

## Αναφορά σχετικά με την απεικόνιση και τις μετασχηματισμούς σε 3D στην Python

### Αναφορά:

ergasia2.py: Περιέχει συναρτήσεις για την απεικόνιση και τους μετασχηματισμούς σε 3D, συμπεριλαμβανομένης της μετατροπής σημείων από τις παγκόσμιες συντεταγμένες σε συντεταγμένες κάμερας, της μετατροπής "look-at", της προοπτικής προβολής και της ραστεροποίησης, όπως και κωδικά της πρώτης εργασίας για την χρήση της `gourand`.

class1.py: Ορίζει μια κλάση για μετασχηματισμούς σε 3D, συμπεριλαμβανομένων των περιστροφών και των μεταφορών, σύμφωνα με τις σημειώσεις του μαθήματος.

demo.py: Χρησιμοποιεί τις συναρτήσεις από τα αρχεία `ergasia2.py` και `class1.py` για να απεικονίσει ένα αντικείμενο 3D με σκίαση (ένα ψάρι?).

---

Περιγραφή των Συναρτήσεων:

### **ergasia2.py**

`world2view(pts, R, c0)`: Μετατρέπει τα 3D σημεία από παγκόσμιες συντεταγμένες σε συντεταγμένες προβολής κάμερας.

`lookat(eye, up, target)`: Υπολογίζει τον πίνακα μετασχηματισμού της κάμερας με βάση τη θέση του ματιού, την κατεύθυνση πάνω και τη θέση του στόχου.

`perspective_project(pts, focal, R, t)`: Προβάλλει τα 3D σημεία στο επίπεδο της εικόνας χρησιμοποιώντας προοπτική προβολή.

`rasterize(pts_2d, plane_w, plane_h, res_w, res_h)`: Μετατρέπει τα προβεβλημένα 2D σημεία σε εικονοστοιχεία στο επίπεδο της εικόνας.

`render_object(v_pos, v_clr, t_pos_idx, plane_h, plane_w, res_h, res_w, focal, eye, up, target)`: Απεικονίζει ένα αντικείμενο 3D σε μια εικόνα δεδομένων παραμέτρων της κάμερας.

## **class1.py**

Transform: Ορίζει έναν τάξη για μετασχηματισμούς 3D.

init(): Αρχικοποιεί τον πίνακα μετασχηματισμού ως μοναδιαίο πίνακα.

rotate(theta, u): Περιστρέφει τον πίνακα μετασχηματισμού γύρω από έναν άξονα και ένα γωνία.

translate(t): Μετακινεί τον πίνακα μετασχηματισμού κατά ένα δεδομένο διάνυσμα.

transform\_pts(pts): Εφαρμόζει τον πίνακα μετασχηματισμού σε ένα σύνολο 3D σημείων.

## **Πώς να Καλέσετε τα Προγράμματα:**

-----

**ergasia2.py**: Εισαγάγετε το πρόσθετο και χρησιμοποιήστε τις συναρτήσεις απευθείας στον κώδικά σας. Για παράδειγμα:

**import ergasia2**

**# Call functions from ergasia2.py**

**result = ergasia2.world2view(pts, R, c0)**

-----

**class1.py:** Εισαγάγετε την κλάση Transform και δημιουργήστε μια νέα οντότητα για να εκτελέσετε μετασχηματισμούς. Για παράδειγμα:

```
-----  
  
from class1 import Transform  
  
# Create a transformation object  
  
transform = Transform()  
  
  
# Call methods to perform rotations and translations  
  
transform.rotate(theta, u)  
  
transform.translate(t)  
  
-----
```

**demo.py:** Εισαγάγετε πρόσθετα και συναρτήσεις από τα αρχεία ergasia2.py και class1.py για να απεικονίσετε ένα αντικείμενο 3D. Για παράδειγμα:

```
-----  
  
import ergasia2  
  
from class1 import Transform  
  
  
# Create a transformation object  
  
transform = Transform()  
  
  
# Perform transformations and render the object  
  
transformed_points = transform.transform_pts(v_pos)  
  
ergasia2.render_object(transformed_points, v_clr, t_pos_idx, plane_h, plane_w,  
res_h, res_w, focal, eye, up, target)
```

## Ενδεικτικά Αποτελέσματα που Παράγονται από τη Δοκιμή:

*ΔΥΣΤΥΧΩΣ ΔΕΝ ΚΑΤΕΦΕΡΑ ΝΑ ΠΑΡΑΞΩ ΕΙΚΟΝΑ  
ΛΟΓΩ ΣΦΑΛΜΑΤΟΣ ΣΤΟΝ ΚΩΔΙΚΑ ΤΟ ΟΠΟΙΟ ΜΕΤΑ  
ΑΠΟ ΠΟΛΥ ΕΡΕΥΝΑ ΣΤΟ ΙΝΤΕΡΝΕΤ ΚΑΙ ΡΩΤΩΝΤΑΣ  
ΓΝΩΣΤΟΥΣ, ΔΕΝ ΕΛΥΣΑ ΔΙΝΩ ΤΟ ΣΦΑΛΜΑ  
ΠΑΡΑΚΑΤΩ:*

Traceback (most recent call last):

```
File "c:\Users\George\Downloads\cg-hw2-2024 (1)\demo.py", line 38, in <module>
    render_object(transformed_points, v_clr, t_pos_idx, plane_h, plane_w, res_h, res_w, focal, eye, up,
target)
File "c:\Users\George\Downloads\cg-hw2-2024 (1)\ergasia2.py", line 81, in render_object
    transformed_vertices = world2view(v_pos, R, eye)
File "c:\Users\George\Downloads\cg-hw2-2024 (1)\ergasia2.py", line 10, in world2view
    Rinv=np.linalg.inv(R)
File
"C:\Users\George\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\
LocalCache\local-packages\Python310\site-packages\numpy\linalg\linalg.py", line 561, in inv
    ainv = _umath_linalg.inv(a, signature=signature, extobj=extobj)
File
"C:\Users\George\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\
LocalCache\local-packages\Python310\site-packages\numpy\linalg\linalg.py", line 112, in
_raise_linalgerror_singular
    raise LinAlgError("Singular matrix")
numpy.linalg.LinAlgError: Singular matrix
```