

Bài tập 3

Học kỳ 2 - Năm học 2017-2018

Lưu ý: Tất cả các bài tập đều phải VIẾT TAY

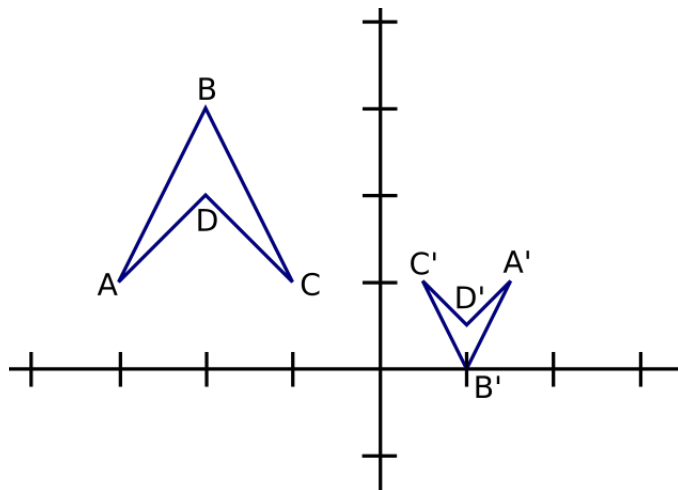
Câu 1

- a) Cho tam giác ABC trong tọa độ 2D với $A = (0, 0)$; $B = (2, 0)$; $C = (0, 2)$. Thu nhỏ tam giác 2 lần theo cả hai trục, tâm scale nằm ở $(0, 0)$; xoay tam giác 90^0 ngược chiều kim đồng hồ với tâm xoay ở $(1, 1)$. Tính ma trận biến đổi trong tọa độ 2D và tọa độ các đỉnh sau khi biến đổi.
- b) Cho tứ diện ABCD với các đỉnh $A = (0, 0, 0)$; $B = (1, 0, 0)$; $C = (0, 1, 0)$; $D = (0, 0, 1)$. Xoay tứ diện 30^0 theo trục z rồi -45^0 theo trục y. Tìm ma trận của phép biến đổi và tọa độ 4 đỉnh của tứ diện sau khi biến đổi.

Câu 2

Biến đổi đa giác ABCD thành đa giác A'B'C'D' (Hình 1).

Hãy tìm ma trận của phép biến đổi, biết tọa độ các đỉnh như sau: $A(-3, 1)$; $B(-2, 3)$; $C(-1, 1)$; $D(-2, 2)$; $A'(1.5, 1)$; $B'(1, 0)$; $C'(0.5, 1)$; $D'(1, 0.5)$.



Hình 1:

Câu 3 Cho đoạn mã sau, biết rằng sử dụng camera mặc định:

```

1 public void display(){
2     glClear(GL_COLOR_BUFFER_BIT);
3     glMatrixMode(GL_MODELVIEW);
4     glLoadIdentity();
5     glTranslatef(3.0f, 2.0f, 1.0f);
6     glScalef(3.0f, 3.0f, 4.0f);
7     glRotatef(60.0f, 0.0f, 0.0f, 1.0f);
8     glRotatef(30.0f, 0.0f, 0.1f, 0.0f);
9     glBegin(GL_TRIANGLES)
10         glVertex3f(-1f, 0f, -1f); //Vertex A
11         glVertex3f(1f, 0f, -1f); //Vertex B
12         glVertex3f(0f, 0f, 1f); //Vertex C
13     glEnd();
14     glFlush();
15 }

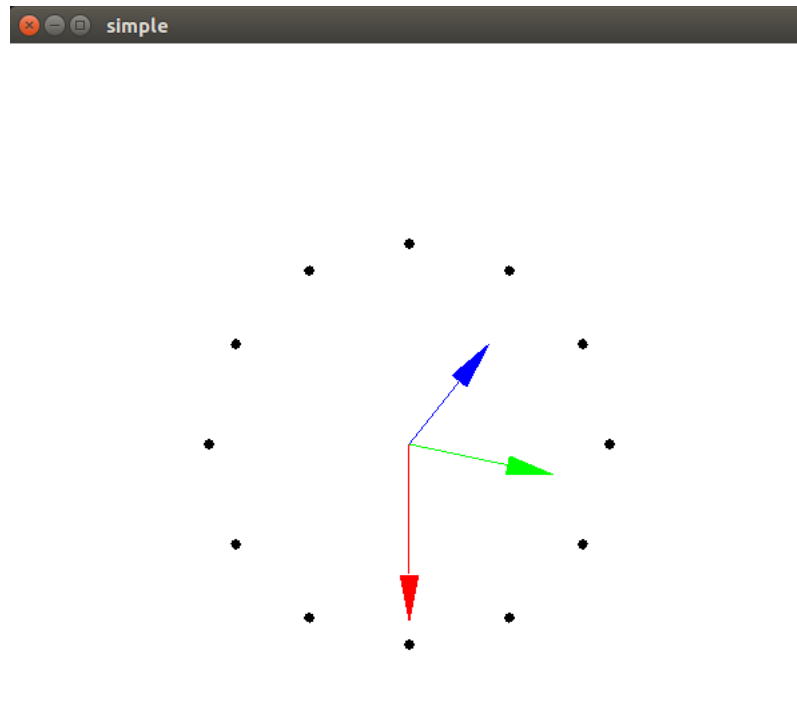
```

- a) Tính ma trận `MODEL_VIEW`. Biết hàm `glGetFloatv` có thể lấy được ma trận `MODEL_VIEW`, viết mã in ra kết quả ma trận, kiểm tra với ma trận vừa tính.
- b) Tính tọa độ các điểm A, B, C sau khi thực hiện các phép biến đổi.

Câu 4

- a) Xét đa giác ABCD ở câu 2 **trong không gian 3D**, giả sử đa giác nằm trên mặt phẳng $z=0$. Xoay góc 45° theo trục z , tâm xoay ở đỉnh B. Tính ma trận biến đổi.
- b) Viết đoạn mã cho hàm *RotatePolygon(double angle)* để xoay hình đa giác quanh tâm với góc cho trước (Đa giác luôn nằm trên mặt phẳng $z = 0$ khi xoay).

Câu 5



Hình 2: 1 giờ 17 phút 30 giây

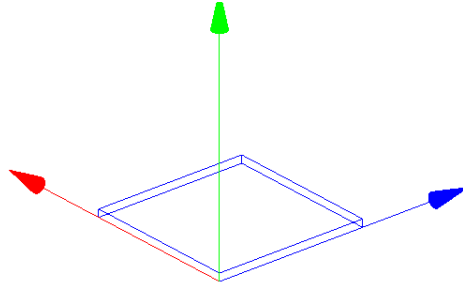
Viết đoạn mã mô phỏng đồng hồ (xem hình tham khảo), với các yêu cầu sau:

- Các hình tròn đen tương ứng với vị trí các giờ trên đồng hồ.
- Kim giờ màu xanh dương, kim phút màu lá cây, kim giây màu đỏ.
- Độ dài kim giây $>$ độ dài kim phút $>$ độ dài kim giờ.
- Cứ mỗi 1 giây sẽ update lại vị trí các kim theo đúng giờ hiện tại trên máy tính.

(Sinh viên tham khảo các hàm `glutSolidSphere`, `glutSolidCone` để vẽ đồng hồ. Hàm `glutTimerFunc`, hoặc `glutIdleFunc` để update thời gian cho đồng hồ, hàm `localtime` của thư viện `time.h` để lấy giờ hiện tại).

Câu 6

Cho đoạn mã vẽ khối hộp (hình 3):

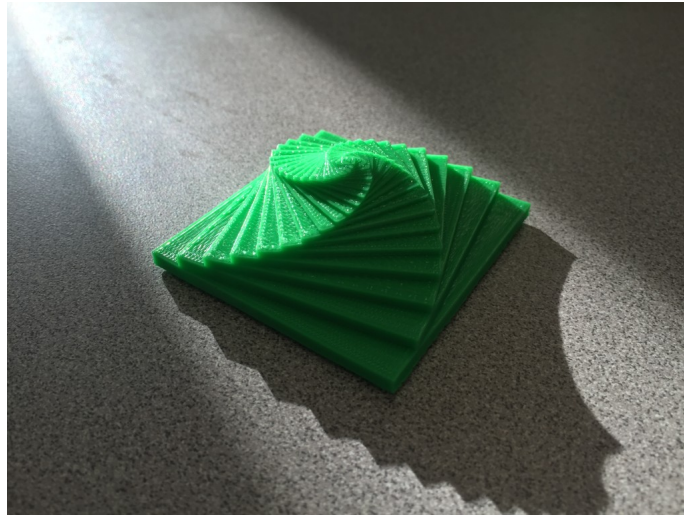


Hình 3:

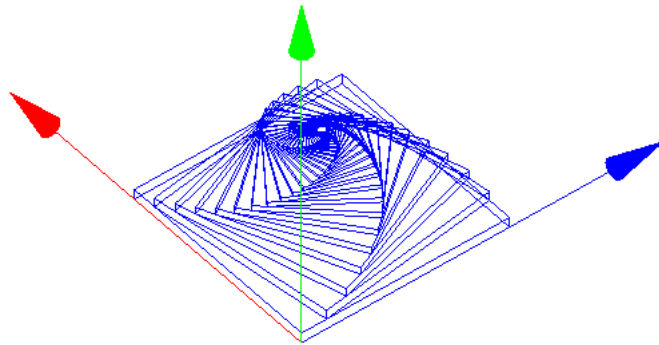
```
1 #define CUBOID_HEIGHT 0.10
2
3 void cuboid()
4 {
5     float vertices[8][3] = {{0, 0, 0}, {2, 0, 0}, {2, 0, 2}, {0,0,2},
6                             {0,CUBOID_HEIGHT, 0}, {2, CUBOID_HEIGHT, 0}, {2, CUBOID_HEIGHT,
7                             2}, {0, CUBOID_HEIGHT, 2}
8 };
9     float line[12][2] = {{0, 1}, {1, 2}, {2, 3}, {3, 0},{4, 5},{5, 6},
10                        {6, 7},{7, 4},
11                        {0, 4}, {1, 5}, {2, 6}, {3, 7}
12 };
13     int i, p1, p2;
14     glBegin(GL_LINES);
15     for (i = 0; i < 12; i++)
16     {
17         p1 = line[i][0];
18         p2 = line[i][1];
19         glVertex3f(vertices[p1][0], vertices[p1][1], vertices[p1][2]);
20         glVertex3f(vertices[p2][0], vertices[p2][1], vertices[p2][2]);
21     }
22     glEnd();
23 }
```

Cho mô hình 3D (Hình 4). Biết mô hình này được tạo ra bằng các chồng các khối có

cùng hình dạng lên nhau, có kích thước giảm dần. Sử dụng các phép biến đổi OpenGL và hàm cuboid cho sẵn để vẽ **wireframe** cho mô hình trên (Hình 5).
(độ xoay và độ scale của các khối hộp sinh viên được tùy ý chọn).



Hình 4:



Hình 5: