

I. (5p) Indicați răspunsul corect.

1. (1p) La apelarea funcției `glm::scale (4.0, 3.0, 8.0)`, matricea 4×4 generată are suma elementelor:
a) 15 b) 16 c) 19
2. (1p) Se presupune că se utilizează modelul de amestecare în care factorul sursă are toate componentele egale cu `GL_SRC_ALPHA`, iar factorul destinație `GL_ONE_MINUS_SRC_ALPHA`. Se desenează un pătrat verde, apoi un pătrat roșu. Care va fi combinația RGB în zona de suprapunere, dacă fundalul este (0.0, 0.0, 0.0, 1.0) și ambele pătrate au componenta ALPHA = 0.5?
a) (0.5, 0.25, 0.0) b) (0.25, 0.5, 0.0) c) (0.5, 0.5, 0.0)
3. (1p) Care dintre funcțiile de mai jos indică revenirea la modul implicit de parcurgere a vârfurilor unui poligon?
a) `glFrontFace (GL_CCW)` b) `glCullFace (GL_FRONT)` c) `glFrontFace (GL_CW)`
4. (1p) Se utilizează `glm::LookAt(1, 2, 3, 4, 5, 6, 7, 8, 9)`. Vectorul **n** al reperului de vizualizare este coliniar cu
a) (1, 2, 3) b) (4, 5, 6) c) (1, 1, 1)
5. (1p) Aplicând algoritmul Cohen-Sutherland pentru $x_{\min} = 50, x_{\max} = 60, y_{\min} = 30, y_{\max} = 40$, codul punctului (55, 25) este:
a) 1000 b) 0100 c) 0010

II. (20p) Completați răspunsul.

6. (2p) Un cod RGB care generează o nuanță de gri deschis (alegeți deschis sau închis) este (0.9, 0.9, 0.9)

7. (2p) Descrieți, pe scurt, două diferențe între un shader de vârfuri și un shader de fragment.

1. Shaderul de vârfuri afectează geometria scenei (proiecție, coordonate de texturale etc.), pe când cel de fragment afectează culoarea pixelilor desenați.

2. Shaderul de vârfuri poate manipula poziția vârfurilor, în timp ce shaderul de fragment nu poate face acest lucru în mod direct.

8. (2p) În planul proiectiv $\mathbb{P}^2\mathbb{R}$ punctul $[4 : 3 : -1]$ coincide cu $[-8, 2\alpha + 4, 2]$ pentru $\alpha = -5$, deoarece $2\alpha + 4 = 3 \cdot (-2)$, $2\alpha + 4 = -6$, $2\alpha = -10$, $\alpha = -5$. (Am obținut -2 din $\frac{-8}{4}$ sau $\frac{2}{-1}$)

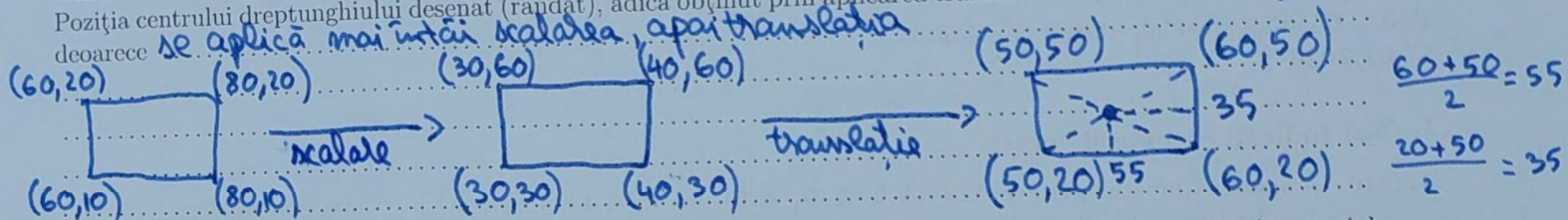
9. (2p) Descrieți, pe scurt, două diferențe între utilizarea funcției `glm::frustum` și a funcției `glm::perspective`.

10. (2p) La apelarea funcției `glDrawArrays(GL_LINES)` pentru un *Vertex Array Object* cu $a = 30$. (alegeți $a > 20$) vârfuri distincte, toate utilizate, numărul de segmente desenate este egal cu 15

11. (2p) La reprezentarea unui segment $[M_0 M_{\text{End}}]$ folosind algoritmul Bresenham s-au obținut parametrii de decizie $p_0 = 2$ și $p_1 = -6$. Se presupune că $M_0 = (25, 41)$. Care este punctul M_{End} ? Justificați!


12. (2p) Un exemplu de alegere a parametrilor funcției `glm::ortho` pentru care paralelipipedul decupat să aibă volumul egal cu 36 este (-3.0, 3.0, -1.0, 1.0, -1.5, 1.5)
(Subiectele continuă pe verso)

13. (2p) Fie $m1 = \text{glm::translate}(20.0, -10.0, 0.0)$, $m2 = \text{glm::scale}(0.5, 3.0, 0.0)$. Se aplică transformarea asociată matricei $m1 \cdot m2$ (vârfurilor) dreptunghiului delimitat de dreptele $x = 60$, $x = 80$, $y = 10$, $y = 20$. Poziția centrului dreptunghiului desenat (randat), adică obținut prin aplicarea transformării indicate, este **(55, 35)** deoarece **se aplică mai întâi scalarea, apoi translația**

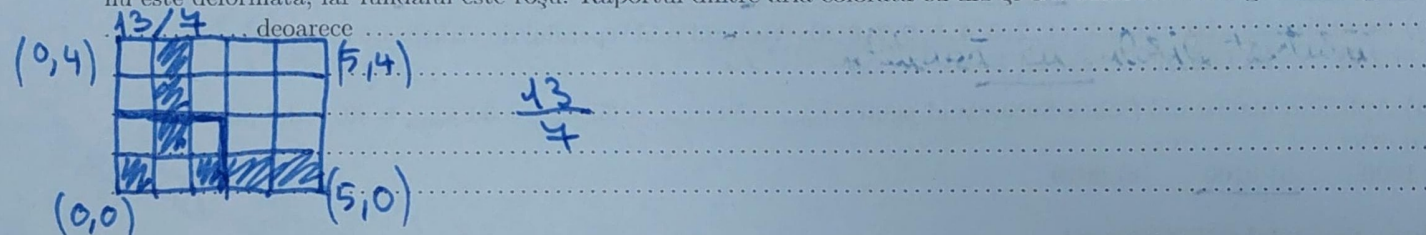


14. (2p) Indicați (și justificați), din curs, o situație / un concept / un rezultat în care este folosit produsul vectorial și o situație / un concept / un rezultat în care este folosit produsul scalar.

produs vectorial: verificarea convexității unui poligon
produs scalar: calcularea intensității luminii (vectorul normal · vectorul direcție al luminii)

15. (2p) Este aplicată unui dreptunghi textura ; coordonatele de texturare asociate vârfurilor sunt (0,0)

(stânga jos), (5,0) (dreapta jos), (5,4) (dreapta sus), (0,4) (stânga sus), este utilizată opțiunea GL_CLAMP, textura nu este deformată, iar fundalul este roșu. Raportul dintre aria colorată cu alb și cea colorată cu negru este egal cu



III. (15p) Rezolvați complet problemele.

16. (5p) Alegeți un dreptunghi $ABCD$ situat într-un plan de forma $y = \text{constant}$ (i.e. paralel cu planul Oxz). Indicați un punct M situat în fața dreptunghiului $ABCD$. Justificați alegerea făcută!

17. (5p) În funcția `createVBO` sunt indicate vârfurile

`GLfloat Vertices[] =`

```
{
    // coordonate          // culori
    4.0f,  4.0f,  -9.0f, 1.0f, 0.0f, 1.0f, 0.0f,
    4.0f, -4.0f,  -9.0f, 1.0f, 0.0f, 1.0f, 0.0f,
    -4.0f, -4.0f, -11.0f, 1.0f, 0.0f, 1.0f, 0.0f,
    -4.0f,  4.0f, -11.0f, 1.0f, 0.0f, 1.0f, 0.0f
};
```

În funcția de desenare se utilizează `glm::ortho(-5,5,-5,5,0,10)` și `glDrawArrays(GL_QUADS, 0, 4)`, dar nu este folosită `glm::lookAt`. Ce arie va avea figura desenată cu verde? Justificați!

18. (5p) Pentru implementarea modelului de iluminare este utilizată formula care ține cont de componenta ambientală și de reflexia difuză, neglijând reflexia speculară, conform codului de mai jos:

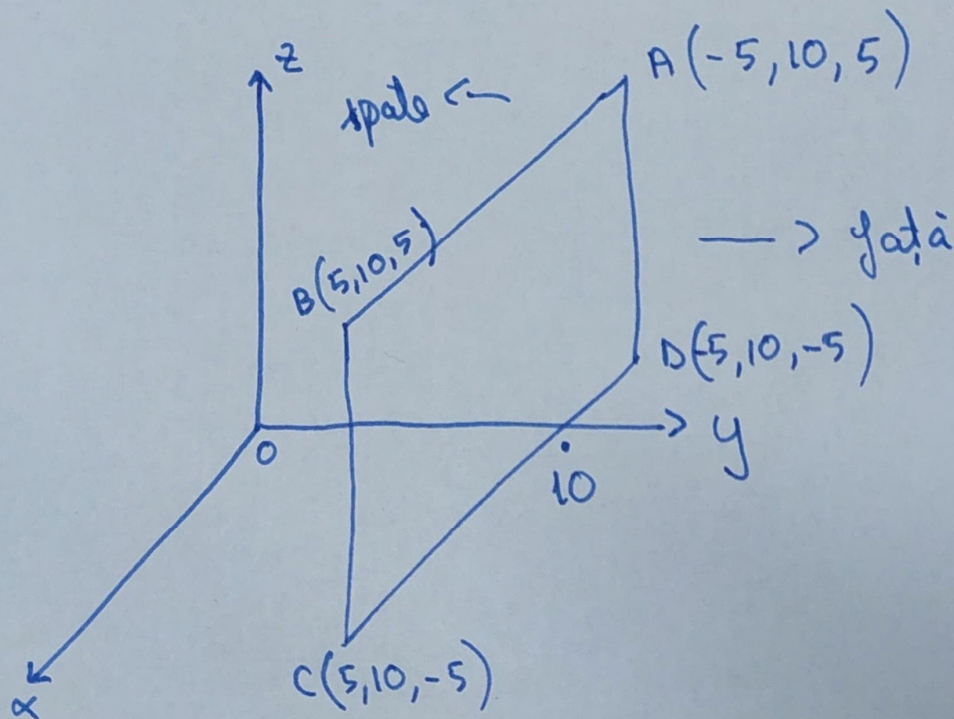
```
vec3 ambient_light = ambientStrength * lightColor; // #1
vec3 ambient_term = ambient_light * objectColor; // #2
vec3 norm = normalize(Normal); // #3
vec3 lightDir = normalize(lightPos - FragPos); // #4
float diff = max(dot(norm, lightDir), 0.0); // #5
vec3 diffuse_light = lightColor; // #6
vec3 diffuse_term = diff * diffuse_light * objectColor; // #7
vec3 result = ambient_term + diffuse_term; // #8
out_Color = vec4(result, 1.0f); // #9
```

Se consideră un fragment de coordonate (5,2,7) cu normala dată de (0,1,0) și cu `objectColor = (0.4,0.2,0.3)`. Sursa de lumină de lumină este situată în punctul (5,4,7). Stabiliți ce alte elemente mai trebuie indicate, alegeți niște valori adecvate nenule și calculați valoarea culorii pentru fragmentul respectiv, explicând pentru fiecare linie calculul efectuat (eticheta liniei i este indicată la final, sub forma $\#i$).

Floreă Mădălin-Horodari
Anul 3, Seria 34, Grupa 343

17.01.2024

Grafică pe calculator
~ lucrare scrisă ~



În sens trigonometric

Paraulatul ABCD este paraulat în sens trigonometric.

→ punctele din dreapta planului ($y > 10$) sunt
situate în fața paraulatului.

→ punctele din stânga planului ($y < 10$) sunt
situate în spatele paraulatului.