Aufgabe 1.a

Es entsteht ein Hell-Dunkel-Verlauf, wobei der Knoten mit der zur Lichtquelle gerichteten Normale sehr hell ist und die Helligkeit in Richtung der beiden anderen Knoten abnimmt.

Aufgabe 1.b

Bei einer statischen Szene hat die Position des Betrachters nur auf die spekuläre Beleuchtung Einfluss. Bei der ambienten Beleuchtung sind die Normalen irrelevant, bei der diffusen Beleuchtung ist nur der Winkel zwischen den Flächennormalen und den Lichtstrahlen relevant. Bei der spekulären Beleuchtung wird zusätzlich aber auch der Sehstrahl des Betrachters berücksichtigt, da es das einfallende Licht auf die Oberfläche zum Betrachter hin reflektiert.

Aufgabe 2

Vorteile:

* Reduziert die Anzahl der zur Darstellung benötigten Knotenpunkte des Polygonmodells
* Komplexität der Geometrie bleibt klein
* Sehr schnelles Rendering

Nachteile:

* Detailgenauigkeit des Objekts nur Illusion
* Bei flachen Blickwinkel kann man die dargestellte Tiefe als Täuschung erkennen
* Beim Schattenwurf kann man ebenfalls die fehlenden Details des Polygonmodells erkennen

Bei Computerspielen wird es verwendet, weil sehr viele Objekte sehr schnell gerendert werden müssen. Man kann sehr viele Objekte mit scheinbar vielen Details bei hohen Bildraten in Echtzeit rendern, was die Verwendung einer detailreichen Spielwelt ermöglicht.

Aufgabe 3

Eine uniform-Variable ist eine globale Variable, welche vom Shaderprogramm nur gelesen werden kann und über alle Bildberechnungen hinweg erhalten bleibt. Das Hauptprogramm kann sie trotzdem manipulieren. Sie eignet sich für Speichern der Modelviewmatrix und der Projektionsmatrix.

Eine attribute-Variable ist nur im Vertexshaderprogramm verwendbar. Sie können nur gelesen werden. Sie gilt nur für einen Knoten und dient als Eingabeparameter für den Vertexshader.

Eine varying-Variable dient dazu, Informationen vom Vertexshader an den Fragmentshader zu übergeben. Sie kann vom Fragmentshader nur gelesen, vom Vertexshader darüber hinaus aber auch manipuliert werden.

Aufgabe 4

Geometrieshader dienen dazu, neue Primitive aus bereits vorhandenen Primitiven zu erzeugen und in die Grafikpipeline zu integrieren. Sie werden beispielsweise zur Erzeugung von Schattenvolumen eingesetzt.

Tesselationsshader dienen speziell dazu, die Polygone der Modelle weiter in Dreiecks- und Vierecksnetze zu unterteilen, da diese hardwareseitig sehr effizient verarbeitet werden können. Er wird eingesetzt um den Detailgrad von Modellen effizient weiter zu erhöhen.

Vertexshader, Tesselationsshader, Geometrieshader und Fragmentshader werden in genau dieser Reihenfolge ausgeführt.

Aufgabe 5

Werden die Dreiecksebenen zur Generierung der Partitionierungsebenen verwendet, dann hat der resultierende BSP die Eigenschaft, dass für jeden Betrachtungspunkt alle Polygone eindeutig sortierbar und mit dem Painter’s-Algorithmus darstellbar sind.

Alternativ kann man den BSP aufbauen, indem man die gesamte Szene in einen Hüllkörper einsetzt und diesen weiter aufteilt. Dabei werden die Partitionierungsebenen stets parallel zu den Koordinatenachsen gesetzt.