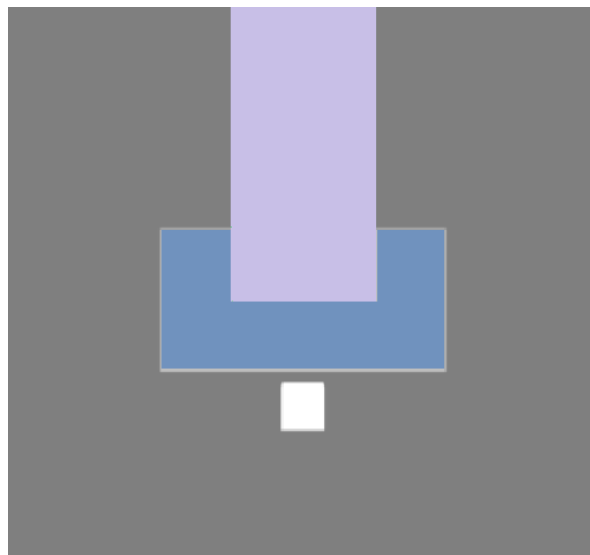




UNIVERSITY
of IOANNINA

ΕΡΓΑΣΙΑ 1B

Σκοπός της άσκησης είναι η δημιουργία ενός παραθύρου στο οποίο θα σχεδιάζουμε τρία 3D αντικείμενα και θα υλοποιήσουμε μια λειτουργία κάμερας.



1.Υλοποίηση άσκησης:

i.

Στην γραμμή 205 του κώδικα με την εντολή αυτή καθορίζονται τα χαρακτηριστικά του παραθύρου ,όπως ζητείται και από την άσκηση με διαστάσεις 950x950 και με τίτλο <<Εργασία 1B>>.

```
204 // Open a window and create its OpenGL context
205 window = glfwCreateWindow(950, 950, "Εργασία 1B", NULL, NULL);
206
```

Η εντολή στην γραμμή 229 παίρνει σαν ορίσματα 4 νούμερα, εκ των οποίων τα τρία πρώτα που μας ενδιαφέρουν είναι τα rgb. Συνεπώς για την απόχρωση του σκούρου γκρι background του παραθύρου δίνουμε τις τιμές 0.3f, 0.3f, 0.3f, 1.0f .

```
227
228 //grey background
229 glClearColor(0.3f, 0.3f, 0.3f, 1.0f);
```

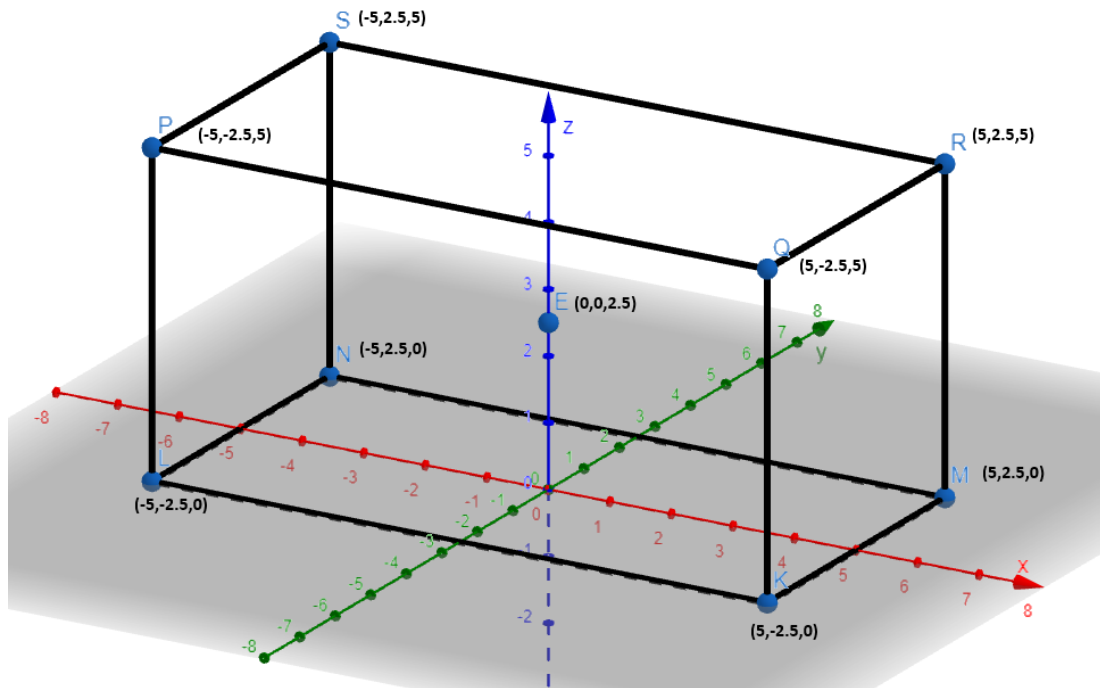
ii . Το πρόγραμμα αρχίζει με τον σχεδιασμό δύο ορθογώνιων παραλληλεπιπέδων (A και B) και ενός κύβου. Το πρώτο ορθογώνιο παραλληλεπίπεδο, A, τοποθετείται κατά μήκος του άξονα x και διαθέτει διαστάσεις με μήκος (k) ίσο με 10, ύψος (h) ίσο με 5, και πλάτος (w) ίσο με 5. Το κέντρο βάρους του A εντοπίζεται στο σημείο $E(0, 0, 2.5)$.

Το ορθογώνιο A "τέμνεται" από το δεύτερο ορθογώνιο παραλληλεπίπεδο, B, με διαστάσεις μήκους (m) ίσο με 5, ύψους (n) ίσο με 14, και πλάτος (v) ίσο με 5. Το B τοποθετείται κατά μήκος του άξονα y, ενώ η μία πλευρά του είναι επάνω στο επίπεδο xz με σημείο $O(0, 0, 0)$ ως κέντρο αυτής της πλευράς.

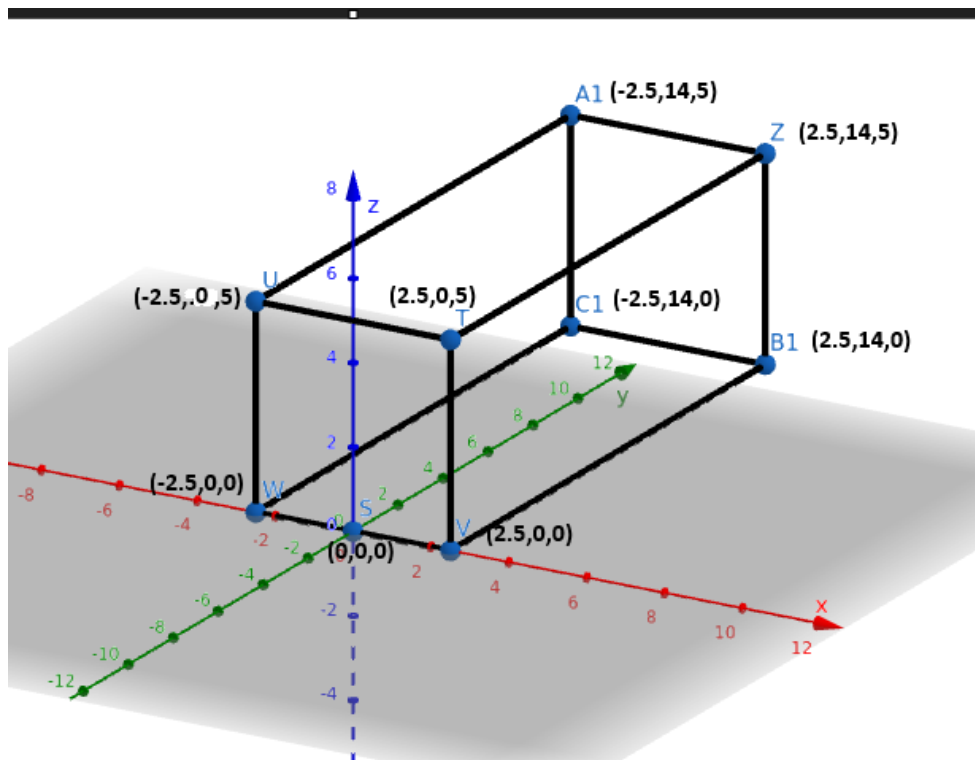
Για τον υπολογισμό των συντεταγμένων των σχημάτων A και B, εφαρμόσαμε μια μέθοδο που περιελάμβανε την αναπαράσταση των κορυφών τους σε ένα επίπεδο για ευκολότερη κατανόηση των σχημάτων και των συντεταγμένων. Επιλέξαμε το GeoGebra Online ως το κατάλληλο εργαλείο για την εισαγωγή και αναπαράσταση των συντεταγμένων των κορυφών, προκειμένου να επιβεβαιώσουμε την ορθότητα των διαστάσεων.

Αξιοποιήσαμε το Paint για να τοποθετήσουμε γραφικά τις συντεταγμένες των κορυφών, ενισχύοντας έτσι την οπτική αναπαράσταση των σχημάτων. Αυτή η διαδικασία λειτούργησε ως αξιόπιστη επιβεβαίωση των υπολογισμένων συντεταγμένων σε σχέση με τους άξονες X, Y, Z και τα κέντρα των σχημάτων.

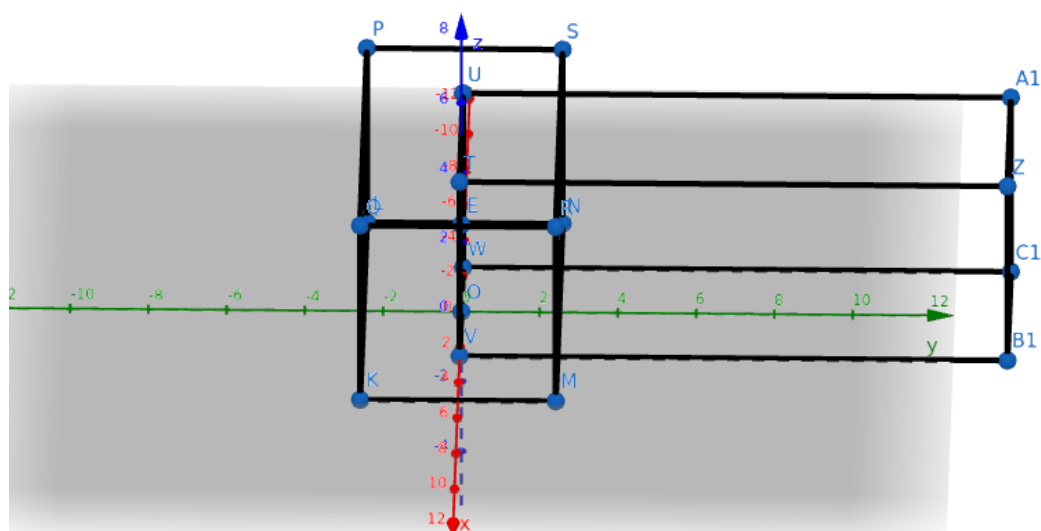
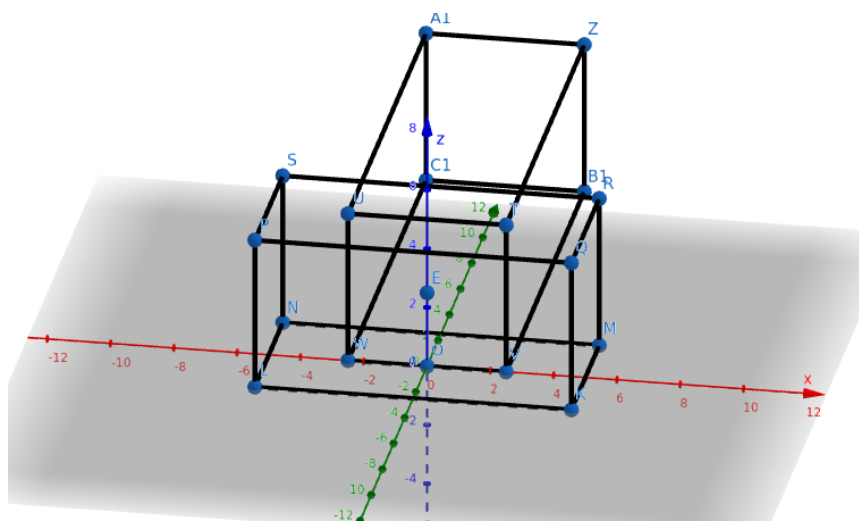
Α ορθογώνιο παραλληλεπίπεδο:



Β ορθογώνιο παραλληλεπίπεδο:



Κοινή απεικόνιση των σχημάτων Α και Β :



Ακολουθούν οι συντεταγμένες που τοποθετήθηκαν μέσα στον κώδικα μας :

```
009 static const GLfloat cube[] = {
010
011
012
013     -5.0f, -2.5f, 0.0f, //1st
014     -5.0f, -2.5f, 5.0f,
015     5.0f, -2.5f, 0.0f,
016     5.0f, -2.5f, 5.0f,
017     -5.0f, -2.5f, 5.0f,
018
019     -5.0f, -2.5f, 5.0f,
020     -5.0f, 2.5f, 5.0f,
021     5.0f, -2.5f, 5.0f,
022     5.0f, 2.5f, 5.0f,
023     -5.0f, 2.5f, 5.0f,
024
025     -5.0f, 2.5f, 0.0f,
026     -5.0f, 2.5f, 5.0f,
027     -5.0f, -2.5f, 0.0f,
028     -5.0f, -2.5f, 5.0f,
029     -5.0f, 2.5f, 5.0f,
030
031     5.0f, -2.5f, 0.0f,
032     5.0f, -2.5f, 5.0f,
033     5.0f, 2.5f, 0.0f,
034     5.0f, 2.5f, 5.0f,
035     5.0f, -2.5f, 5.0f,
036
037     -5.0f, -2.5f, 0.0f,
038     -5.0f, 2.5f, 0.0f,
039     5.0f, -2.5f, 0.0f,
040     5.0f, 2.5f, 0.0f,
041     -5.0f, 2.5f, 0.0f,
042
043     -5.0f, 2.5f, 0.0f,
044     -5.0f, 2.5f, 5.0f,
045     5.0f, 2.5f, 0.0f,
046     5.0f, 2.5f, 5.0f,
047     -5.0f, 2.5f, 5.0f,
048
049
050     -2.5f, 14.0f, 5.0f, //PANW
051     -2.5f, 14.0f, 0.0f,
052     2.5f, 14.0f, 5.0f,
053     2.5f, 14.0f, 0.0f,
054     -2.5f, 14.0f, 0.0f,
055
056     -2.5f, 0.0f, 5.0f,
057     -2.5f, 0.0f, 0.0f, //PLAINH dejia
058     -2.5f, 14.0f, 5.0f,
059     -2.5f, 14.0f, 0.0f,
060     -2.5f, 0.0f, 0.0f,
061
062     2.5f, 0.0f, 0.0f,
063     2.5f, 14.0f, 0.0f, //VB1WC1 mprostini
064     -2.5f, 0.0f, 0.0f,
065     -2.5f, 14.0f, 0.0f,
066     2.5f, 14.0f, 0.0f,
067
068     2.5f, 0.0f, 0.0f,
069     2.5f, 0.0f, 5.0f, //PLAHNH aristerh VTB1Z
070     2.5f, 14.0f, 0.0f,
071     2.5f, 14.0f, 5.0f,
072     2.5f, 0.0f, 5.0f,
073
074     -2.5f, 0.0f, 5.0f,
075     -2.5f, 14.0f, 5.0f, //UA1TZ, piw megalh
076     2.5f, 0.0f, 5.0f,
077     2.5f, 14.0f, 5.0f,
078     -2.5f, 14.0f, 5.0f,
079
080     -2.5f, 0.0f, 0.0f, //2nd
081     -2.5f, 0.0f, 5.0f,
082     2.5f, 0.0f, 0.0f,
083     2.5f, 0.0f, 5.0f, //katv WUVT
084     -2.5f, 0.0f, 5.0f,
085
086 };
```

Οι συντεταγμένες απεικονίζονται με τα παρακάτω DRAW :

```
567 // array buffer offset
568 );
569
570
571 glDrawArrays(GL_TRIANGLES, 0, 3);
572 glDrawArrays(GL_TRIANGLES, 2, 3 );
573
574 glDrawArrays(GL_TRIANGLES, 5, 3 );
575 glDrawArrays(GL_TRIANGLES, 7, 3 );
576
577 glDrawArrays(GL_TRIANGLES, 10, 3 );
578 glDrawArrays(GL_TRIANGLES, 12, 3 );
579
580 glDrawArrays(GL_TRIANGLES, 15, 3 );
581 glDrawArrays(GL_TRIANGLES, 17, 3 );
582
583 glDrawArrays(GL_TRIANGLES, 20, 3 );
584 glDrawArrays(GL_TRIANGLES, 22, 3 );
585
586 glDrawArrays(GL_TRIANGLES, 25, 3 );
587 glDrawArrays(GL_TRIANGLES, 27, 3 );
588
589 glDrawArrays(GL_TRIANGLES, 30, 3 );
590 glDrawArrays(GL_TRIANGLES, 32, 3 );
591
592 glDrawArrays(GL_TRIANGLES, 33, 3);
593 glDrawArrays(GL_TRIANGLES, 35, 3 );
594 glDrawArrays(GL_TRIANGLES, 37, 3 );
595 glDrawArrays(GL_TRIANGLES, 39, 3 );
596
597 glDrawArrays(GL_TRIANGLES, 42, 3 );
598 glDrawArrays(GL_TRIANGLES, 44, 3 );
599
600 glDrawArrays(GL_TRIANGLES, 47, 3 );
601 glDrawArrays(GL_TRIANGLES, 49, 3 );
602
603 glDrawArrays(GL_TRIANGLES, 52, 3 );
604 glDrawArrays(GL_TRIANGLES, 54, 3 );
605
606
```

- Καθώς Ο κύβος C είναι σχεδιασμένος με κέντρο βάρους το σημείο $P(0, -5, -4)$ και πλευρά μήκους $f=2$, αλλάξαμε τις συντεταγμένες του ως εξής:

```
258 static const GLfloat cubel[] = {
259     -2.0f, -7.0f, -6.0f,
260     -2.0f, -7.0f, -2.0f,
261     -2.0f, -3.0f, -2.0f,
262     -2.0f, -3.0f, -6.0f,
263     2.0f, -3.0f, -6.0f,
264     2.0f, -3.0f, -2.0f,
265     2.0f, -7.0f, -2.0f,
266     2.0f, -7.0f, -6.0f,
267     -2.0f, -3.0f, -6.0f,
268     -2.0f, -3.0f, -2.0f,
269     -2.0f, -7.0f, -2.0f,
270     -2.0f, -7.0f, -6.0f,
271     -2.0f, -7.0f, -6.0f,
272     -2.0f, -7.0f, -6.0f,
273     -2.0f, -3.0f, -6.0f,
274     -2.0f, -3.0f, -2.0f,
275     2.0f, -3.0f, -2.0f,
276     2.0f, -7.0f, -2.0f,
277     2.0f, -7.0f, -6.0f,
278     -2.0f, -7.0f, -6.0f,
279     -2.0f, -7.0f, -2.0f,
280     2.0f, -7.0f, -2.0f,
281     2.0f, -7.0f, -2.0f,
282     2.0f, -7.0f, -2.0f,
283     2.0f, -3.0f, -2.0f,
284     -2.0f, -3.0f, -2.0f,
285     -2.0f, -7.0f, -6.0f,
286     -2.0f, -3.0f, -6.0f,
287     2.0f, -3.0f, -6.0f,
288     2.0f, -3.0f, -6.0f,
289     2.0f, -7.0f, -6.0f,
290     -2.0f, -7.0f, -6.0f,
291     -2.0f, -3.0f, -2.0f,
292     -2.0f, -7.0f, -2.0f,
293     2.0f, -7.0f, -2.0f,
294     2.0f, -7.0f, -2.0f,
295     2.0f, -7.0f, -2.0f,
296     2.0f, -3.0f, -2.0f,
297     -2.0f, -3.0f, -2.0f
298 };
299
```

iii. Ακολουθούν τα χρώματα για το Α σχήμα:

```
393 GLfloat a=0.5f;
394 GLfloat b=1.0f;
395 // One color for each vertex
396 static const GLfloat color[] = {
397     0.609f, 0.115f, 0.436f, a,
398     0.609f, 0.115f, 0.436f, a,
399     0.609f, 0.115f, 0.436f, a,
400     0.609f, 0.115f, 0.436f, a,
401     0.609f, 0.115f, 0.436f, a,
402     0.609f, 0.115f, 0.436f, a,
403     0.014f, 0.184f, 0.576f, a,
404     0.014f, 0.184f, 0.576f, a,
405     0.014f, 0.184f, 0.576f, a,
406     0.014f, 0.184f, 0.576f, a,
407     0.014f, 0.184f, 0.576f, a,
408     0.014f, 0.184f, 0.576f, a,
409     0.771f, 0.328f, 0.970f, a,
410     0.771f, 0.328f, 0.970f, a,
411     0.771f, 0.328f, 0.970f, a,
412     0.771f, 0.328f, 0.970f, a,
413     0.771f, 0.328f, 0.970f, a,
414     0.771f, 0.328f, 0.970f, a,
415     0.971f, 0.572f, 0.833f, a,
416     0.971f, 0.572f, 0.833f, a,
417     0.971f, 0.572f, 0.833f, a,
418     0.971f, 0.572f, 0.833f, a,
419     0.971f, 0.572f, 0.833f, a,
420     0.971f, 0.572f, 0.833f, a,
421     0.279f, 0.317f, 0.505f, a,
422     0.279f, 0.317f, 0.505f, a,
423     0.279f, 0.317f, 0.505f, a,
424     0.279f, 0.317f, 0.505f, a,
425     0.279f, 0.317f, 0.505f, a,
426     0.279f, 0.317f, 0.505f, a,
427     0.783f, 0.290f, 0.734f, a,
428     0.783f, 0.290f, 0.734f, a,
429     0.783f, 0.290f, 0.734f, a,
430     0.783f, 0.290f, 0.734f, a,
431     0.783f, 0.290f, 0.734f, a,
432     0.783f, 0.290f, 0.734f, a,
433
```

Ακολουθούν τα χρώματα για το Β σχήμα:


```

0
1
2      0.583f, 0.771f, 0.014f,b, //2nd's colors
3      0.583f, 0.771f, 0.014f,b,
4      0.583f, 0.771f, 0.014f,b,
5      0.583f, 0.771f, 0.014f,b,
6      0.583f, 0.771f, 0.014f,b,
7
8      0.810f, 0.747f, 0.185f,b,
9      0.810f, 0.747f, 0.185f,b,
10     0.810f, 0.747f, 0.185f,b,
11     0.810f, 0.747f, 0.185f,b,
12     0.810f, 0.747f, 0.185f,b,
13
14     0.255f, 0.290f, 0.734f,b,
15     0.255f, 0.290f, 0.734f,b,
16     0.255f, 0.290f, 0.734f,b,
17     0.255f, 0.290f, 0.734f,b,
18     0.255f, 0.290f, 0.734f,b,
19
20     0.0f, 0.0f, 1.0f,b,
21     0.0f, 0.0f, 1.0f,b,
22     0.0f, 0.0f, 1.0f,b,
23     0.0f, 0.0f, 1.0f,b,
24     0.0f, 0.0f, 1.0f,b,
25     0.0f, 0.0f, 1.0f,b,
26
27     0.0f, 0.0f, 0.0f,b,
28     0.0f, 0.0f, 0.0f,b,
29     0.0f, 0.0f, 0.0f,b,
30     0.0f, 0.0f, 0.0f,b,
31     0.0f, 0.0f, 0.0f,b,
32
33     0.714f, 0.505f, 0.345f,b,
34     0.714f, 0.505f, 0.345f,b,
35     0.714f, 0.505f, 0.345f,b,
36     0.714f, 0.505f, 0.345f,b,
37     0.714f, 0.505f, 0.345f,b,
38
39     };
40

```

Ακολουθούν τα χρώματα για το C σχήμα:

```

40
41
42     };
43     GLfloat a=1.0f;
44     static const GLfloat color_cube[] = {
45
46         1.0f, 1.0f, 1.0f,a,
47         1.0f, 1.0f, 1.0f,a,
48         1.0f, 1.0f, 1.0f,a,
49         1.0f, 1.0f, 1.0f,a,
50         1.0f, 1.0f, 1.0f,a,
51
52         1.0f, 1.0f, 1.0f,a,
53         1.0f, 1.0f, 1.0f,a,
54         1.0f, 1.0f, 1.0f,a,
55         1.0f, 1.0f, 1.0f,a,
56         1.0f, 1.0f, 1.0f,a,
57
58         1.0f, 1.0f, 1.0f,a,
59         1.0f, 1.0f, 1.0f,a,
60         1.0f, 1.0f, 1.0f,a,
61         1.0f, 1.0f, 1.0f,a,
62         1.0f, 1.0f, 1.0f,a,
63
64         1.0f, 1.0f, 1.0f,a,
65         1.0f, 1.0f, 1.0f,a,
66         1.0f, 1.0f, 1.0f,a,
67         1.0f, 1.0f, 1.0f,a,
68         1.0f, 1.0f, 1.0f,a,
69
70         1.0f, 1.0f, 1.0f,a,
71         1.0f, 1.0f, 1.0f,a,
72         1.0f, 1.0f, 1.0f,a,
73         1.0f, 1.0f, 1.0f,a,
74         1.0f, 1.0f, 1.0f,a,
75
76         1.0f, 1.0f, 1.0f,a,
77         1.0f, 1.0f, 1.0f,a,
78         1.0f, 1.0f, 1.0f,a,
79         1.0f, 1.0f, 1.0f,a,
80         1.0f, 1.0f, 1.0f,a,
81
82         1.0f, 1.0f, 1.0f,a,
83         1.0f, 1.0f, 1.0f,a,
84         1.0f, 1.0f, 1.0f,a,
85         1.0f, 1.0f, 1.0f,a,
86         1.0f, 1.0f, 1.0f,a,
87
88         1.0f, 1.0f, 1.0f,a,
89         1.0f, 1.0f, 1.0f,a,
90         1.0f, 1.0f, 1.0f,a,
91         1.0f, 1.0f, 1.0f,a,
92         1.0f, 1.0f, 1.0f,a,
93
94         1.0f, 1.0f, 1.0f,a,
95         1.0f, 1.0f, 1.0f,a,
96         1.0f, 1.0f, 1.0f,a,
97         1.0f, 1.0f, 1.0f,a,
98         1.0f, 1.0f, 1.0f,a,
99
100        };
101

```

- για την εκτύπωση του κύβου δημιουργήσαμε δύο επιπλέον buffers:

```
523
524 GLuint vertexbuffer2;
525 glGenBuffers(1, &vertexbuffer2);
526 glBindBuffer(GL_ARRAY_BUFFER, vertexbuffer2);
527 glBufferData(GL_ARRAY_BUFFER, sizeof(cube1), cube1, GL_STATIC_DRAW);
528
529 GLuint colorbuffer2;
530 glGenBuffers(1, &colorbuffer2);
531 glBindBuffer(GL_ARRAY_BUFFER, colorbuffer2);
532 glBufferData(GL_ARRAY_BUFFER, sizeof(color_cube), color_cube, GL_STATIC_DRAW);
533
534
```

```
616
617 // 1st attribute buffer : vertices
618 glEnableVertexAttribArray(0);
619 glBindBuffer(GL_ARRAY_BUFFER, vertexbuffer2);
620 glVertexAttribPointer(
621     0, // attribute 0. No particular reason for 0, but must match
622     3, // size
623     GL_FLOAT, // type
624     GL_FALSE, // normalized?
625     0, // stride
626     (void*)0 // array buffer offset
627 );
628
629 // 2nd attribute buffer : colors
630 glEnableVertexAttribArray(1);
631 glBindBuffer(GL_ARRAY_BUFFER, colorbuffer2);
632 glVertexAttribPointer(
633     1, // attribute. No particular reason for 1, but
634     4, // size
635     GL_FLOAT, // type
636     GL_FALSE, // normalized?
637     0, // stride
638     (void*)0 // array buffer offset
639 );
640
641 glDrawArrays(GL_TRIANGLES, 0, 36);
642
```

- iv. Τοποθετήσαμε την κάμερα αρχικά στο σημείο $(0, 0, 30)$ ώστε να κοιτάει προς το σημείο $E(0, 0, 2.5)$ με ανιόν διάνυσμα (up vector) το $(0, 0, 1)$.

```
// Use our shader
glUseProgram(programID);

glm::mat4 Projection = glm::perspective(glm::radians(45.0f), 4.0f / 4.0f, 0.1f, 100.0f);
// Camera matrix
glm::mat4 View = glm::lookAt(
    glm::vec3(0.0f, 0.0f, 30.0f),
    glm::vec3(0.0f, 0.0f, 2.5f),
    glm::vec3(0.0f, 1.0f, 0.0f)
);
```

- Ως προς τον άξονα x, με τα πλήκτρα <u>και <p>
- Ως προς τον άξονα y, με τα πλήκτρα <i>και <o>
- Ως προς τον άξονα z, με τα πλήκτρα <j>και <k>

```
if (glfwGetKey(window, GLFW_KEY_U) == GLFW_PRESS) {  
    Model = glm::scale(Model, glm::vec3(1.05f, 1.0f, 1.0f));  
}  
  
if (glfwGetKey(window, GLFW_KEY_P) == GLFW_PRESS) {  
    Model = glm::scale(Model, glm::vec3(0.9f, 1.0f, 1.0f));  
}  
  
if (glfwGetKey(window, GLFW_KEY_I) == GLFW_PRESS) {  
    Model = glm::scale(Model, glm::vec3(1.0f, 1.05f, 1.0f));  
}  
  
if (glfwGetKey(window, GLFW_KEY_O) == GLFW_PRESS) {  
    Model = glm::scale(Model, glm::vec3(1.0f, 0.9f, 1.0f));  
}  
  
if (glfwGetKey(window, GLFW_KEY_J) == GLFW_PRESS) {  
    Model = glm::scale(Model, glm::vec3(1.0f, 1.0f, 1.05f));  
}  
  
if (glfwGetKey(window, GLFW_KEY_K) == GLFW_PRESS) {  
    Model = glm::scale(Model, glm::vec3(1.0f, 1.0f, 0.9f));  
}
```

vi .

Η κάμερα θα κινείται στους άξονες του παγκόσμιου συστήματος συντεταγμένων με τους εξής τρόπους:

-γύρω από τον άξονα x με τα πλήκτρα <w>και <x>

-γύρω από τον άξονα y με τα πλήκτρα <q> και <z>

-θα κάνει zoom in/zoom out με κατεύθυνση το σημείο O με τα πλήκτρα <+> και <->του numerical keypad του πληκτρολογίου

```
36 glm::mat4 Model;  
37 glm::mat4 Projection;  
38  
39 float zoom1=40;  
40 glm::mat4 ViewNEW;  
41  
42 glm::mat4 ViewMatrix;  
43 glm::mat4 ProjectionMatrix;  
44  
45 glm::mat4 getViewMatrix() {  
46     return ViewMatrix;  
47 }  
48 glm::mat4 getProjectionMatrix() {  
49     return ProjectionMatrix;  
50 }  
51  
52  
53  
54 void camera_function()  
55 {  
56     if (glfwGetKey(window, GLFW_KEY_W) == GLFW_PRESS){  
57         float rotationn = 0.01;  
58         rotationn+=0.01;  
59         Model = glm::rotate(Model,rotationn, glm::vec3(rotationn,0.0f,0.0f));  
60     }  
61     if (glfwGetKey(window, GLFW_KEY_X) == GLFW_PRESS){  
62         float rotationn = 0.1;  
63         rotationn-=0.15;  
64         Model = glm::rotate(Model,rotationn, glm::vec3(-rotationn,0.0f,0.0f));  
65     }  
66     if (glfwGetKey(window, GLFW_KEY_Q) == GLFW_PRESS){  
67         float rotationn2 = 0.01;  
68         rotationn2+=0.01;  
69         Model = glm::rotate(Model,rotationn2, glm::vec3(0.0f,0.0f,rotationn2));  
70     }  
71     if (glfwGetKey(window, GLFW_KEY_Z) == GLFW_PRESS){  
72         float rotationn2 =0.1;  
73         rotationn2-=0.15;  
74         Model = glm::rotate(Model,rotationn2, glm::vec3(0.0f,0.0f,-rotationn2));  
75     }  
76     if (glfwGetKey( window,GLFW_KEY_KP_ADD) == GLFW_PRESS){  
77         zoom1-=0.2;  
78     }  
79 }  
80  
81  
82  
83  
84  
85  
86  
87  
88
```

```
75 }  
76 if (glfwGetKey( window,GLFW_KEY_KP_ADD) == GLFW_PRESS){  
77     zoom1-=0.2;  
78 }  
79 if (glfwGetKey( window, GLFW_KEY_KP_SUBTRACT) == GLFW_PRESS){  
80     zoom1+=0.1;  
81 }  
82 ViewNEW = glm::lookAt(  
83     glm::vec3(0.0f,zoom1, 0.0f),  
84     glm::vec3(0.0f, 0.0f, 0.0f),  
85     glm::vec3(0.0f, 0.0f, 1.0f)  
86 );  
87 }  
88
```

2. Πληροφορίες σχετικά με την υλοποίηση:

Λειτουργικό Σύστημα: Linux

Text Editor : Sublime Text

3. Αξιολόγηση Ομάδας:

Η εργασία εκπονήθηκε από δύο άτομα, όπου ο καθένας έδινε ιδέες για την σχεδίαση της εφαρμογής. Η συνεργασία ήταν άψογη και παρουσιάστηκε συνέπεια ως προς την υλοποίηση της άσκησης.

4. Αναφορές – Πηγές που χρησιμοποιήθηκαν κατά την εκπόνηση της εργασίας.

<https://www.geogebra.org/>