Name : Abdullah Rajput

Roll no: BSIT-2K22-04

Subject : Computer Graphics

Assignment : Basic Transformations in 2D

# Introduction

The world of computer graphics relies heavily on manipulating objects on the screen. These manipulations are achieved through transformations, which alter the position, size, and orientation of objects. This assignment explores three fundamental 2D transformations: translation, rotation, and scaling. We will delve into their concepts, mathematical representations, and practical applications.

# Translation

Imagine moving a picture frame across a wall. That's essentially translation! In 2D graphics, translation refers to shifting an object from one location to another without changing its size or orientation. It's achieved by specifying a horizontal (x) and vertical (y) distance to move the object.

## Mathematical Representation:

Points in 2D space are represented by coordinates (x, y). To translate a point by a distance of (tx, ty), we simply add these values to the original coordinates:

New X = Original X + tx
New Y = Original Y + ty

## Practical Examples:

- Animating a character walking across the screen involves translating its position frame by frame.
- In image editing software, moving a selected image region employs translation.

# Rotation

Rotation involves turning an object around a fixed point (often the object's center) by a specific angle. Imagine a book spinning on a table. Here, the table acts as the fixed point, and the book rotates around it.

## Mathematical Representation:

Rotation is calculated using trigonometric functions (sine and cosine) based on the desired angle (θ). Here's the transformation matrix for rotating a point (x, y) around the origin (0, 0):

| cos(θ)  -sin(θ) |   | x |
| sin(θ)   cos(θ) |   | y | = (New X, New Y)

## Practical Examples:

- Rotating a game character to face a different direction uses rotation.

- Simulating the movement of a clock's hands involves rotation around the center.

# Scaling

Scaling alters the size of an object, either enlarging or shrinking it proportionally. Think of inflating a balloon – it's getting bigger while maintaining its shape. Scaling is defined by a scaling factor (sx, sy) for the x and y directions, respectively.

## Mathematical Representation:

Scaling is achieved by multiplying the original coordinates with the scaling factors:

New X = Original X * sx
New Y = Original Y * sy

- A scaling factor greater than 1 (e.g., 2) enlarges the object.
- A scaling factor between 0 and 1 (e.g., 0.5) shrinks the object.
- Equal scaling factors (sx = sy) maintain the object's proportions.

## Practical Examples:

- Zooming in on an image involves scaling it up.

- Resizing a window on your computer screen is a scaling operation.

# Combining Transformation

The power of transformations lies in their ability to be combined to create complex effects. For example, imagine animating a ball bouncing across the screen. This animation might involve:

- **Translation:** Moving the ball's position left and right.
- **Scaling:** Slightly compressing the ball as it touches the ground (simulating squish).
- **Rotation:** Rotating the ball slightly with each bounce.

These transformations can be applied sequentially or combined into a single matrix for efficiency.

# Conclusion

This assignment has explored the fundamental 2D transformations of translation, rotation, and scaling. By understanding their concepts and mathematical representations, we can manipulate objects in computer graphics, creating dynamic and visually appealing experiences. With further

exploration, these basic transformations can be combined to achieve more complex effects, pushing the boundaries of computer-generated visuals.

# Further Exploration

This assignment provides a basic understanding. Consider exploring:

- **Shearing:** Tilting an object along a specific axis.
- **Composite transformations:** Applying multiple transformations in sequence.
- **Transformation matrices:** Performing complex transformations with single matrices.