

# Interaktive Computergrafik

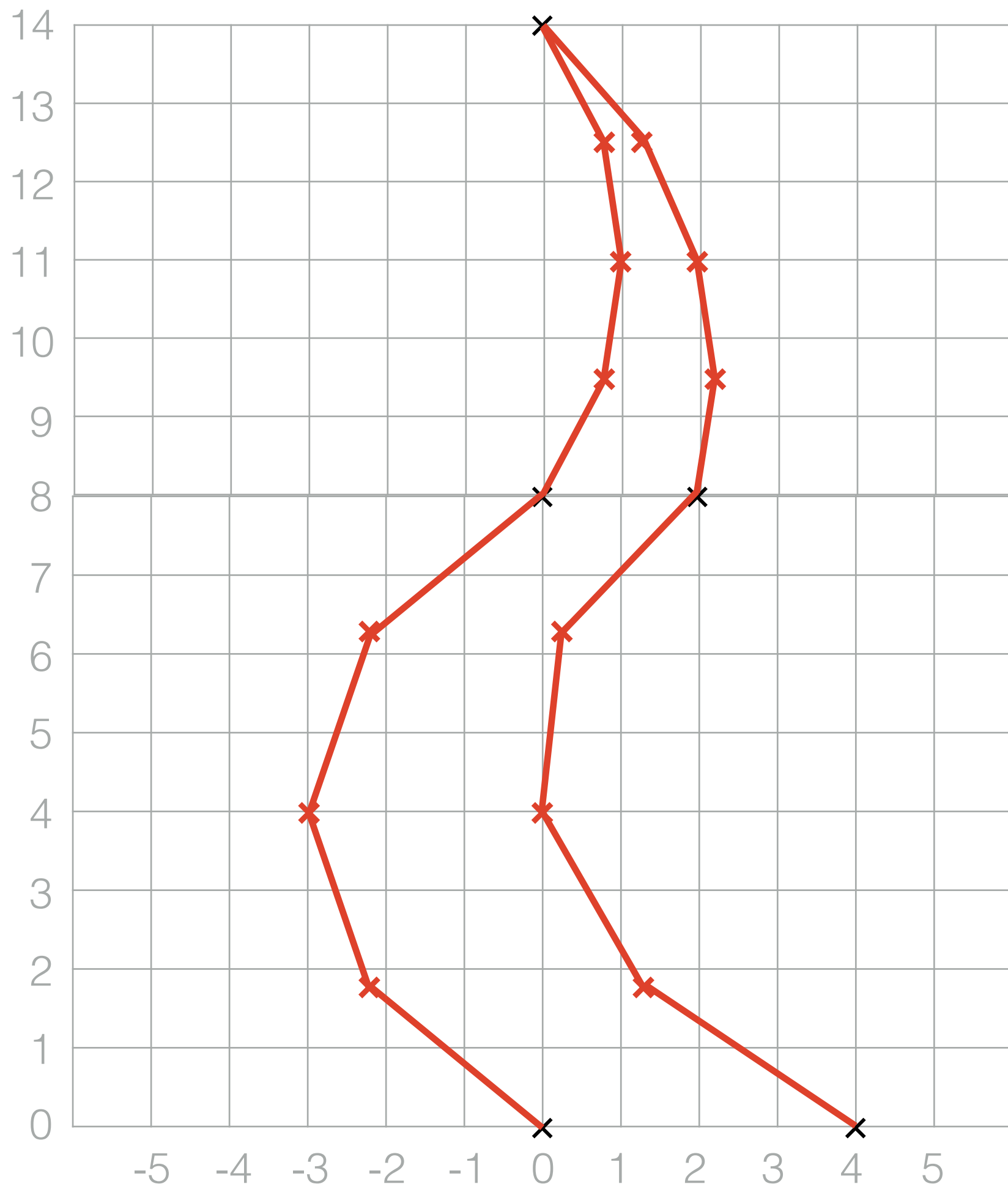


**Prof. Dr. Frank Steinicke**  
Human-Computer Interaction  
Department of Computer Science  
University of Hamburg



# Interaktive Computergrafik

## Übung - Woche 4





# Interaktive Computergrafik

Übung - Woche 4

Objektorientiertes WebGL

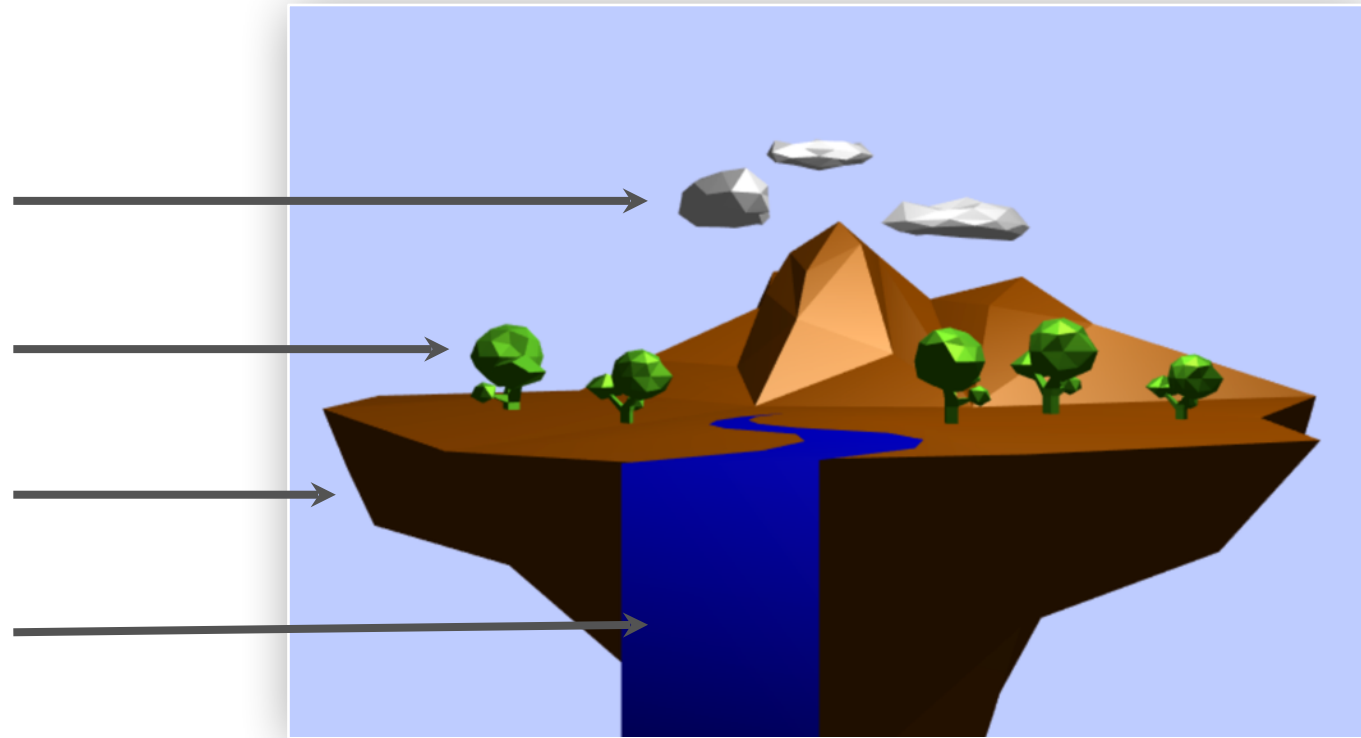
# Problemstellung

Objekt 1: Wolke

Objekt 2: Baum

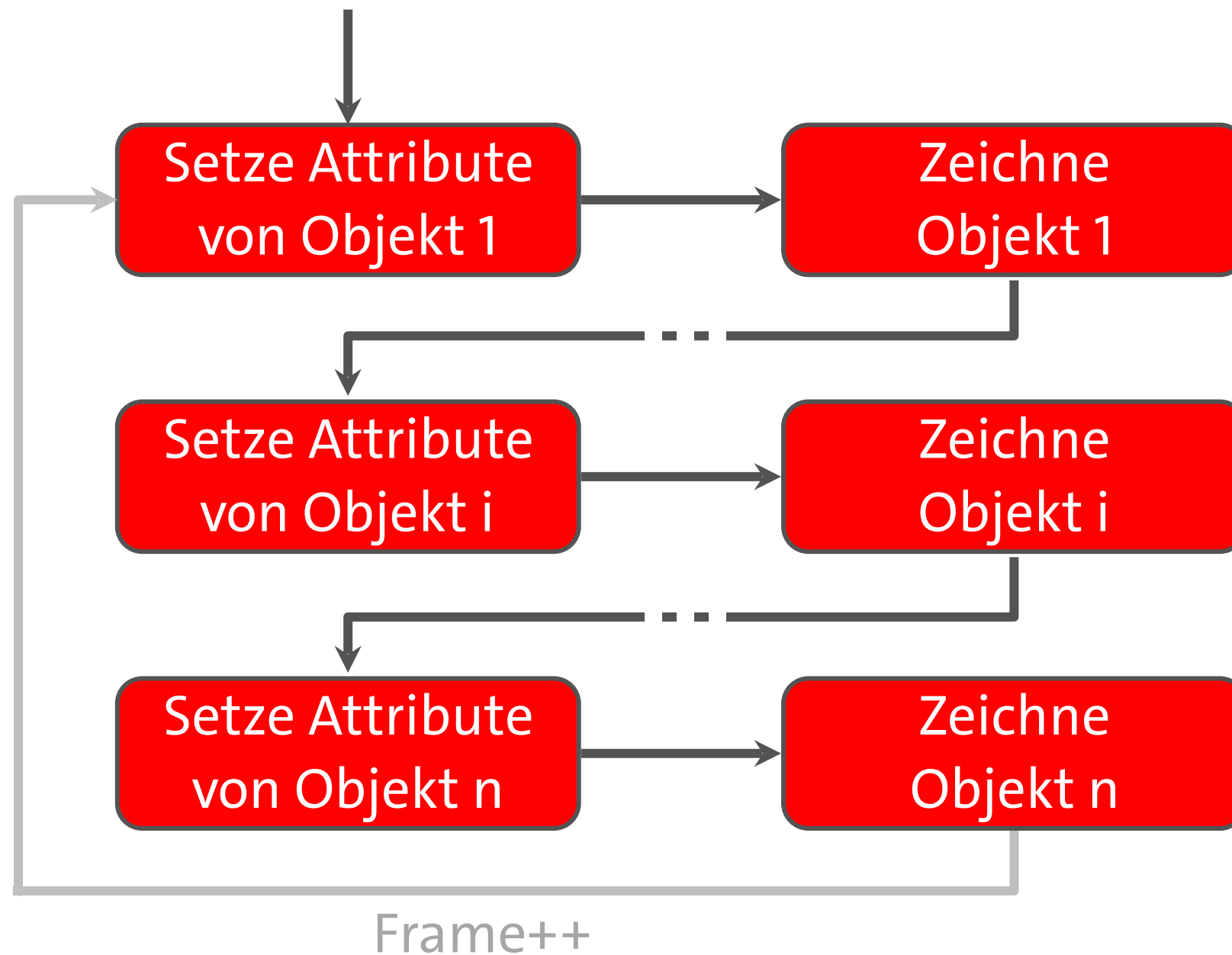
Objekt 3: Insel

Objekt 4: Fluss



Wie können verschiedenen Objekten unterschiedliche Eigenschaften (z.B. Positionen und Farben) zugewiesen werden?

# Grundidee



# Implementierung

- Aufteilung der Vertices in mehrere VBOs sinnvoll (mind. 1 VBO pro Objekt)
- Einmalige Initialisierung:
  1. Generiere ein  $VBO_i$  für Objekt  $i$
  2. Binde  $VBO_i$
  3. Fülle  $VBO_i$

Wiederhole Schritte für alle Objekte

# Implementierung

- Pro Frame:
  1. Binde  $VBO_i$
  2. Definiere Format von  $VBO_i$
  3. Setze objektspezifische Attribute
  4. Rendere Primitiven aus  $VBO_i$

Wiederhole Schritte für alle Objekte



# Implementierung

- Problem: Wiederholung der Schritte für alle Objekte führt zu **Code-Duplikaten**
  - unübersichtlich
  - schwer wartbar
- Lösung: **Objektorientierter** Ansatz → Kapselung aller objektspezifischen Eigenschaften

# Was ist ein Objekt?

- Sammlung von **Eigenschaften** und **Methoden**

```
const square = {  
  _x: 0.4,  
  _y: 0.3,  
  _size: 0.5  
};
```

*Objekteigenschaften  
als Schlüssel-Wert-Paare*



# Klassen in JS ES6

- Zur Repräsentation ähnlicher Objekte

```
const square1 = {  
  _x: 0.4,  
  _y: 0.3,  
  _size: 0.5  
};
```

```
const square2 = {  
  _x: 0.1,  
  _y: 1.0,  
  _size: 0.2  
};
```

# Klassen in JS ES6

- Zur Repräsentation ähnlicher Objekte

```
class Square {  
  constructor(x, y, size) {  
    this._x = x;  
    this._y = y;  
    this._size = size;  
  }  
}
```

# Konstruktor

- **constructor** ist spezielle Methode, die automatisch bei Instanziierung der Klasse aufgerufen wird und der Initialisierung dient

# Instanzen

- **Instanz einer Klasse** ist ein Objekt, das Eigenschaften und Methoden einer Klasse enthält und ihnen spezifische Werte zuweist

```
const square1 = new Square(0.4, 0.3, 0.5);  
const square2 = new Square(0.1, 1.0, 0.2);
```

- Klasse muss vor Zugriff definiert werden!

# Methoden

```
class Square {  
    ...  
    translate(deltax, deltay) {  
        this._x += deltax;  
        this._y += deltay;  
    }  
}  
  
const s = new Square(0.4, 0.3, 0.5);  
s.translate(0.1, 0.1);
```

