

Μετασχηματισμοί - Μοντέλα

Στην εργασία χρησιμοποιήθηκαν 3 αρχεία τύπου .obj για τον Ήλιο, τη Γη και τη Σελήνη αντίστοιχα. Για τον Ήλιο και τη Γη χρησιμοποιήσαμε αντικείμενα που βρήκαμε στο internet ενώ για τη Σελήνη χρησιμοποιήσαμε το planet.zip που δινόταν από το e-class.

Οι μετασχηματισμοί που χρησιμοποιήσαμε σε κάθε αντικείμενο είναι:

- Ήλιος

1. **Αλλαγή κλίμακας** για να μικρύνουμε το αντικείμενο κατάλληλα
2. **Μεταφορά** για να το φέρουμε στο κέντρο των αξόνων

- Γη

1. **Αλλαγή κλίμακας** για να μικρύνουμε το αντικείμενο κατάλληλα
2. **Περιστροφή** γύρω από τον άξονά του
3. **Μεταφορά** για να το φέρουμε στην κατάλληλη θέση / τροχιά γύρω από τον Ήλιο. Αυτό γίνεται αλλάζοντας τις συντεταγμένες x και z της Γης μέσα από τις τριγωνομετρικές συναρτήσεις $\sin()$ και $\cos()$ αντίστοιχα και πολλαπλασιάζοντάς τις με την απόσταση της Γης από τον ήλιο.

- Σελήνη

1. **Αλλαγή κλίμακας** για να μικρύνουμε το αντικείμενο κατάλληλα
2. **Μεταφορά** για να κάνουμε τη Σελήνη να περιστρέφεται γύρω από το κέντρο των αξόνων (όπως και με τη Γη)
3. **Μεταφορά** για να μεταφέρουμε τη περιστροφή της Σελήνης γύρω από το κέντρο των αξόνων, γύρω από το κέντρο της Γης, μέσα από τις τριγωνομετρικές συναρτήσεις $\sin()$ και $\cos()$.

Όλοι οι μετασχηματισμοί στον κώδικα, έχουν γραφτεί με την αντίθετη σειρά για κάθε αντικείμενο γιατί όπως ξέρουμε από την θεωρία, οι μετασχηματισμοί, που είναι στην ουσία πολλαπλασιασμοί πινάκων, γίνεται με την αντίθετη φορά από την οποία γράφονται. Έτσι πρώτα γίνονται οι μετασχηματισμοί αλλαγής κλίμακας, μετά οι περιστροφές και στο τέλος οι μεταφορές για να μην έχουμε αλλοίωση των αντικειμένων.

Φωτισμός

Για το φωτισμό έχουμε τοποθετήσει μία φωτινή πηγή στο κέντρο των συντεταγμένων (0,0,0) μέσα στον ήλιο και έτσι το φως διαχέεται παντού στη σκηνή μας. Για τον Ήλιο έχουμε δύο shaders, έναν vertex shader και έναν fragment shader, ώστε να μην επηρεάζεται το αντικείμενο

του Ήλιου από το φως. Για τη Γη και τη Σελήνη έχουμε 2 άλλους shaders, πάλι έναν vertex shader και έναν fragment shader, για να επηρεάζονται τα αντικείμενα από το φως. Στους shaders της Γης και της Σελήνης υπολογίζονται:

- Φωτισμός περιβάλλοντος (ambient lighting)
- Διάχυτος φωτισμός (diffuse lighting)
- Οπτικός φωτισμός (specular lighting)

και συνδυάζονται όλοι μέσα από το μοντέλο του *Phong*. Επίσης μέσα για το φωτισμό χρησιμοποιούμε τη τεχνική του **σημείου φωτός** (point light), καθώς ένα σημείο φωτός είναι μια πηγή φωτός με μια δεδομένη θέση κάπου στη σκηνή μας (0,0,0), που φωτίζει προς όλες τις κατευθύνσεις όπου οι ακτίνες φωτός ξεθωριάζουν με την απόσταση, όπως γίνεται και με τις ακτίνες του Ήλιου ή μιας απλής λάμπας στην πραγματικότητα. Όλες οι παράμετροι για το μοντέλο του Phong ή για το point light περνάνε στον fragment shader μέσα από τον κώδικά μας και μπορούν εύκολα να αλλάζουν.

Κάμερα

Για την κάμερα έχουμε χρησιμοποιήσει τη βοηθητική κλάση Camera από το tutorial της OpenGL και κάνουμε αντίστοιχους τους μετασχηματισμούς για να την τοποθετήσουμε λίγο πιο μακριά από τη σκηνή μας και να κοιτάμε τα αντικείμενά μας με προοπτική ώστε να μην παραμορφώνονται και έχουμε περίεργα αποτελέσματα.

Αλληλεπίδραση με το χρήστη

Για την αλληλεπίδραση με το χρήστη έχουμε τις εξής λειτουργίες:

- **Άξονας Y:** με τα πλήκτρα **W** και **S** ο χρήστης μπορεί να κινήσει την κάμερα του πάνω και κάτω στον άξονα Y.
- **Άξονας X:** με τα πλήκτρα **A** και **D** ο χρήστης μπορεί να κινήσει την κάμερα του αριστερά και δεξιά στον άξονα X.
- **Άξονας Z:** με τα πλήκτρα **↑** και **↓** ή με το ροδέλα του ποντικιού, ο χρήστης μπορεί να κινήσει την κάμερα του μπροστά και πίσω στον άξονα Z.
- **Παύση / Εκκίνηση** με το πλήκτρο **Space** ο χρήστης μπορεί να σταματήσει τη περιστροφή της Γης και της Σελήνης, και ξαναπατώντας το να ξεκινήσει πάλι από εκεί που είχαν σταματήσει.

- **Τερματισμός** με το πλήκτρο **Escape** ο χρήστης μπορεί να τερματίσει την εφαρμογή.
- ο χρήστης μπορεί επίσης με το ποντίκι να αλλάξει την γωνία που κοιτάει η κάμερα.

Πλατφόρμα - Κώδικας

Η εργασία αναπτύχθηκε στο CLion σε περιβάλλον Linux με βοηθητικό κώδικα από το tutorial της OpenGL που είχε χρησιμοποιηθεί στα εργαστήρια. Πειράξαμε το CMakeLists.txt αρχείο ώστε να το προσαρμόσουμε στις ανάγκες της εργασίας. Μπορείτε αν θέλετε κάνετε build από την αρχή το project, ακολουθώντας τις οδηγίες από το repository του tutorial (<https://github.com/JoeyDeVries/LearnOpenGL>) ή να ανοίξετε το project με το CLion (ή κάποιο άλλο αντίστοιχο editor) και να το τρέξετε.