Experiment No. 10
Program for Animation
Name: AARYAN CHANDRAKANT GOLE
Roll Number: 12
Date of Performance:
Date of Submission:

Experiment No. 10

Aim: To develop programs for making animations such as Solar System.

Objective:

Draw an object and apply various transformation techniques to this object. Translation, scaling and rotation is applied to object to perform animation.

Theory:

- For moving any object, we incrementally calculate the object coordinates and redraw the picture to give a feel of animation by using for loop.
- Suppose if we want to move a circle from left to right means, we have to shift the position of circle along x-direction continuously in regular intervals.
- The below programs illustrate the movement of objects by using for loop and also using transformations like rotation, translation etc.
- For windmill rotation, we use 2D rotation concept and formulas.

Program:

```
#include <conio.h>
#include <dos.h>
#include <graphics.h>
#include <math.h>
#include <stdio.h>

// Function to manipulates the position of planets on the orbit
void planetMotion(int xrad, int yrad,
                  int midx, int midy,
                  int x[70], int y[70])
{
   int i, j = 0;

   // Positions of planets in their corresponding orbits
   for (i = 360; i > 0; i = i - 6) {
      x[j] = midx - (xrad * cos((i * 3.14) / 180));
      y[j++] = midy - (yrad * sin((i * 3.14) / 180));
   }

   return;
}
```

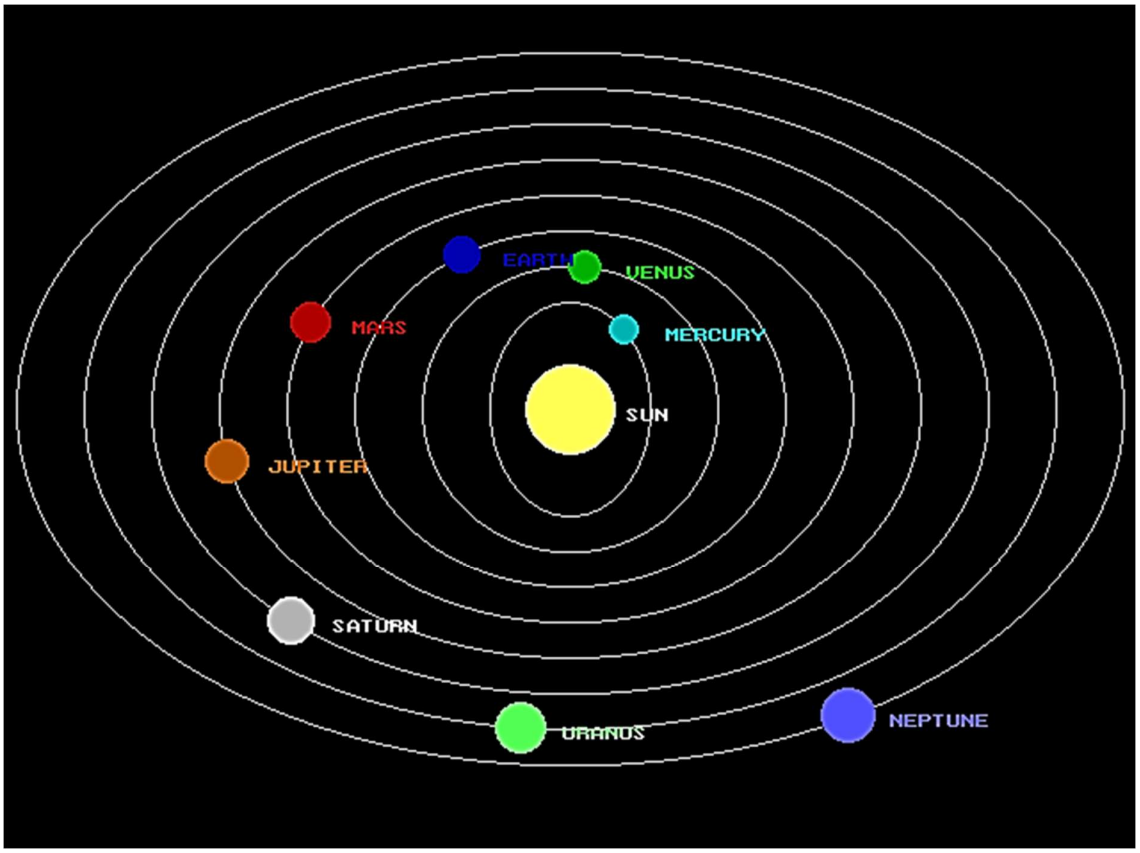
```
// Driver Code
int main()
{

    // Initialize graphic driver
    int gdriver = DETECT, gmode, err;
    int i = 0, midx, midy;
    int xrad[8], yrad[8], x[8][70], y[8][70];
    int pos[8], planet[8], tmp;

    initgraph(&gdriver, &gmode, "C:\\Turboc3\\BGI");
    err = graphresult();

    if (err != grOk) {

        // Error occurred
        printf("Graphics Error: %s",
            grapherrormsg(err));
        return 0;
    }

    // Mid positions at x and y-axis
    midx = getmaxx() - 320;
    midy = getmaxy() - 250;

    // Manipulating radii of all the eight planets
    planet[0] = 8;
    for (i = 1; i < 8; i++) {
        planet[i] = planet[i - 1] + 1;
    }

    // Offset position for the planets on their corresponding orbit
    for (i = 0; i < 8; i++) {
        pos[i] = i * 6;
    }

    // Orbits for all 8 planets
    xrad[0] = 45, yrad[0] = 60;
    for (i = 1; i < 8; i++) {
        xrad[i] = xrad[i - 1] + 38;
        yrad[i] = yrad[i - 1] + 20;
    }

    // Positions of planets on their corresponding orbits
```

```
for (i = 0; i < 8; i++) {
   planetMotion(xrad[i], yrad[i],
            midx, midy, x[i],
            y[i]);
}

while (!kbhit()) {

   // Drawing 8 orbits
   setcolor(WHITE);
   for (i = 0; i < 8; i++) {
      setcolor(WHITE);
      ellipse(midx, midy, 0, 360,
            xrad[i], yrad[i]);
   }

   // Sun at the mid of solar system
   outtextxy(midx, midy, "    SUN");
   setcolor(YELLOW);
   setfillstyle(SOLID_FILL, YELLOW);
   circle(midx, midy, 25);
   floodfill(midx, midy, YELLOW);

   // Mercury in first orbit
   setcolor(CYAN);

   setfillstyle(SOLID_FILL, CYAN);
   outtextxy(x[0][pos[0]],
         y[0][pos[0]],
         "   MERCURY");

   pieslice(x[0][pos[0]],
         y[0][pos[0]],
         0, 360, planet[0]);

   // Venus in second orbit
   setcolor(GREEN);
   setfillstyle(SOLID_FILL, GREEN);
   outtextxy(x[1][pos[1]],
         y[1][pos[1]],
         "   VENUS");
   pieslice(x[1][pos[1]],
         y[1][pos[1]],
         0, 360, planet[1]);
```

```
// Earth in third orbit
setcolor(BLUE);
setfillstyle(SOLID_FILL, BLUE);
outtextxy(x[2][pos[2]],
      y[2][pos[2]],
      "   EARTH");
pieslice(x[2][pos[2]],
      y[2][pos[2]],
      0, 360, planet[2]);

// Mars in fourth orbit
setcolor(RED);
setfillstyle(SOLID_FILL, RED);
outtextxy(x[3][pos[3]],
      y[3][pos[3]],
      "   MARS");
pieslice(x[3][pos[3]],
      y[3][pos[3]],
      0, 360, planet[3]);

// Jupiter in fifth orbit
setcolor(BROWN);
setfillstyle(SOLID_FILL, BROWN);
outtextxy(x[4][pos[4]],
      y[4][pos[4]],
      "   JUPITER");
pieslice(x[4][pos[4]],
      y[4][pos[4]],
      0, 360, planet[4]);

// Saturn in sixth orbit
setcolor(LIGHTGRAY);
setfillstyle(SOLID_FILL, LIGHTGRAY);
outtextxy(x[5][pos[5]],
      y[5][pos[5]],
      "   SATURN");
pieslice(x[5][pos[5]],
      y[5][pos[5]],
      0, 360, planet[5]);

// Uranus in seventh orbit
setcolor(LIGHTGREEN);
setfillstyle(SOLID_FILL, LIGHTGREEN);
      outtextxy (x [6] [pos [6]],
```

```
                y [6] [pos [6]],
                "   URANUS");
            pieslice (x [6] [pos [6]],
                y [6] [pos [6]],
                0, 360, planet [6]);

        // Neptune in eighth orbit
        setcolor (LIGHTBLUE);
        setfillstyle (SOLID_FILL, LIGHTBLUE);
        outtextxy (x [7] [pos [7]],
            y [7] [pos [7]],
            "   NEPTUNE");
        pieslice (x [7] [pos [7]],
            y [7] [pos [7]],
            0, 360, planet [7]);

        // Checking for one complete
        // rotation
        for (i = 0; i < 8; i++) {
            if (pos[i] <= 0) {
                pos[i] = 59;
            }
            else {
                pos[i] = pos[i] - 1;
            }
        }

        // Sleep for 100 milliseconds
        delay (100);

        // Clears graphic screen
        cleardevice ();
    }

    // Deallocate memory allocated for graphic screen
    // or closing the graphics window
    closegraph();

    return 0;
}
```

Output:


EARTH
VENUS
MARS
MERCURY
SUN
JUPITER
SATURN
URANUS
NEPTUNE

Conclusion:

In this program, we utilized various header files like, the graphics.h library for drawing, dos.h for timing, and math.h for mathematical calculations. We set up a simulation of a solar system, which involves initializing the graphics window, running the simulation loop, and finally, cleaning up resources and exiting the program. This program allows for the visual representation of a solar system using basic C graphics capabilities.

Comment on –

1. **Importance of story building-** Story building is crucial as it forms the backbone of any narrative, providing structure, depth, and coherence to the plot. It engages the audience, creates emotional connections, and allows for meaningful exploration of themes and characters.

2. **Defining the basic character of story**- The basic character of a story encompasses its fundamental elements, including the protagonist, antagonist, setting, and central conflict. These elements lay the foundation for the narrative, shaping the events and driving the plot forward. A well-defined basic character is essential for a compelling and relatable story.

3. **Apply techniques to these characters-** Applying techniques to characters involves imbuing them with depth, complexity, and relatability. This can be achieved through methods such as character development arcs, dialogue that reveals their personalities, and vivid descriptions. Techniques also include providing characters with motivations, flaws, and growth, allowing them to resonate with the audience and contribute meaningfully to the narrative.