    showLabel = true;
}

```

//me8odos h opoia krataei mprosta th sknh pou patietai kai diwxnei(pisw apo thn camera) oles tis ipoloipes mazi me ta artifacts pou tous exoume valei panw

```

public void placeScene(string sceneName)
{
    //removes scenes
    foreach (KeyValuePair<string, sceneProp> pair in allScenes)
    {
        sceneProp sceneObj = allScenes[pair.Key];
        sceneObj.getScene().transform.position = new Vector3(0, 0, -15);
        Debug.Log("Remove scenes");
    }
}

```

---

```

//removes artifacts
List<gameEditorObject> objectPerScene = new List<gameEditorObject>();
foreach (KeyValuePair<string, List<gameEditorObject>> pair in goPerScene)
{
    objectPerScene = goPerScene[pair.Key];

    for (int i = 0; i < objectPerScene.Count; i++)
    {
        objectPerScene[i].getGo().transform.position = new Vector3(0, 0, -15);
    }
}

//Places the right scene in front
sceneProp sceneObj1 = allScenes[sceneName];
sceneObj1.getScene().transform.position = new Vector3(1.208702f, 2.067295f,
4.782515f);//test Y = 0.6580517 Y
List<gameEditorObject> objectPerScene1 = new List<gameEditorObject>();
for (int i = 0; i < goPerScene[sceneName].Count; i++)
{
    objectPerScene1.Add(goPerScene[sceneName][i]);
}
for (int i = 0; i < objectPerScene1.Count; i++)
{
    objectPerScene1[i].getGo().transform.position =
    objectPerScene1[i].getPosition();
}
activeSceneHandler.setActiveScene(sceneName);
activeScene = sceneName;
Debug.Log("active scene " + sceneName);
}

// dokimastiki sinartisi gia tin swsti apotiposi ton instatiated object
private Vector3 ToPixel(Vector3 cord)
{
    cord = Camera.main.ScreenToWorldPoint(cord);
    return (cord);
}

//analoga me to poio apo ta library tabs patietai emfanizontai ta katalhla artifacts
public void showArtifacts(string name)
{
    List<string> scripts = new List<string>();
//object instatiation here
if (selectedLibraryTab == "Artifacts")
{
    switch (name)
    {
        case "All":
        case "People":
        case "Objects":
        case "Numbers":
        case "Shapes":
            scripts.Clear();
            scripts.Add("ClickOnEditorObjects");
            show(artifactInfo, scripts, "Artifacts/" + name, name);
            break;
    }
}
}

```

---

```

if (selectedLibraryTab == "Scripts")
{
    switch (name)
    {
        case "All":
        case "Controllers":
        case "Motion":
        case "Sounds":
            scripts.Clear();
            scripts.Add("ScriptAddCollision");
            scripts.Add("ScriptNumbersOnObjects");
            scripts.Add("RandomNumber");
            scripts.Add("ScriptScore");
            scripts.Add("ScriptAddBasket");
            scripts.Add("ScriptTimer");
            scripts.Add("KinectScript");
            scripts.Add("ScriptCharMotor");
            scripts.Add("ScriptMoveBall");
            scripts.Add("ScriptPickFlower");
            scripts.Add("ScriptRotate");
            scripts.Add("ScriptSound");
            show(scriptInfo, scripts, "Scripts/" + name, name);
            break;
    }
}
if (selectedLibraryTab == "Backgrounds")
{
    switch (name)
    {
        case "All":
        case "Landscapes":
        case "Plain":
            scripts.Clear();
            scripts.Add("ScriptBackground");
            show(backgroundInfo, scripts, "Backgrounds/" + name, name);
            break;
    }
}
if (selectedLibraryTab == "Custom")
{
    switch (name)
    {
        case "Artifacts":
            scripts.Clear();
            scripts.Add("ClickOnEditorObjects");
            ShowCustomArtifacts(scripts);
            break;
        case "Background":
            scripts.Clear();
            scripts.Add("ScriptBackground");
            ShowCustomBackgrounds(scripts);
            break;
    }
}
}

private void ShowCustomBackgrounds(List<string> scripts)
{
    throw new NotImplementedException();
}

```

---

```

private void ShowCustomArtifacts(List<string> scripts)
{
    bool loaded = false;
    bool categoryLoaded = false;
    if (libraryObjects.Count == 0)
    {
        StartCoroutine(GetCustomArtifacts(scripts));
    }
    else
    {
        for (int i = 0; i < libraryObjects.Count; i++)
        {
            if (libraryObjects[i].getKind() == selectedLibraryTab)
            {
                placeAll(selectedLibraryTab);
                categoryLoaded = true;
            }
        }
        if (!categoryLoaded)
        {
            StartCoroutine(GetCustomArtifacts(scripts));
            categoryLoaded = false;
        }
    }
}

private void instantiateCustomArtifacts(List<string> scripts)
{
    mother_cube = GameObject.Find("mother_cube");

    for (int i = 0; i < CustomTextures.Count; i++)
    {
        GameObject go = (GameObject)Instantiate(mother_cube, new Vector3(0, 0, -
            10f), mother_cube.transform.rotation);
        go.AddComponent(scripts[0]);
        if (CustomTextures[i] != null)
        {
            go.renderer.material.mainTexture = CustomTextures[i];
        }
        else
        {
            Debug.Log("Texture empty");
        }
        libraryObjects.Add(new LibraryObject("Custom",
            CustomArtifacts.getImage(i).getName(), "Artifacts",
            go));placeArtifacts("Artifacts");
    }
}

public void show(Dictionary<string, string> artifactsInfo, List<string> scripts,
string folderName, string name)
{
    bool loaded = false;
    bool categoryLoaded = false;

    if (libraryObjects.Count == 0)
    {
        instantiateArtifacts(artifactsInfo, scripts, name);
    }
    else

```

```

{
    if (name == "All")
    {
        for (int i = 0; i < libraryObjects.Count; i++)
        {
            if (libraryObjects[i].getKind() == selectedLibraryTab)
            {
                placeAll(selectedLibraryTab);
                categoryLoaded = true;
            }
        }
        if (!categoryLoaded)
        {
            instantiateArtifacts(artifactsInfo, scripts, name);
            categoryLoaded = false;
        }
    }
    else
    {
        for (int i = 0; i < libraryObjects.Count; i++)
        {
            if (libraryObjects[i].getCategory() == name)
            {
                placeArtifacts(name);
                loaded = true;
            }
        }
        if (!loaded)
        {
            instantiateArtifacts(artifactsInfo, scripts, name);
            loaded = false;
        }
    }
}
}
}

```

//Method that places right artifacts in front and displaces the others when changing library tab

```

public void placeArtifacts(string name)
{
    float x = 0; //left
    float y = 6.3f; //height
    int c = 0;
    for (int i = 0; i < libraryObjects.Count; i++)
    {
        if (libraryObjects[i].getCategory().Equals(name))
        {
            if (c % 2 == 0)
            {
                y -= 2.0f;
                x = -5.7f;
                libraryObjects[i].getLibraryGo().transform.position = new
                Vector3(x, y, 0);
                c++;
            }
            else
            {
                x = (x + 1.2f);
                libraryObjects[i].getLibraryGo().transform.position = new
                Vector3(x, y, 0);
                c++;
            }
        }
    }
}

```

---

```

        }
        else
        {
            libraryObjects[i].getLibraryGo().transform.position = new
            Vector3(0,0,-10);
        }
    }
}

//Method that displaces all artifacts behind the camera at scene transition
public void displaceAll()
{
    for (int i = 0; i < libraryObjects.Count; i++)
    {
        libraryObjects[i].getLibraryGo().transform.position = new Vector3(0, 0, -
        10);
    }
}

//Method that places right artifacts in front at scene transition
public void placeAll(string name)
{
    float y = 6.3f;
    float x = 0;
    int c = 0;
    bool loaded = false;

    for (int i = 0; i < libraryObjects.Count; i++)
    {
        if (libraryObjects[i].getKind().Equals(name))
        {
            if (c % 2 == 0)
            {
                y -= 2.0f;
                x = -5.7f;
                libraryObjects[i].getLibraryGo().transform.position = new
                Vector3(x, y, 0);
                c++;
            }
            else
            {
                x = (x + 1.2f);
                libraryObjects[i].getLibraryGo().transform.position = new
                Vector3(x, y, 0);
                c++;
            }
        }
        else
        {
            libraryObjects[i].getLibraryGo().transform.position = new Vector3(0, 0,
            -10);
        }
    }
    if (!loaded)
    {
    }
}
}

```



```

public void instantiateArtifacts(Dictionary<string, string> artifactsInfo,
List<string> scripts, string name)
{
    k = 0;
    c = 5;
    string[] filenames;
    List<GameObject> gameObjectList = new List<GameObject>();
    mother_cube = GameObject.Find("mother_cube");
    foreach (KeyValuePair<string, string> pair in artifactsInfo)
    {
        string foldername = pair.Value;
        filenames = foldername.Split('/');
        GameObject go = (GameObject)Instantiate(mother_cube, new Vector3(0, 0, -
10f),mother_cube.transform.rotation);
        temptexture = (Texture2D)Resources.Load(pair.Value + "/" + pair.Key);
        if (scripts.Count > 1)
        {
            go.AddComponent(scripts[k]);
            k++;
        }
        else
        {
            go.AddComponent(scripts[0]);
        }
        if (temptexture != null)
        {
            go.renderer.material.mainTexture = temptexture;
        }
        else
        {
            Debug.Log("Texture empty");
        }
        libraryObjects.Add(new LibraryObject(filenames[0], pair.Key,
filenames[1], go));
        if (name == "All")
        {
            placeAll(selectedLibraryTab);
        }
        else
        {
            placeArtifacts(name);
        }
    }
}

//components pou exei mesa to pop up window
public void showPopUpWindow(int windowID)
{
    GUI.Label(new Rect(65, 40, 200, 30), "Game Name");
    if (firstTime)
    {
        if (!String.IsNullOrEmpty(SettingsHandler.getGameName()))
        {
            gameName2 = SettingsHandler.getGameName();
            firstTime = false;
        }
    }
    gameName2 = GUI.TextField(new Rect(65, 60, 170, 30), gameName2, 90);
    if (GUI.Button(new Rect(65, 100, 75, 30), "OK"))
    {
        SettingsHandler.setGameName(gameName2);
        if (playButtonEnabled == false)

```

---

```

        {
            saveData();
            playButtonEnabled = true;
        }
        showPopUp = false;
    }
    if (GUI.Button(new Rect(65 + 75 + 20, 100, 75, 30), "Cancel"))
    {
        showPopUp = false;
    }
}

private void saveData()
{
    StartCoroutine("UploadFile");
}
IEnumerator loadXML()
{
    WWW w = new WWW(path);
    while (!w.isDone)
    {
        yield return new WaitForSeconds(0.5f);
    }
    yield return w.text;
    Debug.Log(w.text);
    XmlSerializer xmlSerializer;
    Debug.Log("Serializer rdy");
    byte[] byteArray = Encoding.UTF8.GetBytes(w.text);
    Debug.Log("Bytearray rdy");
    MemoryStream stream = new MemoryStream(byteArray);
    Debug.Log("Memory stream ready");
    try
    {
        Debug.Log("Trying...");
        xmlSerializer = new XmlSerializer(typeof(ExpertXML));
        loadedData = (ExpertXML)xmlSerializer.Deserialize(stream);
    }
    catch { }
    Debug.Log("loading...");
    temptexture = null;
    loadData();
}

private void loadData()
{
    bool firstTabCreated = false;
    for (sceneCounter = 0; sceneCounter < loadedData.getMinScenes();
        sceneCounter++)
    {
        //vazei toses skhnes oses epitrepei o expert(plhroforia apo xml)
        sceneGo = (GameObject)Instantiate(scene, new Vector3(2.2f, 0.35f, 4.782515f),
            scene.transform.rotation);
        sceneGo.name = "Scene " + (sceneCounter + 1);
        sceneprop = new sceneProp(scene.name, sceneGo.GetInstanceID().ToString(),
            sceneGo);
        allScenes.Add("Scene " + (sceneCounter + 1), sceneprop);
        goPerScene.Add("Scene " + (sceneCounter + 1), new List<gameEditorObject>());
        Debug.Log("Scene counter " + sceneCounter);
    }
}

```

---

```

        foreach (KeyValuePair<string, sceneProp> pair in allScenes)
// vazei tosa tabs osa einai kai ta scenes pou dhmiourgh8hkan pio panw
    {
        if (!firstTabCreated)
        {
            TabProperties button1 = new TabProperties(c, false, pair.Key);
            tabs_list[0].Add(button1);
            firstTabCreated = true;
            placeScene(pair.Key);
        }
        else
        {
            TabProperties button = new TabProperties(c, true, pair.Key);
            tabs_list[0].Add(button);
        }
    }
    loadedDat = true;
    Debug.Log("Loading data completed");
}

public void saveScene()
{
    for (int i = 0; i < goObject.Count; i++)
    {
        goPerScene[activeScene].Add(goObject[i]);
    }
    goObject.Clear();
}

//upload script
IEnumerator UploadFile()
{
    string userID;
    if (DEBUG_MODE)
    {
        UserInformationHandler.setUserID(1);
        userID = "1";
    }
    else
    {
        userID = UserInformationHandler.getUserID().ToString();
    }
    XmlSerializer serial = new XmlSerializer(typeof(ObjectsContainer));
    MemoryStream memoryStream = new MemoryStream();
    XmlTextWriter xmlTextWriter = new XmlTextWriter(memoryStream, Encoding.UTF8);
    serial.Serialize(xmlTextWriter, scene_objects);
    memoryStream = (MemoryStream)xmlTextWriter.BaseStream;
    //converting the xml to bytes to be ready for upload
    byte[] Data = memoryStream.ToArray();
    //setting the unique time so the game won't create multiple inserts into
    database

    string fileName;
    fileName = "GameInfo.xml"; //file name where the xml will be saved

    memoryStream.Close();

    //creating form to for the upload.php (server-side)
    WWWForm form = new WWWForm();

    //form elements with data we need to send to php

```

---

```

        form.AddField("action", "upload");
        form.AddField("file", "file");
        form.AddField("user_id", userID);
        form.AddField("type", "game");
        form.AddField("timestamp", SettingsHandler.getUniqueTime());
        form.AddField("Gamename", SettingsHandler.getGameName());

        if (String.IsNullOrEmpty(SettingsHandler.getGameDescription()))
//--just for quick run se periptwsi pou den exei description
        {
            SettingsHandler.setGameDescription("blah blah description");
        }
        form.AddField("GameDesc", SettingsHandler.getGameDescription());

        if (SettingsHandler.getGameVisibility().Equals(null))
//--just for quick run -
        {
            SettingsHandler.setGameVisiblity(0);
        }
        form.AddField("GameVis", SettingsHandler.getGameVisibility());
        form.AddBinaryData("file", Data, fileName, "text/xml");
//in this line we add the binary code
        Debug.Log("binary data added ");
        DEBUG_LOG = "Binary data added ";
//applying the form to the php script
        WWW w = new WWW("http://83.212.105.106:85/upload.php", form);
        yield return w;
//checking for errors
        if (w.error != null)
        {
            Debug.Log(w.error);
            DEBUG_LOG = w.error;
        }
        else
        {
            Debug.Log("Files sent successfully!");
            DEBUG_LOG = "Files sent successfully!";
        }
    }

private Dictionary<string, string> loadTextures(List<string> paths)
{
    Dictionary<string, string> loadedTextures = new Dictionary<string, string>();
    for (int i = 0; i < paths.Count; i++)
    {
        UnityEngine.Object[] textures = Resources.LoadAll(paths[i]);

        for (int j = 0; j < textures.Length; j++)
        {
            loadedTextures.Add(textures[j].name, paths[i]);
            string s = paths[i];
            string[] words = s.Split('/');
            //Debug.Log(words[1]);
            //Debug.Log(textures[j]);
        }
    }

    return loadedTextures;
}
#region Upload Image methods
public void setBaseString(string base64)

```

---

```

{
    s1 = base64.Split(', ')[1];
    imageName = base64.Split(', ')[2];
    imageVisibility = base64.Split(', ')[3];
    basearray = System.Convert.FromBase64String(s1);
    tex = new Texture2D(128, 128);
    tex.LoadImage(basearray);
    StartCoroutine(UploadImage(basearray));
}
IEnumerator UploadImage(byte[] data)
{
    string userID;

    //userID = "1"; //DELETE THIS AND UNCOMMENT THE FOLLOWING LINE BEFORE RELEASE
    userID = UserInformationHandler.getUserID().ToString();

    string fileName;
    fileName = ".png"; //file name where the xml will be saved

    //creating form to for the upload.php (server-side)
    WWWForm form = new WWWForm();

    //form elements with data we need to send to php
    form.AddField("action", "upload");
    form.AddField("file", "file");
    form.AddField("user_id", userID);
    form.AddField("type", "Image");
    form.AddField("ImageName", imageName);
    form.AddField("ImageVisibility", imageVisibility);
    form.AddField("timestamp", SettingsHandler.getUniqueTime());
    form.AddBinaryData("file", data, fileName, "image/png");
//in this line we add the binary code
    Debug.Log("binary data added ");

    //applying the form to the php script
    WWW w = new WWW("http://83.212.105.106:85/upload.php", form);
    yield return w;

    //checking for errors
    if (w.error != null)
    {
        Debug.Log(w.error);
    }
    else
    {
        Debug.Log("Files sended successful!");
    }
}
IEnumerator GetCustomArtifacts(List<string> scripts)
{
    string userID;
    //userID = "1"; //DELETE THIS AND UNCOMMENT THE FOLLOWING LINE BEFORE RELEASE
    userID = UserInformationHandler.getUserID().ToString();
    //creating form to for the upload.php (server-side)
    WWWForm form = new WWWForm();

```

---

```

//form elements with data we need to send to php
form.AddField("action", "images");
form.AddField("userid", userID);
Debug.Log("Form Created");
DEBUG_LOG = "Form Created";
//applying the form to the php script
WWW reply = new WWW("http://83.212.105.106:85/getData.php", form);
yield return reply;
//checking for errors
if (reply.error != null)
{
    Debug.Log(reply.error);
    DEBUG_LOG = reply.error;
}
else
{
    if (reply.text == "-1")
    {
        Debug.Log("Data not found");
        DEBUG_LOG = reply.text;
    }
    else
    {
        Debug.Log(reply.text);
        XmlSerializer xmlSerializer;
        byte[] byteArray = Encoding.UTF8.GetBytes(reply.text);
        MemoryStream stream = new MemoryStream(byteArray);
        try
        {
            xmlSerializer = new XmlSerializer(typeof(CustomImageList));
            CustomArtifacts(CustomImageList)xmlSerializer.Deserialize(stream);
        }
        catch (Exception e)
        {
            Debug.Log(e.Message.ToString());
        }
        StartCoroutine(GetTextures(scripts));
    }
}
}
IEnumerator GetTextures(List<string> scripts)
{
    for (int i = 0; i < CustomArtifacts.getImageList().Count; i++)
    {
        string url = "http://83.212.105.106:85/users/" +
            CustomArtifacts.getImage(i).getOwnerID() + "/images/" +
            CustomArtifacts.getImage(i).getID() + ".png";
        WWW www = new WWW(url);
        yield return www;
        Texture2D ctexture = www.texture;
        ctexture.name = "Custom/" + CustomArtifacts.getImage(i).getOwnerID() +
            "/" + CustomArtifacts.getImage(i).getID();
        CustomTextures.Add(ctexture);
    }
    instantiateCustomArtifacts(scripts);
}
#endregion

```

---

```
public GameObject getObjectByName(string name)
{
    for (int i = 0; i < goPerScene[activeSceneHandler.getActiveScene()].Count;
        i++)
    {
        if (goPerScene[activeSceneHandler.getActiveScene()][i].getGo().name == name)
        {
            goByname = goPerScene[activeSceneHandler.getActiveScene()][i].getGo();
        }
    }
    return goByname;
}

public void removeObjectByName(string name)
{
    for (int i = 0; i < goPerScene[activeSceneHandler.getActiveScene()].Count; i++)
    {
        if (goPerScene[activeSceneHandler.getActiveScene()][i].getGo().name == name)
        {
            Destroy(goByname =
                goPerScene[activeSceneHandler.getActiveScene()][i].getGo());
            goPerScene[activeSceneHandler.getActiveScene()].RemoveAt(i);
        }
    }
}
}
```

---

## References

---

- <sup>1</sup> Λόγου σπουδή έκθεση - έκφραση Γ' λυκείου
- <sup>2</sup> <http://www.methodiko-frontistirio.gr/ep/section01/page02.htm>)
- <sup>3</sup> *Εισαγωγή στις εκπαιδευτικές εφαρμογές των ΤΠΕ(Τεχνολογία πληροφοριών και επικοινωνίας)*
- <sup>4</sup> Ian Sommerville, Βασικές αρχές τεχνολογίας λογισμικού
- <sup>5</sup> Lifelong Kindergarten Group ,MIT media Lab <https://scratch.mit.edu/>
- <sup>6</sup> Carnegie Mellon University School of Computer Science <http://www.alice.org/index.php>
- <sup>7</sup> University of Kent, Oracle <http://www.greenfoot.org/door>
- <sup>8</sup> [https://en.wikipedia.org/wiki/Serious\\_game](https://en.wikipedia.org/wiki/Serious_game)
- <sup>9</sup> Katja Hertzschuch, “Serious games”, Hauptseminar“E-Learning –Sommersemester2008
- <sup>10</sup> Damien Djaouti, Julian Alvarez, Jean-Pierre Jessel, and Olivier Rampnoux “Origins of Serious Games”, chapter 3
- <sup>11</sup> Tarja Susi, Mikael Johannesson, Per Backlund “Serious Games – An Overview”, Technical Report HS- IKI - TR-07-001 School of Humanities and Informatics University of Skövde, Sweden, <http://www.diva-portal.org/smash/get/diva2:2416/FULLTEXT01.pdf>
- <sup>12</sup> Ευρωπαϊκός φορέας για την ανάπτυξη της ειδικής αγωγής “Συμμετοχική εκπαίδευση και πρακτικές στην τάξη στη δευτεροβάθμια εκπαίδευση” 2005
- <sup>13</sup> <http://inclusiveeducation.ca/about/what-is-ie/>
- <sup>14</sup> PBS Parents - <http://www.pbs.org/parents/education/learning-disabilities/inclusive-education/>
- <sup>15</sup> <http://www.inclusionbc.org/our-priority-areas/inclusive-education/what-inclusive-education>
- <sup>16</sup> UNESCO Education Inclusion Policy Guidelines - <http://www.scribd.com/doc/37626440/UNESCO-Education-Inclusion-Policy-Guidelines>
- <sup>17</sup> Assistive technology industry association - <http://www.atia.org/i4a/pages/index.cfm?pageid=3859>
- <sup>18</sup> Matthew T. Marino and Constance C. Beecher Washington State University, Pullman, Washington “Assistive Technology Policy: Promoting Inclusive Education for Students with Reading Disabilities” - [http://public.wsu.edu/~matthewmarino/documents/NWPassage\\_Marino\\_Beecher.pdf](http://public.wsu.edu/~matthewmarino/documents/NWPassage_Marino_Beecher.pdf)
- <sup>19</sup> Fouzia Khursheed Ahmad “Use of Assistive Technology in Inclusive Education: Making Room for Diverse Learning Needs” - [http://www2.hu-berlin.de/transcience/Vol6\\_No2\\_62\\_77.pdf](http://www2.hu-berlin.de/transcience/Vol6_No2_62_77.pdf)
- <sup>20</sup> Ακουμιανάκης Δημοσθένης “Διεπαφή χρήστη υπολογιστή” 2008
- <sup>21</sup> ACM SIGCHI Curricula for Human-Computer Interaction “CHAPTER 2 : Human computer Interaction” - [http://old.sigchi.org/cdg/cdg2.html#2\\_1](http://old.sigchi.org/cdg/cdg2.html#2_1)
- <sup>22</sup> Brad A. Myers “Overview of HCI Design and Implementation” Human Computer Interaction Institute School of Computer Science Carnegie Mellon University, October 28, 1999
- <sup>23</sup> <http://whatis.techtarget.com/definition/natural-user-interface-NUI>



---

<sup>24</sup> Georgios Maninis “ Natural user interfaces (NUIs)” 10 June 2013 -  
<http://www.webcredible.com/blog/natural-user-interfaces-nuis/>

<sup>25</sup> Η πληροφορική στην εκπαίδευση. Τεχνικές εφαρμογές, Κατάρτιση εκπαιδευτικών

<sup>26</sup> <https://unity3d.com/unity/system-requirements>