

COMPUTER GRAPHICS LAB - QUESTION BANK

OpenGL Basics and Environment

1. **Q:** What is OpenGL?
A: OpenGL is a cross-platform API for rendering 2D and 3D vector graphics.
2. **Q:** What is the role of GLUT in OpenGL programs?
A: GLUT manages windows and user input in OpenGL programs.
3. **Q:** What are the main primitives in OpenGL?
A: Points, lines, and polygons.
4. **Q:** How is the background color set in OpenGL?
A: Using `glClearColor(r, g, b, a)`.
5. **Q:** What is `glBegin()` used for?
A: It begins the definition of a geometric primitive.
6. **Q:** How do you set the viewport in OpenGL?
A: Using `glViewport(x, y, width, height)`.
7. **Q:** What does `glFlush()` do?
A: It forces execution of OpenGL commands.
8. **Q:** What is the default coordinate range in OpenGL?
A: From -1 to 1 in both X and Y (normalized device coordinates).
9. **Q:** What function is used to initialize a window with GLUT?
A: `glutCreateWindow("Title")`.
10. **Q:** What is `gluOrtho2D()` used for?
A: It defines a 2D orthographic projection matrix.

Line and Circle Drawing

11. **Q:** What does DDA stand for?
A: Digital Differential Analyzer.

12. **Q:** How does DDA work?
A: It increments one coordinate and calculates the other using slope.
13. **Q:** What is Bresenham's line algorithm based on?
A: It uses integer calculations and decision variables.
14. **Q:** Which is more efficient: DDA or Bresenham?
A: Bresenham is more efficient due to no floating-point operations.
15. **Q:** How can you draw a dotted line?
A: By plotting every alternate pixel.
16. **Q:** How is a dashed line implemented?
A: By skipping a fixed number of pixels after drawing some.
17. **Q:** What are the 4 types of lines you can draw?
A: Simple, Dotted, Dashed, Solid.
18. **Q:** How do you detect mouse input in OpenGL?
A: Using `glutMouseFunc()`.
19. **Q:** How is the screen divided into four quadrants in OpenGL?
A: By mapping coordinates with origin (0, 0) at the center.
20. **Q:** Can Bresenham's algorithm draw lines for all slopes?
A: Yes, by adjusting iteration based on slope.
21. **Q:** What is the basic idea behind Bresenham's circle algorithm?
A: Use symmetry and decision parameter to plot pixels.
22. **Q:** How many octants are used in the circle algorithm?
A: Eight octants (due to symmetry).
23. **Q:** How do you replicate a circle in all quadrants?
A: By mirroring points with respect to center (0, 0).
24. **Q:** How is radius input handled in your circle algorithm?
A: Either via keyboard or mouse selection.
25. **Q:** What happens if radius = 0?
A: Only the center pixel is drawn.
-

Filling and Clipping

26. **Q:** What is flood fill?

A: It fills connected pixels of the same color starting from a seed.

27. **Q:** What is boundary fill?

A: It fills until a boundary color is encountered.

28. **Q:** What causes stack overflow in flood fill?

A: Deep recursion on large areas.

29. **Q:** What is the difference between 4-connected and 8-connected fills?

A: 4-connected checks up/down/left/right; 8-connected includes diagonals.

30. **Q:** What are the inputs needed for flood fill?

A: Seed point, fill color, and background color.

31. **Q:** What is polygon clipping?

A: Trimming parts of a polygon outside the viewport.

32. **Q:** What is a viewport?

A: A rectangular area on the screen where graphics are displayed.

33. **Q:** What is a window in clipping?

A: The logical coordinate region of interest.

34. **Q:** What is the role of outcodes in Cohen-Sutherland?

A: To identify location of endpoints with respect to the window.

35. **Q:** How many regions are defined in Cohen-Sutherland?

A: Nine regions (inside + 8 outside zones).

36. **Q:** What does bitwise AND of outcodes tell you?

A: If both points are outside on the same side.

37. **Q:** What does bitwise OR of outcodes tell you?

A: If at least one point is outside the window.

38. **Q:** What is trivial rejection?

A: Discarding lines completely outside the clipping region.

39. **Q:** How do you accept a line trivially?

A: If both endpoints have outcode 0000.

40. **Q:** How are clipped lines redrawn?

A: By recalculating intersection points.

41. **Q:** What happens to a line entirely inside the window?

A: It is accepted as is.

42. **Q:** How are mouse and keyboard used in clipping programs?

A: To define window corners and polygon points.

43. **Q:** What is the main difference between line and polygon clipping?

A: Polygon clipping is more complex due to multiple edges.

44. **Q:** Can Cohen-Sutherland clip concave polygons?

A: No, it's only suitable for lines and convex polygons.

45. **Q:** How do you handle cases where the polygon lies outside the viewport?

A: Entire polygon is rejected.

Transformations and Fractals

46. **Q:** What are the basic 2D transformations?

A: Translation, scaling, rotation, reflection, and shearing.

47. **Q:** What is scaling?

A: Changing the size of an object.

48. **Q:** What is rotation about origin?

A: Turning the object around the origin point.

49. **Q:** How do you rotate about an arbitrary point?

A: Translate to origin → rotate → translate back.

50. **Q:** What is reflection in 2D?

A: Mirroring an object about a line/axis.

51. **Q:** What is a transformation matrix?

A: A matrix used to apply a transformation to coordinates.

52. **Q:** What are homogeneous coordinates?

A: Coordinates with an extra dimension to simplify transformations.

53. **Q:** What is matrix composition?

A: Applying multiple transformations using one combined matrix.

54. **Q:** What is the order of transformations important?

A: Because matrix multiplication is not commutative.

55. **Q:** What is a Bezier curve?

A: A smooth curve defined by control points.

56. **Q:** How many control points does a cubic Bezier have?
A: Four.
57. **Q:** What is the role of Bernstein polynomials?
A: They define the weights in Bezier curves.
58. **Q:** How are Bezier curves generated in OpenGL?
A: By iteratively calculating points using blending functions.
59. **Q:** What is a Koch curve?
A: A fractal made by recursively replacing line segments with a pattern.
60. **Q:** What is the base case of Koch curve?
A: A straight line.
61. **Q:** How many segments replace a line in Koch recursion?
A: Four.
62. **Q:** What is a fractal?
A: A self-similar and infinitely complex geometric figure.
63. **Q:** What is recursion's role in fractals?
A: It repeatedly applies a pattern.
64. **Q:** What is self-similarity?
A: The property where parts of an object resemble the whole.
65. **Q:** How are Bezier curves useful in graphics?
A: For modeling smooth curves and surfaces.
66. **Q:** What happens when scaling factors are <1 ?
A: The object shrinks.
67. **Q:** What does a negative scaling factor do?
A: It reflects the object.
68. **Q:** How is reflection across Y-axis achieved?
A: Multiply x-coordinates by -1.
46. **Q:** How do you reflect across origin?
A: Multiply both x and y by -1.
47. **Q:** What is the difference between uniform and differential scaling?
A: Uniform scales all dimensions equally.
48. **Q:** What is the identity matrix?
A: A matrix that doesn't change the object when applied.
49. **Q:** What is inverse transformation?

A: A transformation that undoes another.

50. **Q:** What is pivot point in rotation?

A: The point around which rotation is done.

51. **Q:** Why are floating-point operations needed in rotation?

A: To handle trigonometric calculations.

52. **Q:** What happens when you rotate by 360 degrees?

A: The object returns to its original position.

Animation Principles

76. **Q:** What is animation?

A: A sequence of frames creating the illusion of motion.

77. **Q:** What are the 12 principles of animation?

A: Including squash/stretch, anticipation, timing, etc.

78. **Q:** What is keyframe animation?

A: Using key positions at intervals to define motion.

79. **Q:** What is interpolation?

A: Generating intermediate frames between keyframes.

80. **Q:** What is tweening?

A: The process of creating in-between frames.

81. **Q:** What is frame rate?

A: Number of frames displayed per second.

82. **Q:** What is the purpose of double buffering?

A: To avoid flickering during animation.

83. **Q:** What is `glutTimerFunc()`?

A: A function to schedule timed callbacks.

84. **Q:** What is real-time animation?

A: Animation that responds to user inputs or system events.

85. **Q:** What is a transformation hierarchy?

A: Parent-child relationships in animated objects.

86. **Q:** What is looped animation?

A: Repeating an animation cycle indefinitely.

87. **Q:** What is motion blur?

A: Simulated blur to enhance realism.

88. **Q:** What causes animation lag?
A: Slow rendering or unoptimized code.
89. **Q:** How is speed controlled in animation?
A: By adjusting time intervals between frames.
90. **Q:** How can mouse be used in animation?
A: To trigger or control motion.
91. **Q:** How can keyboard control animation?
A: By detecting key presses to move or transform objects.
92. **Q:** What is an animation path?
A: A trajectory followed by an animated object.
93. **Q:** What is the role of `glutIdleFunc()`?
A: It runs a function when the application is idle.
94. **Q:** What is sprite animation?
A: Using 2D images to simulate motion.
95. **Q:** What is morphing?
A: Smooth transformation from one shape to another.
96. **Q:** How do you stop an animation?
A: Disable the timer or idle function.
97. **Q:** What is physics-based animation?
A: Animation driven by physical simulations like gravity.
98. **Q:** How do you animate transformations?
A: By gradually changing parameters (e.g., angle or scale).
99. **Q:** What is the difference between static and dynamic scenes?
A: Static scenes don't change; dynamic ones do.
100. **Q:** How do you synchronize animation with sound?
A: Using time stamps and audio playback APIs.