

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ 1-Α

Γραφικά Υπολογιστών και Συστήματα Αλληλεπίδρασης - ΜΥΥ702

Ακαδ. Έτος 2021-2022

ΚΙΚΙΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ (4387)

ΤΣΑΜΠΙΡΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ (4508)

Λειτουργία του προγράμματος

Το πρόγραμμα είναι σε μεγάλο ποσοστό βασισμένο στο αρχείο που μας δόθηκε στο 1^ο εργαστήριο της OpenGL, όταν τρέξουμε το πρόγραμμα, ανοίγει ένα νέο παράθυρο διαστάσεων 800 x 800 με τίτλο «2Δ Σχήματα», όπου απεικονίζεται το σχήμα 1(α).

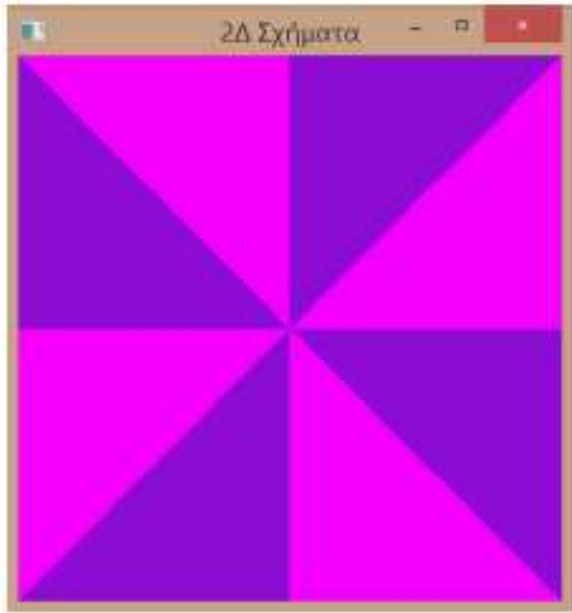


Figure 1: Το σχήμα της εκφώνησης

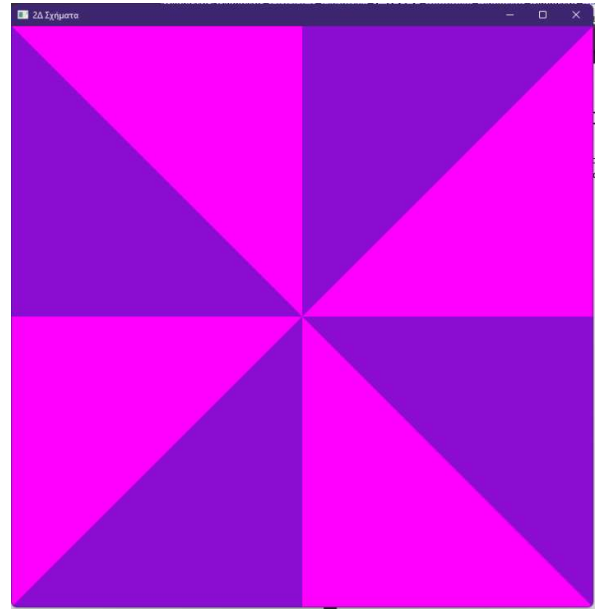


Figure 2: Το σχήμα που φτιάξαμε εμείς

Όταν πατάμε το πλήκτρο “c” τότε, απεικονίζεται το σχήμα 1(β)



Figure 3: Το σχήμα που θα πρέπει να έχουμε πατώντας C

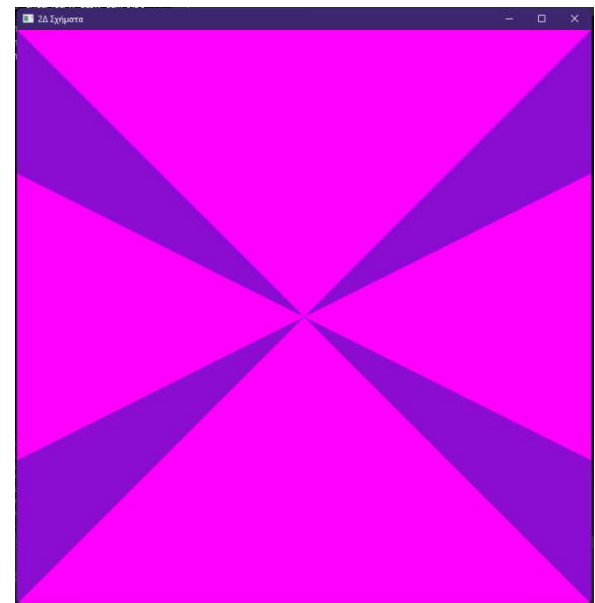


Figure 4: Το σχήμα που έχουμε πατώντας C

Όταν αφήσουμε το “c” θα γυρίσουμε πάλι στο σχήμα 1(α).

Μόλις πατήσουμε το πλήκτρο “n” (ή το x του παραθύρου), η λειτουργία του προγράμματος, τερματίζεται.

Ιδιαιτερότητες/Ειδικές συνθήκες

Αρχικά όταν δημιουργήσαμε το παράθυρο, αλλάζοντας τον τίτλο του σε “2Δ Σχήματα”, μας εμφάνιζε τους ελληνικούς χαρακτήρες, έτσι.



Η λύση σε αυτό το πρόβλημα ήταν η αποθήκευση του αρχείου με κωδικοποίηση Unicode (UTF-8)

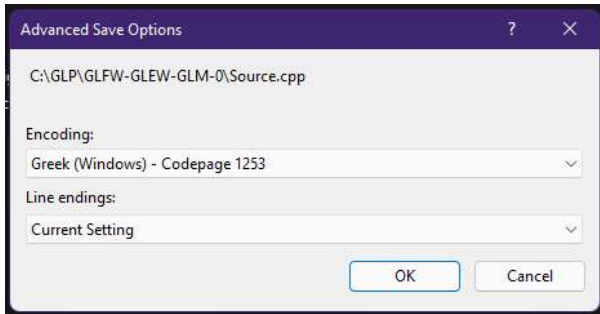


Figure 5: Αρχική ρύθμιση που προκαλούσε το πρόβλημα

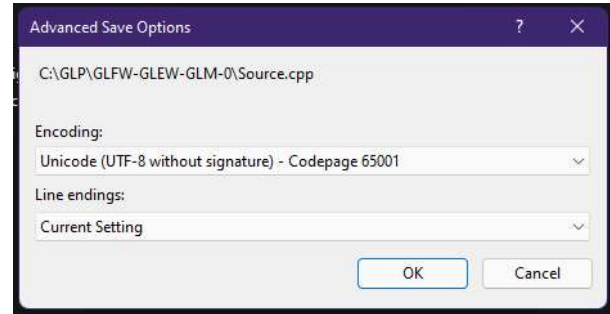
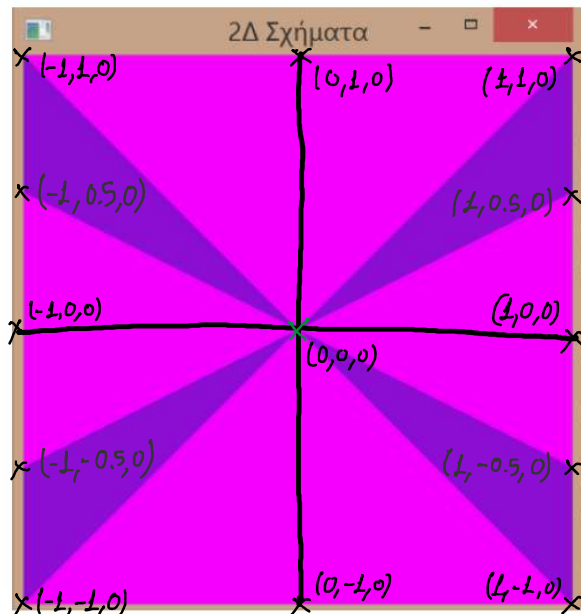
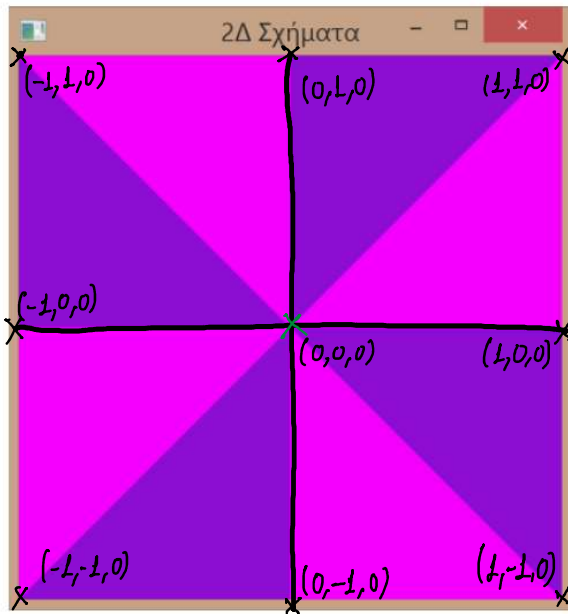


Figure 6: Η λύση στο πρόβλημα

Υπολογισμός Συντεταγμένων

Το κέντρο του παραθύρου, είναι η αρχή των αξόνων $(0,0,0)$, τα υπόλοιπα σημεία είναι μαρκαρισμένα με x στις παρακάτω εικόνες. (οι άξονες που φαίνονται είναι ο x οριζόντια και ο y κατακόρυφα, ο z δεν φαίνεται εδώ)



Ανάλυση του κώδικα που χρησιμοποιήθηκε

Σαν βάση χρησιμοποιήσαμε τον κώδικα που μας δόθηκε στο 1^ο εργαστήριο, εκεί έγιναν τροποποιήσεις στο test.cpp και στο SimpleFragmentShader.fragmentshader

Για το τελευταίο, η αλλαγή που έγινε αφορά το χρώμα όπου βάλαμε το ματζέντα αλλάζοντας την RGB τιμή σε 1,0,1

```
// Output color = Magenta
color = vec3(1,0,1);
```

Στο test.cpp, αλλάξαμε το όνομα σε Source.cpp και πρόσθεσαμε/αλλάξαμε τα παρακάτω

- Γραμμή 139, Διαστάσεις και τίτλος παραθύρου
`window = glfwCreateWindow(800, 800, "2Δ Σχήματα", NULL, NULL); ////`
- Γραμμή 161, Μωβ χρώμα στο background, από το pdf της εκφώνησης πήραμε αυτές τις τιμές, R:142 G:13 B:211. Οι τιμές που βάλαμε είναι από την διαίρεση των παραπάνω τιμών με το 256 (142/256=0.55..., κ.ο.κ.)
`glClearColor(0.55f, 0.05f, 0.82f, 0.0f); ////`
- Γραμμές 181-205, 2 πίνακες που κρατάνε τις συντεταγμένες των κορυφών των τριγώνων για το 1^ο και το 2^ο σχήμα
`static const GLfloat g_vertex_buffer_data[] = {...};
static const GLfloat g_vertex_buffer_data_2[] = {...};`
- Γραμμές 213-216, βάλαμε έναν vertexbuffer για το 2^ο σχήμα, όπως υπάρχει για το 1^ο στις γραμμές 208-211
`GLuint vertexbuffer2;
glGenBuffers(1, &vertexbuffer2);
glBindBuffer(GL_ARRAY_BUFFER, vertexbuffer2);
glBufferData(GL_ARRAY_BUFFER, sizeof(g_vertex_buffer_data_2),
g_vertex_buffer_data_2, GL_STATIC_DRAW);`
- Γραμμή 239, Φτιάχνουμε 12 ακμές, για 4 τρίγωνα (3 ακμές για κάθε τρίγωνο)
`glDrawArrays(GL_TRIANGLES, 0, 4 * 3); // ..`
- Γραμμή 247, αν και για όσο πατιέται το πλήκτρο "c", θα εμφανίζεται το σχήμα 1(β), οι γραμμές 248-275 είναι ίδιες με τις γραμμές 220-245 εκτός των γραμμών 228 και 257 όπου στην πρώτη έχουμε χρήση του vertexbuffer (σχήμα 1(α)) και στη δεύτερη έχουμε τον vertexbuffer2 (σχήμα 1(β)).
`while (glfwGetKey(window, GLFW_KEY_C) == GLFW_PRESS) {...}
glBindBuffer(GL_ARRAY_BUFFER, vertexbuffer);
glBindBuffer(GL_ARRAY_BUFFER, vertexbuffer2);`
- Γραμμή 278, για όσο το πλήκτρο "n" δεν πατιέται, το πρόγραμμα μας τρέχει, ενώ μόλις πατηθεί (ή αν πατηθεί το x του παραθύρου) θα τερματίσει
`while (glfwGetKey(window, GLFW_KEY_N) != GLFW_PRESS && ////`