**Lab Taks-4**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
* Must submit within the announced time.
* Must include resources for all the section in the table

|  |
| --- |
| **Question-1**  Create an animation using two box that will move in the opposite direction. |
| **Graph Plot (Picture)-** |
| #include <windows.h>  #include <GL/glut.h>  float moveBlue = 0.0f;  float moveGreen = 0.0f;  void objectBlue()  {  // glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glTranslatef( 0.0f, moveBlue, 0.0f);  glBegin(GL\_POLYGON);  glColor3f(0.0f, 0.0f, 1.0f);  glVertex2f(0.0f,0.0f);  glVertex2f(2.5f,0.0f);  glVertex2f(2.5f,2.5f);  glVertex2f(0.0f,2.5f);  glEnd();  glPopMatrix();  }  void objectGreen()  {  //glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glTranslatef( 0.0f, moveGreen, 0.0f);  glBegin(GL\_POLYGON);  glColor3f(0.0f, 1.0f, 0.0f);  glVertex2f(0.0f,0.0f);  glVertex2f(-2.5f,0.0f);  glVertex2f(-2.5f,2.5f);  glVertex2f(0.0f,2.5f);  glEnd();  glPopMatrix();  }  void display()  {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  objectBlue();  objectGreen();  glutSwapBuffers();  glFlush();  }  void updateBlue(int value)  {  moveBlue += .04;  if(moveBlue > 10)//10  {  moveBlue = -10.0;  }  glutPostRedisplay();  glutTimerFunc(20, updateBlue, 0);  }  void updateGreen(int value)  {  moveGreen -= .04;  if(moveGreen < -10)  {  moveGreen = 10.0;  }  glutPostRedisplay();  glutTimerFunc(20, updateGreen, 0);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutCreateWindow("Shraboni Biswas Naboni-26");  glutInitWindowSize(470, 470);  gluOrtho2D(-10,10,-10,10);  glutDisplayFunc(display);  glutTimerFunc(20, updateBlue, 0);  glutTimerFunc(20, updateGreen, 0);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-2**  Design a car which will have rotating wheels. |
|  |
| **Code-**  **#include <GL/glut.h>**  **#include <cmath>**  **float \_angle1 = 0.0;**  **float scaleFactor = 0.5;**  **void Car()**  **{**  **glColor3ub(105, 105, 105);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.947279884431,0.9175335415077);//n**  **glVertex2f(13.6559383571238,0.9175335415077);**  **glVertex2f(15.0257997032635,1.1129061741962);**  **glVertex2f(15.0257997032635,1.739716606291);**  **glVertex2f(15.1156647573824,1.9070330455765);**  **glVertex2f(15.1156647573824,1.9070330455765);**  **glVertex2f(15.047886132782,2.0199974199105);**  **glVertex2f(14.9537491541703,2.1743820648337);**  **glVertex2f(14.4524390108801,2.8271481560979);**  **glVertex2f(14,3);//z**  **glVertex2f(12.2266159157762,3.3118063434317);**  **glVertex2f(11.5148279918614,3.3324378774582);**  **glVertex2f(8.5810107975117,4.7066289310759);**  **glVertex2f(4.6645470819727,4.7066289310759);**  **glVertex2f(2.4599151891196,3.6432182533467);**  **glVertex2f(1.3705676655922,3.5654077159519);**  **glVertex2f(1.7855571983646,3.1244813373813);**  **glVertex2f(1.7077466609698,2.5279338840211);**  **glVertex2f(1.2149465908026,2.5538707298193);**  **glVertex2f(1.1111444614174,2.1439570611947);**  **glVertex2f(1.0490752674903,2.0060387600198);**  **glVertex2f(1.0783692372764,1.6195534749397);**  **glVertex2f(1.126999832313,1.4644230916896);**  **glVertex2f(1.0352689485038,1.3399311779486);**  **glVertex2f(1.0568442825556,1.2155046002614);**  **glVertex2f(1.1597608622448,1.1106039684258);**  **glVertex2f(2.0901741123089,0.9631793337325);**  **glEnd();**  **glColor3ub(192, 0, 0); //back light**  **glBegin(GL\_POLYGON);**  **glVertex2f(1.3705676655922,3.5654077159519);//n**  **glVertex2f(1.7855571983646,3.1244813373813);**  **glVertex2f(1.7077466609698,2.5279338840211);**  **glVertex2f(1.2149465908026,2.5538707298193);**  **glEnd();**  **glColor3ub(30, 30, 30); // RGB for darker gray**  **glBegin(GL\_POLYGON);**  **glVertex2f(5.0740348200318,4.3052645840064);**  **glVertex2f(5.076144447425,3.3964723886863);**  **glVertex2f(4.1512055831507,3.8547067801174);**  **glVertex2f(4.1512055831507,3.8547067801174);**  **glEnd();**  **glColor3ub(50, 50, 50); //glass2**  **glBegin(GL\_POLYGON);**  **glVertex2f(4.1512055831507,3.8547067801174);**  **glVertex2f(5.0740348200318,4.3052645840064);**  **glVertex2f(6,4.5);**  **glVertex2f(7,4.5);**  **glVertex2f(7.2128272866857,3.3444364961949);**  **glVertex2f(4.20112055831507,3.3444364961949);**  **glEnd();**  **glColor3ub(50, 50, 50);//glass1**  **glBegin(GL\_POLYGON);**  **glVertex2f(7.4035635090852,4.5092526906923);**  **glVertex2f(8.4802922613314,4.411957923923);**  **glVertex2f(9.0177617008289,4.2895942580031);**  **glVertex2f(10.1643169486462,3.6208729342601);**  **glVertex2f(9.8912143537186,3.3127064225344);**  **glVertex2f(7.9029602885882,3.3127064225344);**  **glEnd();**  **glColor3ub(105, 105, 105);//head light// use border**  **glBegin(GL\_POLYGON);**  **glVertex2f(9.7690362977999,3.6040524810326);**  **glVertex2f(10.4923157865826,3.6040524810326);**  **glVertex2f(10.5427771462651,3.2592331898688);**  **glVertex2f(9.8912143537186,3.2552424322127);**  **glVertex2f(9.8051506092943,3.4375818625029);**  **glEnd();**  **glColor3ub(192, 0, 0);//Front glass**  **glBegin(GL\_POLYGON);**  **glVertex2f(14.0242069189416,2.6119550937212);**  **glVertex2f(14.4862548037162,2.6119550937212);**  **glVertex2f(14.8039127244987,2.2172891921429);**  **glVertex2f(13.8220609693527,2.2172891921429);**  **glEnd();**  **glColor3ub(0, 0,0);//head light// use border**  **glBegin(GL\_LINE\_LOOP);**  **glVertex2f(9.7690362977999,3.6040524810326);**  **glVertex2f(10.4923157865826,3.6040524810326);**  **glVertex2f(10.5427771462651,3.2592331898688);**  **glVertex2f(9.8912143537186,3.2552424322127);**  **glVertex2f(9.8051506092943,3.4375818625029);**  **glEnd();**  **}**  **void border() {**  **// Calculate the center of the circle**  **float circleCenterX = 4.229023150212;**  **float circleCenterY = 1.1418732470549;**  **/\***  **glPushMatrix();**  **glTranslatef(circleCenterX, circleCenterY, 0);**  **glRotatef(\_angle1, 0.0, 0.0, 1.0);**  **\*/**  **glColor3ub(211, 211, 211); // Set border color to light gray**  **glBegin(GL\_POLYGON);**  **// Adjust vertex positions relative to the circle's center**  **glVertex2f((3.5617825514923 - 4.80) \* scaleFactor, (1.4154645467633 - circleCenterY) \* scaleFactor);**  **glVertex2f((4.0818670836389 - 4.80) \* scaleFactor, (1.2235199381484 - circleCenterY) \* scaleFactor);**  **glVertex2f((4.0896308776401 - 4.80) \* scaleFactor, (1.0565983671225 - circleCenterY) \* scaleFactor);**  **glVertex2f((3.5239432537322 - 4.80) \* scaleFactor, (0.8614193146049 - circleCenterY) \* scaleFactor);**  **glEnd();**  **// glPopMatrix();**  **}**  **void border1()**  **{**  **float circleCenterX = 12.4373260390525;**  **float circleCenterY = 1.2647611733131;**  **//12.4373260390525, 1.2647611733131**  **/\*glPushMatrix();**  **glTranslatef(circleCenterX, circleCenterY, 0);**  **glRotatef(\_angle1, 0.0, 0.0, 1.0);**  **\*/**  **glColor3ub(211, 211, 211); // Light Gray**  **glBegin(GL\_POLYGON);**  **// Adjust vertex positions relative to the circle's center**  **glVertex2f((12.3067892620237 - 13.00832) \* scaleFactor, (1.3589203311724 - circleCenterY) \* scaleFactor);**  **glVertex2f((12.3067892620237 - 13.00832) \* scaleFactor, (1.1647620496533- circleCenterY) \* scaleFactor);**  **glVertex2f((11.77443140 - 13.00832) \* scaleFactor, (0.87544089 - circleCenterY) \* scaleFactor);**  **glVertex2f((11.77443140 - 13.00832) \* scaleFactor, (1.4853853 - circleCenterY) \* scaleFactor);**  **glEnd();**  **// glPopMatrix();**  **}**  **void white\_border() {**  **glColor3ub(255, 255, 255); // Set color to white**  **glBegin(GL\_POLYGON);**  **glVertex2f(1.81529, 1.006734);**  **glVertex2f(1.8232217261141, -0.5645063766661);**  **glVertex2f(14.5192132653909, -0.5738074693689);**  **glVertex2f(14.4741651352254, 1.0318433252581);**  **glEnd();**  **}**  **void wheel(float radius, float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 200; i++) {**  **glColor3f(r, g, b);**  **float pi = 3.1416;**  **float A = (i \* 2 \* pi) / 200;**  **float x = radius \* cos(A);**  **float y = radius \* sin(A);**  **glVertex2f(x + xc, y + yc);**  **}**  **glEnd();**  **}**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **Car();**  **wheel(sqrt(1.905916878038), 4.22, 1.09, 0, 0, 0);**  **wheel(sqrt(1.8071686868979), 12.4754573237752, 1.2394829142391, 0, 0, 0);**  **wheel(sqrt(1.6305916878038), 4.22, 1.09, 105 / 255.0, 105 / 255.0, 105 / 255.0);**  **wheel(sqrt(1.4971686868979), 12.4754573237752, 1.2394829142391, 105 / 255.0, 105 / 255.0, 105 / 255.0);**  **white\_border();**  **wheel(sqrt(1.0743966196217), 4.229023150212, 1.1418732470549, 0, 0, 0);**  **wheel(sqrt(1.0606156155846), 12.4373260390525, 1.2647611733131, 0, 0, 0);**  **glPushMatrix();**  **glTranslatef(4.229023150212, 1.1418732470549, 0.0);**  **glRotatef(\_angle1, 0, 0, 1);**  **for (int i = 0; i < 10; i++)**  **{**  **glRotatef(90.0 \* i, 0, 0, 1);**  **border();**  **}**  **glPopMatrix();**    **glPushMatrix();**  **glTranslatef(12.4373260390525, 1.2647611733131, 0.0);**  **glRotatef(\_angle1, 0, 0, 1);**  **for (int i = 0; i < 10; i++)**  **{**  **glRotatef(90.0 \* i, 0, 0, 1);**  **border1();**  **}**  **glPopMatrix();**  **// border();**  **//border1();**  **glutSwapBuffers();**  **}**  **void update(int value) {**  **\_angle1 += 2.0f;**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutCreateWindow("OpenGL Setup Test");**  **glutInitWindowSize(320, 320);**  **gluOrtho2D(-40, 40, -20, 10);**  **glutDisplayFunc(display);**  **glutTimerFunc(20, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-3**  Now move your car of question-2 from left to right in a loop. |
|  |
| **Code-**  **#include <GL/glut.h>**  **#include <cmath>**  **float \_angle1 = 0.0;**  **float scaleFactor = 0.5;**  **float carPosition = -40.0f;//-20.0f; //**  **float carSpeed = 0.10f;**  **void Car() {**  **glColor3ub(105, 105, 105);**  **glBegin(GL\_POLYGON);**  **glVertex2f(2.947279884431, 0.9175335415077); //n**  **glVertex2f(13.6559383571238, 0.9175335415077);**  **glVertex2f(15.0257997032635, 1.1129061741962);**  **glVertex2f(15.0257997032635, 1.739716606291);**  **glVertex2f(15.1156647573824, 1.9070330455765);**  **glVertex2f(15.1156647573824, 1.9070330455765);**  **glVertex2f(15.047886132782, 2.0199974199105);**  **glVertex2f(14.9537491541703, 2.1743820648337);**  **glVertex2f(14.4524390108801, 2.8271481560979);**  **glVertex2f(14, 3); //z**  **glVertex2f(12.2266159157762, 3.3118063434317);**  **glVertex2f(11.5148279918614, 3.3324378774582);**  **glVertex2f(8.5810107975117, 4.7066289310759);**  **glVertex2f(4.6645470819727, 4.7066289310759);**  **glVertex2f(2.4599151891196, 3.6432182533467);**  **glVertex2f(1.3705676655922, 3.5654077159519);**  **glVertex2f(1.7855571983646, 3.1244813373813);**  **glVertex2f(1.7077466609698, 2.5279338840211);**  **glVertex2f(1.2149465908026, 2.5538707298193);**  **glVertex2f(1.1111444614174, 2.1439570611947);**  **glVertex2f(1.0490752674903, 2.0060387600198);**  **glVertex2f(1.0783692372764, 1.6195534749397);**  **glVertex2f(1.126999832313, 1.4644230916896);**  **glVertex2f(1.0352689485038, 1.3399311779486);**  **glVertex2f(1.0568442825556, 1.2155046002614);**  **glVertex2f(1.1597608622448, 1.1106039684258);**  **glVertex2f(2.0901741123089, 0.9631793337325);**  **glEnd();**  **glColor3ub(192, 0, 0); //back light**  **glBegin(GL\_POLYGON);**  **glVertex2f(1.3705676655922, 3.5654077159519); //n**  **glVertex2f(1.7855571983646, 3.1244813373813);**  **glVertex2f(1.7077466609698, 2.5279338840211);**  **glVertex2f(1.2149465908026, 2.5538707298193);**  **glEnd();**  **glColor3ub(30, 30, 30); // RGB for darker gray**  **glBegin(GL\_POLYGON);**  **glVertex2f(5.0740348200318, 4.3052645840064);**  **glVertex2f(5.076144447425, 3.3964723886863);**  **glVertex2f(4.1512055831507, 3.8547067801174);**  **glVertex2f(4.1512055831507, 3.8547067801174);**  **glEnd();**  **glColor3ub(50, 50, 50); //glass2**  **glBegin(GL\_POLYGON);**  **glVertex2f(4.1512055831507, 3.8547067801174);**  **glVertex2f(5.0740348200318, 4.3052645840064);**  **glVertex2f(6, 4.5);**  **glVertex2f(7, 4.5);**  **glVertex2f(7.2128272866857, 3.3444364961949);**  **glVertex2f(4.20112055831507, 3.3444364961949);**  **glEnd();**  **glColor3ub(50, 50, 50); //glass1**  **glBegin(GL\_POLYGON);**  **glVertex2f(7.4035635090852, 4.5092526906923);**  **glVertex2f(8.4802922613314, 4.411957923923);**  **glVertex2f(9.0177617008289, 4.2895942580031);**  **glVertex2f(10.1643169486462, 3.6208729342601);**  **glVertex2f(9.8912143537186, 3.3127064225344);**  **glVertex2f(7.9029602885882, 3.3127064225344);**  **glEnd();**  **glColor3ub(105, 105, 105); //head light**  **glBegin(GL\_POLYGON);**  **glVertex2f(9.7690362977999, 3.6040524810326);**  **glVertex2f(10.4923157865826, 3.6040524810326);**  **glVertex2f(10.5427771462651, 3.2592331898688);**  **glVertex2f(9.8912143537186, 3.2552424322127);**  **glVertex2f(9.8051506092943, 3.4375818625029);**  **glEnd();**  **glColor3ub(192, 0, 0); //Front glass**  **glBegin(GL\_POLYGON);**  **glVertex2f(14.0242069189416, 2.6119550937212);**  **glVertex2f(14.4862548037162, 2.6119550937212);**  **glVertex2f(14.8039127244987, 2.2172891921429);**  **glVertex2f(13.8220609693527, 2.2172891921429);**  **glEnd();**  **glColor3ub(0, 0, 0); //head light border**  **glBegin(GL\_LINE\_LOOP);**  **glVertex2f(9.7690362977999, 3.6040524810326);**  **glVertex2f(10.4923157865826, 3.6040524810326);**  **glVertex2f(10.5427771462651, 3.2592331898688);**  **glVertex2f(9.8912143537186, 3.2552424322127);**  **glVertex2f(9.8051506092943, 3.4375818625029);**  **glEnd();**  **}**  **void border() {**  **float circleCenterX = 4.229023150212;**  **float circleCenterY = 1.1418732470549;**  **glColor3ub(211, 211, 211);**  **glBegin(GL\_POLYGON);**  **glVertex2f((3.5617825514923 - 4.80) \* scaleFactor, (1.4154645467633 - circleCenterY) \* scaleFactor);**  **glVertex2f((4.0818670836389 - 4.80) \* scaleFactor, (1.2235199381484 - circleCenterY) \* scaleFactor);**  **glVertex2f((4.0896308776401 - 4.80) \* scaleFactor, (1.0565983671225 - circleCenterY) \* scaleFactor);**  **glVertex2f((3.5239432537322 - 4.80) \* scaleFactor, (0.8614193146049 - circleCenterY) \* scaleFactor);**  **glEnd();**  **}**  **void border1() {**  **float circleCenterX = 12.4373260390525;**  **float circleCenterY = 1.2647611733131;**  **glColor3ub(211, 211, 211); // Light Gray**  **glBegin(GL\_POLYGON);**  **glVertex2f((12.3067892620237 - 13.00832) \* scaleFactor, (1.3589203311724 - circleCenterY) \* scaleFactor);**  **glVertex2f((12.3067892620237 - 13.00832) \* scaleFactor, (1.1647620496533 - circleCenterY) \* scaleFactor);**  **glVertex2f((11.77443140 - 13.00832) \* scaleFactor, (0.87544089 - circleCenterY) \* scaleFactor);**  **glVertex2f((11.77443140 - 13.00832) \* scaleFactor, (1.4853853 - circleCenterY) \* scaleFactor);**  **glEnd();**  **}**  **void white\_border() {**  **glColor3ub(255, 255, 255);**  **glBegin(GL\_POLYGON);**  **glVertex2f(1.81529, 1.006734);**  **glVertex2f(1.8232217261141, -0.5645063766661);**  **glVertex2f(14.5192132653909, -0.5738074693689);**  **glVertex2f(14.4741651352254, 1.0318433252581);**  **glEnd();**  **}**  **void wheel(float radius, float xc, float yc, float r, float g, float b) {**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 200; i++) {**  **glColor3f(r, g, b);**  **float pi = 3.1416;**  **float A = (i \* 2 \* pi) / 200;**  **float x = radius \* cos(A);**  **float y = radius \* sin(A);**  **glVertex2f(x + xc, y + yc);**  **}**  **glEnd();**  **}**  **void display() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **//////////////////////////////////**  **glPushMatrix();**  **glTranslatef(carPosition, 0.0f, 0.0f);**  **Car();**  **wheel(sqrt(1.905916878038), 4.22, 1.09, 0, 0, 0);**  **wheel(sqrt(1.8071686868979), 12.4754573237752, 1.2394829142391, 0, 0, 0);**  **wheel(sqrt(1.6305916878038), 4.22, 1.09, 105 / 255.0, 105 / 255.0, 105 / 255.0);**  **wheel(sqrt(1.4971686868979), 12.4754573237752, 1.2394829142391, 105 / 255.0, 105 / 255.0, 105 / 255.0);**  **white\_border();**  **wheel(sqrt(1.0743966196217), 4.229023150212, 1.1418732470549, 0, 0, 0);**  **wheel(sqrt(1.0606156155846), 12.4373260390525, 1.2647611733131, 0, 0, 0);**  **glPushMatrix();**  **glTranslatef(4.229023150212, 1.1418732470549, 0.0);**  **glRotatef(\_angle1, 0, 0, 1);**  **for (int i = 0; i < 10; i++) {**  **glRotatef(90.0 \* i, 0, 0, 1);**  **border();**  **}**  **glPopMatrix();**  **glPushMatrix();**  **glTranslatef(12.4373260390525, 1.2647611733131, 0.0);**  **glRotatef(\_angle1, 0, 0, 1);**  **for (int i = 0; i < 10; i++) {**  **glRotatef(90.0 \* i, 0, 0, 1);**  **border1();**  **}**  **glPopMatrix();**  **glPopMatrix();**  **//glPopMatrix();**  **glutSwapBuffers();**  **}**  **void update(int value) {**  **\_angle1 += 2.0f;**  **carPosition += carSpeed;**  **if (carPosition > 40.0f) {**  **carPosition = -40.0f;**  **glutPostRedisplay();**  **glutTimerFunc(20, update, 0);**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutCreateWindow("Shraboni Biswas Naboni-26");**  **glutInitWindowSize(800, 600);**  **gluOrtho2D(-40, 40, -20, 10);**  **glutDisplayFunc(display);**  **glutTimerFunc(20, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question-4**  Design a windmill with rotating blades |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h>**  **#include <GL/freeglut.h>**  **#include <cmath>**  **GLfloat bladeRotation = 0.0;**  **float bladeLength = 3.0;**  **void Blade()**  **{**  **glColor3ub(192, 192, 192);**  **glBegin(GL\_POLYGON);**  **glVertex2f(0.0, 0.0);**  **glVertex2f(bladeLength, 0.5);**  **glVertex2f(bladeLength - 0.5, 0.0);**  **glEnd();**  **}**  **void Body()**  **{**  **glColor3ub(181, 101, 29);**  **glBegin(GL\_QUADS);**  **glVertex2f(7.0, 7.0); // A**  **glVertex2f(7.0, 0.0); // B**  **glVertex2f(7.164499373788, 0.0); // D**  **glVertex2f(7.164499373788, 7.0); // C**  **glEnd();**  **}**  **void wheel(float radius, float xc, float yc, float r, float g, float b)**  **{**  **glColor3ub(192, 192, 192);**  **glBegin(GL\_POLYGON);**  **for (int i = 0; i < 200; i++) {**  **float pi = 3.1416;**  **float A = (i \* 2 \* pi) / 200;**  **float x = radius \* cos(A);**  **float y = radius \* sin(A);**  **glVertex2f(x + xc, y + yc);**  **}**  **glEnd();**  **}**  **void display()**  **{**  **glClearColor(1, 1, 1, 1.0);**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **glPushMatrix();**  **glTranslatef(7.0979109840932, 7.110603473, 0.0);**  **for (int i = 0; i < 4; i++)**  **{**  **glPushMatrix();**  **glRotatef(90.0 \* i + bladeRotation, 0, 0, 1);**  **Blade();**  **glPopMatrix();**  **}**  **glPopMatrix();**  **Body();**  **wheel(sqrt(0.175916878038), 7.0979109840932, 7.110603473, 128/255, 128/255, 128/255);**  **glutSwapBuffers();**  **glFlush();**  **}**  **void update(int value)**  **{**  **bladeRotation += 0.5;**  **if (bladeRotation > 360.0)**  **{**  **bladeRotation = -360.0;**  **}**  **glutPostRedisplay();**  **glutTimerFunc(16, update, 0);**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv);**  **glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA);**  **glutInitWindowSize(800, 600);**  **glutCreateWindow("Shraboni Biswas NAboni-26");**  **gluOrtho2D(-10, 20, -5, 15); glutDisplayFunc(display);**  **glutTimerFunc(0, update, 0);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |