

Εργασία 2

στο μάθημα "Γραφικά με Υπολογιστές"

Καταληκτική ημερομηνία παράδοσης : TBA

Άσκηση 1

Για κάθε Αριθμό Μητρώου (AM) φοιτητή/τρια θα υπολογιστεί ένας αριθμός ο οποίος καθορίζει τις διαφορετικές αναθέσεις. Ο αριθμός αυτός προκύπτει από τον τύπο :

$$U = AM \bmod 6$$

Ανάλογα με το **U** που έχει ο κάθε φοιτητής/τρια θα πρέπει να σχεδιαστούν δύο διαφορετικά πολύγωνα όπως φαίνεται παρακάτω:

Για $U=0$ θα σχεδιαστούν τα πολύγωνα με ID : 0,2

Για $U=1$ θα σχεδιαστούν τα πολύγωνα με ID : 1,3

Για $U=2$ θα σχεδιαστούν τα πολύγωνα με ID : 2,4

Για $U=3$ θα σχεδιαστούν τα πολύγωνα με ID : 3,5

Για $U=4$ θα σχεδιαστούν τα πολύγωνα με ID : 4,0

Για $U=5$ θα σχεδιαστούν τα πολύγωνα με ID : 5,1

Αντιστοιχία αριθμών και τύπου πολυγώνου	
ID	Τύπος Πολυγώνου
0	Πεντάγωνο
1	Εξάγωνο
2	Επτάγωνο
3	Οκτάγωνο
4	Εννεάγωνο
5	Δεκάγωνο

Να δημιουργηθεί αρχείο κώδικα (.cpp) με shaders κατά το οποίο θα παράγεται ένα παράθυρο OpenGL στο οποίο θα εμφανίζονται τα εξής:

Τα δύο διαφορετικά πολύγωνα, με διαφορετικό αντικείμενο VBO (Vertex Buffer Object) για κάθε πολύγωνο, τοποθετημένα σε διαφορετικές θέσεις και με διαφορετικά χρώματα. Τα πολύγωνα δεν χρειάζεται να είναι κανονικά αλλά θα πρέπει να μοιάζουν με κανονικά. Να δημιουργηθούν **έξι (6) ImGui::SliderFloat** με όνοματα R, G, B και R1, G1, B1 τα οποία θα έχουν εύρος [0.0, 1.0] και θα καθορίζουν το χρώμα του κάθε σχήματος, ξεχωριστά για το κάθε χρωματικό κανάλι (για την τιμή μηδέν (0) θα πρέπει να είναι μαύρο και για την τιμή ένα (1) να έχει μέγιστη φωτεινότητα).

Επιπλέον, θα είναι δυνατή η εναλλαγή μεταξύ του "solid mode" και του "wireframe mode" με τη χρήση του πλήκτρου 'W'.

Υποσημείωση 1: Για να γίνεται η αλλαγή με κάθε πάτημα του πλήκτρου μόνο μία φορά και όχι συνεχόμενα, θα χρειαστεί να ορισθούν δύο boolean μεταβλητές, οι οποίες θα σχετίζονται με το state του πλήκτρου στο τρέχον και το προηγούμενο frame.

Υποσημείωση 2: Για την κατασκευή ενός πολυγώνου να χρησιμοποιηθεί το mode GL_POLYGON αντί για το GL_TRIANGLES. Το mode GL_POLYGON συνδέει όλες τις κορυφές με την σειρά όπως έχουν οριστεί στο πίνακα με τις κορυφές και επίσης συνδέει και την τελευταία με την πρώτη κορυφή.

Το αρχείο κώδικα της άσκησης 1 θα έχει όνομα "AM_11.cpp", όπου AM θα είναι ο αριθμός μητρώου σας. Ο vertex shader θα έχει όνομα "VertexShader_11.txt" και ο fragment shader θα έχει όνομα "FragmentShader_11.txt".

Άσκηση 2

Να δημιουργηθεί αρχείο κώδικα (.cpp) με shaders κατά το οποίο θα παράγεται ένα παράθυρο OpenGL στο οποίο θα εμφανίζονται τα εξής:

Ένα πολύγωνο (να επιλέξετε ένα από τα δύο πολύγωνα που έχουν παραχθεί στην άσκηση 1), το οποίο θα έχει τη δυνατότητα να κινείται στην οθόνη με το πάτημα πλήκτρων από τον χρήστη. Για την κίνηση προς τα πάνω (θετικός άξονας Y) θα χρησιμοποιείται το πλήκτρο "W", για την κίνηση προς τα κάτω (αρνητικός άξονας Y) το πλήκτρο "S", για την κίνηση προς τα δεξιά (θετικός άξονας X) το πλήκτρο "D", και για την κίνηση προς τα αριστερά (αρνητικός άξονας X) το πλήκτρο "A". Το πολύγωνο θα μπορεί να τοποθετείται οπουδήποτε στην οθόνη, χωρίς να μετακινείται έξω από τα όριά της.

Το χρώμα (R,G,B) κάθε κορυφής του πολυγώνου (x, y, z), με $-1.0 \leq x, y, z \leq 1.0$ θα προκύπτει από την θέση της κορυφής, αλλά με την κατάλληλη κανονικοποίηση ώστε να τηρούνται οι επόμενες προϋποθέσεις:

- Η τιμή 'R' θα παίρνει τιμή 0.0 στις ακραίες αριστερά θέσεις της οθόνης για $x = -1.0$, τιμή 1.0 στις ακραίες δεξιά θέσεις της οθόνης για $x = 1.0$ και ενδιάμεσες τιμές σε όλες τις άλλες θέσεις.
- Η τιμή 'G' θα παίρνει τιμή 0.0 στις ακραίες κάτω θέσεις της οθόνης για $y = -1.0$, τιμή 1.0 στις ακραίες πάνω θέσεις της οθόνης για $y = 1.0$ και ενδιάμεσες τιμές σε όλες τις άλλες θέσεις.
- Η τιμή 'B' θα είναι πάντα μηδενική.

Το αρχείο κώδικα της άσκησης 2 θα έχει όνομα "AM_12.cpp", όπου AM θα είναι ο αριθμός μητρώου σας. Ο vertex shader θα έχει όνομα "VertexShader_12.txt" και ο fragment shader θα έχει όνομα "FragmentShader_12.txt".

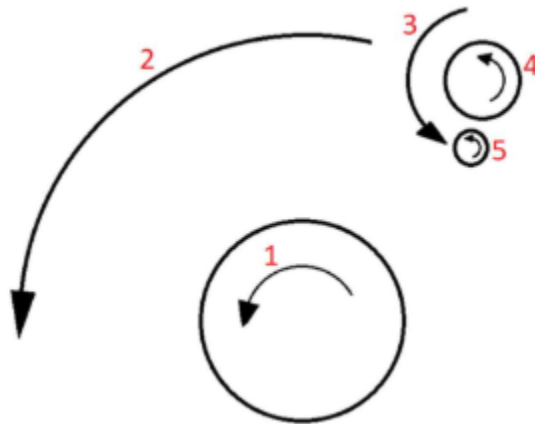
Άσκηση 3

Να δημιουργηθεί αρχείο κώδικα (.cpp) με shaders κατά το οποίο θα παράγεται ένα παράθυρο OpenGL στο οποίο θα εμφανίζονται τα εξής:

Τρεις κύβοι, οι οποίοι θα κινούνται στο χρόνο.

- Ο **πρώτος κύβος**, θα είναι στο κέντρο της οθόνης και θα περιστρέφεται γύρω από τον εαυτό του (περιστροφή 1).
- Ο **δεύτερος κύβος** θα περιστρέφεται γύρω από τον πρώτο (περιστροφή 2) και γύρω από τον εαυτό του (περιστροφή 4).
- Ο **τρίτος κύβος**, θα περιστρέφεται γύρω από τον πρώτο κύβο (περιστροφή 2), γύρω από τον δεύτερο (περιστροφή 3) και γύρω από τον εαυτό του (περιστροφή 5).

Οι περιστροφές των κύβων φαίνονται στο παρακάτω σχήμα:



Οι περιστροφές 2 και 3 να είναι γύρω από τον άξονα (1, 1, 0). Οι περιστροφές 1, 4 και 5 να είναι γύρω από διαφορετικούς άξονες. Να χρησιμοποιηθεί το μοντέλο του κύβου με διαφορετικό χρώμα σε κάθε πλευρά, το οποίο έχουμε χρησιμοποιήσει και στο εργαστήριο.

Το μέγεθος του κάθε κύβου θα καθορίζεται από τα 3 τελευταία ψηφία του AM. Το τρίτο ψηφίο θα καθορίζει το μέγεθος του πρώτου κύβου, το τέταρτο του δεύτερου κύβου και το πέμπτο του τρίτου κύβου. Τα ψηφία αυτά θα αντιπροσωπεύουν το ποσοστό αποκλιμάκωσης για όλους τους άξονες για κάθε διαφορετικό κύβο. Για παράδειγμα για $AM = 56049$, στον πρώτο κύβο θα εφαρμοστεί αποκλιμάκωση κατά 1.0 σε όλους τους άξονες, στον δεύτερο κύβο κατά 1.4 σε όλους τους άξονες και στον τρίτο κύβο κατά 1.9 σε όλους του άξονες. Η τιμή z θα πρέπει να είναι ίδια για όλους τους κύβους ώστε να φαίνεται η διαφορά μεγέθους.

Να οριστούν 2 **ImGui::SliderFloat** με ονόματα 'radius12' και 'radius23' τα οποία θα καθορίζουν την απόσταση του κύβου 1 από τον κύβο 2 και του κύβου 2 από τον κύβο 3 αντίστοιχα. Επίσης να οριστούν πέντε **ImGui::Checkbox** με ονόματα 'rotation1', 'rotation2', ... , 'rotation5' τα οποία θα καθορίζουν το αν θα συμβαίνουν οι περιστροφές 1, 2, 3, 4, 5 αντίστοιχα. Όταν μία περιστροφή σταματήσει και στην συνέχεια ξανά ενεργοποιηθεί θα πρέπει να συνεχίζει από το σημείο που είχε σταματήσει. Τέλος να οριστούν 5 **ImGui::SliderFloat** με ονόματα 'speed1', 'speed2', ... , 'speed5', τα οποία θα καθορίζουν την ταχύτητα των περιστροφών 1, 2, 3, 4, 5 αντίστοιχα (ελάχιστη τιμή 1 έτσι ώστε να μην σταματάει η περιστροφή από τον slider). Το εύρος του κάθε slider θα καθοριστεί από εσάς με λογικές τιμές που θα ορίσετε εσείς. Αν θέλετε μπορείτε να χρησιμοποιήσετε την εντολή **ImGui::SameLine()** έτσι ώστε το checkbox 'rotation1' είναι στην ίδια γραμμή με τον slider 'speed1'. Σε κάθε περίπτωση η κάμερα θα πρέπει να έχει στο οπτικό της πεδίο και τους τρεις κύβους και οι τρεις κύβοι να μην επικαλύπτουν ο ένας τον άλλο.

Το αρχείο κώδικα της άσκησης 3 θα έχει όνομα "AM_13.cpp", όπου AM θα είναι ο αριθμός μητρώου σας. Ο vertex shader θα έχει όνομα "VertexShader_13.txt" και ο fragment shader θα έχει όνομα "FragmentShader_13.txt".

Οι shaders θα πρέπει να τοποθετηθούν σε ένα φάκελο με όνομα "Shaders_AM", όπου AM ο αριθμός μητρώου σας (οι shaders μπορούν να έχουν την μορφή '.glsl', γενικά να φροντίσετε όπως θα στείλετε τους shaders έτσι να διαβάζονται και στον κώδικα).

Φροντίστε ώστε τα ονόματα των path των shaders να έχουν την μορφή "res/Shaders_AM/VertexShader_13.txt" ώστε ο κώδικας να τρέχει απευθείας όταν φορτώσουμε το αρχείο κώδικα της κάθε άσκησης και το φάκελο "Shaders_AM" στο project που έχουμε δημιουργήσει.

Άσκηση 4

Για κάθε Αριθμό Μητρώου (AM) φοιτητή/τρια θα υπολογιστεί ένας αριθμός ο οποίος καθορίζει τις διαφορετικές αναθέσεις. Ο αριθμός αυτός προκύπτει από τον τύπο :

$$U = AM \bmod 5$$

Ανάλογα με το U που έχει ο κάθε φοιτητής/τρια θα πρέπει να σχεδιαστούν διαφορετικές φωτεινές πηγές όπως φαίνεται παρακάτω:

Για $U = 0$ θα σχεδιαστούν οι φωτεινές πηγές με ID : 0, 1, 2, 4, 6

Για $U = 1$ θα σχεδιαστούν οι φωτεινές πηγές με ID : 1, 2, 3, 5, 7

Για $U = 2$ θα σχεδιαστούν οι φωτεινές πηγές με ID : 2, 3, 0, 4, 7

Για $U = 3$ θα σχεδιαστούν οι φωτεινές πηγές με ID : 1, 2, 5, 6, 7

Για $U = 4$ θα σχεδιαστούν οι φωτεινές πηγές με ID : 0, 2, 4, 5, 6

Αντιστοιχία αριθμών και φωτεινών πηγών			
ID	Τύπος φωτεινής πηγής	Θέση	Κατεύθυνση (μόνο για κατευθυντικές)
0	σημειακή	(2.3f, -3.3f, -4.0f)	-
1	σημειακή	(-3.0f, 1.0f, -10.0f)	-
2	σημειακή	(0.0f, 0.0f, -3.0f)	-
3	σημειακή	(3.4f, 4.0f, -14.0f)	-
4	κατευθυντική	(0.0f, 10.0f, -10.0f)	(0.0f, -1.0f, 0.0f)
5	κατευθυντική	(0.0f, -10.0f, -10.0f)	(0.0f, 1.0f, 0.0f)
6	κατευθυντική	(-10.0f, 10.0f, -10.0f)	(1.0f, -1.0f, 0.0f)
7	κατευθυντική	(10.0f, 10.0f, -10.0f)	(-1.0f, -1.0f, 0.0f)

Να δημιουργηθεί αρχείο κώδικα (.cpp) με shaders κατά το οποίο θα παράγεται ένα παράθυρο OpenGL στο οποίο θα εμφανίζονται τα εξής:

Δέκα κύβοι, στις θέσεις που βρίσκονται στο αρχείο "Lab5.cpp" του πέμπτου εργαστηρίου. Ο κάθε κύβος θα πρέπει να αποτελείται από διαφορετικό υλικό. Για την παραμετροποίηση των υλικών να χρησιμοποιηθεί ο πίνακας που είναι διαθέσιμος στο σύνδεσμο [link](#).

- Οι κύβοι θα πρέπει να φωτίζονται από τις φωτεινές πηγές που αντιστοιχούν στην ανάθεση του/της καθενός/μίας.
- Κοινή ανάθεση για όλους/όλες** θα είναι η δημιουργία μίας σημειακής φωτεινής πηγής (επιπρόσθετα με τις υπόλοιπες φωτεινές πηγές) η οποία θα περιστρέφεται συνεχώς γύρω από τον άξονα Z με ακτίνα 5 μονάδες από το κέντρο του Συστήματος Συντεταγμένων (Σ.Σ.).

Επίσης μέσω ενός `ImGui::SliderFloat` με όνομα 'radius' η ακτίνα θα λαμβάνει τιμές από 2 – 10 μονάδες

- Για όλες τις **σημειακές φωτεινές πηγές**, θα υπάρχουν 5 (πέντε) επιλογές για την απόσταση που θα φωτίζουν οι οποίες είναι: 20, 32, 50, 65 και 100 μονάδες και θα καθορίζονται από (5) πέντε **ImGui::RadioButton** (ένα την φορά). Η επιλογή αυτής της απόστασης συνδέεται με τους συντελεστές εξασθένησης K_i όπως φαίνεται στον Πίνακα που υπάρχει στο σύνδεσμο <https://learnopengl.com/Lighting/Light-casters>, (Ενότητα: Point Lights).
- Η συνεισφορά όλων των σημειακών φωτεινών πηγών στον φωτισμό της σκηνής θα ενεργοποιείται και θα απενεργοποιείται από ένα **ImGui::Checkbox** με όνομα 'Point Lights'. Επίσης, **για κάθε** σημειακή φωτεινή πηγή θα πρέπει να υπάρχει ένα **ImGui::Checkbox** με όνομα 'Point Light ID' μέσω του οποίου θα ενεργοποιείται και θα απενεργοποιείται η κάθε σημειακή πηγή ξεχωριστά από τις υπόλοιπες. Για την σημειακή φωτεινή πηγή που αποτελεί κοινή ανάθεση να χρησιμοποιηθεί το όνομα 'Point Light Moving'.
- Η συνεισφορά όλων των κατευθυντικών φωτεινών πηγών στο φωτισμό της σκηνής θα ενεργοποιείται και θα απενεργοποιείται από ένα **ImGui::Checkbox** με όνομα 'Directional Lights'. Επίσης **για κάθε** κατευθυντική φωτεινή πηγή θα πρέπει να υπάρχει ένα **ImGui::Checkbox** με όνομα 'Directional Light ID' μέσω του οποίου θα ενεργοποιείται και θα απενεργοποιείται η κάθε κατευθυντική φωτεινή πηγή ξεχωριστά από τις υπόλοιπες.
- Όταν απενεργοποιείται μία φωτεινή πηγή θα πρέπει να ακυρώνεται η συνεισφορά στον φωτισμό όλων των συνιστωσών του μοντέλου phong (άρα αν είναι όλες απενεργοποιημένες το χρώμα των κύβων θα πρέπει να είναι μαύρο) και να μην φαίνεται οπτικά ο κύβος που την αντιπροσωπεύει στην σκηνή.
- Με πατημένο το πλήκτρο Left Ctrl να εμφανίζεται ο δείκτης του ποντικιού και η κάμερα να μην περιστρέφεται προκυμμένου να πραγματοποιείται η αλληλεπίδραση με την ImGui.

Σημείωση: Για πιο λεπτομερή εξήγηση των επιμέρους φωτεινών πηγών συμβουλευθείτε το σύνδεσμο <https://learnopengl.com/Lighting/Light-casters>, ενώ για το συνδυασμό τους συμβουλευθείτε το σύνδεσμο <https://learnopengl.com/Lighting/Multiple-lights>.

Να κάνετε τις κατάλληλες αλλαγές στην κλάση Camera του αρχείου "Camera.h" που χρησιμοποιείται στις Εργαστηριακές ασκήσεις, ώστε ο χρήστης να έχει την δυνατότητα κίνησης προς τα επάνω και προς τα κάτω **στους άξονες του Σ.Σ. της κάμερας**. Να εκτελείται κίνηση προς τα επάνω με το πλήκτρο 'E' και η κίνηση προς τα κάτω με το πλήκτρο 'Q'. Η ταχύτητα κίνησης της κάμερας να είναι η ίδια με την ταχύτητα στις υπόλοιπες κατευθύνσεις. Με το πλήκτρο 'Left Shift' πατημένο να αυξάνεται η ταχύτητα κίνησης της κάμερας και με την απελευθέρωση του ίδιου πλήκτρου να επανέρχεται στην κανονική ταχύτητα.

Το αρχείο κώδικα θα έχει όνομα "AM_21.cpp", όπου AM θα είναι ο αριθμός μητρώου σας. Ο vertex shader των κύβων εκτός του κύβου της φωτεινής πηγής θα έχει όνομα "VertexShader_21.txt" και ο fragment shader θα έχει όνομα "FragmentShader_21.txt". Ο κύβος της φωτεινής πηγής θα έχει vertex shader με όνομα "Vertex_light_21.txt" και fragment shader με όνομα "Fragment_light_21.txt". Το αρχείο της τροποποιημένης κλάσης Camera θα έχει όνομα "Camera_AM_21.h".

Οι shaders θα πρέπει να τοποθετηθούν σε ένα φάκελο με όνομα "Shaders_AM", όπου AM ο αριθμός μητρώου σας. Φροντίστε ώστε τα ονόματα των path των shaders στα αρχεία κώδικα να είναι έτσι ώστε το πρόγραμμα να τρέχει απευθείας όταν φορτώσουμε το αρχείο κώδικα της κάθε άσκησης και το φάκελο "Shaders_AM" στο project που έχουμε δημιουργήσει.

Τα αρχεία κώδικα και ο φάκελος των shaders να τοποθετηθούν σε έναν φάκελο με όνομα "Ergasia2_AM", ο οποίος θα συμπιεσθεί σε αρχείο (.rar, .zip) και θα υποβληθεί στο Eclass.