

## ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ 1-B

ΟΜΑΔΑ :

ΜΑΡΟΥΛΗΣ ΒΑΣΙΛΕΙΟΣ ΑΜ:4573

ΜΟΤΣΕΝΙΓΟΣ ΣΠΥΡΙΔΩΝ ΑΜ:4426

Για την δευτερη προγραμματιστική άσκηση έχουν παραδοθεί 5 αρχεία.

Υπάρχει ένα αρχείο τύπου Vertex Shader, το οποίο έχουμε ονομάσει **TransformVertexShader** και στο οποίο γίνονται οι μετασχηματισμοί των σημείων που δίνουμε σαν είσοδο στο βασικό αρχείο το main.cpp.

Έπειτα υπάρχει και ένα άλλο αρχείο τύπου fragment shader το οποίο ονομάζουμε **ColorFragmentShader** στο οποίο δίνουμε το χρώμα στα τρίγωνα που φτιάχνουν την πυραμίδα μας .

Τέλος, υπάρχει το αρχείο Main.cpp . Σε αυτό αρχικά εισάγουμε τις GLFW, GLEW και GLM. Στη συνέχεια υλοποιούμε τους δύο Shader, Vertex και Fragment και ελέγχουμε αν έχουν δημιουργηθεί σωστά.

Ύστερα τους συνδέουμε με το πρόγραμμα μας και κατόπιν δημιουργούμε την main του προγράμματός μας. Σε αυτήν αρχικά αρχικοποιούμε την GLFW με την εντολή init και έπειτα δημιουργούμε ένα παράθυρο 800 x 800 με την ονομασία «Πυραμίδα» με χρήση της εντολής glfwCreateWindow.

Στην συνέχεια αφού ελέγξαμε ότι όλα πήγαν καλά, αρχικοποιούμε την GLEW και βάζοντας τις κατάλληλες παραμέτρους στην `glClearColor` πετυχαίνουμε το μαύρο χρώμα που ζητείται για το background. Επίσης, αρχικοποιούμε την GLM.

Έπειτα, σε έναν πίνακα με την ονομασία **`g_vertex_buffer_data`** αποθηκεύουμε τις κατάλληλες συντεταγμένες των 6 τριγώνων ώστε να σχηματιστεί η πυραμίδα μας.

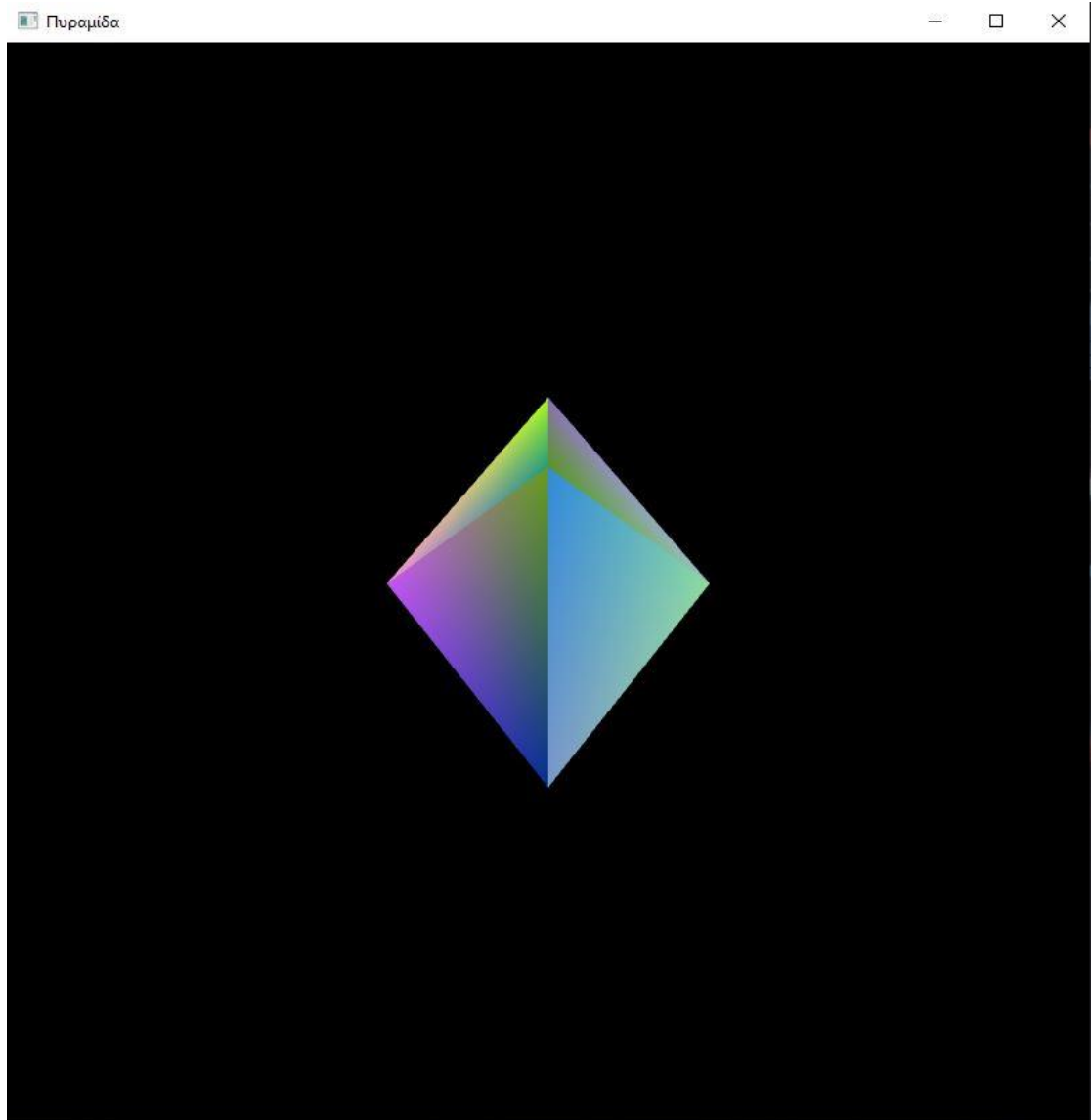
Οι συντεταγμένες υπολογίζονται με βάση το παγκόσμιο σύστημα συντεταγμένων με κέντρο το σημείο (0,0,0) .

Έχουμε ακόμη έναν πίνακα τον **`g_color_buffer_data`** οπού αποθηκεύουμε το χρώμα της κάθε κορυφής. Ο αριθμός έχει παραχθεί τυχαία από γεννήτρια τυχαίων αριθμών.

Στην συνέχεια με μία εντολή `do while` φτιάχνουμε τα 6 τρίγωνα ενώνοντας τις 18 κορυφές τους ανά 3 μεταξύ τους με την χρήση της εντολής `glDrawArrays` με παράμετρο την `GL_TRIANGLES` η οποία είναι προκαθορισμένη εντολή της OpenGL για την δημιουργία τριγώνων.

Έτσι δημιουργούμε την πυραμίδα που μας ζητείται η οποία παραμένει στο παράθυρο μέχρι να το κλείσουμε ή να πατήσουμε το πλήκτρο **Q** το οποίο θα κλείσει το παράθυρο όπως έχουμε προγραμματίσει στις συνθήκες του `while` , με την βοήθεια της συνάρτησης **`GetKeyState(VK_CAPITAL)`** της βιβλιοθήκης **`windows.h`**.

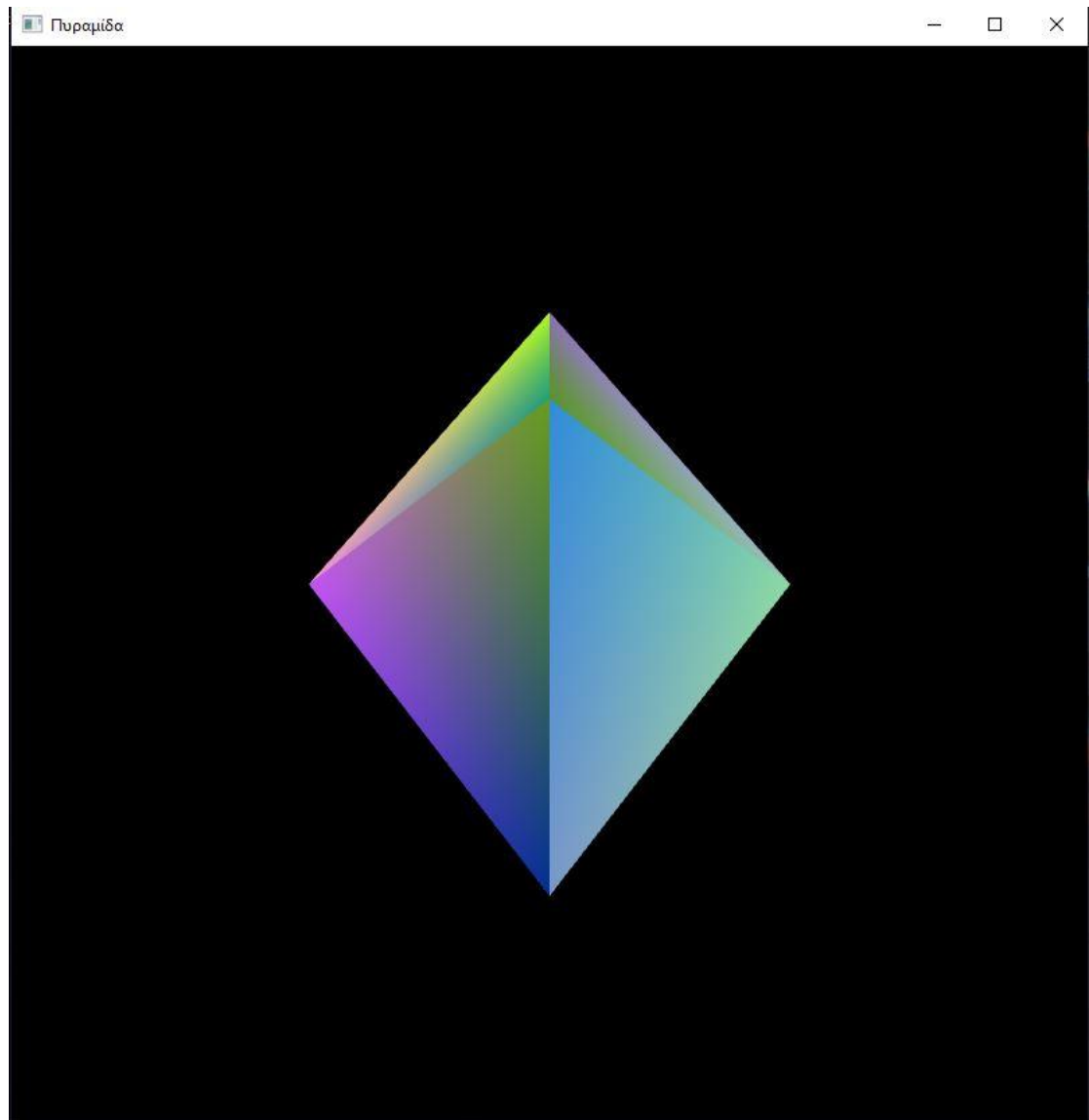
Το ύψος της πυραμίδας είναι τυχαίο τηρώντας όμως τους περιορισμούς , και έτσι κάθε φορά που ο χρήστης τρέχει το πρόγραμμα παίρνει μια νέα πυραμίδα.



Ενδεικτικά μια εικόνα της πυραμίδας.

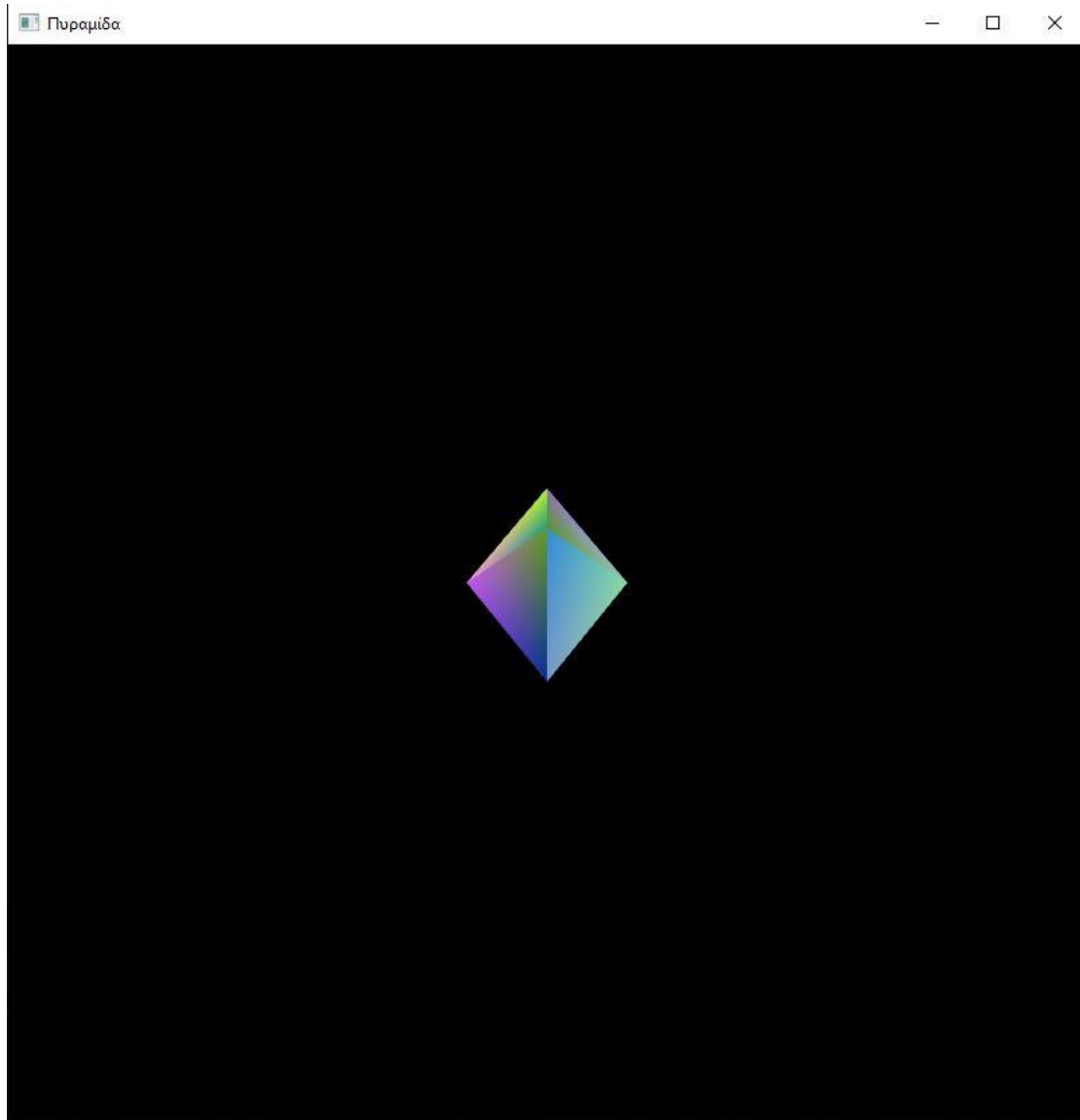
Η κάμερα όπως φαίνεται και στη παραπάνω εικόνα έχει τοποθετηθεί σύμφωνα με τις απαιτήσεις , στο σημείο  $(10.0, 10.0, 30.0)$  ώστε να κοιτάει προς το κέντρο της πυραμίδας με ανιόν διάνυσμα (up vector) το  $(0.0, 0.0, 1.0)$ .

Το πρόγραμμά μας έχει επίσης την δυνατότητα να κάνει upscaling την πυραμίδα πατώντας το πλήκτρο **q**. Το scaling της πυραμίδας μας δεν είναι διαδραστικό και έτσι η πυραμίδα μας μεγαλώνει μόνο όταν ο χρήστης έχει πατήσει το πλήκτρο **q** όταν το αφήσει η πυραμίδα γυρίζει στο κανονικό της μέγεθος.



Upscaling (q πατημένο)

Το πρόγραμμα μπορεί να κάνει και downscaling την πυραμίδα μας πατώντας το **z** . Και σε αυτή την περίπτωση το scaling δεν είναι διαδραστικό και έτσι η πυραμίδα γίνεται πιο μικρή όταν ο χρήστης κρατά πατημένο το **z** όταν το αφήσει η πυραμίδα γυρίζει στο κανονικό της μέγεθος.



Downscaling (z πατημένο)

Στην συνέχεια για το ερώτημα ν) έχουμε φτιάξει το αρχείο `controls.cpp` και το αρχείο `control.hpp` .Σύμφωνα με το παράδειγμα του εργαστήριου . Το αρχείο `controls.hpp` το κάνουμε `include` στην αρχή του `main.cpp`. Το `controls.cpp` περιέχει τον κώδικα για την λειτουργία της κάμερας μόνο με το πληκτρολόγιο.

Έτσι η κάμερα κινείται στους άξονες του παγκόσμιου συστήματος συντεταγμένων σύμφωνα με τα πλήκτρα που δοθήκαν στην εκφώνηση.

Η κάμερα συνεχώς θα κοιτά στο  $(0,0,0)$  δηλαδή στο κέντρο της πυραμίδας.

Η κάμερα αρχικά είναι στο σημείο  $(10,10,30)$ .

Όταν ο χρήστης πατά το πλήκτρο **w** η κάμερα κινείται στον άξονα **χ** και τείνει να δει την πυραμίδα από την πάνω όψη της. Με το πλήκτρο **x** η κάμερα μετακινείται με τέτοιο τρόπο ώστε να δει την κάτω όψη της πυραμίδας.

Πατώντας το πλήκτρο **d** , η κάμερα κινείται η προς τα δεξιά ενώ πατώντας το πλήκτρο **a** κινείται προς τα αριστερά.

Με το πλήκτρο **+** η κάμερα πλησιάζει το κέντρο της πυραμίδας δηλαδή το  $(0,0,0)$ . Ενώ με το πλήκτρο **-** απομακρύνεται από το κέντρο.

Δυστυχώς δεν έχουμε πέτυχει ομογενή περιστροφή και έτσι η κάμερα μας εκτός από της λειτουργίες που αναφέραμε από πάνω τείνει να απομακρύνεται από το κέντρο των αξόνων. Και έτσι δεν μπορούμε να κάνουμε πλήρη περιστροφή γύρο από την πυραμίδα αφού άλλοτε η κάμερα μας απομακρύνεται από την πυραμίδα άλλοτε πλησιάζει την πυραμίδα.

Επίσης αν η κάμερα πλησιάσει και **ΞΕΠΕΡΑΣΕΙ** το κέντρο της πυραμίδας η λειτουργία των **+** , **-** αντιστρέφεται και έτσι ο χρήστης πρέπει να πατήσει το **-** για να πλησιάσει το κέντρο της πυραμίδας.

Για την εύρεση των συντεταγμένων των κορυφών των τριγώνων της πυραμίδας φτιάξαμε μια πρόχειρη εικόνα στην «ζωγραφική» την οποία παραθέτουμε παραθέτουμε.

Φαίνονται οι συντεταγμένες των κορυφών της βάσης της πυραμίδας με μαύρα γράμματα.

Ενώ η κορυφή που μοιράζεται από τα 4 όρθια τρίγωνα είναι με κόκκινα γράμματα και οπου **χ** είναι το τυχαίο ύψος της πυραμίδας μας.

