

Interaktive Computergrafik



Prof. Dr. Frank Steinicke
Human-Computer Interaction
Fachbereich Informatik
Universität Hamburg



Interaktive Computergrafik

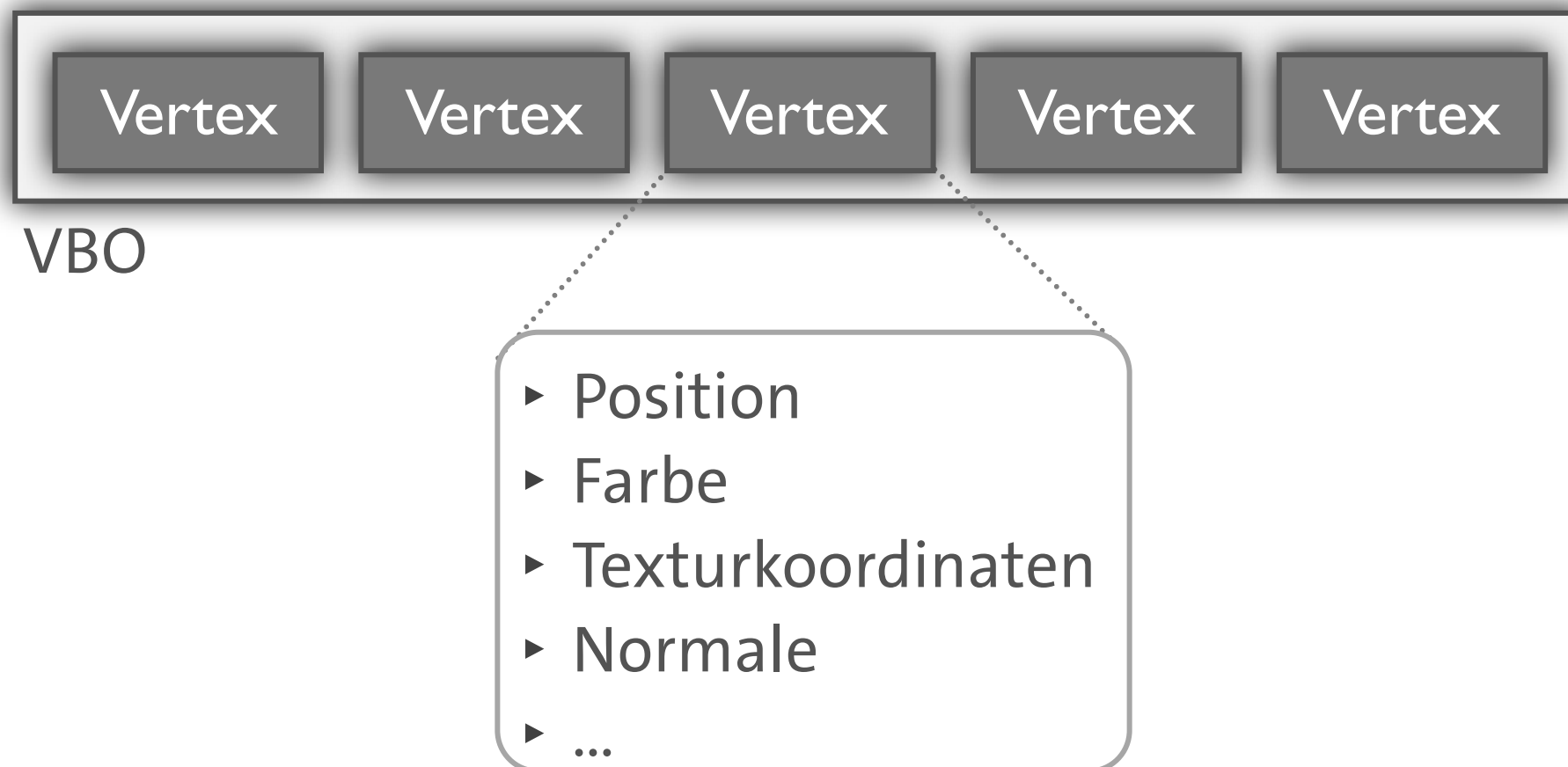
Übung - Woche 2

Human-Computer Interaction, University of Hamburg

Vertex Buffer Objects

Wiederholung: Definition

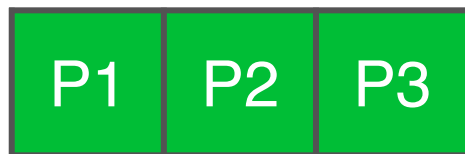
- Eckpunkt eines Primitives = **Vertex**
- Sammlung von Vertices = **VBO**



Vertex Buffer Objects

Wiederholung: Layouts

- ein VBO pro Attribut



- ein VBO für alle Attribute (nacheinander)



- ein VBO für alle Attribute (abwechselnd)



Gruppenarbeit



Welches VBO-Layout ist im aktuellen Code umgesetzt?

```
// 3. Specify geometry
const positions = new Float32Array([
    -1, -1,
    -1, 0,
    1, 0,
    1, -1,
]);

const colors = new Float32Array([
    1, 0, 0, 1,
    1, 1, 0, 1,
    0, 0, 1, 1,
    1, 0, 0, 1,
]);
```

RAM der GPU

vPosition

-1	-1	-1	0	1	0	1	-1
----	----	----	---	---	---	---	----

vColor

1	0	0	1	1	1
---	---	---	---	---	---

```
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
...
gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
```

Gruppenarbeit



Versuchen Sie das VBO-Layout so zu ändern, dass Position und Farbe nacheinander in einem einzelnen VBO gespeichert werden.

```
const vertices = new Float32Array([
    -1, -1,
    -1, 0,
    1, 0,
    1, -1,
    1, 0, 0, 1,
    1, 1, 0, 1,
    0, 0, 1, 1,
    1, 0, 0, 1,
]);
```

RAM der GPU

vPosition

-1	-1	-1	0	1	0	1	-1
----	----	----	---	---	---	---	----

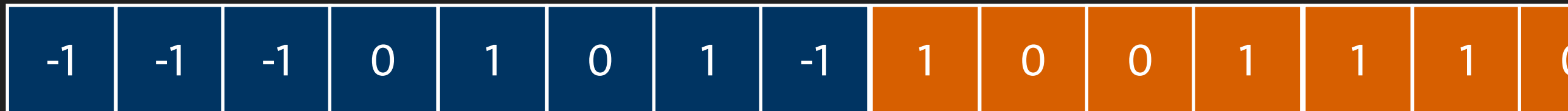
vColor

1	0	0	1	1	1
---	---	---	---	---	---


```
const vertices = new Float32Array([
    -1, -1,
    -1, 0,
    1, 0,
    1, -1,
    1, 0, 0, 1,
    1, 1, 0, 1,
    0, 0, 1, 1,
    1, 0, 0, 1,
]);
```

RAM der GPU

vertex



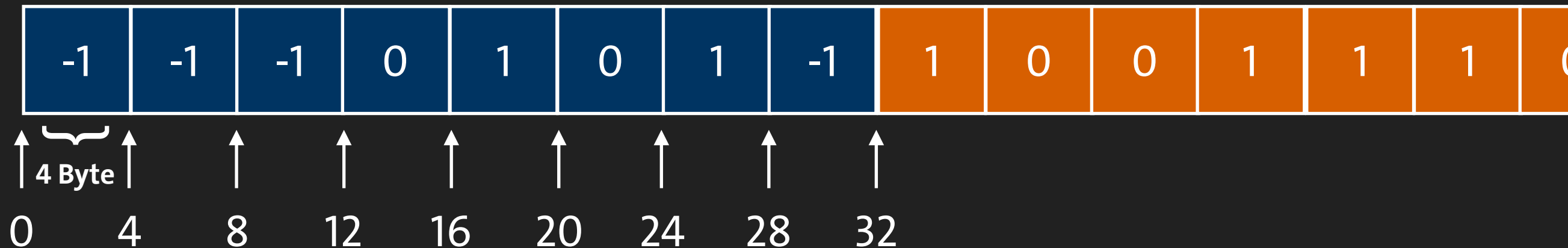
Ab hier: vPosition

Ab da: vColor

```
const vertices = new Float32Array([
    -1, -1,
    -1, 0,
    1, 0,
    1, -1,
    1, 0, 0, 1,
    1, 1, 0, 1,
    0, 0, 1, 1,
    1, 0, 0, 1,
]);
```

RAM der GPU

vertex



```
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 32);
```

Gruppenarbeit



Ändern Sie die vorherige Lösung so, dass Position und Farbe abwechselnd gespeichert werden.

```
const vertices = new Float32Array([
    -1, -1,
    1, 0, 0, 1,
    -1, 0,
    1, 1, 0, 1,
    1, 0,
    0, 0, 1, 1,
    1, -1,
    1, 0, 0, 1,
]);
```

RAM der GPU

vertex

-1	-1	1	0	0	1	-1	0	1	1	0	1	1	0	0
----	----	---	---	---	---	----	---	---	---	---	---	---	---	---

```
gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 24, 0);
gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 24, 8);
```

Gruppenarbeit



(Gehen Sie zurück zum Ausgangslayout.)
Erstellen Sie ein Quadrat mit variabler
Seitenlänge.

Gruppenarbeit



Erstellen Sie in der unteren linken Ecke des Canvas ein Quadrat mit variabler Seitenlänge.

Gruppenarbeit



Füllen Sie mithilfe von for-Schleifen den gesamten Canvas mit 32 x 32 Quadraten.

Gruppenarbeit



Zeichnen Sie ein Quadrat nur, wenn das entsprechende Element im Array
bitcode auf 1 gesetzt ist.

