**BÀI TẬP LAB 7**

1. **Tên thành viên nhóm:**

* Phan Văn Hà
* Hồ Nhật Huy
* Đỗ Quang Pháp
* Nguyễn Cảnh Thông

1. **Bài làm**
2. **Đường cong Bezier:**
3. **Code:**

#code

#code

1. **Dữ liệu thử:**
2. **Màn hình kết quả:**
3. **Mặt cong Bezier:**
4. **Code:**
5. **Dữ liệu thử:**
6. **Màn hình kết quả:**
7. **Mặt có quy tắc:**
8. **Code:**
9. **Dữ liệu thử:**
10. **Màn hình kết quả:**
11. **Mặt tròn xoay Sphere:**
12. **Code:**
13. **Dữ liệu thử:**
14. **Màn hình kết quả:**
15. **Mặt tròn xoay Ellipsoid:**
16. **Code:**
17. **Dữ liệu thử:**
18. **Màn hình kết quả:**
19. **Mặt tròn xoay Toroid:**
20. **Code:**

#code

#include <windows.h>

#include <GL/glut.h>

#include <stdlib.h>

static void resize(int width, int height)

{

const float ar = (float) width / (float) height;

glViewport(0, 0, width, height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glFrustum(-ar, ar, -1.0, 1.0, 2.0, 100.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity() ;

}

static void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glColor3d(1,0,0);

glPushMatrix();

glTranslated(0.0,-1.2,-6);

glutWireTorus(0.4, 0.8, 10, 20);

glPopMatrix();

glutSwapBuffers();

}

const GLfloat light\_ambient[] = { 0.0f, 0.0f, 0.0f, 1.0f };

const GLfloat light\_diffuse[] = { 1.0f, 1.0f, 1.0f, 1.0f };

const GLfloat light\_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };

const GLfloat light\_position[] = { 2.0f, 5.0f, 5.0f, 0.0f };

const GLfloat mat\_ambient[] = { 0.7f, 0.7f, 0.7f, 1.0f };

const GLfloat mat\_diffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };

const GLfloat mat\_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };

const GLfloat high\_shininess[] = { 100.0f };

/\* Program entry point \*/

int main(int argc, char \*argv[])

{

glutInit(&argc, argv);

glutInitWindowSize(640,480);

glutInitWindowPosition(10,10);

glutInitDisplayMode(GLUT\_RGB | GLUT\_DOUBLE | GLUT\_DEPTH);

glutCreateWindow("3D Torus");

glutReshapeFunc(resize);

glutDisplayFunc(display);

glEnable(GL\_DEPTH\_TEST);

glDepthFunc(GL\_LESS);

glEnable(GL\_LIGHT0);

glEnable(GL\_NORMALIZE);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_LIGHTING);

glLightfv(GL\_LIGHT0, GL\_AMBIENT, light\_ambient);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, light\_diffuse);

glLightfv(GL\_LIGHT0, GL\_SPECULAR, light\_specular);

glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, mat\_ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, high\_shininess);

glutMainLoop();

return EXIT\_SUCCESS;

}

#code

1. **Dữ liệu thử:**
2. **Màn hình kết quả:**

