#include <windows.h> // for MS Windows

#include <GL/glut.h> // GLUT, include glu.h and gl.h

#include <GL/gl.h>

/\* Global variables \*/

char title[] = "2D Pig launcher";

GLfloat pigx=0.0f;

int refreshMills = 15; // refresh interval in milliseconds [NEW]

/\* Initialize OpenGL Graphics \*/

void initGL() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque

glClearDepth(1.0f); // Set background depth to farthest

glEnable(GL\_DEPTH\_TEST); // Enable depth testing for z-culling

glDepthFunc(GL\_LEQUAL); // Set the type of depth-test

glShadeModel(GL\_SMOOTH); // Enable smooth shading

glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST); // Nice perspective corrections

}

/\* Handler for window-repaint event. Called back when the window first appears

and

whenever the window needs to be re-painted. \*/

void disp(){

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); // Clear color and depth buffers

glMatrixMode(GL\_MODELVIEW); // To operate on model-view matrix

// Render a color-cube consisting of 6 quads with different colors

glLoadIdentity(); // Reset the model-view matrix

glTranslatef(1.5f, 0.0f, -7.0f); // Move right and into the screen

glBegin(GL\_POLYGON); //Begin drawing the bottom tube

glColor3f(1.0f, 1.0f, 0.0f); //yellow

glVertex3f(4.0f, -0.5f, 0.0f);

glVertex3f(-6.0f, -0.5f, 0.0f);

glVertex3f(-6.0f, -1.5f, 0.0f);

glVertex3f(4.0f, -1.5f, 0.0f);

glEnd(); // End of drawing bottom tube

glBegin(GL\_POLYGON); // drawing the mud

glColor3f(1.10f, 0.6f, 0.5f);

glVertex3f(3.0f, -0.5f, 0.0f);

glVertex3f(1.0f, -0.5f, 0.0f);

glVertex3f(1.0f, -1.5f, 0.0f);

glVertex3f(3.0f, -1.5f, 0.0f);

glEnd(); //end mud

glBegin(GL\_POLYGON); // oil/gas

glColor3f(1.0f, 1.0f, 0.0f);

if (pigx>1.1f)

glColor3f(0.0f, 0.0f, 1.0f);

glVertex3f(-5.0f, -0.5f, 0.0f);

glVertex3f(-4.7f, -0.5f, 0.0f);

glVertex3f(-4.7f, 0.5f, 0.0f);

glVertex3f(-5.0f, 0.5f, 0.0f);

glEnd(); // end oil/gas

glBegin(GL\_POLYGON); // top tube

glColor3f(0.0f, 0.0f, 1.0f);

glVertex3f(-6.0f, 0.5f, 0.0f);

glVertex3f(2.0f, 0.5f, 0.0f);

glVertex3f(2.0f, 1.5f, 0.0f);

glVertex3f(-6.0f, 1.5f, 0.0f);

glEnd(); // end top tube

glBegin(GL\_POLYGON); // connecting tube

glColor3f(1.0f, 1.0f, 1.0f);

glVertex3f(2.0f, 0.5f, 0.0f);

glVertex3f(1.0f, 0.5f, 0.0f);

glVertex3f(1.0f, -0.5f, 0.0f);

glVertex3f(2.0f, -0.5f, 0.0f);

glEnd(); // end of connecting tube

glBegin(GL\_POLYGON); //connecting slim tube

glColor3f(0.0f, 0.0f, 1.0f);

glVertex3f(-6.0f, -0.5f, 0.0f);

glVertex3f(-6.0f, -1.5f, 0.0f);

glVertex3f(-5.9+pigx, -1.5f, 0.0f);

glVertex3f(-5.9+pigx, -0.5f, 0.0f);

glEnd(); // end of conecting tube

glTranslatef(pigx, 0.0, 0.0); //mooving the pig in the tube

glBegin(GL\_POLYGON); // pig

glColor3f(1.0f, 0.0f, 0.0f);

glVertex3f(-6.0f, -0.5f, 0.0f);

glVertex3f(-5.0f, -0.5f, 0.0f);

glVertex3f(-4.5f, -1.0f, 0.0f);

glVertex3f(-5.0f, -1.5f, 0.0f);

glVertex3f(-6.0f, -1.5f, 0.0f);

glEnd(); //end of pig

glutSwapBuffers(); // Swap the front and back frame buffers (double buffering)

// Update the x-coordinate of the pig after each refresh [NEW]

pigx += 0.025f;

}

/\* Called back when timer expired [NEW] \*/

void timer(int value) {

glutPostRedisplay(); // Post re-paint request to activate display()

glutTimerFunc(refreshMills, timer, 0); // next timer call milliseconds later

}

/\* Handler for window re-size event. Called back when the window first appears

and

whenever the window is re-sized with its new width and height \*/

void reshape(GLsizei width, GLsizei height) { // GLsizei for non-negative integer

// Compute aspect ratio of the new window

if (height == 0) height = 1; // To prevent divide by 0

GLfloat aspect = (GLfloat)width / (GLfloat)height;

// Set the viewport to cover the new window

glViewport(0, 0, width, height);

// Set the viewport to cover the new window

glMatrixMode(GL\_PROJECTION); // To operate on the Projection matrix

glLoadIdentity(); // Reset

// Enable perspective projection with fovy, aspect, zNear and zFar

gluPerspective(45.0f, aspect, 0.1f, 100.0f);

}

/\* Main function: GLUT runs as a console application starting at main() \*/

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

{

glutInit(&argc, argv); // Initialize GLUT

glutInitDisplayMode(GLUT\_DOUBLE); // Enable double buffered mode

glutInitWindowSize(640, 480); // Set the window's initial width & height

glutInitWindowPosition(50, 50); // Position the window's initial top-left corner

glutCreateWindow(title); // Create window with the given title

glutDisplayFunc(disp); // Register callback handler for window re-paint event

glutReshapeFunc(reshape); // Register callback handler for window re-size event

initGL(); // Our own OpenGL initialization

glutTimerFunc(0, timer, 0); // First timer call immediately [NEW]

glutMainLoop(); // Enter the infinite event-processing loop

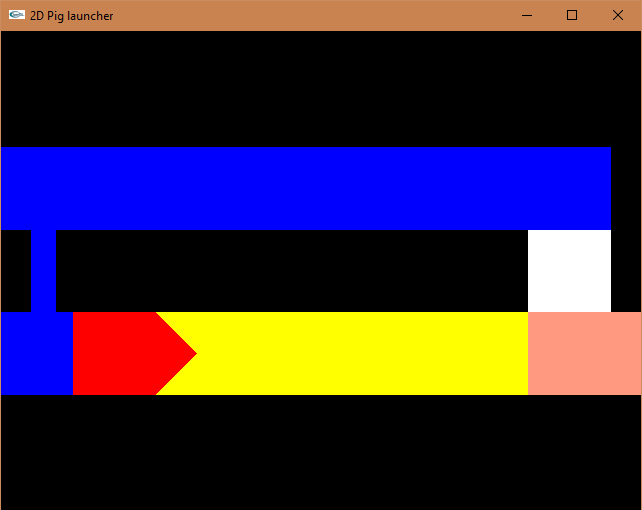
return 0;

}

**INITIAL PHASE**



**INTERMEDIATE PHASE**



**FINAL PHASE**

