



## Εργασία 1: Πλήρωση Τριγώνων

Παλάσκα Αιμιλία 10453 (aimiliapm@ece.auth.gr)

Επιβλέποντες: Αναστάσιος Ντελόπουλος, Αντώνης Καρακώπτας

### Λειτουργία και τρόπος κλήσης των προγραμμάτων

Τα προγράμματα μπορούν να κληθούν είτε από κάποιο IDE π.χ. VSCode είτε με την εντολή τερματικού `/bin/python3 demo_g.py`. Να σημειωθεί πως απαιτούνται μερικά λεπτά για να ολοκληρωθεί η αποθήκευση της τελικής εικόνας. Ο τρόπος που λειτουργούν τα scripts `demo_g.py` και `demo_f.py` παρουσιάζεται στον παρακάτω ψευδοκώδικα. Τέλος, ο κώδικας μπορεί να βρεθεί και στο [GitHub](#).

```
Load the 'hw1.npy' file
Export separate tables (faces, vertices, vcolors, depth)
Perform rendering with the corresponding arguments
Show results
Save results in corresponding file name
```

### Ψευδοκώδικας συναρτήσεων

- ❖ Γραμμική παρεμβολή διανυσμάτων:

```
vector_interp(p1, p2, V1, V2, coord, dim):
    If invalid arguments are given:
        Exit the algorithm
    Calculate the interpolation parameter t
    If dim = 1:
        t = (coord - x1) / (x2 - x1)
    Else if dim = 2:
        t = (coord - y1) / (y2 - y1)
    Else:
        Raise error("Invalid dimension. dim should be 1 or 2.")
    Return the interpolated value V = (1 - t) * V1 + t * V2
```

- ❖ Χάραξη γραμμών μέσω αλγορίθμου Bresenham:

```
bresenham_line(start, end):
    Calculate the needed variables (dx, dy, sx, sy, err)
    Initialize an empty pixels list
    Start from (x,y) = start
    While the ending point isn't reached:
        Add (x,y) to the pixels list
        Decide the next pixel based on the variables
    Return the pixels list
```



❖ Πλήρωση τριγώνου με τεχνική flat shading:

```
f_shading(img, vertices, vcolors):
    Calculate the flat_color as the mean of vcolors along the columns
    Calculate ymin and ymax from the vertices columns
    Initialize an active_edges empty list
    Fill the active_edges by executing bresenham_line on the
    combinations of vertices
    Initialize the updated_image by copying img
    For all the y lines in the scanning range (ymin, ymax):
        Create a current_edges list containing all the pixels with
        the current y from active_edges
        If current_edges has only one pixel (vertex):
            Color the pixel in the updated_img with the flat_color
            Continue to the next scanning line (y)
        Calculate xmin and xmax from the current_edges
        For all the pixels in the x scanning range (xmin, xmax):
            Color the pixel in updated_img with the flat_color
    Return the updated_img
```

❖ Πλήρωση τριγώνου με τεχνική Gouraud shading:

```
g_shading(img, vertices, vcolors):
    Calculate ymin and ymax from the vertices columns
    Initialize the updated_image by copying img
    Initialize an edges list
    Fill each row of edges by executing bresenham_line on the
    combinations of vertices
    For every row of edges:
        If the row is vertical:
            Color all the pixels with vector_intep on the y axis
        Else:
            Color all the pixels with vetor_interp on the x axis
            Color the vertices with the vcolors table
    Create an active_edges list as the concatenation of the edges rows
    For all the y lines in the scanning range (ymin, ymax):
        Create a current_edges list containing all the pixels with
        the current y from active_edges
        Skip the line if it has one pixel (vertex)
        Calculate xmin and xmax from the current_edges
        For all the pixels in the x scanning range (xmin, xmax):
            Color the(x, y) pixel with vector_interp between xmin
            and xmax
    Return the updated_img
```



❖ Αποτύπωση εικόνας:

```
render_img(faces, vertices, vcolors, depth, shading):
    Set canvas dimensions (M, N)
    Initialize the img with MxNx3 dimensions and value 0.99 (white)
    Calculate the depth of each triangle
    Sort the triangle depths and get the indexing
    Sort the triangles in faces using the triangle depth indexing
    If shading is 'f':
        For every triangle in faces:
            Get triangle vertices
            Get triangle colors
            Execute f_shading on img with these arguments
    Else if shading is 'g':
        For every triangle in faces:
            Get triangle vertices
            Get triangle colors
            Execute g_shading on img with these arguments
    Else:
        Raise error("Invalid shading. Value should be 'f' or 'g'.")
    Return img
```

### Παραδοχές που χρησιμοποιήθηκαν

- ❖ Για την λειτουργία των scripts είναι απαραίτητο όλα τα αρχεία να βρίσκονται στον ίδιο φάκελο, όπως επίσης και το δοσμένο αρχείο δεδομένων *hw1.py*.
- ❖ Όλες οι συναρτήσεις πραγματεύονται τρίγωνα, επομένως δεν εξετάζεται η περίπτωση κυρτών σχημάτων
- ❖ Στην συνάρτηση *vector\_interp* γίνεται η παραδοχή πως η συντεταγμένη της μεταβλητής *coord* ανήκει ήδη στην ευθεία που ορίζουν τα σημεία *p1* και *p2*. Ειδάλως, ο αλγόριθμος δεν παράγει κάποια χρήσιμη πληροφορία.
- ❖ Στις συναρτήσεις πλήρωσης τριγώνων γίνεται η παραδοχή πως τα σημεία τους ανήκουν στα περιθώρια του καμβά, άρα δεν εφαρμόζονται τεχνικές clipping πουθενά. Παρόμοια, αποκλείονται τριάδες συνευθειακών σημείων.

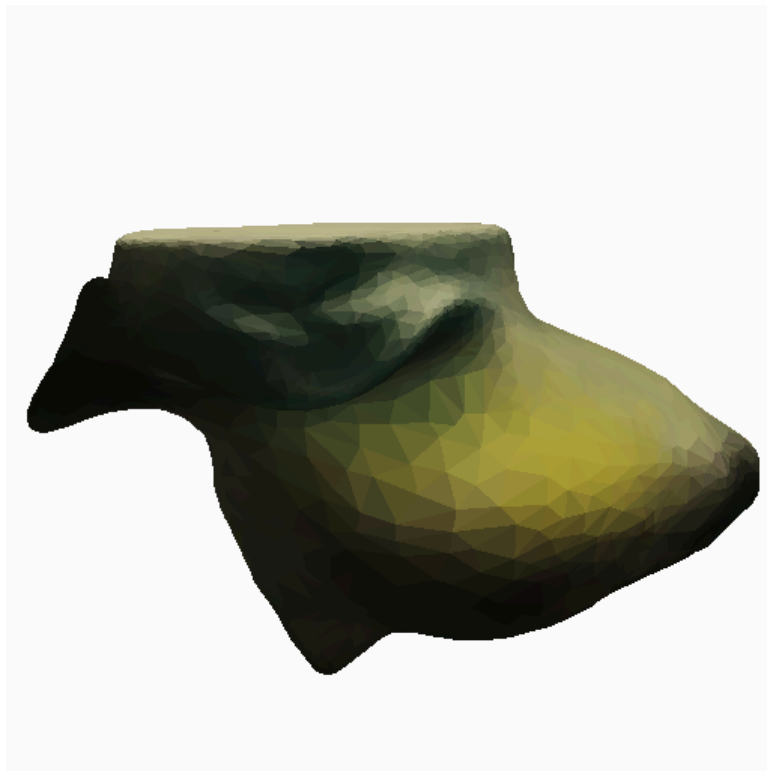
### Πηγές

- ❖ Γλωσσικό μοντέλο [ChatGPT 3.5](#)
- ❖ Επέκταση κειμενογράφου [Code Blocks](#)
- ❖ [Visual Studio Code](#) - Community Edition
- ❖ [Python](#) - Version 3.10.12



### Ενδεικτικά αποτελέσματα από τα demos

Αποτελέσματα τεχνικής flat shading:



Αποτελέσματα τεχνικής Gouraud shading:

