**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 time that will be discussed in class VUES to the section named Lab Tak-4
* Must include resources for all the section in the table

|  |
| --- |
| **Question- 1**  Draw the scenario of a traffic signal |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **Code-**  #include <windows.h>  #ifdef \_\_APPLE\_\_  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #endif  #include <stdlib.h>  #include <GL/gl.h>  #include <GL/glut.h>  #include <math.h>  using namespace std;  void square(float a,float b,float c,float d){  glBegin(GL\_QUADS);  glVertex2f(a,b);  glVertex2f(c,b);  glVertex2f(c,d);  glVertex2f(a,d);  glEnd();  }  void tri(float a,float b,float c,float d,float e){  glBegin(GL\_TRIANGLES);  glVertex2f(a,b);  glVertex2f(c,b);  glVertex2f(d,e);  glEnd();  }  void circle(float a,float b,float d){  glLoadIdentity(); //Reset the drawing perspective  glMatrixMode(GL\_MODELVIEW);  glTranslatef(a, b, 0.0f);  glPushMatrix();  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++){  float pi=3.1416;  float A=(i\*2\*pi)/100;  float r=d;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x,y);  }  glPopMatrix();  glEnd();  }  void divider(){  glPushAttrib(GL\_ENABLE\_BIT);  glColor3f(1.0f,1.0f,1.0f);  glLineStipple(50, 0xAAAA);  glEnable(GL\_LINE\_STIPPLE);  glBegin(GL\_LINES);  glVertex3f(-9.9,-.6,-.5);  glVertex3f(9.9,-.6,-.5);  glEnd();  glPopAttrib();  }  void buildings(){  glColor3f(0.5f, 0.5f, 0.5f);  float a=-0.9;float b=-0.2;float c=-0.6;float d=0.8;  square(a,b,c,d);  square(a+0.4,b,c+0.4,d-0.4);  square(a+0.8,b,c+0.8,d-0.2);  square(a+1.2,b,c+1.2,d);  a=a+0.1;c=c-0.1;d=d-0.8;  glColor4f(1.0f, 1.0f, 0.0f, 0.0f);  square(a,b,c,d);  square(a+0.4,b,c+0.4,d);  square(a+0.8,b,c+0.8,d);  square(a+1.2,b,c+1.2,d);  a=-0.85;b=0.05;c=-0.65;d=0.233;  glColor3f(0.1f, 0.0f, 0.1f);  square(a,b,c,d);  square(a,b+0.25,c,d+0.25);  square(a,b+0.5,c,d+0.5);  a=-0.45;b=0.05;c=-0.25;d=0.2;  square(a,b,c,d);  square(a,b+0.18,c,d+0.18);  a=-0.05;b=0.05;c=0.15;d=0.2;  square(a,b,c,d);  square(a,b+0.18,c,d+0.18);  square(a,b+0.36,c,d+0.36);  a=0.35;b=0.05;c=0.55;d=0.2;  square(a,b,c,d);  square(a,b+0.18,c,d+0.18);  square(a,b+0.36,c,d+0.36);  square(a,b+0.54,c,d+0.54);  }  void tree(){  glColor3f(0.0f, 0.5f, 0.0f);  tri(0.83,0.2,0.99,0.915,0.35);  glColor3f(0.0f, 0.5f, 0.0f);  tri(0.83,0.25,0.99,0.915,0.45);  glColor3f(0.0f, 0.5f, 0.5f);  square(0.9,-0.2,0.93,0.2);  }  void car(){  glColor3f(0.0f, 0.0f, 0.0f);  glLoadIdentity(); //Reset the drawing perspective  glMatrixMode(GL\_MODELVIEW);  glTranslatef(0.0, 0.0, 0.0f);  glPushMatrix();  square(-0.75,-0.45,-0.05,-0.35);  glBegin(GL\_QUADS);  glVertex2f(-0.65,-0.35);  glVertex2f(-0.25,-0.35);  glVertex2f(-0.3,-0.3);  glVertex2f(-0.6,-0.3);  glEnd();  glColor3f(1.0f, 0.5f, 0.0f);  circle(-0.6,-0.45,0.035);  glColor3f(1.0f, 0.5f, 0.0f);  circle(-0.2,-0.45,0.035);  glEnd();  glPopAttrib();  glColor3f(0.0f, 0.5f, 1.0f);  glLoadIdentity(); //Reset the drawing perspective  glMatrixMode(GL\_MODELVIEW);  glTranslatef(0.8,-0.35,0.0f);  glPushMatrix();  square(-0.75,-0.45,-0.05,-0.35);  glBegin(GL\_QUADS);  glVertex2f(-0.65,-0.35);  glVertex2f(-0.25,-0.35);  glVertex2f(-0.3,-0.3);  glVertex2f(-0.6,-0.3);  glEnd();  glColor3f(1.0f, 0.5f, 0.0f);  circle(0.6,-0.8,0.035);  glColor3f(1.0f, 0.5f, 0.0f);  circle(0.2,-0.8,0.035);  glPopAttrib();  glEnd();  }  void trafficLight(){  glColor3f(0.0f, 0.5f, 0.5f);  square(0.7,-0.2,0.73,0.1);  square(0.68,0.1,0.75,0.4);  glColor4f(1.0f, 0.0f, 0.0f, 0.0f);  circle(0.715,0.15,0.025);  glColor3f(0.0f, 0.1f, 0.0f);  circle(0.715,0.25,0.025);  glColor4f(1.0f, 1.0f, 0.0f, 0.0f);  circle(0.715,0.35,0.025);  }  void display(void){  glClear (GL\_COLOR\_BUFFER\_BIT);  glClearColor(1.0f, 1.0f, 1.0f, 0.0f);  glLoadIdentity(); //Reset the drawing perspective  glMatrixMode(GL\_MODELVIEW);  glTranslatef(0.0, 0.0, 0.0f);  glPushMatrix();  //Road  glColor3f(0.556863f, 0.137255f, 0.137255f);  square(-1.0,-1.0,1.0,-0.2);  //Divider  divider();  //Tree  tree();  //Bulidings  buildings();  //Car  car();  //Traffic Light  glLoadIdentity(); //Reset the drawing perspective  glMatrixMode(GL\_MODELVIEW);  glTranslatef(0.0, 0.0, 0.0f);  glPushMatrix();  trafficLight();  glPopAttrib();  glEnd();  glFlush ();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize (1500, 1000);  glutCreateWindow ("Lab Task 004");  glutDisplayFunc(display);  glutMainLoop();  return 0;  } |
| **Output Screenshot (Full Screen)-** |