Westfälische
Hochschule
Gelsenkirchen Bocholt Recklinghausen

Computergrafik

Die Renderpipeline

Geometrie

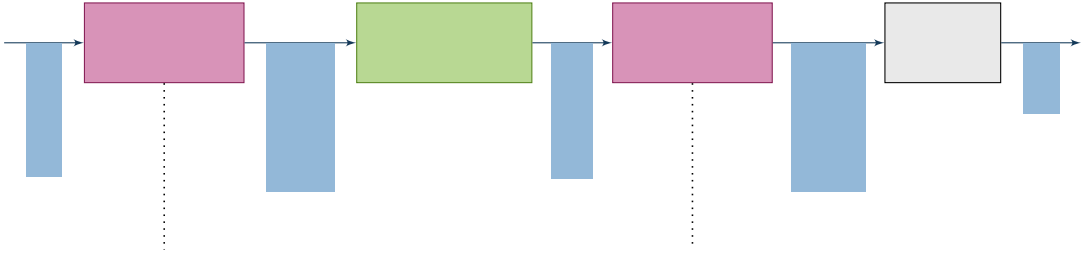
Prof. Dr. Tom Vierjahn

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

Sommersemester 2020

Die Grafik-Pipeline

stark vereinfacht



```
in vec3 position;
void main() {
  // ...
  gl_Position = /*...*/;
}
```

```
out vec3 color;
void main() {
  // ...
  color = /*...*/;
}
```

Geometrie in C++ und OpenGL

Westfälische Hochschule

Geometrie-Objekte:
- C++/CPU:
- OpenGL/GPU:

Platz für Vertices:
- C++/CPU:
- OpenGL/GPU:

erzeuge Dreiecke:
- C++/CPU:
- OpenGL/GPU:

rendere Geometrie:
- C++/CPU:
- OpenGL/GPU:

Vertex Array Object, Vertex Buffer Object

Westfälische Hochschule

Definition: Vertex Array Object[1]

A Vertex Array Object (VAO) is an OpenGL Object that stores all of the state needed to supply vertex data [...]. It stores the format of the vertex data as well as the Buffer Objects providing the vertex data arrays [but not their contents].

Definition: Vertex Buffer Object[1]

A Vertex Buffer Object (VBO) is the common term for a normal Buffer Object when it is used as a source for vertex array data. It is no different from any other buffer object, [...]

[1] Vertex Specification. (2019, December 30). OpenGL Wiki, . Retrieved May 1, 2020 from `http://www.khronos.org/opengl/wiki_opengl/index.php?title=Vertex_Specification&oldid=14620`.

Vertices auf die Grafikkarte bringen

VAO, VBO erstellen, füllen, …

Westfälische Hochschule

VAO, VBO erstellen:

```
void glGenVertexArrays(GLsizei n, GLuint* arrays);
void glGenBuffers(GLsizei n, GLuint* buffers);
```

VAO, VBO binden / aktivieren:

```
void glBindVertexArray(GLuint array);
void glBindBuffer(GLenum target, GLuint buffer);
```

- `target` für Vertex-Attribute:

Vertices auf die Grafikkarte bringen

VAO, VBO erstellen, füllen, …

Westfälische Hochschule

VBO füllen:

```
void glBufferData(GLenum target, GLsizeiptr size, const GLvoid* data,
                  GLenum usage);
```

- Verwendung (usage) „einmal setzen, mehrfach zeichnen“:

Vertex Daten einem Array-Index zuordnen:

```
void glEnableVertexAttribArray(GLuint index);
void glVertexAttribPointer(GLuint index, GLint size, GLenum type,
                           GLboolean normalized, GLsizei stride,
                           const GLvoid* pointer);
```

- type für Vertices mit `float`-Komponenten:

Geometrie rendern

Westfälische Hochschule

Rendern:

```
void glDrawArrays(GLenum mode, GLint first, GLsizei count);
```

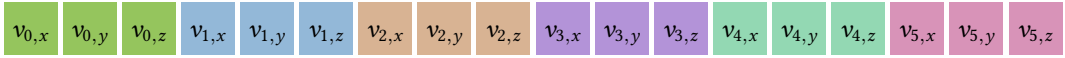
Geometrie	mode
Punkte	
Linien	
Dreiecke	

Wichtig: Die Geometrierepräsentation der Vertex-Liste wird erst hier festgelegt.

Geometrierepräsentation

Westfälische Hochschule

VBO-Inhalt



Daten-Upload

```
glBufferData(GL_ARRAY_BUFFER, ??, ..., GL_STATIC_DRAW);
```

Daten-Zuordnung

```
glVertexAttribPointer(0, ?, GL_FLOAT, GL_FALSE, 0, nullptr);
```

Geometrierepräsentation

Westfälische Hochschule

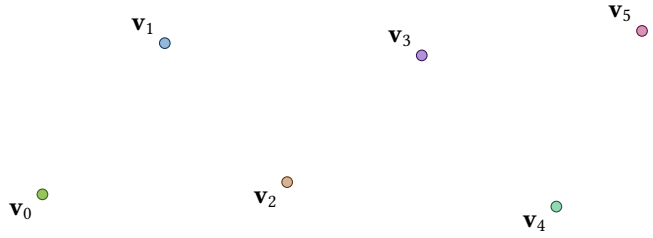
VBO-Inhalt

$\mathbf{v}_0$	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$	$\mathbf{v}_5$

Rendering-Code

```
glDrawArrays(GL_POINTS, 0, 6);
```

Ergebnis



Geometrierepräsentation

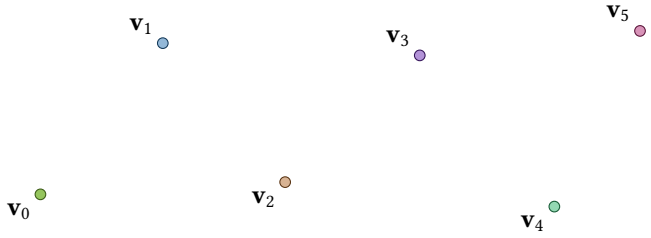
Westfälische Hochschule

VBO-Inhalt

Rendering-Code

```
glDrawArrays(GL_LINES, 0, 6);
```

Ergebnis



Geometrierepräsentation

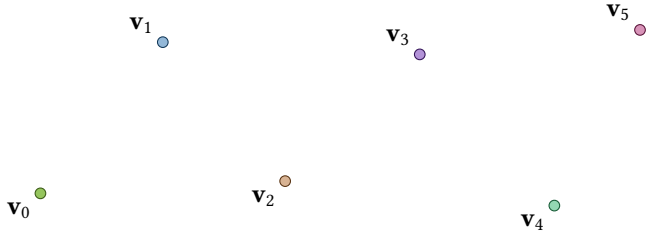
Westfälische Hochschule

VBO-Inhalt

Rendering-Code

```
glDrawArrays(GL_LINE_STRIP, 0, 6);
```

Ergebnis



Geometrierepräsentation

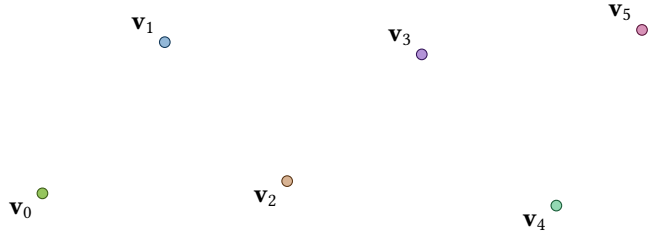
Westfälische Hochschule

VBO-Inhalt

Rendering-Code

```
glDrawArrays(GL_TRIANGLES, 0, 6);
```

Ergebnis



Geometrierepräsentation

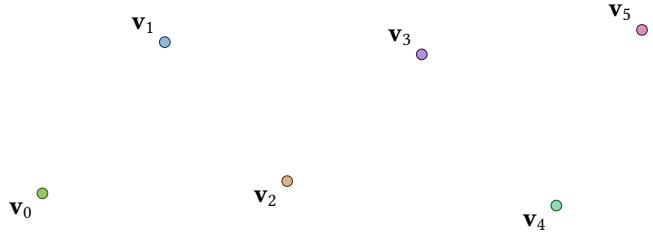
Westfälische Hochschule

VBO-Inhalt

Rendering-Code

```
glDrawArrays(GL_TRIANGLE_STRIP, 0, 6);
```

Ergebnis



Geometrierepräsentation

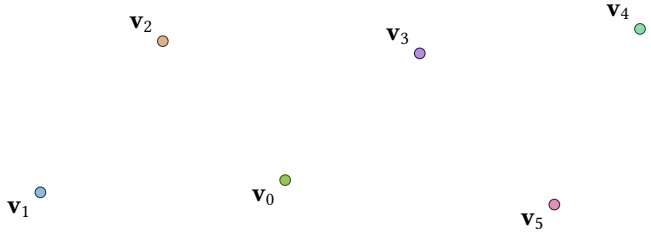
Westfälische Hochschule

VBO-Inhalt

Rendering-Code

```
glDrawArrays(GL_TRIANGLE_FAN, 0, 6);
```

Ergebnis



Zusammenfassung

Westfälische Hochschule

- Pipeline
- Geometrie in C++ und OpenGL
- Vertex Array Object und Vertex Buffer Object
- Upload von Vertices
- Geometrierepräsentation

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