

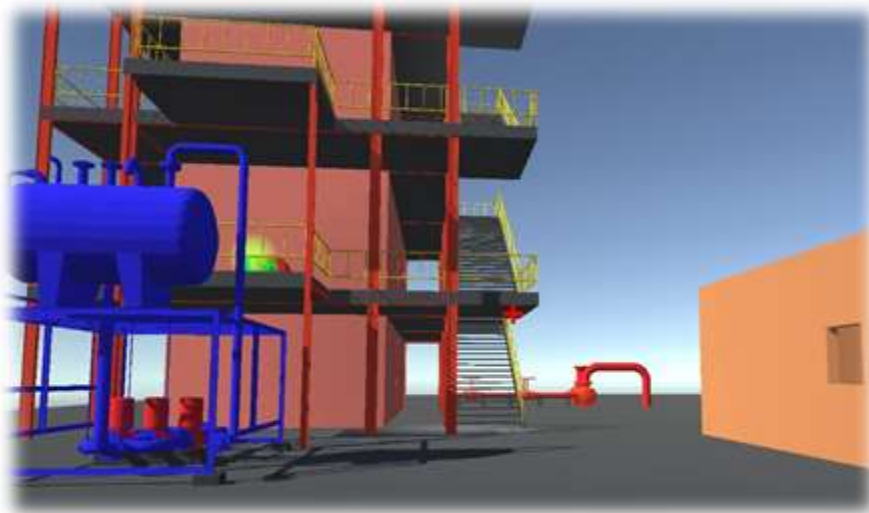


ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ



**«ΠΡΟΣΟΜΟΙΩΤΗΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΓΙΑ ΠΑΡΑΓΩΓΗ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ
ΣΕ ΗΛΕΚΤΡΟΝΙΚΟ ΥΠΟΛΟΓΙΣΤΗ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΗΣ UNITY 3D»**

Όνοματεπώνυμο: Ασπράκης Ελευθέριος-Τσαμπίκος

Αριθμός Μητρώου: 6054

Επιβλέπων καθηγητής: Κωνσταντίνος Βασιλάκης

ΚΡΗΤΗ 2021



ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ



Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη τρισδιάστατου προσομοιωτή βιομηχανικής εγκατάστασης για ηλεκτρονικό υπολογιστή με την βοήθεια της μηχανής Unity 3D. Για τον σχεδιασμό της εγκατάστασης χρησιμοποιήθηκαν προγράμματα της Autodesk κυρίως και για την υλοποίηση των γραφικών χαρακτηριστικών το ελεύθερο λογισμικό Blender. Στον προσομοιωτή ο χρήστης μπορεί να αλληλοεπιδράσει με τα συστήματα της εγκατάστασης και να ορίσει την κατάσταση τους, όπως γίνεται σε μια πραγματική. Τέλος, μέσω της χρήσης του προσομοιωτή επιτυγχάνεται τόσο η εκπαίδευση του χρήστη σε ένα χώρο βιομηχανικής εγκατάστασης όσο και η αξιολόγηση του.

Abstract

This project deals with the development of a 3D industrial computer installation simulator with the help of the Unity 3D software. Autodesk programs were used mainly for the design of the installation and for the implementation of the graphical features the open source Blender software. In the simulator the user could interact with the installation systems and set their status as it does to an actual one. Finally, the simulator aims both to train the user in an industrial installation area and to evaluate them.

Λέξεις Κλειδιά: Unity 3D Editor, C# , Revit, AutoCAD , Blender



Περιεχόμενα

Κεφάλαιο1	6
Γενική περιγραφή εργασίας	6
Κεφάλαιο 2	6
Εργαλεία ανάπτυξης του προσομοιωτή	6
Unity 3D	7
Unity Simulation	8
Unity Reflect	9
Autodesk	13
Revit	13
AutoCAD Plant 3D	14
Blender	14
Κεφάλαιο 3	16
Σκοπός και στόχοι	16
Απαιτήσεις εξοπλισμού	16
Μεθοδολογία	18
Revit	19
Autocad Plant 3D	25
Blender	29
Unity 3D	35
Κεφάλαιο 4	63



ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Ο προσομοιωτής	63
Εγγραφή κίνησης του χρήστη	67
Σενάριο 1.....	68
Εικόνες.....	72
Βιβλιογραφία.....	75



Κεφάλαιο1

Γενική περιγραφή εργασίας

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη ενός προσομοιωτή βιομηχανικής εγκατάστασης παραγωγής ηλεκτρικής ενέργειας για ηλεκτρονικό υπολογιστή που είναι ανεπτυγμένος στην εφαρμογή Unity 3D και χρησιμοποιείται κυρίως για την ανάπτυξη ηλεκτρονικών παιχνιδιών. Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η CSharp(C#).

Στο σχεδιασμό της εγκατάστασης χρησιμοποιήθηκαν προγράμματα σχεδίασης της Autodesk επειδή προσφέρονται δωρεάν σε φοιτητές για δυο χρόνια με δυνατότητα ανανέωσης. Για την υλοποίηση των γραφικών χαρακτηριστικών χρησιμοποιήθηκε η εφαρμογή Blender, το οποίο είναι ανοιχτού κώδικα λογισμικό, που προσφέρεται δωρεάν στους χρήστες.

Στην παρούσα εργασία δεν υλοποιήθηκε πλήρως μια τέτοια εγκατάσταση γιατί αυτό θα απαιτούσε σχέδια και στοιχεία από μια πραγματική και αυτό δεν ήταν δυνατόν, εξαιτίας του κανονισμού της πνευματικής ιδιοκτησίας που ορίζεται στην βιομηχανία.

Κεφάλαιο 2

Εργαλεία ανάπτυξης του προσομοιωτή

Η επιλογή των παρακάτω εφαρμογών έγινε με κριτήριο την συνδεσιμότητα μεταξύ τους καθώς και την αποτελεσματικότητα ως προς την συμβατότητα του τελικού αποτελέσματος τους (σχέδια) ώστε να ενταχθούν στο περιβάλλον της Unity 3D.

Στις επόμενες ενότητες θα παρουσιαστούν αναλυτικά τα εργαλεία, τα οποία χρησιμοποιήθηκαν για την κατασκευή του προσομοιωτή και θα αναφερθούν τα βασικά χαρακτηριστικά τους που ήταν χρήσιμα για την κατασκευή του. Οι εφαρμογές που χρησιμοποιήθηκαν είναι οι εξής:

- Unity 3D
- AutoCAD Plant 3d
- AutoCAD
- Revit
- Blender



Unity 3D

Η Unity3D είναι μια μηχανή κυρίως για την ανάπτυξη παιχνιδιών που κατασκευάστηκε από την «Unity Technologies». Οι αρχικές εκδόσεις της υποστηρίζονταν στο λειτουργικό σύστημα MacOS αλλά πλέον σε Linux και Windows. Η πρώτη κυκλοφορία της παρουσιάστηκε τον Ιούνιο του 2005, ενώ από το 2018 τα «Build» του υποστηρίζονται από περισσότερα από 25 λειτουργικά συστήματα. Κάποια από αυτά είναι τα **Android, HTML5, Xbox, Mac, Windows** και **PlayStation**. (WikiPedia, 2021)

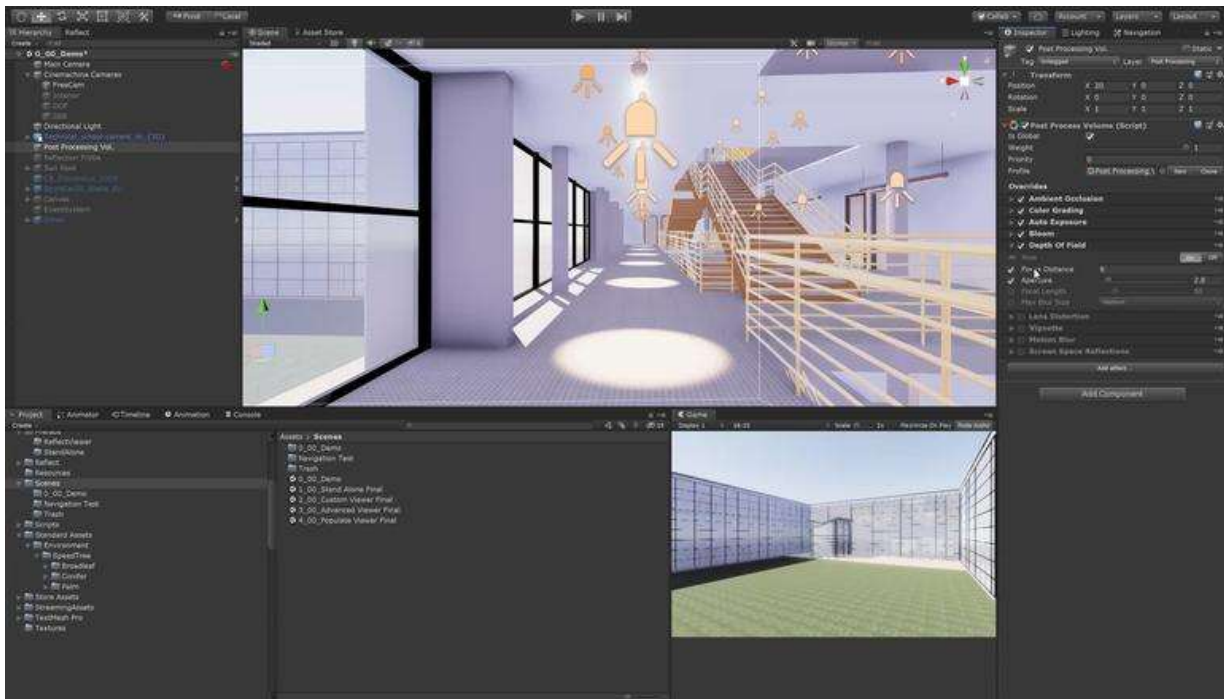
Στην ιστοσελίδα «unity3d.com/get-unity/download» δίνεται η δυνατότητα απόκτησης της **Unity 3D** μηχανής εντελώς δωρεάν και προτείνεται η βοηθητική εφαρμογή **Unity Hub** για την διαχείριση των έργων καθώς και για άλλες δυνατότητες που θα αναφερθούν στα επόμενα κεφάλαια.

Η εφαρμογή Unity 3D έχει τρία κυρία μέρη τα οποία είναι:

- Η μηχανή, επιτρέπει την δημιουργία, δοκιμές, και να τρέχει εφαρμογή/παιχνίδι σε διαφορετικά λειτουργικά συστήματα.
- Η εφαρμογή, είναι η έμμεση επαφή «Interface» του χρήστη με την μηχανή και μέσω αυτής δίνεται η δυνατότητα προεπισκόπησης του έργου που κατασκευάζεται.
- Ο μεταγλωττιστής της γλώσσας, στην προκειμένη περίπτωση είναι το Visual Studio.

Οι κύριοι λόγοι που χρησιμοποιήθηκε αυτή η μηχανή για την κατασκευή του προσομοιωτή είναι οι εξής:

- Προσφέρεται δωρεάν.
- Είναι εύκολη στην εκμάθηση λόγω του υλικού που προσφέρεται από την εταιρία αλλά και από ανεξάρτητους χρήστες όπως ιστοσελίδες και βίντεο (YouTube).
- Υποστηρίζεται στα περισσότερα λειτουργικά συστήματα Linux, Windows και MacOS.
- Ενσωματωμένες βιβλιοθήκες προγραμματισμού.
- Μεγάλη κοινότητα χρηστών.



Εικόνα 1 - το περιβάλλον της εφαρμογής Unity3d

Η μηχανή μπορεί να χρησιμοποιηθεί για την δημιουργία διαδραστικών εφαρμογών δυο διαστάσεων, τριών διαστάσεων, εικονικής πραγματικότητας (VR), καθώς και προσομοιώσεων, αλλά φυσικά δεν περιορίζεται μόνο σε αυτά. Ανάλογα τους στόχους και τις απαιτήσεις του κατασκευαστή, και με συνδυασμό των πολλών δυνατοτήτων που προσφέρει η μηχανή μπορεί να επιτευχθεί ένα πολύ αξιόλογο αποτέλεσμα και για τον λόγο αυτό πλέον συμμετέχει και σε άλλους τεχνολογικούς τομείς. Εκτός της βιομηχανίας παιχνιδιών, η μηχανή έχει υιοθετηθεί και από άλλες βιομηχανίες όπως ο κινηματογράφος, η αυτοκινητοβιομηχανία, η αρχιτεκτονική κτιρίων, η μηχανική και γενικά σε κατασκευές. (Unity, 2021)

Αξίζει να αναφερθούν κάποια συμβατά προγράμματα της Unity3D, όπως το Unity Simulation, Unity Reflect, τα οποία δεν χρησιμοποιούνται στην παρούσα εργασία λόγω τους κόστους αγοράς τους. Στην ιστοσελίδα της εταιρίας «unity.com» παρέχονται πληροφορίες που προσφέρει γενικά η εταιρία. Έχει μεγάλη συχνότητα στις αναβαθμίσεις της και στις καινοτομίες που προσφέρει για διάφορους κλάδους.

Unity Simulation

Το Unity Simulation προσφέρει δυνατότητα δοκιμής, εκπαίδευσης και επικύρωσης ενός συστήματος όπως ρομπότ, αυτοκίνητα, συστήματα εντοπισμού προσώπων ή αντικειμένων σε έναν τρισδιάστατο εικονικό κόσμο σε πραγματικό χρόνο.



Η εφαρμογή είναι μια υπηρεσία «Cloud» που επιτρέπει στους προγραμματιστές προϊόντων, τους ερευνητές και τους μηχανικούς να εκτελούν εύκολα και αποτελεσματικά χιλιάδες περιπτώσεις προσομοιώσεων στο «Cloud» που τις υπολογίζει ένας υπερυπολογιστής.

Αναπτύσσοντας έργα «Projects» στο Unity3D που αντιπροσωπεύουν πραγματικά περιβάλλοντα, προσαρμόζοντας παραμέτρους σε αυτά τα έργα για προσομοίωση και εκτελώντας πολλά προσομοιωμένα σενάρια στο «Cloud» γρηγορότερα από ό, τι σε πραγματικό χρόνο, μπορεί να επιτευχθεί ένας εύλογος χρόνος που στον πραγματικό κόσμο θα απαιτούσε πολλές παραπάνω ώρες προσομοίωσης και μεγαλύτερος κόστος.

(unity3d, n.d.)

Unity Reflect

Με την κυκλοφορία του **Unity Reflect**, του νέου προϊόντος για τη βιομηχανία αρχιτεκτονικής, μηχανικής και κατασκευών (AEC) που ενσωματώνεται με τις παρακάτω εφαρμογές Autodesk Revit, BIM 360, Navisworks, SketchUp και Rhino, δίδονται οι εξής δυνατότητες:

- Οι σχεδιαστές και οι μηχανικοί μπορούν να μεταφέρουν μοντέλα Revit σε πραγματικό χρόνο, συμπεριλαμβανομένης της επαυξημένης πραγματικότητας (AR) και της εικονικής πραγματικότητας (VR)
- Οι προγραμματιστές μπορούν να κατασκευάσουν με τη μηχανή Unity3D εξαιρετικά προσαρμοσμένες εφαρμογές 3D σε πραγματικό χρόνο, αξιοποιώντας τα δεδομένα από το Unity Reflect.
- Οι πελάτες και οι συνεργάτες μπορούν να ελέγχουν τα έργα με καθηλωτικό, διαδραστικό τρόπο σε μια σειρά συσκευών, όπως ηλεκτρονικός υπολογιστής, κινητές συσκευές κ.τ.λ.

(UN3, 2019)

Τι είναι η Unity Reflect

Το Unity Reflect χρησιμοποιείται από προγραμματιστές, αρχιτέκτονες, σχεδιαστές και μηχανικούς κτιρίων που δεν είναι εξοικειωμένοι στον προγραμματισμό. Μέσω του Unity Reflect δίνεται η δυνατότητα μεταφοράς δεδομένων Building Information Modeling (BIM) σε εμπειρίες σε πραγματικό χρόνο με μια σειρά συσκευών όπως κινητά τηλέφωνα και υπολογιστές και σε επαυξημένη πραγματικότητα AR και εικονική πραγματικότητα VR.



Επειδή το Unity 3D στον πυρήνα του είναι μια πλατφόρμα ανάπτυξης, δίνει την δυνατότητα δημιουργίας διαφοροποιημένων εφαρμογών BIM σε πραγματικό χρόνο και προσαρμόζει την εμπειρία του χρήστη δημιουργώντας πάνω από το Unity Reflect με την μηχανή Unity3D .

Εν κατακλείδι, το Unity Reflect είναι ένα εργαλείο το οποίο γεφυρώνει το χάσμα μεταξύ σχεδιασμού και κατασκευής.

(UN3, 2019)

Γιατί κατασκευάστηκε η Unity Reflect

Η ανάγκη της βιομηχανίας αρχιτεκτόνων , μηχανικών κτλ. κατέστησε σαφές ότι υπάρχει ανάγκη ενός εργαλείου το οποίο θα προσφέρει τρισδιάστατη εμπειρία σε πραγματικό χρόνο. Πριν την κατασκευή της Unity Reflect για την πραγματοποίηση μιας τέτοιας εμπειρίας απαιτούσε πολλά βήματα και ανθρώπινο δυναμικό που στο τελικό αποτελέσματα το πιο πιθανόν ήταν να απόφερε σφάλματα γεωμετρικών χαρακτηριστικών που απαιτούσε επιπρόσθετη εργασία για την διόρθωση τους.



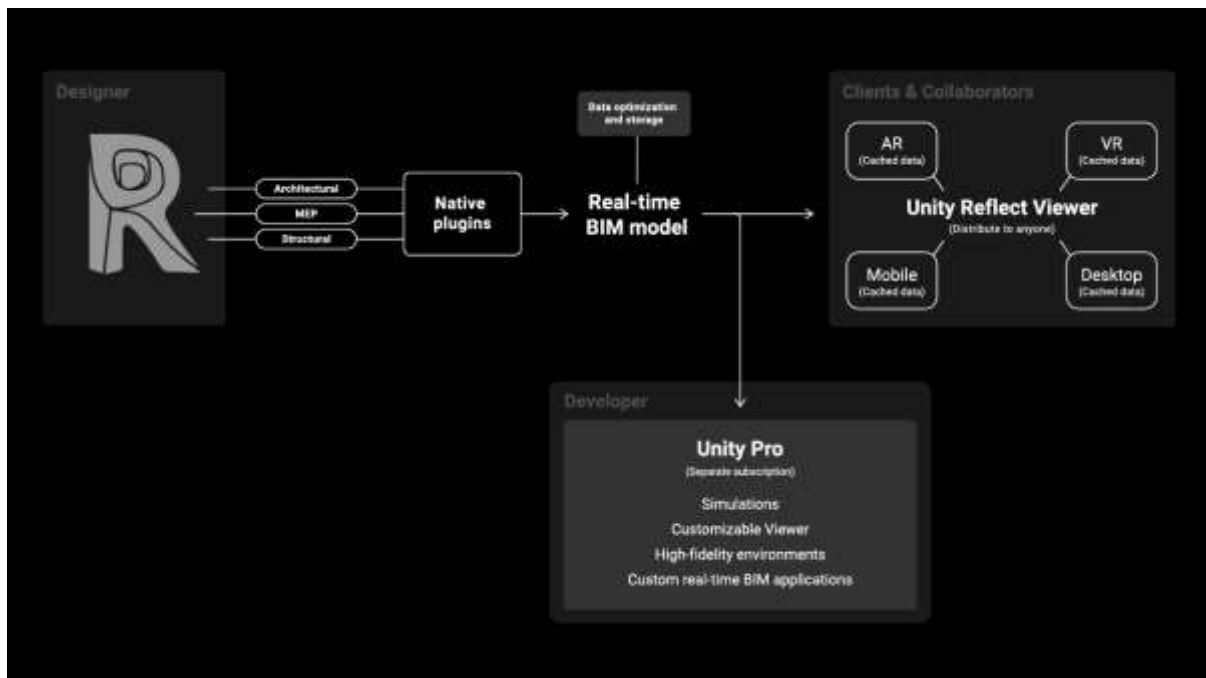
Εικόνα 2 - Απεικόνιση του Revit μοντέλου στον υπολογιστή και αντίστοιχα η απεικόνισή του στην Unity Reflect στο tablet

Είναι γεγονός ότι λίγα τρισδιάστατα σχέδια φτάνουν σε πραγματικό χρόνο σε ένα εργοτάξιο. Σε μια βιομηχανία λοιπόν που ο χρόνος είναι κρίσιμος και οπότε ο σχεδιασμός αλλάζει κατά την ροή του έργου και παράλληλα συνεργάζονται μεταξύ τους διαφορετικά συνεργεία

έρχεται η Unity Reflect για να διευκολύνει την ενσωμάτωση των στοιχείων του έργου σε αυτή και να τα παρουσιάσει σε πραγματικό χρόνο. (UN3, 2019)

Πως δουλεύει

Το Unity Reflect καθιστά τη διαδικασία εισαγωγής μοντέλων Revit σε 3D σε πραγματικό χρόνο εξαιρετικά απλή. Μετά την εγκατάσταση του πρόσθετου Unity Reflect, οι χρήστες απλά κάνουν κλικ στο εικονίδιο Unity που βρίσκεται στη γραμμή εργαλείων του Revit για να αρχίσουν να συγχρονίζουν το μοντέλο τους. (UN3, 2019)



Εικόνα 3 - Σχηματική απεικόνιση λειτουργίας Revit με Unity Reflect

Όλη αυτή η διαδικασία γίνεται μέσα στα γνωστά όρια του λογισμικού BIM. Η ενσωμάτωση του Unity Reflect γίνεται με τις παρακάτω εφαρμογές.

- Revit
- BIM 360
- Navisworks
- SketchUp
- Rhino



Πολλοί σχεδιαστές και μηχανικοί που εργάζονται σε διαφορετικά συστήματα σε ένα μοντέλο-σχέδιο του έργου μπορούν να προωθήσουν τα μεμονωμένα σχέδια τους όπως ο αρχιτεκτονικός σχεδιασμός, ο δομικός ή μηχανικός, ο ηλεκτρικός και ο υδραυλικός στο ίδιο έργο Reflect. Αυτά τα δεδομένα BIM ενώνονται και βελτιστοποιούνται αυτόματα για 3D σε πραγματικό χρόνο και στη συνέχεια αποθηκεύονται σε τοπικό υπολογιστή ή διακομιστή δικτύου. Οποιοσδήποτε υπολογιστής εκτελεί το Unity Reflect μπορεί να λειτουργήσει ως διακομιστής.

Ένας διακομιστής διαχείρισης έργων που φιλοξενείται στο «Cloud» βοηθά το χρήστη να προσδιορίσει σε ποια έργα έχει πρόσβαση, διευκολύνοντας τη διαχείριση έργων και την πρόσβαση σε δικαιώματα σε πολλά γραφεία και τμήματα. Δεν αποθηκεύονται δεδομένα 3D σε αυτόν το διακομιστή, μόνο ονόματα έργων και δικαιώματα.

Τέλος, οι πελάτες και οι συνεργάτες μπορούν να έχουν πρόσβαση στο «Unity Reflect Viewer», οι οποίοι παρακάτω θα αναφέρονται σαν θεατές. Η εφαρμογή είναι αρχικά διαθέσιμη στον υπολογιστή, υποστηρίζοντας το iOS, Android, συμπεριλαμβανομένων των ARKit της Apple και HTC Vive, το πρώτο σετ μικροφώνου-ακουστικών VR που είναι συμβατό από το κουτί με το Unity Reflect. Πολλοί ενδιαφερόμενοι, από τους υπεύθυνους των έργων έως τους συμβούλους και τους πελάτες, μπορούν να έχουν πρόσβαση ταυτόχρονα στη συσκευή της επιλογής τους, διευκολύνοντας τη συνεργασία και την επίτευξη συνεργασίας. Επίσης, οι θεατές έχουν δυνατότητα στο ενσωματωμένο φιλτράρισμα BIM για την απομόνωση διαφόρων στοιχείων του έργου, από μηχανικά, υδραυλικά έως και δομικά στοιχεία.



Εικόνα 4 - Περιβάλλον Unity Reflect για τους θεατές

(UN3, 2019)

Autodesk

Χρησιμοποιήθηκαν κυρίως προγράμματα της Autodesk για τον σχεδιασμό, τα οποία είναι **AutoCAD**, **Plant 3D**, **Revit** γιατί δίνουν ευλυγισία στον σχεδιασμό λόγω του εύρους των εφαρμογών της και εξειδίκευση σε διαφορά αντικείμενα όπως αρχιτεκτονικό, μηχανολογικό και ηλεκτρολογικό σχέδιο. Στην υλοποίηση του προσομοιωτή για την δημιουργία γραφικών χαρακτηριστικών, αντί του 3ds Max χρησιμοποιήθηκε το Blender λόγω της δυσκολίας που συναντήθηκε στην εκμάθηση του 3Ds Max.

Να σημειωθεί ότι όλα τα προγράμματα της Autodesk μεταξύ τους έχουν δυνατότητα σύνδεσης (link), όπως για παράδειγμα ένα τρισδιάστατο σχέδιο από το Revit μπορεί να γίνει εισαγωγή στο AutoCAD χωρίς να αποφέρει γεωμετρικά σφάλματα. Η συμβατότητα μεταξύ τους είναι πλήρης.

Revit

Το Revit είναι μια εφαρμογή σχεδιασμού κτιρίων για αρχιτέκτονες, πολιτικούς μηχανικούς, μηχανολόγους, μηχανικούς ηλεκτρολόγους, υδραυλικούς μηχανικούς, σχεδιαστές και εργολάβους.



Το λογισμικό επιτρέπει στους χρήστες να σχεδιάσουν ένα κτίριο και τα στοιχεία του όπως υδραυλικά, ηλεκτρολογικά, κλιματισμό, είδη υγιεινής, μεταλλικές κατασκευές, κουφώματα και πολλά αλλά σε τρισδιάστατα μοντέλα. Να αναπαραστήσουν το μοντέλο σε δυσδιάστατο σχέδιο και να αποκτήσουν πρόσβαση σε πληροφορίες κτιρίου από τη βάση δεδομένων του κτιριακού μοντέλου. (Revit WikiPedia, 2021)

AutoCAD Plant 3D

Το Plant 3D είναι βοηθητική εφαρμογή του AutoCAD που στοχεύει στην μελέτη και σχεδιασμό βιομηχανικών εγκαταστάσεων και επεξεργασίας. Διαθέτει τα απαραίτητα εργαλεία και χαρακτηριστικά που χρειάζεται ένας σχεδιαστής για να μελετήσει-δημιουργήσει λεπτομερή μοντέλα διαφόρων εγκαταστάσεων, συμπεριλαμβανομένων σωληνώσεων, δομικών και εξοπλισμού που κατασκευάζονται αρχικά στο AutoCAD. Χρησιμοποιείται κυρίως από σχεδιαστές βιομηχανικών εγκαταστάσεων. Υπάρχει δυνατότητα τρισδιάστατης σχεδίασης και δυσδιάστατη σχεδίασης.

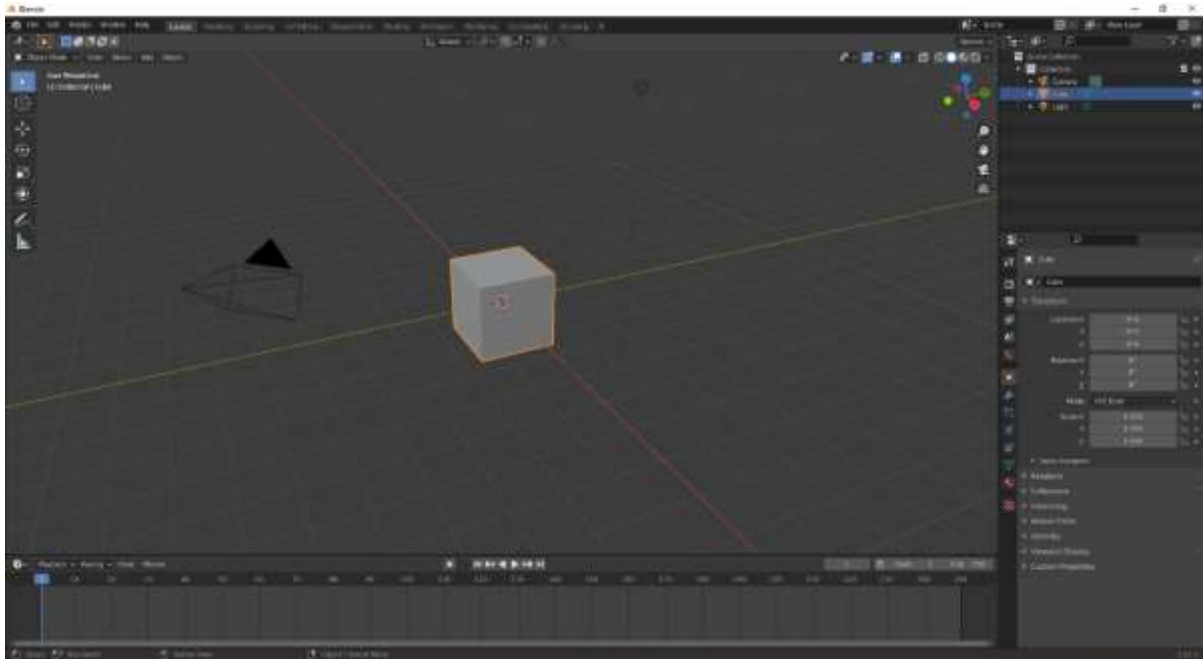
Κάποιες δυνατότητες που προσφέρει το πρόγραμμα σε ένα σχεδιαστή είναι οι εξής:

- Δημιουργία και επεξεργασία P&ID (γραμμικό δυσδιάστατο σχέδιο λειτουργείας) με βιβλιοθήκες συμβόλων για βιομηχανικές εγκαταστάσεις.
- Διάγνωση σχεδίου για πιθανά σφάλματα.
- Δημιουργία και επεξεργασία τρισδιάστατων σχεδίων για δομικό χάλυβα, σωληνώσεις και εξοπλισμό προσαρμοσμένο ώστε να ταιριάζει σε πολλές εφαρμογές.
- Τυποποιημένες σωληνώσεις, δομικός χάλυβας και εξοπλισμός όπως αντλίες, επιστόμια, εναλλάκτες, δεξαμενές κτλ.
- Δυνατότητα εξαγωγής αρχείου για τρισδιάστατο μοντέλο.

(AutoCAD Plant 3D Asti.com, 2021)

Blender

Για την δημιουργία γραφικών χαρακτηριστικών χρησιμοποιήθηκε το Blender, λόγω της μεγάλης κοινότητας που το αντιπροσωπεύει και την μεγάλη γκάμα βιβλιοθηκών και προσθέτων λειτουργιών όπως τα Plugins. Ένα μεγάλο πλεονέκτημα είναι ότι είναι **δωρεάν** και ότι είναι **ανοικτού κώδικα λογισμικό** δίνοντας το πλεονέκτημα ότι με τεχνογνωσία προγραμματισμού στην γλώσσα Python μπορεί να προσαρμοστεί στα χαρακτηριστικά που επιθυμεί ο κάθε χρήστης. Τα ελεύθερα λογισμικά ανοικτού κώδικα συχνά συναγωνίζονται λογισμικά εταιριών όπως τις Autodesk. Οι δυνατότητες του Blender συναγωνίζονται αυτές του 3ds Max και παρέχουν στους χρήστες τους παρόμοια χαρακτηριστικά λειτουργιών και η εκμάθηση του γίνεται πιο εύκολη, λόγω του μεγάλου όγκου πληροφορίας που παρέχεται στο διαδίκτυο από βίντεο, εικόνες και άρθρα. (WikiPedia, 2021)



Εικόνα 5 - Φωτογραφία της εφαρμογής Blender το περιβάλλον του είναι προκαθορισμένο από τους προγραμματιστές. Πηγή: Wikipedia.

Είναι διαθέσιμο για όλα τα κύρια λειτουργικά συστήματα όπως τα Windows της Microsoft το Linux και το Mac OS X. Το Blender διαθέτει προχωρημένα εργαλεία για κινούμενα σχέδια ή ταινίες, διάφορα εργαλεία για σχεδίαση χαρακτήρων και ρούχων για χαρακτήρες, καθώς και κινούμενων γραφικών, προσομοίωσης καπνού και υγρών. Χαρακτηριστικό του είναι η τρισδιάστατη σχεδίαση. Επίσης υποστηρίζεται από τη γλώσσα προγραμματισμού Python για εσωτερικό scripting. (Blender Wikipedia, 2021)

Κεφάλαιο 3

Σκοπός και στόχοι

Σκοπός της κατασκευής του προσομοιωτή, είναι η απεικόνιση ενός περιβάλλον βιομηχανικής εγκατάστασης παραγωγής ηλεκτρικής ενέργειας όσο το δυνατόν πιο ρεαλιστικά.

Στον προσομοιωτή που κατασκευάστηκε δεν σχεδιάστηκαν όλα τα βοηθητικά συστήματα μιας τέτοιας εγκατάστασης διότι για μια τέτοια υλοποίηση θα απαιτούσε πολύ χρόνο και σχέδια από πραγματική εγκατάσταση τα οποία δεν ήταν δυνατόν να αντληθούν.



Εικόνα 6 - Εικόνα από προσομοιωτή καραβιού σε 4D, πηγή: wartsila.com

Ο προσομοιωτής που κατασκευάστηκε απεικονίζει το περιβάλλον της εγκατάστασης σε μια οθόνη ηλεκτρονικού υπολογιστή με την δυνατότητα κίνησης στον τρισδιάστατο χώρο με την χρήση του πληκτρολογίου και ποντικιού. Στον χρήστη δίνεται η δυνατότητα αλληλεπίδρασης με τα συστήματα όπως για παράδειγμα η εκκίνηση μιας αντλίας.

Απαιτήσεις εξοπλισμού

Γενικά είναι μια ελαφριά μηχανή οι οποία δεν έχει μεγάλες απαιτήσεις στον εξοπλισμό.

Οι ελάχιστες απαιτήσεις για τον Unity3D (Editor) ο οποίος χρησιμοποιήθηκε στην παρούσα εργασία στα λειτουργικά συστήματα Windows, MacOS και Linux είναι οι παρακάτω.



Unity Editor system requirements

This section lists the minimum requirements to run the Unity Editor. Actual performance and rendering quality may vary depending on the complexity of your project.

Minimum requirements	Windows	macOS	Linux (Support in Preview)
Operating system version	Windows 7 (SP1+) and Windows 10, 64-bit versions only.	High Sierra 10.13+	Ubuntu 20.4, Ubuntu 18.04, and CentOS 7
CPU	x64 architecture with SSE2 instruction set support.	x64 architecture with SSE2 instruction set support.	x64 architecture with SSE2 instruction set support.
Graphics API	DX10, DX11, and DX12-capable GPUs.	Metal-capable Intel and AMD GPUs.	OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs.
Additional requirements	Hardware vendor officially supported drivers.	Apple officially supported drivers.	Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver or AMD Mesa graphics driver. Other configuration and user environment as provided stock with the supported distribution (Kernel, Compositor, etc.)

For all operating systems, the Unity Editor is supported on workstations or laptop form factors, running without emulation, container or compatibility layer.

Πίνακας 1 - Ελάχιστες απαιτήσεις Unity 3D Editor

Ελάχιστες απαιτήσεις εφαρμογής που γίνεται εξαγωγή (Build) .

Desktop

Operating system	Windows	Universal Windows Platform	macOS	Linux
Operating system version	Windows 7 (SP1+) and Windows 10	Xbox One, HoloLens	High Sierra 10.13+	Ubuntu 20.4, Ubuntu 18.04, and CentOS 7
CPU	x86, x64 architecture with SSE2 instruction set support.	x86, x64 architecture with SSE2 instruction set support, ARM, ARM64.	Apple Silicon, x64 architecture with SSE2.	x64 architecture with SSE2 instruction set support.
Graphics API	DX10, DX11, DX12 capable.	DX10, DX11, DX12 capable GPUs.	Metal capable Intel and AMD GPUs.	OpenGL 3.2+, Vulkan capable.
Additional requirements	Hardware vendor officially supported drivers. For development: IL2CPP scripting backend requires Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.	Hardware vendor officially supported drivers. For development: Windows 10 (64-bit), Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.	Apple officially supported drivers. For development: IL2CPP scripting backend requires Xcode. Targeting Apple Silicon with IL2CPP scripting backend requires macOS Catalina 10.15.4 and Xcode 12.2 or newer.	Gnome desktop environment running on top of X11 windowing system. Other configuration and user environment as provided stock with the supported distribution (such as Kernel or Compositor). Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver.

For all operating systems, the Unity Player is supported on workstations, laptop or tablet form factors, running without emulation, container or compatibility layer.

Πίνακας 2 - Ελάχιστες απαιτήσεις Η/Υ για να τρέξει η εφαρμογή που γίνεται κτίσιμο (build).



Μεθοδολογία

Τα σχέδια που σχεδιαστήκαν δεν ανταποκρίνονται πλήρως σε μια πραγματική βιομηχανική εγκατάσταση παραγωγής ηλεκτρικής ενέργειας με αμμοστρόβιλο. Μια τέτοια εγκατάσταση απαιτεί περαιτέρω βοηθητικά συστήματα για την ορθή λειτουργία της και αυτό στην περίπτωση της παρούσας εργασίας θα απαιτούσε περισσότερο χρόνο από τους έξι μήνες που προϋποθέτει η υλοποίηση της εργασίας.

Στο κεφάλαιο αυτό θα αναφερθούν τα προγράμματα που χρησιμοποιήθηκαν για την υλοποίηση του προσομοιωτή. Τα λογισμικά που χρησιμοποιήθηκαν προσφέρονται δωρεάν σε φοιτητές και καθηγητές μόνο για εκπαιδευτικούς σκοπούς. Σε ένα πραγματικό έργο τα σχέδια θα προσφέρονταν υλοποιημένα και πιθανόν το μόνο που θα έπρεπε να επεξεργαστεί ο κατασκευαστής του προσομοιωτή θα ήταν τα γραφικά με την χρήση του Blender ή άλλου παρόμοιου λογισμικού.

Γενικά

Το τελικό αποτέλεσμα που θα χρειαστεί για να είναι συμβατά τα σχέδια με την Unity 3D απαιτεί μορφοποίηση αρχείου «FBX». Η «FBX» είναι η πιο διαδεδομένη μορφοποίηση για τρισδιάστατα σχέδια. Στο AutoCAD Plant 3D τα σχέδια που σχεδιαστήκαν μορφοποιήθηκαν και έγιναν εισαγωγή-σύνδεση «Link CAD» στο Revit που δίνεται η δυνατότητα εξαγωγής από την εφαρμογή εξαγωγή «Export» του τρισδιάστατου σχεδίου σε μορφοποίηση «FBX».

Σε περίπτωση που υπάρχουν συνεχής αλλαγές σχεδίων ή σχεδίαση περαιτέρω στοιχείων της εγκατάστασης, η δυνατότητα σύνδεσης σχεδίων όπως για παράδειγμα στο Revit είναι η επιλογή «Link CAD» όπως φαίνεται και στην εικόνα 7. Αυτή η λειτουργία δίνει ένα μεγάλο πλεονέκτημα το οποίο είναι όταν γίνεται μια αλλαγή για παράδειγμα στο έργο «Project» που έχει δημιουργηθεί στο AutoCAD Plant 3D οι αλλαγές ενσωματώνονται και στο έργο «Project» της Revit αλλά δεν γίνεται να κάνουμε αλλαγή με το πρόγραμμα το οποίο έχει γίνει σύνδεση. Επίσης μπορεί να χρησιμοποιηθεί και από άλλες εφαρμογές της Autodesk και σχεδόν όλες έχουν την δυνατότητα σύνδεσης μεταξύ τους.



Εικόνα 7 - Σύνδεση αρχείων με μορφοποίηση CAD

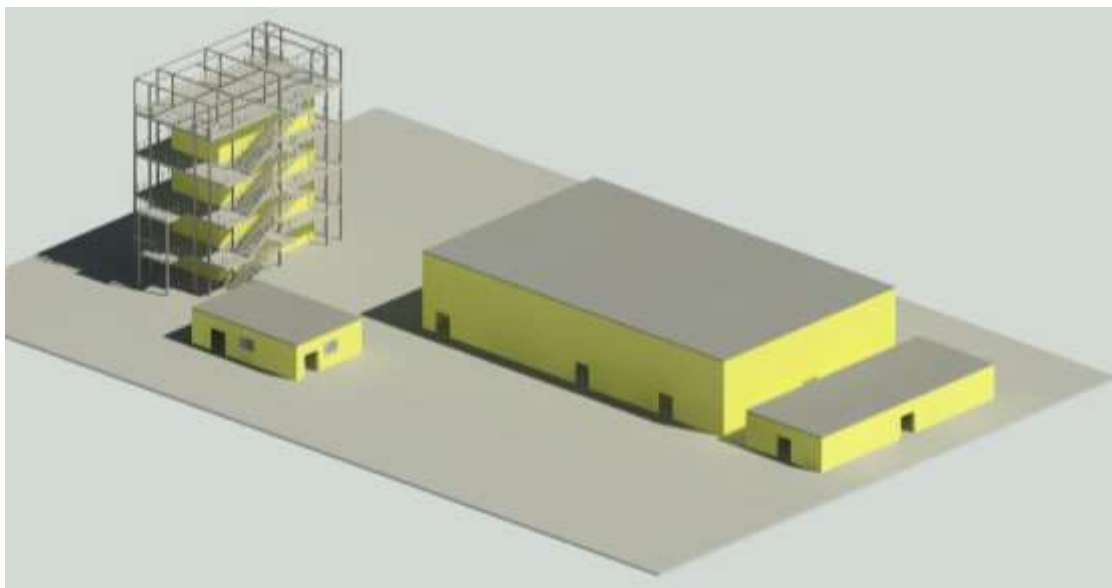


Αυτή η μέθοδος χρησιμοποιείται και από μεγάλες ομάδες, που αναλαμβάνουν ξεχωριστές εργασίες σε διαφορετικές εφαρμογές και υπάρχει ανάγκη ενσωμάτωσης σε μια εφαρμογή και αυτό επιτυγχάνεται αποθηκεύοντας τα «Project» σε ένα σύννεφο «Cloud» ή ακόμα και σε έναν κοινό διακομιστή. Θα μπορούσε να χρησιμοποιηθεί η εφαρμογή Unity Reflect αντί αυτής της μεθόδου οι οποία απαιτεί μεγαλύτερη εργασία και βήματα για την συγχώνευση των σχεδίων αλλά λόγω κόστους δεν χρησιμοποιήθηκε.

Revit

Σχεδιασμός

Στο προσομοιωτή που κατασκευάστηκε χρησιμοποιήθηκε το Revit για τον σχεδιασμό των μεταλλικών κατασκευών και δομικών στοιχείων (π.χ. ο λέβητας και τα κτίρια) και την συγχώνευση των σχεδίων από το AutoCAD Plant 3D. Στην παρακάτω εικόνα 8 φαίνεται η εγκατάσταση που έχει σχεδιαστεί στην εφαρμογή για τον προσομοιωτή.



Εικόνα 8 - - Revit 2020, σχέδιο εγκατάστασης προσομοιωτή , εξαγωγή εικόνας με την μέθοδο Render Render

Στον λέβητα χρησιμοποιήθηκαν μεταλλικοί δοκοί για την υλοποίηση του, στους οποίους το πρόγραμμα δίνει την δυνατότητα επιλογής τυποποιημένου χαρακτηριστικού όπως τις διαστάσεις και το υλικό κατασκευής. Στην παρακάτω εικόνα 9 απεικονίζεται η δυνατότητα ανάλυσης που μπορεί να διαθέσει αλλά λόγω μεγάλων απαιτήσεων που απαιτεί από τον Η/Υ δεν χρησιμοποιήθηκε στον προσομοιωτή που κατασκευάστηκε. Αυτά τα στοιχεία επιλέχτηκαν να μην είναι ορατά στο Revit. Αυτή η δυνατότητα δίνει την επιλογή ποια



στοιχεία να μην μεταφερθούν κατά την εξαγωγή σε μορφοποίηση «FBX» στην οποία θα αναφερθεί πιο αναλυτικά παρακάτω.

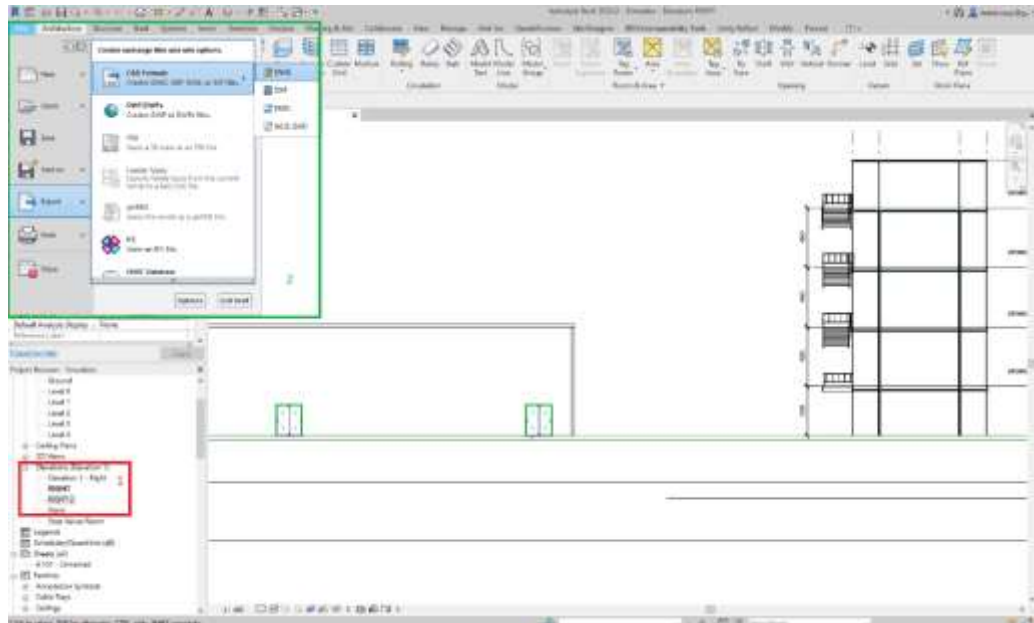


Εικόνα 9 - Στο Revit μεταλλικοί δοκοί με συνδέσμους στον λέβητα (Rendering)

Πριν γίνει εξαγωγή σε μορφοποίηση «FBX» του σχεδίου, η επιλογή που δίνει η εφαρμογή να είναι ορατά ή όχι στοιχεία στο γραφικό περιβάλλον που απεικονίζεται το σχέδιο. Έτσι όταν γίνει εξαγωγή σε μορφοποίηση «FBX» δεν θα συμπεριλαμβάνει τα συγκεκριμένα στοιχεία. Αυτό είναι ένα πλεονέκτημά όταν δεν υπάρχει επιθυμία μεταφοράς συγκεκριμένων λεπτομερειών στο τρισδιάστατο σχέδιο. Στην παρακάτω εικόνα 10 απεικονίζονται κάποιες από τις επιλογές που δίνονται. Στον προσομοιωτή που κατασκευάστηκε δεν συμπεριληφθήκαν τα παρακάτω στοιχεία:

- Structural Connections , διαρθρωτικές συνδέσεις στις μεταλλικούς δοκούς.
- Windows , παράθυρα.
- Doors, πόρτες

- Σύνδεση στο AutoCAD Plant 3D , το οποίο θα εξηγηθεί πιο αναλυτικά στην ενότητα του AutoCAD Plant 3D

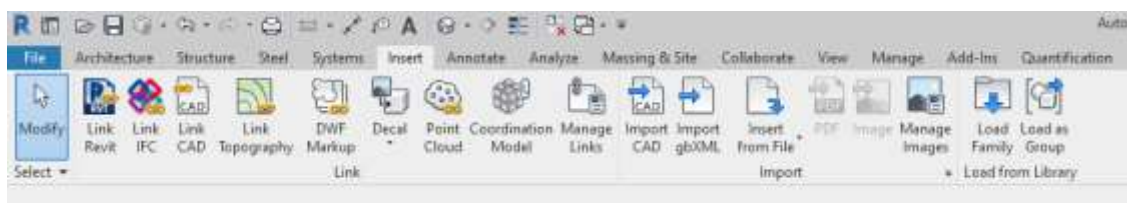


Εικόνα 11 - Εξαγωγή πρόσοψης "RIGHT" σε μορφοποίηση «CAD»

Στην παραπάνω εικόνα 11 απεικονίζεται η διαδικασία εξαγωγής σε «DWG» αρχείο. Στην κόκκινη περιοχή (1) είναι οι πρόσοψες δυο διαστάσεων που έχουν κατασκευαστεί. Στην συγκεκριμένη περίπτωση επιλέχτηκε το «RIGHT». Στην πράσινη περιοχή (2) είναι το μενού για να γίνει η εξαγωγή.

Εισαγωγή σχεδίων από AutoCAD Plant 3D

Εφόσον έχουν σχεδιαστεί τα επιθυμητά σχέδια στο AutoCAD Plant 3D θα πρέπει να τα εισάγουμε στο Revit. Αυτή την δυνατότητα δίνεται από το μενού «Insert» και την επιλογή «Link CAD» όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 12 - Μενού "Insert" στο Revit.



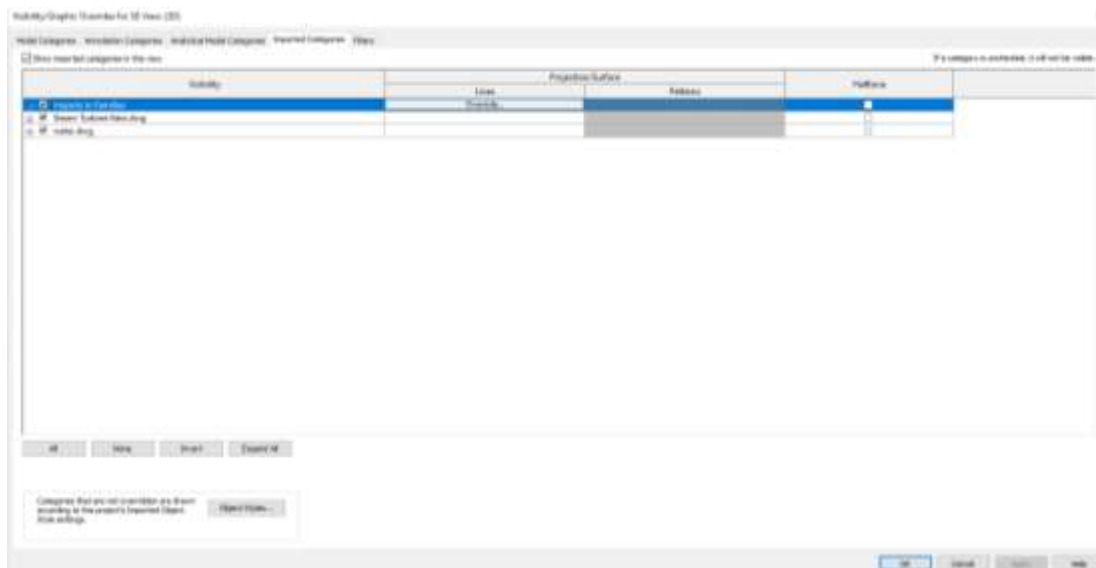
Αυτό για να γίνει προϋποθέτει μια μορφοποίηση αρχείου «AutoCAD Native Drawing» το οποίο θα αναφερθεί στην ενότητα του AutoCAD Plant 3D λόγω ότι η επεξεργασία γίνεται στην συγκεκριμένη εφαρμογή.

Διαμορφώνονται τα σχέδια στο έργο του Revit από την εφαρμογή AutoCAD Plant 3D. Έτσι τώρα είναι έτοιμα για την εξαγωγή σε μορφοποίηση «FBX».

Εξαγωγή σε μορφοποίηση «FBX»

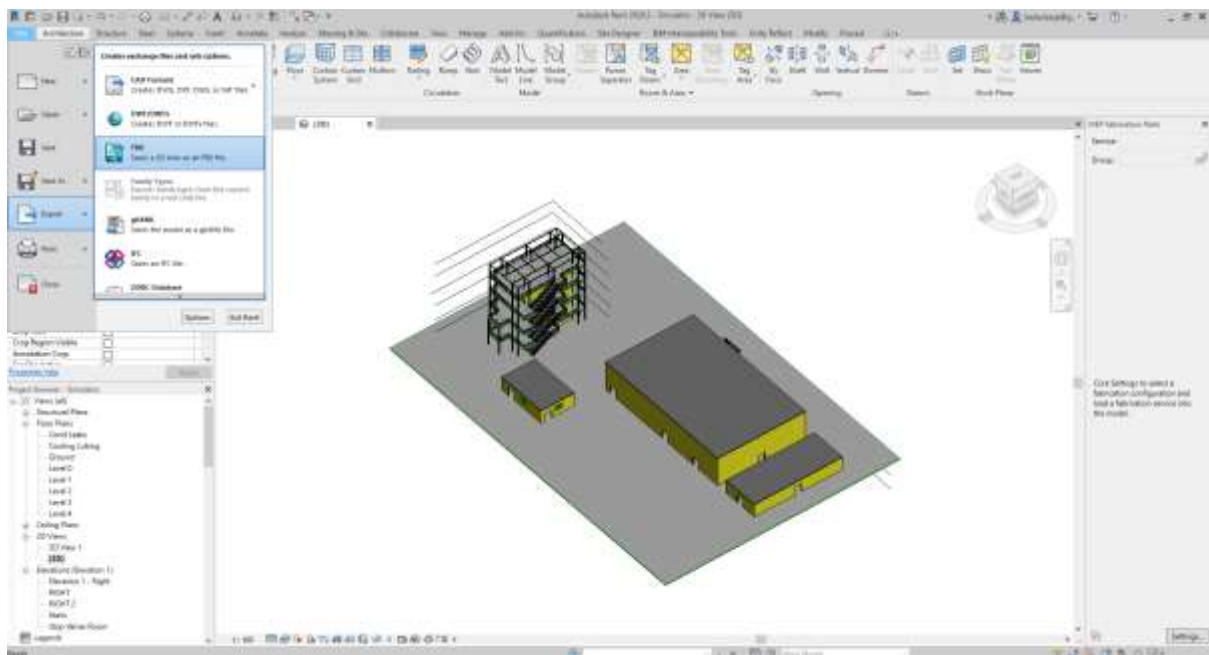
Εφόσον έχει υλοποιηθεί το σχέδιο έγινε εξαγωγή σε «FBX» για να είναι συμβατό με τις εφαρμογές Unity3D και Blender στο οποίο και θα δημιουργηθούν τα γραφικά .

Τα σχέδια τα οποία έγιναν εισαγωγή από την εφαρμογή AutoCAD Plant 3D πρέπει να είναι ενεργοποιημένα από την επιλογή «Visibility/Graphic Overrides for 3D View». Στην καρτέλα «Imported Categories» εμφανίζονται όλα τα σχέδια τα έχουν συνδεθεί με το Revit και τα οποία πρέπει να είναι ενεργοποιημένα σε περίπτωση που δεν είναι.



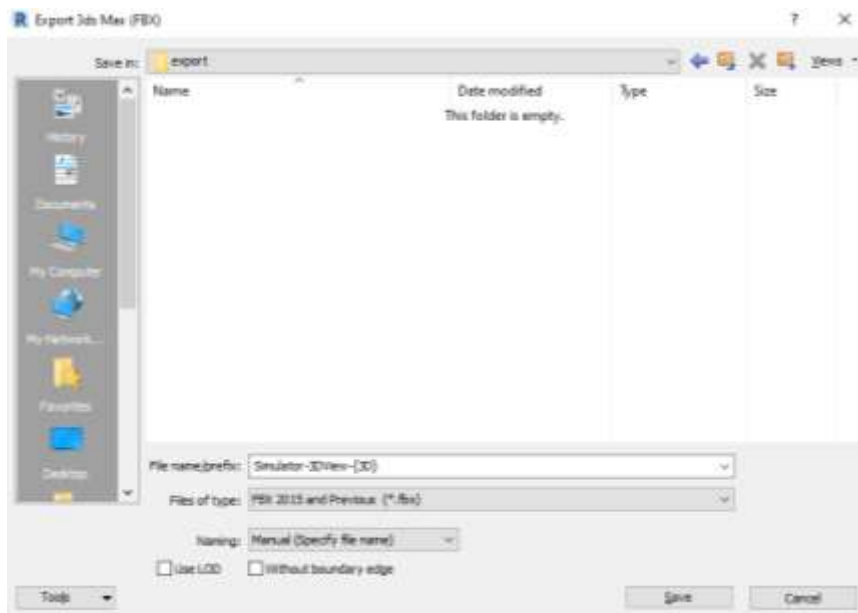
Εικόνα 13 - Συνδεμένα σχέδια στο Revit

Θα γίνει εξαγωγή του τελικού σχεδίου για τον προσομοιωτή από το μενού «File>Export>FBX». Για να γίνει αυτό πρέπει να είναι ανοικτή η σκηνή που υπάρχει επιθυμία να γίνει εξαγωγή. Στην περίπτωση αυτή είναι ανοικτή η «{3D}». Επιλέγοντας το «FBX» γίνεται εξαγωγή του σχεδίου στην μορφοποίηση. Στην παρακάτω εικόνα 14 απεικονίζεται ο τρόπος εξαγωγής.



Εικόνα 14 - Εξαγωγή αρχείου σε μορφοποίηση "FBX"

Εφόσον αποθηκευτεί, το τρισδιάστατο σχέδιο είναι έτοιμο για την υλοποίηση γραφικών χαρακτηριστικών στο Blender. Θα αναφερθεί η διαδικασία πιο αναλυτικά στην αντίστοιχη ενότητα.



Εικόνα 15 - Αποθήκευση του σχεδίου



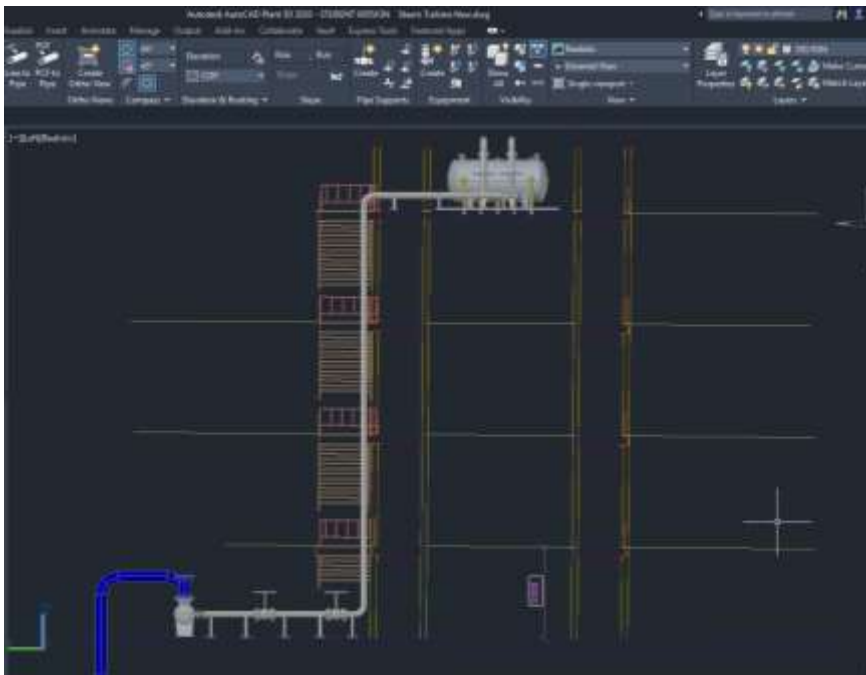
Autocad Plant 3D

Σχεδιασμός

Η συνεισφορά της εφαρμογής AutoCAD Plant 3D στην κατασκευή του προσομοιωτή ήταν στον σχεδιασμό του εξοπλισμού της εγκατάστασης. Ο εξοπλισμός περιλαμβάνει:

- Σωληνώσεις και τα βοηθητικά τους όπως επιστόμια και στηρίγματα.
- Αντλίες
- Δεξαμενές
- Εναλλάκτες Θερμότητας

Για να επιτευχθεί ο ορθός σχεδιασμός μεταξύ δυο διαφορετικών εφαρμογών , απαιτείται σύνδεση «Link CAD» δυσδιάστατης πρόσοψης από την εγκατάσταση που σχεδιάστηκε στο Revit όπως προαναφέρθηκε στην ενότητα Revit «Εξαγωγή δυσδιάστατων σχεδίων». Θα μπορούσε να χρησιμοποιηθεί και τρισδιάστατο σχέδιο αλλά προτιμήθηκε δυσδιάστατο. Στην παρακάτω εικόνα 16 απεικονίζεται ένα παράδειγμα.

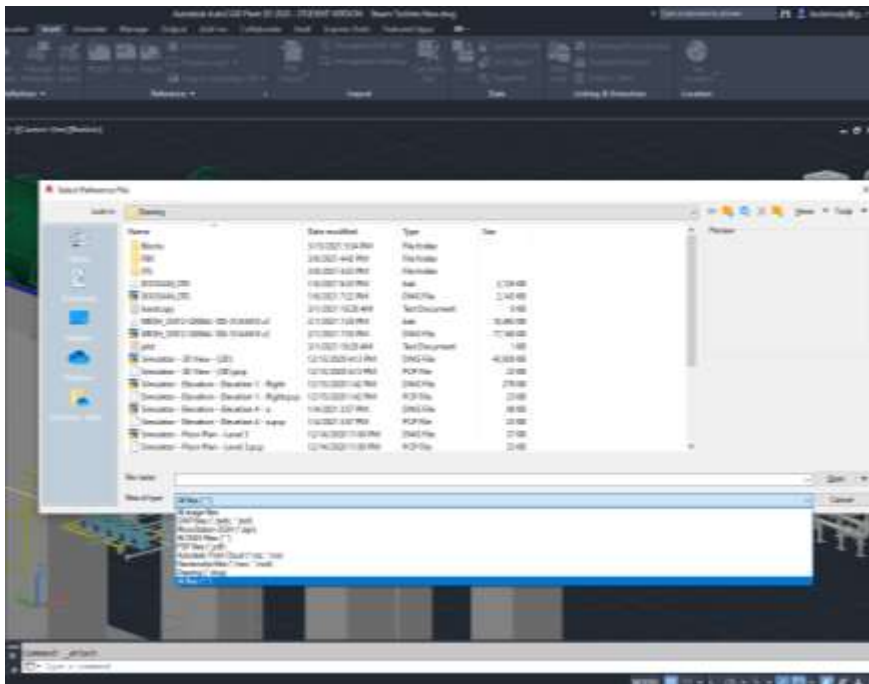


Εικόνα 16 - Απεικόνιση δυσδιάστατου σχεδίου από το Revit στο Plant 3D

Εισαγωγή δυσδιάστατου σχεδίου

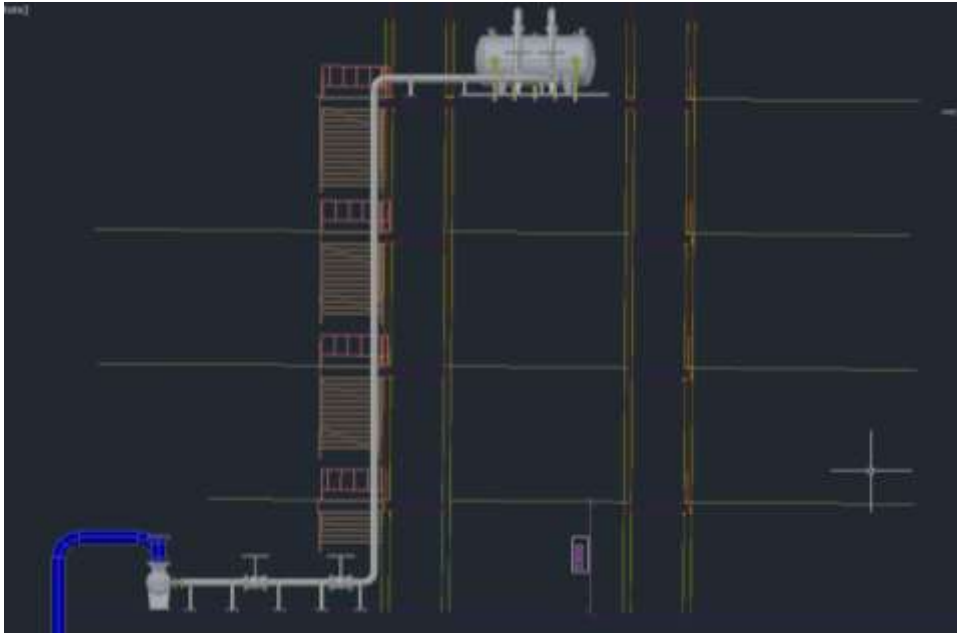
Εφόσον υλοποιήθηκε η διαδικασία εξαγωγής του δυσδιάστατου σχεδίου από την εφαρμογή Revit θα γίνει εισαγωγή στην εφαρμογή από το μενού «Insert». Επιλέγοντας το εικονίδιο

«Attach» θα εμφανιστεί το παράθυρο επιλογής του σχεδίου και οι μορφοποιήσεις όπως απεικονίζονται στην παρακάτω εικόνα 17.



Εικόνα 17 - Απεικονίζεται σύνδεση σχεδίων στο AutoCAD Plant 3D

Έπειτα το δισδιάστατο σχέδιο διαμορφώνεται στους τρισδιάστατους άξονες όπου είναι αναγκαίο με το εργαλείο τρισδιάστατης περιστροφής ή αλλιώς με την εντολή «3DROTATE». Στην παρακάτω εικόνα 18 απεικονίζεται ένα μέρος του τρισδιάστατου σχεδίου όπου σχεδιάστηκε στην εφαρμογή και το δυσδιάστατο σχέδιο σαν βοηθητικό για τον ορθό σχεδιασμό. Αυτό είναι ένα παράδειγμα για την μέθοδο που χρησιμοποιήθηκε. Χρησιμοποιήθηκαν και αλλά δυσδιάστατα σχέδια όπως κατόψεις, πλάγιες όψεις τα οποία δεν παρουσιάζονται. Δεν υπάρχει κάποιος βασικός κανόνας για αυτή την μέθοδο. Είναι στην κρίση του καθενός πως θα το χρησιμοποιήσει.



Εικόνα 18 - Απεικονίζεται το τρισδιάστατο σχέδιο στο AutoCAD Plant 3D και συνδεδεμένο δισδιάστατο σχέδιο από το Revit.

Εξαγωγή σχεδίου σε μορφή CAD

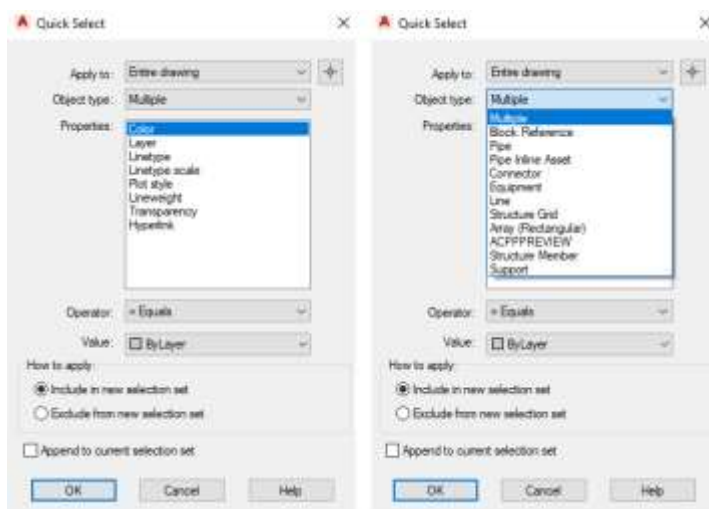
Εφόσον υλοποιήθηκε ο σχεδιασμός θα πρέπει το σχέδιο να εξαχθεί σε μορφοποίηση «AutoCAD Native Drawing» το οποίο είναι συμβατό με την εφαρμογή Revit. Η διαδικασία που εκτελέστηκε είναι η ακόλουθη.

Εφόσον έχουν υλοποιηθεί τα παραπάνω βήματα που προαναφερθήκαν, το σχέδιο είναι στο τελικό του στάδιο ώστε να γίνει εξαγωγή «Export» σε «CAD» μορφοποίηση. Τα βήματα για να επιτευχθεί αυτό από γραμμή εντολών «COMMAND». Οι εντολές είναι οι εξής:

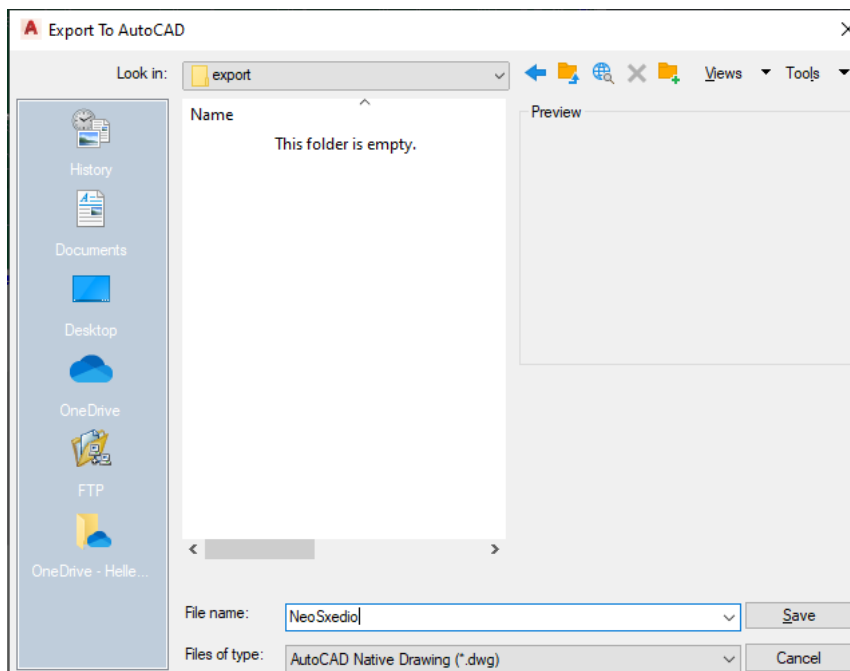
- **QSELECT** , ανοίγει ένα παράθυρο στο οποίο στην παράμετρο «Operator» γίνεται επιλογή «Select all» και έτσι επιλέγονται όλα τα αντικείμενα στο σχέδιο. Επιπλέον αυτή η εντολή δίνει την δυνατότητα επιλογής αντικείμενων με συγκεκριμένα χαρακτηριστικά ακόμα και επιλογή επιπέδων «layers» του σχεδίου. Οι επιλογές που δίνονται φαίνονται στην εικόνα 19.
- **EXPORTTOAUTOCAD**, μια πολύ χρήσιμη εντολή σε περίπτωση που υπάρχει ανάγκη μεταφοράς μόνο των βασικών αντικειμένων (στην περίπτωση μας τρισδιάστατο σχέδιο). Ουσιαστικά αυτή η εντολή «καθαρίζει» τα προσθετά αντικείμενα που προσθέτει η AutoCAD Plant 3D που βοηθάνε στον σχεδιασμό και το σχέδιο γίνεται συμβατό με άλλες εφαρμογές που αναγνωρίζουν «CAD» μορφοποιήσεις (στην περίπτωση μας το Revit) και καθιστά την νέα έκδοση συμβατή με διάφορες εφαρμογές σαν το AutoCAD.



Εφόσον έγιναν τα παραπάνω βήματα το καινούριο σχέδιο που αποθηκεύτηκε με διαφορετικό όνομα μπορεί να γίνει εισαγωγή σε οποιαδήποτε εφαρμογή χωρίς σφάλματα στα γεωμετρικά χαρακτηριστικά, εφόσον υποστηρίζει την συγκεκριμένη μορφοποίηση. Να δοθεί προσοχή ότι αν αποθηκευτεί σε ίδια διεύθυνση και με το ίδιο όνομα με το σχέδιο που υλοποιείται στην εφαρμογή θα αντικαταστήσει το αρχείο και θα χαθούνε τα προσθετά αντικείμενα που προσαρμόζει η εφαρμογή για τις λειτουργίες που παρέχει στον σχεδιασμό.



Εικόνα 19 - Εντολή "QSELECT"



Εικόνα 20 - Αποθήκευση νέου σχεδίου με την εντολή "EXPORTTOAUTOCAD"

Να αναφερθεί ότι τα παραπάνω βήματα είναι αναγκαία για τον εξής λόγο:

Η εφαρμογή χρησιμοποιεί βοηθητικά αντικείμενα τα οποία δεν είναι εμφανή για τις επιπλέον δυνατότητες σχεδιασμού που έχει από την αρχική εφαρμογή AutoCAD. Αν τυχόν δοκιμαστεί να ανοίξει το σχέδιο σε μια άλλη έκδοση όπως για παράδειγμα στο AutoCAD θα ήταν συμβατά αλλά πιθανόν το σχέδιο να έβγαζε σφάλματα σε σχέση με το αρχικό.

Κατά κύριο λόγο η εντολή «EXPORTTOAUTOCAD» μας δίνει την συμβατότητα με άλλες έκδοσης όπως AutoCAD, Revit ακόμα και με παλαιότερες εκδόσεις. Έτσι δημιουργεί μια νέα έκδοση αρχείου «DRW» με νέα μορφοποίηση, στην οποία αν εφαρμοστούν αλλαγές στην κανονική έκδοση δεν επιφέρει αλλαγές στην νέα έκδοση εφόσον αποθηκεύεται σαν διαφορετικό αρχείο. Έτσι λοιπόν το τελικό αποτέλεσμα θα είναι ένα αρχείο με μορφοποίηση «AutoCAD Native Drawing» το οποίο θα είναι συμβατό με το Revit.

Έπειτα αυτή η νέα έκδοση του σχεδίου που έχει τροποποιηθεί σε κατάλληλη μορφοποίηση είναι συμβατή για εισαγωγή της στο Revit με την διαδικασία που προαναφέρθηκε στην ενότητα Revit.

Blender

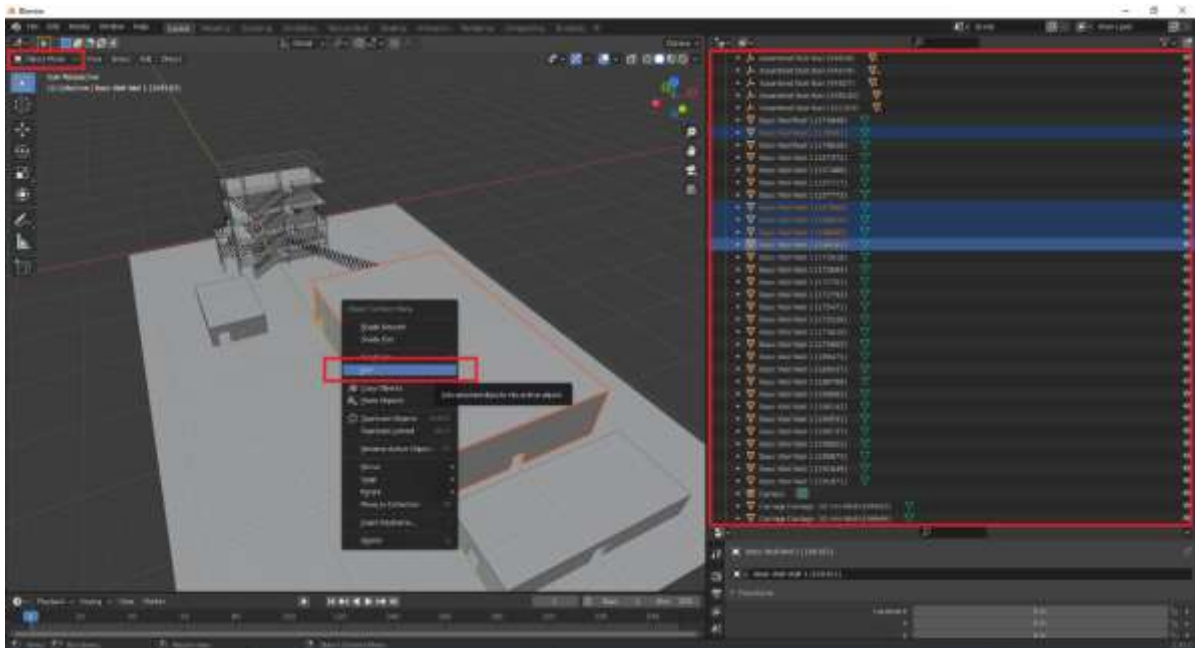
Εφόσον έχουν σχεδιαστεί όλα τα επιμέρους σχέδια στις εφαρμογές που προαναφερθήκαν στις ενότητες Revit, AutoCAD Plant 3D συγχωνεύονται όλα σε ένα τρισδιάστατο σχέδιο με

μορφοποίηση «FBX». Επεξεργάστηκαν τα αντικείμενα «Objects» που προέκυψαν και συγχωνευτήκαν έτσι ώστε να μικρύνει η πληθώρα τους, το οποίο θα διευκολύνει την διαδικασία υλοποίησης γραφικών χαρακτηριστικών.

Η διαδικασία για να κατασκευαστούν τα γραφικά αρχικά είναι να συγχωνευτούν τα αντικείμενα σε επιμέρους ομάδες. Για παράδειγμα τα κτίρια τα κάνουμε ένα αντικείμενο κ.τ.λ. . Αυτό θα κάνει την υλοποίηση των γραφικών και την δυνατότητα αλλαγών μια εύκολη διαδικασία και δεν θα υπάρχουν πολλά αντικείμενα τα οποία θα δυσκολέψουν τις εργασίες στην Unity3D.

Απλοποίηση τρισδιάστατου σχεδίου με συγχώνευση αντικείμενων

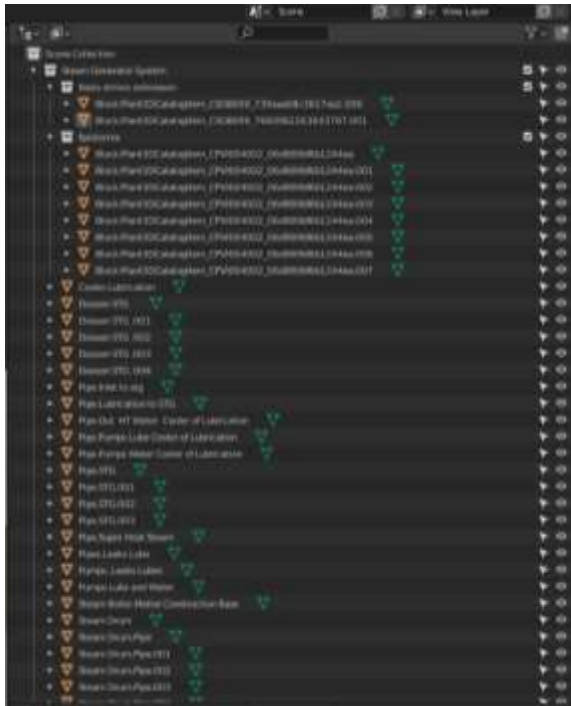
Για να γίνει η συγχώνευση αντικείμενων πρέπει να είναι επιλεγμένο το «Object Mode» όπως φαίνεται στα αριστερά της εικόνας 21. Έπειτα με δεξί κλικ επιλέγεται το «Join» ή πληκτρολόγηση συντόμευσης «CTR + J». Αυτή η εργασία θα συγχωνεύσει τα αντικείμενα που έχουν επιλεχθεί και θα γίνει ένα αντικείμενο. Η διαδικασία αυτή ακολουθείται για τα περισσότερα στοιχεία του τρισδιάστατου σχεδίου.



Εικόνα 21 - Επιλογή αντικειμένων και συγχώνευσή τους με την εντολή "Join"

Το τελικό αποτέλεσμα αυτής της συγχώνευσης των αντικείμενων πρέπει να είναι σε επιμέρους κομμάτια ανάλογα με το είδος του αντικείμενου και την τοποθεσία. Για παράδειγμα τα κάγκελα στο τρισδιάστατο σχέδιο συγχωνευτήκαν σε όσο δυνατόν μικρότερες ομάδες.

Αυτή η διαδικασία κάνει πιο εύκολη την προσθήκη γραφικών χαρακτηριστικών «Materials». Αν γινόταν συγχώνευση όλων των αντικείμεμων του τρισδιάστατου σχεδίου θα καθιστούσε δύσκολο έως αδύνατον την υλοποίηση των γραφικών χαρακτηριστικών «Materials». Στην παρακάτω εικόνα 22 απεικονίζεται το τελικό αποτέλεσμα έπειτα από συγχώνευση και ομαδοποίηση.

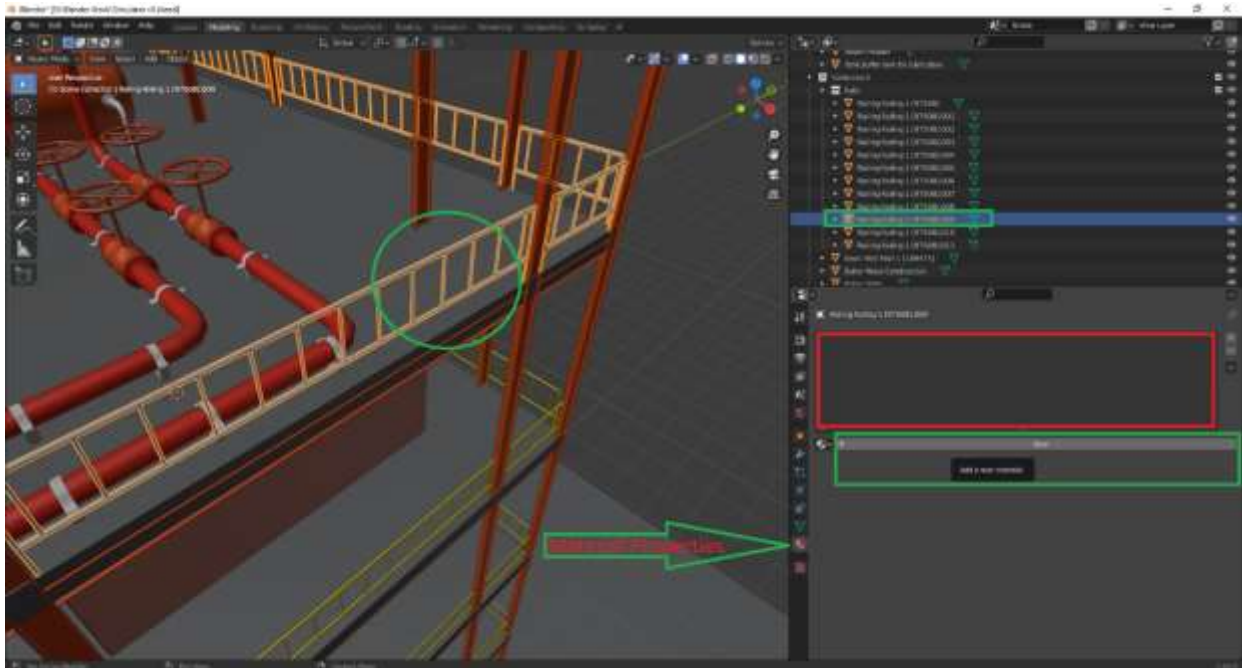


Εικόνα 22 - Απεικονίζει συγχωνεύσεων και ομαδοποίηση

Δημιουργία γραφικών χαρακτηριστικών «Materials»

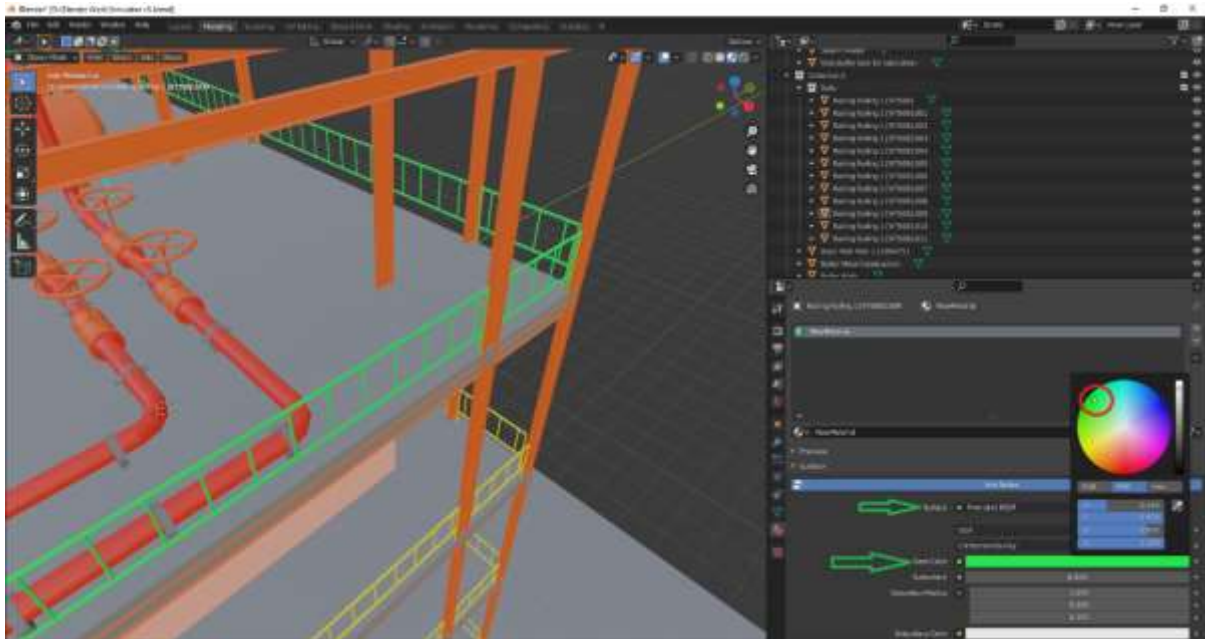
Η εφαρμογή προσφέρει πληθώρα βιβλιοθηκών για γραφικά χαρακτηριστικά. Προτιμήθηκε να χρησιμοποιηθούν τα προσφερόμενα χωρίς προσθήκη ετοιμών βιβλιοθηκών. Δηλαδή τις ήδη υπάρχουσες επιλογές που προφέρει η εφαρμογή.

Αρχικά πρέπει να δημιουργηθεί ένα «Material» για το αντικείμενο που επιλέγεται. Στην οθόνη εργασίας του Blender το αντικείμενο που επιλέχτηκε περιτριγυρίζεται από μια πορτοκαλί διαγράμμιση. Πατώντας την επιλογή «Material Properties» θα εμφανιστεί η επιλογή «New», λόγω του ότι δεν υπάρχει «Material» στο αντικείμενο. Στην παρακάτω εικόνα 23 απεικονίζονται η διαδικασία.



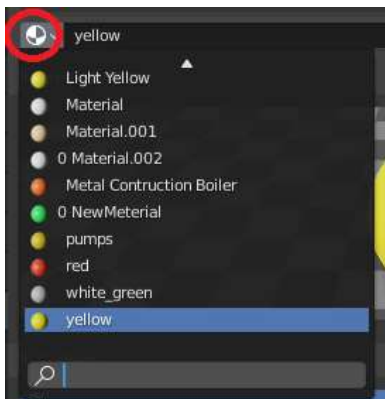
Εικόνα 23 - Προσθήκη νέου Material στην βιβλιοθήκη και στο αντικείμενο

Εφόσον έγιναν τα παραπάνω βήματα θα εμφανιστεί ένα νέο «Material» στην κόκκινη περιοχή της εικόνας 23. Όταν επιλεγθεί θα εμφανιστούν οι ιδιότητες τους οι οποίες ορίζονται όπως για παράδειγμα η επιφάνεια «Surface» που είναι και ένα από τα βασικά χαρακτηριστικά. Στην περίπτωση αυτή επιλέχτηκε για «Surface» η επιλογή «Principled BSDF» και βασικό χρώμα «Base Color» μια πράσινη απόχρωση. Στην παρακάτω εικόνα 24 απεικονίζονται τα προαναφερόμενα και φαίνεται η αλλαγή του «Material» στο αντικείμενο που επιλέχτηκε η πράσινη απόχρωση.



Εικόνα 24 - Προσθήκη ιδιοτήτων στο Material.

Η προσθήκη γραφικών σε άλλα αντικείμενα γίνεται με την ίδια διαδικασία. Αν υπάρχει επιθυμία να μπει ένα «Material» που έχει δημιουργηθεί ήδη, γίνεται επιλέγοντας το εικονίδιο όπως φαίνεται στην παρακάτω εικόνα 25.



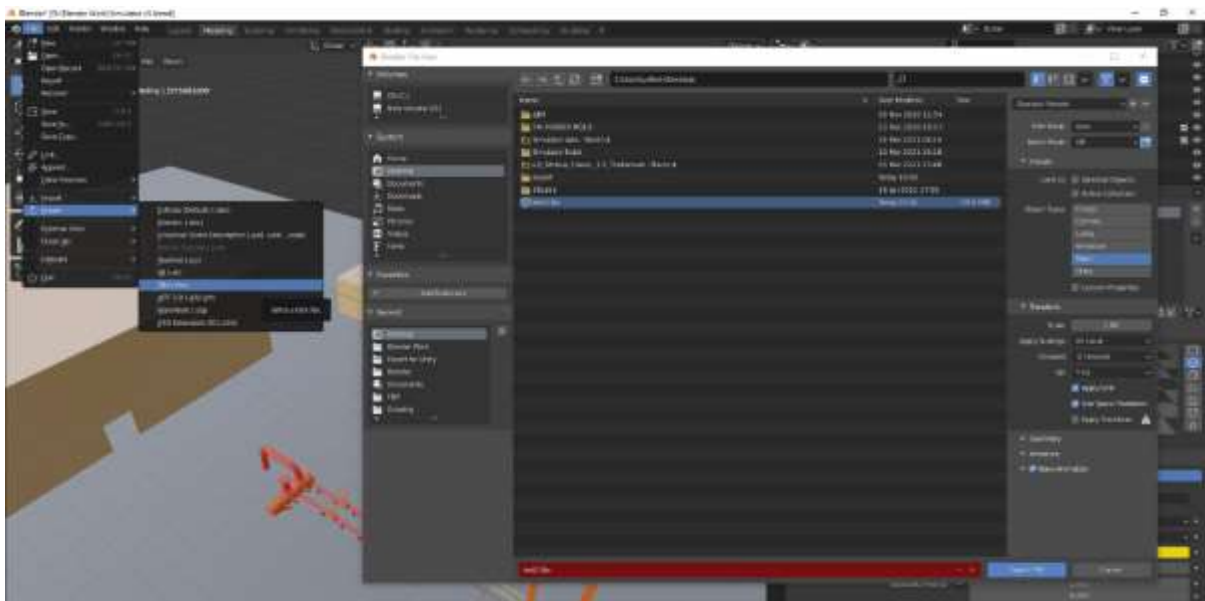
Εικόνα 25 - Επιλογή material

Τα γραφικά χαρακτηριστικά που επιλεχθήκαν «Materials» είναι απλά και δεν εμβαθύνουμε σε κάτι πιο ρεαλιστικό σε αυτά λόγω του ότι δεν είναι τόσο απαραίτητα στον προσομοιωτή παρά μόνο για τον ευκρινή διαχωρισμό των στοιχείων.



Εφόσον υλοποιήθηκαν όλα τα γραφικά χαρακτηριστικά «Materials» σε όλα τα στοιχεία και υπάρχει το επιθυμητό αποτέλεσμα το τρισδιάστατο σχέδιο είναι πλέον έτοιμο για την εισαγωγή του στην μηχανή Unity 3D Editor.

Στη διαδικασία εξαγωγής, η εφαρμογή μας δίνει την δυνατότητα να εξαχθούν φωτισμοί, κάμερες κ.τ.λ. όπως φαίνονται στην παρακάτω εικόνα 26. Στην περίπτωση αυτή επιλέγονται μόνο τα «Mesh» τα οποία είναι αναγκαία και θα βοηθήσουν στην Unity 3D Editor. Ο σκοπός τους θα αναφερθεί στην αντίστοιχη ενότητα.



Εικόνα 26 - Εξαγωγή σε μορφοποίηση «FBX»

Κάποιες δυνατότητες που προσφέρει το Blender κατά την εξαγωγή ενός τρισδιάστατου σχεδίου δεν εμφανίζονται λόγω του ότι η μορφοποίηση «FBX» δεν τις υποστηρίζει. Οι ανάγκες κατασκευής καλύπτονται με την συγκεκριμένη μορφοποίηση. Παρατηρήθηκε επίσης ότι προσθήκες συνθετών γραφικών χαρακτηριστικών «Material» δεν υποστηρίζονται από την συγκεκριμένη μορφοποίηση.



Unity 3D

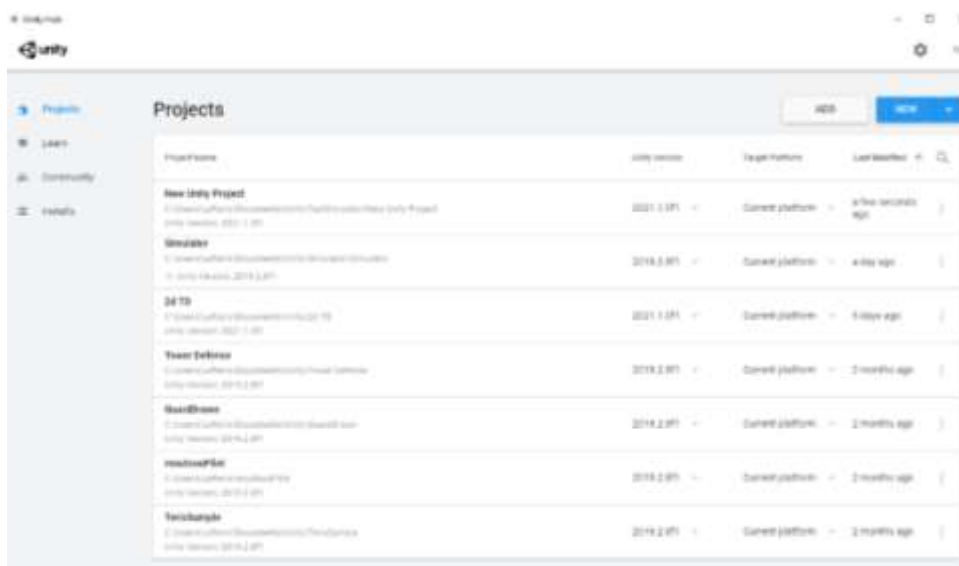
Γενικά

Υλοποιήθηκαν τα γραφικά χαρακτηριστικά στο τρισδιάστατο σχέδιο και δημιουργήθηκε ένα φιλικό αποτέλεσμα ως προς ευκρίνεια για τον χρήστη. Μπορεί το μοντέλο να γίνει εισαγωγή στην Unity 3D για την δημιουργία μιας διαδραστικής εφαρμογής. Σε αυτήν την περίπτωση είναι η προσομοίωση βιομηχανικής εγκατάστασης ηλεκτροπαραγωγής με ατμοστρόβιλο και λέβητα.

Δόθηκε ιδιαίτερη σημασία στον σχεδιασμό ώστε να είναι όσο τον δυνατόν ρεαλιστικός αλλά κυρίως να είναι ευκρινής ο εξοπλισμός της εγκατάστασης και ένας χρήστης με στοιχειώδεις γνώσεις να μπορεί να αναγνωρίσει τι είναι ο κάθε εξοπλισμός και η χρήση του.

Unity Hub

Για να τρέξει η Unity3D χρησιμοποιήθηκαν δυο τρόποι. Ο ένας είναι κατευθείαν από την εγκαταστημένη έκδοση και ο άλλος από το Unity Hub. Επίσης μπορεί να υπάρχουν παραπάνω και από μια εκδόσεις εγκαταστημένες στον υπολογιστή. Έτσι λοιπόν η βοηθητική εφαρμογή Unity Hub προσφέρει μια πλήρη εικόνα από τα έργα «Projects» που έχουν δημιουργηθεί σε οποιασδήποτε έκδοση της Unity3D και δίνει την δυνατότητα να μεταφερθεί ένα έργο σε μια νεότερη ή παλαιότερη έκδοση της. Στην παρακάτω εικόνα απεικονίζονται διάφορα έργα που έχουν δημιουργηθεί στον υπολογιστή σε διαφορετικές εκδόσεις. Επίσης δίνεται η δυνατότητα αλλαγής της πλατφόρμας (Windows,Android,Linux,PlayStation κ.τ.λ.) που στοχεύει το έργο.



Εικόνα 27 - Έργα στο Unity Hub και οι εκδόσεις

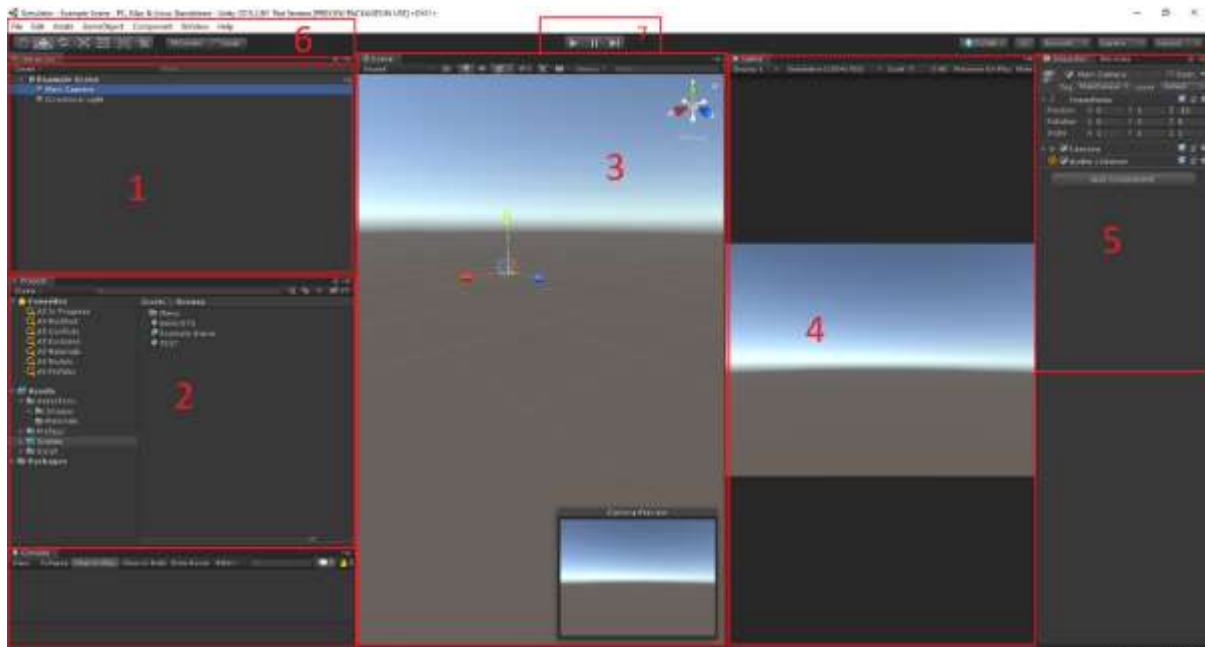
Οι πρόσθετες υπηρεσίες που προσφέρει εκτός αυτής της διαχείρισης έργων «Projects» είναι οι εξής:

- Εκμάθηση «Learn», στο οποίο περιέχει κατασκευή διαφόρων έργων που συνοδεύονται από μια ενότητα βίντεο εκμάθησης βήμα-βήμα και διάφορα βίντεο εκμάθησης «Tutorials». Για έναν εντελώς αρχάριο είναι μια σωστή κατεύθυνση εκμάθησης της μηχανής.
- Κοινότητα «Community», οι επίσημες ιστοσελίδες της εταιρίας.
- Εγκατάσταση «Installs», διαχείριση εγκαταστημένων εκδόσεων της μηχανής όπως προσθήκη «Modules».

Περιβάλλον μηχανής

Στη μηχανή, κατά την δημιουργία ενός έργου, αυτόματα θα δημιουργηθεί μια σκηνή «Scene» στην οποία μπορούν να ξεκινήσουν οι πρώτες εργασίες και δίνεται δυνατότητα μετονομασίας.

Θα αναλυθεί παρακάτω το εργασιακό περιβάλλον της μηχανής. Η αρίθμηση στην παρακάτω εικόνα 28 περιγράφεται στο κείμενο που ακολουθεί.



Εικόνα 28 - Περιβάλλον εργασίας Unity 3D Editor

- 1) Στοιχεία που έχουν δημιουργηθεί ή έχουν γίνει εισαγωγή σαν προκατασκευασμένα στοιχεία/αντικείμενα «Prefab Asset». Όπως, σκηνές, τρισδιάστατο σχέδιο, φωτισμός, κάμερες, ήχος, κώδικας. Σε περίπτωση που δημιουργηθεί μια νέα σκηνή

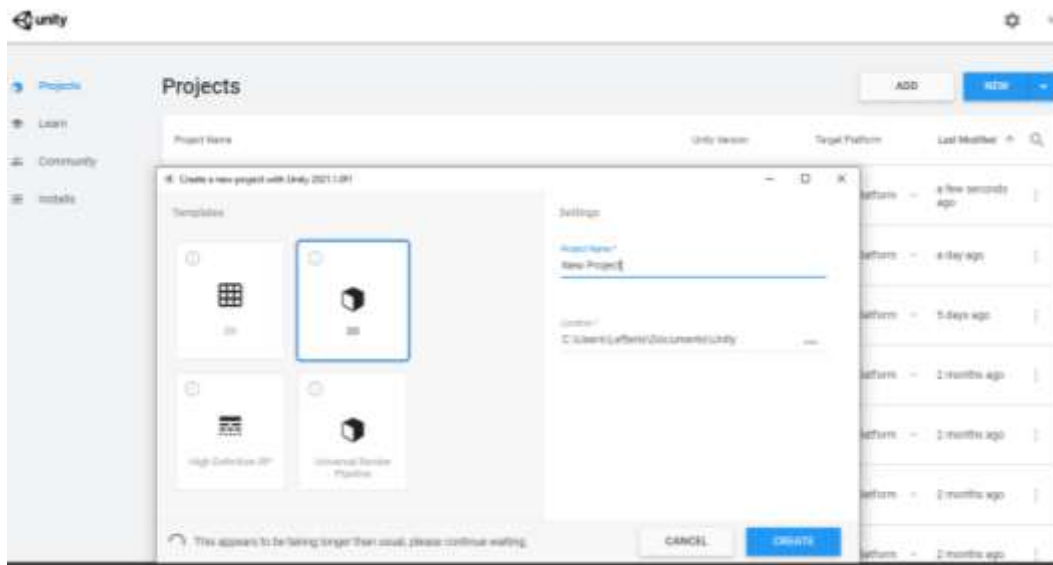


δεν μεταφέρονται τα στοιχεία/αντικείμενα της από την προηγούμενη πάρα μόνο με την επιθυμία του κατασκευαστή. Επίσης μπορούν να υλοποιηθούν παραπάνω από μια σκηνές παράλληλα. Οι σκηνές αποθηκεύονται. Φαίνονται τρεις σκηνές στο σημείο (2) που έχουν αποθηκευτεί.

- 2) Εξερευνητής έργου «Project Explorer», τρισδιάστατα σχέδια, προκατασκευασμένα στοιχεία/αντικείμενα «Prefab», κώδικας, φωτογραφίες, βίντεο, γραφικά «Materials», γραμματοσειρές και οτιδήποτε άλλο εισάγεται στο έργο αποθηκεύονται στο έργο.
- 3) Στο σημείο αυτό ουσιαστικά είναι το περιβάλλον εργασίας. Η επιλεγμένη σκηνή εμφανίζεται και σε αυτήν δίνεται η δυνατότητα επεξεργασίας. Οι δυνατότητες αυτές είναι ο σχεδιασμός χάρτη «Terrain», μετακίνηση, κλιμάκωση «Scale», περιστροφή, διαγραφή, αντιγραφή αντικειμένων/στοιχείων. Απεικονίζονται επίσης οι φωτισμοί, ήχοι, κρυφά στοιχεία/αντικείμενα (π.χ. Colliders).
- 4) Όταν τρέξει το έργο (Play). Σε αυτό το σημείο είναι η οθόνη αλληλεπίδρασης του χρήστη και συνήθως εκεί γίνονται οι δοκιμές. Είναι μεγάλο πλεονέκτημα γιατί με το πάτημα ενός κουμπιού(σημείο 7) δοκιμάζεται το έργο που κατασκευάζεται.
- 5) Όταν επιλέγεται ένα αντικείμενο/στοιχείο στην σκηνή (1) ή ένα προκατασκευασμένο στοιχείο «Prefab» στο σημείο (2), δίνεται η δυνατότητα επεξεργασίας των χαρακτηριστικών «Component» ή εισαγωγής νέου όπως για παράδειγμα κώδικας «Script». Επίσης δίνεται η δυνατότητα ονομασίας και κατηγοριοποίησης του αντικειμένου/στοιχείου από τα «Tag» και «Layer».
- 6) Γραμμή εντολών και εργασιών τα περισσότερα από τα οποία αλληλοεπιδρούν με το σημείο (3), όπως ή μετακίνηση αντικειμένων/στοιχείων που προαναφέρθηκαν στο 3.
- 7) Στο σημείο αυτό είναι τρία κουμπιά. Το πρώτο είναι για δοκιμή του έργου «Play», το δεύτερο παύση «Pause» και το τρίτο βήματα «Step». Είναι χρήσιμο όταν απαιτείται η παρατήρηση του έργου σε βήματα.

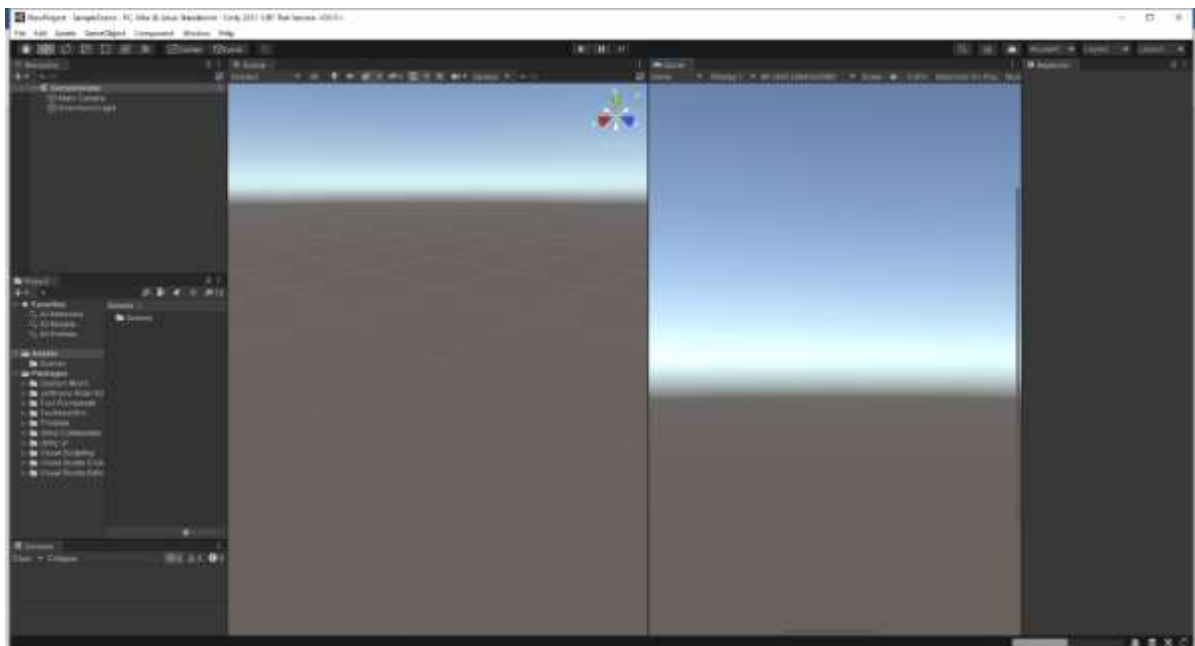
Δημιουργία έργου στο Unity Hub

Αρχικά πρέπει να δημιουργηθεί ένα νέο έργο εφόσον δεν έχει δημιουργηθεί, ανοίγοντας το Unity Hub και πατώντας την επιλογή «New» επιλέγεται η έκδοσή της Unity3D. Όσο νεότερη έκδοση τόσο καλύτερα. Για «Templates» επιλέγεται το «3D». Στην παρακάτω εικόνα απεικονίζεται η δημιουργία ενός έργου.



Εικόνα 29 - Δημιουργία έργου με επιλογή Template 3D

Εφόσον δημιουργήθηκε το έργο και άνοιξε η εφαρμογή, δημιουργείται μια σκηνή «Scene» στην οποία υπάρχουν δυο στοιχεία η «Main Camera» και το «Directional Light» τα οποία και κρατάμε λόγω του ότι θα χρησιμεύσουν αργότερα. Αυτά απεικονίζονται στην παρακάτω εικόνα 30.



Εικόνα 30 - Δημιουργία νέου έργου

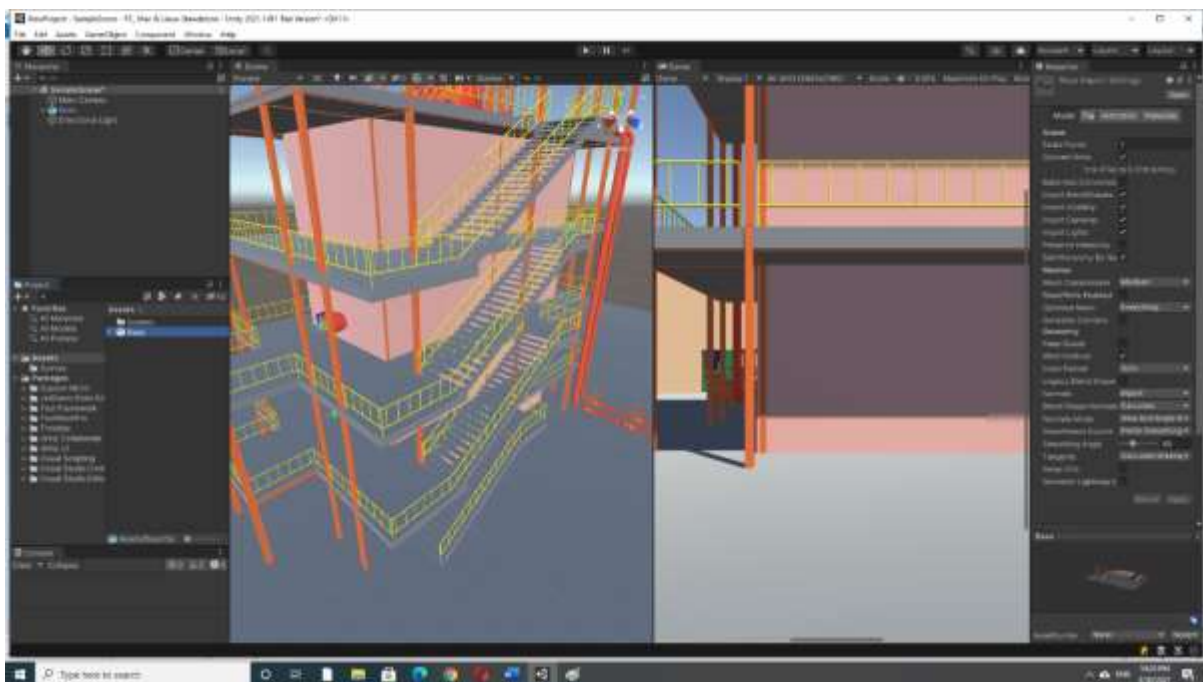


Διαμόρφωση γραφικού περιβάλλοντος

Εφόσον είμαστε σε αυτό το στάδιο εισάγουμε το τρισδιάστατο σχέδιο με μορφοποίηση «FBX» στον φάκελο Assets. Αυτό μπορεί να γίνει πατώντας δεξί κλικ και «Import Asset» και επιλέγεται το τρισδιάστατο σχέδιο ή απλά να σύρουμε το αρχείο στον φάκελο Assets. Το όνομα του τρισδιάστατου σχεδίου είναι «Base».

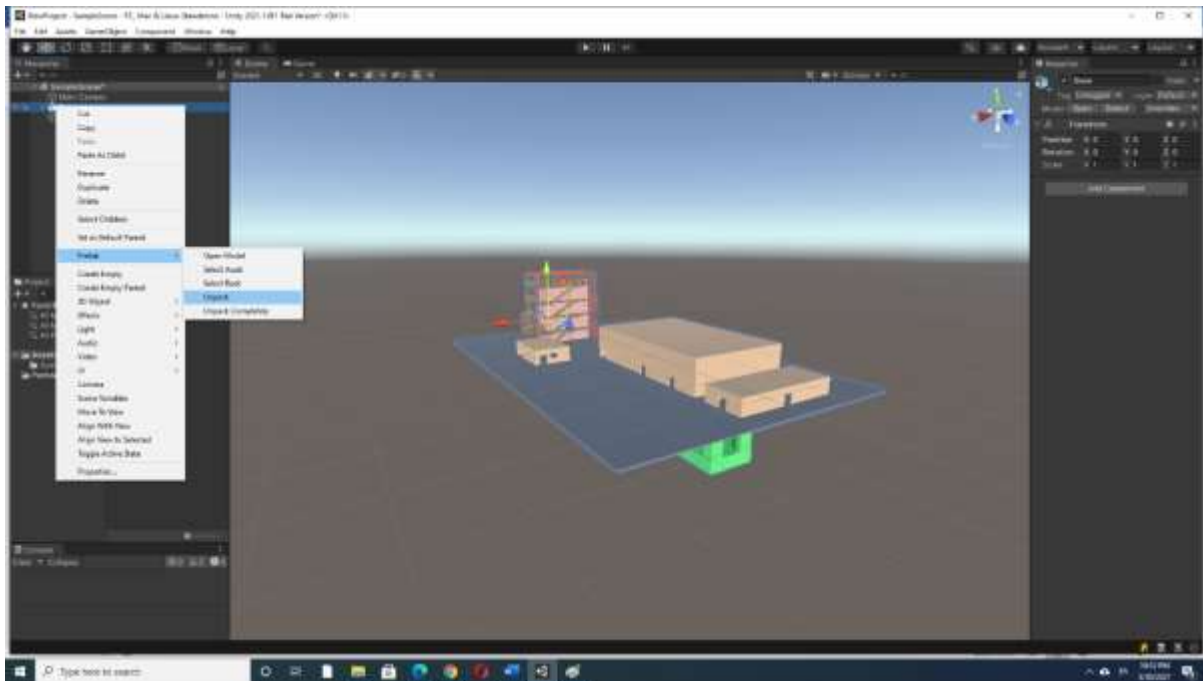
Πατώντας κλικ μια φορά στο «Base» δίνεται η δυνατότητα κάποιων ρυθμίσεων. Εμείς χρησιμοποιήσαμε το «Mesh Compression - Medium» το οποίο χαμηλώνει τις απαιτήσεις στα γραφικά. Οι όποιες αλλαγές γίνονται, εφαρμόζονται και στις σκηνές που έχει τοποθετηθεί το «Base». Ουσιαστικά όταν τοποθετείτε στην σκηνή είναι ο κλώνος του «Base» που υπάρχει στον φάκελο Assets.

Στην παρακάτω εικόνα απεικονίζεται το «Base» επιλεγμένο στο φάκελο Assets και δεξιά οι ρυθμίσεις του και το σχέδιο έχει γίνει εισαγωγή στην σκηνή.



Εικόνα 31 - Εισαγωγή τρισδιάστατου σχεδίου στην σκηνή και στον φάκελο Assets

Εξαιτίας ότι υπάρχει ανάγκη προσθετών χαρακτηριστικών και αντικειμένων στο «Base» θα γίνει μια διαδικασία «Unpack prefab» για να είναι δυνατών αυτές η προσθήκες, όπως είναι τα χαρακτηριστικά «Components» κυρίως. Δεν επηρεάζει το αρχικό που υπάρχει στον φάκελο Assets. Πατώντας το δεξί κλικ στο «Base» που είναι στην σκηνή επιλέγεται Prefab>Unpack όπως φαίνεται στην παρακάτω εικόνα 32.

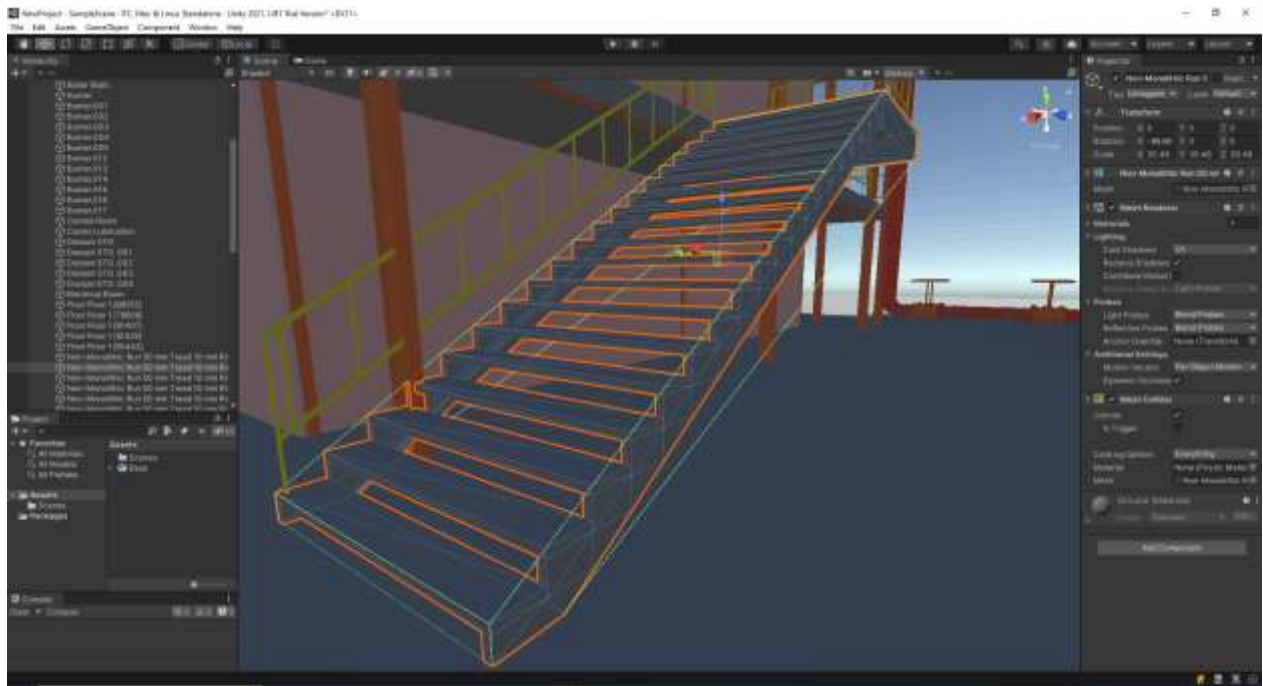


Εικόνα 32 - Διαδικασία Unpack Prefab

Για να υπάρχει ρεαλισμός στον προσομοιωτή ο χρήστης όταν περιφέρεται στο γραφικό περιβάλλον δεν θα πρέπει να περνάει μέσα από τα αντικείμενα όπως επίσης και να ανεβαίνει τα σκαλιά που υπάρχουν κτλ. Αυτή την δυνατότητα την δίνουν τα «Colliders». Η προσθήκη τους είναι απλή και σχετικά γρήγορη. .

Όπως φαίνεται στην παραπάνω εικόνα 32 στα δεξιά υπάρχει η επιλογή «Add Component» η οποία μας δίνει την δυνατότητα για προσθήκη «Colliders». Επιλέγοντας «**Physics>Mesh Collider**» θα γίνει προσθήκη ενός «Collider» το οποίο θα είναι στα γεωμετρικά χαρακτηριστικά του συγκεκριμένου αντικείμενου οπου επιλέχτηκε όπως φαίνεται στην παρακάτω εικόνα33 .

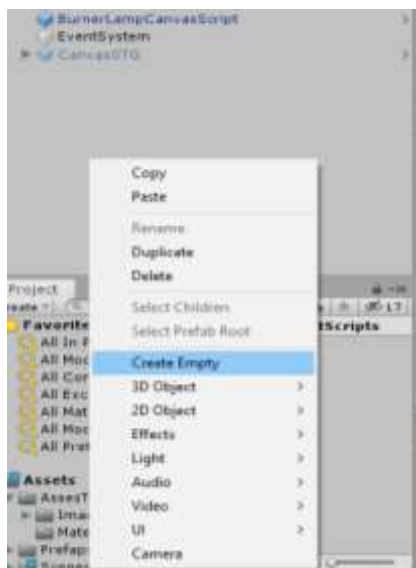
Στα δεξιά της παρακάτω εικόνας 33 φαίνεται το «Mesh Collider» που έγινε προσθήκη. Η επιλογή «Convex» διαμορφώνει το «Collider» με όσο το δυνατόν λιγότερες γωνίες και ομαλές όπως φαίνεται στην εικόνα 33 (σκαλοπάτια) με την πράσινη διαγράμμιση. Θα πρέπει να οριστεί ότι δεν θα είναι διαπερατό από αλλά «Colliders» άρα η επιλογή «Trigger» να μην είναι επιλεγμένη ή false . Η χρησιμότητα τους είναι όταν χρήστης περιηγείται στον περιβάλλον τα «Collider» θα ορίζουν τα χαρακτηριστικά της επιφάνειας αν είναι διαπερατή ή όχι.



Εικόνα 33 - Εισαγωγή Collider στα σκαλοπάτια

Εφόσον τελειώσει αυτή η διαδικασία προσθήκης «Colliders» στο γραφικό περιβάλλον τα οποία δεν είναι διαπερατά γίνεται προσθήκη φωτισμού στο περιβάλλον.

Η δημιουργία του αντικείμενου χωρίς χαρακτηριστικά «Empty» γίνεται πατώντας δεξί κλικ.

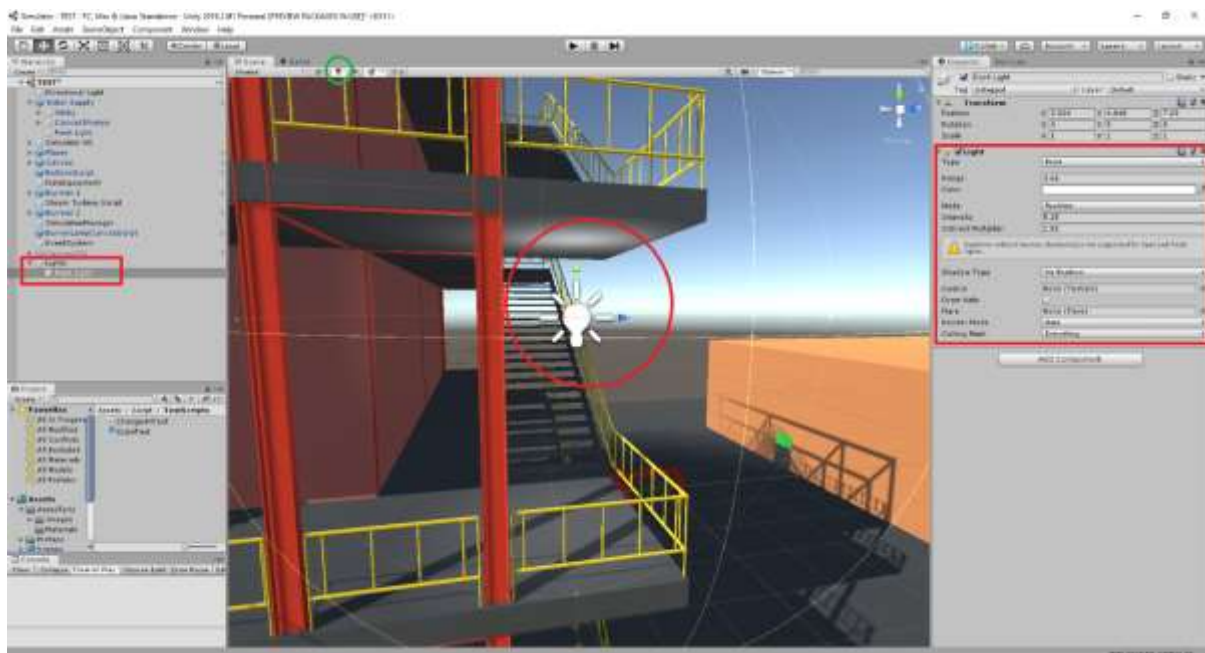


Εικόνα 34 - Δημιουργία αντικείμενου χωρίς χαρακτηριστικά

Σε αυτό το αντικείμενο που δημιουργήθηκε προστέθηκαν στο εσωτερικό του τα «Lights» τα οποία είναι ουσιαστικά ένα αντικείμενο «GameObject» με προσθήκη χαρακτηριστικού «Light». Για να φαίνονται οι φωτισμοί στο «Scene» ενεργοποιούμε τους φωτισμούς από το κουμπί που είναι κυκλωμένο με πράσινη διαγράμμιση στην παρακάτω εικόνα 35 . Στα δεξιά της εικόνας φαίνονται οι δυνατότητες ρυθμίσεων οι τρεις μεταβλητές που χρησιμοποιηθήκαν είναι:

- Type, Point Light.
- Range, το οποίο ορίζει το εύρος-διάσταση του φωτισμού στον τρισδιάστατο χώρο
- Mode, Mixed
- Intensity, ένταση φωτισμού

Η τοποθέτηση έγινε με το εργαλείο μετακίνησης.



Εικόνα 35 - Απεικόνιση αντικειμένου με χαρακτηριστικά φωτισμού "Light"

Components

Τι είναι τα «Components» είναι αυτά που δίνουν χαρακτηριστικά στα αντικείμενα «GameObjects» που δημιουργούνται. Είναι προκατασκευασμένα χαρακτηριστικά που είναι τα ποιο κίονα που χρησιμοποιούνται για δημιουργία παιχνιδιών, όπως για παράδειγμα η τοποθεσία ενός αντικειμένου στο περιβάλλον που είναι το «Transform» παρακάτω θα αναφερθούν τα κυριότερα που χρησιμοποιηθήκανε. Τα Components προσθέτετε σε ένα οποιοδήποτε αντικείμενο «Game Object» είναι χαρακτηριστικό της μηχανής και στα οποία



Components προσφέρονται έτοιμες βιβλιοθήκες προγραμματισμού για ακόμα πιο σύνθετα χαρακτηριστικά.

- **Transform** ορίζει τρία πράγματα σε ένα αντικείμενο την τοποθεσία , περιστροφή και κλίμακα κατά την δημιουργία ενός αντικείμενου GameObject και δημιουργείται υποχρεωτικά για να μπορεί να προσδιορίζεται η θέση του στο περιβάλλον.
- **Collider**, ουσιαστικά δίνει επιφάνεια στο αντικείμενο και μπορεί να οριστεί αν διαπερνάτε ή όχι από τον χρήστη (Trigger). Έχει πολλές δυνατότητες όσο αφορά τα σχήματα που μπορούν να σχεδιαστούνε, όπως η δυνατότητα πλέγματος (Mesh) που το Collider παίρνει το σχήματά του αν το αντικείμενο είναι τρισδιάστατο.
- **Script**, ουσιαστικά δημιουργείται πάλι ένα χαρακτηριστικό του αντικείμενου που ορίζεται από το προγραμματιστεί όπως για παράδειγμα να ορίζουμε μεταβλητές σε ένα αντικείμενο πχ. είδος εξαρτήματος, τύπος, ανοιχτό , κλειστό, κίνηση χαρακτήρα μέσα στο περιβάλλον κτλ.
- **Rigi body**, είναι ένα πολύ χρήσιμο χαρακτηριστικό ουσιαστικά είναι συναρτήσεις φυσικής στο οποίο μπορούμε να ορίσουμε φυσικά χαρακτηριστικά όπως για παράδειγμα βαρύτητα και για πιο σύνθετα προβλήματα προσφέρει βιβλιοθήκες όπως για παράδειγμα προθήκη δυνάμεις στο αντικείμενο και προσομοιάζει την κίνηση της πχ. μια σφαίρα.
- **UI-User Interface**, είναι ουσιαστικά δισδιάστατο και είναι για την χρήση δημιουργία γραφικού περιβάλλοντος αλληλεπίδρασης με το χρήστη όπως κείμενα, κουμπιά (μενού) , εικόνες κτ.λ

Τα κυριότερα components που χρησιμοποιηθήκανε στην κατασκευή του προσομοιωτή είναι τα εξής:

- Collider, δημιουργία μη διαπερατών επιφανειών
- Transform (υποχρεωτικό)
- Script , δημιουργία χαρακτηριστώ σε αντικείμενα
- Rigi Body , για την προσομοίωση κίνησης του χρήστη στο περιβάλλον
- UI, μετάβαση πληροφοριών και αλληλεπίδρασης με τον χρήστη με χρήση εικονικών χαρακτηριστικών όπως εικόνες, γραμματοσειρές , κουμπιά κτλ.

Πηγαίος Κώδικας (Script)

Οι μεταβλητές οπού υπήρξε η επιθυμία να υπάρχει πρόσβαση από όλες τις κλάσεις ταξινομηθήκαν σε ξεχωριστές κλάσεις με σχετικό όνομα και στο τέλος προσθέτοντας την λέξη «Var» όπως για παράδειγμα «GeneralVar.cs». Ένα παράδειγμα δήλωσης μιας τέτοιας κλάσης και μεταβλητής απεικονίζεται στην παρακάτω εικόνα χ.



```
1 public class GeneralVar
2 {
3     private static bool canvasOn;
4
5     public static bool CanvasOn
6     {
7         get { return canvasOn; }
8         set { canvasOn = value; }
9     }
10 }
11
```

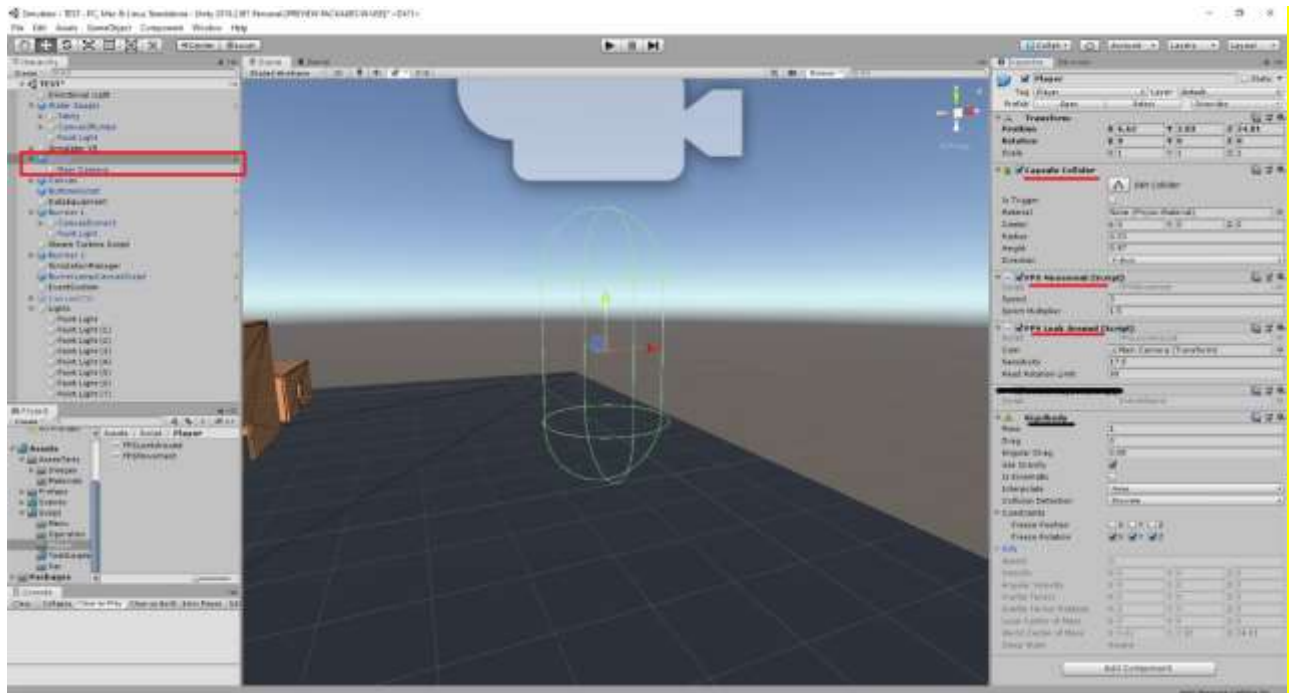
Εικόνα 36 - Παράδειγμα δημιουργίας μια μεταβλητής με ενθυλάκωση "Properties and Encapsulation".

Με αυτή την μέθοδο γίνεται καλύτερος έλεγχος των μεταβλητών. Αυτή η μέθοδος χρησιμοποιείται επίσης για καλύτερή ασφάλεια δεδομένων. Αρκετές πληροφορίες αντλήθηκαν από την πηγή (Docs Microsoft, 2021).

Κίνηση χρήστη στο γραφικό περιβάλλον

Για την κίνηση του χρήστη στο γραφικό περιβάλλον. Δημιουργήθηκε κώδικας που αλληλοεπιδράει με το πληκτρολόγιο και το ποντίκι. Δημιουργήθηκε ένα «GameObject», με χαρακτηριστικά αντικειμένου «Component» τα οποία είναι τα εξής :

- **Rigi Body**, το οποίο προσθέτει χαρακτηριστικά προσομοίωσης φυσικής στο αντικείμενο
- **Collider**, είναι απαραίτητο πριν γίνει η προσθήκη του Rigi Body αλλά και για να μην διαπερνάει τα γραφικά ο χρήστης. Το σχήμα κάψουλας με πράσινη διαγράμμιση που φαίνεται στην παρακάτω εικόνα 37 ουσιαστικά είναι το σώμα το οποίο κινείται στο γραφικό περιβάλλον.
- **Scripts**, δυο πηγαίοι κώδικες «FPSMovement.cs» προσφέρει την κίνηση μέσα στο γραφικό περιβάλλον με το πληκτρολόγιο. Ο «FPSLookAround.cs» ο οποίος περιστρέφει την κάμερα με το ποντίκι και παράλληλα αλλάζει την κατεύθυνση της πορείας.



Εικόνα 37 - Απεικονίζεται το Collider το οποίο κινείται μέσα στο γραφικό περιβάλλον.

Ο κώδικας «FPSMovement.cs» έχει την εξής φιλοσοφία. Ο χρήστης με τα πλήκτρα «W,S,A,D» θα κατευθύνεται μπροστά(W) , πίσω(S), αριστερά(A) και δεξιά(D). Όταν πατάει το πλήκτρο “Shift” και παράλληλα ένα πλήκτρο κατεύθυνσης(W,S,D,A) η ταχύτητα κίνησης αυξάνεται. Παρακάτω απεικονίζεται ο κώδικας με σχόλια στην εικόνα 38.


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class FPSMovement : MonoBehaviour
6 {
7     [SerializeField] float speed;
8     Rigidbody rb;
9     [SerializeField] float sprintMultiplier = 1.5f;
10
11     void Start() {
12         //Το πρώτο Frame θα ορίσει το Rigidbody του αντικείμενου στην μεταβλητή rb
13         rb = GetComponent<Rigidbody>();
14     }
15
16     void Update() {
17
18         //Η τιμές που θα επιστρέψει η συνάρτηση GetAxisRaw() είναι από [-1 έως 1]
19         //και δηλώνει την κατεύθυνση μετακίνησης στους άξονες
20         float x = Input.GetAxisRaw("Horizontal");
21         float z = Input.GetAxisRaw("Vertical");
22
23         Vector3 moveBy = transform.right * x + transform.forward * z;
24
25         float actualSpeed = speed;
26
27         //Πολλαπλασιάζει το actualSpeed για πιο γρήγορη κίνηση.
28         if (Input.GetKey(KeyCode.LeftShift)) {
29             actualSpeed *= sprintMultiplier;
30         }
31
32         //Συνάρτηση πάρνει σαν μεταβλητή την moveBy
33         //ορισμός συνάρτησης rb.MovePosition(new Vector3(x, 0, z));
34         rb.MovePosition(transform.position + moveBy.normalized * actualSpeed * Time.deltaTime);
35     }
36 }
37
38
```

Εικόνα 38 - Κώδικας FPSMovement για την κίνηση μέσω του πληκτρολογίου

Ο κώδικας «FPSLookAround.cs» που απεικονίζεται στην εικόνα 39 δίνει την δυνατότητα περιστροφής της κάμερας στο γραφικό περιβάλλον και ταυτόχρονα όταν ο χρήστης πατάει ένα πλήκτρο κατεύθυνσης (W,S,D,A) αλλάζει η κατεύθυνση της πορείας.


```
1 using UnityEngine;
2
3 public class FPSLookAround : MonoBehaviour
4 {
5     [SerializeField] Transform cam; //MainCamera
6     [SerializeField] float sensitivity; //Ρυθμισή ευαισθησίας
7     float headRotation = 0f;
8     [SerializeField] float headRotationLimit = 90f; //Όριο περιστροφής στον άξονα y
9
10 void Start()
11 {
12     Cursor.visible = false;
13     Cursor.lockState = CursorLockMode.Locked;
14 }
15 void Update() {
16     //Ορισμός μεταβλητών x και y , που ορίζουν την μεταβλητή περιστροφής στους αντίστοιχους άξονες
17     //Input.GetAxis("Mouse X") , η θέση του ποντικιού στον άξονα x και αντίστοιχα για τον y
18     //Time.deltaTime, είναι ο χρόνος εκτέλεσης του frame
19
20     float x = Input.GetAxis("Mouse X") * sensitivity * Time.deltaTime;
21     float y = Input.GetAxis("Mouse Y") * sensitivity * Time.deltaTime * -1f;
22
23     // Αν η μεταβλητή είναι false λειτουργεί η περιστροφή
24     if (!GeneralVar.CanvasOn)//Η μεταβλητή CanvasOn γίνεται true κάθε φορά που ανήγει ένας canvas.
25     {
26         Cursor.visible = false; //Ο κέρσορας του ποντικιου δεν είναι ορατός
27         Cursor.lockState = CursorLockMode.Locked; //Κλειδώνει ο κέρσορας
28
29         //Περιστροφή στον άξονα x , χωρίς όριο
30         transform.Rotate(0f, x, 0f);
31
32         //Περιστροφή στον άξονα y με όριο την μεταβλητή headRotationLimit
33         headRotation += y; //Ορισμός μεταβλητής
34
35         /**
36          * Οριοθέτηση μεταβλητής: με την συνάρτηση Mathf.Clamp(μεταβλητή,ελαχιστο όριο, μέγιστο όριο)
37          * Ουσιαστικά δεν περνει τιμες εξω απο τα ορια που εχουμε οριστεί
38          */
39         headRotation = Mathf.Clamp(headRotation, -headRotationLimit, headRotationLimit);
40         cam.localEulerAngles = new Vector3(headRotation, 0f, 0f);
41     }
42     //Σε περίπτωση true δεν λειτουργεί για να μπορεί ο χρήστης με τον κέρσορα να πατήσει τα Button στο Canvas
43     else if(GeneralVar.CanvasOn)
44     {
45         Cursor.visible = true; //Ο κέρσορας του ποντικιου είναι ορατος
46         Cursor.lockState = CursorLockMode.None; //Εκκλειδώνεται ο κέρσορας
47     }
48 }
49 }
```

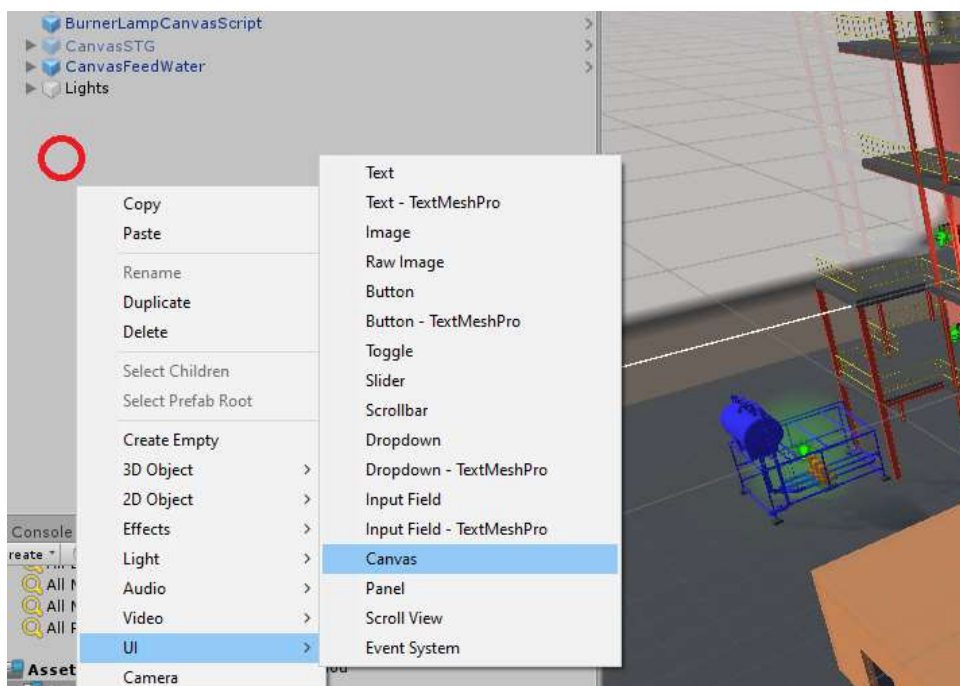
Εικόνα 39 - Κώδικας «FPSLookAround.cs» με σχόλια επεξήγησης

Ένα άλλο χαρακτηριστικό που έχει ο παραπάνω κώδικας είναι ότι όταν η μεταβλητή «GeneralVar.CanvasOn» όταν γίνεται αληθής «True», παγώνει η περιστροφή της κάμερας με το ποντίκι και ο χρήστης μπορεί να κινηθεί πλέον μόνο με το πληκτρολόγιο. Αντίστοιχα όταν γίνεται ψευδής «False», μπορεί να περιστρέφει η κάμερα. Για την περιστροφή της κάμερας πρέπει να κλειδώνει ο κέρσορας στην οθόνη (γραμμή 27) και να μην είναι ορατός (γραμμή 26). Όταν λοιπόν ανοίγει ένας «Canvas» θα πρέπει ο χρήστης να μπορεί να κινηθεί τον κέρσορα στην οθόνη(γραμμή 45) και φυσικά να είναι ορατός (γραμμή 46) για να μπορεί να αλληλεπιδράσει με αυτόν.

Δημιουργία ενός Canvas και αλληλεπίδραση του χρήστη με αυτόν.

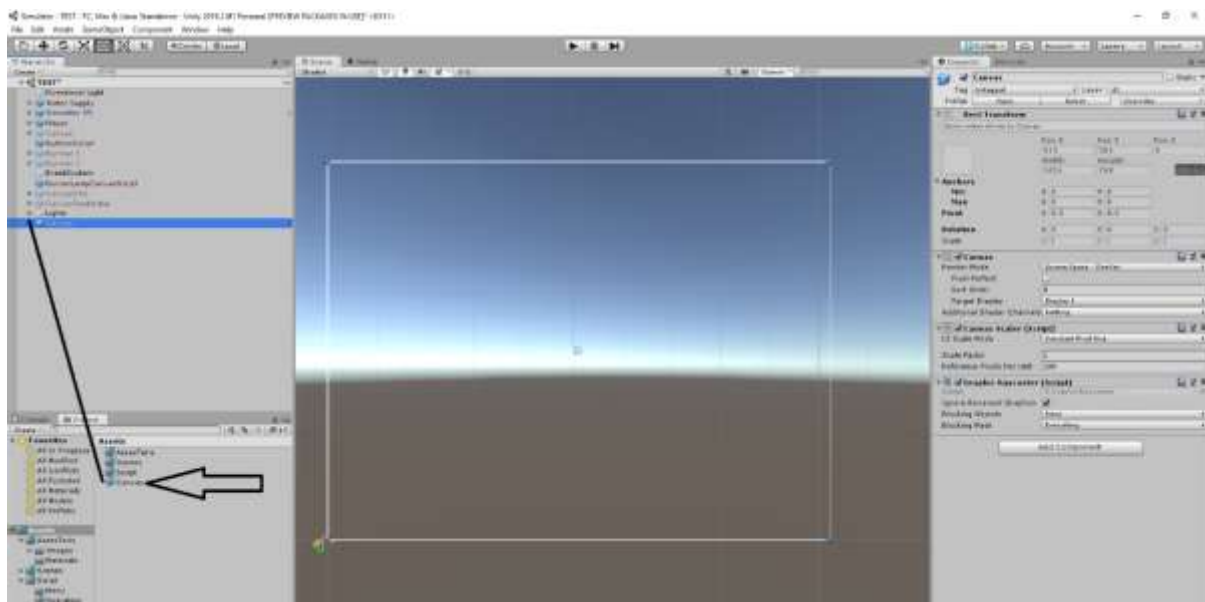
Τα «Canvas» χρησιμοποιήθηκαν κυρίως για αλληλεπίδραση του χρήστη με τον προσομοιωτή. Όπως για παράδειγμα την εκκίνηση αντλιών. Ουσιαστικά εμφανίζει στοιχεία αλληλεπίδρασης του χρήστη «User Interface». Τα οποία αποτελούνται από γραμματοσειρές, κουμπιά «Buttons», εικόνες και άλλα στοιχεία μορφοποίησης τους.

Για την δημιουργία ενός αντικείμενου με χαρακτηριστικά «Canvas» πατώντας δεξί κλικ και επιλέγοντας «UI>Canvas» δημιουργείται ένας «Canvas».



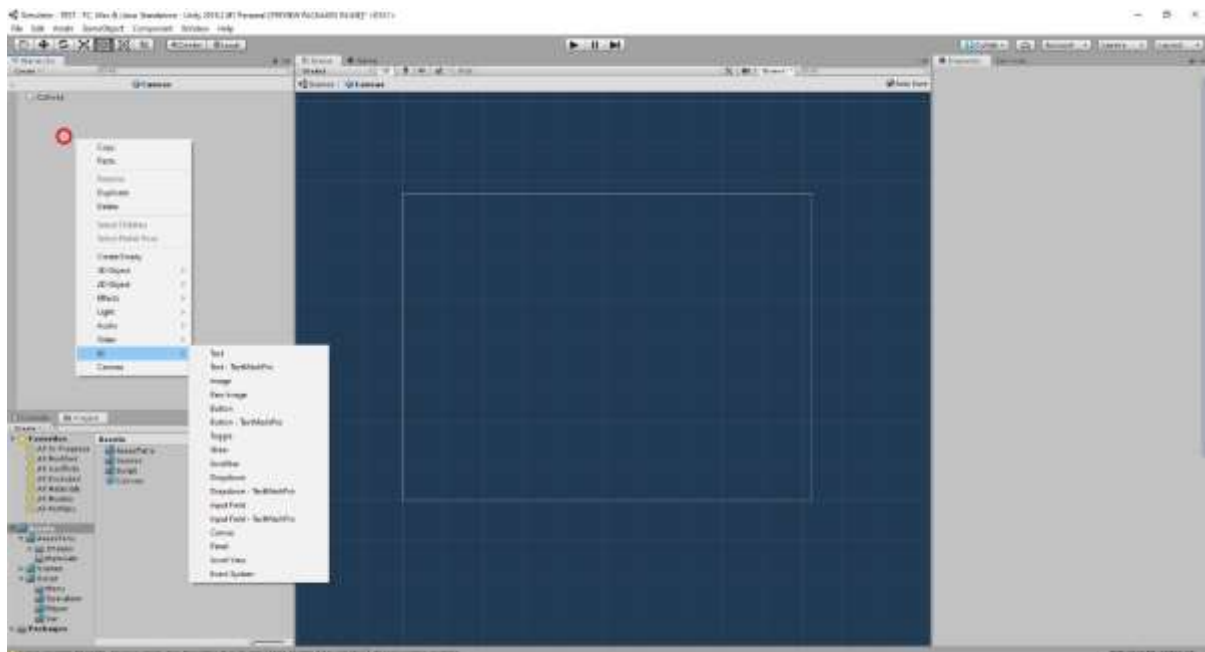
Εικόνα 40 - Δημιουργία "Canvas"

Αυτός ο «Canvas» χρησιμοποιήθηκε σαν πρότυπο αρά δημιουργήθηκε ένα «Prefab Assets» γιατί χρειάστηκε πάνω από μια φορά στον προσομοιωτή που υλοποιήθηκε. Αρά λοιπόν σέρνοντας το αντικείμενο που δημιουργήθηκε στον φάκελο «Assets» και σε έναν οποιοδήποτε υποφάκελό του δημιουργείται αυτόματα το «Prefab Assets» με το όνομα «Canvas». Όπως φαίνεται στην παρακάτω εικόνα 41.



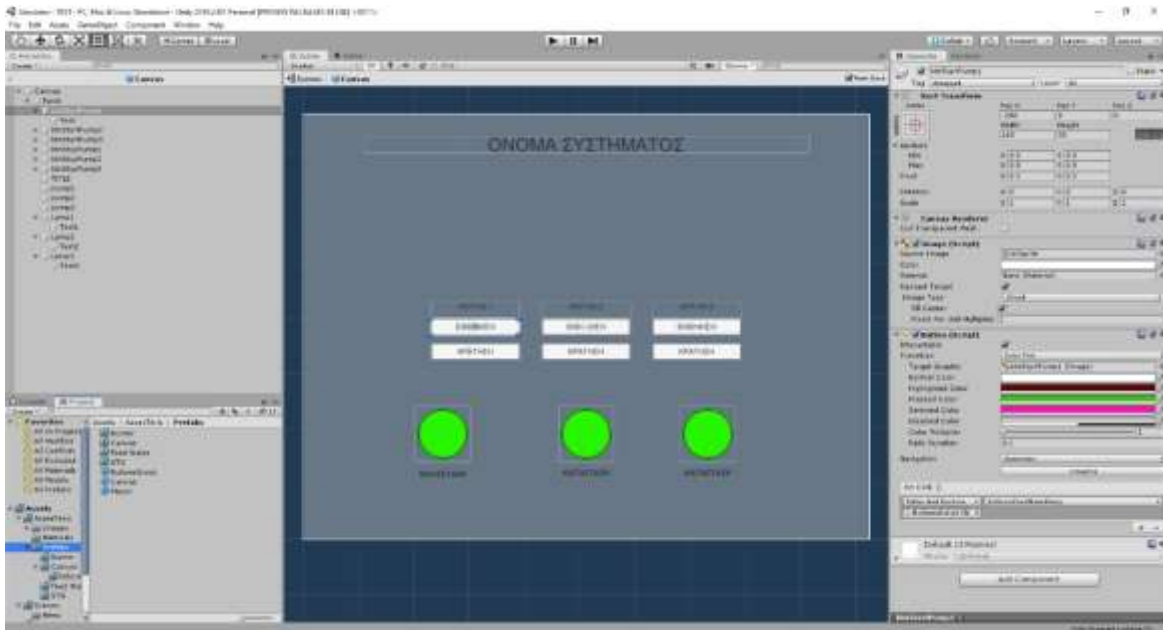
Εικόνα 41 - Δημιουργία "Prefab Assets"

Πατώντας δύο φορές στο αντικείμενο «Canvas» ή δεξί κλικ και «Open» το οποίο βρίσκεται στον φάκελο «Assets» φαίνεται στην παραπάνω εικόνα 41 με βέλος. Θα ανοίξει το αντικείμενο μεμονωμένα και θα απεικονίζεται όπως στην παρακάτω εικόνα 42.



Εικόνα 42 - Μεμονωμένο περιβάλλον επεξεργασίας ενός "Prefab Assets"

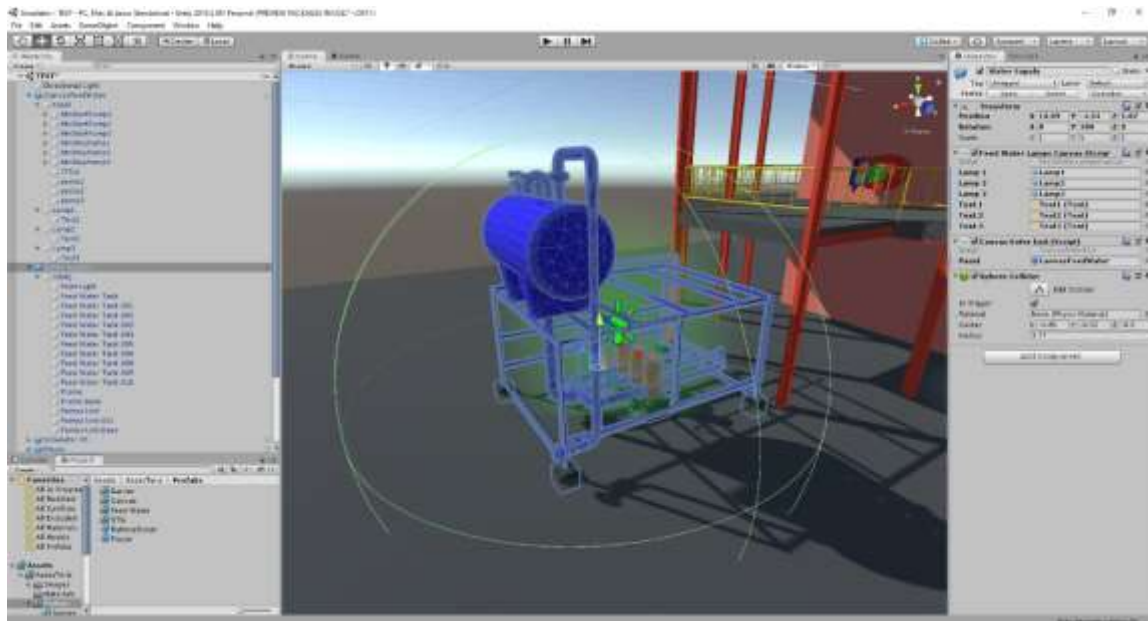
Πατώντας δεξί κλικ και επιλέγοντας «UI» μπορούν να δημιουργηθούν διάφορα αντικείμενα με χαρακτηριστικά. Αρχικά πρέπει να δημιουργηθεί το «Panel» που σαν εσωτερικά του αντικείμενα θα είναι όπως «Text», «Buttons», «Image» κτλ.. Η τελική διαμόρφωση απεικονίζεται στην παρακάτω εικόνα 43.



Εικόνα 43 -Παράδειγμα ενός “Interface” για τρεις αντλίες

Εφόσον υλοποιήθηκε ο «Canvas» μας σαν πρότυπο πλέον, μπορούμε σέρνοντας το στην σκηνή να δημιουργήσουμε ένα αντίγραφο του. Πατώντας δεξί κλικ και κάνοντας την διαδικασία «Unpack Prefab» δημιουργήθηκε ένα νέο αντικείμενο με ακριβώς με τα ίδια χαρακτηριστικά χωρίς να επιφέρει αλλαγές από το πρότυπο «Prefab Assets», το οποίο θα επεξεργαστεί για το σύστημα τροφοδοσίας νερού του λέβητα.

Στην συνέχεια θα αναφερθεί πως δημιουργήθηκε η αλληλοεπίδρασή με το σύστημα τροφοδοσίας νερού στον λέβητα. Στην παρακάτω εικόνα απεικονίζεται το σύστημα με τις τρεις αντλίες και την δεξαμενή.



Εικόνα 44 -Το σύστημα τροφοδοσίας νερού στον λέβητα στο περιβάλλον της Unity 3d Editor.

Στο πρώτο αντικείμενο στην ιεραρχία προστέθηκε το χαρακτηριστικό «Sphere Collider» με την ιδιότητα του επιλεγμένη «Is Trigger» (true). Δηλαδή να το διαπερνούν αλλά αντικείμενα με πρόσθετο χαρακτηριστικό «Collider».

Το πρόσθετο χαρακτηριστικό «Script» με το όνομα «CanvasEnterExit.cs» δηλαδή ο κώδικας αναλαμβάνει την εμφάνιση του «Canvas» που επιθυμούμε να εμφανιστεί ανάλογα το σύστημα. Αυτός ο «Canvas» δηλώνεται στην μεταβλητή «panel» του παρακάτω κώδικα. Θα μπορούσε να οριστεί ένα άλλο οποιοδήποτε αντικείμενο στην μεταβλητή.

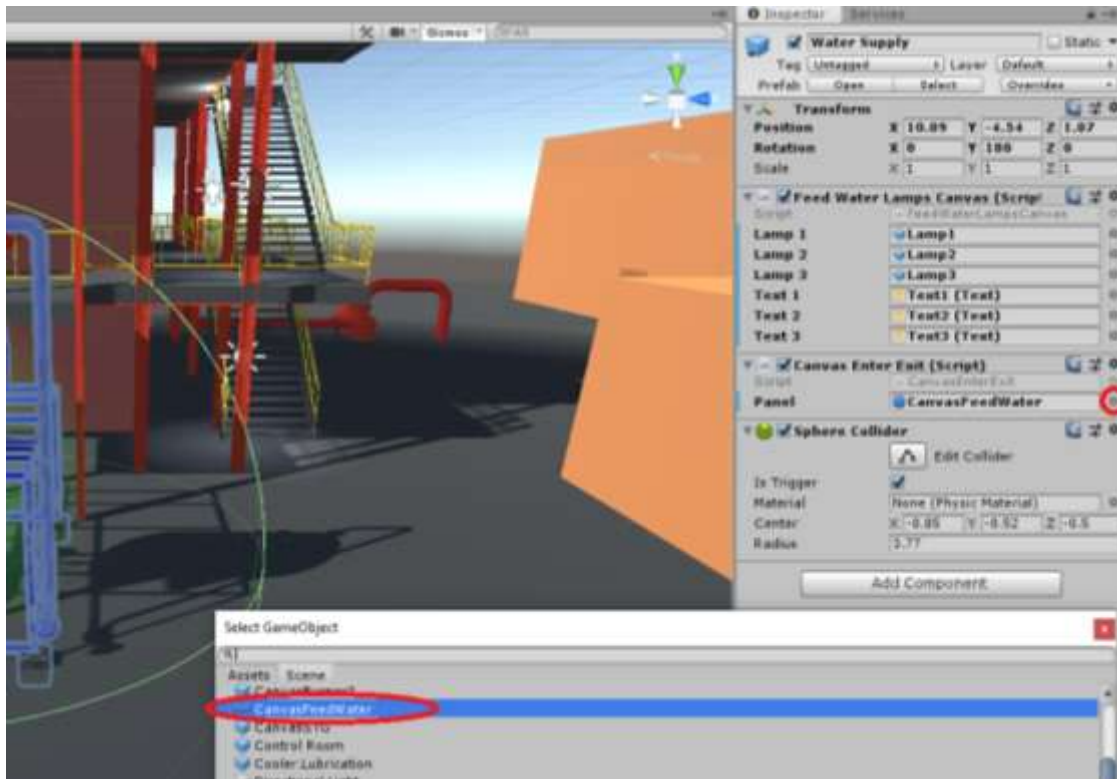


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class CanvasEnterExit : MonoBehaviour
6 {
7     // Στην περίπτωση μας είναι ο Canvas αλληλεπίδρασης του χρήστη με κάποιο σύστημα
8     // Για παράδειγμα το σύστημα τροφοδοσίας νερού στον λέβητα
9     public GameObject panel;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14         // Στο πρώτο frame κάνει απενεργοποιεί το GameObject
15         panel.SetActive(false);
16     }
17     private void OnTriggerEnter(Collider other)
18     {
19         // Όταν μπαίνει στην περιοχή ένα Collider εκτελούνται τα παρακάτω
20         GeneralVar.CanvasOn = true; // Μεταβλητή οι οποία χαρακτηρίζει ότι ανοίξε ένας Canvas
21         panel.SetActive(true); // Ενεργοποιεί το αντικείμενο
22         // Debug.Log("Player enter");
23     }
24     private void OnTriggerExit(Collider other)
25     {
26         // Όταν βγαίνει από την περιοχή εκτελούνται τα παρακάτω
27         GeneralVar.CanvasOn = false;
28         panel.SetActive(false);
29         // Debug.Log("Player exit");
30     }
31 }
```

Εικόνα 45 - Κώδικας "CanvasEnterExit" εμφανίζει οποιοδήποτε αντικείμενο που δηλώνεται στην μεταβλητή panel

Όταν δημιουργήθηκε η κίνηση του χρήστη στο γραφικό περιβάλλον προστέθηκε και ένα «Collider Capsule». Αυτό λοιπόν ουσιαστικά είναι που εκτελεί τις συναρτήσεις «OnTriggerEnter» και «OnTriggerExit» κατά την είσοδο ή έξοδο του στην περιοχή του «Collider» που είναι διαπερατό.

Υπενθυμίζεται ότι όταν ο χρήστης μπαίνει στην περιοχή του «Collider» ενός συστήματος για παράδειγμα τροφοδοσία νερού λέβητα μπορεί πλέον ο χρήστης να κινήσει τον κέρσορα του ποντικιού στην οθόνη και ταυτόχρονα παγώνει η περιστροφή της κάμερας . Έτσι είναι δυνατόν η αλληλεπίδραση με ένα «Canvas» ή ένα «User Interface (UI)». Αν ο χρήστης επιθυμεί να φύγει από αυτήν την περιοχή του «Collider» είναι δυνατόν μόνο με την κίνηση του πληκτρολογίου (W,S,D,A) .



Εικόνα 46 - Ορισμός μεταβλητής panel με το αντικείμενο "CanvasFeedWater"

Η επιλογή της μεταβλητής «Panel» γίνεται με τον εξής τρόπο. Πατώντας την τελεία που σημειώνεται με κόκκινό κύκλο στην παραπάνω εικόνα 46 , θα εμφανιστεί το παράθυρο «Select Objects» και στην οποία μεταβλητή προστέθηκε το αντικείμενο με όνομα «CanvasFeedWater» το οποίο είναι και ο «Canvas» για το συγκεκριμένο σύστημα.

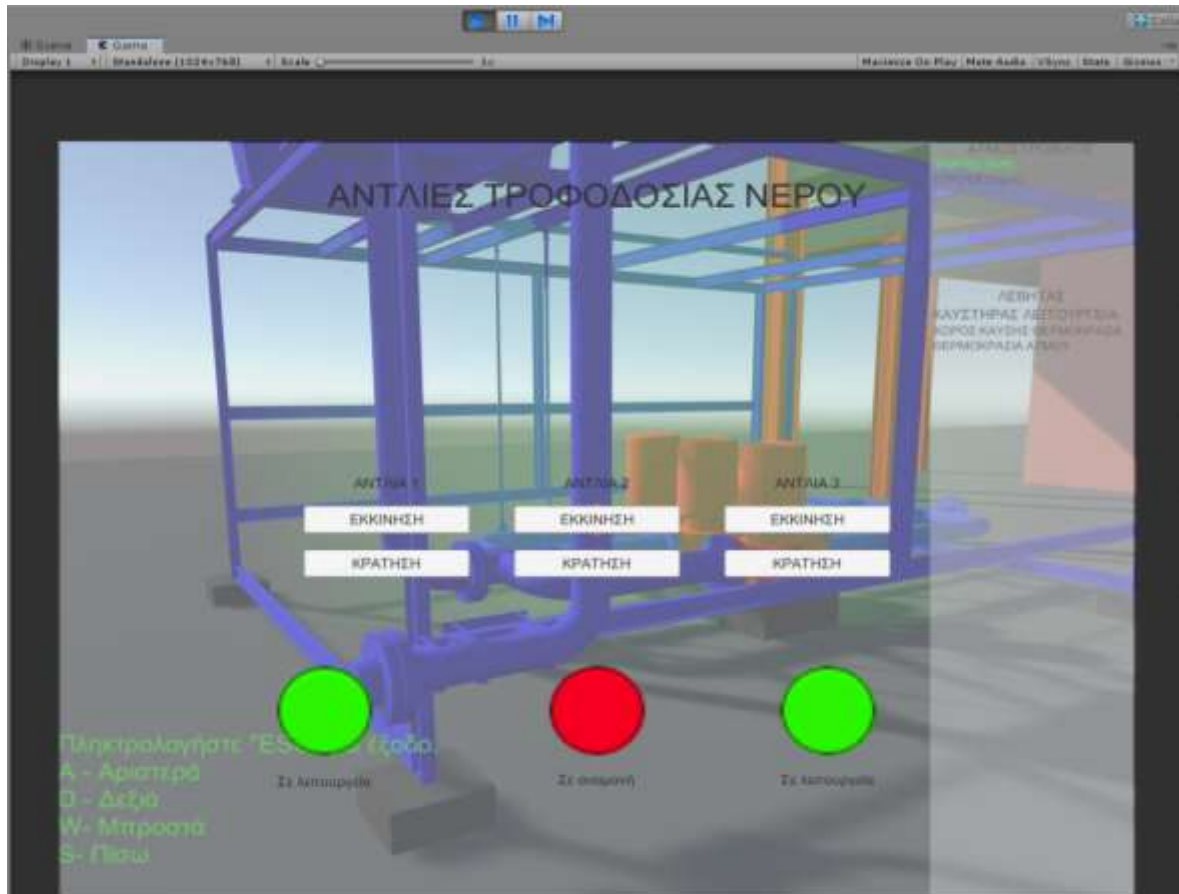


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class FeedWaterLampsCanvas : MonoBehaviour
7 {
8     public GameObject lamp1,lamp2,lamp3;
9     public Text text1, text2, text3;
10
11     // Update is called once per frame
12     void Update()
13     {
14         if(FwVar.Pump1)
15         {
16             lamp1.GetComponent<Image>().color = new Color32(45, 255, 0, 255);
17             text1.text = "Σε λειτουργία";
18         }
19         else
20         {
21             lamp1.GetComponent<Image>().color = new Color32(255, 0, 33, 255);
22             text1.text = "Σε αναμονή";
23         }
24
25         if (FwVar.Pump2)
26         {
27             lamp2.GetComponent<Image>().color = new Color32(45, 255, 0, 255);
28             text2.text = "Σε λειτουργία";
29         }
30         else
31         {
32             lamp2.GetComponent<Image>().color = new Color32(255, 0, 33, 255);
33             text2.text = "Σε αναμονή";
34         }
35
36         if (FwVar.Pump3)
37         {
38             lamp3.GetComponent<Image>().color = new Color32(45, 255, 0, 255);
39             text3.text = "Σε λειτουργία";
40         }
41         else
42         {
43             lamp3.GetComponent<Image>().color = new Color32(255, 0, 33, 255);
44             text3.text = "Σε αναμονή";
45         }
46     }
47 }
```

Εικόνα 47 - Κώδικας *FeedWaterLampsCanvas* τροποποιεί τα αντικείμενα που ορίζονται στις μεταβλητές 8 και 9 γραμμή κώδικα.

Στην παραπάνω εικόνα 47 φαίνεται ακόμα ένα προσθετό χαρακτηριστικό ο κώδικας «*Feed Water Lamps Canvas.cs*», ο οποίος αλλάζει τα χαρακτηριστικά του «*CanvasFeedWater*».

Στις μεταβλητές «*lamp1,lamp2,lamp3*» γραμμή 8 δηλώνονται τρία αντικείμενα που στην περίπτωση αυτή είναι αντικείμενα με χαρακτηριστικό «*Image*». Τρεις ακόμα οι «*text1, text2, text3*» η οποίες αναφέρουν αν είναι σε λειτουργία ή όχι οι αντλίες. Η φιλοσοφία του κώδικα είναι να δείχνει στον χρήστη αν οι αντλίες είναι σε λειτουργία ή όχι με χρήση εικόνας και γράμματα, να αλλάζει την κατάστασή τους το οποίο θα εξηγηθεί παρακάτω. Στην παρακάτω εικόνα 48 απεικονίζεται ο τρόπος λειτουργίας.

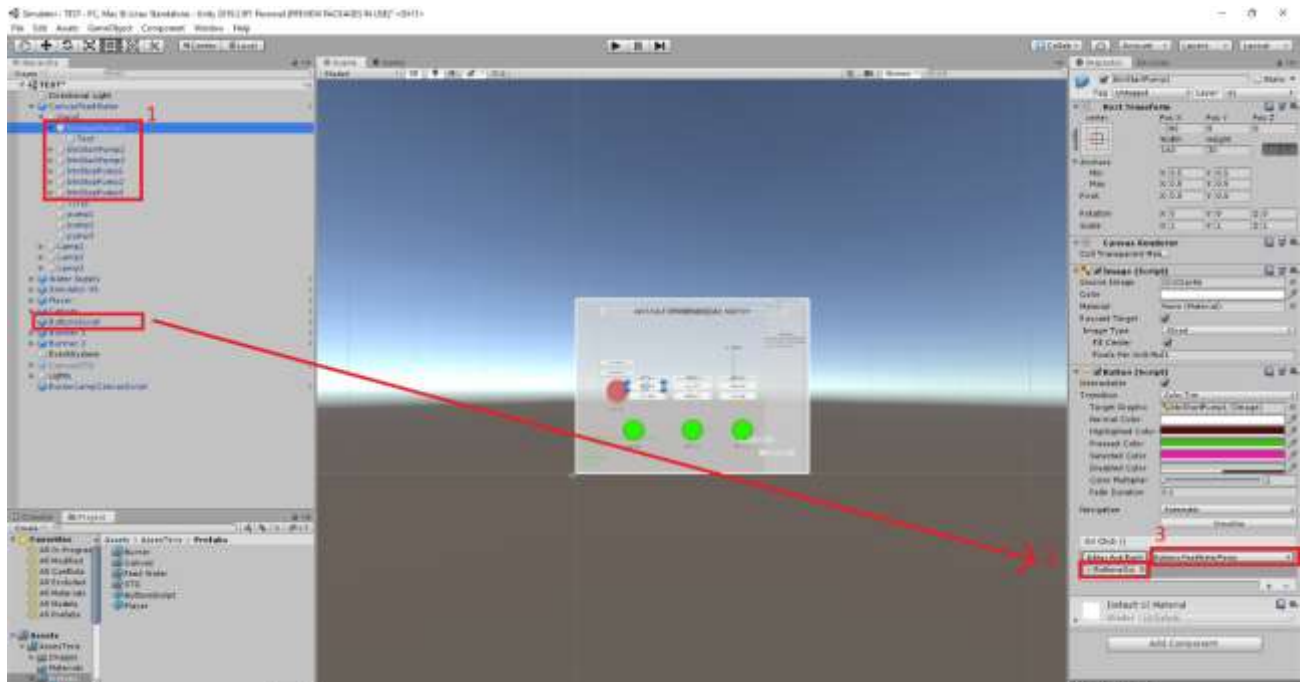


Εικόνα 48 - Σύστημα τροφοδότησης νερού λέβητα, αλληλεπίδραση χρήστη με αυτό.

Εφόσον ο χρήστης πάτησε «ΕΚΚΙΝΗΣΗ» στην αντλία 1 και 3 και εννοώ η αντλία 2 δεν δουλεύει υποθετικά, η κατάσταση του «Canvas» άλλαξε αντίστοιχα όπως φαίνεται στην εικόνα 48 παραπάνω.

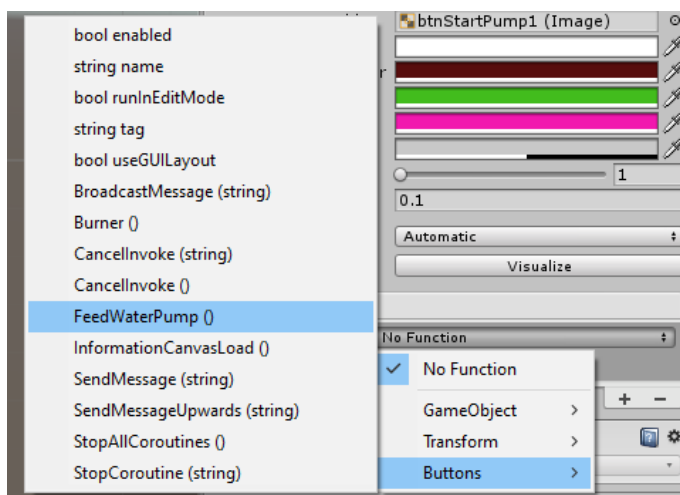
Ο χρήστης όταν πατάει αριστερό κλικ σε ένα «Button» όπως για παράδειγμα «ΕΚΚΙΝΗΣΗ» θα πρέπει να οριστεί ποιο σύστημα είναι και ποια αντλία. Αυτό γίνεται με την βοήθεια του κώδικα «Buttons.cs». Δημιουργήθηκε ένα αντικείμενο στην σκηνή με όνομα «ButtonScript» και προστέθηκε αυτός ο κώδικας σαν χαρακτηριστικό του.

Όταν επιλέγεται ένα «Button» η Unity 3D δίνει την δυνατότητα να οριστεί μια συνάρτηση στην παράμετρο «On Click()» και να εκτελείται όταν πατιέται το «Button». Στην εικόνα 49 έχει επιλεγεί το «btnStartPump1». Ορίσαμε σαν κώδικα τον «ButtonScript.cs» από το αντικείμενο «ButtonScript» ο οποίος προστέθηκε σε αυτό. Επιλέγεται η συνάρτηση «FeedWaterPumps ()» σε αυτήν την περίπτωση.



Εικόνα 49 - Προσθήκη συνάρτησης σε ένα "Button" στην παράμετρο On Click()

Η συνάρτηση επιλέγεται πατώντας στο σημείο τρία (3). Αν δεν έχει οριστεί κάποια συνάρτηση του κώδικα θα λέει «No Function». Η διαδικασία επιλογής συνάρτησης απεικονίζεται στην παρακάτω εικόνα 50.



Εικόνα 50 - Προσθήκη συνάρτησης "FeedWaterPump()"

Εφόσον επιλέχτηκε η συνάρτηση «FeedWaterPump ()» όταν πατιέται το συγκεκριμένο «Button» θα εκτελείται η συνάρτηση και η χρησιμότητα της αναφέρεται παρακάτω.

Στο κώδικα «Buttons.cs» έχει συναρτήσεις για αντίστοιχα συστήματα όπως καυστήρες, αντλίες λίπανσης και τροφοδοσίας νερού λέβητα. Στην παρακάτω εικόνα 51 αναφέρεται η συνάρτηση «FeedWaterPump ()» για το σύστημα τροφοδοσίας νερού στο λέβητα. Αναφέρονται σχόλια στον κώδικα για την καλύτερη κατανόησή του.

```
1 using UnityEngine;
2 using UnityEngine.Events;
3
4 public class Buttons : MonoBehaviour
5 {
6     /// 
7     /// Αυτή η συνάρτηση είναι για τις τροφοδοτικές αντλίες νερού λέβητα για όριζες την κατάσταση τους.
8     /// 
9     public void FeedWaterPump()
10     {
11         //Το go ορίζεται το αντικείμενο που αλλάζουμε στο παιχνίδι μας το button που πατάμε
12         var go = EventSystem.current.currentSelectedGameObject;
13
14         //Το go.name επιστρέφει το όνομα του αντικείμενου
15         switch (go.name)
16         {
17             case "BtnStartPump1": //Όταν ανακλικώνω (button) , κλέβει τα υαίματα αναμεικ
18                 Debug.Log("Start Pump 1");
19                 Filter.Pump1 = true; //Ορίσμός μεταβλητής σε true, δηλαδή η αντλία είναι σε λειτουργία
20                 break;
21             case "BtnStartPump2":
22                 Debug.Log("Start Pump 2");
23                 Filter.Pump2 = true;
24                 break;
25             case "BtnStartPump3":
26                 Debug.Log("Start Pump 3");
27                 Filter.Pump3 = true;
28                 break;
29
30             case "BtnStopPump1":
31                 Debug.Log("Stop Pump 1");
32                 Filter.Pump1 = false; //Ορίσμός μεταβλητής σε false, δηλαδή η αντλία δεν λειτουργεί
33                 break;
34             case "BtnStopPump2":
35                 Debug.Log("Stop Pump 2");
36                 Filter.Pump2 = false;
37                 break;
38             case "BtnStopPump3":
39                 Debug.Log("Stop Pump 3");
40                 Filter.Pump3 = false;
41                 break;
42             default:
43                 Debug.LogError("Method: " + GetCurrentUserMethod() + " * " + gameObject.name + " go.name: " + go.name); //Den kexei kati alla
44                 break;
45         }
46     }
47 }
```

Εικόνα 51 - Κώδικας "Buttons" η συνάρτηση FeedWaterPump

Τα παραπάνω που αναφέρθηκαν έγιναν και στα αντίστοιχα συστήματα που πρέπει να αλληλοεπιδράσει ο χρήστης. Δεν θα είχε νόημα να αναφερθούν γιατί έγινε ακριβώς η ίδια διαδικασία με τα προαναφερόμενα στον κώδικα «Buttons.cs». Δημιουργήθηκε μια αντίστοιχη συνάρτηση για κάθε σύστημα για καλύτερη διαχείριση. Για παράδειγμα για τους καυστήρες δημιουργήθηκε η συνάρτηση «Burner ()» όπως απεικονίζεται στην παρακάτω εικόνα 52.


```
48 public void Burner()  
49 {  
50     var go = EventSystem.current.currentSelectedGameObject;  
51     Debug.Log(go.name);  
52     switch (go.name)  
53     {  
54         //START BURNER  
55         case "btnStartBurn1":  
56             Debug.Log("Start Burn 1");  
57             BurnerVar.Burner1 = true;  
58             break;  
59         case "btnStartBurn2":  
60             Debug.Log("Start Burn 2");  
61             BurnerVar.Burner2 = true;  
62             break;  
63         //STOP BURNER  
64         case "btnStopBurn1":  
65             Debug.Log("Stop Burn 1");  
66             BurnerVar.Burner1 = false;  
67             break;  
68         case "btnStopBurn2":  
69             Debug.Log("Stop Burn 2");  
70             BurnerVar.Burner2 = false;  
71             break;  
72         //Αν δεν υπάρχει το αντικείμενο (Button) σαν όνομα βγάζει το παρακάτω μήνυμα.  
73         default:  
74             Debug.LogError("Συνάρτηση: " + GetCurrentMethodName() + " + gameObject.name + go.name: " + go.name);  
75             break;  
76     }  
77 }  
78  
79 }
```

Εικόνα 52 - Κώδικας "Buttons.cs" και η συνάρτηση Burner() για το σύστημά καυστήρων

Ουσιαστικά η αλληλεπίδραση του χρήστη με τον προσομοιωτή γίνεται με τον κώδικα «Buttons.cs».

Κατασκευή μενού προσομοιωτή

Με την βοήθεια των χαρακτηριστικών «User Interface (UI)» θα δημιουργηθεί το μενού του προσομοιωτή. Δημιουργήθηκε αρχικά ένας «Canvas» σε μια νέα σκηνή «Menu», ο οποίος αποτελείται από μια είσοδο που βάζει ο χρήστης το ονοματεπώνυμο του και τρία «Buttons» τα οποία είναι τα εξής:

1. **Σενάρια**, φορτώνει μια σκηνή την «MenuScenario» , η οποία απαρτίζει τα προκατασκευασμένα σενάρια με ονομασία και περιγραφή. Πατώντας στο «Button» θα φορτώνει η σκηνή με το αντίστοιχο σενάριο.
2. **Πληροφορίες**, απαρτίζει πληροφορίες για τον προσομοιωτή.
3. **Έξοδος**, κλείνει την εφαρμογή

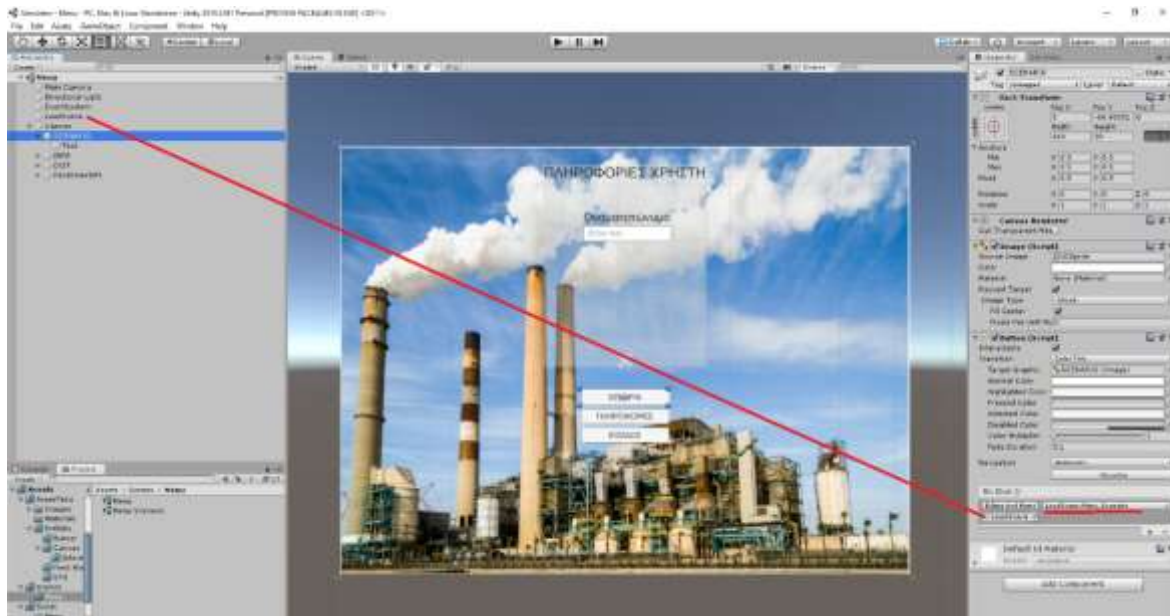
Πατώντας στο αντικείμενο «Canvas» εμφανίζονται οι παράμετροι ρυθμίσεων στα δεξιά. Στο χαρακτηριστικό «Canvas» στην επιλογή «Render Mode» επιλέχτηκε η παράμετρος «Screen Space - Overlay». Αυτό γίνεται για την οριοθέτηση του «Canvas» σε όλο το εύρος της

κάμερας. Η φωτογραφία που είναι στην πίσω όψη δημιουργείται από το χαρακτηριστικό «Raw Image (Script)» πατώντας την επιλογή «Texture». Επιλέχτηκε η φωτογραφία από τον φάκελο «Assets». Ο «Canvas» ο οποίος δημιουργήθηκε για το αρχικό μενού απεικονίζεται στην παρακάτω εικόνα 53.



Εικόνα 53 - Δημιουργία αρχικού μενού

Εφόσον διαμορφώθηκε ο «Canvas» ορίστηκε τι θα κάνει κάθε «Button» με την βοήθεια του κώδικα «LoadScene.cs». Έτσι λοιπόν δημιουργήθηκε ένα «Empty» αντικείμενο στην σκηνή με όνομα «LoadScene» και προστέθηκε σαν χαρακτηριστικό του ο κώδικας. Το αντικείμενο προστέθηκε στην παράμετρο «On Click()» και επιλέχτηκε η συνάρτηση «Menu_Scenario» του κώδικα. Υπενθυμίζεται ότι αυτή η λειτουργία είναι ότι όταν πατάει αριστερό κλικ ο χρήστης το συγκεκριμένο «Button» εκτελείται μια συνάρτηση.



Εικόνα 54 - Εισαγωγή κώδικα και συνάρτησης στην παράμετρο On Click()

Ο κώδικας «LoadScene.cs» περιέχει τις συναρτήσεις που φορτώνει τις σκηνές για το μενού και τα σενάρια. Η Unity 3D με τις έτοιμες συναρτήσεις της καθιστά την αλλαγή σκηνής μια εξαιρετικά απλή διαδικασία. Ο κώδικας απεικονίζεται στην παρακάτω εικόνα 55.

```

1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3
4  public class LoadScene : MonoBehaviour
5  {
6      public void exitSimulator()
7      {
8          Debug.Log("Exit..");
9          Application.Quit(); //Κλείνει την εφαρμογή
10     }
11     public void Test_scenario()
12     {
13         SceneManager.LoadScene("TEST"); //Εάν όρισμα είναι το ονομά της σκηνής
14     }
15     public void Menu_Scenario()
16     {
17         SceneManager.LoadScene("Menu_Scenario");
18     }
19 }
20

```

Εικόνα 55 – Κώδικας «LoadScene.cs»

Στην γραμμή 17 του κώδικα φορτώνει την σκηνή η οποία περιέχει τα σενάρια του προσομοιωτή και από εκεί φορτώνεται η σκηνή του κάθε σεναρίου. Το αντίστοιχο έγινε και στην σκηνή με τα σενάρια όπου φορτώνεται μια σκηνή και με το πάτημα ενός «Button» με την αντίστοιχη συνάρτηση φορτώνει η σκηνή.

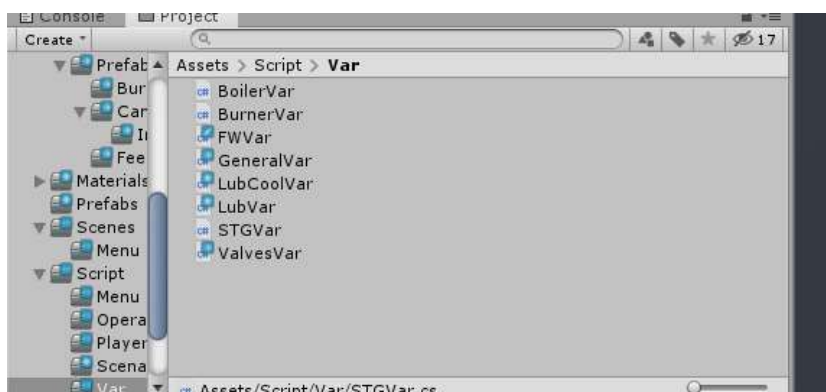


Δημιουργία Σεναρίου

Εφόσον έχουμε ορίσει τα συστήματα μας και τις λειτουργίες τους θα δημιουργηθεί ένα σενάριο βάση αυτών. Τα συστήματα είναι τα εξής:

- Τροφοδοτικές αντλίες νερού λέβητα
- Καυστήρες λέβητα (δυο)
- Αντλίες προλίπανσης και λίπανσης ατμοστρόβιλου
- Αντλίες νερού για την ψύξη του ελαίου λίπανσης με ενναλάκτη

Κατά την διάρκεια υλοποίησης του προσομοιωτή δημιουργήθηκαν κώδικες οι οποίοι περιέχουν μεταβλητές όπως προαναφέρθηκε στην υποενότητα «Πηγαίος Κώδικας». Στην παρακάτω εικόνα 56 απεικονίζονται.



Εικόνα 56 - Κώδικες με περιεχόμενο μεταβλητές για κάθε σύστημα ξεχωριστά

Για παράδειγμα ο κώδικας «LubVar» περιέχει μεταβλητές για το σύστημα λίπανσης του ατμοστροβίλου. Οι μεταβλητές είναι κυρίως για τις αντλίες αν είναι σε λειτουργία ή όχι. Στην δημιουργία σεναρίων χρησιμοποιήθηκαν κυρίως αυτές οι μεταβλητές. Στην παρακάτω εικόνα 47 απεικονίζεται ο κώδικας που περιέχει τις μεταβλητές.



```
1 public class LubVar
2 {
3     private static bool pump1;
4     public static bool Pump1
5     {
6         get { return pump1; }
7         set { pump1 = value; }
8     }
9
10    private static bool pump2;
11    public static bool Pump2
12    {
13        get { return pump2; }
14        set { pump2 = value; }
15    }
16
17    private static bool pump3;
18    public static bool Pump3
19    {
20        get { return pump3; }
21        set { pump3 = value; }
22    }
23
24 }
```

Εικόνα 57 - Κώδικας "LubVar" περιέχει μεταβλητές του συστήματος λίπανσης του ατμοστροβίλου

Δημιουργήθηκε ένα αντικείμενο στην σκηνή με όνομα «Scenario1Script» στο οποίο προστέθηκε ο κώδικας «Scenario1.cs» σαν χαρακτηριστικό αντικειμένου. Αυτός ο κώδικας ελέγχει την λογική του σεναρίου.

Διαδικασία «Build»

Η διαδικασία «Build» είναι το κτίσιμο προσομοιωτή- εφαρμογή σε εκτελέσιμο αρχείο για συγκεκριμένο λειτουργικό σύστημα. Υπάρχουν αρχικά δύο γενικές επιλογές κτισίματος, η μία είναι «Build» και «Build and Run». Οι σκηνές «Scenes» έχουν υλοποιηθεί και για να προστεθούν στο κτίσιμο γίνεται από την επιλογή «Add Open Scenes».



Κεφάλαιο 4

Ο προσομοιωτής

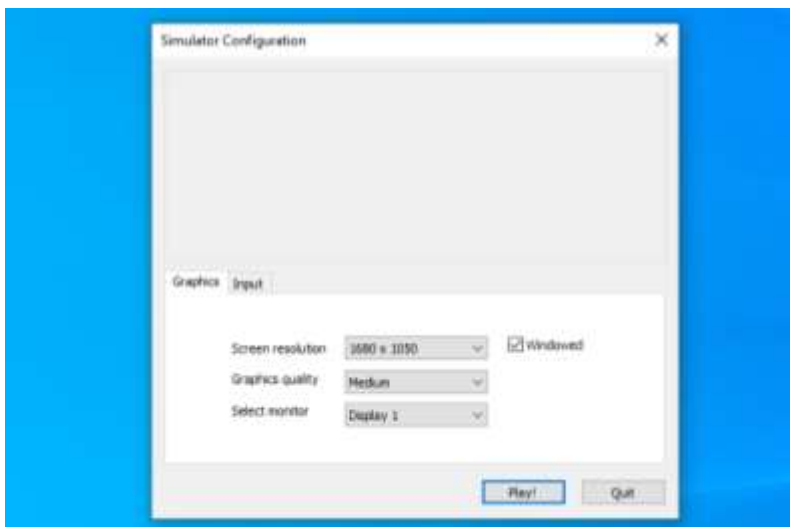
Η εφαρμογή αποτελείται από τα παρακάτω αρχεία όπως φαίνονται στην εικόνα 58.



Name	Date modified	Type	Size
MonobleedingEdge	4/5/2021 11:18 PM	File folder	
Simulator_Data	4/5/2021 11:25 PM	File folder	
Simulator	9/26/2019 4:29 AM	Application	634 KB
UnityContentHandler.dll	9/26/2019 4:54 AM	Application	1,408 KB
UnityPlayer.dll	9/26/2019 4:54 AM	Application exten...	40,880 KB
WinPcEventRuntime.dll	9/26/2019 4:26 AM	Application exten...	33 KB

Εικόνα 58 - Τα στοιχεία από την εγκατάσταση της εφαρμογής.

Για να τρέξει η εφαρμογή πατάμε το «Application» με το όνομα «Simulator». Τρέχει η εφαρμογή και εμφανίζεται το παράθυρο που απεικονίζεται στην παρακάτω εικόνα 59.



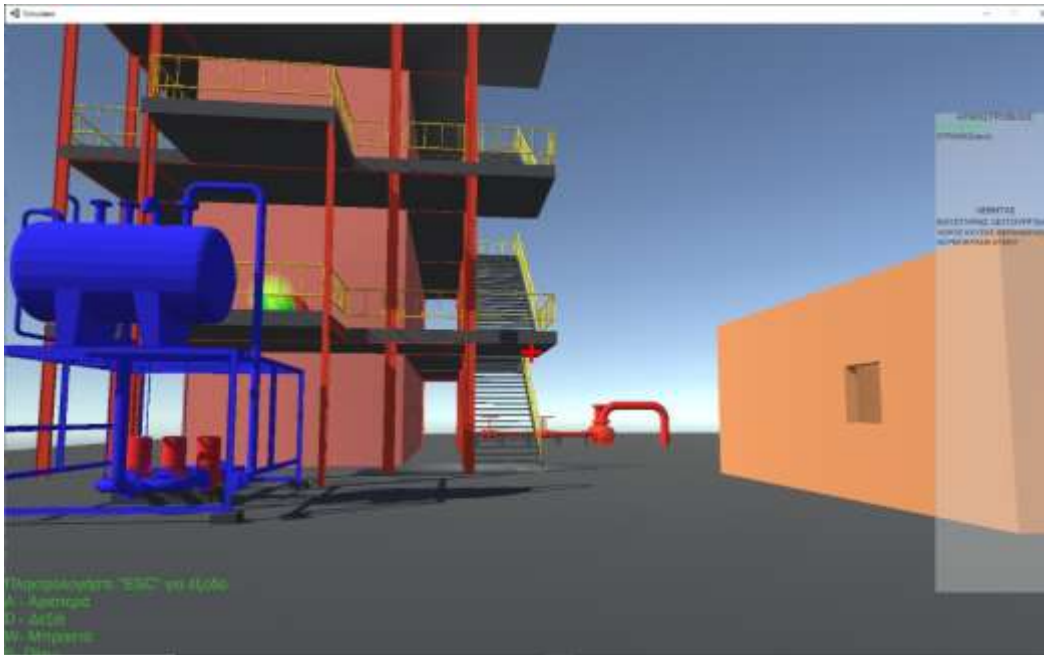
Εικόνα 59 - Παράθυρο ρυθμίσεων εφαρμογής.

Προτείνονται οι προεπιλεγμένες ρυθμίσεις όπως απεικονίζονται στην παραπάνω εικόνα 59. Πατώντας το «Play!» τρέχει την εφαρμογή που υλοποιήθηκε . Εμφανίζεται το μενού του προσομοιωτή.



Εικόνα 60 - Αρχικό μενού προσομοιωτή και καταχώρηση ονοματεπώνυμου

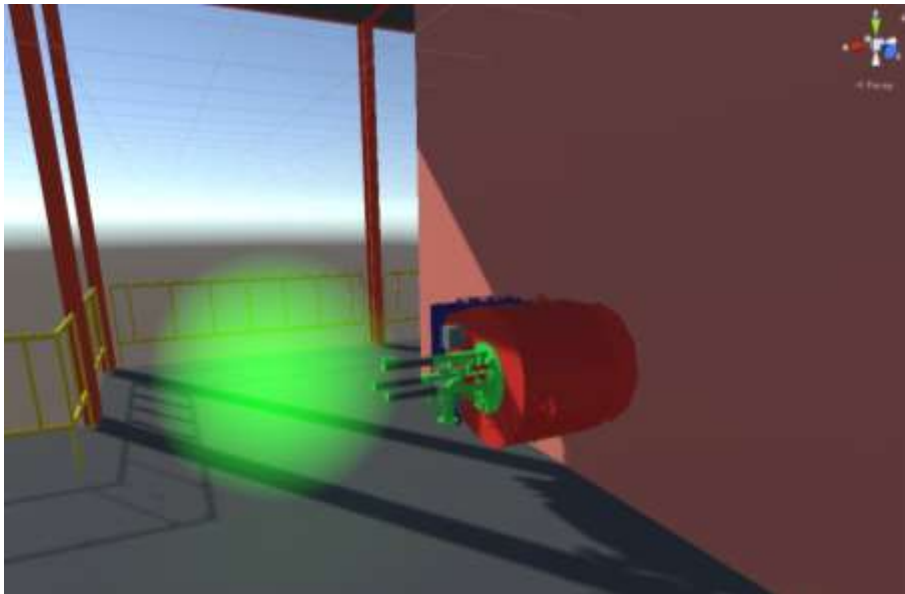
Αρχικά ο χρήστης θα πρέπει να εισάγει το ονοματεπώνυμο του και να πατήσει «ΚΑΤΑΧΩΡΗΣΗ» για να οριστεί το ονοματεπώνυμο. Έπειτα πατάει στο «ΣΕΝΑΡΙΑ» και αφού εμφανιστούν τα σενάρια, επιλέγει το σενάριο που επιθυμεί. Όταν γίνει αυτό θα εμφανιστεί το γραφικό περιβάλλον του προσομοιωτή όπως απεικονίζεται στην παρακάτω εικόνα 61.



Εικόνα 61 - Εμφανίζεται το γραφικό περιβάλλον κατά την είσοδο του χρήστη.

ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΕΞΑΡΤΗΜΑΤΑ

Τα συστήματα και τα εξαρτήματα στα οποία μπορεί ο χρήστης να αλληλεπιδράσει, περιέχουν ένα «Point Light» με πράσινο χρώμα στα οποία ο χρήστης όταν πλησιάζει αλληλοεπιδράει με αυτά εμφανίζοντας ένα «User Interface». Η περιοχή αλληλεπίδρασης είναι με το πράσινο φως όπως απεικονίζεται στην παρακάτω εικόνα 62.



Εικόνα 62 - Στο πράσινο φως όταν ο χρήστης το πλησιάζει αλληλοεπιδράει με το σύστημα.

Υπάρχουν τέτοια σημεία και στα υπόλοιπα αντίστοιχα συστήματα και εξαρτήματα. Έτσι όταν ο χρήστης πλησιάζει τα συστήματα ή εξαρτήματα αλληλοεπιδράει με αυτά και εμφανίζεται ένα σχετικό «User Interface» του συστήματος. Ένα τέτοιο είναι του καυστήρα όπως φαίνεται στην παρακάτω εικόνα 63. Πατώντας ο χρήστης «ΕΚΚΙΝΗΣΗ» ή «ΚΡΑΤΗΣΗ» θα ορίζει αν ο καυστήρας είναι σε λειτουργία ή όχι αντίστοιχα. Στα δεξιά φαίνεται ένα αντικείμενο «Sliderbar» όπου ο χρήστης ορίζει την ισχύ του καυστήρα.



Εικόνα 63 - Αλληλεπίδραση με τον καυστήρα "User Interface"

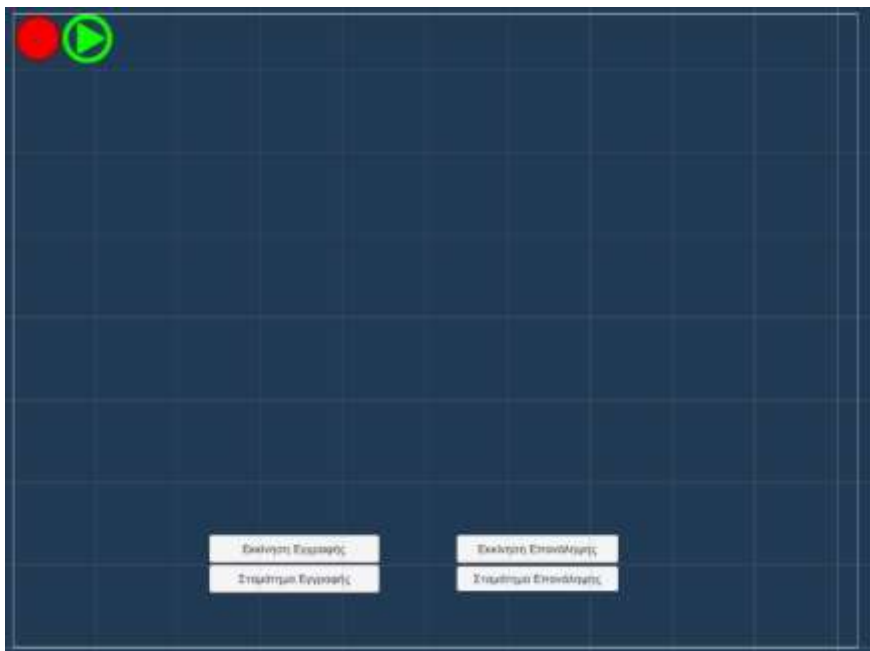


Εγγραφή κίνησης του χρήστη

Η συγκεκριμένη λειτουργία προσφέρει στον επιτηρητή του χρήστη να κάνει εγγραφή την κίνηση του χρήστη και έπειτα απεικόνιση στο ίδιο το γραφικό περιβάλλον. Ο κώδικας που χρησιμοποιήθηκε για την εγγραφή των κινήσεων του χρήστη ανακτήθηκε από την ιστοσελίδα «GitHub» στον παρακάτω σύνδεσμο «<https://github.com/ElmarTalibzade/GhostRecorder>». Η συγκεκριμένη εφαρμογή δημιουργήθηκε ειδικά για έργα στο Unity 3D κυρίως για την εγγραφή και επανάληψη κίνησης ενός αντικείμενου με κυρία στοιχεία αντικείμενου την τοποθεσία.

Στην παρακάτω εικόνα 64 απεικονίζεται το «User Interface» της διαδικασίας εγγραφής και επανάληψης. Στα αριστερά της εικόνας εμφανίζεται το κόκκινο σημάδι εγγραφής όταν κάνει εγγραφή. Το πράσινο όταν γίνεται επανάληψη εγγραφής και τα αντίστοιχα «Buttons» για την αλληλεπίδραση.

Πατώντας το πλήκτρο «R» στο πληκτρολόγιο γίνεται η αλληλεπίδραση με το «User Interface» και δίνεται η δυνατότητα εγγραφής και επανάληψης της κίνησης του χρήστη.

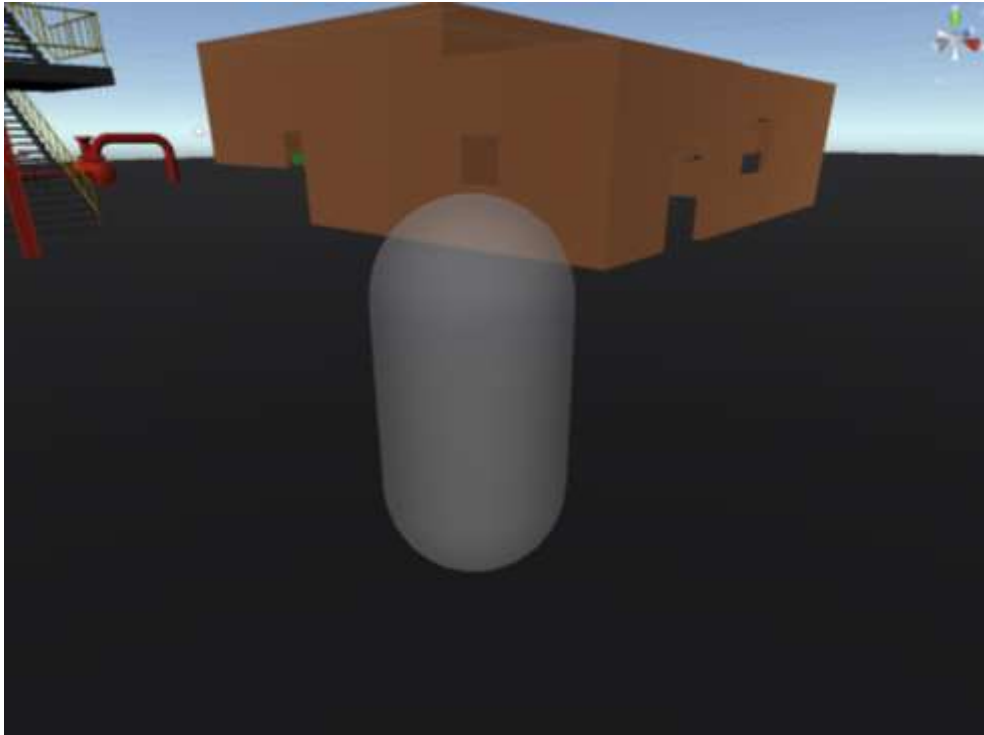


Εικόνα 64 - "User Interface" εγγραφής και επανάληψης κίνησης του χρήστη στο γραφικό περιβάλλον

Πατώντας «Εκκίνηση Εγγραφής» το πρόγραμμα αποθηκεύει την τοποθεσία του χρήστη στο γραφικό περιβάλλον ανά καρτέ και εκεί όπου υπάρχει επιθυμία πατώντας το «Σταμάτημα Εγγραφής». Αποθηκεύει προσωρινά της συντεταγμένες όπου κινήθηκε ο χρήστης στο γραφικό περιβάλλον. Έπειτα όταν πατηθεί «Εκκίνηση επανάληψης» θα εμφανιστεί μια



κάψουλα με διάφανο γκρι χρώμα και θα ξεκινήσει να κινείται στις συντεταγμένες του χρήστη που έγινε η εγγραφή. Δίνεται η δυνατότητα στο πρόγραμμα αυτό να οριστεί ο χρόνος εγγραφής. Στην περίπτωση αυτή έχει οριστεί στο ένα λεπτό διάρκεια. Κάθε νέα εγγραφή σβήνει την προηγούμενη.



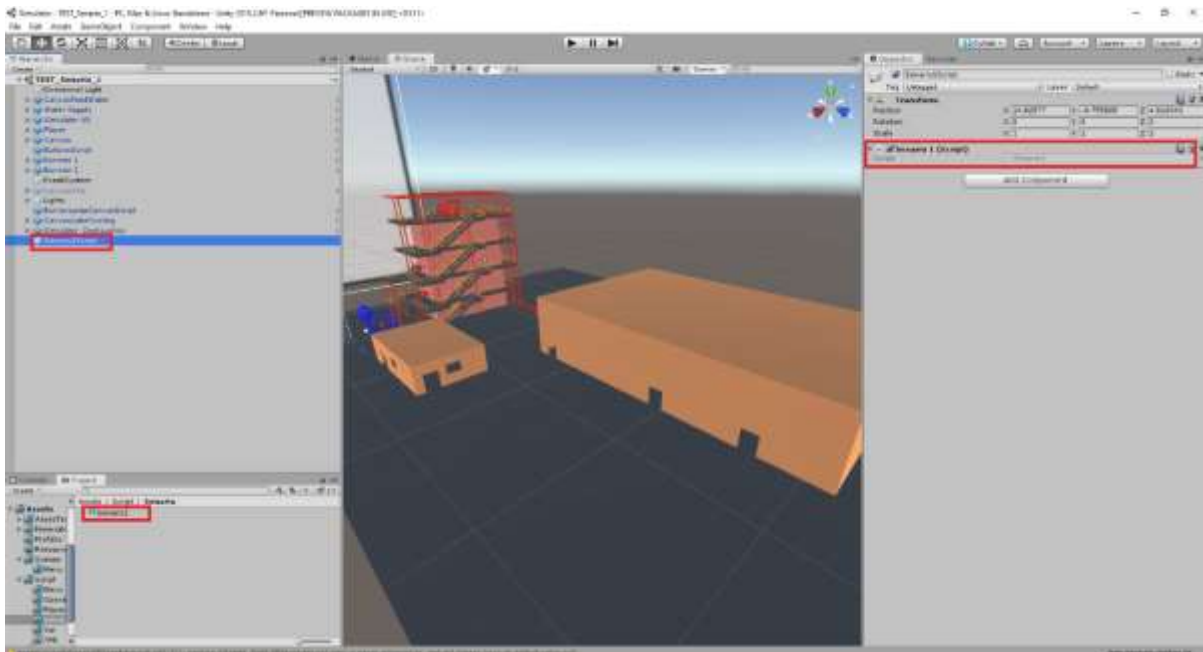
Εικόνα 65 - Κάψουλα που αντιπροσωπεύει τον χρήστη για την επανάληψη κίνησης του στο γραφικό περιβάλλον

Το συγκεκριμένο πρόγραμμα έχει το μειονέκτημα ότι τις συντεταγμένες του χρήστη όταν κινείται στο γραφικό περιβάλλον τις αποθηκεύει σε μια προσωρινή θέση μνήμης και δεν είναι διαθέσιμες όταν δημιουργηθεί μια νέα εγγραφή ή όταν κλείσει η εφαρμογή.

Σενάριο 1

Για την δημιουργία σεναρίου δημιουργήθηκε μια νέα «Scene» με όνομα «Test_Senario_1» στην Unity με όλα τα στοιχεία που έχουε προκατασκευαστεί. Τα πρόσθετα αντικείμενα σε αυτή την σκηνή είναι το «Senario1Script» με πρόσθετό χαρακτηριστικό αντικειμένου τον κώδικα «Senario1.cs». Ο κώδικας είναι αυτός που ελέγχει την λογική του σεναρίου.

Στην παρακάτω εικόνα 66 απεικονίζονται αυτά τα στοιχεία με κόκκινη διαγράμμιση.



Εικόνα 66 - Πρόσθετα στοιχεία για το σενάριο 1

Το σενάριο ένα είναι μια στοιχειώδη προετοιμασία για εκκίνηση ενός ατμοστρόβιλου.

Η λογική του σεναρίου έχει την αναφερόμενη σειρά και είναι η εξής:

1. Εκκίνηση δύο τροφοδοτικών αντλιών νερού λέβητα.
2. Εκκίνηση δύο καυστήρων του λέβητα.
3. Εκκίνηση δύο ή μιας αντλίας λίπανσης ατμοστρόβιλου.
4. Εκκίνηση μιας αντλίας ψύξης λαδιού λίπανσης ατμοστρόβιλου.
5. Κρίκος αργής περιστροφής ατμοστρόβιλου.

Ουσιαστικά ο χρήστης θα πρέπει να υλοποιήσει τις παραπάνω εντολές με την σειρά. Όταν ο χρήστης ολοκληρώσει την διαδικασία θα έχει περάσει το σενάριο με επιτυχία. Για κάθε σφάλμα που κάνει ο χρήστης εμφανίζεται αποτυχία στο σενάριο.

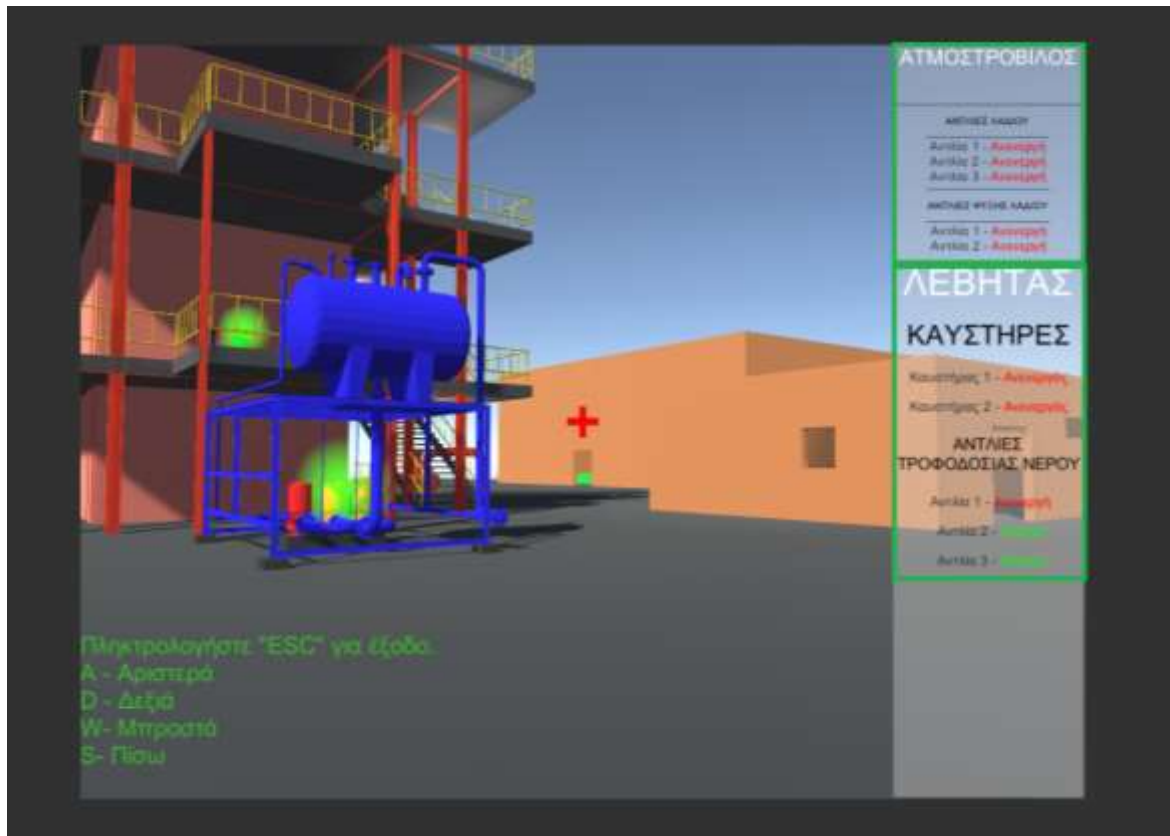
Στην παρακάτω εικόνα 67 στα δεξιά απεικονίζονται τα υποσυστήματα σε τι κατάσταση βρίσκονται. Αυτά είναι τα εξής:

Λέβητας:

- Καυστήρες
- Αντλίες τροφοδοσίας νερού λέβητα.

Ατμοστρόβιλος:

- Κρίκος αργής περιστροφής
- Αντλίες λαδιού
- Αντλίες ψύξης λαδιού



Εικόνα 67 - Απεικονίζει τα συστήματα σε τι κατάσταση βρίσκονται στα δεξιά

Συμπεράσματα

Ο προσομοιωτής προσφέρει στον χρήστη του την εκπαίδευση και την εξοικείωση του με τον χώρο της εγκατάστασης. Για παράδειγμα, αν ο χρήστης δεν έχει βρεθεί ξανά στην εγκατάσταση θα μπορούσε να εξοικειωθεί με αυτήν πριν βρεθεί στην πραγματική.

Στο προσομοιωτή που κατασκευάστηκε δημιουργήθηκαν εξαρχής τα σχέδια της εγκατάστασης. Για μία πραγματική εγκατάσταση δεν θα ήταν αναγκαίο αυτό, εφόσον το πιο πιθανό θα ήταν τα σχέδια να προϋπήρχαν και έτσι ο χρόνος εργασίας που απαιτείται μειώνεται αρκετά. Τα λογισμικά σχεδιασμού που χρησιμοποιήθηκαν είναι διαδομένα σε σχεδιασμό βιομηχανικής εγκατάστασης παραγωγής ηλεκτρικού ρεύματος.



Σε μια εξελιγμένη μορφή του προσομοιωτή θα μπορούσαν να χρησιμοποιηθούν τα ακόλουθα:

- Προσθήκη ηχητικών εφέ
- Δημιουργία ενός συστήματος παρακολούθησης του χρήστη ή περισσότερων από ένα μόνιτορ στο οποίο θα βρίσκεται ο επιτηρητής.
- Δυνατότητα είσοδου περισσότερων χρηστών στον προσομοιωτή παράλληλα μεσώ ενός διακοσμητή στο διαδίκτυο.
- Βελτιώσει της καταγραφής αλληλεπιδράσεων πχ. εξαγωγή σε έγγραφο κειμένου.
- Φιλοξενία του προσομοιωτή σε διακοσμητή στο διαδίκτυο το οποίο θα δίνει το πλεονέκτημα αναβαθμίσεων και απομακρυσμένης σύνδεσης.

Θεωρήθηκε ότι στο πλαίσιο μιας διπλωματικής εργασίας το επίπεδο της παρούσας προσομοίωσης επιτελεί επαρκώς τον σκοπό της.



Εικόνες

Εικόνα 1 - το περιβάλλον της εφαρμογής Unity3d	8
Εικόνα 2 - Απεικόνιση του Revit μοντέλου στον υπολογιστή και αντίστοιχα η απεικόνισή του στην Unity Reflect στο tablet	10
Εικόνα 3 - Σχηματική απεικόνιση λειτουργίας Revit με Unity Reflect	11
Εικόνα 4 - Περιβάλλον Unity Reflect για τους θεατές	13
Εικόνα 5 - Φωτογραφία της εφαρμογής Blender το περιβάλλον του είναι ένα προκαθορισμένο από τους προγραμματιστές. Πηγή: Wikipedia	15
Εικόνα 6 - Εικόνα από προσομοιωτή καραβιού σε 4D, πηγή: wartsila.com	16
Εικόνα 7 - Σύνδεση αρχείων με μορφοποίηση CAD	18
Εικόνα 8 - - Revit 2020, σχέδιο εγκατάστασης προσομοιωτή , εξαγωγή εικόνας με την μέθοδο Render Render	19
Εικόνα 9 - Στο Revit μεταλλικοί δοκοί με συνδέσμους στον λέβητα (Rendering)	20
Εικόνα 10 - Μενού επιλογής ορατών στοιχείων	21
Εικόνα 11 - Εξαγωγή πρόσοψης "RIGHT" σε μορφοποίηση «CAD»	22
Εικόνα 12 - Μενού "Insert" στο Revit.	22
Εικόνα 13 - Συνδεμένα σχέδια στο Revit	23
Εικόνα 14 - Εξαγωγή αρχείου σε μορφοποίηση "FBX"	24
Εικόνα 15 - Αποθήκευση του σχεδίου	24
Εικόνα 16 - Απεικόνιση δυσδιάστατου σχεδίου από το Revit στο Plant 3D	25
Εικόνα 17 - Απεικονίζεται σύνδεση σχεδίων στο AutoCAD Plant 3D	26
Εικόνα 18 - Απεικονίζεται το τρισδιάστατο σχέδιο στο AutoCAD Plant 3D και συνδεδεμένο δυσδιάστατο σχέδιο από το Revit	27
Εικόνα 19 - Εντολή "QSELECT"	28
Εικόνα 20 - Αποθήκευση νέου σχεδίου με την εντολή "EXPORTTOAUTOCAD"	29
Εικόνα 21 - Επιλογή αντικειμένων και συγχώνευσή τους με την εντολή "Join"	30



Εικόνα 22 - Απεικονίσει συγχωνεύσεων και ομαδοποίηση	31
Εικόνα 23 - Προσθήκη νέου Material στην βιβλιοθήκη και στο αντικείμενο	32
Εικόνα 24 - Προσθήκη ιδιοτήτων στο Material.	33
Εικόνα 25 - Επιλογή material	33
Εικόνα 26 - Εξαγωγή σε μορφοποίηση «FBX»	34
Εικόνα 27 - Έργα στο Unity Hub και οι εκδόσεις	35
Εικόνα 28 - Περιβάλλον εργασίας Unity 3D Editor	36
Εικόνα 29 - Δημιουργία έργου με επιλογή Template 3D.....	38
Εικόνα 30 - Δημιουργία νέου έργου	38
Εικόνα 31 - Εισαγωγή τρισδιάστατου σχεδίου στην σκηνή και στον φάκελο Assets	39
Εικόνα 32 - Διαδικασία Unpack Prefab	40
Εικόνα 33 - Εισαγωγή Collider στα σκαλοπάτια.....	41
Εικόνα 34 - Δημιουργία αντικειμένου χωρίς χαρακτηριστικά	41
Εικόνα 35 - Απεικόνιση αντικειμένου με χαρακτηριστικά φωτισμού "Light"	42
Εικόνα 36 - Παράδειγμα δημιουργίας μια μεταβλητής με ενθυλάκωση "Properties and Encapsulation".	44
Εικόνα 37 - Απεικονίζεται το Collider το οποίο κινείται μέσα στο γραφικό περιβάλλον.	45
Εικόνα 38 - Κώδικας FPSMovement για την κίνηση μέσω του πληκτρολογίου.....	46
Εικόνα 39 - Κώδικας «FPSLookAround.cs» με σχόλια επεξήγησης.....	47
Εικόνα 40 - Δημιουργία "Canvas"	48
Εικόνα 41 - Δημιουργία "Prefab Assets"	49
Εικόνα 42 - Μεμονωμένο περιβάλλον επεξεργασίας ενός "Prefab Assets"	49
Εικόνα 43 -Παράδειγμα ενός "Interface" για τρεις αντλίες.....	50
Εικόνα 44 -Το σύστημα τροφοδοσίας νερού στον λέβητα στο περιβάλλον της Unity 3d Editor.	51
Εικόνα 45 - Κώδικας "CanvasEnterExit" εμφανίζει οποιοδήποτε αντικείμενο που δηλώνεται στην μεταβλητή panel	52



Εικόνα 46 - Ορισμός μεταβλητής panel με το αντικείμενο "CanvasFeedWater"	53
Εικόνα 47 - Κώδικας FeedWaterLampsCanvas τροποποιεί τα αντικείμενα που ορίζονται στις μεταβλητές 8 και 9 γραμμής κώδικα.	54
Εικόνα 48 - Σύστημα τροφοδότησης νερού λέβητα, αλληλεπίδραση χρήστη με αυτό.	55
Εικόνα 49 - Προσθήκη συνάρτησης σε ένα "Button" στην παράμετρο On Click()	56
Εικόνα 50 - Προσθήκη συνάρτησης "FeedWaterPump()"	56
Εικόνα 51 - Κώδικας "Buttons" η συνάρτηση FeedWaterPump	57
Εικόνα 52 - Κώδικας "Buttons.cs" και η συνάρτηση Burner() για το σύστημά καυστήρων	58
Εικόνα 53 - Δημιουργία αρχικού μενού	59
Εικόνα 54 - Εισαγωγή κώδικα και συνάρτησης στην παράμετρο On Click()	60
Εικόνα 55 – Κώδικας «LoadScene.cs»	60
Εικόνα 56 - Κώδικες με περιεχόμενο μεταβλητές για κάθε σύστημα ξεχωριστά	61
Εικόνα 57 - Κώδικας "LubVar" περιέχει μεταβλητές του συστήματος λίπανσης του ατμοστροβίλου	62
Εικόνα 58 - Τα στοιχεία από την εγκατάσταση της εφαρμογής	63
Εικόνα 59 - Παράθυρο ρυθμίσεων εφαρμογής.	63
Εικόνα 60 - Αρχικό μενού προσομοιωτή και καταχώρηση ονοματεπώνυμου	64
Εικόνα 61 - Εμφανίζεται το γραφικό περιβάλλον κατά την είσοδο του χρήστη.	65
Εικόνα 62 - Στο πράσινο φως όταν ο χρήστης το πλησιάζει αλληλοεπιδράει με το σύστημα	66
Εικόνα 63 - Αλληλεπίδραση με τον καυστήρα "User Interface"	66
Εικόνα 64 - "User Interface" εγγραφής και επανάληψης κίνησης του χρήστη στο γραφικό περιβάλλον	67
Εικόνα 65 - Κάψουλα που αντιπροσωπεύει τον χρήστη για την επανάληψη κίνησης του στο γραφικό περιβάλλον	68
Εικόνα 66 - Πρόσθετα στοιχεία για το σενάριο 1	69
Εικόνα 67 - Απεικονίζει τα συστήματα σε τι κατάσταση βρίσκονται στα δεξιά	70



Βιβλιογραφία

(2021). Ανάκτηση από unity3d.com.

AutoCAD Plant 3D Asti.com. (2021). Ανάκτηση από <https://www.asti.com/applied-software-guide-to-plant-3d/>

Blender WikiPedia. (2021). Ανάκτηση από [https://en.wikipedia.org/wiki/Blender_\(software\)](https://en.wikipedia.org/wiki/Blender_(software))

Docs Microsoft. (2021). Ανάκτηση από Docs Microsoft: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/using-properties>

Revit WikiPedia. (2021). Ανάκτηση από https://en.wikipedia.org/wiki/Autodesk_Revit

UN3. (2019, Νοέμβριος 19). Ανάκτηση από Blogs Unity 3D: <https://blogs.unity3d.com/2019/11/19/unity-reflect-arrives-in-december/>

Unity. (2021). *Unity3d*. Ανάκτηση από Unity3D: <http://www.unity3d.com>

unity3d. (χ.χ.). Ανάκτηση από unity3d: www.unity3d.com

WikiPedia. (2021). Ανάκτηση από https://en.wikipedia.org/wiki/Autodesk_3ds_Max

WikiPedia. (2021). Ανάκτηση από WikiPedia: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

WikiPedia. (2021). Ανάκτηση από WikiPedia: <https://el.wikipedia.org/wiki/Blender>