



ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ



**8<sup>ο</sup> Εξάμηνο : «Γραφική με Υπολογιστές»**

Ακαδ. έτος 2021-22 | Ημ.Παράδοσης 10/4/2022

**1<sup>η</sup> Εργασία : «Πλήρωση Τριγώνων»**

του Σταύρου Βασίλειου Μπουλιόπουλου 9671

Διδάσκων θεωρίας : Αναστάσιος Ντελόπουλος

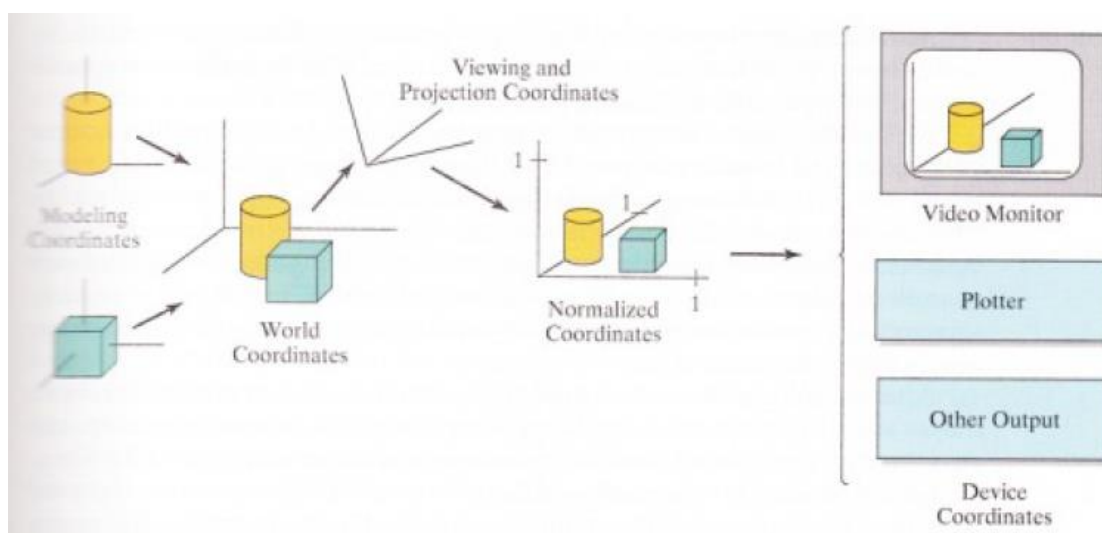
Επιβλέπων εργασίας : Αντώνιος Καρακώττας

## Περιεχόμενα

1.Εισαγωγή και ζητούμενα.....	2
2. Εκτέλεση των αρχείων κώδικα .....	2
3. Επεξήγηση των προβλημάτων που αντιμετωπίσα και της λειτουργικότητας της δομής του κώδικα.....	3
4. Αποτελέσματα των shading .....	6

## 1.Εισαγωγή και ζητούμενα

Η επιστήμη των γραφικών με την βοήθεια των υπολογιστικών συστημάτων έχει πολλές εφαρμογές σήμερα και κύριο στόχο έχει την οπτικοποίηση αντικειμένων σε έναν καμβά προβολής. Η ακολουθία των μετασχηματισμών μιας τρισδιάστατης σκηνής(3D) από συντεταγμένες μοντελοποίησης σε συντεταγμένες συσκευής(2D) απεικονίζεται στην παρακάτω φωτογραφία :



Εικόνα 1: 3D modeling coordinates to 2D image coordinates

Ζητούμενα στην προκειμένη εργασία ήταν η δημιουργία ενός αλγορίθμου(μετατροπή σάρωσης γραμμών,σκίαση με μεθόδους flat ή gouraud,χρωματισμό,ερμηνεία σε καμβά) με τον οποίο επιτυγχάνεται η **πλήρωση** κλειστών πολυγώνων και συγκεκριμένα **τριγώνων** με τελικό στόχο την απεικόνιση μιας εικόνας.

## 2. Εκτέλεση των αρχείων κώδικα

Η απεικόνιση της τελική εικόνας γίνεται χρησιμοποιώντας για γλώσσα προγραμματισμού Python(version=3.7) με δύο μεθόδους σκίασης flat ή gouraud. Κάθε μέθοδος έχει το αντίστοιχο αρχείο εκτέλεσης

demo\_flat.py ή demo\_gouraud.py, όπου το κάθε αρχείο έχει τα πακέτα βιβλιοθηκών που χρειάζεται και τις ζητούμενες συναρτήσεις.

### 3. Επεξήγηση των προβλημάτων που αντιμετωπίσα, παραδοχές και η δομή του κώδικα

Τα προβλήματα που ήρθα αντιμέτωπος ήταν η πρακτική αποτύπωση των αλγορίθμων στο προγραμματιστικό περιβάλλον της Python και ακολούθησα μια **«από πάνω προς τα κάτω προσέγγιση»**(rendering->shading->indexing->scan line conversion). Χρειάστηκα καλή κατανόηση του γενικού αλγορίθμου πλήρωσης πολυγώνων, να εκμεταλλευτώ τις ιδιότητες συνοχής που έχουν τα τρίγωνα ως προς την κυρτότητα και την φύση των τριών ακμών-κορυφών και να μελετήσω online διεξοδικότερα τις ειδικές περιπτώσεις του αλγορίθμου αλλά και τα πακέτα :

- **matplotlib** για να συνειδητοποιήσω την τελική μορφή που ήθελα να έχει η εικόνα για να ερμηνευθεί στον καμβά(rendering).
- **numpy** για την άντληση δεδομένων από το αρχείο hw1.npy και την διαχείριση των στοιχείων.

Η **δομή του κώδικα**(import packages,ορισμός υπολογιστικών συναρτήσεων,αναλυτικά σχόλια για αυτές,εκτέλεση συναρτήσεων) είναι η εξής :

-**Άντληση δεδομένων και μετατροπή** των numpy πινάκων σε λίστες για προσιότερη προσωπική ευκολία

-Κάλεσμα της συνάρτησης **render()** ,η οποία :

i) αρχικοποιεί τον καμβά σε άσπρο φόντο 512x512

ii) επαναληπτικά για κάθε τρίγωνο υπολογίζει το βάθος κάθε κορυφής και τριγώνου συνολικά

iii) ταξινομεί τα τρίγωνα με σειρά τέτοια ώστε πρώτο να είναι αυτό με το μικρότερο βάθος και τελευταίο αυτό με το μεγαλύτερο

iv) επαναληπτικά για κάθε τρίγωνο διαλέγει τις κορυφές κάθε τριγώνου και τις συντεταγμένες αυτών από την RGB λίστα και καλεί την συνάρτηση **shade\_triangle()**

- η **shade\_triangle()** σκιάζει και χρωματίζει το κάθε τρίγωνο ξεχωριστά είτε με flat είτε με gouraud μέθοδο βάσει ορίσματος(η διαφορά στις μεθόδους είναι ο τρόπος επιλογής χρώματος για τον χρωματισμό των διαστημάτων πλήρωσης είτε με μέσο όρο είτε με κυρτή γραμμική παρεμβολή αντίστοιχα). Πιο συγκεκριμένα, υπολογίζονται τα σημεία του εκάστοτε τριγώνου, οι μέγιστες και οι ελάχιστες τιμές της κάθε ακμής του τριγώνου, οι κλίσεις των ακμών του τριγώνου αλλά και το χρώμα με το οποίο θα χρωματιστεί το κάθε σημείο του τριγώνου. Έστερα, ακολουθεί ο υπολογισμός των αρχικών ενεργών γραμμών και των αρχικών ενεργών σημείων του τριγώνου. Στη συνέχεια, υλοποιείται μια διαδικασία κατά την οποία ξεκινά η σάρωση από τις γραμμές σάρωσης. Η σάρωση ξεκινάει από τις ελάχιστες τιμές των τεταγμένων και καταλήγει στις μέγιστες. Έτσι, υπολογίζονται τα διαστήματα πλήρωσης που άκρα τους είναι τα αντίστοιχα ενεργά οριακά σημεία εκείνη την στιγμή. Τελικά,χρωματίζονται αυτά τα διαστήματα από το ένα οριακό σημείο ως το άλλο:

i) **επαναληπτικά** για κάθε ακμή του τριγώνου υπολογίζει τις μέγιστες τεταγμένες και τετμημένες αλλά και την κλίση της(Bresenham)

ii) τοποθετεί τις γραμμές σάρωσης από τη μικρότερη στη μεγαλύτερη

iii) υπολογίζει τα διαστήματα πλήρωσης

iv) **επαναληπτικά** σαρώνοντας την λίστα από τη μικρότερη τεταγμένη μέχρι τη μεγαλύτερη κάνει τα παρακάτω βήματα: βρίσκει τα ενεργά οριακά σημεία, τα ταξινομεί, θέτει την ελάχιστη τετμημένη ως το πρώτο ενεργό οριακό σημείο, θέτει την μέγιστη τετμημένη ως το δεύτερο ενεργό οριακό σημείο και τώρα ξεκινάει **επαναληπτικά** σαρώνοντας την λίστα από την μικρότερη τετμημένη μέχρι τη μεγαλύτερη να χρωματίζει τα εσωτερικά σημεία του διαστήματος πλήρωσης με flat ή gouraud διαβαθμίσεις, μετά από αυτές τις επαναλήψεις ως προς τις τετμημένες και πριν τελειώσει η επαναληπτική σάρωση των τεταγμένων ενημερώνω τις λίστες ενεργών σημείων και ακμών.

-η **interpolate\_color()** υπολογίζει το RGB χρώμα ενός σημείου με κυρτή γραμμική παρεμβολή μεταξύ δύο σημείων

## 4. Αποτελέσματα των shading

*demo\_flat.py -> shade\_flat.png* (ενιαίο χρωματισμό σημείων  
τριγώνου με μέσο όρο χρώματος κορυφών)



*Εικόνα 2: Flat Shading*

*demo\_gouraud.py* -> *shade\_gouraud.png* (ομαλή συνέχεια στον χρωματισμό σημείων παρεμβολικά)



*Εικόνα 3: Gouraud Shading*