

Fragen des Einführungsteils zu OpenGL

`glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);`

Frage 1: Weshalb sollte dies in jedem Frame erfolgen? (Was kann insbesondere bei Animationen passieren, wenn man die Initialisierung weglässt?)

Da im Color- und z-Buffer anfangs undefinierte, vom vorherigen `render()` Aufruf stammende Daten stehen, muss nach jedem neu zeichnen eines Frames „`glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)`“ aufgerufen werden, um den Hintergrund auf den in `init()` definierten Farbe „`glClearColor (0.33f, 0.225f, 0.0f, 1.0f)`“ zu setzen und den z-Buffer auf den in `glClearDepth(1.0)` definierten Wert zu setzen. Auch wenn man ein Objekt vor ein anderes Objekt in einem frame zeichnen will, bspw. Tisch vor Boden, muss nach dem zeichnen von Boden `glClear(GL_DEPTH_BUFFER_BIT)` aufgerufen werden. So wird der Tiefenbuffer auf 1.0 zurückgesetzt und der Tisch wird gezeichnet. Ansonsten würde er verschwinden, wenn er von den z-Koordinaten hinter dem Boden liegt.

`glViewport (0, 0, width, height);`

Frage 2: Setzen Sie versuchsweise statt `width` `width/2` und statt `height` `height/2` ein. Was geschieht? Welche Funktion haben die beiden ersten Parameter, die im Aufruf die Werte 0 besitzen? (Sie finden das z.B. durch Ausprobieren heraus.)

Die ersten beiden Parameter definieren die linke untere Ecke des Viewports. Der Viewport entspricht dem Pixel Rechteck im Fenster auf das das Frustum gemappt wird. Wenn „`width`“ und „`height`“ die Hälfte annehmen, der `x` und `y` Parameter aber trotzdem bei 0 bleibt, wird das komplette Bild halb so groß und in der unteren linken Ecke ausgerichtet, da die beiden Parameter `width` und `height` die Höhe und Breite des Viewports bestimmen und dieser bei (0, 0) beginnt. Standardwerte von `width` und `height` sind somit üblicherweise die Höhe und Breite des Fensters.

Frage 3: Haben Sie zu Beginn dieses Abschnitts 4 den Würfel mit den von Ihnen erwarteten Würfel-Seiten angezeigt bekommen? Da Ihre Grafikkarte zu schnell ist, um den Bildaufbau zu beobachten, sollen Sie die Anzeige „künstlich“ sehr stark verlangsamen

Nein, es wurden nicht die erwarteten Würfel-Seiten angezeigt. Da kein Tiefenpuffer aktiviert ist, werden die Flächen in der Reihenfolge angezeigt, in der sie gezeichnet werden. Somit wird nicht die Seite angezeigt, welche am nächsten zum Betrachter, also der Kamera liegt.

```
glEnable(GL_DEPTH_TEST);  
glClearDepth(1.0);
```

Frage 4: Was hat sich am Ergebnis verändert. Erläutern Sie die Veränderung.

Durch den Tiefenpuffer werden nun die richtigen Seiten angezeigt, nämlich diese, welche am nächsten zum Betrachter bzw. zur Kamera liegen.

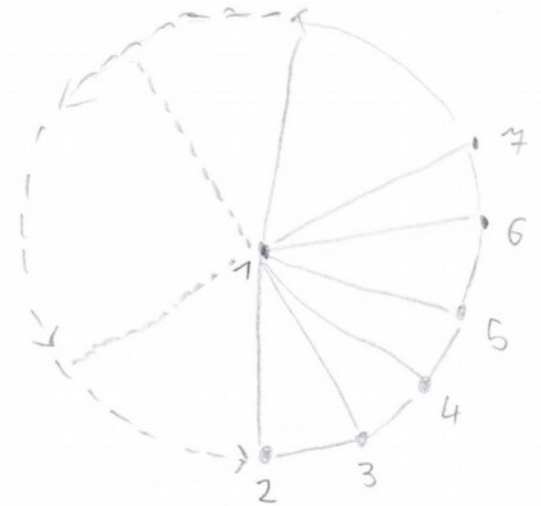
Frage 5: Setzen Sie versuchsweise in `glClearDepth` den Parameter auf den Wert 0.0. Weshalb verschwindet der Würfel? Können Sie den Funktionsaufruf auch weglassen? Was schließen Sie aus Ihrer Antwort?

Der Würfel verschwindet da jeder Wert im Tiefenpuffer auf 0 gesetzt wird und es somit keine anderen z-Werte mehr geben kann, welche kleiner bzw. näher am Betrachter sind. Der Funktionsaufruf kann auch weggelassen werden, da der Tiefenpuffer standardmäßig mit 1.0 gesetzt wird.

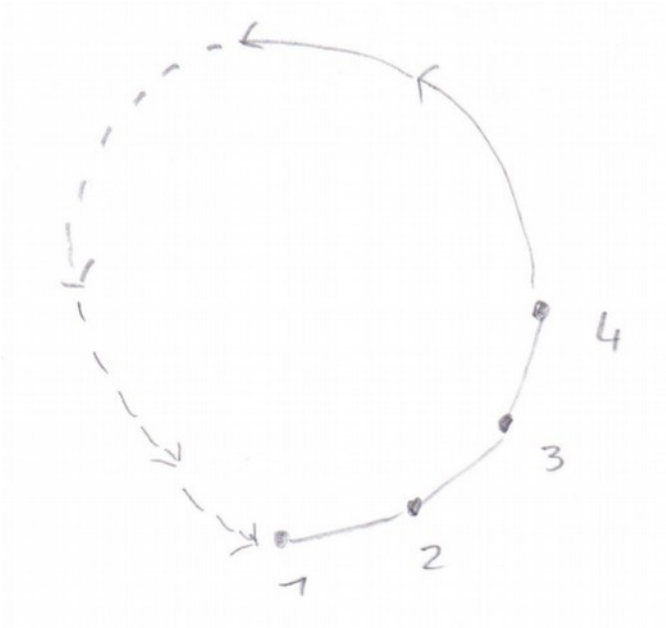
Frage 6: In `Tischplatte.cpp` werden 3 unterschiedliche graphische Primitive verwendet. Welche? Deuten Sie in einer Skizze an, in welcher Reihenfolge dabei die vertices miteinander verbunden werden.

Primitive:

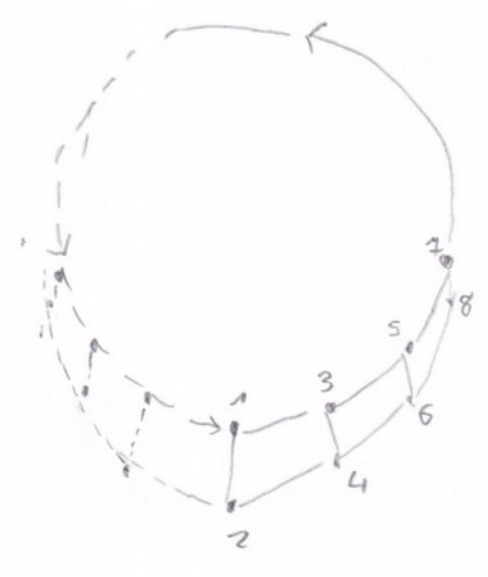
1. Triangle-Fan



2. Polygon

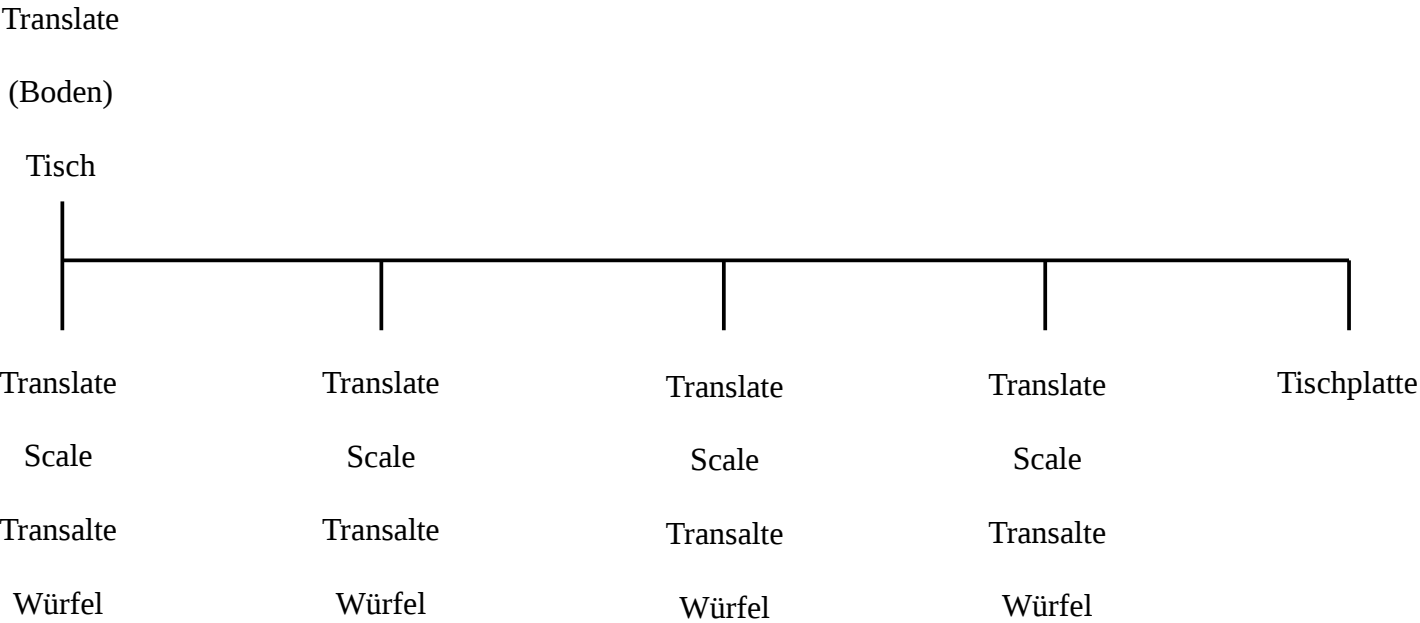


3. Quad-Strip



Frage 7: Fertigen Sie einen Szenengraph für den Tisch an, der alle Objekte und Transformationen unter Benennung ihrer Namen enthält. Achten Sie auf die richtige Reihenfolge. Zeichnen Sie den Tisch im Querschnitt (2D-Seitenansicht) und fügen Sie sein körpereigenes Koordinatensystem hinzu.

Szenengraph:



2D-Schnitt Tisch

