Source Code

```
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>
#include <math.h>

GLint keyl,keyr,flag=0;
float  counter=600.0, cnt=-
150.0,r1=0.0,g1=1.0,b1=0.0,bc=-260.0,tt=-
260.0,ms=400.0,ts=100.0,rs=100.0;
float
r2=0.0,g2=1.0,b2=1.0,r=1.0,g=0.0,b=0.0;
int c=1,d=1;

void  road();
void  grass();
void  grass2();
void  line1();
void  line2();
void  line3();
void  line4();
void  car();
void  truck();
void  bus();
void  sq();
void  text();
void  tc();
void  light();
void  light2();
void track1();


void *currentfont;
```

```
void setFont(void *font)
{
        currentfont=font;
}

void drawstring(float x,float y,float z,char
*string)
{
        char *ct;
        //displaying test
        glRasterPos3f(x,y,z);

        for(ct=string;*ct!='\0';ct++)
        {
          glColor3f(0.0,0.0,0.0);
          //renders a bitmap character
using OpenGL.

        glutBitmapCharacter(currentfont,*ct
);
        }
}

void initOpenGl()
{
  //Background Color

  glClearColor(0.2,0.6,0.99,0);
  //viewing volume
  glMatrixMode(GL_PROJECTION);
  //replace the current matrix with the
identity matrix
  glLoadIdentity();
  //define a 2D orthographic projection
matrix
  gluOrtho2D(0,700,0,500);
```

Source Code

```
   //specify which matrix is the current
matrix
   glMatrixMode(GL_MODELVIEW);
}

void text()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(160,130,0.0,"RUET BUS");
   glColor3f(1,1,1);
}

void credit()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(5,55,0.0,"Animated Road
Crossing Alert System");
   glColor3f(1,1,1);
   drawstring(5,55,0.0,"Developed By:
Ashadullah Shawon");
   glColor3f(1,1,1);
   drawstring(5,30,0.0,"Roll: 133009");
   glColor3f(1,1,1);
   drawstring(5,5,0.0,"CSE, RUET");
}

void msg()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(5,30,0.0,"This system can
alert pedestrian in complex lane too.");
   glColor3f(1,1,1);
   drawstring(5,10,0.0,"That means it can
detect all kinds of vehicles.");

}

void tech()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(5,30,0.0,"Object detection
technology is used in this system.");
   glColor3f(1,1,1);
   drawstring(5,10,0.0,"That's why this
system is reliable.");

}

void developer()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(200,355,0.0,"Animated Road
Crossing Alert System");
   glColor3f(1,1,1);
   drawstring(200,315,0.0,"Developed By:
Ashadullah Shawon");
   glColor3f(1,1,1);
   drawstring(200,275,0.0,"Roll: 133009");
   glColor3f(1,1,1);
   drawstring(200,235,0.0,"CSE, RUET");
}

void endmsg()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(200,355,0.0,"That's it. Be Safe
and Happy");
   glColor3f(1,1,1);
   drawstring(200,315,0.0,"Thank's all for
watching.");
   glColor3f(1,1,1);
   drawstring(200,275,0.0,"The End.");

}

void subtitle()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(5,55,0.0,"Lack of road crossing
alert system people are facing accident
frequently.");
```

Source Code

```
   glColor3f(1,1,1);
   drawstring(5,30,0.0,"This is very dangerous.");
}

void subtitle2()
{
   setFont(GLUT_BITMAP_HELVETICA_18);
   glColor3f(1,1,1);
   drawstring(5,55,0.0,"But road crossing alert system can help people to confirm safety and danger.");
   glColor3f(1,1,1);
   drawstring(5,30,0.0,"So that people can easily walk.");
}

void tc()
{
   glLoadIdentity();
   glColor3f(0.0,0.0,1.0);
   glBegin(GL_POLYGON);
   glVertex2f(310,190);
   glVertex2f(310,390);
   glVertex2f(340,390);
   glVertex2f(340,190);
   glEnd();

}


void night()
{
   glLoadIdentity();
   glColor3f(0.0,0.0,1.0);
   glBegin(GL_POLYGON);
   glVertex2f(310,190);
   glVertex2f(310,390);
   glVertex2f(340,390);
   glVertex2f(340,190);
   glEnd();

}
```

```
void light()
{
   glLoadIdentity();
   glColor3f(1.0,0.0,0.0);
   glBegin(GL_POLYGON);
   glVertex2f(315,330);
   glVertex2f(315,370);
   glVertex2f(335,370);
   glVertex2f(335,330);
   glEnd();
}

void light2()
{
   glLoadIdentity();
   glColor3f(0.0,1.0,0.0);
   glBegin(GL_POLYGON);
   glVertex2f(315,330);
   glVertex2f(315,370);
   glVertex2f(335,370);
   glVertex2f(335,330);
   glEnd();
}

void light3()
{
   glLoadIdentity();
   glColor3f(1.0,0.0,0.0);
   glBegin(GL_POLYGON);
   glVertex2f(315,280);
   glVertex2f(315,320);
   glVertex2f(335,320);
   glVertex2f(335,280);
   glEnd();
}

void light4()
{
   glLoadIdentity();
   glColor3f(0.0,1.0,0.0);
   glBegin(GL_POLYGON);
   glVertex2f(315,280);
   glVertex2f(315,320);
   glVertex2f(335,320);
```

Source Code

```
   glVertex2f(335,280);
   glEnd();
}

void window(int w1,int w2)
{
   glColor3f(0.0,0.0,0.0);
   glBegin(GL_POLYGON);
   glVertex2f(w1,160);
   glVertex2f(w1,185);
   glVertex2f(w2,185);
   glVertex2f(w2,160);
   glEnd();
}

void buswindow(int w1,int w2)
{
   glColor3f(0.0,0.0,0.0);
   glBegin(GL_POLYGON);
   glVertex2f(w1,180);
   glVertex2f(w1,205);
   glVertex2f(w2,205);
   glVertex2f(w2,180);
   glEnd();
}


void wheel(int x,int y)
{
  float th;
  glBegin(GL_POLYGON);
  glColor3f(0,0,0);
  //circle is 360 degree
  for(int i=0;i<360;i++)
  {
     //theta in radian
     th=i*(3.1416/180);
     //drawing circle with coordinates rcosth
and rsinth
     //x and y is for shifting to correct
position
     glVertex2f(x+20*cos(th),y+20*sin(th));
  }
```

```
    glEnd();

}

void road()
{
   glLoadIdentity();
   glColor3f(0.5,0.5,0.5);
   glBegin(GL_POLYGON);
   glVertex2f(0,95);
   glVertex2f(0,260);
   glVertex2f(800,260);
   glVertex2f(800,95);
   glEnd();
}

void grass()
{
   glLoadIdentity();
   glColor3f(0.0,0.5,0.1);
   glBegin(GL_POLYGON);
   glVertex2f(0,0);
   glVertex2f(0,95);
   glVertex2f(800,95);
   glVertex2f(800,0);
   glEnd();
}

void grass2()
{
   glLoadIdentity();
   glColor3f(0.0,0.5,0.1);
   glBegin(GL_POLYGON);
   glVertex2f(0,260);
   glVertex2f(0,300);
   glVertex2f(800,300);
   glVertex2f(800,260);
   glEnd();
}

void line1()
{
   glLoadIdentity();
   glColor3f(1.0,1.0,1.0);
```

Source Code

```
   glBegin(GL_LINE_LOOP);
   glVertex2f(0,190);
   glVertex2f(150,190);
   glEnd();

}

void line2()
{
   glLoadIdentity();
   glColor3f(1.0,1.0,1.0);
   glBegin(GL_LINE_LOOP);
   glVertex2f(200,190);
   glVertex2f(300,190);
   glEnd();

}

void line3()
{
   glLoadIdentity();
   glColor3f(1.0,1.0,1.0);
   glBegin(GL_LINE_LOOP);
   glVertex2f(350,190);
   glVertex2f(500,190);
   glEnd();

}

void line4()
{
   glLoadIdentity();
   glColor3f(1.0,1.0,1.0);
   glBegin(GL_LINE_LOOP);
   glVertex2f(550,190);
   glVertex2f(700,190);
   glEnd();

}

void car()
{

  //Bottom Part
```

```
glLoadIdentity();
counter=counter-0.05;
glColor3f(r1,g1,b1);
glTranslated(counter,80,0.0);
if(counter<-1000.00)
{

   c++;
   counter=700.0;
   //changing color
   if(c%2==0)
   {
      r1=1.0;
      g1=0.0;
      b1=0.0;

   }
   else if(c%3==0)
   {
      r1=0.0;
      g1=2.0+c;
      b1=1.0+c;
   }
   else if(c%5==0)
   {
      r1=1.0;
      g1=1.0;
      b1=0.0;
   }
   else
   {
      r1=0.0;
      g1=1.0;
      b1=0.0;
   }

}
glScaled(0.5,0.5,0.0);
glBegin(GL_POLYGON);
glVertex2f(100,100);
glVertex2f(100,160);
glVertex2f(450,160);
glVertex2f(450,100);
glEnd();
```

Source Code

```
  //Top Part
  glBegin(GL_POLYGON);
  glVertex2f(150,160);
  glVertex2f(200,200);
  glVertex2f(400,200);
  glVertex2f(450,160);

  glEnd();



  window(200,270);
  window(280,330);
  window(340,390);
  wheel(200,100);
  wheel(380,100);


}

void man(int p)
{
   glLoadIdentity();
   glColor3f(1.0,1.0,1.0);
   glScaled(0.4,0.4,0.4);
   // p is for switching man
   if(p==0)
   {
      ms=ms-0.01;
      glTranslated(ms,40,0.0);
   }
   else if(p==1)
   {
      ts=ts+0.29;
      glTranslated(280,ts,0.0);
   }
   else if(p==2)
   {


      ms=ms+0.10;
      glScaled(0.4,0.4,0.4);
      glTranslated(300,ms,0.0);
```

```
   }
   else if(p==3)
   {
      rs=rs+0.19;
      glTranslated(rs,40,0.0);
   }
   //head
   wheel(60,430);
   //body
   glBegin(GL_POLYGON);
   glVertex2f(40,290);
   glVertex2f(40,410);
   glVertex2f(80,410);
   glVertex2f(80,290);
   glEnd();

   glBegin(GL_LINE_LOOP);
   glVertex2f(50,260);
   glVertex2f(50,290);
   glEnd();

   glBegin(GL_LINE_LOOP);
   glVertex2f(70,260);
   glVertex2f(70,290);
   glEnd();

   glBegin(GL_POLYGON);
   glVertex2f(10,380);
   glVertex2f(10,390);
   glVertex2f(110,390);
   glVertex2f(110,380);
   glEnd();

}

void truck()
{

  //Bottom Part

  glLoadIdentity();
  glColor3f(r2,g2,b2);
```

Source Code

```
//speed variable
cnt=cnt+0.04;
//color changing
if(cnt>1300.00)
{
   cnt=-250.0;
   d++;
   if(d%2==0)
   {
      r2=r2+d;
      g2=0.0;
      b2=1.0;

   }
   else if(d%3==0)
   {
      r2=0.0;
      g2=3.0+d;
      b2=5.0+d;
   }
   else if(d%5==0)
   {
      r2=5.0;
      g2=0.0;
      b2=1.0;
   }
   else
   {
      r2=0.0;
      g2=1.0;
      b2=0.0;
   }

}
glTranslated(cnt,200,0.0);
glScaled(0.4,0.4,0.0);
glBegin(GL_POLYGON);
glVertex2f(100,100);
glVertex2f(100,160);
glVertex2f(450,160);
glVertex2f(450,100);
glEnd();

//Top Part
```

```
  glBegin(GL_POLYGON);
  glVertex2f(350,160);
  glVertex2f(350,200);
  glVertex2f(400,200);
  glVertex2f(450,160);

  glEnd();

  window(365,400);
  wheel(200,100);
  wheel(380,100);

}

void sq()
{
  glBegin(GL_POLYGON);
  glColor3f(0.9,0.2,0.1);
  glVertex2f(100,120);
  glVertex2f(100,170);
  glVertex2f(470,170);
  glVertex2f(470,120);
  glEnd();

}

void bus()
{

  glLoadIdentity();
  bc=bc+0.05;
  glColor3f(1.0,1.0,1.0);
  glTranslated(bc,180,0.0);
  //restart from position -260
  if(bc>1300.00)
  {
     bc=-260.0;

  }
  glScaled(0.4,0.4,0.0);
  glBegin(GL_POLYGON);
  glVertex2f(100,100);
  glVertex2f(100,220);
  glVertex2f(470,220);
```

Source Code

```
  glVertex2f(470,100);
  glEnd();

  buswindow(110,160);
  buswindow(170,220);
  buswindow(230,270);
  buswindow(280,330);
  buswindow(340,390);
  buswindow(400,450);
  wheel(200,100);
  wheel(380,100);

}

void rail()
{
   glLoadIdentity();
   glColor3f(0.0,0.0,0.0);
   glBegin(GL_POLYGON);
   glVertex2f(0,50);
   glVertex2f(0,60);
   glVertex2f(850,60);
   glVertex2f(850,50);
   glEnd();
}

void train()
{
  glLoadIdentity();
  //increasing speed variable
  tt=tt+0.05;
  glColor3f(0.7,0.0,0.0);
  //move object to x axis
  glTranslated(tt,10,0.0);
  if(tt>1900.00)
  {
     tt=-260.0;

  }
  glScaled(0.5,0.5,0.0);
  glBegin(GL_POLYGON);
  glVertex2f(90,100);
  glVertex2f(90,220);
  glVertex2f(470,220);
```

```
glVertex2f(470,100);
glEnd();

glBegin(GL_LINE_LOOP);
glVertex2f(20,150);
glVertex2f(90,150);
glEnd();

glBegin(GL_POLYGON);
glVertex2f(-490,100);
glVertex2f(-490,220);
glVertex2f(20,220);
glVertex2f(20,100);
glEnd();

glBegin(GL_LINE_LOOP);
glVertex2f(-530,150);
glVertex2f(-490,150);
glEnd();

glBegin(GL_POLYGON);
glVertex2f(-1000,100);
glVertex2f(-1000,220);
glVertex2f(-530,220);
glVertex2f(-530,100);
glEnd();


glBegin(GL_LINE_LOOP);
glVertex2f(-1030,150);
glVertex2f(-1000,150);
glEnd();

glBegin(GL_POLYGON);
glVertex2f(-1500,100);
glVertex2f(-1500,220);
glVertex2f(-1030,220);
glVertex2f(-1030,100);
glEnd();

buswindow(110,160);
buswindow(170,220);
buswindow(230,270);
buswindow(280,330);
```

Source Code

```
  buswindow(340,390);
  buswindow(400,450);

  buswindow(-470,-430);
  buswindow(-390,-350);
  buswindow(-310,-270);
  buswindow(-230,-190);
  buswindow(-150,-110);
  buswindow(-70,-30);

  buswindow(-970,-930);
  buswindow(-890,-850);
  buswindow(-810,-770);
  buswindow(-730,-690);
  buswindow(-650,-610);
  buswindow(-590,-550);

  buswindow(-1460,-1420);
  buswindow(-1380,-1340);
  buswindow(-1300,-1260);
  buswindow(-1220,-1180);
  buswindow(-1140,-1100);



  //wheel(200,100);
  //wheel(380,100);

}

void display()
{

 glClear(GL_COLOR_BUFFER_BIT);

 //introducing road crossing alert system
 road();
 grass();
 grass2();
 line1();
 line2();
 line3();
 line4();
 car();
 truck();
 bus();
 sq();
 text();
 tc();
 //red signal
 light();
 light3();
 subtitle2();
 //when car crosses the area green signal will be on and man can cross the road
 if(counter<-250)
 {
   light4();
   man(3);
 }
 //green signal when bus and truck are not in the area
 if(bc>650 && cnt>650)
 {
    light2();
    man(2);
 }




 glutSwapBuffers();
 glFlush();


}

void display1()
{

 glClear(GL_COLOR_BUFFER_BIT);
 //introducing train
 road();
 grass();
 grass2();
 line1();
 line2();
```

Source Code

```
  line3();
  line4();
  train();
  rail();
  truck();
  bus();
  sq();
  text();
  tc();
  //red signal
  light();
  light3();
  //green signal when train are not in the
area
  if(tt>1400)
  {
     light4();
  }
  //green signal when bus and track are not
in the area
  if(bc>650 && cnt>650)
  {
      light2();
  }
  tech();

  glutSwapBuffers();
  glFlush();

}

void display2()
{

  glClear(GL_COLOR_BUFFER_BIT);
  glClearColor(0.0,0.0,0.0,0);
  //night scene display
  road();
  grass();
  grass2();
  line1();
  line2();
  line3();
  line4();
  train();
  rail();
  car();
  truck();
  bus();
  sq();
  text();
  tc();
  light();
  light3();
  //green signal when train and car are not
in the area
  if(tt>1400 && counter<-250)
  {
     light4();
  }
  //green signal when bus and track are not
in the area
  if(bc>650 && cnt>650)
  {
      light2();
  }
  //describes the scene
  msg();

  glutSwapBuffers();
  glFlush();

}

void display0()
{

  glClear(GL_COLOR_BUFFER_BIT);
  //starting display
  road();
  grass();
  grass2();
  subtitle();
  line1();
  line2();
  line3();
  line4();
  car();
```

Source Code

```
 truck();
 bus();
 sq();
 text();

 //clashes with car when position of x is less
than 100
 if(counter<100)
 {
   man(1);

 }
 else
 {
   man(0);
 }

 glutSwapBuffers();
 glFlush();

}

void intro()
{
 //clears the window
 glClear(GL_COLOR_BUFFER_BIT);
 //developer information

 developer();

 glutSwapBuffers();
 glFlush();

}

void ending()
{
 //end message
 glClear(GL_COLOR_BUFFER_BIT);
 glClearColor(0.0,0.2,0.0,0);

 endmsg();

 glutSwapBuffers();
```

```
 //force execution of GL commands in finite
time
 glFlush();

}

void MyTimerFunc(int value);

int main(int argc, char **argv)
{
  glutInit(&argc,argv);

glutInitDisplayMode(GLUT_DOUBLE|GLUT_
RGBA|GLUT_DEPTH);
  //initializing window to 700*500
  glutInitWindowSize(700,500);
  //starting position of window
  glutInitWindowPosition(0,0);
  //Window title
  glutCreateWindow("Animated Road
Crossing Alert System");
  //initializing
  initOpenGl();
  //initializing display
  glutDisplayFunc(intro);
  //sets the global idle callback
  glutIdleFunc(intro);
  //switching display after 3 seconds
  glutTimerFunc(3000, MyTimerFunc, 0);
  //enters the GLUT event processing loop
  glutMainLoop();
  return 0;
}

void MyTimerFunc(int value)
{
  if (value == 0) // passed in in main
  {
   glutDisplayFunc(display0);
   glutIdleFunc(display0);

   // Change to a new display function in
25 seconds
   glutTimerFunc(25000, MyTimerFunc, 1);
```

Source Code

```
  }
  else if(value==1)
  {
    glutDisplayFunc(display);
    glutIdleFunc(display);
     //switching display after 50 seconds
    glutTimerFunc(50000, MyTimerFunc, 2);
  }
   else if(value==2)
  {
    glutDisplayFunc(display1);
    glutIdleFunc(display1);
    //switching display after 40 seconds
    glutTimerFunc(40000, MyTimerFunc, 3);

  }
   else if(value==3)
  {
    glutDisplayFunc(display2);
    glutIdleFunc(display2);
    //switching display after 40 seconds
    glutTimerFunc(40000, MyTimerFunc, 4);

  }
   else if(value==4)
  {
    glutDisplayFunc(ending);
    glutIdleFunc(ending);


  }

}
```