Computer Graphics Mid-Term Exam Answers

Q.1 Answer the following.

(a) Short questions (1 mark each)

1. How does antialiasing improve visual quality in graphics?
Antialiasing smooths jagged edges in digital images by averaging color values between pixels, reducing the staircase effect and enhancing visual quality.

2. Define viewport in the context of graphical display.
A viewport is a rectangular area where the content of a window is displayed. It defines the portion of the scene visible to the user.

3. Define multimedia in the context of digital content.
Multimedia refers to the integration of various forms of digital content, such as text, images, audio, video, and animations, to enhance user interaction and communication.

(b) Objective Type/MCQs/True-False/Fill in blanks (1 mark each)

1. In a raster scan, the frame buffer is continuously read to refresh the ____.
Answer: screen

2. What does DDA stand for in the context of line-drawing algorithms?
Answer: (A) Digital Differential Analysis

3. What does the term "viewport" refer to in computer graphics?
Answer: (B) A rectangular area where the window's content is displayed.

4. To combine multiple transformations such as rotation, scaling, and translation into a single transformation is called ____.
Answer: Composite transformation

5. If you want to adjust the properties of an object on the stage, ____ panel would you use?
Answer: Properties

6. True/False: Scan line filling works by processing one horizontal line at a time to determine which pixels should be filled inside a polygon.
Answer: True

7. True/False: The normalized view volume is always a rectangular box, regardless of the projection type used.
Answer: False

Q.2 Answer the following.

(a) Two Questions of 2 Marks

1. Explain what is meant by viewport clipping.
Viewport clipping is the process of restricting the rendering of objects or graphics to the boundaries of a viewport. It ensures that only the visible portion of a scene is displayed on the screen.

2. Explain the difference between boundary fill and flood fill algorithms.

- **Boundary Fill**: Starts filling an enclosed area from a seed point and stops when a boundary color is encountered.
- **Flood Fill**: Fills all connected pixels of a similar color, regardless of predefined boundaries.

(b) Two Questions of 3 Marks

1. How would you implement a circle-drawing algorithm using the mid-point algorithm?
The mid-point circle algorithm determines pixel positions using incremental calculations:

- Start from (0, R) and plot symmetrical points in all eight octants.
- Compute the decision parameter to determine the next pixel.

- Update the coordinates and repeat until x > y.

2. Analyze how geometric transformations such as scaling and rotation affect the position of a graphical object.

- **Scaling**: Changes the size of the object by multiplying its coordinates with scaling factors.
- **Rotation**: Moves the object around a pivot point, changing its orientation.
- **Combined Effects**: Applying these transformations in sequence can alter the position and appearance of an object in a graphical scene.

Q.3 Attempt any TWO. (5 marks each)

1. Given a polygon defined by its vertices, outline the steps to implement the scan line filling algorithm to fill it with color.

- Identify the edges of the polygon and determine the scan lines that intersect it.
- Find intersection points of scan lines with polygon edges.
- Sort the intersection points from left to right.
- Fill pixels between each pair of intersection points.

2. Demonstrate how Bresenham's algorithm can be used to draw a line from (1,1) to (5,4).

Steps:

- **Step 1: Compute Values**

  Δx = 5 - 1 = **4**

  Δy = 4 - 1 = **3**

- **Initial Decision Parameter (P_0):**

  $P_0 = 2\Delta y - \Delta x = 2(3) - 4 = 6 - 4 = 2$

Step 3: move towards (5,4)

- If $\mathbf{P \geq 0}$, move **diagonally** (right + up) and update:

  $P=P+2\Delta y-2\Delta x$

- If $\mathbf{P < 0}$, move **horizontally** (right only) and update:

  $P=P+2\Delta y$

Step: 4 Efficiency Analysis

Step	Current (x, y)	P Value	Move Type	New P Calculation	Next (x, y)
1	(1,1)	2	Diagonal (right + up)	$P_1 = 2 + 6 - 8 = 0$	(2,2)
2	(2,2)	0	Diagonal (right + up)	$P_2 = 0 + 6 - 8 = -2$	(3,3)
3	(3,3)	-2	Horizontal (right only)	$P_3 = -2 + 6 = 4$	(4,3)
4	(4,3)	4	Diagonal (right + up)	$P_4 = 4 + 6 - 8 = 2$	(5,4)

3. Analyze the steps to create a basic masking effect in Flash. What types of objects should be used for effective masking?

- Create a **new file,**
- Insert **an image, text, or any graphic** that you want to mask.
- Draw a shape **circle.**
- If you want the mask to move, **convert it into a movie clip Convert to Symbol.**
- Right-click on the **Mask layer** and **Select "Mask".**
- Now, only the area covered by the mask shape will be visible.
- If you want movement, **create a motion tween** on the mask layer.
- Press **Ctrl + Enter** to preview.

Q.4 Answer the following.

(a) Step-by-step explanation and analysis of Bresenham's line-drawing algorithm. (5 marks)

Step 1: Initialize Variables

- Identify the starting point (x_1, y_1) and the ending point (x_2, y_2).
- $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$.

• Parameter: $P_0 = 2\Delta y - \Delta x$.

Step 2: Plot the First Point

• Start from (x_1, y_1) and plot it on the screen.

Step 3: Iterate and Decide Next Pixel

• For each step in the x-direction:

- If $P \geq 0$, for diagonally: $P = P + 2\Delta y - 2\Delta x$.

- If $P < 0$, for horizontally: $P = P + 2\Delta y$.

• Repeat until the endpoint (x_2, y_2) is reached.

Step 4: Efficiency Analysis

• Uses only integer calculations (addition and subtraction), making it faster than floating-point calculations.

• Ideal for real-time graphics in low-power or embedded systems.

• Works best for shallow slopes (0° to 45°), but can be adapted for steeper slopes using coordinate swapping.

(b) Design an animation using shape motion tweening where a square smoothly transforms into a circle. What key steps are involved? (5 marks)

- Create a new file.
- Select **Frame 1** in **Layer 1**.
- Use the **Rectangle Tool** to draw a **square** on the stage.
- Convert it into a **shape.**
- Right-click on **Frame 1** and select **"Create Shape Tween"**.
- Click on Frame **30fps Press F6** to insert a keyframe.
- Use the **Oval Tool (O)** to replace the square with a **circle** at the same position.
- Ensure the shape tween is applied correctly.
- If needed, use **Shape Hints.**
- Press **Ctrl + Enter**

OR

(b) Demonstrate how to create a smooth motion tween to move an object from left to right. What key steps are involved? (5 marks)

- Create an object (e.g., a rectangle) on the stage.
- Convert it to a symbol (movie clip).
- Insert a new keyframe at a later frame and move the object to the right.
- Apply classic motion tweening.
- Adjust easing and test animation.
