# Beantwortung der Fragen

1. Man sieht ein mittig platziertes, rotes Rechteck mit schwarzem Hintergrund.

2. Die Anweisungen sind notwendig, da man sonst nichts (ein schwarzes Fenster) sieht.

3. Die Anweisung glColor ändert die Farbe der darauffolgend gezeichneten Vertices. glColor ist einmal aufzurufen und gilt solange, bis wieder eine Änderung der Farbe vorgenommen wird. Entsprechend muss der erste Aufruf (rot) vor dem ersten Vertex (-0.5, -0.5, -0.5) stehen, um die untere linke Ecke zu färben, und der Zweite zwischen dem ersten und dem zweiten Vertex, um den Rest blau zu färben.

4. Die geeignetste Position ist in der Methode RenderScene in der ersten Zeile. glClear wird zum Vorbereiten der Szene benötigt und leert die im Parameter festgelegten Buffer mit einem einheitlichen Wert (also im Falle des Anzeigebuffers einer einheitlichen Farbe).

5. glClearColor muss vor glClear aufgerufen werden. Man setzt die Farbe, um sie dann zu verwenden. Theoretisch bräuchte man das nicht beachten, da nach einem „falschen“ Schleifendurchlauf die Löschfarbe gesetzt wurde und somit ab dem 2. Frame korrekt geleert wird.

6. Es würden Spuren entstehen, falls man nicht den gesamten Bildinhalt im neuen Frame überschreibt, da die alten Bilder aus dem vorhergegangenen Frame nicht gelöscht wurden. Falls man den Tiefenbuffer verwendet und ebenfalls nicht leert würde man nur Dinge sehen, solange sie auf den Betrachter zukommen, da die Tiefeninformationen ebenfalls über Frames hinweg erhalten bleiben.

7. Man sieht die grüne Fläche, obwohl sie eigentlich tiefer im Raum liegt (kleinere Z-Koordinate), weil sie als letztes gezeichnet wird.

8. Die zuletzt gezeichnete Fläche sieht man, unabhängig von der Z-Koordinate.

9a. Ja. Man sieht ein rot-blaues Rechteck, obwohl es zuerst gezeichnet wird. Die Z-Koordinate stellt das Rechteck in den Vordergrund, unabhängig von der Zeichenreihenfolge

9b. Der Z-Buffer speichert Tiefeninformationen der bereits gezeichneten Geometrien. Bei jedem neu gezeichneten Punkt wird zunächst der Z-Buffer gelesen und überprüft, ob dieser Punkt sichtbar ist oder nicht, indem die Z-Koordinate des zu zeichnenden Pixels mit dem Wert im Z-Buffer verglichen wird. Ist der Wert näher am Betrachter, wird der Pixel gezeichnet und der Z-Buffer aktualisiert, ansonsten wird der Pixel verworfen, da der bereits vorhandene Pixel den zu Zeichnenden verdeckt.

10. Man sieht die Rückseite des Würfels von innen betrachtet (blau, cyan, schwarz, grün), da sich die Kamera im Ursprung befindet und in Richtung der negativen Z-Achse blickt.

11a. gluLookAt(0,1,1, 0,0,0, 0,1,0). Nein, sie sind falsch gesetzt, da das „far“ zu klein eingestellt ist und somit die Sichtweite nicht bis zum Würfel ausreicht.

11b. glOrtho(-1,1, -1,1, 0.0, 2.0);

gluLookAt(1,0,0, 0,0,0, 0,1,0);

11c. Kameraposition (1,1,0)

12. Ja, der „rotate“ Befehl kann vom „oberen“ Teil in beide Äste des Szenegraphs verlagert werden. (dieselbe Rotation ist dann doppelt im Szenengraph):



13. Wie bereits in Aufgabe 6 vermutet, hinterlassen bewegte Geometrien überall dort, wo im nächsten Frame nichts gezeichnet wird, Spuren.