Continue

: --. . , ! , . . ? ? , . . , . , . , . . . (Glu) , . : C5H10NO3. (), (), () (N) . 19, 146,15 /. , , , . , . C5H10NO3. , , . , . . , . , . ! . , . , . . : ( ) 20 , . (-NH2), (-COOH) , . . , , , . , . , . : --. , , . . : . , . , , . - . , pH . . . : , , . , . , - . , . , . : (), (), (). : --. () 20 , . , , . , . () . . , , . () . . , . -- , . . , , . . , (), () (), --. , . : ntc Verification failed. Please try again.Ti_Fix 23 2014 20:04 , Python . , , , , . , Python , , 3D . . , OpenGL Python. 3D . , , , , . (, OpenGL) , Python. PyOpenGL , Python OpenGL, GLU GLUT, OpenGL. , : Python (). PyCharm () (,). PyOpenGL (). Python (.py). 3D (, OpenGL) :from OpenGL.GL import * from OpenGL.GLU import * from OpenGL.GLUT import * , , glut, . RGB ( ): glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB) (,):glutInitWindowSize(300, 300) :glutInitWindowPosition(50, 50) OpenGl:glutInit(sys.argv) Happy New Year!:glutCreateWindow(b"Happy New Year!") :glutMainLoop() , Happy New Year!. , ! . , , , . . :# def init(): global xrot # x global yrot # y global ambient # global greencolor # global treecolor # global lightpos # xrot = 0.0 # x = 0 yrot = 0.0 # y = 0 ambient = (1.0, 1.0, 1.0, 1) # - RGB, - greencolor = (0.2, 0.8, 0.0, 0.8) # treecolor = (0.9, 0.6, 0.3, 0.8) # lightpos = (1.0, 1.0, 1.0) # xyz glClearColor(0.5, 0.5, 0.5, 1.0) # gluOrtho2D(-1.0, 1.0, -1.0, 1.0) # glRotatef(-90, 1.0, 0.0, 0.0) # 90 glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambient) # glEnable(GL_LIGHTING) # glEnable(GL_LIGHT0) # glLightfv(GL_LIGHT0, GL_POSITION, lightpos) # xrot yrot x y (,). ambient : . (greencolor, treecolor), . glClearColor, , . gluOrtho2D . glRotatef X 90 , . glLightModelfv . glEnable(GL_LIGHTING). , . glEnable(GL_LIGHT0). , , : glLightfv(GL_LIGHT0, GL_POSITION, lightpos) OpenGL . : , , glut (glutSpecialFunc(specialkeys). specialkeys : # def specialkeys(key, x, y): global xrot global yrot # if key == GLUT_KEY_UP: # xrot -= 2.0 # X if key == GLUT_KEY_DOWN: # xrot += 2.0 # X if key == GLUT_KEY_LEFT: # yrot -= 2.0 # Y if key == GLUT_KEY_RIGHT: # yrot += 2.0 # Y glutPostRedisplay() # specialkeys , , , xrot yrot, glutPostRedisplay . , ! specialkeys glut , : glutDisplayFunc(draw). draw :# def draw(): global xrot global yrot global lightpos global greencolor global treecolor glClear(GL_COLOR_BUFFER_BIT) # glPushMatrix() # "" glRotatef(xrot, 1.0, 0.0, 0.0) # X xrot glRotatef(yrot, 0.0, 1.0, 0.0) # Y yrot glLightfv(GL_LIGHT0, GL_POSITION, lightpos) # # # : 2 , , glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, treecolor) glTranslatef(0.0, 0.0, -0.7) # Z -0.7 # 0.1, 0.2 # glutSolidCylinder(0.1, 0.2, 20, 20) # # : 2 , , glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, greencolor) glTranslatef(0.0, 0.0, 0.2) # Z 0.2 # () 0.5, 0.5 # glutSolidCone(0.5, 0.5, 20, 20) glTranslatef(0.0, 0.0, 0.3) # Z -0.3 glutSolidCone(0.4, 0.4, 20, 20) # 0.4, 0.4 glTranslatef(0.0, 0.0, 0.3) # Z -0.3 glutSolidCone(0.3, 0.3, 20, 20) # 0.3, 0.3 glPopMatrix() # "" glutSwapBuffers() # glClear(GL_COLOR_BUFFER_BIT) . glPushMatrix() glPopMatrix() . glLightfv(GL_LIGHT0, GL_POSITION, lightpos) , , . glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, color) , , . glutSolidCone(0.5, 0.5, 20, 20), glutSolidCylinder(0.1, 0.2, 20, 20). , , (). , , glutSwapBuffers() . :# :from OpenGL.GL import *from OpenGL.GLU import *from OpenGL.GLUT import *import sys# global xrot # xglobal yrot # yglobal ambient # global greencolor # global treecolor # global lightpos # # def init(): global xrot # x global yrot # y global ambient # global greencolor # global treecolor # global lightpos # xrot = 0.0 # x = 0 yrot = 0.0 # y = 0 ambient = (1.0, 1.0, 1.0, 1) # RGB, - greencolor = (0.2, 0.8, 0.0, 0.8) # treecolor = (0.9, 0.6, 0.3, 0.8) # lightpos = (1.0, 1.0, 1.0) # xyz glClearColor(0.5, 0.5, 0.5, 1.0) # gluOrtho2D(-1.0, 1.0, -1.0, 1.0) # glRotatef(-90, 1.0, 0.0, 0.0) # 90 glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambient) # glEnable(GL_LIGHTING) # glEnable(GL_LIGHT0) # glLightfv(GL_LIGHT0, GL_POSITION, lightpos) # # def specialkeys(key, x, y): global xrot global yrot # if key == GLUT_KEY_UP: # xrot -= 2.0 # if key == GLUT_KEY_DOWN: # xrot += 2.0 # if key == GLUT_KEY_LEFT: # yrot -= 2.0 # Y if key == GLUT_KEY_RIGHT: # yrot += 2.0 # Y glutPostRedisplay() # # def draw(): global xrot global yrot global lightpos global greencolor global treecolor glClear(GL_COLOR_BUFFER_BIT) # glPushMatrix() # "" glRotatef(xrot, 1.0, 0.0, 0.0) # X xrot glRotatef(yrot, 0.0, 1.0, 0.0) # Y yrot glLightfv(GL_LIGHT0, GL_POSITION, lightpos) # # # : 2 , , glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, treecolor) glTranslatef(0.0, 0.0, -0.7) # Z -0.7 # 0.1, 0.2 # glutSolidCylinder(0.1, 0.2, 20, 20) # # : 2 , , glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, greencolor) glTranslatef(0.0, 0.0, 0.2) # Z 0.2 # () 0.5, 0.5 # glutSolidCone(0.5, 0.5, 20, 20) glTranslatef(0.0, 0.0, 0.3) # Z -0.3 glutSolidCone(0.4, 0.4, 20, 20) # 0.4, 0.4 glTranslatef(0.0, 0.0, 0.3) # Z -0.3 glutSolidCone(0.3, 0.3, 20, 20) # 0.3, 0.3 glPopMatrix() # "" glutSwapBuffers() # # # RGB (, ,)glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB)# (,)glutInitWindowSize(300, 300)# glutInitWindowPosition(50, 50)# OpenGlglutInit(sys.argv)# "Happy New Year!"glutCreateWindow(b"Happy New Year!")# , glutDisplayFunc(draw)# , glutSpecialFunc(specialkeys)# init()# glutMainLoop() ( ): : , , . , - Python OpenGL. , !!!P.S. PyOpenGL .:pythonpython3opengl +1---

How to draw a galvanic cell. How to galvanic cells work. How to make a galvanic cell. Homemade galvanic cell.