AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

FACULTY OF SCIENCE AND TECHNOLOGY

PROJECT FINAL REPORT

COMPUTER GRAPHICS

COURSE SUPERVISOR: DIPTA JUSTIN GOMES

SECTION: F

DYNAMIC WEATHER SIMULATOR

Group Members

Name	ID	Program
TANJIM ISRAK DIHAN	21-45415-3	BSc [CSE]
NAFIUR RAHMAN NILOY	22-46459-1	BSc [CSE]

Introduction:

Welcome to our Weather Simulator project, where we invite you to embark on an exciting journey into the fascinating world of meteorology and computer graphics. In this endeavor, we aim to craft an immersive and interactive experience that allows users to explore the intricacies of weather phenomena within a virtual environment.

Weather is an integral part of our daily lives, influencing everything from our clothing choices to outdoor activities and travel plans. With our Weather Simulator, we seek to provide users with a unique opportunity to interact with and understand the dynamics of various weather patterns firsthand.

Through the power of GLUT in CodeBlocks, we're equipped to create a visually captivating simulation that accurately replicates the behaviors and characteristics of rain, snow, sunshine, clouds, and more. By leveraging advanced rendering techniques and dynamic simulations, we aim to deliver an authentic and engaging experience that mirrors the complexity of real-world weather systems.

But our Weather Simulator is more than just a visual spectacle—it's also an educational tool designed to foster curiosity and learning. As users navigate through different weather scenarios, they'll have the chance to observe how factors such as temperature, humidity, and atmospheric pressure influence the formation and behavior of weather patterns.

Join us as we dive into the realms of computer graphics and meteorology, combining our passion for technology with our fascination for the natural world. Together, let's embark on this adventure to create a Weather Simulator that not only entertains but also enlightens and inspires users of all ages.

Through experimentation, exploration, and discovery, we invite you to unlock the secrets of weather and experience the magic of simulation firsthand. Welcome to the Weather Simulator, where the forecast is always immersive, and the possibilities are endless.

Background:

The concept of weather simulation dates back several decades, with early attempts focused on simple mathematical models and empirical techniques. Over time, advancements in computational science, meteorology, and atmospheric physics have led to the development of more sophisticated weather simulation models and algorithms.

Modern weather simulation systems rely on a combination of numerical models, observational data, and computational algorithms to simulate atmospheric processes and predict future weather conditions. These systems typically involve complex mathematical representations of atmospheric dynamics, thermodynamics, and fluid mechanics, which are solved using numerical integration techniques and computational algorithms.

However, despite these advancements, current weather simulation technologies still face challenges in accurately representing the full spectrum of weather phenomena, especially at fine spatial and temporal scales. Additionally, the growing need for real-time weather updates and adaptive forecasting capabilities has spurred interest in developing more dynamic and responsive weather simulation systems.

The Dynamic Weather Simulator project builds upon the foundation of existing weather simulation research while introducing novel methodologies and technologies to address current limitations. By integrating advanced computational techniques with real-time data processing capabilities, this project aims to push the boundaries of weather simulation and contribute to more accurate, dynamic, and reliable weather forecasting and prediction systems.

Objective of the Project:

The primary objective of the Dynamic Weather Simulator project is to develop a user-friendly weather simulation system focused on basic environmental elements. The project aims to create a simplified yet immersive simulation environment featuring:

1. **Realistic Weather Dynamics**: Implement basic weather phenomena such as clouds during the day, rainfall, and varying intensities of rain. The simulation should capture the essence of weather changes over time, enhancing the overall realism of the environment.

2. **Dynamic Environmental Elements**: Introduce dynamic elements within the simulated environment, including cars moving along a road from left to right and a ship navigating along a river. These elements will add a layer of dynamism to the simulation, creating a more engaging experience for users.

3. **User Interaction**: Design an intuitive user interface for configuring simulation parameters and interacting with the simulated environment. Users should be able to control aspects such as the intensity of rainfall, the speed of cars and the ship, and the time of day to customize their simulation experience.
4. **Visual Representation**: Develop visually appealing graphics and animations to represent the simulated weather conditions and environmental elements. This includes realistic depictions of clouds, rainfall, vehicles, and the river scene, enhancing the overall visual fidelity of the simulation.
5. **Ease of Use**: Ensure that the weather simulation system is easy to set up and use, even for users with limited technical knowledge. Provide clear instructions and intuitive controls to facilitate a seamless user experience.

By achieving these objectives, the Dynamic Weather Simulator project aims to provide users with a simple yet immersive platform for experiencing and exploring basic weather dynamics in a controlled virtual environment. The simulation will serve as a valuable educational tool for understanding fundamental concepts of weather and environmental science, while also offering entertainment value for users of all ages.

Methodology:

The Dynamic Weather Simulator project utilizes OpenGL (Open Graphics Library) for rendering realistic weather conditions and environmental elements in a virtual scene. The methodology involves several key functions and techniques to create an immersive simulation experience.

1. **OpenGL Setup:**
    - The project initializes the OpenGL context and sets up essential parameters such as the depth buffer and clear color using the **MyInit()** and **init()** functions.
    - The **glutCreateWindow()**, **glutInitWindowSize()**, and **glutInitWindowPosition()** functions are utilized to create a new window with specified dimensions and position on the screen.

2. **Rendering the Scene:**
   - The **display()** function serves as the main rendering function, drawing the entire scene using OpenGL commands.
   - Within the **display()** function, various environmental elements such as the sky, mountains, buildings, trees, river, and moving objects like boats and cars are drawn using OpenGL primitives.
   - The **Circle()** function is employed to draw circles representing raindrops with specified centers, radii, and colors.
3. **Animation and Interaction:**
   - Animation of moving objects such as boats, waves, cars, and clouds is achieved through dedicated update functions (**boatUpdate()**, **rightWaveUpdate()**, **leftWaveUpdate()**, **rightCarUpdate()**, **leftCarUpdate()**, **cloud1Update()**, **cloud2Update()**).
   - User interaction is facilitated through the **handleKeypress()** function, which responds to keyboard input by updating the state of the program. For example, pressing the 'd' key toggles between day and night modes.
4. **Rain Simulation:**
   - The **rainFunction()** is responsible for drawing raindrops as lines using OpenGL commands.
   - Raindrop positions are updated dynamically using the **updateRain()** function, creating the illusion of falling rain.
5. **OpenGL Callbacks:**
   - Callback functions such as **glutDisplayFunc()** and **glutKeyboardFunc()** are registered to handle window display and keyboard input events, respectively.
   - The **glutMainLoop()** function initiates the GLUT event loop, which continuously processes events such as keyboard and mouse input and triggers the display callback function to redraw the scene as needed.

By implementing these functions and techniques within the OpenGL framework, the Dynamic Weather Simulator project achieves its objective of simulating basic weather conditions and environmental elements in a user-friendly and interactive manner.

Code:

```
#include<cstdio>
#include <GL/gl.h>
#include <GL/glut.h>
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<math.h>
#define PI 3.141516
using namespace std;

float boatPosition = 0.0f;
GLfloat boatSpeed = 0.05f;


float leftWavePosition = 0.0f;
GLfloat leftWaveSpeed = 0.062f;

float rightWavePosition = 0.0f;
GLfloat rightWaveSpeed = 0.072f;



float leftCarPosition = 0.0f;
GLfloat leftCarSpeed = 0.05f; //0.012

float rightCarPosition = 0.0f;
GLfloat rightCarSpeed = 0.05f;

float cloud1Position = 0.0f;
GLfloat cloud1Speed = 0.05f;

float cloud2Position = 0.0f;
GLfloat cloud2Speed = 0.07f;

bool rain = false;
float xrain = 10.0f;
float yrain = 10.0f;
bool rain1 = false;
bool rain2 = false;

float sun_move = 0.0f;
float moon_move = 0.0f;
bool dayStart1 = false;
bool nightStart2 = false;
bool moonStart3 = false;
bool autumnStart4 = false;


GLfloat mountainColor[3] = {0.5f,0.5f,0.5f}; // 0.0, 0.0, 90.0
GLfloat skyColor1[3] = {0.0752f, 0.839f, 0.940f};
GLfloat skyColor2[3] = {0.279f, 0.854f, 0.930f};
GLfloat skyColor3[3] = {0.534f, 0.901f, 0.890f};
GLfloat skyColor4[3] = {0.754f, 0.901f, 0.920f};
GLfloat cloudColor1[3] = {0.259f, 0.255f, 0.520f};
GLfloat cloudBoxColor1[3] = {0.0752f, 0.839f, 0.940f};
```

```
/// ////////////////////////////////////// Building////////////////////////////////////
GLfloat building1[3] = {0.890, 0.573, 0.0979};
GLfloat building2[3] = {0.820, 0.0410, 0.262};
GLfloat building3[3] = {1.00, 0.00, 0.0333};
GLfloat building4[3] = {0.336, 0.840, 0.655};
GLfloat building5[3] = {0.930, 0.828, 0.0558};
GLfloat building6[3] = {0.990, 0.424, 0.0198};
/// ////////////////////////////////////// Building window////////////////////////////////////
GLfloat building1window1[3] = {0.810, 0.699, 0.203};


GLfloat building2window1[3] = {0.389, 0.770, 0.339};

GLfloat building3window1[3] = {0.248, 0.920, 0.719};

GLfloat building4window2[3] = {0.950, 0.822, 0.361};

GLfloat building5window1[3] = {0.101, 0.183,  0.920};

GLfloat building6window1[3] = {0.683, 0.0486, 0.810};

GLfloat building7window[3] = {0.930, 0.149, 0.227};


GLfloat riverColor[3] = {0.0f,0.0f,90.0f};



void rainFunction(){
   glPushMatrix();

   glTranslatef( xrain,yrain,0.0f);

   glBegin(GL_LINES);

   glVertex2f(-.85,1.9);glVertex2f(-.8,1.8);
   glVertex2f(-.55,1.9);glVertex2f(-.5,1.8);
   glVertex2f(-.25,1.9);glVertex2f(-.2,1.8);
   glVertex2f(.05,1.9);glVertex2f(.1,1.8);
         glVertex2f(.35,1.9);glVertex2f(.4,1.8);
         glVertex2f(.65,1.9);glVertex2f(.7,1.8);
         glVertex2f(.95,1.9);glVertex2f(1.0,1.8);

    glVertex2f(-1.0,1.6);glVertex2f(-.95,1.5);
   glVertex2f(-.7,1.6);glVertex2f(-.65,1.5);
   glVertex2f(-.4,1.6);glVertex2f(-.35,1.5);
   glVertex2f(-.1,1.6);glVertex2f(-.05,1.5);
         glVertex2f(.2,1.6);glVertex2f(.25,1.5);
         glVertex2f(.5,1.6);glVertex2f(.55,1.5);
         glVertex2f(.8,1.6);glVertex2f(.85,1.5);


   glVertex2f(-.85,1.3);glVertex2f(-.8,1.2);
   glVertex2f(-.55,1.3);glVertex2f(-.5,1.2);
   glVertex2f(-.25,1.3);glVertex2f(-.2,1.2);
          glVertex2f(.05,1.3);glVertex2f(.1,1.2);
         glVertex2f(.35,1.3);glVertex2f(.4,1.2);
         glVertex2f(.65,1.3);glVertex2f(.7,1.2);
         glVertex2f(.95,1.3);glVertex2f(1.0,1.2);

   glVertex2f(-1.0,1.0);glVertex2f(-.95,.9);
   glVertex2f(-.7,1.0);glVertex2f(-.65,0.9);
   glVertex2f(-.4,1.0);glVertex2f(-.35,0.9);
   glVertex2f(-.1,1.0);glVertex2f(-.05,0.9);
         glVertex2f(.2,1.0);glVertex2f(.25,.9);
         glVertex2f(.5,1.0);glVertex2f(.55,0.9);
         glVertex2f(.8,1.0);glVertex2f(.85,0.9);



   glVertex2f(-.85,.7);glVertex2f(-.8,.6);
   glVertex2f(-.55,.7);glVertex2f(-.5,0.6);
   glVertex2f(-.25,.7);glVertex2f(-.2,0.6);
        glVertex2f(.05,.7);glVertex2f(.1,.6);
         glVertex2f(.35,.7);glVertex2f(.4,.6);
         glVertex2f(.65,.7);glVertex2f(.7,.6);
         glVertex2f(.95,.7);glVertex2f(1.0,.6);

   glVertex2f(-1.0,.4);glVertex2f(-.95,.3);
   glVertex2f(-.7,.4);glVertex2f(-.65,0.3);
   glVertex2f(-.4,.4);glVertex2f(-.35,0.3);
   glVertex2f(-.1,.4);glVertex2f(-.05,0.3);
         glVertex2f(.2,.4);glVertex2f(.25,.3);
         glVertex2f(.5,.4);glVertex2f(.55,0.3);
         glVertex2f(.8,.40);glVertex2f(.85,0.3);

   glVertex2f(-.85,.1);glVertex2f(-.8,0.0);
   glVertex2f(-.55,.1);glVertex2f(-.5,0.0);
   glVertex2f(-.25,.1);glVertex2f(-.2,0.0);
        glVertex2f(.05,.1);glVertex2f(.1,.0);
         glVertex2f(.35,.1);glVertex2f(.4,.0);
         glVertex2f(.65,.1);glVertex2f(.7,.0);
         glVertex2f(.95,.1);glVertex2f(1.0,.0);


   glVertex2f(-1.0,-.2);glVertex2f(-.95,-.3);
   glVertex2f(-.7,-.2);glVertex2f(-.65,-0.3);
   glVertex2f(-.4,-.2);glVertex2f(-.35,-0.3);
   glVertex2f(-.1,-.2);glVertex2f(-.05,-0.3);
         glVertex2f(.2,-.2);glVertex2f(.25,-.3);
         glVertex2f(.5,-.2);glVertex2f(.55,-.3);
         glVertex2f(.8,-.2);glVertex2f(.85,-.3);

   glVertex2f(-.85,-.5);glVertex2f(-.8,-.6);
   glVertex2f(-.55,-.5);glVertex2f(-.5,-.6);
   glVertex2f(-.25,-.5);glVertex2f(-.2,-.6);
        glVertex2f(.05,-.5);glVertex2f(.1,-.6);
         glVertex2f(.35,-.5);glVertex2f(.4,-.6);
         glVertex2f(.65,-.5);glVertex2f(.7,-.6);
         glVertex2f(.95,-.5);glVertex2f(1.0,-.6);

   glVertex2f(-1.0,-.8);glVertex2f(-.95,-.9);
   glVertex2f(-.7,-.8);glVertex2f(-.65,-0.9);
   glVertex2f(-.4,-.8);glVertex2f(-.35,-0.9);
   glVertex2f(-.1,-.8);glVertex2f(-.05,-0.9);
         glVertex2f(.2,-.8);glVertex2f(.25,-.9);
```

```
            glVertex2f(.5,-.8);glVertex2f(.55,-.9);
            glVertex2f(.8,-.8);glVertex2f(.85,-.9);

    glVertex2f(-.85,-1.1);glVertex2f(-.8,-1.2);
    glVertex2f(-.55,-1.1);glVertex2f(-.5,-1.2);
    glVertex2f(-.25,-1.1);glVertex2f(-.2,-1.2);
          glVertex2f(.05,-1.1);glVertex2f(.1,-1.2);
         glVertex2f(.35,-1.1);glVertex2f(.4,-1.2);
         glVertex2f(.65,-1.1);glVertex2f(.7,-1.2);
         glVertex2f(.95,-1.1);glVertex2f(1.0,-1.2);

    glVertex2f(-1.0,-1.4);glVertex2f(-.95,-1.5);
    glVertex2f(-.7,-1.4);glVertex2f(-.65,-1.5);
    glVertex2f(-.4,-1.4);glVertex2f(-.35,-1.5);
    glVertex2f(-.1,-1.4);glVertex2f(-.05,-1.5);
            glVertex2f(.2,-1.4);glVertex2f(.25,-1.5);
            glVertex2f(.5,-1.4);glVertex2f(.55,-1.5);
            glVertex2f(.8,-1.4);glVertex2f(.85,-1.5);


      glEnd();
      glPopMatrix();
}


void Circle(GLfloat x, GLfloat y,GLfloat radius,int c1, int c2, int c3)
{
    int i;

    int triangleAmount = 100;

    GLfloat twicePi = 2.0f * PI;

    glBegin(GL_TRIANGLE_FAN);
    glColor3ub(c1,c2,c3);
    glVertex2f(x, y);
    for(i = 0; i <= triangleAmount; i++){
      glVertex2f(
             x+(radius * cos(i * twicePi / triangleAmount)),
             y+(radius * sin(i * twicePi / triangleAmount))
             );
    }
    glEnd();
}




void rightCarUpdate(int value) {

    if(rightCarPosition > 1.7)//
      rightCarPosition = -1.6f;

    rightCarPosition += rightCarSpeed;

             glutPostRedisplay();
```

```
        glutTimerFunc(100, rightCarUpdate, 0);
}


void rightWaveUpdate(int value) {

  if(rightWavePosition > 1.0)//1.7
    rightWavePosition = -1.0f; // -1.6

  rightWavePosition += rightWaveSpeed;

        glutPostRedisplay();


        glutTimerFunc(100, rightWaveUpdate, 0);
}

void leftWaveUpdate(int value) {

  if(leftWavePosition < -1.8)//
    leftWavePosition = 1.05f;

  leftWavePosition -= leftWaveSpeed;

        glutPostRedisplay();


        glutTimerFunc(100, leftWaveUpdate, 0);
}

void leftCarUpdate(int value) {

  if(leftCarPosition < -1.7)//
    leftCarPosition = 1.002f;

  leftCarPosition -= leftCarSpeed;

        glutPostRedisplay();


        glutTimerFunc(100, leftCarUpdate, 0);
}

void boatUpdate(int value) {

  if(boatPosition > 1.7)//
    boatPosition = -1.2f;

  boatPosition += boatSpeed;

        glutPostRedisplay();


        glutTimerFunc(100, boatUpdate, 0);
}

void cloud1Update(int value) {
  if(cloud1Position > 0.4)//
    cloud1Position = -0.7f;

  cloud1Position += cloud1Speed;

	glutPostRedisplay();


	glutTimerFunc(100, cloud1Update, 0);
}

void cloud2Update(int value) {

  if(cloud2Position > 0.6)//
    cloud2Position = -0.4f;

  cloud2Position += cloud2Speed;

	glutPostRedisplay();


	glutTimerFunc(100, cloud2Update, 0);
}


///------------------- Handle Keyboard press function------------///////
void handleKeypress(unsigned char key, int x, int y) {

	switch (key) {



case 'n':
  //For night
  nightStart2 = true;
  dayStart1 = false;
  autumnStart4 = false;
  break;
case 'd':
  //For day
  dayStart1 = true;
  nightStart2 = false;
  autumnStart4 = false;
  break;
case 'a':

  autumnStart4 = true;
  dayStart1 = false;
  nightStart2 = false;
  break;
case 'r':

   rain = true;

  break;
case 's':
   rain = false;
   break;


glutPostRedisplay();


             }
}





void display() {
  glClear(GL_COLOR_BUFFER_BIT);
  glLoadIdentity();

 glBegin(GL_QUADS);
    glColor3fv(skyColor1);
    glVertex2f(-1.0f, 0.8f);
    glVertex2f(1.0f, 0.8f);
    glVertex2f(1.0f, 1.0f);
    glVertex2f(-1.0f,1.0f);
 glEnd();

 /// sky Layer 2///
 glBegin(GL_QUADS);
    glColor3fv(skyColor2);
    glVertex2f(-1.0f, 0.6f);
    glVertex2f(1.0f, 0.6f);
    glVertex2f(1.0f, 0.8f);
    glVertex2f(-1.0f,0.8f);
 glEnd();

 /// sky Layer 3///
 glBegin(GL_QUADS);
    glColor3fv(skyColor3);
    glVertex2f(-1.0f, 0.4f);
    glVertex2f(1.0f, 0.4f);
    glVertex2f(1.0f, 0.6f);
    glVertex2f(-1.0f, 0.6f);
 glEnd();

 /// sky Layer 4///
 glBegin(GL_QUADS);
    glColor3fv(skyColor4);
    glVertex2f(-1.0f, 0.24f);
    glVertex2f(1.0f, 0.24f);
    glVertex2f(1.0f, 0.4f);
    glVertex2f(-1.0f,0.4f);
 glEnd();


/// /////////////////////////////////////////////////////////////////////////// ////
///------------------------------SKY Layer END------------------------------////
/// /////////////////////////////////////////////////////////////////////////// ////
```

```
///---------------------------------Day/Night---Start-------------------////////////////
glPushMatrix();
glTranslatef(0.0, sun_move, 0.0);
Circle(-0.35,0.75,0.08,247,247,73);
glPopMatrix();

///--------------------------------Sun/Moon-----Move condition----------////////////////
///--------------------------------Light change function----------------////////////////
if(moonStart3==true)
{
   //Moon();
glPushMatrix();
   Circle(0.55,0.75,0.08,217,217,214);
   glPopMatrix();

}

if(nightStart2 == true)
{
   sun_move -= 0.05; ///
   if(sun_move < -0.32) //
   {
      nightStart2 = false;
      moonStart3= true;
///---------------------Set mountain color for night--------------------//////////////////
    mountainColor[0] = 0.3f; // 1.0
    mountainColor[1] = 0.3f; // 1.0
    mountainColor[2] = 0.3f; // 0.0
    /// //// Set Sky color for night-test  color/////
    ///  sky block 1///////                    ////
    skyColor1[0] = 0.166;
    skyColor1[1] = 0.203;
    skyColor1[2] = 0.920;
    /// -------------------------cloud block night---------------------/////////////////////

    cloudBoxColor1[0] = 0.166;
    cloudBoxColor1[1] = 0.203;
    cloudBoxColor1[2] = 0.920;
    /// /////////////////////////////////////////////////////////////////////////////////
    /// -------------------------cloud color block night-----------------/////////////////////
    cloudColor1[0] = 0.497;
    cloudColor1[1] = 0.515;
    cloudColor1[2] = 0.710;
    /// sky block 2///////                  /////
    skyColor2[0] = 0.191;
    skyColor2[1] = 0.227;
    skyColor2[2] = 0.910;
    /// sky block 3///////                  /////
    skyColor3[0] = 0.340;
    skyColor3[1] = 0.369;
    skyColor3[2] = 0.920;
    /// sky block 4///////                  /////
    skyColor4[0] = 0.448;
    skyColor4[1] = 0.455;
    skyColor4[2] = 0.590;
/// //////////////////////////////////// Building(1-6)/////////////////////////////////
```

```
    building1[0] = 0.663;
    building1[1] = 0.573;
    building1[2] = 0.533;


    building2[0] = 0.999;
    building2[1] = 0.000;
    building2[2] = 0.788;


    building3[0] = 0.0000;
    building3[1] = 0.111;
    building3[2] = 0.222;


    building4[0] = 0.001;
    building4[1] = 0.555;
    building4[2] = 0.6543;


    building5[0] = 0.110;
    building5[1] = 0.339;
    building5[2] = 0.0898;



    building6[0] = 0.99;
    building6[1] = 0.4444;
    building6[2] = 0.0555;
/// ///////////////////////////////////// Building window///////////////////////////////////////////
/// ////////////////////////////////// building1window1//////////////////////////////

    building1window1[0] = 0.610;
    building1window1[1] = 0.110;
    building1window1[2] = 0.800;
    /// ////////////////////////////////// building2window1//////////////////////////////
    building2window1[0] = 0.900;
    building2window1[1] = 0.660;
    building2window1[2] = 0.333;
    /// ////////////////////////////////// building3window1//////////////////////////////
     building3window1[0] = 0.444;
    building3window1[1] = 0.778;
    building3window1[2] = 0.719;
    /// ////////////////////////////////// building4window2//////////////////////////////
    building4window2[0] = 0.920;
    building4window2[1] = 0.929;
    building4window2[2] = 0.928;
     /// ////////////////////////////////// building5window//////////////////////////////
    building5window1[0] = 0.950;
    building5window1[1] = 0.0736;
    building5window1[2] = 0.215;
     /// ////////////////////////////////// building6window////building8window//////////////////////////
    building6window1[0] = 0.870;
    building6window1[1] = 0.818;
    building6window1[2] = 0.880;
    /// ////////////////////////////////// building7window//////////////////////////////

    building7window[0] = 0.290;
    building7window[1] = 0.283;
    building7window[2] = 0.226;
 /// /////////////////////////////////// building window ends/////////////////////////////////
   }

}

if(dayStart1 == true)
{
   moonStart3=false;
   sun_move += 0.05;
   if(sun_move > -0.03) // 0.3
   {
/// ////--------------------- Set mountain color for Day----------------/// ///
    nightStart2 = true;
    moonStart3 = false;

    mountainColor[0] = 0.3f; //0.0
    mountainColor[1] = 0.3f; //0.0
    mountainColor[2] = 0.3f; //90.0
/// //// Set Sky color for Day-test  color/////
///  sky block 1///////                /////
    skyColor1[0] = 0.0752;
    skyColor1[1] = 0.839;
    skyColor1[2] = 0.940;
/// ////////////////////////////////////////cloud block night////////////////////////

    cloudBoxColor1[0] = 0.0752;
    cloudBoxColor1[1] = 0.839;
    cloudBoxColor1[2] = 0.940;
    /// sky block 2///////                /////
    skyColor2[0] = 0.279;
    skyColor2[1] = 0.854;
    skyColor2[2] = 0.930;
    /// sky block 3///////                /////
    skyColor3[0] = 0.534;
    skyColor3[1] = 0.848;
    skyColor3[2] = 0.890;
    /// sky block 4///////                /////
    skyColor4[0] = 0.754;
    skyColor4[1] = 0.901;
    skyColor4[2] = 0.920;

    /// //////////////////////////////////////// Building(1-6)/////////////////////////////////////

    building1[0] = 0.890;
    building1[1] = 0.573;
    building1[2] = 0.0979;


    building2[0] = 0.820;
    building2[1] = 0.0410;
    building2[2] = 0.262;


    building3[0] = 1.00;
    building3[1] = 0.00;
    building3[2] = 0.0333;
    building4[0] = 0.336;
    building4[1] = 0.840;
    building4[2] = 0.655;


    building5[0] = 0.930;
    building5[1] = 0.828;
    building5[2] = 0.0558;



    building6[0] = 0.990; //0.2 //
    building6[1] = 0.424; // 0.2 //
    building6[2] = 0.0198; // 0.2 //
/// //////////////////////////////////////// Building window/////////////////////////////////////

    /// ///////////////////////////////////// building1window//////////////////////////////////
    building1window1[0] = 0.810;
    building1window1[1] = 0.699;
    building1window1[2] = 0.203;

    building2window1[0] = 0.389;
    building2window1[1] = 0.770;
    building2window1[2] = 0.339;

    building3window1[0] = 0.248;
    building3window1[1] = 0.920;
    building3window1[2] = 0.719;

    /// ///////////////////////////////////// building4window2//////////////////////////////////
    building4window2[0] = 0.950;
    building4window2[1] = 0.822;
    building4window2[2] = 0.361;
     //, ,
     /// ///////////////////////////////////// building5window1//////////////////////////////////
    building5window1[0] = 0.101;
    building5window1[1] = 0.183;
    building5window1[2] = 0.920;

    building6window1[0] = 0.683;
    building6window1[1] = 0.0486;
    building6window1[2] = 0.810;

    building7window[0] = 0.930;
    building7window[1] = 0.149;
    building7window[2] = 0.227;
/// ////////////////////////////////////// building window ends//////////////////////////////////
   }
   if(sun_move > 0)
   {
      dayStart1 = false;
   }
}
/// ////////// ////////////// /////AUTUMN SEASON////// /////////// ////////////////////////////// //////

if(autumnStart4 == true)
```

```
{
   moonStart3=false;

/// ////--------------------- Set mountain color for Day----------------/// ///

    mountainColor[0] = 1.0f;
    mountainColor[1] = 0.7f;
    mountainColor[2] = 0.0f;
/// //// Set Sky color for Day-test  color/////
///  sky block 1///////                /////
    skyColor1[0] = 0.95;
    skyColor1[1] = 0.8;
    skyColor1[2] = 0.7;
/// cloudBOX r sky colour1 er value ta same hote hobe /// ///////////
/// ///////////////////////////////////cloud block night/////////////////////

    cloudBoxColor1[0] = 0.95;
    cloudBoxColor1[1] = 0.8;
    cloudBoxColor1[2] = 0.7;
/// ///////////////////////////////////////////////////////////////////////////////
    /// -------------------------cloud color block autumn-----------------////////////////////
    cloudColor1[0] = 1.0;
    cloudColor1[1] = 0.7;
    cloudColor1[2] = 0.3;
/// 1, 0.8, 0.4
    /// sky block 2///////                /////
    skyColor2[0] = 0.90;
    skyColor2[1] = 0.8;
    skyColor2[2] = 0.7;
    /// sky block 3///////                /////
    skyColor3[0] = 0.85;
    skyColor3[1] = 0.8;
    skyColor3[2] = 0.7;
    /// sky block 4///////                /////
    skyColor4[0] = 0.85;
    skyColor4[1] = 0.8;
    skyColor4[2] = 0.7;

    /// //////////////////////////////////// Building(1-6)//////////////////////////////////////

    building1[0] = 0.890;
    building1[1] = 0.573;
    building1[2] = 0.0979;


    building2[0] = 0.820;
    building2[1] = 0.0410;
    building2[2] = 0.262;


    building3[0] = 1.00;
    building3[1] = 0.00;
    building3[2] = 0.0333;


    building4[0] = 0.336;
    building4[1] = 0.840;
    building4[2] = 0.655;
    building5[0] = 0.930;
    building5[1] = 0.828;
    building5[2] = 0.0558;



    building6[0] = 0.990;
    building6[1] = 0.424;
    building6[2] = 0.0198;
/// ////////////////////////////////////// Building window/////////////////////////////////////

    /// ////////////////////////////////////// building1window//////////////////////////////////
    building1window1[0] = 0.810;
    building1window1[1] = 0.699;
    building1window1[2] = 0.203;

    building2window1[0] = 0.389;
    building2window1[1] = 0.770;
    building2window1[2] = 0.339;

    building3window1[0] = 0.248;
    building3window1[1] = 0.920;
    building3window1[2] = 0.719;

    /// ////////////////////////////////////// building4window2/////////////////////////////////
    building4window2[0] = 0.950;
    building4window2[1] = 0.822;
    building4window2[2] = 0.361;
     //, ,
     /// ////////////////////////////////////// building5window1/////////////////////////////////
    building5window1[0] = 0.101;
    building5window1[1] = 0.183;
    building5window1[2] = 0.920;

    building6window1[0] = 0.683;
    building6window1[1] = 0.0486;
    building6window1[2] = 0.810;

    building7window[0] = 0.930;
    building7window[1] = 0.149;
    building7window[2] = 0.227;
/// //////////////////////////////////// building window ends/////////////////////////////////

}


/// /////////// /////////////////// /////AUTUMN SEASON ENDS///////// ////////////// ////////




/// ///////////////////////////// Cloud 1 ////////////////// //////////////////////////
glPushMatrix();
glTranslatef(cloud1Position, 0.0f, 0.0f);
///cloud 1- left-circle
float x,y,i;
```

```
glPushMatrix();

   glTranslatef(-0.7f,0.9f,0.0f);
   glBegin(GL_TRIANGLE_FAN);
   glColor3fv(cloudColor1);
   for(i=0;i<=2*3.14;i+=0.0001)
{
    x=0.02912*sin(i);
    y=0.02912*cos(i);
    glVertex2f(x,y);
}
glEnd();
glPopMatrix();
  ///
  ///
///cloud 1- middle-circle
  ///
 glPushMatrix();
   glTranslatef(-0.64f,0.9f,0.0f);
   glBegin(GL_TRIANGLE_FAN);

   glColor3fv(cloudColor1);
   for(i=0;i<=2*3.14;i+=0.0001)
{
    x=0.04500*sin(i);
    y=0.04500*cos(i);
    glVertex2f(x,y);
}
glEnd();
glPopMatrix();
  ///
  ///cloud 1- left-circle
  ///
glPushMatrix();
   glTranslatef(-0.58f,0.9f,0.0f);
   glBegin(GL_TRIANGLE_FAN);

   glColor3fv(cloudColor1);
   for(i=0;i<=2*3.14;i+=0.0001)
{
    x=0.03123*sin(i);
    y=0.03123*cos(i);
    glVertex2f(x,y);
}
glEnd();
glPopMatrix();
/// //////// //////////////////////////////////////
/// cloud1 back quad////
glBegin(GL_QUADS);
     glColor3fv(cloudBoxColor1);
     glVertex2f(-0.739f, 0.851f);
     glVertex2f(-0.540f ,0.850f);
     glVertex2f(-0.54f, 0.9f);
     glVertex2f(-0.74f, 0.9f);
  glEnd();

  glPopMatrix();
```

```
/// ////////////////////////// Cloud 2 ///////////////// ////////////////////////
glPushMatrix();
glTranslatef(cloud2Position, 0.0f, 0.0f);
///cloud 2- left-circle
glPushMatrix();

    glTranslatef(-0.41f,0.9f,0.0f);
    glBegin(GL_TRIANGLE_FAN);

    glColor3fv(cloudColor1);
    for(i=0;i<=2*3.14;i+=0.0001)
{
     x=0.02745*sin(i);
     y=0.02745*cos(i);
     glVertex2f(x,y);
}
glEnd();
glPopMatrix();
///
///cloud 2- middle-circle
///
 glPushMatrix();
    glTranslatef(-0.358f,0.927f,0.0f);
    glBegin(GL_TRIANGLE_FAN);

    glColor3fv(cloudColor1);
    for(i=0;i<=2*3.14;i+=0.0001)
{
     x=0.05039*sin(i);
     y=0.05039*cos(i);
     glVertex2f(x,y);
}
glEnd();
glPopMatrix();
   ///
   ///cloud 2- left-circle
   ///
glPushMatrix();
    glTranslatef(-0.270f,0.929f,0.0f);
    glBegin(GL_TRIANGLE_FAN);

    glColor3fv(cloudColor1);
    for(i=0;i<=2*3.14;i+=0.0001)
{
     x=0.06058*sin(i);
     y=0.06058*cos(i);
     glVertex2f(x,y);
}
glEnd();
glPopMatrix();
/// //////// /////////////////////////////////
/// cloud2 back quad////
glBegin(GL_QUADS);
      glColor3fv(cloudBoxColor1);
      glVertex2f(-0.470f, 0.849f);
      glVertex2f(-0.2f ,0.85f);
      glVertex2f(-0.2f, 0.9f);
      glVertex2f(-0.47f, 0.9f);
   glEnd();
   glPopMatrix();
/// =======================================Cloud
ENDS===========================================================/////////////////


    // left mountain
    glBegin(GL_TRIANGLES);
    glColor3fv(mountainColor);
    //glColor3f(mountainColor);   /// PC- 1.0f, 0.3f, 0.6f
       glVertex2f(-1.0, 0.24);
       glVertex2f(0.3, 0.24);
       glVertex2f(-0.44, 0.8);
    glEnd();

    // right mountain //
    glBegin(GL_TRIANGLES);
    glColor3fv(mountainColor);
       glVertex2f(-0.5, 0.24);
       glVertex2f(0.24, 0.8);
       glVertex2f(1.0, 0.24);
    glEnd();


    // building - 1... //701
   glBegin(GL_QUADS);
      glColor3fv(building1);
      glColor3f(1.0f, 0.0f, 0.0f);
      glVertex2f(-1.0f, 0.24f);
      glVertex2f(-0.8f, 0.24f);

      glVertex2f(-0.8f, 0.4f);
      glVertex2f(-1.0f, 0.4f);

   // window left bottom
   /// /////////////////////////////////// building1window1///////////////////////////////
      glColor3fv(building1window1);

      glVertex2f(-0.96f, 0.30f);
      glVertex2f(-0.92f, 0.30f);

      glVertex2f(-0.92f, 0.34f);
      glVertex2f(-0.96f, 0.34f);


      // window right bottom
    /// /////////////////////////////////// building1window1///////////////////////////////
      glColor3fv(building1window1);

      glVertex2f(-0.86f, 0.30f);
      glVertex2f(-0.82f, 0.30f);

      glVertex2f(-0.82f, 0.34f);
      glVertex2f(-0.86f, 0.34f);
glEnd();
    // building - 2... //702
   glBegin(GL_QUADS);
```

```
   // upper portion
    //glColor3fv(building2window1);
     glVertex2f(-0.760f, 0.54f);
     glVertex2f(-0.63f, 0.54f);

     glVertex2f(-0.63f, 0.620f);
     glVertex2f(-0.760f, 0.620f);
     //// main building block
      glColor3fv(building2);
     glColor3f(1.0f, 0.8f, 0.4f);

     glVertex2f(-0.8f, 0.24f);
     glVertex2f(-0.6f, 0.24f);

     glVertex2f(-0.6f, 0.55f);
     glVertex2f(-0.8f, 0.55f);
     //-----------------------
     // window left up
     /// //////////////////////////////////// building2window1////////////////////////////////
      glColor3fv(building2window1);

     glVertex2f(-0.76f, 0.46f);
     glVertex2f(-0.72f, 0.46f);

     glVertex2f(-0.72f, 0.5f);
     glVertex2f(-0.76f, 0.5f);

     // window right up
     /// //////////////////////////////////// building2window1////////////////////////////////
     glColor3fv(building2window1);

     glVertex2f(-0.68f, 0.46f);
     glVertex2f(-0.64f, 0.46f);

     glVertex2f(-0.64f, 0.5f);
     glVertex2f(-0.68f, 0.5f);
 //-----------------------
     // window right down
    /// //////////////////////////////////// building2window1////////////////////////////////
     glColor3fv(building2window1);

     glVertex2f(-0.68f, 0.38f);
     glVertex2f(-0.64f, 0.38f);

     glVertex2f(-0.64f, 0.42f);
     glVertex2f(-0.68f, 0.42f);

     // window left down
     /// //////////////////////////////////// building2window1////////////////////////////////
     glColor3fv(building1window1);
     glVertex2f(-0.76f, 0.38f);
     glVertex2f(-0.72f, 0.38f);

     glVertex2f(-0.72f, 0.42f);
     glVertex2f(-0.76f, 0.42f);
     //-----------------------

 // window left bottom
```

```
   /// //////////////////////////////////// building2window1////////////////////////////////
   glColor3fv(building2window1);

   glVertex2f(-0.76f, 0.30f);
   glVertex2f(-0.72f, 0.30f);

   glVertex2f(-0.72f, 0.34f);
   glVertex2f(-0.76f, 0.34f);


   // window right bottom
   /// //////////////////////////////////// building2window1////////////////////////////////
   glColor3fv(building2window1);
   glColor3f(0.0f, 0.8f, 0.4f);

   glVertex2f(-0.68f, 0.30f);
   glVertex2f(-0.64f, 0.30f);

   glVertex2f(-0.64f, 0.34f);
   glVertex2f(-0.68f, 0.34f);
 glEnd();

 glBegin(GL_QUADS);

  glColor3fv(building3);
   //glColor3f(0.3f, 0.3f, 0.3f);

   glVertex2f(-0.6f, 0.24f);
   glVertex2f(-0.4f, 0.24f);

   glVertex2f(-0.4f, 0.480f);
   glVertex2f(-0.6f, 0.480f);


    /// //////////////////////////////////// building3window1////////////////////////////////
  glColor3fv(building3window1);
   glVertex2f(-0.48f, 0.38f);
   glVertex2f(-0.44f, 0.38f);

   glVertex2f(-0.44f, 0.42f);
   glVertex2f(-0.48f, 0.42f);

   // window left down
   glColor3f(0.0f, 0.8f, 0.4f);

   glVertex2f(-0.56f, 0.38f);
   glVertex2f(-0.52f, 0.38f);

   glVertex2f(-0.52f, 0.42f);
   glVertex2f(-0.56f, 0.42f);
   //----------------------
    // window left bottom
   glColor3f(0.0f, 0.8f, 0.4f);

   glVertex2f(-0.56f, 0.30f);
   glVertex2f(-0.52f, 0.30f);

   glVertex2f(-0.52f, 0.34f);
  glVertex2f(-0.56f, 0.34f);


  // window right bottom
  glColor3f(0.0f, 0.8f, 0.4f);

  glVertex2f(-0.48f, 0.30f);
  glVertex2f(-0.44f, 0.30f);

  glVertex2f(-0.44f, 0.34f);
  glVertex2f(-0.48f, 0.34f);
glEnd();


glBegin(GL_QUADS);
   glColor3fv(building4);


  glVertex2f(-0.4f, 0.24f);
  glVertex2f(-0.2f, 0.24f);

  glVertex2f(-0.2f, 0.6f);
  glVertex2f(-0.4f, 0.6f);

  // window left up
  /// /////////////////////////////////// building4window2///////////////////////////////
  glColor3fv(building4window2);

  glVertex2f(-0.36f, 0.46f);
  glVertex2f(-0.32f, 0.46f);

  glVertex2f(-0.32f, 0.5f);
  glVertex2f(-0.36f, 0.5f);

  // window right up
  glVertex2f(-0.28f, 0.46f);
  glVertex2f(-0.24f, 0.46f);

  glVertex2f(-0.24f, 0.5f);
  glVertex2f(-0.28f, 0.5f);

    // window right down
    glColor3fv(building4window2);


  glVertex2f(-0.28f, 0.38f);
  glVertex2f(-0.24f, 0.38f);

  glVertex2f(-0.24f, 0.42f);
  glVertex2f(-0.28f, 0.42f);

  // window left down

  glColor3f(0.7f, 0.2f, 1.0f);
  glVertex2f(-0.36f, 0.38f);
  glVertex2f(-0.32f, 0.38f);

  glVertex2f(-0.32f, 0.42f);
  glVertex2f(-0.36f, 0.42f);
  //-----------------------
   // window left bottom
  glColor3f(0.7f, 0.2f, 1.0f);

  glVertex2f(-0.36f, 0.30f);
  glVertex2f(-0.32f, 0.30f);

  glVertex2f(-0.32f, 0.34f);
  glVertex2f(-0.36f, 0.34f);


  // window right bottom
  glColor3f(0.7f, 0.2f, 1.0f);

  glVertex2f(-0.28f, 0.30f);
  glVertex2f(-0.24f, 0.30f);

  glVertex2f(-0.24f, 0.34f);
  glVertex2f(-0.28f, 0.34f);

glEnd();


glBegin(GL_QUADS);

 glColor3fv(building5);
  //glColor3f(1.0f, 0.7f, 0.2f);

  glVertex2f(-0.2f, 0.24f);
  glVertex2f(0.0f, 0.24f);

  glVertex2f(0.0f, 0.55f);
  glVertex2f(-0.2f, 0.55f);

  // window left up
  glColor3f(0.0f, 0.8f, 0.4f);

  glVertex2f(-0.16f, 0.46f);
  glVertex2f(-0.12f, 0.46f);

  glVertex2f(-0.12f, 0.5f);
  glVertex2f(-0.16f, 0.5f);

  // window right up
  glColor3f(0.0f, 0.8f, 0.4f);

  glVertex2f(-0.08f, 0.46f);
  glVertex2f(-0.04f, 0.46f);

  glVertex2f(-0.04f, 0.5f);
  glVertex2f(-0.08f, 0.5f);

 // window right down
 /// //////////////////////////////////// building5window2////////////////////////////////
 glColor3fv(building5window1);
```

```
   glVertex2f(-0.08f, 0.38f);
   glVertex2f(-0.04f, 0.38f);

   glVertex2f(-0.04f, 0.42f);
   glVertex2f(-0.08f, 0.42f);

   // window left down
   /// /////////////////////////////////// building5window2///////////////////////////////
   glColor3fv(building5window1);


   glVertex2f(-0.16f, 0.38f);
   glVertex2f(-0.12f, 0.38f);

   glVertex2f(-0.12f, 0.42f);
   glVertex2f(-0.16f, 0.42f);
   //-----------------------
    // window left bottom
   glColor3f(0.7f, 0.2f, 1.0f);

   glVertex2f(-0.16f, 0.30f);
   glVertex2f(-0.12f, 0.30f);

   glVertex2f(-0.12f, 0.34f);
   glVertex2f(-0.16f, 0.34f);


   // window right bottom
   glColor3f(0.7f, 0.2f, 1.0f);

   glVertex2f(-0.08f, 0.30f);
   glVertex2f(-0.04f, 0.30f);

   glVertex2f(-0.04f, 0.34f);
   glVertex2f(-0.08f, 0.34f);

 glEnd();

 // building - 6... //706
 glBegin(GL_QUADS);
  glColor3fv(building6);
   //glColor3f(0.4f, 0.8f, 0.1f);

   glVertex2f(0.0f, 0.24f);
   glVertex2f(0.12f, 0.24f);

   glVertex2f(0.12f, 0.72f);
   glVertex2f(0.0f, 0.72f);


   // window 1

   glColor3fv(building6window1);

   glVertex2f(0.04f, 0.64f);
   glVertex2f(0.08f, 0.64f);

   glVertex2f(0.08f, 0.68f);
```

```
    glVertex2f(0.04f, 0.68f);

    // window 2
    glColor3fv(building6window1);

    glVertex2f(0.04f, 0.54f);
    glVertex2f(0.08f, 0.54f);

    glVertex2f(0.08f, 0.58f);
    glVertex2f(0.04f, 0.58f);

    // window 3
   glColor3fv(building6window1);

    glVertex2f(0.04f, 0.44f);
    glVertex2f(0.08f, 0.44f);

    glVertex2f(0.08f, 0.48f);
    glVertex2f(0.04f, 0.48f);

    // window 4
    glColor3fv(building6window1);

    glVertex2f(0.04f, 0.34f);
    glVertex2f(0.08f, 0.34f);

    glVertex2f(0.08f, 0.38f);
    glVertex2f(0.04f, 0.38f);

  glEnd();

  // building - 7... //707
  glBegin(GL_TRIANGLES);
  glColor3f(0.2f, 0.3f, 0.6f);
    glVertex2f(0.12f, 0.24f); //1
    glVertex2f(0.34f, 0.24f); //2
    glVertex2f(0.12f, 0.68f); //4
glColor3f(1.0f, 1.0f, 0.6f);

    glVertex2f(0.34f, 0.24f); //2
    glVertex2f(0.34f, 0.68f); //3
    glVertex2f(0.12f, 0.68f); //4

   glEnd();

    glBegin(GL_QUADS);
    // window 1
    glColor3fv(building7window);
    glVertex2f(0.16f, 0.56f);
    glVertex2f(0.288f, 0.56f);

    glVertex2f(0.288f, 0.62f);
    glVertex2f(0.16f, 0.62f);


    // window 2
    glColor3fv(building7window);
    glVertex2f(0.16f, 0.46f);
    glVertex2f(0.288f, 0.46f);

    glVertex2f(0.288f, 0.52f);
    glVertex2f(0.16f, 0.52f);


    // window 3
    glColor3fv(building7window);

    glVertex2f(0.16f, 0.36f);
    glVertex2f(0.288f, 0.36f);

    glVertex2f(0.288f, 0.42f);
    glVertex2f(0.16f, 0.42f);

  glEnd();


// building - 8... //708
  glBegin(GL_QUADS);

    glColor3f(0.2f, 0.2f, 0.2f);

    glVertex2f(0.34f, 0.24f);
    glVertex2f(0.52f, 0.24f);

    glVertex2f(0.52f, 0.6f);
    glVertex2f(0.34f, 0.6f);


     // window left up
   glColor3fv(building6window1);

    glVertex2f(0.38f, 0.5f);
    glVertex2f(0.42f, 0.5f);

    glVertex2f(0.42f, 0.54f);
    glVertex2f(0.38f, 0.54f);

    // window right up
   glColor3fv(building6window1);

    glVertex2f(0.45f, 0.5f);
    glVertex2f(0.49f, 0.5f);

    glVertex2f(0.49f, 0.54f);
    glVertex2f(0.45f, 0.54f);

     // window left down
   glColor3fv(building6window1);

    glVertex2f(0.38f, 0.4f);
    glVertex2f(0.42f, 0.4f);

    glVertex2f(0.42f, 0.44f);
    glVertex2f(0.38f, 0.44f);
  // window right down
 glColor3fv(building6window1);

  glVertex2f(0.45f, 0.4f);
  glVertex2f(0.49f, 0.4f);

  glVertex2f(0.49f, 0.44f);
  glVertex2f(0.45f, 0.44f);
  //-----------------------
   // window left bottom
  glColor3fv(building6window1);

  glVertex2f(0.38f, 0.3f);
  glVertex2f(0.42f, 0.3f);

  glVertex2f(0.42f, 0.34f);
  glVertex2f(0.38f, 0.34f);

  // window right bottom
  glColor3fv(building6window1);

  glVertex2f(0.45f, 0.3f);
  glVertex2f(0.49f, 0.3f);

  glVertex2f(0.49f, 0.34f);
  glVertex2f(0.45f, 0.34f);


glEnd();

/// Building///code ENDS here///////////////


glBegin(GL_TRIANGLES);
   glColor3f(0.470f, 0.194f, 0.174f);

  glVertex2f(0.72f, 0.24f); //1
  glVertex2f(0.73f, 0.24f); //2
  glVertex2f(0.73f, 0.6f); //3
  //////////////////////////
   glColor3f(0.0909f, 0.850f, 0.0510f);
  glVertex2f(0.73f, 0.6f); //left leaf
  glVertex2f(0.8f, 0.56f);
  glVertex2f(0.8f, 0.64f);
  ///
   glColor3f(0.0805f, 0.700f, 0.0700f);
  glVertex2f(0.73f, 0.6f); //left up leaf
  glVertex2f(0.78f, 0.66f);
  glVertex2f(0.74f, 0.68f);

  ///
   glColor3f(0.124f, 0.710f, 0.114f);
  glVertex2f(0.73f, 0.6f); //left up leaf
  glVertex2f(0.73f, 0.69f);
  glVertex2f(0.68f, 0.68f);

  ///
   glColor3f(0.116f, 0.630f, 0.107f);
```

```
    glVertex2f(0.73f, 0.6f); //left up leaf
    glVertex2f(0.68f, 0.66f);
    glVertex2f(0.64f, 0.56f);

   glEnd();

   /// for river // 710

   glBegin(GL_QUADS);
    glColor3f(0.0f, 1.0f, 1.0f);
    glVertex2f(-1.0f, -0.66f);
    glVertex2f(1.0f, -0.66f);

    glVertex2f(1.0f, -0.2f);
    glVertex2f(-1.0f, -0.2f);
  glEnd();

///-------------------------- TREE SECTION ENDS HERE-------------------------///////

glBegin(GL_QUADS);
    glColor3f(0.188f, 0.188f, 0.188f);//1st point
     glVertex2f(-1.0f, -0.25f);
     glVertex2f(1.0f, -0.25f);
    glVertex2f(1.0f, 0.24f);  //2nd point
    glVertex2f(-1.0f, 0.24f);
  glEnd();
glBegin(GL_QUADS);
    glColor3f(1.0f, 1.0f, 0.0f);//1st point
     glVertex2f(-0.9f, 0.01);
     glVertex2f(-1.0f, 0.01f);
     glVertex2f(-1.0f, -0.01f);
     glVertex2f(-0.9f, -0.01f);
  glEnd();

  glBegin(GL_QUADS);
    glColor3f(1.0f, 1.0f, 0.0f);//1st point
     glVertex2f(-0.8f, 0.01);
     glVertex2f(-0.7f, 0.01f);
     glVertex2f(-0.7f, -0.01f);
     glVertex2f(-0.8, -0.01f);
  glEnd();

  glBegin(GL_QUADS);
    glColor3f(1.0f, 1.0f, 0.0f);//1st point
     glVertex2f(-0.6f, 0.01);
     glVertex2f(-0.5f, 0.01f);
     glVertex2f(-0.5f, -0.01f);
     glVertex2f(-0.6f, -0.01f);
  glEnd();

  glBegin(GL_QUADS);
    glColor3f(1.0f, 1.0f, 0.0f);//1st point
     glVertex2f(-0.4f, 0.01);
     glVertex2f(-0.3f, 0.01f);
     glVertex2f(-0.3f, -0.01f);
     glVertex2f(-0.4f, -0.01f);
  glEnd();
```

```
   glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 0.0f);//1st point
      glVertex2f(-0.2f, 0.01);
      glVertex2f(-0.1f, 0.01f);
      glVertex2f(-0.1f, -0.01f);
      glVertex2f(-0.2f, -0.01f);
  glEnd();

   glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 0.0f);//1st point
      glVertex2f(0.0f, 0.01);
      glVertex2f(0.1f, 0.01f);
      glVertex2f(0.1f, -0.01f);
      glVertex2f(0.0f, -0.01f);
  glEnd();
 ///-/////////////////////////////////////////////Middle-Line--Left Ends & Right Begins
///////////////////////////////////////////////////////////////////////////////
  glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 0.0f);//1st point
      glVertex2f(0.2f, 0.01);
      glVertex2f(0.3f, 0.01f);
      glVertex2f(0.3f, -0.01f);
      glVertex2f(0.2f, -0.01f);
  glEnd();

glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 0.0f);//1st point
      glVertex2f(0.4f, 0.01);
      glVertex2f(0.5f, 0.01f);
      glVertex2f(0.5f, -0.01f);
      glVertex2f(0.4f, -0.01f);
  glEnd();

glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 0.0f);//1st point
      glVertex2f(0.6f, 0.01);
      glVertex2f(0.7f, 0.01f);
      glVertex2f(0.7f, -0.01f);
      glVertex2f(0.6f, -0.01f);
  glEnd();

glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 0.0f);//1st point
      glVertex2f(0.8f, 0.01);
      glVertex2f(0.9f, 0.01f);
      glVertex2f(0.9f, -0.01f);
      glVertex2f(0.8f, -0.01f);
  glEnd();

   glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 0.0f);//1st point
      glVertex2f(0.95f, 0.01);
      glVertex2f(1.0f, 0.01f);
      glVertex2f(1.0f, -0.01f);
      glVertex2f(0.95f, -0.01f);
  glEnd();

/// /////// ---------==========ROAD BLOCK END HERE=====-----------//////////////////
```

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


  glBegin(GL_POLYGON);
  glColor3ub(255, 0, 0);

 glVertex2f(-0.81f, 0.08f); //point-1
 glVertex2f(-0.49f, 0.08f); //point-2
 glVertex2f(-0.5f, 0.12f); //point-3
 glVertex2f(-0.53f, 0.14f); //point-4
 glVertex2f(-0.67f, 0.14f); //point-5

 glVertex2f(-0.64f, 0.14f); //point-6
 glVertex2f(-0.75f, 0.14); //point-7
 glVertex2f(-0.76f, 0.17); //point-8
 glVertex2f(-0.8f, 0.18f); //point-9
 glVertex2f(-0.81f, 0.08f); //point-10
 glEnd();



  glBegin(GL_POLYGON);
  glColor3ub(255, 200, 0);

 glVertex2f(-0.75f, 0.14); //point-1
 glVertex2f(-0.64f, 0.14f); //point-2
 glVertex2f(-0.64f, 0.19f); //point-3
 glVertex2f(-0.71f, 0.19f); //point-4
 glVertex2f(-0.76f, 0.17f); //point-5
 glVertex2f(-0.75f, 0.14f); //point-6
 glEnd();



  glBegin(GL_QUADS);
 glColor3ub(255, 255, 0);
 glVertex2f(-0.67f, 0.14f); //point-1
 glVertex2f(-0.58f, 0.14f); //point-2

 glVertex2f(-0.62f, 0.19f); //point-3
 glVertex2f(-0.67f, 0.19f); //point-4
  glEnd();





 //float x, y, i;

 glPushMatrix();
 glTranslatef(-0.58f, 0.08f,0.0f);
 glBegin(GL_TRIANGLE_FAN);

 glColor3ub(80, 40, 0);
 for(i=0;i<=2*3.14;i+=0.0001)
  {
     x=0.0337*sin(i);
     y=0.0337*cos(i);
     glVertex2f(x,y);
   }

   glEnd();
   glPopMatrix();




      // float x, y, i;

  glPushMatrix();
  glTranslatef(-0.72f,0.08f,0.0f);    //circles er centre er x, y er value
  glBegin(GL_TRIANGLE_FAN);

  glColor3ub(80, 40, 0);
  for(i=0;i<=2*3.14;i+=0.0001)
   {
     x=0.034*sin(i);     //circle er basardher value - point er ber kora value
     y=0.034*cos(i);     //circle er basardher value - point er ber kora value
     glVertex2f(x,y);
   }

   glEnd();
   glPopMatrix();
   glPopMatrix();



   /// ---------------------- Car - 1 END ---------------------- ///



   /// Car - 2 Start --------------------------------------------------

   glPushMatrix();
glTranslatef(leftCarPosition,0.0f, 0.0f);


   glBegin(GL_POLYGON);
   glColor3ub(0, 72, 255);

  glVertex2f(0.535f, -0.13f); //point-1
  glVertex2f(0.84f, -0.13f); //point-2
  glVertex2f(0.83f, -0.04f); //point-3
  glVertex2f(0.80f, -0.05f); //point-4
  glVertex2f(0.78f, -0.07f); //point-5

  glVertex2f(0.71f, -0.07f); //point-6
  glVertex2f(0.62f, -0.07f); //point-7
  glVertex2f(0.57f, -0.07f); //point-8
  glVertex2f(0.545f, -0.09f); //point-9
  glVertex2f(0.535f, -0.13f); //point-10
  glEnd();
```

```
glBegin(GL_QUADS);
  glColor3ub(255, 247, 0);
  glVertex2f(0.62f, -0.07f); //point-1
  glVertex2f(0.71f, -0.07f); //point-2

  glVertex2f(0.71f, -0.02f); //point-3
  glVertex2f(0.65f, -0.02f); //point-4
glEnd();



   glBegin(GL_POLYGON);
   glColor3ub(255, 200, 0);

  glVertex2f(0.71f, -0.07f); //point-1
  glVertex2f(0.78f, -0.07f); //point-2
  glVertex2f(0.80f, -0.05f); //point-3
  glVertex2f(0.77f, -0.02f); //point-4
  glVertex2f(0.71f, -0.02f); //point-5
  glVertex2f(0.71f, -0.07f); //point-6
  glEnd();


  /// Front Tire - Circle // 1009

       // float x, y, i;

   glPushMatrix();
   glTranslatef(0.61f, -0.13f,0.0f);
   glBegin(GL_TRIANGLE_FAN);

   glColor3ub(51, 11, 6);
   for(i=0;i<=2*3.14;i+=0.0001)
    {
       x=0.0337*sin(i);
       y=0.0337*cos(i);
       glVertex2f(x,y);
    }

    glEnd();
    glPopMatrix();


    /// Back Tire - Circle // 1010

     // float x, y, i;

   glPushMatrix();
   glTranslatef(0.74f, -0.13f,0.0f);
   glBegin(GL_TRIANGLE_FAN);

   glColor3ub(80, 10, 0);
   for(i=0;i<=2*3.14;i+=0.0001)
    {
       x=0.034*sin(i);
       y=0.034*cos(i);
       glVertex2f(x,y);
    }
    glEnd();
    glPopMatrix();
glPopMatrix();

    /// ---------------------- Car - 2 END -------------------------------





  ///----------- River Section Start -----------


  /// River Blue Water Block Start ----------
glBegin(GL_QUADS);
   glColor3ub(0, 220, 255);
   glVertex2f(-1.0f, -0.66f);
   glVertex2f(1.0f, -0.66f);

   glVertex2f(1.0f, -0.25f);
   glVertex2f(-1.0f, -0.25f);
  glEnd();
  // River Blue Water Block End ----------



glPushMatrix();
  glTranslatef(boatPosition,0.0f, 0.0f);
  ///-------- Quad Blocks Section --------///


   /*/// Block - 1 ... //501
  glBegin(GL_QUADS);
   glColor3ub(78, 75, 75);
   glVertex2f(0.36f, -0.57f);
   glVertex2f(0.77f, -0.57f);

   glVertex2f(0.77f, -0.55f);
   glVertex2f(0.36f, -0.55f);
  glEnd(); */
  //... Block - 1 ...//End--------------


  /// Block - 2 ... //502
  glBegin(GL_QUADS);
   glColor3ub(138, 3, 42);
   glVertex2f(0.36f, -0.55f);
   glVertex2f(0.77f, -0.55f);

   glVertex2f(0.77f, -0.53f);
   glVertex2f(0.36f, -0.53f);
  glEnd();
  //... Block - 2 ...//End--------------


  /// Block - 3 //503
  glBegin(GL_QUADS);
```

```
    glColor3ub(250, 250, 150);
    glVertex2f(0.45f, -0.53f);
    glVertex2f(0.72f, -0.53f);

    glVertex2f(0.72f, -0.45f);
    glVertex2f(0.45f, -0.45f);
  glEnd();
   //... Block - 3 ...//End--------------


/// Block - 3* //503
  glBegin(GL_QUADS);
    glColor3ub(200, 40, 90);
    glVertex2f(0.45f, -0.5f);
    glVertex2f(0.72f, -0.5f);

    glVertex2f(0.72f, -0.49f);
    glVertex2f(0.45f, -0.49f);
  glEnd();
   //... Block - 3* ...//End--------------


   /// Block - 4 ... //504
  glBegin(GL_QUADS);
    glColor3ub(60, 70, 180);
    glVertex2f(0.51f, -0.53f);
    glVertex2f(0.54f, -0.53f);

    glVertex2f(0.54f, -0.47f);
    glVertex2f(0.51f, -0.47f);
  glEnd();
   //... Block - 4 ...//End--------------


  /// Block - 5 ... //505
  glBegin(GL_QUADS);
    glColor3ub(138, 3, 42);
    glVertex2f(0.58f, -0.53f);
    glVertex2f(0.77f, -0.53f);

    glVertex2f(0.82f, -0.49f);
    glVertex2f(0.63f, -0.49f);
  glEnd();
   //... Block - 5 ...//End--------------


   /// Block - 6 ... //506
  glBegin(GL_QUADS);
    glColor3ub(181, 57, 57);
    glVertex2f(0.43f, -0.45f);
    glVertex2f(0.74f, -0.45f);

    glVertex2f(0.74f, -0.43f);
    glVertex2f(0.43f, -0.43f);
  glEnd();
   //... Block - 6 ...//End--------------
```

```
 /// Block - 7 ... //507
glBegin(GL_QUADS);
  glColor3ub(250, 250, 150);
  glVertex2f(0.52f, -0.43f);
  glVertex2f(0.56f, -0.43f);

  glVertex2f(0.56f, -0.38f);
  glVertex2f(0.52f, -0.38f);
glEnd();
//... Block - 7 ...//End--------------


 /// Block - 8 ... //508
glBegin(GL_QUADS);
  glColor3ub(250, 250, 150);
  glVertex2f(0.56f, -0.43f);
  glVertex2f(0.71f, -0.43f);

  glVertex2f(0.71f, -0.36f);
  glVertex2f(0.56f, -0.36f);
glEnd();
 //... Block - 8 ...//End--------------


 /// Block - 9 ... //509
glBegin(GL_QUADS);
  glColor3ub(60, 70, 180);
  glVertex2f(0.64f, -0.43f);
  glVertex2f(0.67f, -0.43f);

  glVertex2f(0.67f, -0.37f);
  glVertex2f(0.64f, -0.37f);
glEnd();
 //... Block - 9 ...//End--------------


 /// Block - 10 ... //510
glBegin(GL_QUADS);
  glColor3ub(200, 40, 90);
  glVertex2f(0.65f, -0.40f);
  glVertex2f(0.66f, -0.40f);

  glVertex2f(0.66f, -0.38f);
  glVertex2f(0.65f, -0.38f);
glEnd();
 //... Block - 10 ...//End--------------


 /// Block - 11 ... //511
glBegin(GL_QUADS);
  glColor3ub(200, 40, 90);
  glVertex2f(0.59f, -0.40f);
  glVertex2f(0.63f, -0.40f);

  glVertex2f(0.63f, -0.37f);
  glVertex2f(0.59f, -0.37f);
glEnd();
 //... Block - 11 ...//End--------------
 /// Block - 12 ... //512
glBegin(GL_QUADS);
  glColor3ub(200, 40, 90);
  glVertex2f(0.68f, -0.40f);
  glVertex2f(0.70f, -0.40f);

  glVertex2f(0.70f, -0.37f);
  glVertex2f(0.68f, -0.37f);
glEnd();
 //... Block - 12 ...//End--------------


 /// Block - 13 ... //513
glBegin(GL_QUADS);
  glColor3ub(181, 57, 57);
  glVertex2f(0.55f, -0.36f);
  glVertex2f(0.72f, -0.36f);

  glVertex2f(0.72f, -0.35f);
  glVertex2f(0.55f, -0.35f);
glEnd();
 //... Block - 13 ...//End--------------


 /// Block - 14 ... //514
glBegin(GL_QUADS);
  glColor3ub(138, 3, 42);
  glVertex2f(0.55f, -0.35f);
  glVertex2f(0.73f, -0.35f);

  glVertex2f(0.74f, -0.34f);
  glVertex2f(0.55f, -0.34f);
glEnd();
 //... Block - 14 ...//End--------------


 /// Block - 15 ... //515
glBegin(GL_QUADS);
  glColor3ub(181, 57, 57);
  glVertex2f(0.50f, -0.38f);
  glVertex2f(0.58f, -0.38f);

  glVertex2f(0.58f, -0.37f);
  glVertex2f(0.50f, -0.37f);
glEnd();
 //... Block - 15 ...//End--------------


 /// Block - 16 ... //516
glBegin(GL_QUADS);
  glColor3ub(60, 160, 160);
  glVertex2f(0.38f, -0.53f);
  glVertex2f(0.41f, -0.53f);

  glVertex2f(0.41f, -0.52f);
  glVertex2f(0.38f, -0.52f);
```

```
glEnd();
 //... Block - 16 ...//End--------------


 /// Block - 17 ... //517
glBegin(GL_QUADS);
  glColor3ub(60, 160, 160);
  glVertex2f(0.41f, -0.53f);
  glVertex2f(0.43f, -0.53f);

  glVertex2f(0.43f, -0.52f);
  glVertex2f(0.41f, -0.52f);
glEnd();
 //... Block - 17 ...//End--------------


 /// Block - 18 ... //518
glBegin(GL_QUADS);
  glColor3ub(60, 160, 160);
  glVertex2f(0.43f, -0.53f);
  glVertex2f(0.45f, -0.53f);

  glVertex2f(0.45f, -0.52f);
  glVertex2f(0.43f, -0.52f);
glEnd();
 //... Block - 18 ...//End--------------


 /// Block - 19 ... //519
glBegin(GL_QUADS);
  glColor3ub(60, 160, 160);
  glVertex2f(0.38f, -0.52f);
  glVertex2f(0.41f, -0.52f);

  glVertex2f(0.41f, -0.51f);
  glVertex2f(0.38f, -0.51f);
glEnd();
 //... Block - 19 ...//End--------------


 /// Block - 20 ... //520
glBegin(GL_QUADS);
  glColor3ub(60, 160, 160);
  glVertex2f(0.43f, -0.52f);
  glVertex2f(0.45f, -0.52f);

  glVertex2f(0.45f, -0.51f);
  glVertex2f(0.43f, -0.51f);
glEnd();
 //... Block - 20 ...//End--------------


 /// Block - 21 ... //521 //Flag - 1
glBegin(GL_QUADS);
  glColor3ub(60, 70, 180);
  glVertex2f(0.36f, -0.42f);
  glVertex2f(0.38f, -0.42f);
```

```
    glVertex2f(0.38f, -0.4f);
    glVertex2f(0.36f, -0.4f);
  glEnd();
   //... Block - 21 ...//End--------------


   /// Block - 22 ... //522 //Flag - 2
  glBegin(GL_QUADS);
    glColor3ub(60, 70, 180);
    glVertex2f(0.34f, -0.43f);
    glVertex2f(0.36f, -0.43f);

    glVertex2f(0.36f, -0.41f);
    glVertex2f(0.34f, -0.41f);
  glEnd();
   //... Block - 22 ...//End--------------


   /// Block - 23 ... //523 //Light - Back
  glBegin(GL_QUADS);
    glColor3ub(255, 90, 50);
    glVertex2f(0.67f, -0.32f);
    glVertex2f(0.69f, -0.32f);

    glVertex2f(0.69f, -0.31f);
    glVertex2f(0.67f, -0.31f);
  glEnd();
   //... Block - 23 ...//End--------------


   /// Block - 24 ... //524 //Light - Front
  glBegin(GL_QUADS);
    glColor3ub(255, 90, 50);
    glVertex2f(0.70f, -0.33f);
    glVertex2f(0.71f, -0.33f);

    glVertex2f(0.71f, -0.32f);
    glVertex2f(0.70f, -0.32f);
  glEnd();



///-----------------Stand Blocks-----------------///

   /// Block - 25 ... //525 //Flag Stand
  glBegin(GL_QUADS);
    glColor3ub(78, 75, 75);
    glVertex2f(0.38f, -0.53f);
    glVertex2f(0.39f, -0.53f);

    glVertex2f(0.39f, -0.38f);
    glVertex2f(0.38f, -0.38f);
  glEnd();
   //... Block - 25 ...//End--------------


   /// Block - 26 ... //526 //Light Stand - Back
  glBegin(GL_QUADS);
```

```
    glColor3ub(78, 75, 75);
    glVertex2f(0.66f, -0.34f);
    glVertex2f(0.67f, -0.34f);

    glVertex2f(0.68f, -0.32f);
    glVertex2f(0.67f, -0.32f);
  glEnd();
   //... Block - 26 ...//End--------------


   /// Block - 27 ... //527 //Light Stand - Front
  glBegin(GL_QUADS);
    glColor3ub(78, 75, 75);
    glVertex2f(0.70f, -0.34f);
    glVertex2f(0.70f, -0.34f);

    glVertex2f(0.70f, -0.33f);
    glVertex2f(0.70f, -0.33f);
  glEnd();


///-------------------------------------------------------------------------------///



glPopMatrix();



/// ====================--------RIVER--------====================////////////////
// Design For grass background
   glBegin(GL_QUADS);
   glColor3ub(0, 220, 255);
    glVertex2f(-1.0f, -0.8f);
    glVertex2f(1.0f, -0.8f);

    glVertex2f(1.0f, -0.66f);
    glVertex2f(-1.0f, -0.66f);
   glEnd();

// Box for hide tree extra design
   glBegin(GL_QUADS);
   glColor3ub(0, 220, 255);
    glVertex2f(-1.0f, -1.0f);
    glVertex2f(1.0f, -1.0f);

    glVertex2f(1.0f, -0.80f);  /// ei value change korte hobe
    glVertex2f(-1.0f, -0.80f);  // age .85 chilo
   glEnd();

   glMatrixMode(GL_MODELVIEW);
///---------------------------------------------Rain Function
  rainFunction();

  glFlush();
```

```
}
///----------------------------------------Rain update function-----------------------------//////

void updateRain(int value)
{

   if(rain == true){
      if(xrain > 0.1f){
         xrain = -0.2f;
         yrain = -0.5f;
      }
      xrain = xrain + 0.03f;
      yrain = yrain - 0.03f;
   }

   else{

      xrain = 10.0f;
      yrain = 10.0f;

   }
   glutPostRedisplay();

   glutTimerFunc(25, updateRain,0);

    glFlush();


}
///----------------------------------------Rain update function ENDS-----------------------------//////


void MyInit()
{
glEnable(GL_DEPTH_TEST);
gluOrtho2D(0,1600,0,800);
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_COLOR_MATERIAL);

}

void init() {
  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
  //glEnable(GL_DEPTH_TEST);
  gluOrtho2D(0,1600,0,800);
  glEnable(GL_LIGHTING);
  glEnable(GL_LIGHT0);
  glEnable(GL_COLOR_MATERIAL);
}


int main(int argc, char** argv) {



   cout << "///----------------=======================COMPUTER GRAPHICS PROJECT===================---------
----///"<<endl;
```

```
   cout << "///----------------================>>> DYNAMIC WEATHER SIMULATOR <<<==================-----------
--///"<<endl ;

   cout << "Press D : For Day" << endl ;
   cout << "Press N : For Night"<< endl;
   cout << "Press A : For Autumn"<< endl;
   cout << "Press R : For Rain "<< endl;
   cout << "Press S : For stop Rain "<< endl;



  glutInit(&argc, argv);

  glutInitWindowSize(1240, 840);
  glutInitWindowPosition(50, 50);
  glutCreateWindow("DWS");
  glutDisplayFunc(display);
  init();

  glutKeyboardFunc(handleKeypress);



  glutTimerFunc(25, updateRain, 0);
  glutTimerFunc(25, boatUpdate, 0);
  glutTimerFunc(25, rightWaveUpdate, 0);
  glutTimerFunc(25, leftWaveUpdate, 0);
  glutTimerFunc(25, rightCarUpdate, 0);
  glutTimerFunc(25, leftCarUpdate, 0);
  glutTimerFunc(25, cloud1Update, 0);
  glutTimerFunc(25, cloud2Update, 0);
  glutMainLoop();
  return 0;
}
```

Significance of the Project

The Dynamic Weather Simulator project holds significance in various domains, contributing to:

1. **Education and Training:**
    - Serving as an immersive educational tool, the project offers students and enthusiasts an interactive environment to understand fundamental concepts of weather simulation.
    - Its user-friendly interface and realistic simulations enhance learning experiences, fostering a deeper understanding of meteorological phenomena.

2. **Research and Development:**
    - Researchers can leverage insights from the simulator to enhance existing models, driving advancements in accuracy and efficiency in weather prediction and simulation.
3. **Public Awareness and Engagement:**
    - Weather-related events have significant societal impacts, necessitating informed decision-making and community preparedness.
4. **Innovation and Creativity:**
    - Its visually captivating virtual environments stimulate innovation and creativity, encouraging exploration and experimentation in weather simulation technology.

In summary, the Dynamic Weather Simulator project plays a pivotal role in education, research, public engagement, and innovation, contributing to a deeper understanding of weather dynamics and enhancing resilience in the face of weather-related challenges.

Conclusion

As we wrap up, it's clear that the Dynamic Weather Simulator project has truly made waves in providing a user-friendly platform for exploring the wonders of weather dynamics. Through its intuitive interfaces and lifelike simulations, it's not just a tool for learning—it's a window into a world of discovery and understanding.

But its impact goes beyond the realm of academia. By sparking curiosity and engagement, it's empowering everyday folks to connect with and respond to the weather in meaningful ways. And let's not forget its role as a catalyst for creativity and collaboration, bringing people together to tackle real-world challenges.

Looking ahead, the journey of the project is far from over. With new technologies and fresh perspectives on the horizon, there's endless potential for further exploration and innovation. It's exciting to think about how this project will continue to shape our understanding of the natural world and inspire us to embrace its mysteries.

So, in closing, the Dynamic Weather Simulator project isn't just about data and simulations—it's about people and their connection to the world around them. It's a reminder that curiosity knows no bounds and that together, we can unlock the secrets of nature and build a brighter future.

<u>Screenshots of the System</u>:

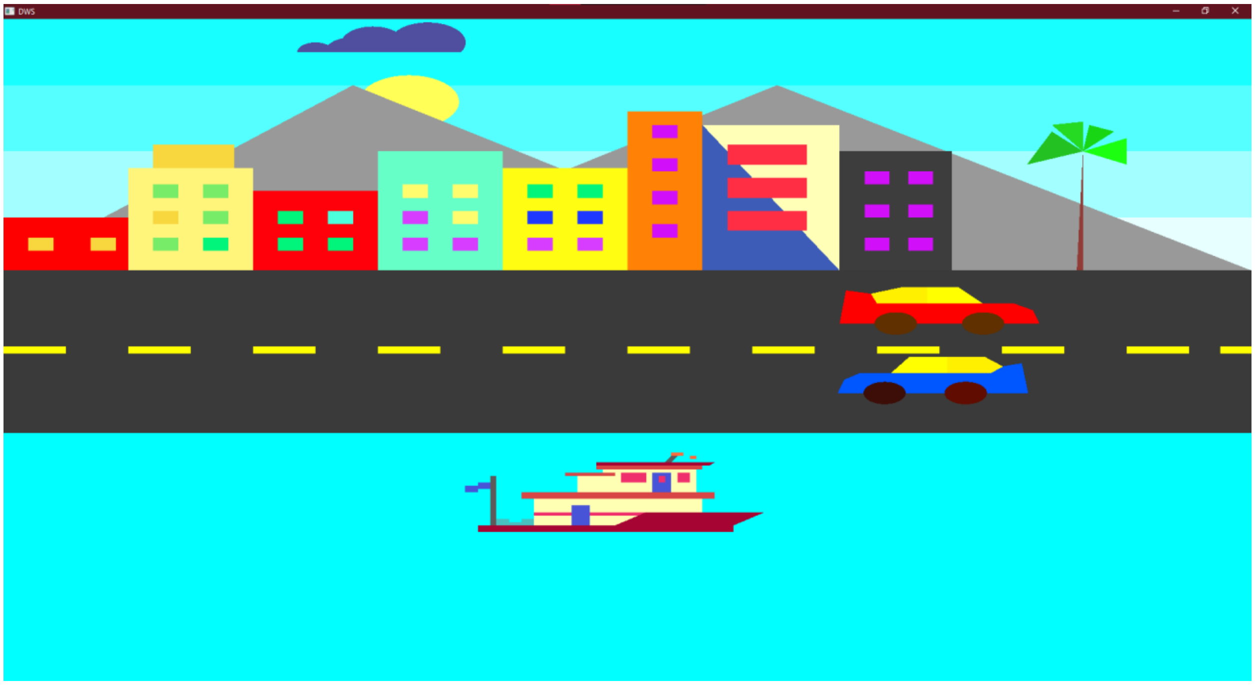


Fig 1: Day Time

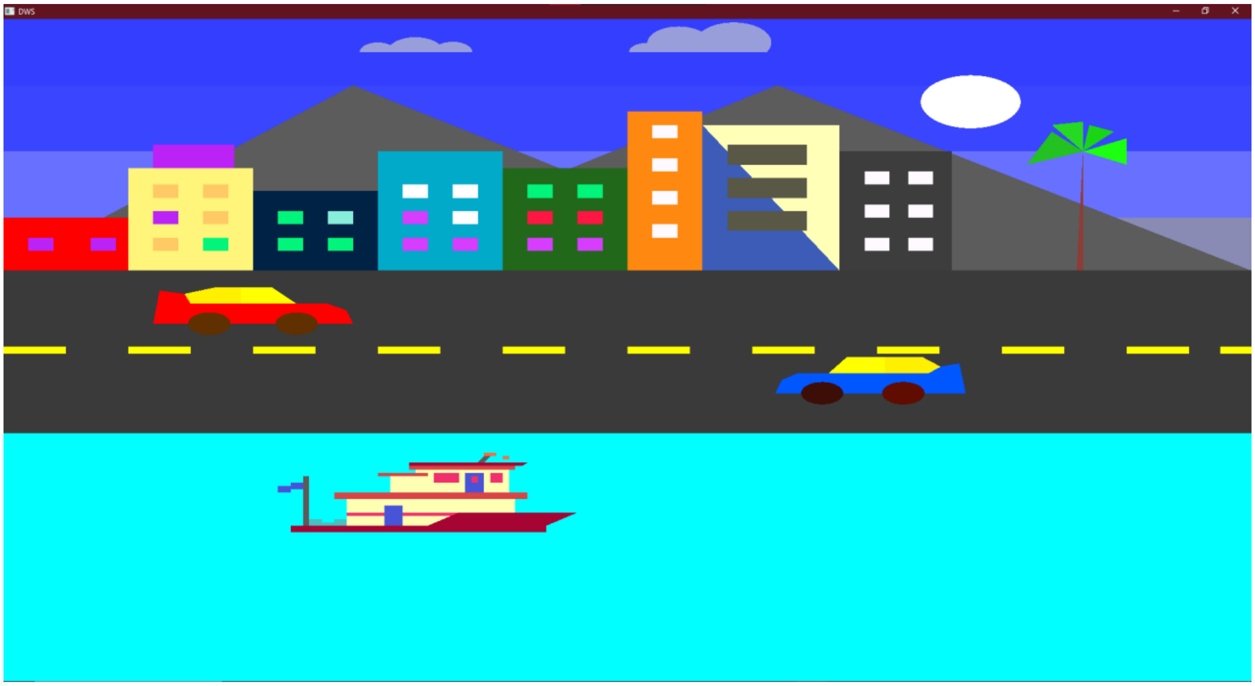


Fig 2: Night Time

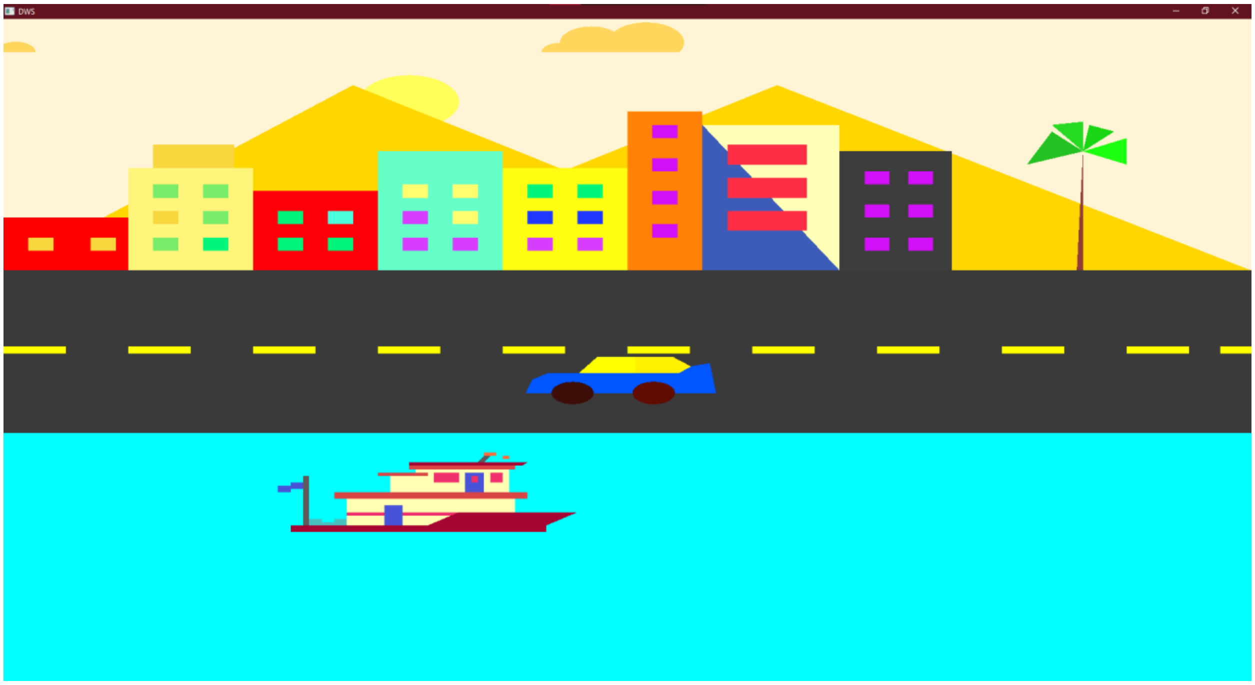


Fig 3: Autumn Season

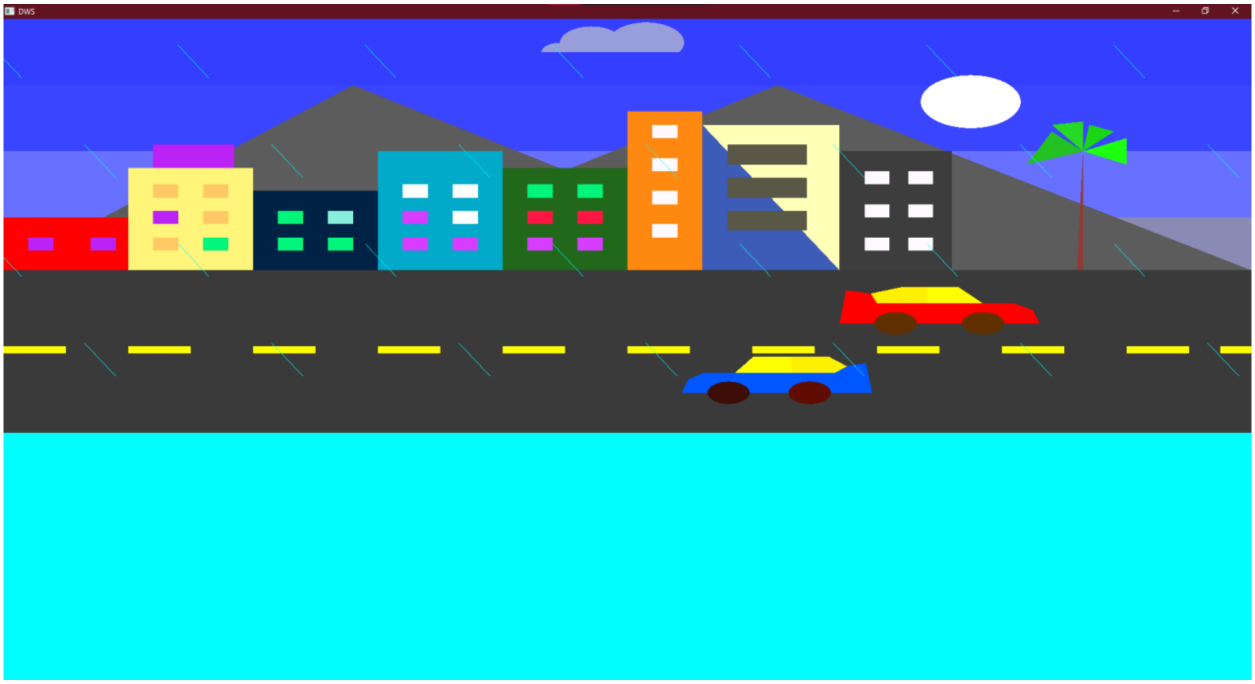
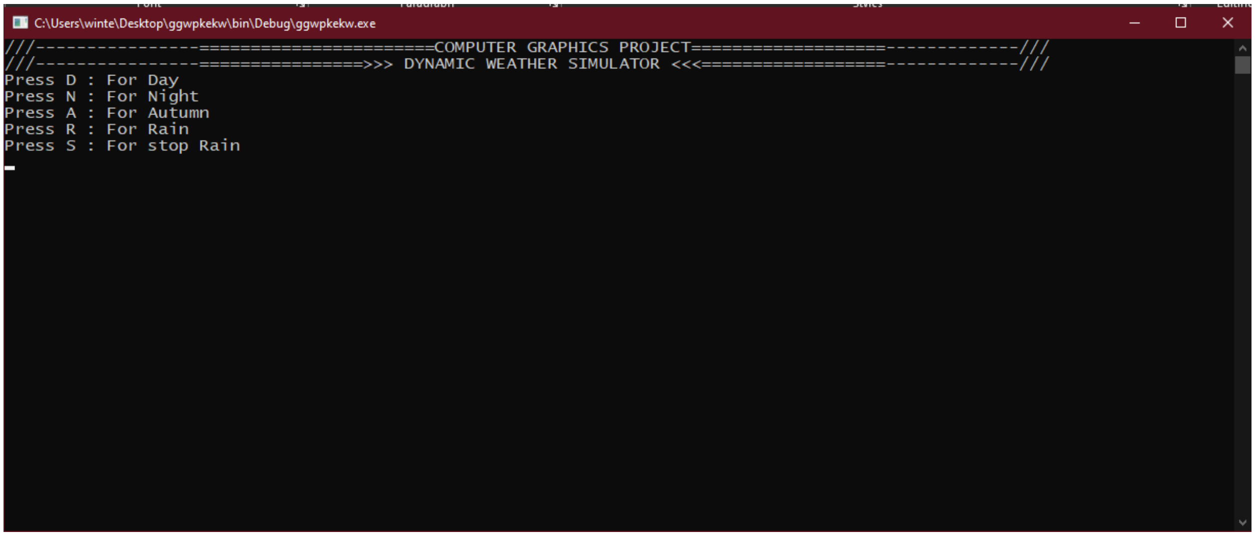


Fig 4: Rain on Night



```
///----------------=======================COMPUTER GRAPHICS PROJECT===================-------------///
///----------------================>>> DYNAMIC WEATHER SIMULATOR <<<==================-------------///
Press D : For Day
Press N : For Night
Press A : For Autumn
Press R : For Rain
Press S : For stop Rain
```

Fig 5: Button Functions