Γραφική με Υπολογιστές

Εργασία #1: Πλήρωση Τριγώνων

Ονομα: ΤΖΟΥΛΙΟ ΤΖΕΛΙΛΑΙ

ΑΕΜ: 9662

Περιγραφή λειτουργίας και κλήσης των προγραμματων

Η δομή των δύο προγραμματων είναι αρκετά απλή. Για τους δυο διαφορετικούς τύπους χρωματισμού των τριγώνων *("gouraud" ή "flat")* υπάρχουν και δύο scripts σε python (*demo_gouraud.py ή demo_flat.py*) αντίστοιχα. Τα συγκεκριμένα scripts περιέχουν την αρχικοποίηση της εικόνας, την εξαγωγή των δεδομένων από το αρχείο *hw1.py* καθώς επίσης και την κατάλληλη επεξεργασία τους ώστε να είναι κατάλληλα ορίσματα για τις συνάρτηση shade_triangle. Το καθε ένα από αυτά τα scripts συνοδεύεται με ένα άλλο αρχείο (*gouraud.py ή flat.py*) αντίστοιχα όπου υλοποιούνται οι συνάρτήσεις interpolate_color και shade_triangle καθώς επίσης και άλλες βοηθητικές συναρτήσεις για την πλήρωση των τριγώνων.

Τέλος, τα δύο scripts αποθηκεύουν τις δύο εικόνες ("*guraud.jpg" ή "flat.jpg*") στο ίδιο αρχείο όπου βρίσκονται και τα ίδια.

Περιγραφή της διαδικασίας χρωματισμού των τριγώνων και παρουσιάση του ανστίστοιχου ψευδοκώδικα.

Η λογική της πλήρωσης των τριγώνων βασίσηκε στον αλγόριθμο που περιγράφεται στις σημειώσεις. Πιο συγκεκριμένα:

1. Ταξινομούμε τις τεταγμένες των κορυφών του τριγώνου (συντεταγμένη y) σε αύξουσα σειρά
2. Διακρίνουμε βάση αυτών τι είδους μορφή είναι το τρίγωνο, δηλαδή είναι ένα σημείο, μια οριζόντια, πλάγια η καθετη γραμμή, ένα τρίγωνο με άνω η κάτω οριζόντια πλευρά ή ένα τυχαίο τρίγωνο
3. Βάση αυτής της διακριτοποίησης “γεμίζουμε” με ελαφρώς διαφορετικό τρόπο το τρίγωνο.

Αλγόριθμος πλήρωσης οριζόντιας γραμμής:

```
// Δεδομένου ότι τα τρίγωνα έχουν ίδια τεταγμένη
// Αντίστοιχος κώδικας και για ίδια τεμημένη, δηλαδή κάθετη γραμμη
For i in range(xmin,xmax+1):
    Img[y][x] = color // flat  or interpolated
```

Αλγόριθμος πλήρωσης τριγώνου με άνω η κάτω οριζόντια πλευρά

/* Βρίσκουμε αρχικά τους δύο συντελεστές διεύθυνσης που δημιουργούνται, έστω m12 αυτό της πρώτης κορυφής με την δεύτερη και m13 αυτό της πρώτης κορυφή με την τρίτη, δηλαδή μιλάμε για ένα τρίγωνο με άνω πλευρά οριζόντια */

```
If x2 <  x3

x_left = x_right = x1 // Δηλαδή το πρώτο σημείο είναι εκείνο

img[y1][x1] = color // της κάτω κορυφής


for y in range(y1+1,y3+1): // Από το κάτω y εώς πάνω

    x_left = x_left + 1/m12 // αριστερό άκρο x

    x_right = x_right + 1/m13 // δεξί άκρο x

    for x in range(x_left, x_right): // από αριστερό εώς δεξί

        img[y][x] = color        // άκρο x, γέμισε.
```

// Παρόμοια λογική και για x3 < x2 καθώς επίσης και για την πλήρωση τριγώνου με κάτω οριζόντια πλευρά.

Αλγόριθμος πλήρωσης τυχαίου τριγώνου

/* Για την πλήρωση τυχαιου τριγώνου έχουμε ταξινομήσει τις τεταγμενες και πραγματοποιούμε την πλήρωση από το μικρότερο ymin στο μεσαίο y, και έπειτα από το μεσαίο y στο μεγιστο ymax */

/* Βρίσκουμε αντίστοιχα τους συντελεστές ευθυών των κορυφών m12,m13,m23 */

```
img[x1][y1] = color
temp_x1 = x1 // Αρχίζουμε από την κάτω κορυφή
temp_x2 = x2
for y in range(y1,y2):
        temp_x1 = temp_x1 + m12 // Βρίσκουμε τα δύο ενεργά σημεία
        temp_x2 = temp_x2 + m13
        xmin ,xmax  = sorted(temp_x1,temp_x2) // Ταξινομούμε τις τεμημένες
        for x in range(xmin,xmax)
                img[y][x] = color
// Το ίδο κάνουμε και για το υπόλοιπο μισό του τριγώνου.  Δηλαδή για y2->y3
```

Αλλαγές στους αλγορίθμους όταν η πλήρωση γίνεται με γραμμική παρεμβολή

/* Όταν σε κάθε περίπτωση έχουμε βρεί τα x_left και x_right τότε βρίσκουμε και το χρώμα που τους αναλογεί με γραμμική παρεμβολή ανάμεσα στα y τους */

```
/Για πραάδειγμα, εάν μιλάμε για ένα τυχαίο τρίγωνο με y1< y2,y3

for y in range(y1,y2)

    x_left = x_left + 1/m13 // αριστερό άκρο x

    x_right = x_right + 1/m12 // δεξί άκρο x

    left_color = interpolate_color(y1,y2,y,C1,C2)

    right color = interpolate_color(y1,y3,y,C1,C3)


    for x in range(x_left, x_right)

        img[y][x] = interpolate_color(x_left,x_right,x,left_color,right_color)
```

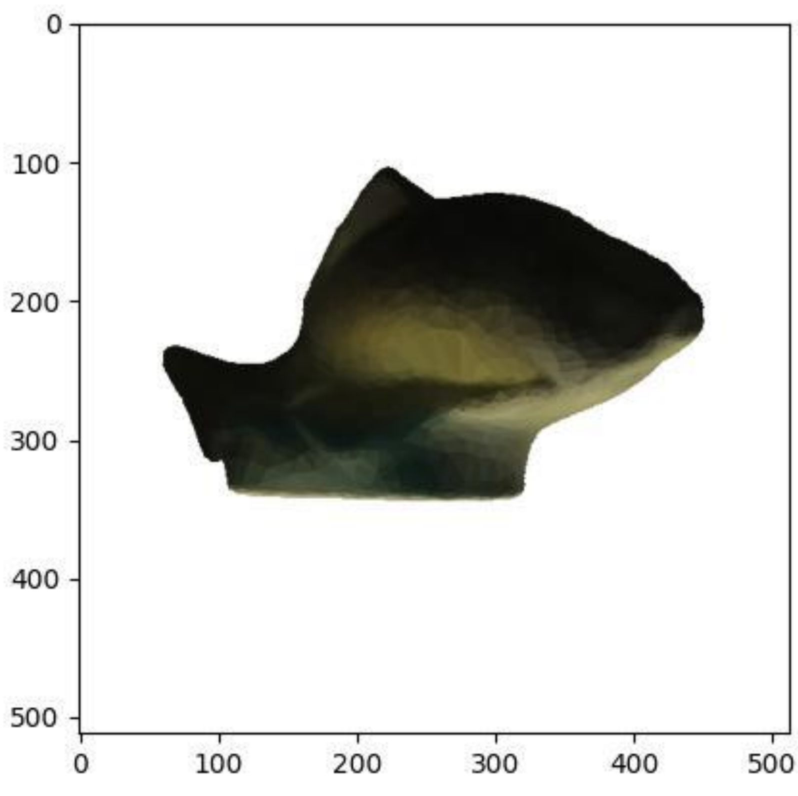
Παραδοχές που πραγματοποιήθηκαν

Στην εργασία έγιναν αρχικά αρκετές παραδοχές οι οποίες είτε προσαρμόστηκαν μετά από πειράματα είτε εγαταλήφθηκαν εντελώς στο τέλος. Οι πιο βασικές είναι οι παρακάτω 3:

1. Όταν δίνεται μόνο έαν σημείο τότε η γραμμική παρεμβολή στο χρώμα τους είναι ίδια με τον μέσο όρο των χρωμάτων τους είτε ζητείται “flat” είτε “gouraud”
2. Τα όρια των τετμημένων παίρνονταν πάντα ώς xleft = int(x_float), δηλαδή του όσο ακριανού γίνεται, xright = πλησιέστερο ακέραιο.
3. Ότι δεν έχουμε σοβαρά προβλήματα όταν $\infty > |m| > 1$.

Αποτελέσματα

FLAT



GOURAUD