



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ
ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΜΕ
ΣΤΟΧΟ ΤΗΝ ΒΕΛΤΙΩΣΗ ΤΩΝ ΜΑΘΗΤΩΝ ΣΤ'
ΔΗΜΟΤΙΚΟΥ ΣΤΟ ΜΑΘΗΜΑ ΤΩΝ
ΜΑΘΗΜΑΤΙΚΩΝ.



Του φοιτητή
Τραχανάς Γεώργιος
Αρ. Μητρώου: 154549

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Αναπληρωτής Καθηγητής

Θεσσαλονίκη 2022

Τίτλος Π.Ε. Ανάπτυξη εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας με στόχο την βελτίωση των μαθητών στ' δημοτικού στο μάθημα των μαθηματικών

Κωδικός Π.Ε. 21331

Όνοματεπώνυμο φοιτητή Γεώργιος Τραχανάς

Όνοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Π.Ε. 12-10-2021

Ημερομηνία περάτωσης Π.Ε. 02-02-2022

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γεώργιου Τραχανά που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Αποτελεί γεγονός πως με την εξέλιξη της τεχνολογίας αυξάνεται και η ευκολία χρήσης της, με αποτέλεσμα να ανοίγονται νέοι ορίζοντες στον τρόπο που μπορεί αυτή να χρησιμοποιηθεί. Μπορεί να παρατηρηθεί πως με την πάροδο του χρόνου βλέπουμε όλο και περισσότερο την αφομοίωση της τεχνολογίας σε τομείς όπως η εκπαίδευση. Η εξέλιξη αυτή μας δίνει δημιουργικούς τρόπους να ανακαλύψουμε νέες μεθόδους εκμάθησης καθώς και να βελτιώσουμε τους ήδη υπάρχοντες. Μέσω της παρούσας Π.Ε. δίνεται ακόμη ένας συνδυασμός της τεχνολογίας, της εκπαίδευσης και της δημιουργικότητας, με κύριο στόχο την καλύτερη κατανόηση του μαθήματος των μαθηματικών καθώς και την εξοικείωση με τεχνολογίες επαυξημένης πραγματικότητας.

Περίληψη

Στόχος της πτυχιακής εργασίας είναι η βελτίωση της εκμάθησης του μαθήματος των μαθηματικών σε μαθητές Στ' δημοτικού, με την χρήση τεχνολογιών εκτεταμένης πραγματικότητας. Το παιχνίδι ονομάζεται «Μαγειρεύοντας Μαθηματικά», είναι διαθέσιμο για tablets καθώς και κινητά και περιέχει εννιά συνταγές οι οποίες καλύπτουν όλα τα κεφάλαια του βιβλίου των μαθηματικών. Το μόνο που χρειάζονται οι μαθητές για να παίξουν το παιχνίδι είναι να σκανάρουν με την κάμερα της συσκευής τους ένα QR code, έπειτα είναι ελεύθεροι να παίξουν και να ολοκληρώσουν το παιχνίδι. Η δημιουργία του παιχνιδιού έγινε μέσω των Unity Game Engine και για το μέρος της επαυξημένης πραγματικότητας χρησιμοποιήθηκε το πακέτο Vuforia Engine. Επίσης χρησιμοποιήθηκε το πρόγραμμα δημιουργίας τρισδιάστατων αντικειμένων Blender, για την δημιουργία όλων των μοντέλων του παιχνιδιού.

Development of an augmented reality educational game that aims at improving 6th grade students in the subject of Mathematics

George Trachanas

Abstract

The aim of this thesis is to enhance the learning experience of mathematics of 6th grade students in a pleasant and creative way. The game makes use of augmented reality, combining math problems with scenarios of cooking and purchasing of materials to better teach and explain the basics of 6th grade mathematics. In order to play the game the students will have to scan a QR code using the camera of their device. The development of the game was done through the Unity Game Engine using the Vuforia Engine package. On a further note every three dimensional model was created using the 3D computer graphics software Blender.

Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	v
Abstract	vii
Κατάλογος Σχημάτων	xii
Κεφάλαιο 1ο: Εισαγωγή.....	1
Κεφάλαιο 2ο: Εκτεταμένη πραγματικότητα και χρήσεις	3
2.1 Εισαγωγή.....	3
2.2 Εικονική Πραγματικότητα	4
2.3 Επαυξημένη Πραγματικότητα.....	4
2.4 Μικτή Πραγματικότητα.....	5
2.5 Χρήσεις της Εκτεταμένης Πραγματικότητας.....	7
2.5.1 Ιατρική.....	7
2.5.2 Μάρκετινγκ	8
2.5.3 Ρομποτική.....	8
2.5.4 Εκπαίδευση.....	9
2.6 Επίλογος.....	10
Κεφάλαιο 3ο: Προγράμματα και πακέτα.....	11
3.1 Εισαγωγή.....	11
3.2 Blender	11
3.3 Unity.....	11
3.3.1 Εγκατάσταση Λογισμικού Unity.....	11
3.3.2 Δημιουργία Προγράμματος και Εγκατάσταση Πακέτου Device Simulator.....	13
3.4 Vuforia	14
3.4.1 Εγκατάσταση και Ρύθμιση Πακέτου Vuforia.....	14
3.4.2 Image Target.....	15
3.4.3 Vuforia AR Camera & Image Target	17
3.4.4 Επιπλέον Ρυθμίσεις.....	20
3.5 Επίλογος.....	21
Κεφάλαιο 4ο: Αρχική Οθόνη και Μενού Εφαρμογής.....	23
4.1 Εισαγωγή.....	23
4.2 Χρήση Σκηνών	23

4.3	Αρχική Οθόνη	23
4.3.1	Ιεραρχία & Δομή Σκηνής	24
4.3.2	Εικόνες Γραφικά και Γραμματοσειρές.....	24
4.3.3	Λογική και Λειτουργία.....	24
4.4	Μενού.....	27
4.4.1	Ιεραρχία & Δομή Σκηνής	27
4.4.2	Εικόνες Γραφικά και Γραμματοσειρές.....	28
4.4.3	Λογική και Λειτουργία.....	28
4.5	Εμφάνιση Διεπαφής	30
4.5.1	Ορισμός εικόνας σε πεδίο ιδιότητας τύπου Image	30
4.5.2	Χειροκίνητη πρόσθεση image αντικειμένου	31
4.6	Επίλογος	32
Κεφάλαιο 5ο: Βασικά Χαρακτηριστικά και Λειτουργίες Σκηνών		33
5.1	Εισαγωγή.....	33
5.2	Vuforia: Target Images	33
5.3	Τρισδιάστατα Αντικείμενα.....	34
5.3.1	Δημιουργία Αντικειμένων και Μετατροπή σε FBX.....	34
5.3.2	Materials Textures και Prefabs.....	35
5.4	Scripts.....	38
5.4.1	DragAndDrop.....	38
5.4.2	CraftingRecipe.....	41
5.4.3	MainUiHandler.....	44
5.4.4	BowlHandler	46
5.4.5	Numpad & NumpadHandler.....	47
5.4.6	Questionnaire & QuestionnaireHandler	50
5.4.7	ChangeValues.....	55
5.5	Επίλογος	61
Κεφάλαιο 6ο: Ανάλυση Παιχνιδιού		63
6.1	Εισαγωγή.....	63
6.2	Ακέραιοι Αριθμοί.....	63
6.2.1	CookingBowl & BowlAnimation.....	65
6.3	Κλασματικοί Αριθμοί.....	66
6.4	Συμμιγείς Αριθμοί & Μετρήσεις Χρόνου	67
6.4.1	TimeUiHandler.....	68
6.4.2	Επεξεργασία τιμών άσκησης τύπου ερώτησης	69

6.5	Αναγωγή Στη Μονάδα	71
6.5.1	TeaCupAnimator	72
6.5.2	Επεξεργασία τιμών άσκησης τύπου συνταγής	75
6.6	Δεκαδικοί Αριθμοί	76
6.6.1	ShopPanel.....	77
6.7	Εξισώσεις	81
6.8	Αναλογίες και Ποσοστά	82
6.9	Συλλογή και επεξεργασία δεδομένων - Μετρήσεις.....	85
6.10	Γεωμετρία.....	88
6.10.1	ChocolateCakeHandler.....	91
6.11	Επίλογος.....	91
Κεφάλαιο 7ο:	Τελικά συμπεράσματα και βελτιώσεις της Π.Ε.....	93
BIBΛΙΟΓΡΑΦΙΑ.....		94

Κατάλογος Σχημάτων

Σχήμα 2.1: Εκτεταμένη Πραγματικότητα και υποκατηγορίες	3
Σχήμα 2.2: Χρήση VR εξοπλισμού	4
Σχήμα 2.3: Χρήση AR εφαρμογής μέσω Smartphone	5
Σχήμα 2.4: Ολογραφικά γυαλιά HoloLens.....	6
Σχήμα 2.5: Ολογραφική συσκευή Holus.....	6
Σχήμα 2.6: Θεραπεία VRET σε ασθενή που πάσχει απο υψοφοβία	7
Σχήμα 2.7: Χρήση AR εφαρμογής IKEA Place.....	8
Σχήμα 2.8: Αλληλεπίδραση MR και μηχανής ρομπότ	9
Σχήμα 2.8: Χρήση AR εφαρμογής σε μάθημα Χημείας	9
Σχήμα 2.9: Χρήση VR εξοπλισμού για την εκπαίδευση πυροσβέστη	10
Σχήμα 3.1: Δημιουργία τρισδιάστατου μολ με αυγά μέσω του λογισμικού Blender.....	11
Σχήμα 3.2: Εγκατάσταση Unity μέσω της πλατφόρμας Unity Hub.....	12
Σχήμα 3.3: Εγκατάσταση πακέτου συμβατότητας Android μέσω Unity Hub	12
Σχήμα 3.4: Δημιουργία 3D Προγράμματος.....	13
Σχήμα 3.5: Εγκατάσταση Device Simulator μέσω του παραθύρου Package Manager.....	14
Σχήμα 3.6: Έκδοση κλειδιού άδειας χρήσης του Vuforia Engine.....	15
Σχήμα 3.7: Πληροφορίες κλειδιού άδειας χρήσης Vuforia Engine.....	15
Σχήμα 3.8: Δημιουργία βάσης δεδομένων αντικειμένων-στόχους	16
Σχήμα 3.9: Διαθέσιμες επιλογές αντικειμένων-στόχους της Vuforia Engine	16
Σχήμα 3.10: Κατέβασμα βάσης σε μορφή πακέτου Unity	17
Σχήμα 3.11: Πρόσθεση Vuforia Engine αντικειμένων σε Unity project.....	18
Σχήμα 3.12: Άνοιγμα επιπλέον ρυθμίσεων συμπεριφοράς AR κάμερας	18
Σχήμα 3.13: Πρόσθεση κλειδιού άδειας χρήσης σε Vuforia AR camera.....	19
Σχήμα 3.14: Ορισμός webcam ως AR κάμερας	19
Σχήμα 3.15: Ορισμός στόχου σε αντικείμενο Image Target.....	20
Σχήμα 3.16: Ορισμός κεντρικού σημείου αναφοράς AR κάμερας	20
Σχήμα 3.17: Επιλογή πεδίου έμφασης της AR κάμερας στην ταχύτητα.....	20
Σχήμα 3.18: Ελάχιστη επιτρεπτή έκδοση Android του παιχνιδιού	21
Σχήμα 4.1: Φάκελος αποθήκευσης σκηνών παρούσας Π.Ε.	23
Σχήμα 4.2: Αρχική οθόνη παιχνιδιού της Π.Ε.	23
Σχήμα 4.3: Ιεραρχία σκηνής Start Screen	24
Σχήμα 4.4: Σκηνές παιχνιδιού Π.Ε. στα Build Settings	25
Σχήμα 4.5: Ορισμός μεθόδου openScen(1) σε OnClick() ενέργεια.	26
Σχήμα 4.6: Κώδικας MenuHandler script	26
Σχήμα 4.7: Βασικό μενού παιχνιδιού Π.Ε.....	27
Σχήμα 4.8: Ιεραρχία βασικού μενού παιχνιδιού Π.Ε.	28
Σχήμα 4.9: Εικόνες μενού και αρχικής οθόνης παιχνιδιού	28
Σχήμα 4.10: Ιδιότητα Scroll Rect αντικειμένου Scroll	29
Σχήμα 4.11: Ιδιότητες αντικειμένου Scrollbar	29
Σχήμα 4.12: Ιεραρχία κουμπιού στην σκηνή μενού.....	30
Σχήμα 4.13: Ιδιότητα Image με χρήση επιπρόσθετου χρωματισμού σε Panel.....	30
Σχήμα 4.14: Ιδιότητα Image σε αντικείμενο button.....	31
Σχήμα 4.15: Τελική απεικόνιση κουμπιού	31
Σχήμα 4.16: Χρήση αντικειμένων Image για την πρόσθεση logo εικόνων στον τίτλο.....	31

Σχήμα 5.1: Ιεραρχία αντικειμένου Image Target	33
Σχήμα 5.2: Πρώτη σκηνή παιχνιδιού Π.Ε.	34
Σχήμα 5.3: Πλατφόρμα Blender.....	34
Σχήμα 5.4: Δημιουργία τρισδιάστατου φούρνου μέσω του Blender.....	35
Σχήμα 5.5: Συνολικά αντικείμενα που αποτελούν τον τρισδιάστατο φουρνο	35
Σχήμα 5.6: Material ανοξίδοτου ατσαλιού με χρήση Texture.....	36
Σχήμα 5.7: Συνολικά Textures παιχνιδιού Π.Ε.	37
Σχήμα 5.8: Συνολικά Materials παιχνιδιού Π.Ε.	37
Σχήμα 5.9: Φούρνος έπειτα από την χρήσης materials και textures	37
Σχήμα 5.10: Συνολικά Prefabs παιχνιδιού Π.Ε.	38
Σχήμα 5.11: Σύρσιμο αυγών σε μπολ μίξης.....	38
Σχήμα 5.12: Στοιχεία αντικειμένου αυγών & μεταβλητές του script DragAndDrop.....	39
Σχήμα 5.13: Script DragAndDrop μέθοδος onTouch πρώτο μέρος	40
Σχήμα 5.14: Script DragAndDrop μέθοδος onTouch δεύτερο μέρος	41
Σχήμα 5.14: Script DragAndDrop μέθοδοι FixedUpdate & OnTriggerEnter	41
Σχήμα 5.15: Script CraftingRecipe scruct Items	42
Σχήμα 5.16: Scriptable objects συνταγών παιχνιδιού Π.Ε.	42
Σχήμα 5.17: Scriptable Object συνταγής κρεπών & τιμές μεταβλητών λίστας Materials	43
Σχήμα 5.18: Script CraftingRecipe κλάση CraftingRecipe τύπου ScriptableObject.....	43
Σχήμα 5.19: Script MainUiHandler μέθοδος showRecipe()	44
Σχήμα 5.20: Εμφάνιση διεπαφής Recipe.....	45
Σχήμα 5.21: Script MainUiHandler μέθοδος onCraftClick()	45
Σχήμα 5.22: Εμφάνιση μηνύματος επιτυχίας κατά την ολοκλήρωση συνταγής.....	46
Σχήμα 5.23: Εμφάνιση μηνύματος αποτυχίας κατά την ολοκλήρωση συνταγής.....	46
Σχήμα 5.24: Script BowlHandler μέθοδος ενεργοποίησης Numpad.....	47
Σχήμα 5.25: Script BowlHandler μέθοδοι OnTriggerEnter & Start.....	47
Σχήμα 5.26: Διεπαφή Numpad παιχνιδιού	48
Σχήμα 5.27: onClick γεγονός στο κουμπί Button1	48
Σχήμα 5.28: Ιεραρχία και δομή διεπαφής Numpad.....	49
Σχήμα 5.29: Script NumpadHandler μέθοδος onClickNumber()	49
Σχήμα 5.30: Script NumpadHandler μέθοδος passValueToBowl()	50
Σχήμα 5.31: Διεπαφή Questionnaire	50
Σχήμα 5.32: Struct QuestionAndAnswer	51
Σχήμα 5.33: Λίστα ερωτήσεων-απαντήσεων	51
Σχήμα 5.34: Script QuestionnaireHandler μέθοδος Start()	52
Σχήμα 5.35: Δομή απάντησης ερωτήσεως τύπου χρόνου	53
Σχήμα 5.36: Μέθοδος getCorrectTimeAfterChange() χρήση κλάσης DateTime	53
Σχήμα 5.37: Κατηγορία ερώτησης χρηματικού ποσού	54
Σχήμα 5.38: Script QuestionnaireHandler μέθοδος onAnswer μέρος πρώτο.....	54
Σχήμα 5.39: Script QuestionnaireHandler μέθοδος onAnswer μέρος δεύτερο	54
Σχήμα 5.40: Εμφάνιση διεπαφής αλλαγής μεταβλητών σε άσκηση-συνταγή	55
Σχήμα 5.41: Απάντηση άσκησης-συνταγής σε μορφή ποσότητας υλικού.....	56
Σχήμα 5.42: Script ChangeValues μέθοδος Start()	58
Σχήμα 5.43: Tag “ChangeableValue” κειμένου στο οποίο αναγράφεται μεταβλητή άσκησης	58
Σχήμα 5.44: Prefab αντικείμενο Input Field, χαρακτηριστικό tag “ChangedValue” και προσθήκη ως εξωτερική μεταβλητή στο script ChangeValues	58

Σχήμα 5.45: Script ChangeValues μέθοδος getValuesForChange() μέρος πρώτο.....	59
Σχήμα 5.46: Script ChangeValues μέθοδος getValuesForChange() μέρος δεύτερο.....	59
Σχήμα 5.47: Script ChangeValues μέθοδος setChangedValuesTypeQuestions().....	60
Σχήμα 5.48: Script ChangeValues μέθοδος setChangedValuesTypeRecipe().....	60
Σχήμα 5.49: Πρόσθεση μεθόδου resetChangedValues στο κουμπί “Ολοκλήρωσε”.....	61
Σχήμα 6.1: Εμφάνιση συνταγής κρεπών με μορφή ακεραίων αριθμών.....	63
Σχήμα 6.2: Εμφάνιση σκηνής μετά απο την επιστροφή από την διεπαφή συνταγής.....	64
Σχήμα 6.3: Διεπαφή επιλογής ποσότητας υλικού.....	64
Σχήμα 6.4: Διεπαφή ερώτησης-απάντησης Questionnaire.....	65
Σχήμα 6.5: Βηματική αλλαγή εμφάνισης μείγματος σε συνταγή τριών υλικών.....	65
Σχήμα 6.6: Διεπαφή συνταγής με χρήση κλασματικών αριθμών.....	66
Σχήμα 6.7: Εμφάνιση σκηνής παιχνιδιού κλασματικών αριθμών.....	66
Σχήμα 6.8: Εμφάνιση διεπαφής συμβουλευτικού μηνύματος στην σκηνή μέτρησης χρόνου.....	67
Σχήμα 6.9: Εμφάνιση σκηνής παιχνιδιού μέτρησης χρόνου.....	67
Σχήμα 6.10: Script TimeUiHandler μέθοδοι Awake() & Update().....	68
Σχήμα 6.11: Script TimeUiHandler μέθοδος δόμησης ώρας LeadingZero().....	68
Σχήμα 6.12: Απάντηση ερώτησης χρόνου μέσω της διεπαφής ερωτήσεων-απαντήσεων.....	69
Σχήμα 6.13: Διεπαφή επεξεργασίας ασκήσεων παιχνιδιού ChangeValues.....	69
Σχήμα 6.14: Αλλαγή μεταβλητής χρόνου της άσκησης σε “60 λεπτά” από “70 λεπτά”.....	70
Σχήμα 6.15: Εμφάνισης της αλλαγής της μεταβλητής χρόνου στην διεπαφή κειμένου.....	70
Σχήμα 6.16: Ολοκλήρωση σκηνής έπειτα από την ορθή απάντηση στην επεξεργασμένη ερώτηση....	70
Σχήμα 6.17: Συμβουλευτικό μήνυμα σκηνής τσαγιού.....	71
Σχήμα 6.18: Όψη σκηνής ετοιμασίας τσαγιού.....	71
Σχήμα 6.19: Animation τσαγιού.....	72
Σχήμα 6.20: Εξωτερικές μεταβλητές του script TeaCupAnimator.....	72
Σχήμα 6.21: Μέθοδοι Update() & OnTriggerEnter του script TeaCupAnimator.....	73
Σχήμα 6.22: Εκφώνηση άσκησης αναγωγής στην μονάδα.....	74
Σχήμα 6.23: Όψη άσκησης αναγωγής στην μονάδα με χρήση αντικειμένων παιχνιδιού.....	74
Σχήμα 6.24: Ορισμός λύσης ασκήσεως μέσω της διεπαφής Numpad.....	74
Σχήμα 6.25: Διαφοροποιημένο μήνυμα λάθους ασκήσεως αναγωγής.....	75
Σχήμα 6.26: Διεπαφή επεξεργασίας ασκήσεως αναγωγής.....	75
Σχήμα 6.27: Αλλαγή των μεταβλητών και λύσεων της άσκησης αναγωγής.....	76
Σχήμα 6.28: Όψη άσκησης αναγωγής έπειτα από την επεξεργασία της.....	76
Σχήμα 6.29: Συμβουλευτικό μήνυμα σκηνής σούπερμάρκετ.....	77
Σχήμα 6.30: Εξωτερική όψη αντικειμένου σουπερμάρκετ.....	77
Σχήμα 6.31: Διεπαφή αγοράς υλικών ShopPanel.....	78
Σχήμα 6.32: Script ShopUiHandler μέθοδοι FixedUpdate() & OpenShop().....	79
Σχήμα 6.33: Δομή κουμπιού-υλικού της διεπαφής ShopPanel.....	79
Σχήμα 6.34: Script Shop μέθοδοι Start(), addItemToBuy() & completeShopping().....	80
Σχήμα 6.35: Εμφάνιση ερώτησης κατηγορίας Money μέσω της διεπαφής QuestionnairePanel.....	81
Σχήμα 6.36: Εκφώνηση εξίσωσης-συνταγής donuts.....	81
Σχήμα 6.37: Όψη εξίσωσης με χρήση αντικειμένων παιχνιδιού.....	82
Σχήμα 6.38: Αλλαγή μεταβλητών άσκησης μέσω διεπαφής ChangeValues.....	82
Σχήμα 6.39: Πρώτη σκηνή υλοποίησης μείγματος καρυδόπιτας.....	83
Σχήμα 6.40: Εκφώνηση αναλογίας-συνταγής καρυδόπιτας.....	83
Σχήμα 6.41: Όψη αναλογίας με χρήση αντικειμένων παιχνιδιού.....	84

Σχήμα 6.42: Συνταγή σιροπιού τρίτης σκηνής.....	84
Σχήμα 6.43: Όψη σκηνής προετοιμασίας σιροπιού	84
Σχήμα 6.44: Εκφώνηση ποσοστού-συνταγής καρυδόπιτας	85
Σχήμα 6.45: Όψη άσκησης ποσοστού με χρήση αντικειμένων παιχνιδιού	85
Σχήμα 6.46: Εκφώνηση άσκησης-ερώτησης μετατροπής βάρους	86
Σχήμα 6.47: Όψη ασκήσεως μετατροπής με χρήση αντικειμένων παιχνιδιού.....	86
Σχήμα 6.48: Ερώτηση ασκήσεως μετατροπής μέσω της διεπαφής QuestionnairePanel.....	87
Σχήμα 6.49: Εκφώνηση άσκησης μέσου όρου συνταγής μικρών κέικ	87
Σχήμα 6.50: Όψη άσκησης μέσου όρου με χρήση αντικειμένων παιχνιδιού	88
Σχήμα 6.51: Αλλαγή μεταβλητών αθροίσματος και λύσης, μέσω της διεπαφής ChangeValues.....	88
Σχήμα 6.52: Όψη προετοιμασίας μείγματος σοκολατόπιτας	89
Σχήμα 6.53: Εμφάνιση βημάτων τελικού σταδίου συνταγής μέσω της διεπαφής TipPanel	90
Σχήμα 6.54: Υλοποίηση πρώτου βήματος και εμφάνιση αντίστοιχης ερώτησης περιμέτρου	90
Σχήμα 6.55: Υλοποίηση δεύτερου βήματος και εμφάνιση αντίστοιχης ερώτησης εμβαδόν	90
Σχήμα 6.56: Υλοποίηση τρίτου βήματος και εμφάνιση τελικής ερώτησης όγκου	90
Σχήμα 6.57: Script ChocolateCakeHandler.....	91

Συντομογραφίες

Π.Ε.	Πτυχιακή Εργασία
AR	Επαυξημένη Πραγματικότητα
VR	Εικονική Πραγματικότητα
MR	Μικτή Πραγματικότητα
UI	User Interface
3D	Τρισδιάστατο

Κεφάλαιο 1ο: Εισαγωγή

Η πτυχιακή εργασία, “ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΜΕ ΣΤΟΧΟ ΤΗΝ ΒΕΛΤΙΩΣΗ ΤΩΝ ΜΑΘΗΤΩΝ ΣΤ’ ΔΗΜΟΤΙΚΟΥ ΣΤΟ ΜΑΘΗΜΑ ΤΩΝ ΜΑΘΗΜΑΤΙΚΩΝ” πρόκειται για ένα παιχνίδι επαυξημένης πραγματικότητας το οποίο συνδυάζει απλές συνταγές μαγειρικής με βασικά προβλήματα μαθηματικών τα οποία πηγάζουν από το βιβλίο του μαθήματος της Στ’ δημοτικού. Η πλατφόρμα που χρησιμοποιήθηκε για την δημιουργία της εφαρμογής είναι το Unity μαζί με την χρήση του πακέτου Vuforia. Μέσω αυτού του συνδυασμού ο χρήστης μπορεί να χρησιμοποιήσει την κάμερα μιας έξυπνης φορητής συσκευής ώστε να εμβαθύνει την κατανόηση του πάνω στα μαθηματικά με διαδραστικό και δημιουργικό τρόπο.

Η δομή της Π.Ε ακολουθεί όπως παρουσιάζεται παρακάτω:

Στο πρώτο κεφάλαιο παρουσιάζονται τα κεφάλαια της Π.Ε. καθώς και μίας μικρής περίληψης γι’αυτών.

Στο δεύτερο κεφάλαιο, περιγράφονται οι τεχνολογίες της εκτεταμένης πραγματικότητας (XR), οι υποκατηγορίες που την αποτελούν, καθώς και οι χρήσεις της. Επίσης, γίνεται αναφορά στους λόγους που η χρήση αυτών των τεχνολογιών, διευκολύνουν και βελτιώνουν διάφορους επαγγελματικούς τομείς όπως και η εκπαίδευση.

Στο τρίτο κεφάλαιο, γίνεται μία ανάλυση των τεχνολογιών και λογισμικών που χρησιμοποιήθηκαν για την υλοποίησης της Π.Ε. Επίσης, εξηγούνται τα απαραίτητα βήματα τα οποία πάρθηκαν για την εγκατάσταση των λογισμικών αυτών και ιδιαίτερα η εγκατάσταση του πακέτου Vuforia στην πλατφόρμα του Unity, καθώς και ο τρόπος με τον οποίο αλληλεπιδρούν.

Στο τέταρτο κεφάλαιο, επεξηγείται ο τρόπος δημιουργίας των βασικών διεπαφών του μενού και της αρχικής οθόνης του παιχνιδιού. Επίσης, γίνεται μία μικρή ανάλυση στον τρόπο με τον οποίο λειτουργούν οι σκηνές σε ένα Unity project.

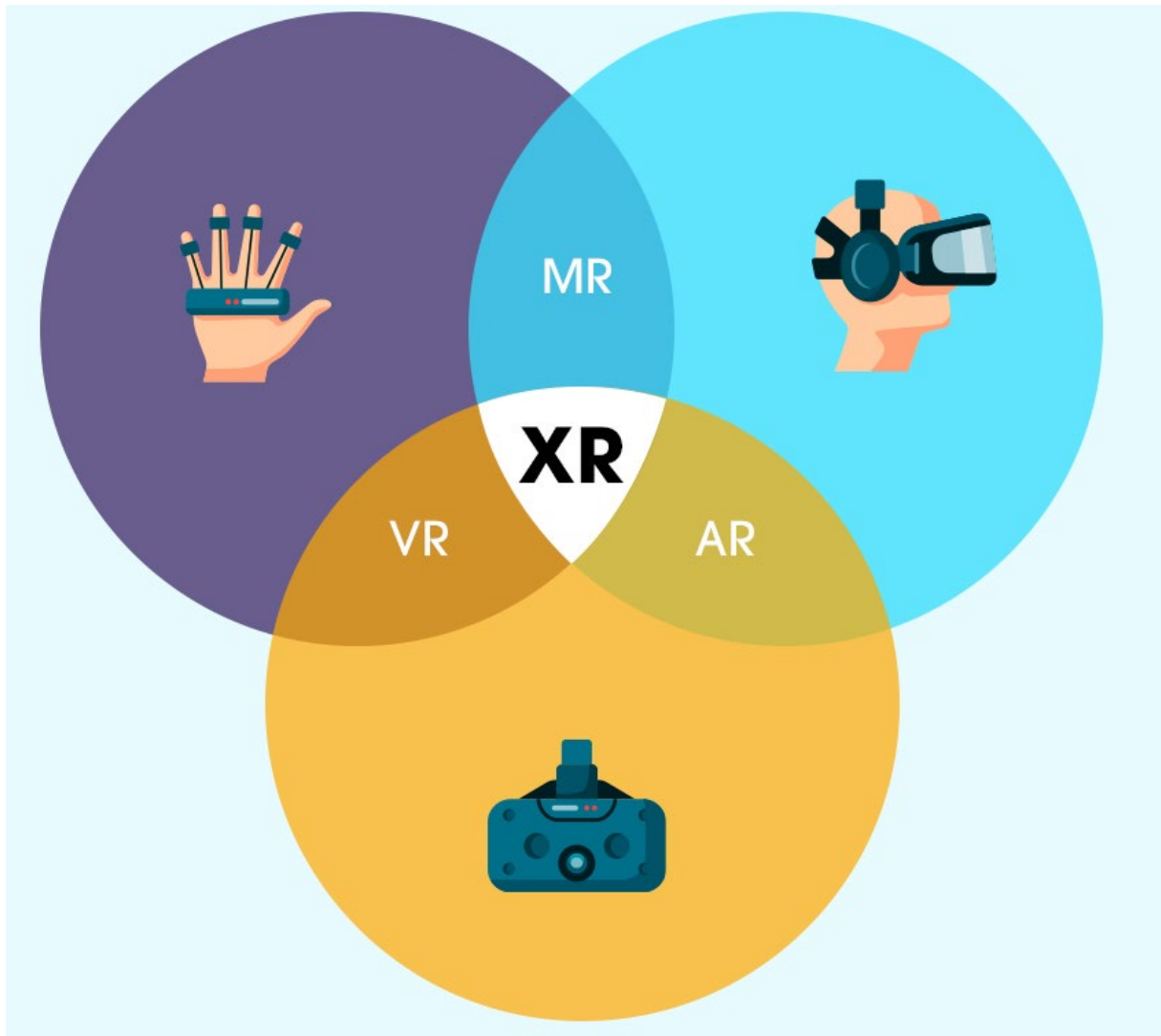
Στο πέμπτο κεφάλαιο, αναλύονται οι λειτουργίες και τα βασικά στοιχεία τα οποία χρησιμοποιούνται στο σύνολο του παιχνιδιού. Συγκεκριμένα, περιγράφεται η δημιουργία των 3D αντικειμένων, ο χρωματισμός τους, οι βασικές διεπαφές του παιχνιδιού καθώς και τα σημαντικά scripts τα οποία διεκπεραιώνουν τις λειτουργίες του.

Στο έκτο κεφάλαιο της Π.Ε., έχοντας αναλύσει τις κύριες λειτουργίες του παιχνιδιού γίνεται μία ανάλυση του περιεχομένου του και συγκεκριμένα τους τρόπους με τους οποίους συνδυάστηκε η διαδικασία της μαγειρικής με την θεωρία των μαθηματικών για την δημιουργία του.

Στο έβδομο και τελευταίο κεφάλαιο, παρουσιάζονται κάποια τελικά συμπεράσματα τα οποία πάρθηκαν κατά την διαδικασία της υλοποίησης της Π.Ε., καθώς και ορισμένες βελτιώσεις οι οποίες μπορούν να γίνουν σε αυτή.

Κεφάλαιο 1ο:

Κεφάλαιο 2ο: Εκτεταμένη πραγματικότητα και χρήσεις



Σχήμα 2.1: Εκτεταμένη Πραγματικότητα και υποκατηγορίες

2.1 Εισαγωγή

Η εκτεταμένη πραγματικότητα πρόκειται για ένα κλάδο της τεχνολογίας ο οποίος ανθίζει καθημερινά. Όλο και περισσότερο βλέπουμε τεχνολογίες οι οποίες κάνουν χρήση της στην καθημερινότητα μας, είτε στον τομέα της αναψυχής είτε στον τομέα της εκπαίδευσης και όχι μόνο. Προτού όμως μιλήσουμε για τις χρήσεις και τα επιμέρους στοιχεία της, θα πρέπει να ξεκαθαρίσουμε τι αποτελεί και ορίζει την εκτεταμένη πραγματικότητα. Ο όρος της εκτεταμένης πραγματικότητας μπορεί να περιγράψει πάρα πολλές εφαρμογές, αυτό συμβαίνει γιατί πρόκειται για έναν όρο ομπρέλα ο οποίος εμπεριέχει τρεις βασικές τεχνολογίες. Αυτές είναι η εικονική πραγματικότητα (VR), η επαυξημένη πραγματικότητα (AR) και η μικτή πραγματικότητα (MR) [3]. Και οι τρεις τεχνολογίες έχουν στόχο την δημιουργία ενός ενιαίου συστήματος το οποίο θα αναπαράγει κάποιο φάσμα της αισθητηριακής αντίληψης. Δηλαδή, κάθε τεχνολογία με τον δικό της τρόπο εμπλουτίζει είτε προσομοιάζει την, ή μία πραγματικότητα.

2.2 Εικονική Πραγματικότητα

Πέρα από τα τρισδιάστατα γραφικά, τα δύο βασικά στοιχεία τα οποία περιγράφουν την εικονική πραγματικότητα (VR) είναι η βύθιση και η διαδραστικότητα [4]. Η βύθιση αφορά την απομάκρυνση του χρήστη από κάθε είδους εξωτερικού περισπασμού, ώστε όλη η συγκέντρωση του να βρίσκεται στις πληροφορίες που του δίνονται εντός του περιβάλλοντος. Η διαδραστικότητα περιγράφει την ικανότητα του χρήστη να αλληλεπιδρά με αντικείμενα και γεγονότα τα οποία προκύπτουν πάλι εντός του περιβάλλοντος το οποίο προσομοιάζεται. Για την λειτουργία αυτών των τριών χαρακτηριστικών, γίνεται χρήση ειδικού εξοπλισμού ο οποίος εμπεριέχει γυροσκόπιο και αισθητήρες κίνησης για τον εντοπισμό των σωματικών κινήσεων του χρήστη, κυρίως για μέρη όπως κεφάλι και χέρια.



Σχήμα 2.2: Χρήση VR εξοπλισμού

Όπως φαίνεται και στο σχήμα 2.2 η χρήση VR εξοπλισμού γίνεται κυρίως με την χρήση μίας μάσκας και δύο χειριστηρίων. Η μάσκα εμπεριέχει δύο φακούς υψηλής ευκρίνειας, ώστε να προσομοιάζει στον χρήστη έναν τρισδιάστατο χώρο και ένα γυροσκόπιο ώστε να αντιλαμβάνεται η εφαρμογή τις κινήσεις του κεφαλιού του. Επίσης, γυροσκόπια εμπεριέχουν και τα χειριστήρια ώστε να αναγνωρίζονται οι κινήσεις των χεριών του χρήστη.

2.3 Επαυξημένη Πραγματικότητα

Η ουσία της επαυξημένης πραγματικότητας (AR) είναι να εμπλουτίζει το περιβάλλον του χρήστη με εικονικά αντικείμενα. Ουσιαστικά συνδυάζει αληθινά και εικονικά αντικείμενα σε ένα πραγματικό περιβάλλον, σε πραγματικό χρόνο. Η λειτουργία της εικονικής πραγματικότητας επιτυγχάνεται με την χρήση μέσων, τα οποία προβάλλουν και τον πραγματικό κόσμο και τα εικονικά στοιχεία στον ίδιο χρόνο [1], [3]. Με αυτόν τον τρόπο, ο χρήστης έχει την δυνατότητα να βιώσει και τα δύο μαζί.



Σχήμα 2.3: Χρήση AR εφαρμογής μέσω Smartphone

Όπως φαίνεται και στο σχήμα 2.3 το μόνο που χρειάζεται για την χρήση AR εφαρμογών είναι μία έξυπνη συσκευή. Οι εφαρμογές χρησιμοποιούν την κάμερα της εκάστοτε συσκευής, μέσω της οποίας σκανάρουν είτε το περιβάλλον είτε ειδικούς κωδικούς QR, ώστε να εφαρμόσουν τα εικονικά αντικείμενα που διαθέτουν.

2.4 Μικτή Πραγματικότητα

Ο όρος μικτή πραγματικότητα (MR) αναφέρεται στον συνδυασμό των δύο προαναφερθέντων τεχνολογιών και η εξέλιξη τους σε μία νέα κατηγορία. Με άλλα λόγια η μικτή πραγματικότητα συνδυάζει τον ανθρώπινο, υπολογιστικό και περιβαλλοντολογικό παράγοντα σε μία μόνο τεχνολογία. Πρόκειται δηλαδή για την μείξη πραγματικού και ψηφιακού κόσμου, μειώνοντας τους περιορισμούς των επί μέρους τεχνολογιών AR και VR [3], [9]. Η χρήση εφαρμογών MR επιτυγχάνονται κυρίως μέσω της χρήσης ολογραφικών συσκευών, όπως στο σχήμα 2.4 και 2.5, είτε μέσω της χρήσης συσκευών VR, όπως στο σχήμα 2.2.



Σχήμα 2.4: Ολογραφικά γυαλιά HoloLens



Σχήμα 2.5: Ολογραφική συσκευή Holus

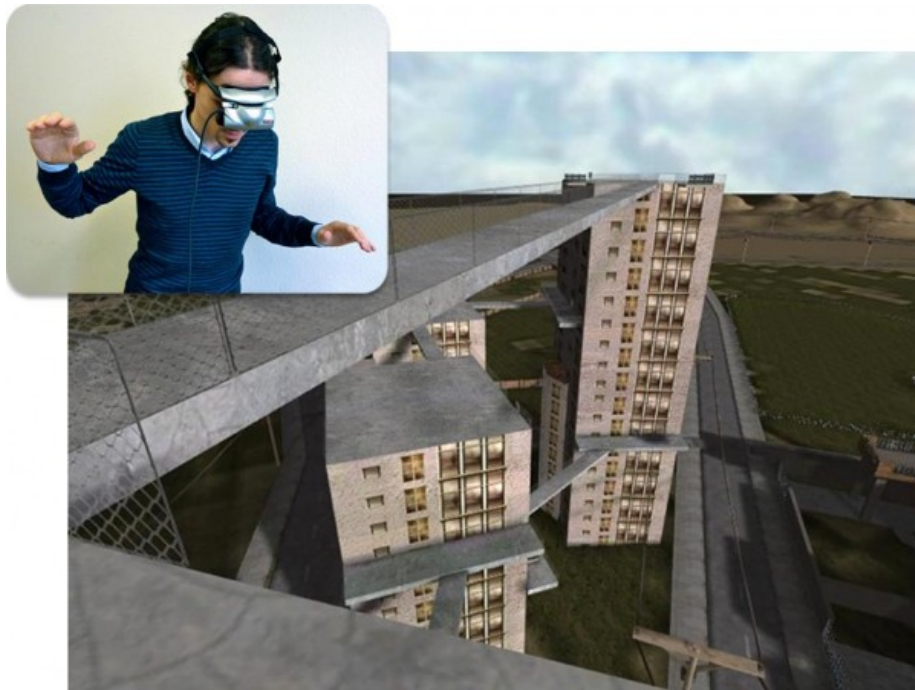
Ο τομέας της μικτής πραγματικότητας πρόκειται για ένα τομέα ο οποίος έχει αρχίσει να αναπτύσσεται τα τελευταία μόλις χρόνια. Αυτό έχει ως αποτέλεσμα να υπάρχει μεγάλη ποικιλομορφία στις συσκευές που χρησιμοποιούν αυτήν την τεχνολογία. Στα παραπάνω σχήματα υπάρχουν δύο εντελώς διαφορετικές συσκευές οι οποίες εκτελούν παρόμοιο έργο, αλλά με διαφορετικό τρόπο. Το σχήμα 2.4 παρουσιάζει ολογραφικά γυαλιά τα οποία διαθέτουν μία διαφανής οθόνη, μέσα από την οποία ο χρήστης μπορεί να βλέπει τα ψηφιακά αντικείμενα στον πραγματικό κόσμο. Αντίθετα στο σχήμα 2.5 πρόκειται για μία συσκευή όπου παρουσιάζει τις ψηφιακές πληροφορίες σε μορφή ολογράμματος χωρίς να απαιτεί την χρήση κάποιου επιπλέον εξοπλισμού από την πλευρά του χρήστη [9].

2.5 Χρήσεις της Εκτεταμένης Πραγματικότητας

Όπως μπορούμε να καταλάβουμε, οι παραπάνω τεχνολογίες μπορούν και έχουν αρχίσει να εφαρμόζονται σε αρκετούς τομείς της καθημερινότητας μας.

2.5.1 Ιατρική

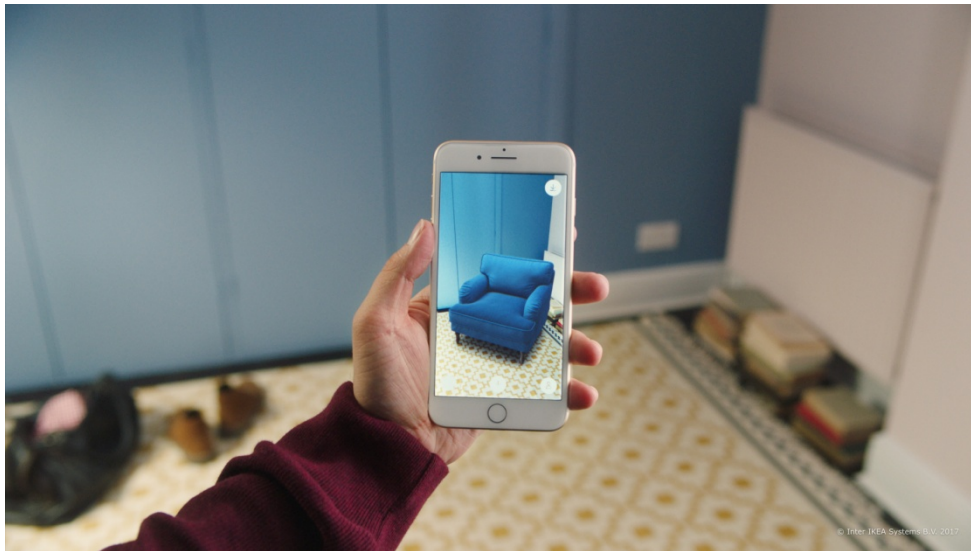
Ένας πολύ σημαντικός κλάδος ο οποίος βελτιώνεται όλο και περισσότερο με την εξέλιξη της τεχνολογίας είναι ο κλάδος της ιατρικής. Συστηματικά τεχνολογίες εκτεταμένης πραγματικότητας εισέρχονται σε αυτόν τον κλάδο, ώστε να βοηθήσουν στην βελτίωση ήδη υπαρχόντων θεραπειών ή στην ανακάλυψη νέων. Ένα πολύ καλό παράδειγμα είναι ένας νέος τρόπος ψυχολογικής θεραπείας με την χρήση VR τεχνολογιών. Η θεραπεία ονομάζεται Virtual Reality Exposure Therapy (VRET) και χρησιμοποιείται σε ασθενείς που πάσχουν από προβλήματα όπως αγχώδες διαταραχές, κρίσεις πανικού και διαταραχή μετατραυματικού στρες (PTSD) [5]. Χρησιμοποιώντας ένα ελεγχόμενο προσομοιωμένο περιβάλλον ο ασθενής μπορεί να έρχεται σταδιακά σε επαφή με καταστάσεις οι οποίες του προκαλούν πρόβλημα μέχρις ότου να μπορέσει, ως ένα βαθμό, να τις ξεπεράσει. Πρόκειται δηλαδή για έναν ήδη υπάρχοντα τρόπον θεραπείας, της σταδιακής έκθεσης του ασθενή στο αντικείμενο ή την κατάσταση που τον διαταράσσει, με στόχο την βελτίωση της διαχείρισης του προβλήματος του, καθώς και την κατανόηση του από αυτόν. Στο παρελθόν η θεραπεία αυτή εφαρμόζονταν με την χρήση φωτογραφιών, κάτι το οποίο δεν ήταν αποτελεσματικό, ειδικά για άτομα τα οποία δεν κατέχουν καλές ικανότητες οπτικοποίησης. Πλέον όμως με την χρήση της εικονικής πραγματικότητας, επιτρέπεται η δημιουργία καλύτερα ελεγχόμενων και πολύ πιο ρεαλιστικών καταστάσεων σε μικρό μάλιστα κόστος.



Σχήμα 2.6: Θεραπεία VRET σε ασθενή που πάσχει από υψοφοβία

2.5.2 Μάρκετινγκ

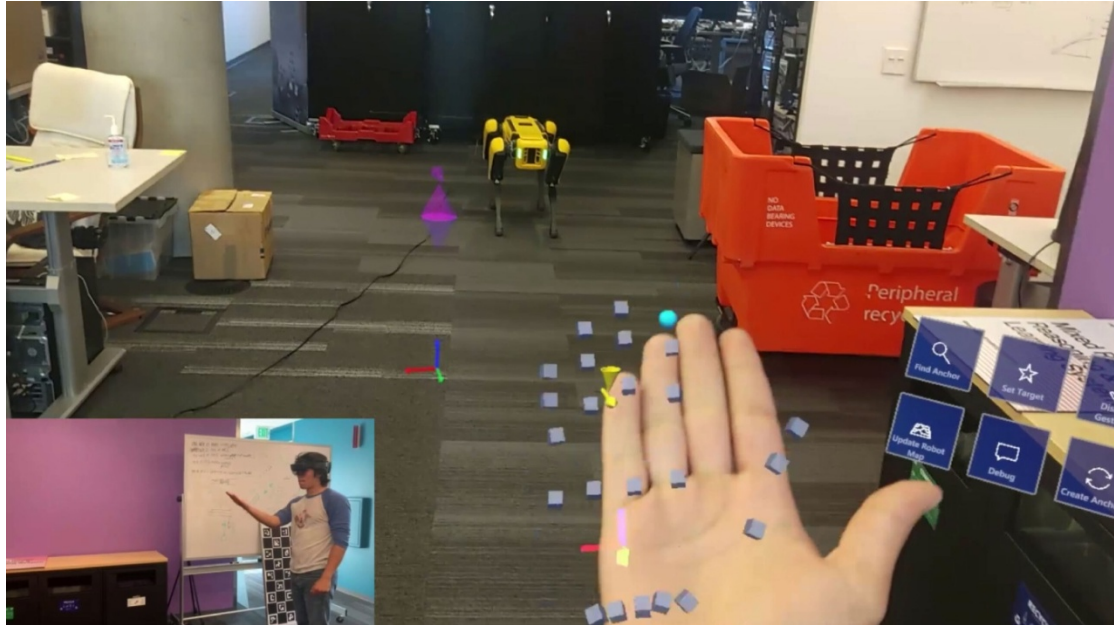
Άλλος ένας ραγδαία εξελισσόμενος κλάδος είναι προφανώς το μάρκετινγκ και η διαφήμιση. Τα τελευταία χρόνια παρατηρούμε όλο και πιο έξυπνους τρόπους στον τομέα του μάρκετινγκ. Πλέον ο περισσότερος κόσμος διαθέτει τουλάχιστον μία έξυπνη συσκευή, όπως ένα Smartphone ή ένα Tablet, με αποτέλεσμα πολλές εταιρίες να έχουν την δυνατότητα της πολύ πιο εύκολης και φθηνότερης διαφήμισης των προϊόντων τους σε πιθανούς αγοραστές, μέσω της χρήσης AR τεχνολογιών. Χαρακτηριστικό παράδειγμα είναι η εφαρμογή IKEA Place της γνωστής εταιρίας επίπλων [6]. Μέσω της εφαρμογής αυτής, ο χρήστης έχει την δυνατότητα όχι μόνο να αναζητήσει ή να περιηγηθεί στο σύνολο των αντικειμένων και επίπλων που προσφέρει η εταιρία, αλλά και να δει στον πραγματικό κόσμο το έπιπλο το οποίο σκέφτεται να αγοράσει. Αυτό έχει σαν αποτέλεσμα την πολύ πιο αποτελεσματική διαφήμιση καθώς κάθε πιθανός αγοραστής μπορεί να δοκιμάζει το προϊόν το οποίο σκέφτεται να αγοράσει στον χώρο του χωρίς επιπλέον κόστος ή κόπο.



Σχήμα 2.7: Χρήση AR εφαρμογής IKEA Place

2.5.3 Ρομποτική

Ο κλάδος της ρομποτικής είναι ένας κλάδος ο οποίος χαρακτηρίζεται πολύ υποσχόμενη καθώς εξελίσσεται μέρα με την μέρα. Ένας από τους λόγους της αυξημένης εξέλιξης αυτής, είναι η προσθήκη τεχνολογιών μικτής πραγματικότητας στην διαδικασία ανάπτυξης και ελέγχου. Οι τεχνολογίες MR δημιουργούν έναν χώρο στον οποίο τόσο το φυσικό περιβάλλον όσο και εικονικά στοιχεία συνυπάρχουν. Αυτό δίνει την δυνατότητα της πολύ πιο ευέλικτης ανάπτυξης καθώς και δοκιμής αλγορίθμων ελέγχου σε εφαρμογές ρομποτικής. Δεδομένου της ευελιξίας που παρέχει το MR, οι χρήστες έχουν την δυνατότητα να διαχειρίζονται τα εικονικά και τα πραγματικά στοιχεία, δημιουργώντας διάφορα ξεχωριστά πειράματα με βάση τους περιορισμούς και τις απαιτήσεις που χρειάζονται [16]. Όπως γίνεται αντιληπτό, το ρίσκο της ασφάλειας που υπάρχει κατά την αλληλεπίδραση μεταξύ του ανθρώπου και της εκάστοτε μηχανής-ρομπότ εξαφανίζεται ολοκληρωτικά. Τα ρομπότ που αναπτύσσονται μπορούν να αλληλεπιδρούν με εικονικούς ανθρώπους σε διάφορα πειράματα. Αυτό έχει ως αποτέλεσμα τον ασφαλή έλεγχο κατά την ανάπτυξη σε όσα διαφορετικά σενάρια χρειάζονται.



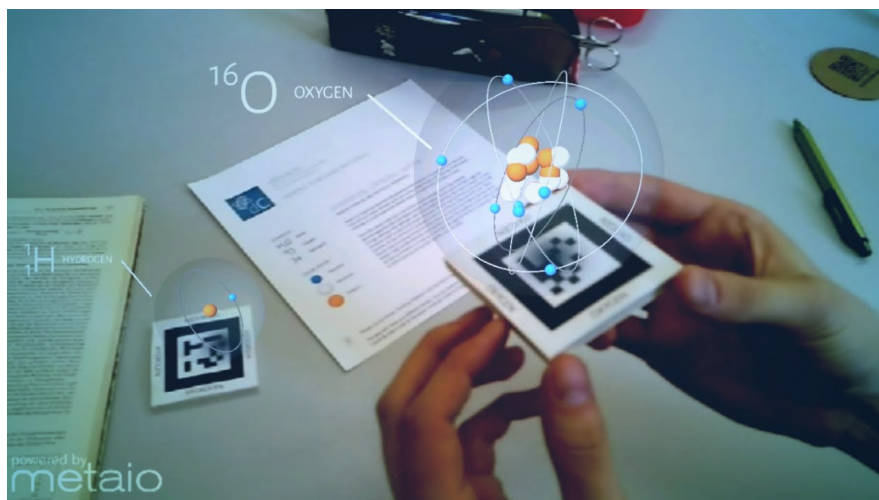
Σχήμα 2.8: Αλληλεπίδραση MR και μηχανής ρομπότ

2.5.4 Εκπαίδευση

Ο τομέας της εκπαίδευσης είναι ίσως και από τους πιο σημαντικούς κλάδους στους οποίους έχουν εισέλθει οι τεχνολογίες της εκτεταμένης πραγματικότητας. Από την εκπαίδευση μαθητών σε σχολεία, μέχρι και νέων εργαζομένων σε εταιρίες, παρατηρούμε μία ραγδαία εξέλιξη στην χρήση αυτών των τεχνολογιών.

2.5.4.1 Σχολική Εκπαίδευση

Πλέον η εκπαίδευση περνάει σε νέο στάδιο διάδρασης και αποτελεσματικότητας, μαθητές έχουν την δυνατότητα να κατανοήσουν καλύτερα έννοιες με διάφορους διασκεδαστικούς και δημιουργικούς τρόπους. Επίσης, σημαντική είναι και η ευκολία από καθηγητές στο να κάνουν πιο διασκεδαστική την διαδικασία της εκπαίδευσης, συνδυάζοντας το παιχνίδι με την μάθηση [2], όπως για παράδειγμα την μαγειρική με τα μαθηματικά. Έτσι, οι μαθητές όχι μόνο κατανοούν καλύτερα το εκάστοτε μάθημα αλλά έρχονται επίσης σε επαφή με νέες τεχνολογίες.



Σχήμα 2.8: Χρήση AR εφαρμογής σε μάθημα Χημείας

2.5.4.2 Εργασιακή Εκπαίδευση - Εξάσκηση

Πέρα από την σχολική εκπαίδευση, άλλος ένας κλάδος στην κατηγορία αυτή είναι η εργασιακή εκπαίδευση. Στον κλάδο αυτό, βλέπουμε μία μεγάλη χρήση τεχνολογιών προσομοίωσης στην εκπαίδευση και τακτική εξάσκηση εργαζομένων σε ένα μεγάλο εύρος εργασιών. Αυτή η χρήση γίνεται κατά κύριο λόγο από εργασίες κατά τις οποίες υπάρχουν υψηλοί κίνδυνοι ή και μεγάλη απαίτηση χρόνου. Χαρακτηριστικά παραδείγματα είναι εργασίες όπως η διάσωση ανθρώπων (πυροσβεστική, μονάδες πρώτων βοηθειών), ο στρατός, καθώς και η ιατρική. Οι προσομοιώσεις μέσω της χρήσης τεχνολογιών εικονικής πραγματικότητας (VR), δίνουν την ευκαιρία της συνεχούς εξάσκησης και ασφαλούς εκπαίδευσης προσωπικού σε πολύ απαιτητικές καταστάσεις, αφαιρώντας από αυτές τις συνέπειες αποτυχίας, καθώς και του μεγάλου κόστους που θα απαιτούσαν αν πραγματοποιούντουσαν σε πραγματικό περιβάλλον [5], [7], [8].



Σχήμα 2.9: Χρήση VR εξοπλισμού για την εκπαίδευση πυροσβέστη

2.6 Επίλογος

Μέρα με την μέρα, όλο και περισσότερο βλέπουμε τεχνολογίες εκτεταμένης πραγματικότητας να διευκολύνουν και να εμπλουτίζουν την καθημερινότητα μας. Νέες ευκαιρίες εμφανίζονται μέσα από τις τεχνολογίες αυτές στην βελτίωση της ποιότητας ζωής, μέσω της εφαρμογής τους σε τομείς όπως η ιατρική και η ψυχολογία. Πέρα από αυτούς τους τομείς, ιδιαίτερη είναι η συμβολή των τεχνολογιών αυτών στον ευρύτερο τομέα της εκπαίδευσης, καθώς με μικρό κόστος μπορεί να αυξηθεί σε μεγάλο βαθμό η αποτελεσματικότητα, διατηρώντας επίσης ένα ασφαλές περιβάλλον. Συμπερασματικά, οι τεχνολογίες εκτεταμένης πραγματικότητας αποτελούν μια πολύ αξιόλογη επιλογή στον τομέα της εκπαίδευσης, κάτι το οποίο αξιοποιεί και η παρούσα Π.Ε.

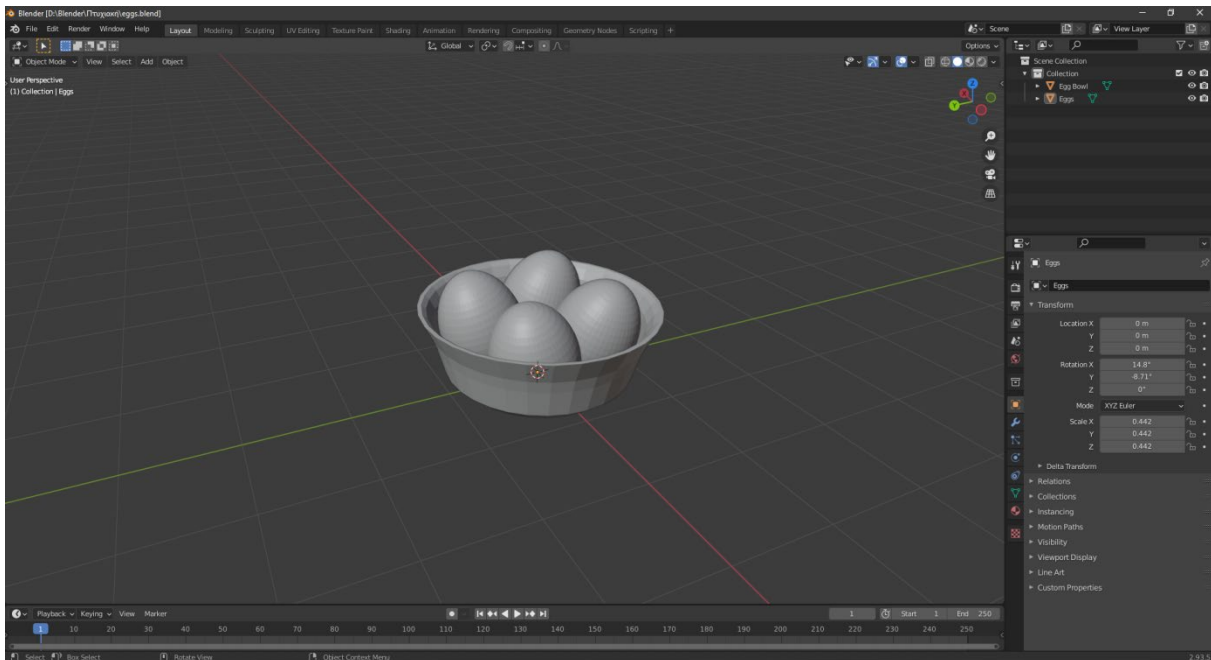
Κεφάλαιο 3ο: Προγράμματα και πακέτα

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύονται όλα τα προγράμματα και πακέτα τα οποία χρησιμοποιήθηκαν για την δημιουργία της Π.Ε. Κατά κύριο λόγο χρησιμοποιήθηκαν τα προγράμματα Unity [10] και Blender [11]. Το πρώτο για την δημιουργία του παιχνιδιού σε προγραμματιστικό επίπεδο και το δεύτερο για τα τρισδιάστατα μοντέλα που χρησιμοποιούνται στο παιχνίδι. Επίσης, χρησιμοποιήθηκε το πακέτο Vuuforia [12] για την υλοποίηση του AR κομματιού της εφαρμογής. Παρακάτω αναλύονται όλα τα βήματα εγκατάστασης και ρύθμισης των προαναφερθέντων.

3.2 Blender

Το Blender πρόκειται για ένα ελεύθερο λογισμικό ανοιχτού κώδικα (open source), το οποίο χρησιμοποιείται για την σχεδίαση τρισδιάστατων γραφικών και όχι μόνο. Η έκδοση που χρησιμοποιήθηκε στην παρούσα Π.Ε. είναι η 2.93.5.



Σχήμα 3.1: Δημιουργία τρισδιάστατου μπολ με αυγά μέσω του λογισμικού Blender

3.3 Unity

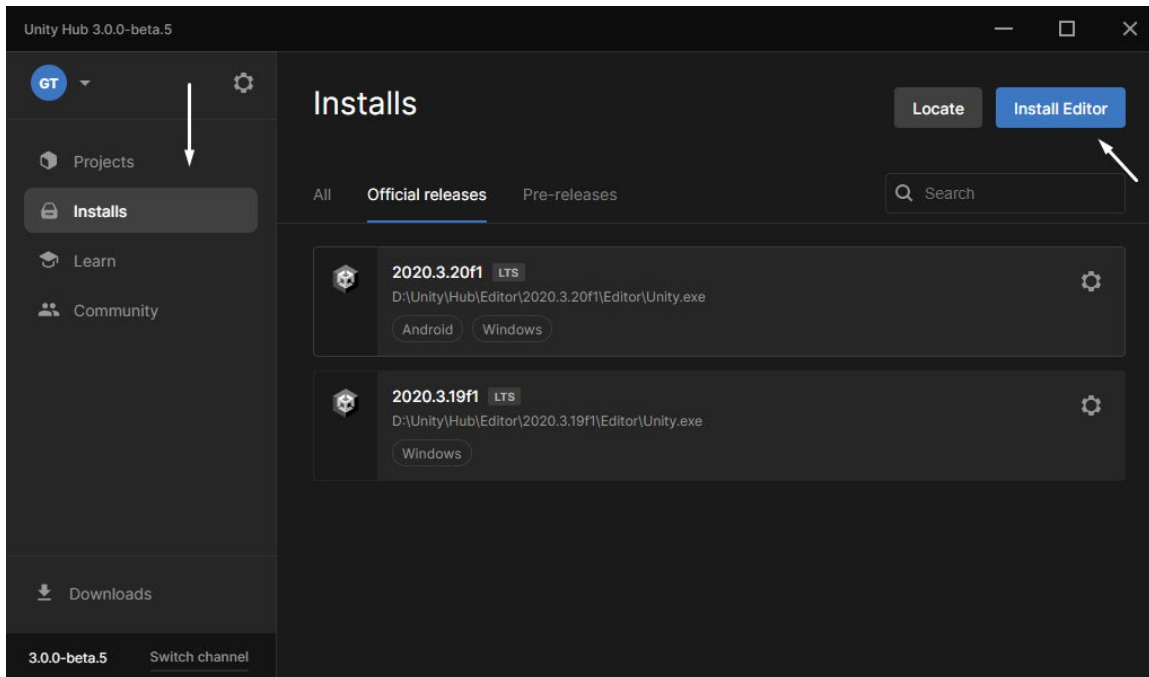
Το βασικό λογισμικό το οποίο χρησιμοποιήθηκε για την δημιουργία της Π.Ε. είναι το Unity. Πρόκειται για ένα επίσης ελεύθερο λογισμικό και χρησιμοποιείται για την δημιουργία παιχνιδιών, κινουμένων σχεδίων (animation), ταινιών και πολλών άλλων. Η γλώσσα προγραμματισμού που χρησιμοποιείται σε αυτό το περιβάλλον είναι η C#, αλλά πολλές λειτουργίες όπως η δημιουργία σκηνών και διεπαφής χρήστη (UI) πραγματοποιούνται μέσω του UI του Unity.

3.3.1 Εγκατάσταση Λογισμικού Unity

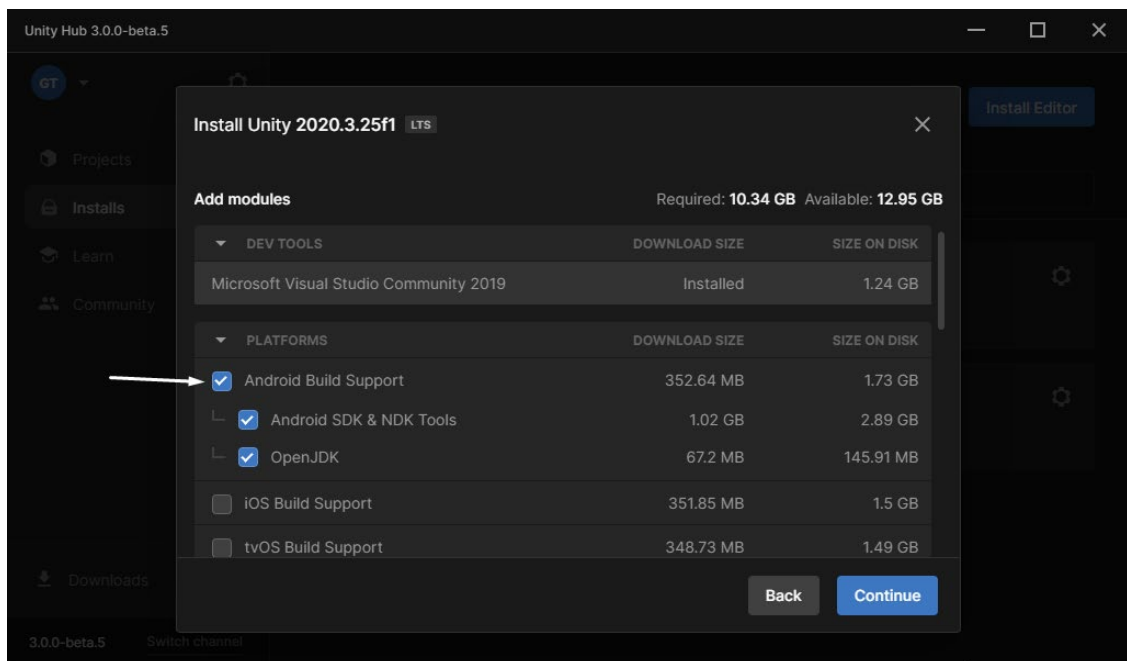
Η πρώτη πράξη για την εγκατάσταση του λογισμικού είναι η εγκατάσταση του Unity Hub [10] το οποίο μπορούμε να βρούμε στην ιστοσελίδα του. Η έκδοση που χρησιμοποιήθηκε σε αυτήν την Π.Ε. είναι η 3.0.0. Μέσω του Unity Hub μπορούμε να εγκαταστήσουμε την πλατφόρμα Unity καθώς και το

Κεφάλαιο 3ο:

απαραίτητο πακέτο Android Build Support, ώστε να το παιχνίδι μας να είναι συμβατό με συσκευές Android. Η έκδοση Unity που χρησιμοποιήθηκε είναι η 2020.3.20f1 και οι διαδικασίες αναδεικνύονται στα παρακάτω σχήματα 3.2 και 3.3.



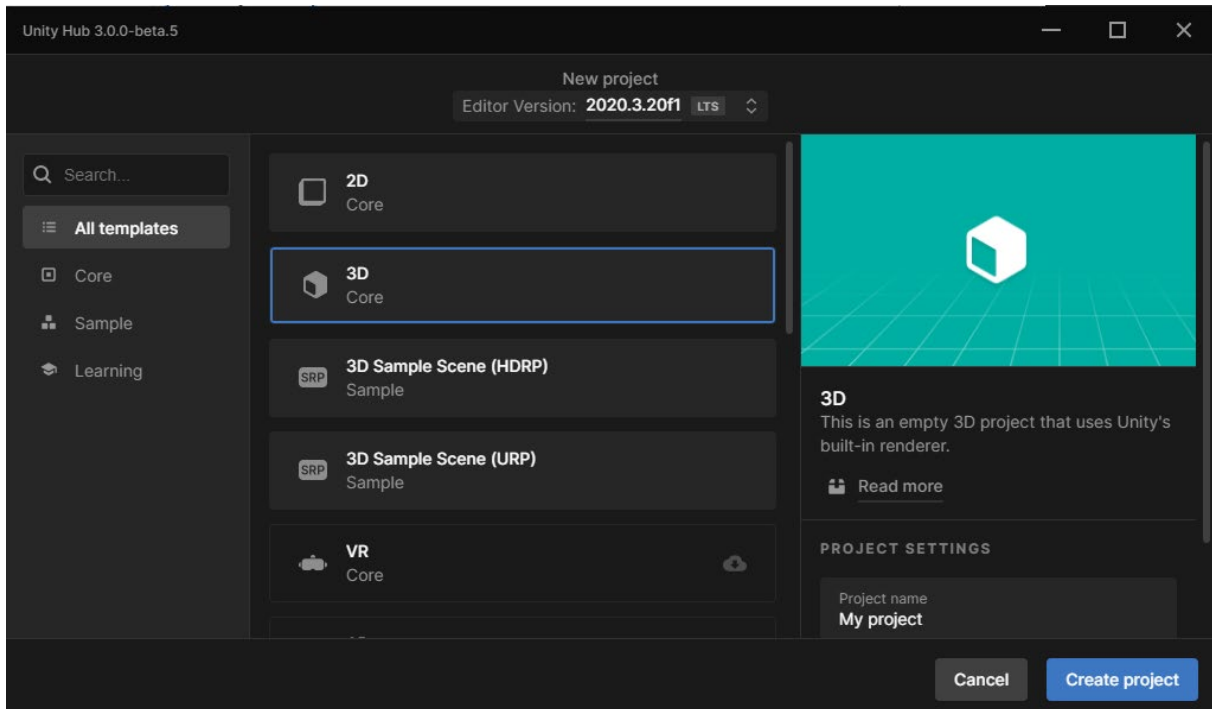
Σχήμα 3.2: Εγκατάσταση Unity μέσω της πλατφόρμας Unity Hub



Σχήμα 3.3: Εγκατάσταση πακέτου συμβατότητας Android μέσω Unity Hub

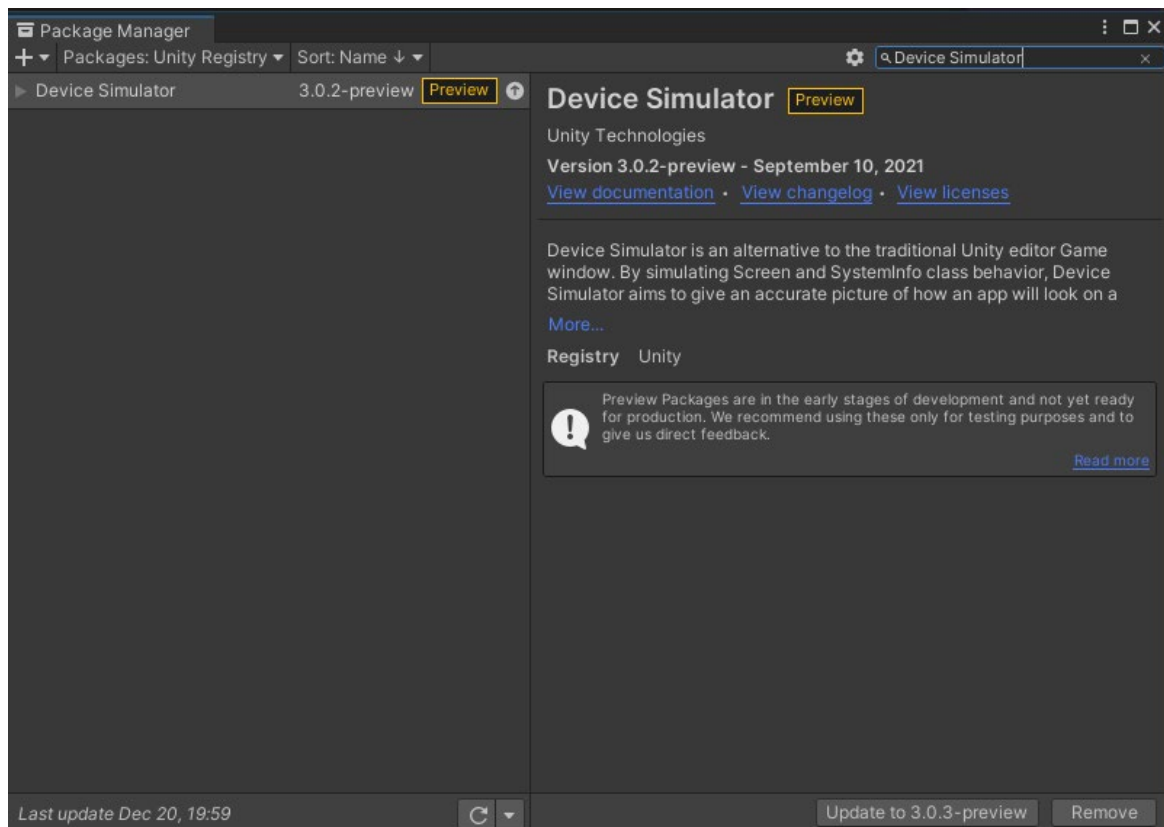
3.3.2 Δημιουργία Προγράμματος και Εγκατάσταση Πακέτου Device Simulator

Έπειτα από την εγκατάσταση του Unity μπορούμε να προχωρήσουμε στην δημιουργία του προγράμματος (project) και την εγκατάσταση του πακέτου Device Simulator. Αρχικά, η πρώτη επιλογή που πρέπει να παρθεί είναι το τί είδους εφαρμογή θα δημιουργήσουμε. Στην συγκεκριμένη εργασία επιλέχθηκε σαν βάση ένα 3D core project, δηλαδή ένα project το οποίο χρησιμοποιεί τις τρεις διαστάσεις με απλά γραφικά.



Σχήμα 3.4: Δημιουργία 3D Προγράμματος

Αφού έχει δημιουργηθεί το project, μια προαιρετική αλλά καλή επιλογή είναι η εγκατάσταση του πακέτου Device Simulator. Μέσω αυτού, έχουμε την δυνατότητα να προσομοιάζουμε την λειτουργία μιας έξυπνης συσκευή μέσα από το Unity. Αυτό αποτελεί μια αρκετά διευκολυντική επιλογή καθώς μπορούμε να τρέχουμε το παιχνίδι όπως θα έτρεχε σε μία έξυπνη συσκευή, απευθείας μέσω της πλατφόρμας του Unity. Για την εγκατάσταση του πακέτου απαραίτητη είναι η επιλογή μίας επιπλέον ρύθμισης ώστε να εμφανίζεται στα διαθέσιμα πακέτα. Η ρύθμιση μπορεί να ενεργοποιηθεί μέσω του Edit→ Project Settings→ Package Manager→ Enable Preview Packages. Ο λόγος του επιπλέον αυτού βήματος είναι διότι το πακέτο βρίσκεται σε πρώιμο στάδιο ανάπτυξης σε αυτήν την χρονική περίοδο. Μετά το παραπάνω βήμα το πακέτο θα είναι διαθέσιμο μέσω του Package Manager, ενός παραθύρου μέσω του οποίο μπορούμε να προσθέσουμε επιπλέον πακέτα στο project μας.



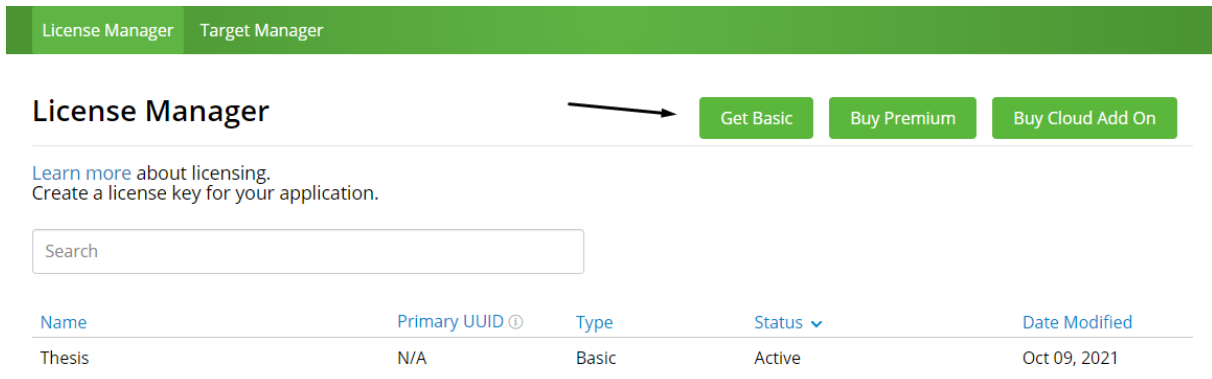
Σχήμα 3.5: Εγκατάσταση Device Simulator μέσω του παραθύρου Package Manager

3.4 Vuforia

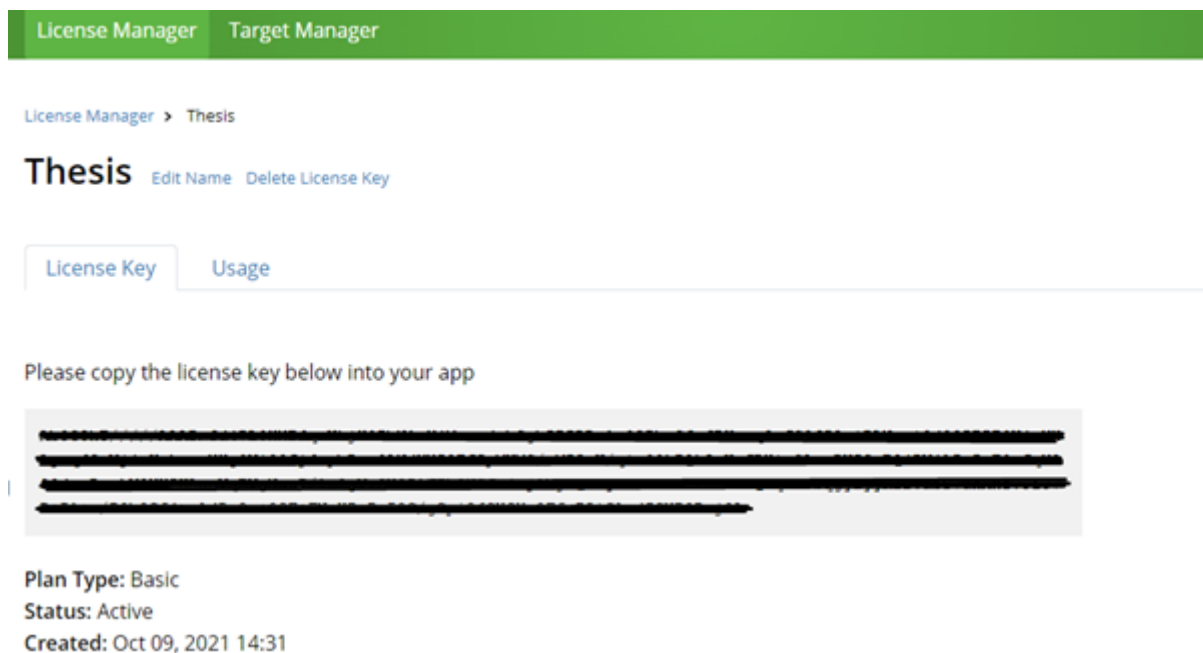
Το Vuforia αποτελεί ένα εργαλείο μέσω του οποίου γίνεται διαθέσιμη η χρήση της επαυξημένης πραγματικότητας. Πρόκειται για ένα από τα πιο διαδεδομένα εργαλεία για την ανάπτυξη AR εφαρμογών, με υποστήριξη σε κινητές συσκευές. Η βασική λειτουργία αυτού του λογισμικού είναι η ικανότητα του να αναγνωρίζει και να ακολουθεί αντικείμενα τα οποία καταγράφηκαν μέσω της κάμερας κάποιας συσκευής. Αυτή η ικανότητα επιτρέπει σε προγραμματιστές να τοποθετούν και να προσανατολίζουν εικονικά αντικείμενα σε σχέση με τον πραγματικό κόσμο προβάλλοντας τα μέσω μίας κάμερας.

3.4.1 Εγκατάσταση και Ρύθμιση Πακέτου Vuforia

Η εγκατάσταση του Vuforia γίνεται εύκολα μέσω της ιστοσελίδας του πακέτου [12], συγκεκριμένα η ιστοσελίδα παρέχει επιλογή κατεβάσματος του πακέτου σε μορφή συμβατή με την πλατφόρμα του Unity ανάμεσα στις επιλογές της. Αφού κατέβει το αρχείο, με μία απλή εκτέλεση προστίθενται το πακέτο στο project. Μετά την εγκατάσταση του πακέτου, είναι απαραίτητη η απόκτηση ενός κλειδιού άδειας χρήσης. Αυτό γίνεται επίσης μέσω της ιστοσελίδας του Vuforia, και συγκεκριμένα στην καρτέλα Develop → License Manager. Εκεί, μέσω της επιλογής του τύπου άδειας που επιθυμείτε, εκδίδεται και η αντίστοιχη άδεια χρήσης. Στην παρούσα Π.Ε. χρησιμοποιήθηκε μια βασική άδεια όπως φαίνεται και στο Σχήμα 3.6. Μετά την έκδοση της, κάνοντας κλικ πάνω στο όνομα του κλειδιού μπορούμε να δούμε το κλειδί καθώς και πληροφορίες για αυτό όπως φαίνεται και στο Σχήμα 3.7.



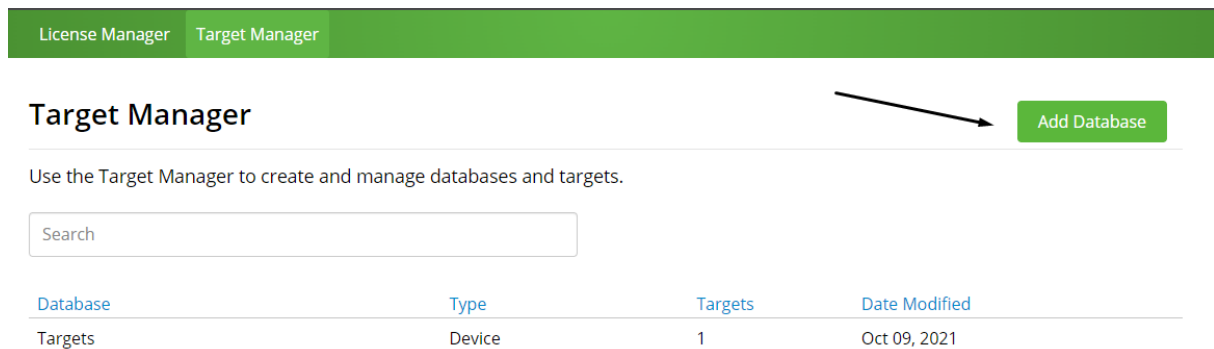
Σχήμα 3.6: Έκδοση κλειδιού άδειας χρήσης του Vuforia Engine



Σχήμα 3.7: Πληροφορίες κλειδιού άδειας χρήσης Vuforia Engine

3.4.2 Image Target

Εξίσου σημαντικό για την λειτουργία μίας AR κάμερας είναι να οριστεί το ποιός και τι είδους θα είναι ο στόχος-αντικείμενο το οποίο θα ακολουθεί και θα καταγράφει. Αρχικά, θα πρέπει να δημιουργηθεί μία βάση δεδομένων η οποία θα περιέχει αυτά τα αντικείμενα. Αυτό μπορεί εύκολα να πραγματοποιηθεί μέσω του site του Vuforia [12]. Εκεί στην καρτέλα Develop→ Database Manager δίνεται η δυνατότητα να προστεθούν βάσεις, όπως απεικονίζεται και στο Σχήμα 3.8.



Σχήμα 3.8: Δημιουργία βάσης δεδομένων αντικειμένων-στόχους

Αφού έχει δημιουργηθεί η βάση, δίνεται η επιλογή της πρόσθεση των αντικειμένων-στόχους. Οι στόχοι αυτοί μπορεί να έχουν την μορφή μίας απλής εικόνας, καθώς και κάποιου είδους τρισδιάστατου αντικειμένου.

Add Target

Type:



File:

Choose File

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Σχήμα 3.9: Διαθέσιμες επιλογές αντικειμένων-στόχους της Vuforia Engine

Στην συγκεκριμένη Π.Ε. χρησιμοποιήθηκε η μορφή μίας απλής εικόνας που απεικονίζει ένα QR Code το οποίο δημιουργήθηκε μέσω της ιστοσελίδας AR Maker [13].

Εφόσον έχει δημιουργηθεί η βάση των στόχων, διαθέσιμη είναι η επιλογή κατεβάσματος (download), καθώς και η επιλογή της πλατφόρμας στην οποία επιθυμούμε να την χρησιμοποιήσουμε. Στην παρούσα εργασία, η βάση πρέπει να είναι σε μορφή αναγνωρίσιμη από το Unity, οπότε επιλέχθηκε η αντίστοιχη μορφή.

Download Database

1 of 1 active targets will be downloaded

Name:
Targets

Select a development platform:

Android Studio, Xcode or Visual Studio

Unity Editor ←



Σχήμα 3.10: Κατέβασμα βάσης σε μορφή πακέτου Unity

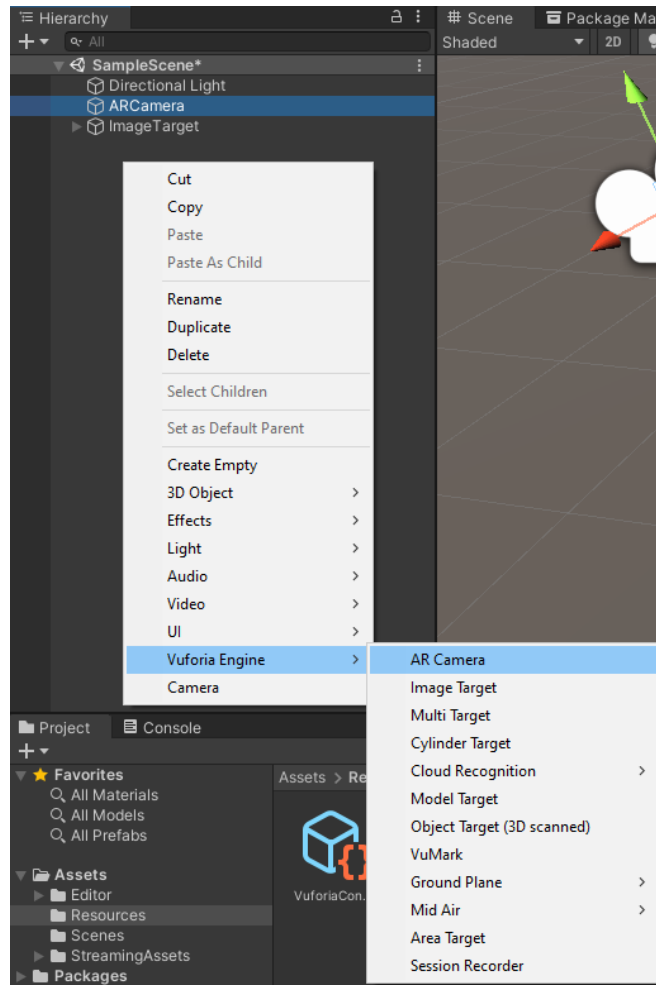
3.4.3 Vuforia AR Camera & Image Target

Όπως προαναφέραμε, βασική λειτουργία του πακέτου Vuforia πραγματοποιείται μέσω της χρήση μίας κάμερας. Προφανώς η βασική κάμερα του Unity, που προστίθενται αυτόματα στην σκηνή μας, δεν έχει τέτοιες δυνατότητες. Ολοκληρώνοντας την εγκατάσταση του πακέτου, έχουμε την δυνατότητα να προσθέσουμε μία κάμερα με AR δυνατότητες. Όπως φαίνεται και στο Σχήμα 3.11 κάνοντας δεξιά κλικ στην ιεραρχία της σκηνής μας, μας εμφανίζονται όλες οι δυνατότητες που προσφέρει το πακέτο Vuforia. Μέσα σε αυτές βρίσκονται τα δύο πολύ σημαντικά στοιχεία μας, η AR Camera και το Image Target.

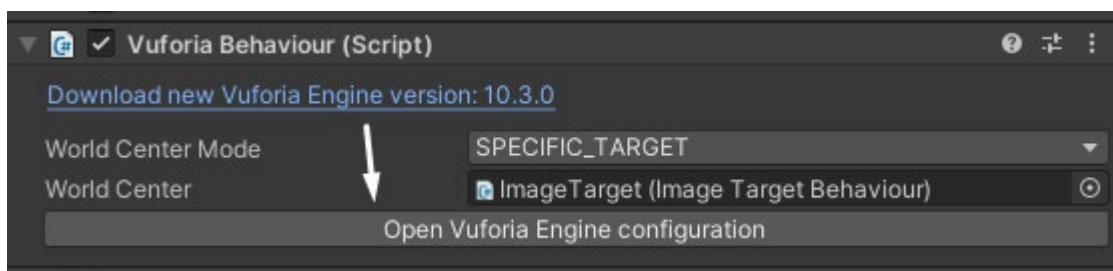
3.4.3.1 AR Camera

Κατά την δημιουργία της AR κάμερας, παρατηρούμε ένα πλήθος επιλογών που μπορούν να αλλάξουν ανάλογα με το επιθυμητό αποτέλεσμα. Οι πιο βασικές επιλογές αφορούν την συμπεριφορά της κάμερας, κάτι το οποίο ρυθμίζεται μέσα απο το script Vuforia Behaviour, το οποίο προστίθενται αυτόματα κατά την δημιουργία της. Πρώτο βήμα για την λειτουργία της κάμερας είναι η χρήση του κλειδιού άδειας χρήσης, το οποίο απεικονίζεται στο Σχήμα 3.7. Πατώντας το διαθέσιμο κουμπί του script το οποίο φαίνεται και στο Σχήμα 3.12, έχουμε πρόσβαση στις βασικές ρυθμίσεις. Εκεί το πρώτο πεδίο το οποίο είναι διαθέσιμο είναι το αντίστοιχο πεδίο στο οποίο θα προσθέσουμε το κλειδί, όπως απεικονίζεται και στο σχήμα 3.13.

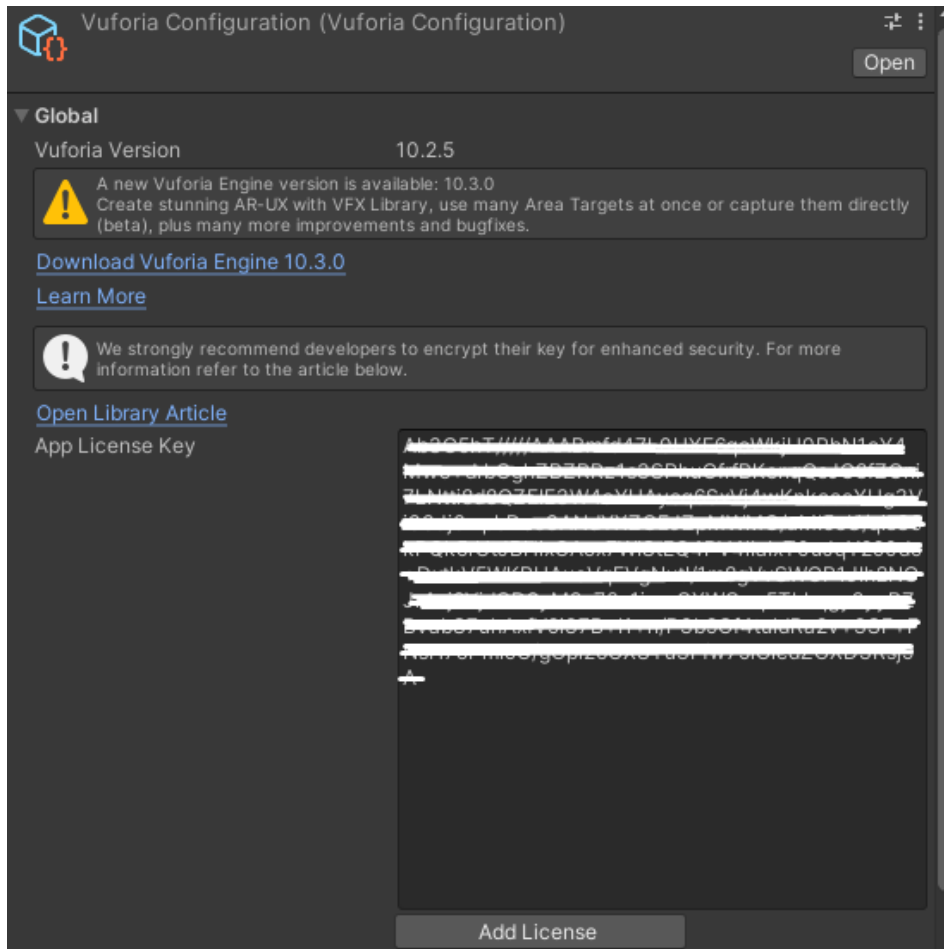
Κεφάλαιο 3ο:



Σχήμα 3.11: Πρόσθεση Vuforia Engine αντικειμένων σε Unity project

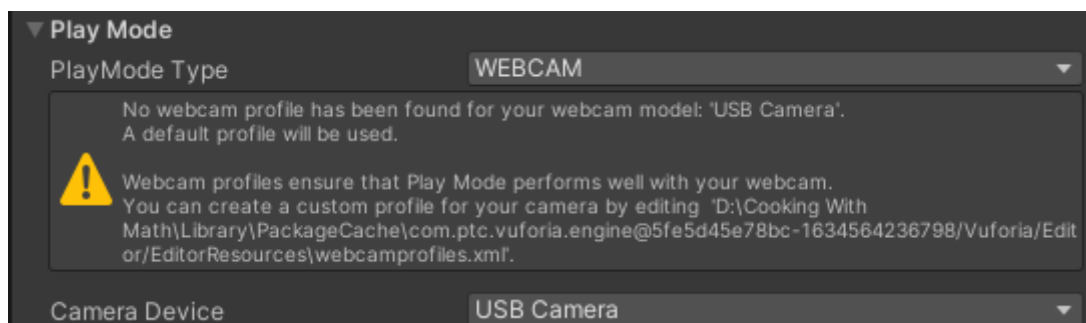


Σχήμα 3.12: Άνοιγμα επιπλέον ρυθμίσεων συμπεριφοράς AR κάμερας



Σχήμα 3.13: Πρόσθεση κλειδιού άδειας χρήσης σε Vuforia AR camera

Επιπλέον με την χρήση του Device Simulator, ακόμη μία επιλογή που θα πρέπει να αλλαχθεί αφορά τις ρυθμίσεις για το Play Mode οι οποίες βρίσκονται στην ίδια καρτέλα και με τις παραπάνω. Συγκεκριμένα θα πρέπει να επιλέξουμε τον τύπο του Play Mode να είναι Webcam και η συσκευή να φαίνεται στην επιλογή Camera Device (Σχήμα 3.14).

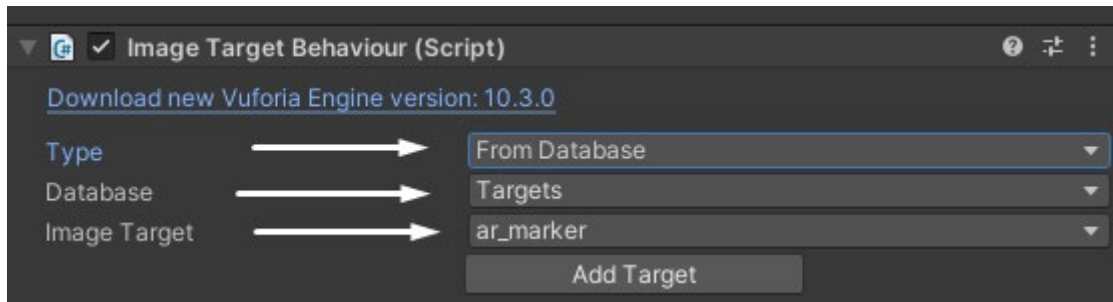


Σχήμα 3.14: Ορισμός webcam ως AR κάμερας

3.4.3.2 Image Target

Το δεύτερο βασικό στοιχείο που θα πρέπει να οριστεί στο project είναι το αντικείμενο-στόχος, κάτι το οποίο μπορεί να προστεθεί όπως προστέθηκε και η AR κάμερα (Σχήμα 3.11). Σε αυτό το αντικείμενο, οι βασικές ρυθμίσεις συμπεριφοράς του ρυθμίζονται μέσω του Image Target Behavior script το οποίο

προστίθενται αυτόματα. Εκεί ορίζεται ο τύπος της βάσης των στόχων, επιλέγεται η βάση που προστέθηκε καθώς και το αντικείμενο μέσα σε αυτήν.

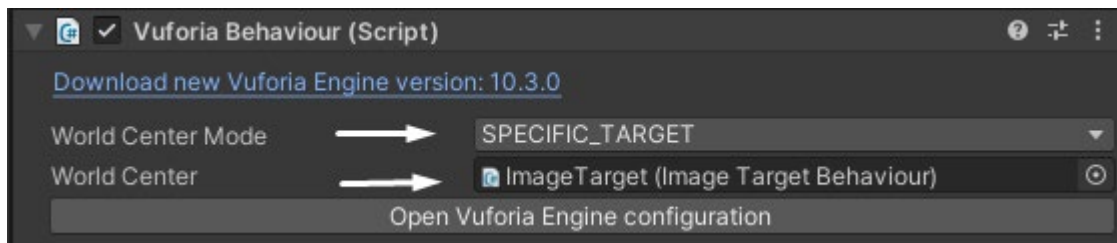


Σχήμα 3.15: Ορισμός στόχου σε αντικείμενο Image Target

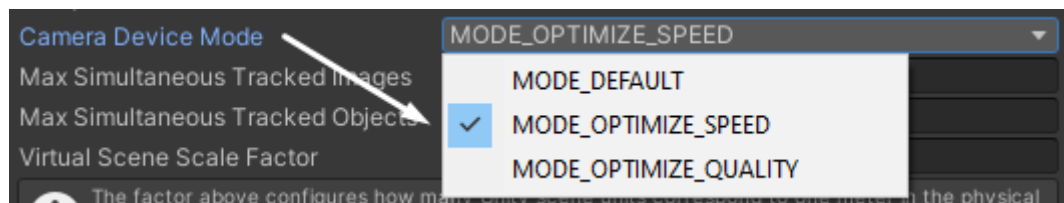
3.4.4 Επιπλέον Ρυθμίσεις

3.4.4.1 AR Camera

Μερικές επιπλέον ρυθμίσεις που χρησιμοποιήθηκαν αφορούν την AR κάμερα. Συγκεκριμένα στην συμπεριφορά της κάμερας επιλέχθηκε το κεντρικό σημείο αναφοράς της να είναι ένας συγκεκριμένος στόχος, το image target (Σχήμα 3.16). Άλλες διαθέσιμες επιλογές είναι το κεντρικό σημείο αναφοράς να είναι το πρώτο αντικείμενο το οποίο θα αναγνωρίσει ή η ίδια η συσκευή που χρησιμοποιείται. Επίσης, για την καλύτερη και πιο γρήγορη λειτουργία του παιχνιδιού επιλέχθηκε στις ρυθμίσεις της κάμερας, να λειτουργεί με βάση την ταχύτητα. Με αυτόν τον τρόπο η κάμερα έχει πιο γρήγορη απόκριση στην αναγνώριση της εικόνας. Άλλες επιλογές αφορούν την προεπιλεγμένη κατάσταση της ή την έμφαση σε ποιότητα (Σχήμα 3.17).



Σχήμα 3.16: Ορισμός κεντρικού σημείου αναφοράς AR κάμερας

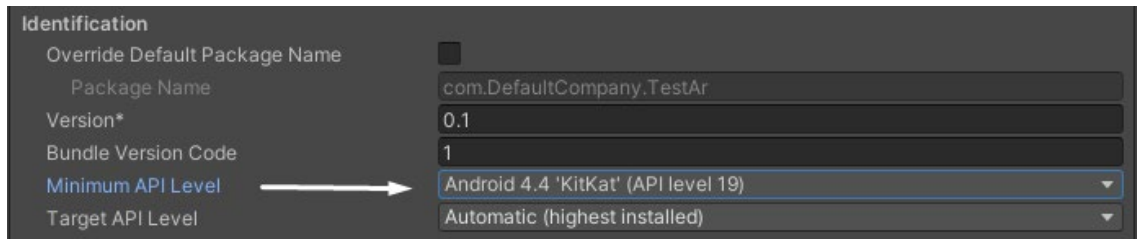


Σχήμα 3.17: Επιλογή πεδίου έμφασης της AR κάμερας στην ταχύτητα

3.4.4.2 Android

Επίσης, μία πολύ σημαντική ρύθμιση αφορά την ελάχιστη επιτρεπτή έκδοση που θα μπορεί να τρέξει το παιχνίδι. Συγκεκριμένα, η εκδοχή αυτή θα πρέπει να συμβαδίζει με την ελάχιστη εκδοχή που υποστηρίζει και το Vuforia Engine. Αυτό μπορεί να οριστεί μέσω του Edit→ Project Settings→ Player. Στην κατηγορία Identification βρίσκονται τα αντίστοιχα πεδία που αφορούν την ελάχιστη

καθώς και την μέγιστη δυνατή έκδοση που επιτρέπει το παιχνίδι (Σχήμα 3.18). Την τρέχων χρονική στιγμή η ελάχιστη έκδοση που υποστηρίζεται από το Vuforia Engine είναι η Android 4.4 'KitKat' (API level 19).



Σχήμα 3.18: Ελάχιστη επιτρεπτή έκδοση Android του παιχνιδιού

3.5 Επίλογος

Με βάση των παραπάνω βημάτων, η δημιουργία του project έχει ολοκληρωθεί και είναι έτοιμο να λειτουργήσει. Όλες οι απαραίτητες ρυθμίσεις σχετικά με το πακέτο Vuforia έχουν οριστεί, οπότε δυνατή είναι πλέον η ανάπτυξη του περιεχομένου του project.

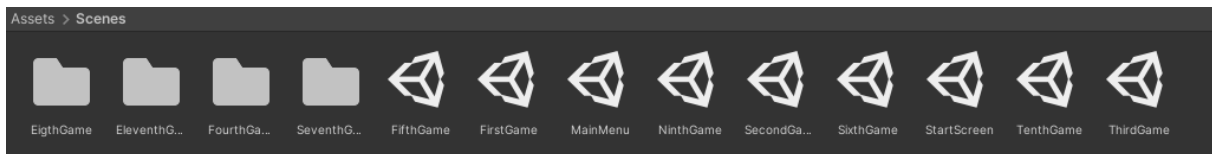
Κεφάλαιο 4ο: Αρχική Οθόνη και Μενού Εφαρμογής

4.1 Εισαγωγή

Το πρώτο και πιο βασικό μέρος μιας εφαρμογής με το οποίο έρχεται σε επαφή ο χρήστης, είναι προφανώς η αρχική οθόνη και το μενού. Απαραίτητα χαρακτηριστικά σε αυτά τα στοιχεία είναι η απλότητα και η ευχρηστία, καθώς θα πρέπει να ωθεί τον χρήστη στο βασικό κομμάτι της εφαρμογής και όχι να τον απωθεί ή να τον μπερδεύει. Σε αυτό το κεφάλαιο θα αναλυθούν τα δύο αυτά κομμάτια του παιχνιδιού σε επιμέρους στοιχεία καθώς και η λογική τους.

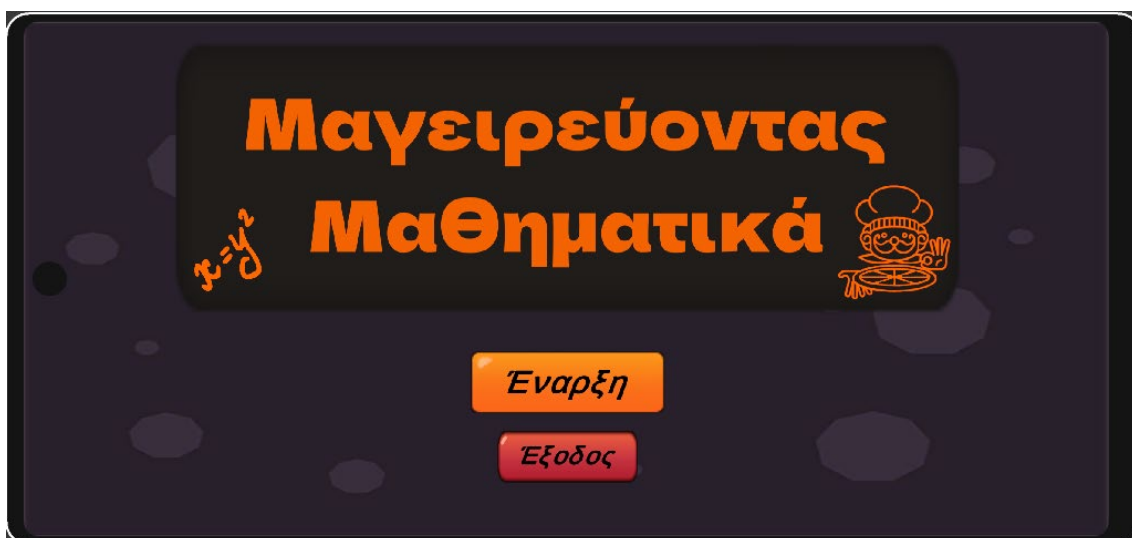
4.2 Χρήση Σκηνών

Προτού μιλήσουμε για τις επιμέρους επιλογές που έχει ο χρήστης, πρέπει να μιλήσουμε για την οργάνωση του προγράμματος σε σκηνές. Το περιεχόμενο του παιχνιδιού, έχει χωριστεί και οργανωθεί σε σκηνές. Ο τρόπος δημιουργίας νέων σκηνών γίνεται απλά μέσω της επιλογής File→ New Scene. Αφού έχουμε δημιουργήσει κάποια σκηνή μπορούμε να την σώσουμε ως αντικείμενο τύπου Scene, στον φάκελο Assets που δημιουργείται αυτόματα από το Unity. Στο συγκεκριμένο project υπάρχουν συνολικά 17 σκηνές, όπου κάθε μία περιγράφει και μία συνταγή, ένα μέρος της ή ένα μενού. Κάθε σκηνή έχει αποθηκευτεί στον φάκελο Asset→ Scenes και μπορεί να προστεθεί στην ροή του παιχνιδιού μέσω της επιλογής File→ Build Settings→ Add Open Scenes. Κατά την πρόσθεση των σκηνών ορίζεται σε κάθε μία ένας αριθμός κλειδί (id) (Σχήμα 4.1), ο οποίος μπορεί να χρησιμοποιηθεί για την προγραμματιστική διαχείριση τους.



Σχήμα 4.1: Φάκελος αποθήκευσης σκηνών παρούσας Π.Ε.

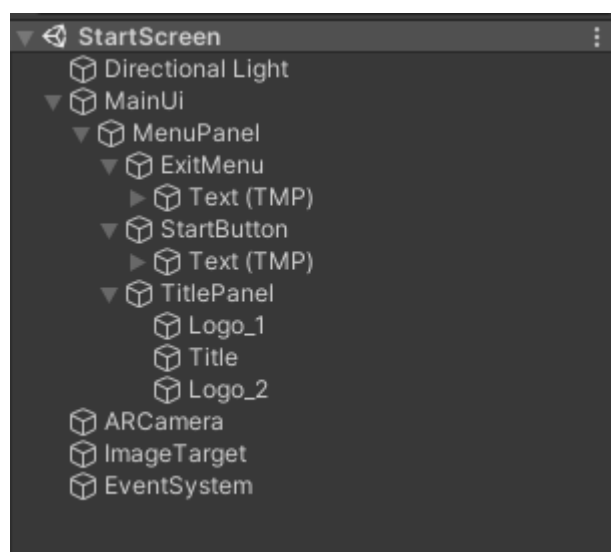
4.3 Αρχική Οθόνη



Σχήμα 4.2: Αρχική οθόνη παιχνιδιού της Π.Ε.

4.3.1 Ιεραρχία & Δομή Σκηνής

Τα στοιχεία τα οποία αποτελούν την παρούσα σκηνή αναδύκνούνται στο σχήμα 4.3, από τα οποία βασικό στοιχείο της σκηνής αποτελεί η διεπαφή χρήστη. Το στοιχείο MainUi πρόκειται για έναν καμβά, ο οποίος είναι απαραίτητος για την λειτουργία μίας διεπαφής. Η πρόσθεση ενός καμβά γίνεται εύκολα μέσω ενός δεξί κλικ και έπειτα τις επιλογές UI→Canvas. Ο καμβάς καταλαμβάνει το σύνολο της οθόνης και μέσα σε αυτόν βρίσκεται ένα πάνελ το οποίο περιέχει τα στοιχεία: τίτλος, κουμπί έναρξης και κουμπί εξόδου. Επίσης, σημαντικό για την λειτουργία μίας διεπαφής χρήστη είναι το στοιχείο EventSystem. Μέσω αυτού καταγράφονται και διαχειρίζονται όλες οι βασικές λειτουργίες που αφορούν την διεπαφή, όπως το ποιά κουμπίά έχουν πατηθεί. Μέσα στο πανελ MenuPanel υπάρχουν τα επιμέρους στοιχεία Title Panel, ExitMenu και StartButton. Το πρώτο στοιχείο πρόκειται για ένα πάνελ, μέσα στο οποίο βρίσκεται ένα στοιχείο κειμένου. Τα στοιχεία ExitMenu και StartButton αποτελούν κουμπίά τα οποία πραγματοποιούν την είσοδο και έξοδο της εφαρμογής



Σχήμα 4.3: Ιεραρχία σκηνής Start Screen

4.3.2 Εικόνες Γραφικά και Γραμματοσειρές

Για την καλύτερη εμφάνιση της εφαρμογής προστέθηκαν γραφικά και επιπλέον γραμματοσειρές στην διεπαφή. Συγκεκριμένα το πάνελ TitlePanel που αφορά τον τίτλο της εφαρμογής περιέχει τρία στοιχεία, δύο από τα οποία είναι εικόνες που περιγράφουν παραστατικά τον τίτλο του παιχνιδιού καθώς και το περιεχόμενό του. Για τον τίτλο του παιχνιδιού χρησιμοποιήθηκε η γραμματοσειρά “DelaGothicOne-Regular” η οποία προστέθηκε στο project μέσω της ιστοσελίδας Google Fonts [14]. Ο λόγος χρήσης πρόσθετης γραμματοσειράς, πέρα της αισθητικής, είναι λόγω της έλλειψης υποστήριξης των γραμματοσειρών του Unity σε ελληνικό αλφάβητο.

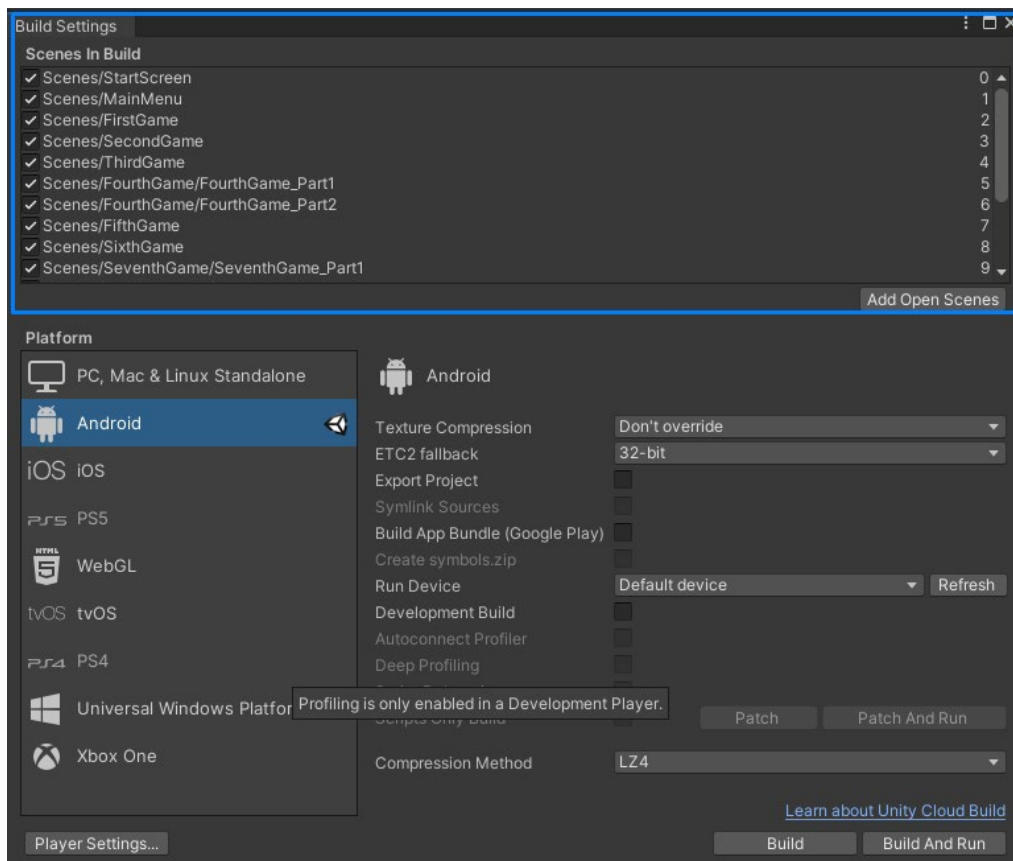
4.3.3 Λογική και Λειτουργία

Η βασική λογική και λειτουργία της αρχικής οθόνης διεκπεραιώνεται μέσω ενός script με όνομα MenuHandler. Το script αυτό προστέθηκε στον καμβά MainUi και εμπεριέχει τις μεθόδους openScene και exitGame (Σχήμα 4.6).

- Η πρώτη μέθοδος δέχεται έναν αριθμό τύπου intiger και χρησιμοποιείται για το άνοιγμα νέας σκηνής με βάση τον αριθμό ο οποίος αντιστοιχεί στο κλειδί αναγνώρισης (id) της. Τα id των σκηνών του παιχνιδιού μπορούν να βρεθούν μέσω του παραθύρου File→ Build Settings, εκεί

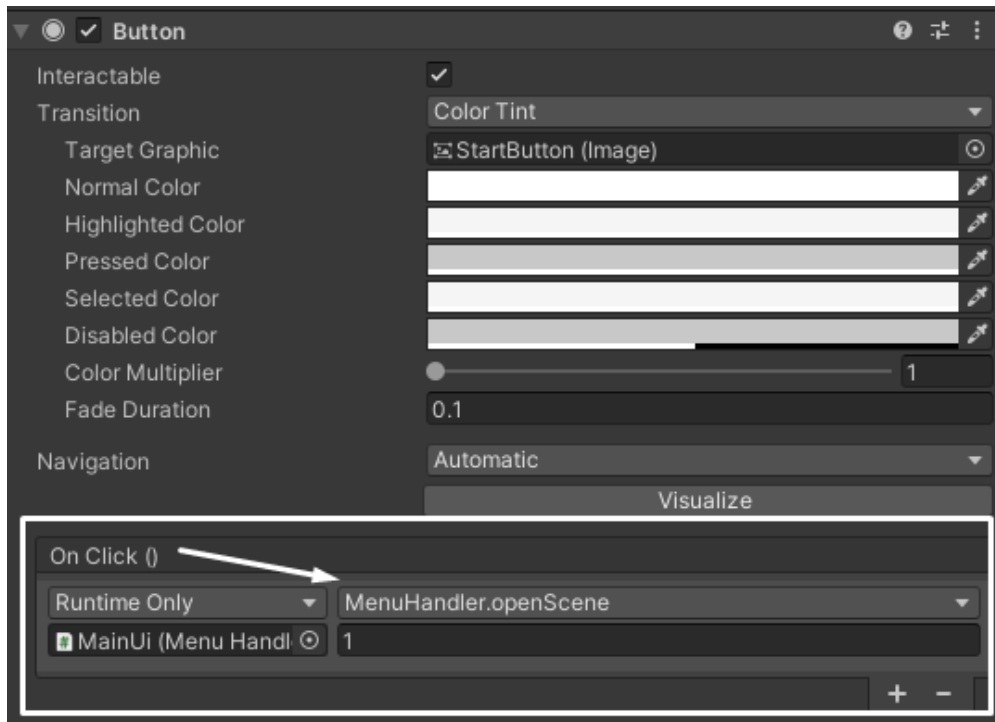
έχουμε την δυνατότητα επίσης να προσθέτουμε νέες σκηνές καθώς και να αλλάζουμε την σειρά προσπέλασης τους (Σχήμα 4.4). Ο τρόπος με τον οποίο επιτυγχάνεται ο προγραμματιστικός έλεγχος των σκηνών είναι μέσω της κλάσης SceneManager, όπου απαραίτητο για την χρήση της είναι η χρήση του API SceneManagement, το οποίο ανήκει στο UnityEngine. Η μέθοδος έχει οριστεί να καλείται όταν ο χρήστης πιάσει το κουμπί 'Έναρξη' με αριθμό σκηνής ίσο με 1, το οποίο αντιστοιχεί στο μενού. Ο ορισμός γίνεται εύκολα μέσω του πεδίου OnClick() το οποίο προϋπάρχει στα στοιχεία κουμπιά (Σχήμα 4.5).

- Η δεύτερη μέθοδος αφορά το κλείσιμο της εφαρμογής και χρησιμοποιεί την κλάση Application, μέσω της οποίας μπορούμε να διαχειριστούμε βασικές λειτουργίες και δεδομένα της εφαρμογής.



Σχήμα 4.4: Σκηνές παιχνιδιού Π.Ε. στα Build Settings

Κεφάλαιο 4ο:



Σχήμα 4.5: Ορισμός μεθόδου openScen(1) σε OnClick() ενέργεια.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

Unity Script (17 asset references) | 0 references
public class MenuHandler : MonoBehaviour
{
    0 references
    public void openScene(int sceneIndex)
    {
        SceneManager.LoadScene(sceneIndex);
        BowlHandler.itemsUsed = new List<Items>(); //reset the list so it doesn't hold any used items
    }

    0 references
    public void exitGame()
    {
        Application.Quit();
    }
}
```

Σχήμα 4.6: Κώδικας MenuHandler script

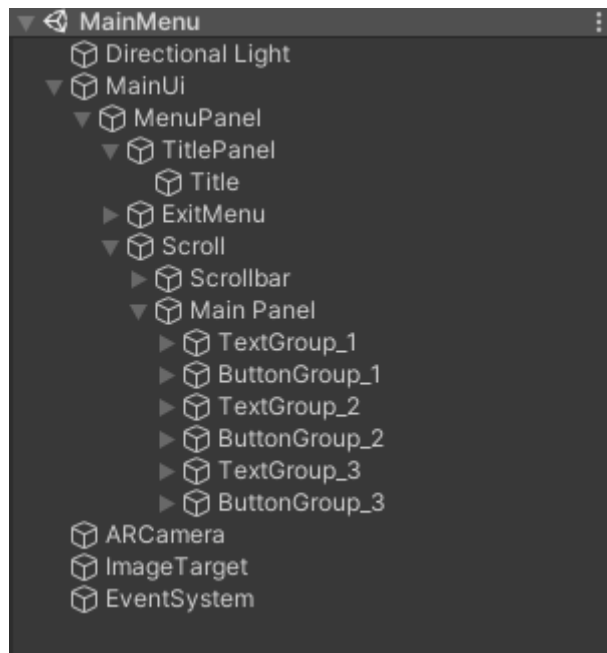
4.4 Μενού



Σχήμα 4.7: Βασικό μενού παιχνιδιού Π.Ε.

4.4.1 Ιεραρχία & Δομή Σκηνής

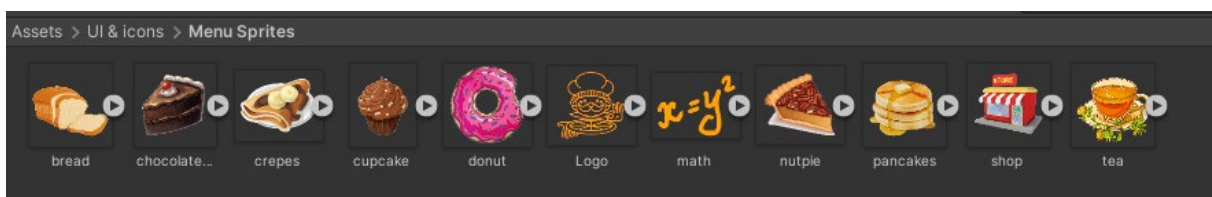
Η ιεραρχία των στοιχείων που αποτελούν την σκηνή του μενού διαφέρουν από την σκηνή της αρχικής οθόνης στο περιεχόμενο του πανελ MenuPanel. Τα στοιχεία TitlePanel καθώς και ExitMenu λειτουργούν όπως και στην σκηνή της αρχικής οθόνης. Το επιπρόσθετο στοιχείο Scroll πρόκειται για ένα αντικείμενο στο οποίο του ανατέθηκε η ιδιότητα Scroll Rect. Με βάση αυτήν την ιδιότητα, ο χρήστης μπορεί να σέρνει την οθόνη του ώστε να έχει πρόσβαση στο υπόλοιπο περιεχόμενο το οποίο δεν εμφανίζεται στην συσκευή του, λόγω του μεγέθους της. Μέσα σε αυτό το αντικείμενο υπάρχει ένα αντικείμενο τύπου UI, ονόματι Scrollbar, το οποίο εκτελεί την λογική της ιδιότητας Scroll Rect, καθώς και άλλο ένα αντικείμενο Panel ονόματι Main Panel, στο οποίο βρίσκεται το βασικό περιεχόμενο της σκηνής. Το αντικείμενο Main Panel έχει διάσταση μεγαλύτερη από αυτή της οθόνης, καθώς και την ιδιότητα Vertical Layout Group, η οποία αφορά την κάθετη στοίχιση των στοιχείων του περιεχομένου του. Το περιεχόμενο του panel χωρίζεται σε δυάδες των δύο ομάδων, η μία αφορά τα ονόματα των συνταγών και η δεύτερη τα κουμπιά που οδηγούν στις αντίστοιχες συνταγές. Και στις δύο ομάδες έχει προστεθεί η ιδιότητα Horizontal Layout Group, ώστε να υπάρχει οριζόντια στοίχιση μεταξύ των στοιχείων.



Σχήμα 4.8: Ιεραρχία βασικού μενού παιχνιδιού Π.Ε.

4.4.2 Εικόνες Γραφικά και Γραμματοσειρές

Όπως και στην παραπάνω σκηνή έτσι και στην παρούσα, προστέθηκαν εικόνες και γραμματοσειρές για την καλύτερη αισθητική του παιχνιδιού. Συγκεκριμένα χρησιμοποιήθηκαν πολλαπλές εικόνες με στόχο την όμορφη και ελκυστική αναπαράσταση των συνταγών σε παιδιά Στ' δημοτικού (Σχήμα 4.9). Οι γραμματοσειρές που χρησιμοποιήθηκαν είναι η “Oi-Regural”, για τον τίτλο της σκηνής, και η “DelaGothicOne-Regular” για τους επιμέρους τίτλους των συνταγών. Οι δύο αυτές γραμματοσειρές βρέθηκαν και προστέθηκαν μέσω της ιστοσελίδας Google Fonts [14].

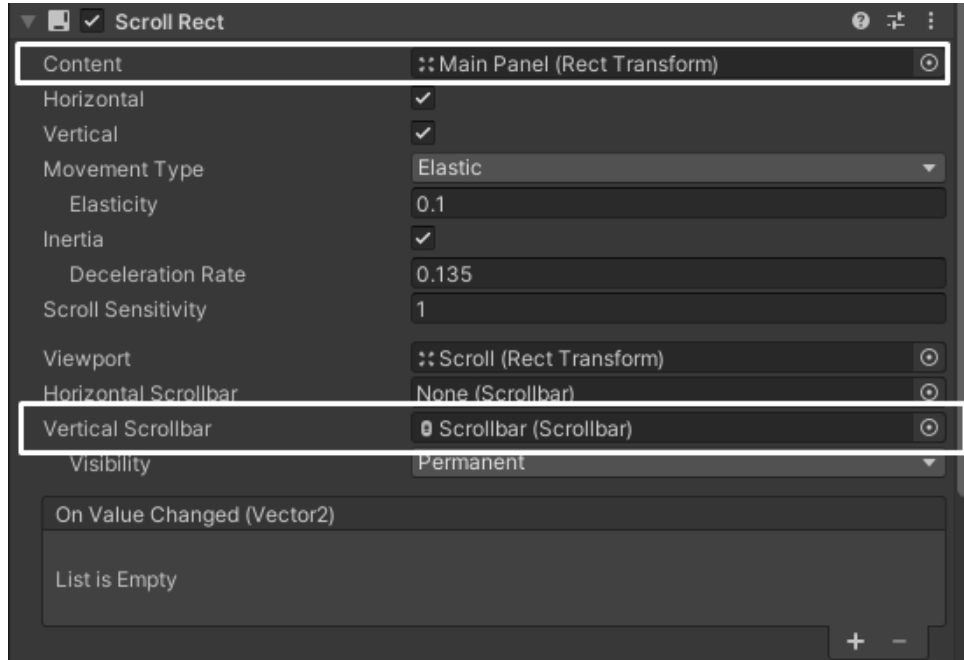


Σχήμα 4.9: Εικόνες μενού και αρχικής οθόνης παιχνιδιού

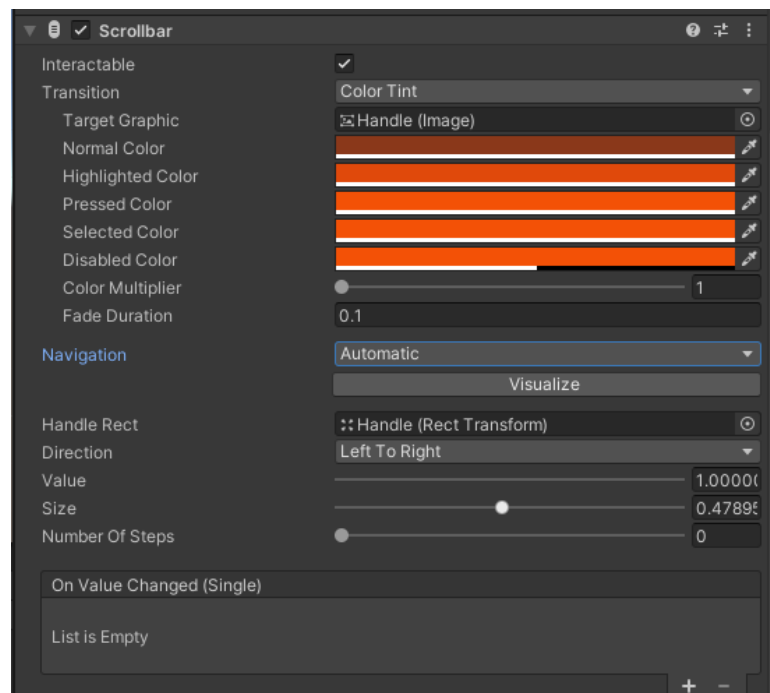
4.4.3 Λογική και Λειτουργία

Η λογική και λειτουργία της σκηνής είναι αρκετά απλή και διεκπεραιώνεται μέσω του script MenuHandler το οποίο έχει ανατεθεί στο αντικείμενο MenuPanel. Κάθε κουμπί χρησιμοποιεί την μέθοδο openScene και το id της κάθε σκηνής για να φέρει τον χρήστη στην αντίστοιχη συνταγή. Το κουμπί ExitMenu σε αυτήν την σκηνή, σε αντίθεση με την προηγούμενη, οδηγεί στην αρχική οθόνη αντί για το κλείσιμο της εφαρμογής, μέσω της ίδιας μεθόδου. Το περιεχόμενο σε αυτήν την περίπτωση είναι το MainPanel, του οποίου οι διαστάσεις είναι αρκετά μεγαλύτερες απο την οθόνη μιας κινητής συσκευής, κάτι το οποίο καθιστά απαραίτητη την λειτουργία ενός τρόπου με τον οποίο ο χρήστης θα έχει πρόσβαση σε όλο το περιεχόμενο, άσχετα από το μέγεθος της οθόνης του. Γι' αυτόν τον λόγο προστέθηκε στην διεπαφή μια λειτουργία scroll. Η λειτουργία scroll λειτουργεί με βάση δύο

στοιχείων, την ιδιότητα Scroll Rect καθώς και το UI αντικείμενο Scrollbar. Στην ιδιότητα ορίζεται το ποιά είναι το περιεχόμενο στο οποίο θέλουμε να έχουμε scroll, καθώς και το αν το scrollbar θα είναι οριζόντιο ή κάθετο (Σχήμα 4.10). Έπειτα, στο αντικείμενο Scrollbar μπορούμε να ορίσουμε τους χρωματισμούς, το εύρος τιμών με το οποίο θα λειτουργεί, καθώς και ότι άλλη ιδιότητα το περιγράφει (Σχήμα 4.11).



Σχήμα 4.10: Ιδιότητα Scroll Rect αντικειμένου Scroll



Σχήμα 4.11: Ιδιότητες αντικειμένου Scrollbar

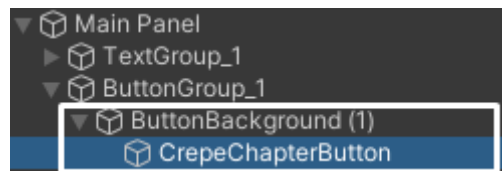
4.5 Εμφάνιση Διεπαφής

Ένα βασικό και εμφανές στοιχείο, κοινό για τις δύο σκηνές είναι η χρήση εικόνων για καλύτερη αναπαράσταση και περιγραφή περιεχομένου. Οι τρόποι με τους οποίους χρησιμοποιήθηκαν αυτές οι εικόνες είναι δύο:

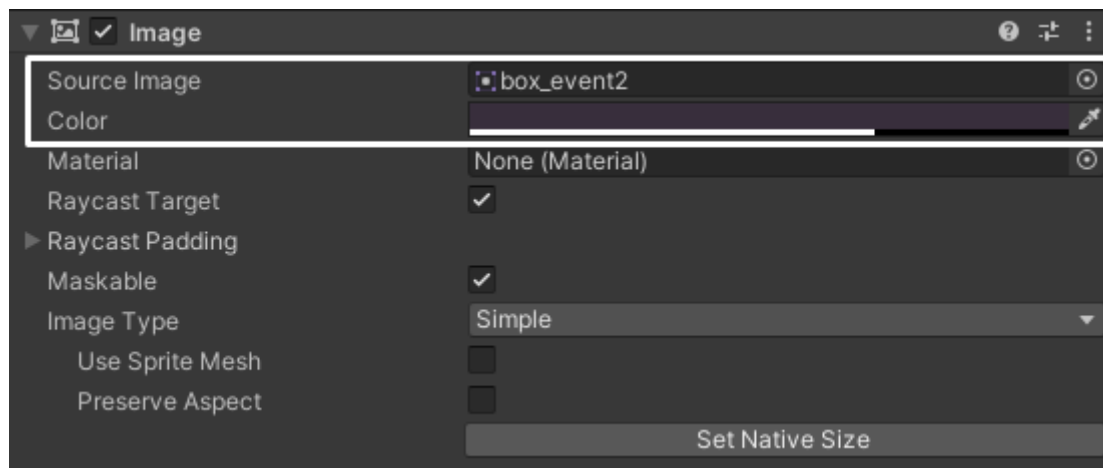
- Ορισμός εικόνας σε πεδίο ιδιότητας τύπου Image
- Χειροκίνητη πρόσθεση image αντικειμένου

4.5.1 Ορισμός εικόνας σε πεδίο ιδιότητας τύπου Image

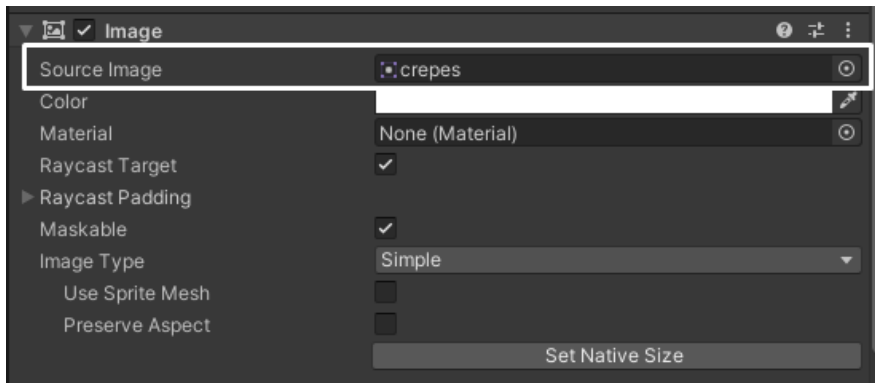
Όλα τα αντικείμενα τύπου UI φέρουν την ιδιότητα Image, μέσα σε οποία μπορούμε να ορίσουμε κάποια εικόνα, έναν επιπρόσθετο χρωματισμό και άλλες επιλογές οι οποίες εμφανίζονται και στο σχήμα 4.13. Χαρακτηριστικό παράδειγμα, είναι τα κουμπιά εικόνες που υπάρχουν στην σκηνή του μενού. Εκεί έχουμε δημιουργήσει ένα πάνελ για κάθε κουμπί το οποίο φέρει σκούρο χρωματισμό και μία εικόνα πλαισίου (Σχήμα 4.13), μέσα σε οποίο βρίσκεται και το εκάστοτε κουμπί-εικόνα (Σχήμα 4.12). Η επιλογή αυτή έγινε για την πιο καθαρή απεικόνιση των εικόνων που αντιστοιχούν στις συνταγές (Σχήμα 4.15).



Σχήμα 4.12: Ιεραρχία κουμπιού στην σκηνή μενού



Σχήμα 4.13: Ιδιότητα Image με χρήση επιπρόσθετου χρωματισμού σε Panel



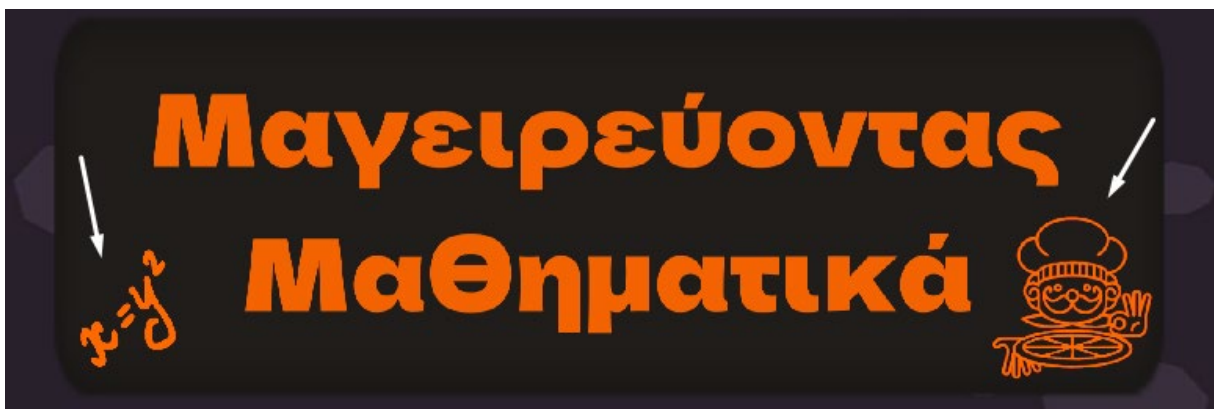
Σχήμα 4.14: Ιδιότητα Image σε αντικείμενο button



Σχήμα 4.15: Τελική απεικόνιση κουμπιού

4.5.2 Χειροκίνητη πρόσθεση image αντικειμένου

Ο δεύτερος και εξίσου απλός τρόπος πρόσθεσης εικόνας, γίνεται απευθείας μέσω του παραθύρου της ιεραρχίας του Unity. Κάνοντας δεξί κλικ και επιλέγοντας UI→ Image, μπορούμε να προσθέσουμε γρήγορα αντικείμενα-εικόνες στην διεπαφή μας. Ο συγκεκριμένος τρόπος χρησιμοποιήθηκε για την πρόσθεση δύο εικόνων στην σκηνή αρχικής οθόνης, για την εικονική περιγραφή του τίτλου.



Σχήμα 4.16: Χρήση αντικειμένων Image για την πρόσθεση logo εικόνων στον τίτλο

Αφού έχει προστεθεί το αντικείμενο, τα βήματα ορισμού εικόνας καθώς και των χαρακτηριστικών της είναι ακριβώς ίδια με τον πρώτο τρόπο (Σχήμα 4.14).

4.6 Επίλογος

Ολοκληρώνοντας αυτό το κεφάλαιο, γίνεται αντιληπτός ο τρόπος λειτουργίας μίας διεπαφής, καθώς και μερικών από τις επιλογές που προσφέρει, όπως η χρήση εικόνων και scroll αντικειμένων, για την δημιουργία μίας εύχρηστης και ευχάριστης διεπαφής.

Κεφάλαιο 5ο: Βασικά Χαρακτηριστικά και Λειτουργίες Σκηνών

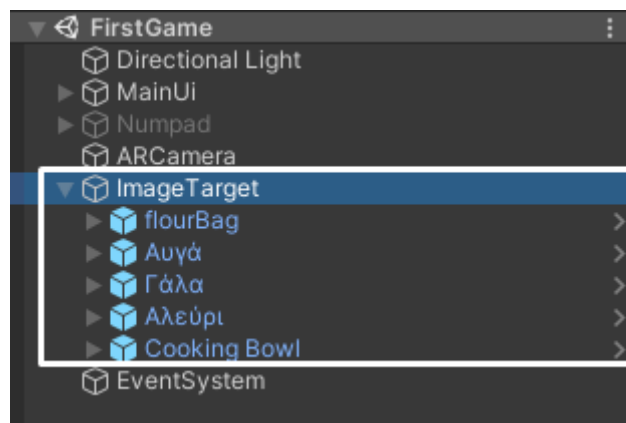
5.1 Εισαγωγή

Όλα τα σενάρια τα οποία αποτελούν το παιχνίδι της παρούσας Π.Ε., διαφέρουν λίγο ή πολύ μεταξύ τους. Όμως, ακόμα και σκηνές με εντελώς διαφορετικούς χειρισμούς έχουν κοινά χαρακτηριστικά και λειτουργίες. Σε αυτό το κεφάλαιο θα αναλύσουμε αυτά τα κοινά χαρακτηριστικά, όπως :

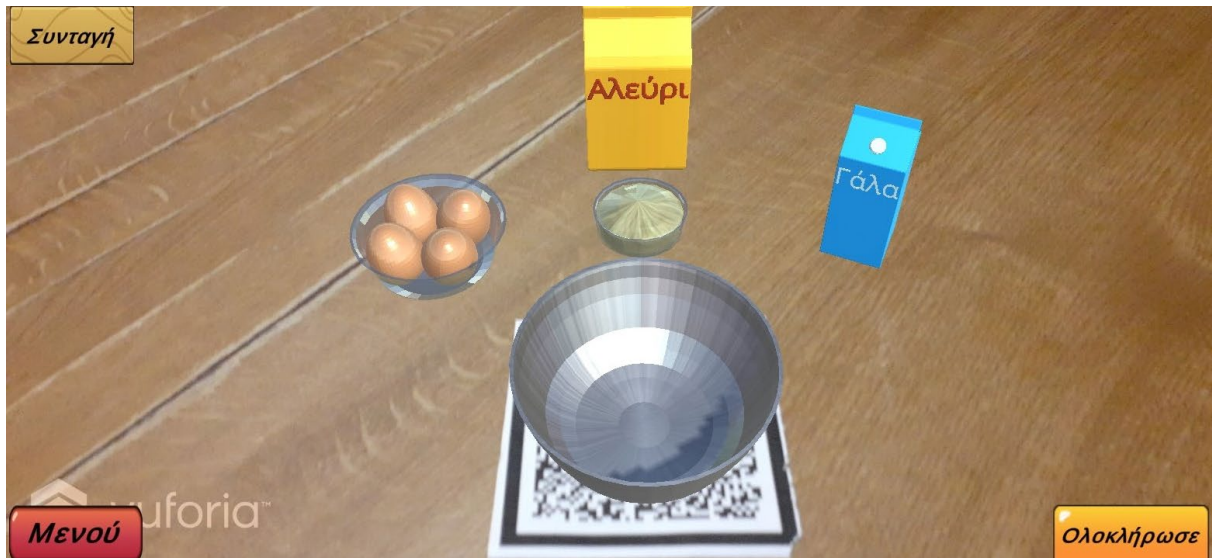
- Vuforia: Target Images
- Τρισδιάστατα Αντικείμενα
- Scripts & UI: Οι βασικές λειτουργίες κορμού πραγματοποιούνται μέσω κάποιων scripts καθώς και διεπαφών

5.2 Vuforia: Target Images

Όπως έχουμε προαναφέρει, η Π.Ε. χρησιμοποιεί τεχνολογίες επαυξημένης πραγματικότητας (AR) σε συνδυασμό με σενάρια μαγειρικής, έχοντας στόχο την εξάσκηση μαθητών Στ' Δημοτικού σε προβλήματα μαθηματικών, με δημιουργικό και διασκεδαστικό τρόπο. Για την υλοποίηση της επαυξημένης πραγματικότητας χρησιμοποιήθηκε το πακέτο Vuforia. Μέσω αυτού, ο χρήστης μπορεί να βλέπει και να αλληλεπιδρά με εικονικά αντικείμενα μέσω της συσκευής του. Αυτό συμβαίνει μέσω μίας AR κάμερας και ενός στόχου, όπου στην συγκεκριμένη εφαρμογή ο στόχος αφορά ένα QR κωδικό. Αφού έχουν δημιουργηθεί τα απαραίτητα στοιχεία στην πλατφόρμα του Unity, όπως αναλύσαμε και στο Κεφάλαιο 3, μπορούμε να αρχίσουμε να προσθέτουμε εικονικά αντικείμενα τα οποία θα φορτώνονται μέσω του Image Target. Συγκεκριμένα, τα αντικείμενα μας μπορούν να προστεθούν ως παιδιά του αντικειμένου Image Target (Σχήμα 5.1), το οποίο σε συνδυασμό με την AR camera και του QR code που ορίσαμε, θα φορτώσουν τα αντικείμενα αυτά στον πραγματικό κόσμο μέσω της οθόνης του χρήστη (Σχήμα 5.2).



Σχήμα 5.1: Ιεραρχία αντικειμένου Image Target



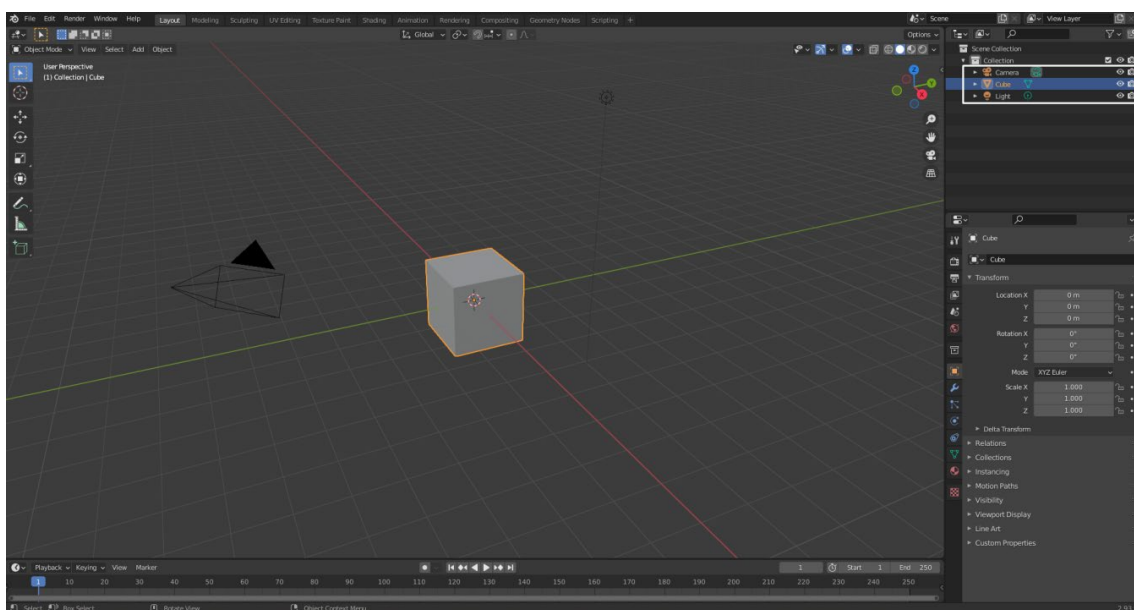
Σχήμα 5.2: Πρώτη σκηνή παιχνιδιού Π.Ε.

5.3 Τρισδιάστατα Αντικείμενα

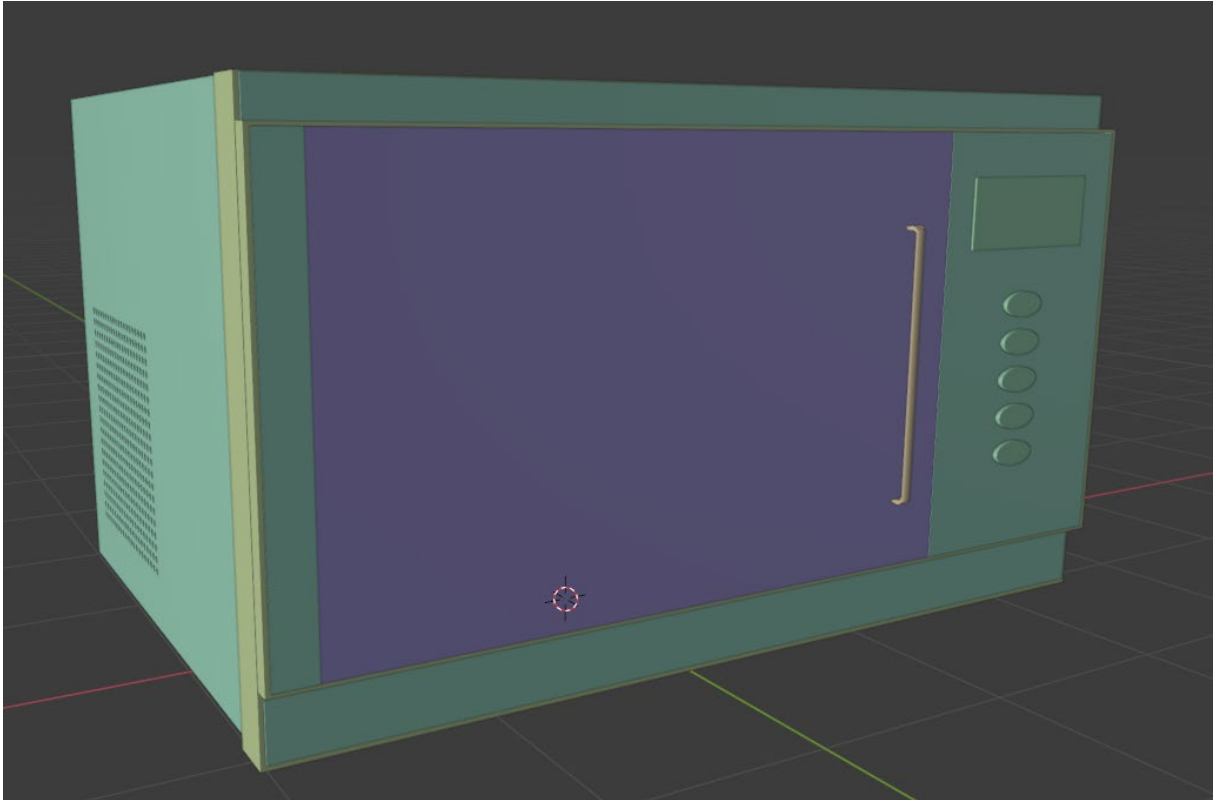
Όλα τα τρισδιάστατα αντικείμενα τα οποία χρησιμοποιούνται στην Π.Ε. έχουν δημιουργηθεί με την χρήση της πλατφόρμας Blender [11].

5.3.1 Δημιουργία Αντικειμένων και Μετατροπή σε FBX

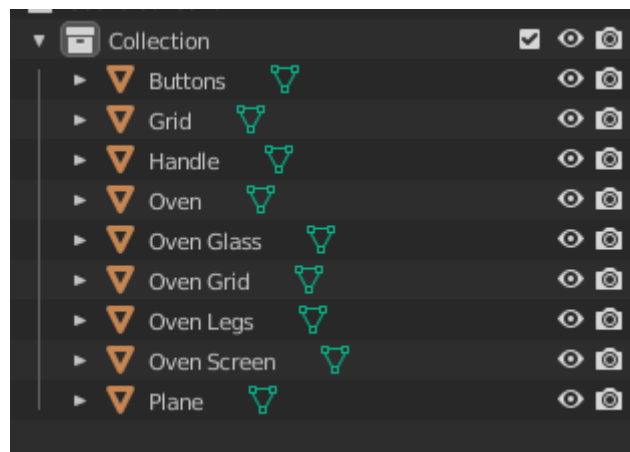
Κατά το άνοιγμα της εφαρμογής Blender, βλέπουμε έναν τρισδιάστατο χώρο και έναν επίσης τρισδιάστατο κύβο (Σχήμα 5.3). Στην σκηνή υπάρχουν επίσης μία κάμερα και ένα σημείο φωτός, τα οποία δεν θα μας χρειαστούν. Μέσω της πλατφόρμας αυτής, έχουμε την δυνατότητα να περιεργαστούμε βασικά αντικείμενα όπως και ο κύβος αυτός, αλλά και άλλα σχήματα, με σκοπό να δημιουργήσουμε τα τρισδιάστατα αντικείμενα που θέλουμε. Χαρακτηριστικό παράδειγμα είναι η επεξεργασία και ο συνδυασμός διαφορετικών σχημάτων (Σχήμα 5.5) για την δημιουργία ενός τρισδιάστατου φούρνου, το οποίο παρουσιάζεται και στο Σχήμα 5.4.



Σχήμα 5.3: Πλατφόρμα Blender



Σχήμα 5.4: Δημιουργία τρισδιάστατου φούρνου μέσω του Blender



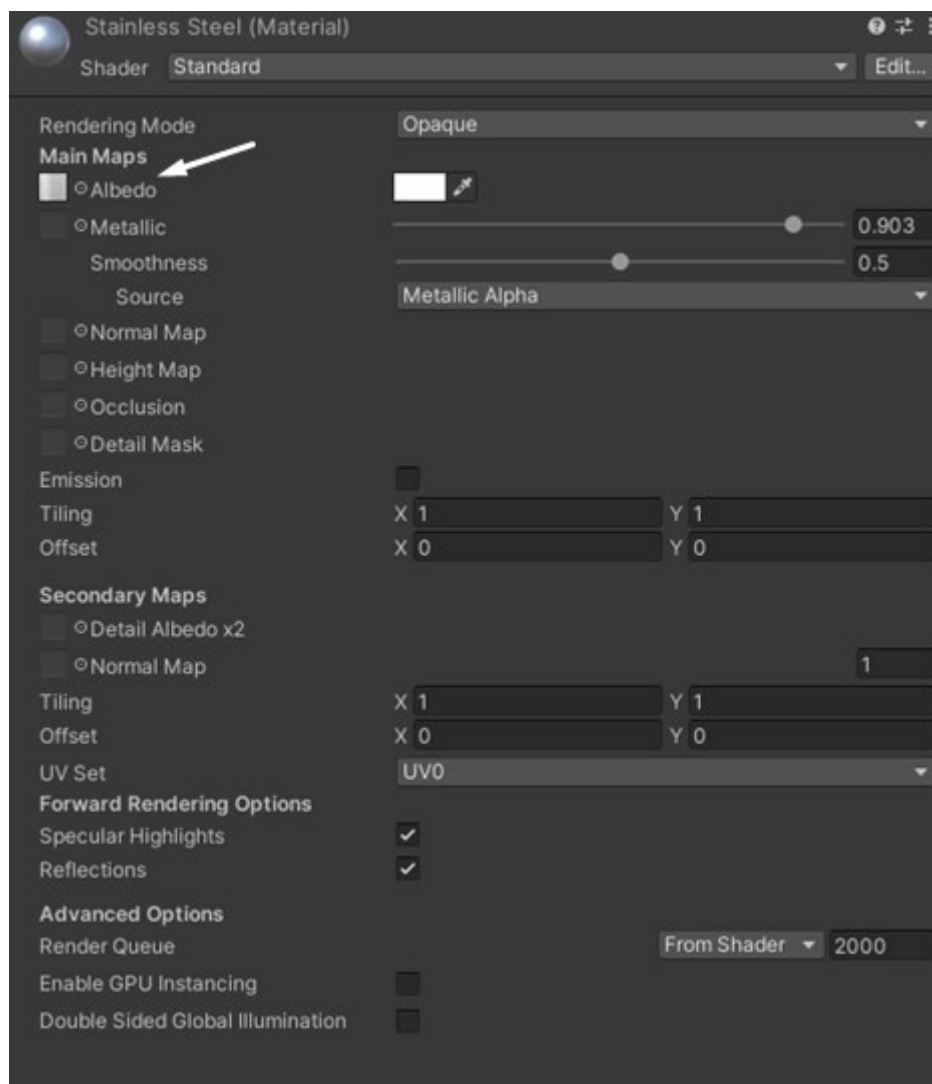
Σχήμα 5.5: Συνολικά αντικείμενα που αποτελούν τον τρισδιάστατο φούρνο

Έχοντας δημιουργήσει τα αντικείμενα, για να προστεθούν στην πλατφόρμα του Unity θα χρειαστεί να τα μετατρέψουμε σε αρχεία τύπου FBX. Αυτό γίνεται εύκολα μέσω της επιλογής του Blender, File→Export→FBX (.fbx). Με αυτόν τον τρόπο, σέρνοντας απλά το αρχείο τύπου FBX στην καρτέλα με τα Assets του Unity, το αντικείμενο που δημιουργήσαμε μπορεί να χρησιμοποιηθεί.

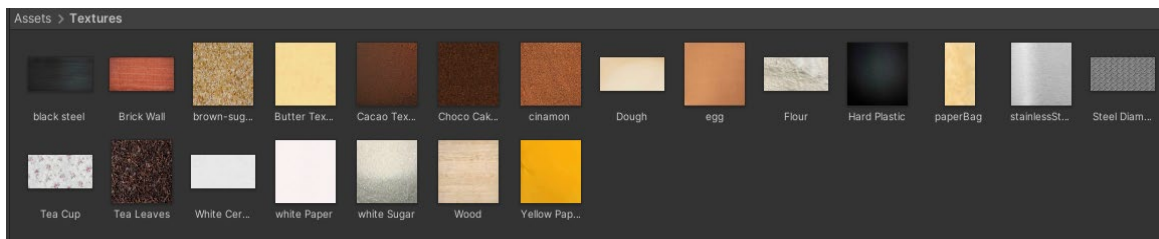
5.3.2 Materials Textures και Prefabs

Το επόμενο και σημαντικό βήμα μετά την πρόσθεση του εκάστοτε αντικειμένου στην πλατφόρμα του Unity, είναι ο σωστός χρωματισμός του και η δημιουργία ενός αντιπροσωπευτικού αντικειμένου Prefab. Αρχικά, ο χρωματισμός και η αίσθηση της επιφάνειας του αντικειμένου δημιουργείται μέσω

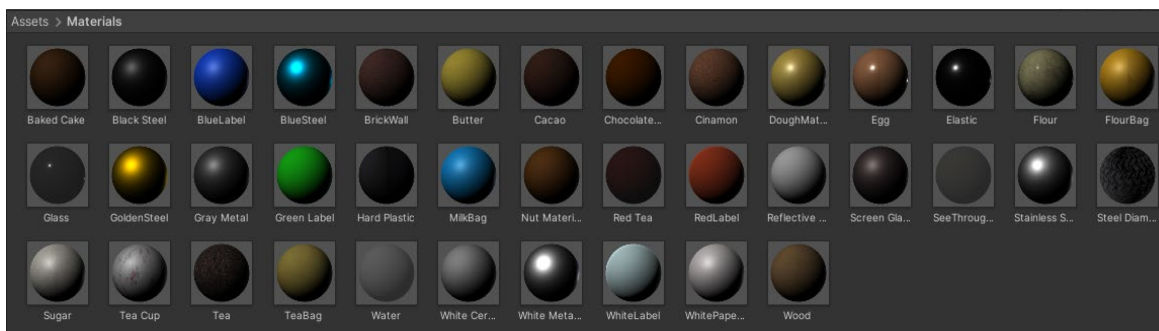
των Materials. Τα materials είναι αρχεία τα οποία προσθέτουμε πάνω σε αντικείμενα, ώστε να ορίσουμε την εμφάνιση τους. Τα αρχεία αυτά έχουν πεδία τα οποία χαρακτηρίζουν τον χρωματισμό, την μεταλλικότητα της επιφάνειας, το πόσο λεία είναι, καθώς και πολλά άλλα χαρακτηριστικά, τα οποία ορίζουν την εκάστοτε εμφάνιση (Σχήμα 5.6). Μέσα σε αυτά τα αρχεία, μπορεί επίσης να οριστεί και κάποιο texture, δηλαδή κάποιο αρχείο το οποίο απεικονίζει πολύ πιο συγκεκριμένα την εμφάνιση μίας επιφάνειας. Για παράδειγμα στο σχήμα 5.6 αντί για χρωματισμός ορίστηκε ένα αρχείο texture, το οποίο πρόκειται για μία εικόνα ατσάλινιας επιφάνειας, σε συνδυασμό με τις τιμές των πεδίων Metallic και Smoothness, ώστε να προσομοιάσουμε εικονικά την επιφάνεια του ατσαλιού. Ένας συνδυασμός Materials και Textures χαρακτηρίζουν όλα τα εικονικά αντικείμενα του παιχνιδιού, τα οποία έπειτα απο την πρόσθεση χαρακτηριστικών όπως colliders και scripts μπορούν να αποθηκευτούν ως Prefabs, δηλαδή αντιπροσωπευτικά αντικείμενα έτοιμα για χρήση χωρίς να χρειάζεται να επαναπροσδιορίσουμε τα χαρακτηριστικά τους και την εμφάνιση τους (Σχήμα 5.10). Η δημιουργία ενός prefab γίνεται εύκολα σέρνοντας το αντικείμενο από την ιεραρχία της σκηνής στην καρτέλα με τα Assets και επιλέγοντας Original Prefab στο μήνυμα που εμφανίζεται.



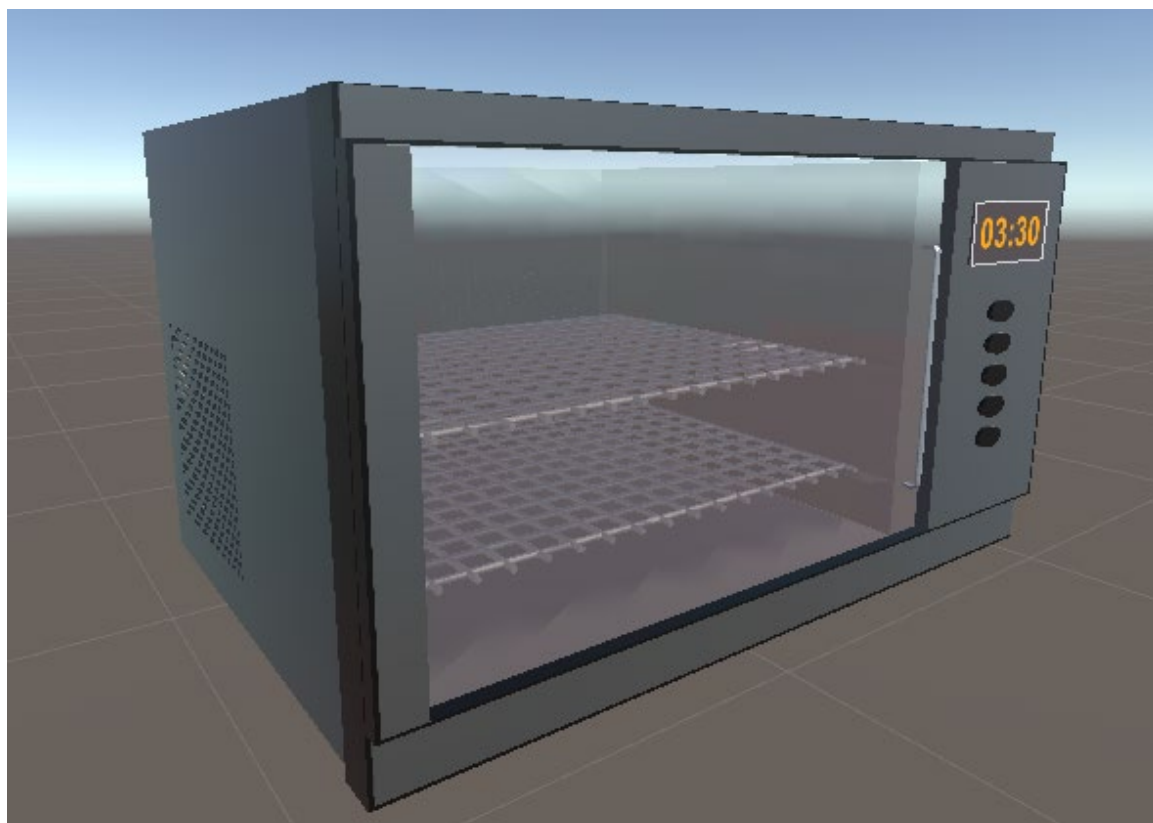
Σχήμα 5.6: Material ανοξιδίου ατσαλιού με χρήση Texture



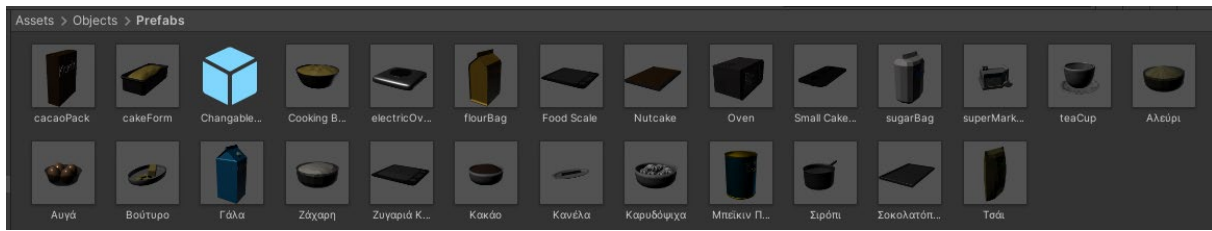
Σχήμα 5.7: Συνολικά Textures παιχνιδιού Π.Ε.



Σχήμα 5.8: Συνολικά Materials παιχνιδιού Π.Ε.



Σχήμα 5.9: Φούρνος έπειτα από την χρήση materials και textures



Σχήμα 5.10: Συνολικά Prefabs παιχνιδιού Π.Ε.

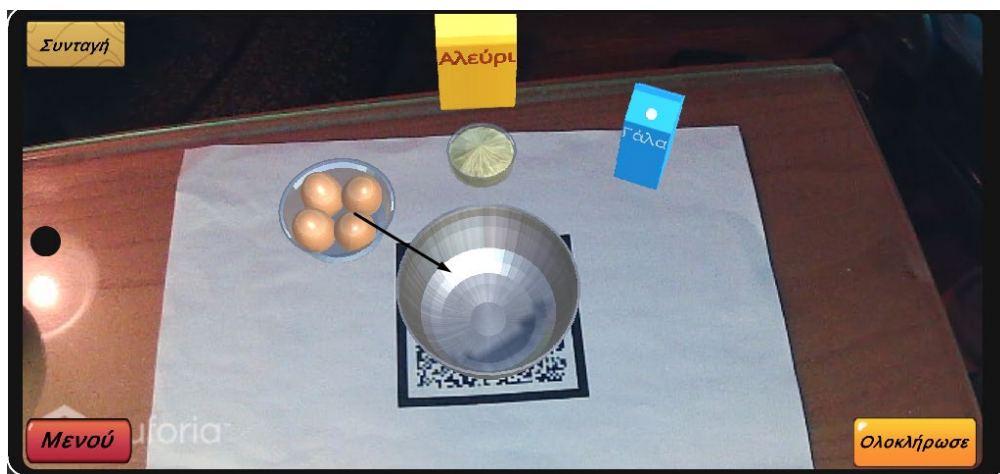
5.4 Scripts

Όλες οι λειτουργίες του παιχνιδιού πραγματοποιούνται με βάση κάποιου αρχείου script, το οποίο περιγράφει μία λειτουργία και τον τρόπο με τον οποίο πραγματοποιείται. Η γλώσσα η οποία χρησιμοποιείται για την γραφή αυτών των αρχείων κώδικα, είναι η C# μέσω της πλατφόρμας Visual Studio [15], η οποία συνεργάζεται μαζί με το Unity. Τα βασικά scripts τα οποία περιγράφουν τις λειτουργίες του παιχνιδιού είναι τα :

1. DragAndDrop
2. CraftingRecipe
3. MainUiHandler
4. BowlHandler
5. Numpad & NumpadHandler
6. Questionnaire & QuestionnaireHandler
7. ChangeValues

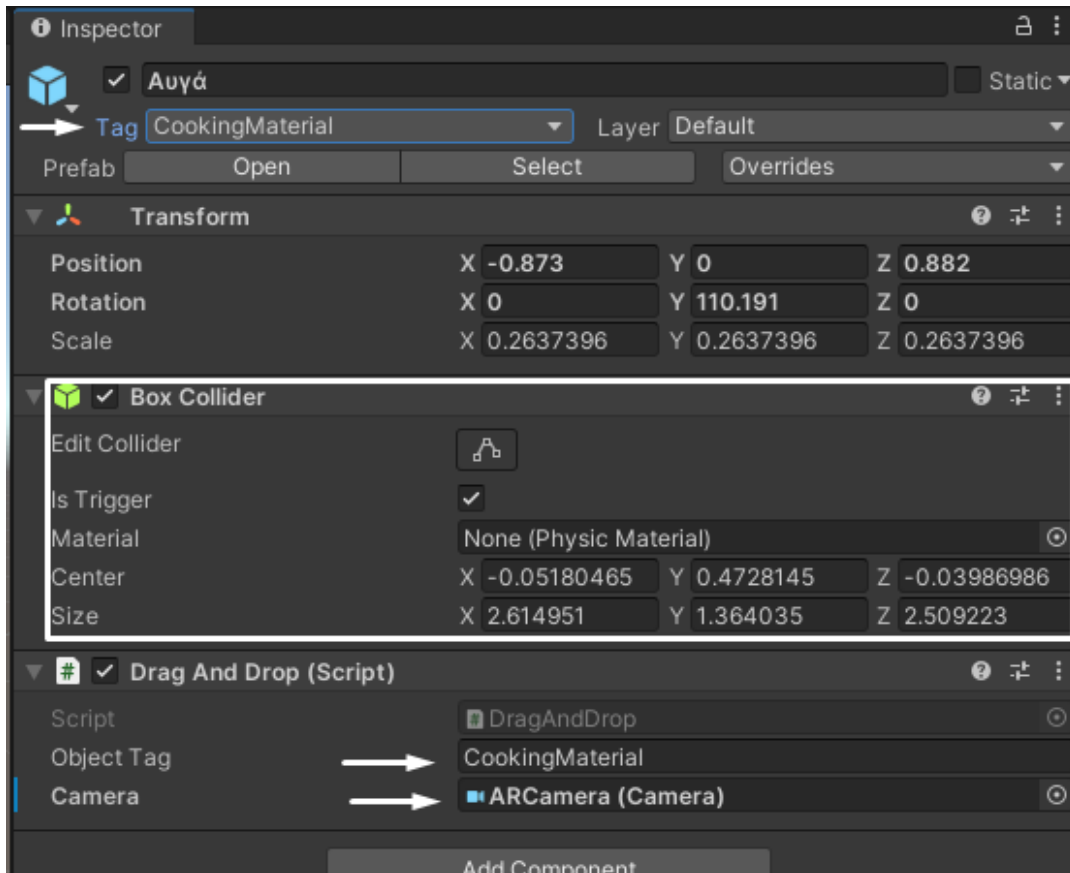
5.4.1 DragAndDrop

Πρώτη και πιο σημαντική λειτουργία που έπρεπε να δημιουργηθεί, είναι ο τρόπος με τον οποίον ο χρήστης θα αλληλεπιδρά με τα εικονικά αντικείμενα του παιχνιδιού. Λόγω της χρήσης AR κάμερας, η δημιουργία ενός τρόπου αλληλεπίδρασης σε συνδυασμό με αυτήν αποτελούσε πρώτο εμπόδιο στην ανάπτυξη του παιχνιδιού. Η λύση σε αυτό το εμπόδιο περιγράφεται από το script DragAndDrop. Το script δίνει στον χρήστη την ικανότητα να σέρνει και να αφήνει αντικείμενα που θέλει να χρησιμοποιήσει. Συγκεκριμένα, στην εφαρμογή η χρήση των αντικειμένων γίνεται σέρνοντας το αντίστοιχο αντικείμενο στο σημείο που πρέπει. Για παράδειγμα έχοντας ένα μπολ στο οποίο αναμειγνύουμε τα υλικά, η χρήση και ανάμιξη των υλικών γίνεται σέρνοντας τα υλικά στο σημείο όπου βρίσκεται το μπολ (Σχήμα 5.11).



Σχήμα 5.11: Σύρσιμο αυγών σε μπολ μίξης

Τα τρία βασικά στοιχεία τα οποία χρησιμοποιεί το script για την λειτουργία του είναι ένα tag, το αντίστοιχο collider του εκάστοτε αντικειμένου και την κάμερα της συσκευής (Σχήμα 5.12).



Σχήμα 5.12: Στοιχεία αντικειμένου αυγών & μεταβλητές του script DragAndDrop

Το tag, πρόκειται απλά για ένα χαρακτηριστικό ορισμό που μπορούμε να προσθέσουμε σε ένα αντικείμενο. Συγκεκριμένα, όλα τα υλικά του παιχνιδιού τα οποία μπορεί να χρησιμοποιήσει ο παίκτης φέρουν το tag 'CookingMaterial'. Άλλο ένα στοιχείο που πρέπει να φέρουν πολλά από τα αντικείμενα μας είναι κάποιο collider, δηλαδή ένα στοιχείο το οποίο ανιχνεύει συγκρούσεις και αντιπροσωπεύει τον χώρο που καταλαμβάνει ένα αντικείμενο σε πραγματικό χρόνο. Μέσω αυτών, μπορούμε να ανιχνεύσουμε πότε ένα αντικείμενο θα έρθει σε επαφή με κάποιο άλλο και να χειριστούμε αυτήν την πληροφορία ανάλογα. Το τελευταίο στοιχείο πρόκειται για την κάμερα του παιχνιδιού, μέσω της οποίας μας δίνεται η δυνατότητα να δημιουργήσουμε μία νοητή γραμμή με σημείο αρχής την ίδια, αυτή η νοητή γραμμή ονομάζεται Ray.

Έχοντας αναλύσει τα απαραίτητα στοιχεία, μπορούμε να προχωρήσουμε στην πράξη και συγκεκριμένα στο πώς χρησιμοποιούνται τα στοιχεία αυτά μέσα στον κώδικα του script. Το script δέχεται ως εξωτερικές μεταβλητές το tag των υλικών που θέλουμε να μπορούν να χρησιμοποιηθούν, καθώς και την κάμερα μέσω της οποίας τα βλέπουμε (Σχήμα 5.12). Επιπλέον το script χρησιμοποιεί τρεις μεθόδους για την διεκπεραίωση της λειτουργίας του: την OnTouch(), την FixedUpdate() και την OnTriggerEnter.

1. Η μέθοδος OnTouch() πρόκειται για τον κορμό του script, καθώς περιγράφει το μεγαλύτερο μέρος της λειτουργίας του. Αρχικά, σε αυτήν την μέθοδο γίνεται έλεγχος για το αν ο χρήστης έχει ακουμπήσει την οθόνη αφής της συσκευής του. Σε περίπτωση που δεν έχει ακουμπήσει, τότε δύο μεταβλητές που λειτουργούν ως flags, μία για το αν ο χρήστης έχει ακουμπήσει κάποιο αντικείμενο και μία για το αν ο χρήστης σέρνει κάποιο αντικείμενο, παίρνουν τιμή

false και η μέθοδος τελειώνει. Αν ο χρήστης χρησιμοποιήσει την αφή, τότε αποθηκεύεται το σημείο στο οποίο ακούμπησε, σε μία μεταβλητή τρισδιάστατου σημείου Vector3. Έπειτα ελέγχονται οι τρεις προκαθορισμένες φάσεις της διαδικασίας αφής, η αρχή, η μετακίνηση και το τέλος. Κατά την φάση αρχής, δημιουργούμε ένα στοιχείο Ray με σημείο αρχής την κάμερα και κατεύθυνση το σημείο αφής του χρήστη. Έπειτα, ελέγχουμε αν το Ray έχει έρθει σε επαφή με κάποιο αντικείμενο και συγκεκριμένα αν το αντικείμενο επαφής φέρει το χαρακτηριστικό tag, 'CookingMaterial'. Στην περίπτωση που το αντικείμενο φέρει αυτό το tag, τότε αποθηκεύεται η αρχική του θέση και ορίζεται η flag μεταβλητή επαφής ως true. Η δεύτερη φάση, αφορά την φάση μετακίνησης και προϋποθέτει ότι ο χρήστης έχει ήδη έρθει σε επαφή με κάποιο αντικείμενο μέσω της αφής. Κατά την στιγμή που ο χρήστης αλλάζει το σημείο αφής του στην οθόνη, αποθηκεύεται η νέα θέση αφής και μετακινείται το αντικείμενο σε αυτή. Τέλος, όταν ο χρήστης αφήσει την οθόνη και πάψει η διαδικασία αφής, τότε το αντικείμενο επιστρέφει στην αρχική του θέση, την οποία είχαμε αποθηκεύσει στην πρώτη φάση και οι μεταβλητές flags παίρνουν πάλι τιμές false.

2. Η μέθοδος FixedUpdate(), πρόκειται για μία προκαθορισμένη λειτουργία-μέθοδος του Unity η οποία τρέχει επαναληπτικά περισσότερο από μία φορές το καρέ. Έχουμε ορίσει η μέθοδος αυτή να καλεί την παραπάνω μέθοδο OnTouch, ώστε να εκτελείται σε πραγματικό χρόνο.
3. Τέλος, έχουμε την επίσης προκαθορισμένη μέθοδο OnTriggerEnter() η οποία καλείται κάθε φορά που ένα αντικείμενο το οποίο διαθέτει collider με ιδιότητα Trigger, έρχεται σε σύγκρουση με κάποιο άλλο. Κατά την μέθοδο αυτή αν το αντικείμενο σύγκρουσης δεν φέρει το tag 'CookingMaterial', ορίζουμε τα προαναφερθέντα flags ως false και επιστρέφουμε το αντικείμενο στην αρχική του θέση, απενεργοποιώντας την ιδιότητα isTrigger. Η λογική πίσω από αυτήν την μέθοδο αφορά την χρήση των υλικών και συγκεκριμένα την μη επιτρεπτή χρήση υλικών περισσότερο της μίας φορές. Για παράδειγμα στο σχήμα 5.11, όταν ο χρήστης προσθέσει τα αυγά στο μπολ μίξης, αυτά θα επιστρέψουν στην αρχική τους θέση και δεν θα μπορούν να ξαναχρησιμοποιηθούν. Αυτό συμβαίνει διότι ακόμη και αν ο χρήστης τα σύρει μέσα στο μπολ, έχοντας απενεργοποιηθεί η ιδιότητα isTrigger, δεν θα αναγνωριστεί η σύγκρουση. Η λογική αυτής της μεθόδου λειτουργεί σε συνδυασμό με άλλα scripts, τα οποία θα αναλύσουμε παρακάτω.

```
private void OnTouch()
{
    if (Input.touchCount != 1) //If there are less or more than one finger touching don't do anything
    {
        dragging = false;
        touched = false;
        return;
    }

    //Get the position of the first touch
    Touch touch = Input.touches[0];
    Vector3 positionOfTouch = touch.position;

    if (touch.phase == TouchPhase.Began)
    {
        rayHitInfo = new RaycastHit();
        Ray ray = camera.ScreenPointToRay(positionOfTouch); //Creates a Ray(line) from the camera to the first position that was touched

        //Returns true if ray collides with any object using {ray} as point of origin and if true {rayHitInfo} contains information about the ob
        if (Physics.Raycast(ray, out rayHitInfo) && rayHitInfo.collider.CompareTag(objectTag)) //And check if the object hit has the appropriate
        {
            toDrag = rayHitInfo.transform; //Get the transform of the object that was touched
            previousPosition = toDrag.position;
            originalPosition = previousPosition;

            distance = camera.WorldToScreenPoint(previousPosition); // Save the screen location and depth of the object that was touched (Vector
            newPosition = new Vector3(distance.x, distance.y, distance.z);

            touched = true;
        }
    }
}
```

Σχήμα 5.13: Script DragAndDrop μέθοδος OnTouch πρώτο μέρος

```

if (touched && touch.phase == TouchPhase.Moved)
{
    dragging = true;

    float newPosX = Input.GetTouch(0).position.x;
    float newPosY = Input.GetTouch(0).position.y;
    Vector3 currentPosition = new Vector3(newPosX, newPosY, distance.z); //Get the new position with each movement

    Vector3 worldPosition = camera.ScreenToWorldPoint(currentPosition); //Make movement relative to the world
    worldPosition = new Vector3(worldPosition.x, 0.0f, worldPosition.z); //remove the z movement

    previousPosition = toDrag.position; //save the current position as the last known position before changing it
    toDrag.position = worldPosition; //change the position of the object to the new world position
}

if (dragging && (touch.phase == TouchPhase.Ended || touch.phase == TouchPhase.Canceled))
{
    dragging = false;
    touched = false;

    previousPosition = new Vector3();
    toDrag.position = originalPosition;
}
}

```

Σχήμα 5.14: Script DragAndDrop μέθοδος onTouch δεύτερο μέρος

```

Unity Message | 0 references
private void FixedUpdate()
{
    OnTouch();
}

Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    if(other.gameObject.tag != "CookingMaterial")
    {
        dragging = false;
        touched = false;
        toDrag.position = originalPosition;
        this.GetComponent<Collider>().isTrigger = false; //disable the trigger when the material is used
    }
}

```

Σχήμα 5.14: Script DragAndDrop μέθοδοι FixedUpdate & OnTriggerEnter

5.4.2 CraftingRecipe

Το script CraftingRecipe περιγράφει τις συνταγές του παιχνιδιού και την λειτουργία τους. Συγκεκριμένα, το script αποτελείται από δύο μέρη, το πρώτο αφορά ένα struct ονόματι Items και το δεύτερο την κλάση CraftingRecipe. Το struct Items πρόκειται για μία παραλλαγή κλάσης, η οποία έχει σχεδόν τις ίδιες ιδιότητες αλλά με μικρότερο υπολογιστικό κόστος για το σύστημα. Μέσα σε αυτό, εμπεριέχονται τρεις μεταβλητές και ένας δομητής. Το struct αυτό περιγράφει ένα υλικό συνταγής και οι τρεις μεταβλητές αφορούν το όνομα του υλικού, την ποσότητα και ένα πεδίο περιγραφής της ποσότητας.

```

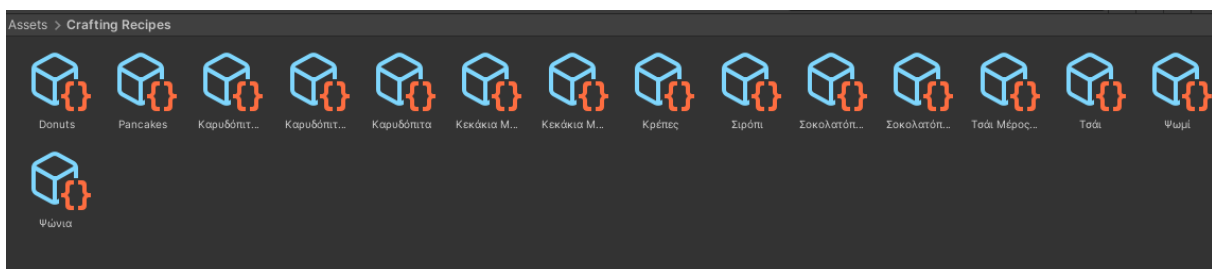
[Serializable]
15 references
public struct Items
{
    [SerializeField]
    public string itemName;
    [SerializeField]
    [Range(1,9999)]
    public float itemAmount;
    [SerializeField]
    public string itemAmountLabel;

    3 references
    public Items(string itemName, float itemAmount, string itemAmountLabel)
    {
        this.itemName = itemName;
        this.itemAmount = itemAmount;
        this.itemAmountLabel = itemAmountLabel;
    }
}

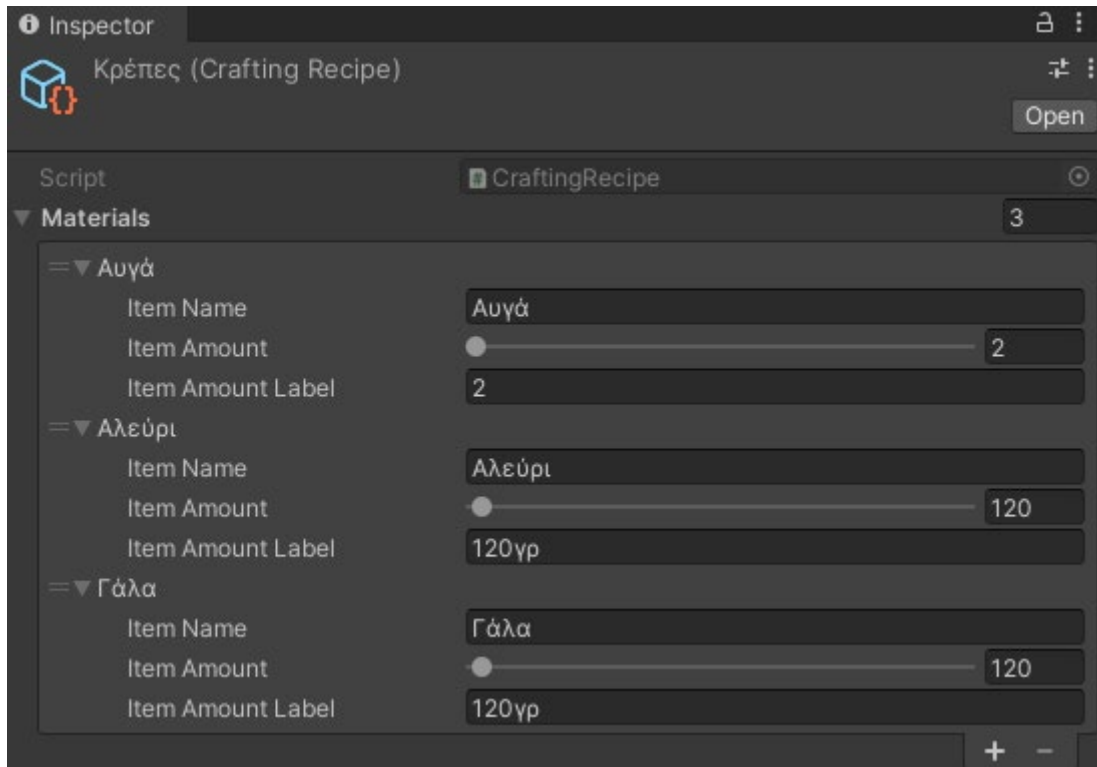
```

Σχήμα 5.15: Script CraftingRecipe struct Items

Το δεύτερο μέρος του script πρόκειται για μία κλάση τύπου ScriptableObject, η οποία περιέχει μία λίστα Materials με αντικείμενα τύπου Items και μία μέθοδο τύπου Boolean, την CanCraft(), η οποία ελέγχει αν ο χρήστης έχει ολοκληρώσει σωστά την συνταγή με βάση τα υλικά που χρησιμοποίησε. Ο ορισμός της κλάσης ως ScriptableObject μαζί με την εντολή CreateAssetMenu, μας δίνει την δυνατότητα να προσθέτουμε με ένα απλό δεξί κλικ αντικείμενα τύπου CraftingRecipe απευθείας στον φάκελο των Assets του Unity. Συγκεκριμένα για κάθε συνταγή του παιχνιδιού έχει δημιουργηθεί και το αντίστοιχο scriptable object (Σχήμα 5.16). Έπειτα, τα αντικείμενα αυτά διαθέτουν την λίστα των Materials που ορίσαμε και μπορούμε εύκολα να προσθέτουμε και να αφαιρούμε υλικά (Σχήμα 5.17)



Σχήμα 5.16: Scriptable objects συνταγών παιχνιδιού Π.Ε.



Σχήμα 5.17: Scriptable Object συνταγής κρεπών & τιμές μεταβλητών λίστας Materials

```
[CreateAssetMenu(fileName = "New Recipe", menuName = "Recipe")]
@ Unity Script | 3 references
public class CraftingRecipe : ScriptableObject
{
    [SerializeField]
    public List<Items> materials;

    1 reference
    public bool CanCraft() {
        bool flag = true;

        //if the player didn't use all the items he need, he won't be able to craft
        if (BowlHandler.itemsUsed.Count != materials.Count)
        {
            Debug.Log(materials.ToString());
            return false;
        }else if (materials.Count == 0) //if there is no materials needed to craft return true;
        {
            return true;
        }

        //if the player used the wrong amounts of an item, the recipe won't be correct
        foreach(Items items in materials)
        {
            for(int i=0; i < BowlHandler.itemsUsed.Count; i++){
                if (items.itemName == BowlHandler.itemsUsed[i].itemName && items.itemAmount != BowlHandler.itemsUsed[i].itemAmount)
                {
                    flag = false;
                }
            }
        }

        return flag;
    }
}
```

Σχήμα 5.18: Script CraftingRecipe κλάση CraftingRecipe τύπου ScriptableObject

5.4.3 MainUiHandler

Άλλο ένα πολύ σημαντικό script είναι το MainUiHandler, καθώς μέσω αυτού πραγματοποιούνται οι περισσότερες λειτουργίες της διεπαφής χρήστη. Το script αυτό δέχεται ως εξωτερικές μεταβλητές όλα τα panels που ανήκουν στην διεπαφή, καθώς και το αντικείμενο συνταγής CraftingRecipe της εκάστοτε σκηνής. Η κλάση MainUiHandler εμπεριέχει πολλές μεθόδους, από τις οποίες οι περισσότερες πραγματοποιούν μία αρκετά απλή διαδικασία, όπως το να εμφανίζουν και να κλείνουν κάποιο panel. Για παράδειγμα, οι μέθοδοι showRecipe(), hideRecipe(), showTipPanel() και hideTipPanel() έχουν ανατεθεί σε κάποιο κουμπί και τρέχουν όταν ο χρήστης κάνει κλικ σε αυτό. Συγκεκριμένα κατα την κλήση της μεθόδου showRecipe() (Σχήμα 5.19), γίνεται έλεγχος για το αν το αντικείμενο CraftingRecipe εμπεριέχει υλικά. Σε περίπτωση που δεν περιέχει υλικά, σημαίνει ότι στο συγκεκριμένο σενάριο ο παίκτης πρέπει να κάνει κάποια διαφορετική ενέργεια από το να ολοκληρώσει την συνταγή και καλείται η showTipPanel(), η οποία θα εμφανίσει στον παίκτη κάποια συμβουλή μέσω του αντίστοιχου panel. Αν όμως η συνταγή έχει υλικά, τότε εμφανίζεται το panel της συνταγής στην οποία έχουν γραφτεί προγραμματιστικά (Σχήμα 5.20).

```

//enable and load the recipe panel
1 reference
public void showRecipe()
{
    //if the recipe is empty then show a tip instead of the recipe
    if (recipe.materials.Count == 0)
    {
        showTipPanel();
    }
    else
    {
        List<Items> materials = recipe.materials;
        RecipePanel.SetActive(true);
        string materialsText = "";
        foreach (Items item in materials)
        {
            materialsText += item.itemAmountLabel + " " + item.itemName + "\n";
        }
        RecipePanel.GetComponentInChildren<TextMeshProUGUI>().SetText(materialsText);
    }
    GameObject.Find("RecipeTitle").GetComponent<Text>().text = recipe.name;
}

```

Σχήμα 5.19: Script MainUiHandler μέθοδος showRecipe()



Σχήμα 5.20: Εμφάνιση διεπαφής Recipe

Άλλη μια αξιοσημείωτη μέθοδος είναι αυτή που έχει ανατεθεί στο κουμπί ‘Ολοκλήρωσε’, κατά την οποία καλείται η μέθοδος CanCraft() του αντικείμενου συνταγής CraftingRecipe και σε περίπτωση που επιστραφεί true, τότε ελέγχεται για το αν το σενάριο εμπεριέχει επιπλέον ερωτήσεις, τις οποίες πρέπει να απαντήσει ο παίκτης. Αν υπάρχουν τότε εμφανίζεται το αντίστοιχο panel, αλλιώς εμφανίζεται μήνυμα επιτυχίας (Σχήμα 5.22) και σε περίπτωση λάθους εμφανίζεται μήνυμα αποτυχίας (Σχήμα 5.23).

```

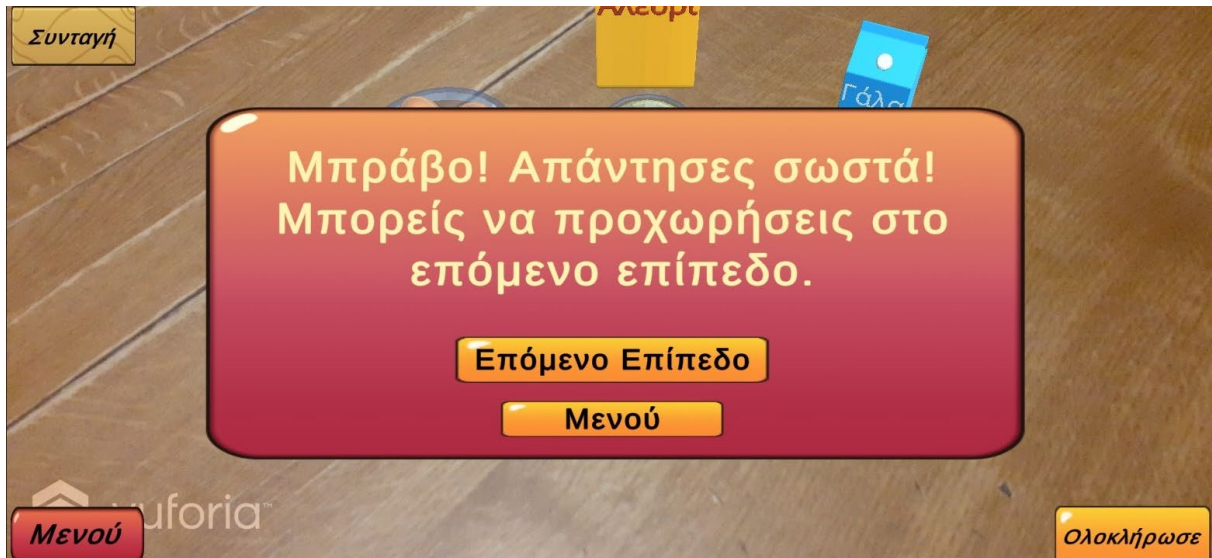
0 references
public void onCraftClick(CraftingRecipe recipe)
{
    //get the full list of questions and answers from the QuestionnaireHandler
    List<QuestionAndAnswer> qna = QuestionnairePanel.GetComponent<QuestionnaireHandler>().QnA;

    //enable the questionnaire panel
    bool canBeCrafted = recipe.CanCraft();

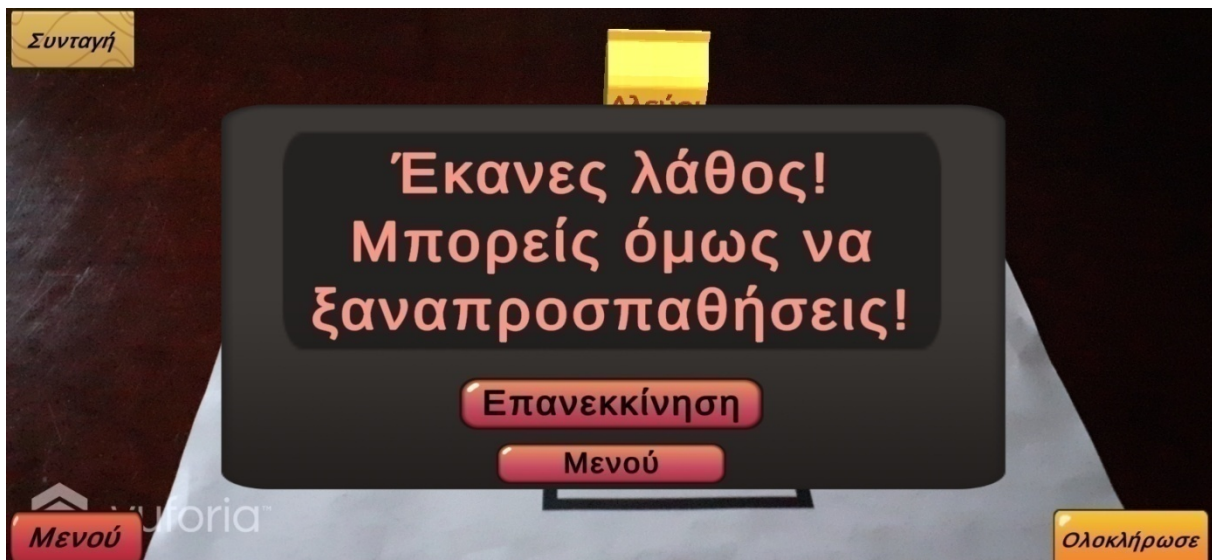
    if (canBeCrafted && qna.Count!=0)
    {
        QuestionnairePanel.SetActive(true);
    }
    else if(canBeCrafted && qna.Count == 0)
    {
        CompletedTaskPanel.SetActive(true);
    }
    else
    {
        FailedTaskPanel.SetActive(true);
    }
}

```

Σχήμα 5.21: Script MainUiHandler μέθοδος onCraftClick()



Σχήμα 5.22: Εμφάνιση μηνύματος επιτυχίας κατά την ολοκλήρωση συνταγής



Σχήμα 5.23: Εμφάνιση μηνύματος αποτυχίας κατά την ολοκλήρωση συνταγής

5.4.4 BowlHandler

Το script BowlHandler αφορά την λειτουργία ενός μπολ μίξης, το οποίο ελέγχει το πόσα και ποιά υλικά χρησιμοποιήθηκαν κατα την διάρκεια εκτέλεσης μίας συνταγής. Η λογική του εφαρμόζεται σε οποιοδήποτε σενάριο χρειάζεται να γίνει σωστή χρήση κάποιου υλικού, χωρίς αναγκαστικά να αφορά μία κλασική συνταγή. Το script ορίζει μία στατική λίστα με όνομα itemsUsed, στην οποία αποθηκεύονται τα ονόματα των υλικών κατα την χρήση τους απο τον παίκτη. Συγκεκριμένα, έχει οριστεί η μέθοδος OnTriggerEnter(), η οποία κατα την στιγμή που ένα αντικείμενο έρθει σε επαφή με το αντικείμενο το οποίο φέρει το script, αποθηκεύεται το όνομα του σε μία λίστα υλικών. Επίσης χρησιμοποιείται και η μέθοδος Start(), η οποία κατά την αρχή της σκηνής βρίσκει τον αριθμό των υλικών που μπορούν να χρησιμοποιηθούν, έτσι ώστε σε συνδιασμό με την απενεργοποίηση της ιδιότητας isTrigger, που συμβαίνει στο script DragAndDrop, δεν μπορεί να γίνει χρήση υλικών

περισσότερο απο μία φορά το κάθε ένα. Μέσω του script BowlHandler ενεργοποιείται και η διεπαφή Numpad (Σχήμα 5.25), η οποία φέρει το script NumpadHandler το οποίο εξηγείται παρακάτω.

```

1 reference
private void activateNumpadUI()
{
    amountUI.SetActive(true);
}

```

Σχήμα 5.24: Script BowlHandler μέθοδος ενεργοποίησης Numpad

```

[SerializeField]
private GameObject amountUI;

private int collisionCounter;
private int numberOfMaterials;

public static List<Items> itemsUsed = new List<Items>();

Unity Message | 0 references
private void Start()
{
    numberOfMaterials = GameObject.FindGameObjectsWithTag("CookingMaterial").Length;
}

Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    activateNumpadUI();

    collisionCounter += 1;
    //limits how many materials can be used to fill the bowl
    if (collisionCounter <= numberOfMaterials)
    {
        string itemNameCol = other.name;

        var newItemAdded = new Items(itemNameCol, 0, "");
        itemsUsed.Add(newItemAdded);
    }
}

```

Σχήμα 5.25: Script BowlHandler μέθοδοι OnTriggerEnter & Start

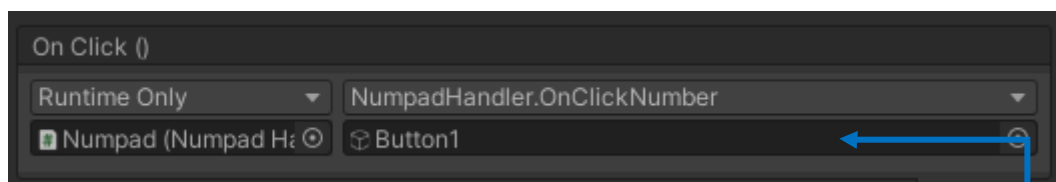
5.4.5 Numpad & NumpadHandler

Άλλη μία βασική ικανότητα η οποία πρέπει να διατίθεται στον παίκτη, είναι η επιλογή του αριθμού της ποσότητας που θέλει να χρησιμοποιήσει από κάποιο υλικό. Η ικανότητα αυτή δίνεται με την χρήση μίας διεπαφής, η οποία λειτουργεί με βάση το script NumpadHandler. Η διεπαφή πρόκειται για ένα απλό αριθμητικό πληκτρολόγιο το οποίο χρησιμοποιεί ο παίκτης για δώσει την εκάστοτε τιμή που

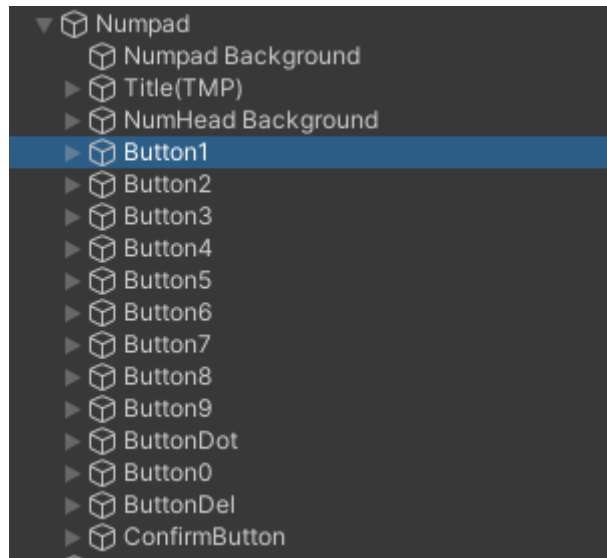
επιθυμεί (Σχήμα 5.26). Το script NumpadHandler αποτελείται από δύο μεθόδους, την OnClickNumber() και την passValueToBowl(). Η πρώτη μέθοδος δέχεται σαν μεταβλητή ένα αντικείμενο και εξάγει το κείμενο που είναι γραμμένο σε αυτό. Συγκεκριμένα, κάθε κουμπί της διεπαφής έχει οριστεί να καλεί αυτήν την μέθοδο κατά την στιγμή που γίνει κλικ και σαν μεταβλητή περνάει τον εαυτό του. Έπειτα, γίνεται έλεγχος για το αν το κείμενο που βρίσκεται μέσα στο κουμπί που πατήθηκε αντιστοιχεί σε αριθμό. Αν αντιστοιχεί σε αριθμό, τότε το κείμενο προστίθεται σε μία μεταβλητή τύπου string, αν αντιστοιχεί στο κουμπί σβήσιματος τότε σβήνεται ο τελευταίος αριθμός που προστέθηκε, αλλιώς αν αντιστοιχεί στο κουμπί 'Επιβεβαίωση' τότε καλείται η δεύτερη μέθοδος passValueToBowl(). Η μέθοδος αυτή ενημερώνει την στατική λίστα του script BowlHandler, η οποία κατέχει ήδη το όνομα του τελευταίου υλικού που χρησιμοποιήθηκε, και προσθέτει στην λίστα το ποσό του υλικού. Αφού ανανεωθεί η λίστα, η διεπαφή κλείνει και επαναφέρει τις μεταβλητές της στην αρχική τους τιμή.



Σχήμα 5.26: Διεπαφή Numpad παιχνιδιού



Σχήμα 5.27: OnClick γεγονός στο κουμπί Button1



Σχήμα 5.28: Ιεραρχία και δομή διεπαφής Numpad

```
Unity Script (15 asset references) | 0 references
public class NumpadHandler : MonoBehaviour
{
    [SerializeField]
    private GameObject numHead;
    private string numInput="";

    public float numValue = 0;

    0 references
    public void OnClickNumber(GameObject button)
    {
        string inputText = button.GetComponentInChildren<Text>().text;
        string oldText = numHead.GetComponent<Text>().text;

        if (!inputText.Equals("Επιβεβαίωση"))
        {
            if (inputText.Equals("C") && oldText.Length != 0)
            {
                string newText = oldText.Remove(oldText.Length - 1);
                numInput = newText;
            }
            else if (!inputText.Equals("C"))
            {
                numInput = numHead.GetComponent<Text>().text + inputText;
            }

            numHead.GetComponent<Text>().text = numInput;
        }
        else
        {
            //set the value of the amount chosen
            numValue = float.Parse(numInput);
            passValueToBowl(numValue);
        }
    }
}
```

Σχήμα 5.29: Script NumpadHandler μέθοδος OnClickNumber()

```

1 reference
private void passValueToBowl(float numValue)
{
    int lastItemUsed = BowlHandler.itemsUsed.Count - 1; //get the index of the last item used
    //creates the item that will update the values{itemAmount,itemAmountLabel} of the list itemsUsed
    Items updateItem = BowlHandler.itemsUsed[lastItemUsed];
    updateItem.itemAmount = numValue;
    updateItem.itemAmountLabel = numInput;
    //update the list item. It was originally holding only the name of the item, now it has both the values
    BowlHandler.itemsUsed[lastItemUsed] = updateItem;

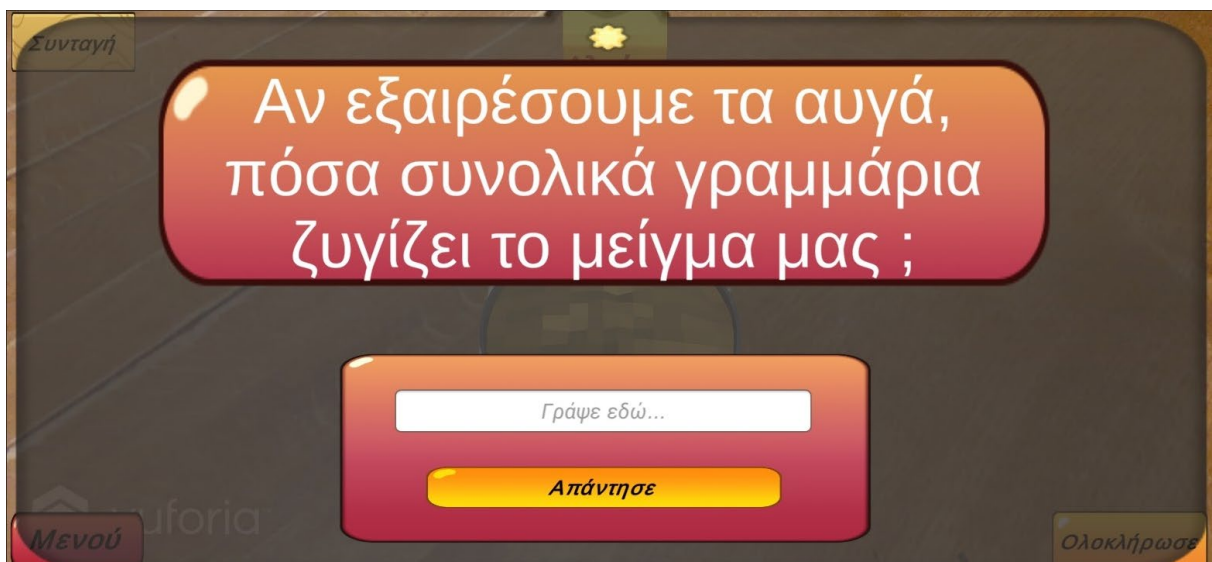
    //reset the numpad values and turn the ui off
    numValue = 0;
    numInput = "";
    numHead.GetComponent<Text>().text = numInput;
    this.gameObject.SetActive(false);
}
}

```

Σχήμα 5.30: Script NumpadHandler μέθοδος passValueToBowl()

5.4.6 Questionnaire & QuestionnaireHandler

Για την ολοκλήρωση μερικών συνταγών, ο παίκτης χρειάζεται να απαντήσει σε κάποια ερώτηση σχετική με την συνταγή, αλλά κυρίως σχετική με τα μαθηματικά. Αυτή η ανάγκη πρόσθεσης μικρών ερωτήσεων σε μερικές απο τις συνταγές μας οδήγησε στην δημιουργία της διεπαφής Questionnaire, καθώς και του script QuestionnaireHandler. Η διεπαφή παρουσιάζει την ερώτηση μέσα σε ένα περίγραμμα και ακριβώς απο κάτω διαθέτει ένα αντικείμενο τύπου Input Field, στο οποίο μπορεί ο παίκτης να πληκτρολογήσει την απάντηση του. Η διεπαφή εμφανίζεται με την προϋπόθεση ότι ο παίκτης έχει ολοκληρώσει σωστά την συνταγή την στιγμή που θα πιέσει το κουμπί ‘Ολοκλήρωσε’.



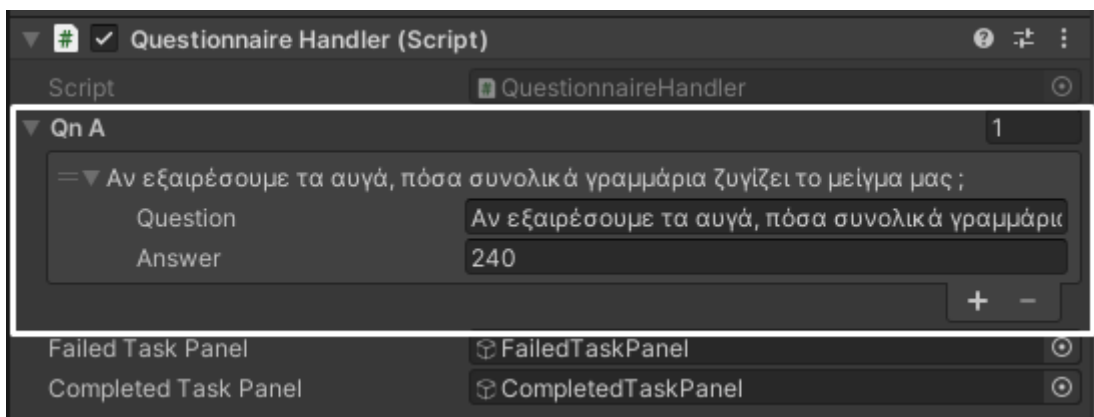
Σχήμα 5.31: Διεπαφή Questionnaire

Η λειτουργία της διεπαφής διεκπεραιώνεται μέσω του script QuestionnaireHandler και του struct QuestionAndAnswer. Το struct QuestionAndAnswer πρόκειται για μία παραλλαγή κλάσεως η οποία περιγράφει μία ερώτηση-απάντηση (Σχήμα 5.32). Αυτό χρησιμοποιείται από το script QuestionnaireHandler για την δημιουργία μίας λίστας η οποία θα περιέχει όσες ερωτήσεις-απαντήσεις θέλουμε (Σχήμα 5.33).

```
[Serializable]
11 references
public struct QuestionAndAnswer
{
    [SerializeField]
    public string question;
    [SerializeField]
    public string answer;

    2 references
    public QuestionAndAnswer(string question, string answer)
    {
        this.question = question;
        this.answer = answer;
    }
}
```

Σχήμα 5.32: Struct QuestionAndAnswer



Σχήμα 5.33: Λίστα ερωτήσεων-απαντήσεων

Το script λειτουργεί κυρίως με τρεις μεθόδους, την Start() την onAnswer() και την getCorrectTimeAfterChange(). Η πρώτη μέθοδος καλείται κατά την στιγμή που θα ανοίξει η διεπαφή και ορίζει το περιεχόμενο της, ώστε οι ερωτήσεις που έχουμε προσθέσει στην λίστα του αντικειμένου να εμφανίζονται στο αντίστοιχο μέρος της (Σχήμα 5.34). Έπειτα, χρησιμοποιείται μία ουρά (queue), στην οποία προστίθενται όλες οι ερωτήσεις και απαντήσεις της λίστας. Ο λόγος ύπαρξης της ουράς είναι για την καλύτερη επεκτασιμότητα του κώδικα, καθώς σε επόμενες σκηνές υπάρχουν παραπάνω απο μία ερωτήσεις. Με την χρήση ουράς μπορούμε να αφαιρούμε βηματικά ερωτήσεις κατά την ορθή απάντηση τους, χωρίς να πειράζουμε την αρχική λίστα που τις διαθέτει. Έτσι, όταν ο παίκτης απαντήσει στην πρώτη ερώτηση, αυτή θα αφαιρεθεί από την ουρά και θα αλλάξει η εμφανιζόμενη ερώτηση απευθείας στην αμέσως επόμενη. Ουσιαστικά, χρησιμοποιείται η λειτουργία FIFO (First In First Out) που χαρακτηρίζει τις ουρές, ώστε να έχουμε καλύτερη και πιο ασφαλή διαχείριση των ερωτήσεων και απαντήσεων μας.

```

public class QuestionnaireHandler : MonoBehaviour
{
    [SerializeField]
    public List<QuestionAndAnswer> QnA;

    [SerializeField]
    private GameObject FailedTaskPanel;

    [SerializeField]
    private GameObject CompletedTaskPanel;

    private GameObject QuestionnairePanel;

    private Queue<QuestionAndAnswer> QnAQueue = new Queue<QuestionAndAnswer>();

    Unity Message | 0 references
    public void Start()
    {
        QuestionnairePanel = GameObject.Find("QuestionnairePanel");

        //set the first question from the existing serialized list
        QuestionnairePanel.GetComponentInChildren<TextMeshProUGUI>().SetText(QnA[0].question);

        //store the questions with the corresponding answers on a queue for better Extensibility
        foreach (QuestionAndAnswer qna in QnA)
        {
            QnAQueue.Enqueue(qna);
        }
    }
}

```

Σχήμα 5.34: Script QuestionnaireHandler μέθοδος Start()

Η δεύτερη μέθοδος του script πρόκειται για την onAnswer() και καλείται την στιγμή που ο παίκτης πιέσει το κουμπί ‘Απάντησε’. Η μέθοδος αυτή αρχικά, παίρνει την απάντηση που δίνει ο παίκτης καθώς και την σωστή απάντηση η οποία είναι αποθηκευμένη στην ουρά. Έπειτα, ελέγχει τον τύπο της ερώτησης, κάτι το οποίο εξαρτάται από την απάντηση που είναι αποθηκευμένη. Συγκεκριμένα, υπάρχουν τρεις κατηγορίες ερώτησης:

➤ **Basic:** “Απλό κείμενο”

Στην συγκεκριμένη κατηγορία, η απάντηση δίνεται απευθείας χωρίς να χρειαστεί περαιτέρω επεξεργασία. Δηλαδή, γίνεται ένας απλός έλεγχος για το αν ο παίκτης έδωσε την ίδια απάντηση με αυτήν η οποία είναι αποθηκευμένη. Αν η απάντηση είναι ίδια, τότε σημαίνει ο παίκτης απάντησε σωστά και η ερώτηση-απάντηση αφαιρείται από την ουρά, αν όμως η απάντηση είναι διαφορετική, τότε εμφανίζεται μήνυμα λάθους. Έπειτα, αν δεν υπάρχουν άλλες αποθηκευμένες ερωτήσεις στην ουρά εμφανίζεται μήνυμα επιτυχίας, αλλιώς εμφανίζεται η επόμενη ερώτηση μέχρι να έχουν απαντηθεί όλες (Σχήμα 5.39).

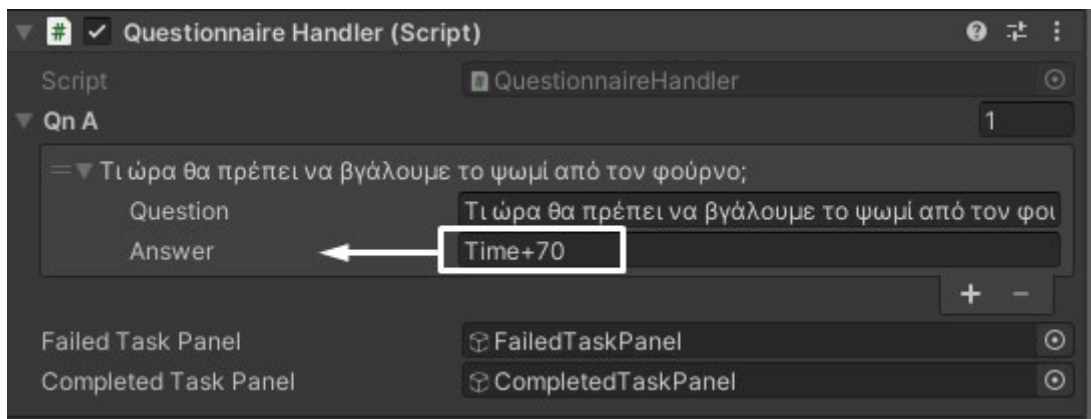
➤ **Time:** “Time+...”

Στην κατηγορία αυτή, οι απαντήσεις αποθηκεύονται με βάση της δομής “Time+χρονικό διάστημα”, δηλαδή αν το χρονικό διάστημα είναι εβδομήντα λεπτά, η απάντηση θα αποθηκευτεί σε μορφή “Time+70” (Σχήμα 5.35). Ο λόγος ύπαρξης της δομής αυτής είναι πως οι ερωτήσεις που σχετίζονται με χρονικά διαστήματα θα απαντηθούν με βάση την τρέχων εκάστοτε ώρα. Για τον λόγο αυτό, αποθηκεύεται η χρονική πράξη που πρέπει να εκτελεστεί και όχι έτοιμη η απάντηση της ερώτησης. Απο μεριά κώδικα, στην μέθοδο onAnswer γίνεται ένας έλεγχος για το αν τα πρώτα τέσσερα γράμματα της αποθηκευμένης απάντησης, αποτελούν την λέξη “Time” (Σχήμα 5.38). Σε περίπτωση που ισχύει, σημαίνει οτι είμαστε στην συγκεκριμένη κατηγορία απάντησης, εξάγεται απο αυτήν το χρονικό διάστημα που

αναγράφεται μετά την φράση “Time+” και καλείται μία επιπρόσθετη μέθοδος ονόματι `getCorrectTimeAfterChange()`. Στην μέθοδο `getCorrectTimeAfterChange()`, χρησιμοποιείται μια μεταβλητή της κλάσης `DateTime`, η οποία μας δίνει την δυνατότητα να πάρουμε την τρέχων ώρα του συστήματος, καθώς και να εκτελέσουμε χρονικές πράξεις σε αυτήν (Σχήμα 5.36). Έτσι, προστίθεται στην τρέχων ώρα το χρονικό διάστημα που δίνεται απο την ερώτηση και έχουμε πλέον την σωστή απάντηση, η οποία έπειτα απο μία δόμηση επιστρέφεται σε μορφή `string`.

➤ **Money:** “Money”

Η τελευταία κατηγορία ερώτησης έχει να κάνει με χρηματικά ποσά και αποσπά την απάντηση, από μία διεπαφή η οποία έχει δημιουργηθεί για την αγορά υλικών. Οι απαντήσεις σε τέτοιου είδους ερωτήσεις δεν αποθηκεύονται απευθείας στην λίστα καθώς εξαρτώνται από την εκάστοτε αγορά του παίκτη. Για τον λόγο αυτό, η απάντηση αποσπάται απο διαφορετική διεπαφή, η οποία θα αναλυθεί στο επόμενο κεφάλαιο της Π.Ε.



Σχήμα 5.35: Δομή απάντησης ερωτήσεως τύπου χρόνου

```

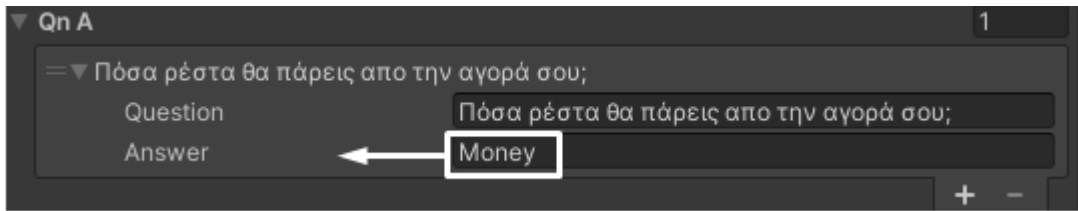
1 reference
private string getCorrectTimeAfterChange(string timeNeeded)
{
    double timeToAdd = double.Parse(timeNeeded);
    DateTime currentTime = DateTime.Now;

    DateTime correctTime = currentTime.AddMinutes(timeToAdd);
    string hour = TimeUIHandler.LeadingZero(correctTime.Hour);
    string minute = TimeUIHandler.LeadingZero(correctTime.Minute);

    string correctTimeFormatted = hour + ":" + minute;

    return correctTimeFormatted;
}
    
```

Σχήμα 5.36: Μέθοδος `getCorrectTimeAfterChange()` χρήση κλάσης `DateTime`



Σχήμα 5.37: Κατηγορία ερώτησης χρηματικού ποσού

```
public void onAnswer(Text answerHolder)
{
    string answerGiven = answerHolder.text;
    QuestionAndAnswer questionAndAnswer = QnAQueue.Peek(); //gets the first item without removing it
    string correctAnswer = questionAndAnswer.answer;

    //if the answer depends on the current time
    if (correctAnswer.Length>4 && correctAnswer.Substring(0, 4) == "Time")
    {
        //get the corrected time given the change needed. Example currentTime + 40minutes
        correctAnswer = getCorrectTimeAfterChange(correctAnswer.Substring(4));
    }
    else if (correctAnswer.Contains("Money")) //if the question has to do with the shop
    {
        //get the value of the amount of money left from shopping and then disable the ui of the shop
        decimal amountLeft = GameObject.Find("ShopPanel").GetComponent<Shop>().amountLeft;
        GameObject.Find("ShopPanel").SetActive(false);

        correctAnswer = amountLeft.ToString();
        correctAnswer = correctAnswer.Substring(0, 4);
    }
}
```

Σχήμα 5.38: Script QuestionnaireHandler μέθοδος onAnswer μέρος πρώτο

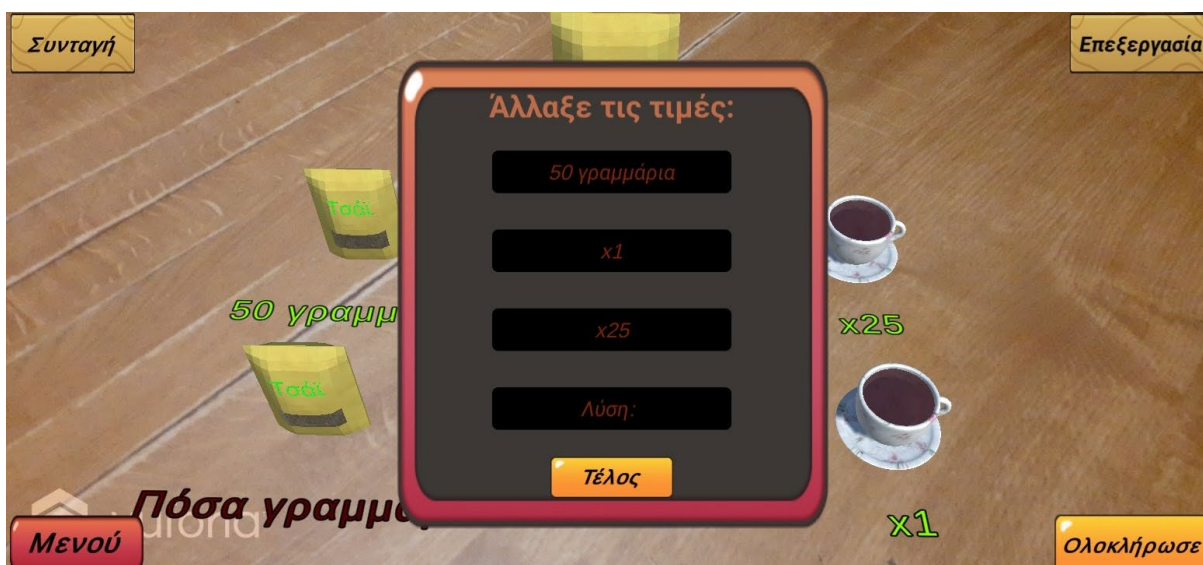
```
if(answerGiven.Equals(correctAnswer))
{
    //get the first item and remove it from the queue
    questionAndAnswer = QnAQueue.Dequeue();

    if (QnAQueue.Count == 0) //if there is no other question then show the completedTask Panel
    {
        CompletedTaskPanel.SetActive(true);
        QuestionnairePanel.SetActive(false);
    }
    else
    {
        questionAndAnswer = QnAQueue.Peek();
        QuestionnairePanel.GetComponentInChildren<TextMeshProUGUI>().SetText(questionAndAnswer.question);
        QuestionnairePanel.GetComponentInChildren<InputField>().text = "";
        QuestionnairePanel.SetActive(false);
    }
}
else
{
    FailedTaskPanel.SetActive(true);
    QuestionnairePanel.SetActive(false);
}
}
```

Σχήμα 5.39: Script QuestionnaireHandler μέθοδος onAnswer μέρος δεύτερο

5.4.7 ChangeValues

Το παιχνίδι πέρα από συνταγές προσφέρει και μαθηματικές ασκήσεις οι οποίες έχουν διαμορφωθεί γύρω από ορισμένες συνταγές του παιχνιδιού. Μία επιλογή η οποία δίνεται σε αυτές τις ασκήσεις, είναι η δυνατότητα τροποποίησης της εκάστοτε άσκησης, μέσω της αλλαγής των μεταβλητών καθώς και των απαντήσεων της. Η συγκεκριμένη επιλογή, στοχεύει στην καλύτερη χρήση του παιχνιδιού στο πλαίσιο της μάθησης, καθώς ένας καθηγητής έχει πλέον την δυνατότητα να αλλάζει τις μεταβλητές των ασκήσεων, διατηρώντας τις μεθόδους του παιχνιδιού, όσες φορές θέλει. Η διαδικασία αυτή γίνεται μέσω της διεπαφής ChangeValues panel, καθώς και του script ChangeValues. Στο πάνω δεξί μέρος της οθόνης έχει τοποθετηθεί ένα κουμπί στο οποίο αναγράφεται “Επεξεργασία”, με το πάτημα του κουμπιού εμφανίζεται και η αντίστοιχη διεπαφή επεξεργασίας άσκησης (Σχήμα 5.40).



Σχήμα 5.40: Εμφάνιση διεπαφής αλλαγής μεταβλητών σε άσκηση-συνταγή

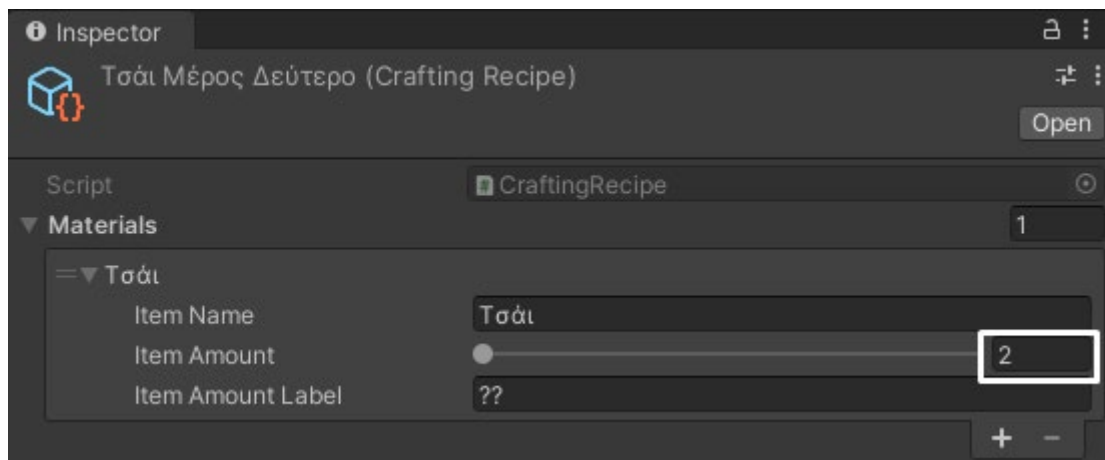
Η λειτουργία της διεπαφής και γενικότερα της αλλαγής των μεταβλητών, πραγματοποιείται μέσω του script ChangeValues. Το script αποτελείται από έξι μεθόδους, όμως ανάλογα το είδος της άσκησης χρησιμοποιούνται οι πέντε. Αυτό συμβαίνει διότι οι ασκήσεις του παιχνιδιού έχουν χωριστεί σε δύο κατηγορίες:

➤ **Άσκηση-Συνταγής:**

Οι ασκήσεις αυτού του τύπου χαρακτηρίζονται ως ασκήσεις τύπου συνταγής λόγω της δομής τους. Συγκεκριμένα, οι ασκήσεις αυτές βασίζονται στην κλασική δομή του παιχνιδιού. Δηλαδή, υπάρχει κάποιο υλικό το οποίο ο παίκτης πρέπει να σύρει σε ένα δοχείο και να επιλέξει την σωστή ποσότητα. Έτσι και εδώ, ο παίκτης έχει στην διάθεση του ένα υλικό και κάποιον άγνωστο αριθμό τον οποίο θα πρέπει να υπολογίσει. Η απάντηση του θα δοθεί σέρνοντας το υλικό στο κατάλληλο σημείο και δίνοντας, με βάση της διεπαφής Numpad, την σωστή ποσότητα. Επίσης, άλλο ένα χαρακτηριστικό αυτών των ασκήσεων είναι ο τρόπος αποθήκευσης των απαντήσεων τους. Συγκεκριμένα, οι ασκήσεις χρησιμοποιούν αντικείμενα τύπου CraftingRecipe, τα οποία έχουν δημιουργηθεί ώστε να έχουν μέσα τους μόνο ένα υλικό, στο οποίο αποθηκεύεται η σωστή απάντηση σε μορφή ποσότητας (Σχήμα 5.41). Οι ασκήσεις αυτές θα γίνουν ακόμα πιο κατανοητές στο επόμενο κεφάλαιο.

➤ **Άσκηση-Ερωτήσεως:**

Η δομή των ασκήσεων τύπου ερώτησης είναι απλή, καθώς σε αυτές τις περιπτώσεις ο παίκτης αρκεί απλά να απαντήσει σε κάποια ερώτηση η οποία θα του δοθεί στο τέλος της σκηνής. Για τον λόγο αυτό, οι λύσεις των ασκήσεων αποθηκεύονται στην λίστα των ερωτήσεων-απαντήσεων της εκάστοτε σκηνής.



Σχήμα 5.41: Απάντηση άσκησης-συνταγής σε μορφή ποσότητας υλικού

Ανάλογα λοιπόν από την κατηγορία των ασκήσεων αλλά και όχι μόνο, αλλάζει η συμπεριφορά των παρακάτω μεθόδων:

1. **Start():** Η μέθοδος Start() καλείται κατά το άνοιγμα της διεπαφής και ελέγχει για το είδος της άσκησης της οποίας θα αλλάξουν οι μεταβλητές. Μετά τον έλεγχο, σε περίπτωση που πρόκειται για άσκηση τύπου ερώτησης, λαμβάνεται η λίστα των ερωτήσεων και αποθηκεύονται ως αντίγραφα οι απαντήσεις τους σε μία λίστα τύπου string. Στην περίπτωση όμως που πρόκειται για τον τύπο συνταγής, κάτι το οποίο αναγνωρίζεται από το γεγονός ότι η συνταγή της σκηνής εμπεριέχει κάποιο υλικό, ορίζεται μία μεταβλητή isRecipe τύπου boolean ως αληθής και αποθηκεύεται η αρχική απάντηση της άσκησης. Η μεταβλητή αυτή παίζει τον ρόλο σημαίας στις παρακάτω μεθόδους, καθώς πρέπει να γνωρίζουμε για το αν η άσκηση αφορά συνταγή ή ερώτηση. Τέλος, ορίζεται άλλη μία boolean μεταβλητή (variablesChanged) ως ορθή, η οποία συμβολίζει πως πρόκειται να υπάρξουν αλλαγές στην άσκηση, και έπειτα καλείται η επόμενη μέθοδος getValuesForChange() (Σχήμα 5.42).
2. **getValuesForChange():** Ο ρόλος της μεθόδου getValuesForChange() είναι να βρίσκει τον αριθμό των μεταβλητών, καθώς και τις λύσεις της άσκησης που μπορούν να αλλαχθούν, να τις αποθηκεύει και να δημιουργεί τον αντίστοιχο αριθμό αντικειμένων πληκτρολόγησης κειμένου (Input Fields), στα οποία θα πληκτρολογήσει ο παίκτης τις αλλαγές του. Συγκεκριμένα κάθε μεταβλητή η οποία έχει την δυνατότητα αλλαγής, φέρει ένα tag με όνομα "ChangeableValue" (Σχήμα 5.43). Το tag αυτό χρησιμοποιείται από μεριά κώδικα ώστε να βρεθούν όλα τα αντικείμενα τα οποία το έχουν και να αποθηκευτούν σε ένα array τύπου GameObject[] με όνομα valuesToChange. Έπειτα, για κάθε αντικείμενο το οποίο βρέθηκε δημιουργείται δυναμικά μέσω της εντολής Instantiate() και το αντίστοιχο πεδίο inputField. Η εντολή Instantiate δέχεται σαν μεταβλητή κάποιο ήδη υπάρχων αντικείμενο, από το οποίο θα δημιουργήσει αντίγραφο. Για τον λόγο αυτόν, έχει ήδη δημιουργηθεί ένα Prefab αντικείμενο inputField και έχει αποθηκευτεί στον φάκελο των Prefabs (Σχήμα 5.44), το οποίο έπειτα χρησιμοποιείται από την εντολή αυτή για την δημιουργία των αντιγράφων. Με αυτόν τον τρόπο, δημιουργούνται δυναμικά όσα πεδία πληκτρολόγησης χρειάζονται ανάλογα με τον αριθμό των μεταβλητών και λύσεων της άσκησης. Αφού έχουν δημιουργηθεί τα πεδία αυτά, τους προστίθενται σαν προϋπάρχων κείμενο οι τιμές τις οποίες έχουν οι μεταβλητές πριν πραγματοποιηθεί κάποια αλλαγή σε αυτές. Έπειτα, υπάρχει και σε αυτήν την μέθοδο ο έλεγχος του είδους της άσκησης, όπου ανάλογα το είδος της δημιουργούνται επιπλέον πεδία, τα οποία αντιστοιχούν στις απαντήσεις και φέρουν κάποιο κείμενο το οποίο τα αναδεικνύει,

όπως φαίνεται και στο τελευταίο πεδίο στο σχήμα 5.40. Πέρα από τον έλεγχο για το είδος της άσκησης, αν πρόκειται για άσκηση-ερώτηση γίνεται ένας επιπλέον έλεγχος για την κατηγορία της απάντησης. Αυτό συμβαίνει διότι αν επιτραπεί επεξεργασία άσκησης της κατηγορίας “Time”, θα πρέπει η αλλαγή του παίκτη να αποθηκευτεί με την αντίστοιχη δομή. Έτσι, σε περίπτωση που ανήκει σε αυτήν την κατηγορία ορίζεται ως αληθής μία μεταβλητή boolean (timeRelated), την οποία θα χρειαστούμε αργότερα.

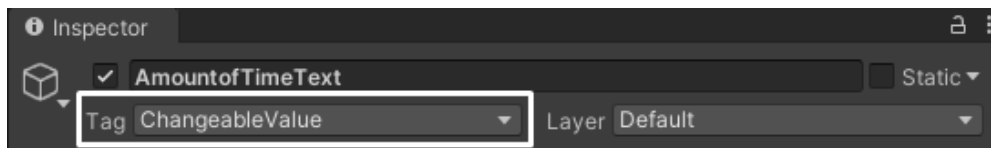
3. **setChangedValuesTypeQuestions():** Ρόλος αυτής της μεθόδου είναι να ορίζει τις τιμές που έδωσε ο παίκτης ως τις νέες μεταβλητές και απαντήσεις, σε άσκηση τύπου ερωτήσεων την στιγμή που πιεστεί το κουμπί “Τέλος” της διεπαφής (Σχήμα 5.40). Αρχικά, η μέθοδος βρίσκει και αποθηκεύει όλα τα inputFields τα οποία δημιουργήθηκαν μέσω της προηγούμενης μεθόδου, χρησιμοποιώντας επίσης ένα χαρακτηριστικό tag που τους έχει προστεθεί κατά την δημιουργία τους, το tag “ChangedValue” (Σχήμα 5.44). Έπειτα, γίνεται έλεγχος για το αν η ερώτηση έχει κατηγορία απάντησης “Time”, με βάση την τιμή που πήρε η μεταβλητή timeRelated της προηγούμενης μεθόδου. Από την στιγμή που είναι αληθής, θα γίνει ένας ακόμη έλεγχος για το αν ο παίκτης άλλαξε την τιμή της λύσης της άσκησης και αφού υπήρξε αλλαγή, χωρίζεται το κείμενο που πληκτρολόγησε ο παίκτης, κρατώντας από αυτό, το μέρος στο οποίο αναγράφεται το χρονικό διάστημα. Αυτό πραγματοποιείται μέσω της εντολής Split. Η εντολή αυτή μας δίνει την δυνατότητα να χωρίζουμε ένα κείμενο σε επιμέρους στοιχεία με βάση κάποιο διαχωριστικό χαρακτήρα. Στην συγκεκριμένη περίπτωση, ο διαχωρισμός γίνεται με βάση τα κενά και κρατάμε το πρώτο μέρος της πρότασης. Δηλαδή, αν η φράση που γράψει ο παίκτης για την αλλαγή της μεταβλητής χρόνου είναι “60 λεπτά”, θα κρατήσουμε μόνο το “60”. Μαζί με τον διαχωρισμό της φράσης προσθέτουμε και μπροστά από αυτήν την φράση “Time+”, με αποτέλεσμα η νέα τιμή που έδωσε ο παίκτης να έχει πάρει την πλέον έγκυρη μορφή απάντησης με βάση την κατηγορία ερωτήσεων χρόνου, η οποία και αποθηκεύεται ως η νέα απάντηση στην ερώτηση. Στην περίπτωση όμως που η ερώτηση δεν είναι της κατηγορίας χρόνου, η νέα απάντηση της ερώτησης αποθηκεύεται απευθείας στην λίστα των ερωτήσεων-απαντήσεων. Τέλος, καλείται η μέθοδος με όνομα changeUiText() (Σχήμα 5.47).
4. **setChangedValuesTypeRecipe():** Η μέθοδος υπεύθυνη για τον χειρισμό των απαντήσεων στις ασκήσεις τύπου συνταγής, είναι η setChangedValuesTypeRecipe(). Η λογική της μεθόδου είναι παραπλήσια με την αντίστοιχη μέθοδο που χρησιμοποιείται στις ασκήσεις τύπου ερωτήσεων. Η διαφορά της μεθόδου είναι στον χειρισμό της απάντησης, καθώς η νέα απάντηση πρέπει να περάσει ως υλικό της αντίστοιχης συνταγής και όχι στις απαντήσεις της λίστας των ερωτήσεων-απαντήσεων, όπως στην παραπάνω μέθοδο. Επίσης, σε αυτήν την κατηγορία άσκησης υπάρχει μόνο ένα είδος απάντησης οπότε δεν χρειάζονται περαιτέρω έλεγχοι. Τέλος, και αυτή η μέθοδος θα καλέσει την μέθοδο changeUiText() (Σχήμα 5.48).
5. **changeUiText():** Η μέθοδος changeUiText() είναι υπεύθυνη για την ενημέρωση της εμφάνισης των μεταβλητών της άσκησης, έπειτα των αλλαγών του παίκτη. Πρόκειται για μία απλή διαδικασία, καθώς οι απαντήσεις, οι νέες μεταβλητές και τα αντικείμενα διεπαφής τα οποία εμφανίζουν αυτές τις μεταβλητές, έχουν ήδη βρεθεί και αποθηκευτεί μέσω των προηγούμενων μεθόδων. Έτσι, με την χρήση ενός επαναληπτικού βήματος, καθώς και ενός αρχικού ελέγχου για τον αριθμό των απαντήσεων της άσκησης, διεκπεραιώνεται η λειτουργία αυτή. Αξίζει να σημειωθεί, πως ο έλεγχος του αριθμού των απαντήσεων είναι απαραίτητος καθώς πρέπει να γνωρίζουμε πόσες τιμές δεν θα λάβουμε υπόψη από την λίστα στην οποία έχουν αποθηκευτεί (changedValues), οι απαντήσεις δεν θα εμφανίζονται προφανώς στην διεπαφή της άσκησης. Επίσης, και σε αυτήν την μέθοδο γίνεται ένας έλεγχος σε περίπτωση που δεν υπάρξουν αλλαγές σε κάποια μεταβλητή, το αντίστοιχο αντικείμενο διεπαφής να κρατάει την αρχική του τιμή .
6. **resetChangedValues():** Η τελευταία και πολύ σημαντική μέθοδος που χρησιμοποιείται είναι η resetChangedValues(). Ο ρόλος της μεθόδου είναι απλός αλλά πολύ σημαντικός, καθώς είναι υπεύθυνη για την επιστροφή των μεταβλητών και λύσεων που άλλαξε ο παίκτης στις αρχικές τους τιμές. Αρχικά, γίνεται ένας έλεγχος για την boolean μεταβλητή

variablesChanged, η οποία κατά την δημιουργία της έχει ψευδείς τιμή (false). Η μεταβλητή αυτή, λειτουργεί σαν σημαία για το αν κατά την διάρκεια της σκηνής έχει ανοίξει η διεπαφή αλλαγής τιμών. Αυτό συμβαίνει μέσω της παραπάνω μεθόδου Start(), η οποία αν τρέξει θα αλλάξει την τιμή της μεταβλητής σε αληθή (true). Μετά τον έλεγχο αυτό, γίνεται ένας επιπλέον έλεγχος για το είδος της άσκησης και ανάλογα το είδος επιστρέφονται και οι αντίστοιχες λύσεις στην αρχική τιμή τους, η οποία είχε προηγουμένως αποθηκευτεί. Η μέθοδος resetChangedValues() καλείται σαν επιπρόσθετη μέθοδος όταν πιέσει ο παίκτης το κουμπί ολοκλήρωσε (Σχήμα 5.49), καθώς και στις μεθόδους αλλαγής τιμών setChangedValuesTypeQuestions() και setChangedValuesTypeRecipe(), στην περίπτωση που δεν εντοπιστούν αλλαγές από τον παίκτη.

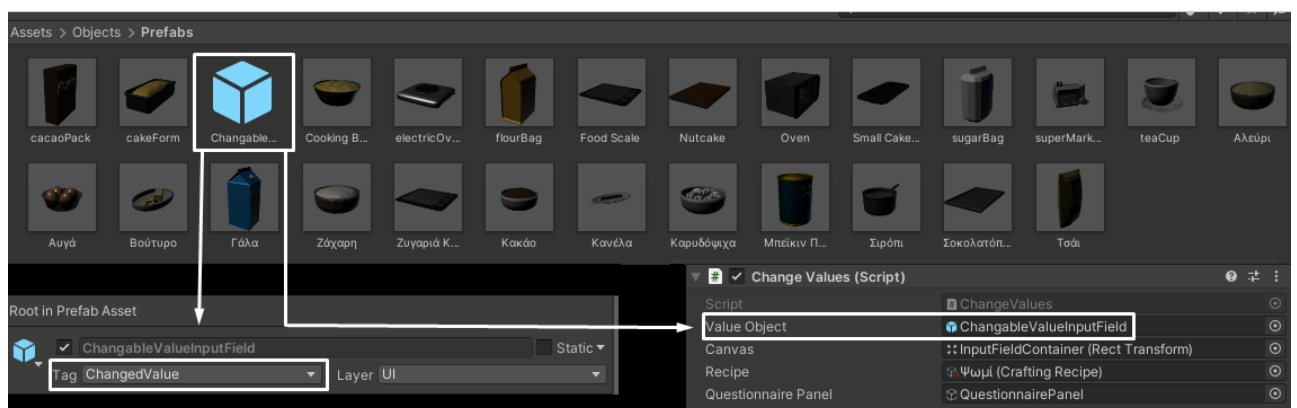
```

Unity Message | 0 references
private void Start()
{
    if (recipe.materials.Count != 0)
    {
        isRecipe = true;
        //set the original amount of the recipe so it is reset once the problem is solved with the changed values
        originalAmountOfRecipe = recipe.materials[0].itemAmount;
    }
    else
    {
        //save original list of questions and answers
        QnA = QuestionnairePanel.GetComponent<QuestionnaireHandler>().QnA;
        foreach (QuestionAndAnswer qna in QnA)
        {
            answers.Add(qna.answer);
        }
    }
    variablesChanged = true;
    getValuesForChange();
}
    
```

Σχήμα 5.42: Script ChangeValues μέθοδος Start()



Σχήμα 5.43: Tag “ChangeableValue” κειμένου στο οποίο αναγράφεται μεταβλητή άσκησης



Σχήμα 5.44: Prefab αντικείμενο Input Field, χαρακτηριστικό tag “ChangedValue” και προσθήκη ως εξωτερική μεταβλητή στο script ChangeValues

```

1 reference
public void getValuesForChange()
{
    valuesToChange = GameObject.FindGameObjectsWithTag("ChangeableValue");
    //for every changeable value create its representative input field
    for(int i = 0; i <= valuesToChange.Length-1; i++)
    {
        GameObject value = Instantiate(valueObject);
        value.transform.SetParent(canvas,false);
        value.GetComponentInChildren<Text>().text = valuesToChange[i].GetComponent<TMP_Text>().text;
        value.SetActive(true);
    }

    if (isRecipe)
    {
        GameObject answer = Instantiate(valueObject);
        answer.transform.SetParent(canvas, false);
        answer.GetComponentInChildren<Text>().text = "Λύση: ";
    }
}

```

Σχήμα 5.45: Script ChangeValues μέθοδος getValuesForChange() μέρος πρώτο

```

else
{
    if (answers[0].Length > 4 && answers[0].Substring(0, 4) == "Time")
    {
        timeRelated = true;
    }
    else
    {
        //set the containers based on the answers saved on the list of questions and answers
        List<string> answerLabels = new List<string> { "Περίμετρος", "Εμβαδόν", "Όγκος με ύψος 1.5" };
        int i = 0;
        foreach (string answer in answers)
        {
            GameObject answerContainer = Instantiate(valueObject);
            answerContainer.transform.SetParent(canvas, false);
            answerContainer.GetComponentInChildren<Text>().text = answerLabels[i].ToString() + ": ";
            i++;
        }
    }
}
}

```

Σχήμα 5.46: Script ChangeValues μέθοδος getValuesForChange() μέρος δεύτερο

```

0 references
public void setChangedValuesTypeQuestions()
{
    GameObject[] changedValues = GameObject.FindGameObjectsWithTag("ChangedValue");

    questionnaireHandler = QuestionnairePanel.GetComponent<QuestionnaireHandler>();
    string changedAnswer;

    if (timeRelated)
    {
        changedAnswer = changedValues[0].GetComponentInChildren<InputField>().text;

        if (changedAnswer == "")
        {
            resetChangedValues();
        }
        else
        {
            changedAnswer = "Time+" + changedValues[0].GetComponentInChildren<InputField>().text.Split(' ')[0];
            questionnaireHandler.QnA[0] = new QuestionAndAnswer(QnA[0].question, changedAnswer);
        }
    }
    else
    {
        //if it is not time related then changable questions exist only in the last scene that holds multiple questions and answers
        for (int i = 0; i < QnA.Count; i++)
        {
            changedAnswer = changedValues[i + 2].GetComponentInChildren<InputField>().text;

            if (changedAnswer == "")
            {
                resetChangedValues();
            }
            else
            {
                questionnaireHandler.QnA[i] = new QuestionAndAnswer(QnA[i].question, changedAnswer);
            }
        }
    }
    changeUiText(changedValues);
}

```

Σχήμα 5.47: Script ChangeValues μέθοδος setChangedValuesTypeQuestions()

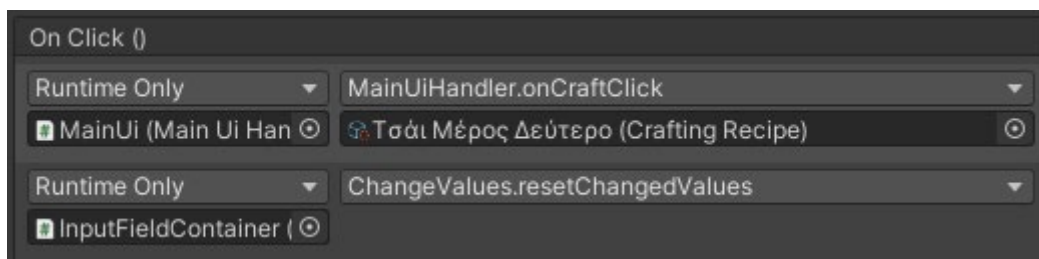
```

0 references
public void setChangedValuesTypeRecipe()
{
    GameObject[] changedValues = GameObject.FindGameObjectsWithTag("ChangedValue");
    changeUiText(changedValues);

    string newValue = changedValues[changedValues.Length - 1].GetComponent<InputField>().text;
    if (newValue == "")
    {
        resetChangedValues();
    }
    else
    {
        //update recipe with the new value
        recipe.materials[0] = new Items(recipe.materials[0].itemName, float.Parse(newValue), recipe.materials[0].itemAmountLabel);
    }
}

```

Σχήμα 5.48: Script ChangeValues μέθοδος setChangedValuesTypeRecipe()



Σχήμα 5.49: Πρόσθεση μεθόδου resetChangedValues στο κουμπί “Ολοκλήρωσε”

5.5 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκαν οι βασικοί μηχανισμοί του παιχνιδιού, ο τρόπος με τον οποίο αλληλεπιδρούν καθώς και η λογική με την οποία διεκπεραιώνονται. Προχωρώντας στο επόμενο κεφάλαιο, γίνεται ακόμα πιο αντιληπτός ο τρόπος με τον οποίο έχει χτιστεί το περιεχόμενο του παιχνιδιού γύρω από αυτούς τους μηχανισμούς.

Κεφάλαιο 6ο: Ανάλυση Παιχνιδιού

6.1 Εισαγωγή

Κύριος στόχος του παιχνιδιού, είναι η βοήθεια μαθητών Στ' δημοτικού στην κατανόηση του μαθήματος των μαθηματικών. Για να επιτευχθεί αυτός ο στόχος, έχουν δημιουργηθεί πολλαπλά σενάρια παιχνιδιού, με τέτοιο τρόπο ώστε η διαδικασία αυτή πέρα από εκπαιδευτική να είναι και διασκεδαστική. Σε αυτό το κεφάλαιο θα αναλύσουμε τα μέρη του παιχνιδιού με βάση τα κεφάλαια του βιβλίου των μαθηματικών τα οποία καλύπτουν.

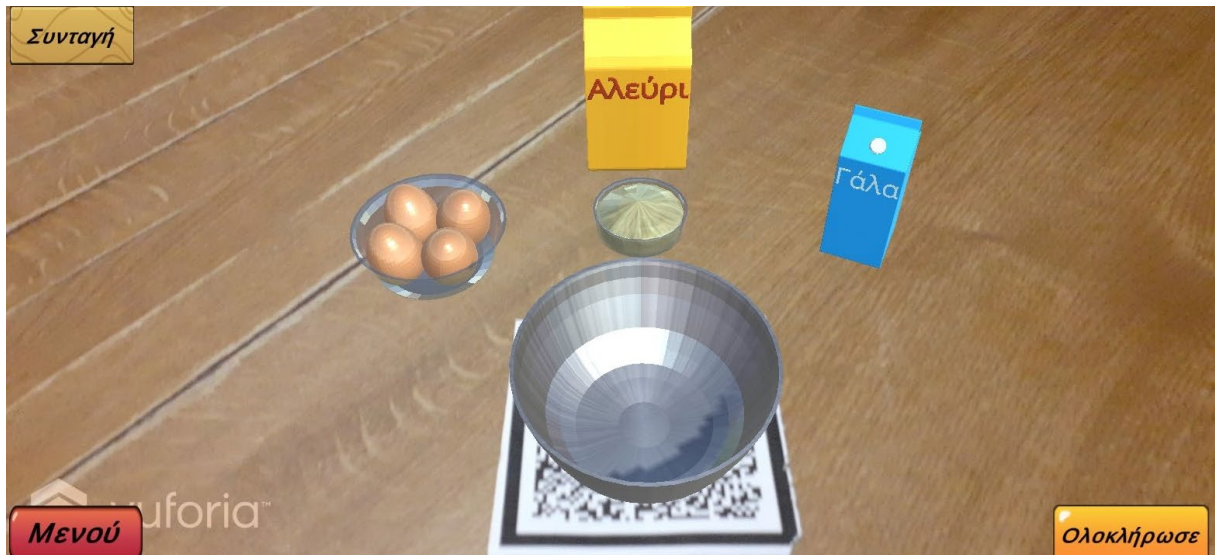
6.2 Ακέραιοι Αριθμοί

Το πρώτο κεφάλαιο των μαθηματικών αφορά τους ακέραιους αριθμούς. Το παιχνίδι ξεκινάει με μία απλή συνταγή για την δημιουργία κρεπών, η οποία απαιτεί από τον παίκτη να χρησιμοποιήσει τις σωστές ποσότητες υλικών. Η συνταγή λειτουργεί ως εισαγωγή στους μηχανισμούς του παιχνιδιού και δίνεται με χρήση ακέραιων αριθμών. Στο τέλος της ζητείται από τον παίκτη να υπολογίσει το συνολικό βάρος του μείγματος, με βάση την ποσότητα των υλικών που χρησιμοποίησε.



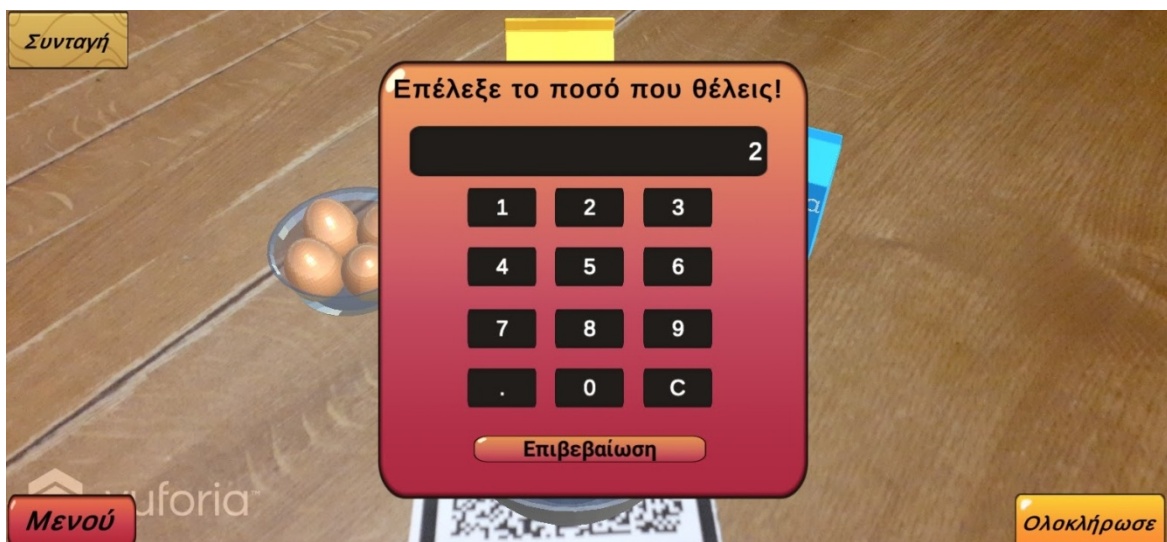
Σχήμα 6.1: Εμφάνιση συνταγής κρεπών με μορφή ακεραίων αριθμών

Για την ολοκλήρωση του παιχνιδιού, ο παίκτης πρέπει να σύρει τα υλικά με όποια σειρά θέλει αρκεί να ορίσει τις σωστές ποσότητες. Η συνταγή εμφανίζεται κατά την αρχή της σκηνής και υπάρχει η επιλογή να ξαναεμφανιστεί απλά με την πίεση του κουμπιού στο πάνω αριστερό μέρος της οθόνης (Σχήμα 6.1). Πιέζοντας το κουμπί επιστροφή στην διεπαφή εμφάνισης συνταγής, ο παίκτης είναι ελεύθερος να αλληλεπιδράσει με τα εικονικά αντικείμενα (Σχήμα 6.2).

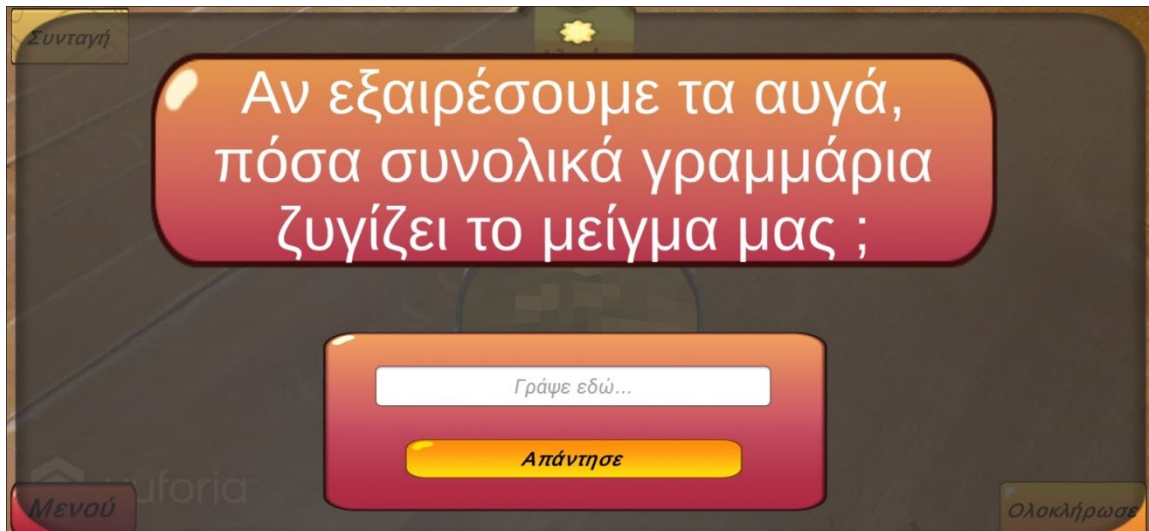


Σχήμα 6.2: Εμφάνιση σκηνής μετά απο την επιστροφή από την διεπαφή συνταγής

Την στιγμή κατά την οποία ο παίκτης σύρει κάποιο υλικό στο μπολ μίξης, θα εμφανιστεί η αντίστοιχη διεπαφή επιλογής ποσότητας (Σχήμα 6.3). Όταν ο παίκτης έχει τελειώσει με το κομμάτι της συνταγής και εφόσον έχει προσθέσει τις σωστές ποσότητες των υλικών, θα πρέπει να απαντήσει μία ερώτηση πριν ολοκληρώσει την σκηνή του παιχνιδιού (Σχήμα 6.4). Η ερώτηση αφορά τους ακέραιους αριθμούς και συγκεκριμένα την πρόσθεση των ποσοτήτων που χρησιμοποίησε στην συνταγή. Η ερώτηση εμφανίζεται μέσω της διεπαφής QuestionnairePanel και λειτουργεί με βάση το script QuestionnaireHandler. Επίσης, η διεπαφή επιλογής ποσότητας υλικού λειτουργεί με βάση το script NumpadHandler, ενώ η εμφάνιση όλων των παραπάνω διεπαφών πραγματοποιείται μέσω του MainUiHandler script. Όλα τα scripts που αναφέρθηκαν έχουν αναλυθεί στο προηγούμενο κεφάλαιο, καθώς αποτελούν κοινές και βασικές λειτουργίες σκηνών.



Σχήμα 6.3: Διεπαφή επιλογής ποσότητας υλικού



Σχήμα 6.4: Διεπαφή ερώτησης-απάντησης Questionnaire

6.2.1 CookingBowl & BowlAnimation

Όσο ο παίκτης προσθέτει υλικά στο μπολ, παρατηρείται ένα μικρό animation, το οποίο προσομοιάζει την προσθετική αύξηση ενός μίγματος. Το animation αυτό λειτουργεί με βάση το script BowlAnimation. Το script αυτό έχει προστεθεί στο αντικείμενο Cooking Bowl και δέχεται σαν εξωτερική μεταβλητή ένα μέρος του αντικειμένου. Συγκεκριμένα κατά την δημιουργία του τρισδιάστατου αντικειμένου Cooking Bow, έχει δημιουργηθεί σαν μέρος του και το μίγμα που περιέχει. Το script χρησιμοποιεί τέσσερις μεθόδους, την Start(), την OnTriggerEnter(), την setOriginalPositionOfBowContent() και την animateBowl(). Η πρώτη μέθοδος τρέχει κατά την αρχή της σκηνής μετρώντας τον αριθμό των υλικών που υπάρχουν σε αυτή και καλεί την μέθοδο setOriginalPositionOfBowContent(). Η μέθοδος αυτή, βρίσκει την αρχική κλίμακα και θέση του τρισδιάστατου αντικειμένου μίγματος και θέτει και τα δύο μηδενικά. Αυτό έχει σαν αποτέλεσμα, το μίγμα να μην εμφανίζεται στην αρχή της σκηνής, καθώς ο παίκτης δεν έχει προσθέσει ακόμα υλικά στο μπολ. Έπειτα, η μέθοδος υπολογίζει τον προσθετικό αριθμό με τον οποίο θα αυξάνει την κλίμακα και θέση του μίγματος κατά την πρόσθεση νέου υλικού, μέχρις ότου αυτό να φτάσει στην αρχική του μορφή. Ο υπολογισμός γίνεται διαιρώντας την αρχική τιμή κλίμακας και θέσης αντίστοιχα, με τον αριθμό των υλικών που πρόκειται να χρησιμοποιηθούν. Έτσι, με την μέθοδο OnTriggerEnter(), η οποία θα κληθεί όταν ο χρήστης ακουμπήσει με το υλικό το μπολ, και με την σειρά της θα καλέσει την animateBowl(), προστίθενται οι αριθμοί αυτοί στις τρέχων τιμές κλίμακας και θέσης επαναληπτικά, μέχρις ότου να χρησιμοποιηθούν όλα τα υλικά και το μίγμα να έχει την αρχική του εμφάνιση (Σχήμα 6.5).



Σχήμα 6.5: Βηματική αλλαγή εμφάνισης μίγματος σε συνταγή τριών υλικών

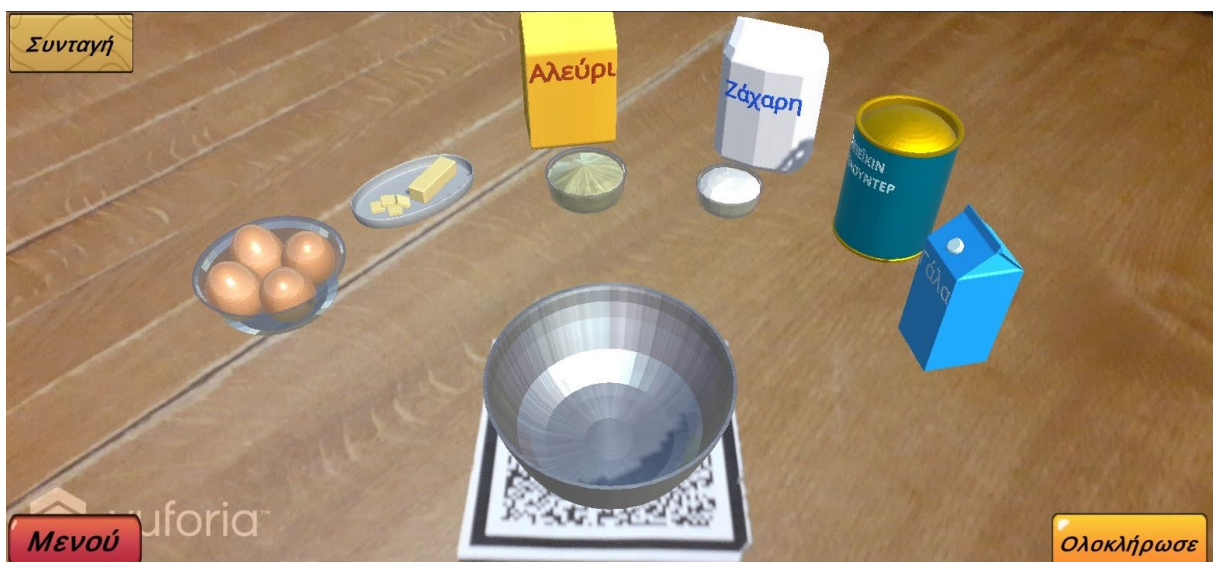
6.3 Κλασματικοί Αριθμοί

Προχωρώντας στο επόμενο κεφάλαιο των μαθηματικών, μπαίνουμε στους κλασματικούς αριθμούς. Η χρήση κλασματικών αριθμών χρησιμοποιήθηκε σε αυτήν την σκηνή, ως μέρος της συνταγής. Συγκεκριμένα, η σκηνή αφορά μία συνταγή για τηγανίτες (pancakes) και ορισμένα από τα υλικά της συνταγής δίνονται σε μορφή κλασμάτων του κιλού (Σχήμα 6.6).



Σχήμα 6.6: Διεπαφή συνταγής με χρήση κλασματικών αριθμών

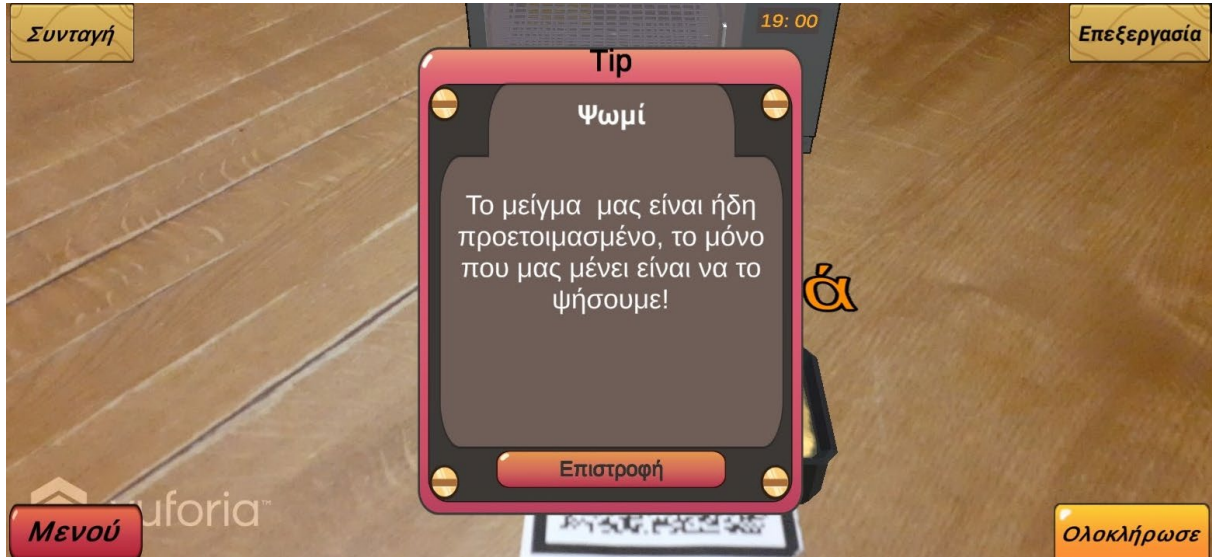
Η λογική του παιχνιδιού παραμένει ίδια με την προηγούμενη σκηνή, δηλαδή ο παίκτης θα πρέπει να χρησιμοποιήσει τις σωστές ποσότητες των υλικών με βάση την συνταγή που του δίνεται. Η βασική διαφορά αυτής της σκηνής, είναι πως εδώ το μαθηματικό μέρος του παιχνιδιού έχει συγχωνευτεί άμεσα με αυτό. Γι' αυτό τον λόγο δεν υπάρχει κάποια ερώτηση στο τέλος της συνταγής, αρκεί μόνο να ολοκληρωθεί σωστά.



Σχήμα 6.7: Εμφάνιση σκηνής παιχνιδιού κλασματικών αριθμών

6.4 Συμμιγείς Αριθμοί & Μετρήσεις Χρόνου

Οι συμμιγείς αριθμοί αποτελούν ακέραιους αριθμούς οι οποίοι δηλώνουν διαφορετικές μονάδες τάξης. Στο σύνολο του παιχνιδιού χρησιμοποιούνται συμμιγείς αριθμοί, καθώς όλες οι συνταγές δίνονται με χρήση κάποιας μονάδας βάρους. Στο συγκεκριμένο σημείο γίνεται μία διαφορετική χρήση των συμμιγών αριθμών, συνδυαστικά με το κεφάλαιο των μαθηματικών που αφορά τις μετρήσεις και συγκεκριμένα τις μετρήσεις του χρόνου. Η σκηνή αρχίζει σε αντίθεση με τις προηγούμενες, με ένα συμβουλευτικό μήνυμα αντί της συνταγής (Σχήμα 6.8).



Σχήμα 6.8: Εμφάνιση διεπαφής συμβουλευτικού μηνύματος στην σκηνή μέτρησης χρόνου

Δίνεται στον παίκτη το μείγμα ήδη προετοιμασμένο και ζητείται από αυτόν να υπολογίσει σωστά, με βάση το χρονικό διάστημα που χρειάζεται το ψωμί, το πότε αυτό θα είναι έτοιμο. Πρόκειται δηλαδή, για μία άσκηση μέτρησης χρόνου με βάση την τρέχων ώρα, καθώς και το χρονικό διάστημα που δίνεται.



Σχήμα 6.9: Εμφάνιση σκηνής παιχνιδιού μέτρησης χρόνου

6.4.1 TimeUiHandler

Ένα script το οποίο παίζει σημαντικό ρόλο σε αυτήν την σκηνή είναι το TimeUiHandler. Η λειτουργία αυτού του script, αφορά την σωστή δόμηση της ώρας μέσω της μεθόδου LeadingZero, καθώς και την εμφάνιση της τρέχων ώρας στην οθόνη του φούρνου. Το script χρησιμοποιεί τρεις μεθόδους, την Awake(), την Update() και την LeadingZero. Με τον συνδυασμό των δύο πρώτων μεθόδων, το script παίρνει το text στοιχείο της διεπαφής το οποίο θα εμφανίζει την ώρα και με την χρήση της κλάσης DateTime εμφανίζεται σωστά η τρέχων ώρα του συστήματος (Σχήμα 6.10). Η μέθοδος LeadingZero, πρόκειται για μία απλή μέθοδο η οποία δημιουργήθηκε για την καλύτερη δόμηση της ώρας. Συγκεκριμένα, ορίζει την εμφάνιση της ώρας και των λεπτών να είναι πάντα σε μορφή δύο ψηφίων προσθέτοντας μηδενικά όπου χρειαστεί (Σχήμα 6.11). Δηλαδή, αν η ώρα είναι δώδεκα και πέντε (12:5), η εμφάνιση της θα είναι σε μορφή 12:05 ή αν η ώρα είναι μία και τέταρτο (1:15) η ώρα θα εμφανίζεται ως 01:15. Η μέθοδος αυτή χρησιμοποιείται επίσης από το script QuestionnaireHandler στην κατηγορία ερώτησης “Time” και συγκεκριμένα στην μέθοδο getCorrectTimeAfterChange, για τον ίδιο σκοπό.

```

public class TimeUiHandler : MonoBehaviour
{
    private Text textOfClock;

    Unity Message | 0 references
    private void Awake()
    {
        textOfClock = GetComponentInChildren<Text>();
    }

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        DateTime time = DateTime.Now;
        string hour = LeadingZero(time.Hour);
        string minute = LeadingZero(time.Minute);

        textOfClock.text = hour + ": " + minute;
    }
}

```

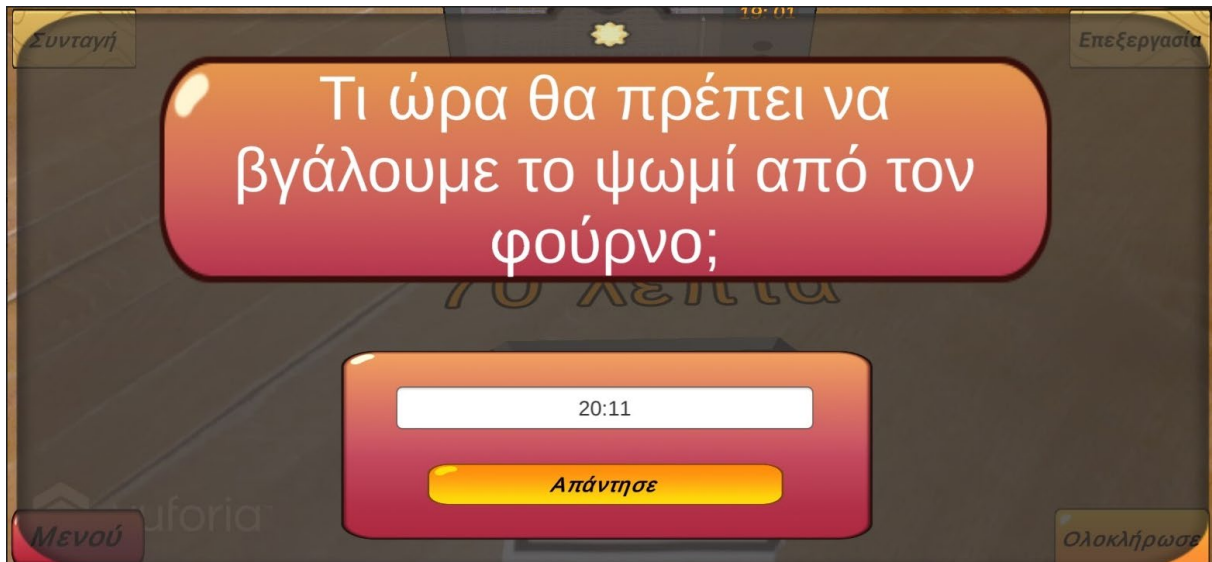
Σχήμα 6.10: Script TimeUiHandler μέθοδοι Awake() & Update()

```

4 references
public static string LeadingZero(int time)
{
    //this way the time will always have a length of 2 digits. Example: 1:30 => 01:30
    return time.ToString().PadLeft(2, '0');
}

```

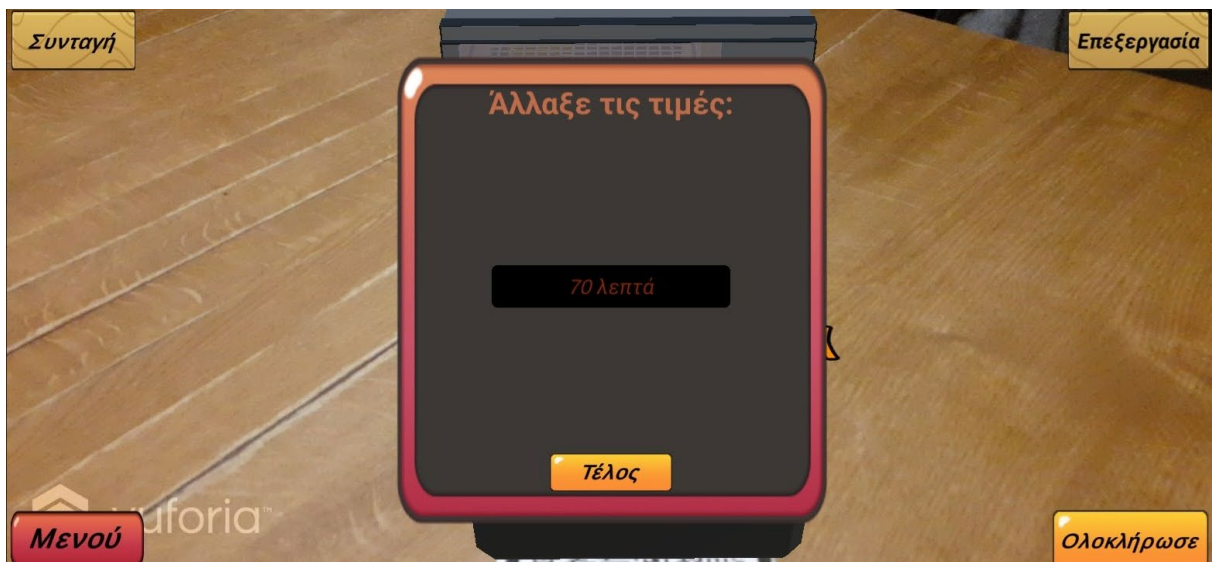
Σχήμα 6.11: Script TimeUiHandler μέθοδος δόμησης ώρας LeadingZero()



Σχήμα 6.12: Απάντηση ερώτησης χρόνου μέσω της διεπαφής ερωτήσεων-απαντήσεων

6.4.2 Επεξεργασία τιμών άσκησης τύπου ερώτησης

Στην παρούσα σκηνή δίνεται στον παίκτη η επιλογή επεξεργασίας του προβλήματος. Συγκεκριμένα ο παίκτης έχει την δυνατότητα να αλλάξει το ζητούμενο χρονικό διάστημα της συνταγής προκειμένου να αλλάξει και το τελικό ζητούμενο. Η επιλογή αυτή έχει στόχο την χρήση από καθηγητές των τάξεων ΣΤ' δημοτικού, προκειμένου να έχουν την δυνατότητα να δημιουργούν πολλαπλές ασκήσεις με την ίδια μαθηματική βάση. Ο τρόπος με τον οποίο μπορούν να κάνουν αυτές τις αλλαγές είναι μέσω του κουμπιού “Επεξεργασία”, το οποίο θα εμφανίσει την αντίστοιχη διεπαφή επεξεργασίας ασκήσεων (Σχήμα 6.13).

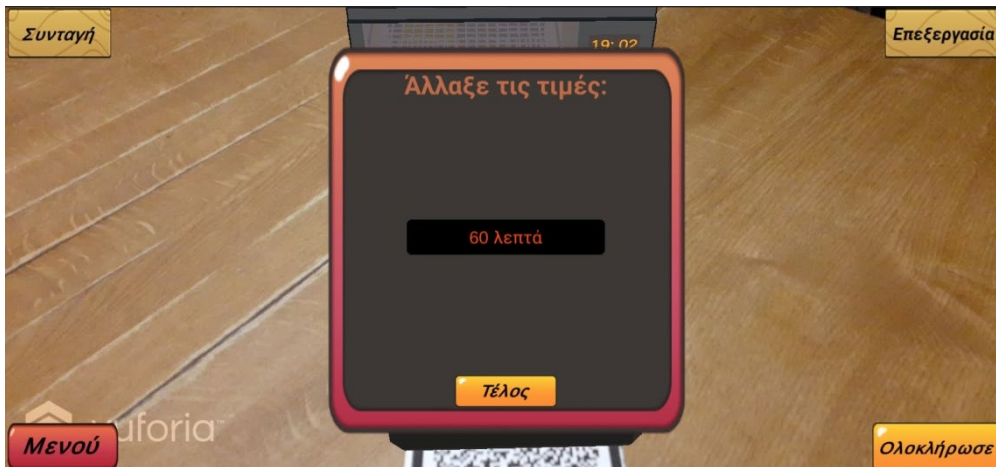


Σχήμα 6.13: Διεπαφή επεξεργασίας ασκήσεων παιχνιδιού ChangeValues

Ανοίγοντας την διεπαφή εμφανίζεται το inputField το οποίο αντιστοιχεί στην μεταβλητή της άσκησης και μέσα στο οποίο αναγράφεται η αρχική της τιμή. Πιέζοντας μέσα σε αυτό, ο παίκτης μπορεί να πληκτρολογήσει το νέο χρονικό διάστημα το οποίο θα πρέπει να υπολογίσει και πιέζοντας τέλος η αλλαγή ορίζεται τόσο στην εμφάνιση της διεπαφής όσο και στην απάντηση της τελικής ερώτησης. Η

Κεφάλαιο 6ο:

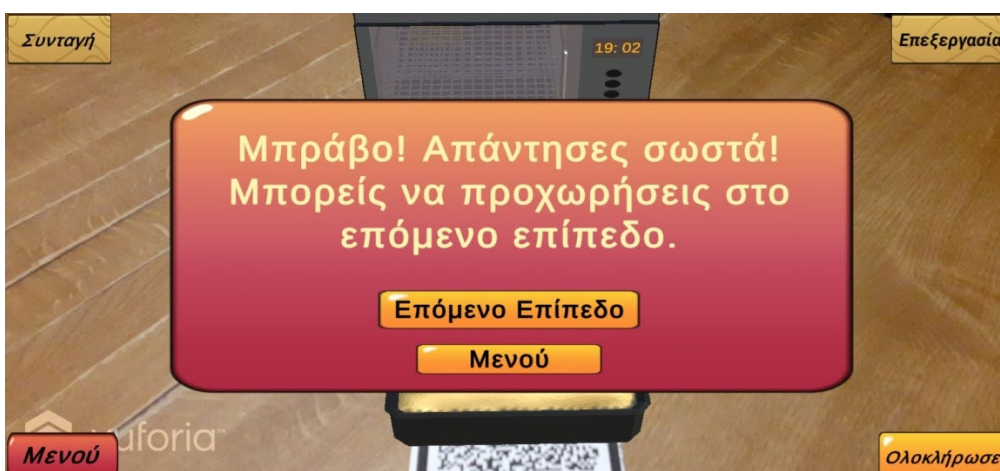
διαδικασία αυτή πραγματοποιείται μέσω του script ChangeValues, μπορεί να επαναληφθεί όσες φορές χρειαστεί και παρουσιάζεται στα παρακάτω σχήματα (6.13, 6.14, 6.15, 6.16).



Σχήμα 6.14: Αλλαγή μεταβλητής χρόνου της άσκησης σε "60 λεπτά" από "70 λεπτά"



Σχήμα 6.15: Εμφάνιση της αλλαγής της μεταβλητής χρόνου στην διεπαφή κειμένου



Σχήμα 6.16: Ολοκλήρωση σκηνής έπειτα από την ορθή απάντηση στην επεξεργασμένη ερώτηση

6.5 Αναγωγή Στη Μονάδα

Προχωρώντας στα μαθηματικά, παίζουμε στην θεωρία της αναγωγής στην μονάδα. Το μέρος αυτό των μαθηματικών αφορά την εύρεση ενός κομματιού μίας ποσότητας, με σκοπό να υπολογιστεί το ζητούμενο της κάθε άσκησης. Από την μεριά του παιχνιδιού, η θεωρία αυτή συνδυάστηκε με μία απλή διαδικασία ετοιμασίας τσαγιού στο πρώτο του μέρος και μία άσκηση αναγωγής στο δεύτερο μέρος. Με άλλα λόγια, το συγκεκριμένο μέρος του παιχνιδιού πρόκειται για συνδυασμό δύο σκηνών. Στην πρώτη σκηνή το πρώτο πράγμα το οποίο βλέπει ο παίκτης, είναι ένα συμβουλευτικό μήνυμα για το πώς να προχωρήσει (Σχήμα 6.17).



Σχήμα 6.17: Συμβουλευτικό μήνυμα σκηνής τσαγιού

Κλείνοντας το μήνυμα, ο παίκτης έχει μπροστά του μία κούπα με νερό, καθώς και ένα σακουλάκι τσάι (Σχήμα 6.18). Στο συγκεκριμένο σημείο ο παίκτης δεν χρειάζεται να λύσει κάποιο πρόβλημα ή να κάνει κάποια περίπλοκη πράξη, παρα μόνο να σύρει το τσάι στην κούπα με το νερό και να παρατηρήσει ένα μικρό animation (Σχήμα 6.19).



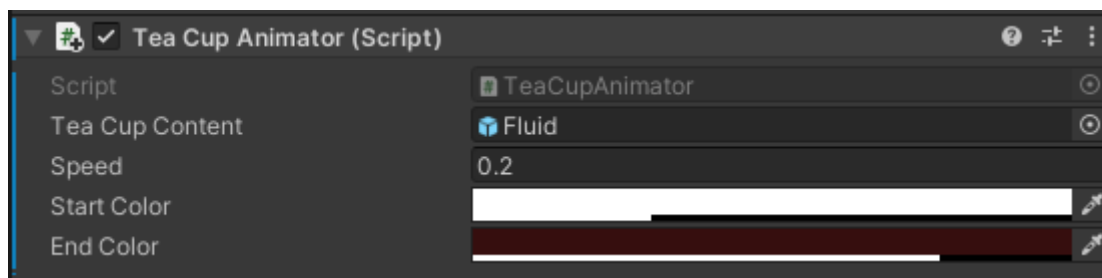
Σχήμα 6.18: Όψη σκηνής ετοιμασίας τσαγιού



Σχήμα 6.19: Animation τσαγιού

6.5.1 TeaCupAnimator

Το παραπάνω animation πραγματοποιείται μέσω της χρήσης ενός αρκετά απλού script με όνομα TeaCupAnimator. Το script αποτελείται από δύο μεθόδους, την Update() και την OnTriggerEnter(). Προτού προχωρήσουμε στις μεθόδους, σημαντικό είναι να αναφέρουμε τις εξωτερικές μεταβλητές τις οποίες δέχεται το script (Σχήμα 6.20). Οι μεταβλητές αυτές, αφορούν το τρισδιάστατο αντικείμενο το οποίο αναπαριστά το νερό της κούπας, το αρχικό χρώμα και το τελικό χρώμα του animation, καθώς και την ταχύτητα αλλαγής του από το πρώτο στο δεύτερο. Επιστρέφοντας στις μεθόδους, η μέθοδος Update() καλείται σε κάθε καρέ της σκηνής και ρόλος της είναι να ελέγχει μέσω μίας boolean μεταβλητής (startChangeOfMaterial) το πότε να ξεκινήσει το animation. Η μεταβλητή αυτή έχει ψευδής (false) τιμή κατά τον ορισμό της και παίρνει αληθής τιμή μέσω της μεθόδου OnTriggerEnter(). Αυτό συμβαίνει καθώς η μέθοδος OnTriggerEnter() θα τρέξει όταν το τσάι ακουμπήσει την κούπα, με αποτέλεσμα ο συνεχής έλεγχος της Update() να είναι αληθής στην ίδια χρονική στιγμή. Έτσι, έχοντας περάσει τον έλεγχο εντός της μεθόδου Update(), χρησιμοποιείται η μέθοδος Lerp() της κλάσης Color. Η μέθοδος αυτή δέχεται τρία ορίσματα, τα δύο χρώματα αρχής και τέλους και έναν αριθμό μετάβασης, με αποτέλεσμα να αλλάζει τον αρχικό χρωματισμό προς τον τελικό, με βήμα τον αριθμό που ορίστηκε. Έτσι, η μέθοδος χρησιμοποιείται για την αλλαγή του χρώματος του νερού προς το χρώμα του τσαγιού, με βάση ενός χρονικού διαστήματος επηρεασμένου από την ταχύτητα που ορίσαμε .



Σχήμα 6.20: Εξωτερικές μεταβλητές του script TeaCupAnimator

```

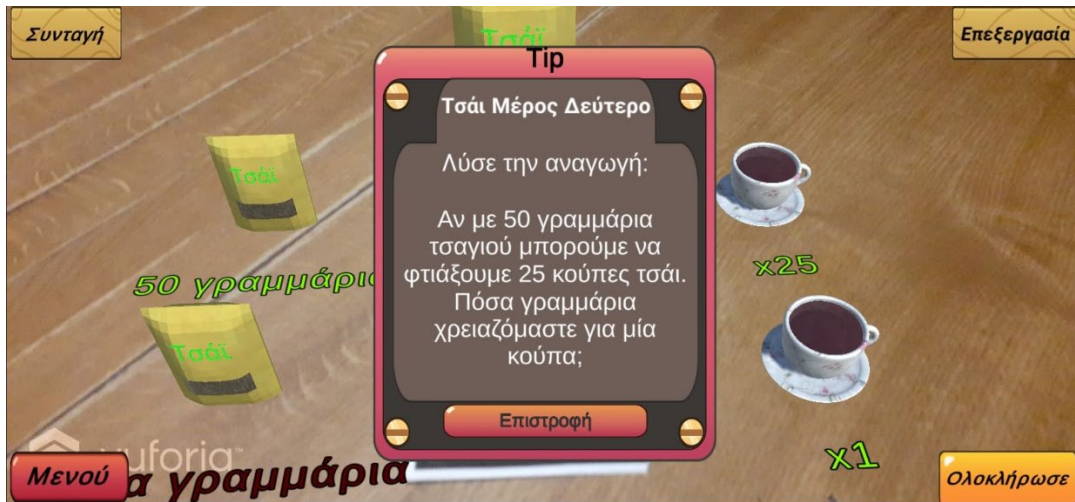
Unity Message | 0 references
private void Update()
{
    if (startChangeOfMaterial)
    {
        float t = (Time.time - startTime) * speed;
        teaCupContent.GetComponent<Renderer>().material.color = Color.Lerp(startColor, endColor,t);
    }
}

Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    startChangeOfMaterial = true;
    startTime = Time.time;
}

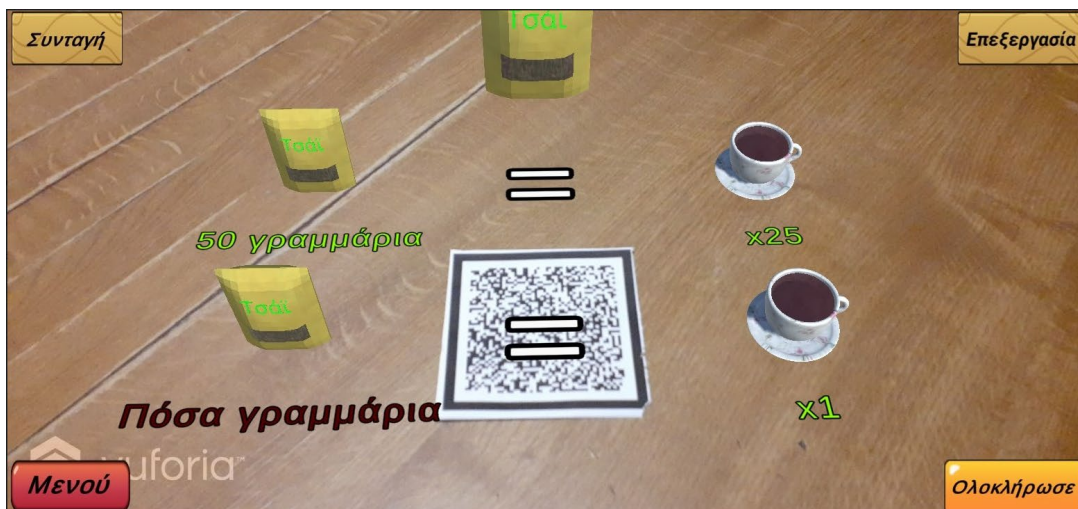
```

Σχήμα 6.21: Μέθοδοι Update() & OnTriggerEnter του script TeaCupAnimator

Ολοκληρώνοντας το animation ο παίκτης μπορεί να προχωρήσει στο επόμενο μέρος του παιχνιδιού, το οποίο αφορά μία άσκηση αναγωγής σχετική με γραμμάρια τσαγιού και αντίστοιχες κούπες. Η σκηνή ξεκινάει δίνοντας στον παίκτη μία μικρή περιγραφή της άσκησης που βρίσκεται μπροστά του, με την μορφή μίας συμβουλής (tip) (Σχήμα 6.22). Στο μήνυμα αυτό, περιγράφεται μία άσκηση αναγωγής στην μονάδα και πρόκειται για το δεύτερο είδος άσκησης το οποίο υπάρχει στο παιχνίδι. Όπως προαναφέραμε το παιχνίδι έχει δύο είδη ασκήσεων, σε αυτό το σημείο του παιχνιδιού παρουσιάζεται το δεύτερο είδος, η άσκηση-συνταγή. Η βασική δομή των ασκήσεων αυτών αφορά ένα αντικείμενο CraftingRecipe το οποίο περιέχει μόνο ένα υλικό, το οποίο είναι και το ζητούμενο της άσκησης. Έτσι, ο παίκτης έχει μπροστά του τις μεταβλητές της άσκησης, έναν άγνωστο ο οποίος έχει διαφορετικό χρωματισμό και έχει τον ρόλο του “δοχείου”, καθώς και το υλικό το οποίο θα πρέπει να χρησιμοποιήσει για να απαντήσει σωστά (Σχήμα 6.23). Συγκεκριμένα, σέρνοντας το τσάι, το οποίο βρίσκεται στο πάνω μέρος της οθόνης στην άγνωστη ποσότητα κάτω αριστερά, εμφανίζεται η διεπαφή επιλογής ποσότητας (Σχήμα 6.24). Αν ο παίκτης λύσει σωστά την άσκηση, δηλαδή δώσει μέσω της διεπαφής την σωστή ποσότητα και πιάσει το κουμπί “Ολοκλήρωσε”, θα εμφανιστεί το αντίστοιχο μήνυμα επιτυχίας που υπάρχει σε όλες τις σκηνές. Σε περίπτωση όμως που ο παίκτης απαντήσει λάθος ή δεν δώσει απάντηση, θα εμφανιστεί ένα διαφορετικό μήνυμα λάθους το οποίο θα εξηγήσει καλύτερα την πράξη που πρέπει να πραγματοποιήσει ο παίκτης (Σχήμα 6.25). Όλες αυτές οι λειτουργίες πραγματοποιούνται μέσω των scripts BowlHandler, MainUiHandler, CraftingRecipe, NumpadHandler και DragAndDrop, τα οποία αναλύθηκαν στο προηγούμενο κεφάλαιο.



Σχήμα 6.22: Εκφώνηση άσκησης αναγωγής στην μονάδα



Σχήμα 6.23: Όψη άσκησης αναγωγής στην μονάδα με χρήση αντικειμένων παιχνιδιού



Σχήμα 6.24: Ορισμός λύσης ασκήσεως μέσω της διεπαφής Numpad



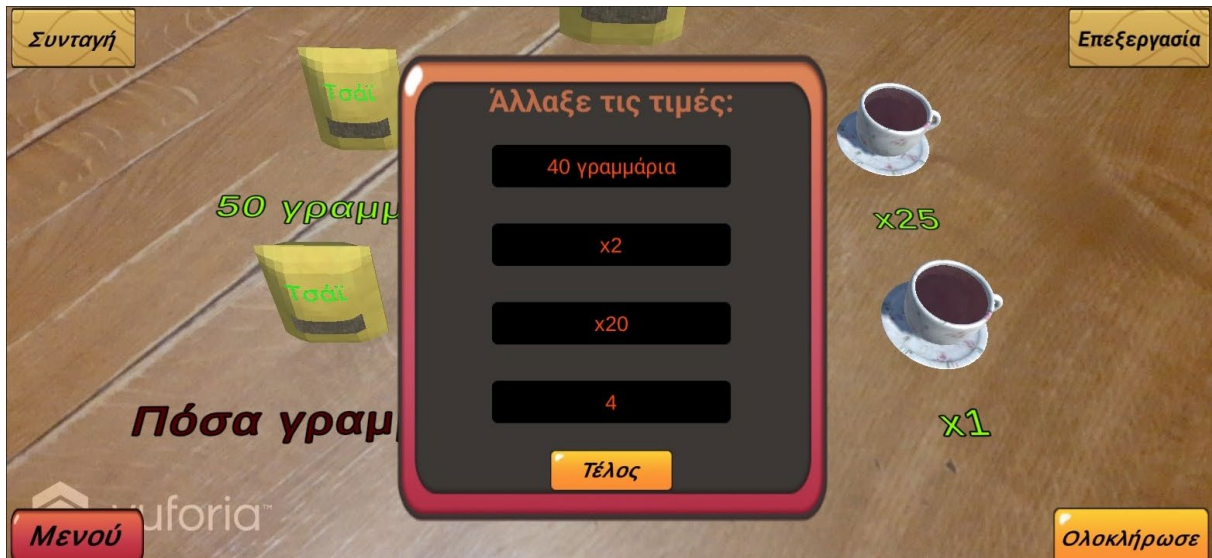
Σχήμα 6.25: Διαφοροποιημένο μήνυμα λάθους ασκήσεως αναγωγής

6.5.2 Επεξεργασία τιμών άσκησης τύπου συνταγής

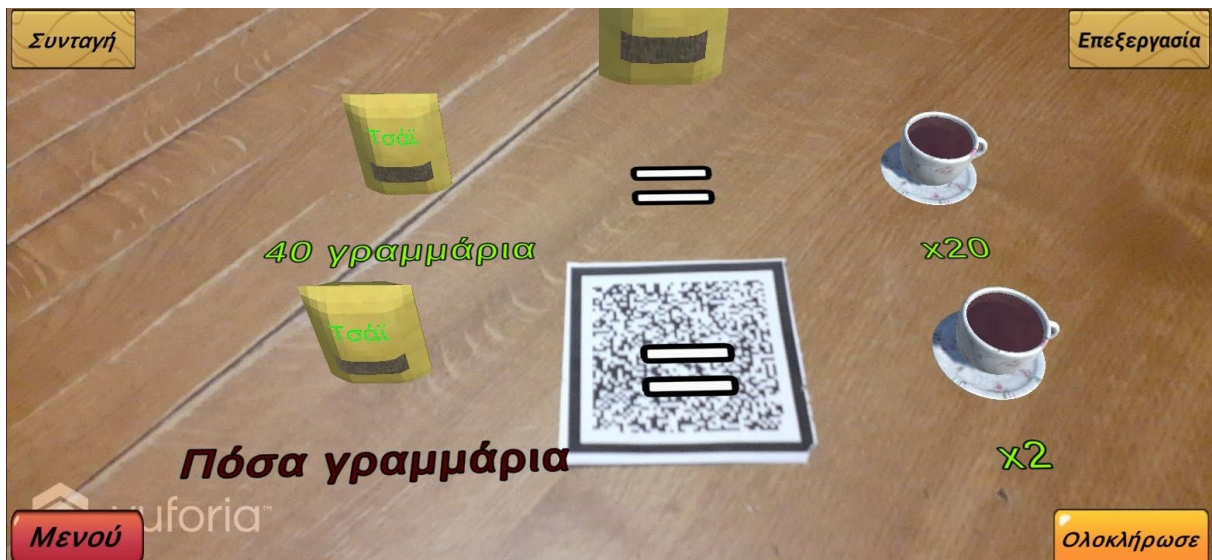
Το αρχικό πρόβλημα που δίνεται αφορά μία απλή αναγωγή στην μονάδα. Δίνεται όμως και η δυνατότητα αλλαγής των μεταβλητών της άσκησης, ώστε να καλύπτει καλύτερα τα προβλήματα αναγωγής των μαθηματικών. Η αλλαγή αυτή πραγματοποιείται πάλι μέσω της διεπαφής ChangeValuesPanel και του script ChangeValue, με βασική διαφορά πως το είδος της άσκησης πρόκειται για άσκηση-συνταγή. Πιέζοντας το κουμπί επεξεργασίας, εμφανίζεται η υπεύθυνη διεπαφή η οποία περιέχει όλες τις μεταβλητές της άσκησης καθώς και ένα επιπλέον πεδίο υπεύθυνο για την πληκτρολόγηση της νέας λύσης (Σχήμα 6.26). Η διαδικασία ορισμού νέων τιμών καθώς και λύσης της νέας άσκησης παραμένει ίδια από την πλευρά του παίκτη, κάτι το οποίο μπορεί να παρατηρηθεί και από τα επόμενα σχήματα (6.26, 6.27, 6.28).



Σχήμα 6.26: Διεπαφή επεξεργασίας ασκήσεως αναγωγής



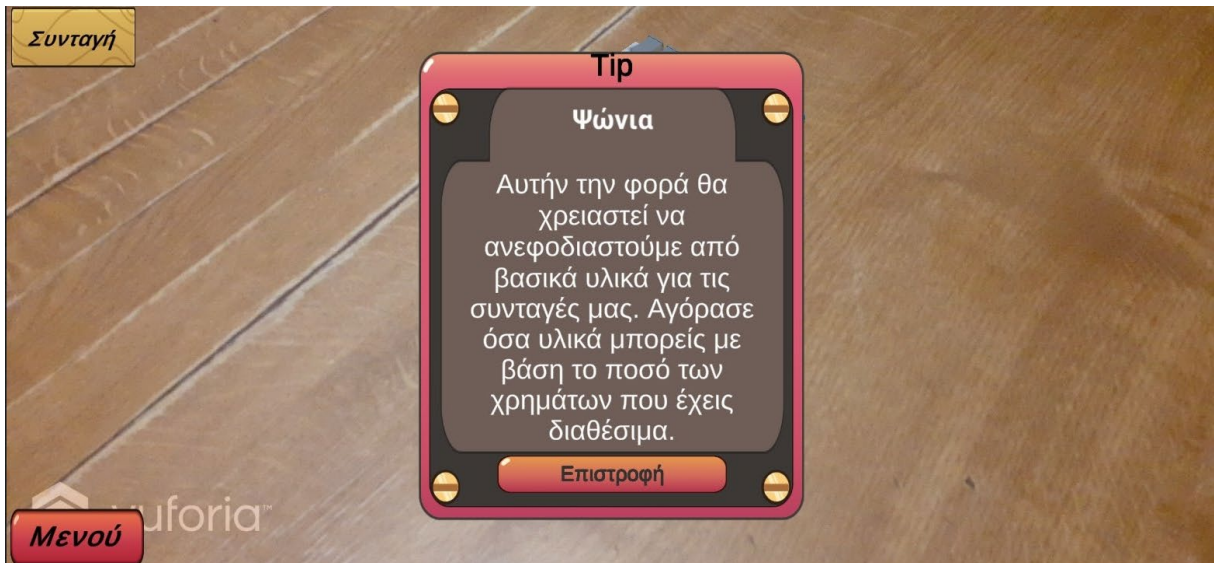
Σχήμα 6.27: Αλλαγή των μεταβλητών και λύσεων της άσκησης αναγωγής



Σχήμα 6.28: Όψη άσκησης αναγωγής έπειτα από την επεξεργασία της

6.6 Δεκαδικοί Αριθμοί

Άλλο ένα μεγάλο μέρος των μαθηματικών αφορά τους δεκαδικούς αριθμούς. Για την καλύτερη αντιπροσώπευση των δεκαδικών αριθμών, επιλέχθηκε η δημιουργία ενός σεναρίου κατά του οποίου ο παίκτης καλείται να αγοράσει κάποια υλικά, τα οποία θα χρειαστεί στις επόμενες συνταγές του. Με άλλα λόγια επιλέχθηκε η δημιουργία μίας άσκησης χρημάτων, κατά την οποία ο παίκτης έχει ένα αρχικό ποσό το οποίο πρέπει να χρησιμοποιήσει σωστά ώστε να αγοράσει ένα αριθμό υλικών. Η σκηνή μας ξεκινάει για ακόμη μία φορά με ένα συμβουλευτικό μήνυμα, το οποίο περιγράφει στον παίκτη την διαδικασία την οποία πρέπει να ακολουθήσει (Σχήμα 6.29). Κλείνοντας το μήνυμα αυτό, ο παίκτης βλέπει μπροστά του ένα σουπερμαρκετ στο οποίο μπορεί να “εισέλθει” με ένα απλό κλικ πάνω στο αντικείμενο (Σχήμα 6.30). Επίσης σε αυτήν την σκηνή δεν είναι διαθέσιμο του κουμπί “Ολοκλήρωσε”, καθώς είναι απαραίτητο για τον παίκτη να εισέλθει στο σουπερμαρκετ ώστε να ολοκληρώσει την σκηνή.



Σχήμα 6.29: Συμβουλευτικό μήνυμα σκηνής σούπερμαρκετ



Σχήμα 6.30: Εξωτερική όψη αντικειμένου σουπερμάρκετ

6.6.1 ShopPanel

Πιέζοντας το αντικείμενο το οποίο προσομοιάζει την εξωτερική εμφάνιση ενός σουπερμάρκετ, ο παίκτης εισέρχεται στην διεπαφή ShopPanel (Σχήμα 6.31). Η διεπαφή αυτή αποτελείται κυρίως από τρία υλικά τα οποία μπορεί να αγοράσει ο παίκτης. Πάνω από τα υλικά υπάρχουν κάποιες αριθμητικές ενδείξεις, οι οποίες έχουν ρόλο να μετράνε τον αριθμό των υλικών που προσθέτει ο παίκτης στο καλάθι του. Μαζί με τα υλικά δίνεται και η τιμή του κάθε υλικού αντίστοιχα, καθώς και τα διαθέσιμα χρήματα τα οποία διαθέτει ο παίκτης. Τα υλικά της διεπαφής αποτελούν κουμπιά στα οποία τους έχει προστεθεί από μία εικόνα αντίστοιχα. Η αγορά πραγματοποιείται πιέζοντας τα κουμπιά ώστε να προστεθούν τα αντίστοιχα υλικά στο καλάθι και πατώντας το κουμπί “Αγορά”, εφόσον έχει προσθέσει

τουλάχιστον ένα. Η λειτουργία της διεπαφής διεκπεραιώνεται μέσω των script Shop και ShopUiHandler.

- **ShopUiHandler:** Το script ShopUiHandler βρίσκεται πάνω στο τρισδιάστατο αντικείμενο supermarket και έχει παρόμοια λειτουργία με το script DragAndDrop, με την βασική διαφορά πως πρόκειται για μία απλοϊκή παραλλαγή η οποία ελέγχει συγκεκριμένα αν ο παίκτης έχει πιάσει το συγκεκριμένο τρισδιάστατο αντικείμενο (Σχήμα 6.30). Ο έλεγχος αυτός πραγματοποιείται μέσω δύο μεθόδων ονόματι FixedUpdate() και OpenShop(). Η μέθοδος OpenShop() λειτουργεί σχεδόν με τον ίδιο τρόπο όπως και η μέθοδος onTouch() του script DragAndDrop, με την διαφορά πως εδώ γίνεται ένας και μόνο έλεγχος, για την αφή του παίκτη προς το αντικείμενο supermarket. Η μέθοδος αυτή αποτελεί δηλαδή έναν έλεγχο, ο οποίος τρέχει περισσότερες από μία φορές ανά καρέ μέσω της FixedUpdate(). Στην περίπτωση που βγει αληθής, σημαίνει πως ο παίκτης πιάσε το αντικείμενο supermarket και ανοίγει η αντίστοιχη διεπαφή ShopPanel.
- **Shop:** Το δεύτερο και πιο σημαντικό script περιγράφει την βασική λειτουργία της διεπαφής, από την πλευρά της αγοράς και του υπολογισμού του κόστους. Το script βρίσκεται πάνω στην διεπαφή ShopPanel και αποτελείται από τρεις μεθόδους ονόματι Start(), addItemToBuy() και completeShopping(). Η μέθοδος start(), ορίζει το αντίστοιχο κείμενο στο πάνω δεξί μέρος της διεπαφής να εμφανίζει το ποσό το οποίο διαθέτει ο παίκτης. Η πρόσθεση αντικειμένων στο καλάθι, καθώς και ο υπολογισμός του συνολικού κόστους των αντικειμένων των οποίων προσθέτει ο παίκτης πραγματοποιείται μέσω της μεθόδου addItemToBuy(). Η μέθοδος έχει προστεθεί ως onClick γεγονός σε κάθε κάθε κουμπί-υλικό και δέχεται ως όρισμα ολόκληρο το κουμπί, καθώς μέσα σε αυτό εμπεριέχονται και τα αντικείμενα τα οποία διαθέτουν την τιμή του υλικού και την ένδειξη ποσότητας του (Σχήμα 6.33). Τέλος, η μέθοδος completeShopping() καλείται όταν ο παίκτης πιάσει το αντίστοιχο κουμπί ολοκλήρωσης της αγοράς και ελέγχει για το αν ο παίκτης αγόρασε παραπάνω υλικά απο αυτά τα οποία μπορεί, με βάση τα διαθέσιμα του χρήματα. Στην περίπτωση που ισχύει, εμφανίζεται μήνυμα λάθους, ενώ στην περίπτωση που δεν ξεπέρασε το διαθέσιμο ποσό εμφανίζεται η διεπαφή ερωτήσεων.



Σχήμα 6.31: Διεπαφή αγοράς υλικών ShopPanel

```

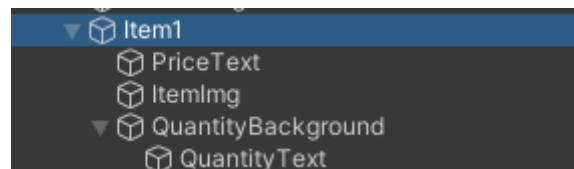
Unity Message | 0 references
void FixedUpdate()
{
    OpenShop();
}

1 reference
private void OpenShop()
{
    //If there are less or more than one finger touching don't do anything
    if (Input.touchCount != 1)
    {
        return;
    }
    //Get the position of the first touch
    Touch touch = Input.touches[0];
    Vector3 positionOfTouch = touch.position;

    if (touch.phase == TouchPhase.Began)
    {
        raycastHitInfo = new RaycastHit();
        Ray ray = camera.ScreenPointToRay(positionOfTouch);
        if (Physics.Raycast(ray, out raycastHitInfo) && raycastHitInfo.collider.name == "superMarket")
        {
            ShopPanel.SetActive(true);
            raycastHitInfo.collider.enabled = false; //Disable the shop collider after it is opened
        }
    }
}

```

Σχήμα 6.32: Script ShopUiHandler μέθοδοι FixedUpdate() & OpenShop()



Σχήμα 6.33: Δομή κουμπιού-υλικού της διεπαφής ShopPanel

```
void Start()
{
    moneyText.text = "Χρήματα: "+money.ToString() + " €";
}

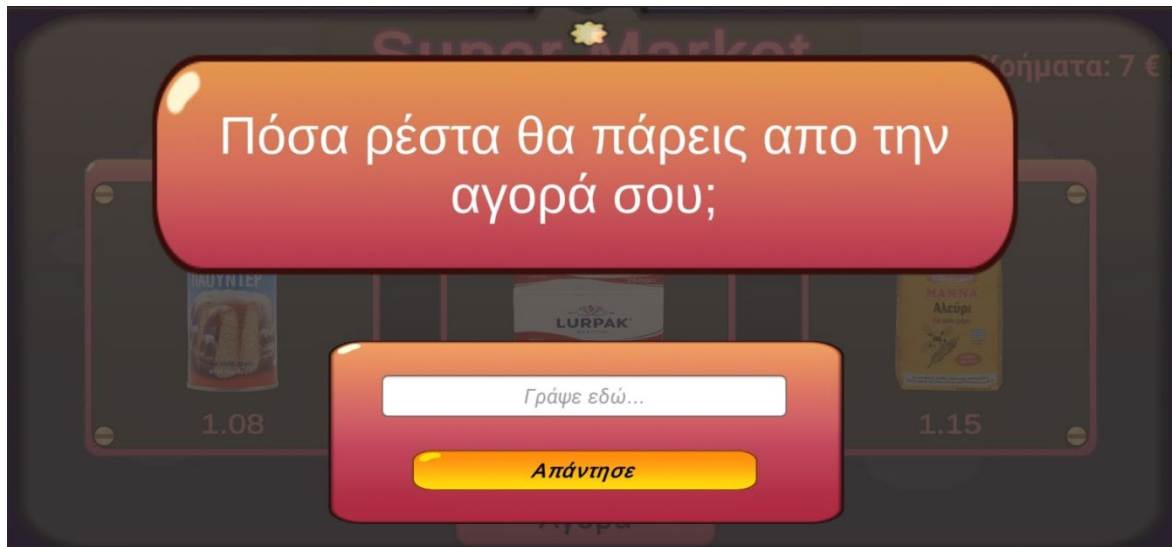
public void addItemToBuy(GameObject item)
{
    itemAmount = int.Parse(item.GetComponentInChildren<Text>()[1].text) + 1;
    sumPrice += Decimal.Parse(item.GetComponentInChildren<Text>()[0].text);

    item.GetComponentInChildren<Text>()[1].text = itemAmount.ToString();
}

public void completeShopping()
{
    if (sumPrice > money || sumPrice==0)
    {
        FailedTaskPanel.SetActive(true);
    }
    else
    {
        amountLeft =money - sumPrice;
        QuestionnairePanel.SetActive(true);
    }
}
```

Σχήμα 6.34: Script Shop μέθοδοι Start(), addItemToBuy() & completeShopping()

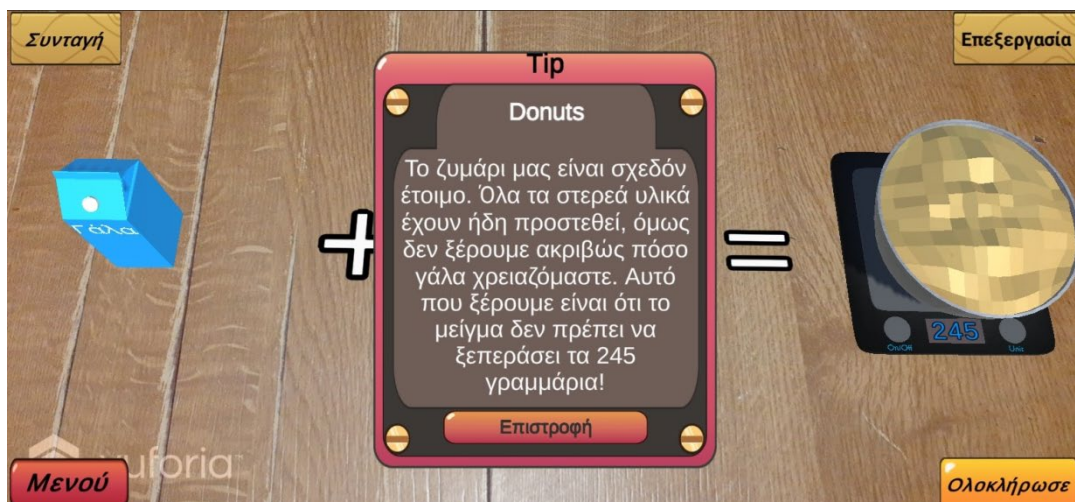
Έτσι λοιπόν, εφόσον ο παίκτης έχει επιλέξει ορθά τον αριθμό των υλικών των οποίων μπορεί να αγοράσει, θα του εμφανιστεί μία ερώτηση η οποία αφορά το υπόλοιπο ποσό των χρημάτων έπειτα από την αγορά του (Σχήμα 6.35). Η ερώτηση αυτή, πρόκειται ουσιαστικά για την τρίτη κατηγορία ερωτήσεων “Money”, την οποία αναφέραμε στο προηγούμενο κεφάλαιο. Η απάντηση της ερώτησης περνάει στην λίστα των ερωτήσεων-απαντήσεων μέσω της διεπαφής ShopPanel, καθώς έχει ήδη υπολογιστεί δυναμικά μέσω της μεθόδου addItemToBuy(). Οπότε, αν ο παίκτης απαντήσει σωστά στην τελική ερώτηση περί των χρημάτων, θα εμφανιστεί το μήνυμα επιτυχίας και μπορεί να προχωρήσει στο επόμενο σενάριο του παιχνιδιού.



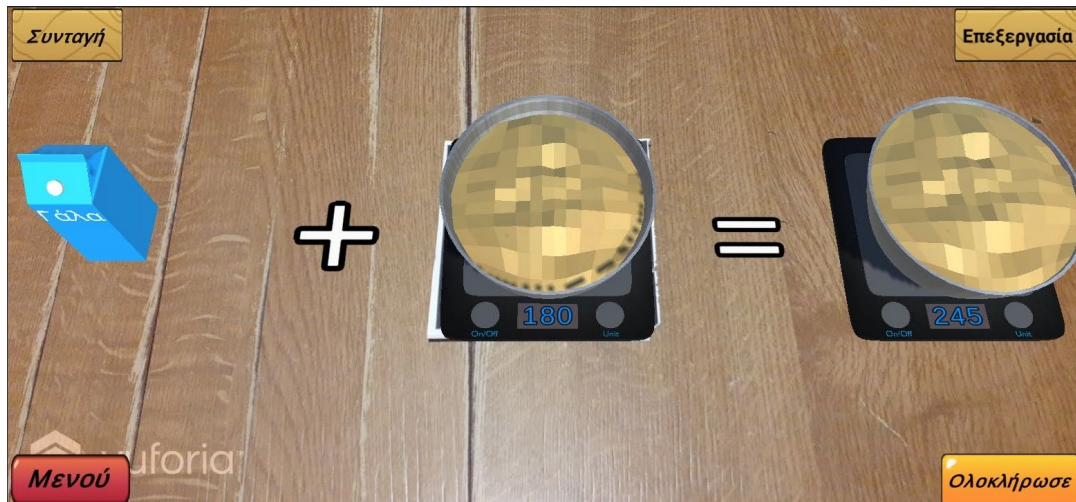
Σχήμα 6.35: Εμφάνιση ερώτησης κατηγορίας Money μέσω της διεπαφής QuestionnairePanel

6.7 Εξισώσεις

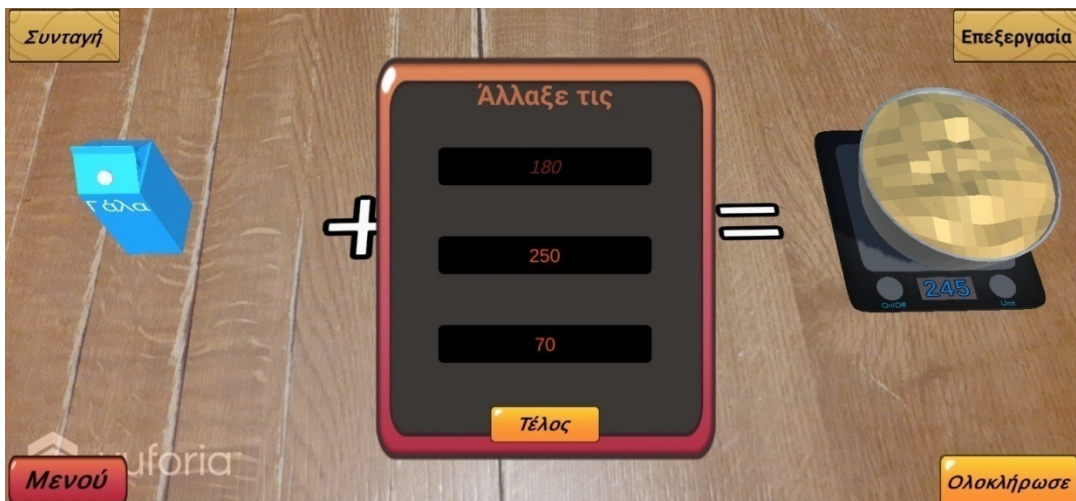
Προχωρώντας στα μαθηματικά καθώς και στο παιχνίδι, μπαίνουμε στο κεφάλαιο των εξισώσεων και συγκεκριμένα σε απλές εξισώσεις πρώτου βαθμού, στις οποίες γνωρίζοντας δύο ποσά υπολογίζουμε τον άγνωστο x . Για την προσομοίωση μίας τέτοιας εξίσωσης, χρησιμοποιήθηκε ένα σενάριο κατά το οποίο ο παίκτης έχει έτοιμο το μείγμα μπροστά του, όμως λείπει από την συνταγή του το πόσο γάλα χρειάζεται για να την ολοκληρώσει. Η μόνη πληροφορία την οποία διαθέτει αφορά το βάρος του μείγματος και πως δεν πρέπει να ξεπεράσει τα 245 γραμμάρια. Έτσι λοιπόν, το πρώτο πράγμα το οποίο αντικρίζει ο παίκτης ανοίγοντας την σκηνή, είναι αυτή η εκφώνηση σε μορφή συμβουλής (Σχήμα 6.36). Η συνταγή αφορά την δημιουργία λουκουμάδων και μπροστά του ο παίκτης διαθέτει δύο ζυγαριές φαγητού, οι οποίες παρουσιάζουν το αρχικό και τελικό βάρος του μείγματος. Επίσης διατίθεται το γάλα το οποίο θα χρησιμοποιήσει ο παίκτης για να λύσει το πρόβλημα. Η άσκηση αυτή πρόκειται για άσκηση-συνταγή, οπότε για να την λύσει ο παίκτης χρειάζεται να σύρει το υλικό στο πρώτο δοχείο μείγματος και να ορίσει την σωστή ποσότητα. Αξίζει να σημειωθεί, πως και σε αυτή την άσκηση δίνεται η επιλογή της επεξεργασίας και αλλαγής των τιμών της, μέσω της διεπαφής ChangeValues (Σχήμα 6.38).



Σχήμα 6.36: Εκφώνηση εξίσωσης-συνταγής donuts



Σχήμα 6.37: Όψη εξίσωσης με χρήση αντικειμένων παιχνιδιού



Σχήμα 6.38: Αλλαγή μεταβλητών άσκησης μέσω διεπαφής ChangeValues

6.8 Αναλογίες και Ποσοστά

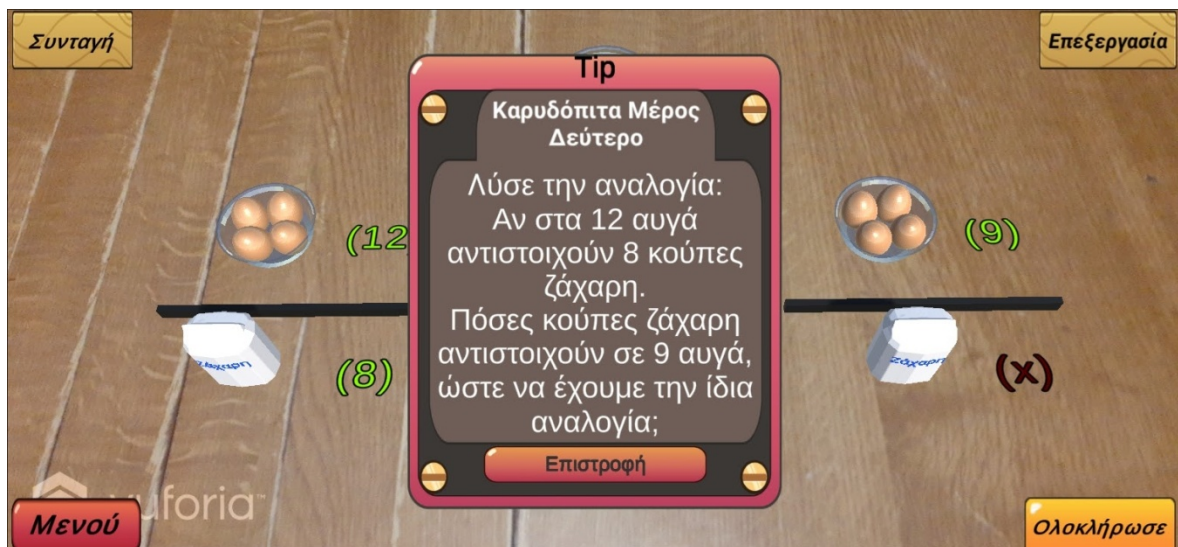
Το επόμενο μέρος του παιχνιδιού, αποτελείται από τέσσερις σκηνές κατά τις οποίες ο παίκτης υλοποιεί μία συνταγή καρυδόπιτας, η οποία έχει χωριστεί σε τέσσερα μέρη. Δύο από αυτά αποτελούν ασκήσεις αναλογιών και ποσοστών αντίστοιχα.

Η πρώτη σκηνή επιστρέφει στις ρίζες του παιχνιδιού, δηλαδή της υλοποίησης μίας απλής συνταγής κατά την οποία ο παίκτης διαθέτει ένα σύνολο υλικών, καθώς και ένα μπολ μίξης, στο οποίο θα πρέπει να προσθέσει τις αντίστοιχες ποσότητες που του δίνονται (Σχήμα 6.39). Η συνταγή διατηρείται και στην επόμενη σκηνή, ως η βάση γύρω από την οποία δημιουργείται η άσκηση.



Σχήμα 6.39: Πρώτη σκηνή υλοποίησης μείγματος καρυδόπιτας

Η δεύτερη σκηνή πρόκειται για μία ακόμα άσκηση τύπου συνταγής, κατά την οποία ο παίκτης καλείται να λύσει μία αναλογία, με βάση των αριθμό αυγών και ζάχαρης που χρησιμοποίησε στην πρώτη σκηνή. Η εκφώνηση της άσκησης, δίνεται μέσω της διεπαφής TipPanel (Σχήμα 6.40), την οποία όταν κλείσει ο παίκτης μπορεί να προχωρήσει στην λύση της (Σχήμα 6.41). Όπως και σε προηγούμενες ασκήσεις, έτσι και εδώ δίνεται η επιλογή επεξεργασίας.



Σχήμα 6.40: Εκφώνηση αναλογίας-συνταγής καρυδόπιτας



Σχήμα 6.41: Όψη αναλογίας με χρήση αντικειμένων παιχνιδιού

Η τρίτη σκηνή, προχωράει στην προετοιμασία του σιροπιού το οποίο θα χρησιμοποιήσει ο παίκτης σαν τελικό βήμα στην υλοποίηση της καρυδόπιτας. Η σκηνή ξεκινάει με μία μικρή εκφώνηση, την οποία όταν κλείσει ο παίκτης θα δει τις ποσότητες των υλικών, τις οποίες χρειάζεται για την προετοιμασία του σιροπιού. Αυτή η σκηνή αποτελεί όπως και η πρώτη μία απλή υλοποίηση ενός μείγματος, στην οποία ο παίκτης πρέπει απλά να προσθέσει τις σωστές ποσότητες υλικών.

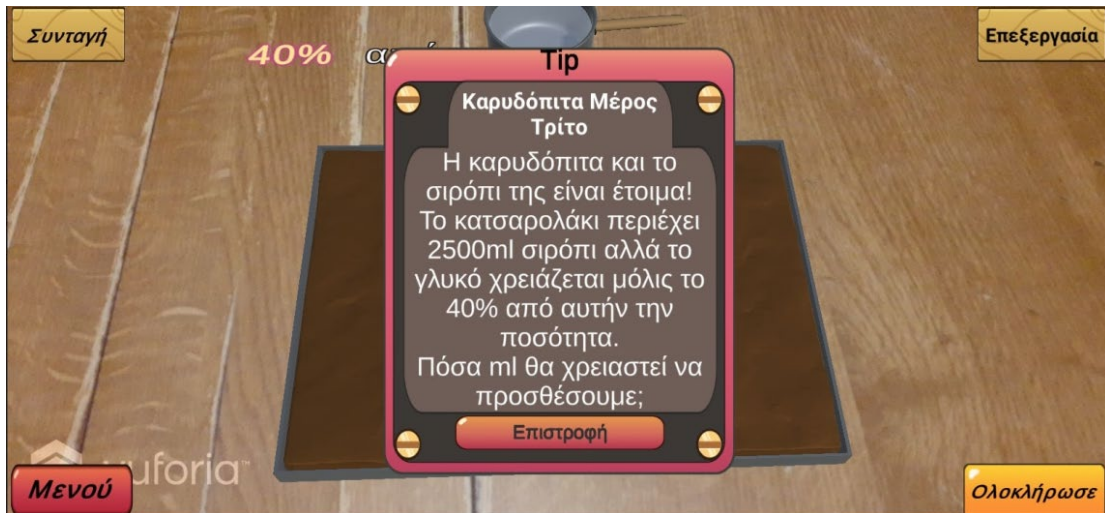


Σχήμα 6.42: Συνταγή σιροπιού τρίτης σκηνής

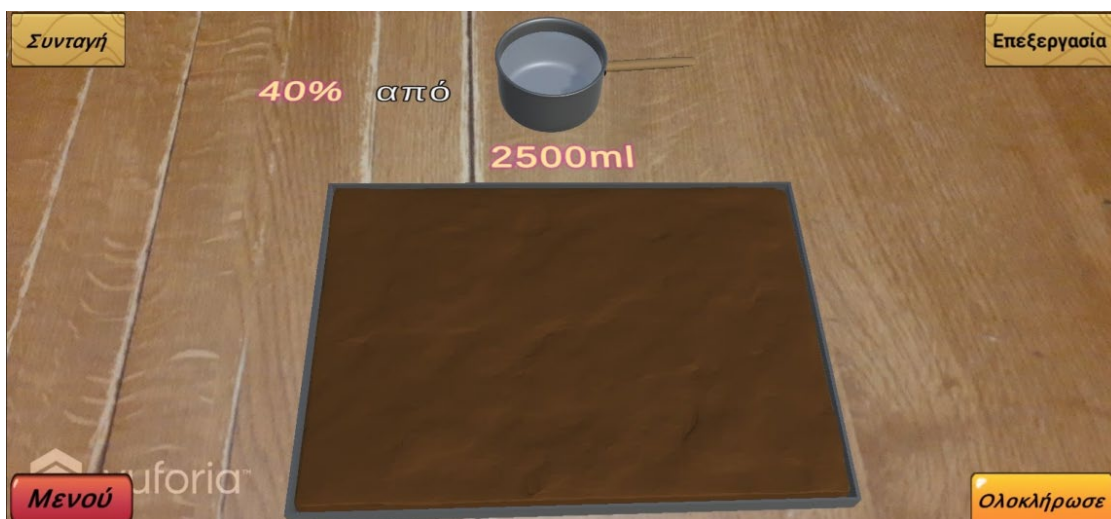


Σχήμα 6.43: Όψη σκηνής προετοιμασίας σιροπιού

Τελευταίο μέρος της συνταγής αποτελεί μία άσκηση ποσοστών, η οποία βασίζεται στην υλοποίηση της προηγούμενης σκηνής. Συγκεκριμένα, ενημερώνεται ο παίκτης πως το τελευταίο βήμα στην ολοκλήρωση της καρυδόπιτας είναι η πρόσθεση του σιροπιού, το οποίο ετοίμασε στην προηγούμενη σκηνή (Σχήμα 6.44). Η ποσότητα σιροπιού την οποία πρέπει να προσθέσει δίνεται σε μορφή ποσοστού, κάτι το οποίο καθιστά την συγκεκριμένη σκηνή ως άσκηση-συνταγής, στην οποία επίσης δίνεται η επιλογή επεξεργασίας των μεταβλητών.



Σχήμα 6.44: Εκφώνηση ποσοστού-συνταγής καρυδόπιτας

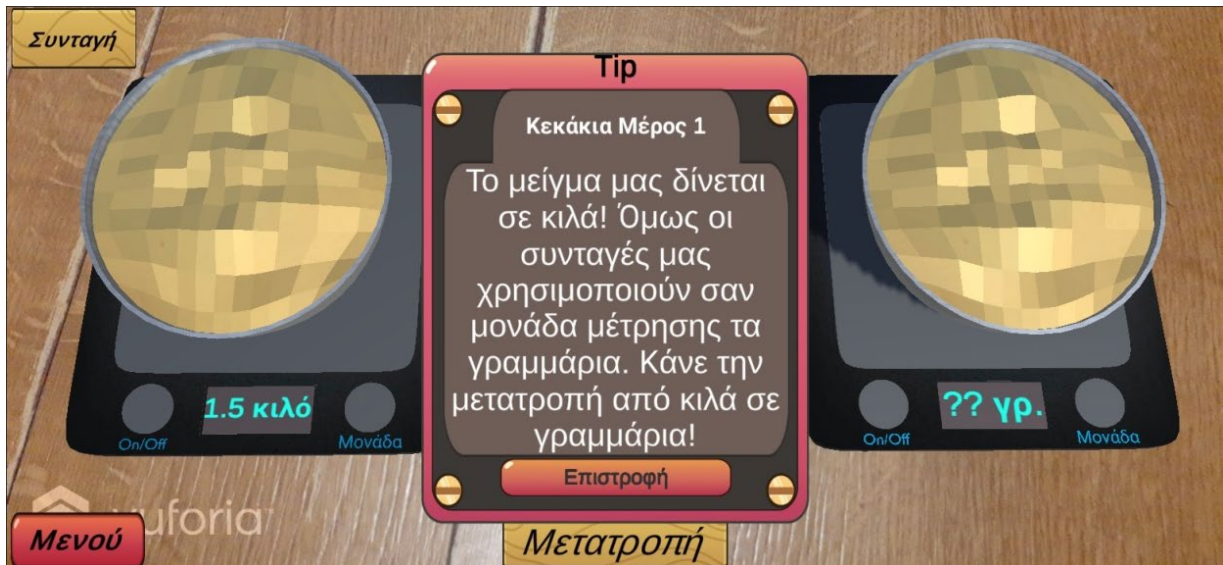


Σχήμα 6.45: Όψη άσκησης ποσοστού με χρήση αντικειμένων παιχνιδιού

6.9 Συλλογή και επεξεργασία δεδομένων - Μετρήσεις

Προχωρώντας στα κεφάλαια των μαθηματικών, μπαίνουμε στην θεωρία της συλλογής και επεξεργασίας δεδομένων, καθώς και των μετρήσεων. Τα κεφάλαια αυτά είναι μικρού περιεχομένου και περιγράφουν κυρίως την ανάλυση δεδομένων, τα γραφήματα και διάφορα είδη μετρήσεων όπως βάρους, χρόνου και χρημάτων. Τα δύο κεφάλαια χρησιμοποιήθηκαν συνδυαστικά για την δημιουργία ενός σεναρίου το οποίο αποτελείται από δύο σκηνές και αφορούν την δημιουργία μικρών κέικ. Αρχικά, το περιεχόμενο του κεφαλαίου των μετρήσεων έχει καλυφθεί και σε άλλες σκηνές του

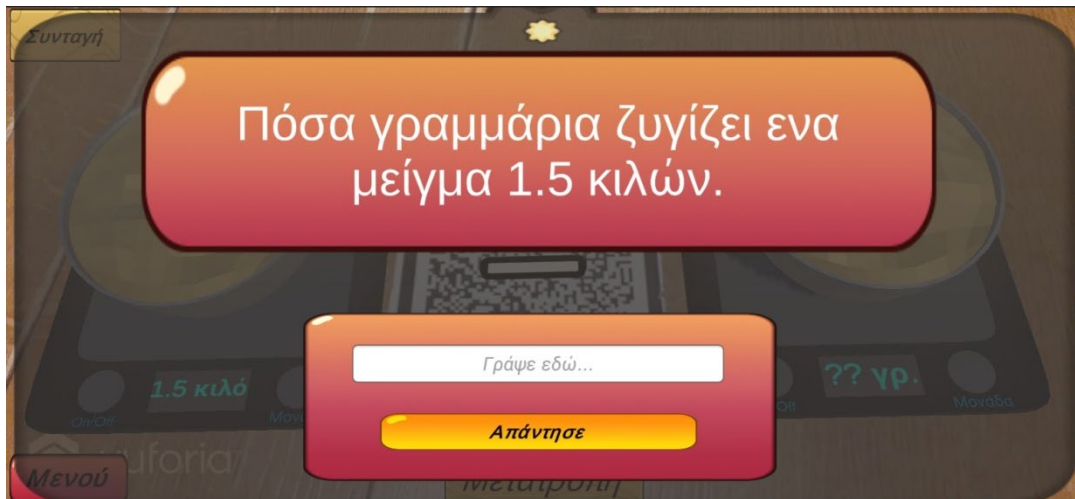
παιχνιδιού, όπως την σκηνή υπολογισμού χρόνου ή την σκηνή αγοράς υλικών. Για να ολοκληρωθεί το περιεχόμενο του, προστέθηκε επίσης μία σκηνή στην οποία ο παίκτης πρέπει να υλοποιήσει μία μετατροπή βάρους. Η σκηνή ξεκινάει δίνοντας στο παίκτη μία εκφώνηση μέσω της διεπαφής TipPanel, η οποία περιγράφει το πρόβλημα μετατροπής (Σχήμα 6.46). Το περιεχόμενο της αποτελείται από δύο ζυγαριές κουζίνας, οι οποίες περιγράφουν την ποσότητα του μείγματος πριν και μετά την μετατροπή, καθώς και ένα κουμπί μέσω του οποίου ο παίκτης θα την υλοποιήσει (Σχήμα 6.47). Η σκηνή πρόκειται για μία άσκηση-ερώτηση, στην οποία όταν ο παίκτης πιάσει το κουμπί της μετατροπής θα εμφανιστεί και η αντίστοιχη ερώτηση μέσω της διεπαφής QuestionnairePanel (Σχήμα 6.48).



Σχήμα 6.46: Εκφώνηση άσκησης-ερώτησης μετατροπής βάρους

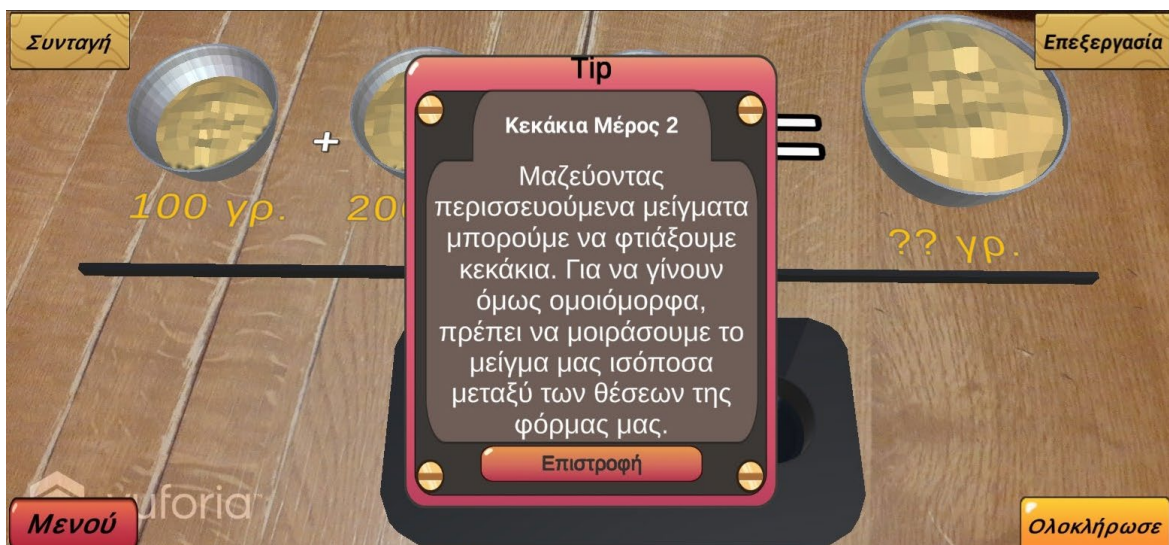


Σχήμα 6.47: Όψη ασκήσεως μετατροπής με χρήση αντικειμένων παιχνιδιού



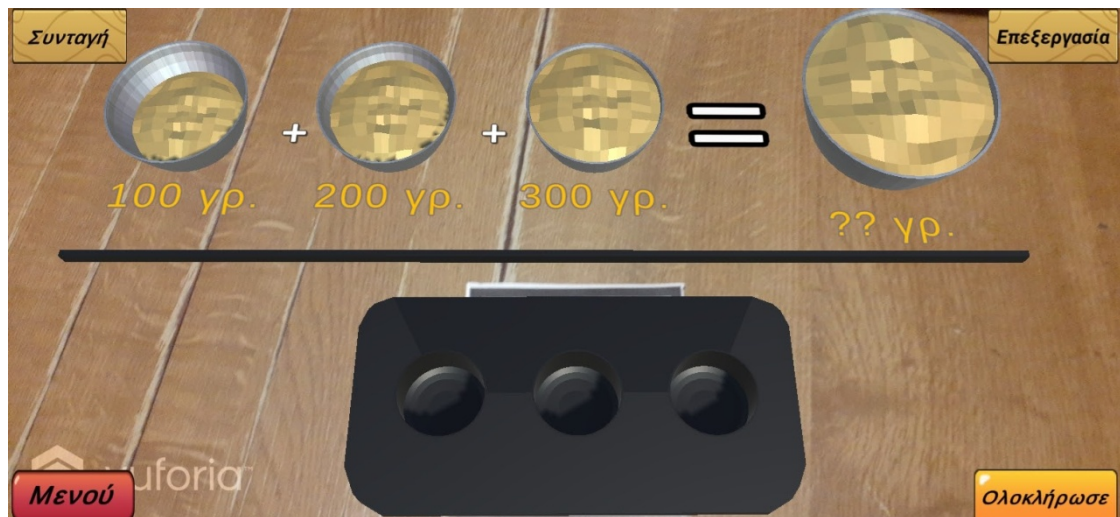
Σχήμα 6.48: Ερώτηση ασκήσεως μετατροπής μέσω της διεπαφής QuestionnairePanel

Μετά την ολοκλήρωση του παραπάνω μέρους της συνταγής, ο παίκτης μπορεί να προχωρήσει στο δεύτερο μέρος της, το οποίο αφορά την συλλογή και επεξεργασία δεδομένων. Σε αυτό το κεφάλαιο των μαθηματικών, οι μαθητές έρχονται σε επαφή με την έννοια του μέσου όρου, κάτι το οποίο αποτελεί και την βάση της επόμενης άσκησης. Σε αυτήν την σκηνή, ο παίκτης ενημερώνεται πως έχει περισσέψει αρκετή ποσότητα απο το προηγούμενο μείγμα για την υλοποίηση περισσότερων μικρών κέικ. Όμως, η ποσότητα αυτή έχει χωριστεί σε διαφορετικά μπόλ και ο παίκτης πρέπει να βρεί πως θα χωρίσει το συνολικό μείγμα ίσα στις τρεις θέσεις της φόρμας (Σχήμα 6.49).

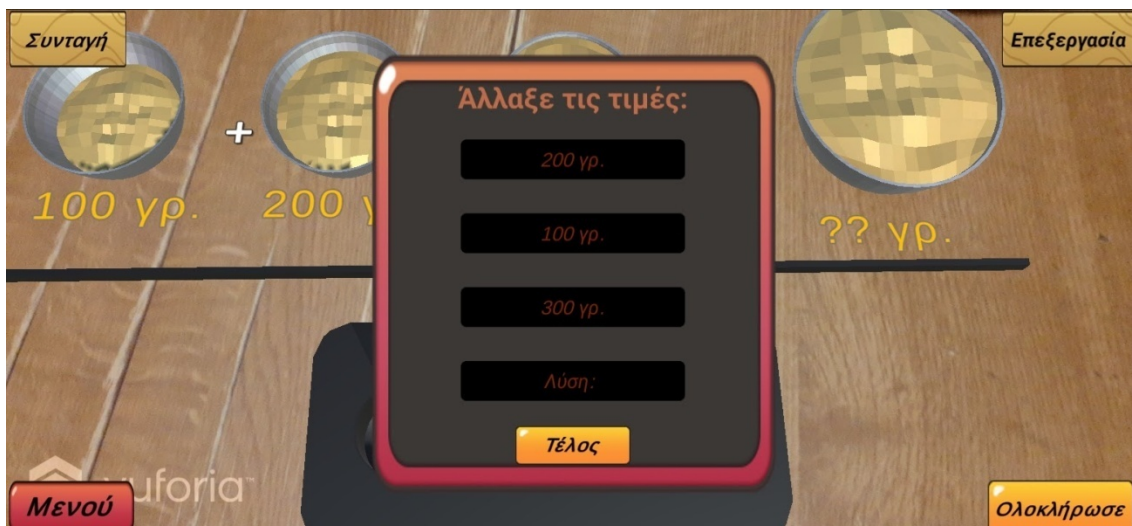


Σχήμα 6.49: Εκφώνηση άσκησης μέσου όρου συνταγής μικρών κέικ

Η άσκηση πρόκειται για άσκηση-συνταγή, στην οποία ο παίκτης πρέπει να υπολογίσει τον μέσο όρο με βάση του αθροίσματος των μειγμάτων, καθώς και των θέσεων της φόρμας και να χρησιμοποιήσει το αποτέλεσμα ώστε να την γεμίσει σωστά. Όπως και σε προηγούμενες ασκήσεις τέτοιου τύπου έτσι και εδώ, ο παίκτης μπορεί να απαντήσει σέρνοντας το ολικό μείγμα στη φόρμα που βρίσκεται μπροστά του (Σχήμα 6.50). Επίσης, δίνεται η επιλογή της επεξεργασίας της άσκησης, κατά την οποία επιτρέπεται η αλλαγή των επιμέρους ποσών τα οποία αποτελούν το άθροισμα του μείγματος (Σχήμα 6.51).



Σχήμα 6.50: Όψη άσκησης μέσου όρου με χρήση αντικειμένων παιχνιδιού



Σχήμα 6.51: Αλλαγή μεταβλητών αθροίσματος και λύσης, μέσω της διεπαφής ChangeValues

6.10 Γεωμετρία

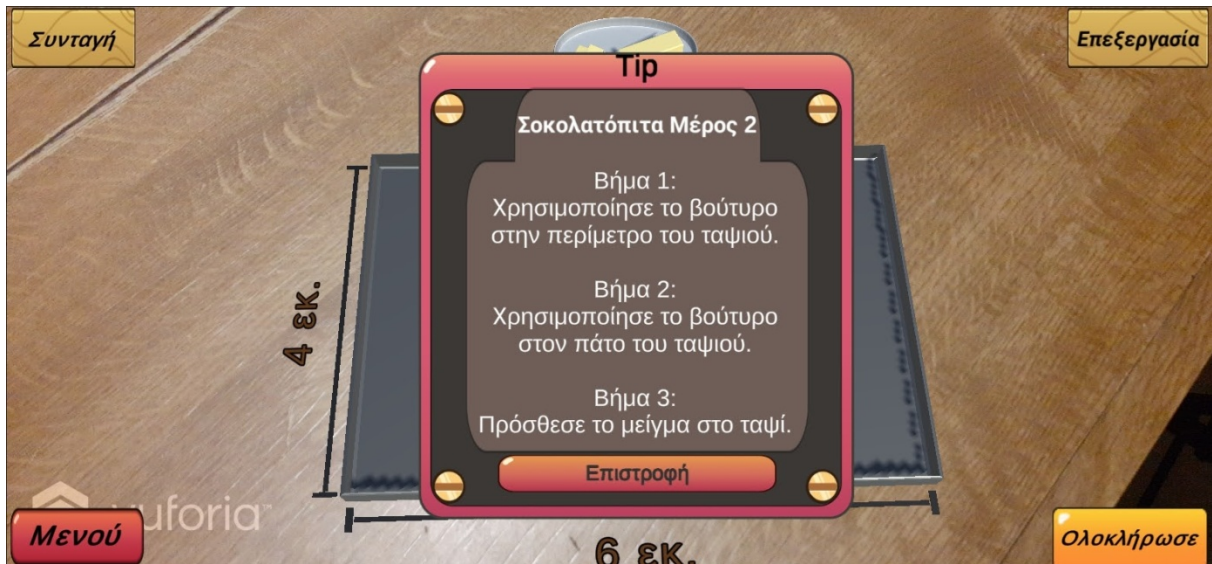
Τελευταίο κεφάλαιο του βιβλίου των μαθηματικών Στ' Δημοτικού, αποτελεί το κεφάλαιο της γεωμετρίας. Στο κεφάλαιο αυτό, περιγράφονται διάφορα γεωμετρικά σχήματα καθώς και οι έννοιες της περιμέτρου, του εμβαδόν και του όγκου. Από την πλευρά του παιχνιδιού, τελευταία συνταγή αποτελεί η συνταγή μιας σοκολατόπιτας η οποία αποτελείται από δύο μέρη, δηλαδή δύο σκηνών.

Η πρώτη σκηνή, αφορά την προετοιμασία του μείγματος της σοκολατόπιτας και πρόκειται για μία απλή σκηνή κατά την οποία ο παίκτης, με βάση την συνταγή που του δίνεται, προσθέτει ένα ένα τα υλικά στο μπολ μίξης (Σχήμα 6.52). Από την στιγμή που ο παίκτης έχει προσθέσει σωστά τα υλικά, μπορεί να προχωρήσει στην επόμενη σκηνή. Αυτό το μέρος του σεναρίου δεν αποτελεί κάποια άσκηση και επιστρέφει τον παίκτη στον βασικό τρόπο παιχνιδιού του παιχνιδιού, χωρίς να του ζητηθεί κάποιος επιπλέον υπολογισμός ή ερώτηση.

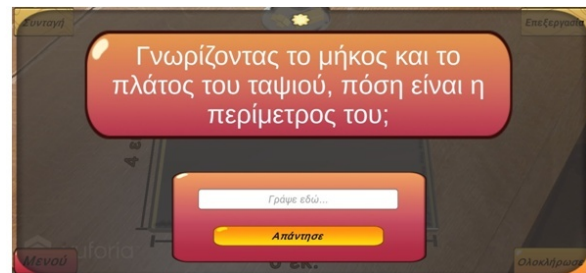
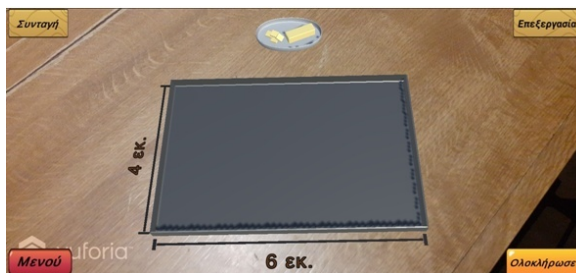


Σχήμα 6.52: Όψη προετοιμασίας μείγματος σοκολατόπιτας

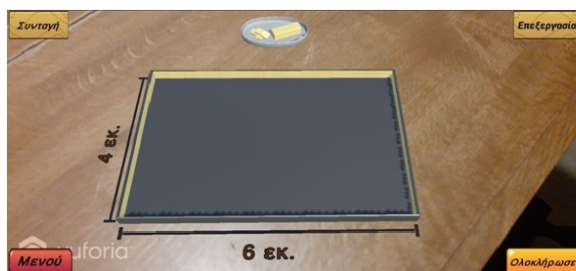
Προχωρώντας όμως στο επόμενο μέρος της συνταγής, ο παίκτης έρχεται σε επαφή και με το μαθηματικό κομμάτι της και συγκεκριμένα με το κομμάτι της γεωμετρίας. Σε αυτήν την σκηνή δίνονται στο παίκτη τρία βήματα τα οποία πρέπει να ακολουθήσει ώστε να ολοκληρώσει την συνταγή (Σχήμα 6.53). Μπροστά του βρίσκεται ένα ταψί σχήματος ορθογώνιου παραλληλογράμμου, το μήκος των πλευρών του σε μέτρα, καθώς και το βούτυρο με το οποίο θα πρέπει να το αλείψει. Κάθε βήμα της σκηνής πρόκειται και για ένα στάδιο το οποίο εμπεριέχει μία ερώτηση. Οι ερωτήσεις έχουν θέμα την γεωμετρία και συγκεκριμένα αφορούν τον υπολογισμό της περιμέτρου, του εμβαδόν και του όγκου του ταψιού. Στο πρώτο στάδιο, ο παίκτης καλείται να απλώσει το βούτυρο στις πλευρές του ταψιού, σέρνοντας το υλικό στο ταψί. Την στιγμή που έρθει σε επαφή το βούτυρο με το ταψί, εμφανίζεται η διεπαφή ερωτήσεων και ζητείται από τον παίκτη να υπολογίσει την περίμετρο του ταψιού (Σχήμα 6.54). Έχοντας απαντήσει σωστά, μπορεί να προχωρήσει στο δεύτερο βήμα, κατά το οποίο πρέπει να αλείψει το βούτυρο και στο υπόλοιπο ταψί. Όπως και στο προηγούμενο βήμα, κατά την επαφή των δύο αντικειμένων εμφανίζεται στον παίκτη η δεύτερη ερώτηση, στην οποία καλείται να υπολογίσει το εμβαδόν του ταψιού (Σχήμα 6.55). Απαντώντας σωστά, ο παίκτης προχωράει και στο τελευταίο βήμα της σκηνής, στο οποίο μπορεί να προσθέσει στο ταψί το μείγμα το οποίο ετοίμασε στην προηγούμενη σκηνή. Εφόσον ο παίκτης προσθέσει το μείγμα και πιάσει το κουμπί ολοκλήρωσης, εμφανίζεται και η τελευταία ερώτηση της σκηνής, η οποία ζητά από τον παίκτη να υπολογίσει τον όγκο του ταψιού με βάση την τιμή του ύψους του (Σχήμα 6.56). Αξίζει να σημειωθεί, πως και σε αυτήν την άσκηση δίνεται η επιλογή επεξεργασίας των τιμών και λύσεων της. Οι ερωτήσεις της σκηνής, βασίζονται στους γεωμετρικούς τύπους των ορθογώνιων παραλληλογράμμων και απαντώντας σωστά σε όλες ο παίκτης έχει πλέον ολοκληρώσει το παιχνίδι. Επίσης, δίνεται και σε αυτήν την άσκηση η δυνατότητα επεξεργασίας της.



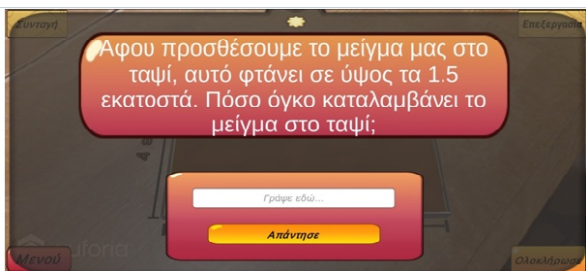
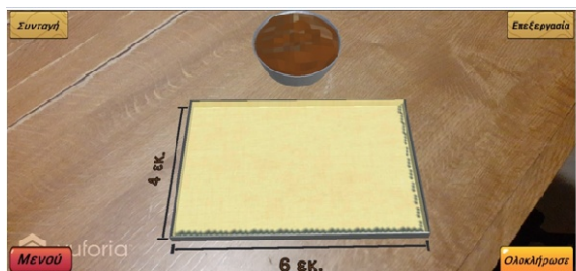
Σχήμα 6.53: Εμφάνιση βημάτων τελικού σταδίου συνταγής μέσω της διεπαφής TipPanel



Σχήμα 6.54: Υλοποίηση πρώτου βήματος και εμφάνιση αντίστοιχης ερώτησης περιμέτρου



Σχήμα 6.55: Υλοποίηση δεύτερου βήματος και εμφάνιση αντίστοιχης ερώτησης εμβαδόν



Σχήμα 6.56: Υλοποίηση τρίτου βήματος και εμφάνιση τελικής ερώτησης όγκου

6.10.1 ChocolateCakeHandler

Πέρα των βασικών script, σε αυτήν την σκηνή χρησιμοποιείται ένα επιπλέον script ChocolateCakeHandler για την εμφάνιση των ερωτήσεων μετά απο κάθε βήμα. Η λογική του είναι αρκετά απλή, καθώς χρησιμοποιεί την μέθοδο OnTriggerEnter και μία int μεταβλητή η οποία μετράει τον αριθμό των συγκρούσεων. Έτσι κάθε φορά που ένα αντικείμενο έρχεται σε επαφή με το ταψί, η μεταβλητή αυξάνεται κατά ένα και εφόσον ο αριθμός των συγκρούσεων είναι μικρότερος του δύο, εμφανίζεται η διεπαφή των ερωτήσεων (Σχήμα 6.57).

```

public class ChocolateCakeHandler : MonoBehaviour
{
    [SerializeField]
    private GameObject Questionnaire;

    private int numberOfCollisions = 0;

    private void OnTriggerEnter(Collider other)
    {
        numberOfCollisions += 1;
        if (numberOfCollisions <= 2)
        {
            Questionnaire.SetActive(true);
        }
        else
        {
            Questionnaire.SetActive(false);
        }
    }
}

```

Σχήμα 6.57: Script ChocolateCakeHandler

6.11 Επίλογος

Ολοκληρώνοντας αυτό το κεφάλαιο, έχει ολοκληρωθεί και η ανάλυση της Π.Ε από την πλευρά του περιεχομένου της, αλλά και τον τρόπο με τον οποίον συνδυάζει την μαγειρική με την θεωρία των μαθηματικών για την δημιουργία των σεναρίων της.

Κεφάλαιο 7ο: Τελικά συμπεράσματα και βελτιώσεις της Π.Ε.

Ολοκληρώνοντας την Π.Ε. καθώς και κατά την διάρκεια υλοποίησης της εφαρμογής, λήφθηκαν κάποια τελικά συμπεράσματα. Τα συμπεράσματα αυτά είναι πως:

- Η διδασκαλία ενός μαθήματος μπορεί να βελτιωθεί θεαματικά, τόσο από την πλευρά της κατανόησης του αντικειμένου, όσο και από την πλευρά της κινητοποίησης και διέγερσης του ενδιαφέροντος των μαθητών. Μάλιστα, η βελτίωση έρχεται σε μικρό κόστος καθώς απαιτεί μόνο την απλή εγκατάσταση μίας εφαρμογής και χρήσης ενός τυπωμένου QR κωδικού.
- Η διαδικασία της μάθησης γίνεται πολύ πιο διασκεδαστική, ειδικά σε παιδιά μικρής ηλικίας. Σήμερα, παρατηρείται ένα αυξητικό ενδιαφέρον καθώς και εξοικείωση των παιδιών προς τον τομέα της τεχνολογίας. Αυτό σε συνδυασμό με την μάθηση μπορεί να δώσει στον μαθητή μια πολύ μεγαλύτερη ικανοποίηση στην κατανόηση κάποιων εννοιών, καθώς επιβραβεύεται οπτικοακουστικά γι' αυτό.
- Η εξέλιξη της τεχνολογίας έχει δώσει πλέον αρκετά εργαλεία, ώστε ο καθένας να μπορεί να ασχοληθεί με την δημιουργία τέτοιων παιχνιδιών. Το κόστος υλοποίησης μπορεί εύκολα να είναι μηδαμινό, με μόνη απαίτηση μία μικρή εξοικείωση με τις τεχνολογίες που πρόκειται να χρησιμοποιηθούν.

Το παιχνίδι της Π.Ε. βρίσκεται σε έκδοση που μπορεί να χρησιμοποιηθεί εύκολα από μαθητές και εκπαιδευτικούς, όμως υπάρχουν κάποιες βελτιώσεις οι οποίες μπορούν να γίνουν ώστε το παιχνίδι να είναι πιο εύχρηστο, καθώς και πιο αποτελεσματικό. Οι βελτιώσεις αυτές είναι:

- Πρόσθεση δυνατότητας επεξεργασίας ασκήσεων σε περισσότερα σημεία του παιχνιδιού.
- Βελτίωση των γραφικών του παιχνιδιού από έμπειρα άτομα στην σχεδίαση 3D γραφικών.
- Δημιουργία εναλλακτικού μενού, στο οποίο αναγράφονται τα κεφάλαια των μαθηματικών αντί των συνταγών.
- Εμπλούτιση του μαγειρικού μέρους του παιχνιδιού.
- Δημιουργία περισσότερων σεναρίων για την καλύτερη κάλυψη της ύλης των μαθηματικών
Στ' Δημοτικού

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] S. Chi-Yin Yuen, G. Yaoyuneyong, E. Johnson, “Augmented Reality: An Overview and Five Directions for AR in Education” 2011, pp. 119-140.
- [2] H. F. Tobar, S. Baldiris, R. Fabregat, “ Augmented Reality Game-Based Learning: A Review of Applications and Design Approaches” 2017, pp 1-7
- [3] F. Palmas, G. Klinker, “Defining Extended Reality Training: A Long-Term Definition for All Industries” 2020, pp 322-324
- [4] J. M. Zheng, K. W. Chan, I. Gibson, “Virtual reality” 1998, pp 20-23
- [5] H. Aziz, “Virtual Reality Programs Applications in Healthcare” 2018, pp 1-3
- [6] S. Ozturkcan, “Service innovation: Using augmented reality in the IKEA Place app” 2020, pp 8-13
- [7] R. Looker, “Virtual reality enhances training for country EMTs”, *National Association of Countries*, 2019. [Online]. Available: [Virtual reality enhances training for county EMTs \(naco.org\)](http://naco.org)
- [8] A. D. Kaplan, J. Cruit, M. Endsley, S. M. Beers, B. D. Sawyer, P. A. Hancock “The effects of Virtual Reality, Augmented Reality, and Mixed Reality as Training Enhancement Methods: A Meta-Analysis” 2020, pp 1-21
- [9] Mixed Reality Microsoft Documentation. [Online]. Available: [What is mixed reality? - Mixed Reality | Microsoft Docs](#)
- [10] Unity [Online]. Available: [Unity Real-Time Development Platform | 3D, 2D VR & AR Engine](#)
- [11] Blender [Online]. Available: [blender.org - Home of the Blender project - Free and Open 3D Creation Software](http://blender.org)
- [12] Vuforia [Online]. Available: [Vuforia Developer Portal |](#)
- [13] ARMaker [Online]. Available: [ARMaker - Augmented Reality Marker Generator \(shawnlehner.github.io\)](http://shawnlehner.github.io)
- [14] Google Fonts [Online]. Available: [Browse Fonts - Google Fonts](#)
- [15] Visual Studio [Online]. Available: [Visual Studio: IDE and Code Editor for Software Developers and Teams \(microsoft.com\)](#)
- [16] W. Honig, C. Milanes, L. Scaria, T. Phan, M. Bolas, N. Ayanian, “Mixed Reality for Robotics” 2015, pp 1-6
- [17] Ο. Κασσώτη, Π. Κλιάπης, Θ. Οικονόμου, *Μαθηματικά Στ’ Δημοτικού*, Αθήνα: Ινστιτούτο Τεχνολογίας Υπολογιστών και Εκδόσεων «ΔΙΟΦΑΝΤΟΣ» 2009