American International University Bangladesh

Computer Graphics

Course Code: CSC4118

Fall Semester 2023-24

Project Report

[Five-time Prayer Weather Scenario]

Under the Guidance of

Shahnaj Parvin

Lecturer

Department of Computer Science, FST

I. Group Member Details

Group Member Name	ID
1.MD SHANJID HASAN ABIR	22-47202-1
2.MD TANVIR RAHMAN MOLLA	22-46052-1
3.MD ASAD-UZ-ZAMAN	22-46582-1
4.S.M. SAYED ARIFIN OMI	22-46613-1
Team Leader Name: MD. SHANJID HASAN ABIR	22-47202-1
Section: J	

II. Table of Contents:

1. Introduction:

Background:

Muslim people pray five times a day. The five daily prayers are offered at five fixed times of the day. Every time the environment is different. In this project we have shown that scenario. A mosque located on the bank of a river, a river, mountains, and natural scenario are presented here. In this scenario there are also some boats and ships underway in river, some birds are flying, and clouds are moving in the sky. In this project, there are 5 transitions depending on the five scenario including Fajar, Johar, Asar, Maghrib and Esha. 5 different keyboard keys are used to show these transitions which are F for Fajar, J for Johar, A for Asar, M for Maghrib and E for Esha. Here in this project, the clouds and boats will keep moving whenever we run the project.

Motivation:

The main motivation behind this project is to show the significance of five-time prayers. The main duty of every Muslim is to perform his/her duty towards Allah. In this project we tried to show the varieties and beauties of the weather and other scenarios in the specific five-time prayers. Besides, the project also tried to explore the beauty of nature and objects during these five-time prayers.

2. List of objects:

- A mosque
- River
- Mountain
- Cloud
- Bird
- Sun & moon
- Steamer & boat
- Tree
- Table & tool
- A wharf
- Bench

3. Project Requirements

Software Requirements:

- **Operating System:** GLUT is typically available on various operating systems, including Windows, Linux, and macOS.
- **Development Environment:** A programming IDE or text editor that supports C/C++ programming.
- **OpenGL Library:** GLUT is an auxiliary library for OpenGL, so you'll need the OpenGL library installed on your system.
- **GLUT library:** Install the GLUT library itself, which includes header files and libraries required for linking.

Hardware Requirements:

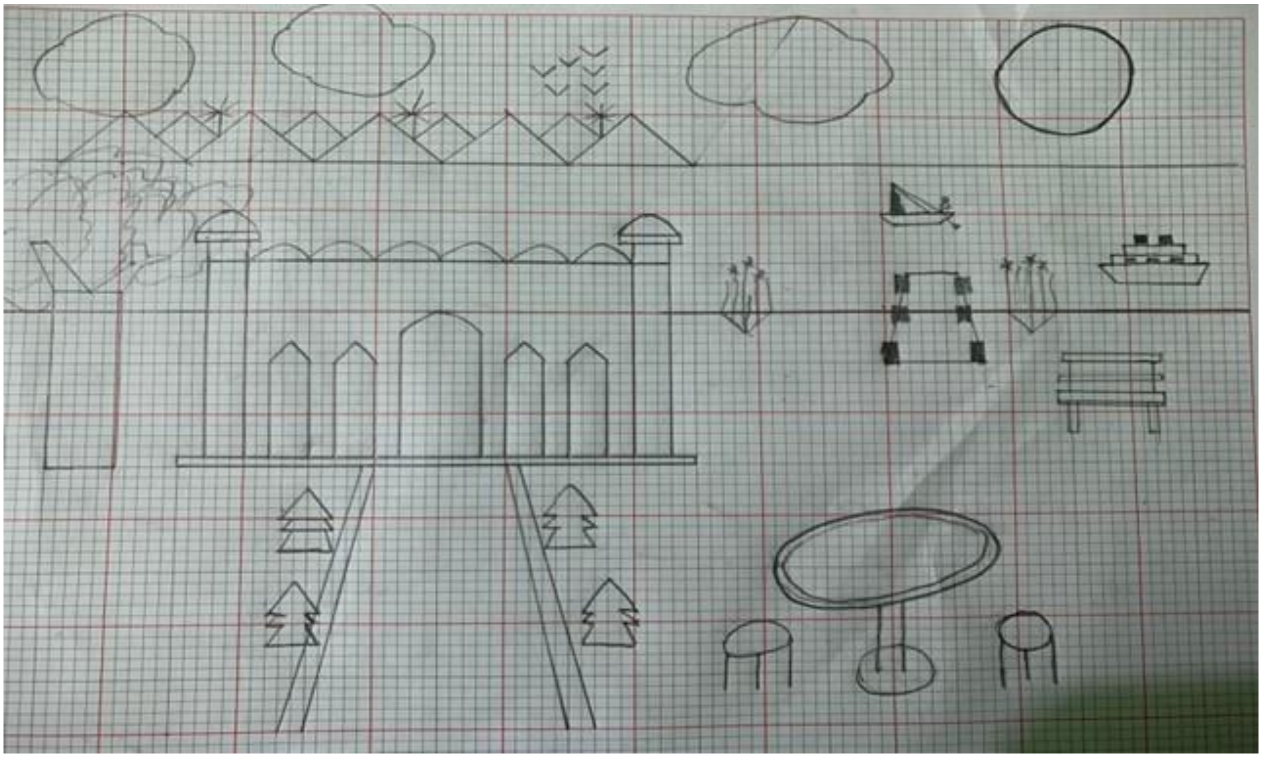
- **Computer:** A desktop or laptop capable of running the chosen operating system and supporting the OpenGL/GLUT requirement.
- **Graphics Card:** While not explicitly demanding, for better performance in rendering and manipulating graphics, having a dedicated graphics card can be beneficial.

Here are some requirements of this project:

1. **User Interface (UI):**
    - Design an intuitive and user-friendly interface for users to interact with the virtual environment.
2. **Graphics Rendering:**
    - Implement graphics rendering techniques to create realistic representations of weather conditions, including lighting, shading, and atmospheric effects.
    - Use textures and visual elements to animate various weather phenomena such as sunlight, clouds, air flow.
3. **Time-of-Day Transitions:**
    - Implement smooth transitions between different times of the day to animate the changing lighting conditions from dawn to dusk.
4. **Educational Content:**
    - Integrate educational content that provides information about the significance of the five daily prayers in Islam.
5. **Accessibility:**
    - Ensure that the project is accessible to a diverse audience by considering factors such as clear visuals and options for users with different abilities.
6. **Performance Optimization:**

- Optimize graphics rendering to ensure smooth performance, especially if the project is intended for use on a variety of devices.

4. Design



5. Functions to represent each object:

- **Mosque:**

```
glBegin(GL_QUADS);//Mosque

            glColor3ub(168, 102, 50);
            glVertex2f(-0.75f, -0.20f);
            glVertex2f(0.20f, -0.20f);
            glVertex2f(0.20f, 0.50f);
            glVertex2f(-0.75f, 0.50f);
            glEnd();
            glBegin(GL_QUADS);//Mosque upper border
            glColor3f(0.54f, 0.21f, 0.060f);
            glVertex2f(-0.75f, 0.50f);
            glVertex2f(0.20f, 0.50f);
```

```
    glVertex2f(0.20f, 0.515f);
    glVertex2f(-0.75f, 0.515f);
    glEnd();
glBegin(GL_QUADS);//Mosque bottom border
    glColor3f(0.54f, 0.21f, 0.060f);
    glVertex2f(-0.825f, -0.25f);
    glVertex2f(0.275f, -0.25f);
    glVertex2f(0.275f, -0.20f);
    glVertex2f(-0.825f, -0.20f);
    glEnd();

    glBegin(GL_POLYGON);//Big gate outside
    glColor3f(0.59, 0.27f, 0.080f);
    glVertex2f(-0.35f, -0.20f);
    glVertex2f(-0.20f, -0.20f);
    glVertex2f(-0.20f, 0.20f);
    glVertex2f(-0.225f, 0.30f);
    glVertex2f(-0.275f, 0.35f);
    glVertex2f(-0.325f, 0.30f);
    glVertex2f(-0.35f, 0.20f);
    glEnd();

    glBegin(GL_POLYGON);//Big gate inside
    glColor3f(0.87f, 0.68f, 0.45f);
    glVertex2f(-0.325f, -0.20f);
    glVertex2f(-0.21f, -0.20f);
    glVertex2f(-0.21f, 0.20f);
    glVertex2f(-0.228f, 0.28f);
    glVertex2f(-0.275f, 0.32f);
    glVertex2f(-0.31f, 0.28f);
    glVertex2f(-0.325f, 0.20f);
    glEnd();

    glBegin(GL_POLYGON);//right 1st gate outside
    glColor3f(0.59, 0.27f, 0.080f);
    glVertex2f(-0.15f, -0.20f);
    glVertex2f(-0.05f, -0.20f);
    glVertex2f(-0.05f, 0.15f);
    glVertex2f(-0.075f, 0.2125f);
    glVertex2f(-0.10f, 0.25f);
    glVertex2f(-0.125f, 0.2125f);
    glVertex2f(-0.15f, 0.15f);
    glEnd();
```

```
glBegin(GL_POLYGON);//right 1st gate inside
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(-0.135f, -0.20f);
glVertex2f(-0.058f, -0.20f);
glVertex2f(-0.058f, 0.15f);
glVertex2f(-0.085f, 0.2125f);
glVertex2f(-0.10f, 0.225f);
glVertex2f(-0.112f, 0.2125f);
glVertex2f(-0.135f, 0.15f);
glEnd();

glBegin(GL_POLYGON);//right last gate outside
glColor3f(0.59, 0.27f, 0.080f);
glVertex2f(0.0f, -0.20f);
glVertex2f(0.10f, -0.20f);
glVertex2f(0.10f, 0.15f);
glVertex2f(0.075f, 0.2125f);
glVertex2f(0.050f, 0.25f);
glVertex2f(0.025f, 0.2125f);
glVertex2f(0.0f, 0.15f);
glEnd();

glBegin(GL_POLYGON);//right last gate inside
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(0.015f, -0.20f);
glVertex2f(0.092f, -0.20f);
glVertex2f(0.092f, 0.15f);
glVertex2f(0.065f, 0.2125f);
glVertex2f(0.050f, 0.225f);
glVertex2f(0.037f, 0.2125f);
glVertex2f(0.015f, 0.15f);
glEnd();

glBegin(GL_POLYGON);//left 1st gate outside
glColor3f(0.59, 0.27f, 0.080f);
glVertex2f(-0.50f, -0.20f);
glVertex2f(-0.40f, -0.20f);
glVertex2f(-0.40f, 0.15f);
glVertex2f(-0.425f, 0.2125f);
glVertex2f(-0.45f, 0.25f);
glVertex2f(-0.475f, 0.2125f);
glVertex2f(-0.50f, 0.15f);
```

```
glEnd();

glBegin(GL_POLYGON);//left 1st gate inside
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(-0.492f, -0.20f);
glVertex2f(-0.415f, -0.20f);
glVertex2f(-0.4155f, 0.15f);
glVertex2f(-0.437f, 0.2125f);
glVertex2f(-0.45f, 0.225f);
glVertex2f(-0.465f, 0.2125f);
glVertex2f(-0.492f, 0.15f);
glEnd();

glBegin(GL_POLYGON);//left last gate outside
glColor3f(0.59, 0.27f, 0.080f);
glVertex2f(-0.65f, -0.20f);
glVertex2f(-0.55f, -0.20f);
glVertex2f(-0.55f, 0.15f);
glVertex2f(-0.575f, 0.2125f);
glVertex2f(-0.60f, 0.25f);
glVertex2f(-0.625f, 0.2125f);
glVertex2f(-0.65f, 0.15f);
glEnd();

glBegin(GL_POLYGON);//left last gate inside
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(-0.642f, -0.20f);
glVertex2f(-0.565f, -0.20f);
glVertex2f(-0.565f, 0.15f);
glVertex2f(-0.587f, 0.2125f);
glVertex2f(-0.60f, 0.225f);
glVertex2f(-0.615f, 0.2125f);
glVertex2f(-0.642f, 0.15f);
glEnd();


glBegin(GL_QUADS);//left minar
glColor3ub( 143, 81, 34);
glVertex2f(-0.80f, -0.20f);
glVertex2f(-0.70f, -0.20f);
glVertex2f(-0.70f, 0.55f);
glVertex2f(-0.80f, 0.55f);
glEnd();
```

```
glBegin(GL_QUADS);//left miner border
glColor3f(0.54f, 0.21f, 0.060f);
glVertex2f(-0.825f, 0.25f);
glVertex2f(-0.675f, 0.25f);
glVertex2f(-0.675f, 0.30f);
glVertex2f(-0.825f, 0.30f);
glEnd();
glBegin(GL_QUADS);
glColor3f(0.54f, 0.21f, 0.060f);
glVertex2f(-0.825f, 0.525f);
glVertex2f(-0.675f, 0.525f);
glVertex2f(-0.675f, 0.55f);
glVertex2f(-0.825f, 0.55f);
glEnd();
glBegin(GL_POLYGON);//window
glColor3f(0.59, 0.27f, 0.080f);
glVertex2f(-0.78f, 0.30f);
glVertex2f(-0.72f, 0.30f);
glVertex2f(-0.72f, 0.45f);
glVertex2f(-0.75f, 0.475f);
glVertex2f(-0.78f, 0.45f);
glEnd();

int t;// Right 2nd gombuj

GLfloat x7=0.11f; GLfloat h7= 0.515f; GLfloat Gr5 = 0.07f;
int Gtringle25=40;

GLfloat tp25 =1.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(161, 82, 26);
glVertex2f (x7,h7);
for(t= 0;t<=Gtringle25; t++)
{
glVertex2f (
x7+(Gr5*cos(t*tp25/Gtringle25)),
h7+(Gr5*sin(t*tp25/Gtringle25))
);
}
glEnd ();

glBegin(GL_QUADS);//right minar
glColor3ub( 143, 81, 34);
```

```
    glVertex2f(0.15f, -0.20f);
    glVertex2f(0.25f, -0.20f);
    glVertex2f(0.25f, 0.55f);
    glVertex2f(0.15f, 0.55f);
    glEnd();
    glBegin(GL_QUADS);//right miner border-1
    glColor3f(0.54f, 0.21f, 0.060f);
    glVertex2f(0.125f, 0.25f);
    glVertex2f(0.275f, 0.25f);
    glVertex2f(0.275f, 0.30f);
    glVertex2f(0.125f, 0.30f);
    glEnd();
    glBegin(GL_QUADS);//right miner border -2
    glColor3f(0.54f, 0.21f, 0.060f);
    glVertex2f(0.125f, 0.525f);
    glVertex2f(0.275f, 0.525f);
    glVertex2f(0.275f, 0.55f);
    glVertex2f(0.125f, 0.55f);
    glEnd();
    glBegin(GL_POLYGON);//window
    glColor3f(0.59, 0.27f, 0.080f);
    glVertex2f(0.170f, 0.30f);
    glVertex2f(0.230f, 0.30f);
    glVertex2f(0.230f, 0.45f);
    glVertex2f(0.20f, 0.475f);
    glVertex2f(0.170f, 0.45f);
    glEnd();

int l;//left minar gombuj

    GLfloat p17=-0.751f; GLfloat q17= 0.55f; GLfloat r17= 0.07f;
    int tringle17=40;

    GLfloat tp17 =1.0f * PI ;
    glBegin (GL_TRIANGLE_FAN);
    glColor3ub(161, 82, 26);
    glVertex2f (p17,q17);
    for(l= 0;l<=tringle17; l++)
    {
    glVertex2f (
    p17+(r17*cos(l*tp17/tringle17)),
    q17+(r17*sin(l*tp17/tringle17))
    );
}
glEnd ();

int s;// Right minar gombuj

GLfloat x18=0.2f; GLfloat h18= 0.55f; GLfloat Gr18 = 0.07f;
int Gtringle18=40;

GLfloat tp18 =1.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(161, 82, 26);
glVertex2f (x18,h18);
for(s= 0;s<=Gtringle18; s++)
{
glVertex2f (
x18+(Gr18*cos(s*tp18/Gtringle18)),
h18+(Gr18*sin(s*tp18/Gtringle18))
);
}
glEnd ();

//2nd left gombuj
GLfloat x1=-0.604f; GLfloat h1= 0.515f; GLfloat ra = 0.07f;
int Gtringle=40;

GLfloat tp19 =1.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(161, 82, 26);
glVertex2f (x1,h1);
for(l= 0;l<=Gtringle; l++)
{
glVertex2f (
x1+(ra*cos(l*tp19/Gtringle)),
h1+(ra*sin(l*tp19/Gtringle))
);
}
glEnd ();

int m;// 3rd left gombuj

GLfloat x2=-0.460f; GLfloat h2= 0.515f; GLfloat Gr = 0.07f;
int Gtringle1=40;
```

```
GLfloat tp20 =1.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(161, 82, 26);
glVertex2f (x2,h2);
for(m= 0;m<=Gtringle1; m++)
{
glVertex2f (
x2+(Gr*cos(m*tp20/Gtringle1)),
h2+(Gr*sin(m*tp20/Gtringle1))
);
}
glEnd ();

int n;// 4th left gombuj

GLfloat x3=-0.315f; GLfloat h3= 0.515f; GLfloat Gr1 = 0.07f;
int Gtringle21=40;

GLfloat tp21 =1.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(161, 82, 26);
glVertex2f (x3,h3);
for(n= 0;n<=Gtringle21; n++)
{
glVertex2f (
    x3+(Gr1*cos(n*tp21/Gtringle21)),
    h3+(Gr1*sin(n*tp21/Gtringle21))
);
}
glEnd ();

int p;// 5th left gombuj

GLfloat x4=-0.170f; GLfloat h4= 0.515f; GLfloat Gr2 = 0.07f;
int Gtringle22=40;

GLfloat tp22=1.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(161, 82, 26);
glVertex2f (x4,h4);
for(p= 0;p<=Gtringle22; p++)
{
glVertex2f (
x4+(Gr2*cos(p*tp22/Gtringle22)),
h4+(Gr2*sin(p*tp22/Gtringle22))
);
}
glEnd ();

int q;// 6th left gombuj

GLfloat x5=-0.028f; GLfloat h5= 0.515f; GLfloat Gr3 = 0.07f;
int Gtringle23=40;

GLfloat tp23 =1.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(161, 82, 26);
glVertex2f (x5,h5);
for(q= 0;q<=Gtringle23; q++)
{
glVertex2f (
x5+(Gr3*cos(q*tp23/Gtringle23)),
h5+(Gr3*sin(q*tp23/Gtringle23))
);
}
glEnd ();
```

- **Bench ,table & tools:**

```
int a;//under stand circle

   GLfloat p100=0.63f; GLfloat q100= -0.83f; GLfloat r100 = 0.08f;
   int tringle100=40;

   GLfloat tp100 =2.0f * PI  ;

   glBegin (GL_TRIANGLE_FAN);
   glColor3ub ( 160,157,156);
   glVertex2f (p100,q100);
   for(a= 0;a<=tringle100; a++)
   {
      glVertex2f (
               p100+(r100*cos(a*tp100/tringle100)),
               q100+(r100*sin(a*tp100/tringle100))
               );
}
glEnd ();


glBegin(GL_QUADS);//toll stand
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(0.425f, -0.83f);
glVertex2f(0.435f, -0.83f);
glVertex2f(0.435f, -0.65f);
glVertex2f(0.425f, -0.65f);
glEnd();
glBegin(GL_QUADS);//toll stand
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(0.39f, -0.78f);
glVertex2f(0.40f, -0.78f);
glVertex2f(0.40f, -0.65f);
glVertex2f(0.39f, -0.65f);
glEnd();
glBegin(GL_QUADS);//toll stand
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(0.46f, -0.78f);
glVertex2f(0.47f, -0.78f);
glVertex2f(0.47f, -0.65f);
glVertex2f(0.46f, -0.65f);
glEnd();

GLfloat p30=0.43f; GLfloat q30= -0.63f; GLfloat r30 = 0.05f;
int tringle30=40;

GLfloat tp30 =2.0f * PI  ;

glBegin (GL_TRIANGLE_FAN);
glColor3ub ( 160,157,156);
glVertex2f (p30,q30);
for(a= 0;a<=tringle30; a++)
{
    glVertex2f (p30+(r30*cos(a*tp30/tringle30)),q30+(r30*sin(a*tp30/tringle30)));
}
glEnd ();


glBegin(GL_QUADS);//toll stand
glColor3f(0.87f, 0.68f, 0.45f);
```

```
glVertex2f(0.825f, -0.83f);
glVertex2f(0.835f, -0.83f);
glVertex2f(0.835f, -0.65f);
glVertex2f(0.825f, -0.65f);
glEnd();
glBegin(GL_QUADS);//toll stand
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(0.79f, -0.78f);
glVertex2f(0.80f, -0.78f);
glVertex2f(0.80f, -0.65f);
glVertex2f(0.79f, -0.65f);
glEnd();
glBegin(GL_QUADS);//toll stand
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(0.86f, -0.78f);
glVertex2f(0.87f, -0.78f);
glVertex2f(0.87f, -0.65f);
glVertex2f(0.86f, -0.65f);
glEnd();

GLfloat p31=0.83f; GLfloat q31= -0.63f; GLfloat r31 = 0.05f;
int tringle31=40;

GLfloat tp31 =2.0f * PI  ;

glBegin (GL_TRIANGLE_FAN);
glColor3ub ( 160,157,156);
glVertex2f (p31,q31);
for(a= 0;a<=tringle31; a++)
{
   glVertex2f (p31+(r31*cos(a*tp31/tringle31)),q31+(r31*sin(a*tp31/tringle31)));
}
glEnd ();


glBegin(GL_QUADS);//CHAIR
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(0.35f, -0.15f);
glVertex2f(0.50f, -0.15f);
glVertex2f(0.525f, -0.05f);
glVertex2f(0.375f, -0.05f);
glEnd();
```

```
glBegin(GL_LINES);
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(0.36f, -0.15f);
glVertex2f(0.36f, -0.225f);

glVertex2f(0.49f, -0.15f);
glVertex2f(0.49f, -0.225f);

glVertex2f(0.52f, -0.05f);
glVertex2f(0.52f, -0.145f);

glVertex2f(0.52f, -0.05f);
glVertex2f(0.52f, 0.05f);

glVertex2f(0.38f, -0.05f);
glVertex2f(0.38f, 0.05f);

glVertex2f(0.52f, 0.05f);
glVertex2f(0.38f, 0.05f);

glVertex2f(0.52f, 0.025f);
glVertex2f(0.38f, 0.025f);

glVertex2f(0.52f, 0.00f);
glVertex2f(0.38f, 0.00f);

glEnd();


glBegin(GL_QUADS);//table stand
glColor3ub(112,128,144);
glVertex2f(0.62f, -0.81f);
glVertex2f(0.64f, -0.81f);
glColor3ub(119,136,153);
glVertex2f(0.64f, -0.64f);
glVertex2f(0.62f, -0.64f);
glEnd();
```

- **Tree:**

```
glBegin(GL_TRIANGLES);//right side  small tree triangle-1
 glColor3ub(47, 92, 13);
  glVertex2f(0.00f, -0.35f);//middle line
  glVertex2f(0.10f, -0.35f);
  glVertex2f(0.05f, -0.15f);
```

```
    glEnd();
    glBegin(GL_TRIANGLES);//right side  small tree triangle-2
   glColor3ub(47, 92, 13);
    glVertex2f(0.05f, -0.25f);//middle line
    glVertex2f(0.10f, -0.4f);
    glVertex2f(0.00f, -0.4f);

    glEnd();
    glBegin(GL_TRIANGLES);//right side  small tree triangle-3
   glColor3ub(47, 92, 13);
    glVertex2f(0.05f, -0.30f);//middle line
    glVertex2f(0.10f, -0.45f);
    glVertex2f(0.00f, -0.45f);

    glEnd();



    glBegin(GL_TRIANGLES);//left side  small tree triangle-1
    glColor3ub(47, 92, 13);
    glVertex2f(-0.55f, -0.35f);//middle line
    glVertex2f(-0.65f, -0.35f);
    glVertex2f(-0.60f, -0.15f);

    glEnd();
        glBegin(GL_TRIANGLES);//left side  small tree triangle-2
 glColor3ub(47, 92, 13);
    glVertex2f(-0.55f, -0.4f);//middle line
    glVertex2f(-0.65f, -0.4f);
    glVertex2f(-0.60f, -0.25f);

    glEnd();
    glBegin(GL_TRIANGLES);//left side  small tree triangle-3
glColor3ub(47, 92, 13);
    glVertex2f(-0.55f, -0.45f);//middle line
    glVertex2f(-0.65f, -0.45f);
    glVertex2f(-0.60f, -0.30f);

    glEnd();

    glBegin(GL_TRIANGLES);//right side  middle tree triangle-1
glColor3ub(47, 92, 13);
```

```
    glVertex2f(0.15f, -0.40f);//middle line
    glVertex2f(0.1f, -0.6f);
    glVertex2f(0.2f, -0.6f);

    glEnd();
    glBegin(GL_TRIANGLES);//right side  middle tree triangle-2
    glColor3ub(47, 92, 13);
    glVertex2f(0.15f, -0.50f);//middle line
    glVertex2f(0.1f, -0.65f);
    glVertex2f(0.2f, -0.65f);

    glEnd();
    glBegin(GL_TRIANGLES);//right side  middle tree triangle-3
glColor3ub(47, 92, 13);
    glVertex2f(0.15f, -0.55f);//middle line
    glVertex2f(0.1f, -0.7f);
    glVertex2f(0.2f, -0.7f);
    glEnd();

    glBegin(GL_TRIANGLES);//left side  middle tree triangle-1
    glColor3ub(47, 92, 13);
    glVertex2f(-0.70f, -0.40f);//middle line
    glVertex2f(-0.75f, -0.60f);
    glVertex2f(-0.65f, -0.60f);

    glEnd();

    glBegin(GL_TRIANGLES);//left side  middle tree triangle-1
    glColor3ub(47, 92, 13);
    glVertex2f(-0.70f, -0.50f);//middle line
    glVertex2f(-0.75f, -0.65f);
    glVertex2f(-0.65f, -0.65f);

    glEnd();


    glBegin(GL_TRIANGLES);//left side  middle tree triangle-2
    glColor3ub(47, 92, 13);
    glVertex2f(-0.70f, -0.55f);//middle line
    glVertex2f(-0.75f, -0.70f);
    glVertex2f(-0.65f, -0.70f);

    glEnd();
```

```
 glBegin(GL_TRIANGLES);//right side last tree triangle-1
glColor3ub(47, 92, 13);
 glVertex2f(0.25f, -0.65f);//middle line
 glVertex2f(0.2f, -0.85f);
 glVertex2f(0.3f, -0.85f);

 glEnd();
 glBegin(GL_TRIANGLES);//right side last tree triangle-2
 glColor3ub(47, 92, 13);
 glVertex2f(0.25f, -0.75f);//middle line
 glVertex2f(0.2f, -0.90f);
 glVertex2f(0.3f, -0.90f);

 glEnd();
 glBegin(GL_TRIANGLES);//right side  last tree triangle-3
 glColor3ub(47, 92, 13);
 glVertex2f(0.25f, -0.80f);//middle line
 glVertex2f(0.2f, -0.95f);
 glVertex2f(0.3f, -0.95f);

 glEnd();

 glBegin(GL_TRIANGLES);//left side  last tree triangle-1
glColor3ub(47, 92, 13);
 glVertex2f(-0.80f, -0.65f);//middle line
 glVertex2f(-0.85f, -0.85f);
 glVertex2f(-0.75f, -0.85f);

 glEnd();
 glBegin(GL_TRIANGLES);//left side  last tree triangle-2
glColor3ub(47, 92, 13);
 glVertex2f(-0.80f, -0.75f);//middle line
 glVertex2f(-0.85f, -0.9f);
 glVertex2f(-0.75f, -0.9f);

 glEnd();

 glBegin(GL_TRIANGLES);//left side  last tree triangle-3
 glColor3ub(47, 92, 13);
 glVertex2f(-0.80f, -0.8f);//middle line
 glVertex2f(-0.85f, -0.95f);
 glVertex2f(-0.75f, -0.95f);
```

```
    glEnd();


GLfloat p26=-0.95f; GLfloat q26= 0.7f; GLfloat r26 = 0.15f;// big tree
int tringle26=40;
GLfloat tp26 =11.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(47, 92, 13);
glVertex2f (p26,q26);
for(i= 0;i<=tringle26; i++)
{
glVertex2f(p26+(r26*cos(i*tp26/tringle26)),q26+(r26*sin(i*tp26/tringle26)));
}
glEnd ();


//int j; //Right Leaf
GLfloat p27=-0.75f; GLfloat q27= 0.7f; GLfloat r27 = 0.15f;
int tringle27=40;
GLfloat tp27 =11.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(47, 92, 13);
glVertex2f (p27,q27);
for(i= 0;i<=tringle27; i++)
{
glVertex2f (p27+(r27*cos(i*tp27/tringle27)),q27+(r27*sin(i*tp27/tringle27)));
}
glEnd ();


//int k; //Upper Leaf
GLfloat p28=-0.85f; GLfloat q28= 0.85f; GLfloat r28 = 0.15f;
int tringle28=40;
GLfloat tp28 =11.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(47, 92, 13);
glVertex2f (p28,q28);
for(i= 0;i<=tringle28; i++)
{
glVertex2f (
p28+(r28*cos(i*tp28/tringle28)),q28+(r28*sin(i*tp28/tringle28)));
}
glEnd ();


glBegin(GL_QUADS);//Tree ground
glColor3ub(58,24,16);
glVertex2f(-0.95f, 0.5f);
glVertex2f(-0.85f, 0.5f);
glVertex2f(-0.85f, -0.2f);
glVertex2f(-0.95f, -0.2f);
glEnd();
```

- **Mountain:**

```
glBegin(GL_TRIANGLES);//mountain
    glColor3ub(22, 82, 23);
    glVertex2f(-0.50f, 0.55f);
    glVertex2f(-0.30f, 0.55f);
    glVertex2f(-0.40f, 0.65f);
    glEnd();

    glBegin(GL_TRIANGLES);//mountain
    glColor3ub(22, 82, 23);
    glVertex2f(-0.30f, 0.55f);
    glVertex2f(-0.10f, 0.55f);
    glVertex2f(-0.20f, 0.65f);
    glEnd();

    glBegin(GL_TRIANGLES);//mountain
    glColor3ub(22, 82, 23);
    glVertex2f(-0.10f, 0.55f);
    glVertex2f(0.10f, 0.55f);
    glVertex2f(0.0f, 0.65f);
    glEnd();



    glBegin(GL_TRIANGLES);//mountain in near
    glColor3ub(8, 115, 10);
    glVertex2f(-1.0f, 0.55f);
    glVertex2f(-0.80f, 0.55f);
    glVertex2f(-0.90f, 0.65f);
    glEnd();

    glBegin(GL_TRIANGLES);//mountain in near
```

```
glColor3ub(8, 115, 10);
glVertex2f(-0.80f, 0.55f);
glVertex2f(-0.60f, 0.55f);
glVertex2f(-0.70f, 0.65f);
glEnd();

glBegin(GL_TRIANGLES);//mountain in near
glColor3ub(8, 115, 10);
glVertex2f(-0.60f, 0.55f);
glVertex2f(-0.40f, 0.55f);
glVertex2f(-0.50f, 0.65f);
glEnd();

glBegin(GL_TRIANGLES);//mountain in near
glColor3ub(8, 115, 10);
glVertex2f(-0.40f, 0.55f);
glVertex2f(-0.20f, 0.55f);
glVertex2f(-0.30f, 0.65f);
glEnd();

glBegin(GL_TRIANGLES);//mountain in near
glColor3ub(8, 115, 10);
glVertex2f(-0.20f, 0.55f);
glVertex2f(0.0f, 0.55f);
glVertex2f(-0.10f, 0.65f);
glEnd();

glBegin(GL_TRIANGLES);//mountain in near
glColor3ub(8, 115, 10);
glVertex2f(0.20f, 0.55f);
glVertex2f(0.0f, 0.55f);
glVertex2f(0.10f, 0.65f);
glEnd();
```

- **Steamer & boat:**

```
glPushMatrix();
 glTranslatef(position,0.0f, 0.0f);
 glBegin(GL_QUADS);// streamer ist part
 glColor3f(0.36f, 0.20f, 0.09f);
 glVertex2f(0.70f, 0.125f);
 glVertex2f(0.90f, 0.125f);
 glVertex2f(0.95f, 0.175f);
 glVertex2f(0.65f, 0.175f);
 glEnd();
```

```
glBegin(GL_QUADS);//2nd part
glColor3f(0.0f, 0.45f, 0.26f);
glVertex2f(0.70f, 0.175f);
glVertex2f(0.90f, 0.175f);
glVertex2f(0.90f, 0.225f);
glVertex2f(0.70f, 0.225f);
glEnd();

glBegin(GL_QUADS);// streamer window
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(0.725f, 0.1875f);
glVertex2f(0.75f, 0.1875f);
glVertex2f(0.75f, 0.2125f);
glVertex2f(0.725f, 0.2125f);
glEnd();

glBegin(GL_QUADS);// streamer window
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(0.775f, 0.1875f);
glVertex2f(0.80f, 0.1875f);
glVertex2f(0.80f, 0.2125f);
glVertex2f(0.775f, 0.2125f);
glEnd();

glBegin(GL_QUADS);// streamer window
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(0.825f, 0.1875f);
glVertex2f(0.85f, 0.1875f);
glVertex2f(0.85f, 0.2125f);
glVertex2f(0.825f, 0.2125f);
glEnd();

glBegin(GL_QUADS);// streamer window
glColor3f(1.0f, 1.0f, 1.0f);
glVertex2f(0.875f, 0.1875f);
glVertex2f(0.90f, 0.1875f);
glVertex2f(0.90f, 0.2125f);
glVertex2f(0.875f, 0.2125f);
glEnd();

glBegin(GL_QUADS);//3rd part
glColor3f(0.0f, 0.f, 0.26f);
```

```
glVertex2f(0.725f, 0.225f);
glVertex2f(0.875f, 0.225f);
glVertex2f(0.875f, 0.255f);
glVertex2f(0.725f, 0.255f);
glEnd();

glBegin(GL_QUADS);// air out
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.745f, 0.255f);
glVertex2f(0.760f, 0.255f);
glVertex2f(0.760f, 0.305f);
glVertex2f(0.745f, 0.305f);
glEnd();

glBegin(GL_QUADS);// air out
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.775f, 0.255f);
glVertex2f(0.79f, 0.255f);
glVertex2f(0.79f, 0.305f);
glVertex2f(0.775f, 0.305f);
glEnd();

glBegin(GL_QUADS);// air out
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.805f, 0.255f);
glVertex2f(0.82f, 0.255f);
glVertex2f(0.82f, 0.305f);
glVertex2f(0.805f, 0.305f);
glEnd();

glBegin(GL_QUADS);// air out
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.835f, 0.255f);
glVertex2f(0.85f, 0.255f);
glVertex2f(0.85f, 0.305f);
glVertex2f(0.835f, 0.305f);
glEnd();

glPopMatrix();


glPushMatrix();
glTranslatef(position3,0.0f, 0.0f);
```

```
glBegin(GL_QUADS);//boat
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.47f, 0.325f);
glVertex2f(0.65f, 0.325f);
glVertex2f(0.70f, 0.375f);
glVertex2f(0.42f, 0.375f);
glEnd();

glBegin(GL_LINES);
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.49f, 0.375f);
glVertex2f(0.49f, 0.475f);

glVertex2f(0.50f, 0.375f);
glVertex2f(0.50f, 0.525f);

glVertex2f(0.50f, 0.575f);
glVertex2f(0.55f, 0.375f);

glVertex2f(0.50f, 0.575f);
glVertex2f(0.57f, 0.375f);

glEnd();

glBegin(GL_TRIANGLES);
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.455f, 0.38f);
glVertex2f(0.49f, 0.38f);
glVertex2f(0.49f, 0.475f);
glEnd();

glBegin(GL_TRIANGLES);
glColor3f(0.0f, 0.f, 0.26f);
glVertex2f(0.50f, 0.38f);
glVertex2f(0.535f, 0.38f);
glVertex2f(0.50f, 0.575f);
glEnd();

glBegin(GL_TRIANGLES);//maji
glColor3ub(21, 176, 34);
glVertex2f(0.63f, 0.40f);
glVertex2f(0.67f, 0.40f);
glVertex2f(0.65f, 0.45f);
```

```
glEnd();

int v;//maji

GLfloat p24=0.65f; GLfloat q24= 0.45f; GLfloat r24 = 0.01f;
int Gtringle24=40;

GLfloat tp24 =2.0f * PI ;
glBegin (GL_TRIANGLE_FAN);
glColor3ub(207, 202, 155);
glVertex2f (p24,q24);
for(v= 0;v<=Gtringle24; v++)
{
glVertex2f (
p24+(r24*cos(v*tp24/Gtringle24)),
q24+(r24*sin(v*tp24/Gtringle24))
);
}
glEnd ();

glBegin(GL_LINES);//maji
glColor3ub(207, 202, 155);
glVertex2f(0.645f, 0.40f);
glVertex2f(0.645f, 0.375f);

glVertex2f(0.655f, 0.40f);
glVertex2f(0.655f, 0.375f);

glVertex2f(0.66f, 0.425f);
glVertex2f(0.62f, 0.415f);

glVertex2f(0.64f, 0.425f);
glVertex2f(0.61f, 0.435f);

glColor3ub(145, 86, 3);
glVertex2f(0.61f, 0.445f);
glVertex2f(0.66f, 0.30f);

glEnd();
   glPopMatrix();
```

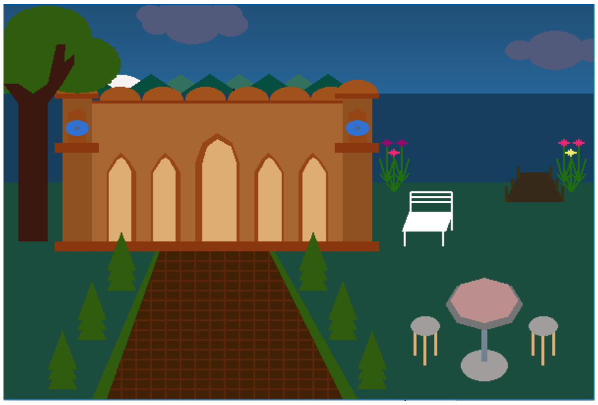
- **Road:**

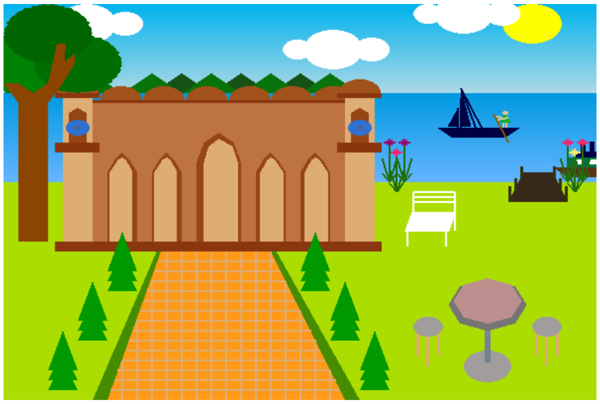
```
glBegin(GL_QUADS);//road
glColor3f(1.0f, 0.6f, 0.09f);
```

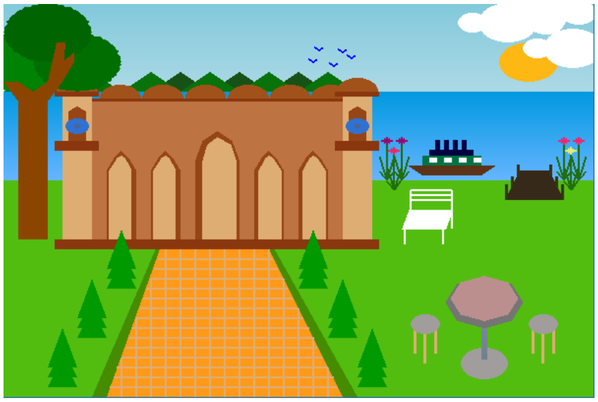
```
//glColor3f(0.58f, 0.23f, 0.19f);
glVertex2f(-0.47f, -0.25f);
glVertex2f(-0.1f, -0.25f);
glVertex2f(0.15f, -1.0f);
glVertex2f(-0.65f, -1.0f);

glEnd();

glBegin(GL_LINES);//road
glColor3f(0.87f, 0.68f, 0.45f);
glVertex2f(-0.475f, -0.30f);
glVertex2f(-0.0875f, -0.30f);
glVertex2f(-0.490f, -0.35f);
glVertex2f(-0.07f, -0.35f);
glVertex2f(-0.5f, -0.4f);
glVertex2f(-0.05f, -0.4f);
glVertex2f(-0.515f, -0.45f);
glVertex2f(-0.035f, -0.45f);
glVertex2f(-0.525f, -0.5f);
glVertex2f(-0.02f, -0.5f);
glVertex2f(-0.54f, -0.55f);
glVertex2f(0.00f, -0.55f);
glVertex2f(-0.554f, -0.6f);
glVertex2f(0.017f, -0.6f);
glVertex2f(-0.562f, -0.65f);
glVertex2f(0.0355f, -0.65f);
glVertex2f(-0.575f, -0.7f);
glVertex2f(0.052f, -0.7f);
glVertex2f(-0.587f, -0.75f);
glVertex2f(0.064f, -0.75f);
glVertex2f(-0.6f, -0.8f);
glVertex2f(0.084f, -0.8f);
glVertex2f(-0.615f, -0.85f);
glVertex2f(0.1f, -0.85f);
glVertex2f(-0.625f, -0.9f);
glVertex2f(0.12f, -0.9f);
glVertex2f(-0.64f, -0.95f);
glVertex2f(0.135f, -0.95f);
glVertex2f(-0.60f, -0.8f);
glVertex2f(-0.60f, -1.0f);
glVertex2f(-0.55f, -0.6f);
glVertex2f(-0.55f, -1.0f);
glVertex2f(-0.5f, -0.4f);
```

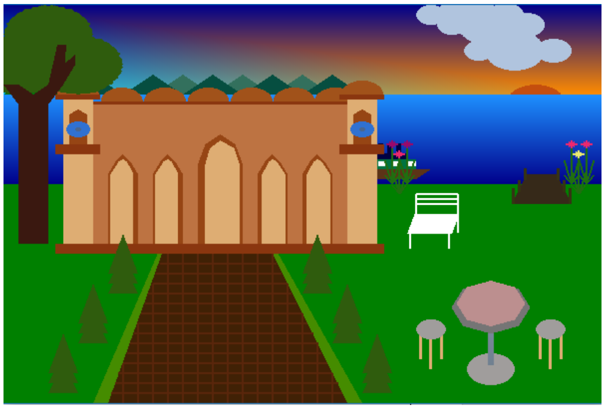
```
glVertex2f(-0.5f, -1.0f);
glVertex2f(-0.45f, -0.25f);
glVertex2f(-0.45f, -1.0f);
glVertex2f(-0.4f, -0.25f);
glVertex2f(-0.4f, -1.0f);
glVertex2f(-0.35f, -0.25f);
glVertex2f(-0.35f, -1.0f);
glVertex2f(-0.3f, -0.25f);
glVertex2f(-0.3f, -1.0f);
glVertex2f(-0.25f, -0.25f);
glVertex2f(-0.25f, -1.0f);
glVertex2f(-0.2f, -0.25f);
glVertex2f(-0.2f, -1.0f);
glVertex2f(-0.15f, -0.25f);
glVertex2f(-0.15f, -1.0f);
glVertex2f(-0.1f, -0.25f);
glVertex2f(-0.1f, -1.0f);
glVertex2f(-0.05f, -0.4f);
glVertex2f(-0.05f, -1.0f);
glVertex2f(0.0f, -0.55f);
glVertex2f(0.0f, -1.0f);
glVertex2f(0.05f, -0.7f);
glVertex2f(0.05f, -1.0f);
glVertex2f(0.10f, -0.85f);
glVertex2f(0.10f, -1.0f);
glEnd();
```

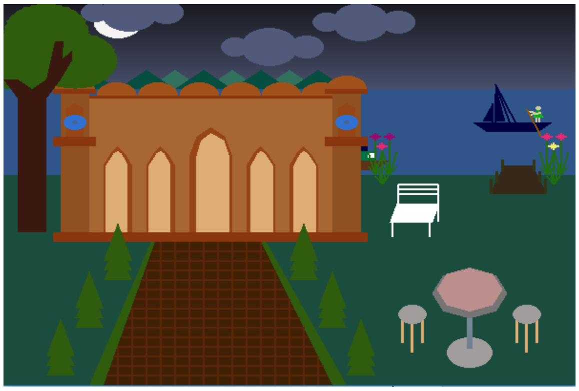
6. Output Screenshot:











7. Demo Link:

Five time Prayer Scenario .mp4

8. Uniqueness of your Project:

- Visual Representation of Prayer Times.
- Interactive Elements for Prayer Reminders.
- Customizable Environment.
- Five different scenarios.
- Changes the scenario by keypress.

9. Conclusion:

The weather plays a crucial role in shaping the practice of the five-time prayers within the Islamic faith. Throughout this project, we have delved into the significant influence that weather conditions exert on prayer timings, the adaptability of prayer routines during varying weather scenarios, and the importance of understanding meteorological patterns for devout individuals. The flexibility inherent in Islamic teachings enables believers to adjust their prayer schedules and methods in response to these weather fluctuations, emphasizing the essence of devotion over rigidity in practice. In conclusion, this project highlights the dynamic interplay between weather and the practice of the five-time prayers, emphasizing the adaptability and dedication of individuals striving to maintain their religious observances in various meteorological scenarios.

10. Future Work:

In Islam, there are specific times for each of the five daily prayers. Weather conditions such as sunrise and sunset times can vary based on location and season. A tool or application could provide real-time adjustments to prayer times based on the current weather conditions. The project might include educational resources on how weather conditions were historically considered in Islamic practices or teachings related to adaptability in prayer routines based on environmental factors. This project will create a mobile application that provides real-time weather updates and prayer time adjustments based on the user's location. This could enhance accessibility and usability for individuals on the go.

11. Contribution: (Mention each of the team-mate's contribution)

Name & Id		**Work Details**
MD SHANJID HASAN ABIR	22-47202-1	Trees, mountains, fields and roads, color changes
MD TANVIR RAHMAN MOLLA	22-46052-1	River, Boat, Steamer, Cloud and Birds.
MD ASAD-UZ-ZAMAN	22-46582-1	Bench, table & tools, a wharf
S.M. SAYED ARIFIN OMI	22-46613-1	Mosque

12.Reference:

- https://webeduclick.com/computer-graphics-tutorial/
- https://www.tutorialspoint.com/computer_graphics/index.htm