

Αθήνα, Οκτώβριος 2021

Ανάπτυξη εκπαιδευτικού 3D
Puzzle με αντικείμενα
πολιτιστικής κληρονομιάς σε
περιβάλλον Εικονικής
Πραγματικότητας

Διπλωματική Εργασία

Ρουμανά Αντιγόνη – Μαρίνα

Επιβλέπων καθηγητής: Ανδρέας Γεωργόπουλος

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ & ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ – ΜΗΧΑΝΙΚΩΝ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΟΜΕΑΣ ΤΟΠΟΓΡΑΦΙΑΣ
ΕΡΓΑΣΤΗΡΙΟ ΦΩΤΟΓΡΑΜΜΕΤΡΙΑΣ



ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας την διπλωματική μου εργασία, θα ήθελα να ευχαριστήσω όσους με την βοήθεια και την υποστήριξή τους συνέβαλαν στην επιτυχή εκπόνησή της.

Αρχικά, οφείλω ένα μεγάλο ευχαριστώ στον καθηγητή κ. Ανδρέα Γεωργόπουλο, ο οποίος με στήριξε στην επιλογή του θέματος και με βοήθησε να φέρω εις πέρας την εργασία μου. Τίποτα από όλα αυτά δεν θα είχε επιτευχθεί χωρίς την υποστήριξή του, την καθοδήγησή του σε όλα τα στάδια της εργασίας και την εμπιστοσύνη του στην υλοποίηση της ιδέας.

Επιπλέον, θέλω να ευχαριστήσω τον Δρ. Ανέστη Κουτσούδη, διευθυντή ερευνών στο Ερευνητικό Κέντρο Αθηνά, για την πολύτιμη βοήθειά του σε όλα τα στάδια της εργασίας. Ήταν πάντα πρόθυμος να συνεισφέρει στην επίλυση των προβλημάτων που προέκυπταν και χωρίς την συνεισφορά του η εργασία δεν θα είχε αποκτήσει την επιθυμητή μορφή.

Στην συνέχεια, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους φίλους και συμφοιτητές μου που με υποστήριξαν και συμμετείχαν στην αξιολόγηση της εφαρμογής. Οι παρατηρήσεις τους βοήθησαν στην βελτίωση του παιχνιδιού και στον εντοπισμό προβλημάτων και οι ιδέες τους συνέβαλλαν στην απόκτηση της τελικής του μορφής.

Τέλος, θέλω να ευχαριστήσω την οικογένειά μου για την στήριξη και το ενδιαφέρον τους όλο αυτό το διάστημα.

ΠΕΡΙΛΗΨΗ

Ο βασικός στόχος της παρούσας διπλωματικής εργασίας είναι η δημιουργία ενός παιχνιδιού σοβαρού σκοπού με εκπαιδευτικό χαρακτήρα σε περιβάλλον Εικονικής Πραγματικότητας. Υποστηρίζει την χρήση τρισδιάστατων μοντέλων αντικειμένων πολιτιστικής κληρονομιάς για εκπαιδευτικούς σκοπούς, χρησιμοποιώντας μοντέλα που έχουν δημιουργηθεί σε προηγούμενες εργασίες.

Κατά την βιβλιογραφική ανασκόπηση δίνονται πολλοί ορισμοί για τα παιχνίδια σοβαρού σκοπού, τα οποία εν συντομία αποτελούν παιχνίδια που δεν έχουν την διασκέδαση ως βασικότερο στόχο τους αλλά την εκπαίδευση σε ποικίλα αντικείμενα. Μέχρι σήμερα έχουν χρησιμοποιηθεί κυρίως για διδασκαλία σε σχολεία και πανεπιστήμια, για εκπαίδευση ιατρών και στρατιωτικών αλλά και για την προσομοίωση προβλημάτων που χρήζουν αντιμετώπισης. Η πολιτιστική κληρονομιά αποτελεί κεντρικό χαρακτήρα των εκπαιδευτικών παιχνιδιών σοβαρού σκοπού με χρήση της ψηφιακής μορφής που εκείνη αποκτά προκειμένου να επιτευχθεί η αναβίωση και διατήρησή της. Η έννοια της Εικονικής Πραγματικότητας αναλύεται εκτενώς και γίνεται λόγος για τις εφαρμογές της σε σκοπούς πέρα από την διασκέδαση. Η ικανότητά της να εμβυθίζει το χρήστη σε ένα περιβάλλον που ο ίδιος αντιλαμβάνεται ως πραγματικό και που αλληλεπιδρά άμεσα με αυτό την καθιστά κατάλληλη για εκπαιδευτικούς σκοπούς, για επίλυση προβλημάτων υγείας όπως το μετατραυματικό στρες, καθώς και για την αρχιτεκτονική και τη πολεοδομία.

Με χρήση της μηχανής παιχνιδιών Unity 3D, της γλώσσας προγραμματισμού C# και του εξοπλισμού Εικονικής Πραγματικότητας HTC Vive και με κατάλληλη φωτογραμμετρική επεξεργασία των τρισδιάστατων μοντέλων δύο αντικειμένων δημιουργείται ένα παιχνίδι με τη μορφή παζλ, το οποίο οι παίκτες καλούνται να συναρμολογήσουν. Αποκτά εκπαιδευτικό χαρακτήρα καθώς προκειμένου να είναι εφικτή η συναρμολόγηση πρέπει να απαντηθούν σωστά ερωτήσεις που αφορούν στα αντικείμενα, τόσες όσες και τα κομμάτια στα οποία χωρίζονται. Η σωστή απάντηση επιτρέπει την μετακίνηση του κομματιού και την τοποθέτησή του στην σωστή θέση, ενώ με λάθος απάντηση εμφανίζεται καινούρια ερώτηση. Οι ερωτήσεις πηγάζουν από την ιστορία και την αρχιτεκτονική των αντικειμένων και για υποβοήθηση του παίκτη διατίθεται σε αυτόν πριν ξεκινήσει το παιχνίδι ένα σύντομο κείμενο με σχετικές πληροφορίες. Το παιχνίδι χωρίζεται σε δύο επίπεδα δυσκολίας, το εύκολο να αποτελείται από πέντε κομμάτια ενώ το δύσκολο από δέκα και ολοκληρώνεται όταν όλα τα κομμάτια είναι στη σωστή θέση.

Η ολοκληρωμένη εφαρμογή αξιολογείται ερασιτεχνικά ως προς την εφικτότητά της και την ικανότητά της να επιτύχει τον σκοπό που προσβέυει. Ειδικότερα, ερμηνεύεται η εμβύθιση στο περιβάλλον Εικονικής Πραγματικότητας κι αν αυτή επηρέασε αρνητικά την ικανότητα του παίκτη να ολοκληρώσει το παιχνίδι, κι ακόμη εάν οι πληροφορίες που παρέχονται σε

αυτόν και τα βοηθητικά εργαλεία καθιστούν την περαιώση του παιχνιδιού εφικτή. Η αξιολόγηση από την πλειονότητα των χρηστών είναι θετική, δηλώνοντας ενδιαφέρον για την χρήση της νέας τεχνολογίας της Εικονικής Πραγματικότητας και προθυμία για να δοκιμάσουν εφαρμογές παρόμοιου βεληνεκούς.

Τα αντικείμενα που επιλέγονται για την εφαρμογή είναι ο «Ναός της Δήμητρας» στη Νάξο και το άγαλμα «Αυτοκράτορας και Δούλος», που βρέθηκε στην αρχαία Αγορά της Αθήνας.

ABSTRACT

The main goal of this diploma thesis is the development of an educational Serious Games application in Virtual Reality environment, supporting the use of 3D Models of cultural assets for educational purposes. The models of the objects of cultural heritage that were selected have already been created and used in different projects.

Many definitions have been given to Serious Games, with the more inclusive one being “games which have been designed with a primary purpose other than entertainment, aiding users to achieve learning targets through a fun experience”. Until today, they have mostly been used for teaching at schools and universities, training in fields such as Health Care or the Military, as well as emulating real – life problems which require resolving. Cultural Heritage is used extensively for Serious Games applications in digital platforms, since more and more monuments and artifacts acquire digital form in order to be preserved and promoted. At the same time, Virtual Reality has seen a great increase in usage not only for entertainment but for Serious Games applications as well, due to its ability to immerse the user in an environment they interpret as the real one and with which they interact directly. Some of these applications worth mentioning are training future doctors in anatomy education, overcoming phobias and psychological trauma such as fear of heights or PTSD, along with architecture and design.

The application is developed using the game engine Unity 3D by coding in C# language, and the HTC Vive virtual reality system. The two cultural assets which are used were processed photogrammetrically to fix their surface and texture errors. The game created is a 3D puzzle which gets its educational quality by adding a type of choice quiz, with “True or False” questions deriving from the history and architecture of the two assets. In order for the player to move the puzzle pieces they have to correctly answer one question for each one of them, otherwise they cannot interact with them. If they select the wrong answer, a different question is generated and they can try again. To assist the players in answering the questions, at the beginning of the game a brief text is provided which includes information about each of the assets, depending on which one the player has chosen to proceed with. The game consists of two levels, easy (five pieces) and hard (ten pieces) and is complete when every piece is in its correct position.

Upon completion, the application is evaluated by other students on the degree of its feasibility and its ability to meet the goal it represents. More precisely, it is observed whether the immersion of Virtual Reality affects negatively the user’s ability to complete the game, due to the feelings of dizziness and isolation that usually come with the VR applications. Additionally, it is examined if the information and supplementary tools of the application made the completion of the game easier. The majority of users made positive remarks of the game, stating their interest in the growing technology of Virtual Reality and willingness to try similar applications in the future.

Due to lack of time, no professional evaluation was made, but it is suggested by the international literature.

The cultural assets that were selected to create the game are the “Temple of Demeter” in Naxos and the small statue “Emperor and Barbarian” which was found in Ancient Agora of Athens.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	1
1.1: Σκοπός της εργασίας	1
1.2: Διάρθρωση του κειμένου	2
ΚΕΦΑΛΑΙΟ 2: ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....	3
2.1: Βασικές έννοιες και ορισμοί	3
2.1.1: Πολιτιστική κληρονομιά.....	3
2.1.2: Serious Games	4
2.1.3: Εικονική Πραγματικότητα – Virtual Reality (VR).....	5
2.1.4: Επαυξημένη Πραγματικότητα – Augmented Reality (AR)	10
2.1.5: Μεικτή ή Υβριδική Πραγματικότητα – Mixed Reality (MR)	12
2.1.6: Εκτεταμένη Πραγματικότητα – Extended Reality (XR).....	13
2.2: Εφαρμογές.....	14
2.2.1: Serious Games	14
2.2.2: Εφαρμογές VR.....	17
2.2.3: Serious Games και VR στην πολιτιστική κληρονομιά	21
2.2.4: VR Serious Games	27
ΚΕΦΑΛΑΙΟ 3: ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ	29
3.1: Λογισμικά που χρησιμοποιήθηκαν	29
3.1.1: Unity Game Engine	29
3.1.2: Blender Game Engine	36
3.2: Προγράμματα επεξεργασίας τρισδιάστατων μοντέλων	38
3.3: Εξοπλισμός εικονικής πραγματικότητας HTC Vive	38
3.3.1: Πλεονεκτήματα & Μειονεκτήματα του συστήματος HTC Vive ^{[28][28]}	41
3.4: Πειραματική εφαρμογή.....	42
3.4.1: Κατάκτηση του αντικειμένου.....	42
3.4.2: Υλοποίηση του παιχνιδιού	43
ΚΕΦΑΛΑΙΟ 4: ΑΝΤΙΚΕΙΜΕΝΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	59
4.1: Ναός της Δήμητρας	59
4.1.1: Ιστορική ανασκόπηση	60
4.1.2: Αρχιτεκτονικά στοιχεία του ναού.....	61
4.1.3: Ο μύθος της θεάς Δήμητρας	62
4.1.4: Ο μύθος του θεού Απόλλωνα.....	63

4.2: Ο αυτοκράτορας και ο δούλος.....	65
4.2.1: Ιστορικά και αρχιτεκτονικά στοιχεία.....	65
4.2.2: Ο αυτοκράτορας.....	66
ΚΕΦΑΛΑΙΟ 5: ΑΝΑΛΥΣΗ ΤΟΥ ΠΑΙΧΝΙΔΙΟΥ.....	71
5.1: Σκοπός του παιχνιδιού	71
5.2: Προετοιμασία των αντικειμένων	71
5.2.1: Ο ναός της Δήμητρας.....	72
5.2.2: Το άγαλμα «Αυτοκράτορας και Δούλος».....	76
5.3: Διαδικασία δημιουργίας του παιχνιδιού	78
5.3.1: Σχηματισμός των κομματιών.....	78
5.3.2: Εισαγωγή στη μηχανή παιχνιδιών unity	78
5.3.3: Προσαρμογή του κώδικα C#	80
5.3.4: Δημιουργία των ερωτήσεων	86
5.3.5: Προσθήκη ηχητικών εφέ	94
5.3.6: Μετατροπή σε VR	96
5.4: Επίλυση προβλημάτων.....	123
5.4.1: Διόρθωση του κώδικα	123
5.4.2: Βελτιστοποίηση της εφαρμογής	126
ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ ΣΧΟΛΙΑ	127
6.1: Ανακεφαλαίωση της εφαρμογής	127
6.2: Αξιολόγηση Εφαρμογής.....	128
6.3: Επέκταση παρούσας εφαρμογής	130
ΒΙΒΛΙΟΓΡΑΦΙΑ	133
Δημοσιεύσεις	133
Ιστοσελίδες	135

Περιεχόμενα εικόνων

Εικόνα 1: Παραδείγματα κληρονομιάς. Στην πρώτη σειρά απεικονίζονται ακίνητα, στη δεύτερη φυσικά τοπία και στην τρίτη κινητά αντικείμενα	3
Εικόνα 2: Sketch up – walk: χαρακτηριστικό παράδειγμα παθητικού συστήματος. Μια κάμερα ακολουθεί μια διαδρομή που χαράζει ο χρήστης με τα βελάκια του πληκτρολογίου	6
Εικόνα 3: Συσκευή πολλαπλών σημείων αλληλεπίδρασης για απτική αντίληψη.....	8
Εικόνα 4: Κάμερα ανίχνευσης κινήσεων του κεφαλιού	8
Εικόνα 5: Εξοπλισμός VR με τη σειρά: φόρμες (data suits), γάντια (data gloves) και οθόνες προσαρμογής στο κεφάλι (HMD).....	9
Εικόνα 6: Σύστημα ηλεκτρονικού υπολογιστή με χειριστήρια (Oculus Quest 2)	9
Εικόνα 7: Σύστημα για κινητό τηλέφωνο (Google Daydream View)	10
Εικόνα 8: Παράδειγμα επαυξημένης πραγματικότητας στη διακόσμηση εσωτερικών χώρων	11
Εικόνα 9: Το παιχνίδι επαυξημένης πραγματικότητας Pokémon Go	11
Εικόνα 10: Το συνεχές της μεικτής πραγματικότητας	12
Εικόνα 11: Αλληλεπίδραση υπολογιστή, ανθρώπου και περιβάλλοντος	13
Εικόνα 12: Στιγμιότυπο από το παιχνίδι “America’s Army”	14
Εικόνα 13: Στιγμιότυπο από το παιχνίδι “Food Force”	15
Εικόνα 14: Στιγμιότυπο από το παιχνίδι “Wu’s Castle”	16
Εικόνα 15: Στιγμιότυπο από το παιχνίδι “Making History”	16
Εικόνα 16: Ενδεικτική χρήση της ΕΠ στην αρχιτεκτονική εσωτερικού χώρου, Redshift.....	17
Εικόνα 17: Εξομοιωτές μάχης για εκπαίδευση στρατιωτικών.....	18
Εικόνα 18: Ανάπτυξη συστήματος ΕΠ για αντιμετώπιση υψοφοβίας.....	19
Εικόνα 19: Ανάπτυξη συστήματος ΕΠ για αντιμετώπιση κοινωνικών φοβιών με εξομίωση δημόσιας ομιλίας	19
Εικόνα 20: εργαστήριο επ, nasa	20
Εικόνα 21: Εικονική περιήγηση και ανακατασκευή του εσωτερικού χώρου μιας οικείας σε χαλκολιθικό οικισμό με δυνατότητες 3D πλοήγησης και μεταγωγής στον προϊστορικό χρόνο με βάση τα ευρήματα του οικισμού στην Alba Iouliia της Ρουμανίας	20
Εικόνα 22: Στιγμιότυπα από την εφαρμογή “Roma Nova”.....	21
Εικόνα 23: Το θέατρο της Πομπηίας.....	22
Εικόνα 24: Παραδείγματα αλληλεπίδρασης χρήστη με εκθέματα στο google’s arts and culture.....	24
Εικόνα 25: Blob opera, machine learning experiment	25
Εικόνα 26: Αιγυπτιακός Ναός της Περιόδου Νέου Βασιλείου	25
Εικόνα 27: Το κύριο menu του παιχνιδιού, (Kontogianni, Georgia The Contribution of 3D Models to Serious Games Applications).....	26
Εικόνα 28: Η «Θόλος» του Ελληνικού Κόσμου τη νύχτα	27
Εικόνα 29: Παράδειγμα επιλογής κομματιού και προβολής του ονόματός του – Supporting Anatomy Education with a 3D puzzle in a Virtual Reality Environment.....	28
Εικόνα 30: Παράδειγμα της προς αφοπλισμό βόμβας σε ένα από τα σενάρια.....	28
Εικόνα 31: Unity Game Engine Logo	29
Εικόνα 32: Η επιφάνεια εργασίας της μηχανής παιχνιδιών Unity	30
Εικόνα 33: Παράδειγμα sprites στη Unity Game Engine	30
Εικόνα 34: Παράδειγμα mirror της Unity Game Engine.....	30
Εικόνα 35: Το ρομπότ Adam από την ταινία μικρού μήκους της Unity.....	35
Εικόνα 36: Σκηνικό της ταινίας Lion King (2019) που δημιουργήθηκε στην Unity.....	35
Εικόνα 37: Στιγμιότυπο από την δημιουργία της ταινίας Lion King σε περιβάλλον ΕΠ.....	35
Εικόνα 38: Blender Logo.....	36

Εικόνα 39: Παράδειγμα χρήσης των λογικών τούβλων στο περιβάλλον εργασίας της Blender και του προγραμματισμού με Python.....	36
Εικόνα 40: Το logo του παιχνιδιού Yo Frankie!.....	37
Εικόνα 41: Το logo του παιχνιδιού Sintel.....	37
Εικόνα 42: Το σύστημα εικονικής πραγματικότητας HTC Vive.....	39
Εικόνα 43: vive tracker.....	40
Εικόνα 44: vive deluxe audio strap.....	40
Εικόνα 45: vive facial tracker.....	41
Εικόνα 46: Τα 7 κομμάτια στα οποία χωρίστηκε το πειραματικό αντικείμενο.....	42
Εικόνα 47: Τα κομμάτια αποθηκεύτηκαν σε μορφή .obj.....	43
Εικόνα 48: Περιεχόμενα του καταλόγου σε μία νέα σκηνή.....	43
Εικόνα 49: Τα 7 κομμάτια του κύβου όπως εισήχθησαν στο περιβάλλον της Unity.....	43
Εικόνα 50: Η μορφή της ετικέτας των κομματιών.....	44
Εικόνα 51: Η θέση του κύβου στο τρισδιάστατο περιβάλλον της Unity.....	44
Εικόνα 52: η μορφή του κύβου πριν (αριστερά) και μετά (μεσαία και δεξιά) την τυχαιοποίηση.....	47
Εικόνα 53: Η λογική λειτουργίας της ακτίνας στην Unity.....	48
Εικόνα 54: Μορφή του πίνακα με τις νέες θέσεις των κομματιών.....	51
Εικόνα 55: Η ταχύτητα περιστροφής των κομματιών.....	52
Εικόνα 56: Τυχαίες τιμές Quaternion κατά τη στροφή ενός κομματιού (πάνω) και η απόκλιση του σε μοίρες από την αρχική γωνία στροφής (κάτω).....	54
Εικόνα 57: Rigidbody component.....	57
Εικόνα 58: Box Collider.....	58
Εικόνα 59: Ο Ναός της Δήμητρας στην Νάξο.....	59
Εικόνα 60: Το εσωτερικό του μουσείου Γύρουλα.....	60
Εικόνα 61: Μοντέλο του Ναού της Δήμητρας.....	62
Εικόνα 62: Η αρπαγή της Περσεφόνης από τον Πλούτωνα (Lorenzo Bernini, 1621).....	63
Εικόνα 63: Ο θεός Απόλλωνας, άγαλμα στην Ακαδημία Αθηνών.....	64
Εικόνα 64: Το άγαλμα «Αυτοκράτορας και Δούλος».....	65
Εικόνα 65: Κομμάτι του μανδύα του αυτοκράτορα που έχει διασωθεί.....	66
Εικόνα 66: Ο αυτοκράτορας Τραϊανός.....	67
Εικόνα 67: Ο αυτοκράτορας Αδριανός.....	67
Εικόνα 68: Το Πάνθεον με τη μορφή που απέκτησε μετά την ανακατασκευή του από τον Αδριανό.....	68
Εικόνα 69: Η βίλλα του Αδριανού στο Τίβολι.....	68
Εικόνα 70: Η βιβλιοθήκη του Αδριανού στην Αθήνα.....	69
Εικόνα 71: Η αρχική μορφή του τρισδιάστατου μοντέλου του ναού της Δήμητρας.....	73
Εικόνα 72: Η μορφή του ναού της Δήμητρας μετά την επεξεργασία.....	73
Εικόνα 73: Το μοντέλο του ναού της Δήμητρας μετά την απόδοση υλικού.....	74
Εικόνα 74: Το άγαλμα «Αυτοκράτορας και Δούλος» μετά την επεξεργασία.....	77
Εικόνα 75: Τα δύο αντικείμενα «αυτοκράτορας και δούλος» (πάνω) και «Ναός της Δήμητρας» (κάτω) όπως εισάγονται στο περιβάλλον της unity.....	79
Εικόνα 76: Τα 10 κομμάτια του ναού όπως εισήχθησαν στο περιβάλλον της Unity.....	79
Εικόνα 77: Τα 5 κομμάτια του αγάλματος όπως εισήχθησαν στο περιβάλλον της Unity.....	79
Εικόνα 78: Τυχαίες θέσεις των κομματιών του ναού όπως φαίνονται στο περιβάλλον της unity (πάνω) και οι συντεταγμένες τους (κάτω).....	83
Εικόνα 79: Τυχαίες θέσεις των κομματιών του αγάλματος όπως φαίνονται στο περιβάλλον της unity (πάνω) και οι συντεταγμένες τους (κάτω).....	83
Εικόνα 80: Το μαθηματικό μοντέλο Quaternion για το άγαλμα.....	84
Εικόνα 81: Τα αντικείμενα κειμένου.....	86
Εικόνα 82: Τα χαρακτηριστικά που αποδίδονται σε κάθε αντικείμενο κειμένου.....	87

Εικόνα 83: Οι παράμετροι του ηχητικού εφέ για τη δημιουργία των κομματιών	95
Εικόνα 84: Οι παράμετροι της μουσικής υποβάθρου	96
Εικόνα 85: Το πρόσθετο steamvr.....	97
Εικόνα 86: Το αντικείμενο που ορίζει τον παίκτη σε ένα VR παιχνίδι	98
Εικόνα 87: Steam vr input	99
Εικόνα 88: Binding Ui	99
Εικόνα 89: Διαθέσιμα πλήκτρα των χειριστηρίων.....	100
Εικόνα 90: Το άγαλμα σε περιβάλλον Εικονικής Πραγματικότητας.....	100
Εικόνα 91: Ο ναός σε περιβάλλον Εικονικής Πραγματικότητας.....	101
Εικόνα 92: Παράδειγμα της τηλεμεταφοράς του παίκτη σε επιτρεπτά σημεία (πάνω) και μη επιτρεπτά (κάτω)...	102
Εικόνα 93: Εξάρτημα που χρησιμοποιείται για την κίνηση του παίκτη στη σκηνή.....	103
Εικόνα 94: Καθορισμός της ενέργειας στο περιβάλλον της Unity.....	105
Εικόνα 95: Παράδειγμα τυχαιοποίησης των θέσεων των κομματιών των δύο αντικειμένων	105
Εικόνα 96: Interactable και Throwable εξαρτήματα.....	106
Εικόνα 97: Τα layer με τα οποία μπορεί να αλληλεπιδράσει ο χρήστης.....	107
Εικόνα 98: Εμφάνιση της ερώτησης για ένα κομμάτι του αγάλματος όταν πλησιάζει ο χρήστης το χέρι του σε αυτό	108
Εικόνα 99: Οι ερωτήσεις για το άγαλμα	109
Εικόνα 100: Μορφή του πλαισίου των ερωτήσεων	109
Εικόνα 101: Φόρτωση συνάρτησης για το σωστό (πάνω) και το λάθος (κάτω).....	111
Εικόνα 102: Παράδειγμα εμφάνισης ερωτήσεων για το άγαλμα (αριστερά) και το ναό (δεξιά)	112
Εικόνα 104: Παράδειγμα λάθος απάντησης σε ερώτηση για το άγαλμα (αριστερά) και για το ναό (δεξιά)	113
Εικόνα 103: Παράδειγμα σωστής απάντησης σε ερώτηση για το άγαλμα (αριστερά) και το ναό (δεξιά).....	113
Εικόνα 105: Εμφάνιση μηνύματος «Τέλος Παιχνιδιού».....	115
Εικόνα 106: Αρχικό μενού επιλογής	116
Εικόνα 107: Τα εισαγωγικά κείμενα για το ναό και το άγαλμα.....	117
Εικόνα 108: Build Settings του παιχνιδιού.....	118
Εικόνα 109: Το μοντέλο που χρησιμοποιείται για το αριστερό χέρι.....	119
Εικόνα 110: Το άγαλμα - φάντασμα	120
Εικόνα 111: Ο ναός – φάντασμα.....	120
Εικόνα 112: Μενού υπενθύμισης χειρισμών.....	121
Εικόνα 113: Χειρισμοί	122

Περιεχόμενα Πινάκων

Πίνακας 1: Ψηφιακές αναπαραστάσεις της Ακρόπολης των Αθηνών ανά ιστορικές περιόδους.....	22
Πίνακας 2: Γνωστά παιχνίδια που αναπτύχθηκαν σε περιβάλλον Unity.....	31
Πίνακας 3: Κομμάτι του κώδικα για την δημιουργία του πίνακα θέσεων.....	45
Πίνακας 4: Η διαδικασία τυχαιοποίησης των κομματιών.....	47
Πίνακας 5: Εκκίνηση της τυχαιοποίησης με το <i>space</i>	47
Πίνακας 6: Ανίχνευση του αντικειμένου με χρήση ακτινών.....	49
Πίνακας 7: Ανίχνευση κίνησης ποντικιού και εισαγωγή νέων συντεταγμένων για κάθε κομμάτι που επιλέγεται ...	49
Πίνακας 8: Ανίχνευση αριστερού κλικ και κίνησης ποντικιού.....	50
Πίνακας 9: Έλεγχος της απόστασης από την <i>default</i> θέση των κομματιών.....	51
Πίνακας 10: Τυχαιοποίηση θέσεων και προσανατολισμού των κομματιών.....	52
Πίνακας 11: Ανίχνευση περιστροφής ποντικιού και στρέψη του κομματιού κατάλληλα.....	53
Πίνακας 12: Ολοκληρωμένος έλεγχος για δεξιά ή αριστερό κλικ και επίκληση αντίστοιχης συνάρτησης.....	53
Πίνακας 13: Έλεγχος στροφής από τον <i>default</i> προσανατολισμό των κομματιών.....	54
Πίνακας 14: Στροφή της κάμερας γύρω από το αντικείμενο.....	55
Πίνακας 15: Τερματισμός της περιστροφής με το <i>space</i>	55
Πίνακας 16: Στροφή της κάμερας με το πληκτρολόγιο.....	56
Πίνακας 17: <i>Zoom in</i> και <i>zoom out</i> με την ροδέλα του ποντικιού.....	56
Πίνακας 18: Μετακίνηση της κάμερας με τα πλήκτρα <i>WASD</i>	57
Πίνακας 19: Υλικά που χρησιμοποιήθηκαν στον ναό της Δήμητρας.....	75
Πίνακας 20: Ο κώδικας για την περιστροφή της κάμερας.....	81
Πίνακας 21: Ο κώδικας τυχαιοποίησης θέσεων και προσανατολισμών για το ναό.....	82
Πίνακας 22: Ο κώδικας τυχαιοποίησης θέσεων και προσανατολισμών για το άγαλμα.....	82
Πίνακας 23: Έλεγχος της θέσης των κομματιών για το ναό και αυτόματη τοποθέτησή τους στην σωστή.....	85
Πίνακας 24: Έλεγχος της γωνίας στροφής κατά την περιστροφή των κομματιών.....	85
Πίνακας 25: Οι ερωτήσεις και απαντήσεις για το Ναό της Δήμητρας.....	87
Πίνακας 26: Οι ερωτήσεις και απαντήσεις για το άγαλμα.....	88
Πίνακας 27: Οι ερωτήσεις και απαντήσεις για τα δύο αντικείμενα.....	88
Πίνακας 28: Οι δύο λίστες που καθορίζουν την πρώτη (σωστό) και την δεύτερη (λάθος) επιλογή σε κάθε ερώτηση.....	89
Πίνακας 29: Εμφάνιση της ερώτησης στο άγαλμα.....	90
Πίνακας 30: Εμφάνιση της ερώτησης στο ναό.....	91
Πίνακας 31: Ενεργοποίηση του κώδικα που περιέχει την εμφάνιση των ερωτήσεων εντός του <i>Drag and Drop</i>	91
Πίνακας 32: Η συνάρτηση <i>Awake</i> του κώδικα " <i>Text Control</i> ".....	92
Πίνακας 33: Η διαδικασία που εκτελείται όταν ο χρήστης απαντάει σωστά σε μια ερώτηση.....	92
Πίνακας 34: Η διαδικασία που εκτελείται όταν ο χρήστης απαντάει λάθος σε μια ερώτηση.....	93
Πίνακας 35: Η διαδικασία επιλογής της ερώτησης που θα εμφανιστεί στον ναό.....	94
Πίνακας 36: Κλήση του ήχου <i>breaking</i>	95
Πίνακας 37: Ορισμός του ήχου <i>matching</i>	95
Πίνακας 38: Τυχαιοποίηση των θέσεων των κομματιών του αγάλματος.....	104
Πίνακας 39: Τυχαιοποίηση των θέσεων των κομματιών του ναού.....	104
Πίνακας 40: Καθορισμός του πλήκτρου που πραγματοποιεί την συγκεκριμένη ενέργεια.....	104
Πίνακας 41: Η αλλαγή του <i>layer</i> του κομματιού.....	107
Πίνακας 42: Τοποθέτηση των κομματιών στην αρχική θέση και αλλαγή του <i>layer</i>	107
Πίνακας 43: Εμφάνιση των ερωτήσεων και επιστροφή εάν έχει ήδη εμφανιστεί ερώτηση για αυτό το κομμάτι... ..	108
Πίνακας 44: Διαχείριση των ερωτήσεων και απαντήσεων.....	110

Πίνακας 45: Κομμάτι κώδικα για την ανίχνευση της θέσης της ακτίνας και την αλληλεπίδρασή της με κουμπιά..	112
Πίνακας 46: Έλεγχος τέλους παιχνιδιού	114
Πίνακας 47: Έξοδος από το παιχνίδι.....	115
Πίνακας 48: Παράδειγμα του κώδικα που εμφανίζει τη σκηνή	118
Πίνακας 49: Εμφάνιση των αντικειμένων – φάντασμα με χρήση της σκανδάλης.....	121
Πίνακας 50: Εμφάνιση του μενού υπενθύμισης πλήκτρων με το πάτημα του αντίστοιχου κουμπιού.....	122
Πίνακας 51: Απλοποίηση <code>for loop</code>	124
Πίνακας 52: Βελτιστοποίηση <code>Update()</code>	125
Πίνακας 53: Διορθωμένη εμφάνιση ερώτησης	126

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1: Σκοπός της εργασίας

Σκοπό της παρούσας διπλωματικής εργασίας αποτελεί η ανάπτυξη ενός εκπαιδευτικού τρισδιάστατου puzzle με χρήση αντικειμένων πολιτιστικής κληρονομιάς σε περιβάλλον Εικονικής Πραγματικότητας.

Τα παιχνίδια σοβαρού σκοπού (serious games) εισάγονται όλο και περισσότερο στην εκπαιδευτική διαδικασία με διάφορες μορφές επιτυγχάνοντας έναν διαφορετικό τρόπο μάθησης μέσω μιας διασκεδαστικής εμπειρίας (edutainment). Η Εικονική Πραγματικότητα (Virtual Reality – VR) χρησιμοποιείται εκτενώς για αυτόν τον σκοπό, καθώς τοποθετεί τον χρήστη σε ένα περιβάλλον που ο ίδιος αντιλαμβάνεται ως πραγματικό, και αλληλοεπιδρά άμεσα με αυτό μέσω ηλεκτρονικών συσκευών. Ταυτόχρονα, η ανάγκη διατήρησης και αναβίωσης της πολιτιστικής κληρονομιάς έχει οδηγήσει στη δημιουργία τρισδιάστατων μοντέλων χώρων και αντικειμένων, με στόχο την φωτορεαλιστική απόδοση στοιχείων του πολιτισμού συγκεκριμένων ιστορικών εποχών για ακαδημαϊκούς, κυρίως, σκοπούς. Συνδυάζοντας τα παραπάνω, δημιουργούνται εφαρμογές σε περιβάλλον Εικονικής Πραγματικότητας με απώτερο σκοπό την εκπαίδευση χρηστών πάνω σε αντικείμενα πολιτιστικής κληρονομιάς και την απόδοση αρχαιολογικών χώρων με την πρότερη μορφή τους.

Στη συγκεκριμένη εφαρμογή, χρησιμοποιήθηκαν δύο τρισδιάστατα μοντέλα αντικειμένων πολιτιστικής κληρονομιάς, τα οποία βελτιστοποιήθηκαν με κατάλληλες μεθόδους φωτογραμμετρικής επεξεργασίας. Στη συνέχεια, δημιουργήθηκε ένα παιχνίδι σε περιβάλλον Εικονικής Πραγματικότητας με τη μορφή ενός παζλ, με σκοπό την εκμάθηση πληροφοριών για τα συγκεκριμένα αντικείμενα. Ο χρήστης, διαβάζοντας μια σύντομη περιγραφή των αντικειμένων και απαντώντας σε σχετικές ερωτήσεις καλείται να ενώσει τα κομμάτια που αποτελούν κάθε αντικείμενο για να ολοκληρώσει το παιχνίδι.

1.2: Διάρθρωση του κειμένου

Η παρούσα διπλωματική εργασία εκτείνεται σε έξι κεφάλαια, με την εξής διάρθρωση:

- Στο Κεφάλαιο 1, δηλαδή το παρόν κεφάλαιο, γίνεται μια εισαγωγή για τον σκοπό της διπλωματικής εργασίας και αναλύεται η διάρθρωση όλων των κεφαλαίων.
- Στο Κεφάλαιο 2 γίνεται βιβλιογραφική ανασκόπηση, δηλαδή αναλύονται έννοιες που αναφέρονται στην εργασία και εφαρμογές και δημοσιεύσεις σχετικές με τα παιχνίδια σοβαρού σκοπού, την Εικονική Πραγματικότητα και την πολιτιστική κληρονομιά.
- Στο Κεφάλαιο 3 επεξηγούνται τα λογισμικά που χρησιμοποιήθηκαν για την παρούσα εφαρμογή, τόσο για την φωτογραμμετρική επεξεργασία των αντικειμένων όσο και για την σύσταση του παιχνιδιού και την μετατροπή του σε Εικονικής Πραγματικότητας καθώς και ο εξοπλισμός Εικονικής Πραγματικότητας που χρησιμοποιήθηκε. Επιπλέον, αναλύεται η πειραματική εφαρμογή που δημιουργήθηκε, η οποία χρησιμοποιεί ένα απλούστερο αντικείμενο και αποτελεί την βάση για το τελικό παιχνίδι.
- Στο Κεφάλαιο 4 πραγματοποιείται μια ανάλυση των ιστορικών και αρχιτεκτονικών στοιχείων των αντικειμένων που χρησιμοποιούνται στην διπλωματική εργασία.
- Στο Κεφάλαιο 5 αναλύεται η διαδικασία δημιουργίας του τελικού παιχνιδιού, από την επεξεργασία των τρισδιάστατων μοντέλων, την δημιουργία των κομματιών του παζλ, την σύνταξη ενός απλού τρισδιάστατου παιχνιδιού και τέλος την μετατροπή του σε παιχνίδι Εικονικής Πραγματικότητας. Επιπλέον, αναφέρονται όλες οι ερωτήσεις που γίνονται στον χρήστη κατά τη διάρκεια του παιχνιδιού καθώς και τα κείμενα τα οποία θα πρέπει να διαβάσει προκειμένου να απαντήσει σε αυτές.
- Στο Κεφάλαιο 6 γίνεται μια ανακεφαλαίωση της διαδικασίας δημιουργίας του παιχνιδιού. Επιπλέον, παρατίθεται μια σύντομη αξιολόγηση της εφαρμογής και αναφέρονται πιθανές μελλοντικές επεκτάσεις ή τροποποιήσεις της, ώστε να βελτιστοποιηθεί.

ΚΕΦΑΛΑΙΟ 2: Βιβλιογραφική ανασκόπηση

2.1: Βασικές έννοιες και ορισμοί

2.1.1: Πολιτιστική κληρονομιά

Η πολιτιστική κληρονομιά αποτελεί μια έκφραση του τρόπου ζωής μιας κοινότητας που κληροδοτείται από γενιά σε γενιά και περιλαμβάνει ήθη, έθιμα, πρακτικές, αντικείμενα και καλλιτεχνικές εκφράσεις. Μπορεί να διακριθεί σε^[12]:

- **υλική πολιτιστική κληρονομιά**, η οποία περιλαμβάνει κινητά και ακίνητα αντικείμενα τόσο στη στεριά όσο και υποθαλάσσια. Μερικά παραδείγματα κινητής πολιτιστικής κληρονομιάς είναι οι πίνακες ζωγραφικής, τα κείμενα που έχουν σωθεί από τα αρχαία χρόνια, τα νομίσματα και τα αγάλματα. Τα ακίνητα αντικείμενα μπορεί να είναι μνημεία, αρχαιολογικοί χώροι και ναοί ενώ υποθαλάσσια συναντώνται ναυάγια ή βυθισμένες πόλεις.
- **άυλη πολιτιστική κληρονομιά**, με την οποία περιγράφονται τα ήθη, τα έθιμα, η γλώσσα και οι τελετουργίες μιας κοινωνίας που έχουν διαδοθεί από γενιά σε γενιά.
- **φυσική πολιτιστική κληρονομιά**, δηλαδή βιολογικοί ή γεωλογικοί σχηματισμοί του εδάφους, τοπία με ιδιαίτερα χαρακτηριστικά, και γενικότερα η βιοποικιλότητα και γεωποικιλότητα ενός τόπου.



Εικόνα 1: Παραδείγματα κληρονομιάς. Στην πρώτη σειρά απεικονίζονται ακίνητα, στη δεύτερη φυσικά τοπία και στην τρίτη κινητά αντικείμενα

Σύμφωνα με τον ορισμό που έχει δοθεί από την UNESCO το 1989^[1] η Πολιτιστική Κληρονομιά είναι το συνολικό έργο των υλιστικών υπογραφών, καλλιτεχνικών ή συμβολικών, που κληροδοτήθηκαν σε κάθε κουλτούρα από το παρελθόν και ως συνέπεια σε όλη την ανθρωπότητα. Ως συστατικό στοιχείο για την επιβεβαίωση και τον εμπλουτισμό της πολιτιστικής ταυτότητας ως κληροδότημα ολόκληρης της ανθρωπότητας, η πολιτιστική κληρονομιά δίνει σε κάθε συγκεκριμένο μέρος τα διακριτά χαρακτηριστικά του και είναι η αποθήκη των ανθρωπινων εμπειριών.

2.1.2: Serious Games

Ως “serious games” ορίζονται τα ηλεκτρονικά και άλλα παιχνίδια και τρισδιάστατες εφαρμογές που έχουν κυρίως εκπαιδευτικό χαρακτήρα, βοηθώντας τον χρήστη να επιτύχει στόχους μάθησης μέσω μιας διασκεδαστικής εμπειρίας. Ο απλούστερος ορισμός που δίνεται στα παιχνίδια σοβαρού σκοπού είναι «παιχνίδια που δεν έχουν τη διασκέδαση ως βασικότερο στόχο τους» (Michael, et al., 2006). Δύο ακόμη ορισμοί που έχουν δοθεί είναι ότι πρόκειται για μια εγκεφαλική δοκιμασία που πραγματοποιείται σε υπολογιστή ακολουθώντας συγκεκριμένους κανόνες, η οποία χρησιμοποιεί την διασκέδαση ως επιπρόσθετη παράμετρο της εξάσκησης σε ακαδημαϊκά ή ιδιωτικά περιβάλλοντα στους τομείς της εκπαίδευσης, της υγείας, της πολιτικής προστασίας και της στρατηγικής επικοινωνίας (Zyda, 2005; Tarja, et al., 2007) ή αλλιώς «εφαρμογή υπολογιστή ο σκοπός της οποίας είναι να συνδυάσει ταυτόχρονα σημαντικά αντικείμενα όπως η διδασκαλία, η μάθηση, η επικοινωνία ή η πληροφόρηση, με όσα προσφέρει ένα video game» (Alvarez & Michaud, 2008).

Τα παιχνίδια σοβαρού σκοπού συναντώνται σε μεγάλο βαθμό στο ακαδημαϊκό περιβάλλον τόσο για την βελτίωση της μάθησης όσο και για την διαμόρφωση χαρακτήρα και χρησιμοποιούνται ευρέως στην διδασκαλία μαθημάτων σε σχολεία και πανεπιστήμια, στην ιατρική για αποκατάσταση τραυματισμών μεταφράζοντας κινήσεις του σώματος σε κινήσεις εντός του παιχνιδιού και για εκμάθηση της ανατομίας του ανθρωπίνου σώματος και την εξάσκηση ιατρικών μεθόδων σε προσομοιωτικό περιβάλλον, στην εκπαίδευση εργαζομένων σε νέους τομείς και τέλος στην ευαισθητοποίηση των χρηστών σε θέματα βιωσιμότητας, όπως την προστασία του περιβάλλοντος και της πολιτιστικής κληρονομιάς. Ειδικότερα, τα SG κατατάσσονται σε πέντε κατηγορίες^[2]:

- Μεταβίβασης γνώσης
- Εκμάθησης δεξιοτήτων
- Επαγρύπνησης
- Αλλαγής συμπεριφοράς
- Κινητοποίησης

Ένα ολοκληρωμένο SG χαρακτηρίζεται από: (Andreoli, et al., 2018)

- Αποτελεσματική απόκτηση της γνώσης
- Ικανοποίηση από τον χρήστη
- Χρησιμότητα
- Δυνατότητα αναπαραγωγής

Για τη σωστή λειτουργία ενός SG είναι απαραίτητη και η διασκέδαση του χρήστη. Εδώ έγκειται και η σημαντικότερη διαφορά ανάμεσα στα SG και τα κοινά παιχνίδια, δηλαδή ότι τα πρώτα εστιάζουν στην επίλυση προβλημάτων και τα στοιχεία που είναι απαραίτητα να κατακτήσουν οι χρήστες με στόχο να μάθουν ένα αντικείμενο, ενώ τα δεύτερα έχουν ως απώτερο στόχο την διασκέδαση και άρα απαιτούν λιγότερο περίπλοκες προσομοιώσεις.

Οι μέχρι τώρα εφαρμογές έχουν αναδείξει τα σημαντικότερα πλεονεκτήματα των SG, με κυριότερο την τεταμένη ασχολία του χρήστη με το αντικείμενο μελέτης του. Όπως όλα τα παιχνίδια, έτσι και τα SG χρησιμοποιούν «ανταμοιβές» και διαρκή επιβράβευση για τις νέες γνώσεις που αποκτώνται έτσι ώστε να κρατήσουν τον παίκτη απασχολημένο και κινητοποιημένο να συνεχίσει. Ως αποτέλεσμα ο χρήστης δε χάνει την επαφή με το αντικείμενο και το μελετά εις βάθος. Επιπλέον, το εικονικό περιβάλλον των παιχνιδιών τα καθιστά ασφαλή για πειραματισμούς οποιασδήποτε φύσης, αφού οι συνέπειες πιθανών λαθών δεν επηρεάζουν την πραγματικότητα και μπορεί έτσι να βρεθεί η καλύτερη δυνατή λύση σε ένα πρόβλημα (trial and error). Τέλος, είναι αποδεδειγμένο πως η εκμάθηση με διασκεδαστικό τρόπο οδηγεί σε γρηγορότερη και αποτελεσματικότερη απόκτηση γνώσης, είτε αφορά μαθητές είτε εργαζόμενους.

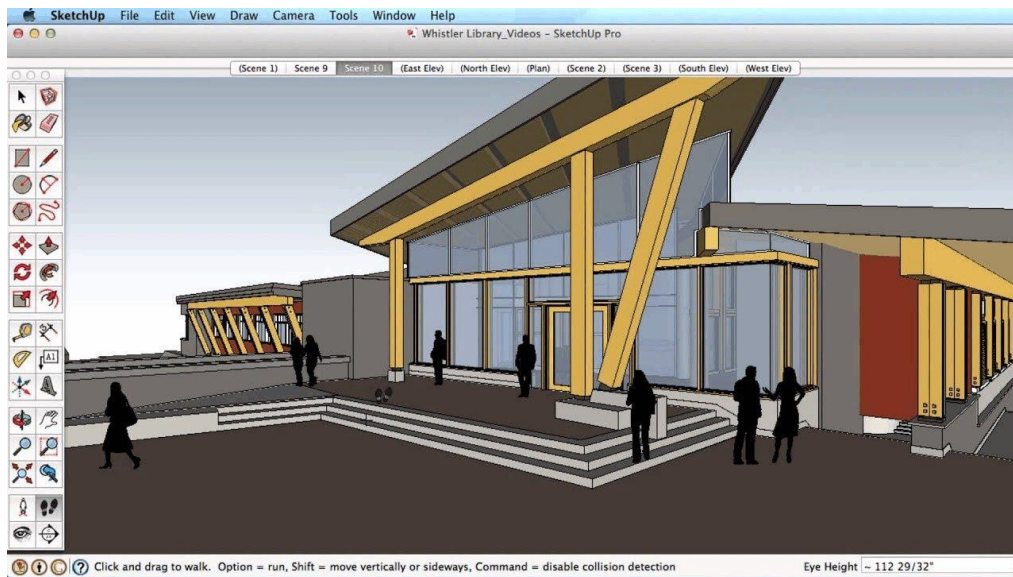
Ωστόσο, υπάρχουν ακόμα αρκετές αμφιβολίες γύρω από τη χρήση των SG, οι οποίες πηγάζουν κυρίως από τις ψυχολογικές συνέπειες που μπορεί να έχει η εκτεταμένη έκθεση του χρήστη στο περιβάλλον των παιχνιδιών και την πιθανότητα να επηρεάσει αυτό την παραγωγικότητά του.

2.1.3: Εικονική Πραγματικότητα – Virtual Reality (VR)

Η Εικονική Πραγματικότητα (VR)^[3] αναφέρεται στη χρήση αληθοφανών προσομοιώσεων που βασίζονται σε υπολογιστή μέσω των οποίων ο χρήστης έρχεται σε επαφή με ένα τεχνητό τρισδιάστατο περιβάλλον χρησιμοποιώντας ηλεκτρονικές συσκευές. Το περιβάλλον στη βέλτιστη μορφή του το αντιλαμβάνεται ο χρήστης ως πραγματικό και σε πραγματικό χρόνο το τροποποιεί. Η βασικότερη έννοια της εικονικής πραγματικότητας^[4] είναι η **εμβύθιση** (immersion) η οποία ορίζεται ως η ψευδαίσθηση που έχει ο χρήστης αναφορικά με την ύπαρξή του στο εικονικό περιβάλλον και αποτελεί μέτρο αξιολόγησης της επιτυχίας του συστήματος Εικονικής Πραγματικότητας ανάλογα με τον βαθμό που καταφέρνει αυτό να

απομονώσει τον χρήστη από το εξωτερικό περιβάλλον του. Άλλες σημαντικές έννοιες είναι η **αλληλεπίδραση** (interaction), δηλαδή η δυνατότητα του υπολογιστή να διαμορφώνει τον ψηφιακό κόσμο ανάλογα με τις κινήσεις του χρήστη και η **ένταση πληροφορίας** (information intensity), που αφορά στην πληθώρα της πληροφορίας και την ποικιλία των διαύλων επικοινωνίας που παρέχουν την πληροφορία στον χρήστη.

Τα συστήματα Εικονικής πραγματικότητας χωρίζονται ανάλογα με τον βαθμό που επιτρέπουν στον χρήστη να αλληλεπιδράσει με το περιβάλλον του, σε **ενεργητικά**, κατά τα οποία ο χρήστης ασκεί ένα επίπεδο ελέγχου και **παθητικά**, στα οποία ο χρήστης μετακινείται στο εσωτερικό ενός εικονικού κόσμου χωρίς τη δυνατότητα άσκησης ελέγχου (Παγουλάτου, et al., 2017). Τα ενεργητικά συστήματα χωρίζονται στα εξερευνητικά περιβάλλοντα, τα οποία χαρακτηρίζονται από ελεύθερη κίνηση της κάμερας χωρίς όμως δυνατότητα αλληλεπίδρασης με τα αντικείμενα και στα αλληλεπιδραστικά περιβάλλοντα, τα οποία παρέχουν όλες τις δυνατότητες των ενεργητικών συστημάτων με επιπλέον ιδιότητα ότι οι χρήστες μπορούν να αλληλοεπιδράσουν με άλλους χρήστες που βρίσκονται στον ίδιο εικονικό χώρο την ίδια στιγμή.



Εικόνα 2: Sketch up – walk: χαρακτηριστικό παράδειγμα παθητικού συστήματος. Μια κάμερα ακολουθεί μια διαδρομή που χαράζει ο χρήστης με τα βελάκια του πληκτρολογίου

Ανάλογα με τον τύπο της συσκευής που χρησιμοποιείται, διαφέρει ο τρόπος ανίχνευσης σημάτων εισόδου στο σύστημα Εικονικής Πραγματικότητας^[5]. Οι συνηθέστερες τεχνικές ανίχνευσης είναι:

- **Ανίχνευση κίνησης του ματιού (eye tracking):** με αναφορά τη θέση του κεφαλιού του χρήστη ανιχνεύεται είτε η κατεύθυνση που εκείνος κοιτάει είτε η κίνηση του ματιού. Πραγματοποιείται μέσω συσκευών που εφάπτονται στο μάτι (φακοί επαφής), οι οποίοι με μαγνητικούς αισθητήρες καταγράφουν κατακόρυφη, οριζόντια κίνηση και συστροφή ματιού, μέσω υπέρυθρης ακτινοβολίας που ανακλάται από το μάτι και ανιχνεύεται από οπτικούς αισθητήρες ή κάμερα είτε μέσω μέτρησης ηλεκτρικής δραστηριότητας με χρήση ηλεκτροδίων γύρω από το μάτι, παράγοντας έτσι ένα Ηλεκτροφθαλμογράφημα.
- **Ανίχνευση θέσης (position tracking)** η οποία ονομάζεται Motion Capture εάν ανιχνεύεται μόνο η θέση και ο προσανατολισμός ενός μέλους του σώματος και Motion Tracking εάν παρακολουθείται συνεχώς η κίνηση κάποιου μέλους του σώματος. Η ανίχνευση αυτή πραγματοποιείται με χρήση κάμερας και απαιτεί οι κινήσεις να γίνονται σε χαμηλότερη ταχύτητα από την συχνότητα δειγματοληψίας της κάμερας (frame rate). Η κάμερα λαμβάνει τις κινήσεις και εξάγεται το περίγραμμα του στόχου σε κάθε καρέ.
- **Ανίχνευση χειρονομιών (gesture recognition):** με χρήση κάμερας αναγνωρίζονται κινησιολογικά μοτίβα είτε του προσώπου είτε του σώματος του χρήστη. Το αποτέλεσμα είναι ένα ογκομετρικό μοντέλο που περιγράφει κινήσεις χεριών και σώματος.
- **Απτική αντίληψη (haptics):** με χρήση απτικών αισθητήρων και γαντιών εικονικής πραγματικότητας μεταφέρεται η πληροφορία της αφής, δηλαδή των ασκούμενων δυνάμεων, δονήσεων και κινήσεων, στο σύστημα προσομοίωσης εικονικού κόσμου. Οι απτικές συσκευές λειτουργούν ταυτόχρονα ως συσκευές εισόδου και εξόδου και αντιδρούν στις κινήσεις του χρήστη. Μπορεί να είναι συσκευές μονού σημείου αλληλεπίδρασης, οι οποίες διαθέτουν έναν ακροδέκτη σε σχήμα ράβδου τον οποίο ο χρήστης χειρίζεται σαν στυλό και έχουν έξι βαθμούς ελευθερίας ως προς την κίνηση του χρήστη και τρεις ως προς την ανάδραση δύναμης ή συσκευές πολλαπλών σημείων αλληλεπίδρασης, που προσομοιάζουν ένα γάντι και τοποθετούνται στο χέρι του χρήστη. Αυτές οι συσκευές έχουν έναν βαθμό ελευθερίας σε κάθε δάχτυλο του χεριού.



Εικόνα 3: Συσκευή πολλαπλών σημείων αλληλεπίδρασης για απτική αντίληψη



Εικόνα 4: Κάμερα ανίχνευσης κινήσεων του κεφαλιού

Ο συνηθής εξοπλισμός εικονικής πραγματικότητας (hardware) είναι οι οθόνες προσαρμογής στο κεφάλι (HMD) , τα γάντια, οι φόρμες και τα χειριστήρια τύπου Wii.

«Ανάπτυξη εκπαιδευτικού 3D Puzzle με αντικείμενα πολιτιστικής κληρονομιάς σε περιβάλλον Εικονικής Πραγματικότητας»



Εικόνα 5: Εξοπλισμός VR με τη σειρά: φόρμες (data suits), γάντια (data gloves) και οθόνες προσαρμογής στο κεφάλι (HMD)

Η εμπύθιση επιτυγχάνεται με τη βοήθεια των масκών και των παρελκόμενών τους, ανάλογα με το σύστημα που χρησιμοποιείται. Τα σύγχρονα συστήματα μπορεί να χωριστούν σε 3 επιπλέον μεγάλες κατηγορίες:

- Συστήματα ηλεκτρονικού υπολογιστή: στα συστήματα αυτά τα δεδομένα στην μάσκα εικονικής πραγματικότητας παρέχει ένας ηλεκτρονικός υπολογιστής και φαίνονται στον χρήστη μέσω μιας οθόνης κρυστάλλων. Έχουν ως μειονεκτήματα ότι μόνο ένας χρήστης μπορεί να κάνει χρήση του συστήματος κάθε φορά και είναι υψηλού κόστους.



Εικόνα 6: Σύστημα ηλεκτρονικού υπολογιστή με χειριστήρια (Oculus Quest 2)

- Συστήματα smartphone: δεν περιέχουν οθόνη που οπτικοποιεί δεδομένα όπως τα προηγούμενα, διότι τον ρόλο της οθόνης έχει το κινητό τηλέφωνο. Είναι χαμηλότερου κόστους από τα πρώτα, όμως και χαμηλότερης ποιότητας.



Εικόνα 7: Σύστημα για κινητό τηλέφωνο (Google Daydream View)

- Αυτόνομα συστήματα: οι οθόνες είναι ενσωματωμένες και τα δεδομένα παρέχονται από εξωτερικές μνήμες αποθήκευσης

2.1.4: Επαυξημένη Πραγματικότητα – Augmented Reality (AR)

Η Επαυξημένη Πραγματικότητα (AR)^[6] διαφέρει από την Εικονική Πραγματικότητα στο ότι ενισχύει τον πραγματικό κόσμο όπως αυτός είναι μέσω γραφικών (ήχος, εικόνα, ψηφιακά αντικείμενα) και δε δημιουργεί το αίσθημα της εμβύθισης ή την απομόνωση από τον πραγματικό κόσμο. Τα τρία χαρακτηριστικά που καθορίζουν την επαυξημένη πραγματικότητα είναι ότι συνδυάζει το εικονικό με το πραγματικό, ότι είναι διαδραστική σε πραγματικό χρόνο και η πληροφορία χωροθετείται σε τρεις διαστάσεις^[7]. Η επαυξημένη πραγματικότητα χρησιμοποιείται εκτενώς σε παιχνίδια του κινητού τηλεφώνου ή και άλλων φορητών συσκευών με χρήση της κάμερας (π.χ. Pokémon Go, το επιτραπέζιο παιχνίδι Pictionary Air κ.α.), σε εφαρμογές πλοήγησης, σε περιγραφή αθλητικών αγώνων για ανάλυση των κινήσεων των παικτών, στη διακόσμηση εσωτερικών χώρων, ακόμα και σε αρχαιολογικούς χώρους.

«Ανάπτυξη εκπαιδευτικού 3D Puzzle με αντικείμενα πολιτιστικής κληρονομιάς σε περιβάλλον Εικονικής
Πραγματικότητας»



Εικόνα 8: Παράδειγμα επανξημένης πραγματικότητας στη διακόσμηση εσωτερικών χώρων



Εικόνα 9: Το παιχνίδι επανξημένης πραγματικότητας Pokémon Go

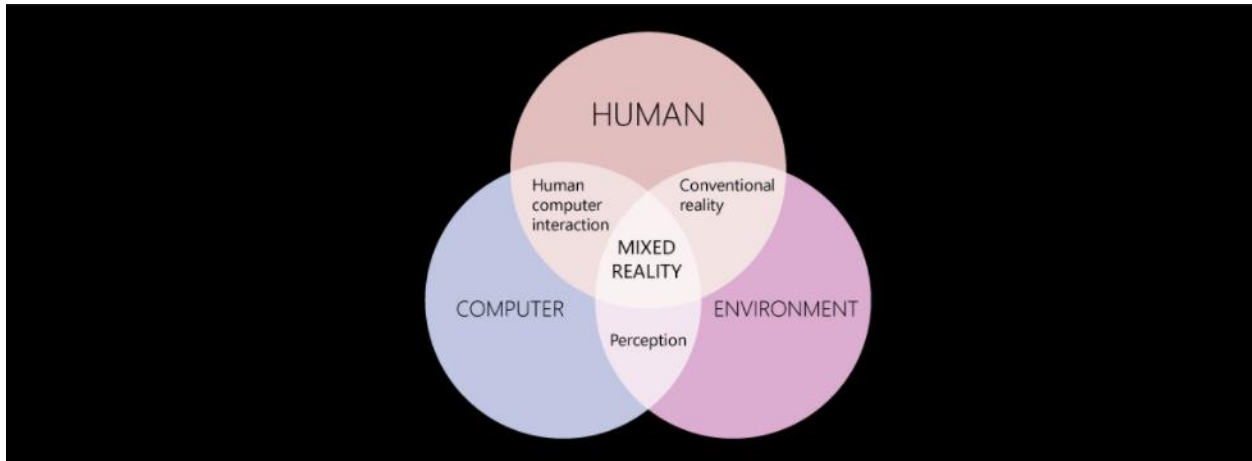
2.1.5: Μεικτή ή Υβριδική Πραγματικότητα – Mixed Reality (MR)

Η Μεικτή Πραγματικότητα (MR) είναι μια ανάμειξη του πραγματικού και του ψηφιακού κόσμου που δημιουργεί συνδέσεις μεταξύ ανθρώπου, υπολογιστή και περιβάλλοντος. Σύμφωνα με τους Miligram & Kishino (1994) η μεικτή πραγματικότητα απεικονίζεται ως ένα συνεχές (Reality – Virtuality Continuum), το ένα άκρο του οποίου αποτελεί το πραγματικό περιβάλλον και το άλλο άκρο το εικονικό. Ανάμεσα στα δύο άκρα εντοπίζεται η επαυξημένη πραγματικότητα και η επαυξημένη εικονικότητα (ψηφιακή υπέρθεση πραγματικών αντικειμένων σε εικονικό περιβάλλον) ενώ η μεικτή πραγματικότητα είναι η ένωση των δύο άκρων. Το σχήμα φαίνεται στην Εικόνα 10.



Εικόνα 10: Το συνεχές της μεικτής πραγματικότητας

Εμβαθύνοντας, η μεικτή πραγματικότητα επεκτείνεται στην ανίχνευση ερεθισμάτων από το περιβάλλον, ήχου και θέσης τόσο στον πραγματικό όσο και στον εικονικό χώρο και βασίζεται στην αλληλεπίδραση χρήστη – υπολογιστή – περιβάλλοντος όπως φαίνεται και στην εικόνα 11. Αναλυτικότερα, η μεικτή πραγματικότητα συνδυάζει την συνηθισμένη πλέον αλληλεπίδραση χρήστη – υπολογιστή, την συμβατική πραγματικότητα την οποία αντιλαμβάνεται ο άνθρωπος και την αντιληπτική ικανότητα του υπολογιστή όσον αφορά το περιβάλλον, καθώς χωρίς εισροές από αυτό δεν μπορεί να γίνει ανάμειξη φυσικού και ψηφιακού κόσμου.



Εικόνα 11: Αλληλεπίδραση υπολογιστή, ανθρώπου και περιβάλλοντος

Η μεικτή πραγματικότητα χρησιμοποιείται σήμερα στην εφαρμογή HoloLens της Microsoft^[8]

2.1.6: Εκτεταμένη Πραγματικότητα – Extended Reality (XR)

Ο όρος της Εκτεταμένης Πραγματικότητας (XR) αποτελεί την ομπρέλα για όλους τους προαναφερθέντες τύπους πραγματικότητας και όσους θα δημιουργηθούν στο μέλλον.

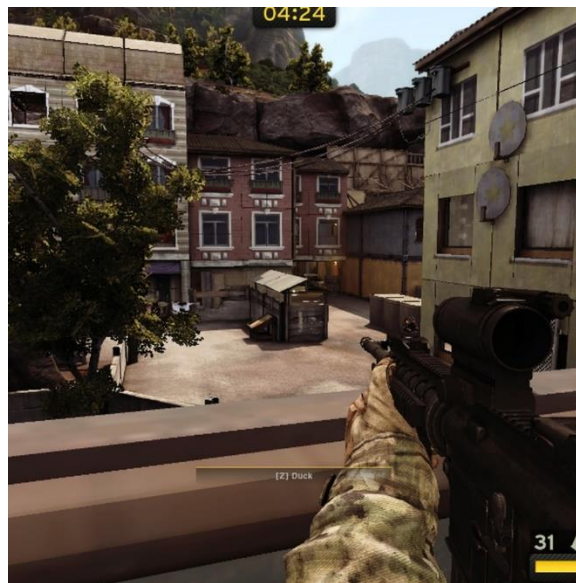
2.2: Εφαρμογές

2.2.1: Serious Games

Τα SG έχουν χρησιμοποιηθεί μέχρι σήμερα για στρατιωτικούς σκοπούς (βλ. πολεμικά παιχνίδια και στρατιωτικές προσομοιώσεις), για εξάσκηση αντιμετώπισης κρίσεων σε κυβερνητικό επίπεδο, σε σχολεία και πανεπιστήμια, σε εταιρίες για εκπαίδευση προσωπικού και στην βιομηχανία της υγείας για εξάσκηση και εκπαίδευση (Michaela, et al., 2014). Κάποιες δημοσιευμένες εφαρμογές αναλύονται παρακάτω.

Στρατιωτικά παιχνίδια

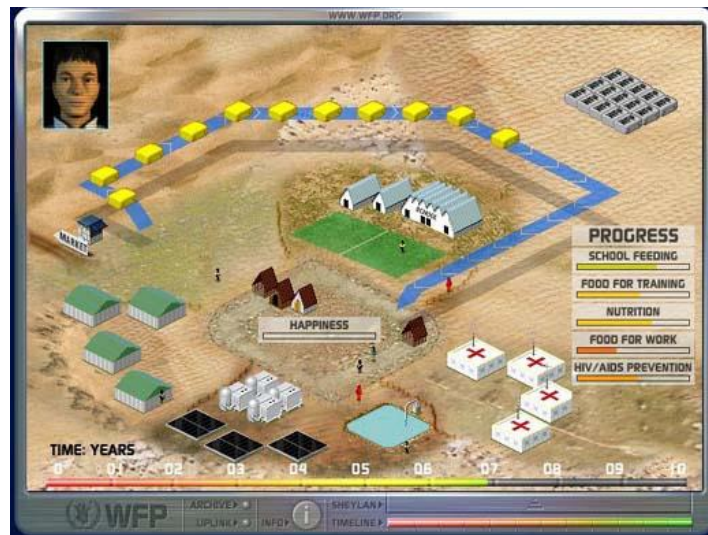
Εδώ και πολλά χρόνια χρησιμοποιούνται παιχνίδια για την εκπαίδευση στρατιωτών, όπως είναι το γνωστό επιτραπέζιο παιχνίδι *chaturanga*, που πρωτοεμφανίστηκε στην αρχαία Ινδία και είναι πρόγονος του σκακιού. Στην ψηφιακή τους μορφή, τα στρατιωτικά παιχνίδια ήταν τα πρώτα που χρησιμοποιήθηκαν για σκοπούς πέρα από τη διασκέδαση προκειμένου να εκπαιδεύσουν στρατιώτες σε ψηφιακό περιβάλλον προσομοιώνοντας πραγματικά γεγονότα. Το πιο γνωστό παιχνίδι που δημιουργήθηκε για αυτό το σκοπό είναι το “America’s Army” (2002)^[9].



Εικόνα 12: Στιγμιότυπο από το παιχνίδι “America’s Army”

Παιχνίδια για κυβερνητικούς σκοπούς

Το περιεχόμενο των συγκεκριμένων παιχνιδιών αφορά κυρίως στην αφύπνιση των πολιτών και σε πολιτικές καμπάνιες. Για παράδειγμα το παιχνίδι *FloodSim*^[10] δημιουργήθηκε το 2007 στο Ηνωμένο Βασίλειο με αφορμή τις εκτεταμένες πλημμύρες και τις συνέπειές τους και σκόπευε στην επαγρύπνηση των πολιτών γύρω από τους κινδύνους και την αντιμετώπιση του φαινομένου, βάζοντας τους παίκτες στη θέση λήψης σχετικών αποφάσεων. Ένα ακόμα παράδειγμα αποτελεί το παιχνίδι *Food Force*, το οποίο δημιουργήθηκε από το Παγκόσμιο Πρόγραμμα Σίτισης των Ηνωμένων Εθνών (United Nations World Food Program) με σκοπό να τοποθετήσει τους παίκτες στη θέση ενός νεοσύλλεκτου μέλους των ΗΕ που πρέπει να διαχειριστεί τον υποσιτισμό και τον εμφύλιο πόλεμο σε ένα νησί του Ειρηνικού Ωκεανού.



Εικόνα 13: Στιγμιότυπο από το παιχνίδι "Food Force"

Εκπαιδευτικά παιχνίδια

Όπως προαναφέρθηκε, τα serious games χρησιμοποιούνται για τη διδασκαλία διαφόρων αντικειμένων τόσο σε μαθητές όσο και σε φοιτητές. Στο συγκεκριμένο τομέα ξεχωρίζουν τα παιχνίδια *Dragon Box Elements*, το οποίο απευθύνεται σε μαθητές του δημοτικού και για να κερδίσουν οι παίκτες πρέπει να επιλύσουν προβλήματα γεωμετρίας, και το *Making History* για διδασκαλία της ιστορίας γύρω από τον 2^ο Παγκόσμιο Πόλεμο. Ιδιαίτερα γνωστή είναι και η εφαρμογή *Duolingo*, η οποία χρησιμοποιεί το παιχνίδι για την εκμάθηση ξένων γλωσσών.

Αξίζει να αναφερθεί το παιχνίδι *Wu's castle* στο οποίο οι χρήστες επιλύουν προβλήματα συντάσσοντας κώδικα C++^[1].



Εικόνα 14: Στιγμιότυπο από το παιχνίδι "Wu's Castle"



Εικόνα 15: Στιγμιότυπο από το παιχνίδι "Making History"

Παιχνίδια στον τομέα της υγείας

Πέρα από την εκπαίδευση στρατιωτών, τα SG χρησιμοποιούνται ευρέως και για την εκπαίδευση μελλοντικών νοσοκόμων, όπως για παράδειγμα το παιχνίδι *Pulse* στο οποίο οι παίκτες μπορούν να εξασκήσουν τις θεωρητικές τους γνώσεις προσπαθώντας να εντοπίσουν το πρόβλημα κάθε ασθενή αντιμετωπίζοντάς τον κατάλληλα. Σημαντικά είναι, επίσης, τα παιχνίδια *Underground*, που βοηθά χειρουργούς να εξασκούνται συχνότερα και

Gryphon Rider, το οποίο έχει ως αντικείμενο την αποκατάσταση ασθενών και χρησιμοποιείται και από γιατρούς και από ασθενείς.

2.2.2: Εφαρμογές VR

Η εικονική πραγματικότητα χρησιμοποιείται πλέον ευρέως όχι μόνο για ψυχαγωγικούς σκοπούς, αλλά και σε ερευνητικούς και επαγγελματικούς χώρους. Στην συνέχεια περιγράφονται χώροι οι οποίοι υιοθέτησαν την ΕΠ και εμπλέκουν τον άνθρωπο σε εικονικούς κόσμους.

Αρχιτεκτονική και Πολεοδομικές Εφαρμογές

Η δυσκολία αποτύπωσης σε χαρτί ορισμένων αστικών δομών οδήγησε στην στροφή των αρχιτεκτόνων στην Εικονική Πραγματικότητα για πιο πιστή αναπαράσταση των κατασκευών που πρόκειται να γίνουν. Η ΕΠ εισάγει την τρίτη διάσταση, κάνοντας τις κατασκευές πιο ρεαλιστικές, αλλά παράλληλα γλιτώνει τους σχεδιαστές από την εκπαίδευση στα αρχιτεκτονικά σύμβολα. Το αποτέλεσμα που προκύπτει είναι, ουσιαστικά, μια ψηφιακή μακέτα, στην οποία μπορεί να προστεθούν ακόμα και οι συνθήκες φωτισμού ενός εσωτερικού χώρου και στην οποία ο εκάστοτε ενδιαφερόμενος μπορεί να περιηγηθεί πριν υλοποιηθεί στο έδαφος.



Εικόνα 16: Ενδεικτική χρήση της ΕΠ στην αρχιτεκτονική εσωτερικού χώρου, Redshift

Αντίστοιχα, οι πολεοδομικές εφαρμογές μελετούν την αστική ανάπτυξη και τις κυκλοφοριακές συνθήκες σε ένα αστικό περιβάλλον. Η χρήση της ΕΠ κάνει την επικοινωνία του πολεοδόμου με το κοινό πιο εύκολη, καθώς μπορεί εκείνο να δει άμεσα τα πλεονεκτήματα και μειονεκτήματα μιας παρέμβασης στο αστικό τοπίο.

Στρατιωτικές εφαρμογές

Όπως και τα παιχνίδια σοβαρού σκοπού έτσι και η εικονική πραγματικότητα έχει χρησιμοποιηθεί εκτενώς για την εξάσκηση στρατιωτών σε ποικίλες καταστάσεις εμπλοκής. Ιδιαίτερα γνωστοί στο ευρύ κοινό είναι οι εξομοιωτές μάχης με χρήση όπλων^[13].

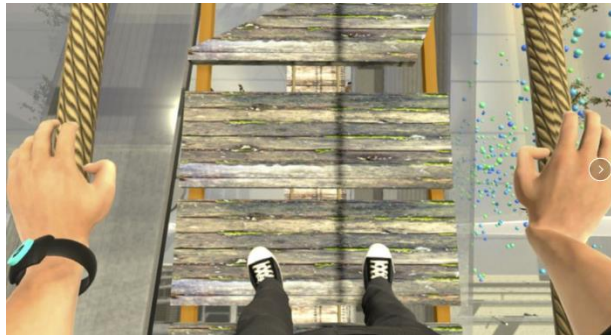


Εικόνα 17: Εξομοιωτές μάχης για εκπαίδευση στρατιωτικών

Ιατρική

Η εικονική πραγματικότητα χρησιμοποιείται σε μεγάλο βαθμό από χειρουργούς και κλινικούς γιατρούς για να αντιμετωπίσουν καταστάσεις κοντά στις πραγματικές ή και για να εκπαιδευτούν, καθώς παρέχεται η δυνατότητα παρατήρησης του εσωτερικού του ανθρώπινου σώματος, φτιαγμένο με 360° CGI. Επιπλέον, η ΕΠ και ειδικότερα η τεχνική της εμπύθισης αποτελεί σημαντικό μέσο παροχής υπηρεσιών για πρόληψη και αντιμετώπιση ψυχιατρικών παθήσεων, βασιζόμενο σε υπολογιστικά περιβάλλοντα (Computer Based Treatment). Ανάλογα την πάθηση, είτε δίνεται στο χρήστη η δυνατότητα ελέγχου του περιβάλλοντός του, είτε εκτίθεται αυτός σε ένα περιβάλλον. Η δεύτερη μέθοδος

χρησιμοποιείται για την αντιμετώπιση φοβιών, διαταραχών άγχους ή μετατραυματικού στρες^{[14][15]}. Ένα άλλο παράδειγμα εφαρμογής εικονικής πραγματικότητας για την ιατρική είναι η χρήση της σε ασθενείς με Αλτσχάιμερ.



Εικόνα 18: Ανάπτυξη συστήματος ΕΠ για αντιμετώπιση υψοφοβίας



Εικόνα 19: Ανάπτυξη συστήματος ΕΠ για αντιμετώπιση κοινωνικών φοβιών με εξομίωση δημόσιας ομιλίας

Εκπαίδευση και επαγγελματική κατάρτιση

Η αξιοποίηση της ΕΠ για εκπαιδευτικούς σκοπούς βασίζεται στην αρχή της «βιωματικής μάθησης». Αναλυτικότερα, ο μαθητής αποκτά άμεση επαφή με το προς μελέτη αντικείμενο και συμμετέχει στην προσέγγιση της ύλης του μαθήματος με αποτέλεσμα να αποκτά περισσότερο ενδιαφέρον και καλύτερη γνώση αυτού. Τη σημαντικότερη συνεισφορά σε αυτό έχουν τα τρισδιάστατα εικονικά περιβάλλοντα μάθησης (Virtual Reality Learning Environments), τα οποία ενσωματώνουν περιβάλλοντα ΕΠ με χαρακτηριστικά κοινωνικών

δικτύων με αποτέλεσμα να μην διαμοιράζουν μόνο πληροφορίες μεταξύ των χρηστών, αλλά ολοκληρωμένες εκπαιδευτικές εμπειρίες.

Αξίζει να αναφερθεί η χρήση ΕΠ για κατάρτιση αστροναυτών σε αποστολές που απαιτούν ιδιαίτερη εξοικείωση με μέσα και εργαλεία^[16].



Εικόνα 20: εργαστήριο επ, nasa

Πολιτισμός

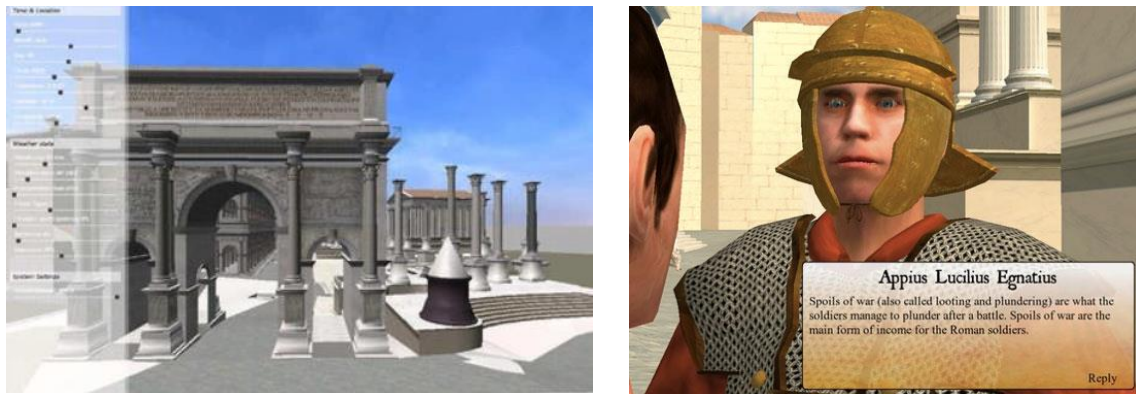
Στον χώρο του πολιτισμού, και ειδικότερα στους αρχαιολογικούς χώρους και τα μουσεία, εμφανίστηκαν οι πρώτες χρήσεις Εικονικής Πραγματικότητας. Όπως αναφέρεται αναλυτικότερα στις επόμενες ενότητες, έχουν αναπτυχθεί πολλές εφαρμογές για ψυχαγωγία και εκπαίδευση μέσω αλληλεπίδρασης του χρήστη με εκθέματα μουσείων ή ευρήματα αρχαιολογικών χώρων.



Εικόνα 21: Εικονική περιήγηση και ανακατασκευή του εσωτερικού χώρου μιας οικείας σε χαλκολιθικό οικισμό με δυνατότητες 3D πλοήγησης και μεταγωγής στον προϊστορικό χρόνο με βάση τα ευρήματα του οικισμού στην Alba Iulia της Ρουμανίας

2.2.3: Serious Games και VR στην πολιτιστική κληρονομιά

Πριν αναφερθούν σημαντικά παιχνίδια σοβαρού σκοπού με αντικείμενο την πολιτιστική κληρονομιά, αξίζει να γίνει μια εισαγωγή στις εφαρμογές που έχουν γίνει στην κατεύθυνση της διατήρησης και αναβίωσης της πολιτιστικής κληρονομιάς. Σύμφωνα με τη δημοσιευμένη έρευνα “Developing Serious Games for cultural heritage: a state – of – the – art review” (Falk Anderson, et al., 2010), έχουν γίνει πολλές προσπάθειες οπτικής ανακατασκευής αρχαιολογικών χώρων για ακαδημαϊκούς κυρίως σκοπούς, με σημαντικότερη εφαρμογή την “Roma Nova” (Rome Reborn) η οποία αναπαριστά την Ρώμη στη μορφή που είχε το 320 μ.Χ. Σκοπεύει στην εξοικείωση του χρήστη με την κοινωνικοπολιτική ζωή της εποχής και δίνει την δυνατότητα στους αρχαιολόγους να ελέγξουν παρελθοντικές και παροντικές υποθέσεις σχετικά με την αρχιτεκτονική, την τοπογραφία, την πολεοδομία και την κοινωνική ζωή. Για να επιτευχθεί αυτό χρησιμοποιούνται εικονικοί χαρακτήρες (avatar) με διαφορετικές συμπεριφορές ο καθένας (behavior- driven) με τους οποίους ο χρήστης κάνει διάλογο.



Εικόνα 22: Στιγμιότυπα από την εφαρμογή “Roma Nova”

Παρόμοια εφαρμογή έχει αναπτυχθεί για την αρχαία Πομπηία ((Maim, et al., 2007), (Muller, et al., 2005)) χωρίς όμως να πραγματοποιείται αλληλεπίδραση χρήστη - περιβάλλοντος παρά μόνο αναπαράσταση της ζωής πριν την καταστροφή της Ρωμαϊκής πόλης το 79 μ.Χ. Αντίστοιχα, αναπτύχθηκε σε περιβάλλον Unity το τρισδιάστατο μοντέλο του αρχαίου θεάτρου της Πομπηίας με μουσική και ήχο^[17].








Εικόνα 23: Το θέατρο της Πομπηίας

Επιπλέον, έχει δημιουργηθεί ψηφιακή αναπαράσταση του Παρθενώνα και των αγαλμάτων που βρίσκονται στο Βρετανικό Μουσείο ώστε να ενωθούν εικονικά και να δοθεί η παλαιότερη μορφή του ναού. Πέρα από τον Παρθενώνα έχει αναπαρασταθεί σε ψηφιακή μορφή η μεγαλύτερη έκταση της Αρχαίας Αθήνας στα πλαίσια του προγράμματος “Ancient Athens 3D”^[18]. Στο πρόγραμμα αυτό παρέχεται ψηφιακή ξενάγηση στην Αθήνα 7 ιστορικών περιόδων, της Μυκηναϊκής (1200 π.Χ.), της Αρχαϊκής (600 - 480 π.Χ.), της Κλασικής (479 – 323 π.Χ.), της Ελληνιστικής (323 – 86 π.Χ.), της Ρωμαϊκής (86 π.Χ. – 267 μ.Χ.), της Μεσαιωνικής (267 – 1458 μ. Χ.) και της Οθωμανικής (1458 – 1833).

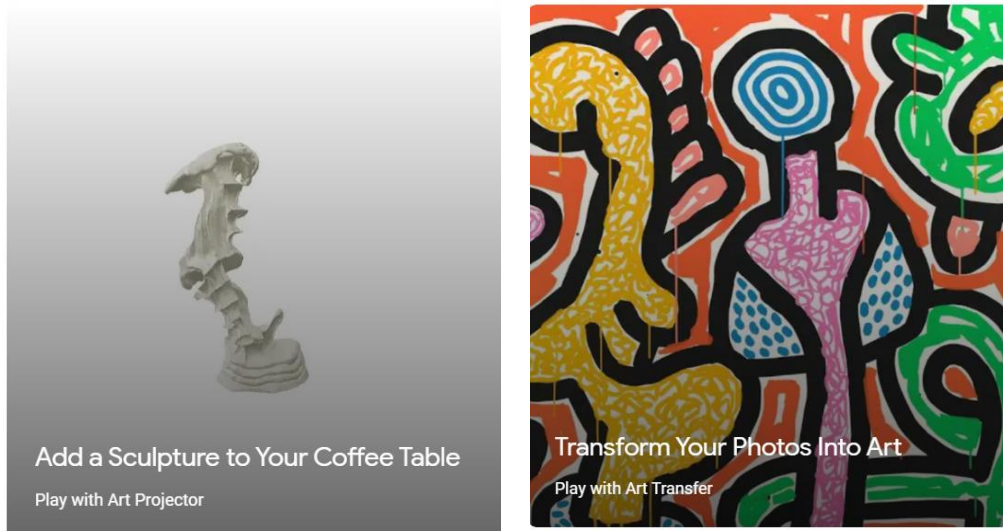
Πίνακας 1: Ψηφιακές αναπαραστάσεις της Ακρόπολης των Αθηνών ανά ιστορικές περιόδους

Η ΑΚΡΟΠΟΛΗ ΤΩΝ ΑΘΗΝΩΝ ΣΤΙΣ 7 ΙΣΤΟΡΙΚΕΣ ΠΕΡΙΟΔΟΥΣ	
	ΜΥΚΗΝΑΪΚΗ ΠΕΡΙΟΔΟΣ
	ΑΡΧΑΪΚΗ ΠΕΡΙΟΔΟΣ

 <p>www.ancientathens3d.com</p>	<p>ΚΛΑΣΙΚΗ ΠΕΡΙΟΔΟΣ</p>
 <p>www.AncientAthens3d.com</p>	<p>ΕΛΛΗΝΙΣΤΙΚΗ ΠΕΡΙΟΔΟΣ</p>
 <p>www.ancientathens3d.com</p>	<p>ΡΩΜΑΪΚΗ ΠΕΡΙΟΔΟΣ</p>
 <p>www.ancientathens3d.com</p>	<p>ΜΕΣΑΙΩΝΙΚΗ ΠΕΡΙΟΔΟΣ</p>
 <p>www.ancientathens3d.com</p>	<p>ΟΘΩΜΑΝΙΚΗ ΠΕΡΙΟΔΟΣ</p>

Πέρα από την Ακρόπολη, η συγκεκριμένη εφαρμογή πραγματοποιεί ξεναγήσεις και στην Αρχαία Αγορά των Αθηνών καθώς και σε άλλα ιστορικά μέρη της πόλης.

Τα εικονικά μουσεία αποτελούν μερικές από τις σημαντικότερες συμβολές των νέων τεχνολογιών στην αναβίωση της πολιτιστικής κληρονομιάς, καθώς διατίθενται σε ψηφιακή μορφή σημαντικά εκθέματα. Παράδειγμα αποτελεί η εφαρμογή της Google “Google’s Arts and Culture”, μέσω της οποίας ο χρήστης μπορεί να επιλέξει από μια λίστα μουσείων και να αποκτήσει πρόσβαση στα εκθέματα αυτών και την ιστορία τους. Η συγκεκριμένη εφαρμογή παρέχει στους χρήστες και τη δυνατότητα να αλληλεπιδράσουν με διάφορα εκθέματα μέσω της κάμερας του κινητού τηλεφώνου. Για παράδειγμα, μπορούν να φωτογραφηθούν με τη στολή του αστροναύτη Neil Armstrong ή να βρουν με ποιο πρόσωπο γνωστών έργων τέχνης μοιάζουν ή ακόμα και να δοκιμάσουν πώς ταιριάζουν διάφορα εκθέματα στο σπίτι τους. Επίσης, βοηθάει τον χρήστη να βρει ανάμεσα σε πάνω από 100000 εκθέματα εκείνα που ταιριάζουν σε κάποιο χρώμα της επιλογής τους ή να συνθέσουν ένα δικό τους κομμάτι όπερα με χρήση όρασης υπολογιστών (blob opera). Επιπλέον, περιλαμβάνει διάφορα παιχνίδια για εξοικείωση του χρήστη με τον πολιτισμό, όπως για παράδειγμα το παιχνίδι “Can you guess what came first?”, στο οποίο ο χρήστης πρέπει να διαλέξει ανάμεσα σε 2 αντικείμενα πολιτιστικής κληρονομιάς ποιο προηγείται χρονικά^[19].

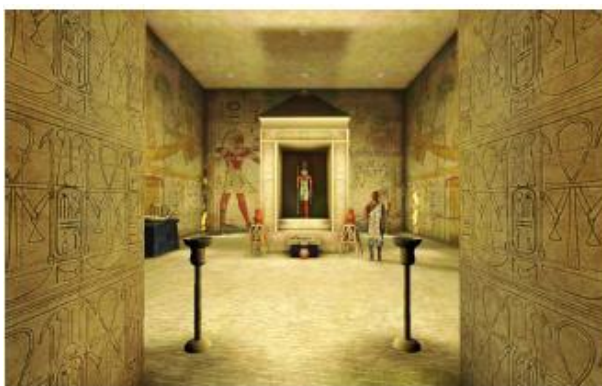


Εικόνα 24: Παραδείγματα αλληλεπίδρασης χρήστη με εκθέματα στο google’s arts and culture



Εικόνα 25: Blob opera, machine learning experiment

Πέρα από τις εφαρμογές που παρέχονται εντός της Google, έχουν ψηφιοποιηθεί πολλά μουσεία ως μέρος άλλων εφαρμογών (Falk Anderson, et al., 2010). Παράδειγμα αποτελεί ο εικονικός αιγυπτιακός ναός, ο οποίος απεικονίζει ένα καθαρά υποθετικό περιβάλλον που ταιριάζει πλήρως με την πραγματική μορφή που θα είχε ο ναός κατά την περίοδο του Νέου Βασιλείου (1550 – 1070 π.Χ.). Ο ναός είναι χωρισμένος σε τέσσερα βασικά μέρη, το καθένα από τα οποία φιλοξενεί μια φιγούρα του αρχιερέα. Στο ψηφιακό περιβάλλον του ναού στήνεται ένα παιχνίδι, με σκοπό την εξερεύνηση του μοντέλου και τη συλλογή πληροφοριών ώστε να απαντηθούν οι ερωτήσεις του ιερέα.



Εικόνα 26: Αιγυπτιακός Ναός της Περιόδου Νέου Βασιλείου

Τέλος, αξίζει να αναφερθεί η χρήση ψηφιακών χαρτών και ξεναγών μέσω εφαρμογών, που κάνουν την ξενάγηση σε αρχαιολογικούς χώρους πολύ πιο απλή (π.χ. “Discover Pompeii – Pompeii audio tour”)^[20].

Πλέον τα παιχνίδια σοβαρού σκοπού σε συνδυασμό με τα τριδιάστατα μοντέλα και την φωτογραμμετρία γενικότερα, έχουν κάνει την προσέγγιση πολύ πιο ενδιαφέρουσα καθώς οι νέες τεχνολογίες επιτρέπουν την κατάλληλη εξαγωγή τρισδιάστατων μοντέλων που θα χρησιμοποιηθούν για αυτό το σκοπό (Kontogianni, et al., 2016). Αξίζει να αναφερθεί η δημοσιευμένη εργασία “The contribution of 3D models to serious games applications”

(Kontogianni, 2015) η οποία πραγματεύεται τη δημιουργία ενός παιχνιδιού στην Αρχαία Αγορά της Αθήνας με σκοπό την εξοικείωση των παικτών με τα αρχαία μνημεία χρησιμοποιώντας ένα παιχνίδι ερωτήσεων. Τα μνημεία παράγονται με φωτογραμμετρικές μεθόδους και το παιχνίδι αναπτύσσεται στο λογισμικό Unity 3D.



Εικόνα 27: Το κύριο menu του παιχνιδιού, (Kontogianni, Georgia *The Contribution of 3D Models to Serious Games Applications*)

Το 2004 δημιουργήθηκε από το Ίδρυμα Μείζονος Ελληνισμού (Foundation of Hellenic World)^[21] ένα σύνολο παιχνιδιών με αντικείμενο τους Ολυμπιακούς Αγώνες της αρχαίας Ελλάδας (2004). Ξεχώρισε ιδιαίτερα το παιχνίδι «Εργαστήριο του Φειδία» (Feidias' Workshop), το οποίο έδινε στους χρήστες τον ρόλο του γλύπτη στην κατασκευή του χρυσού αγάλματος του Δία χρησιμοποιώντας εξοπλισμό εικονικής πραγματικότητας ώστε να πραγματοποιήσουν τις κατάλληλες κινήσεις για να τοποθετήσουν τα κομμάτια στις σωστές θέσεις. Από τότε μέχρι και σήμερα το Ίδρυμα Μείζονος Ελληνισμού έχει αξιοποιήσει σε μεγάλο βαθμό την Εικονική Πραγματικότητα με σκοπό την αναβίωση της πολιτιστικής κληρονομιάς προσφέροντας περιηγήσεις σε αρχαίες πόλεις και μνημεία τόσο στην «Θόλο» του Κέντρου Πολιτισμού «Ελληνικός Κόσμος» όσο και διαδικτυακά. Αναλυτικότερα, πραγματοποιούνται περιηγήσεις:

1. στην Αρχαία Αγορά της Αθήνας, η οποία διαδραματίζεται σε 3 στάδια 50 λεπτών το καθένα: «Διαδραστική περιήγηση στην Αρχαία Αγορά», «Η Αθηνά στην Αρχαία Αγορά» και «Δρώμενο στην Αρχαία Αγορά». Στην περιήγηση αυτή συμμετέχουν και ηθοποιοί
2. στην Αρχαία Ολυμπία, η οποία δίνει την δυνατότητα στους επισκέπτες να παρακολουθήσουν αγωνίσματα και τελετουργίες των Ολυμπιακών Αγώνων και να δουν τα μνημεία και τα ιερά κτίρια του χώρου της Αρχαίας Ολυμπίας
3. στην Αρχαία Μίλητο με τη μορφή που είχε πριν 2000 χρόνια
4. στην Αρχαία Πριήνη
5. στην Ακρόπολη της εποχής του Περικλή
6. στην Αγία Σοφία



Εικόνα 28: Η «Θόλος» του Ελληνικού Κόσμου τη νύχτα

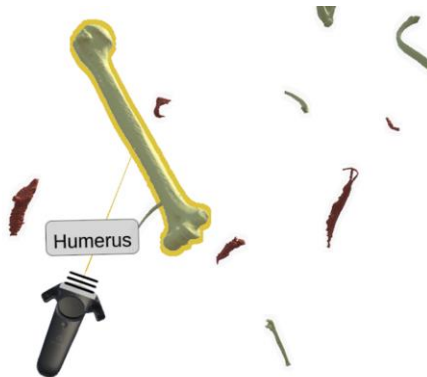
Το 2007 αναπτύχθηκε από το History Channel ένα ιστορικό παιχνίδι στρατηγικής – ντοκιμαντέρ με αντικείμενο τις μάχες της Ρώμης (Slitherine Strategies, 2007)^[22], το οποίο συνδυάζει προσομοίωση της μάχης με ιστορικές πληροφορίες.

Το 2018 αναπτύχθηκε μια εφαρμογή εικονικής πραγματικότητας με αντικείμενο την πραγματοποίηση και εξαγωγή γεωδαιτικών μετρήσεων αποστάσεων, τομών και συντεταγμένων (Κοντός, 2019). Μετρητικό αντικείμενο αποτέλεσε το Ιερό Κουβούκλιο του Πανάγιου Τάφου και η εφαρμογή αξιολογήθηκε ως προς τα μετρητικά αποτελέσματα και την αξιοπιστία της, καθώς και τη χρηστικότητα και λειτουργικότητά της. Όπως αναφέρεται και στη δημοσιευμένη εργασία, οι εφαρμογές της εικονικής πραγματικότητας που έχουν παρεμβατική μορφή ελλοχεύουν κίνδυνο αλλοίωσης των μνημείων, επομένως είναι σημαντικό να πραγματοποιούνται σε περιβάλλον εμπύθισης.

2.2.4: VR Serious Games

Το 2019 πραγματοποιήθηκε μια μελέτη με αντικείμενο τα παιχνίδια σοβαρού σκοπού στην εκπαίδευση ιατρών, στοχεύοντας στην εκμάθηση της ανατομίας του ανθρώπινου σώματος (Pohlandt, et al., 2019) (Michaela & Lawrence, 2001)). Στη δημοσιευμένη εργασία αναλύονται τα τέσσερα βήματα της διαδικασίας της μάθησης (preparation, presentation, practice, performance) και τονίζεται η σημασία της συναρμολόγησης και αποσυναρμολόγησης ενός παζλ στην καλλιέργεια ενδιαφέροντος για μάθηση και βελτίωση επαφής με το αντικείμενο. Το παιχνίδι βοηθά τους χρήστες να μάθουν την ανατομία του ανθρώπινου σώματος ονομάζοντας όλα τα μέρη του σώματος και ενώνοντάς τα καταλλήλως και χωρίζεται σε τρία μέρη: παροχή γνώσης στον χρήστη, επανάληψη της γνώσης που ήδη έχει ή εφαρμογή αυτής

σε ένα σενάριο. Η πιλοτική αυτή εφαρμογή έγινε σε γενικές γραμμές αποδεκτή από την εκπαιδευτική κοινότητα, δίνοντας το πράσινο φως και σε εφαρμογές παρόμοιου βεληνεκούς.



Εικόνα 29: Παράδειγμα επιλογής κομματιού και προβολής του ονόματός του – Supporting Anatomy Education with a 3D puzzle in a Virtual Reality Environment

Το παιχνίδι Keep Talking and Nobody Explodes της εταιρείας Steel Crate Games αποτελεί ένα χαρακτηριστικό παράδειγμα VR παιχνιδιού σοβαρού σκοπού. Πρωτοεμφανίστηκε με στόχο την εκπαίδευση πυροτεχνουργών στον αφοπλισμό βόμβων και πλέον αποτελεί ένα παιχνίδι συνεργασίας από 2 έως 6 παίκτες οι οποίοι είναι εγκλωβισμένοι σε ένα δωμάτιο με μία βόμβα. Ζητούμενο είναι ο ένας παίκτης να αφοπλίσει την βόμβα σύμφωνα με οδηγίες που του δίνονται από τους υπόλοιπους χωρίς όμως εκείνοι να βλέπουν την βόμβα. Το συγκεκριμένο παιχνίδι εκτός από σοβαρού σκοπού αποτελεί και χαρακτηριστικό παράδειγμα χρήσης περιβάλλοντος εικονικής πραγματικότητας για video games, καθώς δημιουργεί στους χρήστες το αίσθημα της εμπύθισης λόγω της αγωνίας να αφοπλίσουν την βόμβα πριν εκείνη εκραγεί.



Εικόνα 30: Παράδειγμα της προς αφοπλισμό βόμβας σε ένα από τα σενάρια

ΚΕΦΑΛΑΙΟ 3: Ανάπτυξη εφαρμογής

Στην παρούσα ενότητα αναλύεται η προεργασία για την ανάπτυξη της εφαρμογής. Συγκεκριμένα, είναι απαραίτητο να χρησιμοποιηθούν διάφορα λογισμικά και περιβάλλοντα και ένα πειραματικό αντικείμενο στο οποίο θα γίνουν όλες οι δοκιμές που θα οδηγήσουν στο ιδανικό αποτέλεσμα. Απαιτείται μια μηχανή παιχνιδιών, δηλαδή ένα σύστημα λογισμικού σχεδιασμένο για την ανάπτυξη παιχνιδιών και άλλων εφαρμογών, και ένα εργαλείο επεξεργασίας τρισδιάστατων μοντέλων. Επιπλέον χρησιμοποιείται μια ακόμα μηχανή παιχνιδιών για την επίλυση του ζητήματος της τομής του αρχικού αντικειμένου σε επιμέρους κομμάτια.

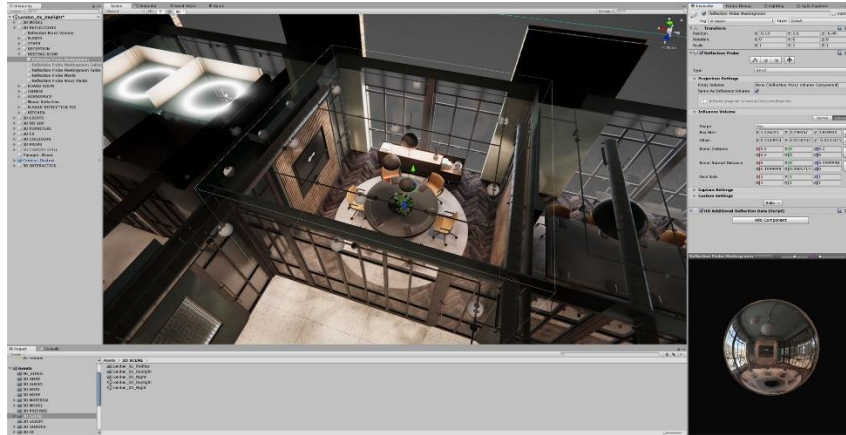
3.1: Λογισμικά που χρησιμοποιήθηκαν

3.1.1: Unity Game Engine

Η μηχανή παιχνιδιών Unity^[23] αναπτύχθηκε από την εταιρεία Unity Technologies τον Ιούνιο του 2005 και οι εφαρμογές που δημιουργούνται σε αυτήν υποστηρίζονται από πολλές πλατφόρμες, δηλαδή υπολογιστή, κινητό τηλέφωνο, κονσόλες και πλατφόρμες Εικονικής Πραγματικότητας. Ξεκίνησε ως αποκλειστική μηχανή παιχνιδιών για Mac OS, με την πάροδο των χρόνων όμως υποστηρίζει και άλλα λογισμικά. Πέρα από τα παιχνίδια, η Unity χρησιμοποιείται και στον κινηματογράφο, την αρχιτεκτονική, τις κατασκευές και τις μετακινήσεις. Από τον Ιούνιο του 2020 υποστηρίζει εφαρμογές σε Μικτή και Επαυξημένη Πραγματικότητα (Mixed and Augmented Reality Studio – MARS).

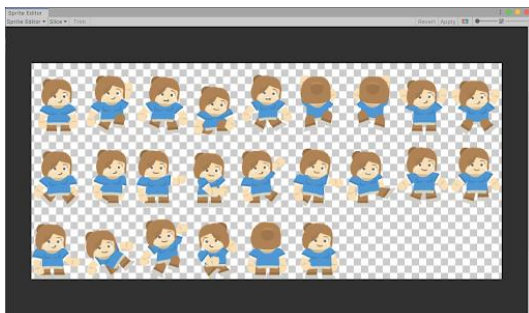


Εικόνα 31: Unity Game Engine Logo

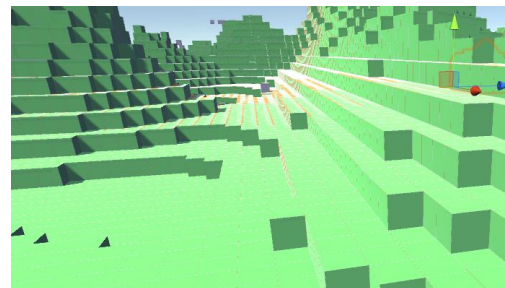


Εικόνα 32: Η επιφάνεια εργασίας της μηχανής παιχνιδιών Unity

Παρέχει τη δυνατότητα δημιουργίας τόσο διδιάστατων (2D) όσο και τρισδιάστατων (3D) παιχνιδιών και διαδραστικών προσομοιώσεων και χρησιμοποιεί για τον προγραμματισμό εφαρμογών (Application Programming Interface – API) την γλώσσα C#, η οποία αποτελεί μια αντικειμενοστραφή γλώσσα προγραμματισμού και συντάσσεται στην επέκταση Visual Studio της Microsoft. Επιπλέον, δεδομένα μπορεί να εισαχθούν στην μηχανή με τη μέθοδο drag and drop, είτε με όσα παρέχονται από την βιβλιοθήκη της Unity είτε εκτός αυτής. Στις διδιάστατες εφαρμογές είναι δυνατή η εισαγωγή απλών αντικειμένων που έχουν υφή ή γραφικές εικόνες (sprites) και η απεικόνιση του διδιάστατου κόσμου ενώ στις τρισδιάστατες είναι εφικτές ακόμα περισσότερες ενέργειες, όπως για παράδειγμα η χρήση εικόνων με απεικόνιση της υφής σε επίπεδα (mirmaps). Επιπλέον, στις εφαρμογές μπορεί να προστεθεί ήχος και ειδικός φωτισμός.



Εικόνα 33: Παράδειγμα sprites στη Unity Game Engine











Εικόνα 34: Παράδειγμα mirmap της Unity Game Engine


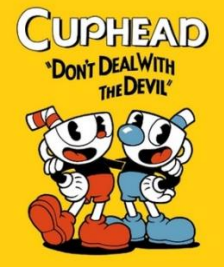




Γνωστές εφαρμογές στην Unity Game Engine




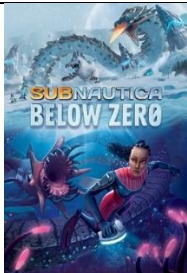
Από το 2005 και μετά η μηχανή παιχνιδιών Unity έχει χρησιμοποιηθεί για την δημιουργία πολλών εφαρμογών, οι περισσότερες από τις οποίες γνώρισαν μεγάλη επιτυχία στην κοινότητα των χρηστών. Στον πίνακα 2 συνοψίζονται τα πιο γνωστά παιχνίδια, ταξινομημένα με την χρονολογία κυκλοφορίας τους^[24].

Πίνακας 2: Γνωστά παιχνίδια που αναπτύχθηκαν σε περιβάλλον Unity

Χρονολογία Κυκλοφορίας	Όνομα Παιχνιδιού	Logo Παιχνιδιού	Πλατφόρμα	Σύντομη Περιγραφή
2005	GooBall		Mac OS X	Αποτελεί παιχνίδι δράσης. Ο παίκτης παίρνει το ρόλο ενός εξωγήινου, του Goober, ο οποίος βρίσκεται κολλημένος σε ένα μηχάνημα υποστήριξης ζωής από πρωτόπλασμα.
2008	Dead Frontier		Web browser	Αποτελεί παιχνίδι συνεργασίας πολλών παικτών και επιβίωσης, στο οποίο οι παίκτες πρέπει να πολεμήσουν ζόμπι
2009	Fusion Fall		Desktop	Αποτελεί παιχνίδι συνεργασίας πολλών παικτών στο οποίο πρέπει να εμποδίσουν οι παίκτες την καταστροφή της γης.
2010	PGA Tour		Wii, Xbox, PlayStation, Desktop	Είναι σειρά παιχνιδιών golf, δημοσιευμένα από την εταιρεία EA sports.
2011	Battlestar Galactica Online		Desktop	Αποτελεί παιχνίδι συνεργασίας πολλών παικτών, στο οποίο οι παίκτες είναι πιλότοι στο σύμπαν Battlestar Galactica.
	Temple Run		Mobile Phone	Είναι ένα τρισδιάστατο παιχνίδι στο οποίο ο χαρακτήρας τρέχει συνεχώς και αποφεύγει εμπόδια με σκοπό να σωθεί από πλάσματα που τον κυνηγάνε

	Monkey Quest		Web browser	Αποτελεί παιχνίδι φαντασίας, στο οποίο οι παίκτες πρέπει να σώσουν τον βασιλιά που πεθαίνει εξερευνώντας τον κόσμο.
2012	Subway Surfers		Desktop & Mobile Phone	Παιχνίδι παρόμοιο με το Temple Run, στο οποίο οι παίκτες στο ρόλο καλλιτεχνών γκράφιτι τρέχουν να σωθούν από τον φύλακα ενός σταθμού
	Slender, the eight pages		Desktop	Αποτελεί παιχνίδι τρόμου και επιβίωσης, στο οποίο ο παίκτης παίζοντας σε πρώτο πρόσωπο καλείται να μαζέψει 7 σελίδες από ένα δάσος χωρίς να τον πιάσει ο Slender Man.
	The Room		Desktop & Mobile Phone	Αποτελεί παιχνίδι γρίφων, στο οποίο ο παίκτης πρέπει να βρει τον τρόπο να φύγει από ένα δωμάτιο. Ακολούθησαν τα The Room 2 & 3.
2014	Hearthstone (Heroes of Warcraft)		Desktop & Mobile Phone	Είναι παιχνίδι με κάρτες στο οποίο κάθε παίκτης πρέπει να νικήσει έναν αντίπαλο.
2015	Fallout Shelter		Desktop & Mobile Phone	Στο συγκεκριμένο παιχνίδι προσομοίωσης οι παίκτες πρέπει να διαχειριστούν μια αποθήκη ώστε να παρέχουν προμήθειες στους κατοίκους μιας κοινότητας
2016	Overcooked		Nintendo, Xbox, PlayStation, Microsoft Windows	Αποτελεί προσομοιωτικό παιχνίδι στο οποίο οι παίκτες παίρνουν ρόλο σεφ σε μια κουζίνα.
	Pokémon Go		Mobile Phone	Αποτελεί παιχνίδι επαυξημένης πραγματικότητας στο οποίο οι παίκτες πρέπει να επισκεφτούν πραγματικές

				τοποθεσίες για να αιχμαλωτίσουν Pokémon.
	Poly Bridge		Desktop & Mobile Phone	Στο συγκεκριμένο παιχνίδι οι παίκτες πρέπει να κατασκευάσουν γέφυρες ώστε να βοηθήσουν οχήματα να προχωρήσουν στο χάρτη
2017	Cuphead		Desktop, Nintendo, Xbox, PlayStation	Αποτελεί παιχνίδι μορφής run and gun με μάχες μεταξύ χαρακτήρων.
	Life is Strange		Desktop, PlayStation, Xbox, Mobile Phone	Αποτελεί παιχνίδι επιλογών, στο οποίο ο παίκτης βλέπει τον χαρακτήρα σε τρίτο πρόσωπο και αποφασίζει για την εξέλιξη του παιχνιδιού.
2018	Among Us		Mobile Phone, Desktop, PlayStation, Xbox, Nintendo	Το συγκεκριμένο παιχνίδι πολλών παικτών διαδραματίζεται σε ένα διαστημόπλοιο, και μεταξύ των χαρακτήρων υπάρχει κάποιος υποκριτής (impostor) που σκοτώνει τους υπόλοιπους. Σκοπός είναι να προλάβουν οι υπόλοιποι παίκτες να ολοκληρώσουν κάποιες αγγαρείες ή να βρουν τον υποκριτή.
	Rick and Morty: Virtual Rick-ality		VR	Ο παίκτης ενσαρκώνει τον χαρακτήρα Morty ως κλώνος αυτού με σκοπό να βοηθήσει τους αρχικούς χαρακτήρες να φέρουν εις πέρας εργασίες και αποστολές.
	The Forest		Microsoft Windows, PlayStation	Αποτελεί παιχνίδι τρόμου και επιβίωσης, στο οποίο οι επιζώντες μιας πτώσης αεροπλάνου καλούνται να επιβιώσουν σε ένα νησί.

2019	Call of Duty: Mobile		Mobile Phone	Αποτελεί την εκδοχή για το κινητό τηλέφωνο του γνωστού πολεμικού παιχνιδιού Call of Duty
2020	Fall Guys		Mobile Phone, Desktop, PlayStation, Xbox, Nintendo	Μέχρι 60 παίκτες κινούνται ταυτόχρονα σε έναν τρισδιάστατο χώρο και πρέπει να ολοκληρώσουν δοκιμασίες. Κερδίζει μόνο ένας παίκτης, ο τελευταίος που θα επιβιώσει όλες τις δοκιμασίες.
	League of Legends: Wild Rift		Mobile Phone	Αποτελεί την έκδοση για κινητό τηλέφωνο του γνωστού παιχνιδιού μάχης League of Legends
2021	Subnautica: Below Zero		Desktop, PlayStation, Xbox, Nintendo	Αποτελεί παιχνίδι δράσης και επιβίωσης σε οπτική πρώτου προσώπου κατά το οποίο ο παίκτης παίρνει το ρόλο επιστήμονα που αναζητά ζωή σε έναν άλλο πλανήτη.

Όπως φαίνεται από τον πίνακα, τα παιχνίδια που έχουν αναπτυχθεί προορίζονται για όλων των ειδών τις πλατφόρμες και προσφέρουν διαφορετικές μορφές ψυχαγωγίας.

Πέρα από τα παιχνίδια, η μηχανή Unity χρησιμοποιήθηκε το 2010 για τη δημιουργία μιας ταινίας μικρού μήκους (Adam) για ένα ρομπότ που δραπέτευσε από τη φυλακή. Στην ταινία προστέθηκαν αργό δύο ακόμα μέρη (Adam: The Mirror & Adam: The Prophet).

«Ανάπτυξη εκπαιδευτικού 3D Puzzle με αντικείμενα πολιτιστικής κληρονομιάς σε περιβάλλον Εικονικής Πραγματικότητας»



Εικόνα 35: Το ρομπότ Adam από την ταινία μικρού μήκους της Unity

Πιο πρόσφατα, με χρήση της Unity και με Εικονική Πραγματικότητα δημιουργήθηκαν τα σκηνικά για την ταινία «Ο Βασιλιάς των Λιονταριών» της Disney, του 2019.



Εικόνα 36: Σκηνικό της ταινίας Lion King (2019) που δημιουργήθηκε στην Unity



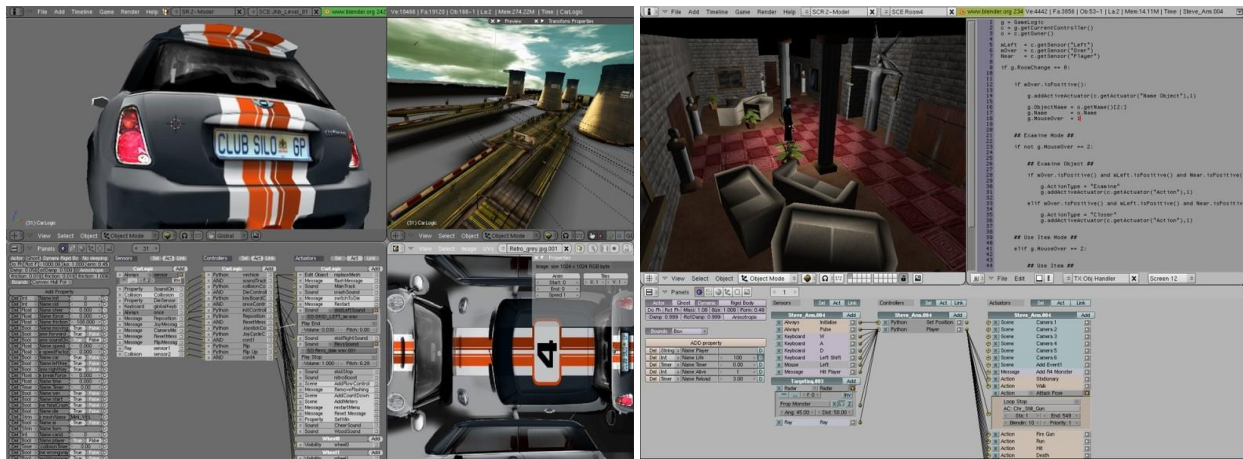
Εικόνα 37: Στιγμιότυπο από την δημιουργία της ταινίας Lion King σε περιβάλλον ΕΠ

3.1.2: Blender Game Engine



Εικόνα 38:
Blender Logo

Η μηχανή παιχνιδιών Blender^[25] αποτελεί κομμάτι της πλατφόρμας Blender για την παραγωγή ρεαλιστικού διαδραστικού περιεχομένου. Αναπτύχθηκε το 2000 με σκοπό την εύκολη δημιουργία παιχνιδιών και περιεχομένου είτε ως εφαρμογές είτε ως ιστοσελίδες. Είναι γραμμένη σε γλώσσα προγραμματισμού C++ αλλά στηρίζει και την Python. Για τον έλεγχο των κινήσεων και της απεικόνισης αντικειμένων, η μηχανή παιχνιδιών Blender χρησιμοποιεί ένα σύστημα γραφικών «λογικών τούβλων», δηλαδή ένα συνδυασμό από αισθητήρες, ελεγκτές και ενεργοποιητές.



Εικόνα 39: Παράδειγμα χρήσης των λογικών τούβλων στο περιβάλλον εργασίας της Blender και του προγραμματισμού με Python.

Τον Σεπτέμβρη του 2015 δημιουργήθηκε μια παραλλαγή της Blender, η UPBGE (Uchronia Project Blender Game Engine) με σκοπό την βελτίωση του κώδικα της μηχανής παιχνιδιών Blender και τον πειραματισμό με νέες λειτουργίες. Μέχρι σήμερα έχουν γίνει πολλές προσαρμογές της μηχανής παιχνιδιών Blender για την εξυπηρέτηση διαφορετικών σκοπών. Για παράδειγμα, δημιουργήθηκε η μηχανή Blend4Web για τη δημιουργία διαδραστικού τρισδιάστατου περιεχομένου σε φυλλομετρητές ιστοσελίδων (web browser) ή η μηχανή Panda3D για την δημιουργία τρισδιάστατων παιχνιδιών με το επιπλέον χαρακτηριστικό την ανίχνευση συγκρούσεων μεταξύ των αντικειμένων.

Γνωστές εφαρμογές στην Blender Game Engine

Η εταιρεία Blender είναι γνωστή και για την παραγωγή παιχνιδιών αλλά και για ταινίες. Αξίζει να αναφερθούν δύο από τις σημαντικότερες εφαρμογές που αναπτύχθηκαν στην μηχανή παιχνιδιών Blender:

- **Yo Frankie!** : αποτελεί παιχνίδι ηλεκτρονικού υπολογιστή που κυκλοφόρησε το 2008, βασισμένο στην ταινία Big Buck Bunny παραγωγής Blender Institute. Ανήκει στην κατηγορία των παιχνιδιών δράσης με χαρακτηριστικό τα άλματα και την αναρρίχηση στο περιβάλλον.



Εικόνα 40: Το logo του παιχνιδιού Yo Frankie!

- **Sintel the Game**: Αποτελεί παιχνίδι ηλεκτρονικού υπολογιστή βασισμένο στην ταινία Sintel παραγωγής Blender Institute. Ανήκει στην κατηγορία παιχνιδιών δράσης και αναφέρεται σε ένα έφηβο κορίτσι που αναζητεί έναν δράκο.



Εικόνα 41: Το logo του παιχνιδιού Sintel

3.2: Προγράμματα επεξεργασίας τρισδιάστατων μοντέλων

Η επεξεργασία των τρισδιάστατων μοντέλων που αποτέλεσαν τα αντικείμενα του παιχνιδιού πραγματοποιείται σε προγράμματα που διατίθενται στο διαδίκτυο για ελεύθερη χρήση. Τα προγράμματα αυτά είναι:

- **Instant meshes**: αποτελεί ένα πρόγραμμα επαναπροσδιορισμού μιας τρισδιάστατης επιφάνειας σε ισοτροπική τριγωνική ή τετραγωνική επιφάνεια χρησιμοποιώντας έναν χειριστή που βελτιστοποιεί τους προσανατολισμούς των ακμών και τις θέσεις των σημείων. Έχει τη δυνατότητα να επεξεργαστεί νέφη σημείων, τριγωνικές επιφάνειες ή σαρώσεις ανεξάρτητα από το μέγεθός τους.
- **MeshLab**: αποτελεί ανοιχτό πρόγραμμα επεξεργασίας τριγωνικών επιφανειών. Είναι ιδανικό για την επισκευή και προετοιμασία αντικειμένων που μελλοντικά θα υποστούν τρισδιάστατη εκτύπωση.
- **Autodesk Meshmixer**: διαθέτει παρόμοιες λειτουργίες με το MeshLab και χρησιμοποιείται για τον ίδιο σκοπό. Χρησιμοποιήθηκε διότι προσέγγιζε με διαφορετικό τρόπο τις επιφάνειες και ο συνδυασμός των δύο εφαρμογών δίνει το καλύτερο δυνατό αποτέλεσμα

3.3: Εξοπλισμός εικονικής πραγματικότητας HTC Vive

Για την μεταφορά της εφαρμογής σε περιβάλλον εικονικής πραγματικότητας χρησιμοποιείται το σύστημα HTC Vive, το οποίο αναπτύχθηκε από τις εταιρείες HTC και Valve Corporation και κυκλοφόρησε τον Απρίλιο του 2016^[26]. Είναι συμβατό με λειτουργικά συστήματα Windows, Linux και MacOS και είναι σύστημα υψηλού κόστους.

Αποτελείται από:

- **Μάσκα εικονικής πραγματικότητας**, με δύο οθόνες οργανικών διόδων φωτοεκπομπής (OLED), μία για κάθε μάτι. Η κάθε οθόνη είναι ανάλυσης 1080 x 1200 pixel και η μάσκα παρέχει ρυθμό ανανέωσης 90Hz και οπτικό πεδίο 110 μοίρες. Επιπροσθέτως, διαθέτει μια εμπρόσθια κάμερα που επιτρέπει την παρατήρηση του φυσικού περιβάλλοντος από το χρήστη χωρίς την αφαίρεση της μάσκας και μπορεί να λειτουργήσει και ως βοήθημα αναγνώρισης εμποδίων.
- **Χειριστήρια** με πολλαπλούς τρόπους διάδρασης και εισαγωγής δεδομένων που λειτουργούν με μπαταρία διάρκειας έξι ωρών.

- **Βάσεις – Φάροι**, δύο συσκευές εκπομπής μη ορατού φωτός περιόδου 8,33 ms που βοηθούν στον εντοπισμό και την χωροθέτηση της μάσκας και των χειριστηρίων και ορίζουν τον εικονικό χώρο του συστήματος. Λειτουργούν με χρήση άορατων ακτινών λέιζερ και μιας ορθογώνιας διάταξης από λαμπτήρες LED υπέρυθρου φωτός.
- **Έξοδος ήχου** και ενσωματωμένο **μικρόφωνο**, που ενισχύουν την εμπειρία εμβύθισης.



Εικόνα 42: Το σύστημα εικονικής πραγματικότητας HTC Vive

Πέρα από όσα αναφέρονται πιο πάνω, διαθέτει επιπλέον εξαρτήματα που μπορεί να χρησιμοποιηθούν για καλύτερη εμπειρία χρήστη. Μερικά από αυτά είναι:

- **Vive Tracker**: αποτελεί ένα σύστημα εντοπισμού κίνησης που ενσωματώνεται στα χειριστήρια για να μπορεί να εντοπιστεί από τις βάσεις – φάρους. Έχει χρησιμοποιηθεί μέχρι σήμερα σε πολλές εφαρμογές και με πολλούς τρόπους, όπως για παράδειγμα ως ρακέτα σε παιχνίδια με αθλήματα.



Εικόνα 43: vive tracker

- Vive Deluxe Audio Strap: αποτελεί σύστημα ήχου ενσωματωμένο στη μάσκα, το οποίο αναβαθμίζει την εμπειρία εμβύθισης καθώς ο ήχος είναι πιο ποιοτικός και απομονώνει το χρήστη από εξωτερικά ερεθίσματα. Επιπλέον, ο τρόπος που είναι ενσωματωμένο στη μάσκα εξασφαλίζει καλύτερη στερεοσκοπική όραση.



Εικόνα 44: vive deluxe audio strap

- Vive Facial Tracker: αποτελεί μια επέκταση που κυκλοφόρησε το Μάρτιο του 2021 και αναγνωρίζει τις κινήσεις του στόματος του χρήστη και γενικότερα του χαμηλότερου μέρους του προσώπου.



Εικόνα 45: vive facial tracker

Μέχρι σήμερα έχουν γίνει πολλές εφαρμογές και αναβαθμίσεις του συστήματος HTC Vive (Vive Pro, Vive Pro Eye, Vive Focus, Vive Cosmos, Vive Pro 2) και έχει χρησιμοποιηθεί για την δημιουργία περίπου 100 παιχνιδιών.

3.3.1: Πλεονεκτήματα & Μειονεκτήματα του συστήματος HTC Vive^{[27][28]}

Το συγκεκριμένο σύστημα εικονικής πραγματικότητας έχει τη δυνατότητα χρήσης σε καθήμενη ή όρθια στάση ή ακόμα και σε κλίμακα δωματίου, γεγονός που το καθιστά ιδιαίτερα χρήσιμο για πολλές εφαρμογές και λογισμικά. Επιπλέον παρέχει τη δυνατότητα μη χρήσης χειριστηρίων και η κίνηση του χρήστη στο φυσικό περιβάλλον γίνεται χωρίς κίνδυνο, αφού διαθέτει την εμπρόσθια κάμερα η οποία δίνει ένα μονόχρωμο περιγράμμα του περιβάλλοντα χώρου.

Ωστόσο, η μάσκα έχει μεγάλο βάρος με αποτέλεσμα να δυσκολεύει αρκετά τον χρήστη. Ακόμη, απαιτούνται οι βάσεις – φάροι για οποιαδήποτε εφαρμογή, επομένως δεν είναι εύκολο να μεταφερθεί το σύστημα σε άλλο χώρο πέρα από εκείνον που θα εγκατασταθεί. Επιπλέον, το σύστημα έχει υψηλές απαιτήσεις σε χαρακτηριστικά ηλεκτρονικού υπολογιστή για να δώσει το επιθυμητό αποτέλεσμα. Φυσικά, τα χαρακτηριστικά του συστήματος Η/Υ εξαρτώνται άμεσα και από την πολυπλοκότητα της εφαρμογής, δηλαδή οι ταυτόχρονοι υπολογισμοί που πρέπει να εκτελεστούν χωρίς να παρεμποδιστεί η εμπύθιση.

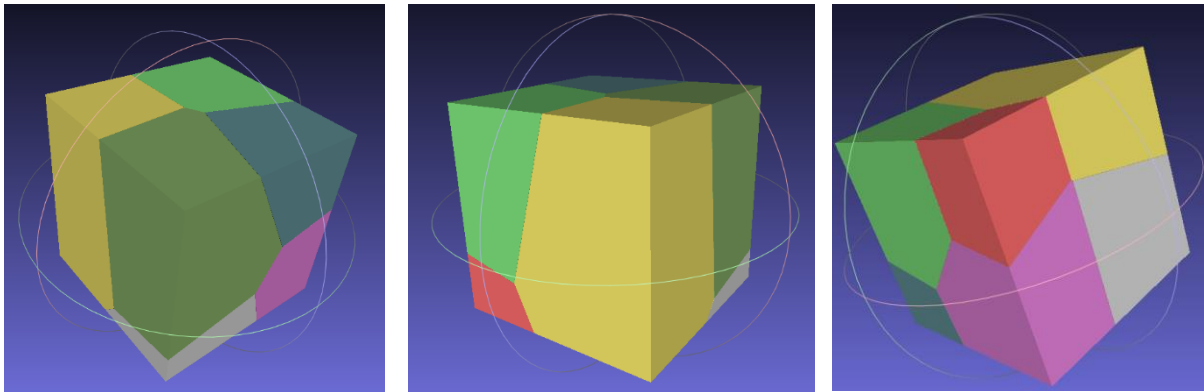
3.4: Πειραματική εφαρμογή

Όπως προαναφέρθηκε, προκειμένου να υλοποιηθεί η εφαρμογή που πραγματεύεται η διπλωματική εργασία προηγήθηκε πειραματική εφαρμογή σε διαφορετικό και απλούστερο αντικείμενο ώστε να στηθεί το λογικό τμήμα του παιχνιδιού. Το πειραματικό αντικείμενο αποτελεί ένας κύβος. Τα βήματα της διαδικασίας περιγράφονται αναλυτικά στην συνέχεια του κεφαλαίου.

3.4.1: Κατάτμηση του αντικειμένου

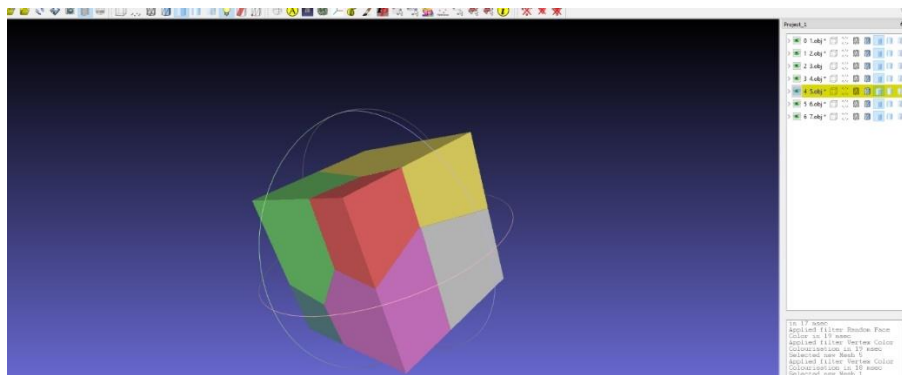
Χρησιμοποιώντας το εργαλείο Cell fracture της μηχανής παιχνιδιών Blender χωρίζεται ένας κύβος σε 7 τυχαία κομμάτια, τα οποία στον τρισδιάστατο χώρο βρίσκονται στην αρχική τους θέση, δηλαδή ενωμένα μεταξύ. Η θέση αυτή αποτελεί την αφετηρία του παιχνιδιού και σημείο αναφοράς.

Η κατάτμηση του κύβου φαίνεται στην εικόνα 47. Κάθε χρώμα αντιστοιχεί σε ένα κομμάτι.



Εικόνα 46: Τα 7 κομμάτια στα οποία χωρίστηκε το πειραματικό

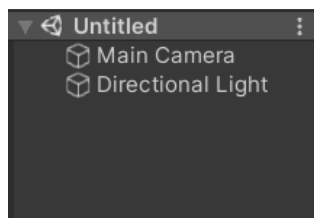
Στη συνέχεια, κάθε κομμάτι αποθηκεύεται στην κατάλληλη μορφή ώστε να εισαχθεί στην μηχανή παιχνιδιών Unity και να υλοποιηθεί το παιχνίδι.



Εικόνα 47: Τα κομμάτια αποθηκεύτηκαν σε μορφή .obj

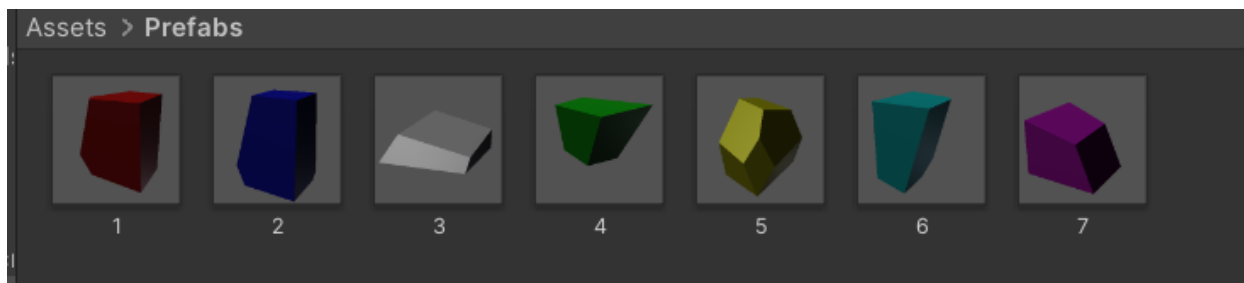
3.4.2: Υλοποίηση του παιχνιδιού

Ως πρώτο βήμα δημιουργείται στο περιβάλλον της Unity ένα νέο project με μία νέα σκηνή, η οποία έχει ως αρχικά δεδομένα μία κάμερα (main camera) και φωτισμό (directional light).



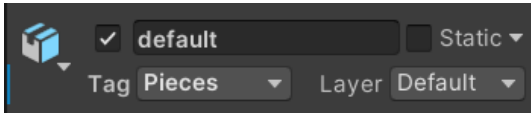
Εικόνα 48: Περιεχόμενα του καταλόγου σε μία νέα σκηνή

Τα 7 κομμάτια εισάγονται στην μηχανή παιχνιδιών Unity ως προκατασκευασμένα αντικείμενα (prefabs) κατευθείαν από τον φάκελο στον οποίο έχουν εξαχθεί μετά την επεξεργασία τους στην Blender και σε κάθε ένα αποδίδεται ένα διαφορετικό χρώμα.

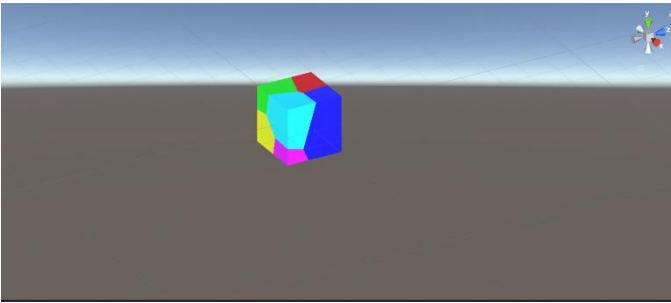


Εικόνα 49: Τα 7 κομμάτια του κύβου όπως εισήχθησαν στο περιβάλλον της Unity

Όλα τα κομμάτια τοποθετούνται σε μια προκαθορισμένη θέση, η οποία είναι και η θέση εκκίνησης του παιχνιδιού και έχουν την ίδια ετικέτα, "Pieces", ώστε να είναι εύκολο να εντοπιστούν.



Εικόνα 50: Η μορφή της ετικέτας των κομματιών



Εικόνα 51: Η θέση του κύβου στο τρισδιάστατο περιβάλλον της Unity

Στη συνέχεια, συντάσσεται ο κώδικας για την υλοποίηση του παιχνιδιού. Η λογική πάνω στην οποία στηρίζεται το παιχνίδι είναι, συνοπτικά, η εξής:

1. Σε έναν πίνακα *Initial Coordinates Array* (Πίνακας 3) αποθηκεύονται οι αρχικές θέσεις του κέντρου βάρους των κομματιών κατά X, Y και Z.
2. Όλα τα κομμάτια εκτός από ένα, έτσι ώστε να υπάρχει κομμάτι αναφοράς, διασκορπίζονται σε τυχαίες θέσεις στο χώρο εντός ενός πλαισίου μέγιστων και ελάχιστων τιμών προκειμένου να μην βρεθούν εκτός του πεδίου ορατότητας του χρήστη.
3. Ακολουθώντας έναν μηχανισμό drag and drop, τα κομμάτια τοποθετούνται σε νέες θέσεις τις οποίες ορίζει ο χρήστης με το ποντίκι.
4. Όταν ένα κομμάτι είναι στη σωστή του θέση, δηλαδή την αρχική, κλειδώνει η δυνατότητα drag and drop για αυτό και ο παίκτης συνεχίζει με τα υπόλοιπα, μέχρι όλα να είναι στη σωστή θέση. Φυσικά, για λόγους ευκολίας, ο έλεγχος της θέσης του κομματιού γίνεται με περιθώρια σφάλματος, ώστε να λαμβάνεται σωστά η θέση ακόμα και αν είναι αρκετά κοντά και όχι ακριβώς.

Η αλλαγή της θέσης των κομματιών συμπληρώνεται σε επόμενο στάδιο με αλλαγή του προσανατολισμού τους. Συγκεκριμένα, τα κομμάτια με αποθηκευμένο τον αρχικό τους προσανατολισμό στρίβουν σε τυχαίες γωνίες γύρω από τους άξονες X, Y, Z και ο παίκτης πέρα από το να τα τοποθετήσει στη σωστή θέση καλείται και να τα στρέψει κατάλληλα

προκειμένου να ολοκληρωθεί το παιχνίδι. Όπως στις θέσεις, έτσι και στους προσανατολισμούς, ορίζεται ένα πλαίσιο περιορισμού των τυχαίων τιμών και η ανίχνευση των νέων προσανατολισμών γίνεται με μεγάλα περιθώρια σφάλματος ως προς τις αρχικές, για πιο εύκολη τοποθέτηση.

Όλα τα βήματα αναλύονται περαιτέρω στις επόμενες ενότητες, συμπληρωματικά με κομμάτια του κώδικα σε C# γραμμένου σε περιβάλλον Visual Studio 2019.

Initial Coordinates Array

Αρχικά, εντοπίζονται στο περιβάλλον της Unity όλα τα κομμάτια με την ετικέτα "Pieces" και αποθηκεύονται σε ένα πίνακα. Η κίνηση αυτή επιτρέπει, εάν δημιουργηθούν περισσότερα κομμάτια, να ανανεώνεται αυτόματα το μέγεθος του πίνακα με την τοποθέτηση αυτών κάτω από την ίδια ετικέτα και να μην χρειάζεται να γίνεται χειροκίνητα η αλλαγή. Στην συνέχεια, ο πίνακας των Game Objects μετατρέπεται σε πίνακα μετασχηματισμού (transform) ώστε να αποθηκευτούν σε αυτόν οι αρχικές θέσεις. Τελικά, δημιουργείται ο πίνακας θέσης (positions), που περιλαμβάνει τις αρχικές θέσεις των κομματιών του κύβου.

Πίνακας 3: Κομμάτι του κώδικα για την δημιουργία του πίνακα θέσεων

```
public void Start()
{
    //creating a Transform Object from a GameObject
    GameObject[] cubes = GameObject.FindGameObjectsWithTag("Pieces");
    Transform[] cubes_position = new Transform[cubes.Length];

    for (int i = 0; i < cubes.Length; i++)
    {
        cubes_position[i] = cubes[i].transform;
    }

    positions = new Vector3[cubes_position.Length];
    for (int i = 0; i < cubes_position.Length; i++)
    {
        positions[i] = cubes_position[i].position;

        //Debug.Log(positions[i]);
    }
}
```

Σημειώνεται πως με τον όρο Vector3 ορίζονται μαθηματικά μοντέλα του τρισδιάστατου χώρου που περιγράφουν διανύσματα με κατεύθυνση και μήκος αλλά χωρίς θέση, σημεία στον τρισδιάστατο χώρο, χρώμα (RGB) και γενικώς οντότητες με τρεις παραμέτρους, συνήθως δεκαδικούς αριθμούς. Χρησιμοποιούνται κυρίως για να αλλάξουν οι θέσεις των αντικειμένων στο χώρο. Το μήκος των διανυσμάτων Vector3 δίνεται από τον μαθηματικό τύπο:

$$\text{magnitude} = \sqrt{x^2 + y^2 + z^2}$$

- ✓ Με την εξέλιξη της πειραματικής εφαρμογής διαπιστώθηκε πως δεν είναι απαραίτητη η αποθήκευση των αρχικών συντεταγμένων σε πίνακα, διότι οι προκαθορισμένες θέσεις του κέντρου βάρους των κομματιών είναι (0,0,0) και επομένως η σύγκριση κατά το βήμα drag and drop μπορεί να γίνει απευθείας.

Random Positions

Επόμενο στάδιο αποτελεί η τυχαιοποίηση των θέσεων των κομματιών του κύβου εντός ενός συγκεκριμένου εύρους τιμών. Αναλυτικότερα, ορίζεται μια συνάρτηση (randomizePiecesPositions) η οποία όταν κληθεί μετατρέπει τον πίνακα που περιέχει τα αντικείμενα σε λίστα προκειμένου να είναι δυνατό να εξαιρεθεί ένα από την τυχαιοποίηση για να υπάρχει το σημείο αναφοράς, και καθορίζει το εύρος των τιμών που θα πάρουν οι νέες συντεταγμένες. Πιο συγκεκριμένα, ορίζονται τρεις μεταβλητές (x_a , y_a , z_a) με τυχαίες τιμές από 0 έως 1. Εάν η τυχαία τιμή είναι μεγαλύτερη από το 0.5, δίνεται στην μεταβλητή η τιμή -1, ενώ αν είναι μικρότερη η τιμή 1. Για μεγαλύτερη τυχαιοποίηση, η μεταβλητή y_a λειτουργεί αντίθετα, δηλαδή παίρνει την τιμή -1 εάν είναι μικρότερη από το 0.5. Οι τρεις αυτές μεταβλητές πολλαπλασιάζονται αντίστοιχα με τις μεταβλητές x , y , z , οι οποίες παίρνουν τυχαίες δεκαδικές τιμές από το 0.5 έως το 3. Τέλος, κάθε κομμάτι αλλάζει θέση και μεταφέρεται σε ένα διαφορετικό σημείο (x , y , z). Η διαδικασία φαίνεται στον **παρακάτω** πίνακα.

Πίνακας 4: Η διαδικασία τυχαιοποίησης των κομματιών

```
void randomizePiecesPositions()
{
    //converting array of gameobjects to list
    list = pieces.ToList();
    foreach (GameObject piece in list.Skip(1))
    {
        float xa = Random.Range(0.0f, 1.0f) > 0.5f ? -1 : 1; //if >0.5, result= -1,
alternatively result = 1
        float ya = Random.Range(0.0f, 1.0f) < 0.5f ? -1 : 1;
        float za = Random.Range(0.0f, 1.0f) > 0.5f ? -1 : 1;

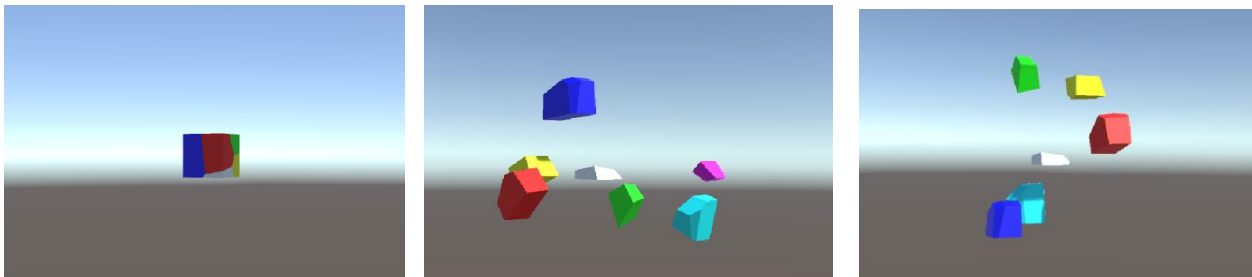
        float x = Random.Range(0.5f, 3.0f) * xa;
        float y = Random.Range(0.5f, 3.0f) * ya;
        float z = Random.Range(0.5f, 3.0f) * za;

        piece.transform.position = new Vector3(x, y, z);
    }
}
```

Για καλύτερη οπτικοποίηση της διαδικασίας τυχαιοποίησης, επιλέγεται να κληθεί η συνάρτηση και άρα ο κύβος να «σπάσει» και τα κομμάτια να αλλάξουν θέση, όταν ο χρήστης πατάει το πλήκτρο “space” του πληκτρολογίου.

Πίνακας 5: Εκκίνηση της τυχαιοποίησης με το space

```
if (!isSpacePressed && Input.GetKeyDown(KeyCode.Space))
{
    isSpacePressed = true;
    randomizePiecesPositions();
}
```



Εικόνα 52: η μορφή του κύβου πριν (αριστερά) και μετά (μεσαία και δεξιά) την τυχαιοποίηση.

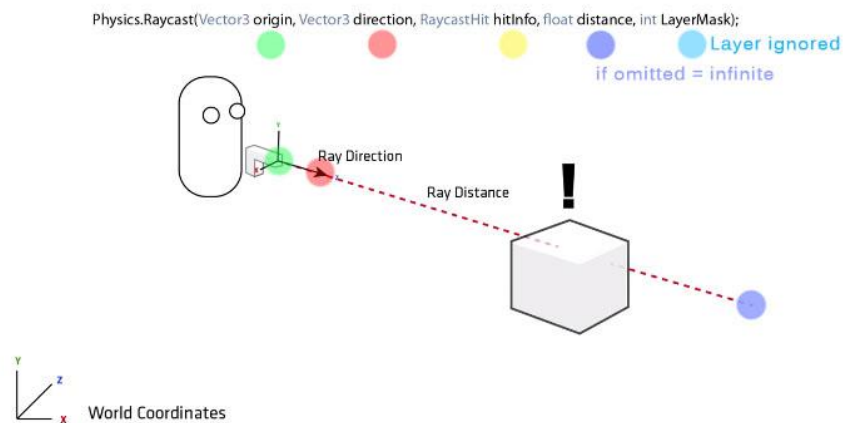
Στην εικόνα 52 φαίνεται πως κάθε φορά που ξεκινάει το παιχνίδι τα κομμάτια παίρνουν και νέες τυχαίες θέσεις, ενώ το κομμάτι λευκού χρώματος παραμένει πάντα στην ίδια (αποτελεί στο σημείο αναφοράς). Ωστόσο, η θέση μπορεί να αλλάξει με το space μόνο μία φορά, στην αρχή του παιχνιδιού. Η λειτουργία κλειδώνει με την πρώτη τυχαιοποίηση, ώστε να μην αλλάξουν οι θέσεις των κομματιών κατά λάθος όταν ο παίκτης έχει τοποθετήσει κάποια κομμάτια σωστά. Ο έλεγχος πραγματοποιείται με χρήση της Boolean μεταβλητής

“isSpacePressed”, η οποία παίρνει την τιμή «Αληθές» όταν γίνει η τυχαιοποίηση και όσο είναι αληθής, δεν λειτουργεί η συνάρτηση.

Drag and Drop

Αφού τα κομμάτια έχουν βρεθεί σε τυχαίες θέσεις στο χώρο, ο χρήστης καλείται να τα τοποθετήσει ξανά στις σωστές για να ενώσει το αντικείμενο στην αρχική του μορφή. Η δυνατότητα αυτή παρέχεται μέσω της λειτουργίας drag and drop. Αυτό σημαίνει πως χρησιμοποιώντας το ποντίκι ο παίκτης παίρνει κάθε ένα κομμάτι και το τοποθετεί σε νέες θέσεις που θεωρεί ότι είναι οι σωστές.

Η ανίχνευση της επιλογής του κομματιού γίνεται με χρήση ακτινών (raycast) παρέχοντας στο πρόγραμμα μια αφετηρία της ακτίνας, μια κατεύθυνση και μια «μάσκα» ώστε να φιλτράρονται τα αντικείμενα που δεν συμπεριλαμβάνονται στην επιθυμητή επιλογή. Η αφετηρία της ακτίνας καθορίζεται από την θέση του ποντικιού όπως το βλέπει η κάμερα του παιχνιδιού, η κατεύθυνση αυτόματα από τη βιβλιοθήκη της C# και ο προορισμός της ακτίνας από το αντικείμενο – στόχο, δηλαδή το αντικείμενο που επιλέγει ο χρήστης με το κλικ από το ποντίκι.



Εικόνα 53: Η λογική λειτουργίας της ακτίνας στην Unity

Πίνακας 6: Ανίχνευση του αντικειμένου με χρήση ακτινών

```
//Method to Return Clicked Object
GameObject ReturnClickedObject(out RaycastHit hit)
{
    GameObject target = null;
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    if (Physics.Raycast(ray.origin, ray.direction * 10, out hit))
    {
        target = hit.collider.gameObject;
    }
    return target;
}
```

Μόλις καθοριστεί η ακτίνα της ανίχνευσης, ακολουθούν δύο βήματα: πρώτον, να οριστεί ότι η ανίχνευση ξεκινά όταν ο παίκτης κρατάει πατημένο το αριστερό κλικ του ποντικιού και δεύτερον να αλλάζει η θέση του επιλεγμένου αντικειμένου όταν το σέρνει. Για το δεύτερο βήμα ορίζεται μια συνάρτηση που παίρνει τιμές «αληθές» ή «ψευδές» (Boolean) και ανιχνεύει τότε το ποντίκι κινείται. Όταν η συνάρτηση είναι σωστή, ορίζονται οι νέες θέσεις των κομματιών ανάλογα με τη θέση του ποντικιού κάθε φορά.

Πίνακας 7: Ανίχνευση κίνησης ποντικιού και εισαγωγή νέων συντεταγμένων για κάθε κομμάτι που επιλέγεται

```
if (isMouseDowning)
{
    //tracking mouse position.
    Vector3 currentScreenSpace = new Vector3(Input.mousePosition.x,
    Input.mousePosition.y, positionOfScreen.z);

    //converting screen position to world position with offset changes.
    //Vector3 currentPosition = Camera.main.ScreenToWorldPoint(currentScreenSpace)
    + offsetValue;

    float disX = Input.mousePosition.x - posX;
    float disY = Input.mousePosition.y - posY;
    float disZ = Input.mousePosition.z - posZ;
    Vector3 currentPosition = Camera.main.ScreenToWorldPoint(new Vector3(disX,
    disY, disZ));
}
```

Η παραπάνω συνάρτηση καλείται μαζί με την ανίχνευση του στόχου, δηλαδή του αντικειμένου που θέλει να μετακινήσει ο χρήστης, η οποία πραγματοποιείται όταν είναι πατημένο το αριστερό κλικ του ποντικιού, το οποίο συμβολίζεται με τον αριθμό «0» στην C#.

Όταν το αριστερό κλικ δεν είναι πια πατημένο, δεν γίνεται καμία κίνηση στα κομμάτια, δηλαδή η συνάρτηση είναι ψευδής.

Πίνακας 8: Ανίχνευση αριστερού κλικ και κίνησης ποντικιού

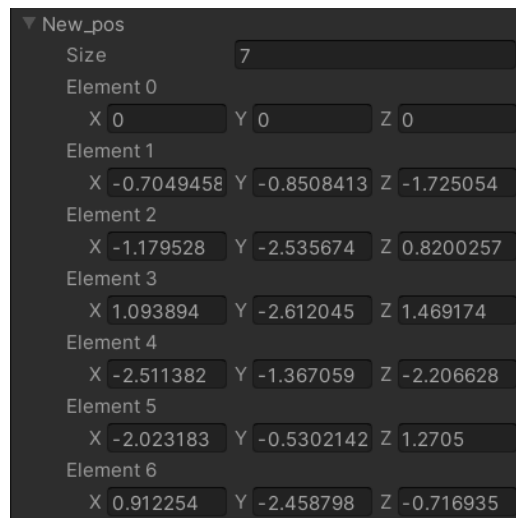
```
if (Input.GetMouseButtonDown(0))
{
    RaycastHit hitInfo;
    getTarget = ReturnClickedObject(out hitInfo);
    if (getTarget != null)
    {
        if (Input.GetMouseButtonDown(0))
        {
            isMouseDragging = true;
        }
    }
}
if (Input.GetMouseButtonUp(0))
{
    isMouseDragging = false;
}
```

Προκειμένου να ολοκληρωθεί η διαδικασία του Drag and Drop, οι νέες θέσεις των κομματιών πρέπει να συγκριθούν με τις αρχικές ώστε να εντοπιστεί εάν είναι σωστά τοποθετημένα. Για να γίνει αυτό ορίζεται μια μεταβλητή (defPosition) στην οποία αποθηκεύονται οι αρχικές θέσεις (0,0,0) και μια μεταβλητή που μετράει την απόσταση ανάμεσα στην τρέχουσα θέση και την αρχική. Εάν η απόσταση είναι μικρότερη από 0.45, τότε το κομμάτι πηγαίνει αυτόματα στην σωστή του θέση ενώ αν είναι μεγαλύτερη μένει στη θέση που το τοποθετεί ο χρήστης. Όταν τοποθετηθεί το κομμάτι στην σωστή θέση, ο χρήστης δεν μπορεί πια να το μετακινήσει.

Πίνακας 9: Έλεγχος της απόστασης από την default θέση των κομματιών

```
// If the distance is less than 0.45 then set it to (0,0,0) and you can't move it anymore.  
Else move it  
  
if (distance < 0.45f)  
    {  
        getTarget.transform.position = defPostion;  
    }  
else  
    {  
        //It will update target gameobject's current postion.  
        getTarget.transform.position = currentPosition;  
    }
```

Για λόγους οπτικοποίησης των νέων θέσεων, δημιουργείται δυναμικός πίνακας που αποθηκεύει κάθε αλλαγή που πραγματοποιείται στις θέσεις. Ο πίνακας αυτός έχει σειρές όσες τα κομμάτια που βρίσκονται κάτω από την ίδια ετικέτα ('pieces') και στήλες X, Y, Z. Η μορφή του πίνακα στο περιβάλλον του παιχνιδιού φαίνεται στην Εικόνα 54. Το κομμάτι με συντεταγμένες (0,0,0) είναι εκείνο που χρησιμοποιείται ως σημείο αναφοράς.



Element	X	Y	Z
Element 0	0	0	0
Element 1	-0.7049458	-0.8508413	-1.725054
Element 2	-1.179528	-2.535674	0.8200257
Element 3	1.093894	-2.612045	1.469174
Element 4	-2.511382	-1.367059	-2.206628
Element 5	-2.023183	-0.5302142	1.2705
Element 6	0.912254	-2.458798	-0.716935

Εικόνα 54: Μορφή του πίνακα με τις νέες θέσεις των κομματιών

Rotations

Όπως προαναφέρθηκε, η αλλαγή της θέσης των κομματιών συμπληρώθηκε στην πορεία του πειράματος με ταυτόχρονη αλλαγή του προσανατολισμού τους. Η λογική είναι ίδια με την αλλαγή της θέσης και χρησιμοποιείται το δεξί κλικ του ποντικιού για τις περιστροφές των κομματιών. Αναλυτικότερα, για τις στροφές χρησιμοποιείται το μαθηματικό μοντέλο Quaternion, που ορίζεται με τέσσερις παραμέτρους (x, y, z, w), των οποίων οι τιμές κυμαίνονται μεταξύ -1 και 1. Οι τρεις πρώτες προσδιορίζουν τις στροφές γύρω από τους άξονες x, y και z ενώ η τέταρτη αντιστοιχεί σε ένα διάνυσμα και δεν επηρεάζει άμεσα τις γωνίες. Το Quaternion μετριέται σε μονάδες (units) και δεν αντιστοιχεί σε μοίρες ή ακτίνια. Συχνά για να χρησιμοποιηθεί μετατρέπεται σε Euler Angles που μετρούνται σε μοίρες, ωστόσο υπάρχει δυσκολία αντιστοίχισης των γωνιών που χρησιμοποιεί η μηχανή παιχνιδιών Unity και των γωνιών Euler καθώς το (0,0,0) της Unity αντιστοιχεί σε (180,180,180) σε γωνίες Euler, επομένως αυτή η μετατροπή συνήθως αποφεύγεται.

Κατά τη διαδικασία τυχαιοποίησης των συντεταγμένων των κομματιών όπως φαίνεται στον πίνακα 10 πραγματοποιείται και στροφή αυτών γύρω από τον άξονα X με τυχαίες γωνίες από 90 έως 180 μοίρες. Η στροφή γύρω από τους άλλους άξονες δεν γίνεται για λόγους απλοποίησης της εφαρμογής.

Πίνακας 10: Τυχαιοποίηση θέσεων και προσανατολισμού των κομματιών

```
piece.transform.Rotate(Random.Range(90.0f, 180.0f), 0.0f, 0.0f);
piece.transform.position = new Vector3(x, y, z);
```

Η επιλογή των κομματιών ανιχνεύεται όπως αναφέρεται στην ενότητα Drag and Drop, δηλαδή με τη χρήση ακτινών. Για να πραγματοποιηθούν οι στροφές δημιουργείται και πάλι μια συνάρτηση με τιμές «αληθές» ή «ψευδές» που ανιχνεύει εάν το ποντίκι πραγματοποιεί περιστροφές. Όταν η συνάρτηση είναι αληθής, το επιλεγμένο κομμάτι περιστρέφεται γύρω από τους άξονες X ή Y ανάλογα με τη θέση του ποντικιού με μία συγκεκριμένη ταχύτητα η οποία μπορεί να αλλάξει και από τον χρήστη. Με το ποντίκι δεν είναι εφικτό να πραγματοποιηθεί περιστροφή ή κίνηση στον άξονα Z.

Rot Speed 100

Εικόνα 55: Η ταχύτητα περιστροφής των κομματιών

Πίνακας 11: Ανίχνευση περιστροφής ποντικιού και στρέψη του κομματιού κατάλληλα

```
if (isMouseRotating)
{
    float rotx = Input.GetAxis("Mouse X") * rotSpeed;
    float roty = Input.GetAxis("Mouse Y") * rotSpeed;
    getTarget.transform.Rotate(-rotx, roty, 0);
}
```

Σε αυτό το σημείο, ταυτόχρονα με τον έλεγχο που πραγματοποιείται στον Πίνακας 8 για το πατημένο αριστερό κλικ, πραγματοποιείται αντ' αυτού αντίστοιχος έλεγχος για πατημένο δεξί κλικ, καθώς αυτό χρησιμοποιείται για τις περιστροφές των κομματιών. Αναλυτικότερα, το πρόγραμμα ανιχνεύει εάν ο χρήστης κρατάει πατημένο το αριστερό (τιμή 0) ή το δεξί κλικ (τιμή 1) και ανάλογα καλεί την αντίστοιχη συνάρτηση. Ο παρακάτω πίνακας δείχνει τον ολοκληρωμένο έλεγχο για το ποντίκι, όταν είναι και όταν δεν είναι πατημένο κάποιο από τα δύο κουμπιά.

Πίνακας 12: Ολοκληρωμένος έλεγχος για δεξί ή αριστερό κλικ και επίκληση αντίστοιχης συνάρτησης

```
if (Input.GetMouseButtonDown(0) || Input.GetMouseButtonDown(1))
{
    RaycastHit hitInfo;
    getTarget = ReturnClickedObject(out hitInfo);
    if (getTarget != null)
    {
        if (Input.GetMouseButtonDown(0))
        {
            isMouseDragging = true;
        }

        else if (Input.GetMouseButtonDown(1))
        {
            isMouseRotating = true;
        }
    }
}

if (Input.GetMouseButtonUp(0))
{
    isMouseDragging = false;
}

if (Input.GetMouseButtonUp(1))
{
    isMouseRotating = false;
}
```

Τέλος, όπως και στην αλλαγή των θέσεων, πραγματοποιείται έλεγχος για την διαφορά προσανατολισμού των κομματιών όπως τα τοποθετεί ο χρήστης από τον προκαθορισμένο. Για το σκοπό αυτό ορίζεται μια αρχική τιμή Quaternion (0,0,0,1) και συγκρίνεται κάθε νέα γωνία στροφής με αυτήν. Το αποτέλεσμα της σύγκρισης προκύπτει σε μοίρες και εάν είναι μικρότερο από 25°, το κομμάτι παίρνει αυτόματα τον σωστό του προσανατολισμό και η δυνατότητα περιστροφής αυτού τερματίζει.

Πίνακας 13: Έλεγχος στροφής από τον default προσανατολισμό των κομματιών

```
Quaternion quaternion_def = new Quaternion(0, 0, 0, 1);

angle = Quaternion.Angle(getTarget.transform.localRotation, quaternion_def); //angle in
degrees from local rotation to default rotation

if (angle < 25.0f)
{
    getTarget.transform.localRotation = quaternion_def;
    isMouseRotating = false;
}
```

Για λόγους οπτικοποίησης και πάλι, οι τυχαίες στροφές που παίρνουν τα κομμάτια κατά την εκκίνηση και οι νέες που τα τοποθετεί ο χρήστης φαίνονται σε πίνακα παρόμοιο με εκείνο των θέσεων, με γραμμές όσες τα κομμάτια και στήλες x, y, z και w. Επίσης, είναι ορατή και η γωνία απόκλισης του κομματιού από την προκαθορισμένη του θέση.

▼ Element 3	
X	0.5640113
Y	-0.4080814
Z	-0.5407232
W	0.4722069

Angle	116.8441
-------	----------

Εικόνα 56: Τυχαίες τιμές Quaternion κατά τη στροφή ενός κομματιού (πάνω) και η απόκλιση του σε μοίρες από την αρχική γωνία στροφής (κάτω)

Το πειραματικό παιχνίδι ολοκληρώνεται όταν ο χρήστης έχει τοποθετήσει τα κομμάτια στη σωστή τους θέση και με σωστό προσανατολισμό.

Συμπληρωματικές λειτουργίες

Αφού ολοκληρώνονται οι βασικές δυνατότητες του παιχνιδιού όπως περιγράφονται παραπάνω, εμπλουτίζονται με λεπτομέρειες για καλύτερη εμπειρία του χρήστη.

1. Περιστροφή της κάμερας στην αρχή του παιχνιδιού

Όταν το παιχνίδι ξεκινάει και πριν γίνει το «σπάσιμο» του κύβου σε κομμάτια, η κάμερα του παιχνιδιού περιστρέφεται γύρω από το αντικείμενο ώστε να παρέχει στον χρήστη πλήρη εποπτεία της μορφής του και της σωστής θέσης των κομματιών. Η στροφή γίνεται γύρω από τον άξονα Y με προκαθορισμένη ταχύτητα.

Πίνακας 14: Στροφή της κάμερας γύρω από το αντικείμενο

```
void Update()
{
    cam.transform.RotateAround(target.transform.position, Vector3.up, Time.deltaTime
* speed);
}
```

Η λειτουργία αυτή ενεργοποιείται με το που γίνεται εκκίνηση του παιχνιδιού και τερματίζει όταν ο χρήστης πατήσει το πλήκτρο “space”, όπου και γίνεται η τυχαιοποίηση θέσεων και προσανατολισμών. Καθώς ο κώδικας της περιστροφής συντάσσεται σε διαφορετικό αρχείο από τον υπόλοιπο κώδικα, πρέπει να κληθεί κατάλληλα για την απενεργοποίησή του.

Πίνακας 15: Τερματισμός της περιστροφής με το space

```
if (!isSpacePressed && Input.GetKeyDown(KeyCode.Space))
{
    isSpacePressed = true;
    randomizePiecesPositions();
    c.GetComponent<CameraRotator>().enabled = false;
}
```

2. Κίνηση της κάμερας κατά τη διάρκεια του παιχνιδιού

Καθώς το πειραματικό παιχνίδι αναπτύσσεται σε υπολογιστή και οι κινήσεις πραγματοποιούνται μόνο με το ποντίκι, δεν είναι εφικτή η μετακίνηση των κομματιών στην τρίτη διάσταση, δηλαδή στον άξονα Z. Για να επιλυθεί αυτό το πρόβλημα, ενεργοποιείται η δυνατότητα περιστροφής της κάμερας γύρω από τον άξονα Y του αντικειμένου, ώστε να μπορεί ο χρήστης να το πλησιάσει από όλες τις πλευρές και να τοποθετήσει τα κομμάτια κατάλληλα. Η περιστροφή αυτή γίνεται μέσω των πλήκτρων Q και E του πληκτρολογίου σύμφωνα με μια καθορισμένη ταχύτητα.

Πίνακας 16: Στροφή της κάμερας με το πληκτρολόγιο

```

if (Input.GetKey(KeyCode.Q))
    {
        transform.RotateAround(point, Vector3.up, Time.deltaTime * rotationSpeed);
    }
    ναι}
if (Input.GetKey(KeyCode.E))
    {
        transform.RotateAround(point, -Vector3.up, Time.deltaTime * rotationSpeed);
    }
}

```

Πέρα από την περιστροφή, δίνεται στον χρήστη και η δυνατότητα να μετακινήσει την κάμερα οριζόντια ή κατακόρυφα για καλύτερη εποπτεία του αντικειμένου και να πλησιάσει ή απομακρυνθεί από αυτό (zoom in & zoom out). Οι οριζόντιες και κατακόρυφες κινήσεις ορίζονται με τα πλήκτρα WASD ενώ η εστίαση ρυθμίζεται από την ροδέλα του ποντικιού.

Πίνακας 17: Zoom in και zoom out με την ροδέλα του ποντικιού

```

//zooming in and out
if (Input.GetAxis("Mouse ScrollWheel") < 0)
    {
        transform.Translate(0, Y, -1);
    }
if (Input.GetAxis("Mouse ScrollWheel") > 0)
    {
        transform.Translate(0, B, 1);
    }
}

```

Πίνακας 18: Μετακίνηση της κάμερας με τα πλήκτρα WASD

```
//moving WASD
if (Input.GetKey(KeyCode.D))
{
    transform.Translate(new Vector3(speed * Time.deltaTime, 0, 0));
}

if (Input.GetKey(KeyCode.A))
{
    transform.Translate(new Vector3(-speed * Time.deltaTime, 0, 0));
}

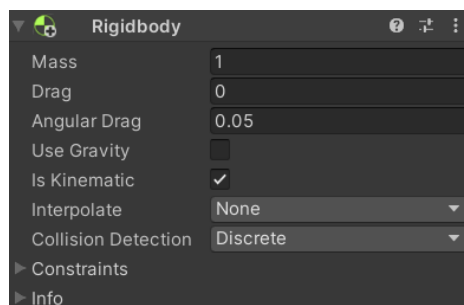
if (Input.GetKey(KeyCode.W))
{
    transform.Translate(new Vector3(0, speed * Time.deltaTime, 0));
}

if (Input.GetKey(KeyCode.S))
{
    transform.Translate(new Vector3(0, -speed * Time.deltaTime, 0));
}
```

Physics

Για να είναι εφικτή η επιλογή του αντικειμένου από τον χρήστη και για να μην εισέρχεται το ένα κομμάτι μέσα στο άλλο αλλά να αντιμετωπίζονται σαν κανονικά στερεά αντικείμενα, είναι απαραίτητη η εισαγωγή φυσικών χαρακτηριστικών. Συγκεκριμένα, σε κάθε κομμάτι τοποθετούνται εξαρτήματα για:

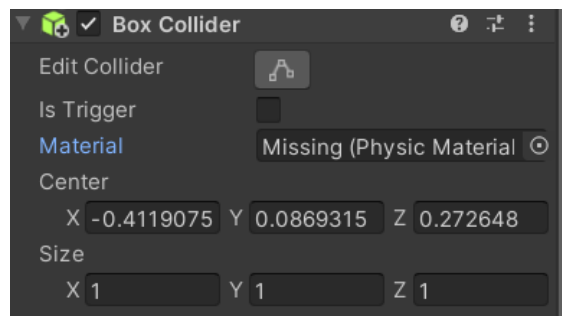
- ✓ **Rigidbody:** αποτελούν εξαρτήματα τα οποία επιτρέπουν στα αντικείμενα να υπακούν στους νόμους της φυσικής. Περιλαμβάνει τις αντιδράσεις σε δυνάμεις, στη βαρύτητα, στην μετακίνηση στο χώρο (drag) και την ορμή.



Εικόνα 57: Rigidbody component

Όπως φαίνεται στην Εικόνα 57, για κάθε rigidbody ορίζεται η μάζα του σε κιλά (*mass*), η αντίσταση του αέρα κατά την μετακίνηση του αντικείμενου από άλλες δυνάμεις (*drag*) η οποία όταν ισούται με το μηδέν σημαίνει πως δεν υπάρχει αντίσταση αέρα ενώ εάν πάρει άπειρη τιμή σημαίνει ότι το αντικείμενο σταματάει απευθείας να μετακινείται και αντίστοιχα ορίζεται τιμή αντίστασης αέρα για την περιστροφή του αντικείμενου (*angular drag*). Στην εικόνα φαίνονται οι προεπιλεγμένες τιμές για τις συγκεκριμένες μεταβλητές. Επιπλέον, αποφασίζεται εάν θα γίνει χρήση της βαρύτητας ή όχι, στην προκειμένη περίπτωση είναι από-επιλεγμένο, καθώς η επιλογή θα έκανε τα κομμάτια να πέφτουν και να χάνονται από την οθόνη και θα δημιουργούσε δυσκολία στον χρήστη να τα μετακινήσει. Η επιλογή “*Is Kinematic*” σημαίνει πως το αντικείμενο δε θα υπακούει στους νόμους της φυσικής όπως ορίζονται από τη μηχανή παιχνιδιών αλλά θα μπορεί να ελεγχθεί μόνο από εντολές που συμπεριλαμβάνουν την θέση του (“*Transform*”) και για αυτό το λόγο είναι ενεργοποιημένο στην συγκεκριμένη εφαρμογή. Η επιλογή “*Interpolate*” ενεργοποιείται όταν είναι επιθυμητή η αστάθεια ενός αντικείμενου, και για αυτό εδώ είναι συμπληρωμένο ως «καμία» (none). Τέλος, η ανίχνευση σύγκρουσης (*collision detection*) είναι ορισμένη ως «διακριτή» (discrete) διότι είναι επιθυμητή η ανίχνευση ως προς όλα τα άλλα αντικείμενα της σκηνής. Οι υπόλοιπες επιλογές καθορίζουν εάν το αντικείμενο θα μείνει ακίνητο ως προς ένα άξονα X, Y ή Z, αν κινείται με μια συγκεκριμένη ταχύτητα και γενικότερες πληροφορίες που αφορούν το κέντρο βάρους του αντικείμενου.

- ✓ **Box Collider:** αποτελεί ένα κυβοειδές πρωτόγονο σύγκρουσης και χρησιμοποιείται για ορισμό υλικού σε ένα αντικείμενο, μεγέθους, θέσης, και για τον καθορισμό της πηγής ανίχνευσης των φυσικών κανόνων.



Εικόνα 58: Box Collider

ΚΕΦΑΛΑΙΟ 4: Αντικείμενα της εφαρμογής

Στην παρούσα ενότητα αναλύονται τα ιστορικά στοιχεία των δύο αντικειμένων πάνω στα οποία συντάσσεται το παιχνίδι, δηλαδή του αρχαίου ναού της Δήμητρας στην Νάξο και του αγάλματος «Αυτοκράτορας και Δούλος» που βρίσκεται στο μουσείο της Αρχαίας Αγοράς των Αθηνών.

4.1: Ναός της Δήμητρας

Ο ναός της Δήμητρας^{[29][30]} εντοπίζεται στην Νάξο, στον Γύρουλα Σαγκρίου και είναι αφιερωμένος στον Απόλλωνα, την Δήμητρα και την κόρη της Περσεφόνη. Χρονολογείται στο 530 π. Χ, δηλαδή περίπου έναν αιώνα πριν τον Παρθενώνα του οποίου θεωρείται και πρόδρομος. Εκείνη την περίοδο τύραννος του νησιού ήταν ο Λύγδαμης, ο οποίος ανέπτυξε πολύ την περιοχή αρχιτεκτονικά. Ο ναός είναι χτισμένος εξ ολοκλήρου από λευκό μάρμαρο Νάξου και εντάσσεται στον ιωνικό ρυθμό, με κάποιες παραλλαγές. Η τοποθεσία του στην εύφορη κοιλάδα αιτιολογείται από το γεγονός ότι η θεά Δήμητρα, την οποία και τιμούσαν με θρησκευτικές τελετές, είναι η θεά της γεωργίας. Τα πρώτα χριστιανικά χρόνια ο ναός μετατράπηκε σε βασιλική εκκλησία αφιερωμένη στον Άγιο Ιωάννη το Θεολόγο. Πλέον έχει αναστηλωθεί και στον εσωτερικό του χώρο λειτουργεί μουσείο.



Εικόνα 59: Ο Ναός της Δήμητρας στην Νάξο

4.1.1: Ιστορική ανασκόπηση

Ο ναός ανακαλύπτεται το 1949 από τον Νικόλαο Κοντολέοντα, όποτε και ξεκίνησε μια σειρά ανασκαφών στην περιοχή. Ο ναός βρισκόταν κάτω από την εκκλησία του Αγίου Ιωάννη και αποτέλεσε έναν από τους λόγους που πραγματοποιήθηκαν ανασκαφές στην περιοχή, καθώς είναι αρκετά συχνό να χρησιμοποιούνται κομμάτια αρχαίων ναών σε βυζαντινές εκκλησίες. Από το 1976 έως το 1985 πραγματοποιείται μια σειρά εργασιών για έρευνα του ναού και των γειτονικών περιοχών υπό την διεύθυνση των Βασίλη Λαμπρινουδάκη και Γκότφριντ Γκρούμπεν του Πανεπιστημίου Αθηνών και του Τεχνικού Πανεπιστημίου του Μιλάνου. Τα ευρήματα των ανασκαφών έχουν συλλεχθεί στο μουσείο «Γύρουλα».

Όταν οι έρευνες επεκτάθηκαν εντός του ναού διαπιστώθηκε πως μεταξύ αυτού και της μετέπειτα βυζαντινής εκκλησίας υπήρχαν τρία στρώματα. Το πρώτο αποτελείται από χοντρή λατύπη (μικρό κομμάτι πέτρας) και κατεργασμένο μάρμαρο και δημιουργήθηκε από την καταστροφή του ναού προκειμένου να ταιριάξει στις ανάγκες της εκκλησίας. Το δεύτερο στρώμα είναι πιο λεπτό, σχηματίζεται από λεπτή λατύπη και εντοπίζεται κάτω από τη θεμελίωση του ναού. Το τρίτο στρώμα είναι το πιο βαθύ και παρουσιάζει σκληρή σύσταση, ενώ εντοπίζονται και πολλά όστρακα που αποδίδονται στη μυκηναϊκή και πρωτοκυκλαδική περίοδο. Οι φάσεις του οικοδομήματος και τα στάδια μετατροπής του σε βασιλική εκκλησία τεκμηριώνονται από κατασκευές που εντοπίζονται στα θεμέλια του ναού και στα δάπεδα μεταξύ τους.

Για την πιστή αποκατάσταση του ναού ακολουθήθηκε μια διαδικασία ανεύρεσης μεμονωμένων αρχιτεκτονικών μελών για να ενταχθούν στην συνολική σύνθεσή του και διάλυσης του βυζαντινού ναού, ο οποίος χτίστηκε εκ νέου το 1977, είκοσι μέτρα πιο μακριά. Εντοπίστηκαν 36 αρχιτεκτονικά μέλη κατασκευασμένα από μάρμαρο (Στεφάνου, 2018).



Εικόνα 60: Το εσωτερικό του μουσείου Γύρουλα

4.1.2: Αρχιτεκτονικά στοιχεία του ναού

Ο ναός της Δήμητρας διαθέτει χαρακτηριστικά που τον διαφοροποιούν από τους υπόλοιπους ναούς της εποχής του. Πιο συγκεκριμένα, διαθέτει σχεδόν τετράγωνη κάτοψη (13,29 x 12,73 m), σε αντίθεση με τους περισσότερους ναούς που είναι επιμήκεις, και είναι προσανατολισμένος στον Νότο αντί για Ανατολή – Δύση. Επιπλέον, δεν ορθώνεται σε κρηπίδα, αλλά μέσω ευθυνητριάς, η οποία αποτελεί και τον στυλοβάτη των κίωνων.

Η πρόσοψη αποτελείται από πέντε κίονες, σε ιωνικές βάσεις αλλά χωρίς κυματοειδές σχήμα. Η διάμετρος των κίωνων αυξάνεται όσο αυξάνεται το ύψος, αντίθετα από τους συνήθεις κίονες που μειώνεται. Τα κιονόκρανα διαθέτουν στεφάνι φύλλων αλλά ζωγραφισμένο, όχι τρισδιάστατο και ο άβακας είναι διακοσμημένος με κορδέλα. Στο επιστύλιο εντοπίζονται τα μπροστινά δοκάρια της οροφής, η οποία εκτείνεται στον πρόναο. Οι τεγίδες της οροφής σχηματίζονται από επτά δοκούς μήκους τεσσάρων μέτρων, οι οποίες είναι λυγισμένες προς τα πάνω κατά δύο εκατοστά (2 cm) ώστε να αποδίδουν καμπυλότητα.

Η πρόσβαση στο σηκό γίνεται με δύο πόρτες που βρίσκονται στην 2^η και 5^η ενδιάμεση στήλη. Τα κουφώματα είναι διακοσμημένα με κορδέλες με χάντρες και τα κατώτατα όρια βαμμένα με κύμα. Ο σηκός αποτελείται από 2 μέρη από 5 στήλες, οι οποίες αποτελούν και τους κίονες της πρόσοψης. Οι εσωτερικοί κίονες έχουν ύψος από 5,40 έως 6,46 μέτρα και η διάμετρος της βάσης τους είναι σταθερή στα 50 cm, παραβιάζοντας την αρχαϊκή αρχιτεκτονική, γιατί είναι χωρίς κωνικότητα. Στέκονται σε ιωνική βάση χωρίς δακτύλιο και στηρίζουν την δίρριχτη οροφή μήκους πλευράς τεσσάρων μέτρων. Καθώς οι πλάκες έχουν πάχος 2 – 4 cm, το φως του ήλιου εισέρχεται στο εσωτερικό του ναού χωρίς πρόβλημα ακόμα και με κλειστές πόρτες. Τα τοιχώματα των σηκών στηρίζονται σε στυλοβάτη ύψους 28 εκατοστών ύψους και πλάτους 70 εκατοστών και αποτελούνται από δύο στρώματα. Το εξωτερικό αποτελείται από ορθογώνιες πλάκες και έχει εξωτερική κλίση 3%, αντίθετα από τις κοινές αρχές σχεδιασμού κατά τις οποίες αναμένεται εσωτερική κλίση, ενώ το εσωτερικό έχει ακανόνιστο σχήμα, από πέτρες διαφόρων σχημάτων. Ο εξωτερικός τοίχος είναι επιχρισμένος.

Η καμπυλότητα του ναού καθιστά την κατασκευή αρκετά εντυπωσιακή, καθώς αποτελείται από μαρμάρινα κομμάτια και η απόδοση καμπυλότητας μπορεί να γίνει μόνο τοποθετώντας μεμονωμένα στοιχεία σε κατάλληλη θέση. Επιπλέον, διευκολύνει πολύ την ανοικοδόμηση, καθώς η αρχική θέση μπορεί να προσδιοριστεί με σχετικά μεγάλη ακρίβεια.



Εικόνα 61: Μοντέλο του Ναού της Δήμητρας

4.1.3: Ο μύθος της θεάς Δήμητρας

Σύμφωνα με την ελληνική μυθολογία^[31], η θεά Δήμητρα είναι μια ανθρωπόμορφη θεότητα της καλλιέργειας και της ελεύθερης βλάστησης και θεωρείται προστάτιδα του γάμου και της μητρότητας των ανθρώπων. Το όνομά της (Δημήτηρ, Δαμάτηρ) σημαίνει *Μητέρα Γη*, από της λέξεις Μήτηρ (Μητέρα) και Δα (Γη). Είναι κόρη του Κρόνου και της Ρέας, αδελφή της Ήρας, της Εστίας, του Δία και του Ποσειδώνα και μητέρα της Περσεφόνης. Η αρπαγή της Περσεφόνης από τον Άδη (ή Πλούτωνα) στον Κάτω Κόσμο οδήγησε στον μαρασμό της Δήμητρας και όταν την ζήτησε πίσω συμφώνησαν να ανεβαίνει η Περσεφόνη έξι μήνες στον πάνω κόσμο, όποτε και επικρατούσε καλοκαιρία, και έξι μήνες να παραμένει στον κάτω κόσμο, όποτε και επικρατούσε κακοκαιρία. Με τον τρόπο αυτό εξηγούσαν τις αλλαγές στην εποχή και τις καλλιέργειες.

Θεωρείται ότι κάποτε έφτασε στην Ελευσίνα, όπου και ανέλαβε την ανατροφή του Δημοφώντα. Αρνήθηκε το θρόνο που της προσέφερε η μητέρα του Μετάνειρα και διασκεδάζε μόνο με τραγούδια της υπηρέτριας Ιάμβης. Έτσι, σύμφωνα με το μύθο, δημιουργήθηκε το ιαμβικό μέτρο.



Εικόνα 62: Η αρπαγή της Περσεφόνης από τον Πλούτωνα (Lorenzo Bernini, 1621)

Μαζί με την Περσεφόνη αποτελούν τα κεντρικά πρόσωπα στα Ελευσίνια μυστήρια και συχνά θεωρούνται το ίδιο πρόσωπο ή μια θεά με δύο πρόσωπα. Τα Μικρά Ελευσίνια διαδραματίζονταν το διάστημα από Φλεβάρη μέχρι Μάρτη (μήνας Ανθεστηρίωνας), δηλαδή όταν η Περσεφόνη βρισκόταν στον πάνω κόσμο και θεωρείται ότι ιδρύθηκαν από την Δήμητρα για εξαγνισμό του Ηρακλή από τους φόνους των Κενταύρων. Τα Μεγάλα Ελευσίνια γίνονταν τους μήνες Αύγουστο – Σεπτέμβρη, (μήνας Βοηδρομίωνας) όταν δηλαδή έπρεπε η Περσεφόνη να επιστρέψει στον κάτω κόσμο.

Τα Θεσμοφόρια αποτελούν την μεγαλύτερη γιορτή που γινόταν προς τιμήν της Δήμητρας στο διάστημα Σεπτέμβριος – Οκτώβριος, την εποχή της σποράς. Το διάστημα αυτό αποκαλούνταν «μήνας Πυανεψίωνας».

4.1.4: Ο μύθος του θεού Απόλλωνα

Ο Απόλλωνας^[32] (ή Φοίβος) ήταν θεός της μουσικής, του φωτός, προστάτης των τεχνών και της μαντείας. Κατά πλειοψηφία, θεωρείται πως γεννήθηκε στην σημερινή Δήλο από τη Λητώ, θεά της έναστρης νύχτας, μαζί με την αδερφή του Άρτεμις. Άλλες απόψεις υποστηρίζουν πως γεννήθηκε στην Κρήτη, ενώ σύγχρονοι ερευνητές αμφισβητούν την

ελληνική καταγωγή του και θεωρούν ότι προερχόταν από το Βορρά ή την Ασία. Ήταν ένας από τους 12 θεούς του Ολύμπου και ιδρυτής του Μαντείου των Δελφών. Σύμφωνα με την «Πολιτεία» του Πλάτωνα κοινοποιεί στους ανθρώπους όσα χρειάζεται να ξέρουν για τις θυσίες, τους ναούς και τη μεταθανάτια ύπαρξη. Στην Ιλιάδα εμφανίζεται να πολεμάει στο πλευρό των Τρώων μαζί με τους άλλους θεούς, εκτός από τον Άρη και την Αφροδίτη, καθώς και ως Κιθαρωδός στο τέλος της πρώτης ραψωδίας.

Η λατρεία του Απόλλωνα διαδόθηκε πολύ νωρίς στη Ρώμη, ξεκινώντας από την Κύμη και την νότια Ετρουρία. Το 433 π.Χ. κατασκευάστηκε ο πρώτος ναός αφιερωμένος στον θεό, με αφορμή μια επιδημία και με σκοπό να κατευναστεί η οργή του. Κοντά στο Πεδίο του Άρεως της Ρώμης υπάρχει το ιερό Απολλινάριον, αφιερωμένο στον Ρωμαίο Απόλλωνα, ο οποίος ήταν ιατρός και θεραπευτής (Apollo Medicus). Αποτελέσε πολύ σημαντικό παράγοντα στον εξελληνισμό της ρωμαϊκής θρησκείας.

Η σύγκριση του Απόλλωνα, φωτεινού, λογικού, συνετού θεού της αρμονίας και της τάξης, με τον συναισθηματικό και παράφορο θεό της μέθης και του παρορμητισμού Διόνυσο οδήγησε στη δημιουργία του απολλώνιου και διονυσιακού στοιχείου. Η ανοικοδόμηση του Ναού του Απόλλωνος, ο οποίος αναπαριστά και τα δύο στοιχεία, κάνει εμφανή την προσπάθεια γεφύρωσης των διαφορών των δύο θεοτήτων και παγίωσης μιας νέας εποχής για τη θρησκευτική λατρεία.



Εικόνα 63: Ο θεός Απόλλωνας, άγαλμα στην Ακαδημία Αθηνών

4.2: Ο αυτοκράτορας και ο δούλος

4.2.1: Ιστορικά και αρχιτεκτονικά στοιχεία

Το άγαλμα «Αυτοκράτορας και Δούλος» (Emperor and Barbarian) βρέθηκε στην αρχαία Αγορά της Αθήνας από την Αμερικανική Σχολή Κλασικών Σπουδών στην Αθήνα (American School of Classical Studies at Athens – ASCSA) και αποτελεί μέρος των ευρημάτων των ανασκαφών που διενεργεί η σχολή στην περιοχή από το 1931. Όλες οι πληροφορίες σχετικά με αυτό αντλούνται από τη βάση δεδομένων της σχολής^[33].



Εικόνα 64: Το άγαλμα «Αυτοκράτορας και Δούλος»

Αναλυτικότερα, το αντικείμενο βρέθηκε στις 23-24 Μαΐου του 1972 στην νότια στοά της Αρχαίας Αγοράς της Αθήνας. Χαρακτηρίζεται από μεγάλη λεπτομέρεια και είναι φτιαγμένο από λευκό μάρμαρο. Όπως φαίνεται και στην εικόνα, το άγαλμα είναι σπασμένο από τα γόνατα και πάνω και λείπει μεγάλο κομμάτι της βάσης του. Το χαμηλότερο κομμάτι του αγάλματος που είναι προσκολλημένο στη βάση αποτελείται από πόδια με κυνηγετικές μπότες, με το αριστερό να είναι λυγισμένο και να μην ακουμπάει στη βάση όλο το πόδι. Φαίνεται στη μέση του ποδιού το τέρμα κάποιου μανδύα που ερμηνεύεται πως πέφτει κατακόρυφα από τον αριστερό ώμο. Ένα κομμάτι του μανδύα έχει διασωθεί, αλλά είναι πολύ βαρύ για να ενωθεί με το υπόλοιπο άγαλμα. Το δεξί πόδι του αυτοκράτορα πατάει στο έδαφος με μια υποστήριξη για να στηρίζει και το υπόλοιπο σώμα, και φαίνεται και η φιγούρα

του δούλου σχεδόν ολόκληρη, ο οποίος στέκεται σκυμμένος σε υποτακτική στάση κοιτάζοντας τον αυτοκράτορα.



Εικόνα 65: Κομμάτι του μανδύα του αυτοκράτορα που έχει διασωθεί.

4.2.2: Ο αυτοκράτορας

Ο αυτοκράτορας θεωρείται πως είναι ο Τραϊανός^[34], Ρωμαίος Αυτοκράτορας από τον Ιανουάριο του 98 έως το 117 μ.Χ. Η παραδοσιακή ιστοριογραφία της συγκλήτου τον θεωρεί τον καλύτερο Ρωμαίο Princeps (optimus princeps) διότι, μεταξύ άλλων, θεμελίωσε το σύστημα της αυτοκρατορικής υιοθεσίας, το οποίο έδωσε λύση στο ζήτημα της διαδοχής. Κατά την διάρκεια της ηγεμονίας του επέφερε την μέγιστη επέκταση της αυτοκρατορίας του με την κατάκτηση της Αρμενίας, της Μεσοποταμίας και της Δακίας και στην εσωτερική του πολιτική προσπάθησε να ενισχύσει την Ιταλία με οικοδομικά και κοινωνικά έργα.



Εικόνα 66: Ο αυτοκράτορας Τραϊανός



Εικόνα 67: Ο αυτοκράτορας Αδριανός

Μια άλλη θεωρία αποδίδει την φιγούρα του αυτοκράτορα στον Αδριανό^[35], ο οποίος ήταν Ρωμαίος Αυτοκράτορας κατά τα έτη 117-138, ανιψιός και διάδοχος του Τραϊανού και στωικός και επικύρειος φιλόσοφος. Ήταν ο τελευταίος από τους «Πέντε Καλούς Αυτοκράτορες» και υπήρξε προστάτης των τεχνών και των γραμμάτων, αφού διέσωσε και κατασκεύασε σημαντικά μνημεία με χαρακτηριστικά παραδείγματα την Βίλλα του Αδριανού στο Τίβολι,

που αποτέλεσε αλεξανδρινό κήπο, και το Πάνθεον, το οποίο ανακατασκεύασε μετά από καταστροφή από πυρκαγιά και πήρε τη μορφή που έχει σήμερα, επηρεάζοντας πολλούς μεγάλους αρχιτέκτονες της Ιταλικής Αναγέννησης και του Μπαρόκ. Στην Αθήνα κατασκεύασε την Πύλη του Αδριανού και την Βιβλιοθήκη του Αδριανού, που σώζονται ακόμη σήμερα. Επιπλέον, φρόντισε την αποπεράτωση του Ναού του Ολυμπίου Διός με σκοπό να τον κάνει έδρα του Πανελληνίου. Πέρα από την αρχιτεκτονική του επιρροή, έγραψε ποίηση στα λατινικά και στα ελληνικά και συνέγραψε την αυτοβιογραφία του. Κοινωνικά, μείωσε τη δουλειά, εξανθρώπισε τον νομικό κώδικα και απαγόρευσε τα βασανιστήρια, κατανοώντας τις ανάγκες του λαού και κερδίζοντας τον τίτλο του «Πρώτου υπηρέτη της Αυτοκρατορίας» (*Friedrich Schiller*).



Εικόνα 68: Το Πάνθεον με τη μορφή που απέκτησε μετά την ανακατασκευή του από τον Αδριανό



Εικόνα 69: Η βίλλα του Αδριανού στο Τιβολι



Εικόνα 70: Η βιβλιοθήκη του Αδριανού στην Αθήνα

ΚΕΦΑΛΑΙΟ 5: Ανάλυση του παιχνιδιού

Η πειραματική μορφή του παιχνιδιού όπως αναλύεται στο 3^ο κεφάλαιο επεκτείνεται και προσαρμόζεται κατάλληλα, ώστε τα αντικείμενα να μην είναι ο πειραματικός κύβος αλλά το τρισδιάστατο μοντέλο του ναού της Δήμητρας στη Νάξο και το άγαλμα «Αυτοκράτορας και Δούλος» του μουσείου της Αρχαίας Αγοράς της Αθήνας. Στο παρόν κεφάλαιο αναλύονται ο σκοπός του παιχνιδιού, οι φωτογραμμετρικές επεξεργασίες των τρισδιάστατων μοντέλων και η διαδικασία δημιουργίας και ολοκλήρωσης του παιχνιδιού σε περιβάλλον εικονικής πραγματικότητας.

5.1: Σκοπός του παιχνιδιού

Το παιχνίδι που δημιουργείται στο πλαίσιο της παρούσας διπλωματικής εργασίας εντάσσεται στα παιχνίδια σοβαρού σκοπού (serious games) με αντικείμενο την πολιτιστική κληρονομιά σε περιβάλλον εικονικής πραγματικότητας. Αποτελεί ένα τρισδιάστατο παζλ, του οποίου τα κομμάτια ο χρήστης πρέπει να ενώσει με χρήση εξοπλισμού εικονικής πραγματικότητας ώστε να δώσει στα αντικείμενα την αρχική τους μορφή. Προκειμένου να μπορέσει ο παίκτης να σύρει τα κομμάτια και να τα τοποθετήσει στη σωστή θέση πρέπει να απαντήσει σε κάποιες απλές ερωτήσεις τύπου «Σωστό – Λάθος» σχετικές με το εκάστοτε αντικείμενο αποδεικνύοντας ότι έχει αποκτήσει στοιχειώδεις γνώσεις για αυτό. Το παιχνίδι δεν προϋποθέτει προ υπάρχουσες γνώσεις για τα αντικείμενα, καθώς οι ερωτήσεις βασίζονται σε πληροφοριακά κείμενα δύο παραγράφων που εμφανίζονται στον παίκτη στην αρχή του παιχνιδιού.

5.2: Προετοιμασία των αντικειμένων

Τα δύο αντικείμενα είναι απαραίτητο να υποστούν μια σειρά επεξεργασιών ώστε να είναι δυνατό να χρησιμοποιηθούν στο παιχνίδι. Οι επεξεργασίες αυτές αφορούν στο μέγεθος και στη μορφή του τρισδιάστατου μοντέλου και στην επίλυση προβλημάτων αυτού, όπως για παράδειγμα την έλλειψη πληροφορίας. Καθώς τα αντικείμενα δεν έχουν τις ίδιες ανάγκες επεξεργασίας, αναλύονται διαφορετικά.

5.2.1: Ο ναός της Δήμητρας

Διαδικασία λήψης δεδομένων

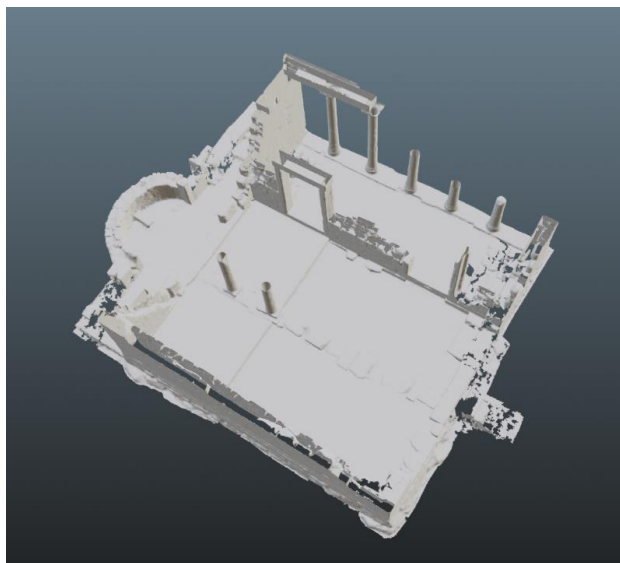
Το τρισδιάστατο μοντέλο του ναού της Δήμητρας σχηματίστηκε το 2018 στο πλαίσιο της μεταπτυχιακής εργασίας με θέμα «Η συμβολή των νέων τεχνολογιών στην αρχαιολογία: Η περίπτωση του ναού της Δήμητρας στο Σαγκρί Νάξου» (Στεφάνου, 2018). Χρησιμοποιώντας φωτογραμμετρικές μεθόδους υλοποιήθηκε ένα σύστημα συντεταγμένων σε κοινό σύστημα αναφοράς που καθορίστηκε από φωτοσταθερά σημεία (τρειςδιάστατα σημεία). Πιο αναλυτικά, φωτογραφήθηκε ο ναός περιμετρικά με συγκλίνοντες άξονες φωτογραφικής μηχανής με τέτοιο τρόπο ώστε οι φωτογραφίες να έχουν μεταξύ τους επικαλύψεις και τοποθετήθηκαν σημεία – στόχοι σε χαρακτηριστικά σημεία σε όλη την έκταση του μνημείου. Στη συνέχεια, ιδρύθηκε και επιλύθηκε ένα τετράπλευρο δίκτυο περιμετρικά του ναού, με χρήση γεωδαιτικού σταθμού.

Ταυτόχρονα, χρησιμοποιήθηκε ένας σαρωτής Laser, ο οποίος στέλνοντας και επιστρέφοντας παλμούς μέτρησε κατακόρυφες και οριζόντιες γωνίες και δημιούργησε ένα νέφος σημείων. Για το ναό σχηματίστηκαν έντεκα νέφη, τα οποία ενώθηκαν σε ένα ενιαίο με κοινό σύστημα αναφοράς.

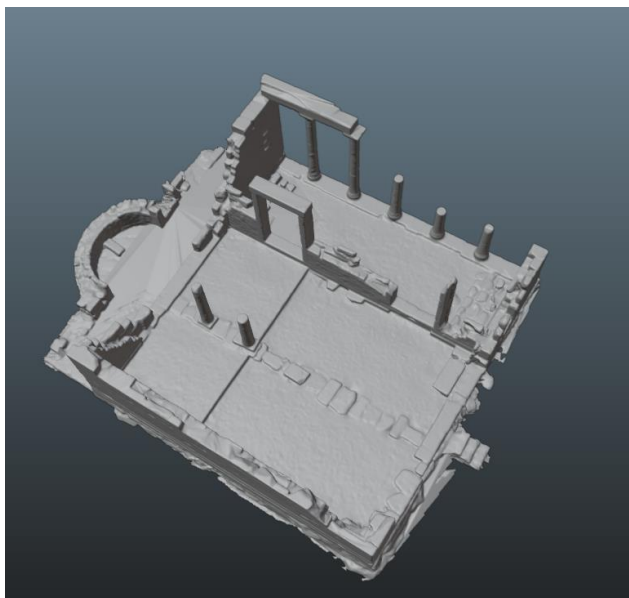
Το τρισδιάστατο μοντέλο υλοποιήθηκε στο λογισμικό Agisoft Photoscan με χρήση των αλγορίθμων Structure from Motion (SfM) και Multi View Stereo (MVS). Οι φωτογραφίες που είχαν ληφθεί προσανατολίστηκαν σχετικά και χρησιμοποιήθηκαν για να δημιουργηθεί ένα αραιό νέφος σημείων. Στη συνέχεια, χρησιμοποιώντας τα φωτοσταθερά σημεία (Ground Control Points) γεωαναφέρθηκε το νέφος σημείων και αποδόθηκε η κλίμακα 1:1.

Επεξεργασία του μοντέλου

Το μοντέλο του ναού της Δήμητρας, καθώς είναι αρκετά λεπτομερές και περίπλοκο, αποτελείται από πολλά τρίγωνα γεγονός που καθιστά το μέγεθος του αρχείου μεγάλο και μη διαχειρίσιμο. Προτεραιότητα, επομένως, αποτελεί η μείωση των τριγώνων που σχηματίζουν το αντικείμενο, χωρίς όμως να χαθεί πληροφορία. Για το σκοπό αυτό χρησιμοποιείται το πρόγραμμα “Instant Meshes”. Επόμενο βήμα αποτελεί η συμπλήρωση των κενών της επιφάνειας, που δημιουργήθηκαν λόγω έλλειψης πληροφορίας, έτσι ώστε το αντικείμενο να γίνει συμπαγές και να μπορεί να χρησιμοποιηθεί. Τα κενά καλύπτονται με ταυτόχρονη χρήση των προγραμμάτων “MeshLab” και “Meshmixer”, και με το τελευταίο το αντικείμενο έγινε τελικά συμπαγές.

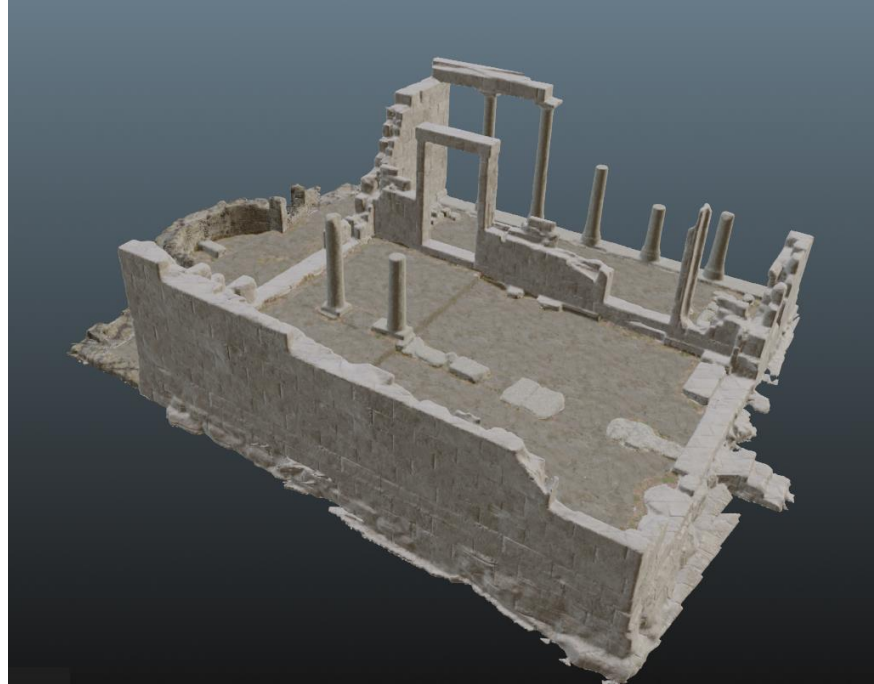


Εικόνα 71: Η αρχική μορφή του τρισδιάστατου μοντέλου του ναού της Δήμητρας



Εικόνα 72: Η μορφή του ναού της Δήμητρας μετά την επεξεργασία







Όπως φαίνεται από τις Εικόνα 71 και Εικόνα 72, μετά την επεξεργασία έχουν χαθεί κάποια στοιχεία λεπτομέρειας και ειδικότερα ο χρωματισμός της επιφάνειας του μοντέλου. Το πρόβλημα αυτό διορθώνεται στο επόμενο στάδιο, με την απόδοση υλικού στο μοντέλο ώστε να προσεγγίζει το πραγματικό.



Εικόνα 73: Το μοντέλο του ναού της Δήμητρας μετά την απόδοση υλικού.

Η απόδοση υλικού στο μοντέλο γίνεται με χρήση της μηχανής παιχνιδιών Blender και συγκεκριμένα της βιβλιοθήκης υλικών Blenderkit^[36]. Η διαδικασία γίνεται χειροκίνητα, «βάφοντας» κάθε περιοχή του μοντέλου με το επιθυμητό υλικό. Για το συγκεκριμένο αντικείμενο επιλέγονται τα υλικά “Stone Wall”, για τους εξωτερικούς τοίχους, “Old Stone Wall” για το ημικυκλικό εξωτερικό κομμάτι, “Concrete 07” για το έδαφος, “Marble Cracks” για τους κίονες, “Marble White” για τις πέτρες που βρίσκονται σκορπισμένες στο έδαφος και τέλος “Grass 03” για να γίνει το έδαφος πιο ρεαλιστικό.

Πίνακας 19: Υλικά που χρησιμοποιήθηκαν στον ναό της Δήμητρας

Υλικό	Όνομα
	Stone Wall
	Marble Cracks
	Concrete 07
	Marble White
	Old Stone Wall
	Grass 03

5.2.2: Το άγαλμα «Αυτοκράτορας και Δούλος»

Διαδικασία λήψης δεδομένων

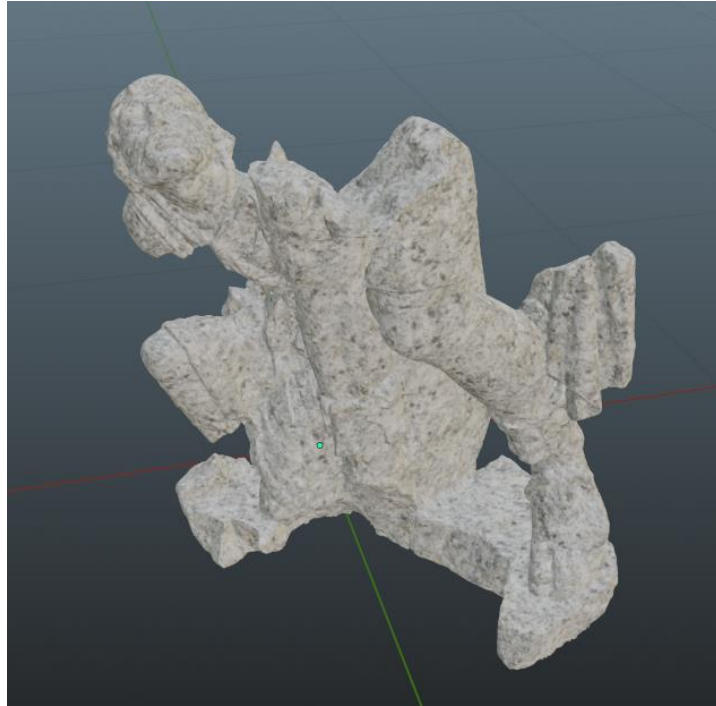
Το τρισδιάστατο μοντέλο του αγάλματος σχηματίστηκε στο πλαίσιο της διπλωματικής εργασίας με τίτλο «Τρισδιάστατη μοντελοποίηση γλυπτών με χρήση ψηφιακών εικόνων» (Ραβανη, 2018). Η συλλογή των δεδομένων πραγματοποιήθηκε εντός ενός εργαστηρίου της Αρχαίας Αγοράς.

Το αντικείμενο τοποθετήθηκε στο κέντρο της αίθουσας προκειμένου να φωτογραφηθεί περιμετρικά, χωρίς μεγέθυνση και χωρίς φλας. Για να επιτευχθεί η απόδοση της κλίμακας τοποθετήθηκε δίπλα στο άγαλμα ένας μεταλλικός χάρακας. Ελήφθησαν συνολικά 169 φωτογραφίες, οι οποίες εισήχθησαν στο πρόγραμμα Photoscan.

Αρχικά, οι εικόνες προσανατολίστηκαν σχετικά και σχημάτισαν ένα αραιό νέφος σημείων. Στη συνέχεια, αποδόθηκε στο μοντέλο με χρήση του χάρακα η επιθυμητή κλίμακα και μειώθηκε χειροκίνητα το νέφος των σημείων ώστε να περιέχει μόνο σημεία πάνω στο άγαλμα. Επόμενο βήμα αποτελεί η δημιουργία του πυκνού νέφους των σημείων, σε αυθαίρετο σύστημα αναφοράς. Τέλος, το νέφος επεξεράστηκε στο λογισμικό Geomagic ώστε να μην έχει κενά στην επιφάνειά του και αποδόθηκε σε αυτό υφή.

Επεξεργασία του μοντέλου

Το άγαλμα διατίθεται και αυτό σε μεγάλο μέγεθος αρχείου, επομένως με το πρόγραμμα “Instant Meshes” μειώνεται ο αριθμός των τριγώνων που σχηματίζουν την επιφάνεια και μετατρέπονται σε τετράγωνα για μεγαλύτερη απλότητα. Η επιφάνεια του μοντέλου δεν έχει κενά, επομένως αρκεί η απόδοση υλικού σε για να είναι το αντικείμενο έτοιμο να εισαχθεί στο παιχνίδι. Από την ίδια βιβλιοθήκη της μηχανής παιχνιδιών Blender επιλέγεται το υλικό “Marble White”, έτσι ώστε να ανταποκρίνεται πιο πολύ στην πραγματικότητα. Όπως και στην περίπτωση του ναού, έχουν χαθεί κάποια σημεία λεπτομέρειας αλλά δεν αλλοιώνεται η πληροφορία.



Εικόνα 74: Το άγαλμα «Αυτοκράτορας και Δούλος» μετά την επεξεργασία

5.3: Διαδικασία δημιουργίας του παιχνιδιού

5.3.1: Σχηματισμός των κομματιών

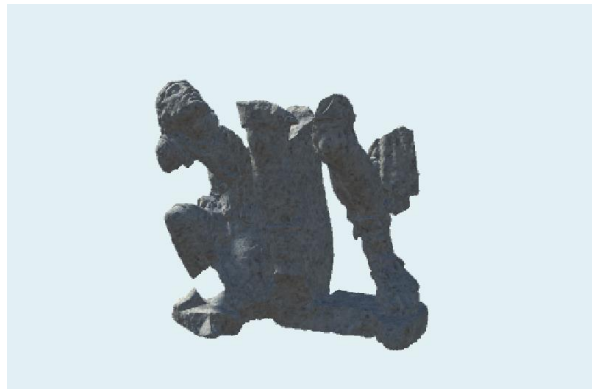
Τα δύο αντικείμενα κόβονται, στη συνέχεια, σε συγκεκριμένο αριθμό κομματιών ώστε να δημιουργηθεί το παζλ. Τα κομμάτια σχηματίζονται χειροκίνητα στην μηχανή παιχνιδιών Blender επιλέγοντας μέρη της επιφάνειας και αποκόβοντάς τα από την υπόλοιπη, ώστε να αποτελέσουν ένα ξεχωριστό αντικείμενο. Δεν επιλέγεται ο αυτόματος τρόπος δημιουργίας κομματιών που είχε εφαρμοστεί στον πειραματικό κύβο, δηλαδή η επέκταση Cell Fracture Tool της μηχανής παιχνιδιών Blender, καθώς τα αντικείμενα αποτελούνται από πολλά τρίγωνα και η τομή δεν γίνεται με τον επιθυμητό τρόπο. Το άγαλμα κόβεται σε πέντε συγκεκριμένα κομμάτια ενώ ο ναός στα διπλάσια, τόσο επειδή διαθέτει περισσότερα σημεία λεπτομέρειας, όσο και για να έχει ο χρήστης του παιχνιδιού την δυνατότητα να διαλέξει επίπεδο δυσκολίας.

Τα κομμάτια εξάγονται από την μηχανή παιχνιδιών Blender σε μορφή fbx (filmbox), καθώς έχει την μεγαλύτερη συμβατότητα μεταξύ εφαρμογών ψηφιακού περιεχομένου.

5.3.2: Εισαγωγή στη μηχανή παιχνιδιών unity

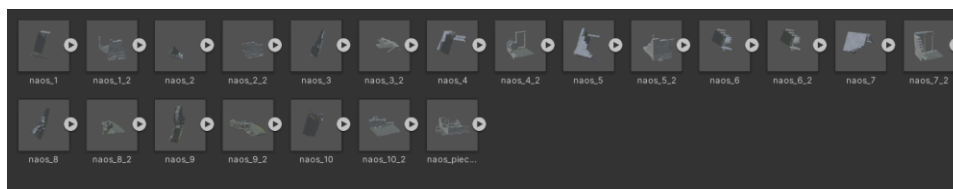
Σε αυτό το στάδιο δημιουργείται ένα νέο project στην μηχανή παιχνιδιών unity με δύο διαφορετικές σκηνές για τα δύο αντικείμενα και εισάγονται τα κομμάτια μαζί με τα στοιχεία που αποτελούν το υλικό τους.

Λόγω των διαφορετικών συστημάτων αναφοράς, χρειάζονται συγκεκριμένες προσαρμογές της κάμερας του παιχνιδιού προκειμένου να φαίνονται σωστά τα αντικείμενα. Πιο συγκεκριμένα, η κάμερα μετατοπίζεται στην επιφάνεια εργασίας της μηχανής παιχνιδιών ανάλογα με την θέση των αντικειμένων (GameObject → Align with View) και στρέφεται κατάλληλα. Αντίστοιχα, ρυθμίζεται και η πηγή φωτός ώστε οι σκιές από αυτήν να μην καλύπτουν σημαντικά σημεία λεπτομέρειας των αντικειμένων και το κενό αντικείμενο που χρησιμοποιείται ως στόχος εντός του κώδικα (target) για την πραγματοποίηση διαφόρων λειτουργιών τοποθετείται στο κέντρο των αντικειμένων.

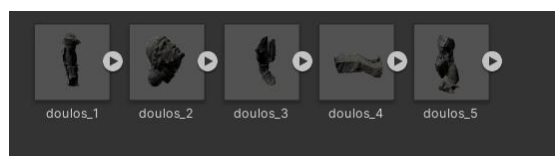


Εικόνα 75: Τα δύο αντικείμενα «αντοκράτορας και δούλος» (πάνω) και «Ναός της Δήμητρας» (κάτω) όπως εισάγονται στο περιβάλλον της unity

Τα κομμάτια τοποθετούνται υπό την ετικέτα «Pieces» όπως είχε γίνει και στο πειραματικό αντικείμενο, ώστε να μη χρειάζεται ο κώδικας να αλλάζει για κάθε ένα ξεχωριστά και τοποθετούνται στη θέση (0,0,0).



Εικόνα 76: Τα 10 κομμάτια του ναού όπως εισήχθησαν στο περιβάλλον της Unity



Εικόνα 77: Τα 5 κομμάτια του αγάλματος όπως εισήχθησαν στο περιβάλλον της Unity

5.3.3: Προσαρμογή του κώδικα C#

Ο κώδικας C# που έχει συνταχθεί για το πειραματικό αντικείμενο χρησιμοποιείται αυτούσιος και σε αυτό το στάδιο του παιχνιδιού, με κάποιες αναγκαίες προσαρμογές. Όπως έχει αναφερθεί και στο κεφάλαιο 3.4: Πειραματική εφαρμογή, η λογική πάνω στην οποία δημιουργείται ο κώδικας είναι η εξής:

1. Τα κομμάτια που βρίσκονται σε συγκεκριμένες θέσεις στο χώρο με την ετικέτα "Pieces" σκορπίζονται – εκτός από ένα - εντός ενός συγκεκριμένου πλαισίου σε τυχαίες θέσεις και προσανατολισμούς.
2. Με ανίχνευση της θέσης του ποντικιού και χρήση του μηχανισμού drag and drop τα κομμάτια τοποθετούνται στις σωστές θέσεις και στρέφονται κατάλληλα ώστε να έχουν σωστό προσανατολισμό.
3. Όταν η θέση και ο προσανατολισμός είναι κοντά στην αρχική, και άρα σωστή, τιμή δεν είναι πλέον δυνατή η στροφή και η μετατόπισή τους.
4. Το παιχνίδι ολοκληρώνεται όταν όλα τα κομμάτια είναι στη σωστή θέση.

Για ευκολία προστίθενται κάποιες επιπλέον λειτουργίες, όπως η μετακίνηση και περιστροφή της κάμερας του παιχνιδιού ώστε να μπορεί ο παίκτης να μετακινήσει τα κομμάτια και στους τρεις άξονες, κίνηση που δεν είναι εφικτή μόνο με το ποντίκι.

Οι προσαρμογές του κώδικα αναλύονται παρακάτω με τη σειρά με την οποία συναντώνται στο παιχνίδι.

Camera Rotator

Στην αρχή του παιχνιδιού πριν τυχαιοποιηθούν οι θέσεις των κομματιών η κάμερα πραγματοποιεί περιστροφή γύρω από το αντικείμενο ώστε ο χρήστης να αποκτήσει πλήρη εποπτεία αυτού. Η περιστροφή γίνεται με χρήση της εντολής "Rotate Around" και της συντομογραφίας "Vector3.down" για την αριστερή περιστροφή γύρω από τον άξονα Y (0, -1, 0). Η ταχύτητα περιστροφής μπορεί να καθοριστεί από τον χρήστη, αλλά έχει οριστεί για αρχή ίση με 20.

Πίνακας 20: Ο κώδικας για την περιστροφή της κάμερας

```
void Update()  
{  
    cam.transform.RotateAround(target.transform.position, Vector3.down,  
    Time.deltaTime * speed);  
}
```

Για λόγους ευκολίας, η περιστροφή πραγματοποιείται γύρω από ένα κενό αντικείμενο (target) που είναι τοποθετημένο στο κέντρο της σκηνής. Η περιστροφή τερματίζει όταν ο χρήστης «σπάει» το αντικείμενο σε κομμάτια με χρήση του πλήκτρου space.

Randomize pieces' position & rotation

Ο κώδικας για την τυχαιοποίηση των θέσεων των κομματιών ενεργοποιείται πατώντας το πλήκτρο space του πληκτρολογίου, οπότε τα κομμάτια διασκορπίζονται σε τυχαίες θέσεις στο χώρο εντός συγκεκριμένου εύρους τιμών. Το εύρος αυτό αλλάζει στα δύο αντικείμενα ώστε να εξασφαλίζεται ότι θα είναι πάντα ορατά από το χρήστη. Συγκεκριμένα, οι τιμές του των συντεταγμένων των κομματιών του ναού κυμαίνονται από 0.1 έως 10 ενώ του αγάλματος από 0.5 έως 3. Αντίστοιχα, τα κομμάτια αποκτούν τυχαίους προσανατολισμούς με στροφές κατά τυχαίες γωνίες γύρω από τον άξονα X εντός του εύρους τιμών από 0 έως 90 μοίρες, ώστε να μη δημιουργούνται πολύ μεγάλες διαφορές και να είναι πιο εύκολο να τοποθετηθούν τα κομμάτια στις σωστές θέσεις.

Όπως έχει αναφερθεί και στο κεφάλαιο 3.4: Πειραματική εφαρμογή, από την τυχαιοποίηση των θέσεων εξαιρείται ένα κομμάτι για κάθε αντικείμενο, ώστε να αποτελέσει σημείο αναφοράς. Οι τυχαίες θέσεις, καθώς και κάθε νέα θέση που θα προκύψει, αποθηκεύονται σε ένα πίνακα και οι προσανατολισμοί ερμηνεύονται με το μαθηματικό μοντέλο Quaternion.

Πίνακας 21: Ο κώδικας τυχαιοποίησης θέσεων και προσανατολισμών για το ναό

```

void randomizePiecesPositions()
{
    //converting array of gameobjects to list
    list = pieces.ToList();
    foreach (GameObject piece in list.Skip(1))
    {
        float xa = Random.Range(0.0f, 0.5f) > 0.5f ? -1 : 1; //if >0.5, result= -1,
alternatively result = 1
        float ya = Random.Range(0.0f, 0.5f) < 0.5f ? -1 : 1;
        float za = Random.Range(0.0f, 0.5f) > 0.5f ? -1 : 1;
        //Debug.Log("xa" + xa + ", ya:" + ya + ", za" + za);
        float x = Random.Range(0.1f, 10.0f) * xa;
        float y = Random.Range(0.1f, 10.0f) * ya;
        float z = Random.Range(0.1f, 10.0f) * za;

        //place pieces to random positions and rotations
        piece.transform.Rotate(Random.Range(0f, 90.0f), 0.0f, 0.0f);
        piece.transform.position = new Vector3(x, y, z);
    }
}

```

Πίνακας 22: Ο κώδικας τυχαιοποίησης θέσεων και προσανατολισμών για το άγαλμα

```

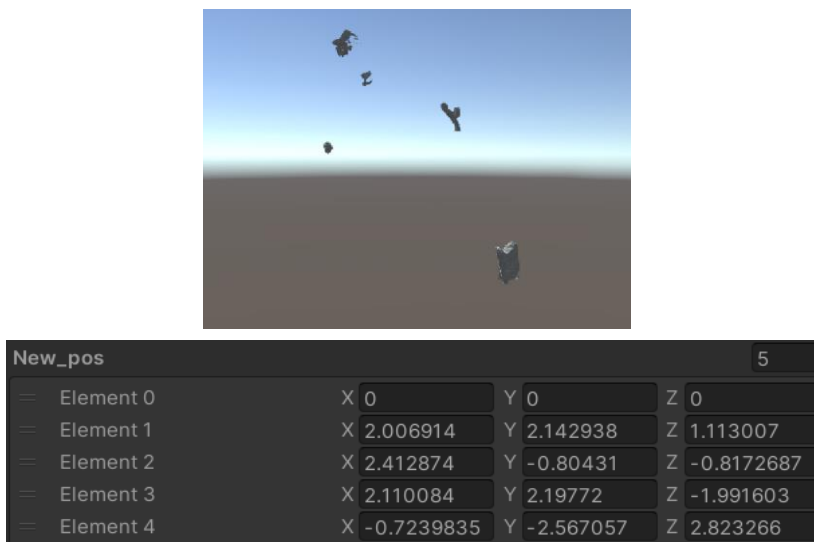
void randomizePiecesPositions()
{
    //converting array of gameobjects to list
    list = pieces.ToList();
    foreach (GameObject piece in list.Skip(1))
    {
        float xa = Random.Range(0.0f, 1.0f) > 0.5f ? -1 : 1; //if >0.5, result= -1,
alternatively result = 1
        float ya = Random.Range(0.0f, 1.0f) < 0.5f ? -1 : 1;
        float za = Random.Range(0.0f, 1.0f) > 0.5f ? -1 : 1;
        //Debug.Log("xa" + xa + ", ya:" + ya + ", za" + za);
        float x = Random.Range(0.5f, 3.0f) * xa;
        float y = Random.Range(0.5f, 3.0f) * ya;
        float z = Random.Range(0.5f, 3.0f) * za;

        //place pieces to random positions and rotations
        piece.transform.Rotate(Random.Range(0.0f, 90.0f), 0.0f, 0.0f);
        piece.transform.position = new Vector3(x, y, z);
    }
}

```




Εικόνα 78: Τυχαίες θέσεις των κομματιών του ναού όπως φαίνονται στο περιβάλλον της unity (πάνω) και οι συντεταγμένες τους (κάτω)



Εικόνα 79: Τυχαίες θέσεις των κομματιών του αγάλματος όπως φαίνονται στο περιβάλλον της unity (πάνω) και οι συντεταγμένες τους (κάτω)

Element 0	
X	8.146034e-08
Y	0
Z	0
W	1
Element 1	
X	0.5572668
Y	0
Z	0
W	0.8303335
Element 2	
X	0.3063918
Y	0
Z	0
W	0.9519055
Element 3	
X	0.5823629
Y	0
Z	0
W	0.8129289
Element 4	
X	0.409503
Y	0
Z	0
W	0.9123088

Εικόνα 80: Το μαθηματικό μοντέλο Quaternion για το άγαλμα

Drag and drop & περιστροφές

Η λογική της επιλογής των κομματιών και της τοποθέτησής τους στη σωστή θέση είναι ίδια με το πειραματικό αντικείμενο, δηλαδή γίνεται με τη χρήση ακτίνας ανίχνευσης. Σε αυτό το στάδιο η επιλογή και τοποθέτηση γίνονται με το αριστερό κλικ του ποντικιού (συμβολίζεται με το 0) ενώ οι περιστροφές των κομματιών με το δεξί (συμβολίζεται με το 1) και με μια συγκεκριμένη ταχύτητα περιστροφής, η οποία μπορεί να καθοριστεί και από τον χρήστη. Για το άγαλμα έχει προκαθοριστεί 100 ενώ για το ναό 5, καθώς τα κομμάτια έχουν διαφορετικό μέγεθος στα δύο αντικείμενα.

Η δυνατότητα μετακίνησης και περιστροφής των κομματιών κλειδώνει όταν τα κομμάτια είναι τοποθετημένα στη σωστή θέση. Για λόγους ευκολίας, δίνεται μια επιτρεπτή απόκλιση από αυτήν, η οποία είναι ίση με 0.05 μέτρα και 20 μοίρες για το ναό και 0.45 μέτρα και 20 μοίρες για το άγαλμα, για την μετατόπιση και την στροφή αντίστοιχα.

Πίνακας 23: Έλεγχος της θέσης των κομματιών για το ναό και αυτόματη τοποθέτησή τους στην σωστή

```
if (!checkIfAllPiecesAreInTheirDefaultPositions())
{
    // If the distance is less than 0.45 then set it to (0,0,0) and you cant
    move it anymore. Else move it
    if (distance < 0.05f)
    {
        getTarget.transform.position = defPostion;
    }
    else
    { //It will update target gameobject's current postion.
        getTarget.transform.position = currentPosition;
    }
}
```

Πίνακας 24: Έλεγχος της γωνίας στροφής κατά την περιστροφή των κομματιών

```
if (isMouseRotating)
{
    //store the default rotations of the gameobjects
    Quaternion quaternion_def = new Quaternion(0, 0, 0, 1);
    //Vector3 defRotation = new Vector3(180,180,180);
    //Debug.Log(defRotation);
    //enable rotation with mouse dragging
    float rotx = Input.GetAxis("Mouse X") * rotSpeed;
    //float roty = Input.GetAxis("Mouse Y") * rotSpeed;
    getTarget.transform.Rotate(-rotx, 0, 0);

    //check the distance from default position
    angle = Quaternion.Angle(getTarget.transform.localRotation, quaternion_def);
    //angle in degrees from local rotation to default rotation

    if (angle < 20f)
    {
        getTarget.transform.localRotation = quaternion_def;
        isMouseRotating = false;
    }
}
```

Επιπρόσθετες λειτουργίες

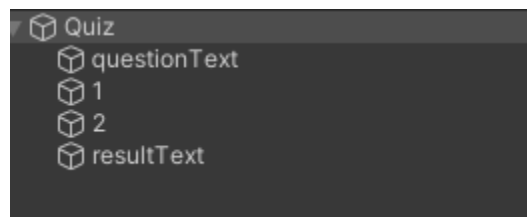
Όπως και στο πειραματικό αντικείμενο, παραμένει η μετακίνηση και στροφή της κάμερας με τα πλήκτρα WASD και QE αντίστοιχα για να μπορεί ο χρήστης να περιηγηθεί στον τρισδιάστατο χώρο και να τοποθετήσει σωστά τα κομμάτια. Ο κώδικας δεν έχει αλλάξει σε αυτό το στάδιο.

5.3.4: Δημιουργία των ερωτήσεων

Προκειμένου το παιχνίδι να αποκτήσει μορφή σοβαρού σκοπού με εκπαιδευτικό χαρακτήρα, εισάγονται ερωτήσεις γνώσεων σχετικές με το αντικείμενο τις οποίες ο χρήστης πρέπει να απαντήσει σωστά, ώστε να προχωρήσει στην τοποθέτηση κάθε κομματιού στη θέση του. Οι ερωτήσεις είναι μορφής «Σωστό – Λάθος» και τόσες ώστε να αντιστοιχούν σίγουρα μία για κάθε κομμάτι και κάποιες επιπλέον σε περίπτωση που ο χρήστης κάνει κάποιο λάθος.

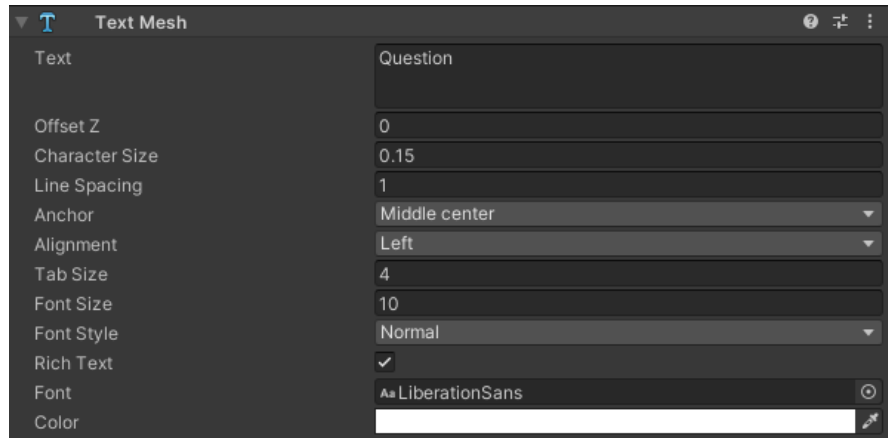
Δημιουργία των αντικειμένων κειμένου

Οι ερωτήσεις εισάγονται στο παιχνίδι ως αντικείμενα κειμένου και καλούνται όπως ορίζεται σε ξεχωριστό κώδικα. Συγκεκριμένα, δημιουργούνται 4 αντικείμενα για την τοποθέτηση της ερώτησης, των δύο επιλογών Σωστό ή Λάθος (1 & 2) και για το μήνυμα κειμένου που εμφανίζεται όταν ο χρήστης εισάγει την απάντησή του (result Text) και επισυνάπτονται σε ένα κενό αντικείμενο ώστε να μπορεί να διαγραφούν από την σκηνή και να εμφανίζονται σαν σύνολο μόνο όταν καλούνται.



Εικόνα 81: Τα αντικείμενα κειμένου

Κάθε ένα αντικείμενο ορίζεται ως κείμενο και παίρνει επιθυμητό μέγεθος και γραμματοσειρά.



Εικόνα 82: Τα χαρακτηριστικά που αποδίδονται σε κάθε αντικείμενο κειμένου

Κωδικοποίηση των ερωτήσεων και απαντήσεων

Οι ερωτήσεις και οι απαντήσεις σχηματίζονται σε κώδικα C# και επισυνάπτονται στο κατάλληλο αντικείμενο. Αναλυτικότερα, συντάσσεται κώδικας με τίτλο “Text Control” ο οποίος χρησιμοποιώντας λίστες καθορίζει τις ερωτήσεις και σε ποιο αντικείμενο (1 ή 2) βρίσκεται η σωστή απάντηση. Έχει οριστεί το αντικείμενο «1» να αντιστοιχεί στην απάντηση «Σωστό» και το αντικείμενο «2» στην απάντηση «Λάθος».

Πίνακας 25: Οι ερωτήσεις και απαντήσεις για το Ναό της Δήμητρας

```
//questions asked
public static List<string> questions_naos = new List<string>()
{
    "Ο ναός της Δήμητρας βρίσκεται στην Νάξο.", "Η Δήμητρα ήταν θεά του κυνηγιού.", "Ο ναός μετατράπηκε σε εκκλησία τα πρώτα χριστιανικά χρόνια", "Ο ναός υπάγεται στον δωρικό ρυθμό.", "Ο ναός της Δήμητρας θεωρείται πρόδρομος του Παρθενώνα", "Η θεά Δήμητρα είχε κόρη την Περσεφόνη", "Τα Ελευσίνια μυστήρια τελούνταν προς τιμήν της θεάς Δήμητρας", "Κατά τις ανασκαφές στην περιοχή δεν υπήρξαν άλλα ευρήματα", "Ο ναός χαρακτηρίζεται από καμπυλότητα", "Ο Απόλλωνας ήταν ο θεός της μουσικής.", "Ο ναός χρονολογείται στο 300 π.Χ.", "Σύμφωνα με το μύθο, την Περσεφόνη άρπαξε ο Άδης", "Ο ναός έχει αναστηλωθεί πλήρως", "Ο ναός ανακαλύφθηκε το 1949.", "Ο ναός είναι προσανατολισμένος στην διεύθυνση Ανατολή - Δύση.", "Στις ανασκαφές της περιοχής συμμετείχε το Πανεπιστήμιο του Μιλανο", "Ο Απόλλωνας θεωρείται πως ίδρυσε το Μαντείο των Δελφών."
};

//object that contains the correct answer

List<string> correctAnswer_naos = new List<string>()
{
    "1", "2", "1", "2", "1", "1", "1", "2", "1", "1", "2", "1", "1", "1", "2", "1", "1"
};
```

Πίνακας 26: Οι ερωτήσεις και απαντήσεις για το άγαλμα

```

//questions asked
public static List<string> questions = new List<string>()
{
    "Το άγαλμα βρέθηκε στην Αρχαία Αγορά της Αθήνας.", "Το κομμάτι που έχει απομείνει
απεικονίζει τον αυτοκράτορα.", "Το άγαλμα είναι φτιαγμένο από λευκό μάρμαρο.", "Εικάζεται πως ο
αυτοκράτορας είναι ο Τραϊανός.", "Το άγαλμα βρέθηκε την άνοιξη του 1972.", "Ο αυτοκράτορας
φορούσε μανδύα και κυνηγετικές μπότες.", "Έχει προσκολληθεί στο άγαλμα ό,τι βρέθηκε στις
ανασκαφές.", "Ο Ρωμαίος αυτοκράτορας Αδριανός δεν έκανε αρχιτεκτονικά έργα στην Αθήνα."
};

//object that has the correcre answer

List<string> correctAnswer = new List<string>()
{
    "1", "2", "1", "1", "1", "1", "2", "2"
};

```

Όπως φαίνεται παραπάνω στα δύο κομμάτια κώδικα που επισυνάπτονται, οι ερωτήσεις και οι απαντήσεις τους είναι:

Πίνακας 27: Οι ερωτήσεις και απαντήσεις για τα δύο αντικείμενα

ΕΡΩΤΗΣΗ	ΑΠΑΝΤΗΣΗ
Για τον Ναό της Δήμητρας:	
Ο Ναός της Δήμητρας βρίσκεται στην Νάξο.	Σωστό
Η Δήμητρα ήταν η θεά του κυνηγιού.	Λάθος
Ο Ναός μετατράπηκε σε εκκλησία τα πρώτα χριστιανικά χρόνια.	Σωστό
Ο Ναός υπάγεται στον δωρικό ρυθμό.	Λάθος
Ο Ναός της Δήμητρας θεωρείται πρόδρομος του Παρθενώνα.	Σωστό
Η θεά Δήμητρα είχε κόρη την Περσεφόνη.	Σωστό
Τα Ελευσίνια μυστήρια τελούνταν προς τιμήν της θεάς Δήμητρας.	Σωστό
Κατά τις ανασκαφές στην περιοχή δεν υπήρξαν άλλα ευρήματα.	Λάθος
Ο ναός χαρακτηρίζεται από καμπυλότητα.	Σωστό
Ο Απόλλωνας ήταν ο θεός της μουσικής.	Σωστό
Ο ναός χρονολογείται στο 300 π.Χ.	Λάθος
Σύμφωνα με το μύθο, την Περσεφόνη άρπαξε ο Άδης	Σωστό
Ο ναός έχει αναστηλωθεί πλήρως.	Σωστό
Ο ναός ανακαλύφθηκε το 1949.	Σωστό
Ο ναός είναι προσανατολισμένος στην διεύθυνση Ανατολή – Δύση.	Λάθος

Στις ανασκαφές της περιοχής συμμετείχε το Πανεπιστήμιο του Μιλάνο.	Σωστό
Ο Απόλλωνας θεωρείται πως ίδρυσε το Μαντείο των Δελφών.	Σωστό
Για το άγαλμα Αυτοκράτορας και Δούλος	
Το άγαλμα βρέθηκε στην Αρχαία Αγορά της Αθήνας	Σωστό
Το κομμάτι που έχει απομείνει απεικονίζει τον αυτοκράτορα	Λάθος
Το άγαλμα είναι φτιαγμένο από λευκό μάρμαρο	Σωστό
Εικάζεται πως ο αυτοκράτορας είναι ο Τραϊανός	Σωστό
Το άγαλμα βρέθηκε την άνοιξη του 1972.	Σωστό
Ο αυτοκράτορας φορούσε μανδύα και κυνηγετικές μπότες.	Σωστό
Έχει προσκολληθεί στο άγαλμα ό,τι βρέθηκε στις ανασκαφές.	Λάθος
Ο Ρωμαίος αυτοκράτορας Αδριανός δεν έκανε αρχιτεκτονικά έργα στην Αθήνα.	Λάθος

Σε κάθε αντικείμενο από τα «1» και «2» συντάσσεται κώδικας για να ανιχνεύει εάν το αντικείμενο έχει επιλεχθεί και τότε η απάντηση είναι σωστή καθώς επίσης και τι θα εμφανίζεται στην οθόνη, με χρήση λίστας.

Πίνακας 28: Οι δύο λίστες που καθορίζουν την πρώτη (σωστό) και την δεύτερη (λάθος) επιλογή σε κάθε ερώτηση

<pre>List<string> right = new List<string>() { "Σωστό", "Σωστό", "Σωστό", "Σωστό", "Σωστό", "Σωστό", "Σωστό", "Σωστό" };</pre>
<pre>List<string> wrong = new List<string>() { "Λάθος", "Λάθος", "Λάθος", "Λάθος", "Λάθος", "Λάθος", "Λάθος", "Λάθος" };</pre>

Μέχρι να γίνει η μετατροπή σε VR η επιλογή γίνεται με τα πλήκτρα 1 & 2 του πληκτρολογίου, 1 για το «Σωστό» και 2 για το «Λάθος».

Εμφάνιση των ερωτήσεων

Επόμενο βήμα αποτελεί η εμφάνιση των ερωτήσεων. Αναλυτικότερα, ορίζεται να εμφανίζεται κάθε ερώτηση όταν ο χρήστης πατάει πάνω στο κάθε κομμάτι. Κατά την εμφάνιση της ερώτησης απενεργοποιείται η δυνατότητα μετακίνησης των κομματιών και ενεργοποιείται ξανά όταν ο χρήστης δίνει τη σωστή απάντηση. Προκειμένου να εμφανίζεται ερώτηση μόνο την πρώτη φορά που επιλέγει ο χρήστης το κομμάτι και όχι κάθε φορά που πραγματοποιεί οποιαδήποτε ενέργεια πάνω σε αυτό, ο κώδικας που καλεί το αντικείμενο "Quiz" που περιέχει τις ερωτήσεις και απαντήσεις, θέτει μια Boolean μεταβλητή ("hasAppeared") από ψευδή σε αληθή όταν εμφανίζεται η ερώτηση, εμποδίζοντας έτσι την κλήση του αντικειμένου "Quiz" ξανά για το ίδιο κομμάτι και άρα την διατύπωση νέας ερώτησης. Η ανίχνευση της επιλογής του κομματιού λειτουργεί όπως και στο βασικό παιχνίδι, δηλαδή με ακτίνες (Raycast). Η ερώτηση εμφανίζεται στη θέση που βρίσκεται το εκάστοτε κομμάτι, ώστε να μην χρειάζεται ο παίκτης να ψάχνει στον χώρο προκειμένου να απαντήσει. Για λόγους ευκολίας, και εφόσον επιτρέπεται λόγω του μικρού πλήθους των κομματιών, η ανίχνευση της επιλογής γίνεται για εκάστοτε κομμάτι και όχι μαζικά.

Στο παρακάτω κομμάτι κώδικα φαίνεται ένα παράδειγμα εμφάνισης ερώτησης όταν ο παίκτης επιλέγει ένα κομμάτι από το άγαλμα (doulos_1) και ένα κομμάτι από το ναό (naos_1.002).

Πίνακας 29: Εμφάνιση της ερώτησης στο άγαλμα

```

if(Physics.Raycast(ray, out hit))
    {
        if(hit.collider.gameObject.name == "doulos_1" &&!hasAppeared1)
            {
                Instantiate(Quiz, hit.collider.gameObject.transform.position,
Quaternion.identity);
                hasAppeared1 = true;
            }
    }

```


Πίνακας 30: Εμφάνιση της ερώτησης στο ναό

```
if (Physics.Raycast(ray, out hit))
{
    if (hit.collider.gameObject.name == "naos_1.002" && !hasAppeared1)
    {
        Instantiate(Quiz_2, hit.collider.gameObject.transform.position,
Quaternion.identity);
        hasAppeared1 = true;
    }
}
```

Ο κώδικας που καλεί τις ερωτήσεις ενεργοποιείται, όπως προαναφέρθηκε, κατά την επιλογή ενός κομματιού, επομένως η εντολή ενεργοποίησης πραγματοποιείται στον αρχικό κώδικα (DragDropScript) κατά τον έλεγχο ανίχνευσης στόχου από ακτίνα. Η σειρά στον κώδικα που πραγματοποιεί την εμφάνιση είναι υπογραμμισμένη με κίτρινο χρώμα στον Πίνακας 31.

Πίνακας 31: Ενεργοποίηση του κώδικα που περιέχει την εμφάνιση των ερωτήσεων εντός του Drag and Drop

```
GameObject target = null;
Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
if (Physics.Raycast(ray.origin, ray.direction * 10, out hit))
{
    target = hit.collider.gameObject;
    c.GetComponent<QuestionAppearance>().enabled = true;
}
return target;
```

Όταν εμφανίζεται η ερώτηση, απενεργοποιείται ο κώδικας που επιτρέπει την μετακίνηση των κομματιών. Αυτό πραγματοποιείται εντός της συνάρτησης Awake του κώδικα που περιέχει τις ερωτήσεις και τις θέσεις των απαντήσεων ("Text Control"). Το κομμάτι του κώδικα που πραγματοποιεί την απενεργοποίηση είναι υπογραμμισμένο στον Πίνακας 32 με κίτρινο χρώμα.

Πίνακας 32: Η συνάρτηση Awake του κώδικα "Text Control"

```

void Awake()
{
    //set the position of random Question to -1 ==> no question as the positions start from 0
    randQuestion = -1;

    //fill the array of previous questions
    for(int i=0; i<9; i++)
    {
        previousQuestions.Add(i);
    }

    drag = GameObject.Find("Code"); //get access to the gameObject that contains the code
    drag.GetComponent<DragDropScript>().enabled = false; //object cannot be dragged
}

```

Όταν ο χρήστης απαντήσει σωστά την ερώτηση εμφανίζεται ένα μήνυμα «Μπράβο!», εκείνη εξαφανίζεται από την οθόνη μετά από 2 δευτερόλεπτα (Destroy) και ο χρήστης έχει ξανά την δυνατότητα μετακίνησης του κομματιού που έχει επιλέξει και τοποθέτησής του σε σωστή θέση και προσανατολισμό. Με επιλογή κάθε νέου κομματιού, εμφανίζεται νέα ερώτηση και η διαδικασία επαναλαμβάνεται μέχρι όλα τα κομμάτια να είναι στη σωστή θέση.

Πίνακας 33: Η διαδικασία που εκτελείται όταν ο χρήστης απαντάει σωστά σε μια ερώτηση.

```

if (correctAnswer_naos[randQuestion] == selectedAnswer)
{
    resultObj.GetComponent<TextMesh>().text = "Μπράβο!";
    Destroy(Quiz_2, 2); //destroy the gameObject after 2 seconds
    drag.GetComponent<DragDropScript_naos>().enabled = true; //object can be dragged
again
}

```

Εάν ο χρήστης δεν απαντήσει σωστά, εμφανίζεται το μήνυμα «Προσπάθησε ξανά!» και διατυπώνεται μια νέα ερώτηση μετά από δύο δευτερόλεπτα, που ορίζονται σε μια νέα συνάρτηση (MyDelay).

Πίνακας 34: Η διαδικασία που εκτελείται όταν ο χρήστης απαντάει λάθος σε μια ερώτηση

```
else //if the wrong answer is selected
{
    //show a new question
    resultObj.GetComponent<TextMesh>().text = "Προσπάθησε ξανά!";
    Invoke("myDelay", 2);
}
}
}

private void myDelay()
{
    if(correctAnswer[randQuestion] != selectedAnswer)
    {
        randQuestion = -1;
    }
}
```

Προκειμένου οι ερωτήσεις να μην εμφανίζονται με μία συγκεκριμένη σειρά χρησιμοποιείται γεννήτρια τυχαίων αριθμών που διατρέχει τους αύξοντες αριθμούς στους οποίους βρίσκονται οι ερωτήσεις και οι αντίστοιχες απαντήσεις μέσα στη λίστα. Επιπλέον, για να εξασφαλίζεται ότι δεν θα υπάρξουν επαναλήψεις των ερωτήσεων, δημιουργείται μία ακόμα λίστα που αποθηκεύει τους αύξοντες αριθμούς των ερωτήσεων που έχουν ερωτηθεί. Εάν επιλεγεί ένας αριθμός που υπάρχει ήδη στη λίστα, επαναλαμβάνεται η τυχαιοποίηση και επιλέγεται ένας νέος αριθμός. Η λίστα αυτή γεμίζει κάθε φορά που καλείται το αντικείμενο που περιέχει την ερώτηση, δηλαδή εντός της συνάρτησης Awake του κώδικα, όπως φαίνεται στον Πίνακας 35: Η διαδικασία επιλογής της ερώτησης που θα εμφανιστεί στον ναό

Ορίζεται ως αφετηρία ο αριθμός -1, καθώς οι θέσεις σε κάθε λίστα ξεκινάνε από τον αύξοντα αριθμό 0 κι έτσι είναι σίγουρο ότι το -1 δεν θα αντιστοιχεί σε κάποια ερώτηση.

Πίνακας 35: Η διαδικασία επιλογής της ερώτησης που θα εμφανιστεί στον ναό

```
if (randQuestion == -1)
{
    randQuestion = Random.Range(0, 9);

    for (int i = 0; i < 10; i++)
    {
        if (randQuestion == previousQuestions[i])
        {
            randQuestion = -1;
        }

        else
        {
            randQuestion = previousQuestions[questionNumber];
        }
    }
}
```

5.3.5: Προσθήκη ηχητικών εφέ

Προκειμένου να επιτευχθεί καλύτερη εμπειρία για τον παίκτη, προστίθενται στο παιχνίδι ηχητικά εφέ.

Εντοπίζονται και εισάγονται στην εφαρμογή ηχητικά εφέ από τη βιβλιοθήκη της Unity^[37] και από άλλες πλατφόρμες που τα διαθέτουν δωρεάν^[38]. Επιλέγονται ήχοι που να ταιριάζουν στην καταστροφή των αντικειμένων όταν σπάνε σε κομμάτια, στην ένωση των κομματιών και μια μελωδία για ήχος υποβάθρου όσο διαρκεί το παιχνίδι.

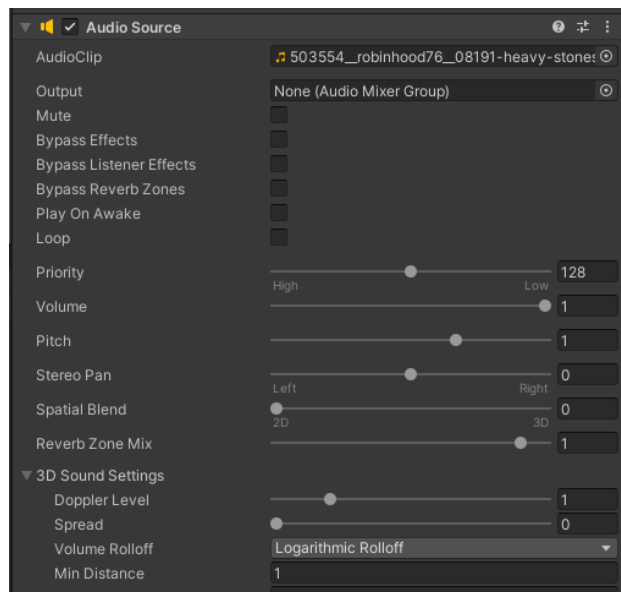
Για την εισαγωγή των ηχητικών εφέ χρησιμοποιούνται από τη βιβλιοθήκη της C# οι εντολές “AudioClip” και “AudioSource” που αναφέρονται στην πηγή του ήχου και το αρχείο .wav που χρησιμοποιείται. Το αρχείο καλείται εντός του κώδικα στην θέση που πρέπει κάθε φορά, δηλαδή ο ήχος της καταστροφής των αντικειμένων (breaking) προστίθεται όταν ο χρήστης σπάει το αντικείμενο σε κομμάτια ενώ ο ήχος της ενοποίησης όταν το κομμάτι μπαίνει στη σωστή του θέση (matching). Και στους δύο ήχους δίνεται συχνότητα 1. Οι παράμετροι στην Unity ρυθμίζονται αναλόγως ώστε οι ήχοι αυτοί να έχουν προτεραιότητα και να μην καλύπτονται από την μουσική του υποβάθρου.

Πίνακας 36: Κλήση του ήχου *breaking*

```
void Update()
{
    //used to randomize objects positions & rotations
    if (!isSpacePressed && Input.GetKeyDown(KeyCode.Space))
    {
        isSpacePressed = true;
        randomizePiecesPositions();
        c.GetComponent<CameraRotator_naos>().enabled = false;
        source.PlayOneShot(breaking, 1F);
    }
}
```

Πίνακας 37: Ορισμός του ήχου *matching*

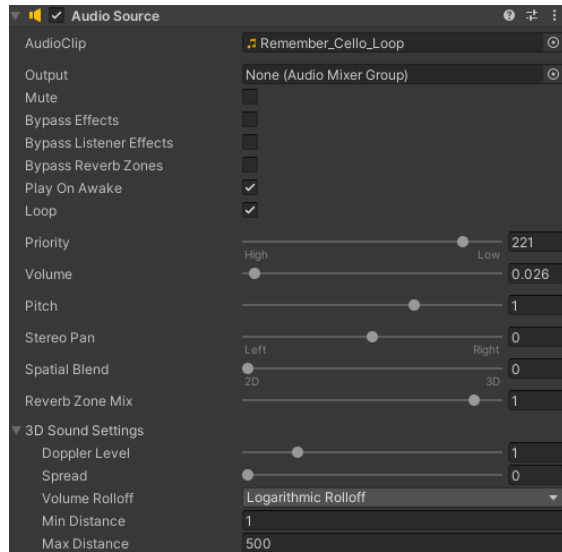
```
if (!checkIfAllPiecesAreInTheirDefaultPositions())
{
    // If the distance is less than 0.45 then set it to (0,0,0) and you cant
    move it anymore. Else move it
    if (distance < 0.05f)
    {
        getTarget.transform.position = defPostion;
        source_2.PlayOneShot(matching, 1F);
    }
}
```



Εικόνα 83: Οι παράμετροι του ηχητικού εφέ για τη δημιουργία των κομματιών

Όπως φαίνεται στην εικόνα 11, έχει ρυθμιστεί έτσι ώστε το αρχείο να μην καλείται όταν ξεκινάει το παιχνίδι (Play On Awake) αλλά όποτε ορίζει ο κώδικας και να μην παίζεται συνεχώς (Loop). Η προτεραιότητα του ήχου είναι στη μέση και η ένταση υψηλή.

Ως μουσική υποβάθρου επιλέγεται ένα κομμάτι κλασικής μουσικής από την βιβλιοθήκη της Unity σε χαμηλή ένταση και προτεραιότητα.



Εικόνα 84: Οι παράμετροι της μουσικής υποβάθρου

5.3.6: Μετατροπή σε VR

Λογισμικά και επεκτάσεις που χρησιμοποιήθηκαν

Ακολουθώντας τις εκπαιδευτικές οδηγίες της μηχανής παιχνιδιών Unity^[39] και ειδικές οδηγίες για την εισαγωγή του εξοπλισμού HTC Vive στην μηχανή παιχνιδιών Unity^[40] το παιχνίδι από απλή τρισδιάστατη εφαρμογή μετατρέπεται σε εφαρμογή περιβάλλοντος Εικονικής Πραγματικότητας. Για το σκοπό αυτό χρησιμοποιήθηκε ο εγκατεστημένος εξοπλισμός του εργαστηρίου Φωτογραμμετρίας της Σχολής Αγρονόμων και Τοπογράφων Μηχανικών – Μηχανικών Γεωπληροφορικής του Εθνικού Μετσόβιου Πολυτεχνείου.

Προκειμένου να αναγνωριστεί ο εξοπλισμός τόσο από τον υπολογιστή όσο και από την μηχανή παιχνιδιών είναι απαραίτητη η εγκατάσταση δύο λογισμικών:

- Vive Port: παρέχεται από την εταιρεία HTC και χρειάζεται για την αναγνώριση του χώρου που στήνεται ο εξοπλισμός εικονικής πραγματικότητας
- SteamVR: είναι απαραίτητο για την εισαγωγή του headset HTC Vive στην Unity.

Επιπλέον, εγκαθίστανται στο περιβάλλον της Unity τα εξής πρόσθετα:

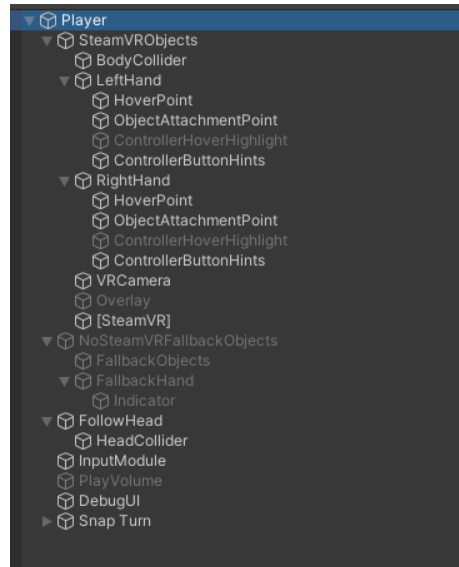
- XR plugin management
- XR Interaction Toolkit
- Open XR plugin
- Universal Rp
- Steam VR

ώστε να αναγνωρίζεται ο εξοπλισμός και να γίνει η μετάβαση εύκολα.



Εικόνα 85: Το πρόσθετο steamvr

Το πρόσθετο Steam VR εισάγει στην Unity συγκεκριμένα αντικείμενα και κώδικες που με κατάλληλη προσαρμογή μπορεί να χρησιμοποιηθούν για την δημιουργία εφαρμογής Εικονικής Πραγματικότητας με οποιονδήποτε εξοπλισμό. Το σημαντικότερο είναι το αντικείμενο “Player”, το οποίο αντικαθιστά την απλή κάμερα του δισδιάστατου ή τρισδιάστατου παιχνιδιού με αντικείμενα που ελέγχουν την θέση των χεριών και του κεφαλιού του παίκτη αλλά και καθορίζει τον χώρο στον οποίο θα διαδραματίζεται το παιχνίδι. Επιπλέον, παρέχει σημεία και περιοχές τηλεμεταφοράς του παίκτη μέσα στο χώρο αλλά και κώδικες που καθιστούν δυνατή την αλληλεπίδραση του χρήστη με αντικείμενα.

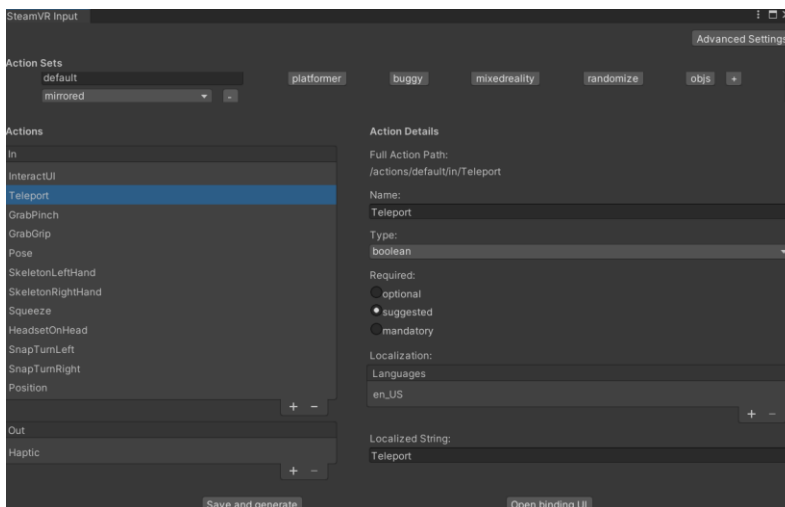


Εικόνα 86: Το αντικείμενο που ορίζει τον παίκτη σε ένα VR παιχνίδι

Οι ενέργειες που πραγματοποιούνται με κάθε κίνηση καθορίζονται από το παράθυρο SteamVR Input, το οποίο έχει ήδη κάποιες προκαθορισμένες τιμές και μπορεί να προστεθούν νέες ανάλογα τις ανάγκες της κάθε εφαρμογής. Οι ενέργειες μπορεί να είναι:

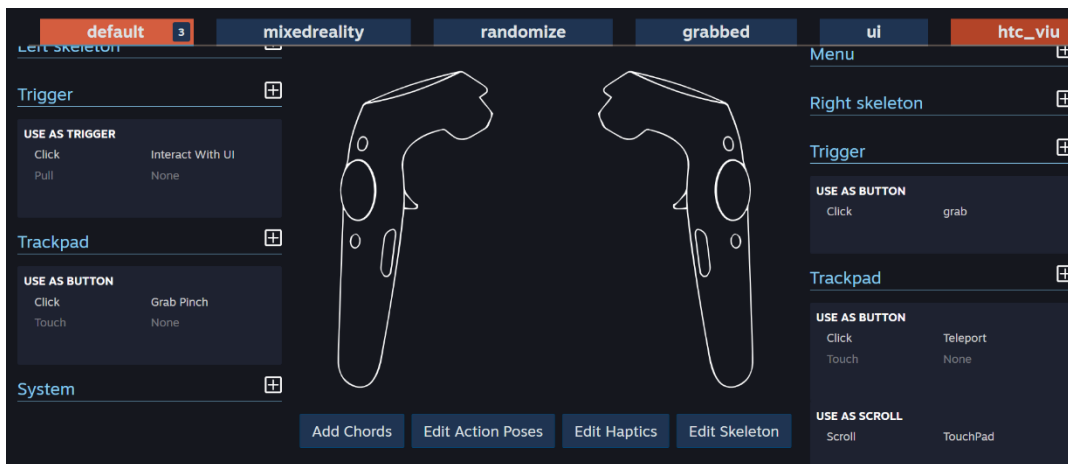
- **Boolean**, δηλαδή να παίρνει τιμές σωστή ή λάθος, όπως για παράδειγμα το κράτημα ενός αντικειμένου από τον παίκτη – είτε το κρατάει είτε όχι
- **Vector** 1, 2 ή 3, ανάλογα με την κίνηση στον έναν άξονα, σε δύο ή και στους 3. Το Vector 1 συνήθως χρησιμοποιείται για καθορισμό ταχύτητας, το 2 για την κίνηση σε δύο διαστάσεις X και Y ενώ το 3 χρησιμοποιείται σπάνια
- **Pose**, η οποία καθορίζει την θέση και περιστροφή σε έναν τρισδιάστατο χώρο. Χρησιμοποιείται συνήθως στον εντοπισμό των χειριστηρίων
- **Skeleton**, η οποία ανιχνεύει τις κινήσεις των δακτύλων του χρήστη όσο κρατάει το κάθε χειριστήριο
- **Vibration**, η οποία προκαλεί μια δόνηση των χειριστηρίων.

«Ανάπτυξη εκπαιδευτικού 3D Puzzle με αντικείμενα πολιτιστικής κληρονομιάς σε περιβάλλον Εικονικής Πραγματικότητας»



Εικόνα 87: Steam vr input

Η ενέργεια που πραγματοποιείται με κάθε κουμπί του χειριστηρίου καθορίζεται από το Binding UI του παραθύρου SteamVR Input.

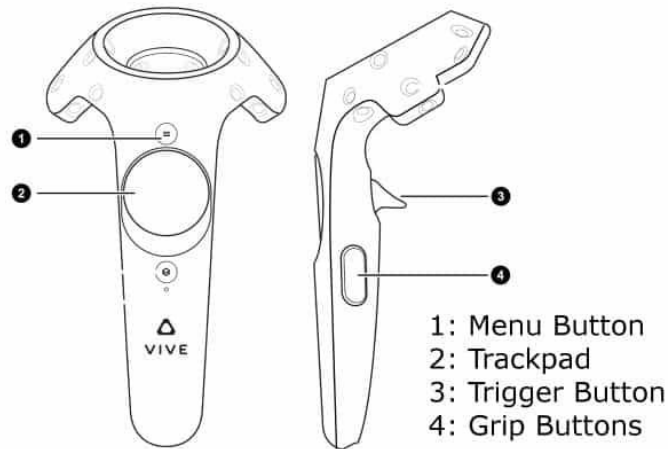


Εικόνα 88: Binding Ui

Κάθε χειριστήριο HTC Vive έχει τέσσερα κουμπιά:

1. Κουμπί menu
2. Κουμπί αφής, που λειτουργεί και με ανίχνευση κίνησης από τον αντίχειρα και με πάτημα (Trackpad)
3. Σκανδάλη (Trigger)
4. Λαβή (Grip)

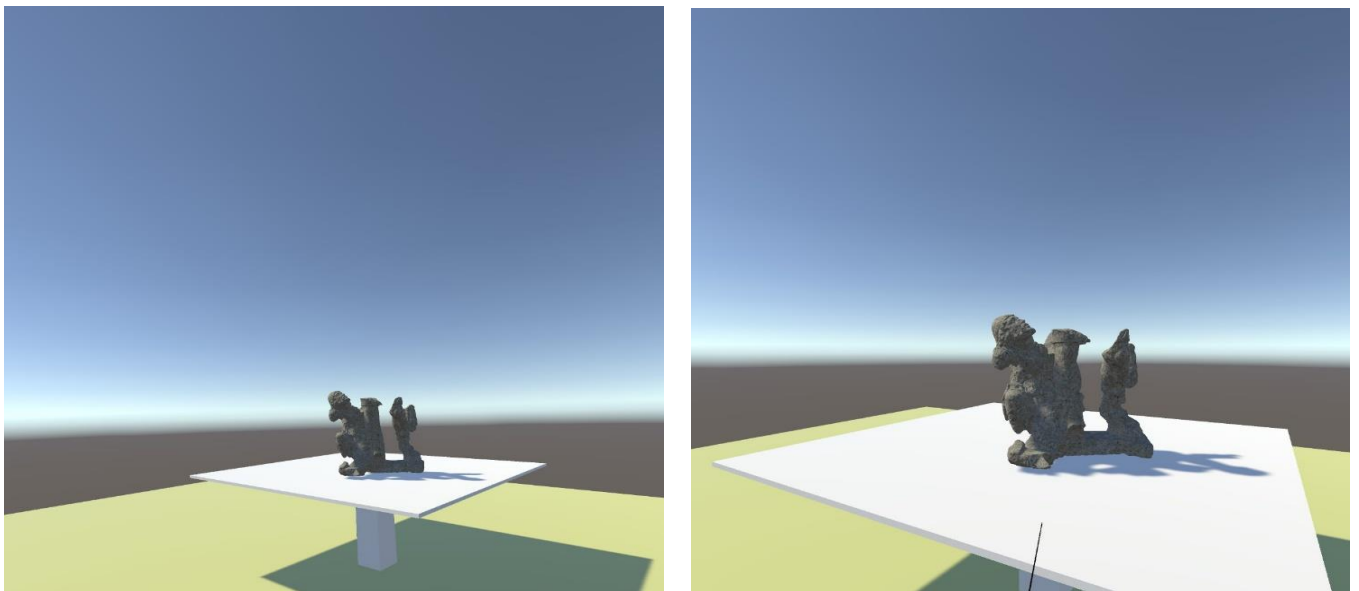
Στην πορεία του κεφαλαίου αναλύεται για ποιο σκοπό χρησιμοποιείται κάθε ένα από αυτά.



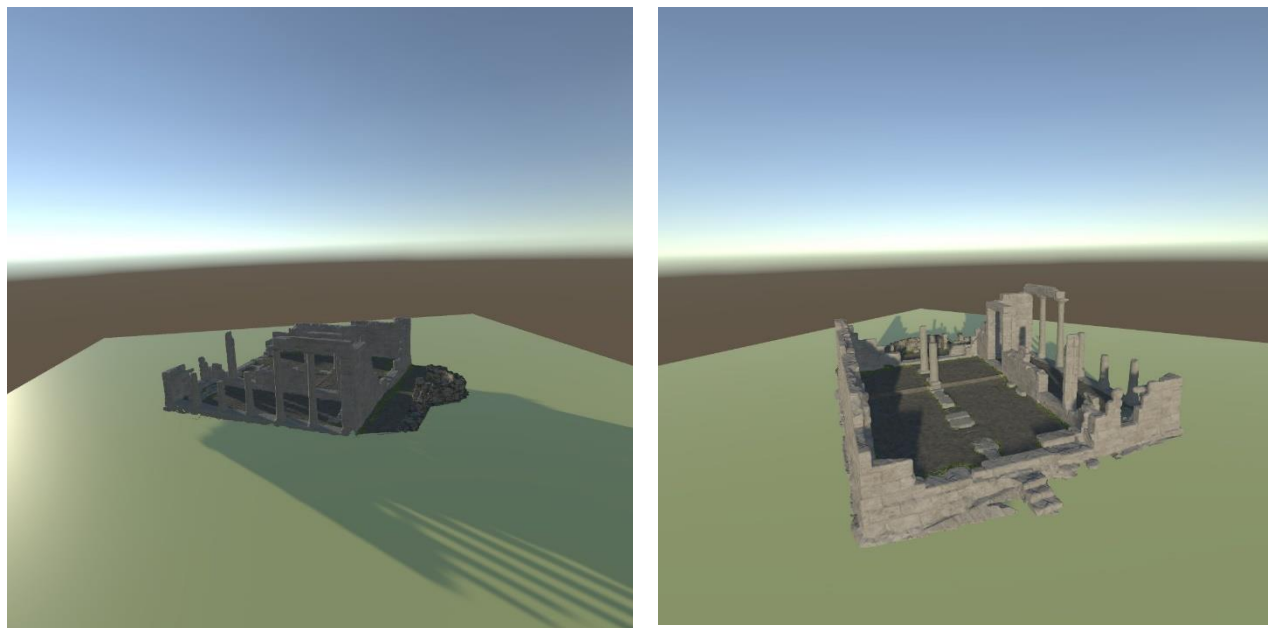
Εικόνα 89: Διαθέσιμα πλήκτρα των χειριστηρίων

Μετατροπή του παιχνιδιού

Αρχικά, δημιουργείται το περιβάλλον στο οποίο θα στηθεί το παιχνίδι. Τα δύο αντικείμενα, καθώς έχουν διαφορετικό μέγεθος, χρειάζονται και διαφορετικό περιβάλλον για να είναι εύκολη η αλληλεπίδραση του χρήστη με αυτά. Έτσι, το άγαλμα τοποθετείται πάνω σε ένα τραπέζι ενώ ο ναός παραμένει στο έδαφος, σε μεσαίου μεγέθους κομμάτια τα οποία είναι ρεαλιστικό να μετακινεί ο παίκτης με τα χέρια.

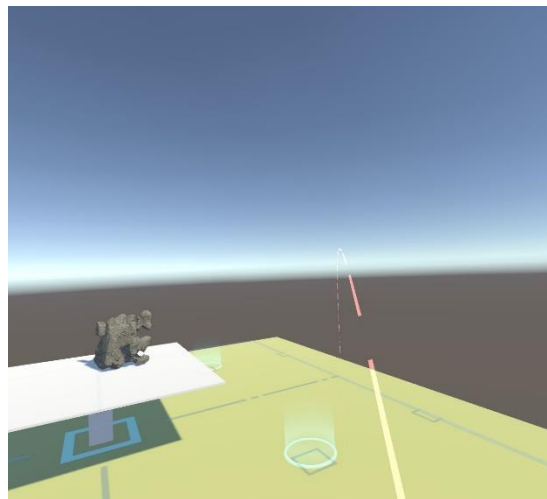
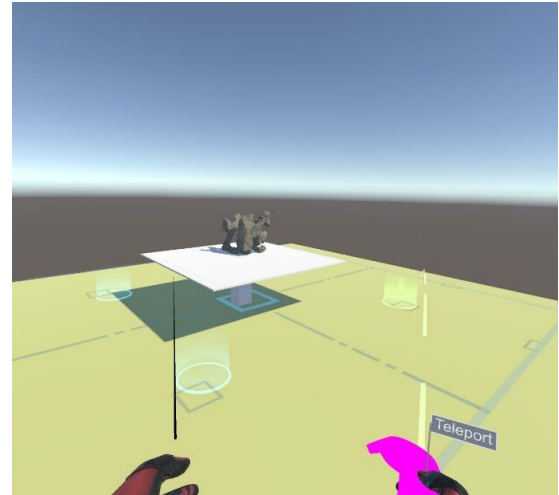
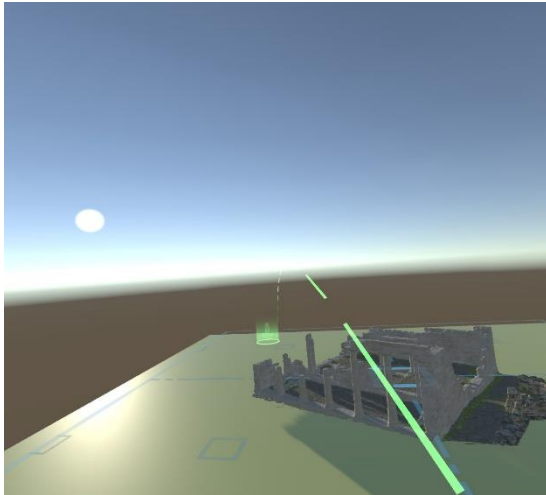


Εικόνα 90: Το άγαλμα σε περιβάλλον Εικονικής Πραγματικότητας



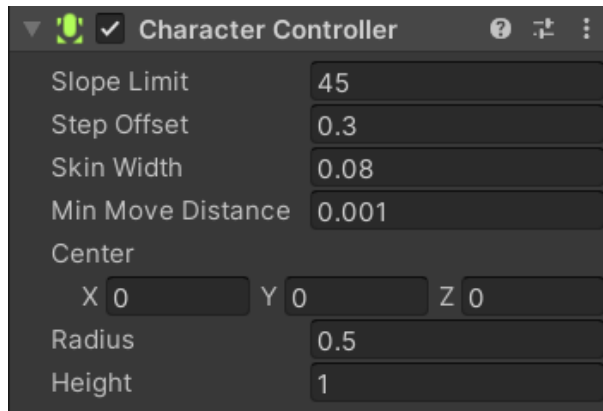
Εικόνα 91: Ο ναός σε περιβάλλον Εικονικής Πραγματικότητας

Επόμενο στάδιο αποτελεί ο καθορισμός του τρόπου με τον οποίο θα μετακινείται ο παίκτης μέσα στο χώρο. Προκειμένου να αποφευχθεί ο κίνδυνος ζαλάδας που προκύπτει συχνά σε περιβάλλοντα Εικονικής Πραγματικότητας, επιλέγεται ο παίκτης να μετακινείται με τηλεμεταφορά και όχι περπατώντας στο χώρο. Αυτό καθίσταται δυνατό εξαιτίας των δυνατοτήτων του προσθέτου Steam VR, το οποίο παρέχει περιοχές και σημεία τηλεμεταφοράς (Teleport Points & Teleport Area). Οι περιοχές τοποθετούνται στο κέντρο της εκάστοτε σκηνής ώστε να μπορεί ο χρήστης να μεταφερθεί οπουδήποτε, ενώ τα σημεία περιμετρικά των αντικειμένων, και συγκεκριμένα στις τέσσερις πλευρές του τραπέζιου στο άγαλμα και διαγωνίως στην περιοχή τηλεμεταφοράς στον ναό. Η τηλεμεταφορά πραγματοποιείται πατώντας το δεξί πλήκτρο αφής (TrackPad) όπως καθορίζεται στις ενέργειες του δεξιού χεριού, όπου και βγαίνει από το εικονικό χειριστήριο ένα τόξο πράσινου χρώματος όταν αγγίζει περιοχές που η μεταφορά είναι επιτρεπτή και κόκκινου σε κάθε αντίθετη περίπτωση.



Εικόνα 92: Παράδειγμα της τηλεμεταφοράς του παίκτη σε επιτρεπτά σημεία (πάνω) και μη επιτρεπτά (κάτω)

Για επιπλέον μετακίνηση, ο παίκτης μπορεί να σύρει τον αντίχειρα του δεξιού χεριού στο αντίστοιχο πλήκτρο αφής και να μετακινηθεί μπροστά ή πίσω ανάλογα με την θέση του κεφαλιού. Προκειμένου να επιτευχθεί αυτό εισάγεται στο αντικείμενο που αντικαθιστά τον παίκτη (Player) το εξάρτημα "Character Controller", το οποίο καθορίζει έναν χαρακτήρα συγκεκριμένου ύψους εντός της σκηνής και συντάσσεται κώδικας που ανιχνεύει την τρισδιάστατη κίνηση αυτού του χαρακτήρα.



Εικόνα 93: Εξάρτημα που χρησιμοποιείται για την κίνηση του παίκτη στη σκηνή

Η λογική του παιχνιδιού από εδώ και πέρα παραμένει ίδια και στο περιβάλλον Εικονικής Πραγματικότητας, αλλάζουν όμως οι μέθοδοι που επιτυγχάνεται αυτό. Αναλυτικότερα, οι ενέργειες που πραγματοποιούνται μέσα στο παιχνίδι είναι:

- Τυχαιοποίηση των θέσεων των κομματιών
- Επιλογή κομματιού προς μετακίνηση και μετακίνηση αυτού
- Εμφάνιση ερώτησης και επιλογή σωστής απάντησης

Η τυχαιοποίηση γίνεται με τον ίδιο τρόπο με την απλή τρισδιάστατη εφαρμογή, απλώς αντί για το πλήκτρο space του πληκτρολογίου χρησιμοποιείται το αριστερό TrackPad. Οι θέσεις των κομματιών (εκτός από 1 που χρησιμοποιείται κάθε φορά για σημείο αναφοράς) τυχαιοποιούνται μόνο κατά τους άξονες x και z ώστε να παραμείνουν τα κομμάτια πάνω στην επιφάνεια που θα φτιαχτεί το puzzle ενώ οι ελάχιστες και μέγιστες τιμές που μπορεί να πάρουν οι συντεταγμένες καθορίζονται από το μέγεθος της κάθε επιφάνειας. Όπως φαίνεται και στα παρακάτω κομμάτια κώδικα, το άγαλμα παίρνει τιμές από -0.5 έως 0.8 στον x και -0.5 έως 1 στον z προκειμένου να μην βγουν τα κομμάτια εκτός του τραπεζιού, το οποίο είναι τετράγωνο πλευράς 1, ενώ ο ναός από -1.2159 έως -0.2159 στον x και 0.37 έως 1.37 στον z για να μην βγουν τα κομμάτια εκτός της σκηνής.

Πίνακας 38: Τυχαιοποίηση των θέσεων των κομματιών του αγάλματος

```

void RandomizePiecesPosition()
{

    foreach (GameObject piece in list.Skip(1))
    {

        float x = Random.Range(-0.5f, 0.8f);
        float z = Random.Range(-0.5f, 1f);

        piece.transform.position = new Vector3(x, 0, z); //we dont want it to move
        vertically, we want it to stay on the table

    }

}

```

Πίνακας 39: Τυχαιοποίηση των θέσεων των κομματιών του ναού

```

void RandomizePiecesPosition()
{

    foreach (GameObject piece in list.Skip(1))
    {

        float x = Random.Range(-1.2159f, -0.2159f);
        float z = Random.Range(0.37f, 1.37f);

        piece.transform.position = new Vector3(x, 0, z); //we dont want it to move
        vertically, we want it to stay on the ground

    }

}

```

Η εισαγωγή του πλήκτρου με το οποίο πραγματοποιείται η τυχαιοποίηση στον κώδικα γίνεται χρησιμοποιώντας την βιβλιοθήκη Valve VR, καθορίζοντας μια μεταβλητή τύπου Boolean (μπορεί να πάρει τιμές αληθές ή ψευδές).

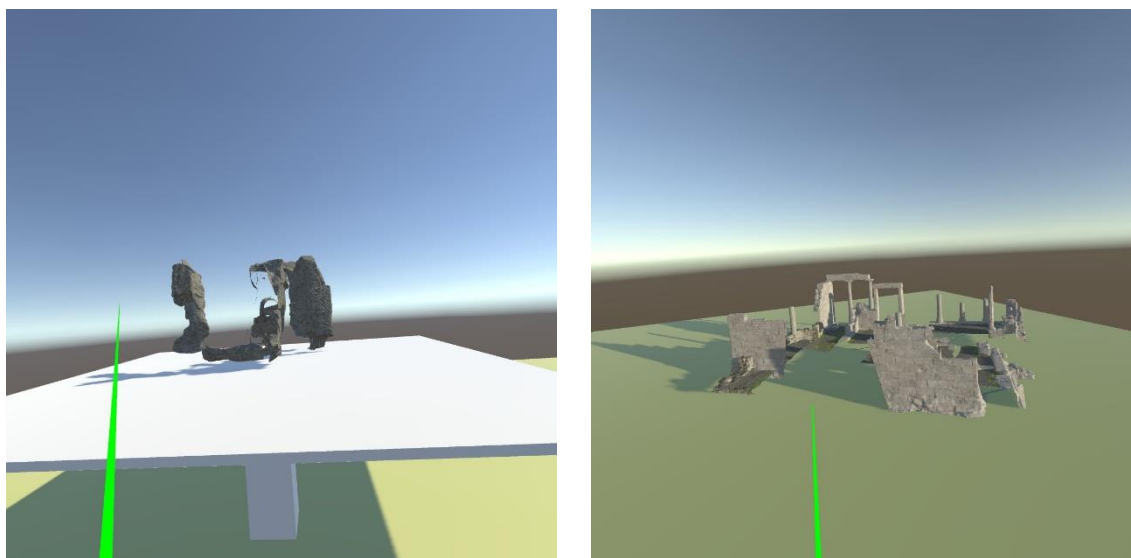
Πίνακας 40: Καθορισμός του πλήκτρου που πραγματοποιεί την συγκεκριμένη ενέργεια

```
public SteamVR_Action_Boolean randomAction;
```

Ο παραπάνω τρόπος καθορισμού του πλήκτρου εμφανίζει στο περιβάλλον της unity μια λίστα με όλες τις ενέργειες που παρέχει το πρόσθετο Steam VR και επιλέγεται η επιθυμητή, δηλαδή εκείνη που έχει οριστεί στο αντίστοιχο πλήκτρο εντός του Steam VR Input.



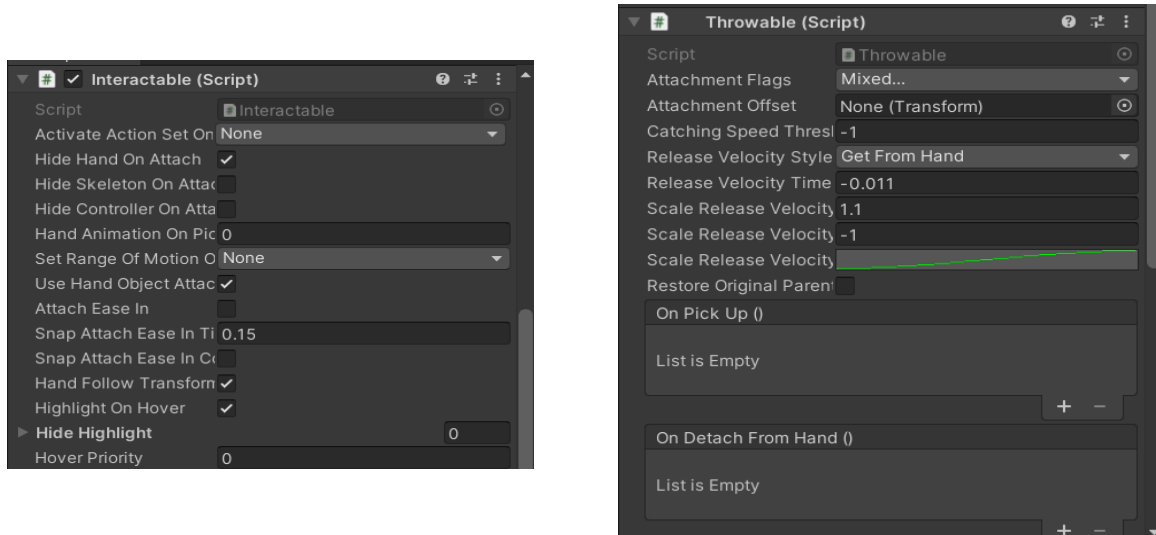
Εικόνα 94: Καθορισμός της ενέργειας στο περιβάλλον της Unity



Εικόνα 95: Παράδειγμα τυχαιοποίησης των θέσεων των κομματιών των δύο

Μόλις πραγματοποιείται η τυχαιοποίηση των θέσεων των κομματιών, ο κώδικας που την επιτρέπει κλειδώνει, έτσι ώστε να μην μπορεί να ξαναγίνει εάν ο παίκτης πατήσει κατά λάθος το αντίστοιχο κουμπί.

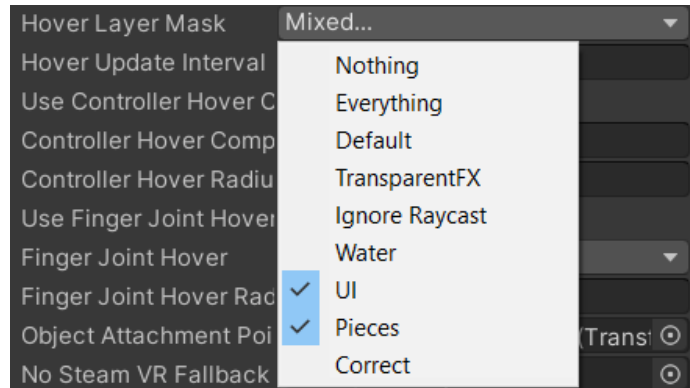
Από αυτό το σημείο ξεκινάει το παιχνίδι και ο χρήστης καλείται να πάρει τα κομμάτια και να τα τοποθετήσει στη σωστή τους θέση. Σε αντίθεση με την απλή τρισδιάστατη εφαρμογή στην οποία χρειαζόταν ανίχνευση της θέσης του ποντικιού για να μπορέσει ο χρήστης να αλληλεπιδράσει με τα κομμάτια, εδώ αρκεί αυτά να διαθέτουν δύο εξαρτήματα: “Interactable” και “Throwable”, τα οποία παρέχονται από την Steam και επιτρέπουν στον παίκτη να πάρει τα αντικείμενα, να τα μετακινήσει και να πραγματοποιήσει οποιαδήποτε ενέργεια με αυτά. Το κουμπί που χρειάζεται να πατήσει ο χρήστης προκειμένου να πάρει τα κομμάτια είναι το πλαϊνό κουμπί του δεξιού χειριστηρίου (Grip).



Εικόνα 96: Interactable και Throwable εξαρτήματα

Όταν τα κομμάτια πλησιάσουν την αρχική και άρα σωστή θέση (0, 0, 0), και συγκεκριμένα σε απόσταση 0,1 από αυτή, μεταφέρονται αυτόματα εκεί. Προκειμένου να κλειδώσει η δυνατότητα μετακίνησής τους μόλις έχουν τοποθετηθεί σωστά, αλλάζει το layer στο οποίο βρίσκονται και τοποθετούνται σε ένα νέο layer, το 10°, με όνομα “Correct”. Το layer αυτό αφαιρείται από την λίστα όσων μπορεί να αλληλεπιδράσει ο χρήστης, και σε αυτή παραμένει μόνο το 9° layer με όνομα “Pieces”, δηλαδή εκείνο που τοποθετούνται όλα τα κομμάτια εξαρχής και το layer που επιτρέπει την εισαγωγή δεδομένων από τον χρήστη (User Interface). Όταν ένα κομμάτι τοποθετηθεί σωστά, καλείται ένας κώδικας που εισάγεται σε κάθε κομμάτι και ορίζει την μεταφορά αυτού από το 9° στο 10° layer και δεν μπορεί να μετακινηθεί πλέον.

Ο κώδικας που στέλνει τα κομμάτια στη σωστή θέση διαθέτει μια Boolean μεταβλητή, η οποία χρησιμεύει στην πορεία για την ανίχνευση των κομματιών που έχουν τοποθετηθεί σωστά.



Εικόνα 97: Τα layer με τα οποία μπορεί να αλληλεπιδράσει ο χρήστης

Πίνακας 41: Η αλλαγή του layer του κομματιού

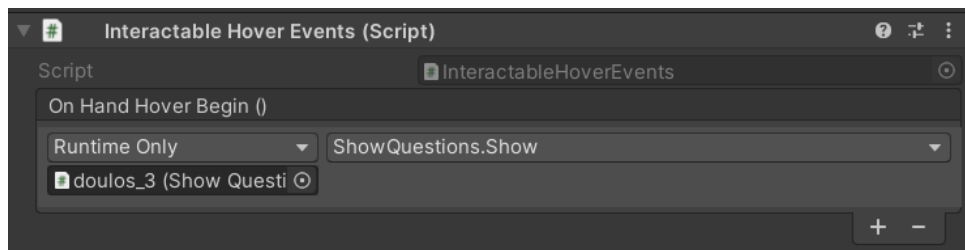
```
public class ChangeLayer : MonoBehaviour
{
    void OnDisable()
    {
        gameObject.layer = 9;
    }

    void OnEnable()
    {
        gameObject.layer = 10;
    }
}
```

Πίνακας 42: Τοποθέτηση των κομματιών στην αρχική θέση και αλλαγή του layer

```
float distance = Vector3.Distance(gameObject.transform.position, defPosition);
if (distance < 0.1f && vr_c.GetComponent<RandomizePositions_doulos>().enabled == false)
{
    gameObject.transform.position = defPosition;
    gameObject.GetComponent<ChangeLayer>().enabled = true;
    isCorrect = true;
}
```

Ανάμεσα στην επιλογή του κομματιού προς μετακίνηση και την μετακίνησή του, εμφανίζονται οι ερωτήσεις. Αναλυτικότερα, όταν ο χρήστης πλησιάζει το χέρι του σε ένα κομμάτι (on Hand Hover Begin), εμφανίζεται μια ερώτηση με χρήση του εξαρτήματος “Interactive Hover Events” και πρέπει να απαντηθεί σωστά για να μπορέσει να πιάσει το κομμάτι και να το τοποθετήσει στη σωστή θέση. Μέχρι να απαντηθεί η ερώτηση, το κομμάτι αλλάζει layer με την ίδια λογική όπως παραπάνω, έτσι ώστε να μην μπορεί ο παίκτης να αλληλεπιδράσει με αυτό. Οι ερωτήσεις τύπου Σωστό – Λάθος που έχουν αναφερθεί και σε προηγούμενο κεφάλαιο καθορίζονται σε μία λίστα και για κάθε μια από αυτές υπάρχει το πεδίο “is True”, που επιλέγεται όταν η ερώτηση είναι σωστή και αποεπιλέγεται όταν είναι λάθος.



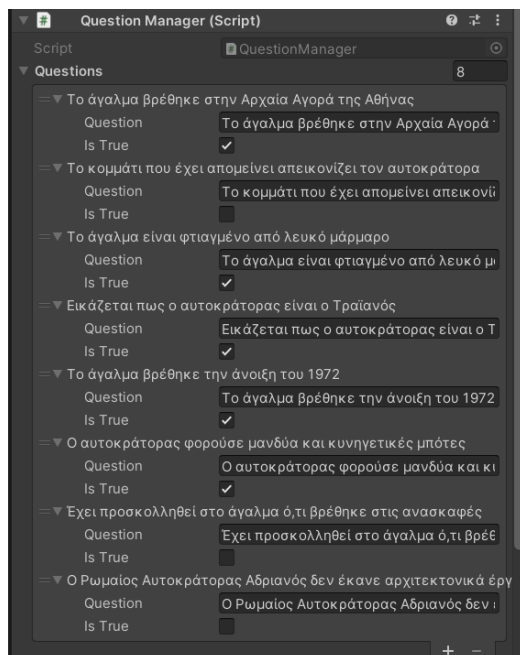
Εικόνα 98: Εμφάνιση της ερώτησης για ένα κομμάτι του αγάλματος όταν πλησιάζει ο χρήστης το χέρι του σε αυτό

Προκειμένου να μην εμφανίζεται ξανά ερώτηση όταν πάει ξανά ο παίκτης να πιάσει το κομμάτι εάν έχει ήδη απαντηθεί σωστά μία, εισάγεται μια Boolean μεταβλητή στον κώδικα που εμφανίζει την ερώτηση η οποία είναι αληθής όταν η ερώτηση έχει εμφανιστεί και ψευδής σε κάθε άλλη περίπτωση.

Πίνακας 43: Εμφάνιση των ερωτήσεων και επιστροφή εάν έχει ήδη εμφανιστεί ερώτηση για αυτό το κομμάτι

```
public void Show()
{
    if (!hasAppeared)
    {
        Instantiate(Quiz);

        hasAppeared = true;
    }
}
```



Εικόνα 99: Οι ερωτήσεις για το άγαλμα

Οι ερωτήσεις εμφανίζονται σε ένα πλαίσιο, το οποίο έχει δύο κουμπιά, «Σωστό» και «Λάθος», τα οποία διαθέτουν και box colliders για να είναι εφικτή η αλληλεπίδραση με αυτά.



Εικόνα 100: Μορφή του πλαισίου των ερωτήσεων

Ο καθορισμός των ερωτήσεων και των απαντήσεων γίνεται εντός του κώδικα, ο οποίος εξασφαλίζει ότι κάθε ερώτηση θα εμφανιστεί τυχαία, ότι δε θα επαναληφθεί καμία και πώς ερμηνεύεται η επιλογή του κουμπιού «Σωστό» ή «Λάθος». Όπως φαίνεται και στο παρακάτω κομμάτι κώδικα στον Πίνακα 44, σχηματίζεται μια λίστα στην οποία αποθηκεύονται όλες οι ερωτήσεις που δεν έχουν απαντηθεί και όταν ζητείται μια ερώτηση, η επιλογή γίνεται από

αυτή τη λίστα. Ανάλογα με το εάν το πεδίο "is True" είναι επιλεγμένο ή όχι, καθορίζονται και οι αντίστοιχες συναρτήσεις για επιλογή «Σωστό» ή «Λάθος» (User Select True & User Select False). Εάν ο χρήστης απαντήσει σωστά εμφανίζεται μήνυμα «Μπράβο!» και το κομμάτι επιστρέφει στο αρχικό του layer ενώ αν απαντήσει λάθος εμφανίζεται μήνυμα «Προσπάθησε ξανά» και φορτώνει μια νέα ερώτηση που καθορίζεται από την συνάρτηση Transition to Next Question.

Πίνακας 44: Διαχείριση των ερωτήσεων και απαντήσεων

```

void GetRandomQuestion()
{
    int randomQuestionIndex = Random.Range(0, unansweredQuestions.Count);
    currentQuestion = unansweredQuestions[randomQuestionIndex];

    questionText.text = currentQuestion.question;
}

IEnumerator TransitionToNextQuestion()
{
    unansweredQuestions.Remove(currentQuestion);
    yield return new WaitForSeconds(timeUntilNextQuestion);
    GetRandomQuestion();
}

public void UserSelectTrue()
{
    if (currentQuestion.isTrue)
    {
        //Debug.Log("Correct"); //mporei pleon na metakinhsei ta kommatia mexri na kanei
spawn h epomenh erwthsh
        resultText.GetComponent<TMPPro.TextMeshProUGUI>().text = "Μπράβο!";

        Destroy(Quiz, 2);
        unansweredQuestions.Remove(currentQuestion);

        for (int i = 0; i < pieces.Length; i++)
        {
            pieces[i].GetComponent<ChangeLayer>().enabled = false;
        }

    }

    else
    {
        resultText.GetComponent<TMPPro.TextMeshProUGUI>().text = "Προσπάθησε ξανά!";
        StartCoroutine(TransitionToNextQuestion());
    }
}

public void UserSelectFalse()
{
    if (!currentQuestion.isTrue)

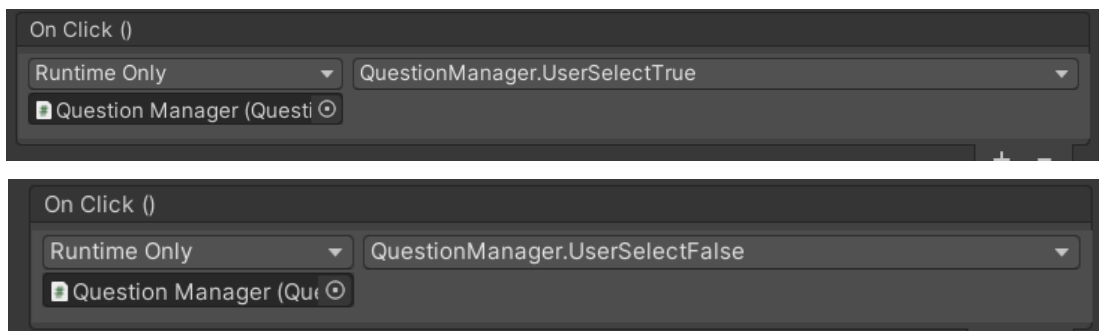
```

```
{
    resultText.GetComponent<TMPPro.TextMeshProUGUI>().text = "Μπράβο!";
    unansweredQuestions.Remove(currentQuestion);
    Destroy(Quiz, 2);

    for (int i = 0; i < pieces.Length; i++)
    {
        pieces[i].GetComponent<ChangeLayer>().enabled = false;
    }
}

else
{
    resultText.GetComponent<TMPPro.TextMeshProUGUI>().text = "Προσπάθησε ξανά!";
    StartCoroutine(TransitionToNextQuestion());
}
}
```

Η αλληλεπίδραση με τα κουμπιά γίνεται με το αριστερό χειριστήριο με χρήση της ακτίνας laser που παρέχεται από την Steam, Steam VR Laser Pointer. Η ακτίνα έχει μαύρο χρώμα σε όλη την διάρκεια του παιχνιδιού και γίνεται πράσινη όταν ο παίκτης πατήσει την αριστερή σκανδάλη για να επιλέξει ένα από τα δύο κουμπιά. Το επιλεγμένο κουμπί αποκτά πράσινο χρώμα και φορτώνει την αντίστοιχη συνάρτηση μέσω της ενσωματωμένης σε αυτό κλάσης onClick.



Εικόνα 101: Φόρτωση συνάρτησης για το σωστό (πάνω) και το λάθος (κάτω)

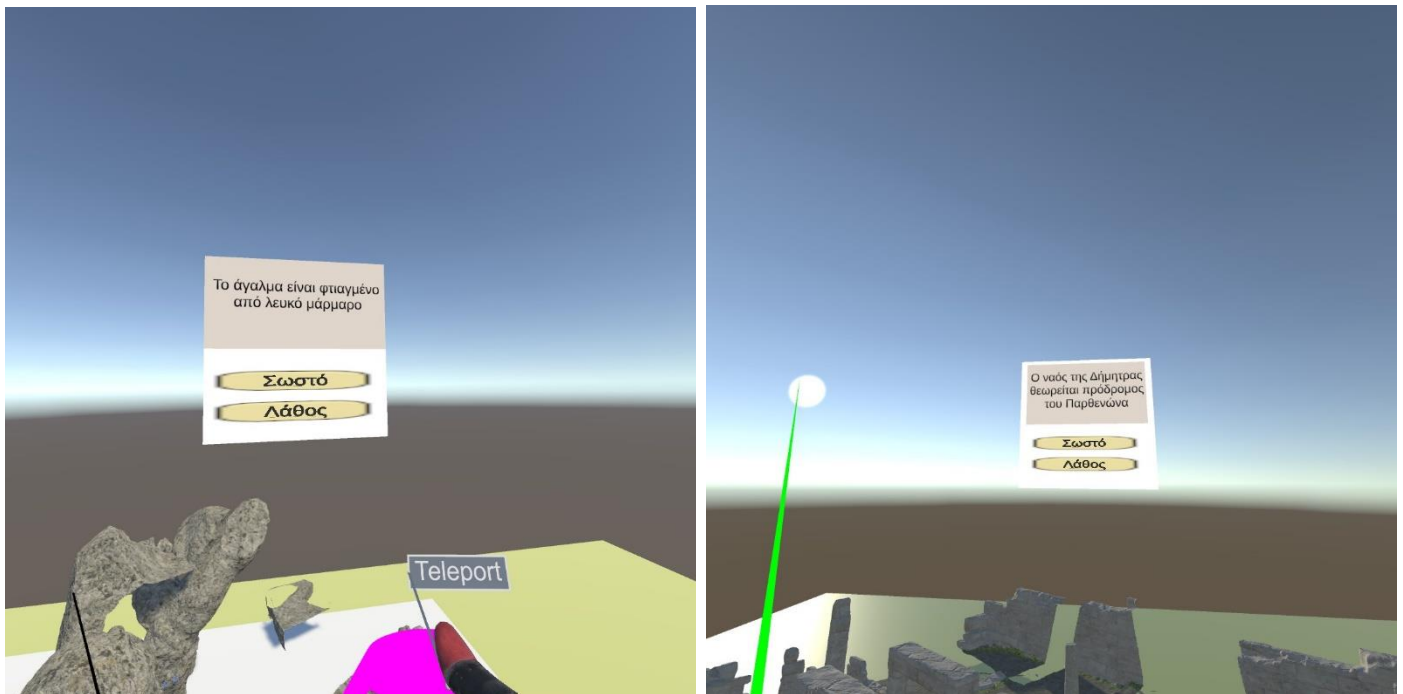
Προκειμένου να είναι δυνατή η επιλογή ενός κουμπιού με το laser συντάσσεται νέος κώδικας για την ακτίνα ο οποίος ανιχνεύει την θέση της. Όταν βρίσκεται εντός ενός από τα δύο κουμπιά και πατηθεί η σκανδάλη, καλείται η κλάση onClick.

Πίνακας 45: Κομμάτι κώδικα για την ανίχνευση της θέσης της ακτίνας και την αλληλεπίδρασή της με κουμπιά

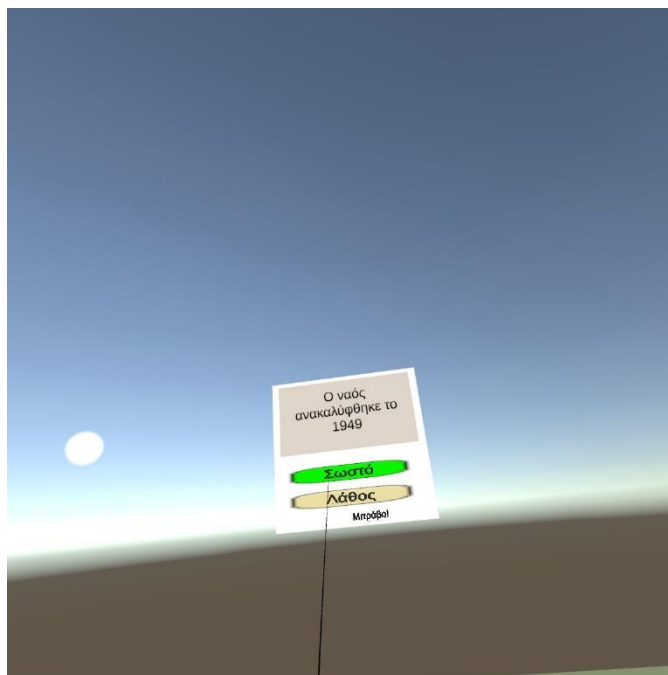
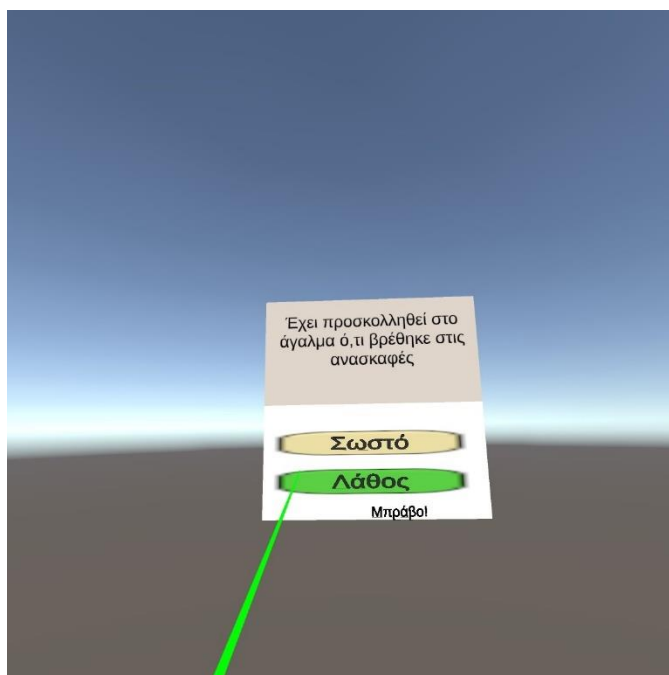
```
private void LaserPointer_PointerIn(object sender, PointerEventArgs e)
{
    if(e.target.gameObject.GetComponent<Button>() != null && btn == null)
    {
        btn = e.target.gameObject.GetComponent<Button>();
        btn.Select();

        pointerOnButton = true;
    }
}

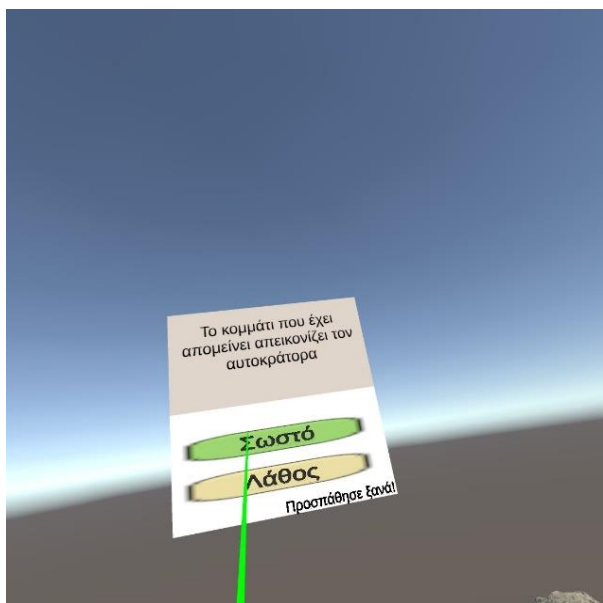
void Update()
{
    if(pointerOnButton)
    {
        if(leftTrigger.GetState(inputSource))
        {
            btn.onClick.Invoke();
        }
    }
}
```



Εικόνα 102: Παράδειγμα εμφάνισης ερωτήσεων για το άγαλμα (αριστερά) και το ναό (δεξιά)



Εικόνα 1043: Παράδειγμα σωστής απάντησης σε ερώτηση για το άγαλμα (αριστερά) και το ναό (δεξιά)



Εικόνα 1034: Παράδειγμα λάθος απάντησης σε ερώτηση για το άγαλμα (αριστερά) και για το ναό (δεξιά)

Το παιχνίδι ολοκληρώνεται όταν όλα τα κομμάτια τοποθετηθούν στις σωστές θέσεις. Στην παρούσα εφαρμογή συλλέγονται σε μία λίστα όλα τα κομμάτια που είναι στο layer 10 (Correct). Ταυτόχρονα ελέγχεται εάν η μεταβλητή “isCorrect” του κώδικα που στέλνει τα κομμάτια στις σωστές θέσεις όπως φαίνεται στον Πίνακας 42, είναι αληθής και εάν τα κομμάτια στη νέα λίστα είναι όσα τα κομμάτια με την ετικέτα “Pieces”. Όταν ισχύει αυτό, εμφανίζεται ένα μήνυμα που ανακοινώνει την ολοκλήρωση του παιχνιδιού. Χρησιμοποιείται η συνάρτηση IEnumerator ώστε να εφαρμοστεί μια καθυστέρηση στον έλεγχο και να μη γίνει εάν τα κομμάτια περάσουν κατά την τυχαίωσή τους από τις αρχικές τους θέσεις.

Πίνακας 46: Έλεγχος τέλους παιχνιδιού

```

void Update()
{
    StartCoroutine(CheckPositions());
}

IEnumerator CheckPositions()
{
    yield return new WaitForSeconds(2);

    foreach(GameObject go in correctList)
    {
        script = go.GetComponent<SendToCorrectPositions_naos>();
        if (go.layer == 10 && script.isCorrect == true)
        {
            correctPositions.Add(go);
            correctList.Remove(go);
        }
    }

    for(int i=0; i<correctPieces.Length; i++)
    {
        if (correctPositions.Count == correctPieces.Length )
        {
            gameComplete.GetComponent<Canvas>().enabled = true;
        }
    }
}
}

```


Μαζί με το μήνυμα, εμφανίζονται και δύο κουμπιά για έξοδο από το παιχνίδι ή για μετάβαση στην επόμενη πίστα, δηλαδή στο άγαλμα εάν η παρούσα είναι ο ναός και το αντίθετο. Η διαδικασία της μετάβασης γίνεται με χρήση της συνάρτησης `onClick` που διαθέτουν τα κουμπιά.



Εικόνα 105: Εμφάνιση μηνύματος «Τέλος Παιχνιδιού»

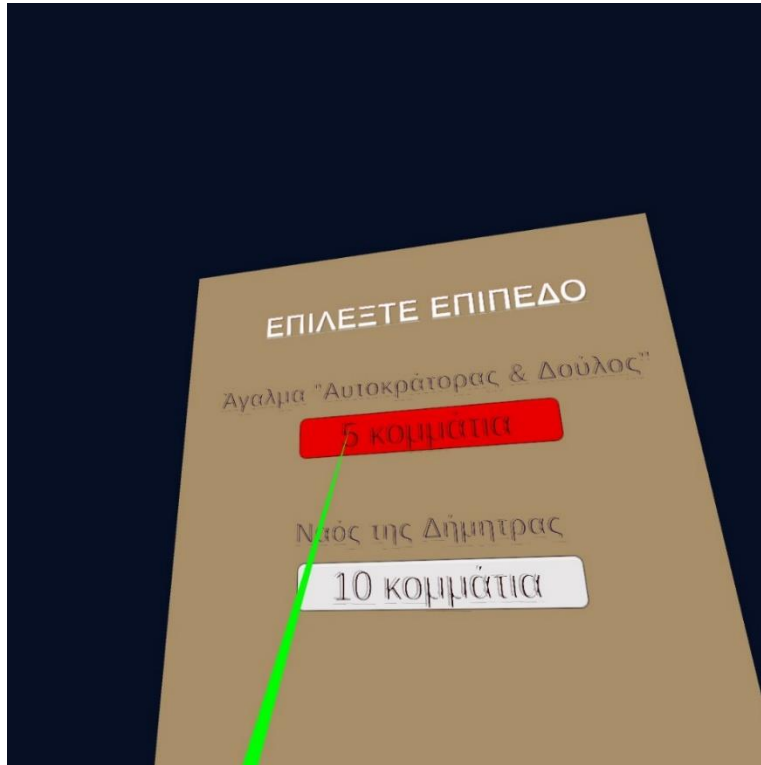
Πίνακας 47: Έξοδος από το παιχνίδι

```
public void doExitGame()
{
    Application.Quit();
}
```

Επιπρόσθετες λειτουργίες

1. Μενού επιλογών

Εφόσον μελετώνται δύο αντικείμενα, είναι σημαντικό να δίνεται η δυνατότητα στο χρήστη να επιλέξει με ποιο από τα δύο θέλει να παίξει. Έτσι, δημιουργείται ένα αρχικό μενού όπως φαίνεται στην εικόνα 106 στο οποίο ο χρήστης πατώντας το ανάλογο κουμπί μεταβαίνει στην αντίστοιχη σκηνή. Ο κώδικας που καθορίζει ποια σκηνή θα φορτώσει εισάγεται στην κλάση `onClick()` που έχει το κάθε κουμπί, όπως γίνεται και κατά την απάντηση των ερωτήσεων.



Εικόνα 106: Αρχικό μενού επιλογής

2. Εισαγωγικά πληροφοριακά κείμενα

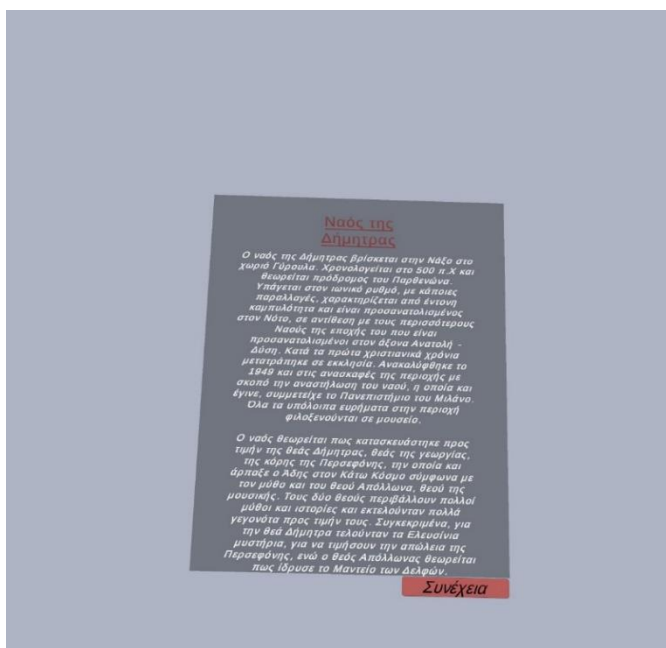
Στη συνέχεια, προκειμένου να επεκταθεί το κοινό του παιχνιδιού και να μην απευθύνεται μόνο σε όσους έχουν ήδη γνώσεις πάνω στα αντικείμενα, προστίθενται δύο επιπλέον σκηνές οι οποίες περιέχουν ένα σύντομο κείμενο για κάθε ένα από αυτά. Αναλυτικότερα, όταν ο χρήστης επιλέγει στην αρχική σκηνή με ποιο από τα δύο αντικείμενα θέλει να συνεχίσει, αντί να φορτώσει απευθείας το παιχνίδι εμφανίζεται ένα κείμενο που περιγράφει το επιλεγμένο αντικείμενο. Στο τέλος αυτού υπάρχει το κουμπί «Συνέχεια» το οποίο ο παίκτης πρέπει να πατήσει προκειμένου να προχωρήσει στο παιχνίδι.

Τα δύο κείμενα περιέχουν συνοπτικά σημαντικές πληροφορίες για τα αντικείμενα. Αναλυτικότερα, το κείμενο για το ναό είναι το εξής: «Ο ναός της Δήμητρας βρίσκεται στην Νάξο στο χωριό Γύρουλα. Χρονολογείται στο 500 π.Χ. και θεωρείται πρόδρομος του Παρθενώνα. Υπάγεται στον ιωνικό ρυθμό, με κάποιες παραλλαγές, χαρακτηρίζεται από έντονη καμπυλότητα και είναι προσανατολισμένος στον Νότο, σε αντίθεση με τους περισσότερους Ναούς της εποχής του που είναι προσανατολισμένοι στον άξονα Ανατολή - Δύση. Κατά τα πρώτα χριστιανικά χρόνια μετατράπηκε σε εκκλησία. Ανακαλύφθηκε το 1949

και στις ανασκαφές της περιοχής με σκοπό την αναστήλωση του ναού, η οποία και έγινε, συμμετείχε το Πανεπιστήμιο του Μιλάνο. Όλα τα υπόλοιπα ευρήματα στην περιοχή φιλοξενούνται σε μουσείο.

Ο ναός θεωρείται πως κατασκευάστηκε προς τιμήν της θεάς Δήμητρας, θεάς της γεωργίας, της κόρης της Περσεφόνης, την οποία και άρπαξε ο Άδης στον Κάτω Κόσμο σύμφωνα με τον μύθο και του θεού Απόλλωνα, θεού της μουσικής. Τους δύο θεούς περιβάλλουν πολλοί μύθοι και ιστορίες και εκτελούνταν πολλά γεγονότα προς τιμήν τους. Συγκεκριμένα, για την θεά Δήμητρα τελούνταν τα Ελευσίνια μυστήρια, για να τιμήσουν την απώλεια της Περσεφόνης, ενώ ο θεός Απόλλωνας θεωρείται πως ίδρυσε το Μαντείο των Δελφών.» ενώ για το άγαλμα: «Το άγαλμα "Αυτοκράτορας και Δούλος" βρέθηκε στην Αρχαία Αγορά της Αθήνας τον Μάιο του 1972 από την Αμερικανική Σχολή Κλασικών Σπουδών. Χαρακτηρίζεται από μεγάλη λεπτομέρεια και είναι φτιαγμένο από λευκό μάρμαρο. Το ολόκληρο κομμάτι απεικονίζει τα πόδια του αυτοκράτορα, που φαίνεται να φορούσε μανδύα και κυνηγετικές μπότες, και τον δούλο σκυμμένο. Ότι άλλο βρέθηκε στις ανασκαφές και ανήκει στο άγαλμα δεν έχει προσκολληθεί σε αυτό.

Εικάζεται πως ο αυτοκράτορας είναι ή ο Τραϊανός ή ο Αδριανός, και οι δύο σημαντικοί Ρωμαίοι Αυτοκράτορες. Ο Αδριανός, μάλιστα, φημίζεται για τα μεγάλα αρχιτεκτονικά του έργα τόσο στην Ιταλία όσο και στην Αθήνα.»



Εικόνα 107: Τα εισαγωγικά κείμενα για το ναό και το άγαλμα

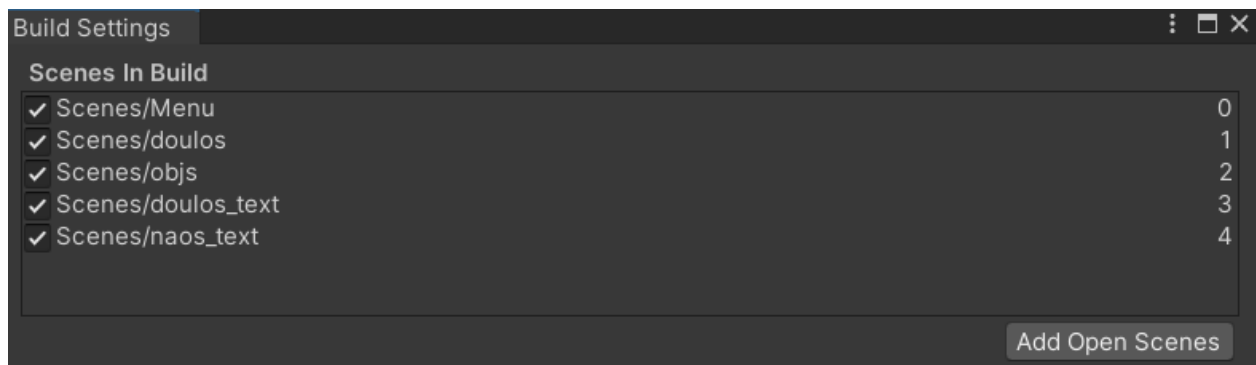
Για να φορτώσει η κάθε σκηνή, δηλαδή τα εισαγωγικά κείμενα για κάθε αντικείμενο στο αρχικό μενού επιλογών και το κάθε αντικείμενο με το κουμπί «Συνέχεια», καθώς και η έξοδος ή η επόμενη σκηνή κατά την εμφάνιση του μηνύματος «Τέλος Παιχνιδιού» συντάσσεται κομμάτι κώδικα βασιζόμενο στο Unity Scene Management.

Πίνακας 48: Παράδειγμα του κώδικα που εμφανίζει τη σκηνή

```
using UnityEngine.SceneManagement;

public class LoadScene1: MonoBehaviour
{
    public void Loaddoulos()
    {
        SceneManager.LoadScene(3);
    }
}
```

Ο αύξων αριθμός της σκηνής δίνεται από τις ρυθμίσεις του παιχνιδιού κατά το κτίσιμο αυτού (Build Settings). Όπως φαίνεται στην Εικόνα 107, στην αρχική σκηνή που είναι το μενού δίνεται ο αριθμός μηδέν (0), στην σκηνή του αγάλματος ο αριθμός ένα (1) και στο ναό ο αριθμός δύο (2) ενώ οι σκηνές που περιέχουν τα δύο κείμενα είναι τρία (3) και τέσσερα (4)



Εικόνα 108: Build Settings του παιχνιδιού

3. Οπτικοποίηση των χεριών του χρήστη

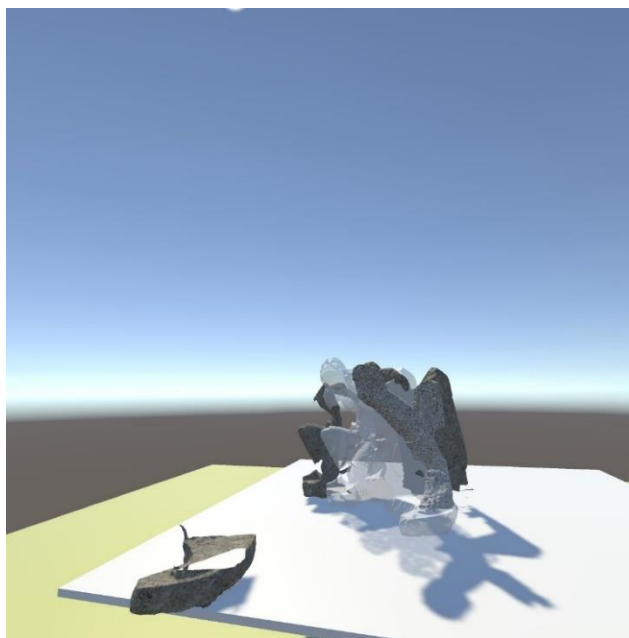
Επιπλέον, για να είναι πιο διαδραστικό το παιχνίδι, τα χειριστήρια αντικαθίστανται από γάντια που έχουν σχήμα χεριού και εξαφανίζονται μόνο όταν χρησιμοποιείται η ακτίνα laser. Τα γάντια που χρησιμοποιούνται παρέχονται από την βιβλιοθήκη της SteamVR και καλύπτουν όλες τις πραγματικές κινήσεις που κάνει το χέρι και τα δάχτυλα του χρήστη όταν αλληλοεπιδρά με τα κουμπιά των χειριστηρίων.



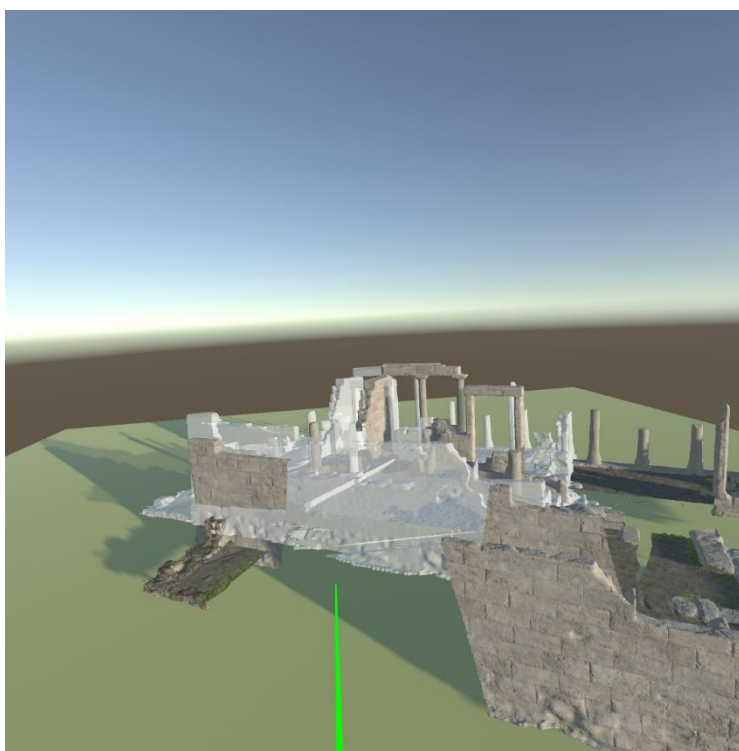
Εικόνα 109: Το μοντέλο που χρησιμοποιείται για το αριστερό χέρι

4. Εμφάνιση αντικειμένων – φάντασμα

Για διευκόλυνση του χρήστη κατά τη συναρμολόγηση των κομματιών, δημιουργούνται αντικείμενα – φάντασμα που δείχνουν ολόκληρο κάθε αντικείμενο στη σωστή θέση. Εισάγονται στο περιβάλλον της unity τα τρισδιάστατα μοντέλα των δύο αντικειμένων στην σωστή θέση και τους δίνεται λευκό ημιδιάφανο χρώμα. Ο χρήστης πατώντας την δεξιά σκανδάλη (trigger) εμφανίζει τα αντικείμενα – φάντασμα, και αυτά εξαφανίζονται όταν αφήνει την σκανδάλη.



Εικόνα 110: Το άγαλμα - φάντασμα



Εικόνα 111: Ο ναός - φάντασμα

Πίνακας 49: Εμφάνιση των αντικειμένων – φάντασμα με χρήση της σκανδάλης

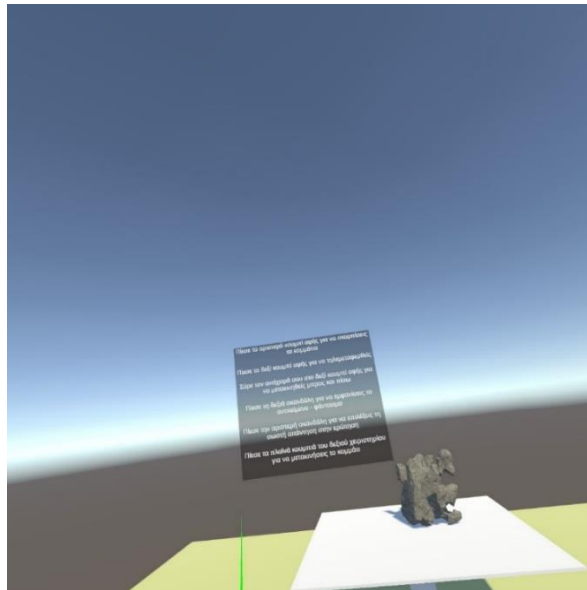
```
void Start()
{
    objsOnOff.AddOnStateDownListener(TriggerDown, handType);
    objsOnOff.AddOnStateUpListener(TriggerUp, handType);
}

public void TriggerUp(SteamVR_Action_Boolean fromAction, SteamVR_Input_Sources
fromSource)
{
    obj.GetComponent<MeshRenderer>().enabled = false;
}

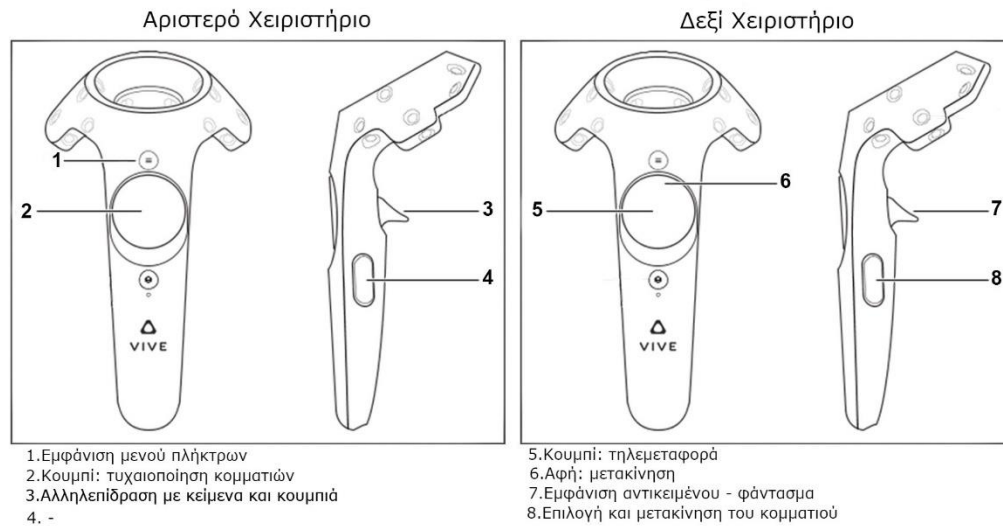
public void TriggerDown(SteamVR_Action_Boolean fromAction, SteamVR_Input_Sources
fromSource)
{
    obj.GetComponent<MeshRenderer>().enabled = true;
}
}
```

5. Μενού υπενθύμισης χειρισμών

Τέλος, αξιοποιείται το κουμπί «μενού» των χειριστηρίων ώστε να εμφανίζεται κατά το πάτημα αυτού μια καρτέλα που υπενθυμίζει στον χρήστη με ποιο κουμπί πραγματοποιείται κάθε κίνηση. Οι χειρισμοί οπτικοποιούνται στην Εικόνα 113.



Εικόνα 112: Μενού υπενθύμισης χειρισμών



Εικόνα 113: Χειρισμοί

Πίνακας 50: Εμφάνιση του μενού υπενθύμισης πλήκτρων με το πάτημα του αντίστοιχου κουμπιού

```
void Start()
{
    menuButton.AddOnStateDownListener(ButtonDown, handType);
    menuButton.AddOnStateUpListener(ButtonUp, handType);
}

// Update is called once per frame
public void ButtonDown(SteamVR_Action_Boolean fromAction, SteamVR_Input_Sources
fromSource)
{
    menu.GetComponent<Canvas>().enabled = true;
}

public void ButtonUp(SteamVR_Action_Boolean fromAction, SteamVR_Input_Sources
fromSource)
{
    menu.GetComponent<Canvas>().enabled = false;
}
```


5.4: Επίλυση προβλημάτων

Κατά την ολοκλήρωση και έκδοση του παιχνιδιού εντοπίστηκαν σημαντικά προβλήματα τα οποία επηρέαζαν την απόδοσή του και άρα την εμπειρία του χρήστη. Τα προβλήματα αυτά πηγάζουν είτε από λάθη του κώδικα είτε από ζητήματα της ίδιας της εφαρμογής. Σημαντικό ρόλο έχει και η δύναμη του συστήματος που υποστηρίζει το παιχνίδι, σε αυτή την περίπτωση ένας υπολογιστής.

5.4.1: Διόρθωση του κώδικα

Καθώς η Κεντρική Μονάδα Επεξεργασίας της Unity (CPU) μπορεί να «τρέξει» μόνο κώδικες που είναι σε πολύ απλή μορφή (machine code)^[41] και όχι σε σύνθετη όπως είναι η C#, είναι απαραίτητο κατά τη διάρκεια του παιχνιδιού να γίνεται μετατροπή από τη σύνθετη γλώσσα που χρησιμοποιείται στην πιο απλή που αντιλαμβάνεται η μηχανή. Η διαδικασία της μετάφρασης (compiling) χρειάζεται αρκετή από τη μνήμη της μηχανής, επομένως προκαλεί καθυστερήσεις. Εάν ο κώδικας δεν είναι γραμμένος με τον βέλτιστο τρόπο, οι καθυστερήσεις αυτές πολλαπλασιάζονται και η απόδοση μειώνεται ακόμα περισσότερο.

Προκειμένου να βελτιωθεί η απόδοση του κώδικα, συνιστανται αρκετές συμβουλές αναφορικά με τον τρόπο σύνταξης του^[42]. Στο κεφάλαιο αυτό αναφέρονται μόνο όσες εφαρμόστηκαν στο πλαίσιο της εργασίας.

Τα κυκλώματα επαναλαμβανόμενων συναρτήσεων (for loops) είναι πολύ συχνά αιτία δυσλειτουργίας κώδικα, καθώς «τρέχουν» συνεχώς μέχρι να εντοπίσουν τις συνθήκες που ικανοποιούν. Για να ελαφρύνει η διαδικασία απλοποιούνται οι ενέργειες που καλούνται εντός αυτών και αφαιρούνται όσες δεν είναι απαραίτητο να εκτελεστούν σε εκείνο το σημείο. Στο κομμάτι κώδικα που περιέχεται στον Πίνακα 51 φαίνεται πως εντός μιας συνάρτησης εξετάζεται η υποθετική συνθήκη, κι αν αυτή είναι αληθής τότε πραγματοποιείται ένας βρόγχος (for loop), ώστε να περιορίζεται το εύρος αναζήτησης των τιμών του βρόγχου.

```
public void UserSelectTrue()
{
    if (currentQuestion.isTrue)
    {
        resultText.GetComponent<TMPPro.TextMeshProUGUI>().text = "Μπράβο!";

        Quiz.GetComponent<Canvas>().enabled = false;
        System.GC.Collect();

        for (int i = 0; i < pieces.Length; i++)
        {
            pieces[i].GetComponent<ChangeLayer>().enabled = false;
        }

        StartCoroutine(TransitionToNextQuestion());
    }
}
```

Επιπλέον, καθώς η συνάρτηση Update καλείται σε κάθε καρέ του παιχνιδιού (per frame), είναι καλό να χρησιμοποιείται όσο γίνεται λιγότερο και για μικρού κόστους ενέργειες. Πιο συγκεκριμένα, η αναζήτηση εξαρτημάτων (Components), αντικειμένων και ετικετών δεν πρέπει να γίνεται στην Update, αλλά στην Start ή Awake συνάρτηση και να κωδικοποιηθεί με ένα όνομα ώστε χρησιμοποιείται αυτό όπου χρειάζεται. Ακόμη, όταν πραγματοποιούνται αλλαγές σε θέση ή μορφή αντικειμένου συνίσταται η αλλαγή της Update από την FixedUpdate, που σε έναν συμβατικό υπολογιστή μειώνει τις κλήσεις ανά καρέ. Στον Πίνακας 52 φαίνονται και οι δύο περιπτώσεις.

Πίνακας 52: Βελτιστοποίηση Update()

```
void Start()
{
    defPosition = new Vector3(0.0f, 0.0f, 0.0f);

    layerChange = GetComponent<ChangeLayer>();
    layerChange.enabled = false;
    randomize = vr_c.GetComponent<RandomizePositions_doulos>();
    //GetComponent<ChangeLayer>().enabled = false;
    isCorrect = false;
}

// Update is called once per frame
void FixedUpdate()
{
    float distance = Vector3.Distance(gameObject.transform.position, defPosition);

    if (distance < 0.1f && randomize.enabled == false)
    {
        gameObject.transform.position = defPosition;
        layerChange.enabled = true;
        isCorrect = true;
    }
}
}
```

Στην συνέχεια, άλλαξε ο τρόπος με τον οποίο εμφανίζονται οι ερωτήσεις. Αναλυτικότερα, καθώς η επίκληση (Instantiate) και μετά η καταστροφή (Destroy) του αντικειμένου των ερωτήσεων άφηνε υπολείμματα στην εφαρμογή, επιλέγεται να ενεργοποιείται και απενεργοποιείται κατάλληλα ο καμβάς που περιέχει την ερώτηση και τις απαντήσεις. Οι κώδικες που αλλάζουν είναι εκείνος που εμφανίζει τις ερωτήσεις όταν ο παίκτης πλησιάζει σε κάποιο κομμάτι και εκείνος που διαχειρίζεται τις ερωτήσεις. Η δεύτερη διόρθωση φαίνεται στον Πίνακα 51 υπογραμμισμένη με γαλάζιο χρώμα ενώ η πρώτη στον Πίνακα 53.

Πίνακας 53: Διορθωμένη εμφάνιση ερώτησης

```
public void Show()
{
    if (!hasAppeared)
    {
        Quiz.GetComponent<Canvas>().enabled = true;

        hasAppeared = true;
    }
}
```

Τέλος, είναι σημαντικό ο κάθε κώδικας να καλείται μόνο όπου χρειάζεται, όπως επίσης να αποφεύγεται η σύνταξη νέου κώδικα εάν δεν είναι απαραίτητο.

5.4.2: Βελτιστοποίηση της εφαρμογής

Στη συνέχεια, διορθώνονται κάποιες ρυθμίσεις των δεδομένων της εφαρμογής ώστε να επιτευχθεί η βέλτιστη δυνατή απόδοση. Αρχικά, ελαττώνονται οι πηγές φωτός εντός της σκηνής και αλλάζουν μορφή από την προκαθορισμένη σε “Baked”. Αυτό σημαίνει πως η Unity πραγματοποιεί εκ των προτέρων όλες τις ρυθμίσεις του φωτισμού και τις αποθηκεύει στον δίσκο, μειώνοντας την επεξεργασία την ώρα του παιχνιδιού^[44]. Στη συνέχεια ελαττώνεται η ανάλυση των υλικών που αποτελούν τις επιφάνειες των αντικειμένων από 4096 pixel που είναι η προκαθορισμένη σε 512. Τέλος, οργανώθηκε η επιφάνεια εργασίας της εφαρμογής ώστε όλα τα αντικείμενα που εξυπηρετούν τον ίδιο σκοπό να «είναι παιδιά» του ίδιου κενού αντικειμένου (parenting).

ΚΕΦΑΛΑΙΟ 6: Συμπερασματικά Σχόλια

6.1: Ανακεφαλαίωση της εφαρμογής

Η υλοποίηση της παρούσας εφαρμογής οδηγεί στο συμπέρασμα ότι η Εικονική Πραγματικότητα μπορεί να εφαρμοστεί με επιτυχία σε διάφορους τομείς και τα παιχνίδια σοβαρού σκοπού αποτελούν ένα χαρακτηριστικό τέτοιο παράδειγμα. Επιπλέον, η χρήση τρισδιάστατων μοντέλων αντικειμένων μπορεί να αλλάξει αρκετά τον τρόπο που προσεγγίζεται η πολιτιστική κληρονομιά, όχι μόνο για εκπαιδευτικούς σκοπούς αλλά και γενικότερα για την μελέτη, την διατήρηση και την αναβίωσή της.

Ειδικότερα, στην παρούσα εργασία δημιουργείται ένα παιχνίδι σοβαρού σκοπού μορφής παζλ, το οποίο αποκτά εκπαιδευτικό χαρακτήρα αφού «αναγκάζει» τον χρήστη να μελετήσει δύο αντικείμενα πολιτιστικής κληρονομιάς και να απαντήσει σε σχετικές ερωτήσεις τύπου «Σωστό – Λάθος» προκειμένου να μπορέσει να αλληλοεπιδράσει με αυτά. Για την δημιουργία του παιχνιδιού χρησιμοποιείται η μηχανή Unity 3D συντάσσοντας κώδικα σε γλώσσα C#. Η εμπύθιση σε περιβάλλον Εικονικής Πραγματικότητας επιτυγχάνεται μέσω του συστήματος HTC Vive, το οποίο πλεονεκτεί από τα υπόλοιπα συστήματα ως προς την δυνατότητά του να εντοπίσει τη θέση του χρήστη σε επίπεδο δωματίου.

Η διαδικασία ολοκλήρωσης του παιχνιδιού γίνεται σε τέσσερα στάδια. Αρχικά, εξετάζεται ένα πειραματικό αντικείμενο, συγκεκριμένα ένας κύβος, ο οποίος χωρίζεται σε επτά κομμάτια και αποτελεί την βάση για την δημιουργία του τρισδιάστατου παιχνιδιού. Η ύπαρξή του θεωρείται αναγκαία καθώς τα τρισδιάστατα μοντέλα των αντικειμένων που επιλέχθηκαν είναι πολύ βαριά και περίπλοκα, καθιστώντας τον πειραματισμό για τον καλύτερο τρόπο περαίωσης του παιχνιδιού δύσκολο και χρονοβόρο. Σε δεύτερο στάδιο επιλέγονται τα τρισδιάστατα μοντέλα, βελτιστοποιούνται φωτογραμμετρικά ώστε να αποδίδουν την καλύτερη δυνατή πληροφορία με τον λιγότερο δυνατό αριθμό τριγώνων επιφάνειας και να μην υπάρχουν κενά στα μοντέλα και χωρίζονται σε κομμάτια. Τα λογισμικά που χρησιμοποιούνται για την βελτιστοποίηση είναι το MeshLab, το Instant Meshes και το Meshmixer της Autodesk ενώ τα κομμάτια σχηματίζονται χρησιμοποιώντας την μηχανή παιχνιδιών Blender. Το μικρό αντικείμενο, δηλαδή το άγαλμα «Αυτοκράτορας και Δούλος» της Αρχαίας Αγοράς της Αθήνας χωρίζεται σε πέντε κομμάτια ενώ το μεγάλο, «ο ναός της Δήμητρας» της Νάξου, σε δέκα. Στη συνέχεια, τα κομμάτια που αποτελούν τα αντικείμενα εισάγονται στην μηχανή παιχνιδιών Unity ενωμένα μεταξύ τους σε μια αρχική θέση και στήνεται ένα ολοκληρωμένο παιχνίδι σε περιβάλλον υπολογιστή, βασιζόμενο πλήρως στο πειραματικό αντικείμενο με κάποιες αναγκαίες προσαρμογές. Θεωρήθηκε

απαραίτητο να μην γίνει κατευθείαν η εφαρμογή σε περιβάλλον Εικονικής Πραγματικότητας, που είναι το τέταρτο και τελευταίο στάδιο, διότι είναι αρκετά περίπλοκο επομένως ήταν καλύτερο ο σκελετός του παιχνιδιού να είναι έτοιμος και να γίνουν μόνο τροποποιήσεις στον κώδικα και τους χειρισμούς. Τα δύο αντικείμενα αποτέλεσαν δύο διαφορετικά επίπεδα δυσκολίας. Επιλέχθηκε το άγαλμα, που είναι και μικρότερο αντικείμενο, να χωριστεί σε πέντε κομμάτια και ο ναός, που είναι μεγαλύτερος, σε δέκα. Η επιλογή αυτή ήταν τυχαία, θα μπορούσε κάθε αντικείμενο να χωριστεί σε οποιονδήποτε αριθμό κομματιών.

Στην πορεία περιγράφεται το ολοκληρωμένο παιχνίδι ανά στάδιο, όπως αναλύεται και στο Κεφάλαιο 5.3: Διαδικασία δημιουργίας του παιχνιδιού Ξεκινώντας, ο χρήστης συναντά μια οθόνη αρχικού μενού, στην οποία επιλέγει το επίπεδο δυσκολίας, και άρα το αντικείμενο με το οποίο θα συνεχίσει. Ανάλογα με την επιλογή του, μεταβαίνει στο κείμενο περιγραφής του αντικειμένου, στο οποίο κείμενο βασίζονται οι μετέπειτα ερωτήσεις που τίθενται για την διεξαγωγή του παιχνιδιού. Έπειτα μεταφέρεται στην σκηνή που βρίσκεται το επιλεγμένο αντικείμενο. Από εδώ και πέρα, καλείται να σπάσει το αντικείμενο σε κομμάτια με το κατάλληλο κουμπί και να το συναρμολογήσει ξανά. Όταν πλησιάζει το χέρι του σε ένα κομμάτι με σκοπό να το πιάσει για να το τοποθετήσει στη σωστή θέση, εμφανίζεται μια ερώτηση την οποία πρέπει να απαντήσει σωστά για να μπορέσει να μετακινήσει το κομμάτι. Αν απαντήσει λάθος, εμφανίζεται νέα ερώτηση.

Για διευκόλυνση της συναρμολόγησης, με επιλογή του κατάλληλου πλήκτρου από το χειριστήριο εμφανίζεται στην σωστή θέση το αντικείμενο ολόκληρο, με λευκό ημιδιάφανο χρώμα (αντικείμενο – φάντασμα).

Μόλις όλα τα κομμάτια είναι στη σωστή τους θέση, το παιχνίδι ολοκληρώνεται και ο παίκτης μπορεί να επιλέξει εάν θέλει να βγει από αυτό ή να προχωρήσει στην επόμενη πίστα.

Όλοι οι χειρισμοί και οι κινήσεις που πρέπει να πραγματοποιήσει ο χρήστης υπενθυμίζονται σε αυτόν με επιλογή του κατάλληλου κουμπιού.

6.2: Αξιολόγηση Εφαρμογής

Ο περιορισμένος χρόνος και οι συνθήκες της πανδημίας της COVID -19 δεν επέτρεψε την αξιολόγηση της συγκεκριμένης εφαρμογής από επαγγελματίες και ειδικούς του χώρου των παιχνιδιών και της Εικονικής Πραγματικότητας. Δοκιμάστηκε μόνο από μικρό αριθμό φοιτητών και αξιολογήθηκε ως προς την χρησιμότητά της, την ευκολία ολοκλήρωσής της και την ικανότητά της να εξυπηρετήσει τον σκοπό της και να δώσει στους χρήστες εμπειρία εικονικής πραγματικότητας.

Αρχικά, οι χρήστες δεν συνάντησαν κάποια δυσκολία στον χειρισμό του εξοπλισμού εικονικής πραγματικότητας και συγκεκριμένα στα πλήκτρα των χειριστηρίων, καθώς τους υπενθυμίζονται όλες οι κινήσεις και τα πλήκτρα που τις πραγματοποιούν κατά τη διάρκεια του παιχνιδιού. Ωστόσο από πολλούς η μάσκα θεωρήθηκε βαριά και δύσκολε τις κινήσεις του κεφαλιού. Επιπλέον, η μετακίνηση στο χώρο με μεθόδους τηλεμεταφοράς δεν προκαλούσε αίσθημα ζαλάδας, όπως συνηθίζεται σε εφαρμογές εικονικής πραγματικότητας, και ήταν κατανοητές από όλους τους χρήστες. Η αλληλεπίδραση με τα κομμάτια και τις ερωτήσεις έγινε με ευκολία, με εξαίρεση κάποια κομμάτια του ναού που αντιστοιχούσαν στο δάπεδο αυτού, τα οποία ήταν πολύ χαμηλά και οι παίκτες έπρεπε να σκύβουν αρκετά για να τα πιάσουν. Η προσθήκη των αντικειμένων – φάντασμα θεωρήθηκε απαραίτητη για την συναρμολόγηση του παζλ, καθώς ήταν πολύ δύσκολο έως ανέφικτο να ενωθούν τα κομμάτια χωρίς να ξέρουν οι παίκτες πού πρέπει να τοποθετηθούν, ακόμα και αν σε κάθε τυχαιοποίηση των θέσεων των κομματιών ένα κομμάτι παρέμενε πάντα στη θέση του για να χρησιμοποιείται ως σημείο αναφοράς. Απαραίτητα θεωρήθηκαν και τα εισαγωγικά κείμενα με πληροφορίες για κάθε αντικείμενο προκειμένου να μπορέσουν οι παίκτες να απαντήσουν τις ερωτήσεις, κι άρα να αλληλοεπιδράσουν με τα κομμάτια και να τα τοποθετήσουν στις σωστές θέσεις.

Παρατηρήθηκε, επιπλέον, πως τα κομμάτια δεν «κολλούσαν» τέλεια μεταξύ τους. Αυτό συνέβη διότι κάθε κομμάτι περιβάλλεται από ένα ορθογωνικό πλέγμα το οποίο επιτρέπει την αλληλεπίδραση των κομματιών μεταξύ τους, δηλαδή να τοποθετούνται το ένα δίπλα στο άλλο χωρίς να εισέρχονται ή καλύπτονται μεταξύ τους, και δίνει την δυνατότητα στον χρήστη να επιλέξει και μετακινήσει το κομμάτι με το χειριστήριο. Καθώς το πλέγμα αυτό δεν ταυτίζεται με το σχήμα κάθε κομματιού, η ένωση δεν γίνεται τέλεια και δημιουργούνται κάποια κενά μερικών χιλιοστών μεταξύ των κομματιών. Ακόμη, καθώς τα κομμάτια σχηματίζονται «κόβοντας» το ενιαίο μοντέλο σε επιμέρους επιφάνειες, έχουν κάποια κενά στις πλευρές που εφάπτονται με τα άλλα κομμάτια. Το ζήτημα αυτό μπορεί να επιλυθεί με φωτογραμμετρική επεξεργασία των επιμέρους κομματιών, όπως πραγματοποιήθηκε και για ολόκληρα τα αντικείμενα.

Οι χρήστες ήταν σε γενικές γραμμές ευχαριστημένοι από την εμπειρία εμπύθισης και πρόθυμοι να δοκιμάσουν παρόμοιες εφαρμογές. Η παρούσα θεωρήθηκε χρήσιμη για την εκμάθηση πληροφοριών τόσο για αντικείμενα πολιτιστικής κληρονομιάς όσο και άλλων εννοιών, αφού αποτελεί έναν ευχάριστο και διαφορετικό τρόπο εξέτασης.

6.3: Επέκταση παρούσας εφαρμογής

Λαμβάνοντας υπόψη την τελική μορφή του παιχνιδιού, προκύπτουν αρκετές δυνατότητες επέκτασής της σε άλλα αντικείμενα και μορφές αλλά και βελτίωση της ίδιας της εφαρμογής.

Αρχικά, θεωρείται σημαντικό να μην καλύπτεται με επιπλέον ερωτήσεις μόνο η πιθανότητα ενός λάθους ανά κομμάτι (δηλαδή διπλάσιες ερωτήσεις από τα κομμάτια που πρέπει να τοποθετηθούν) αλλά να είναι αρκετά περισσότερες ώστε να είναι σίγουρο ότι ο παίκτης θα μπορέσει να προχωρήσει ακόμα και αν κάνει αρκετά λάθη. Μια εναλλακτική πρόταση για επίλυση αυτού του ζητήματος είναι η αλλαγή της μορφής των ερωτήσεων από «Σωστό – Λάθος» σε πολλαπλής επιλογής. Έτσι, ο παίκτης θα μπορεί να δοκιμάσει μια άλλη απάντηση στην ίδια ερώτηση αντί να αλλάζει ερώτηση και άρα θα χρειάζονται λιγότερες ερωτήσεις.

Δεύτερον, σημαντική βελτίωση της εφαρμογής αποτελεί η ανίχνευση της σωστής τοποθέτησης των κομματιών και μεταξύ τους, όχι μόνο με την αρχική τους θέση. Δηλαδή, εάν ο χρήστης καταφέρει να βρει δύο ή περισσότερα κομμάτια που ταιριάζουν μεταξύ τους, να μπορεί να τα ενώσει και να τα μεταφέρει σαν ένα στην αρχική σωστή θέση. Αυτό δεν εφαρμόστηκε στο πλαίσιο της παρούσας εργασίας, διότι τα κομμάτια είναι λίγα και ο χώρος στον οποίο σκορπίζονται περιορισμένος και κοντά στην αρχική θέση. Θεωρήθηκε, επομένως, δευτερεύον ζήτημα.

Επιπλέον, η συναρμολόγηση του ναού της Δήμητρας μπορεί να πάρει μορφή ανακατασκευής και να επεκταθεί και σε άλλους ναούς. Αναλυτικότερα, τα κομμάτια που θα αποτελούν το παζλ θα είναι συγκεκριμένα, για παράδειγμα θα χωριστεί στους κίονες, τα τοιχώματα ή τις πύλες και ο παίκτης θα καλείται να κατασκευάσει ξανά τον ναό, αντί απλώς να ενώσει τυχαία κομμάτια μεταξύ τους.

Η συγκεκριμένη εφαρμογή μπορεί ακόμη να επεκταθεί δημιουργώντας περισσότερα από δύο επίπεδα δυσκολίας. Μερικά παραδείγματα είναι να χωρίζονται οι ερωτήσεις σε επίπεδα δυσκολίας, δηλαδή στο εύκολο επίπεδο να είναι καθαρά ερωτήσεις που καλύπτονται από τα εισαγωγικά κείμενα ενώ στο δύσκολο πιο λεπτομερείς ερωτήσεις. Επιπλέον, θα μπορούσε να σπάσουν τα αντικείμενα σε περισσότερα ή λιγότερα κομμάτια, ή να προστεθούν μοντέλα άλλων αντικειμένων ώστε ο χρήστης να έχει πιο πολλές επιλογές. Τέλος, ένα επίπεδο αυξημένης δυσκολίας θα ήταν να μην είχε τη δυνατότητα ο χρήστης να εμφανίσει το αντικείμενο – φάντασμα, οπότε και να πρέπει να ενώσει τα κομμάτια χωρίς υποβοήθηση από το παιχνίδι.

Τέλος, προτείνεται να προστεθεί «βραβείο» στο τέλος του παιχνιδιού για την επιτυχημένη συναρμολόγησή του. Το βραβείο αυτό θα μπορούσε να είναι ένα ελαφρύ τρισδιάστατο μοντέλο του αντικειμένου ή των αντικειμένων με τα οποία έπαιζε, σε μορφή έτοιμη για τρισδιάστατη εκτύπωση.

Σε γενικότερο πλαίσιο, εφαρμογές σαν την παρούσα μπορεί να χρησιμοποιηθούν για εξέταση μαθητών ή φοιτητών με έναν πιο ευχάριστο τρόπο. Αναλυτικότερα, μπορεί να ενταχθεί στην εκμάθηση ιστορικών γεγονότων, στην ιατρική για την ανατομία των ανθρώπινων ή και ζωικών οργανισμών, στην αρχαιολογία και την αρχιτεκτονική για εξοικείωση με τη μορφή μνημείων και κτιρίων αλλά και την απόδοση μορφής σε αντικείμενα που δεν υπάρχουν πλέον. Η μοναδική προϋπόθεση για όλα τα παραπάνω είναι η ύπαρξη ή η δημιουργία ενός τρισδιάστατου μοντέλου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Δημοσιεύσεις

Alvarez, J. & Michaud, L., 2008. *Serious games: Advergaming, educaming, training*. Montpellier: IDATE.978-2-84822-169-4 επιμ.

Andreoli, R. και συν., 2018. A Framework to Design, Develop and Evaluate Immersive and Collaborative Serious Games in Cultural Heritage. *Journal on Computing and Cultural Heritage*, 11(1), pp. 1-22.

Falk Anderson, E. και συν., 2010. Developing serious games for cultural heritage: a state-of-the-art review. *Virtual Reality*, 14(4), pp. 255-275.

Kontogianni, G., 2015. *The contribution of 3D models to serious games applications*, Athens: s.n.

Kontogianni, G. και συν., 2016. *Developing and Exploiting 3D Textured Models for a Serious Game Application*, s.l.: s.n.

Maim, J. και συν., 2007. Populating ancient Pompeii with crowds of virtual romans. *VAST '07: Proceedings of the 8th International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, 26 November. pp. 109-116.

Michaela, M. και συν., 2014. Learning cultural heritage by serious games. *Journal of Cultural Heritage*, May - June, 15(3), pp. 318-325.

Michaela, M. & Lawrence, R., 2001. Virtual 3D Puzzles: A new method for exploring geometric models in VR. *Projects in VR*, 21(5), pp. 11-13.

Michael, D., Chen & Sande, 2006. *Serious Games: Games that educate, Train and Inform*. 1st επιμ. s.l.: Course Technology PTR.978-1592006229 επιμ.

Milgram, P. & Kishino, F., 1994. A Taxonomy of Mixed Reality in Visual Displays. *IEICE TRANSACTIONS on Information and Systems*, 25 12, pp. 1321-1329.

Muller, P., Vereenooghe, T., Ulmer, A. & Van Gool, L., 2005. Automatic reconstruction of roman housing architecture. *Recoding, Modeling and Visualization of Cultural Heritage*, pp. 287-299.

Pohlandt, D., Preim, B. & Saalfeld, P., 2019. *Supporting Anatomy Education with a 3D Puzzle in a VR Environment - Results from a pilot study*. Hamburg, Association for Computing Machinery, pp. 91-102.

Tarja, S., Mikael, J. & Backlund, P., 2007. *Serious Games- An Overview*, s.l.: s.n.

Zyda, M., 2005. From visual simulation to virtual reality to games. *Computer*, 19 September, 38(9), pp. 25-32.

Κοντός, Δ., 2019. Ανάπτυξη συστήματος μετρήσεων σε περιβάλλον Εικονικής Πραγματικότητας.

Κυριακή, Δ. & Νεφέλη, Η., 2004. *Εικονική Πραγματικότητα*, s.l.: s.n.

Παγουλάτου, Χ., Λάζου, Α. & Ντούρμα, Δ., 2017. *Συστήματα Εικονικής Πραγματικότητας*, Πάτρα: s.n.

Ραβανη, Ν., 2018. *Τρισδιάστατη μοντελοποίηση γλυπτών με χρήση ψηφιακών εικόνων*, Αθήνα: s.n.

Στεφάνου, Α. -. Β., 2018. *Η Συμβολή των νέων τεχνολογιών στην αρχαιολογία. Η περίπτωση του κλασικού Ναού της Δήμητρας στο Σαγκρί Νάξου.*

Ιστοσελίδες

1. <http://www.unesco.org/>
2. <https://old.grendelgames.com/>
3. https://en.wikipedia.org/wiki/Virtual_reality
4. <https://link.springer.com/referenceworkentry>
5. <https://repository.kallipos.gr/>
6. <https://www.investopedia.com/>
7. http://repfiles.kallipos.gr/html_books/
8. : <https://www.microsoft.com/en-us/hololens>
9. <https://www.americasarmy.com/>
10. <https://games4sustainability.org/gamepedia/floodsim/>
11. <https://www.game-learn.com/>
12. <http://www.cultureindevelopment.nl/>
13. <https://www.wearable.com/vr/>
14. <https://apnews.com/article/>
15. <https://visualise.com/>
16. <https://www.nasa.gov/>
17. <https://publicvr.info/>
18. <https://ancientathens3d.com/>
19. <https://artsandculture.google.com/>
20. <https://play.google.com/store/apps/>
21. <http://vr.hellenic-cosmos.gr/>
22. <https://www.slitherine.com/>
23. <https://en.wikipedia.org/wiki/Unity>
24. https://en.wikipedia.org/wiki/List_of_Unity_games
25. <https://en.wikipedia.org/wiki/Blender>
26. <https://en.wikipedia.org/>
27. <https://www.theverge.com/>
28. <https://www.engadget.com/>
29. <https://el.wikipedia.org/wiki/>
30. <https://www.mixanitouxronou.gr>
31. <https://el.wikipedia.org/wiki/>
32. <https://el.wikipedia.org/wiki/>
33. <http://agora.ascsa.net/id/agora/object/s>
34. <https://el.wikipedia.org/wiki>
35. <https://el.wikipedia.org/wiki/>
36. <https://www.blenderkit.com/>
37. <https://assetstore.unity.com/>

38. <https://freesound.org/s>
39. <https://learn.unity.com/>
40. <https://www.raywenderlich.com/>
41. https://en.wikipedia.org/wiki/Machine_code
42. <https://learn.unity.com/tutorial/fixing-performance-problems>
43. <https://resources.unity.com/>
44. <https://docs.unity3d.com/Manual/>
45. <https://store.steampowered.com>
46. <https://stackoverflow.com/>
47. <https://www.tutorialspoint.com/>
48. <https://ambientcg.com/>
49. <https://github.com/>
50. <https://forum.unity.com/>
51. <https://answers.unity.com/>
52. <https://www.vive.com/us/>
53. <https://eranistis.net/>
54. <https://dogana.gov.al/>

«Ανάπτυξη εκπαιδευτικού 3D Puzzle με αντικείμενα πολιτιστικής κληρονομιάς σε περιβάλλον Εικονικής
Πραγματικότητας»