

Οπτικοποίηση και Ανάλυση Δεδομένων σε Περιβάλλον Εικονικής Πραγματικότητας με τη μηχανή γραφικών Unity 3D

Ασημακόπουλος Γεώργιος (Α.Μ: 2216)

Διπλωματική Εργασία

Επιβλέπων: Φούντος Ιωάννης

Ιωάννινα , Οκτώβριος , 2018



ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY OF IOANNINA

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω το καθηγητή μου, κύριο Φούντο Ιωάννη για τη πρόταση του θέματος αυτού και για τη βοήθεια και τις συμβουλές που μου παρείχε για τη τελική δημιουργία της εργασίας μου.

Επίσης, ένα μεγάλο ευχαριστώ στην οικογένεια μου και στους φίλους μου, που με στήριξαν όλα αυτά τα χρόνια των σπουδών μου και με βοήθησαν με κάθε δυνατό τρόπο!

Ιωάννινα, Οκτώβριος 2018.

Ασημακόπουλος Γιώργος.

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με την οπτικοποίηση δεδομένων σε ένα περιβάλλον εικονικής πραγματικότητας. Πιο συγκεκριμένα, τα δεδομένα που διαχειριζόμαστε αφορούν τα ατμοσφαιρικά χαρακτηριστικά μιας περιοχής της Ελλάδας. Διαβάζοντας τις τιμές των pixels μιας εικόνας τύπου GeoTIFF μπορούμε να εξάγουμε από αυτήν διάφορα δεδομένα ανάλογα με το τύπο της. Στη περίπτωση αυτή προσπαθούμε μέσω της μηχανής γραφικών Unity 3D να αποτυπώσουμε τη περιοχή αυτή με γραφικά στοιχεία με τη μορφή ενός Terrain και πάνω σε αυτό να απεικονίσουμε γραφήματα τα οποία μας υποδεικνύουν τις τιμές θερμοκρασίας της περιοχής όπως αυτές διαβάστηκαν από τα pixels της εικόνας. Ο χρήστης έχει τη δυνατότητα να θέσει αυτός το πλήθος των γραφημάτων, το χρώμα τους αλλά και τη σχηματική τους μορφή μέσα από ένα μενού επιλογών. Για την υλοποίηση της εργασίας χρησιμοποιήθηκε όπως αναφέρθηκε η μηχανή γραφικών Unity 3D, το πρόγραμμα Visual Studio 2017 για τη συγγραφή κώδικα στη γλώσσα προγραμματισμού C# και οι βιβλιοθήκες (πακέτα) BitMiracle.LibTiff και GDAL για το άνοιγμα και το διάβασμα εικόνων τύπου GeoTIFF.

Λέξεις κλειδιά: Οπτικοποίηση, Unity 3D, GeoTIFF, Terrain.

Abstract

This diploma thesis is about the visualization of data in a virtual reality environment. Specifically the data that we handle have to do with atmospheric characteristics of some area in Greece. By reading the values of the pixels of an image type GeoTIFF we can extract data depending type of image. In this case we try via game engine Unity 3D to capture this region by the form of a Terrain, in this Terrain we represent graphs all of which indicate the values of temperatures in this region such as reading by the pixels of the image. The user has the potential to set the number of graphs, the color and the shape type of the graph through a menu. For the implementation of the project was used the game engine Unity 3D, the program Visual Studio 2017 for writing code in programming language C# and the libraries (packages) BitMiracle.LibTiff and GDAL for the opening and reading images of type GeoTIFF.

Keywords: Visualization, Unity 3D, GeoTIFF, Terrain.

Πίνακας Περιεχομένων

Κεφάλαιο 1. Εισαγωγή.....	8
1.1 Η βασική δομή της Εργασίας.....	8
1.2 Ορισμός του προβλήματος.....	9
1.3 Πως αντιμετωπίσα το πρόβλημα.....	10
1.4 Άλλες προσεγγίσεις στο θέμα.....	11
Κεφάλαιο 2. Βασικές έννοιες και στοιχεία υποβάθρου.....	14
2.1 Μηχανή γραφικών Unity 3D.....	14
2.1.1 Γενικές Πληροφορίες.....	14
2.1.2 Ιστορικά στοιχεία.....	15
2.1.3 Περιβάλλον (Editor) της Unity 3D.....	15
2.1.4 Προγραμματισμός με Unity 3D.....	18
2.2 Αρχεία Εικόνων TIFF.....	19
2.2.1 Γενικές Πληροφορίες.....	19
2.2.2 Ιστορικά στοιχεία.....	21
2.2.3 Εικόνες GeoTIFF.....	21
2.2.4 Λήψη GeoTIFF εικόνων.....	23
2.2.5 Διαχείριση εικόνων GeoTIFF μέσω κώδικα.....	25
2.3 Terrain.....	26
2.3.1 Γενικές Πληροφορίες.....	26
2.3.2 Ο ρόλος του Terrain στη Unity 3D.....	27
2.4 Υψομετρικοί Χάρτες (Height Maps).....	28
2.4.1 Γενικές Πληροφορίες.....	28
2.4.2 Δημιουργία Υψομετρικών Χαρτών.....	29
2.4.3 Προγράμματα και Unity 3D.....	29
Κεφάλαιο 3. Οπτικοποίηση δεδομένων σε περιβάλλον εικ. πραγματικότητας.....	32
3.1 Εισαγωγή - Περίληψη του κεφαλαίου.....	32
3.2 Προγράμματα και βιβλιοθήκες.....	32
3.3 Σκηνές του προγράμματος.....	33
3.4 Τα αντικείμενα της κάθε σκηνής.....	36
3.5 Σχεδιασμός του Terrain από Height Map.....	38
3.6 Λειτουργίες του προγράμματος.....	43
3.7 Scripts.....	47
3.8 Δημιουργία εκτελέσιμου αρχείου.....	54

Κεφάλαιο 4. Πειραματική Αξιολόγηση.....	56
4.1 Ποιοτική Αξιολόγηση.....	56
4.2 Ποσοτική Αξιολόγηση.....	59
Κεφάλαιο 5. Συμπεράσματα	62
5.1 Προβλήματα που αντιμετώπισα.....	62
5.2 Γνώσεις που αποκόμισα.....	62
5.3 Γιατί επέλεξα τη Unity 3D ;.....	63
Βιβλιογραφία.....	64

Κεφάλαιο 1. Εισαγωγή

1.1 Η βασική δομή της Εργασίας

Στην ενότητα αυτή θα περιγραφεί η βασική δομή της εργασίας και θα αναλυθούν περιληπτικά τα στοιχεία και οι πληροφορίες των κεφαλαίων που θα ακολουθήσουν. Πιο αναλυτικά:

- Κεφάλαιο 1: Το πρώτο κεφάλαιο θα έχει εισαγωγικό χαρακτήρα. Αρχικά θα δοθεί ο ορισμός του προβλήματος που θα επιδιώξουμε να επιλύσουμε σε αυτή τη διπλωματική εργασία. Στη συνέχεια θα περιγραφούν συνοπτικά τα κυριότερα βήματα που ακολουθήθηκαν για την επίλυση του προβλήματος αυτού. Στο τελευταίο κομμάτι του κεφαλαίου θα αναλυθούν παρόμοιες εργασίες που έχουν ασχοληθεί με την Οπτικοποίηση δεδομένων σε περιβάλλον εικονικής πραγματικότητας.
- Κεφάλαιο 2: Το κεφάλαιο περιέχει βασικές έννοιες και στοιχεία υπόβαθρου, τα όποια επίδρασαν καταλυτικά στη πορεία της εργασίας και έπαιξαν αρκετά σημαντικό ρόλο στη δημιουργία της, πιο συγκεκριμένα στο κεφάλαιο αναφέρονται βασικοί ορισμοί, πληροφορίες, ιστορικά στοιχεία, εικόνες αλλά και παραδείγματα για τους παρακάτω όρους:
 1. Μηχανή γραφικών Unity 3D.
 2. Αρχεία εικόνων τύπου TIFF και GeoTIFF.
 3. Terrain.
 4. Υψομετρικοί Χάρτες (Height Maps).
- Κεφάλαιο 3: Σε αυτό το κεφάλαιο θα αναλυθεί βήμα προς βήμα η εργασία και πως αυτή υλοποιήθηκε. Αρχικά, θα αναφερθούν τα προγράμματα και οι βιβλιοθήκες (πακέτα) που χρησιμοποιήθηκαν και λίγα λόγια για αυτά. Στη συνέχεια θα περιγράψουν όλες οι λειτουργίες και σκηνές του προγράμματος, ο τρόπος που σχεδιάστηκε το Terrain και τα Scripts που δημιουργήθηκαν σε κώδικα C#, ποιο ρόλο έχει το κάθε ένα και πως βοήθησαν στη τελική δημιουργία της εργασίας.

- Κεφάλαιο 4: Στο κεφάλαιο αυτό υπάρχουν δύο βασικές παράγραφοι. Η πρώτη αφορά τη ποιοτική αξιολόγηση της εργασίας, εδώ θα υπάρχουν εικόνες (screenshots) από τη λειτουργία του προγράμματος και από την απεικόνιση των γραφημάτων πάνω στο Terrain. Η δεύτερη παράγραφος αφορά τη ποσοτική αξιολόγηση της εργασίας, σε αυτή παρουσιάζονται γράφοι όπου απεικονίζουν πως μεταβάλλονται τα FPS (Frames Per Second) της προσομοίωσης ανάλογα με το μέγεθος του πλέγματος (πλήθος γραφημάτων) πάνω στο Terrain.
- Κεφάλαιο 5: Το τελευταίο κεφάλαιο δρα ως επίλογος της εργασίας. Περιέχει τα δικά μου συμπεράσματα, τις γνώσεις που αποκόμισα από τη δημιουργία της εργασίας, τα προβλήματα και τις δυσκολίες που αντιμετώπισα κατά τη κατασκευή της αλλά και τους λόγους που επέλεξα το τομέα των Γραφικών σε Υπολογιστή και τη μηχανή γραφικών Unity 3D.

1.2 Ορισμός του προβλήματος

Το κύριο πρόβλημα που επιδιώκουμε να επιλύσουμε με αυτή τη διπλωματική εργασία, είναι να βρούμε ένα τρόπο έτσι ώστε τα ατμοσφαιρικά χαρακτηριστικά (θερμοκρασίες στο πρόβλημα μας) μια περιοχής της Ελλάδας και πιο συγκεκριμένα οι περιοχές γύρω από τη πόλη των Ιωαννίνων να αποτυπωθούν με τη μορφή γραφημάτων πάνω σε ένα Terrain (έδαφος) το οποίο θα περιγράφει γεωμορφολογικά την εν λόγω περιοχή. Ειδικότερα, αυτό που θέλουμε είναι, κάνοντας λήψη και διαβάζοντας τα βασικά χαρακτηριστικά μιας εικόνας τύπου GeoTIFF (εικόνες που περιγράφουν γεωγραφικά ένα κομμάτι/περιοχή της Γης) να τοποθετήσουμε πάνω στο Terrain γραφήματα όπου οι συντεταγμένες τους θα προσδιορίζονται από τις αντίστοιχες συντεταγμένες (γεωγραφικό μήκος και πλάτος) που θα εξάγουμε από την εικόνα και το ύψος των γραφημάτων θα προσδιορίζεται από τη τιμή που θα έχει η θερμοκρασία σε εκείνο το κομμάτι της περιοχής. Όσο μεγαλύτερος θα είναι ο μέσος όρος της θερμοκρασίας, τόσο μεγαλύτερο θα είναι και το ύψος του γραφήματος στη τρέχουσα τοποθεσία. Τα γραφήματα αυτά θα προβάλλονται στο χρήστη με τη βοήθεια γραφικών στοιχείων, σε αυτό το κομμάτι καταλυτικό ρόλο θα παίξει η μηχανή γραφικών Unity 3D, όπου είναι και το βασικό πρόγραμμα υλοποίησης της παρούσας διπλωματικής εργασίας.

1.3 Πως αντιμετωπίσα το πρόβλημα

Στην ενότητα αυτή, θα παρατεθούν κυρίως επιγραμματικά τα σημαντικότερα βήματα που έγιναν για τη δημιουργία ενός προγράμματος εικονικής πραγματικότητας όπου επιλύει το παραπάνω πρόβλημα όπως αυτό περιγράφηκε. Περισσότερες λεπτομέρειες για την υλοποίηση του προγράμματος και το πρακτικό κομμάτι της εργασίας θα αναφερθούν στο Κεφάλαιο 3 (με τίτλο: Οπτικοποίηση δεδομένων σε περιβάλλον εικονικής πραγματικότητας). Τα βασικά στάδια που ακολουθήθηκαν είναι τα εξής:

1. Αναζήτηση και εύρεση από το διαδίκτυο μιας εικόνας τύπου GeoTIFF από την ευρύτερη περιοχή των Ιωαννίνων, από την εικόνα αυτή θα εξαχθούν οι συντεταγμένες τοποθέτησης των γραφημάτων αλλά και ο μέσος όρος των θερμοκρασιών ανά περιοχή, οι μέσοι όροι θα υπολογιστούν από τις τιμές των pixel της εικόνας.
2. Άνοιγμα και διάβασμα της παραπάνω εικόνας με τη βιβλιοθήκη BitMiracle.LibTiff που παρέχεται δωρεάν με τη μορφή NuGet από το πρόγραμμα δημιουργίας κώδικα στη γλώσσα C# Visual Studio 2017. Αφού διαβάστηκαν επιτυχώς οι συντεταγμένες και οι τιμές θερμοκρασίας από την εικόνα, αποθηκεύτηκαν σε απλά αρχεία κειμένου (txt) για πιο εύκολη διαχείριση τους.
3. Λήψη, εγκατάσταση και δημιουργία ενός νέου project μέσα στο περιβάλλον της μηχανής γραφικών Unity 3D. Με το πρόγραμμα αυτό θα υλοποιηθεί και ο βασικός κορμός της διπλωματικής εργασίας. Η δημιουργία του Terrain που απεικονίζει τη περιοχή και που εν τέλει πάνω σε αυτό θα τοποθετηθούν τα γραφήματα είναι ένα τα πιο σημαντικά κομμάτια της εργασίας. Πιο συγκεκριμένα, το Terrain δημιουργήθηκε με τη βοήθεια μιας εικόνας που περιγράφει τη συγκεκριμένη περιοχή, η εικόνα αυτή μεταφέρθηκε μέσα στο φάκελο αποθήκευσης της εργασίας και χρησιμοποιώντας ένα script κώδικα στη γλώσσα JavaScript πάνω στο Terrain αποτυπώθηκαν τα μορφολογικά στοιχεία της περιοχής (βουνά, λόφοι κλπ), επίσης το Terrain χρωματίστηκε έτσι ώστε να μοιάζει πιο αληθοφανής στο ανθρώπινο μάτι.
4. Τα αρχεία κειμένου που δημιουργήθηκαν στο βήμα 2, διαβάστηκαν μέσω της Unity 3D και του Visual Studio 2017 και οι τιμές τους αποθηκεύτηκαν σε λίστες. Τα γραφήματα δημιουργούνται με τη μορφή grid (πλέγματος) και τοποθετούνται πάνω στο Terrain. Ο χρήστης έχει τη δυνατότητα να επιλέξει το μέγεθος, το χρώμα και το τύπο του σχήματος με το οποίο θα προβληθούν τα γραφήματα μέσα από ένα μενού επιλογών.

1.4 Άλλες προσεγγίσεις στο θέμα

Στην ενότητα αυτή, θα αναλυθούν περιληπτικά κυρίως άλλες εργασίες οι οποίες έχουν ασχοληθεί με το κομμάτι της οπτικοποίησης ατμοσφαιρικών δεδομένων. Στο κομμάτι της Βιβλιογραφίας αναφέρεται για κάθε εργασία ο τίτλος της και οι συγγραφείς της.

Εργασία 1 - Περίληψη:

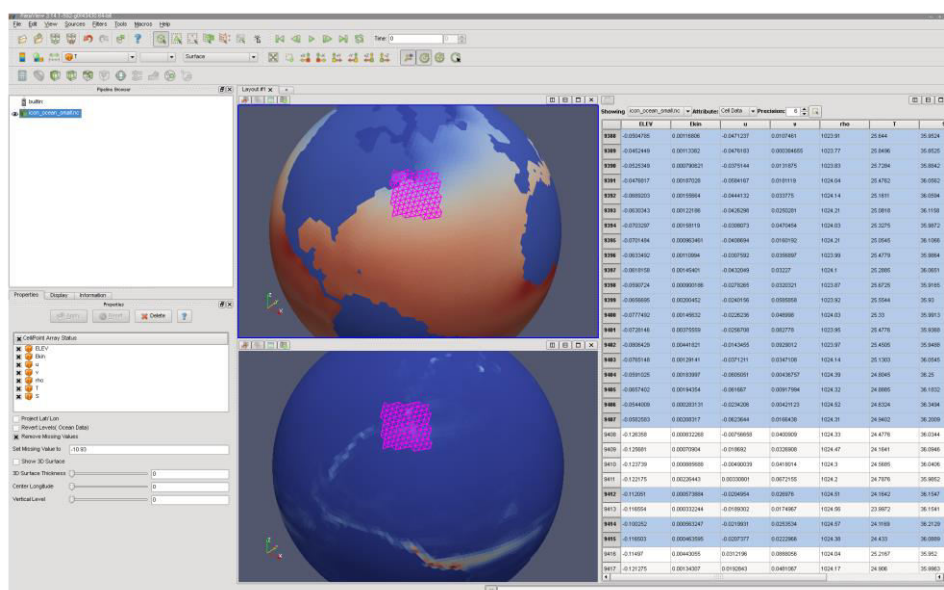
Στην εργασία [1] οι συγγραφείς μελετούν τις κλιματικές αλλαγές και το πως αυτές επιδράσουν στο περιβάλλον αλλά και στις συνθήκες ζωής των ανθρώπων. Οι προσομοιώσεις του καιρού και του κλίματος οι οποίες προβλέπουν πιθανές αλλαγές παράγουν μεγάλο όγκο δεδομένων. Ο συνδυασμός διαφόρων μεταβλητών που προέρχονται από κλιματικά μοντέλα και χωρικών δεδομένων από διαφορετικές πηγές μας βοηθούν να αναγνωρίσουμε συσχετίσεις και να μελετήσουμε σημαντικές διαδικασίες. Σε αυτή την εργασία, τα αποτελέσματα ερευνών και προγνωστικών μοντέλων καιρού οπτικοποιούνται για δύο περιοχές. Για το σκοπό αυτό, ένα συνεχόμενο σύνολο εργασιών αναπτύχθηκε το οποίο μεταφέρει τα ανεπεξέργαστα δεδομένα σε 3D οπτικοποίηση η οποία μπορεί να προβληθεί μέσω υπολογιστή ή μέσω ενός περιβάλλοντος εικονικής πραγματικότητας. Για την υλοποίηση της εργασίας οι συγγραφείς ακολούθησαν τρία βασικά στάδια, τα όποια είναι:

1. Ενοποίηση των δεδομένων: στο στάδιο αυτό έγινε η προ-επεξεργασία των αρχικών δεδομένων.
2. Μέθοδοι οπτικοποίησης και σύνθεσης δεδομένων: εδώ περιγράφονται μέθοδοι οπτικοποίησης οι οποίες ελέγχθηκαν και χρησιμοποιήθηκαν στο πρόβλημα.
3. Παρουσίαση και αξιολόγηση: στο τελικό αυτό στάδιο προσδιορίζονται και αποτιμούνται όλες οι διαδικασίες και οι μέθοδοι που υλοποιήθηκαν.

Αξίζει να σημειωθεί ότι ο τομέας της οπτικοποίησης ατμοσφαιρικών δεδομένων απαιτεί τη στενή συνεργασία ειδικών της οπτικοποίησης αλλά και της μετεωρολογίας. Η εργασία αυτή μας δείχνει τα κυριότερα αποτελέσματα αυτής της συνεργασίας. Για περαιτέρω ανάπτυξη της ιδέας, σχεδιάζεται και η υποστήριξη περιβάλλοντος εικονικής πραγματικότητας (VR). Αυτό το κομμάτι μπορεί να επιτευχθεί χρησιμοποιώντας τη μηχανή γραφικών Unity 3D, ένα λογισμικό το οποίο προσφέρει δημιουργία εικονικής πραγματικότητας.

Εργασία 2 - Περίληψη:

Στην εργασία [2] πραγματοποιείται χρήση και ανάλυση του προγράμματος ParaView. Το ParaView είναι ένα εκ των κορυφαίων λογισμικών οπτικοποίησης για 4D (3D χώρος + χρόνος) δεδομένα. Είναι ένα πανίσχυρο εργαλείο, σχεδιασμένο να οπτικοποιεί δεδομένα σε πολλά διαφορετικά πεδία. Ωστόσο, λόγω αυτής της ιδιότητας, συχνά απαιτείται μεγάλη προ-επεξεργασία και χειρισμός δεδομένων για τη δημιουργία σωστών και εξειδικευμένων οπτικοποιήσεων. Με το πακέτο `pv_atmos` επιτρέπεται η δημιουργία 4D οπτικοποιήσεων ατμοσφαιρικών δεδομένων σε `netCDF` (network Common Data Form) μορφή. Χρησιμοποιώντας κώδικα σε γλώσσα Python, το λογισμικό ParaView φορτώνει, επεξεργάζεται και οπτικοποιεί δεδομένα. Το πακέτο αυτό είναι αποκλειστικά γραμμένο στη γλώσσα Python και δεν απαιτείται εγκατάσταση της, ούτε κάποιο άλλο βοηθητικό πακέτο ή πρόγραμμα εκτός από το ParaView. Οι ρουτίνες που περιέχει έχουν σχεδιαστεί να τρέχουν με το Shell της Python που βρίσκεται μέσα στο ParaView. Ο κώδικας αυτοματοποιεί τη φόρτωση των δεδομένων γεωγραφικού μήκους και πλάτους (longitude - latitude) και δημιουργεί ένα πλέγμα. Στη συνέχεια, ξανά υπολογίζει τη μορφή του πλέγματος και παράγει μια σφαιρική αναπαράσταση όλων των δεδομένων.



Εικόνα 1.1. Διαχείριση ατμοσφαιρικών δεδομένων με το πρόγραμμα ParaView.
(Πηγή: Google)

Εργασία 3 - Περίληψη:

Η εργασία [3] ερευνά την ιστορία αλλά και τη τρέχουσα κατάσταση της οπτικοποίησης δεδομένων στο κλάδο της Μετεωρολογίας, εστιάζοντας κυρίως σε τεχνικές και εργαλεία τα οποία χρησιμοποιούνται για την ανάλυση τέτοιων δεδομένων. Οι συγγραφείς εξετάζουν τα χαρακτηριστικά των δεδομένων και αναλύουν εργασίες και περιγράφουν την εξέλιξη των γραφικών μεθόδων σε υπολογιστή για οπτικοποίηση στη Μετεωρολογία από τη δεκαετία του '60 μέχρι και σήμερα. Επίσης, παρουσιάζουν τις τελευταίες τεχνικές και εργαλεία που χρησιμοποιούνται για τη πρόγνωση του καιρού και γενικότερα για την ατμοσφαιρική έρευνα. Προσεγγίζουν το θέμα και από τη πλευρά της οπτικοποίησης αλλά και της Μετεωρολογίας, προβάλλοντας τεχνικές που χρησιμοποιούνται συνήθως σε εφαρμογές της. Η επισκόπηση αυτή καλύπτει τεχνικές οπτικοποίησης από τα πεδία της 3D οπτικοποίησης, της δυναμικής ροής, της διαδραστικής οπτικής ανάλυσης, της αποτελεσματικής απόδοσης και της κλιμάκωσης. Το άρθρο πραγματεύεται απαιτήσεις και προκλήσεις για έρευνα στο κομμάτι της οπτικοποίησης η οποία στοχεύει στην ανάλυση μετεωρολογικών δεδομένων, τονίζοντας απόψεις για τα πλεονεκτήματα της διαδραστικής οπτικής ανάλυσης, της 3D οπτικοποίησης αλλά και άλλων τεχνικών θεμάτων.

Κεφάλαιο 2. Βασικές έννοιες και στοιχεία υποβάθρου

2.1 Μηχανή γραφικών Unity 3D

2.1.1 Γενικές Πληροφορίες

Η Unity είναι μια μηχανή γραφικών [4] η οποία αναπτύχθηκε από την εταιρεία Unity Technologies [5] με πρωταρχικό σκοπό την ανάπτυξη 3D αλλά και 2D βίντεο παιχνιδιών και προσομοιώσεων για υπολογιστές, κονσόλες και κινητά. Αρχικά δημιουργήθηκε μόνο για συστήματα OS X το 2005, αλλά στη συνέχεια επεκτάθηκε και σήμερα υποστηρίζει συνολικά 27 πλατφόρμες. Έξι σημαντικές εκδόσεις της μηχανής έχουν εκδοθεί από την αρχική της δημιουργία. Μέχρι πρόσφατα, υποστήριζε τη κατασκευή μόνο 3D παιχνιδιών αλλά αυτό άλλαξε μετά την έκδοση 4.6. Πλέον υποστηρίζει και τη δημιουργία παιχνιδιών εικονικής πραγματικότητας. Μέσω αυτής έχουν δημιουργηθεί αρκετά δημοφιλή και εμπορικά βίντεο παιχνίδια. Επίσης, οι χρήστες μέσω της βάσης δεδομένων που προσφέρει το Unity Store [8], έχουν τη δυνατότητα να κατεβάσουν δωρεάν, να αγοράσουν ή ακόμη και να πουλήσουν assets σε άλλους χρήστες. Είναι μια μηχανή πολλαπλών χρήσεων η οποία υποστηρίζει 2D και 3D γραφικά και λειτουργίες "drag-and-drop". Η Unity υποστηρίζει πολλές διαφορετικές πλατφόρμες, οι σημαντικότερες από αυτές είναι οι εξής: iOS, Android, Windows, Mac, Linux, PlayStation 4, Xbox One, Nintendo Switch και άλλες. Στο παρελθόν υποστηρίζονταν άλλες 7 πλατφόρμες συμπεριλαμβανομένου και του δικού της Web Player, ο οποίος ήταν ένα είδος επέκτασης (plug-in) για browsers.



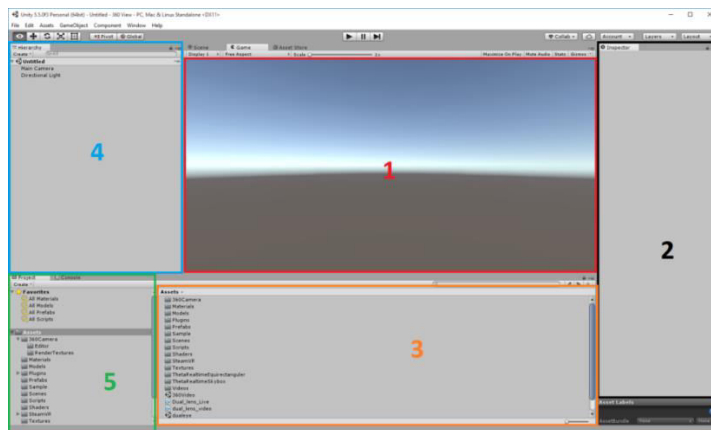
Εικόνα 2.1. Το επίσημο λογότυπο της εταιρείας. (Πηγή: Google)

2.1.2 Ιστορικά στοιχεία

Ιστορικά, η πρώτη έκδοση της Unity (1.0.0) δημιουργήθηκε από τους φοιτητές David Helgason, Joachim Ante και Nicholas Francis στη Δανία το 2004, η αρχική ιδέα όμως ξεκίνησε αρκετά νωρίτερα (2002) από ένα δημοσίευμα του Δανού Nicholas Francis και έπειτα ακολούθησαν και οι υπόλοιποι δύο προγραμματιστές. Το αρχικό προϊόν εμπορικής χρήσεως παρουσιάστηκε τον Ιούνιο του 2005. Ο στόχος ήταν να δημιουργηθεί μια προσιτή σε όλους μηχανή γραφικών με επαγγελματικά εργαλεία οι οποία βοηθά ερασιτέχνες σχεδιαστές βίντεο παιχνιδιών να είναι σε θέση να δημιουργήσουν το δικό τους παιχνίδι εύκολα και απλά. Η εταιρεία τελικά ιδρύθηκε στη Δανία την ίδια χρονιά με το όνομα Over the Edge I/S και άλλαξε το 2006 αφού μετονομάστηκε σε Unity Technologies ApS. Το 2009, μετακίνησε τα κεντρικά της γραφεία στο San Francisco και σήμερα απασχολεί πάνω από 1500 υπαλλήλους παγκοσμίως, πλέον το επίσημο όνομα της είναι Unity Technologies SF. Η τελευταία έκδοση της μηχανής (2018.2.11) παρουσιάστηκε τον Οκτώβριο του 2018. Η μηχανή μπορεί εύκολα να χρησιμοποιηθεί από όλους τους χρήστες αφού υπάρχει δωρεάν έκδοση της στην [επίσημη σελίδα](#) της [6], βέβαια υπάρχουν και εκδόσεις επί πληρωμή οι οποίες προσφέρουν περισσότερες δυνατότητες για επαγγελματικούς και εμπορικούς σκοπούς.

2.1.3 Περιβάλλον (Editor) της Unity 3D

Το περιβάλλον της μηχανής αποτελείται από διάφορα παράθυρα τα οποία κάνουν την εργασία εύκολα διαχειρίσιμη. Ο χρήστης έχει τη δυνατότητα να δημιουργήσει το δικό του Interface καθώς μπορεί να μεταφέρει τα παράθυρα εύκολα και να επιλέξει αυτός πως θα φαίνεται ο Editor της μηχανής. Στην εικόνα 2.2 παρουσιάζονται τα κύρια παράθυρα και τα σημαντικότερα χαρακτηριστικά αυτών που χρησιμοποιούνται από τη μηχανή και βοήθησαν στη δημιουργία της εργασίας.



Εικόνα 2.2. Το περιβάλλον (editor) της μηχανής γραφικών Unity 3D.

Στις παραγράφους που ακολουθούν αναφέρονται λίγα λόγια για το κάθε παράθυρο.

1. Scene: Ένας αρχικά άδειος χώρος, στον οποίο μπορούμε να βλέπουμε τα αντικείμενα (game objects) που έχουμε προσθέσει στο παιχνίδι μας. Υπάρχουν κάποια έτοιμα και διαθέσιμα αντικείμενα από τη μηχανή, αν θέλουμε κάτι εξειδικευμένο υπάρχουν κάποια πακέτα μέσω του Store που παρέχεται. Υπάρχει η καρτέλα "Game" η οποία μας δείχνει πως είναι ακριβώς το παιχνίδι μας αφού το θέσουμε σε λειτουργία. Στο σημείο αυτό αξίζει να σημειωθεί ότι για τη καρτέλα αυτή υπάρχουν τρία βασικά κουμπιά, αρχικά το Play με το οποίο τρέχει η εργασία μας και μπορούμε να έχουμε μια ιδέα της τελικής της μορφής, το Pause όπου μπορούμε να "παγώσουμε" το παιχνίδι και να επεξεργαστούμε λεπτομέρειες αντικειμένων που κρίνουμε εμείς και τέλος το κουμπί Next Frame με το οποίο βλέπουμε με ακρίβεια το κάθε frame (καρέ) του παιχνιδιού.
2. Inspector: Το παράθυρο αυτό περιέχει τα στοιχεία και τις ιδιότητες ενός επιλεγμένου αντικειμένου από την ιεραρχία (παράθυρο 4). Μας υποδεικνύει τη θέση του αντικειμένου στη σκηνή, τη κλιμάκωση που έχει υποστεί αλλά και τα scripts που είναι συνδεδεμένα με αυτό.
3. Assets: Στο κομμάτι αυτό αποθηκεύονται όλα τα αρχεία που έχουν σχέση με την εργασία μας. Στο παράθυρο αυτό μπορούμε να δημιουργήσουμε φακέλους για καλύτερη διαχείριση μεγάλων projects, στους φακέλους μπορούμε να έχουμε εικόνες/γραφικά αλλά και scripts που έχουν σχέση με τα game objects της σκηνής μας.
4. Hierarchy: Περιέχει τη λίστα με τα game objects που έχουμε στη σκηνή μας, επιλέγοντας ένα από τα αντικείμενα μπορούμε να το επεξεργαστούμε μέσω του Inspector και να βλέπουμε άμεσα το αποτέλεσμα στη σκηνή.
5. Project/Debug Console: Το παράθυρο αυτό περιέχει σε μορφή λίστας τους φακέλους που έχουμε δημιουργήσει στο Assets. Επίσης, υπάρχει και η κονσόλα η οποία μας δείχνει ότι εκτυπώνουμε μέσω των scripts αλλά και τα όποια σφάλματα και ειδοποιήσεις υπάρχουν καθώς εκτελείται το παιχνίδι.

Εκτός από τα κύρια παράθυρα του Editor που περιγράφηκαν παραπάνω, υπάρχει και το βασικό μενού (μπάρα) της μηχανής στο πάνω αριστερό κομμάτι της οθόνης. Στις παρακάτω παραγράφους θα δοθεί μια συνοπτική περιγραφή για τη κάθε επιλογή που περιέχει το βασικό μενού όπως δείχνει και η εικόνα 2.3:

File Edit Assets GameObject Component Window Help

Εικόνα 2.3. Το βασικό μενού (μπάρα) της Unity 3D.

Αναλυτικότερα για τις επιλογές που μας προσφέρει το βασικό μενού:

- **File**: Με τη καρτέλα αυτή μπορούμε να δημιουργήσουμε μια νέα σκηνή στην εργασία μας, να αποθηκεύσουμε ή να εισάγουμε μια ήδη υπάρχουσα, να αποθηκεύσουμε όλη την εργασία ή να εισάγουμε μια παλιότερη, επίσης εκεί υπάρχει και η επιλογή Build, με την οποία η εργασία μας γίνεται εκτελέσιμο αρχείο σε όποια πλατφόρμα επιλεγεί από το δημιουργό της.
- **Edit**: Η καρτέλα αυτή, επιτρέπει στο χρήστη της μηχανής να κάνει ενέργειες πάνω στην εργασία, όπως θα έκανε και με το πληκτρολόγιο (επικόλληση, αντιγραφή, αναίρεση, αποκοπή), επίσης μπορεί να θέσει και διάφορες προτιμήσεις σχετικά με την εργασία και τον Editor της μηχανής.
- **Assets**: Στη καρτέλα αυτή, υπάρχουν επιλογές εισαγωγής εξωτερικών αρχείων μέσα στη μηχανή, όπως έτοιμα πακέτα και γραφικά στοιχεία όπου βοηθούν την εργασία μας να γίνει καλύτερη. Επίσης υπάρχει και η δυνατότητα εξαγωγής τέτοιων πακέτων, από στοιχεία που έχουμε δημιουργήσει εμείς και θέλουμε να τα μοιραστούμε με άλλους.
- **GameObject**: Αποτελεί τη καρτέλα με τη περισσότερη χρήση. Είναι η κεντρική πηγή αντικειμένων που ο δημιουργός του παιχνιδιού μπορεί να χρησιμοποιήσει στη σκηνή. Εδώ μπορούν να παραχθούν 3D και 2D αντικείμενα όπως σφαίρες, κύβοι, κύλινδροι, τετράγωνα, terrain, στοιχεία κειμένου, εφέ, φωτισμός, βίντεο και ήχο ακόμη και η κάμερα που τελικά θα προβάλει τη σκηνή μας.
- **Component**: Με τη καρτέλα αυτή, επιλέγοντας ένα αντικείμενο από το Hierarchy της σκηνής μας, μπορούμε να του δώσουμε διάφορες ιδιότητες. Επίσης υπάρχει και η δυνατότητα προσθήκης ενός script στο επιλεγμένο αντικείμενο.
- **Window**: Μας βοηθά στη διαχείριση όλου του περιβάλλοντος της μηχανής. Από αυτό μπορούμε να επεξεργαστούμε όλα τα παράθυρα του Editor, όπως αυτά περιγράφηκαν στις παραπάνω παραγράφους, στην αρχή της τρέχουσας ενότητας.
- **Help**: Περιέχει διάφορα βοηθητικά εργαλεία για το χρήστη της μηχανής. Μέσω της καρτέλας αυτής, μπορούμε να μεταβούμε στο επίσημο forum της Unity, να μάθουμε βασικές πληροφορίες για την έκδοση που έχουμε (με την επιλογή About Unity) και άλλα όπως την αναφορά ενός σφάλματος (Report a Bug) ή τον έλεγχο για το αν υπάρχει κάποια νέα ενημέρωση (Check for updates) για τη μηχανή.

2.1.4 Προγραμματισμός με Unity 3D

Οι βασικές λειτουργίες της μηχανής στηρίζονται στη δημιουργία Scripts σε κώδικα. Εκτός από μερικές έτοιμες λειτουργίες που προσφέρει, οι σημαντικότερες εκτελούνται μέσω κώδικα από τα Scripts που μπορούμε εμείς να δημιουργήσουμε.

Υποστηρίζονται τρεις βασικές γλώσσες προγραμματισμού, οι:

1. C#, αντικειμενοστραφής γλώσσα προγραμματισμού ανεπτυγμένη από τη Microsoft με την οποία γράφεται ο μεγαλύτερος όγκος του κώδικα.
2. JavaScript, διερμηνευμένη γλώσσα επηρεασμένη από τη C και ανεπτυγμένη από τη Netscape.
3. Boo, στατική γλώσσα γενικού σκοπού επηρεασμένη από τη σύνταξη της Python και ανεπτυγμένη από τον Rodrigo B. Oliveira.

Όταν δημιουργούμε ένα Script, αυτόματα μέσα σε αυτό παράγονται δύο συναρτήσεις, η Start και η Update. Οι συναρτήσεις αυτές θα αναλυθούν εκτενέστερα στο Κεφάλαιο 3 (παράγραφος 6). Για τη συγγραφή και επεξεργασία του κώδικα υπάρχουν δύο βασικά προγράμματα. Όταν δημιουργούμε ένα Script στη μηχανή, κάνοντας το διπλό κλικ μας ανοίγει ένα από τα παρακάτω, όποιο εμείς προτιμούμε:

1. Microsoft Visual Studio.
2. MonoDevelop.



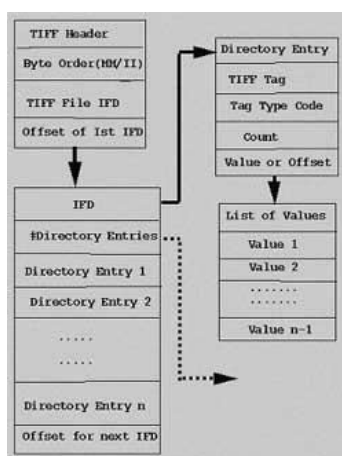
Εικόνα 2.4. Το επίσημα λογότυπα των προγραμμάτων, αριστερά: Visual Studio, δεξιά: MonoDevelop. (Πηγή: Google)

2.2 Αρχεία εικόνων TIFF

2.2.1 Γενικές Πληροφορίες

Οι ψηφιακές εικόνες τύπου TIFF (Tagged Image File Format) είναι κοινά αποδεκτές και στη σημερινή εποχή βρίσκουν χρήση σε πολλά πεδία. Το TIFF είναι ένας ιδανικός τύπος για αποθήκευση, μεταφορά, προβολή και εκτύπωση ψηφιακών εικόνων όπως κλιπ αρτ, λογότυπα και σκαναρισμένα έγγραφα [9]. Ο τύπος αυτός μπορεί επίσης να χρησιμοποιηθεί για αποθήκευση και μεταφορά ψηφιακών εικόνων που προέρχονται από δορυφόρους, αεροφωτογραφίες αλλά και χάρτες. Το TIFF είναι μια μορφή αρχείου ψηφιακής εικόνας που βασίζεται στις ετικέτες (tag-based). Τα κύρια πλεονεκτήματα αυτών των εικόνων είναι η συμβατότητα τους με ένα μεγάλο εύρος εφαρμογών, είναι ανεξάρτητες από την αρχιτεκτονική του υπολογιστή, από το λειτουργικό σύστημα και από τη κάρτα γραφικών. Είναι αρκετά συμπιεσμένα αρχεία και μπορούν να διαχειριστούν ασπρόμαυρες, grayscale αλλά και έγχρωμες εικόνες. Τα TIFF επιτρέπουν ανάλυση χρωμάτων μέχρι και 48 bits (υπάρχει ένα 16-bit πεδίο για κάθε ένα από τα R,G,B) [10]. Οι προδιαγραφές για την έκδοση 6.0 εικόνων TIFF, κυκλοφόρησαν τον Ιούνιο του 1992 και αναφέρονται ως GeoTIFF. Όπως φαίνεται και στην εικόνα 2.5, τα TIFF αρχεία έχουν μια ιεραρχία τριών επιπέδων. Από το υψηλότερο προς το χαμηλότερο, τα επίπεδα είναι τα εξής:

1. Η κεφαλίδα του αρχείου.
2. Ένα ή περισσότερα ευρετήρια τα όποια καλούνται IFDs (Image File Directories), περιέχουν κώδικες και τα δεδομένα τους ή δείκτες προς τα δεδομένα.
3. Πραγματικά δεδομένα.



Εικόνα 2.5. Η δομή ενός αρχείου TIFF. (Πηγή: <https://www.geospatialworld.net/>)

Πιο αναλυτικά, τα αρχεία TIFF ξεκινούν με μια κεφαλίδα των 8 byte, η οποία μας δίνει βασικές πληροφορίες σχετικά με το αρχείο όπως τη διάταξη των byte (Little endian ή Big endian), το κωδικό του αρχείου TIFF ή τον αριθμό έκδοσης (είναι πάντα 42) και έναν δείκτη προς το πρώτο ευρετήριο (IFD). Η επόμενη δομή του αρχείου μετά τη κεφαλίδα είναι το πρώτο και ίσως το μοναδικό ευρετήριο (IFD). Από το σημείο αυτό μπορούμε να εξάγουμε πληροφορίες ακολουθώντας τους δείκτες, για τη μετάβαση στο πρώτο ευρετήριο χρησιμοποιούμε το δείκτη που έχει η κεφαλίδα. Ένα ευρετήριο αποτελείται από εγγραφές των 12 byte, οι οποίες περιέχουν και ετικέτες. Η βασική δομή ενός ευρετηρίου και των εγγραφών του περιγράφεται στο πίνακα 2.1 που ακολουθεί:

Offset	Μέγεθος	Περιγραφή
0	2	Ετικέτα (tag)
2	2	Τύπος δεδομένων (Type of data)
4	4	Μετρητής πεδίου (Count field)
8	4	Δείκτης δεδομένων ή πεδίο δεδομένων (Data pointer or data field)

Πίνακας 2.1. Η δομή του ευρετηρίου ενός TIFF αρχείου. (Πηγή: <https://www.geospatialworld.net/>)

Συγκεκριμένα, για το κάθε πεδίο που απεικονίζει ο παραπάνω πίνακας:

- Ετικέτα (tag): Τα πρώτα δύο bytes είναι η ετικέτα. Οι κώδικες αυτοί εκχωρούνται από το διαχειριστή των εικόνων TIFF σε μπλοκ των πέντε. Ο σχεδιασμός αυτός των ετικετών επεκτάθηκε έτσι ώστε να υποστηρίζει και αρχεία εικόνων GeoTIFF.
- Τύπος δεδομένων: Τα επόμενα δύο bytes αποτελούνται από ένα κώδικα ο οποίος υποδεικνύει τον τύπο δεδομένων στο συγκεκριμένο πεδίο.
- Μετρητής πεδίου: Αυτό το 4-byte πεδίο προσδιορίζει τον αριθμό των τιμών μέσα στο πεδίο δεδομένων και όχι τον αριθμό των bytes.
- Δείκτης δεδομένων: Τα 4 τελευταία bytes συνήθως, είναι ένας δείκτης στην αρχή του πεδίου των δεδομένων. Μερικές φορές, το πεδίο αυτό δεν περιέχει ένα δείκτη αλλά τα πραγματικά δεδομένα.

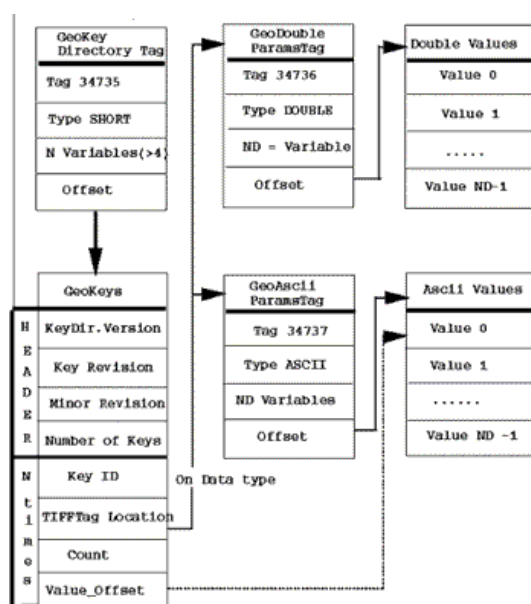
2.2.2 Ιστορικά στοιχεία

Τα αρχεία TIFF δημιουργήθηκαν ως μια προσπάθεια λήψης σκαναρισμένων εικόνων σε ηλεκτρονικό υπολογιστή στα μέσα της δεκαετίας του '80. Στα αρχικά τους στάδια, οι εικόνες TIFF είχαν μόνο δυαδική μορφή (μόνο δύο πιθανές τιμές για κάθε ένα από τα πίξελ), επειδή οι υπολογιστές της εποχής εκείνης μπορούσαν να διαχειριστούν μόνο αυτή τη μορφή. Όσο το σκανάρισμα εικόνων γινόταν όλο και πιο ισχυρό και ο αποθηκευτικός χώρος των σκληρών δίσκων των υπολογιστών όλο και πιο άφθονος, τα TIFF αναπτύχθηκαν έτσι ώστε να υποστηρίζουν εικόνες επιπέδου του γκρι και στη συνέχεια έγχρωμες εικόνες. Στη σημερινή εποχή, οι εικόνες τύπου TIFF μαζί με τους τύπους JPEG και PNG είναι οι πιο δημοφιλείς. Η πρώτη έκδοση των προδιαγραφών μιας εικόνας TIFF εκδόθηκε από την Aldus Corporation το φθινόπωρο του 1986 μετά από δύο σημαντικές προσχεδιασμένες εκδόσεις. Η έκδοση αυτή μπορεί να χαρακτηριστεί ως Revision 3.0, η οποία δημοσιεύθηκε μετά από μια σειρά συναντήσεων με διάφορους κατασκευαστές σκάνερ και υπευθύνων ανάπτυξης λογισμικού. Τον Απρίλιο του 1987 δημοσιεύθηκε η έκδοση Revision 4.0 η οποία περιείχε μικρές βελτιώσεις. Τον Οκτώβριο του 1988 δημοσιεύθηκε η 5η έκδοση η οποία πρόσθεσε επιλογή διαχείρισης έγχρωμων εικόνων. Η τελευταία έκδοση (6.0) των εικόνων TIFF δημοσιεύθηκε τον Ιούνιο του 1992, πλέον η ανάπτυξη τους γίνεται από τον οργανισμό Adobe Systems Incorporated. [10]

2.2.3 Εικόνες GeoTIFF

Οι περισσότεροι χρήστες τέτοιων εικόνων ένιωθαν την ανάγκη να αποθηκεύσουν σε αυτές με κάποιο τρόπο πληροφορία γεωγραφικής φύσεως (γεωγραφικό μήκος, πλάτος αλλά και χάρτες). Η απαίτηση αυτή δημιούργησε την ιδέα ανάπτυξης GeoTIFF αρχείων, είναι εικόνες τύπου TIFF που έχουν ενσωματωμένα γεωγραφικά στοιχεία. Οι προδιαγραφές ενός GeoTIFF προσδιορίζουν ένα σύνολο ετικετών TIFF εικόνων οι οποίες παρέχουν περιγραφή "χαρτογραφημένης" πληροφορίας. Η μορφή των εικόνων αυτών είναι σε πλήρη συμφωνία με τις προδιαγραφές της έκδοσης 6.0 των TIFF, όπως αναφέρθηκε και στη παραπάνω παράγραφο. Χρησιμοποιούν ένα μικρό σύνολο κατοχυρωμένων ετικετών TIFF για να αποθηκεύσουν ένα ευρύ φάσμα γεωγραφικών πληροφοριών. Η αποθήκευση της πληροφορίας προσεγγίζεται με ένα "MetaTag" (GeoKey) το οποίο κωδικοποιεί ένα μεγάλο όγκο στοιχείων πληροφορίας σε 6 ετικέτες. Τα κλειδιά αυτά σχεδιάστηκαν έτσι ώστε να πληρούν τα πρότυπα των TIFF ετικετών. Νέα κλειδιά μπορούν να δημιουργηθούν όπου χρειάζεται χωρίς τη συγκατάθεση των αρχικών κατασκευαστών. Οι εικόνες GeoTIFF αρχικά δημιουργήθηκαν από τον Dr. Niles Ritter καθώς εργαζόταν στα εργαστήρια της NASA. [9]

Η ιεραρχική δομή των κλειδιών και των αρχείων GeoTIFF γενικότερα παρουσιάζεται στην εικόνα 2.6:



Εικόνα 2.6. Η βασική δομή των κλειδιών μιας GeoTIFF εικόνας.

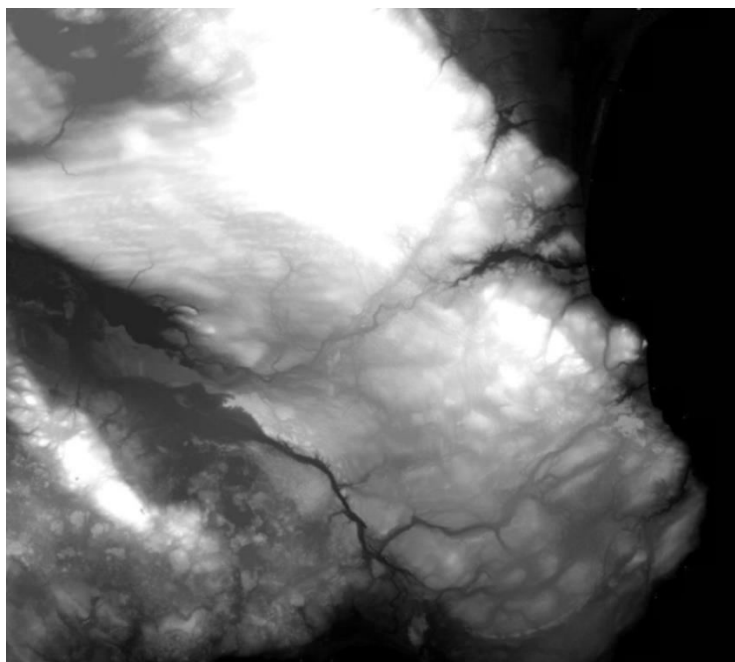
(Πηγή: <https://www.geospatialworld.net/>)

Η εικόνα αυτή μας δίνει περιληπτικά τη μορφή του αρχείου και του μηχανισμού αποθήκευσης που χρησιμοποιείται από τις εικόνες τύπου GeoTIFF. Επίσης, υπάρχουν και άλλες πληροφορίες που μπορούν να εξαχθούν από αυτές τις εικόνες όπως το γεωγραφικό μήκος και πλάτος μιας περιοχής (latitude και longitude) αλλά και ατμοσφαιρικά δεδομένα που αφορούν την εικόνα, ανάλογα με το τύπο της. Στη παρούσα εργασία χρησιμοποιήθηκαν αρκετά αυτές οι πληροφορίες. Ένας μεγάλος όγκος εμπορικών GIS (Geographic Information Systems, Γεωγραφικά Συστήματα Πληροφορίας) και πακέτων διαχείρισης εικόνων πλέον υποστηρίζουν GeoTIFF επειδή ενσωματώνουν γεωγραφική πληροφορία. Μερικά παραδείγματα τέτοιων συστημάτων:

1. ARC/INFO.
2. ERDAS IMAGINE.
3. PCI'S EASI/PACE.
4. MAP INFO and Python Imaging Library.

Καθώς όλο και περισσότερα συστήματα στις μέρες μας υποστηρίζουν TIFF εικόνες, η μορφή αυτή είναι πλέον μια από τις δημοφιλέστερες και έχει υιοθετηθεί από πολλές εφαρμογές διαχείρισης τέτοιων δεδομένων. Η ευελιξία των εικόνων TIFF στη προσθήκη νέων ετικετών και η μεταφερισιμότητα τους, τους έχει δώσει μεγάλο εύρος επέκτασης για το μέλλον.

Παράδειγμα αρχείου GeoTIFF ακολουθεί στην εικόνα 2.7, στο σημείο αυτό επισημαίνεται ότι δεν είναι το αρχείο που χρησιμοποιήθηκε στην εργασία:



Εικόνα 2.7. Οπτική αναπαράσταση ενός GeoTIFF αρχείου.
(Πηγή: Google)

2.2.4 Λήψη GeoTIFF εικόνων

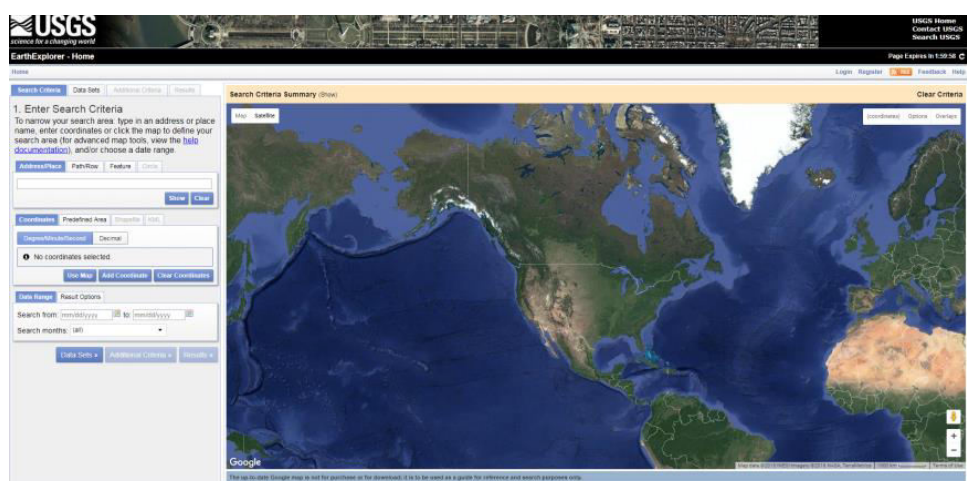
Για να έχουμε τη δυνατότητα διαχείρισης τέτοιων εικόνων, πρέπει πρώτα να της έχουμε κάνει λήψη και να της έχουμε αποθηκεύσει στον υπολογιστή μας. Υπάρχουν πολλές σελίδες στο διαδίκτυο για το σκοπό αυτό, η μέθοδος που ακολουθήσαμε στην εργασία αυτή, περιγράφεται με τα παρακάτω βήματα:

1. Αρχικά, πήγαμε στη σελίδα <https://earthexplorer.usgs.gov/> [13] και δημιουργούμε ένα λογαριασμό (register) γιατί διαφορετικά δεν μας επιτρέπεται η λήψη εικόνων από τη συγκεκριμένη σελίδα. Η διαδικασία είναι χρονοβόρα αλλά είναι δωρεάν.
2. Αφού δημιουργηθεί ο λογαριασμός, η σελίδα μας εμφανίζει το παγκόσμιο χάρτη, μέσω αυτού μπορούμε να αναζητήσουμε όποια περιοχή επιθυμούμε. Επίσης προσφέρεται και η δυνατότητα γρήγορης αναζήτησης μέσω της καρτέλας Search Criteria στο αριστερό μέρος της σελίδας.
3. Αφού επιλέξουμε τη περιοχή, μπορούμε σε αυτή να τοποθετήσουμε pins κάνοντας ένα κλικ πάνω στο χάρτη. Στην εργασία τα pins που εισάγαμε ήταν 4 στον αριθμό, έτσι ώστε να δημιουργηθεί μια τετραγωνική περιοχή

γύρω από τη πόλη των Ιωαννίνων. Η περιοχή που θέλουμε τώρα έχει μαρκαριστεί με ένα κόκκινο πλαίσιο.

4. Στη συνέχεια, μεταβαίνουμε στη καρτέλα Data Sets (δεξιά της καρτέλας Search Criteria), εκεί μας εμφανίζεται μια λίστα με όλους τους τύπους δεδομένων που έχουμε στη διάθεση μας και μπορούμε να επιλέξουμε.
5. Τέλος, κάνουμε κλικ στη τέταρτη κατά σειρά καρτέλα (Results) και μας εμφανίζεται μια λίστα από διαθέσιμες εικόνες για λήψη. Επιλέγουμε την εικόνα που θέλουμε και κάνουμε κλικ στο πέμπτο εικονίδιο (Download Options) κάτω από τις πληροφορίες της εικόνας μας. Αυτό μας εμφανίζει ένα νέο πλαίσιο μέσα στη σελίδα με τις επιλογές λήψης που έχουμε στη διάθεση μας. Επιλέγουμε το αρχείο εικόνας GeoTIFF και κάνουμε κλικ στο κουμπί Download. Βέβαια, υπάρχουν και τύποι εικόνων όπου η λήψη τους γίνεται με διαφορετικό τρόπο, είτε μέσω e-mail αφού πρώτα γίνει order της εικόνας από τη σελίδα είτε επί πληρωμή (υποδεικνύονται με το σύμβολο του δολαρίου). [23]

Επίσης, η σελίδα παρέχει και βοήθεια (Help) στους χρήστες της. Για να μπορέσει κάποιος απλός χρήστης να κατανοήσει τι δεδομένα και πληροφορίες περιέχει το κάθε σύνολο δεδομένων πρέπει να ανατρέξει στον αντίστοιχο πίνακα που προσφέρει η σελίδα. Για το κομμάτι της βοήθειας: Help (πάνω δεξί μέρος) > EarthExplorer Tutorial > Select Data Set > Products, ανοίγοντας κάθε φορά τη καρτέλα που μας ενδιαφέρει, συλλέγουμε πληροφορίες για τα σύνολα δεδομένων. Να σημειωθεί βέβαια ότι βοήθεια παρέχεται και για πολλές άλλες πληροφορίες, αλλά αυτή είναι η σημαντικότερη.



Εικόνα 2.8. Η σελίδα Earth Explorer. (Πηγή: <https://earthexplorer.usgs.gov/>)

2.2.5 Διαχείριση εικόνων GeoTIFF μέσω κώδικα

Υπάρχουν δύο βασικές βιβλιοθήκες (πακέτα/προγράμματα) για το άνοιγμα και τη διαχείριση εικόνων GeoTIFF μέσω κώδικα. Το πρώτο πρόγραμμα ονομάζεται GDAL (Geospatial Data Abstraction Library) [12]. Το GDAL είναι μια βιβλιοθήκη για ανάγνωση και για εγγραφή ψηφιακών δεδομένων που προέρχονται από εικόνες οι οποίες αποθηκεύουν γεωγραφικά δεδομένα όπως μια εικόνα τύπου GeoTIFF στη δική μας περίπτωση. Εγκατάσταση του μπορεί να γίνει σε όλα τα διαθέσιμα λειτουργικά συστήματα (Windows, Linux, Mac) καθώς η εφαρμογή είναι cross-platform. Υπάρχουν δύο διαφορετικοί τρόποι εκτέλεσης της εφαρμογής. Ο πρώτος τρόπος είναι μέσω του τερματικού που προσφέρει το GDAL, δίνοντας τις απαραίτητες εντολές σε αυτό μπορούμε να μάθουμε πολλές πληροφορίες για το αρχείο εικόνας GeoTIFF που έχουμε. Ο δεύτερος τρόπος είναι μέσω κώδικα. Το GDAL υποστηρίζει τις γλώσσες προγραμματισμού C, C++, Python και υπάρχει δυνατότητα επέκτασης και στη γλώσσα C# μέσω εγκατάστασης NuGet στο Visual Studio. Ιστορικά, η εφαρμογή δημιουργήθηκε αρχικά από τον Frank Warmerdam τον Ιούνιο του 2000 μέχρι και την έκδοση 1.3.2, η τωρινή έκδοση της είναι η 2.3.2 η οποία εκδόθηκε το Σεπτέμβριο του 2018, πλέον η εφαρμογή αναπτύσσεται από τον οργανισμό Open Source Geospatial Foundation. Πολλά προγράμματα χρησιμοποιούν τη βιβλιοθήκη GDAL για τη διαχείριση τέτοιων εικόνων. Μερικά από αυτά είναι τα εξής: ArcGIS, Biosphere3D, Biotop Invent, GeoView Pro, Google Earth, JMAP και άλλα. Τέλος, το πρόγραμμα υποστηρίζει μια μεγάλη πληθώρα από διαφορετικούς τύπους εικόνων και δεδομένων, επίσης υπάρχει και δυνατότητα εύκολης επέκτασης του.

Η δεύτερη βιβλιοθήκη ονομάζεται BitMiracle.LibTiff [11]. Η Unity και το πρόγραμμα Visual Studio προσφέρουν τη βιβλιοθήκη αυτή (NuGet) η οποία μπορεί να ανοίξει και να διαβάσει βασικές πληροφορίες αρχείων τύπου TIFF. Υποστηρίζει μόνο τη γλώσσα προγραμματισμού C#, η βιβλιοθήκη αναπτύχθηκε από τον Sam Leffler μέσω του οργανισμού Silicon Graphics.



Εικόνα 2.9. Το επίσημο λογότυπο της εφαρμογής GDAL. (Πηγή: Wikipedia)

2.3 Terrain

2.3.1 Γενικές Πληροφορίες

Ένα Terrain (έδαφος) ή ανάγλυφο (κάποιες φορές αναφέρεται και ως τοπογραφικό ανάγλυφο) εμπεριέχει τις κάθετες και τις οριζόντιες διαστάσεις μιας περιοχής [14]. Η λατινική λέξη "terra" (η ρίζα της λέξης terrain) σημαίνει "Γη". Στην επιστήμη της γεωγραφίας, το Terrain είναι κομμάτι μιας περιοχής. Συνήθως αναφέρεται στους όρους υψόμετρο, πλαγιά ή προσανατολισμός των χαρακτηριστικών ενός εδάφους. Το Terrain επηρεάζει σημαντικά τη ροή και τη κατανομή της επιφάνειας του νερού. Σε μεγάλες περιοχές, μπορεί να επιδράσει στο καιρό αλλά και σε ατμοσφαιρικά δείγματα. Τέλος, ένα μεγάλο κομμάτι της γεωμορφολογίας ασχολείται με το σχηματισμό Terrain. Η κατανόηση του όρου είναι αρκετά σημαντική για πολλούς λόγους, μερικοί από αυτούς είναι οι εξής:

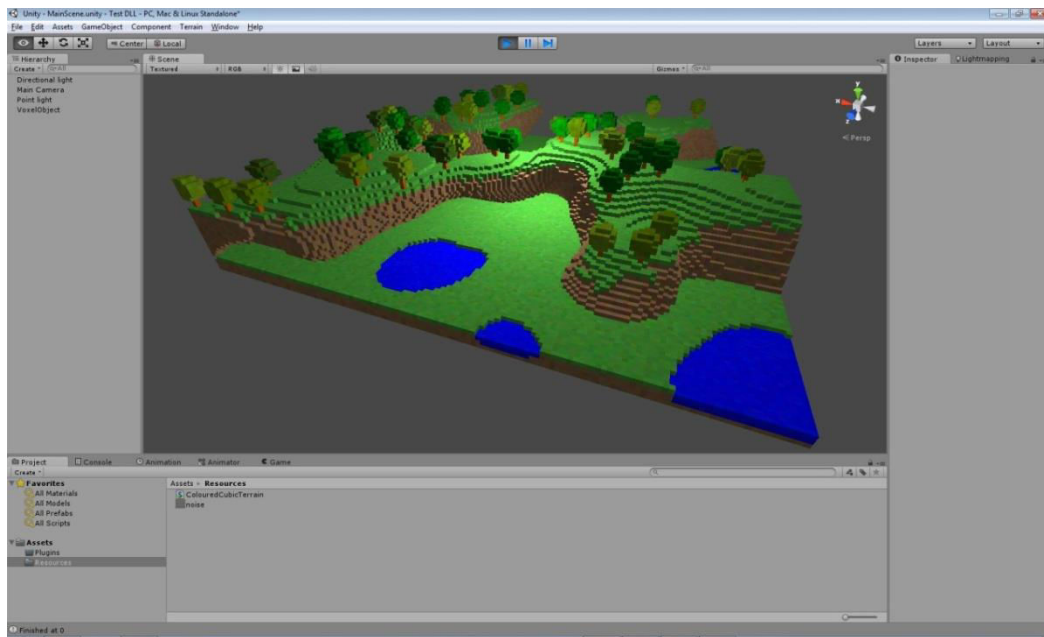
1. Το Terrain μιας ευρείας περιοχής καθορίζει τη καταλληλότητα της για τη δημιουργία ανθρώπινων οικισμών.
2. Επίσης, η αντίληψη του υποστηρίζει την εδαφική συντήρηση μιας περιοχής, ειδικότερα στο τομέα της γεωργίας.
3. Το Terrain είναι ύψιστης σημασίας και για στρατιωτικούς λόγους, γιατί προσδιορίζει την ικανότητα των ενόπλων δυνάμεων να καταλάβουν και να κρατήσουν περιοχές, στη μετακίνηση των στρατευμάτων τους και του εξοπλισμού μέσα στη περιοχή.
4. Τέλος, είναι αρκετά σημαντικό στο προσδιορισμό καιρικών μοτίβων. Δύο γεωγραφικά κοντινές περιοχές μπορεί να διαφέρουν ριζικά ως προς θέμα αυτό.



Εικόνα 2.10. Παράδειγμα ενός Terrain σε μορφή εικόνας.
(Πηγή: Google)

2.3.2 Ο ρόλος του Terrain στη Unity 3D

Το Terrain στη Unity έχει σημαντικό ρόλο και επιδρά καταλυτικά στη δημιουργία μιας σκηνής ή ενός ολοκληρωμένου παιχνιδιού. Όπως αναφέραμε και στη παραπάνω παράγραφο έτσι και στη μηχανή το Terrain παίζει το ρόλο του εδάφους. Αποτελεί ένα αντικείμενο (game object) και μπορεί εύκολα και γρήγορα να δημιουργηθεί είτε χειροκίνητα (μέσω των επιλογών στο μενού) ή μέσω κώδικα, γράφοντας script σε C#. Πάνω του μπορούμε να εισάγουμε έτοιμα αντικείμενα που προσφέρει η μηχανή όπως κύβους, κυλίνδρους, σφαίρες ή ακόμη και πιο σύνθετα game objects (όπως δέντρα) που δημιουργούμε εμείς. Επίσης, μπορούμε να αλλάξουμε το μέγεθος του (μήκος και πλάτος), το ύψος του καθώς και το χρώμα του. Υπάρχει και η δυνατότητα εισαγωγής textures σε αυτό, με το τρόπο αυτό γίνεται πιο αληθοφανής στο μάτι του χρήστη. Κατά την αρχική του δημιουργία το Terrain στη μηχανή είναι μια επίπεδη επιφάνεια. Στην εργασία αυτή επιδιώκουμε να εισάγουμε ένα αρχείο εικόνας πάνω σε ένα κενό Terrain, έτσι ώστε αυτό να πάρει τη μορφή (γεωμορφολογικά στοιχεία) της περιοχής που περιγράφει η εικόνα. Ένα τυχαίο παράδειγμα Terrain μέσω της Unity περιγράφεται στην εικόνα 2.11:



Εικόνα 2.11. Απεικόνιση ενός Terrain στη Unity.
(Πηγή: Google)

2.4 Υψομετρικοί Χάρτες (Height Maps)

2.4.1 Γενικές Πληροφορίες

Στο τομέα των γραφικών σε υπολογιστή, ένας Υψομετρικός Χάρτης [15] (heightmap) ή ένα υψομετρικό πεδίο (heightfield) είναι μια ψηφιακή εικόνα η οποία χρησιμοποιείται για την αποθήκευση τιμών, όπως υψομετρικά δεδομένα για επιφάνειες, έτσι ώστε να προβληθούν μέσω 3D γραφικών. Ένας υψομετρικός χάρτης μπορεί να χρησιμοποιηθεί στη χαρτογράφηση ενός Terrain, αφού πρώτα γίνει μετατροπή του σε 3D πλέγμα (mesh). Περιλαμβάνει ένα κανάλι το οποίο αντιλαμβάνεται το ύψος από το έδαφος για μια επιφάνεια και μερικές φορές το οπτικοποιεί ως μια ασπρόμαυρη εικόνα, όπου με μαύρο χρώμα απεικονίζεται το μικρότερο ύψος και με άσπρο το μεγαλύτερο (εικόνα 2.13). Όταν ο χάρτης δημιουργείται, ο σχεδιαστής του μπορεί να προσδιορίσει το εύρος της μετατόπισης του, για κάθε ενότητα του καναλιού. Οι υψομετρικοί χάρτες μπορούν από μόνοι τους να αποθηκευθούν σε ήδη υπάρχουσες ασπρόμαυρες εικόνες, με ή χωρίς το προσδιορισμό ειδικών δεδομένων. Μια απλή RGB 8-bit εικόνα μπορεί να δείξει μόνο 256 τιμές του γκρι και συνεπώς μόνο 256 τιμές ύψους. Χρησιμοποιώντας χρώματα, ένας μεγαλύτερος αριθμός από τιμές ύψους μπορούν να αποθηκευτούν. Η τεχνική αυτή είναι ιδιαίτερα χρήσιμη στα σημεία όπου το ύψος μεταβάλλεται ελάχιστα. Από την άλλη πλευρά χρησιμοποιώντας μόνο τιμές του γκρι, επειδή το ύψος πρέπει να αποτυπωθεί μόνο με 256 τιμές, το παραγόμενο Terrain εμφανίζεται αρκετά επίπεδο. Οι Υψομετρικοί Χάρτες συνήθως χρησιμοποιούνται στα Γεωγραφικά Συστήματα Πληροφορίας (Geographic Information Systems, GIS), όπου αποκαλούνται Ψηφιακά Μοντέλα Ύψους (Digital Elevation Models). Στην εικόνα 2.12 βλέπουμε έναν υψομετρικό χάρτη της επιφάνειας της Γης (περιέχει νερό και πάγο), η οποία έχει κανονικοποιηθεί ως μια 8-bit εικόνα επιπέδων του γκρι (grayscale):



Εικόνα 2.12. Υψομετρικός Χάρτης ολόκληρης της Γης (Πηγή: Wikipedia).

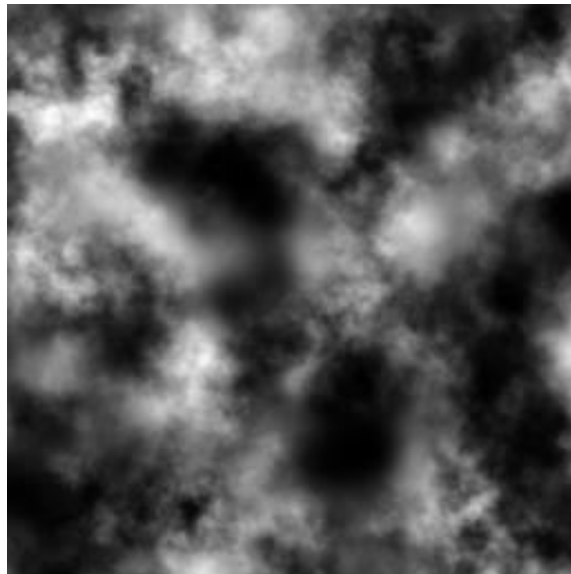
2.4.2 Δημιουργία Υψομετρικών Χαρτών

Υψομετρικοί Χάρτες μπορούν να δημιουργηθούν χειροκίνητα με ένα κλασικό πρόγραμμα ζωγραφικής ή με ένα ειδικό επεξεργαστή (editor) για Terrain. Αυτοί οι editors οπτικοποιούν το Terrain σε 3D μορφή και δίνουν τη δυνατότητα στο χρήστη να επεξεργάζεται την επιφάνεια του. Στη πραγματικότητα, τα εργαλεία αυτά μπορούν να ανυψώσουν, να μειώσουν, να ομαλοποιήσουν ή ακόμη και να διαβρώσουν το Terrain. Άλλος ένας τρόπος δημιουργίας Terrain είναι η χρήση ενός αλγόριθμου παραγωγής του. Τέλος, μια άλλη μέθοδος είναι η χρήση πραγματικών δεδομένων από διάφορες περιοχές του κόσμου όπως για παράδειγμα δεδομένα από δορυφόρους ή αεροπλάνα. Πραγματικά δεδομένα θα χρησιμοποιηθούν και στη παρούσα εργασία, μέσω των εικόνων GeoTIFF που αναλύσαμε σε προηγούμενη ενότητα.

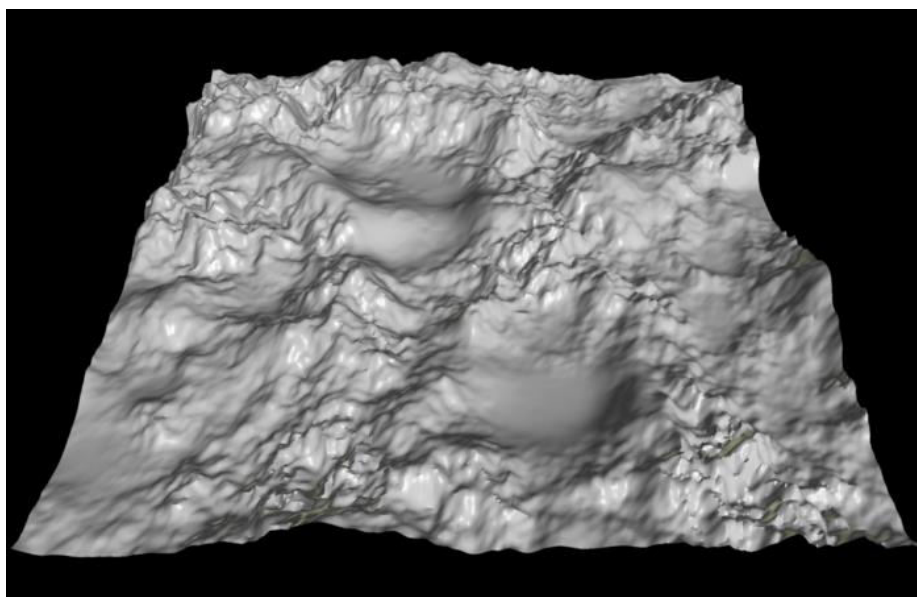
2.4.3 Προγράμματα και Unity 3D

Η παράγραφος αυτή, θα επικεντρωθεί στα προγράμματα τα οποία χρησιμοποιούν τέτοια είδη χαρτών αλλά και το ρόλο που έχει η Unity στο κομμάτι αυτό. Οι Υψομετρικοί Χάρτες χρησιμοποιούνται ευρέως στη δημιουργία ενός Terrain ή ακόμη και στα σύγχρονα βίντεο παιχνίδια. Οι χάρτες αυτοί προσφέρουν έναν ιδανικό τρόπο αποθήκευσης υψομετρικών Terrain σε ψηφιακή μορφή. Σε σύγκριση με ένα πολυγωνικό πλέγμα (mesh), οι χάρτες αυτοί απαιτούν σημαντικά λιγότερη μνήμη. Τα περισσότερα σύγχρονα προγράμματα 3D μοντελισμού για υπολογιστές μπορούν να χρησιμοποιήσουν δεδομένα από Υψομετρικούς Χάρτες για να δημιουργήσουν γρήγορα και με ακρίβεια πολύπλοκα Terrain και άλλες επιφάνειες. Για να γίνει σύνδεση με την εργασία και η Unity μπορεί να χρησιμοποιήσει Υψομετρικούς Χάρτες,

ειδικότερα βασικά στοιχεία σύγχρονων βίντεο παιχνιδιών αναπαριστούν τη συντεταγμένη του ύψους των πολυγώνων ως ένα πλέγμα. Η μηχανή προσφέρει μέσω του μενού της, εύκολους τρόπους εισαγωγής Υψομετρικών Χαρτών, σε αυτούς μπορούμε να εισάγουμε και textures έτσι ώστε η μορφή τους να είσαι όσο πιο κοντά γίνεται στο πραγματικό κόσμο. Υπάρχουν δύο βασικοί τρόποι εισαγωγής Υψομετρικών Χαρτών στη Unity. Ο πρώτος και αυτός που ακολουθήσαμε σε αυτή την εργασία θα περιγραφεί εκτενέστερα στο Κεφάλαιο 3. Ο άλλος τρόπος είναι με τη βοήθεια του προγράμματος Photoshop, το αρχείο εικόνας που απεικονίζει τη γεωγραφική περιοχή που θέλουμε πρέπει αρχικά να μετατραπεί σε μορφή εικόνας τύπου RAW μέσω του προγράμματος. Αφού ολοκληρωθεί με επιτυχία η μετατροπή, επιλέγοντας το Terrain που υπάρχει στη σκηνή μας μπορούμε μέσω των ρυθμίσεων του (Inspector > Terrain Settings) να μεταβούμε στην επιλογή Heightmap και να κάνουμε κλικ στο κουμπί Import Raw, αυτό θα έχει ως αποτέλεσμα την εισαγωγή της γεωγραφικής μορφολογίας του χάρτη πάνω στο επιλεγμένο Terrain. Τέλος, αξίζει να σημειωθεί ότι υπάρχουν δύο βασικά προγράμματα απόδοσης τέτοιων χαρτών, τα Terragen και Picogen. Η εικόνα 2.13 απεικονίζει έναν Υψομετρικό Χάρτη που δημιουργήθηκε με το πρόγραμμα Terragen, στην εικόνα 2.14 απεικονίζεται ο ίδιος χάρτης αφού έχει μετατραπεί σε ένα 3D πλέγμα (mesh) με τη βοήθεια του προγράμματος Anim8or.



Εικόνα 2.13. Υψομετρικός Χάρτης από το Terragen (Πηγή: Wikipedia).



Εικόνα 2.14. Ο παραπάνω χάρτης αφού έχει μετατραπεί σε 3D πλέγμα μέσω του προγράμματος Anim8or (Πηγή: Wikipedia).

Κεφάλαιο 3. Οπτικοποίηση δεδομένων σε περιβάλλον εικονικής πραγματικότητας

3.1 Εισαγωγή - Περίληψη του κεφαλαίου

Το κεφάλαιο αυτό θα επικεντρωθεί στο πρακτικό κομμάτι της διπλωματικής εργασίας. Θα αναλυθεί και θα περιγραφεί βήμα προς βήμα η υλοποίηση της και όλα τα επιμέρους υποσυστήματα που την απαρτίζουν, καθώς και ο σχεδιασμός του Terrain μέσω της Unity. Επίσης, θα αναλυθούν όλα τα αρχεία scripts που δημιουργήθηκαν και χρησιμοποιήθηκαν για τη κατασκευή της εργασίας. Τέλος, θα υπάρχουν και εικόνες (screenshots) από τη λειτουργία του προγράμματος. Πιο αναλυτικά οι ενότητες που θα ακολουθήσουν, θα είναι οι:

1. Προγράμματα και βιβλιοθήκες που χρησιμοποιήθηκαν.
2. Σκηνές του προγράμματος.
3. Τα αντικείμενα της κάθε σκηνής.
4. Σχεδιασμός του Terrain από Height Map.
5. Λειτουργίες του προγράμματος.
6. Scripts.
7. Δημιουργία του τελικού εκτελέσιμου αρχείου.

3.2 Προγράμματα και βιβλιοθήκες

Τα προγράμματα και οι βιβλιοθήκες (πακέτα) που χρησιμοποιήθηκαν για την ανάπτυξη και υλοποίηση της εργασίας είναι τα εξής:

1. Μηχανή γραφικών Unity 3D (έκδοση 2017.4.3-64 bit): το βασικό πρόγραμμα που βοήθησε στη δημιουργία της εργασίας, μέσω της μηχανής σχεδιάστηκαν όλες οι σκηνές και το περιβάλλον του προγράμματος.
2. Microsoft Visual Studio 2017 (έκδοση 15.3.4): το πρόγραμμα αυτό βοήθησε στη δημιουργία των αρχείων κώδικα (scripts).
3. Βιβλιοθήκη (NuGet) BitMiracle.LibTiff: το πακέτο αυτό το οποίο έγινε εγκατάσταση του στο Visual Studio, μας προσφέρει τη δυνατότητα ανοίγματος και διαβάσματος εικόνων τύπου TIFF.
4. Βιβλιοθήκη GDAL (έκδοση 203-MSVC 2017 Win64): βοήθησε στην εύρεση των ακραίων σημείων της εικόνας GeoTIFF. Μέσω του τερματικού που μας προσφέρει η βιβλιοθήκη και εκτελώντας την εντολή gdalinfo.exe

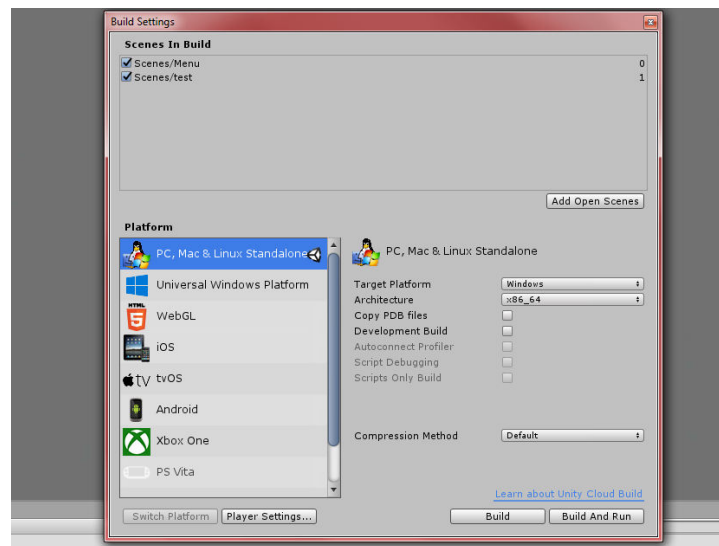
<όνομα εικόνας TIFF> υπολογίζουμε τα ακραία σημεία της εικόνας σε μορφή (γεωγραφικό μήκος, πλάτος). Έπειτα γίνεται αντιστοίχιση μεταξύ των σημείων αυτών και των σημείων της Unity για να τοποθετηθούν τα γραφήματα στο σωστό κομμάτι του Terrain.

Τέλος, έγινε χρήση των γλωσσών προγραμματισμού C# και JavaScript και η εργασία υλοποιήθηκε σε λειτουργικό σύστημα Windows 7 (Professional, Service Pack 1).

3.3 Σκηνές του προγράμματος

Τα παιχνίδια της Unity βασίζονται στη δημιουργία σκηνών. Σε μεγάλες και πολύπλοκες εργασίες οι σκηνές μπορεί να είναι πολλές στον αριθμό. Βέβαια σε μικρές εργασίες μπορεί να υπάρχει και μια σκηνή. Για να γίνει πλήρως κατανοητός ο όρος "σκηνή" ένα απλό παράδειγμα είναι τα επίπεδα ενός βίντεο παιχνιδιού. Ξεκινά από το κεντρικού μενού (π.χ. σκηνή 1) και όσο προχωράμε μέσα στο παιχνίδι και αλλάζουμε επίπεδα, οι σκηνές εναλλάσσονται μεταξύ τους, εδώ τα επίπεδα αντιπροσωπεύουν τις σκηνές. Για να δημιουργήσουμε μια νέα σκηνή στη Unity, από το κεντρικό μενού της επιλέγουμε File > New Scene, αφού την επεξεργαστούμε όπως θέλουμε μπορούμε να την αποθηκεύσουμε με CTRL+S. Έπειτα μπορούμε εύκολα με διπλό κλικ να γυρίσουμε ξανά στη σκηνή που θέλουμε αν έχουμε πολλές, αφού αποθηκεύονται στο μέρος Project του Editor της Unity. Κάθε σκηνή έχει το δικό της Hierarchy και σε αυτό υπάρχουν μόνο τα δικά της αντικείμενα. Για να πραγματοποιηθεί επιτυχής εναλλαγή μεταξύ δύο σκηνών πρέπει οι σκηνές να έχουν προστεθεί στη λίστα Scenes In Build (File > Build Settings), κάθε σκηνή παίρνει ένα μοναδικό αριθμό, ξεκινώντας από το μηδέν. Η τελική αλλαγή των σκηνών γίνεται μέσω κώδικα, όταν κριθεί απαραίτητο από το δημιουργό του προγράμματος, η Unity προσφέρει έτοιμες εντολές για το σκοπό αυτό, αρκεί βέβαια στην εντολή να δοθεί το όνομα ή ο αριθμός που αντιπροσωπεύει τη κάθε σκηνή όπως αναφέρθηκε. Απαραίτητη προϋπόθεση είναι σε κάθε σκηνή να υπάρχει και μια κεντρική κάμερα, η οποία προβάλλει όλα τα αντικείμενα που αυτή περιέχει. Για τη δημιουργία της, από το μενού > GameObject > Camera. Τη κάμερα μπορούμε να την επεξεργαστούμε, αλλάζοντας της θέση και κλίση έτσι ώστε να δείχνει εκείνο το σημείο της σκηνής που θέλουμε, επίσης χρησιμοποιώντας scripts μπορούμε να κινούμε τη κάμερα με τη βοήθεια του πληκτρολογίου ή του ποντικιού. Αν δεν υπάρχει κάποια κάμερα στη σκηνή μας, τότε στη καρτέλα Game του Editor της μηχανής εμφανίζεται το αντίστοιχο μήνυμα. [17]

Το παράθυρο Build Settings και όλες οι επιλογές που αυτό προσφέρει απεικονίζεται στην εικόνα 3.1:



Εικόνα 3.1. Το παράθυρο Build Settings της Unity 3D.

Η παρούσα διπλωματική εργασία αποτελείται από δύο βασικές σκηνές, όπως μας δείχνει και η εικόνα 3.1. Η πρώτη αποτελεί το βασικό μενού του προγράμματος και η δεύτερη είναι η σκηνή στην οποία έχει δημιουργηθεί το Terrain και πραγματοποιείται η οπτικοποίηση των δεδομένων πάνω σε αυτό. Πιο αναλυτικά:

Σκηνή 1: όπως σε κάθε παιχνίδι απλό ή σύνθετο υπάρχει το βασικό μενού, μια εισαγωγική σκηνή στη περίπτωση μας όπως φαίνεται και στην εικόνα 3.2:



Εικόνα 3.2. Σκηνή 1 - Το βασικό μενού του προγράμματος.

Η πρώτη σκηνή αποτελείται από τρία κουμπιά τα οποία επιτελούν τις παρακάτω λειτουργίες:

- Κουμπί Start: όταν πατηθεί ξεκινά το πρόγραμμα και ο χρήστης μεταβαίνει στη δεύτερη σκηνή, όπου πραγματοποιείται η οπτικοποίηση.
- Κουμπί About: αναφέρει βασικές πληροφορίες για το πρόγραμμα, όλες οι εφαρμογές και οι σύγχρονες σελίδες έχουν μια τέτοια επιλογή για πληροφορίες προς το χρήστη που τις χρησιμοποιεί.
- Κουμπί Quit: όπως σε όλες τις εφαρμογές, υπάρχει η δυνατότητα εξόδου από το πρόγραμμα, όταν επιλεγεί από το χρήστη, η εφαρμογή τερματίζει και το παράθυρο που τη τρέχει κλείνει.

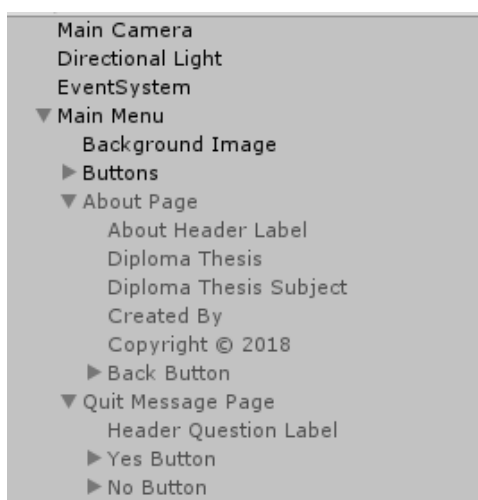
Η Unity μέσω του μενού της προσφέρει έναν εύκολο τρόπο δημιουργίας κουμπιών στη σκηνή του παιχνιδιού μας. Ο τρόπος εμφάνισης τους εξαρτάται από το δημιουργό της εφαρμογής και από τη λειτουργία που θέλουμε αυτά να επιτελούν. Για τη δημιουργία ενός κουμπιού πάμε GameObject > UI > Button. Αφού το τοποθετήσουμε όπου θέλουμε μέσα στη σκηνή μας, μπορούμε να το χρωματίσουμε, να του αλλάξουμε το κείμενο και τη λειτουργία που θα εκτελέσει όταν αυτό πατηθεί από το χρήστη. Για να θέσουμε στο κουμπί μας μια συγκεκριμένη λειτουργία, το επιλέγουμε και μεταβαίνουμε στον Inspector του, εκεί υπάρχει η επιλογή On Click (), κάνουμε κλικ στο σύμβολο "+" και επιλέγουμε εκείνο το αντικείμενο από την Hierarchy της σκηνής μας το οποίο περιέχει το script που προσδίδει μια λειτουργία στο κουμπί μας, τέλος επιλέγουμε τη μέθοδο μέσα από το script που θέλουμε αυτό να εκτελεστεί. [18]

Σκηνή 2: Η σκηνή αυτή περιέχει το Terrain της εργασίας που πάνω σε αυτό γίνεται η οπτικοποίηση και η προβολή των γραφημάτων. Τα δεδομένα έχουν εξαχθεί από το αρχείο εικόνας GeoTIFF που περιγράφει τη συγκεκριμένη περιοχή η οποία έχει αποτυπωθεί με τη μορφή γραφικών που προσφέρει η Unity πάνω στο Terrain (αναλυτικά ο σχεδιασμός του περιγράφεται στην ενότητα 3.5). Στη σκηνή, ο χρήστης έχει τρεις δυνατές επιλογές/κουμπιά. Υπάρχουν τα κουμπιά Grid Options, Clear και Hide Buttons, επίσης αν πατηθεί το πλήκτρο Escape από το πληκτρολόγιο, ο χρήστης μεταβαίνει στο μενού παύσης του προγράμματος. Η σκηνή της οπτικοποίησης θα αναλυθεί πλήρως στην ενότητα 3.6 (Λειτουργίες του προγράμματος), εκεί θα υπάρχουν επεξηγήσεις για το τι ενέργεια επιτελεί το κάθε κουμπί της σκηνής, εικόνες και διάφορα παραδείγματα εκτέλεσης του προγράμματος.

3.4 Τα αντικείμενα της κάθε σκηνής

Όπως αναφέραμε και στη προηγούμενη ενότητα, κάθε σκηνή περιέχει τα δικά της αντικείμενα (game objects) και τα κρατά σε μια λίστα κάτω από το κομμάτι Hierarchy του Editor της Unity. Υπάρχουν πολλοί και διαφορετικοί τύποι αντικειμένων που μπορεί να δημιουργήσει ο χρήστης της μηχανής, ανάλογα με τη φύση και τη χρήση που θέλει να κάνει πάνω σε αυτό. Στην ενότητα αυτή θα περιγραφούν συνοπτικά όλα τα αντικείμενα που έχουν δημιουργηθεί και στις δύο σκηνές που απαρτίζουν το πρόγραμμα αυτό. Αναλυτικές πληροφορίες για τις λειτουργίες και το ρόλο που επιτελεί το κάθε αντικείμενο από αυτά θα δοθούν στην ενότητα 3.6 (Λειτουργίες του προγράμματος).

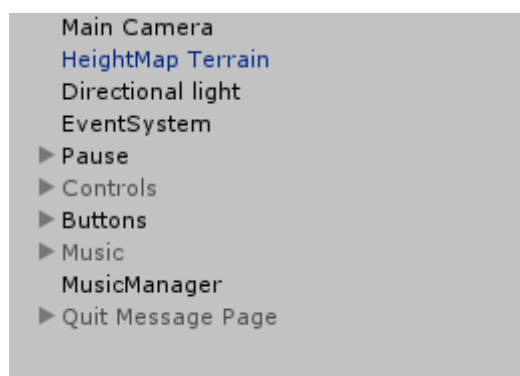
Σκηνή 1: Τα αντικείμενα της πρώτης σκηνής απεικονίζονται στην εικόνα 3.3:



Εικόνα 3.3. Η λίστα αντικειμένων της πρώτης σκηνής.

- **Main Camera:** η κεντρική κάμερα της σκηνής, βοηθά στη προβολή των αντικειμένων που βρίσκονται σε αυτή.
- **Directional Light:** το αντικείμενο αυτό δημιουργείται κάθε φορά που υπάρχει νέα σκηνή στην εργασία μας, βοηθά στο φωτισμό της σκηνής.
- **Event System:** και αυτό το αντικείμενο δημιουργείται κάθε φορά που υπάρχει μια νέα σκηνή στο πρόγραμμά μας, βοηθά στην αποστολή γεγονότων (events) σε όλα τα αντικείμενα της σκηνής έτσι ώστε αυτά να αλληλεπιδρούν με τις ενέργειες του χρήστη (π.χ. πληκτρολόγιο, ποντίκι).
- **Main Menu:** το αντικείμενο αυτό περιέχει όλο τα κουμπιά του κεντρικού μενού της πρώτης σκηνής. Υπάρχει η εικόνα του φόντου (background image), τα κουμπιά Start, About και Quit όπως και οι σελίδες τους που όταν πατηθούν ενεργοποιούνται.

Σκηνή 2: Τα αντικείμενα της δεύτερης σκηνής απεικονίζονται στην εικόνα 3.4:



Εικόνα 3.4. Η λίστα αντικειμένων της δεύτερης σκηνής.

- Pause: το μενού παύσης του προγράμματος, εμφανίζεται όταν πατηθεί το πλήκτρο Escape από το χρήστη.
- Controls: στο μενού παύσης υπάρχει η επιλογή Controls, το παράθυρο που εμφανίζεται πληροφορεί το χρήστη για τα κουμπιά που έχει στη διάθεση του και μπορεί να χρησιμοποιήσει.
- Buttons: το αντικείμενο αυτό περιέχει όλα τα βασικά κουμπιά της δεύτερης σκηνής (Grid Options, Clear & Hide Buttons).
- Music & Music Manager: το αντικείμενο αυτό διαχειρίζεται τη μουσική που υπάρχει στο πρόγραμμα.
- Quit Message Page: είναι το πλαίσιο με το μήνυμα που εμφανίζεται στο χρήστη όταν αυτός πατήσει το κουμπί Quit από το μενού παύσης.
- Height Map Terrain: το Terrain που πάνω σε αυτό πραγματοποιείται η οπτικοποίηση των δεδομένων. Αναλυτικές πληροφορίες για το σχεδιασμό του υπάρχουν στην επόμενη ενότητα 3.5 (Σχεδιασμός Terrain από Height Map).
- Τα υπόλοιπα αντικείμενα (Main Camera, Directional Light και Event System) είναι ακριβώς ίδια όπως αυτά της πρώτης σκηνής, τα οποία περιγράφηκαν παραπάνω.

3.5 Σχεδιασμός του Terrain από Height Map

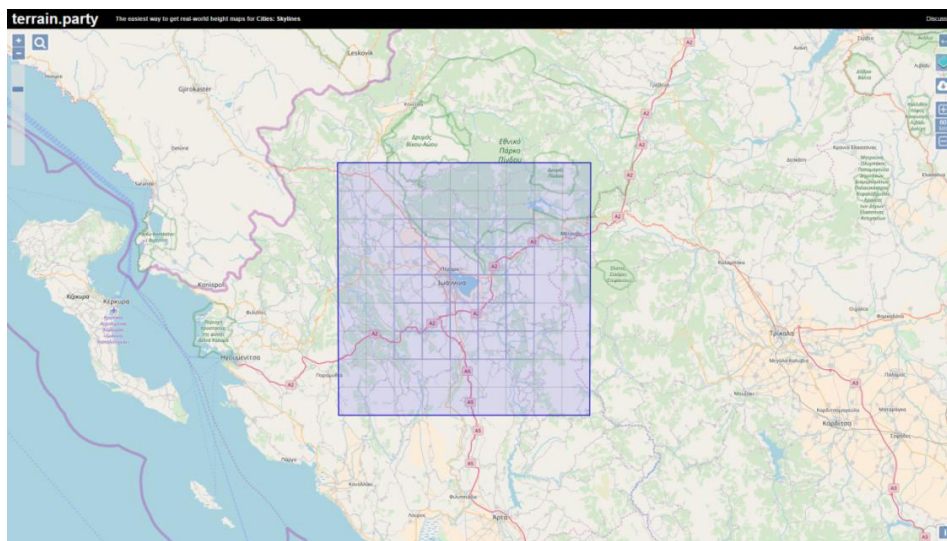
Η δημιουργία του Terrain από έναν Υψομετρικό Χάρτη (Height Map) αποτελεί ένα πολύ σημαντικό κομμάτι στη κατασκευή της εργασίας αλλά και στο οπτικό κομμάτι της. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο υπάρχουν δύο βασικοί τρόποι εισαγωγής Υψομετρικών Χαρτών στο περιβάλλον της Unity. Επιγραμματικά τα κύρια βήματα εισαγωγής του Υψομετρικού Χάρτη στη μηχανή και δημιουργίας του Terrain:

1. Δημιουργία ενός αρχείου JavaScript που εκτελεί τη μετατροπή από την εικόνα του χάρτη σε Terrain στη μηχανή.
2. Λήψη της εικόνας του χάρτη. Την εικόνα τη δίνουμε ως είσοδο (input) στο αρχείο JavaScript.
3. Το JavaScript ανοίγει και διαβάζει τα υψομετρικά δεδομένα της εικόνας και την "κολλάει" πάνω στο Terrain ως texture.
4. Ως έξοδο (output) λαμβάνουμε το Terrain στη μηχανή, μορφοποιημένο σύμφωνα με τα υψομετρικά δεδομένα του χάρτη.

Η μέθοδος [22] που ακολουθήσαμε στη παρούσα εργασία, αναλύεται εκτενέστερα στα παρακάτω βήματα:

1. Μέσω ενός browser, μεταβαίνουμε στη σελίδα: <http://terrain.party/>. [16]
2. Η σελίδα μας εμφανίζει το παγκόσμιο χάρτη και μας ξεκινά από ένα τυχαίο σημείο αναφοράς το οποίο έχουν ορίσει οι δημιουργοί της. Μπορούμε να μεταβούμε σε όποια περιοχή του κόσμου επιθυμούμε είτε χειροκίνητα είτε μέσω της αναζήτησης (πάνω αριστερά της σελίδας). Στην εικόνα 3.5 έχουμε επιλέξει τη πόλη των Ιωαννίνων ως κέντρο και έχουμε πάρει όλες τις περιοχές οι οποίες βρίσκονται σε απόσταση 60 χιλιομέτρων από αυτό (το μπλε πλαίσιο πάνω στο χάρτη μας δείχνει το τελικό Terrain). Την επιθυμητή απόσταση μπορούμε να τη θέσουμε εμείς, μειώνοντας την ή αυξάνοντας την αφού υπάρχει η αντίστοιχη επιλογή στο δεξί κομμάτι της σελίδας. Όσο μεγαλύτερη είναι η περιοχή τόσο πιο ρεαλιστικό θα είναι το Terrain όταν γίνει η εισαγωγή του στη Unity, για αυτό και εδώ επιλέξαμε τη μέγιστη απόσταση που μας επιτρέπει η σελίδα.
3. Αφού γίνει η τελική επιλογή της περιοχής μας, κάνουμε κλικ στο κουμπί Export που βρίσκεται στο δεξί μέρος της σελίδας (ακριβώς πάνω από εκεί που επιλέξαμε τη χιλιομετρική απόσταση στο βήμα 2). Στη συνέχεια ο browser μας εμφανίζει ένα μήνυμα και μας ζητά όνομα για τη περιοχή που επιλέξαμε. Δίνουμε το όνομα (εδώ ioannina) και κάνουμε κλικ στο OK, θα γίνει λήψη ενός συμπίεσμένου αρχείου (zip) με τελικό όνομα ioannina terrain.
4. Μεταβαίνουμε στο φάκελο αποθήκευσης του zip και το κάνουμε αποσυμπίεση (extract). Θα εμφανιστούν συνολικά 5 αρχεία, 4 εικόνες και ένα αρχείο κειμένου (txt) που μας δίνει κάποιες πληροφορίες για τις εικόνες. Εμείς

κρατάμε μόνο την εικόνα με όνομα *ioannina Height Map (Merged)*, τις υπόλοιπες αν θέλουμε τις διαγράφουμε. Η αρχική εικόνα έχει διαστάσεις 1081 x 1081, τις αλλάζουμε σε 1024 x 1024 μέσω κάποιου προγράμματος επεξεργασίας εικόνων ή πιο εύκολα online. Αποθηκεύουμε και κρατάμε τη νέα εικόνα, αυτή θα είναι και η τελική εικόνα που θα γίνει προβολή της γεωγραφικής μορφολογίας της πάνω στο Terrain της Unity. Να σημειωθεί ότι η εικόνα αυτή αναπαριστά την αντίστοιχη περιοχή όπως το GeoTIFF αρχείο που περιγράψαμε στο προηγούμενο κεφάλαιο, έτσι ώστε να γίνει σωστά η οπτικοποίηση των δεδομένων πάνω στη περιοχή, που είναι και ο βασικός σκοπός της διπλωματικής εργασίας.



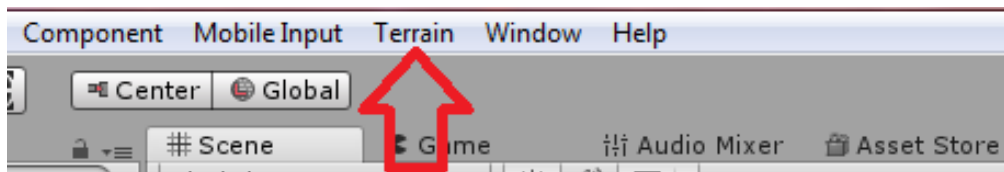
Εικόνα 3.5. Η σελίδα <http://terrain.party/>.

Τα παραπάνω βήματα αποτελούν το πρώτο κομμάτι του σχεδιασμού του Terrain, και αφορούν τη λήψη και την επεξεργασία της εικόνας. Το δεύτερο κομμάτι αφορά την εισαγωγή της εικόνας στη Unity και τη κατασκευή του τελικού Terrain. Τα βήματα που ακολουθήθηκαν είναι τα παρακάτω:

1. Στη Unity, σε νέο ή ήδη υπάρχων project δημιουργούμε ένα κενό Terrain (στο κεντρικό μενού > GameObject > 3D Object > Terrain). Σημαντικό είναι να αλλάξουμε τις διαστάσεις του, επιλέγοντας το αντικείμενο Terrain πάμε δεξιά στο παράθυρο Inspector > Terrain Settings > Resolution και επεξεργαζόμαστε μόνο τις τιμές Terrain Width, Terrain Length και Terrain Height και τις θέτουμε 600, 600 και 60 αντίστοιχα. Οι τιμές αυτές δεν είναι τυχαίες, τις θέτουμε με αυτό το τρόπο γιατί η απόσταση του Terrain από το κέντρο του είναι 60 χιλιόμετρα όπως αναφέραμε στη προηγούμενη σελίδα και στο βήμα 2.
2. Στο κομμάτι Project της Unity δημιουργούμε ένα JavaScript αρχείο. Αρχικά το ανοίγουμε με το Visual Studio 2017 και διαγράφουμε ότι υπάρχει μέσα στο

κώδικα και κάνουμε αντιγραφή και επικόλληση ένα ειδικό κομμάτι κώδικα που θα μας βοηθήσει στη δημιουργία του Υψομετρικού Χάρτη πάνω στο Terrain.

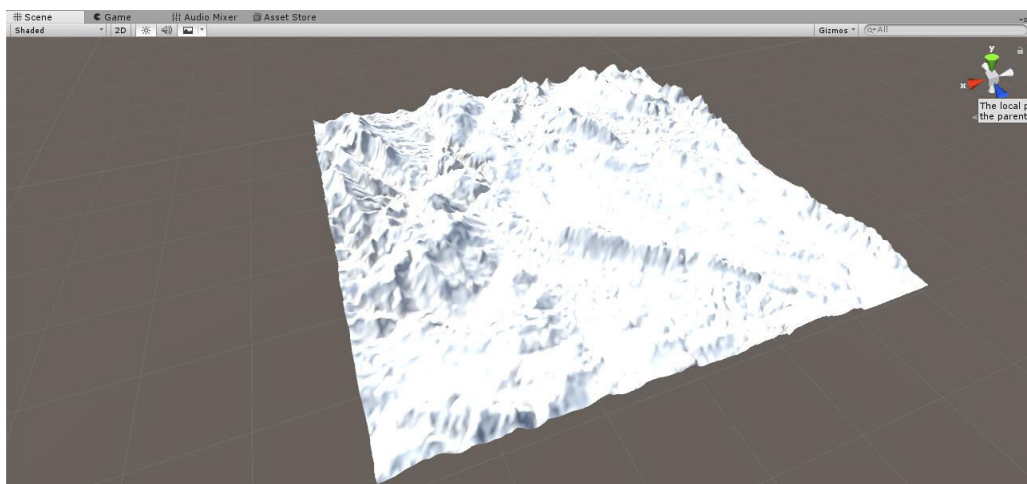
Ο κώδικας JavaScript υπάρχει σε αυτό το σύνδεσμο: [HeightmapFromTexture.js](#). Του δίνουμε το όνομα αυτό και τον αποθηκεύουμε μέσα στο περιβάλλον της μηχανής. Αφού αυτό το βήμα πραγματοποιηθεί επιτυχώς στο κεντρικό μενού της Unity θα εμφανιστεί η επιλογή που υποδεικνύει η εικόνα 3.6:



Εικόνα 3.6. Η επιλογή Terrain που εμφανίζεται στο κεντρικό μενού της Unity.

3. Στη συνέχεια, κάνουμε εισαγωγή της εικόνας που αποθηκεύσαμε και κλιμακώσαμε στα αρχικά βήματα μέσα στη Unity. Η εισαγωγή της μπορεί να γίνει είτε με τη τεχνική drag & drop είτε χειροκίνητα περνώντας την εικόνα στο φάκελο αποθήκευσης της εργασίας. Επιλέγουμε την εικόνα μέσα στη Unity και πάμε δεξιά του παραθύρου στον Inspector της, ανοίγουμε τη καρτέλα Advanced και τσεκάρουμε την ιδιότητα Read/Write Enabled.

4. Τέλος, επιλέγουμε την εικόνα μέσα στη Unity με ένα απλό κλικ, έπειτα μεταβαίνουμε στο κεντρικό μενού της, κάνοντας κλικ την επιλογή Terrain > Heightmap From Texture, η γεωγραφική μορφολογία που περιγράφει η εικόνα εμφανίζεται πάνω στο Terrain, και το κάνει να πάρει τη παρακάτω μορφή:



Εικόνα 3.7. Η αρχική μορφή του Terrain.

Για να φαίνεται πιο αληθινό και όμορφο στο μάτι του χρήστη, το Terrain πρέπει να χρωματιστεί με textures τα οποία αντιπροσωπεύουν μια περιοχή της γης όπως γρασίδι, άμμος, δέντρα, βράχοι και άλλα παρόμοια στοιχεία. Για το χρωματισμό του Terrain ακολουθήθηκε η παρακάτω τεχνική:

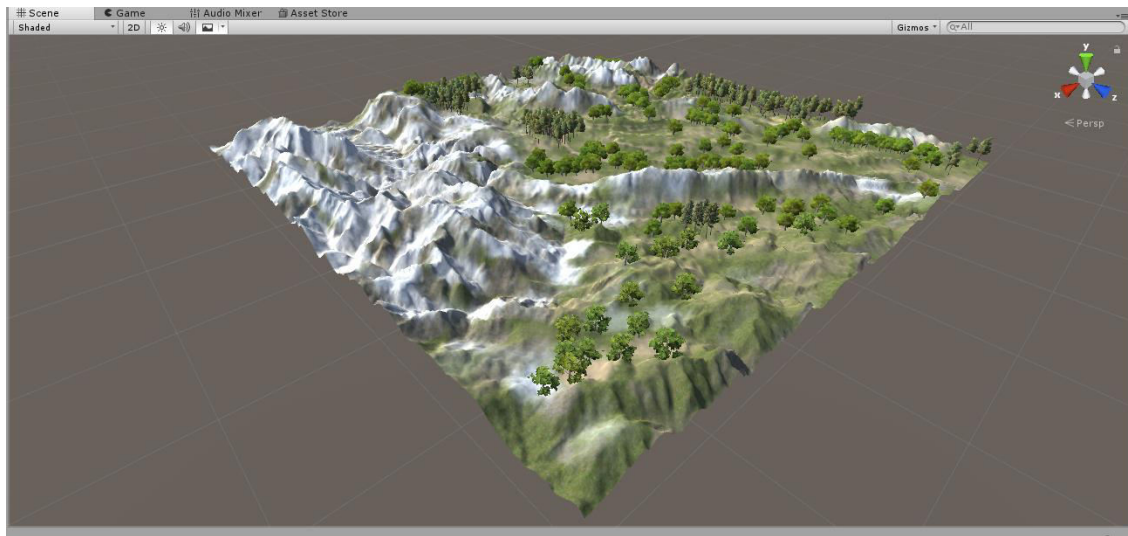
1. Η Unity για το σκοπό αυτό προσφέρει ένα έτοιμο πακέτο που μπορούμε να το εγκαταστήσουμε σε αυτή εύκολα και γρήγορα. Για την εγκατάσταση του πακέτου: από το κεντρικού μενού > Assets > Import Package > Environment, μας ανοίγει ένα νέο μικρό παράθυρο και κάνουμε κλικ στο κουμπί Import. Αφού η εγκατάσταση είναι επιτυχής, στο κομμάτι Project του Editor μας εμφανίζονται πολλοί φάκελοι, scripts και textures όπου βοηθούν την ορθή χρήση του πακέτου. Ο δημιουργός του κάθε Terrain μπορεί να ακολουθήσει πολλούς διαφορετικούς τρόπους χρωματισμού του, ανάλογα με τις προτιμήσεις του.

2. Επιλέγουμε το αντικείμενο Terrain από το Hierarchy και μεταβαίνουμε στον Inspector του. Εκεί πάμε στο Component Terrain που έχει και κάνουμε κλικ στο τέταρτο κατά σειρά εικονίδιο (Paint Texture), το πεδίο Textures είναι αρχικά κενό, με το κουμπί Edit Textures μπορούμε να επιλέξουμε και να εισάγουμε Textures που υπάρχουν ήδη στο φάκελο της εργασίας μας. Το πρώτο texture που θα κάνουμε εισαγωγή θα χρωματίσει όλη την επιφάνεια του Terrain. Με τον ίδιο τρόπο μπορούμε να εισάγουμε και άλλα, επιλέγοντας μια από τις Brushes και το texture που θέλουμε, μπορούμε κρατώντας πατημένο το αριστερό κλικ να χρωματίσουμε εύκολα όποιο κομμάτι της επιφάνειας του Terrain θέλουμε. Επίσης μπορούμε να ρυθμίσουμε και το μέγεθος της Brush (Settings > Brush Size) όπως και την αδιαφάνεια του texture (Settings > Opacity). Στο σημείο αυτό να σημειωθεί, ότι το texture που αφορά το χιόνι πάνω στο Terrain δεν υπάρχει μέσα στο πακέτο Environment, εύκολα όμως μπορούμε με μια αναζήτηση στο Google να βρούμε texture σε μορφή εικόνων PNG (ή JPG) συνήθως, να τα κάνουμε εισαγωγή και να τα χρησιμοποιήσουμε μέσα στη Unity.

3. Για την εισαγωγή άλλων στοιχείων όπως δέντρων, ακολουθούμε ένα παρόμοιο τρόπο. Όπως και πριν, στο Component Terrain κάνουμε κλικ στο πέμπτο κατά σειρά εικονίδιο (Paint Trees), και στη συνέχεια στο πεδίο Trees > Edit Trees > Add Tree, επιλέγουμε ένα από τα μοντέλα δέντρων που μας προσφέρει το πακέτο. Και σε αυτή τη περίπτωση επιλέγουμε το δέντρο και κάνοντας ένα απλό κλικ πάνω στο Terrain, εμφανίζεται. Η Unity μας δίνει τη δυνατότητα, να αλλάξουμε το μέγεθος της Brush (Settings > Brush Size), τη πυκνότητα των δέντρων (Settings > Tree Density) αλλά και το ύψος τους (Settings > Tree Height) όπως και άλλα στοιχεία.

Αυτή είναι η τεχνική που υλοποιήθηκε σε αυτή την εργασία. Όπως αναφέρθηκε αρχικά ο κάθε δημιουργός μιας εργασίας μπορεί να ακολουθήσει εντελώς διαφορετικό τρόπο χρωματισμού του Terrain του, σημαντικός παράγοντας είναι και ο σκοπός που θέλει να επιτελεί το Terrain του στο περιβάλλον της εργασίας του.

Η τελική μορφή που πήρε το Terrain, έπειτα από όλα αυτά τα βήματα απεικονίζεται στην εικόνα 3.8:



Εικόνα 3.8. Η τελική μορφή του Terrain.

Για τη δημιουργία μεγάλων αντικειμένων όπως ένα Terrain στη περίπτωση μας, η Unity εκτελεί μια διαδικασία που ονομάζεται Baking (στο κάτω δεξί μέρος του παραθύρου, υπάρχει μια μπάρα η οποία γεμίζει με μπλε χρώμα και υποδεικνύει το χρόνο που απομένει μέχρι να ολοκληρωθεί η διαδικασία), η διάρκεια της εξαρτάται σημαντικά από την ανάλυση και το μέγεθος του Terrain, τα αντικείμενα που υπάρχουν πάνω σε αυτό αλλά και από την υπολογιστική ισχύ του συστήματος που χρησιμοποιούμε. Σε γενικά πλαίσια η διαδικασία αυτή είναι αρκετά χρονοβόρα και σπαταλά αρκετούς πόρους (μνήμη RAM, επεξεργαστική ισχύ).

3.6 Λειτουργίες του προγράμματος

Στην ενότητα αυτή θα περιγραφούν αναλυτικά όλες οι λειτουργίες του προγράμματος. Τα κουμπιά και οι ενέργειες της 1ης σκηνής περιγράφηκαν στην ενότητα 3.3, εδώ θα περιγραφούν όλες οι ενέργειες των κουμπιών της 2ης σκηνής στην οποία πραγματοποιείται η οπτικοποίηση των δεδομένων. Επίσης για κάθε ενέργεια θα υπάρχουν και εικόνες μέσα από το πρόγραμμα (screenshots) οι οποίες θα επιβεβαιώνουν τη λειτουργία τους.

Η αρχική μορφή της δεύτερης σκηνής απεικονίζεται στην εικόνα 3.9, στην εικόνα αυτή βλέπουμε ότι το Terrain είναι κενό και επάνω του δεν υπάρχει κάποιο γράφημα οπτικοποίησης. Επίσης, παρατηρούμε ότι η σκηνή περιέχει και 4 κουμπιά, τα οποία θα αναλυθούν ένα προς ένα. Παρακάτω, θα αναφερθεί η διαδικασία που πρέπει να ακολουθήσει ο χρήστης για να εμφανιστούν τα γραφήματα πάνω στο Terrain.



Εικόνα 3.9. Η αρχική μορφή της δεύτερης σκηνής.

Τα κουμπιά της δεύτερης σκηνής επιτελούν τις παρακάτω λειτουργίες:

1. Κουμπί Grid Options: εμφανίζει στο χρήστη ένα παράθυρο επιλογών για το Grid (πλέγμα), δηλαδή για τα γραφήματα της οπτικοποίησης τα οποία θα προβληθούν πάνω στο Terrain. Εκεί ο χρήστης θα μπορεί να θέσει το μέγεθος του πλέγματος, το χρώμα του αλλά και τι μορφή του γραφήματος (το είδος του σχήματος που θα απεικονιστεί το γράφημα, εικόνα 3.10).

2. Κουμπί Clear: όταν πατηθεί από το χρήστη, διαγράφει το γράφημα που έχει επιλέξει και αφήνει το Terrain στην αρχική του μορφή, δηλαδή κενό. Είναι χρήσιμο μόνο αν υπάρχει γράφημα στο πρόγραμμα, διαφορετικά δεν επιτελεί κάποια λειτουργία.
3. Κουμπί Hide Buttons: για να βλέπει καλύτερα ο χρήστης το Terrain και τα γραφήματα, μπορεί να πατήσει το συγκεκριμένο κουμπί το οποίο κρύβει και τα 4 κουμπιά της σκηνής. Για να έρθουμε ξανά στην αρχική μορφή, το πρόγραμμα εμφανίζει ένα νέο κουμπί Show Buttons το οποίο επαναφέρει όλα τα κουμπιά (εικόνα 3.11).
4. Μενού Παύσης: υπάρχει και η δυνατότητα παύσης του προγράμματος, πατώντας το κουμπί Escape από το πληκτρολόγιο ο χρήστης μεταβαίνει στο μενού αυτό, όπου εκεί έχει διάφορες επιλογές (εικόνα 3.12).



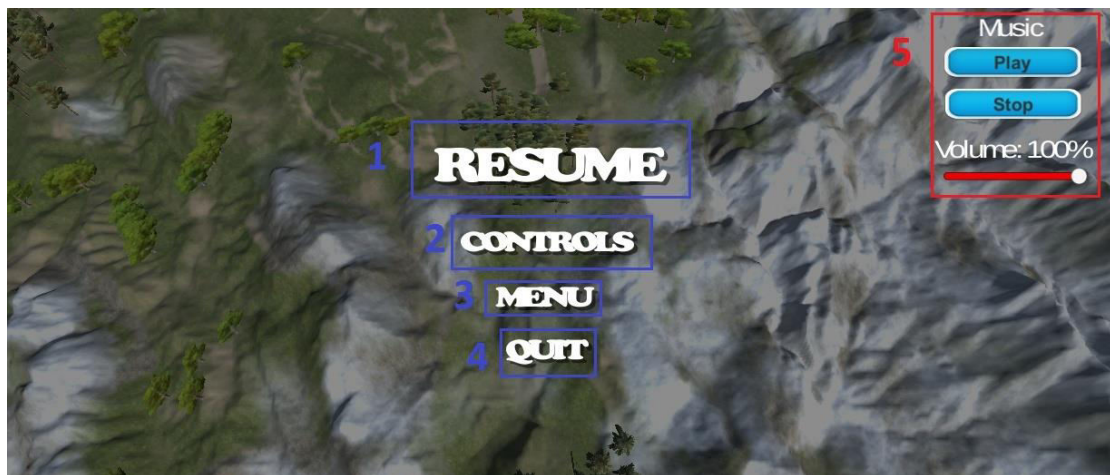
Εικόνα 3.10. Το μενού ρύθμισης του πλέγματος.

Στην εικόνα 3.10 βλέπουμε το μενού ρύθμισης του πλέγματος, δηλαδή των γραφημάτων της οπτικοποίησης. Εδώ, ο χρήστης μπορεί να επιλέξει το μέγεθος, το χρώμα και το σχήμα του γραφήματος. Για παράδειγμα αν ο χρήστης της εφαρμογής επιλέξει μέγεθος πλέγματος 50x50, τότε πάνω στο Terrain της σκηνής θα εμφανιστεί ένα γράφημα με το παραπάνω μέγεθος, χρώματος πράσινου και η μορφή του κάθε γράφου θα είναι ένας κύβος (Cube). Το ύψος του κάθε γράφου προσαρμόζεται ανάλογα με τη τιμή της θερμοκρασίας που έχει η συγκεκριμένη περιοχή. Αντίστοιχα και για τις υπόλοιπες επιλογές. Με το κουμπί Back επιστρέφει, χωρίς όμως να έχει εμφανιστεί στη σκηνή κάποιο νέο γράφημα.



Εικόνα 3.11. Το κουμπί Show Buttons.

Το μενού παύσης του προγράμματος απεικονίζεται στη παρακάτω εικόνα 3.12:



Εικόνα 3.12. Το μενού παύσης.

Τα κουμπιά του μενού παύσης επιτελούν τις παρακάτω λειτουργίες:

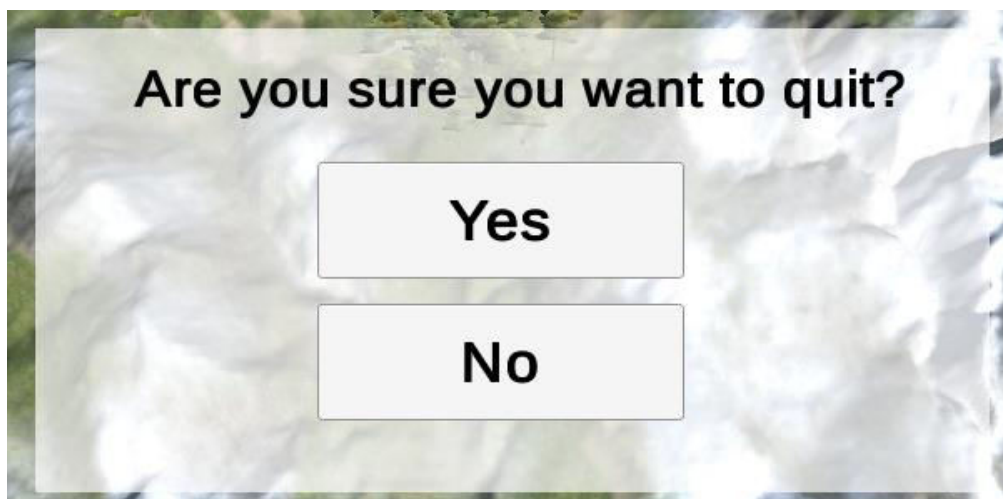
1. Κουμπί Resume: συνεχίζει το πρόγραμμα και κάνει έξοδο από το μενού παύσης.
2. Κουμπί Controls: ανοίγει ένα νέο παράθυρο και πληροφορεί το χρήστη για τα κουμπιά που μπορεί να χρησιμοποιήσει από το πληκτρολόγιο (εικόνα 3.13).
3. Κουμπί Menu: επιστρέφει το πρόγραμμα στο βασικό μενού (εικόνα 3.2).
4. Κουμπί Quit: κάνει έξοδο από το πρόγραμμα, αφού πρώτα εμφανίσει αντίστοιχο μήνυμα ερώτησης προς το χρήστη (εικόνα 3.14).
5. Μουσική: υπάρχει και η επιλογή μουσικής, πατώντας το Play αρχίζει και με το Stop σταματά, επίσης με τη μπάρα Volume ελέγχουμε την ένταση της.



Εικόνα 3.13. Τα διαθέσιμα controls του χρήστη.

Κάμερα: όπως υποδεικνύει και η εικόνα 3.13, για τη λειτουργία και κίνηση της κάμερας του προγράμματος, ο χρήστης μπορεί να χρησιμοποιήσει:

- Τα γράμματα W, A, S, D (κεφαλαία και μικρά) από το πληκτρολόγιο έτσι ώστε η κάμερα να κινηθεί μπροστά, αριστερά, πίσω και δεξιά αντίστοιχα.
- Τη scroll wheel (ροδέλα) του ποντικιού έτσι ώστε να κάνει zoom in/out στη σκηνή και στα γραφήματα της οπτικοποίησης.
- Κρατώντας πατημένο το δεξί κλικ και κινώντας το ποντίκι, μπορούμε να περιστρέφουμε τη κάμερα όπου θέλουμε μέσα στη σκηνή.



Εικόνα 3.14. Το μήνυμα ερώτησης για την έξοδο.

3.7 Scripts

Όπως αναφέρθηκε και στο Κεφάλαιο 2 (παράγραφος 2.1.4), η Unity υποστηρίζει δημιουργία αρχείων κώδικα (scripts) τα οποία μας βοηθούν να εκτελέσουμε διάφορες λειτουργίες. Η βασική γλώσσα προγραμματισμού είναι η C#. Για να δημιουργήσουμε ένα νέο script πάμε στο κομμάτι Project του Editor της μηχανής, κάνουμε δεξί κλικ και μετά Create > C# Script. Για να εκτελεστεί ένα script με επιτυχία πρέπει να γίνει διασύνδεση του (attach) με ένα αντικείμενο που υπάρχει στη σκηνή του παιχνιδιού μας. Η διασύνδεση μπορεί να γίνει με δύο τρόπους, είτε κάνοντας drag & drop το αρχείο script στον Inspector του επιλεγμένου αντικειμένου είτε κάνοντας κλικ στην επιλογή Add Component και μετά δίνουμε στην αναζήτηση το όνομα του script. Πριν την ανάλυση των scripts που χρησιμοποιήθηκαν στη παρούσα διπλωματική εργασία, στο σημείο αυτό αξίζει να αναφέρουμε τα βασικά χαρακτηριστικά και μεθόδους ενός script. Πιο αναλυτικά:

1. Awake: χρησιμοποιείται για την αρχικοποίηση μεταβλητών ή για τη κατάσταση του παιχνιδιού πριν αυτό ξεκινήσει. Η μέθοδος αυτή καλείται μια φορά κατά τη διάρκεια ζωής ενός script και πριν από τη μέθοδο Start.
2. Start: καλείται σε εκείνο το frame όπου το script ενεργοποιείται πριν από την Update, όπως και η Awake καλείται μια φορά.
3. Update: καλείται για κάθε frame, είναι η περισσότερο χρησιμοποιούμενη μέθοδος για τη δημιουργία οποιουδήποτε script.
4. LastUpdate: και αυτή η μέθοδος τρέχει για κάθε frame, η μόνη διαφορά είναι ότι τρέχει μετά τη μέθοδο Update.

Όλες αυτές οι μέθοδοι είναι void (δηλαδή δεν επιστρέφουν κάποια τιμή) με τις Start και Update να δημιουργούνται αυτόματα κάθε φορά που υπάρχει νέο script στην εργασία μας. Βέβαια η Unity εκτός από τις 4 παραπάνω βασικές μεθόδους, μας δίνει τη δυνατότητα να παράγουμε δικές μας μεθόδους, όπου επιτελούν μια συγκεκριμένη λειτουργία που επιλέγουμε εμείς. Στο αρχικό κομμάτι του κάθε script υπάρχουν και οι βιβλιοθήκες που χρησιμοποιεί, κάποιες από αυτές δηλώνονται αυτόματα από τη Unity όταν δημιουργείται το script, αν θέλουμε εμείς επιπλέον μπορούμε εύκολα να της δηλώσουμε χειροκίνητα. Για τη δήλωση τους χρησιμοποιείται η δεσμευμένη λέξη using και ακολουθεί το όνομα της βιβλιοθήκης. Στη συνέχεια αναγράφεται το όνομα του κάθε script που είναι και το όνομα της κλάσης. Σωστή τακτική όπως συμβαίνει και σε όλες τις γλώσσες προγραμματισμού είναι το όνομα της κλάσης να υποδηλώνει και τη λειτουργία που επιτελεί το συγκεκριμένο script, έτσι ώστε ο δημιουργός του προγράμματος να μπορεί εύκολα να τα διαχειριστεί αν είναι πολλά στον αριθμό. Τέλος, μέσα στο σώμα της κλάσης μπορούμε να γράψουμε το κώδικα (μεταβλητές, μεθόδους) του κάθε script. [5]

Στο σημείο αυτό θα συνεχίσουμε με την ανάλυση των scripts που δημιουργήσαμε και τα οποία βοήθησαν στην υλοποίηση της εργασίας. Αρχικά, θα αναγράφετε το όνομα του κάθε script (όνομα κλάσης) και οι μέθοδοι που περιέχει (σκοπός της μεθόδου, ορίσματα που δέχεται και επιστρεφόμενη τιμή).

1. GeoTIFFReader: η κλάση αυτή χρησιμοποιείται γιατί το άνοιγμα και το διάβασμα βασικών στοιχείων της εικόνας GeoTIFF που αναπαριστά το Terrain του προγράμματος. Με τη χρήση της βιβλιοθήκης (NuGet) BitMiracle.LibTiff.Classic [11] εξάγουμε το γεωγραφικό μήκος και πλάτος της εικόνας και τα αποθηκεύουμε σε λίστες από double δεδομένα [29]. Οι μέθοδοι της κλάσης είναι οι παρακάτω:

- CreateGridList: δημιουργεί τις λίστες από τις οποίες θα διαβάσουμε τα στοιχεία τους και θα κατασκευάσουμε τα πλέγματα (grids), δέχεται ως όρισμα ολόκληρη τη λίστα του γεωγραφικού μήκους και πλάτους και επιστρέφει τη λίστα που αντιπροσωπεύει το αντίστοιχο πλέγμα κάθε φορά.
- CreateGridTxtFile: δημιουργεί αρχεία κειμένου (.txt), σε αυτά αποθηκεύουμε το γεωγραφικό μήκος και πλάτος του κάθε πλέγματος, στη συνέχεια θα γίνει εισαγωγή των αρχείων αυτών στη Unity για να διαβαστούν εν τέλει από εκεί και να κατασκευαστούν σωστά τα πλέγματα πάνω στο Terrain της σκηνής. Δέχεται ως ορίσματα το όνομα του αρχείου κειμένου και τη λίστα του πλέγματος, δεν επιστρέφει κάποια τιμή (είναι void).

2. ReadRGBFromGeoTIFF: η κλάση αυτή χρησιμοποιείται για το άνοιγμα και διάβασμα των τιμών που περιέχουν τα pixels της εικόνας GeoTIFF, από τις τιμές αυτές θα προκύψουν τα δεδομένα θερμοκρασίας της περιοχής που περιγράφει η εικόνα. Κάθε pixel περιέχει τρεις τιμές, μια για κάθε χρώμα RGB (R: red, G: green, B: blue). Αφού διαβάσουμε τις τιμές των pixels τις αποθηκεύουμε σε τρεις διαφορετικές λίστες από double τιμές, μια λίστα για κάθε χρώμα. Και σε αυτή τη κλάση έγινε χρήση της βιβλιοθήκης BitMiracle.LibTiff.Classic [11]. Οι μέθοδοι της κλάσης είναι οι παρακάτω:

- findAverageTemp: υπολογίζει το μέσο όρο των τιμών της θερμοκρασίας (δηλαδή των τιμών από τα pixels) για κάθε περιοχή των παραπάνω πλεγμάτων (grids) όπως αυτά προέκυψαν από την ίδια εικόνα. Δέχεται ως ορίσματα τις 3 λίστες των χρωμάτων και επιστρέφει τη λίστα που κρατά τους μέσου όρους.
- CreateAverageTempTxtFile: δημιουργεί αρχεία κειμένου (.txt) τα οποία περιέχουν τους μέσους όρους κάθε περιοχής για κάθε πλέγμα όπως υπολογίστηκαν στη παραπάνω μέθοδο. Δέχεται ως όρισμα το όνομα του αρχείου και τη λίστα με τους μέσους όρους, δεν επιστρέφει κάποια τιμή.

3. MainMenu: η κλάση αυτή διαχειρίζεται το βασικό μενού το προγράμματος και αφορά τη πρώτη σκηνή (εικόνα 3.2). Δίνει λειτουργίες στα τρία κουμπιά που περιέχονται σε αυτό (Start, About, Quit) με τη βοήθεια των παρακάτω μεθόδων:

- PlayGame: με την ειδική μέθοδο `SceneManager.LoadScene()` που προσφέρει η Unity, όταν πατηθεί το κουμπί Start του βασικού μενού μεταβαίνουμε στη δεύτερη σκηνή, δεν δέχεται ορίσματα και δεν έχει επιστρεφόμενη τιμή [18].
- AboutGame: εμφανίζεται ένα πλαίσιο που μας δίνει πληροφορίες για βασικά στοιχεία της διπλωματικής εργασίας, δεν δέχεται ορίσματα και δεν έχει επιστρεφόμενη τιμή. [20]
- BackFromAboutPage: όταν ο χρήστης ανοίξει τη σελίδα About που αναφέρθηκε παραπάνω, πρέπει με κάποιο κουμπί να επιστρέψει πίσω στο βασικό μενού, αυτό το σκοπό επιτελεί η μέθοδος αυτή. [20]
- QuitGame: όταν ο χρήστης επιλέξει έξοδο από το πρόγραμμα του εμφανίζεται ένα μήνυμα ερώτησης εξόδου με δύο επιλογές Yes ή No, αν επιλέξει "Yes" τότε καλείται η μέθοδος `YesQuitGame()` η οποία κάνει έξοδο από το πρόγραμμα και κλείνει το παράθυρο, διαφορετικά καλείται η `NoQuitGame()` η οποία μας γυρνά πίσω στο βασικό μενού. Καμία από τις παραπάνω μεθόδους δεν δέχεται ορίσματα, ούτε επιστρέφει κάποια τιμή. [19]

4. CameraController: η κλάση αυτή βοηθά στη διαχείριση της κεντρικής κάμερας της δεύτερης σκηνής του προγράμματος (εικόνα 3.9). Το script αυτό έχει γίνει attach με το αντικείμενο Main Camera της δεύτερης σκηνής. Περιέχει μόνο τη μέθοδο `Update` στην οποία αν το πρόγραμμα δεχτεί κάποιους από τους χαρακτήρες W, A, S, D από το πληκτρολόγιο κάνει και την αντίστοιχη κίνηση όπως αναφέραμε. Επίσης δίνει λειτουργία και στη ροδέλα του ποντικιού για zoom in/out στη σκηνή, τέλος θέτει όρια στη ταχύτητα κίνησης της κάμερας αλλά και όρια έτσι ώστε η κάμερα να μην εστιάζει σε όλη τη σκηνή παρά μόνο στο κομμάτι που έχει ενδιαφέρον για το χρήστη, δηλαδή το Terrain και τα γραφήματα οπτικοποίησης. Η μέθοδος δεν δέχεται ορίσματα και δεν επιστρέφει κάποια τιμή. [18]

5. ShowMusicVolume: η κλάση αυτή περιέχει μια και μόνο μέθοδο. Η μέθοδος αυτή ονομάζεται `textUpdate`, αυτό που κάνει είναι να μας εμφανίζει το κείμενο το οποίο αντιπροσωπεύει την ένταση της μουσικής του προγράμματος σε ποσοστό επί τοις εκατό (εικόνα 3.12). Δέχεται ως όρισμα μια float μεταβλητή που δείχνει τη τιμή της έντασης, δεν επιστρέφει τίποτα. [21]

6. ChangeMusicVolume: η κλάση αυτή περιέχει μόνο τη μέθοδο `Update`. Μέσω αυτής ρυθμίζουμε την ένταση της μουσικής (αύξηση ή μείωση) ανάλογα με τη κίνηση του slider που έχουμε (εικόνα 3.12) δεξιά ή αριστερά. Επίσης, όταν η

μουσική είναι ανενεργή (Stop) ο slider της έντασης εξαφανίζεται, όταν όμως ενεργοποιηθεί ξανά (Play) εμφανίζεται. [21]

7. PauseGameMenu: η μέθοδος αυτή διαχειρίζεται το μενού παύσης του προγράμματος (εικόνα 3.12). Δίνει λειτουργίες σε όλα τα κουμπιά του μενού με τη βοήθεια των παρακάτω μεθόδων:

- Update: όταν πατηθεί το πλήκτρο Escape από το πληκτρολόγιο ξεκινά η κλήση του συγκεκριμένου script. Δε δέχεται ορίσματα, δεν έχει επιστρεφόμενη τιμή.
- Resume: όταν πατηθεί το κουμπί "Resume" στο μενού παύσης, γυρνάμε ξανά στην αρχική σκηνή της οπτικοποίησης των γραφημάτων και το μενού παύσης εξαφανίζεται. Δε δέχεται ορίσματα, δεν έχει επιστρεφόμενη τιμή. [20]
- Pause: εκτελεί την αντίθετη λειτουργία από τη παραπάνω μέθοδο. Όταν πατηθεί το Escape καλείται η μέθοδος και εμφανίζεται στο χρήστη το μενού παύσης. Δεν δέχεται ορίσματα, δεν έχει επιστρεφόμενη τιμή. [24]
- ControlsMenu: όταν πατηθεί το κουμπί "Controls" εμφανίζονται στο χρήστη τα διαθέσιμα πλήκτρα που μπορεί να χρησιμοποιήσει (εικόνα 3.13). Δεν δέχεται ορίσματα, δεν έχει επιστρεφόμενη τιμή.
- BackFromControlsMenu: όταν πατηθεί το κουμπί "Back" που βρίσκεται στο μενού των πλήκτρων επιστρέφουμε στο αρχικό μενού παύσης. Δεν δέχεται ορίσματα, δεν έχει επιστρεφόμενη τιμή.
- LoadMenu: όταν πατηθεί το κουμπί "Menu", το πρόγραμμα μας επιστρέφει στο βασικό μενού (σκηνή 1). Δεν δέχεται ορίσματα, δεν έχει επιστρεφόμενη τιμή. [20]
- QuitGame: ο χρήστης με το κουμπί "Quit" μπορεί να κάνει οριστική έξοδο από το πρόγραμμα και να κλείσει το παράθυρο του. Όταν πατηθεί το κουμπί εμφανίζεται το μήνυμα ερώτησης εξόδου (εικόνα 3.14) με δύο επιλογές Yes ή No. Αν γίνει κλικ στο κουμπί "Yes" καλείται η μέθοδος YesQuitGame() όπου κλείνει το πρόγραμμα οριστικά, αντίθετα αν γίνει κλικ στο κουμπί "No" το πρόγραμμα μας επιστρέφει στο μενού παύσης. [18]

8. ReadTxtFiles: η κλάση αυτή διαβάζει τα αρχεία κειμένου txt που δημιουργήσαμε και εξηγήσαμε στις 2 αρχικές κλάσεις. Τα αρχεία αυτά περιέχουν το γεωγραφικό μήκος και πλάτος (latitude & longitude) για το κάθε grid όπως και τους μέσους όρους των θερμοκρασιών. Στην αρχή της κλάσης πριν από τον ορισμό των μεθόδων της, υπάρχουν global private (δεν μπορούν να προσπελαστούν από άλλη κλάση) μεταβλητές όπου σε μορφή λίστας από double τιμές αποθηκεύουν όλα τα στοιχεία των grids αλλά και των θερμοκρασιών που

διαβάζουμε από τα αρχεία κειμένου. Οι μέθοδοι που περιέχει η κλάση είναι οι παρακάτω:

- ReadTxtFiles: αποτελεί τον constructor της κλάσης. Ο constructor είναι μια ειδική μέθοδος η οποία έχει ίδιο όνομα με αυτό της κλάσης που τη περιέχει. Καλείται αυτόματα κάθε φορά που δημιουργείται ένα αντικείμενο της παρούσας κλάσης από κάποια άλλη. Στην αρχή του υπάρχουν όλα τα μονοπάτια (paths) των αρχείων κειμένου txt έτσι ώστε να διαβαστούν από τη Unity, τα μονοπάτια των αρχείων τα αποθηκεύουμε σε μεταβλητές τύπου string, μια μεταβλητή για κάθε αρχείο ξεχωριστά. Στη συνέχεια καλούνται οι άλλες μέθοδοι της κλάσης οι οποίες θα αναλυθούν στις παρακάτω παραγράφους. Ο συγκεκριμένος constructor δεν δέχεται ορίσματα και δεν επιστρέφει κάποια τιμή. [22]
- ReadFile: είναι η μέθοδος η οποία διαβάζει κάθε αρχείο κειμένου γραμμή προς γραμμή και αποθηκεύει ότι διαβάζει στην αντίστοιχη λίστα. Δέχεται ένα όρισμα, μια string μεταβλητή όπου περιγράφει το μονοπάτι του αρχείου και επιστρέφει τη λίστα (τύπου double) που έχει αποθηκεύσει τα δεδομένα του αρχείου.
- ConvertLatToUnity: είναι η μέθοδος που μετατρέπει το αρχικό γεωγραφικό πλάτος (latitude) της περιοχής που διαβάζουμε από την εικόνα GeoTIFF σε συντεταγμένες της μηχανής Unity έτσι ώστε τα γραφήματα οπτικοποίησης να "πέσουν" πάνω στο Terrain που έχουμε δημιουργήσει. Η μετατροπή δεν πραγματοποιείται τυχαία. Με τη βοήθεια της βιβλιοθήκης GDAL μπορούμε να γνωρίζουμε με ακρίβεια τα 4 ακραία σημεία της περιοχής που περιγράφει η εικόνα, επίσης μέσω της Unity γνωρίζουμε και τα 4 ακραία σημεία του Terrain στη σκηνή μας οπότε κάνοντας την αντιστοίχιση, προκύπτουν οι συντεταγμένες των γραφημάτων στο Terrain. Δέχεται ως όρισμα την αρχική λίστα με τα γεωγραφικά πλάτη κάνει τη μετατροπή και επιστρέφει μια νέα λίστα τύπου double που περιέχει τις νέες συντεταγμένες.
- ConvertLonToUnity: εκτελεί την ίδια διαδικασία όπως και η παραπάνω μέθοδος, αλλά αυτή τη φορά για το γεωγραφικό μήκος (longitude) της περιοχής. Δέχεται ως όρισμα την αρχική λίστα με τα γεωγραφικά μήκη κάνει τη μετατροπή και επιστρέφει τη νέα λίστα τύπου double που κρατά τις νέες συντεταγμένες.
- getGrid4x4Lat: όπως αναφέρθηκε στην αρχή της κλάσης αυτής όλες οι global μεταβλητές που περιέχει είναι private (ιδιωτικές) και έτσι δεν μπορούν να προσπελαστούν από άλλη διαφορετική κλάση. Ο μόνος τρόπος για τη προσπέλαση των μεταβλητών αυτών είναι να δημιουργήσουμε μέσα στη κλάση μεθόδους οι οποίες να τις επιστρέφουν απλά με την εντολή return χωρίς κάποιο όρισμα. Οι μέθοδοι αυτοί ονομάζονται getters, για αυτό και στην αρχή του ονόματος της μεθόδου υπάρχει η λέξη get. Μια τέτοια μέθοδος έχει δημιουργηθεί για κάθε

μέγεθος grid (λίστες με τις συντεταγμένες) και για κάθε λίστα που περιέχει τους μέσους όρους των θερμοκρασιών, έτσι ώστε όλες οι λίστες αυτές να μπορέσουν να προσπελαστούν από την επόμενη κλάση η οποία δημιουργεί και τα γραφήματα οπτικοποίησης. Στη συγκεκριμένη περίπτωση όπως λέει και το όνομα της μεθόδου, επιστρέφουμε το γεωγραφικό μήκος του grid μεγέθους 4x4. Αντίστοιχα ονόματα υπάρχουν και στις άλλες μεθόδους.

9. CreateGameObjects: μαζί με τη προηγούμενη κλάση είναι οι πιο σημαντικές για τη δημιουργία των γραφημάτων οπτικοποίησης. Η κλάση αυτή διαβάζει τα αρχεία κειμένου και κατασκευάζει πάνω στο Terrain της σκηνής τα γραφήματα. Στην αρχή της, πριν τη δήλωση των μεθόδων η κλάση περιέχει global private μεταβλητές. Στις μεταβλητές αυτές υπάρχουν οι λίστες τύπου double που είναι αποθηκευμένες οι συντεταγμένες για κάθε γράφημα του Terrain (γεωγραφικό μήκος & πλάτος) αλλά και οι λίστες των μέσων όρων των θερμοκρασιών που θα καθορίσουν το ύψος του κάθε γραφήματος. Πιο αναλυτικά, οι μέθοδοι που περιέχονται στη κλάση είναι οι:

- Start: στη μέθοδο αυτή γίνεται η αποθήκευση των τιμών σε όλες τις λίστες. Αρχικά, δημιουργούμε ένα αντικείμενο της κλάσης ReadTxtFiles, με αυτό το τρόπο έχουμε πρόσβαση στις μεθόδους της. Η κάθε μια από τις λίστες "γεμίζει" με τιμές καλώντας την αντίστοιχη μέθοδο της κλάσης που αναφέρθηκε παραπάνω. Δεν δέχεται ορίσματα, δεν έχει επιστρεφόμενη τιμή.
- CreateGrid4x4: η μέθοδος αυτή δημιουργεί το grid με μέγεθος 4x4. Στο σώμα της μεθόδου καλούνται άλλες δύο μέθοδοι, οι BackFromGridOptionsMenu() και η CreateGrid() οι οποίες θα εξηγηθούν παρακάτω. Αντίστοιχη μέθοδος υπάρχει για όλα τα μεγέθη γραφημάτων, οι μέθοδοι αυτοί εκτελούνται όταν ο χρήστης πατήσει το αντίστοιχο κουμπί/μέγεθος για το grid στο Grid Options μενού (εικόνα 3.10). Δε δέχεται ορίσματα, δεν επιστρέφει τιμή.
- CreateGrid: είναι η τελική μέθοδος που δημιουργεί το γράφημα πάνω στο Terrain της σκηνής. Στην αρχή της μεθόδου υπάρχει ένας έλεγχος, αν πάνω στο Terrain υπάρχει ήδη άλλο γράφημα το διαγράφει και δημιουργεί το καινούργιο που επιλέγει ο χρήστης, ο έλεγχος αυτός πραγματοποιείται με μια μεταβλητή Boolean. Μετά ξεκινά ένας διπλός βρόγχος for, μέσα σε αυτόν δημιουργούνται τα γραφήματα και τοποθετούνται στις σωστές θέσεις πάνω στο Terrain. Η μέθοδος αυτή δέχεται 5 ορίσματα και δεν επιστρέφει κάποια τιμή. Τα ορίσματα της είναι οι 3 λίστες (γεωγραφικό μήκος, πλάτος, θερμοκρασία), μια ακέραια μεταβλητή η οποία προσδιορίζει το τύπο (σχήμα) και το χρώμα των γραφημάτων και τέλος μια μεταβλητή τύπου float η οποία προσδιορίζει τη κλιμάκωση (scale) των γραφημάτων. [24]

- DestroyGrid: ο σκοπός της είναι να "καταστρέφει" το γράφημα της σκηνής όταν ο χρήστης πατήσει το κουμπί "Clear". Επίσης, χρησιμοποιείται και όταν υπάρχει ήδη άλλο πλέγμα στη σκηνή και ο χρήστης πατήσει να δει κάποιο διαφορετικό, τότε διαγράφει το παλιό και προβάλλει το καινούργιο. Δεν δέχεται ορίσματα, δεν επιστρέφει τιμή. [24]
- GridOptionsButtonClicked: καλείται όταν ο χρήστης πατήσει το κουμπί "Grid Options" στη σκηνή της προσομοίωσης, εμφανίζει το μενού ρυθμίσεων του πλέγματος (εικόνα 3.10). Δεν δέχεται ορίσματα, δεν επιστρέφει τιμή.
- BackFromGridOptionsMenu: όταν ο χρήστης κάνει κλικ στο κουμπί "Back" του μενού ρυθμίσεων του πλέγματος, το πρόγραμμα μας επιστρέφει στην αρχική μορφή της δεύτερης σκηνής (εικόνα 3.9). Δεν δέχεται ορίσματα, δεν επιστρέφει τιμή.
- ClearButtonClicked: όπως αναφέρθηκε παραπάνω, όταν πατηθεί το κουμπί "Clear" το γράφημα της σκηνής διαγράφεται. Δεν δέχεται ορίσματα, δεν επιστρέφει τιμή.
- HideButtonClicked: όταν πατηθεί το κουμπί "Hide Buttons", τότε όλα τα κουμπιά της σκηνής εξαφανίζονται έτσι ώστε ο χρήστης να μπορεί να δει καλύτερα το Terrain και τα γραφήματα οπτικοποίησης. Δεν δέχεται ορίσματα, δεν επιστρέφει τιμή.
- ShowButtonClicked: εκτελεί την αντίστροφη διαδικασία σε σχέση με τη παραπάνω μέθοδο, όταν πατηθεί το κουμπί "Show Buttons", όλα τα κουμπιά της σκηνής εμφανίζονται ξανά. Δεν δέχεται ορίσματα, δεν επιστρέφει τιμή.
- ColorGrid: δίνει χρώμα στα γραφήματα οπτικοποίησης, καλείται μέσα στη μέθοδο CreateGrid() και ανάλογα με τα ορίσματα που τις δίνονται δημιουργεί και το ανάλογο χρώμα, δεν επιστρέφει τιμή. [24]

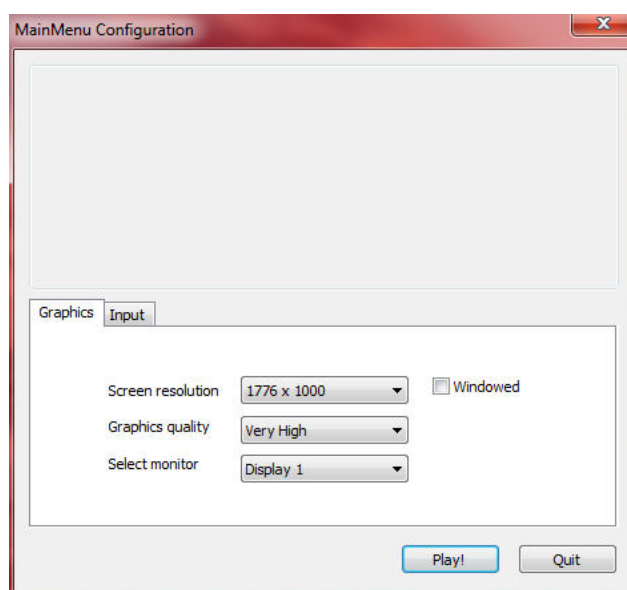
Τέλος, αξίζει να σημειωθεί ότι τα αρχεία κώδικα στη γλώσσα C# ή και σε οποιαδήποτε άλλη γλώσσα εκτός από το όνομα τους (όνομα κλάσης) περιέχουν και μια κατάληξη η οποία προσδιορίζει τη γλώσσα που έχει χρησιμοποιηθεί στο συγκεκριμένο αρχείο κώδικα. Στη περίπτωση μας, τα αρχεία του κώδικα έχουν κατάληξη .cs (C Sharp), δηλαδή για παράδειγμα CreateGameObjects.cs. Εκτός του περιβάλλοντος της Unity μπορούν να δημιουργηθούν τέτοια αρχεία και να μεταφερθούν από εμάς χειροκίνητα μέσα στο φάκελο της εργασίας, θα έχουμε ακριβώς το ίδιο αποτέλεσμα. Για τη δημιουργία και επεξεργασία τέτοιων αρχείων εκτός από τα Visual Studio 2017 και MonoDevelop, μπορούμε να χρησιμοποιήσουμε και άλλα προγράμματα, όπως απλούς editors. (Notepad++, Sublime Text, Atom, Coda και άλλοι)

3.8 Δημιουργία εκτελέσιμου αρχείου

Στην ενότητα αυτή θα αναλύσουμε το τελικό στάδιο της εργασίας, τη δημιουργία του εκτελέσιμου αρχείου [18]. Η Unity 3D προσφέρει εύκολα και γρήγορα ένα τρόπο δημιουργίας εκτελέσιμων αρχείων σε ποικίλες πλατφόρμες. Η δωρεάν έκδοση της μηχανής, που χρησιμοποιήθηκε σε αυτή την εργασία δε προσφέρει τη δυνατότητα έκδοσης της εργασίας σε όλα τα διαθέσιμα συστήματα που υποστηρίζονται. Βέβαια, η δημιουργία σε λειτουργικά συστήματα Windows, Mac & Linux είναι εφικτή. Η διαδικασία αυτή στη Unity ονομάζεται Build. Στη περίπτωση που υπάρχουν σφάλματα (compile errors) στην εργασία μας, το Build του προγράμματος θα αποτύχει. Τα κύρια βήματα που πρέπει να ακολουθηθούν είναι τα παρακάτω:

1. Από το κεντρικό μενού της Unity επιλέγουμε File > Build Settings.
2. Ανοίγει το σχετικό παράθυρο και στο πλαίσιο Scenes In Build κάνουμε εισαγωγή όλες τις σκηνές που έχουμε δημιουργήσει στην εργασία μας.
3. Στο πλαίσιο Platform επιλέγουμε τη πλατφόρμα που θέλουμε η εργασία μας να κάνει Build.
4. Τέλος, κάνουμε κλικ στην επιλογή Build στο κάτω δεξιά μέρος του παραθύρου και μας ανοίγει ο File Explorer του λειτουργικού μας συστήματος, δίνουμε ένα όνομα στο εκτελέσιμο αρχείο μας και επιλέγουμε το φάκελο αποθήκευσης του.

Αφού δημιουργηθεί το εκτελέσιμο αρχείο (στα Windows έχει κατάληξη .exe) μπορούμε εύκολα να κάνουμε διπλό κλικ στο σχετικό εικονίδιο και πριν ξεκινήσει το παιχνίδι, μας εμφανίζεται το παράθυρο της εικόνας 3.19:



Εικόνα 3.19. Το παράθυρο πριν την εκκίνηση του παιχνιδιού.

Να σημειωθεί ότι μαζί με το εκτελέσιμο (exe) αρχείο παράγεται και ένα dll (Dynamic-link library) αρχείο και ένας φάκελος που περιέχει διάφορα δεδομένα για την εργασία. Για να την ορθή εκτέλεση της, πρέπει όλα αυτά να συνυπάρχουν στον ίδιο φάκελο με το exe. Στο παράθυρο της εικόνας 3.19 βλέπουμε ότι πριν την εκκίνηση του προγράμματος, η Unity μας δίνει τη δυνατότητα να θέσουμε εμείς κάποιες βασικές ρυθμίσεις στο παιχνίδι μας. Υπάρχουν δύο βασικές καρτέλες, οι Graphics και Input. Στη πρώτη, μπορούμε να καθορίσουμε την ανάλυση του παιχνιδιού, τη ποιότητα των γραφικών στοιχείων του, την οθόνη που θα προβληθεί αλλά και αν επιθυμούμε εκτέλεση σε πλήρη οθόνη ή όχι (με την επιλογή Windowed). Η δεύτερη καρτέλα χρησιμοποιείται συνήθως σε πιο μεγάλες εργασίες. Μας πληροφορεί για τα πλήκτρα που έχουμε στη διάθεση μας, καθώς το παιχνίδι μας εκτελείται. Επίσης, παρέχει και τη δυνατότητα προσθήκης gaming controller. Τέλος, κάνουμε κλικ στο κουμπί Play στο κάτω μέρος του παραθύρου και είμαστε έτοιμοι να παίξουμε! Όπως αναφέραμε και στο Κεφάλαιο 2.1 η Unity προσφέρει έκδοση των παιχνιδιών της σε μια μεγάλη ποικιλία πλατφόρμων, όπως υποδεικνύει και η παρακάτω εικόνα 3.20:

BUILD ONCE **DEPLOY ANYWHERE**

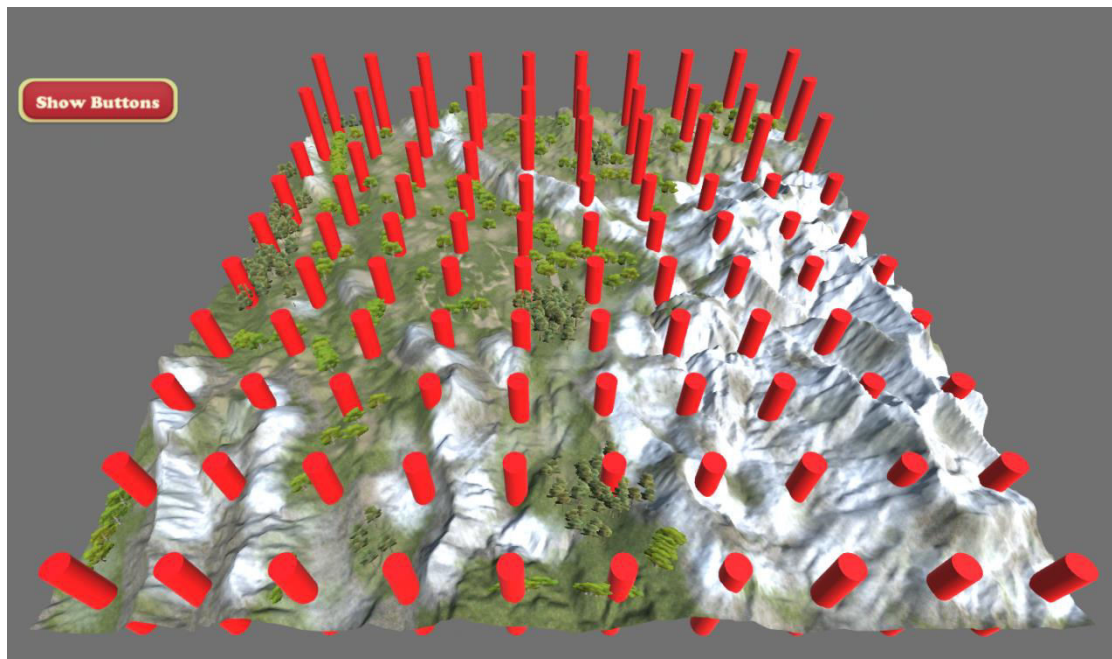


Εικόνα 3.20. Οι πλατφόρμες που υποστηρίζει η Unity 3D. (Πηγή: Google)

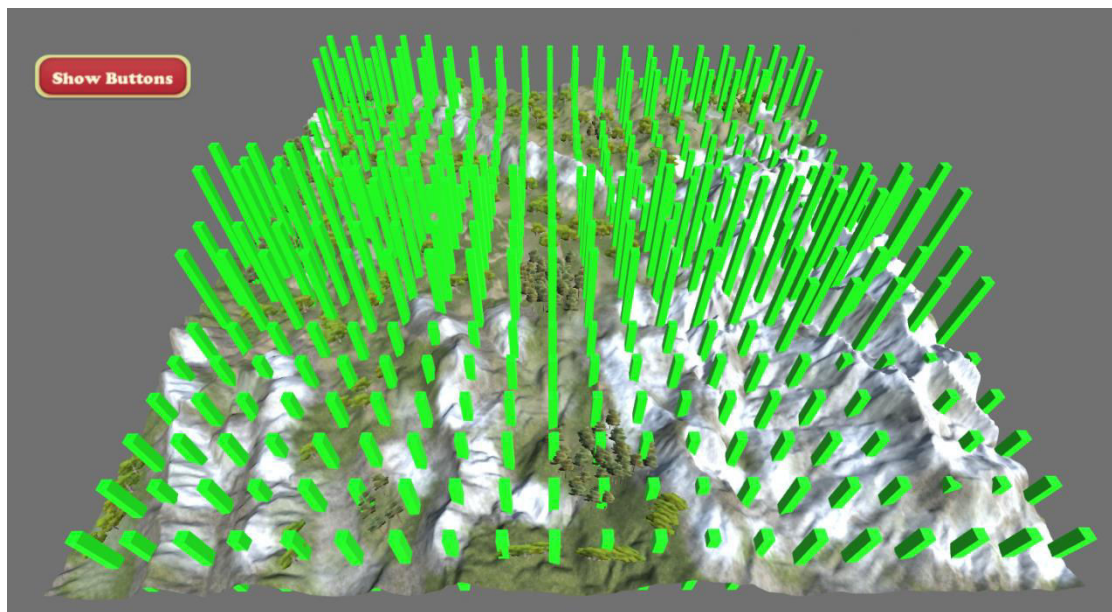
Κεφάλαιο 4. Πειραματική Αξιολόγηση

4.1 Ποιοτική Αξιολόγηση

Στη παράγραφο αυτή θα παραθέσουμε κάποιες εικόνες (screenshots) μέσα από το πρόγραμμα κατά την εκτέλεση του. Πιο συγκεκριμένα, θα δώσουμε εικόνες των γραφημάτων θερμοκρασίας καθώς αυτά εμφανίζονται πάνω στο Terrain της σκηνής.

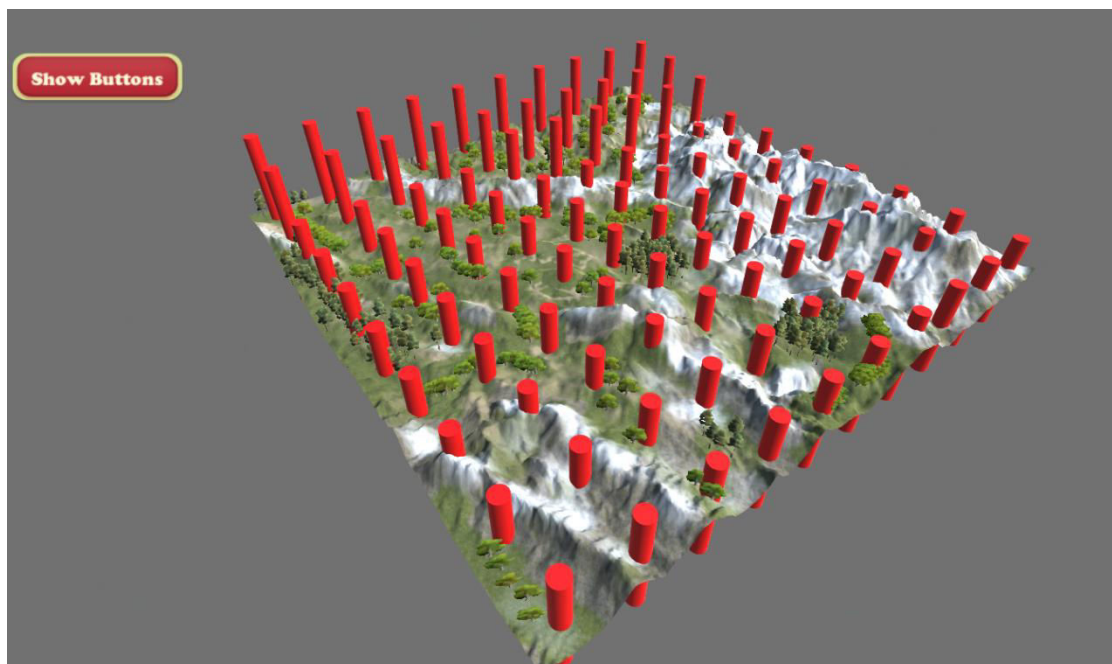


Εικόνα 3.21. Grid 10x10, χρώμα: κόκκινο, σχήμα απεικόνισης: Cylinder.

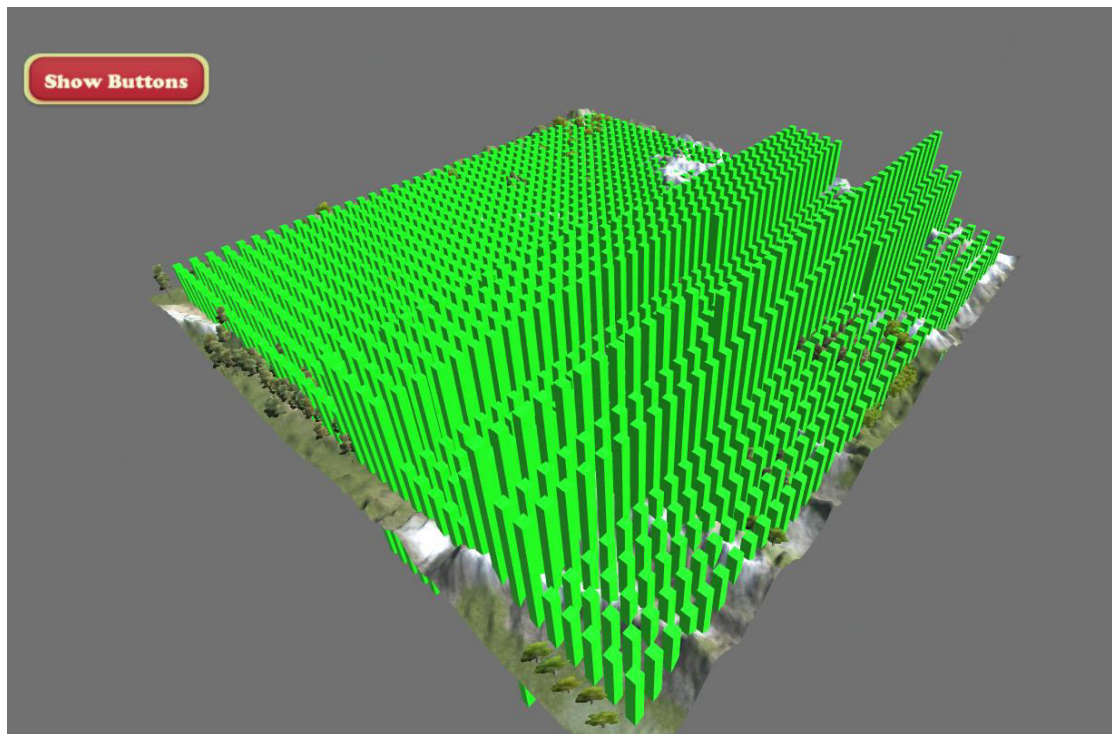


Εικόνα 3.22. Grid 20x20, χρώμα: πράσινο, σχήμα απεικόνισης: Cube.

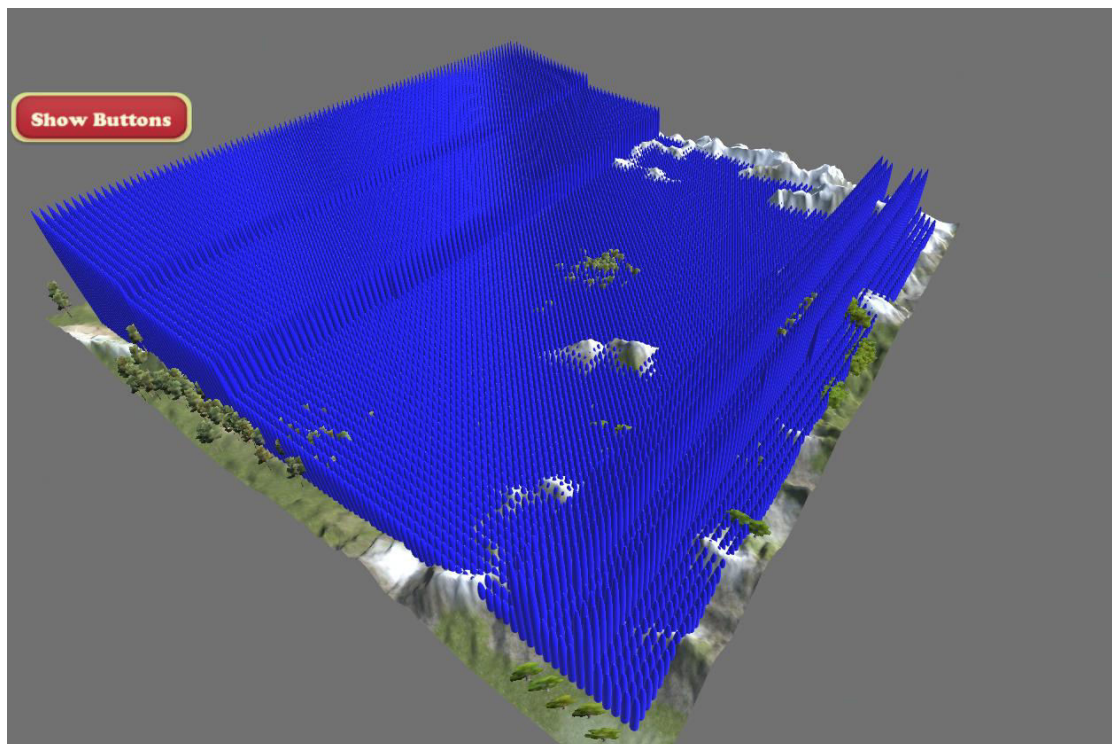
Εικόνες από διαφορετική οπτική, όπου απεικονίζονται με πιο εμφανή τρόπο τα μέγιστα και ελάχιστα σημεία των γραφημάτων οπτικοποίησης:



Εικόνα 3.22. Grid 10x10, χρώμα: κόκκινο, σχήμα απεικόνισης: Cylinder.



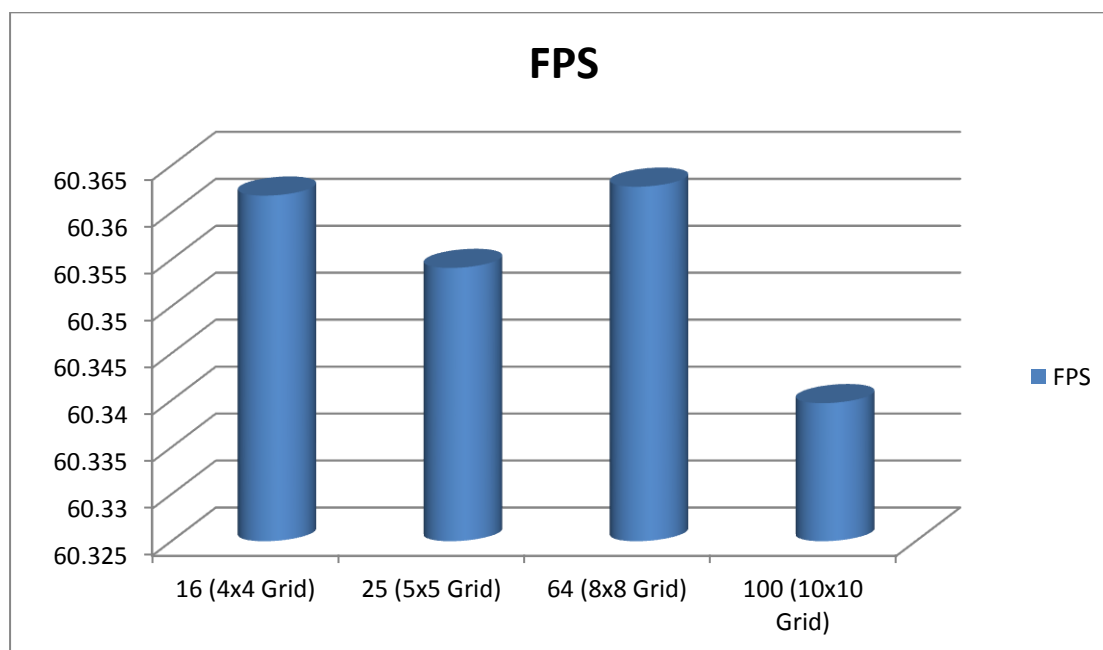
Εικόνα 3.23. Grid 40x40, χρώμα: πράσινο, σχήμα απεικόνισης: Cube.



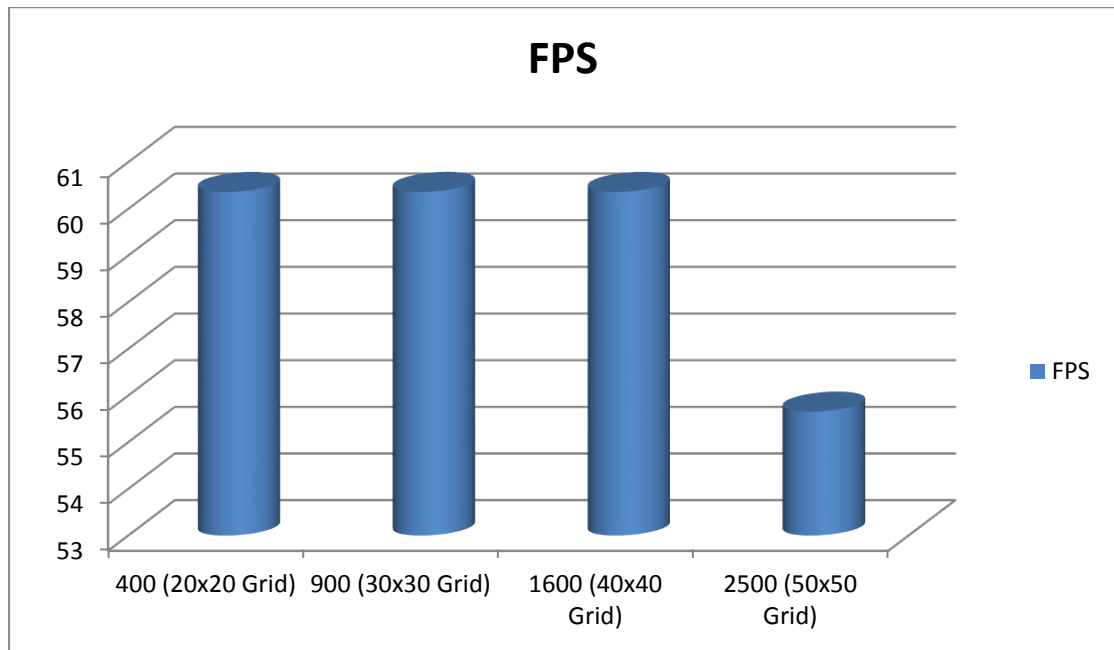
Εικόνα 3.24. Grid 100x100, χρώμα: μπλε, σχήμα απεικόνισης: Capsule.

4.2 Ποσοτική Αξιολόγηση

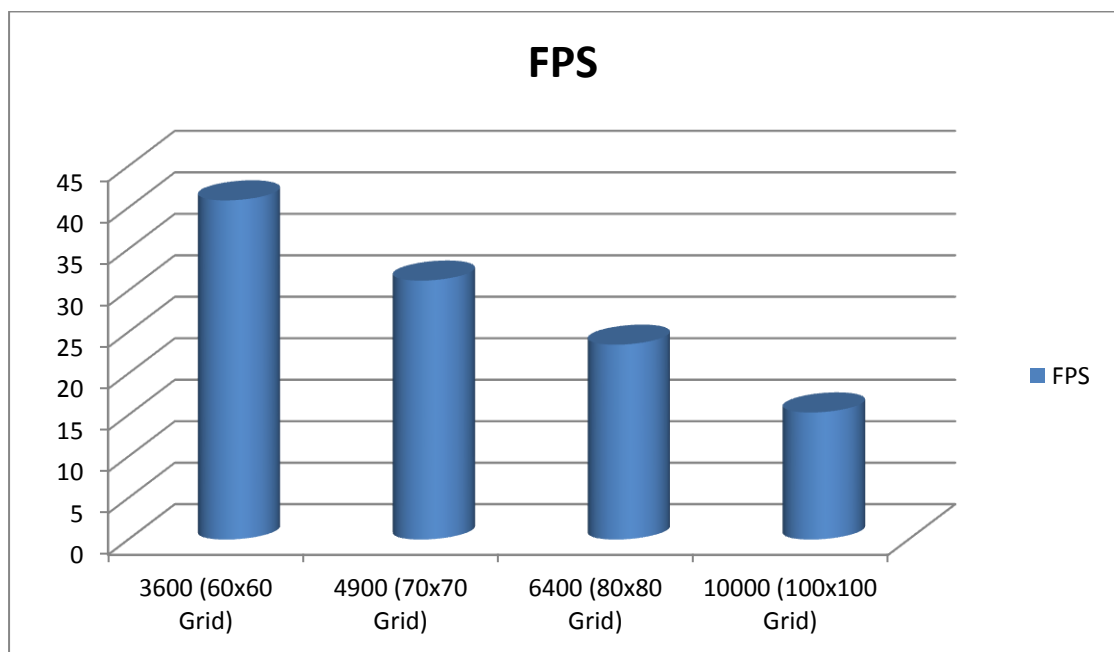
Υπάρχουν πολλές μετρικές οι οποίες αξιολογούν τη ποιότητα ενός βίντεο παιχνιδιού ή μιας προσομοίωσης. Μια από αυτές τις μετρικές είναι και τα FPS (Frames Per Second) [27]. Ο όρος αυτός στα Ελληνικά μπορεί να ερμηνευθεί ως "καρέ ανά δευτερόλεπτο". Στην επίσημη βιβλιογραφία μπορούμε να συναντήσουμε τη μετρική αυτή και με το όνομα Frame rate (Ρυθμός καρέ) [26]. Πιο συγκεκριμένα, η μετρική εκφράζει τη συχνότητα (ρυθμό) που διαδοχικές εικόνες που καλούνται καρέ εμφανίζονται στην οθόνη. Ο όρος αυτό βρίσκεται εφαρμογή με τον ίδιο τρόπο και σε ταινίες, βίντεο κάμερες και γραφικά υπολογιστών. Το Frame rate μπορεί να χρησιμοποιηθεί και με το όνομα frame frequency (Συχνότητα των καρέ), όπου μετράται σε Hertz. Όσο περισσότερα είναι τα FPS, τόσο πιο "όμορφο" φαίνεται το βίντεο. Τα σημερινά βίντεο έχουν συνήθως 30 FPS ή περισσότερα. Διαφορετικοί τύποι αρχείων βίντεο παρουσιάζουν και διαφορετικούς ρυθμούς FPS. Μικροί ρυθμοί FPS δημιουργούν μικρότερα αρχεία ως προς το μέγεθος. Η παράγραφος αυτή έχει ως απώτερο σκοπό να μας δείξει με τη βοήθεια γράφων τα FPS που παράγονται ανάλογα με το πλήθος (μέγεθος grid) των γραφημάτων θερμοκρασίας. Στο κάθετο άξονα του γράφου θα υπάρχουν τα παραγόμενα FPS και στον οριζόντιο άξονα το πλήθος των γραφημάτων (μέγεθος grid) του προγράμματος. Πιο αναλυτικά:



Γράφημα 4.1. FPS για μεγέθη Grid 4x4, 5x5, 8x8, 10x10



Γράφημα 4.2. FPS για μεγέθη Grid 20x20, 30x30, 40x40, 50x50



Γράφημα 4.3. FPS για μεγέθη Grid 60x60, 70x70, 80x80, 100x100

Παρατηρούμε ότι στους αρχικούς γράφους και όσο το μέγεθος του πλέγματος (grid) παραμένει μικρό, ο αριθμός των FPS δεν έχει σημαντικές μεταβολές και κυμαίνεται κοντά στη τιμή 60. Όσο αυξάνεται όμως το μέγεθος του πλέγματος και συνεπώς το πλήθος των γραφημάτων θερμοκρασίας, ο αριθμός των FPS της προσομοίωσης μειώνεται μέχρι που λαμβάνει τη κατώτερη τιμή του για μέγεθος πλέγματος 100x100. Αυτό το φαινόμενο συμβαίνει για το λόγο ότι η μηχανή γραφικών Unity 3D έχει να διαχειριστεί πολλά αντικείμενα και γραφικά στοιχεία όσο το μέγεθος του πλέγματος αυξάνεται, έτσι η προσομοίωση αρχίζει να "κολλάει" και ο αριθμός των FPS αρχίζει να μειώνεται αισθητά, αυτό φαίνεται εύκολα καθώς τρέχει το πρόγραμμα ακόμη και από ένα "μη έμπειρο" μάτι. Για την εύρεση του αριθμού των FPS μπορούμε να χρησιμοποιήσουμε διάφορα εργαλεία. Στη παρούσα εργασία χρησιμοποιήθηκε ένα script αρχείο σε γλώσσα C#. Το αρχείο αυτό έγινε attach σε ένα αντικείμενο της σκηνής και εμφάνισε τα FPS στο πάνω αριστερό κομμάτι της οθόνης. Υπάρχουν φυσικά και άλλοι τρόποι υπολογισμού των FPS σε ένα πρόγραμμα της Unity 3D. Η μηχανή παρέχει ένα έτοιμο πακέτο μέσω του Asset Store της. Για την εγκατάσταση του πακέτου μεταβαίνουμε στη μπάρα του μενού > Assets > Import Package > Utility, στο πακέτο αυτό υπάρχουν αρχεία κώδικα όπου η χρήση τους μας υπολογίζει τον αριθμό των FPS. Τέλος, αξίζει να σημειωθεί ότι υπάρχουν και έτοιμα προγράμματα που μπορούμε να βρούμε εύκολα και δωρεάν online, όπως το Fraps το οποίο μας βοηθά στον υπολογισμό αυτό.

Κεφάλαιο 5. Συμπεράσματα

5.1 Προβλήματα που αντιμετώπισα

Το μεγαλύτερο πρόβλημα που αντιμετώπισα κατά τη διάρκεια δημιουργίας της εργασίας μου ήταν η εύρεση και η εξαγωγή των δεδομένων από τις εικόνες GeoTIFF που έπρεπε να διαχειριστώ. Το πρόγραμμα Visual Studio 2017 που με βοήθησε στη δημιουργία κώδικα στη γλώσσα C# μέσω της Unity 3D δεν εγκαθιστούσε με επιτυχία τη βιβλιοθήκη (NuGet) BitMiracle.LibTiff [11]. Για το λόγο αυτό αναγκάστηκα να κατεβάσω παλιότερη έκδοση του προγράμματος, στην έκδοση αυτή πραγματοποιήθηκε με επιτυχία η εγκατάσταση της παραπάνω βιβλιοθήκης και έτσι μπόρεσα να εξάγω και να αποθηκεύσω σε αρχεία κειμένου τα δεδομένα που χρειάζονται από την εικόνα GeoTIFF [10] (γεωγραφικό μήκος, πλάτος και τιμές θερμοκρασιών). Ένα άλλο πρόβλημα παρουσιάστηκε στη τοποθέτηση των γραφημάτων πάνω στην επιφάνεια του Terrain μέσα στο περιβάλλον της μηχανής. Οι συντεταγμένες του Terrain και της εικόνας ήταν διαφορετικές, έτσι μέσω ενός μηχανισμού έπρεπε να βρεθεί τρόπος η τοποθέτηση των γραφημάτων που αντιπροσωπεύουν τους μέσους όρους των θερμοκρασιών να γίνει στη σωστή περιοχή. Για το σκοπό αυτό βοήθησε η βιβλιοθήκη (πρόγραμμα) GDAL [12] το οποίο δίνοντας του ως είσοδο το όνομα της GeoTIFF εικόνας, μας εξάγει με ακρίβεια τα 4 ακραία σημεία της με τη μορφή συντεταγμένων. Στη συνέχεια έγινε αντιστοίχιση των σημείων αυτών με τις συντεταγμένες της Unity 3D και μέσω δύο μεθόδων έγινε και η μετατροπή τους.

5.2 Γνώσεις που αποκόμισα

Οι κυριότερες και πιο σημαντικές γνώσεις που "κέρδισα" από την εκπόνηση της εργασίας μου ήταν πάνω στο κομμάτι των γραφικών σε υπολογιστή, αντικείμενο που ήθελα εξ αρχής να ασχοληθώ και να εμβαθύνω σε αυτό το κλάδο της επιστήμης της Πληροφορικής. Έμαθα να διαχειρίζομαι ένα σύνθετο περιβάλλον όπως αυτό της μηχανής γραφικών Unity 3D, η οποία προσφέρει μια πληθώρα από επιλογές για όλα τα επίπεδα προγραμματιστών, σε αυτό βέβαια βοήθησε και το online forum στο οποίο άλλοι πιο έμπυροι χρήστες δίνουν απαντήσεις σε όποιο πρόβλημα έχεις. Επίσης, έμαθα και πολλές πληροφορίες για τις εικόνες τύπου TIFF και ειδικότερα GeoTIFF τις οποίες διαχειρίστηκα, τι δεδομένα αποθηκεύουν στις ετικέτες τους, για ποιο λόγο χρησιμοποιούνται και

σε ποιους κλάδους της επιστήμης μας επιδρούν καταλυτικά έτσι ώστε να εξαχθούν πληροφορίες και δεδομένα από τους επιστήμονες για σημαντικά περιβαλλοντικά ζητήματα της σύγχρονης εποχής. Ανέπτυξα και τις γνώσεις πάνω στο κομμάτι του προγραμματισμού με τη δημιουργία κώδικα στη γλώσσα C#. Η γλώσσα C# βασίζεται στη δημιουργία κλάσεων, μεθόδων και αντικειμένων, στη διάρκεια των προπτυχιακών μου σπουδών ασχολήθηκα με αντικειμενοστραφείς γλώσσες όπως η Java και η Python, αλλά ποτέ σε συνδυασμό με βιβλιοθήκες οι οποίες προσφέρουν δημιουργία γραφικών στοιχείων όπως στη παρούσα φάση μέσω της Unity 3D και του Visual Studio. Τέλος, γνώρισα και εμπέδωσα από την αρχή τους όρους Terrain και Υψομετρικός Χάρτης (Height Map), έμαθα βασικές πληροφορίες για αυτούς, πως και για ποιους λόγους χρησιμοποιούνται στη πραγματική ζωή και πως μπορούν να συνδυαστούν με άλλους τομείς της Πληροφορικής αλλά και άλλων επιστημών που δεν σχετίζονται με τους υπολογιστές.

5.3 Γιατί επέλεξα τη Unity 3D ;

Ήθελα να ασχοληθώ από την αρχή των σπουδών μου με γραφικά υπολογιστών και τη δημιουργία ενός βίντεο παιχνιδιού λόγω της στενής μου "σχέσης" και της αγάπης μου για τα βίντεο παιχνίδια από παιδική ηλικία. Η επιλογή της Unity έγινε επειδή είναι μια αρκετά διαδεδομένη μηχανή γραφικών στις μέρες μας, με την οποία ο καθένας εύκολα και γρήγορα μπορεί να μάθει και να δημιουργήσει όμορφα περιβάλλοντα. Επίσης, παρέχει και δωρεάν έκδοση η οποία δε διαφέρει σχεδόν καθόλου από τις επαγγελματικές και ένα μεγάλο Forum-Community [7] όπου άλλοι χρήστες παρέχουν βοήθεια στις απορίες σου, άμεσα και γρήγορα. Τέλος, υπάρχει και το Asset Store όπου μπορείς εκεί να βρεις δωρεάν γραφικά μοντέλα ή ακόμη και scripting code με μια απλή αναζήτηση. Σίγουρα μελλοντικά θα ήθελα να συνεχίσω τη περαιτέρω εκμάθηση μου πάνω στη Unity ή ακόμη και σε πιο σύνθετες μηχανές γραφικών είτε σε μεταπτυχιακό επίπεδο είτε σε εργασιακό.

Βιβλιογραφία

[1] Carolin Helbig, Hans-Stefan Bauer, Karsten Rink, Volker Wulfmeyer, Michael Frank, Olaf Kolditz. *Concept and workflow for 3D visualization of atmospheric data in a virtual reality environment for analytical approaches*. November 2014, Volume 72, Issue 10, pp. 3767-3780.

Link: <https://link.springer.com/article/10.1007/s12665-014-3136-6>

[2] Martin Jucker. *Scientific Visualization of Atmospheric Data With ParaView*. Journal of Open Research Software (2014).

Link: <https://openresearchsoftware.metajnl.com/articles/10.5334/jors.al/>

[3] Marc Rautenhaus, Michael Bottinger, Stephan Siemen. *Visualization in Meteorology - A Survey of Techniques and Tools for Data Analysis Tasks*. December 2017, IEEE Computer Society.

Link: <https://ieeexplore.ieee.org/document/8126857/>

[4] Μηχανή γραφικών Unity 3D - Unity Game Engine (Wikipedia).

Link: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

[5] Μηχανή γραφικών Unity 3D - Unity Technologies (Wikipedia).

Link: https://en.wikipedia.org/wiki/Unity_Technologies

[6] Μηχανή γραφικών Unity 3D - Επίσημη σελίδα.

Link: <https://unity3d.com/>

[7] Μηχανή γραφικών Unity 3D- Επίσημο forum.

Link: <https://forum.unity.com/>

[8] Μηχανή γραφικών Unity 3D- Επίσημο store.

Link: <https://store.unity.com/>

[9] Εικόνες TIFF & GeoTIFF - Βασικές Πληροφορίες.

Link: <https://www.geospatialworld.net/article/geotiff-a-standard-image-file-format-for-gis-applications/>

[10] Εικόνες TIFF & GeoTIFF - Βασικός ορισμός (Wikipedia).

Link: <https://en.wikipedia.org/wiki/GeoTIFF>

[11] Εικόνες TIFF & GeoTIFF - Βιβλιοθήκη για διαχείριση TIFF μέσω C#.

Link: <https://bitmiracle.github.io/libtiff.net/>

- [12] Εικόνες TIFF & GeoTIFF - Βιβλιοθήκη GDAL.
Link: <https://www.gdal.org/>
- [13] Εικόνες TIFF & GeoTIFF - Λήψη εικόνων GeoTIFF.
Link: <https://earthexplorer.usgs.gov/>
- [14] Terrain - Βασικός ορισμός & πληροφορίες (Wikipedia).
Link: <https://en.wikipedia.org/wiki/Terrain>
- [15] Υψομετρικοί Χάρτες - Βασικός ορισμός & πληροφορίες (Wikipedia).
Link: <https://en.wikipedia.org/wiki/Heightmap>
- [16] Υψομετρικοί Χάρτες - Λήψη αρχείου για υψομετρικό χάρτη.
Link: <https://terrain.party/>
- [17] YouTube Κανάλι - Username: Unity Official Channel.
Link: <https://www.youtube.com/user/Unity3D>
- [18] YouTube Κανάλι - Username: Brackeys.
Link: <https://www.youtube.com/user/Brackeys>
- [19] YouTube Κανάλι - Username: Sloan Kelly.
Link: <https://www.youtube.com/user/sloankelly>
- [20] YouTube Κανάλι - Username: SpeedTutor.
Link: <https://www.youtube.com/user/SpeedTutor>
- [21] YouTube Κανάλι - Username: Charger Games.
Link: <https://www.youtube.com/channel/UChvLQbvpw2pfODgOmdS4ZSQ>
- [22] YouTube Κανάλι - Username: HelloWorld Studios.
Link: https://www.youtube.com/channel/UCD_Q6mBgigza-s2O3AqBlFw
- [23] YouTube Κανάλι - Username: USGS.
Link: <https://www.youtube.com/user/usgs>
- [24] YouTube Κανάλι - Username: Stuart Spence.
Link: <https://www.youtube.com/channel/UCb0DBpjN2lZhJCflvMUCIpw>
- [25] YouTube Κανάλι - Username: Muaaz.
Link: <https://www.youtube.com/channel/UCOCBH9Xcdf9vN6gpWxM0POA>

- [26] Frame rate - Βασικός ορισμός & πληροφορίες (Wikipedia).
Link: https://en.wikipedia.org/wiki/Frame_rate
- [27] Frames per second - Βασικός ορισμός & πληροφορίες.
Link: <https://www.techopedia.com/definition/7297/frames-per-second-fps>
- [28] OBS Studio - Πρόγραμμα καταγραφής βίντεο.
Link: <https://obsproject.com/>
- [29] Εικόνες TIFF & GeoTIFF - Εξαγωγή γεωγραφικού μήκους, πλάτους.
Link: <http://build-failed.blogspot.com/2014/12/processing-geotiff-files-in-net-without.html>