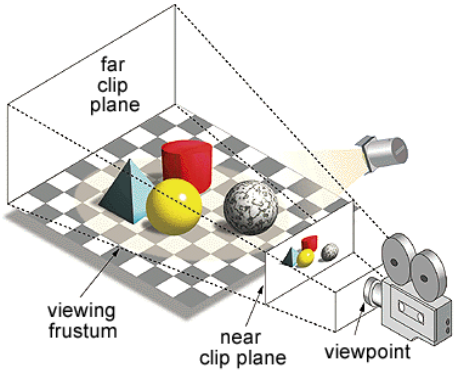
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ **ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ




# 8ο Εξάμηνο : **«Γραφική με Υπολογιστές»**

## Ακαδ. έτος 2021-22 |Ημ.Παράδοσης 25/5/2022

# **2η Εργασία : «Μετασχηματισμοί και προβολές»**

του Σταύρου Βασίλειου Μπουλιόπουλου 9671

Διδάσκων θεωρίας : Αναστάσιος Ντελόπουλος

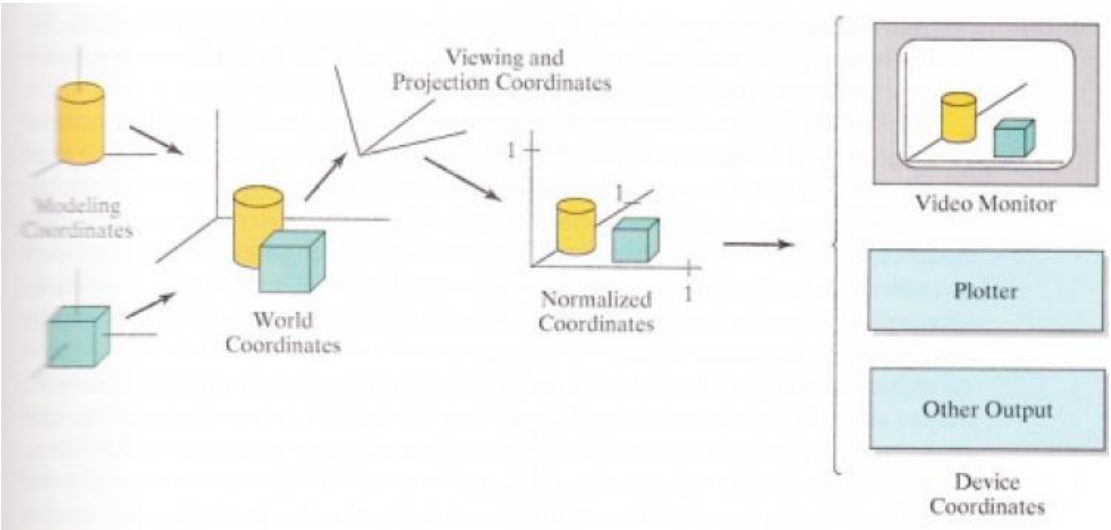
Επιβλέπων εργασίας : Αντώνιος Καρακώττας

## Περιεχόμενα

# 1.Εισαγωγή και ζητούμενα

Στην προηγούμενη εργασία στην οποία είχαμε δημιουργήσει αλγόριθμο πλήρωσης τριγώνων και σάρωσης γραμμών για την απεικόνιση σε εικόνα ενός δωσμένου συνόλου 2D σημείων με συγκεκριμένο χρώμα το κάθε ένα . Σε αυτή την εργασία σκοπός είναι η προβολή τρισδιάστατων σκηνικών στο πέτασμα της κάμερας και η αποτύπωση της εικόνας του σκηνικού στον καμβά του υπολογιστή,δηλαδή ένα προηγούμενο βήμα πριν μας δωθούν οι 2D συντεταγμένες προς απεικόνιση σε καμβά υπολογιστή. Προοπτική προβολή είναι η τέχνη της προβολής μιας τρισδιάστατης εικόνας και της δημιουργίας της αίσθησης του βάθους σε μια επίπεδη επιφάνεια.

3D WCS(WorldCoordinateSystem)->3D CCS(CameraCS)->2D CCS




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

Ζητήθηκαν συναρτήσεις με τις οποίες υπολογίζουμε πίνακες γραμμικού κι affine μετασχηματισμού και υλοποιούμε προβολή 3D αντικειμένου στην 2D προοπτική της κάμερας

συγκεκριμένων ορισμένων διαστάσεων,έτσι ώστε να συνεχίσουμε με την προβολή και τον χρωματισμό του αντικειμένου με Gouraud προσέχοντας να κάνουμε σωστό clipping.

## 2. Εκτέλεση των αρχείων κώδικα

Η απεικόνιση των αποτελεσμάτων(4 προβολές) και η αποθήκευση αυτών γίνεται χρησιμοποιώντας για γλώσσα προγραμματισμού Python(version=3.7) στο αρχείο εκτέλεσης **demo.py** ,το οποίο περιέχει τις απαιτούμενες βιβλιοθήκες και τις ζητούμενες συναρτήσεις.

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

Τα προβλήματα που ήρθα αντιμέτωπος ήταν η διεξοδικότερη ανάλυση γραμμικής άλγεβρας($5^o$,$6^o$ κεφάλαιο σημειώσεων) και πειραματισμός με την **numpy** για τους μετασχηματισμούς πινάκων(διάβασμα για rotation βάσει Rodriguez, για τον affine-translation μετασχηματισμό που δεν είναι γραμμικός), την κατανόηση των ομογενών συντεταγμένων και της χρησιμότητας αυτών για ευκολότερους-συμπτηγμένους μετασχηματισμούς και καλύτερη διαχώριση χώρου σημείων(points) και χώρου διανυσμάτων(vectors).

Οι **παραδοχές** που έκανα ήταν οι εξής:

-μικροαλλαγές συνάρτησης **render()** προηγούμενης εργασίας για να κάνει viewport clipping προσθέτοντας οριακές συνθήκες ελέγχου για να μην συμπεριληφθούν στην εικόνα σημεία εκτός καμβά π.χ. εκτός (0,imageHeight)x(0,imageWidth).

-uncomment line87(αν επιθυμήσουμε τεχνητή περιστροφή για να φαίνεται ίσιο το αντικείμενο διακοσμητικού ψαριού)

-τα ορίσματα εισόδου στις συναρτήσεις μετασχηματισμού δέχονται μαζικά τις 3D συντεταγμένες όλων των σημείων και όχι μόνο ένα-ένα τα σημεία.

Η **ροή του κώδικα** στο αρχείο είναι η εξής:

-import packages

-ορισμός υπολογιστικών συναρτήσεων με αναλυτικά σχόλια

-άντληση και επεξεργασία δεδομένων από **hw2.npy**

-κάλεσμα συναρτήσεων μετασχηματισμού αναλόγως αν θέλουμε μετατόπιση ή περιστροφή του 3D αντικειμένου

-κάλεσμα της **render_object()** για την προβολή του αντικειμένου και την δημιουργία 2D εικόνας

-απεικόνιση και αποθήκευση εικόνας

Η **λειτουργία των συναρτήσεων** είναι η εξής :

-η **affine_transform()** παίρνει ένα σύνολο από σημεία και τα μετατρέπει στις αντίστοιχες τους ομογενείς συντεταγμένες. Αυτές πολλαπλασιάζονται με τον πίνακα T του αντικειμένου και δίνουν τις μετασχηματισμένες ομογενείς συντατεγμένες,έτσι ώστε να αγνοηθεί η τελευταία γραμμή των νέων ομογενών συντεταγμένων και να επιστραφούν τα νέα σημεία.

-η **system_transform()** παίρνει τις 3D συντεταγμένες ενός σημείου βάσει του νέου συστήματος συντεταγμένων,το οποίο αποτελείται από διαφορετική αρχή και διαφορετικούς άξονες.Ο R είναι μορφής SO3 οπότε μπορώ να εκμεταλλευτώ για λιγότερες πράξεις ανάστροφος=αντεσταμμένος για την σχέση (5.28) των σημειώσεων.

-η **project_cam()** παίρνει την απόσταση **f** του πετάσματος από το κέντρο της κάμερας , τις 3D συντεταγμένες εκφρασμένες ως

προς το WCS των **διανυσμάτων x,y,z** ,τα οποία είναι τα μοναδιαία διανύσματα των συντεταγμένων της κάμερας, το **διάνυσμα v** ,το οποίο είναι η απόσταση των συντεταγμένων της κάμερας από την αρχή των αξόνων του WCS και τέλος παίρνει τις συντεταγμένες του σημείου **p** που θέλουμε να παράγουμε την προοπτική προβολή του ως προς την κάμερα. Ύστερα, κατασκευάζουμε πίνακα περιστροφής **R** βάσει των **x,y, z** για να αλλάξουμε τις συντεταγμένες του σημείου **p** στο νέο σύστημα συντεταγμένων και υπολογίζουμε στην συνέχεια το βάθος των σημείων πάνω πέτασμα βάσει της νέας z συνιστώσας του σημείου **p** και την προοπτική προβολή αυτών πάνω στο πέτασμα της κάμερας.

-η **project_cam_lookat()** παίρνει τις συντεταγμένες του σημείου προς το οποίο «κοιτάει» το κέντρο της κάμερας και το προς τα πάνω διάνυσμά της u,του οποίου η προβολή στο επίπεδο της κάμερας είναι παράλληλη προς το **y** της κάμερας ως προς το WCS. Σκοπός της συνάρτησης αυτής είναι να βρει τα διανύσματα του νέου συστήματος συντεταγμένων και να καλέσει την **project_cam()** βασιζόμενη στις σχέσεις (6.6,6.7,6.8).

-η **rasterize()** παίρνει τις διαστάσεις του πετάσματος της κάμερας και τις διαστάσεις του καμβά της εικόνας. Βρίσκουμε τον λόγο της κάθε φυσικής διάστασης προς την αντίστοιχη διάσταση της εικόνας,ο οποίος συμβολίζει τις φυσικές διαστάσεις ενός pixel. Γνωρίζοντας ότι το κέντρο της κάμερας θεωρείται το (0,0) τότε μετακινούμε το πέτασμα κατά **camW/2** προς τα θετικά x και κατά **camH/2** κατά τα θετικά y για να «ευθυγραμμίσουμε» τις δύο επιφάνειες και να πετύχουμε την απεικόνιση στις νέες διαστάσεις. Τέλος, διαιρούμε αυτές τις διαστάσεις x και y με τον λόγο που βρήκαμε πριν και στρογυλλοποιούμε το αποτέλεσμα.

-η **render_object()** υλοποιεί την τελική απεικόνιση του αντικειμένου βάσει των δωσμένων δεδομένων αυτού. Συγκεκριμένα, καλεί την συνάρτηση προοπτικής κάμερας με στόχο(**project_cam_lookat()**) ,στην συνέχεια την συνάρτηση απεικόνισης(**rasterize()**) και τέλος την συνάρτηση της 1ης εργασίας(**render()**) με χρωματισμό κατά Gouraud.

## 4. Αποτελέσματα των μετασχηματισμών και προβολών

*Μετασχηματισμοί κατά διαδοχή*

*Εικόνα 2: Αρχικό αντικείμενο μετά από προοπτική προβολή 0.jpg*

*Εικόνα 3: Αντικείμενο μετατοπισμένο κατά*

*t1= [0 0 -5000] (απομάκρυνση-μείωση κατά z) 1.jpg*

*Εικόνα 4: Αντικείμενο μετά από περιστροφή κατά γωνία φ=π rad προς άξονα u= [0 1 0] 2.jpg*

*Εικόνα 5: Αντικείμενο μετατοπισμένο κατά*

*t2= [0 500 -10000] (μετατόπιση κατά y και απομάκρυνση-μείωση κατά z) 3.jpg*