|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) | attribute | uniform | const | varying |
| Kann im VS gelesen werden | x | x |  |  |
| Kann im VS geschrieben werden |  |  |  | x |
| Kann im Hauptprogramm geschrieben werden |  |  | x |  |
| Kann im FS gelesen werden |  | x |  | x |
| Kann im FS geschrieben werden |  |  |  |  |
| Kann im VS innerhalb einer Fkt definiert werden |  |  |  |  |
| Kann im Fragment Shader innerhalb einer Funktion deﬁniert werden |  |  |  |  |
| Hat im VS grundsätzlich für jeden Vertex den gleichen Wert |  | x |  |  |
| Hat im FS grundsätzlich für jedes Fragment den gleichen Wert |  | x |  |  |
|  |  |  |  |  |

b)

Die ﬁnale Position eines Vertex muss im Vertex Shader in die eingebaute gl\_Position Variable.geschrieben werden.

Die ﬁnale Farbe eines Fragments muss im Fragment Shader in die eingebaute Variable gl\_FragColor geschrieben werden.

c)

|  |  |
| --- | --- |
|  | ModelView-Transformation |
| x | Rasterisierung |
| x | Projektion |
|  | z-Buﬀer-Test |

d)

e)

vec3 v2 = v.xyzw; //vec3 kann nur 3 Variablen übernehmen

vec3 v4 = v.gb; // nur 2 von 3 benötigten Werten gegeben

vec3 v3 = v.def; // Ungültiger index Zugriff

vec3 d = dot(w, v); // Dimensionen von w & v stimmen nicht überein

vec3 e = dot(w.xy, v.yz); // dot(…) ergibt 2 Dim. Vektor, vec3 gefragt

mat2 m = mat2(1.0, 2.0, 1.0, 0.0, 0.1); // ungerade Anzahl an Parametern für nxn Matrix