1 Computer Graphics

1. What is a pixel?
2. What is a pixmap?
3. How can pixmaps be created?
4. How can pixmaps be displayed?
5. What is a raster display? Give examples.
6. What is a vector display? Give examples.
7. What is the vertical refresh rate of a raster display? How is it related to the pixel clock?
8. What determines the frame rate of a vector display?

2 OpenGL Basics

1. What is the rendering pipeline? Where does it start and where does it end?
2. What is OpenGLs function?
3. What is GLUT?
4. What is the function of the GLUT callback functions?
5. What is typically done in a GLUT display function?
6. What is the structure of the OpenGL function names?
7. How are graphical objects created using glBegin() glVertex and glEnd()?
8. What is “algorithmic drawing”? Give an example.
9. What is “relative drawing”? Give an example.

3 Vector and curve basics

1. How can curves be represented? What is the use of the different representations in computer graphics?
2. What is clipping? Why do we need clipping?
3. Give an example for a clipping algorithm.
4. How can you detect the intersection between a polygon and a line? Is there an advantage if the line is aligned with the coordinate system?
5. How can points and vectors be represented in a common format? Why is this common format useful?
6. What classes of transformations are used in computer graphics? Where are they used?

7. In order to scale, rotate and move an object, what transforms would be applied to its vector coordinates? Could you do it in a single matrix? What would the matrix look like for a rotation of +90 degrees around the z axis, a scaling by 2 and a translation by 4 along the z axis?

8. How can points, lines and planes be represented using vectors? How can you find different representations (parametric, implicit...) from these vectors?

4 OpenGL Coordinates, Transformations and Projections

1. How are the OpenGL coordinate systems defined?

2. What is the “camera”? What different methods do you know to “place” the camera?

3. OpenGL uses a number of matrices that represent transformations. What matrices are there, how can they be modified, what is their function?

4. OpenGL maintains a stack for one matrix type. Why? What relationship does this stack has with hierarchical modeling?

5. How dows OpenGL implement perspective? What types of perspective can be used in OpenGL? What do these different types of perspective do?

6. What is the view volume? How is it related to the viewport?

5 Modeling solid objects

1. How can solid objects be represented for computer graphics?

2. What is a polyhredron?

3. What physical processes can OpenGL model for surfaces? What coefficients can be specified?

4. How does OpenGL create the color of a surface? Is this the only way of specifying a color?

5. When rendering colored objects, two coefficients are used, α and γ. What is their function?

6. How does OpenGL create a textured object?

7. What methods of shading does OpenGL implement? Are there other methods?

8. What is the difference between shading and ray tracing?

9. What is Lambert’s Law?

10. Mr. Phong is famous for two methods used in shading, both published in 1975. One ended up in OpenGL, the other one could not be implemented. What are these? Why could one not be implemented?

11. Finding the mirror direction $\vec{r}$ for one of the methods of Mr. Phong is expensive. why? What is the alternative?

12. What is the relationship between textures and shading? How can OpenGL assure that textures are rendered with the right perspective distortion? What is “wrong” with Gouraud shading in this respect?

13. What types of light can be used in OpenGL? How can these be specified?

14. How can smooth objects be approximated? What is the advantage of using generic shapes for approximation?

15. Why do approximated objects appear smooth when shading is used?

16. How can a z-Buffer be used to remove hidden parts of objects?

17. What is pseudodepth? How can it be computed using the prespective transformation?