



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ**

**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΤΡΙΣΔΙΑΣΤΑΤΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΨΑΡΙΩΝ ΚΑΙ ΑΝΑΠΤΥΞΗ
ΕΦΑΡΜΟΓΗΣ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΗΝ
ΑΝΑΔΕΙΞΗ ΘΑΛΑΣΣΙΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ**

Μπατανάρας Θωμάς

Επιβλέπων: Χρυσόστομος Στύλιος
Καθηγητής

**3D fish modeling and development of an augmented reality
application to highlight the marine environment**

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

Όνομα Επίθετο,

2. Μέλος επιτροπής

Όνομα Επίθετο,

3. Μέλος επιτροπής

Όνομα Επίθετο,

© Μπατανάρας, Θωμάς, 2023.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα προπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Μπατανάρας, Θωμάς

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Θέλω να ευχαριστήσω τους καθηγητές μου και συγκεκριμένα τον Κύριο Χρυσόστομο Στύλιο που μου εμπιστευτηκε το συγκεκριμένο θέμα και τον Κύριο Μποτίλια Ιωάννη για την αμέριστη βοήθεια και καθοδήγηση του που με βοήθησαν να βρω λύσεις σε κρίσιμα σημεία της εργασίας.

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία επιδιώκει να εξετάσει πως η τρισδιάστατη μοντελοποίηση και η τεχνολογία της επαυξημένης πραγματικότητας μπορούν να χρησιμοποιηθούν για την ανάδειξη του θαλάσσιου περιβάλλοντος. Ως περιοχή μελέτης επιλέχθηκε ο Αμβρακικός κόλπος, ενώ ως αντικείμενα μελέτης χρησιμοποιήθηκαν γνωστά ψάρια της περιοχής, όπως η Τσιπούρα, η Σαρδέλα, ο Κέφαλος και η Κουτσομούρα. Η διαδικασία περιλάμβανε την τρισδιάστατη μοντελοποίηση αυτών των ειδών. Παράλληλα, αναπτύχθηκε μια εφαρμογή επαυξημένης πραγματικότητας, η οποία ενσωμάτωσε αυτά τα τρισδιάστατα μοντέλα, προσφέροντας μια διαδραστική εμπειρία εκπαίδευσης και εξερεύνησης του θαλάσσιου κόσμου του Αμβρακικού.

Στο πλαίσιο της εργασίας δημιουργήθηκαν τρισδιάστατα(3D) μοντέλα ψαριών και συγκεκριμένα τέσσερα είδη που είναι η κουτσομούρα, η τσιπούρα, η σαρδέλα και ο κέφαλος επίσης δόθηκε στα μοντέλα ένα είδος κίνησης(animation). Έπειτα αναπτύχθηκε μια εφαρμογή επαυξημένης πραγματικότητας(AR) που σε συνδυασμό με τα 3D μοντέλα που δημιουργήθηκαν έχουν σκοπό την ανάδειξη θαλάσσιου περιβάλλοντος.

Λέξεις-κλειδιά: 3D μοντελοποίηση, Επαυξημένη Πραγματικότητα, Εφαρμογή

ABSTRACT

This thesis aims to examine how three-dimensional modeling and augmented reality technology can be used to highlight the marine environment. The Ambracian Gulf was selected as the study area, while known fish species of the area, such as Sea Bream, Sardine, Mullet and Goby, were used as study objects. The process involved the three-dimensional modeling of these species. Additionally, an augmented reality application was developed, which incorporated these three-dimensional models, providing an interactive educational and explanatory experience of the Ambracian marine world.

Within the scope of this thesis, three-dimensional(3D) models of fish were created, specifically four species which are Goby, Sea Bream, Sardine and Mullet. Additionally a type of motion(animation) was applied to the models. Subsequently an augmented reality(AR) application was developed which, in combination with the created 3D models, aims to highlight the marine environment.

Keywords: 3D Modeling, Augmented Reality, Application

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΕΡΙΛΗΨΗ	7
ABSTRACT	8
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	9
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	10
1. Εισαγωγή	12
2. Θεωρητικό Υπόβαθρο	19
2.1 3D Μοντελοποίηση.....	19
2.1.1 Δομικά στοιχεία και τρόποι αναπαράστασης 3D μοντέλων.....	21
2.1.2 Παράγοντες της 3D μοντελοποίησης.....	21
2.1.3 Εργαλεία 3D μοντελοποίησης.....	22
2.2 Επαυξημένη Πραγματικότητα (AR).....	22
2.2.1 Τύποι AR.....	23
2.2.2 Συσκευές AR.....	24
2.2.3 Ανάπτυξη Εφαρμογών AR.....	24
2.3 Unity 3D.....	24
2.3.1 Περιβάλλον Εργασίας.....	24
2.3.2 Προγραμματισμός και Ανάπτυξη.....	27
2.3.3 Γραφικά, Ήχος και Κίνηση.....	28
3. Σχεδιασμός και Ανάπτυξη	29
3.1 Σχεδιασμός 3D Αντικειμένων.....	29
3.1.1 Τεχνικές και Εργαλεία	29
3.2.1 Διαδικασία Σχεδιασμού.....	29
3.2 Σχεδίαση και Ανάπτυξη AR εφαρμογής.....	40
3.2.1 Τεχνικές και Εργαλεία.....	40
3.2.2 Ρυθμίσεις AR στο Unity.....	41
3.2.3 Ενσωμάτωση 3D Αντικειμένων και Animation.....	41
3.2.4 Προγραμματισμός Σεναρίων (Scripts) με C#.....	46
3.2.5 Σχεδιασμός UI.....	49
3.2.6 Ενσωμάτωση Ειδικών Εφέ (Special effects).....	51
4. Παρουσίαση AR Εφαρμογής	55
5. Συμπεράσματα και μελλοντικές επεκτάσεις	56
ΒΙΒΛΙΟΓΡΑΦΙΑ	57
ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ	59

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.1	3D μοντέλο για βιντεοπαιχνίδι.....	13
Εικόνα 1.2	3D εκτύπωση μιας φιγούρας.....	13
Εικόνα 1.3	3D απεικόνιση από σχέδιο κτιρίου.....	14
Εικόνα 1.4	3D Animation.....	14
Εικόνα 2.1.1	Wireframe.....	19
Εικόνα 2.1.2	Surface.....	19
Εικόνα 2.1.3	Solid.....	20
Εικόνα 2.1.4	Sculpting.....	20
Εικόνα 2.1.1.1	Δομικά Στοιχεία.....	21
Εικόνα 2.1.2.1	Βασικό περιβάλλον UV.....	22
Εικόνα 2.3.1.1	Βασικό γραφικό περιβάλλον Unity.....	30
Εικόνα 2.3.1.2	Build Settings.....	31
Εικόνα 2.3.2.1	Visual Scripting με nodes.....	32
Εικόνα 3.1.2	Γραφικό Περιβάλλον Blender.....	34
Εικόνα 3.1.2.1	Τσιπούρα.....	35
Εικόνα 3.1.2.2	Σαρδέλα.....	35
Εικόνα 3.1.2.3	Κέφαλος.....	35
Εικόνα 3.1.2.4	Κουτσομούρα.....	35
Εικόνα 3.1.2.5	Δημιουργία κυλίνδρου στο Blender.....	36
Εικόνα 3.1.2.6	Προσθήκη Ψαριού.....	36
Εικόνα 3.1.2.7	Προσθήκη επιπέδου(plane).....	36
Εικόνα 3.1.2.8	Επεξεργασία επιπέδου.....	36
Εικόνα 3.1.2.9	Δημιουργία material.....	37
Εικόνα 3.1.2.10	Προσθήκη modifier (mirror).....	37
Εικόνα 3.1.2.11	Προσθήκη κοκάλων (Bones).....	38
Εικόνα 3.1.2.12	Ρύθμιση κοκάλων.....	38
Εικόνα 3.2.2.1	AR Core.....	40
Εικόνα 3.2.2.2	AR Kit.....	40
Εικόνα 3.2.3.1	Τοποθέτηση ψαριών.....	41
Εικόνα 3.2.3.2	Σύνδεση animator controller με armature.....	41
Εικόνα 3.2.3.3	Προσθήκη animator στα μοντέλα.....	41
Εικόνα 3.2.3.4	Spline.....	42
Εικόνα 3.2.4.1	Δήλωση μεταβλητών σε C# Script.....	43
Εικόνα 3.2.4.2	Κώδικας.....	44
Εικόνα 3.2.4.3	Script Machine Component.....	45
Εικόνα 3.2.4.4	Blackboard.....	45
Εικόνα 3.2.4.5	Κώδικας.....	45
Εικόνα 3.2.4.6	Κώδικας.....	46

Εικόνα 3.2.6.1 Bubbles.....	47
Εικόνα 3.2.6.2 Post process layer.....	48
Εικόνα 3.2.6.3 Post process volume.....	48
Εικόνα 3.2.6.4 Box Collider.....	48
Εικόνα 4.1 Αρχικό μενού.....	49
Εικόνα 4.2 AR σκηγή 1.....	49
Εικόνα 4.3 AR σκηγή 2.....	49

1. Εισαγωγή

Τρισδιάστατη Μοντελοποίηση (3D Modeling)

Η τρισδιάστατη μοντελοποίηση είναι η διαδικασία δημιουργίας ενός πραγματικού ή φανταστικού αντικειμένου σε έναν τρισδιάστατο εικονικό κόσμο (Thilakanathan, 2016). Αυτή η τεχνική, βρίσκει εφαρμογή σε διάφορους τομείς, όπως τον τομέα των βιντεοπαιχνιδιών, όπου η χρήση των τρισδιάστατων μοντέλων είναι κρίσιμη για τη δημιουργία ρεαλιστικών κόσμων και χαρακτήρων (Thilakanathan, 2016). Επίσης, στον τομέα της σχεδίασης των προϊόντων, τα τρισδιάστατα μοντέλα χρησιμοποιούνται για τη δημιουργία εικονικών πρωτοτύπων, για τον έλεγχο των διαστάσεων και την πραγματοποίηση τυχόν διορθώσεων (Thilakanathan, 2016). Ακόμα ένα παράδειγμα αποτελεί ο τομέας της εκπαίδευσης, όπου δημιουργούνται προσομοιώσεις αρχαίων πόλεων, αρχιτεκτονικών επιτευγμάτων ή ιστορικών γεγονότων, προκειμένου να υποστηρίξουν την εκπαίδευση σε φοιτητές αρχαιολογίας (Thilakanathan, 2016).

Επιπρόσθετα, η τρισδιάστατη μοντελοποίηση χρησιμοποιείται και στον τομέα της ιατρικής, είτε για τη διενέργεια διαγνώσεων σε ασθενείς είτε για την εκπαίδευση φοιτητών ιατρικής σχολής, μέσω τρισδιάστατων μοντέλων ανθρώπινων οργάνων (Thilakanathan, 2016). Ακόμα, στον τομέα των καλών τεχνών, δίνεται η δυνατότητα δημιουργίας εικονικών έργων τέχνης και δυναμικής διαμόρφωσής τους. Τέλος, το πλέον διαδεδομένο παράδειγμα εφαρμογής της τρισδιάστατης μοντελοποίησης είναι η παραγωγή κινούμενων σχεδίων, με την χρήση κάποιου λογισμικού όπως το Blender, το Autocad, το Autodesk, το Maya, το 3DS Max κ.α. (FutureLearn, 2022).

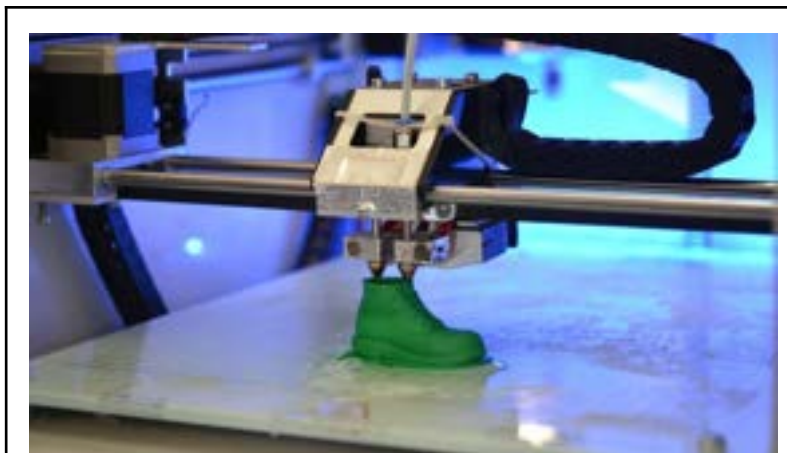
Συνολικά, η τρισδιάστατη μοντελοποίηση αντιπροσωπεύει μια τεχνική που εφαρμόζεται σε διάφορους τομείς, με βασικό σκοπό την αναπαράσταση αντικειμένων ή συμβάντων σε τρεις διαστάσεις. Η εκτενής χρήση της περιλαμβάνει ποικίλες εφαρμογές σε διάφορες βιομηχανίες, με τις πιο κοινές να είναι οι εξής (FutureLearn, 2022):

- *Βιομηχανία Ανάπτυξης Παιχνιδιών*: Με την βοήθεια της τρισδιάστατης μοντελοποίησης δίνεται η δυνατότητα στους προγραμματιστές παιχνιδιών (Game Developers) να φτιάξουν χαρακτήρες, σκηνικά, ακόμα και ολόκληρους κόσμους στα βιντεοπαιχνίδια τους.



Εικόνα 1.1 3D μοντέλο χαρακτήρα για Βιντεοπαιχνίδι
(πηγή: <https://retrostylegames.com/blog/3d-modeling-in-gaming-industry/>)

- *Τρισδιάστατη Εκτύπωση (3D Printing)*: Αποτελεί τη διαδικασία όπου η τρισδιάστατη μοντελοποίηση επιτρέπει στα μοντέλα να μετατραπούν σε φυσικά αντικείμενα. Αυτά τα αντικείμενα μπορεί να ποικίλουν από απλές φιγούρες έως και την κατασκευή προσθετικών, εμφυτευμάτων και ανατομικών αντικειμένων για χειρουργικό σχεδιασμό.



Εικόνα 1.2 3D εκτύπωση μιας φιγούρας με σχήμα παπουτσιού
(πηγή: <https://www.popsoci.com/technology/how-does-a-3d-printer-work/>)

- *Αρχιτεκτονική (Architecture)*: Οι αρχιτέκτονες μπορούν να επωφεληθούν από την τρισδιάστατη μοντελοποίηση για τη δημιουργία ενός μοντέλου κτιρίου πριν από την πραγματική του κατασκευή. Αυτό τους επιτρέπει να έχουν μια ολοκληρωμένη

εικόνα του τελικού αποτελέσματος χωρίς την ανάγκη για χρηματική δαπάνη και ταυτόχρονα εξοικονομείται χρόνος.

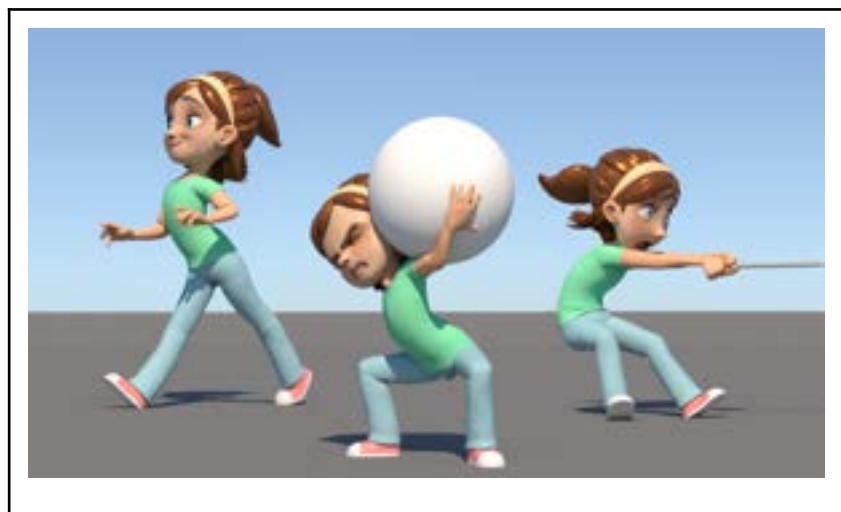


Εικόνα 1.3 3D απεικόνιση από το σχέδιο του κτιρίου

(πηγή:

<https://sgodleyarchitects.com/3-reasons-to-use-3d-rendering/>)

- *Τρισδιάστατα Κινούμενα Σχέδια (3D Animation)*: Αφού ολοκληρωθεί ένα τρισδιάστατο μοντέλο, ένας Animator μπορεί να αναλάβει την δημιουργία κινούμενων σχεδίων. Αυτό μπορεί να ενσωματωθεί σε ταινίες, τηλεοπτικές σειρές, ή ακόμη και στη δημιουργία σκηνικών, χαρακτήρων, και άλλων στοιχείων. (FutureLearn, 2022).



Εικόνα 1.4 3D Animation

<https://www.vrhotwires.com/how-to-make-3d-animations-in-simple-steps/>

Επαυξημένη Πραγματικότητα (AR) και εφαρμογές

Για την παρουσίαση ενός 3D μοντέλου στον πραγματικό κόσμο, απαιτείται η κάμερα μιας συσκευής και η χρήση μιας εφαρμογής με δυνατότητες Επαυξημένης Πραγματικότητας (Augmented Reality - AR) (Pavlova, n.d). Η επαυξημένη πραγματικότητα είναι ο συνδυασμός ψηφιακών δεδομένων μαζί με αισθητήρες κίνησης του πραγματικού κόσμου σε πραγματικό χρόνο. Η ιδέα της επαυξημένης πραγματικότητας άρχισε στην δεκαετία του 1990 σε ερευνητικά εργαστήρια. Από την δεκαετία του 2010 και με την εισαγωγή των smartphones και των tablets στην καθημερινότητα μας υπάρχει αύξηση στον βιομηχανικό και καταναλωτικό ενδιαφέρον της AR τεχνολογίας (Linowes & Babilinski, 2017).

Υπάρχουν δύο μέθοδοι που συνδυάζουν τον πραγματικό κόσμο με τον ψηφιακό. Η πρώτη μέθοδος, απαιτεί τη χρήση κάμερας μιας φορητής συσκευής όπως Smartphone ή Tablet, και κάποιο πρόγραμμα δημιουργίας ενός 3D περιβάλλον. Η δεύτερη μέθοδος επιτυγχάνεται μέσω έξυπνων AR γυαλιών (AR Smartglasses), τα οποία επιτρέπουν την προβολή του ψηφιακού κόσμου σε πραγματικό χρόνο. Οι εφαρμογές επαυξημένης πραγματικότητας αναπτύσσονται με εξειδικευμένα λογισμικά που υποστηρίζουν 3D δυνατότητες που επιτρέπουν στους προγραμματιστές να συνδυάσουν κινούμενα σχέδια ή ψηφιακές πληροφορίες με το περιβάλλον του πραγματικού κόσμου. (Linowes & Babilinski, 2017).

Μια εφαρμογή επαυξημένης πραγματικότητας (AR) αποτελεί λογισμικό που ενσωματώνει ψηφιακό οπτικό περιεχόμενο (π.χ. ήχο, εικόνα, 3D μοντέλα) στο πραγματικό περιβάλλον. Οι εφαρμογές που αναπτύσσονται με λειτουργίες AR χρησιμοποιούνται σε διάφορους κλάδους συμπεριλαμβανομένων της εκπαίδευσης (π.χ. Photomath), των εφαρμογών καταναλωτών (π.χ. IKEA place), της υγειονομικής περίθαλψης (π.χ. HealthVu), της δημόσιας ασφάλειας (π.χ. iOnRoad), του τουρισμού (π.χ. WorldAroundMe), του μάρκετινγκ (π.χ. HarleyDavidson) και της ψυχαγωγίας (π.χ. Pokemon GO, φίλτρα σε κάμερες διάφορων μέσων κοινωνικής δικτύωσης κ.α.). Οι πιο διαδεδομένοι τομείς, που χρησιμοποιούνται εφαρμογές AR περιγράφονται συνοπτικά παρακάτω:

- *Ιατρική Εκπαίδευση (Medical Training)*: Η επαυξημένη πραγματικότητα δίνει την δυνατότητα στους μαθητές ιατρικής σχολής την λειτουργία ενός εξοπλισμού μαγνητικής τομογραφίας έως την εκτέλεση σύνθετων χειρουργικών επεμβάσεων (Behringer et al., 1999).
- *Σχεδιασμός & Μοντελοποίηση (Design & Modeling)*: Οι αρχιτέκτονες χρησιμοποιούν εφαρμογές επαυξημένης πραγματικότητας για να οπτικοποιήσουν τα τελικά προϊόντα τους κατά τη δημιουργική διαδικασία (Behringer et al., 1999).
- *Τουριστική Βιομηχανία (Tourism Industry)*: Εφαρμογές AR προσφέρουν την δυνατότητα σε ταξιδιωτικές εταιρείες να παρουσιάζουν σε τουρίστες σημεία ενδιαφέροντος από το μέρος που θέλουν να επισκεφθούν.
- *Εκπαίδευση (Education)*: Οι εκπαιδευτικοί μπορούν να χρησιμοποιήσουν μια εφαρμογή Επαυξημένης Πραγματικότητας (AR) για τη διδασκαλία διαφόρων μαθημάτων. Για παράδειγμα, μπορούν να διδάξουν αστρονομία παρουσιάζοντας στους μαθητές έναν πλήρη χάρτη του ηλιακού συστήματος. Επίσης, μπορούν να διδάσκουν μουσική παρουσιάζοντας σε πραγματικό χρόνο τις νότες που πρέπει να παίξουν σε ένα μουσικό όργανο. (Linowes & Babilinski, 2017).
- *Ψυχαγωγία (Entertainment)*: Στην ψυχαγωγία συναντούμε εφαρμογές AR σε βιντεοπαιχνίδια, ταινίες, σειρές ακόμα και σε βιβλία. Αυτές προσφέρουν πληθώρα δυνατοτήτων, όπως η δυνατότητα για κάποιον να διαβάζει ένα βιβλίο και με τη χρήση μιας εφαρμογής AR να δημιουργείται η ψευδαίσθηση ότι ένας χαρακτήρας ζωντανεύει. Σε βιντεοπαιχνίδια, ο χαρακτήρας μπορεί να μετακινείται στον πραγματικό κόσμο, ενώ μέσω εφαρμογής AR ο χρήστης βλέπει έναν διαφορετικό κόσμο με ζώα, κτίρια κ.ά., όπως συνέβη και στο γνωστό παράδειγμα του Pokemon Go.
- *Αποθήκες Επιχείρησης (Business Logistics)*: Με την χρήση AR γυαλιών βοηθάει τους εργαζόμενους στις αποθήκες εταιρειών να βρίσκουν πιο εύκολη διαδρομή για τα προϊόντα που θα πρέπει να σταλούν στους πελάτες (Linowes & Babilinski, 2017).
- *Λιανεμπόριο (Retail)*: Με τη χρήση AR εφαρμογών, ο πελάτης μπορεί να οπτικοποιήσει το προϊόν που τον ενδιαφέρει και να το προσαρμόσει σύμφωνα με τις προτιμήσεις του. Ένα ενδεικτικό παράδειγμα είναι η εφαρμογή της Harley Davidson, όπου ο πελάτης μπορεί να προσαρμόσει την εμφάνιση της μοτοσυκλέτας που σκοπεύει να αγοράσει, προσδίδοντας την επιθυμητή εμφάνιση (Linowes & Babilinski, 2017).

Τεχνολογικές Προκλήσεις και Εφαρμογές στη Βιώσιμη Αλιεία και Υδατοκαλλιέργεια

Παράλληλα, η παγκόσμια κατανάλωση ψαριών ως τροφή έχει διπλασιαστεί κατά τις τελευταίες δεκαετίες, σημειώνοντας ανάλογη αύξηση στην υδατοκαλλιέργεια και τη σημασία της για την παροχή ψαριών και θαλασσινών. Επομένως, είναι απαραίτητη η υιοθέτηση υπεύθυνων μεθόδων υδατοκαλλιέργειας που σέβονται και προστατεύουν τις θάλασσες. Μόνο με τη βιώσιμη αλιεία και την υπεύθυνη κατανάλωση ψαρικών μπορούμε να διασφαλίσουμε την προστασία της θάλασσας, της ζωής που φιλοξενεί, και των πολύτιμων θαλάσσιων πόρων. (Βιώσιμη Ανάπτυξη, n.d.).

Η τεχνολογία μπορεί να διαδραματίσει ένα σημαντικό ρόλο στην πρόοδο και την εξέλιξη του τομέα της αλιείας και υδατοκαλλιέργειας. Οι τεχνολογικές εξελίξεις συμβάλλουν θετικά σε πολλούς τομείς, όπως η εφαρμογή συστημάτων GPS (Global Positioning System) για την προσδιορισμό ακριβούς θέσης στη θάλασσα (Παπαδόπουλος, 2016). Η καταγραφή και ανάλυση δεδομένων με τη χρήση τεχνητής νοημοσύνης (AI) και Μηχανικής Μάθησης (ML) (Αλεξίου, 2021) συνεισφέρουν επίσης στην βελτίωση της αλιείας. Η τεχνολογία δορυφορικής παρακολούθησης (Gould, 2023), η χρήση αισθητήρων για την καταγραφή δεδομένων (Tamm, 2022), και η εφαρμογή τεχνολογίας 3D Stereo Camera για την υποστήριξη της βιώσιμης αλιείας (NOAA, 2023) συμβάλλουν στην αποτελεσματικότερη και πιο βιώσιμη διαχείριση των αλιευτικών πόρων.

- Συστήματα GPS (Global Positioning System): Ένα σύστημα GPS χρησιμοποιείται για την παρακολούθηση της θέσης των αλιευτικών σκαφών, βοηθώντας έτσι στην βελτίωση της παραγωγικότητας και της ασφάλειας (Παπαδόπουλος, 2016).
- Καταγραφή και Ανάλυση δεδομένων με χρήση τεχνητής νοημοσύνης (AI) και Μηχανικής Μάθησης (ML): Χρησιμοποιούνται κυρίως για να προβλέψουν και να προτείνουν την καλύτερη διαχείριση αποθεμάτων και πόρων (Αλεξίου, 2021).
- Τεχνολογία Δορυφορικής Παρακολούθησης: Λαμβάνουμε εικόνες από δορυφόρους που μπορούν να χρησιμοποιηθούν για την παρακολούθηση θαλάσσιων περιοχών, τον εντοπισμό αλιευτικών σκαφών και την αξιολόγηση των περιβαλλοντικών συνθηκών (TS2, 2023).
- Καταγραφή Δεδομένων με Χρήση Αισθητήρων: Η χρήση αισθητήρων θερμοκρασίας και οξυγόνου βοηθάει στην παρακολούθηση των περιβαλλοντικών συνθηκών στα υδάτινα περιβάλλοντα (ThisFish, 2022).

Σκοπός εργασίας

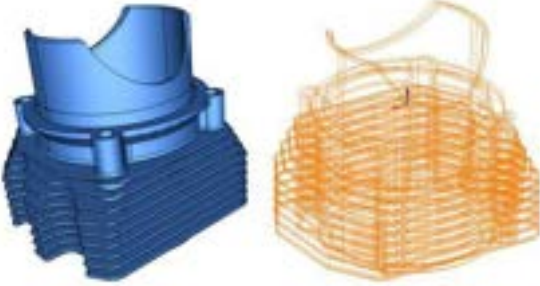

Η παρούσα εργασία επιδιώκει να εξετάσει πως η τρισδιάστατη μοντελοποίηση και η τεχνολογία της επαυξημένης πραγματικότητας μπορούν να χρησιμοποιηθούν για την ανάδειξη του θαλάσσιου περιβάλλοντος. Ως περιοχή μελέτης επιλέχθηκε ο Αμβρακικός κόλπος, ενώ ως αντικείμενα μελέτης χρησιμοποιήθηκαν γνωστά ψάρια της περιοχής, όπως η Τσιπούρα, η Σαρδέλα, ο Κέφαλος και η Κουτσομούρα. Η διαδικασία περιλάμβανε την τρισδιάστατη μοντελοποίηση αυτών των ειδών. Παράλληλα, αναπτύχθηκε μια εφαρμογή επαυξημένης πραγματικότητας, η οποία ενσωμάτωσε αυτά τα τρισδιάστατα μοντέλα, προσφέροντας μια διαδραστική εμπειρία εκπαίδευσης και εξερεύνησης του θαλάσσιου κόσμου του Αμβρακικού.

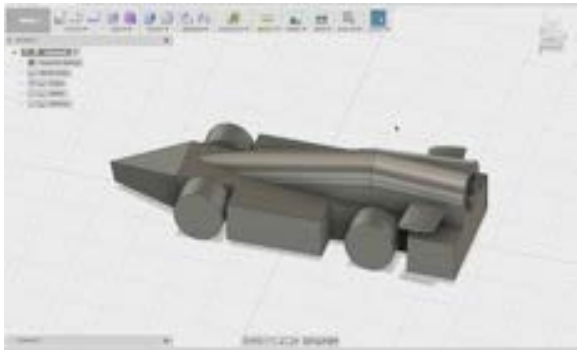

2. Θεωρητικό Υπόβαθρο

2.1 3D Μοντελοποίηση

Η τρισδιάστατη μοντελοποίηση (3D modelling) είναι μια διαδικασία γραφικών υπολογιστή για τη δημιουργία μιας μαθηματικής αναπαράστασης ενός τρισδιάστατου αντικειμένου ή σχήματος χρησιμοποιώντας εξειδικευμένο λογισμικό (Thilakanathan, 2016). Η ψηφιακή απόδοση του φυσικού αντικειμένου ονομάζεται τρισδιάστατο μοντέλο και κατά της δημιουργίας του χρησιμοποιούνται εργαλεία και τεχνικές τρισδιάστατης μοντελοποίησης όπως η μετακίνηση (Move), η εξώθηση (Extrude), η περιστροφή (Rotate) κ.α. Σε πολλές περιπτώσεις κρίνεται απαραίτητο η προσθήκη σημείων στον τρισδιάστατο μοντελοποιημένο χώρο του, που ονομάζονται κορυφές (Vertices) για την υποδιαίρεση των πολύγωνων και την δημιουργία ενός νέου τρισδιάστατου αντικειμένου (Khemlani, 2004). Οι τέσσερις βασικές κατηγορίες της τρισδιάστατης μοντελοποίησης είναι:

- **Wireframe:** Είναι μια σκελετική αναπαράσταση οποιουδήποτε αντικειμένου του πραγματικού κόσμου (Khemlani, 2004).
- **Surface:** Είναι η μέθοδος εμφάνισης ή παρουσίασης στερεών αντικειμένων (Spatial). (Khemlani, 2004).
- **Solid:** Είναι η υπολογιστική μοντελοποίηση τρισδιάστατων στερεών αντικειμένων. Ο στόχος του Solid Modelling είναι να διασφαλίσει ότι κάθε επιφάνεια είναι γεωμετρικά σωστή (*Solid Modeling*, n.d.).
- **Sculpting (or Polygonal):** Είναι η κατηγορία που ολοκληρώνει το πολυγωνικό πλέγμα ενός τρισδιάστατου μοντέλου με περίπλοκες λεπτομέρειες και υφές.

	
Εικόνα 2.1.1 Wireframe (πηγή: https://it-s.com/what-is-3d-wireframe-modeling/)	Εικόνα 2.1.2 Surface (πηγή: https://it-s.com/what-is-3d-surface-modeling/)

	
<p align="center">Εικόνα 2.1.3 Solid (πηγή: https://www.spatial.com/resources/glossary/what-is-solid-modeling)</p>	<p align="center">Εικόνα 2.1.4 Sculpting (πηγή: https://formlabs.com/blog/practice-3d-sculpting-custom-model/)</p>

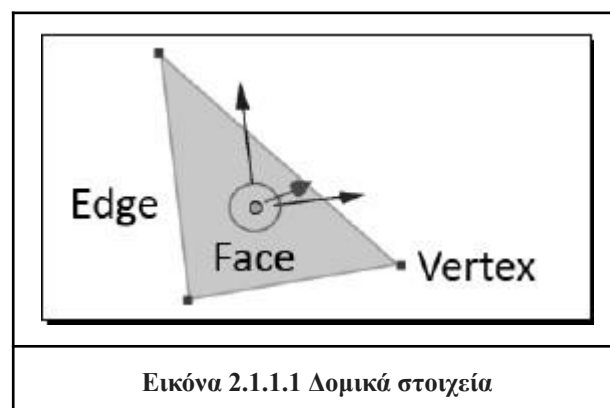
Τα τρισδιάστατα μοντέλα δημιουργήθηκαν με την τεχνική του Low Poly Modelling που σημαίνει ότι ένα μοντέλο low-poly συνήθως περιγράφεται ως ένα τρισδιάστατο μοντέλο με χαμηλό αριθμό πολυγώνων. Τα πολύγωνα είναι γεωμετρικά σχήματα (συνήθως τρίγωνα ή τετράπλευρα) που συνθέτουν ένα πολύγωνο πλέγμα και είναι υπεύθυνα για τη δημιουργία λεπτομερειών σε ένα τρισδιάστατο μοντέλο. Ο εμπειρικός κανόνας είναι ότι όσο υψηλότερο είναι το πλήθος πολυγώνων, τόσο μεγαλύτερο είναι το επίπεδο λεπτομέρειας (Low Poly, n.d.) (Fisher, 2014). Τα πλεονεκτήματα που προσφέρει η τεχνική του low poly modelling σε σχέση με την high poly modelling είναι τα εξής:

- *Γρηγορότερα αποτελέσματα:* Συνήθως χρησιμοποιούνται για εικόνες VR/AR, παιχνίδια, διαδραστικές εικόνες 360° κ.α. οπότε πρέπει να αποτελούνται από μικρό αριθμό πολυγώνων, ώστε οποιαδήποτε εφαρμογή να μπορεί να τα επεξεργαστεί γρήγορα (*High-Poly Models Vs Low-Poly Ones*, n.d.) (Fisher, 2014).
- *Εύκολη Επεξεργασία:* Η γεωμετρία των πολυγώνων μπορεί να γίνει αρκετά περίπλοκη και χρονοβόρα στην επεξεργασία της, οπότε ρίχνουμε τον αριθμό των πολυγώνων στο πλέγμα για πιο εύκολες και γρήγορες πιθανές αλλαγές που πρέπει να γίνουν στο μοντέλο (*High-Poly Models Vs Low-Poly Ones*, n.d.) (Fisher, 2014).
- *Λιγότερη Υπολογιστική Ισχύ:* Τα low-poly μοντέλα σχεδίασης είναι βελτιστοποιημένα για άμεση ή ακόμη και πραγματικού χρόνου απόδοση, έχοντας επομένως απλή και ελαφριά δομή σε όλα τα στοιχεία τους (*High-Poly Models Vs Low-Poly Ones*, n.d.) (Fisher, 2014).

2.1.1 Δομικά Στοιχεία και Τρόποι Αναπαράστασης 3D μοντέλων

-Δομικά στοιχεία:

- Κόμβοι (Vertices): Ο κόμβος είναι ένα σημείο στον χώρο που περιγράφεται με X, Y, Z συντεταγμένες.
- Συνδέσεις (Edges): Είναι μια γραμμή που ενώνει δύο κόμβους.
- Επιφάνειες (Faces): Είναι η επίπεδη περιοχή μεταξύ τριών ή περισσότερων κόμβων, που περιγράφονται από συνδέσεις. Αναφέρεται και ως πολύγωνο (Fisher, 2014).



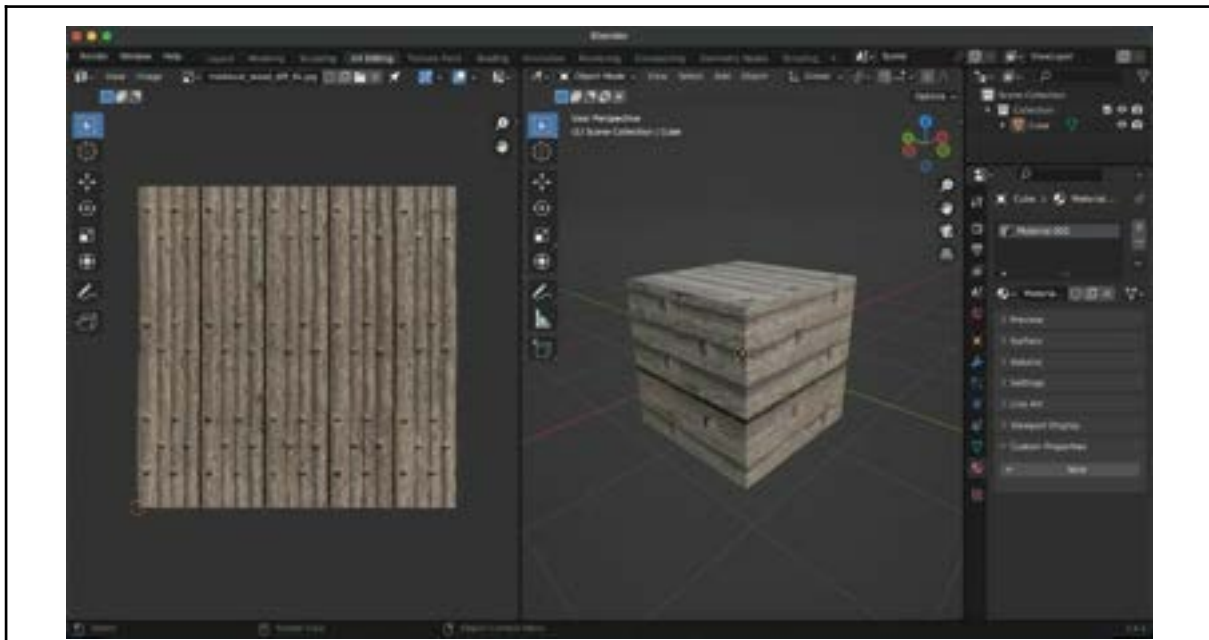
-Τρόποι Αναπαράστασης:

- **Polygonal Modelling:** Χρησιμοποιεί κόμβους, συνδέσεις και επιφάνειες για να σχηματίσει μοντέλα. Σχεδιάζονται από απλά σχήματα μέχρι σύνθετα με την προσθήκη επιπλέον συνδέσεων και κόμβων.
- **Curves and Surfaces:** Είναι συγκεκριμένοι τύποι 3D μοντέλων που εκφράζονται με μαθηματικές συναρτήσεις. Υπάρχουν δύο είδη, τα Bezier και τα NURBS που χρησιμοποιούν μια σειρά από κόμβους και συνδέσεις για να σχηματίσουν μια διαδρομή.
- **Volumetric Modeling:** Χρησιμοποιεί ένα 3D πλέγμα(grid) για να καθορίσει τις ιδιότητες ενός αντικειμένου σε σημεία(points).
- **Point Cloud:** Είναι ένα σύνολο σημείων σε έναν 3D χώρο. Αντιπροσωπεύει την επιφάνεια ενός 3D αντικειμένου. Περιέχει τρεις συντεταγμένες που προσδιορίζουν τη θέση του αντικειμένου, επίσης μπορεί να περιέχει επιπλέον πληροφορία όπως RGB χρώμα. (Liu et al., 2021).
- **Procedural Modeling:** Περιέχει ένα σύνολο τεχνικών για την δημιουργία 3D μοντέλων. Μερικές από τις τεχνικές αυτές είναι τα Συστήματα(Systems), Η παραγωγική μοντελοποίηση(Generative Modeling) και τα Fractals.(Adão et al., 2016).

2.1.2 Παράγοντες της 3D μοντελοποίησης

UV Mapping

Το UV mapping είναι η διαδικασία αντιστοίχισης των διαστάσεων της επιφάνειας ενός 3D μοντέλου σε ένα επίπεδο, γνωστό ως χάρτης UV. Το UV είναι αρχικά που αντιπροσωπεύουν τους δύο άξονες U και V σε ένα 2D texture. Εφαρμόζει εικόνες, μοτίβα και χρώματα στις επιφάνειες του μοντέλου, δίνοντάς του μια πιο ρεαλιστική και λεπτομερή εμφάνιση. Η δημιουργία ενός UV αναδεικνύει τη γεωμετρία του μοντέλου, λειτουργώντας ως οδηγός για την τοποθέτηση των υφών στις επιφάνειές του. Λειτουργεί σαν ένα σχέδιο, καθορίζοντας τις θέσεις όπου θα πρέπει να εφαρμοστούν οι ραφές, προκειμένου να γίνει η κοπή του τρισδιάστατου μοντέλου και η ισοπέδωσή του στο επίπεδο 2D (UV Mapping, 2023).



Εικόνα 2.1.2.1 Βασικό παράδειγμα UV στο Blender
(πηγή: <https://all3dp.com/2/blender-uv-mapping-simply-explained/>)

Animation

Το τρισδιάστατο animation αποτελεί την τέχνη της δημιουργίας κινούμενων τρισδιάστατων εικόνων μέσα σε ένα ψηφιακό πλαίσιο. Αυτές οι γραφικές αναπαραστάσεις δημιουργούνται με τη χρήση εξειδικευμένου λογισμικού 3D, συνδυάζοντας οπτικά εφέ και ακριβή χρονισμό. Με αυτήν τη διαδικασία, μπορούμε να δημιουργήσουμε οποιαδήποτε σκηνή, από έναν χαρακτήρα βιντεοπαιχνιδιού μέχρι ένα αυτοκίνητο σε μια διαφήμιση, το οποίο φαίνεται να

κινείται σε έναν τρισδιάστατο χώρο (3D Animation, 2021). Οι πιο γνωστοί τρόποι για να χρησιμοποιήσει κάποιος μια 3D animated τεχνική είναι για:

- *Σειρές και Ταινίες:* 3D animation συχνά χρησιμοποιείται για τη δημιουργία κινούμενων χαρακτήρων σε τηλεοπτικές σειρές και ταινίες. Αυτή η τεχνική μπορεί να προσδώσει μια αίσθηση ζωντάνιας στους χαρακτήρες, ενισχύοντας τη συναισθηματική σύνδεση που δημιουργείται με τους θεατές (3D Animation, 2021).
- *Βιντεοπαιχνίδια:* Το 3D animation μπορεί επίσης να αναβαθμίσει την εμπειρία σε βιντεοπαιχνίδια, προσδίδοντας μια πιο ζωντανή αίσθηση και ενισχύοντας τη δράση για τον παίκτη. Επιπλέον, το τρισδιάστατο animation μπορεί να χρησιμοποιηθεί για ειδικά εφέ σε περιβάλλοντα εικονικής πραγματικότητας και επαυξημένης πραγματικότητας, δημιουργώντας την αίσθηση ότι ο παίκτης βρίσκεται ενσωματωμένος στον κόσμο του παιχνιδιού σε πραγματικό χρόνο (3D Animation, 2021).
- *Αρχιτεκτονική και Σχεδιασμός:* Τα τρισδιάστατα κινούμενα σχέδια εξυπηρετούν επίσης στη δημιουργία ρεαλιστικών απεικονίσεων σπιτιών και κτιρίων πριν από την πραγματική κατασκευή τους. Σε συνδυασμό με εργαλεία εικονικής πραγματικότητας (VR) και επαυξημένης πραγματικότητας (AR), οι αρχιτέκτονες μπορούν ακόμη και να εισέλθουν και να περιηγηθούν μέσα στα σχέδιά τους. Αυτό αποτελεί ένα χρήσιμο εργαλείο για την ακριβή προειδοποίηση της κατασκευής, ενώ παράλληλα εξυπηρετεί τους σχεδιαστές εσωτερικής διακόσμησης, παρέχοντας τους μια σαφή εικόνα για το πώς θα διαμορφωθεί ένα δωμάτιο (3D Animation, 2021).

2.1.3 Τεχνική Ανάλυση Λογισμικών που χρησιμοποιούνται για 3D μοντελοποίηση.

- **Autodesk Maya:** Κυρίως χρησιμοποιεί δύο τρόπους για την μοντελοποίηση, την πολυγωνική(Polygonal) και την NURBS. Η πολυγωνική μέθοδος χρησιμοποιεί γεωμετρικά σχήματα που τα επεξεργαζόμαστε για να φτιάξουμε το επιθυμητό αποτέλεσμα. Η μέθοδος NURBS χρησιμοποιείται για οργανικά σχήματα επειδή περιέχει Curves(καμπύλες). Η τεχνική που χρησιμοποιεί το Maya για χρώματα και textures(υφές) για να δημιουργηθούν τα Materials ονομάζεται Shader. Για τον φωτισμό(Lighting) περιέχει έξι τύπους οι οποίοι είναι (Derakhshani, 2014):
 - Ambient
 - Directional
 - Point

- Spot
 - Area
 - Volume
- 3ds Max: Περιέχει πολλούς τρόπους μοντελοποίησης όπως τα Primitives(πρωτόγονοι) που είναι απλά γεωμετρικά σχήματα(κύβοι, κύλινδροι, πυραμίδες). Τα Meshes(πλέγματα) είναι σύνθετα αντικείμενα που έχουν δημιουργηθεί από πολλές επιφάνειες πολυγώνων, συνδεδεμένες μεταξύ τους. Τα Patches(μπαλώματα) είναι βασισμένα σε Spline Curves που μπορούν να τα τροποποιηθούν χρησιμοποιώντας Control Points. Τα Compound Objects περιέχουν Booleans, Scatter Objects κ.α. χρησιμοποιούνται κυρίως για την δημιουργία Terrain(έδαφος). Τα Particle Systems που είναι συστήματα από μικρά αντικείμενα που χρησιμοποιούνται μαζί ως ομάδα(group) (Murdock, 2012, #).
 - ZBrush: Έχει μια διαφορετική μέθοδο μοντελοποίησης η οποία είναι με το να χρησιμοποιεί brushes για γλυπτική(Sculpting), έτσι μπορεί οποιοσδήποτε να φτιάξει πολυγωνικά μοντέλα με υψηλή ανάλυση. Η τεχνολογία που χρησιμοποιεί το ZBrush ονομάζεται Pixols τα οποία είναι pixels που διατηρούν τις πληροφορίες τους όπως τον προσανατολισμό(orientation) τους και το βάθος(depth) τους (Patnode, 2008, #).
 - Blender: Χρησιμοποιεί γεωμετρικά σχήματα(Cube(κύβος), Cylinder(Κύλινδρος), Κύκλος(Circle) κ.α.) που ονομάζονται Primitives. Με μεθόδους επεξεργασίας όπως το Direct Mesh Editing που χρησιμοποιεί εργαλεία όπως, το Extrude, το Scale, το Rotate κ.α. Μία μέθοδος επεξεργασίας του μοντέλου είναι το Sculpting(γλυπτική) που δίνει την δυνατότητα να διαμορφωθεί το μοντέλο με υψηλή ακρίβεια και ανάλυση. Υπάρχουν διάφορα εργαλεία για τον φωτισμό της σκηνής όπως το Point Light που είναι ένα σημείο φωτός που ακτινοβολεί προς όλες τις κατευθύνσεις. Το Spot Light που λειτουργεί σαν κώνος, δηλαδή ακτινοβολεί πιο έντονα στην αρχική του θέση και λιγότερο στην τελική. Το Sun Light που μας δίνει την ψευδαίσθηση του ήλιου στην σκηνή. Για να δώσουμε στα αντικείμενα κάποιο χρώμα χρησιμοποιούμε τα Materials τα οποία είναι data-blocks και περιέχουν τρία είδη shaders (Flavell, 2010, #):
 - Surface: Ορίζει την εμφάνιση
 - Volume: Βρίσκεται μέσα στο αντικείμενο
 - Displacement: Ορίζει την μετατόπιση της επιφάνειας(Surface)

Υπάρχουν τα Image Textures που χρησιμοποιούν μια εικόνα που την εφαρμόζουν στα αντικείμενα. Ένας τρόπος για να εφαρμοστούν Textures στο αντικείμενο είναι με το UV Mapping το οποίο είναι σαν να έχεις μια εικόνα και να περιτυλίγεις το αντικείμενο με αυτήν. Το Blender μας δίνει την δυνατότητα να δημιουργήσουμε Animation. Ένας τρόπος να φτιαχτεί είναι με armature που λειτουργούν σαν κόκκαλα και εφαρμόζονται στο μοντέλο. Τα Armatures μπορούν να κινηθούν στον χώρο και μαζί τους το μοντέλο. Άλλοι τρόποι animation στο Blender είναι με την χρήση

Lattice, Markers, Motion Paths κ.α. Για το τελικό αποτέλεσμα της σκηνής υπάρχει η διαδικασία του Rendering η οποία παίρνει τα μοντέλα, τον φωτισμό και το animation εφόσον υπάρχει και βγάζει την εικόνα ή το βίντεο. Υπάρχουν δύο τρόποι Rendering στο Blender που είναι:

1. Cycles: Είναι ιδανικό για ρεαλιστικά αποτελέσματα. Χρειάζεται μεγαλύτερη υπολογιστική ισχύ από άλλες μεθόδους αλλά τα αποτελέσματα της είναι πιο ακριβές.
2. Eevee: Είναι Real-time render μηχανή που εστιάζει κυρίως στην ταχύτητα των αποτελεσμάτων.

2.2 Επαυξημένη Πραγματικότητα (AR)

Η επαυξημένη πραγματικότητα (AR) είναι η χρήση πληροφοριών σε πραγματικό χρόνο με τη μορφή κειμένου, γραφικών, ήχου και άλλων εικονικών βελτιώσεων που ενσωματώνονται με αντικείμενα του πραγματικού κόσμου. Η συγκεκριμένη τεχνολογία δίνει στον χρήστη την δυνατότητα να αλληλοεπιδρά με τον πραγματικό κόσμο (Linowes & Babilinski, 2017), εξελίσσεται συνέχεια και είναι πλέον από τις κύριες λειτουργίες στον τομέα της τεχνολογίας.

2.2.1 Τύποι AR

Η τεχνολογία AR περιλαμβάνει διάφορους τύπους που επιλέγονται ανάλογα με τις απαιτήσεις του υλικού και τον τομέα εφαρμογής. Κάποιοι από τους πιο διαδεδομένους περιγράφονται στη συνέχεια.

1. **Marker-Based AR (AR Βασιζόμενο σε Δείκτες):** Χρησιμοποιεί στόχους στον πραγματικό χώρο, όπως εικόνες (Image Targets), που όταν ανιχνευθούν από την κάμερα της συσκευής, τοποθετούν ψηφιακά αντικείμενα στην καθορισμένη θέση. (Linowes & Babilinski, 2017).
 - *Marker* : Ο βασικός τύπος στόχου, συνήθως σε τετράγωνο σχήμα, περιέχει ένα απλό περίγραμμα.
 - *Coded Marker*: Παρόμοιο με τον Marker, επιτρέπει τη χρήση του ίδιου αντικειμένου σε διάφορα χρώματα ή σχήματα. Με αυτό τον τρόπο, κερδίζεται ο χρόνος που θα χρειαζόταν να σχεδιάσουμε το κάθε Marker ξεχωριστά.
 - *Multi-Targets*: Επιτρέπει τη χρήση περισσότερων από έναν στόχο ταυτόχρονα για δημιουργία σχεδίου ή μονοπατιού από αντικείμενα..

- *Text Recognition*: Αναγνωρίζει και μεταφράζει το κείμενο σε αντικείμενο.
 - *Simple Shapes*: Αναγνωρίζει βασικά γεωμετρικά σχήματα και τα αλλάζει με τα καθορισμένα αντικείμενα. Για παράδειγμα αν αναγνωρίσει ένα κύβο, μπορεί να εμφανίσει μια ντουλάπα ρούχων.
 - *Spatial Mapping*: Αναγνωρίζει ολόκληρα πραγματικά μοντέλα ή αντικείμενα και τα μεταφράζει στον ψηφιακό κόσμο για να φτιαχτεί μια ολόκληρη ψηφιακή σκηνή. (Shrestha, 2021).
2. **Markerless AR (AR Άνευ Δεικτών)**: Χρησιμοποιεί τεχνολογίες όπως GPS, QR κώδικες ή αισθητήρες για τον προσδιορισμό της θέσης και του προσανατολισμού του χρήστη, κυρίως χρησιμοποιούμενη σε εφαρμογές πλοήγησης (Peddie, 2017).
 3. **Wearable AR (Φορητό AR)**: Χρησιμοποιεί φορητές συσκευές όπως έξυπνα γυαλιά (Smart Glasses) ή ακουστικά (Headsets), εμφανίζοντας ψηφιακό περιεχόμενο στο οπτικό πεδίο του χρήστη, συνήθως με διαφανείς οθόνες (Peddie, 2017).
 4. **Superimposition-based AR (AR με Υπερθέτηση)**: Συμπεριλαμβάνει εικονικά αντικείμενα, σχολιασμούς κειμένων ή γραφικά που συνυπάρχουν με τον πραγματικό κόσμο, ευρέως χρησιμοποιούμενα σε διαφημιστικές καμπάνιες. (Peddie, 2017).
 5. **Recognition-based AR (AR με Αναγνώριση)**: Αναγνωρίζει αντικείμενα ή σκηνές στον φυσικό κόσμο και προσθέτει ψηφιακό περιεχόμενο μέσω μιας AR εφαρμογής, συνήθως χρησιμοποιούμενη σε εγχειρίδια, προσομοιώσεις εκπαίδευσης και οπτικοποίηση προϊόντων. (Peddie, 2017).

2.2.2 Συσκευές AR

- Κινητά τηλέφωνα ή τάμπλετ χρησιμοποιούν την τεχνολογία της επαυξημένης πραγματικότητας με την χρήση της κάμεράς τους για να απαθανατίσουν τον πραγματικό κόσμο και να το συνδυάσουν με ψηφιακά αντικείμενα. Κάποια από τα χαρακτηριστικά που έχουν οι κινητές συσκευές για AR είναι (Linowes & Babilinski, 2017, #):

- Οθόνη αφής για την είσοδο πληροφορίας.
- Μπροσ-πίσω Κάμερα
- CPU(Κύριος Επεξεργαστής), GPU(Επεξεργαστής γραφικών), Μνήμη
- Αισθητήρες κίνησης όπως το γυροσκόπιο(Gyroscope) για περιστροφική κίνηση και το επιταχυνσιόμετρο(Accelerometer) για την γραμμική κίνηση.

- Τα γυαλιά επαυξημένης πραγματικότητας ή αλλιώς τα smart glasses απεικονίζουν μέσω αυτών τον ψηφιακό κόσμο που έχει φτιαχτεί μέσω κάποιου υπολογιστή. Έχουν μια see-through οθόνη που αναμειγνύει το φως από τον πραγματικό κόσμο με το εικονικό αντικείμενο. Επίσης όσο περισσότερο φως έχει η περιοχή γύρω μας τόσο καλύτερα θα φαίνονται τα αντικείμενα μέσω των γυαλιών. Υπάρχουν τα AR Headsets που έχουν τα AR γυαλιά και σε συνδυασμό με κάποιο hardware τα χρησιμοποιούμε για να μπορούμε να κινηθούμε στον χώρο. Τυπικά τα Headsets περιλαμβάνουν (Linowes & Babilinski, 2017):

- Οπτικό φακό με συγκεκριμένο οπτικό πεδίο
- Μπροστά Κάμερα
- Αισθητήρες βάθους για εντοπισμό θέσης και αναγνώριση χεριού
- Μικρόφωνο

2.2.3 Ανάπτυξη Εφαρμογών AR

- Τεχνολογία ARKit: Είναι το πλαίσιο ανάπτυξης AR για κινητές συσκευές της Apple. Με την χρήση της κάμερα των συσκευών και με το ARKit μας δίνεται η δυνατότητα να δημιουργήσουμε 2D και 3D εικονικό περιεχόμενο για μια καλή εμπειρία προς τον χρήστη. Η διαφορά του ARKit σε σχέση με άλλα πλαίσια ανάπτυξης AR είναι ότι περιέχει τα markerless tracking (AR δίχως δείκτη). Χρησιμοποιείται για την δημιουργία κάποιων emoji που εφαρμογές όπως το iMessage και το Facetime χρησιμοποιούν. Το IKEA επίσης χρησιμοποιεί στην εφαρμογή του IKEA Place για την απεικόνιση του χώρου και την πιθανή τοποθέτηση κάποιου επίπλου. Οι τεχνολογίες πίσω από το ARKit είναι (Language & raywenderlich.com Tutorial Team, 2022, #):
 - AVFoundation: Ελέγχει τις εισόδους και εξόδους ήχου και βίντεο.
 - CoreMotion: Ανιχνεύει την κίνηση της συσκευής στον χώρο.
 - CoreML: Με την βοήθεια μηχανικής μάθησης (machine learning) εντοπίζει τις επιφάνειες που θα μπορεί να τοποθετηθεί κάποιο αντικείμενο.
- Τεχνολογία ARCore: Είναι το πλαίσιο ανάπτυξης AR της Google για κινητές συσκευές με λειτουργικό Android. Το ARCore περιέχει αισθητήρα κίνησης για τον εντοπισμό θέσης στον πραγματικό κόσμο, Αισθητήρα για το περιβάλλον που ανιχνεύει το μέγεθος και την θέση όλων των επιφανειών (Grubert & Grasset, 2013, #).

- Vuforia: Είναι Software Development Kit(SDK) για την δημιουργία εφαρμογών επαυξημένης πραγματικότητας. Προσθέτει λειτουργίες όπως την αναγνώριση εικόνας, αντικειμένων, και επιφανειών για την αλληλεπίδραση του πραγματικού κόσμου με τον ψηφιακό. Η χρήση της Vuforia μέσω της Unity υποστηρίζει την ανάπτυξη εφαρμογών για πολλές πλατφόρμες όπως Android συσκευές, iOS συσκευές και Windows 10 συσκευές. Με το που εγκατασταθεί η Vuforia σε κάποια εφαρμογή της Unity τότε μας δίνει την δυνατότητα της κάμερας AR που μπορεί να ανιχνεύσει διάφορους στόχους(markers) όπως:
 - Image Targets: Ανιχνεύει επίπεδες εικόνες.
 - Multi Targets: Ανιχνεύει παραπάνω από ένα στόχο παράλληλα.
 - Barcode Scanner: Εντοπίζει Barcodes για κάποια λειτουργία στην AR εφαρμογή.
 - Model Targets: Εντοπίζει αντικείμενα βάση των γεωμετρικών σχημάτων τους.

2.3 Unity 3D

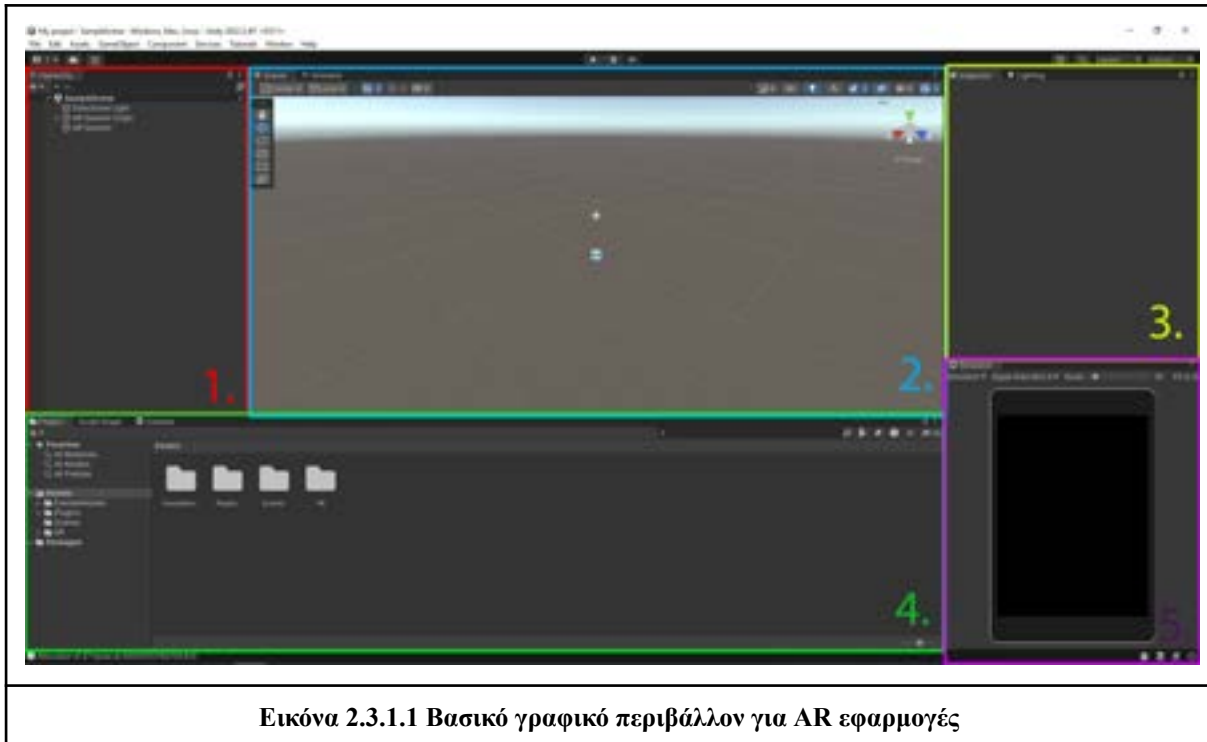
Για τη δημιουργία εφαρμογών Επαυξημένης Πραγματικότητας (AR), συχνά χρησιμοποιείται το λογισμικό Unity, μια πανίσχυρη μηχανή γραφικών που κυκλοφορεί από το 2005. Το Unity 3D αποτελεί βασική επιλογή για προγραμματιστές, παρέχοντας εύκολη πρόσβαση σε εργαλεία ανάπτυξης παιχνιδιών (What Is Unity, 2023). Το εν λόγω λογισμικό εκμεταλλεύεται το AR Foundation, το οποίο ενσωματώνει τα ARCore και ARKit. Αυτά τα εργαλεία αποτελούν βασικά εργαλεία ανάπτυξης εφαρμογών Επαυξημένης Πραγματικότητας (AR), και παρέχουν τη δυνατότητα δημιουργίας εφαρμογών AR για περιβάλλοντα τόσο Android όσο και iOS. Μέσω αυτών των εργαλείων, οι προγραμματιστές μπορούν να προσαρμόσουν το ψηφιακό περιεχόμενο των εφαρμογών τους στον πραγματικό κόσμο, λαμβάνοντας υπόψη τον φωτισμό και τις κάμερες των συσκευών. Αυτό επιτρέπει την δημιουργία πιο ρεαλιστικών και αληθοφανών εμπειριών AR για τους χρήστες (AR Engines — Unity, 2023).

2.3.1 Περιβάλλον Εργασίας

Το γραφικό περιβάλλον του Unity 3D σχεδιάστηκε ώστε να προσφέρει στον προγραμματιστή μια καλά δομημένη και ευέλικτη πλατφόρμα για τη δημιουργία εφαρμογών και παιχνιδιών. Οι δυνατότητες του Unity 3D επιτρέπουν στους δημιουργούς να αναπτύσσουν ποικίλες εφαρμογές, από απλές προσομοιώσεις μέχρι σύνθετα παιχνίδια

και εφαρμογές επαυξημένης πραγματικότητας. Τα βασικά στοιχεία του γραφικού περιβάλλοντος απαρτίζονται από τα παρακάτω παράθυρα (windows):

1. **Hierarchy (Ιεραρχία)**: Αποτελεί τον κατάλογο όλων των GameObject που περιέχει η σκηνή. Αποδεικνύεται ουσιαστικό εργαλείο για την ταξινόμηση και ομαδοποίηση αντικειμένων, διευκολύνοντας έτσι τη διαχείριση του περιεχομένου.
2. **Scene View (Προβολή Σκηνής)**: Προσφέρει οπτικοποίηση των αντικειμένων στον επεξεργαστή, επιτρέποντας επιλογή, χειραγώγηση και τροποποίηση διαφόρων στοιχείων όπως κάμερες, χαρακτήρες, φωτισμοί κ.ά.
3. **Inspector Window (Παράθυρο Επιθεωρητή)**: Εμφανίζει τις ιδιότητες όλων των αντικειμένων στον επεξεργαστή, δίνοντας λεπτομερή έλεγχο και προσαρμογή.
 - **Lighting Window (Παράθυρο Φωτισμού)**: Επιτρέπει τη ρύθμιση του φωτισμού στη σκηνή, ενισχύοντας τη ρεαλιστικότητα των γραφικών.
4. **Project Window (Παράθυρο Έργου)**: Περιλαμβάνει έναν οργανωμένο κατάλογο με όλα τα αρχεία του project, βοηθώντας στον εύκολο εντοπισμό και τη διαχείρισή τους.
 - **Script Graph (Γράφημα Σεναρίου)**: Στέκεται στην καρδιά του περιβάλλοντος προγραμματισμού, διευκολύνοντας τη δήλωση μεταβλητών και τον προγραμματισμό λειτουργιών.
5. **Simulator View (Προβολή Προσομοίωσης)**: Προσφέρει ρεαλιστική προσομοίωση της εφαρμογής σε επιθυμητή συσκευή, με δυνατότητα ρύθμισης ανάλυσης και προσανατολισμού.
6. **GameObject (Αντικείμενο)**: Αντιπροσωπεύει το σύνολο όλων των αντικειμένων της εφαρμογής, περιλαμβάνοντας χαρακτήρες, φώτα, κάμερες και ειδικά εφέ.

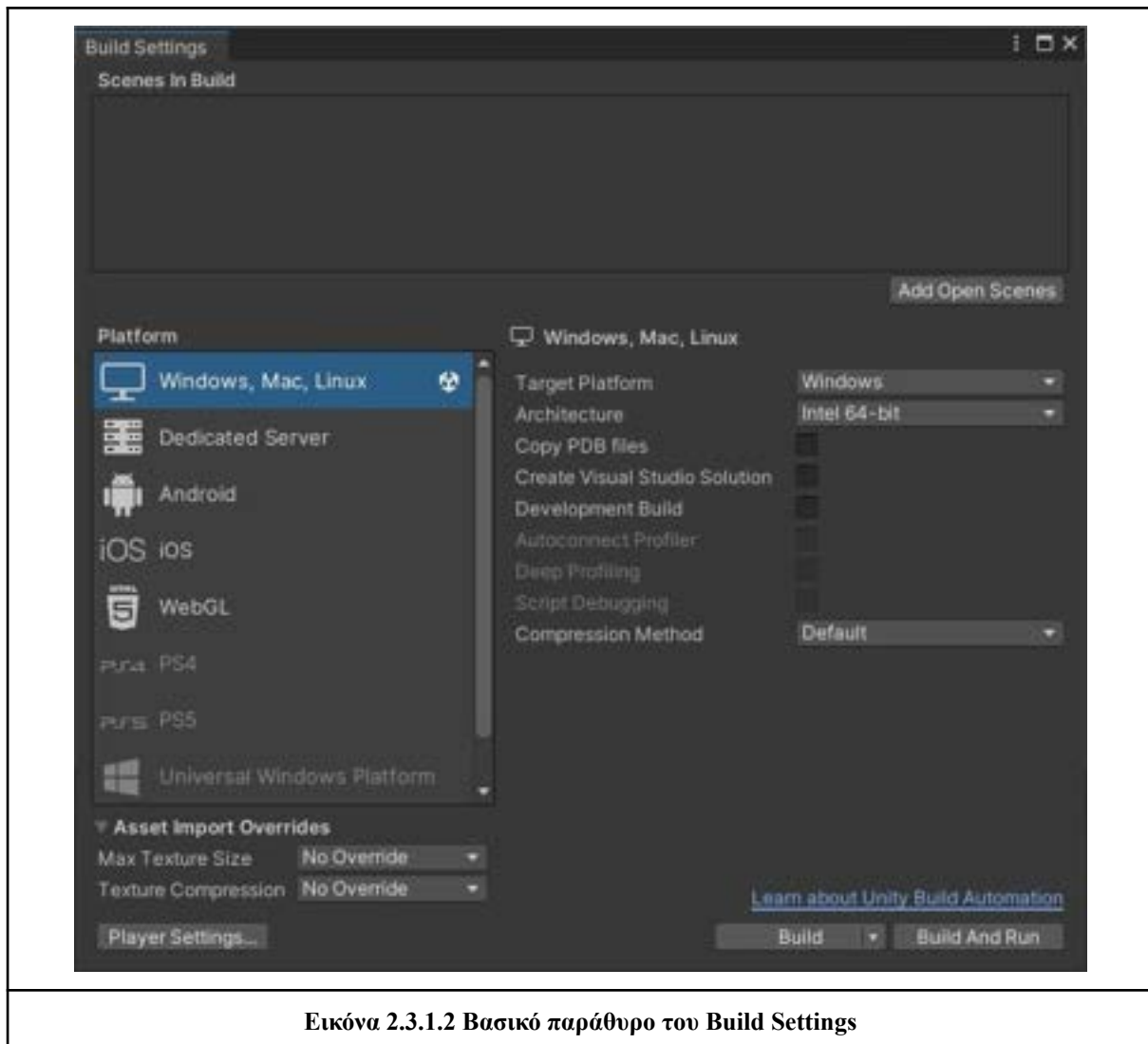


Εικόνα 2.3.1.1 Βασικό γραφικό περιβάλλον για AR εφαρμογές

Ένα ακόμα βασικό στοιχείο του γραφικού περιβάλλοντος του Unity αποτελεί η λειτουργία Build Settings. Το Build Settings είναι προσφέρει τη δυνατότητα επιλογής της πλατφόρμας στην οποία θα αναπτυχθεί η εκάστοτε εφαρμογή. Η επιλογή αυτή καθορίζει τον τρόπο με τον οποίο η εφαρμογή θα λειτουργεί σε διάφορες συσκευές και λειτουργικά συστήματα.

Κατά την ανάπτυξη μιας εφαρμογής, ο προγραμματιστής χρησιμοποιεί το Build Settings για να ορίσει ποια πλατφόρμα θα είναι η στόχος του τελικού build της εφαρμογής. Για παράδειγμα, μπορεί να επιλέξει Android εάν στοχεύει σε συσκευές Android, iOS για συσκευές Apple, ή άλλες πλατφόρμες ανάλογα με τις ανάγκες του έργου.

Η επιλογή της σωστής πλατφόρμας είναι ουσιώδης για τη σωστή λειτουργία της εφαρμογής σε διάφορες συσκευές. Επιπλέον, το Build Settings επιτρέπει τη ρύθμιση διαφόρων παραμέτρων σχετικά με το build, όπως ο βαθμός συμπίεσης, οι ρυθμίσεις φωτισμού, και άλλες σημαντικές παράμετροι που επηρεάζουν την απόδοση και την εμφάνιση της εφαρμογής.



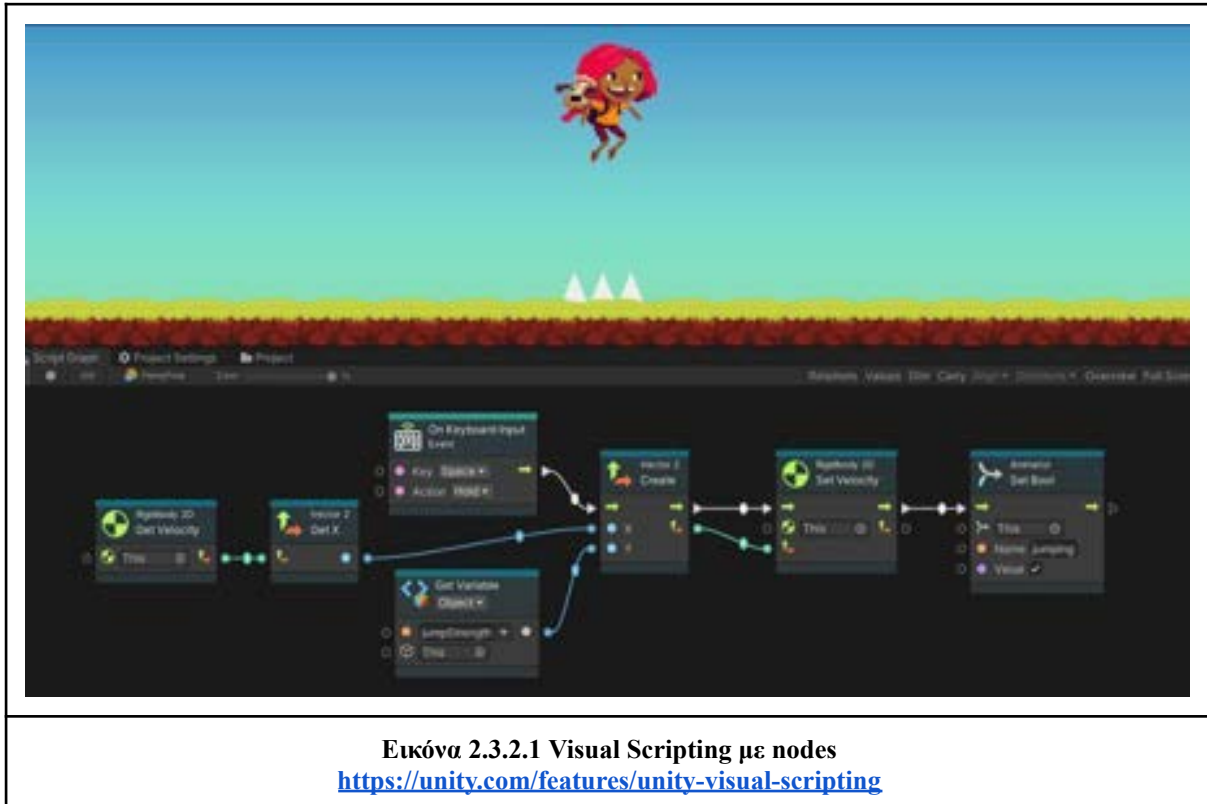
Εικόνα 2.3.1.2 Βασικό παράθυρο του Build Settings

2.3.2 Προγραμματισμός και Ανάπτυξη

Η πλατφόρμα Unity παρέχει δύο διαφορετικούς τρόπους δημιουργίας σεναρίων (scripts): το Visual Scripting και τη χρήση της γλώσσας προγραμματισμού C#. Το Visual Scripting απλοποιεί την διαδικασία δημιουργίας λογικής και σεναρίων για παιχνίδια ή εφαρμογές, εξαλείφοντας την ανάγκη για γραφή κώδικα. Αυτό καθιστά το Visual Scripting ιδανικό για σχεδιαστές χωρίς προηγούμενη εμπειρία στον προγραμματισμό.

Στο Visual Scripting, τα σενάρια δημιουργούνται με τη χρήση κόμβων (nodes), που προσφέρουν μια οπτική προσομοίωση της προγραμματιστικής λογικής. Επιπλέον, χρησιμοποιεί γραφικά στοιχεία που αναπαριστούν συναρτήσεις, τελεστές και μεταβλητές, οι οποίες συνδέονται μεταξύ τους μέσω θυρών που υπάρχουν στους κόμβους. Αυτή η προσέγγιση αντικαθιστά την ανάγκη για γραφή κώδικα σε κάποια γλώσσα προγραμματισμού (Unity Visual Scripting, n.d.).

Για τον προγραμματισμό σε γλώσσα C#, η πλατφόρμα Unity συνεργάζεται με το λογισμικό συγγραφής κώδικα Visual Studio. Αυτό παρέχει πρόσβαση στα C# Scripts, τα οποία συνδέονται αυτόματα με τις εσωτερικές λειτουργίες της Unity μέσω της κλάσης MonoBehaviour. Τα C# Scripts μπορούν να συνδεθούν με ένα GameObject και να ορίσουν τις ενέργειές του, όπως κίνηση, περιστροφή, κ.ά. (Scripts, n.d.).



2.3.3 Γραφικά, Ήχος και Κίνηση

- **Shaders:** Εκτελούν μαθηματικούς υπολογισμούς που καθορίζουν τα χρώματα των pixels που βλέπουμε και πραγματοποιούνται στον επεξεργαστή γραφικών. Στην Unity χρησιμοποιούμε συνήθως τα Shaders με κάποια αντικείμενα(Objects). Η Unity μας δίνει διάφορους τρόπους για να χρησιμοποιήσουμε ένα shader όπως με το Shader Variant που περιέχει μια συλλογή από shaders που μπορούμε να προσθέσουμε στην εφαρμογή μας. Έχει το Shader Asset που είναι ένα αρχείο κειμένου και εφαρμόζει τα shaders μέσω του κώδικα.

- **Visual Effects:** Ειδικά εφέ στην Unity μπορούν να εφαρμοστούν σε κάμερες, GameObjects, φώτα κ.α. Μερικά από τα εφέ είναι (Lanzinger, 2022, #):

- **Particle System:** Είναι ιδανικό για όταν θέλουμε να προσθέσουμε κάποιο εφέ που είναι υγρό(liquid), τα σύννεφα(clouds), κάποιες φωτιές(flames). Δίνει στα εφέ μια κίνηση και δείχνουν πιο αληθοφανή.
- **Trail Renderer:** Δημιουργεί μια διαδρομή που ακολουθεί ένα GameObject κάνοντάς την κίνησή του πιο αληθοφανή.

- **Projector:** Χρησιμοποιείται για την δημιουργία εφέ όπως τις τρύπες που μπορεί να δημιουργηθούν από σφαίρες ή από σκάψιμο του χώματος με κάποιο φτυάρι.
- **Splines:** Δημιουργούν ένα μονοπάτι που μπορούν να τοποθετηθούν τα GameObjects και να δημιουργείται η κίνηση των αντικειμένων σε αυτό.
- **Sound:** Η μηχανή γραφικών Unity υποστηρίζει διαφορετικούς τύπους ήχων όπως ένα κανονικό τραγούδι σε μορφή WAV,MP3 κ.α. που θα παίζει στην σκηνή μας, ένα Sound Effect (SFX) που μπορούμε να το τοποθετήσουμε σε κάποιο αντικείμενο που θα αναπαράγει τον ήχο που θέλουμε.
- **Animation:** Η Unity περιέχει δύο συστήματα animation με το ένα να ονομάζεται Mecanim και περιέχει το αντικείμενο animation(animation component) που θα τοποθετηθεί στο αντικείμενο που θέλουμε να έχει κίνηση και περιέχει δύο παράθυρα(Animation Window, Animator Window) για την ρύθμιση των κινήσεων. Ο δεύτερος τρόπος είναι το Legacy Animation που χρησιμοποιεί ένα animation component.

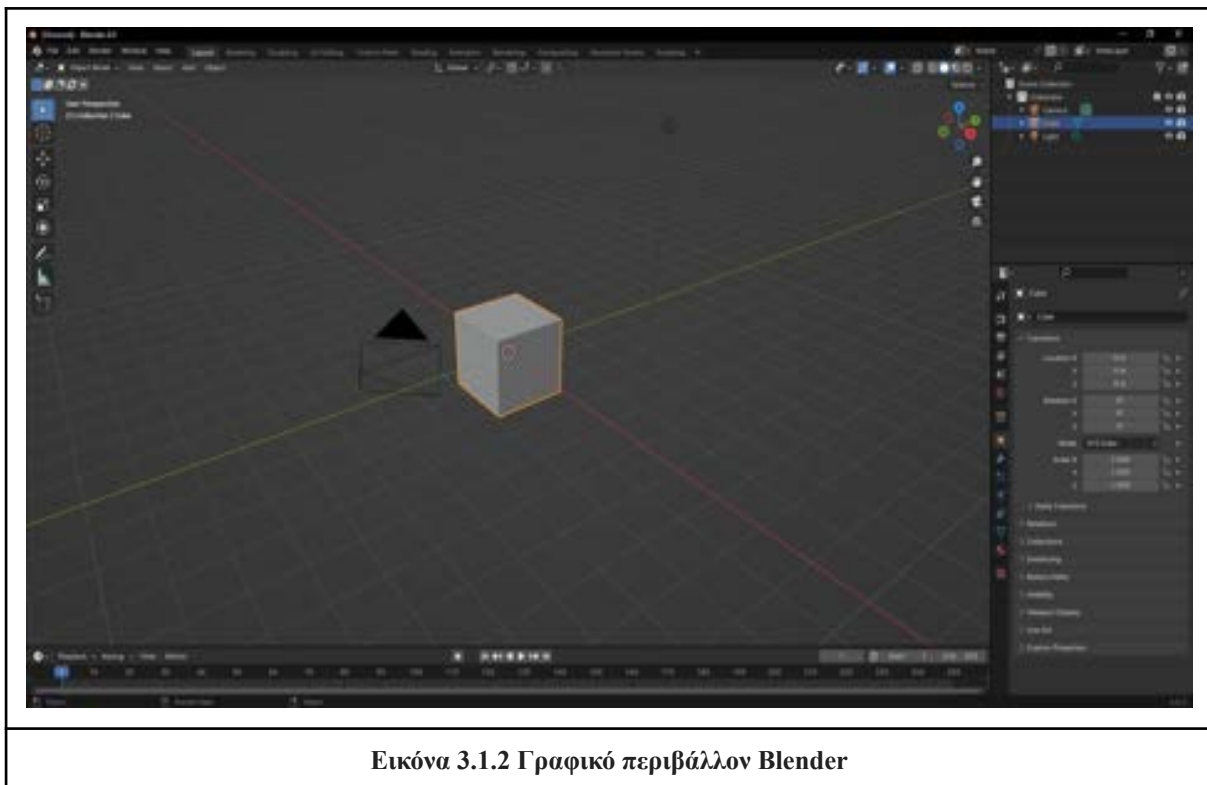
3. Σχεδιασμός και Ανάπτυξη

3.1 Σχεδιασμός 3D Αντικειμένων

3.1.1 Τεχνικές και Εργαλεία

Για την δημιουργία των μοντέλων χρησιμοποιήθηκε η τεχνική του Low Poly Modeling δηλαδή έχουμε σχήματα χαμηλής γεωμετρίας. Επιλέχθηκε επειδή είναι πιο εύκολο και γρήγορο στην υλοποίησή του, είναι πιο ελαφρύ στην επεξεργασία για την υπολογιστική ισχύ και για την εφαρμογή δεν χρειαζόμασταν υψηλής ανάλυσης λεπτομέρειας. Η τεχνική που χρησιμοποιήθηκε για την ενσωμάτωση της εικόνας των ψαριών στα μοντέλα είναι με το UV Mapping και επιλέχθηκε λόγω της ευκολίας για την υλοποίησή του. Το στάδιο του Animation έγινε μέσω Armature και συγκεκριμένα με Bones για να φτιαχτεί ο σκελετός του ψαριού και να ελέγχει το κάθε Bone ποιο σημείο θα κινείται.

Για την τρισδιάστατη (3D) μοντελοποίηση, επιλέχθηκε το λογισμικό Blender. Η απόφαση αυτή βασίστηκε εκτός από την πληθώρα δυνατοτήτων που προσφέρει στον προγραμματιστή, και στην ευκολία μάθησης του προγράμματος σε σύγκριση με άλλα διαθέσιμα λογισμικά. Το Blender διακρίνεται για τον πλούτο των διαθέσιμων μαθημάτων (tutorials) που προσφέρει, καθιστώντας προσιτή την εκμάθησή του. Επιπλέον, ενεργά φόρουμ συμμετοχής αποτελούν πολύτιμη πηγή πληροφοριών για την κατασκευή 3D μοντέλων.



Εικόνα 3.1.2 Γραφικό περιβάλλον Blender





3.1.2 Διαδικασία Σχεδιασμού

Η διαδικασία σχεδιασμού περιλάμβανε τη συλλογή εικόνων τεσσάρων διαφορετικών ειδών ψαριών, χρησιμοποιώντας τις ως παραδείγματα για την ανάπτυξη των 3D μοντέλων. Στη συνέχεια, χρησιμοποιώντας τις εικόνες ως αναφορά, ακολούθησε η διαδικασία δημιουργίας 3D μοντέλων των ψαριών, δίνοντας σημασία στις λεπτομέρειες, τα χρώματα και τη δομή ώστε να παραμείνει πιστή η απεικόνιση του πραγματικού ψαριού. Έπειτα, εφαρμόστηκε η τεχνική του UV mapping για τη σωστή αντιστοίχιση των 2D εικόνων στις 3D επιφάνειες των μοντέλων, επιτυγχάνοντας ομοιόμορφη και ρεαλιστική όψη. Τέλος, ενσωματώθηκε η τεχνική του Animation στα 3D μοντέλα, προσδίδοντας ζωντάνια και φυσικότητα στην απεικόνιση των ψαριών. Με αυτήν τη διαδικασία, δημιουργήσαμε ρεαλιστικά 3D μοντέλα ψαριών, με σκοπό την ενσωμάτωσή τους στην εφαρμογή επαυξημένης πραγματικότητας.

Στάδιο συλλογής υλικού

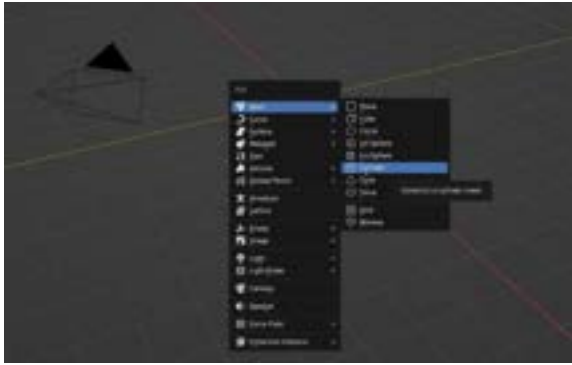
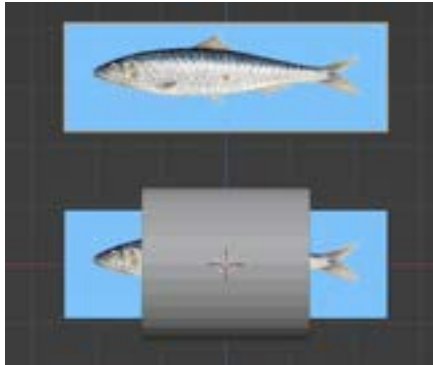
Στο πρώτο στάδιο της διαδικασίας σχεδιασμού, πραγματοποιήσαμε το βήμα της συλλογής εικόνων από τέσσερα διαφορετικά είδη ψαριών. Κατά τη διάρκεια αυτής της αναζήτησης, καταγράψαμε ένα εκτενές υλικό εικονογράφησης με ποικιλία γωνιών λήψης. Η επιλογή

των εικόνων πραγματοποιήθηκε με γνώμονα την υψηλή ποιότητα και τις λεπτομέρειες που κατέχει κάθε μία από αυτές.



	
<p>Εικόνα 3.1.2.1 Τσιπούρα https://allabout.fish/tsipoura/</p>	<p>Εικόνα 3.1.2.2 Σαρδέλα https://www.psaria.eu/2015/07/blog-post_41.html</p>
	
<p>Εικόνα 3.1.2.3 Κέφαλος https://www.aliavnra.gr/kefalos-amvrakikou/</p>	<p>Εικόνα 3.1.2.4 Κουτσομούρα https://www.hellasfishing.gr/%CE%BA%CE%B F%CF%85%CF%84%CF%83%CE%BF%CE %BC%CE%BF%CF%8D%CF%81%CE%B1/</p>

Στάδιο μοντελοποίησης

Στο στάδιο του modeling, χρησιμοποιήθηκε το λογισμικό Blender για τη δημιουργία του 3D μοντέλου του ψαριού. Η διαδικασία περιλάμβανε πρώτον την προσθήκη ενός Mesh τύπου Cylinder, το οποίο αντιπροσωπεύει ένα πλέγμα σχημάτων που είναι προσαρμόσιμο σε διάφορα σχήματα. Στη συνέχεια, ενσωματώθηκε η εικόνα του επιθυμητού μοντέλου ως υπόβαθρο, δημιουργώντας ένα αντίγραφο ακριβώς από πάνω του.

	
<p align="center">Εικόνα 3.1.2.5</p> <p align="center">Shift + A → Mesh → Cylinder Δημιουργία κολίνδρου στο Blender.</p>	<p align="center">Εικόνα 3.1.2.6</p> <p align="center">Προσθήκη εικόνας ψαριού που θα δημιουργηθεί.</p>

Χρησιμοποιώντας τα εργαλεία μετακίνησης (G), περιστροφής (R) και εξώθησης (E), προσαρμόστηκε το Mesh τύπου Cylinder στα χαρακτηριστικά του επιλεγμένου μοντέλου. Στη συνέχεια, δημιουργήθηκαν τα πτερύγια χρησιμοποιώντας ένα επίπεδο (Plane), αντί για κύλινδρο, προσδίδοντας ελαφριές και ρεαλιστικές μορφές.

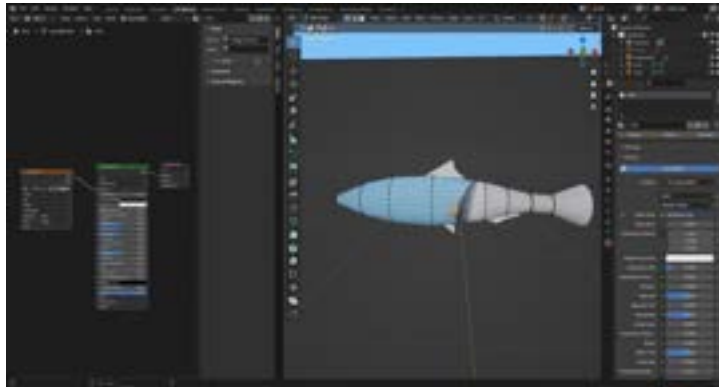
	
<p align="center">Εικόνα 3.1.2.7</p> <p align="center">Προσθήκη επιπέδου(Plane) για την δημιουργία των πτερυγίων</p>	<p align="center">Εικόνα 3.1.2.8</p> <p align="center">Επεξεργασία του επιπέδου(plane) για να ταιριάζει στο μοντέλο</p>

Τέλος, προστέθηκε ένας τροποποιητής (Modifier) με την ονομασία Solidify, με σκοπό να καταστήσει πιο ομαλές τις γωνίες του μοντέλου και να προσθέσει όγκο. Με αυτά τα

βήματα, δημιουργήθηκε ένα βασικό 3D μοντέλο του ψαριού, έτοιμο για περαιτέρω επεξεργασία και προσθήκη λεπτομερειών.

Στάδιο UV Mapping

Στο στάδιο του UV Mapping, πραγματοποιήθηκαν τα παρακάτω βήματα για τη δημιουργία και εφαρμογή του υλικού (Material) και τον χαρτογραφικό προσδιορισμό UV στο 3D μοντέλο του ψαριού. Αρχικά, δημιουργήθηκε το Material που θα χρησιμοποιηθεί για την απεικόνιση των επιφανειών του μοντέλου.



Εικόνα 3.1.2.9

Δημιουργία Material για το μοντέλο. Στα αριστερά γίνεται η δημιουργία του υλικού(Material) στο οποίο βάζουμε την εικόνα που θέλουμε να απεικονίζει το μοντέλο που έχουμε δημιουργήσει στα δεξιά

Στη συνέχεια, στην καρτέλα UV Editing, το Material τοποθετήθηκε πάνω στο μοντέλο, ενώ προηγουμένως το μοντέλο "ξεδιπλώθηκε" (unwrapped) για τη σωστή γεωμετρία πριν την εφαρμογή του UV. Προκειμένου να εφαρμοστούν οι αλλαγές σε ολόκληρο το μοντέλο, προστέθηκε ο τροποποιητής (Modifier) Mirror, εξασφαλίζοντας τη συμμετρία σε οποιαδήποτε αλλαγή και στις δύο πλευρές του μοντέλου.



Εικόνα 3.1.2.10

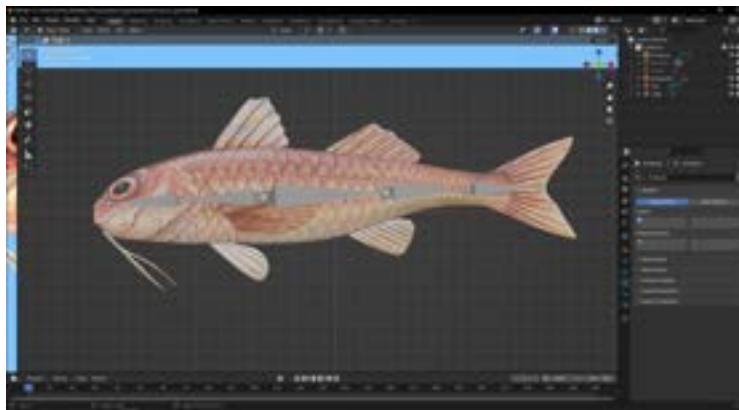
Προσθήκη Modifier Mirror στα UV του μοντέλου.

Τέλος, μετά την ολοκλήρωση του μοντέλου και του UV Mapping, επιλέχθηκαν όλα τα πτερύγια και το ψάρι, και με τον συνδυασμό Ctrl + J πραγματοποιήθηκε η ένωσή τους, δημιουργώντας ένα ενιαίο μοντέλο που περιλαμβάνει όλα τα στοιχεία. Αυτή η διαδικασία

εφαρμόστηκε αντίστοιχα και σε όλα τα τέσσερα είδη ψαριών, ολοκληρώνοντας έτσι τη διαδικασία του UV Mapping για τα μοντέλα ψαριών.

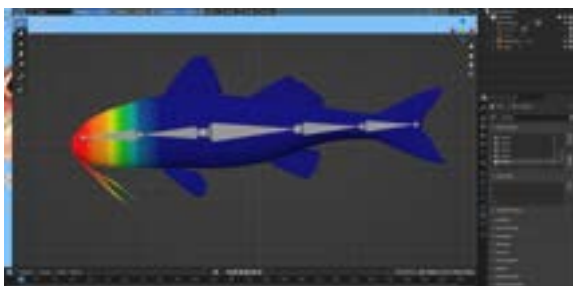
Στάδιο Animation

Στο τελικό στάδιο του Animation, προχωρήσαμε στα εξής βήματα για τα 3D μοντέλα των ψαριών. Αρχικά, με τη χρήση του εργαλείου Armature, προστέθηκαν κόκαλα (Bones) στα μοντέλα, τα οποία τοποθετήθηκαν στις κατάλληλες θέσεις. Ακολούθως, πραγματοποιήθηκε η ενοποίηση των κοκάλων με τα μοντέλα, εξασφαλίζοντας έτσι ότι τα μοντέλα θα ακολουθούν τις κινήσεις που θα καθορίσουμε.

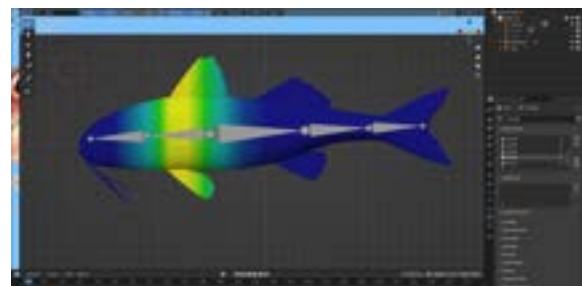


Εικόνα 3.1.2.11
Προσθήκη και ενοποίηση (με αυτόματα Βάρη) κόκκαλων(Bones) στο μοντέλο

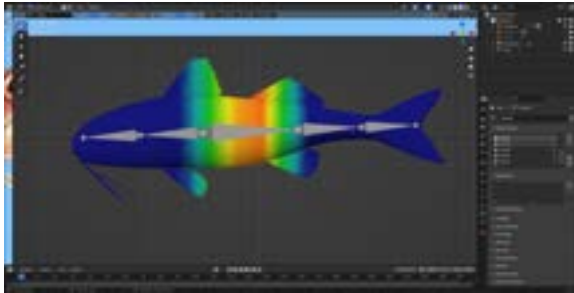
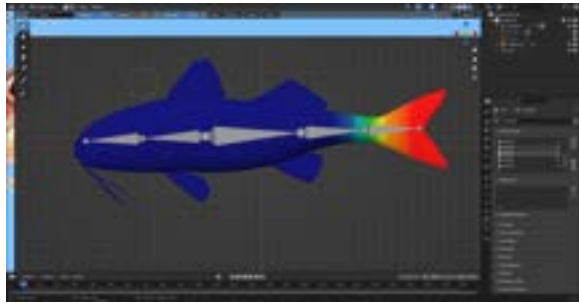
Στη συνέχεια, προχωρήσαμε στη ρύθμιση κάθε κόκαλου για να ελέγχει μια συγκεκριμένη περιοχή του μοντέλου. Για παράδειγμα, ένα κόκαλο μπορεί να είναι υπεύθυνο για την κίνηση του κεφαλιού, ενώ άλλα κόκαλα μπορεί να ελέγχουν διάφορα τμήματα του σώματος. Αυτή η διαδικασία επιτρέπει την ακριβή προγραμματισμό των κινήσεων, προσδίδοντας ρεαλιστική κίνηση στα ψάρια σε κάθε περιοχή που επιθυμούμε.



Εικόνα 3.1.2.12
Ρύθμιση κόκκαλου για την περιοχή του κεφαλιού του μοντέλου.



Εικόνα 3.1.2.13
Ρύθμιση κόκκαλου για την περιοχή της άνω κοιλιάς του μοντέλου.

	
<p style="text-align: center;">Εικόνα 3.1.2.14</p> <p style="text-align: center;">Ρύθμιση κόκκαλου για την περιοχή της κάτω κοιλιάς του μοντέλου.</p>	<p style="text-align: center;">Εικόνα 3.1.2.15</p> <p style="text-align: center;">Ρύθμιση κόκκαλου για την περιοχή της ουράς του μοντέλου.</p>

3.2 Σχεδίαση και Ανάπτυξη AR εφαρμογής

Για την ανάπτυξη της AR εφαρμογής χρησιμοποιήθηκε η μηχανή γραφικών Unity που μέσω των δυνατοτήτων της έχουμε έτοιμα εργαλεία για την ανάπτυξη της. Η εφαρμογή αναπτύχθηκε για την ανάδειξη θαλάσσιου περιβάλλοντος και την συλλογή πληροφοριών για τέσσερα είδη ψαριών (τσιπούρα, σαρδέλα, κουτσομούρα, κέφαλος). Η εφαρμογή περιέχει ένα αρχικό μενού καλωσορίσματος και την σκηνή με τοποθετημένα ψάρια που μπορεί ο χρήστης να αλληλεπιδράσει μαζί τους για να πάρει πληροφορίες για αυτά.

3.2.1 Τεχνικές και Εργαλεία

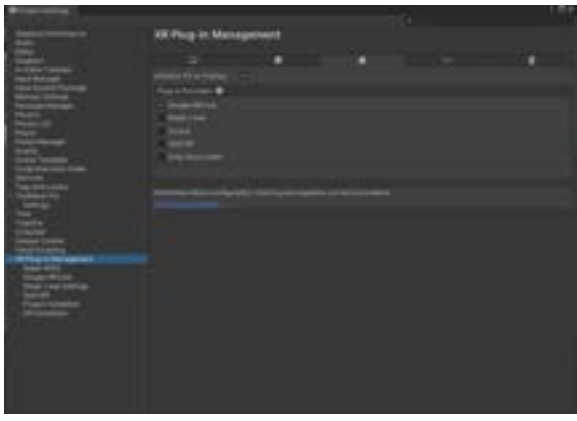
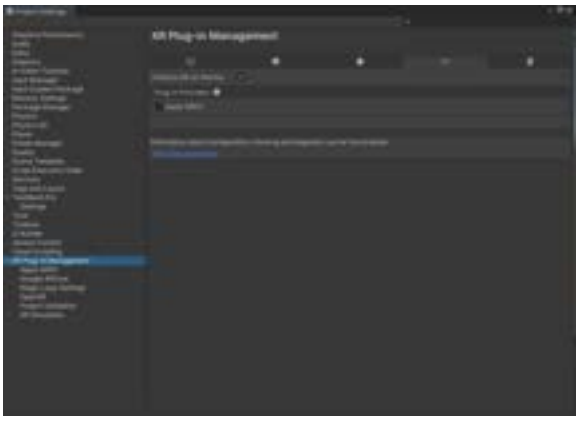
Τα εργαλεία που χρησιμοποιήθηκαν στην Unity είναι:

- **Animator Controller:** Για να μπορούν τα ψάρια να έχουν την κίνηση που τους δώσαμε μέσω του Blender με το armature τους.
- **Spline:** Δίνουμε μια κυκλική κίνηση στα ψάρια μέσα στην σκηνή.
- **C# Scripts:** Έγινε ο προγραμματισμός της AR σκηνής και τις λειτουργίες της.
- **Visual Scripts:** Έγινε ο προγραμματισμός των UI(μενού διεπαφής)
- **UI:** Δημιουργία ενός Canvas που περιέχει Text καλωσορίσματος, Button για την αλλαγή στην AR σκηνή και εικόνες για το Background και το Logo.
- **Special Effects:** Δημιουργία ενός Particle System Component για τις φυσαλίδες που υπάρχουν μέσα στην σκηνή.
- **Post Process:** Εισαγωγή ενός Box Collider για το χρώμα που έχει η AR σκηνή.

Για την δημιουργία της AR εφαρμογής χρησιμοποιήθηκε η μηχανή γραφικών Unity και συγκεκριμένα η έκδοση 2022.3.4f1

3.2.2 Ρυθμίσεις AR στο Unity

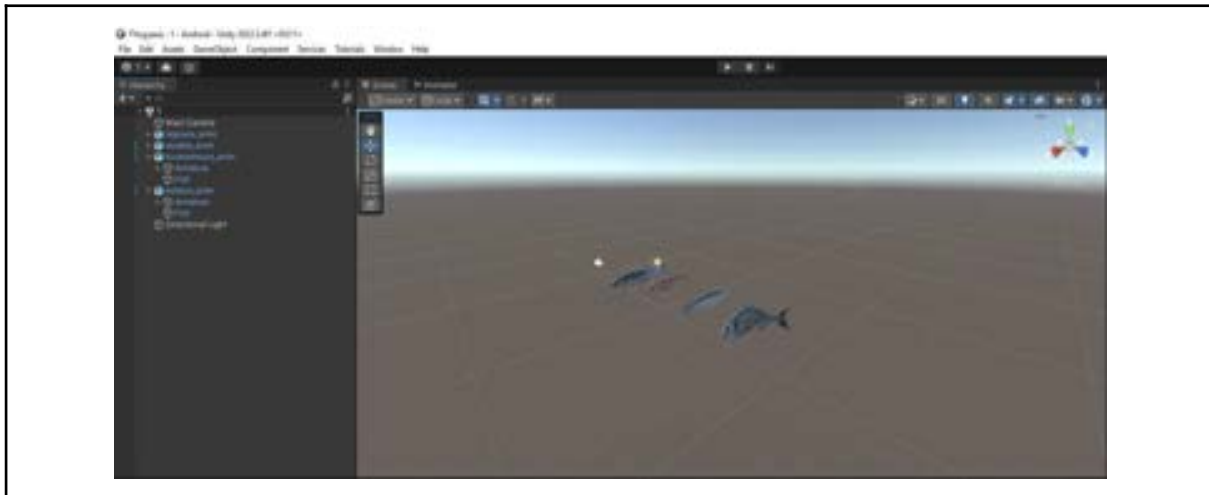
Η προσθήκη των απαραίτητων ρυθμίσεων για λειτουργίες Επαυξημένη Πραγματικότητα (AR) στο Unity, πραγματοποιείται μέσα από τις ρυθμίσεις της πλατφόρμας, ανοίγοντας το μενού "Edit" στην κορυφή του Unity και επιλέγοντας "Project Settings", στην ενότητα "Player" επιλέγει την υποκατηγορία που σχετίζεται με τη συγκεκριμένη πλατφόρμα (π.χ., Android ή iOS). Στη συνέχεια, εντός της υποκατηγορίας "XR Plugin Management", βρίσκει τις ρυθμίσεις που σχετίζονται με την Επαυξημένη Πραγματικότητα (AR) ή AR Foundation. Εκεί, μπορεί να ενεργοποιήσει και να ρυθμίσει τα αντίστοιχα πακέτα.

	
<p>Εικόνα 3.2.2.1. Προσθήκη Google AR Core</p>	<p>Εικόνα 3.2.2.2. Προσθήκη Apple ARKit</p>

3.2.3 Ενσωμάτωση 3D Αντικειμένων και Animation

Ενσωμάτωση 3D αντικειμένων στο Unity

Τα μοντέλα των ψαριών έγιναν από το Blender export στην μορφή .fbx, αυτό έγινε επειδή η επέκταση .fbx κρατάει στο αρχείο την γεωμετρία του μοντέλου, το material που του έχουμε δώσει, τον φωτισμό που μπορεί να έχει η σκηνή όπως και το animation του. Για να γίνει το export στο Blender ανοίγουμε το μενού "File", στην ενότητα "Export" επιλέγουμε FBX(.fbx). Τα αρχεία των ψαριών είναι έτοιμα για χρήση στην μηχανή γραφικών που χρησιμοποιούμε (Unity).

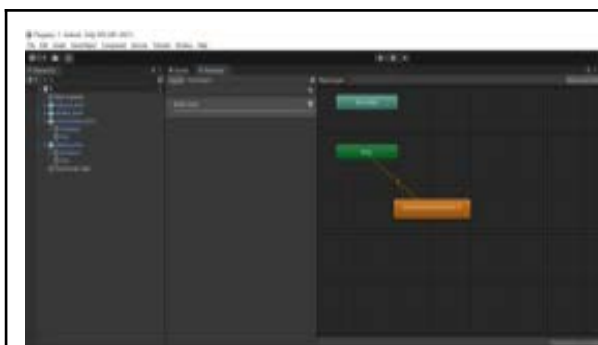


Εικόνα 3.2.3.1

Ενσωμάτωση μοντέλων σε σκηνή του Unity.

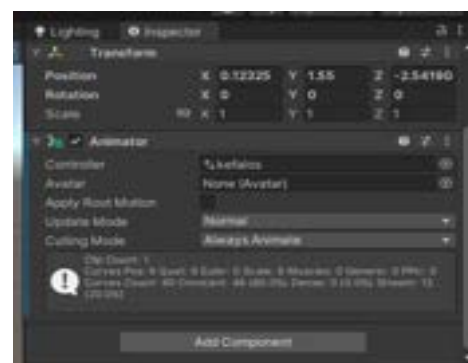
Ενσωμάτωση Animation

Για τη διαδικασία ενσωμάτωσης animation στα μοντέλα των ψαριών, δημιουργείται ένας Animator Controller (Create → Animator Controller) για κάθε μοντέλο στο οποίο θα εφαρμοστεί το Armature. Το Armature περιέχει το animation που έχει δημιουργηθεί στο Blender. Στη συνέχεια, οι Animator Controllers προστέθηκαν στα μοντέλα που τοποθετήσαμε στη σκηνή.



Εικόνα 3.2.3.2.

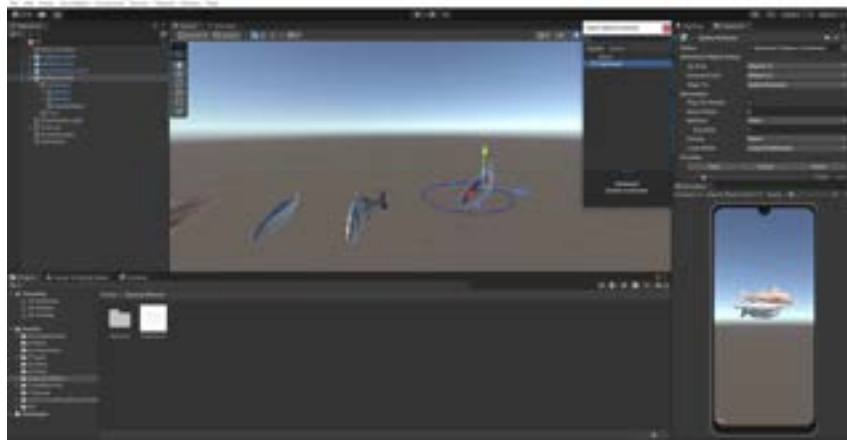
Σύνδεση του Animator Controller με το Armature.



Εικόνα 3.2.3.3

Προσθήκη του Animator στα μοντέλα.

Τέλος, ακολουθήθηκε η διαδικασία ενεργοποίησης του πακέτου (asset) Spline μέσω του Package Manager στο Unity. Το πακέτο Spline μας παρέχει τη δυνατότητα να χρησιμοποιήσουμε το Spline Animate για να ορίσουμε μια κίνηση στο μοντέλο που επιθυμούμε. Αυτή η λειτουργία επιτρέπει την ακριβή και ευέλικτη κίνηση των μοντέλων στον χώρο.



Εικόνα 3.2.3.4

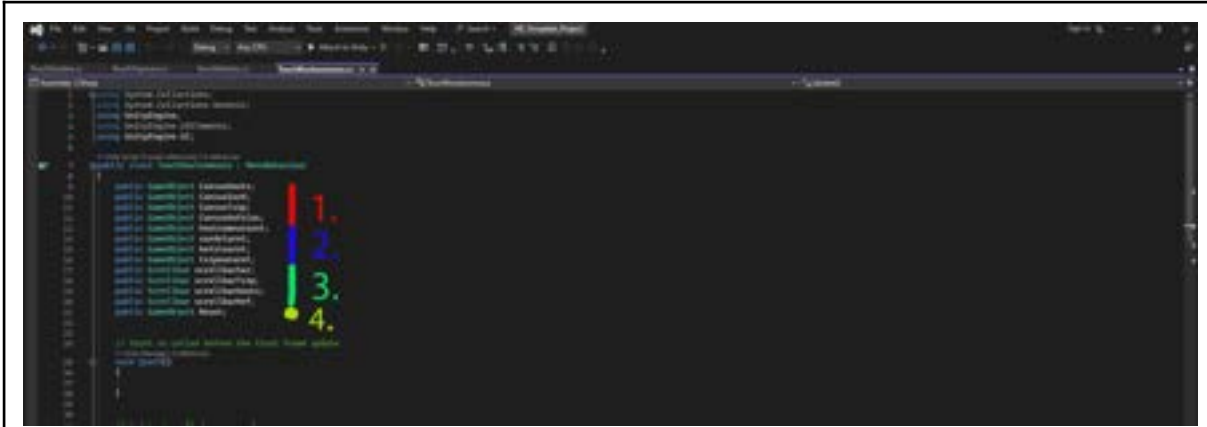
Σύνδεση Spline με το μοντέλο

3.2.4 Προγραμματισμός Σεναρίων (Scripts) με C#

Τα C# Scripts είναι κομμάτια κώδικα που χρησιμοποιούνται στις εφαρμογές για να δώσουν κάποια λειτουργικότητα. Δημιουργούνται σε λογισμικό όπως το Visual Studio που έχει την δυνατότητα για υλοποίηση κώδικα. Μέσα από την Unity στο παράθυρο Project μπορούμε πατώντας δεξί κλικ και στην καρτέλα “Create” να δημιουργήσουμε ένα C# Script. Στο παράθυρο Hierarchy βρίσκουμε το GameObject που θέλουμε να πάρει κάποιο script και στο παράθυρο Inspector πατάμε το κουμπί “Add Component” για να προσθέσουμε το C# Script που έχουμε δημιουργήσει.

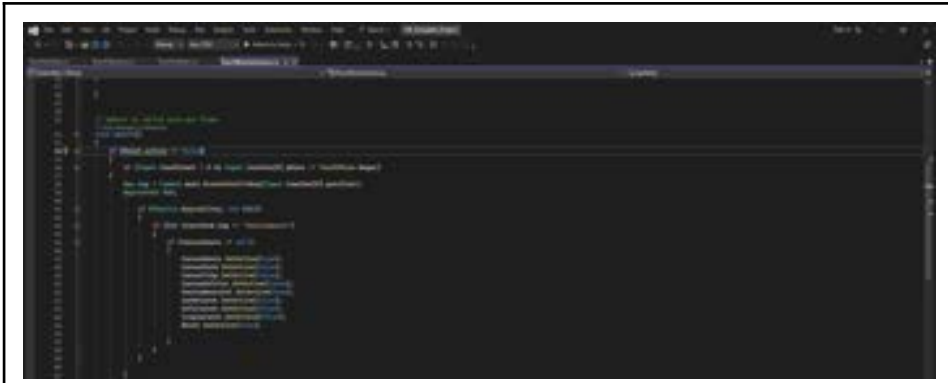
Για την εφαρμογή δημιουργήθηκαν τέσσερα διαφορετικά C# Scripts που συνδέθηκαν με το αντίστοιχο μοντέλο ψαριού. Προγραμματίστηκε έτσι ώστε να διαβάζει μέσω του Ray το άγγιγμα(touch) του χρήστη στην οθόνη της συσκευής, να καταλαβαίνει ποιο ψάρι έχει

πατήσει ο χρήστης και να ανοίγει τον αντίστοιχο Canva με τις πληροφορίες για το ψάρι που έχει πατήσει ο χρήστης. Βλέπουμε ένα από τα τέσσερα Script που δημιουργήθηκαν, συγκεκριμένα το Touch Koutsomoura που προγραμματίστηκε για τα μοντέλα Koutsomoura και εκτελούν λειτουργίες μόνο για αυτά.



Εικόνα 3.2.4.1 Δήλωση μεταβλητών σε C# Script

1. Δηλώνουμε τους Canvas του κάθε ψαριού που θα περιέχουν τις πληροφορίες τους.(CanvasKouts, CanvasSard, CanvasTsip, CanvasKefalos).
2. Δηλώνουμε ένα μοντέλο για κάθε ψάρι το οποίο θα ανοίγει μαζί με τον Canva του ψαριού που έχει πατηθεί και θα βρίσκεται από πάνω και θα περιστρέφεται με το που ο χρήστης θα πατάει στο αντίστοιχο ψάρι.(koutsomourarot, sardelarot, kefalosrot, tsipourarot).
3. Δηλώνουμε Scrollbars τα οποία όποτε κλείνει ένα Canvas και ξανα ανοίγει τότε να δείχνει την πάνω πάνω πληροφορία(scrollbarSar, scrollbarTsip, scrollbarKouts, scrollbarKef).
4. Δηλώνουμε ένα GameObject με όνομα Reset το οποίο έχει τοποθετηθεί στην σκηνή και αποτρέπει τον χρήστη να πατάει σε κάποιο ψάρι ενώ υπάρχει ήδη κάποιο ανοιχτό. Έτσι η μόνη ενέργεια που μπορεί να κάνει είναι να κλείσει τον Canva με το κουμπί X.(Reset).



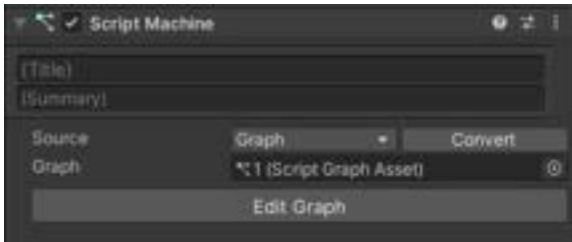
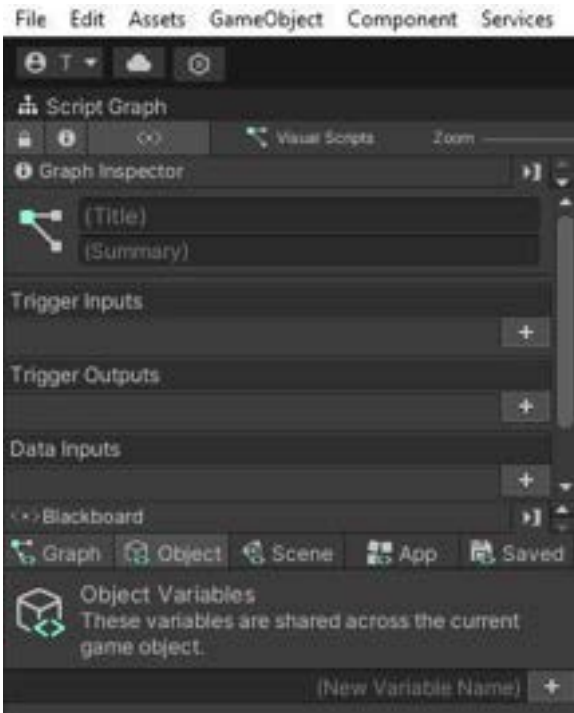
Εικόνα 3.2.4.2 Ο κώδικας που εκτελείται όταν ο χρήστης πατήσει σε ένα ψάρι

Ελέγχει αν το Reset είναι ανενεργό τότε: ο τρόπος για να καταλάβει η Unity ότι ο χρήστης έχει πατήσει στην οθόνη της συσκευής κάποιο μοντέλο είναι μέσω του RaycastHit, οπότε έχουμε ορίσει κάποια Tags στα μοντέλα μας για να μπορεί να καταλάβει ποιο από τα τέσσερα ψάρια πατάει. Έπειτα ελέγχει αν ο Canvas δεν είναι ενεργός τότε ανοίγει(true) τον canva του ψαριού που πατήσαμε και κλείνει(false) τους υπόλοιπους. Το ίδιο και με τα Scrollbar. Τέλος ενεργοποιεί το Reset για να αποτρέπει τον χρήστη να πατήσει κάποιο άλλο ψάρι.

Ο ίδιος κώδικα έγινε και για τα τέσσερα ψάρια με τις μόνες διαφορές να είναι τα Tags και ποιοι Canvas και ποια Scrollbar θα γίνονται True/False.

Visual Scripts

Σε ένα κενό(empty) GameObject που θα το ονομάσουμε Visual Scripts προσθέτουμε ένα Component Script Machine. Μέσα στο Script Machine και συγκεκριμένα στο Blackboard δηλώνουμε μεταβλητές που θα χρησιμοποιήσουμε.

	
<p>Εικόνα 3.2.4.3 Ένα Script Machine Component</p>	<p>Εικόνα 3.2.4.4 Blackboard που δηλώνουμε τις μεταβλητές.</p>

Στα Visual Scripts προγραμματίστηκε το κλείσιμο των Canvas με κουμπιά(Κλείσιμο) όπως και η αλλαγή σκηνής δηλαδή από το ARScene στο αρχικό μενού(MainScene).


<p>Εικόνα 3.2.4.5 Κώδικας που εκτελείται όταν πατηθεί το κουμπί Κλείσιμο</p>

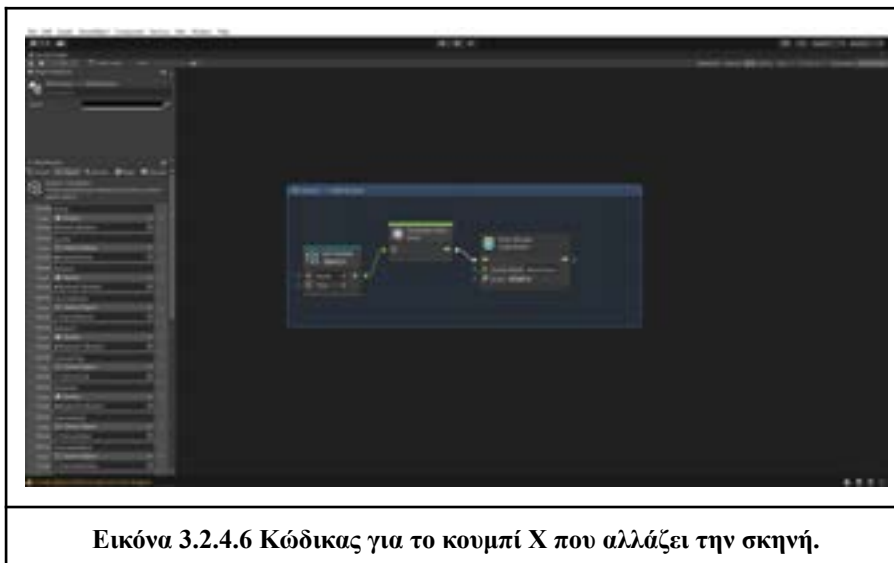
Οι μεταβλητές που χρησιμοποιούνται είναι:

- ButtonX
- ScrollbarKouts

- CanvasKouts
- koutsomourarot
- Reset

Όταν πατήσει(On Button Click) το κουμπί Κλείσιμο τότε κάνει set το ScrollbarKouts στην αρχική τιμή του, κλείνει τον CanvasKouts που ήταν ανοιχτός, κλείνει το μοντέλο(koutsomourarot) που έκανε περιστροφή πάνω από τον Canva και κλείνει το Reset για να μπορεί ο χρήστης να πατήσει σε κάποιο άλλο ψάρι.

Ο ίδιος κώδικας έγινε για το κάθε ψάρι ξεχωριστά με μόνη διαφορά τις μεταβλητές.



Εικόνα 3.2.4.6 Κώδικας για το κουμπί X που αλλάζει την σκηνή.

Με το που πατήσει το κουμπί X, τότε επιστρέφει στην αρχική σκηνή(MainScene).

3.2.5 Σχεδιασμός UI

Για την δημιουργία του 2D UI(User Interface) στην κύρια σκηνή χρησιμοποιήθηκε ένα Canvas που περιέχει:

- Μια εικόνα που τοποθετήθηκε ως Background
- Μια εικόνα που μπήκε με το λογότυπο της εφαρμογής
- Text που εξηγεί σε μια πρόταση τι βλέπουμε στην εφαρμογή
- Text που καλωσορίζει τους χρήστες
- Κουμπί που όταν πατηθεί αλλάζει στην AR σκηνή

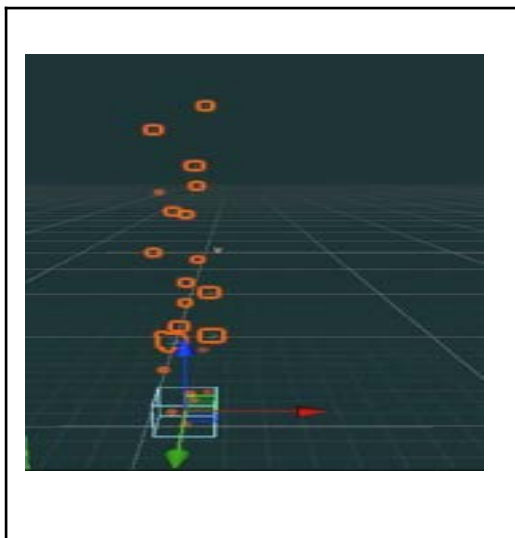
Στην AR σκηνή:

- Χρησιμοποιήθηκε ένα Text που έχει τις ονομασίες των ψαριών και τοποθετήθηκε πάνω από τα μοντέλα.
- Ένα Panel που περιέχει Text με τις πληροφορίες του ψαριού επίσης έχει ένα Button για το κλείσιμο του Panel και την επιστροφή στην AR σκηνή.

3.2.6 Ενσωμάτωση Ειδικών Εφέ (Special effects)

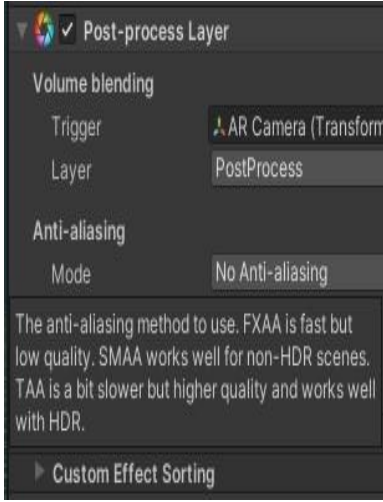

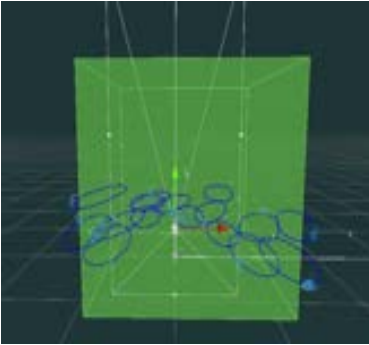
Για να βελτιωθεί η εμπειρία του χρήστη στην εφαρμογή, προστέθηκαν ειδικά εφέ που περιλαμβάνουν φουσαλίδες και ένα background βυθού. Το εφέ με τις φουσαλίδες σχεδιάστηκε με στόχο να προσδώσει μια gamification πτυχή στην εφαρμογή, ενθαρρύνοντας τον χρήστη να αλληλοεπιδράσει με το περιβάλλον Επαυξημένης Πραγματικότητας (AR). Αντίστοιχα, το εφέ με το background του βυθού προσφέρει στον χρήστη την αίσθηση ότι βρίσκεται πραγματικά μέσα στον βυθό, ενισχύοντας την αναπαράσταση του εικονικού περιβάλλοντος και προσδίδοντας ρεαλιστικότητα στην εφαρμογή.

Για το εφέ με τις φουσαλίδες, δημιουργούμε ένα κενό GameObject στο οποίο προσθέτουμε ένα Component Particle System. Αυτό το Component μας επιτρέπει να δημιουργήσουμε το εφέ των φουσαλίδων.



Εικόνα 3.2.6.1 Το GameObject που του αναθέσαμε τα Bubbles

Για το εφέ του μπλε βυθού, χρησιμοποιήθηκε η τεχνική Post Process. Αρχικά, προσθέσαμε σε ένα κενό GameObject ένα Component "Post Process Volume", μέσω του οποίου καθορίζουμε το χρώμα της σκηνής. Επιπλέον, προσθέσαμε ένα ακόμα Component "Box Collider", το οποίο ορίζεται σε όλη τη σκηνή. Τέλος, προσθέσαμε ένα Component "Post Process Layer" στο GameObject της κάμερας (AR Camera) και το συνδέσαμε με το GameObject που δημιουργήσαμε προηγουμένως.

		
<p>Εικόνα 3.2.6.2</p> <p>Component Post process Layer</p>	<p>Εικόνα 3.2.6.3</p> <p>Component Post process volume</p>	<p>Εικόνα 3.2.6.4</p> <p>Box Collider</p>

4. Παρουσίαση AR Εφαρμογής



Στην εικόνα 5.1 έχουμε το αρχικό μενού της εφαρμογής που μαζί με το λογότυπο της έχουμε ένα μήνυμα καλωσορίσματος και το κουμπί “έναρξη” που θα μας μεταφέρει στην AR σκηνή. Έπειτα στην εικόνα 5.2 έχουμε το αρχικό στάδιο της AR σκηνής που δουλεύει με την κάμερα της συσκευής που την έχουμε ανοίξει. Στην σκηνή έχουμε τα μοντέλα των ψαριών που έχουμε δημιουργήσει μαζί με κάποια εφέ. Τέλος όταν ο χρήστης πατήσει σε κάποιο από τα ψάρια θα βγει το μενού της εικόνας 5.3 που θα έχει το μοντέλο να περιστρέφεται πάνω από τις πληροφορίες του ψαριού που έχει πατήσει. Με το κουμπί “Κλείσιμο” κλείνει ο καμβάς με τις πληροφορίες του ψαριού και με το κουμπί X επιστρέφει στο αρχικό μενού.

5. Συμπεράσματα και μελλοντικές επεκτάσεις

Έπειτα από την ολοκλήρωση της εφαρμογής κατέληξα σε κάποια συμπεράσματα σχετικά με τα 3D μοντέλα και την ανάπτυξη εφαρμογής. Για έναν αρχάριο σαν εμένα ήταν αρκετά χρονοβόρο και δύσκολο στην σχεδίαση και την υλοποίηση είτε των 3D μοντέλων είτε στην ανάπτυξη της εφαρμογής χωρίς αυτό να σημαίνει ότι δεν ευχαριστήθηκα την όλη διαδικασία. Ευτυχώς υπάρχει μεγάλος αριθμός μαθημάτων(courses, tutorials) στο διαδίκτυο για την εκμάθησή τους.

Όσον αφορά τα 3D κατάλαβα ότι μπορείς να δημιουργήσεις ότι φανταστείς από μοντέλα ζωντανών οργανισμών όπως ψάρια, ανθρώπους κτλ., σε ολόκληρα κτίρια, διάφορα πάρκα κ.α. Μπορείς να δημιουργήσεις κινούμενες εικόνες που θα απεικονίζουν κάποιο γεγονός ή κάποια ιστορία. Στην εργασία ασχολήθηκα με ένα μικρό μέρος από τις δυνατότητες που μας παρέχει το 3D αλλά μου φάνηκε χρήσιμη και διασκεδαστική η εκμάθησή του.

Για την δημιουργία της εφαρμογής υπήρχαν αρκετά προβλήματα όπως ότι δεν είχα την δυνατότητα να αναπτύξω την εφαρμογή σε iOS διότι δεν έχω στην κατοχή μου κάποια Apple συσκευή. Παρόλα αυτά η διαδικασία της σχεδίασης και ανάπτυξης της εφαρμογής ήταν ευχάριστη και χρήσιμη ως την εκμάθηση των εργαλείων της Unity για μια AR(επαυξημένη πραγματικότητα) εφαρμογή.

Στην εφαρμογή μπορούν να προστεθούν και άλλα ψάρια για περισσότερη συλλογή πληροφοριών. Επίσης έχουμε την δυνατότητα με την Unity να προστεθούν κάποια planes ή markers που με το οποία να τα εντοπίζει η κάμερα της συσκευής και να εμφανίζει τότε το αντίστοιχο ψάρι. Μια επέκταση που μπορεί να γίνει στην εφαρμογή είναι να έχει πιο αναλυτικές πληροφορίες για τα ψάρια όπως διάφορες συνταγές που μπορούν να μαγειρευτούν με αυτά.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Βιώσιμη Ανάπτυξη*. (n.d.). αλιΑμβρα. Retrieved December 29, 2023, from <https://www.aliamvra.gr/sustainable-development/>
- Παπαδόπουλος*. (2016, October 9). Retrieved December 29, 2023, from <https://maredu.hcg.gr/modules/document/file.php/MAK264/%CE%9D%CE%B1%CF%85%CF%84%CE%B9%CE%BA%CE%AD%CF%82%20%CE%95%CF%80%CE%B9%CE%BA%CE%BF%CE%B9%CE%BD%CF%89%CE%BD%CE%AF%CE%B5%CF%82%20-%20%CE%9D%CE%97%CE%9F/%CE%91%CE%BE%CE%B9%CE%BF%CF%80%CE%BF%CE%AF%CE%B7%CF%>
- Αλεξίου, Κ. (2021, March 19). *Ιδρυματικό Αποθετήριο Πανεπιστημίου Δυτικής Αττικής*. Retrieved December 29, 2023, from <https://polynoe.lib.uniwa.gr/xmlui/bitstream/handle/11400/516/ntst8056121.pdf?sequence=1&isAllowed=y>
- AR Engines — Unity*. (2023, April 2). Makaka Games. Retrieved January 2, 2024, from <https://makaka.org/unity-tutorials/top-ar-engines#what-the-top-ar-engine-for-unity>
- Augmented Reality*. (2018, May 30). Inc. Magazine. Retrieved December 29, 2023, from <https://www.inc.com/james-paine/10-real-use-cases-for-augmented-reality.html>
- Augmented reality app (AR app)*. (n.d.). TechTarget. Retrieved December 26, 2023, from <https://www.techtarget.com/whatis/definition/augmented-reality-app-AR-app>
- Build Settings*. (n.d.). Unity - Manual. Retrieved January 3, 2024, from <https://docs.unity3d.com/Manual/BuildSettings.html>
- eLearning Industry*. (2023, March 19). eLearning Industry. Retrieved December 26, 2023, from <https://elearningindustry.com/making-science-come-alive-3d-modeling-in-science-education>
- The Hierarchy*. (n.d.). Unity - Manual. Retrieved January 3, 2024, from <https://docs.unity3d.com/Manual/Hierarchy.html>
- High-Poly Models vs Low-Poly Ones*. (n.d.). CGIFURNITURE. Retrieved December 29, 2023, from <https://cgifurniture.com/high-poly-models-vs-low-poly-ones/>
- The Inspector window*. (n.d.). Unity - Manual. Retrieved January 3, 2024, from <https://docs.unity3d.com/Manual/UsingTheInspector.html>
- ITS*. (2021, November 23). Information Transformation Services. Retrieved December 27, 2023, from <https://it-s.com/what-is-3d-wireframe-modeling/>
- ITS*. (2021, November 30). Information Transformation Services. Retrieved December 27, 2023, from <https://it-s.com/what-is-3d-surface-modeling/>
- Low Poly*. (n.d.). Domestika. Retrieved December 29, 2023, from <https://www.domestika.org/en/blog/6206-what-is-low-poly-learn-about-polygon-art-for-video-games-and-more>
- Modeling*. (n.d.). Blender Documentation. Retrieved January 4, 2024, from <https://docs.blender.org/manual/en/latest/modeling/meshes/introduction.html>
- The Scene view*. (n.d.). Unity - Manual. Retrieved January 3, 2024, from <https://docs.unity3d.com/Manual/UsingTheSceneView.html>
- Scripts*. (n.d.). Unity - Manual. Retrieved January 4, 2024, from <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>
- The Simulator view*. (n.d.). Unity - Manual. Retrieved January 3, 2024, from <https://docs.unity3d.com/Manual/device-simulator-view.html>
- Solid Modeling*. (n.d.). Spatial Corp. Retrieved December 27, 2023, from <https://www.spatial.com/resources/glossary/what-is-solid-modeling>
- ThisFish*. (2022, February 23). ThisFish. Retrieved December 29, 2023, from <https://this.fish/el/blog/ai-guide-globally-scaling-electronic-monitoring/>
- 3D Animation*. (2021, December 21). Upwork. Retrieved December 30, 2023, from <https://www.upwork.com/resources/what-is-3d-animation>
- 3D in Medical Training and Simulations*. (2023, September 22). IWConnect. Retrieved December 26, 2023, from <https://iwconnect.com/3d-in-medical-training/>

TS2. (2023, October 9). YouTube. Retrieved December 29, 2023, from <https://ts2.space/el/%CF%84%CE%B1-%CE%BF%CF%86%CE%AD%CE%BB%CE%B7-%CF%84%CE%B7%CF%82-%CE%B4%CE%BF%CF%81%CF%85%CF%86%CE%BF%CF%81%CE%B9%CE%BA%CE%AE%CF%82-%CF%84%CE%B5%CF%87%CE%BD%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1%CF%82-4/#gsc.tab=0>

Unity Visual Scripting. (n.d.). Unity. Retrieved January 4, 2024, from <https://unity.com/features/unity-visual-scripting>

UV Mapping. (2023, March 24). All3DP. Retrieved December 30, 2023, from <https://all3dp.com/2/blender-uv-mapping-simply-explained/>

What is 3D modelling. (2022, March 18). FutureLearn. Retrieved December 26, 2023, from https://www.futurelearn.com/info/blog/general/what-is-3d-modelling#What_is_3D_modelling

What is augmented reality app (AR app). (n.d.). TechTarget. Retrieved December 29, 2023, from <https://www.techtarget.com/whatis/definition/augmented-reality-app-AR-app>

What Is Unity. (2023, July 1). GameDev Academy. Retrieved January 2, 2024, from https://gamedevacademy.org/what-is-unity/#What_is_Unity_and_Who_Owns_Unity_Game_Engine

ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ

C# Script

TouchKefalos.cs

```
using System.Collections;
using System.Collections.Generic;
using Unity.VisualScripting;
using UnityEngine;
using UnityEngine.UIElements;
using UnityEngine.UI;

public class TouchKefalos : MonoBehaviour
{
    public GameObject CanvasKouts;
    public GameObject CanvasSard;
    public GameObject CanvasTsip;
    public GameObject CanvasKefalos;
    public GameObject kefalosrot;
    public GameObject sardelarot;
    public GameObject koutsomourarot;
    public GameObject tsipourarot;
    public Scrollbar scrollbarSar;
    public Scrollbar scrollbarTsip;
    public Scrollbar scrollbarKouts;
    public Scrollbar scrollbarKef;
    public GameObject Reset;
```

```
// Start is called before the first frame update
```

```
void Start()
```

```
{
```

```
}
```

```
// Update is called once per frame
```

```
void Update()
```

```
{
```

```
if (Reset.active == false)
```

```
{
```

```
if (Input.touchCount > 0 && Input.touches[0].phase ==
```

```
TouchPhase.Began)
```

```
{
```

```
Ray ray = Camera.main.ScreenPointToRay(Input.touches[0].position);
```

```
RaycastHit hit;
```

```
if (Physics.Raycast(ray, out hit))
```

```
{
```

```
if (hit.transform.tag == "kefalos")
```

```
{
```

```

if (CanvasKefalos != null)
{
    CanvasKouts.SetActive(false);
    CanvasSard.SetActive(false);
    CanvasTsip.SetActive(false);
    CanvasKefalos.SetActive(true);
    kefalosrot.SetActive(true);
    sardelarot.SetActive(false);
    koutsomourarot.SetActive(false);
    tsipourarot.SetActive(false);
    Reset.SetActive(true);
}
}
}
}
}
}
}

```

TouchKoutsomoura.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UIElements;
using UnityEngine.UI;

```

```
public class TouchKoutsomoura : MonoBehaviour
```

```
{
```

```
    public GameObject CanvasKouts;
```

```
    public GameObject CanvasSard;
```

```
    public GameObject CanvasTsip;
```

```
    public GameObject CanvasKefalos;
```

```
    public GameObject koutsomourarot;
```

```
    public GameObject sardelarot;
```

```
    public GameObject kefalosrot;
```

```
    public GameObject tsipourarot;
```

```
    public Scrollbar scrollbarSar;
```

```
    public Scrollbar scrollbarTsip;
```

```
    public Scrollbar scrollbarKouts;
```

```
    public Scrollbar scrollbarKef;
```

```
    public GameObject Reset;
```

```
// Start is called before the first frame update
```

```
void Start()
```

```
{
```

```
}
```

```
// Update is called once per frame
```



```

void Update()
{
    if (Reset.active == false)
    {
        if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
        {
            Ray ray = Camera.main.ScreenPointToRay(Input.touches[0].position);
            RaycastHit hit;

            if (Physics.Raycast(ray, out hit))
            {
                if (hit.transform.tag == "koutsomoura")
                {
                    if (CanvasKouts != null)
                    {
                        CanvasKouts.SetActive(true);
                        CanvasSard.SetActive(false);
                        CanvasTsip.SetActive(false);
                        CanvasKefalos.SetActive(false);
                        koutsomourarot.SetActive(true);
                        sardelarot.SetActive(false);
                        kefalosrot.SetActive(false);
                        tsipourarot.SetActive(false);
                        Reset.SetActive(true);
                    }
                }
            }
        }
    }
}

```

```
    }  
  }  
}  
}  
}
```

TouchTsipoura.cs

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UIElements;  
using UnityEngine.UI;
```

public class TouchKoutsomoura : MonoBehaviour

```
{  
  public GameObject CanvasKouts;  
  public GameObject CanvasSard;  
  public GameObject CanvasTsip;  
  public GameObject CanvasKefalos;  
  public GameObject koutsomourarot;  
  public GameObject sardelarot;  
  public GameObject kefalosrot;  
  public GameObject tsipourarot;  
  public Scrollbar scrollbarSar;
```

```

public Scrollbar scrollbarTsip;

public Scrollbar scrollbarKouts;

public Scrollbar scrollbarKef;

public GameObject Reset;

// Start is called before the first frame update

void Start()
{
}

// Update is called once per frame

void Update()
{
    if (Reset.active == false)
    {
        if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
        {
            Ray ray = Camera.main.ScreenPointToRay(Input.touches[0].position);

            RaycastHit hit;

            if (Physics.Raycast(ray, out hit))
            {
                if (hit.transform.tag == "koutsomoura")
                {
                    if (CanvasKouts != null)
                    {
                        CanvasKouts.SetActive(true);
                    }
                }
            }
        }
    }
}

```

```
CanvasSard.SetActive(false);  
CanvasTsip.SetActive(false);  
CanvasKefalos.SetActive(false);  
koutsomourarot.SetActive(true);  
sardelarot.SetActive(false);  
kefalosrot.SetActive(false);  
tsipourarot.SetActive(false);  
Reset.SetActive(true);  
    }  
    }  
    }  
    }  
    }  
    }  
}
```

TouchSardela.cs

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UIElements;  
using UnityEngine.UI;  
  
public class TouchSardela : MonoBehaviour  
{  
    public GameObject CanvasKouts;
```

```

public GameObject CanvasSard;
public GameObject CanvasTsip;
public GameObject CanvasKefalos;
public GameObject sardelarot;
public GameObject kefalosrot;
public GameObject tsipourarot;
public GameObject koutsomourarot;
public Scrollbar scrollbarSar;
public Scrollbar scrollbarTsip;
public Scrollbar scrollbarKouts;
public Scrollbar scrollbarKef;
public GameObject Reset;

// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{
if (Reset.active == false)
{
if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
{

```

```

Ray ray = Camera.main.ScreenPointToRay(Input.touches[0].position);

RaycastHit hit;

if (Physics.Raycast(ray, out hit))
{
    if (hit.transform.tag == "sardela")
    {
        if (CanvasKefalos != null)
        {
            CanvasKouts.SetActive(false);
            CanvasSard.SetActive(true);
            CanvasTsip.SetActive(false);
            CanvasKefalos.SetActive(false);
            sardelarot.SetActive(true);
            kefalosrot.SetActive(false);
            koutsomourarot.SetActive(false);
            tsipourarot.SetActive(false);
            Reset.SetActive(true);
        }
    }
}
}
}
}
}
}
}
}
}

```