Experiment 9

Student Name: Gaganjot Singh **UID: 22BCS14843**
Branch: B.E. CSE III Yr **Section:** 22BCS-IOT-612-B
Semester: 6th
Subject Name: Computer Graphics with Lab **Subject Code:** 22CSH-352

1. **Aim:** Demonstrate the result of window-to-viewport transformation by implementing and visualizing the process.

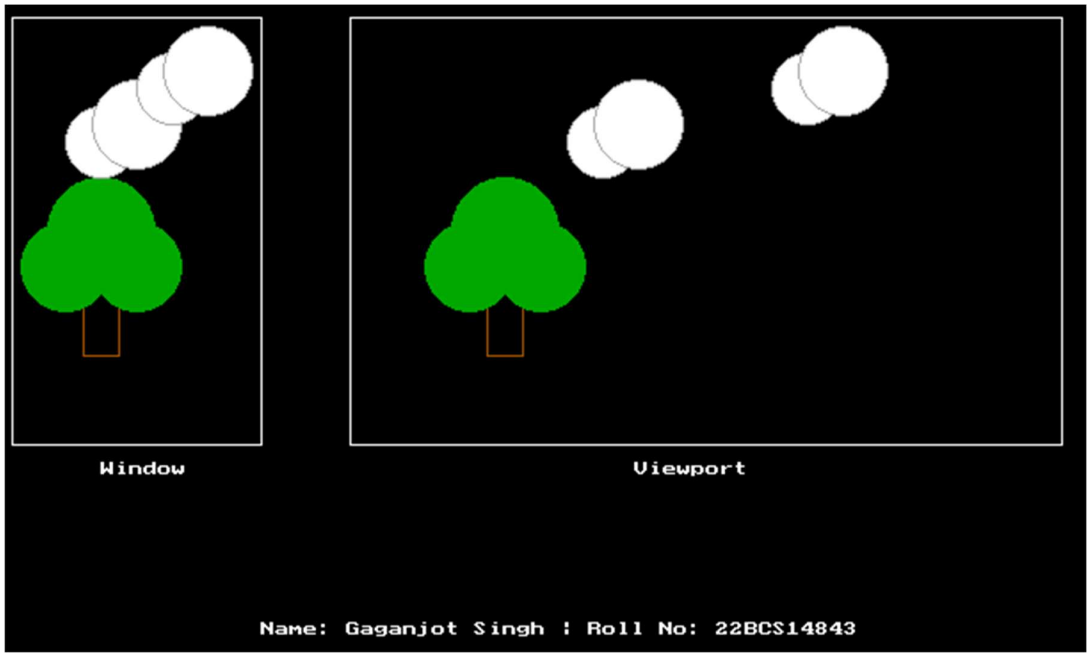
2. **Objective:** Window-to-viewport transformation by mapping world coordinates (window) to device coordinates (viewport) through scaling and translation, visually showcasing the resulting changes in size and position.

3. **Code:**

```
#include <iostream.h>
#include <graphics.h>
#include <conio.h>

void drawTree(int baseX, int baseY)
{
    setcolor(BROWN);
    rectangle(baseX + 20, baseY - 60, baseX + 40, baseY);

    setcolor(GREEN);
    setfillstyle(SOLID_FILL, GREEN);
    fillellipse(baseX + 30, baseY - 70, 30, 30);
    fillellipse(baseX + 10, baseY - 50, 25, 25);
    fillellipse(baseX + 50, baseY - 50, 25, 25);}

void drawCloud(int cx, int cy)
{
    setcolor(LIGHTGRAY);
    setfillstyle(SOLID_FILL, WHITE);
    fillellipse(cx, cy, 20, 20);
    fillellipse(cx + 20, cy - 10, 25, 25);}

void main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

    outtextxy(150, 350, "Name: Gaganjot Singh | Roll No: 22BCS14843");

    float wxmin = 10, wymin = 10, wxmax = 150, wymax = 250;
    float vxmin = 200, vymin = 10, vxmax = 600, vymax = 250;
    rectangle(wxmin, wymin, wxmax, wymax);
    rectangle(vxmin, vymin, vxmax, vymax);
    outtextxy(60, 260, "Window");
```

```
outtextxy(360, 260, "Viewport");

float sx = (vxmax - vxmin) / (wxmax - wxmin);
float sy = (vymax - vymin) / (wymax - wymin);

drawTree(30, 200);

drawCloud(60, 80);
drawCloud(100, 50);


int tx = 30, ty = 200;
int tx_view = sx * (tx - wxmin) + vxmin;
int ty_view = sy * (ty - wymin) + vymin;
drawTree(tx_view, ty_view);


int cx1 = 60, cy1 = 80;
int cx1_view = sx * (cx1 - wxmin) + vxmin;
int cy1_view = sy * (cy1 - wymin) + vymin;
drawCloud(cx1_view, cy1_view);

int cx2 = 100, cy2 = 50;
int cx2_view = sx * (cx2 - wxmin) + vxmin;
int cy2_view = sy * (cy2 - wymin) + vymin;
drawCloud(cx2_view, cy2_view);

getch();
closegraph();}
```

4. Output:



5. Learning Outcome:

i. Learnt how to represent a scene (like a tree and clouds) inside a "window" and how to map that same scene into another area called the "viewport." The window represents the original coordinates or design area, while the viewport is where the scaled output is displayed.

ii. Scaling objects from one coordinate system (window) to another (viewport) using scaling factors for both x and y directions. This helps in keeping the proportions of the objects the same while fitting them into a different space.

iii. sx = (vxmax - vxmin) / (wxmax - wxmin);
sy = (vymax - vymin) / (wymax - wymin);
Here, sx is the scaling factor in the x-direction and sy is for the y-direction. (wxmin, wymin) and (wxmax, wymax) define the window's corners. (vxmin, vymin) and (vxmax, vymax) define the viewport's corners.

iv. After calculating the scaling factors, used them to transform each object's position from the window to the viewport using the formula:
vx = sx * (wx - wxmin) + vxmin;
vy = sy * (wy - wymin) + vymin;

v. Created functions like drawTree() and drawCloud() to organize my code and draw objects more easily.