Westfälische
Hochschule
Gelsenkirchen Bocholt Recklinghausen

Computergrafik

Die Renderpipeline

Vertex- und Fragment-Shader

Prof. Dr. Tom Vierjahn

Visual Computing (https://vc.w-hs.de)
Fachbereich Wirtschaft und Informationstechnik
Campus Bocholt

Sommersemester 2020

Vertex-Shader

für den Anfang

Westfälische Hochschule

VBO-Inhalt:

Shader-Code:

```
#version 330 core

layout(location = 0) in vec3 vertex_position;

void main() {
  gl_Position.xyz = vertex_position;
  gl_Position.w = 1.0;
}
```

- Shader wird für jeden Vertex aufgerufen.
- `gl_Position` wird von OpenGL für jeden Vertex bereitgestellt.

Fragment-Shader

für den Anfang

Westfälische Hochschule

Shader-Code:

```
#version 330 core

layout(location = 0) out vec3 color;

void main() {
  color = vec3(0.443, 0.694, 0.153);
}
```

- Rasterisierung erzeugt Fragmente.
- Shader wird für jedes Fragment aufgerufen.
- Ausgabe geht ans Blending.

Einen Shader anlegen

Westfälische Hochschule

Shader erzeugen:

```
GLuint glCreateShader(GLenum shaderType);
```

- shaderType für Vertex-Shader:
- shaderType für Fragment-Shader:

Shader-Source zuweisen und kompilieren:

```
void glShaderSource(GLuint shader, GLsizei count, const GLchar** string,
                    const GLint* length);
void glCompileShader(GLuint shader);
```

Ein Shader-Programm anlegen und verwenden

Westfälische Hochschule

Shader-Programm erzeugen:

```
GLuint glCreateProgram();
```

Shader zuweisen und Programm linken:

```
void glAttachShader(GLuint program, GLuint shader);
void glLinkProgram(GLuint program);
```

Shader-Prgramm für Draw-Call verwenden:

```
void glUseProgram(GLuint program);
```

Falls etwas schiefgeht

Westfälische Hochschule

Fehlerbehandlung für Shader:

```
void glGetShaderiv(GLuint shader, GLenum pname, GLint* params);
void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei* length,
                        GLchar* infoLog);
```

- pname, um zu überprüfen, ob Kompilieren erfolgreich war:
- pname, um die Länge des Logs abzufragen:

Fehlerbehandlung für Shader-Programme:

```
void glGetProgramiv(GLuint program, GLenum pname, GLint* params);
void glGetProgramInfoLog(GLuint program, GLsizei maxLength,
                         GLsizei* length, GLchar* infoLog);
```

- pname, um zu überprüfen, ob Linken erfolgreich war:
- pname, um die Länge des Logs abzufragen:

Uniforms

Eingaben pro Geometrie

Westfälische Hochschule

Definition: Uniform[1]

A uniform is a global Shader variable declared with the uniform storage qualifier. These act as parameters that the user of a shader program can pass to that program. [...]
Uniforms [...] do not change from one shader invocation to the next within a particular rendering call.

Als Eingabe im (Fragment)-Shader-Code:

```
uniform vec3 geometry_color;
```

Zuweisung im Programmcode:

```
GLint glGetUniformLocation(GLuint program, const GLchar* name);
void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2);
void glUniform3fv(GLint location, GLsizei count, const GLfloat* value);
```

[1] Uniform (GLSL). (2019, April 25). OpenGL Wiki, . Retrieved May 1, 2020 from http://www.khronos.org/opengl/wiki_opengl/index.php?title=Uniform_(GLSL)&oldid=14539.

Zusammenfassung

Westfälische Hochschule

- Vertex-Shader
- Fragment-Shader
- Shader-Programm
- Uniforms

Prof. Dr. Tom Vierjahn
- E-Mail: tom.vierjahn@w-hs.de

Visual Computing
- Web: `https://vc.w-hs.de`
- YouTube: Visual Computing WH
- Twitter: @VisComputingWH

Westfälische Hochschule
Fachbereich Wirtschaft und Informationstechnik
Campus Bocholt