Malaviya National Institute of Technology Jaipur
Department of Computer Science Engineering
Computer Graphics Lab
Quiz, Date: April 25, 2024

Time: 10:00 am-11:00 am (1 hour)   Spring 2024, VI Semester

1. In the context of 3D Computer graphics, which of the following statements is/are correct?
   (I). Under perspective projection, each set of parallel lines in the object does not stay parallel in the image (except those that are parallel to the view plane to start with).
   (II). Applying a perspective transformation in the graphics pipeline to a vertex involves dividing by its 'z' coordinate.
   (III). Perspective transformation is a linear transformation.
   Choose the correct answer from the options given below:
   (a). (I) and (II) only
   (b). (I) and (III) only
   (c). (II) and (III) only
   (d). (I), (II), and (III)

2. In the context of 3D computer graphics, which of the following statements is/are true?
   P: Orthographic transformations keep parallel lines parallel.
   Q: Orthographic transformations are affine transformations.
   Select the correct answer from the options given below:
   (a). Both P and Q
   (b). Neither P nor Q
   (c). Only P
   (d). Only Q

3. In 2D-translation, a point (x, y) can move to the new position (x', y') by using the equation
   (a). x'=x+dx and y'=y+dx
   (b). x'=x+dx and y'=y+dy
   (c). X'=x+dy and Y'=y+dx
   (d). X'=x-dx and y'=y-dy

4. In OpenGL, select the correct option(s) related to the gluPerspective(fovy , aspect , zNear , zFar) function:
   (a). fovy specifies the aspect ratio that determines the field of view in the x direction.
   (b). fovy Specifies the field of view angle, in degrees, in the y direction.

(c). zNear Specifies the field of view angle, in degrees, in the y direction.

(d). zFar Specifies the distance from the viewer to the far clipping plane

5. Complete the following python code for implementation of Bresenham's Line Drawing Algorithm:

```python
def draw_line(x0, y0, x1, y1):
    dx = abs(x1 - x0)
    dy = abs(y1 - y0)
    sx = 1 if x0 < x1 else -1
    sy = 1 if y0 < y1 else -1

    if dy > dx:
        dx, dy = dy, dx
        interchange = True
    else:
        interchange = False


    _____
    x = x0
    y = y0

    points = [(x, y)]

    for i in range(dx):
        while e >= 0:
            if interchange:
                x += sx
            else:
                y += sy
            e -= 2 * dx
        if interchange:

            _____
        else:
            x += sx
        e += 2 * dy
        points.append((x, y))

    return points
```

(a). e = dy - dx , y += sy

(b). e = 2 * dx - dy , y += sy

(c). e = 2 * dy - dx , y += sy

(d). e = 2 * dy - dx , y += sx