AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)

FACULTY OF SCIENCE sTECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE COMPUTER GRAPHICS

Spring 2024-2025

Section: O, Group: 10

Project: "Village Scenario"

Supervise By:

ANEEM AL AHSAN RUPAI

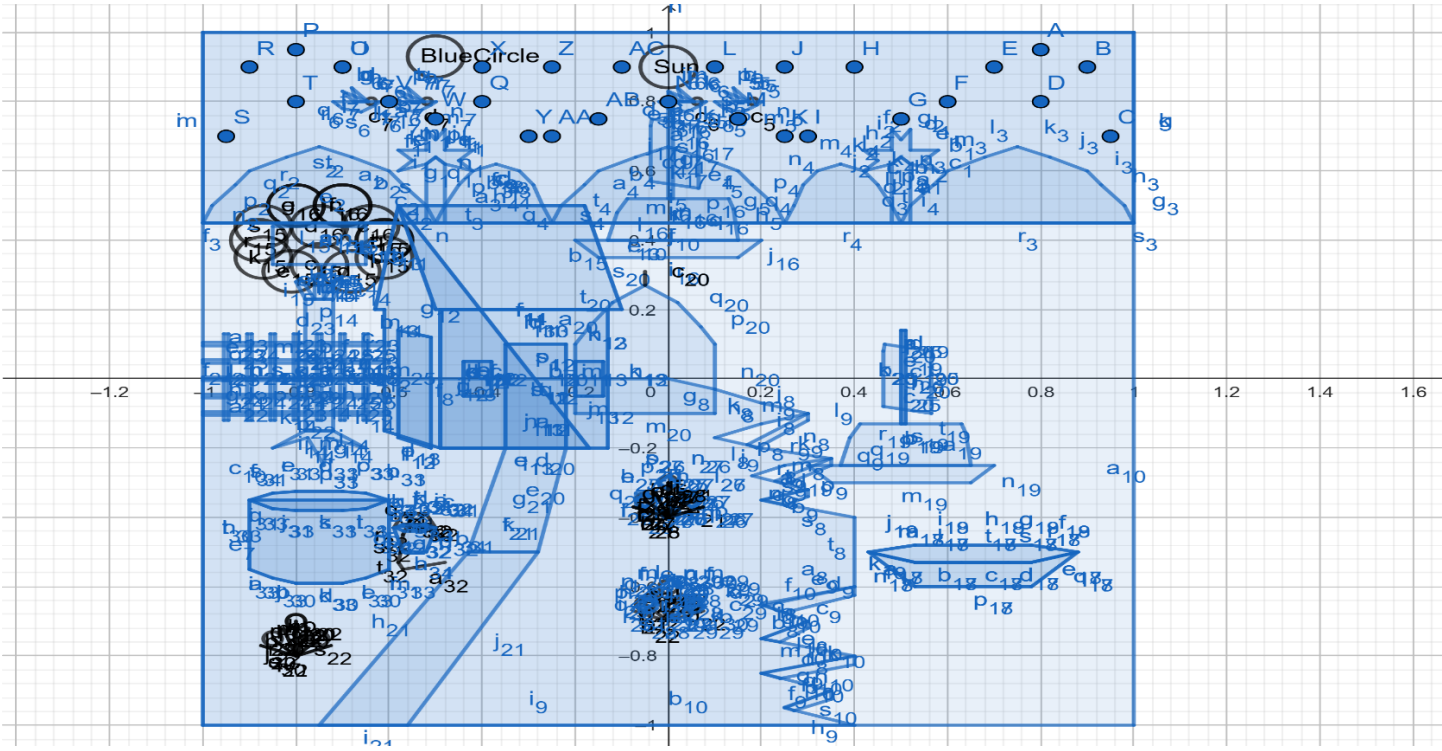
Student Information		
SL No.	**Name**	**ID**
1.	MD . MOZAHIDUL ISLAM	23-51293-1

Introduction:

This project presents a 2D animated village scene implemented using OpenGL (GLUT) in C++. The main objective is to create a visually engaging simulation of a serene village environment with dynamic elements and user interactivity. The scene includes:

1. Day and Night Modes with sun, moon, and stars, toggled via keyboard input ('D'/'N').
2. Animated Rain Effect activated with 'R' key for a realistic weather simulation.
3. Moving Clouds and Birds with smooth translations across the sky.
4. Animated Boat moving across the river, enhancing the dynamic environment.
5. Detailed Village Elements including huts with doors/windows, trees, a well, straw piles, and fences.
6. Interactive Pathways and Grass with colorful flower accents for added realism.
7. Sound Effects for day (birds), night (crickets), and rain, triggered with scene changes.

Project Graph:



List of Functions Used for Creating Objects
The following functions are defined in your code to create and render the objects in the village scene:

1. **cloud1()**: Draws the first cloud using multiple `GL_TRIANGLE_FAN` primitives to create circular shapes, forming a cloud with a light pink color (`255, 217, 255`) at coordinates centered around `(0.5, 0.86)` and nearby points.

2. **cloud2()**: Draws the second cloud, similar to `cloud1`, using `GL_TRIANGLE_FAN` primitives with the same color, positioned around `(-0.5, 0.86)`.

3. **cloud3()**: Draws the third cloud, also using `GL_TRIANGLE_FAN` primitives, colored light pink, centered around `(0.0, 0.86)`.

4. **sky()**: Renders the daytime sky as a blue quad (`51, 204, 255`) covering the upper part of the scene from `y=0.45` to `y=1.0`.

5. **sky2()**: Renders the nighttime sky as a darker blue quad (`0, 51, 204`) covering the same region as `sky()`.

6. **backgroundtree()**: Draws multiple trees in the background using `GL_POLYGON` primitives. Includes palm trees with brown trunks (`102, 51, 0`) and green foliage (`34, 139, 34`) at various positions (e.g., `(-0.5, 0.45)`, `(0.5, 0.45)`).

7. **bird()**: Renders four animated birds, each with a body (`GL_POLYGON`), wings (`GL_TRIANGLES`), and head (`GL_TRIANGLE_FAN`) in shades of gray (`225, 225, 208`, `217, 217, 217`, `242, 242, 242`).

8. **stars()**: Draws stars in the night sky using `GL_POINTS` with white color (`255, 255, 255`) at various coordinates across the upper part of the scene.

9. **sun()**: Renders the sun as a yellow circle (`255, 204, 0`) using `GL_TRIANGLE_FAN` centered at `(0.0, 0.9)`.

10. **moon()**: Draws the moon using two `GL_TRIANGLE_FAN` primitives: a dark blue circle (`0, 51, 204`) and a white crescent (`242, 242, 242`) at `(-0.5, 0.9)`.

11. **ground()**: Creates the ground terrain using a green `GL_POLYGON` (`102, 255, 51`) for the main area and multiple brown quads (`153, 153, 102`) for paths or patches.

12. **river()**: Renders the river as a blue quad (`38, 154, 214`) covering the lower part of the scene from `y=-1.0` to `y=0.45`.

13. **hut()**: Draws a hut with a sloped roof and walls using `GL_POLYGON` primitives in brown (`204, 153, 0`) and darker brown (`153, 115, 0`) for doors and windows, outlined with black lines (`0, 0, 0`).

14. **hut1()**: An alternative rendering of the hut, similar to `hut()`, with the same structure but includes only the outline and filled polygons, possibly for a different view or state.

15. **tree()**: Renders a detailed tree with a brown trunk (`102, 51, 0`) using `GL_POLYGON` and `GL_QUADS`, and green foliage (`51, 204, 51`) using multiple `GL_TRIANGLE_FAN` circles.

16. **boat()**: Draws a boat (assumed to be animated due to the `boat()` timer function) with a specific structure, though the rendering code is not fully shown in the provided snippet. It is referenced in the `display()` function with translation for

animation.

17. **boat2()**: Renders a second boat using `GL_LINES` and `GL_POLYGON` primitives with black (`0, 0, 0`), gray (`122, 122, 82`), and brown (`204, 153, 0`) colors, positioned around `(0.43, -0.5)`.

18. **boat3()**: Draws a third boat, similar to `boat2`, with black and gray polygons and lines, positioned slightly differently in the scene.

19. **boat4()**: Renders a fourth boat with black (`0, 0, 0`), orange (`255, 153, 0`), red (`255, 25, 25`), and green (`136, 204, 0`) polygons, positioned around `(0.3, -0.25)`.

20. **Straw()**: Draws a straw pile using a yellow `GL_POLYGON` (`255, 219, 77`) and a brown line (`153, 153, 102`) for a pole, centered around `(-0.05, 0.27)`.

21. **way()**: Renders pathways using brown quads (`153, 153, 102`) connecting the hut to other parts of the scene, such as from `(-0.35, -0.2)` to `(-0.75, -1.0)`.

22. **grass1()**, **grass2()**, **grass3()**, **grass4()**, **grass5()**, **grass6()**: Draw grass patches with green lines (`0, 102, 0`) and colorful flowers (`255, 51, 0`, `255, 102, 0`, `255, 255, 0`) using `GL_LINES` and `GL_TRIANGLE_FAN` at various positions (e.g., `(0.0, -0.4)`, `(0.0, -0.7)`, `(-0.8, -0.8)`).

23. **fence()**: Renders a fence using yellow lines (`255, 255, 102`) with horizontal and vertical segments from `(-1.0, -0.1)` to `(-0.6, 0.1)`.

24. **well()**, **well1()**: Draw a well with a red-brown base (`204, 51, 0`), orange top (`255, 102, 51`), gray structure (`194, 194, 163`), and blue water (`38, 154, 214`) using `GL_POLYGON` and `GL_LINES`, centered around `(-0.75, -0.35)`.

Notes
- **Animation Functions**: While not directly rendering objects, the following timer functions control the animation of specific objects:
  - `cloud(int value)`: Animates clouds by translating their positions (`position2`, `speed2 = 0.004`).
  - `birdd(int value)`: Animates birds (`position22`, `speed22 = 0.007`).
  - `sunn(int value)`: Animates the sun (`position4`, `speed4 = -0.01`).

  - `boat(int value)`: Animates the first boat (`position1`, `speed1 = -0.005`).
  - `rain(int value)`: Animates rain particles (`position3`, `speed3 = -0.5`).
- **Other Functions**: The following functions are not directly for creating objects but are critical for the scene:
  - `PointLight()`: Sets up lighting for the scene, affecting object appearance.
  - `StartingText()`: Displays text (title, instructions, credits) but does not create scene objects.
  - `DrawSphere()`, `display()`, `display1()`, `display2()`: Composite display functions that call object-rendering functions for different modes (day, rain, night).
  - `reshape()`, `init()`, `handleKeypress()`: Handle window setup, initialization, and user input for mode switching (not object creation).

Summary
The 24 functions listed above (`cloud1`, `cloud2`, `cloud3`, `sky`, `sky2`, `backgroundtree`, `bird`, `stars`, `sun`, `moon`, `ground`, `river`, `hut`, `hut1`, `tree`, `boat`, `boat2`, `boat3`, `boat4`, `Straw`, `way`, `grass1`, `grass2`, `grass3`, `grass4`, `grass5`, `grass6`, `fence`, `well`, `well1`) are directly responsible for creating and rendering the objects in your village scene. These functions use OpenGL primitives to draw the visual elements, with some objects (e.g., clouds, birds, boat, sun, rain) animated via timer functions. If you need further details about any specific function or want to refine this list (e.g., to include animation functions or exclude certain elements), please let me know!

Output Image(Day):