SIMULATION OF A VILLAGE

CG MINI PROJECT REPORT SUBMITTED BY

Jayesh Kumar
4NM17CS071
VI Semester, B Section

Joshni Princia Saldanha
4NM17CS072
VI Semester, B section

Jovita Andrews
4NM17CS073
VI Semester, B Section

Joyston Menezes
4NM17CS074
VI Semester, B section

UNDER THE GUIDANCE OF

Mr. Pradeep Kanchan
Assistant professor GD III
Department of Computer Science and Engineering

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Bachelor of Engineering in Computer Science & Engineering

from

Visvesvaraya Technological University, Belagavi

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution under VTU, Belgaum)
AICTE approved, (ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
NITTE -574 110, Udupi District, KARNATAKA.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering
B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

"Simulation of a Village" is a bonafide work carried out by Jayesh Kumar (4NM17CS071), Joshni Princia Saldanha (4NM17CS072), Jovita Andrews (4NM17CS073) and Joyston Menezes (4NM17CS074) in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2019-2020.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide Signature of HOD

ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, our sincere thanks to our beloved principal, Dr. Niranjan N. Chiplunkar for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We sincerely thank Dr. K.R. Udaya Kumar Reddy, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

We express our deep sense of gratitude and indebtedness to our guide Mr. Pradeep Kanchan, Assistant Professor GD III, Department of Computer Science and Engineering, for her inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We thank all the teaching and non-teaching staff members of the Computer Science and Engineering Department and our parents and friends for their honest opinions and suggestions throughout the course of our project.

Finally, we thank all those who have supported us directly or indirectly throughout the project and making it a grand success.

Jayesh Kumar
(4NM17CS071)

Joshni Princia Saldanha
(4NM17CS072)

Jovita Andrews
(4NM17CS073)

Joyston Menezes
(4NM17CS074)

ABSTRACT

The main theme behind the project is to use the basic concepts of the computer graphics to draw a package from the scratch. Concepts involved in this project are polygon drawing, color filling, the movement of objects and simulation of village. What we will learn from this project is how to build a package from scratch and basics of computer graphics by programming in OpenGL.

We are developing this using OpenGL. OpenGL is an application program interface (API) offering various functions to implement primitives, models and images. This offer functions to create and manipulate render lighting, coloring, viewing the models.

TABLE OF CONTENTS

INTRODUCTION

OpenGL is a graphical library which is an open-source project. OpenGL can be used to build even complex graphics including 3D shapes. The project aims at providing a 2D model of the **"Simulation of a Village"** which depicts the major features of the village.

An animation made in OpenGL using C programming. In this mini project, all the things, shapes, colors are drawn and filled from the scratch. There are several moving components in the project namely train, birds, clouds, rain, boat, aircraft, falling of fruits and numerous stationary components such as trees, water, houses, sun, moon etc.

IMPLEMENTATION DETAILS

The functions mentioned below are involved in creating the crucial components in the project.

1. Displays the first introductory window

```
void display()
{
  glClear(GL_COLOR_BUFFER_BIT);
  drawrectangle();
  glColor3f(1.0,0.0,0.0);
  glRasterPos2i(15,40);
  glRasterPos3f(16,40,0);
  Display_on_screen("SIMULATION OF A VILLAGE");
  glRasterPos3f(12,26,0);
  Display_on_screen("JAYESH KUMAR");
  glRasterPos3f(23,26,0);
  Display_on_screen("-");
  glRasterPos3f(26,26,0);
  Display_on_screen("4NM17CS071");
  glRasterPos3f(12,24,0);
  Display_on_screen("JOSHNI SALDANHA");
  glRasterPos3f(23,24,0);
  Display_on_screen("-");
  glRasterPos3f(26,24,0);
  Display_on_screen("4NM17CS072");
  glRasterPos3f(12,22,0);
  Display_on_screen("JOVITA ANDREWS");
  glRasterPos3f(23,22,0);
  Display_on_screen("-");
```

```
    glRasterPos3f(26,22,0);
    Display_on_screen("4NM17CS073");
    glRasterPos3f(12,20,0);
    Display_on_screen("JOYSTON MENEZES");
    glRasterPos3f(23,20,0);
    Display_on_screen("-");
    glRasterPos3f(26,20,0);
    Display_on_screen("4NM17CS074");
    glRasterPos3f(37,12,0);
    Display_on_screen("Next");
    glColor3f(1.0,0.0,0.0);
    glLineWidth(5);
    glBegin(GL_LINES);
    glVertex2i(16,39);
    glVertex2i(28,39);
    glEnd();
    glColor3f(1.0,0.0,0.0);
    glFlush();
 }
```

2. void plane() function displays the aircraft and void planeMove() function is used to display the movement of aircraft.

```
void Plane()
{
      glBegin(GL_POLYGON);
      glColor3ub(33,69,200);
      glVertex2f(88,90);
      glVertex2f(92,90);
      glVertex2f(93,94);
      glVertex2f(88,92);
```

```
        glVertex2f(86,90);
        glEnd();
        glColor3f(1,1,1);
        circle(.2,.3,88.5,91);
        circle(.2,.3,89.5,91);
        circle(.2,.3,90.5,91);
        glColor3f(1,0,0);
        circle(.2,.3,92.5,93.4);
}
void planemove()
{
      pl1 -= 1;
      if(pl1 < -100)
         pl1 = 5;
      glutPostRedisplay();
      pl2 += 1;
      if(pl2 >100)
      pl2 = -7;
      glutPostRedisplay();
}
void planeMove()
{
     planemove();
     glPushMatrix();
     glTranslated(pl1,0,0);
     Plane();
     glPopMatrix();
     glPushMatrix();
     glTranslated(pl2,0,0);
     Plane2();
     glPopMatrix();
}
```

3. void house() function not only displays the house but upon clicking it, one can see the inside view of it. The inside view contains components like the dining table with a couple of chairs, the clock which reciprocates the system time, a fan that can be switched on by pressing S which also allows the user to modulate the frequency of the speed with ↑ & ↓ buttons. Another component is the bedroom door which can be opened upon pressing O.

```
void house(void)
{
GLfloat mat_ambient[]={1,0,0,1};
GLfloat mat_specular[]={1,1,1,1};
GLfloat mat_diffuse[]={1,1,.7,1};
GLfloat mat_shininess[]={50};
matprop(mat_ambient,mat_diffuse,mat_specular,mat_shininess);
GLfloat lightIntensity4[]={.7,.7,.7,.7};
GLfloat light_position4[]={3,1,.5,1};
glLightfv(GL_LIGHT6,GL_POSITION,light_position4);
glLightfv(GL_LIGHT6,GL_DIFFUSE,lightIntensity4);
glEnable(GL_LIGHT6);
glPushMatrix();
glTranslated(0,.15,0);
//roof
glPushMatrix();
glTranslated(-.02*4,3.9,-.01*4-.25);
glScaled(1.5+.05,1.5,1.1);
wall(0.08);
glPopMatrix();
GLfloat ambient2[]={1,0,0,1};
GLfloat specular2[]={1,1,1,1};
GLfloat diffuse2[]={.7,1,0.8,1};
```

```
GLfloat shininess[]={50};
matprop(ambient2,diffuse2,specular2,shininess);
//floor
glPushMatrix();
glTranslated(-.02*3,-0.05,-.01*4);
glScaled(1.5+.01,1.5,1);
wall(0.08);
glPopMatrix();
GLfloat ambient1[]={1,0,0,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={1,1,.7,1};
GLfloat shininess1[]={50};
matprop(ambient1,diffuse1,specular1,shininess1);
//left wall
glPushMatrix();
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//right wall
glPushMatrix();
glTranslated(6,0,0);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
 //back wall
glPushMatrix();
glTranslated(-.08,0,0);
glScaled(1.5+.02,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
```

```
//room vertical wall
glPushMatrix();
glTranslated(4,0,0);
glScaled(1,1,.5);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//room horizontal wall
glPushMatrix();
glTranslated(4.4,0,2);
glScaled(.4,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//wall above the room door
glPushMatrix();
glTranslated(4,3,2);
glScaled(.11,.25,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//left room horizontal wall
glPushMatrix();
glTranslated(0,0,2);
glScaled(.4,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//lroom vertical wall
glPushMatrix();
glTranslated(1.6,0,0);
glScaled(1,1,.35);
```

```
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//entrance room right wall
glPushMatrix();
glTranslated(1.6,0,2.59);
glScaled(1,1,.35);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();

window();
fan();
cot(.6,.9,.06,.35,.009);
diningtable();
myclock();
solar();

GLfloat ambient[]={1,0.5,.5,1};
GLfloat specular[]={1,1,1,1};
GLfloat diffuse[]={1,.5,.5,1};
matprop(ambient,diffuse,specular,mat_shininess);
//below room door
glPushMatrix();
glTranslated(4,0,(2-.025));
glRotated(romo,0,1,0);
glTranslated(-4,0,-(2-.025));
glPushMatrix();
glTranslated(4,0,2);
glScaled(.099,.75,1);
glRotated(-90.0,1,0,0);
wall(0.01);
```

```
glPopMatrix();
glPushMatrix();
glTranslated(4.01,0,2-.025);
glScaled(.5,1,.6);
glRotated(-90,1,0,0);
gluCylinder(Cylinder, 0.05, 0.05, 3, 16, 16);
glPopMatrix();
glPopMatrix();
glPopMatrix();
glFlush();
}
```

4. void tree() is the function that draws a tree in the window frame.

```
void tree(float x1, float y1)
{
   float x2 = x1+5,y2=y1, x3=x1+2.5,y3=y1+6;
   glColor3ub(11, 50, 11);
   glBegin(GL_TRIANGLES);
   glVertex2d(x1, y1);
   glVertex2d(x2, y2);
   glVertex2d(x3, y3);
    glEnd();
    glBegin(GL_QUADS);
    glVertex2d(x1+2, y1);
    glVertex2d(x1, y1-5);
    glVertex2d(x2, y1-5);
    glVertex2d(x2-2, y1);
    glVertex2d(x1+2, y1-5);
    glVertex2d(x1, y1-10);
```

```
    glVertex2d(x2, y1-10);
    glVertex2d(x2-2, y1-5);
    glColor3ub(68, 43, 2);
    glVertex2d(x1+2, y1-10);
    glVertex2d(x1+1.5, y1-18);
    glVertex2d(x2-1.5, y1-18);
    glVertex2d(x2-2, y1-10);
    glEnd();
}
```

5. void train() is function that helps to draw the train component and also helps it to move.

```
void train()
{
  glBegin(GL_QUADS);
  //back cabin
  glColor3ub(58, 90, 145);
  //top
  glVertex2d(61,55);
  glVertex2d(70,55);
  glVertex2d(69,59);
  glVertex2d(62,59);
  //front
  glVertex2d(61,55);
  glVertex2d(61,52);
  glVertex2d(70,52);
  glVertex2d(70,55);
  //front plate
  glColor3ub(225, 43, 145);
  glVertex2d(61.2,52);
```

```
glVertex2d(61.2,48);
glVertex2d(69.8,48);
glVertex2d(69.8,52);
//front plate small for aesthetics
glColor3ub(22, 23, 145);
glVertex2d(63,51);
glVertex2d(63,49);
glVertex2d(68.4,49);
glVertex2d(68.4,51);
//joint
glColor3f(1,0,0);
glVertex2d(65.1,48);
glVertex2d(66.1,48);
glVertex2d(66,46);
glColor3f(1,.9,.30);
glVertex2d(65,46);
//joint shadow
glColor3f(1,.9,0);
glVertex2d(64.8,48);
glVertex2d(65.1,48);
glVertex2d(65,46);
glVertex2d(64.7,46);
//front cabin
glColor3ub(58, 90, 145);
//top
glVertex2d(61,43);
glVertex2d(70,43);
glVertex2d(69,46);
glVertex2d(62,46);
//front
glVertex2d(61,43);
glVertex2d(61,39);
```

```
    glVertex2d(70,39);
    glVertex2d(70,43);
    //front plate
    glColor3ub(225, 43, 145);
    glVertex2d(61.2,40);
    glVertex2d(61.2,35);
    glVertex2d(69.8,35);
    glVertex2d(69.8,40);
    //front plate small for aesthetics
    glColor3ub(22, 23, 145);
    glVertex2d(63,39);
    glVertex2d(63,36);
    glVertex2d(68.4,36);
    glVertex2d(68.4,39);
    glEnd();
    //top plate front cabin
    glColor3f(1,0,0);
    circle(0.5,0.7,62,42);
    circle(0.5,0.7,69,42);
    //front plate
    glColor3f(1,1,1);
    circle(0.5,0.7,64,37.5);
    circle(0.5,0.7,65.5,37.5);
    circle(0.5,0.7,67.3,37.5);
    //rear cabin
    glColor3f(1,1,1);
    circle(0.3,0.5,63.5,50.2);
    circle(0.3,0.5,65,50.2);
    circle(0.3,0.5,66.5,50.2);
    circle(0.3,0.5,67.7,50.2);
}
```

6. void boat() is the function that draws the boat component and also helps it to move upon clicking the boat and returning back to the same position.

```
void boat(float x1, float y1, float boatLength)
{
   float x2 = x1 + boatLength, y2 = y1, x3 = x2+(boatLength/2)-1,y3 =
y2+boatLength/2, x4 = x2, y4 = y3-1, x5 = x1, y5 = y4, x6=x1-boatLength/2-
1,y6=y3;
      glBegin(GL_POLYGON);
       glColor3d(1,0,0);
        glColor3ub(85, 28, 22);
       glVertex2d(x1,y1);
       glVertex2d(x2,y2);
       glVertex2d(x3,y3);
       glVertex2d(x4,y4);
       glVertex2d(x5,y5);
       glVertex2d(x6,y6);
       glEnd();
       glBegin(GL_POLYGON);
       glColor3d(1,1,1);
       glColor3ub(160, 155, 136);
       glVertex2d(x3,y3);
       glVertex2d(x4,y4);
       glVertex2d(x5,y5);
       glVertex2d(x6,y6);
       glVertex2d(x1,y6+.7);
       glVertex2d(x2,y3+.7);
       glEnd();
       glColor3d(.90,1,1);
       glRectf(x5-.2,y5+1,x5+.2,y5+12);
       glBegin(GL_TRIANGLES);
       glColor3d(0,.1,1);
```

```
        glVertex2d(x5+.5,y5+11);
        glVertex2d(x5+.5,y5+2.5);
        glVertex2d(x4-3,y5+2.3);
        glEnd();
        glBegin(GL_LINES);
        glColor3d(1,0,0);
        glLineWidth(20);
        glVertex2d(x5+.5,y5+11);
        glVertex2d(x5+.2,y5+11);
        glVertex2d(x5+.5,y5+2.5);
        glVertex2d(x5+.2,y5+2.5);
        glVertex2d(x4-3,y5+2.3);
        glVertex2d(x4-2.8,y5+2);
        glEnd();
}
```

CONCLUSION

We have successfully completed the animation of village scenario with several stationary and moving components with the OpenGL tool implemented in CodeBlocks. OpenGL supports enormous flexibility in the design and the use of OpenGL graphics programs. The presence of many built in classes methods take care of much functionality and reduce the job of coding as well as makes the implementation simpler. We have implemented the project making it user-friendly and error free as possible.

REFERENCES

https://docs.microsoft.com/en-us/windows/win32/opengl/gl-functions

https://www.opengl.org/resources/libraries/glut/spec3/spec3.html

https://lazyfoo.net/tutorials/OpenGL/index.php

https://www.opengl.org/resources/libraries/

APPENDIX

Figure1: NAME USN front window

Figure 2: Village scenario window with moving components namely boat, train, birds, clouds & aircraft and Stationary components namely trees, sun, sky & water

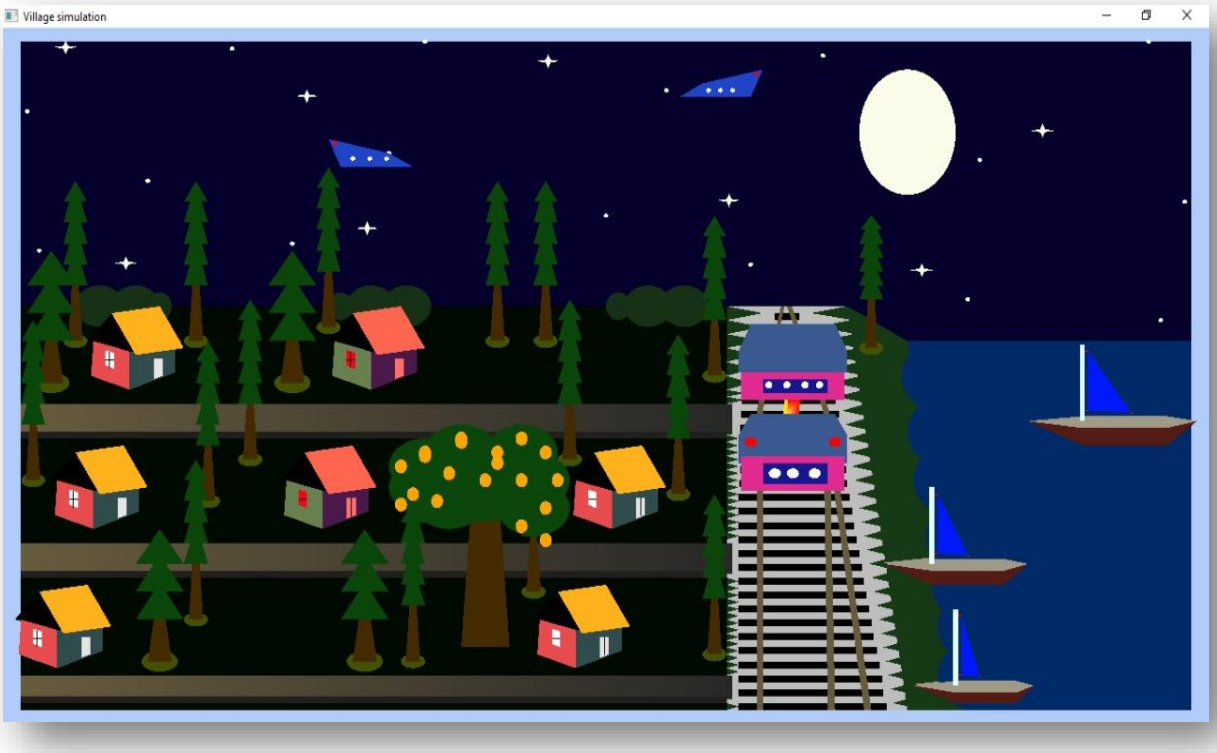

Figure 3: Night Mode

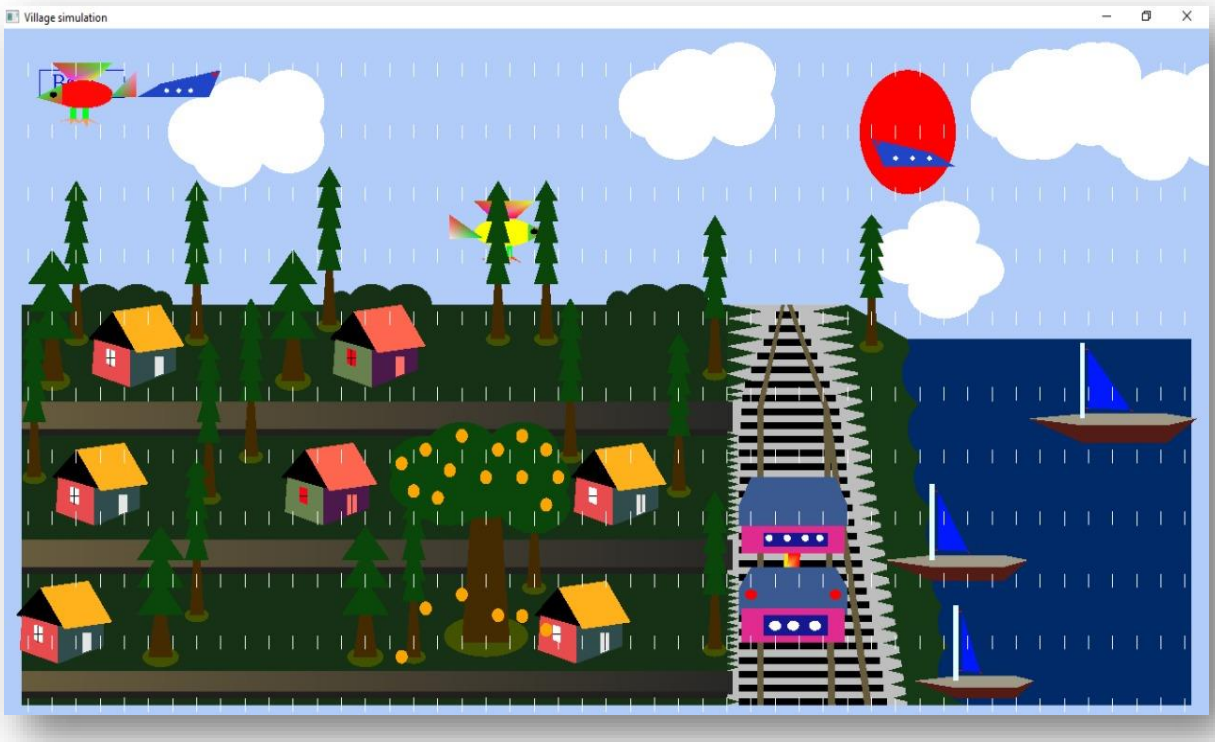

Figure 4: Rain & Day Mode

Figure 5: Rain & Night Mode

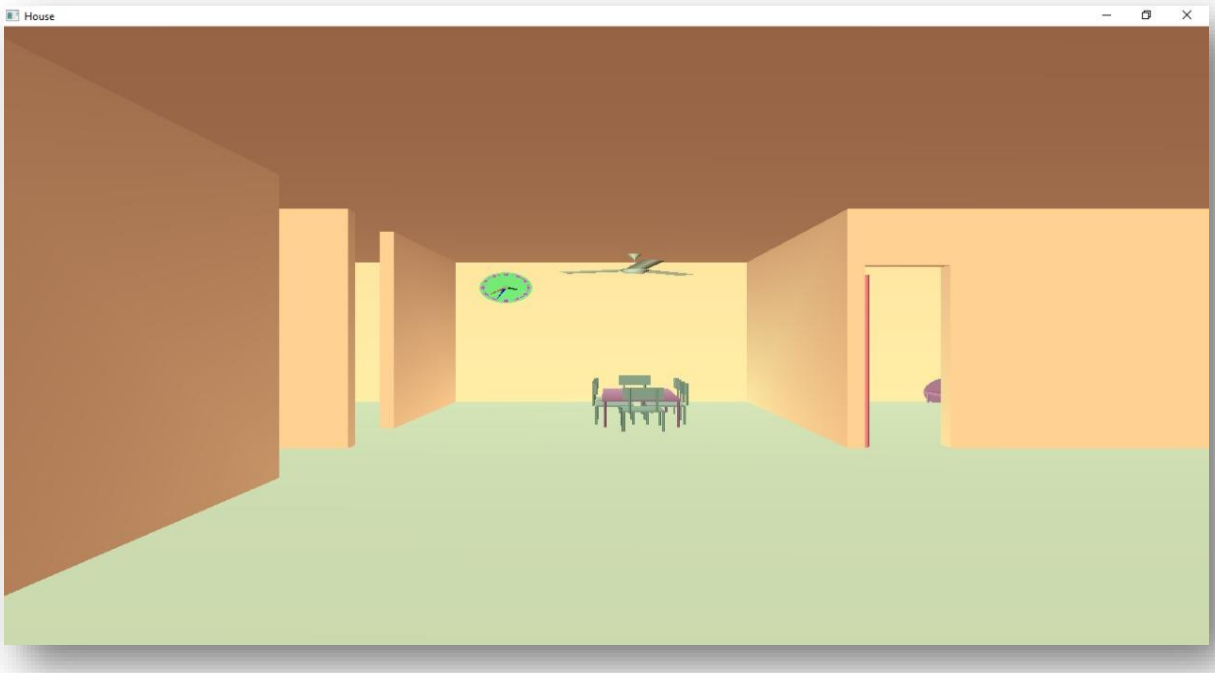

Figure 6: House view with a clock that shows the system time, dining table, moving components namely door of the bedroom & fan