

# Computergrafik

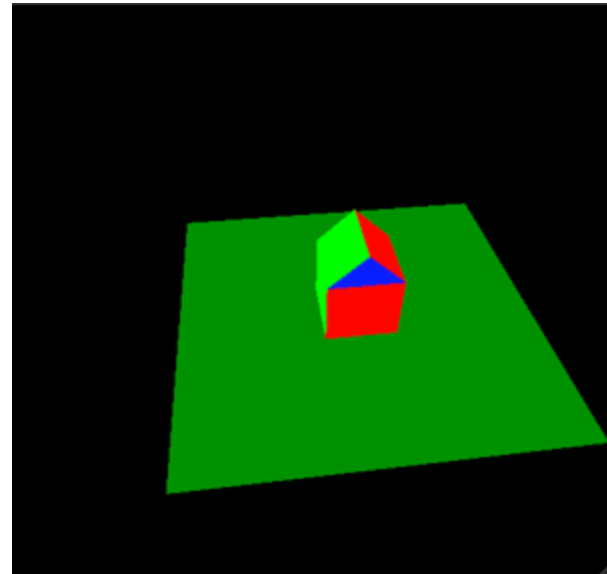
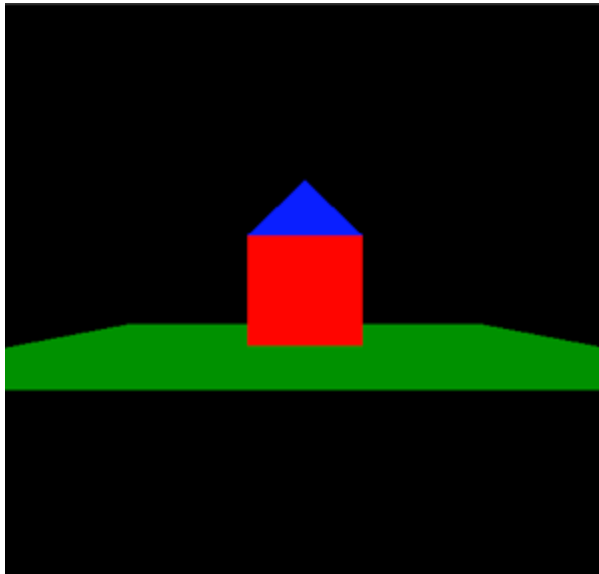
Universität Bern  
Herbst 2010

# Übung 2: Benutzerinteraktion

- Abgabe am 14. Oktober
- Code bis Mittag 12 Uhr im Ilias hochladen!
- Für Abgabe an Assistenten online neu einschreiben
- Für jede Teilaufgabe separate Java Klasse machen
  - Einfachere Präsentation
  - Projekt „simple“ kopieren und erweitern

# 1. Kamera & View Frustum

- Klassen in Projekt jrtr modifizieren
- Kamera- & Projektionsmatrizen gemäss Formeln aus Vorlesung setzen
- Test wie auf Webpage beschrieben

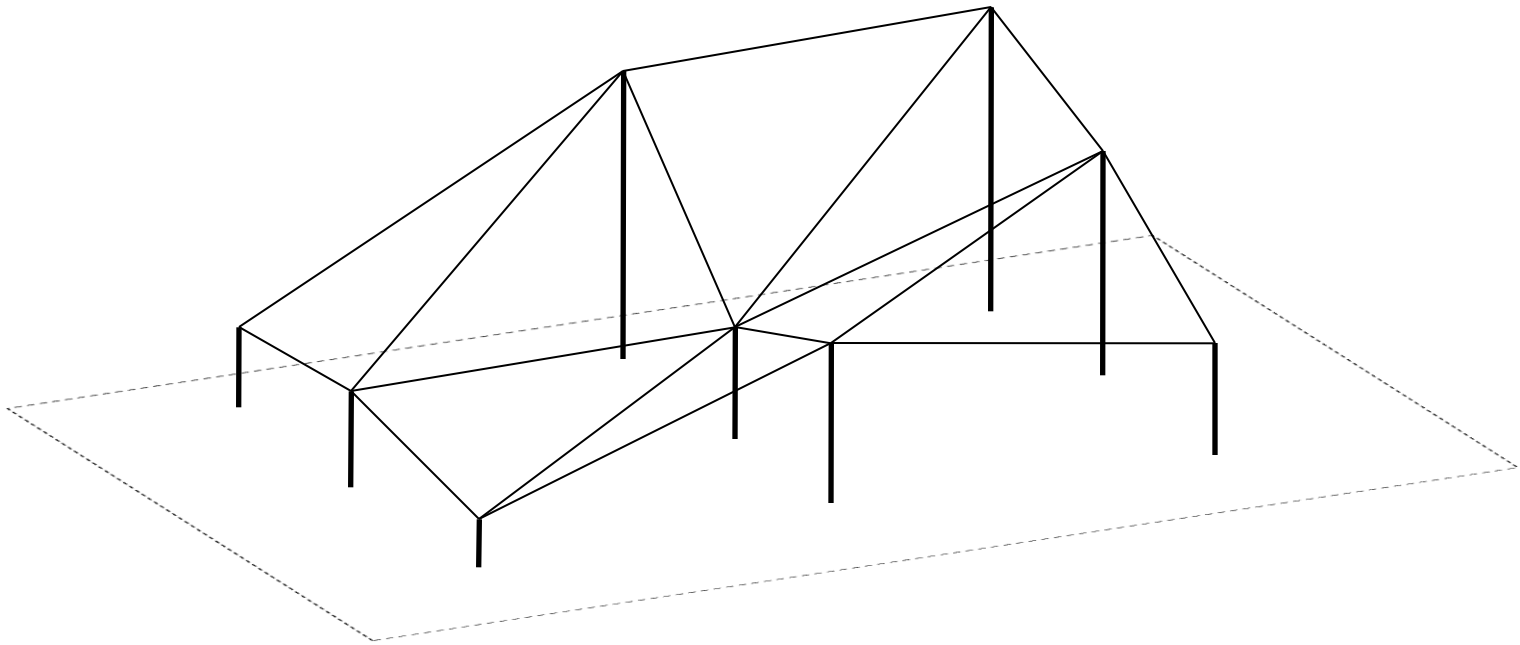


## 2. Virtual Trackball

- Demonstration
- Einbauen von vorgegebenem Code wie auf Webpage beschrieben
  - Klasse zum Lesen von obj Dateiformat
  - Shader für bessere Visualisierung der Oberflächen

# 3. Fraktale Landschaft

- Landschaft = 2D Feld von Höhenwerten
- Zu Dreiecksgitter verbunden

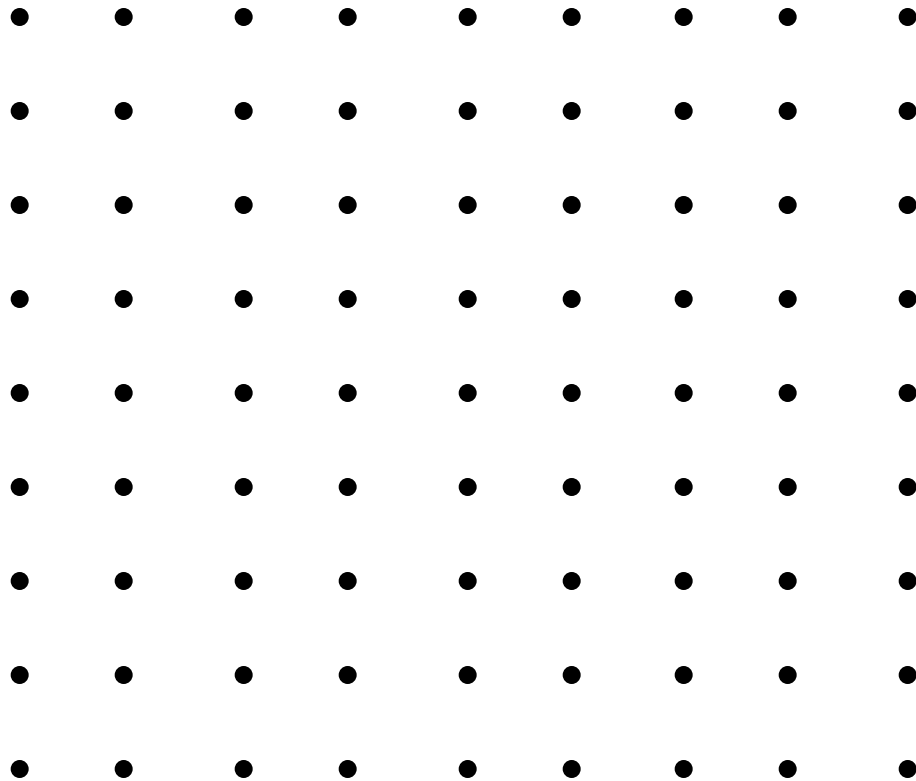


# Höhenwerte

- Zuerst Feld von Werten allozieren
  - Grösse  $(2^{n+1}) \times (2^{n+1})$
- Höhenwerte zufällig generieren, aber „von grob nach fein“
  - Squares & Diamond Algorithmus
- Jeder neue Höhenwert ist Durchschnitt von 4 schon berechneten Werten plus zufällige Abweichung

# Squares & Diamonds

- Beispiel: Feld mit  $(2^{3+1}) \times (2^{3+1})$  Werten

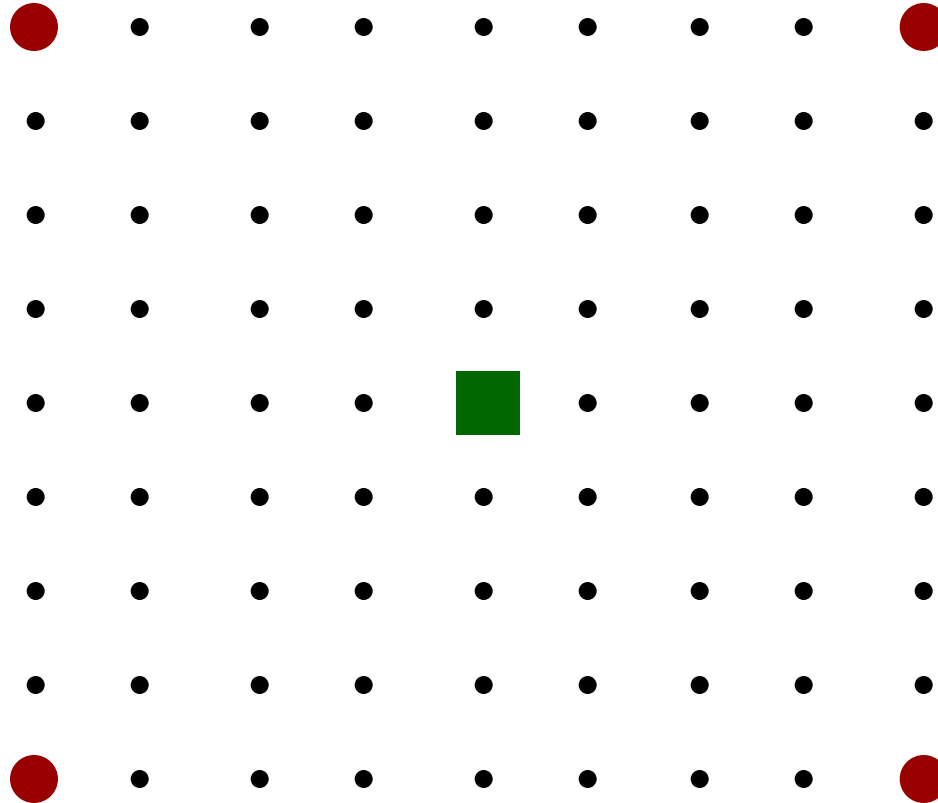






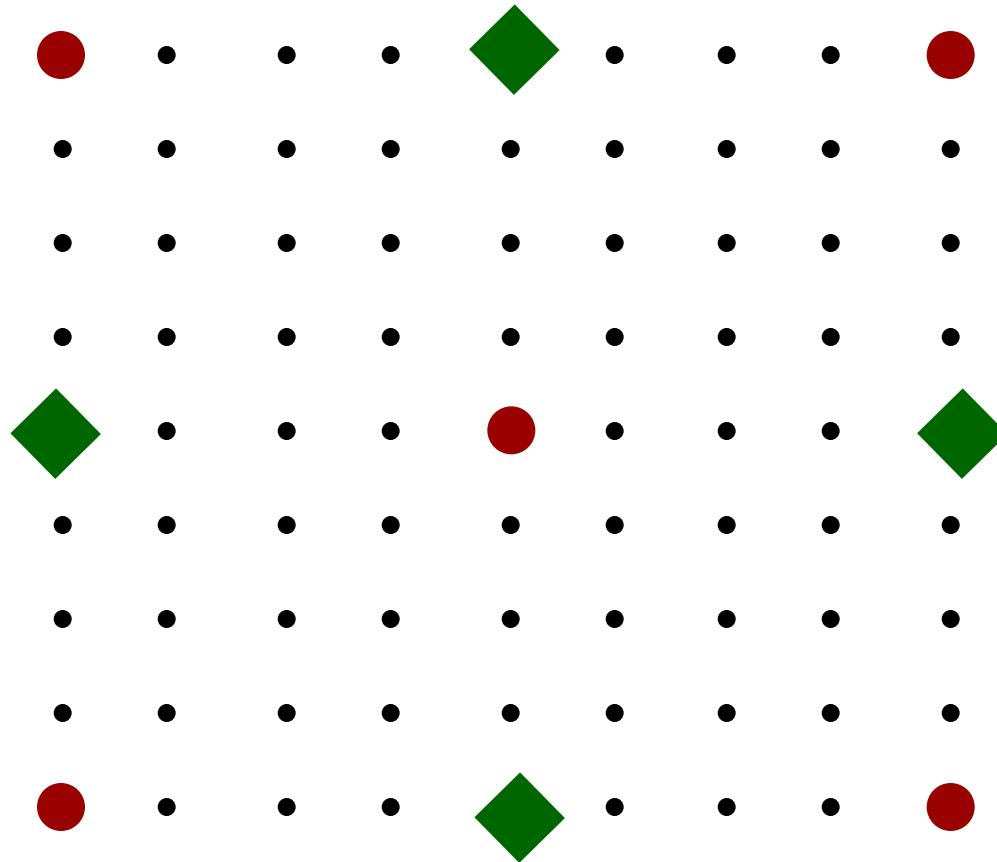
# Squares & Diamonds

- Square Schritt, grün wird neu berechnet



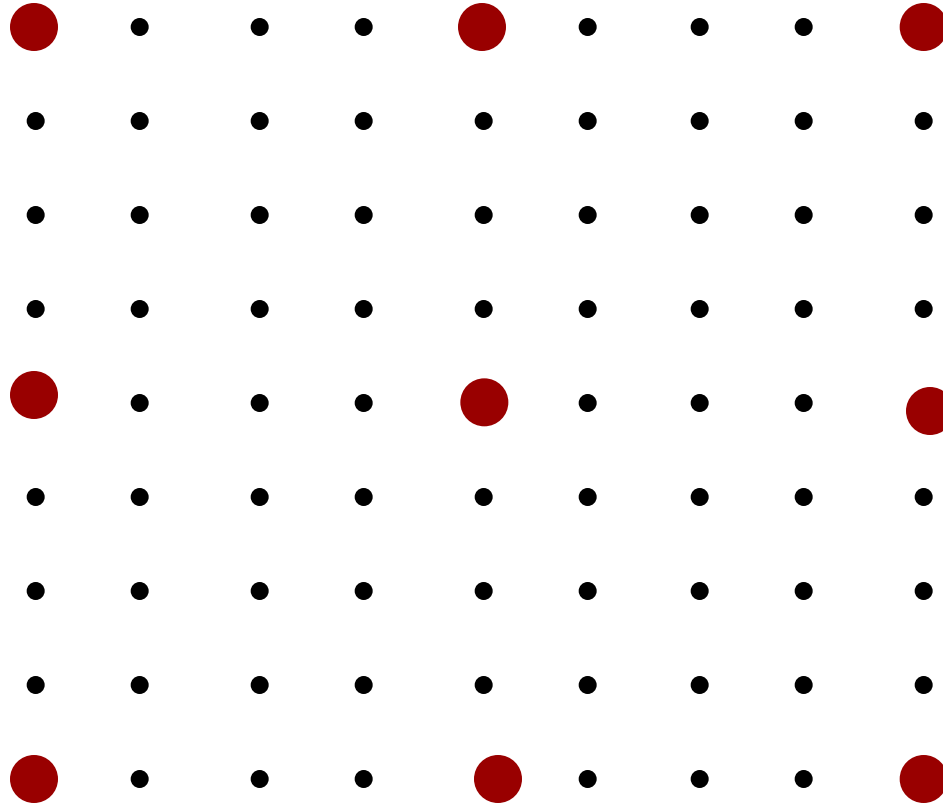
# Squares & Diamonds

- Diamond Schritt, grün wird neu berechnet



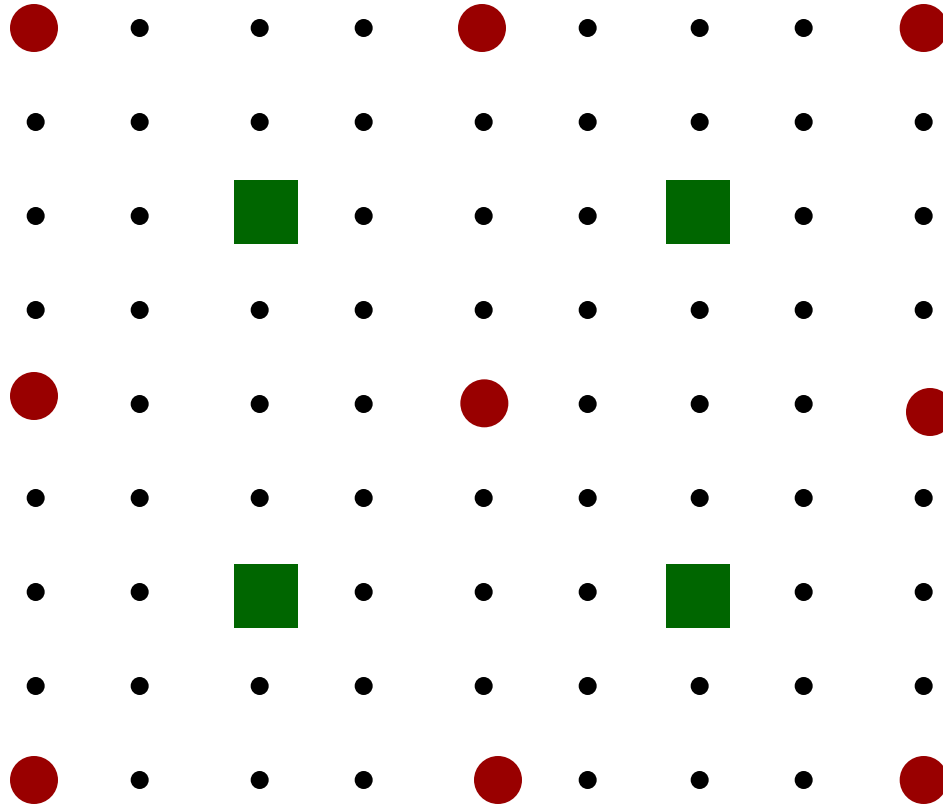
# Squares & Diamonds

- Ausgangslage nächster „grob nach fein“ Schritt



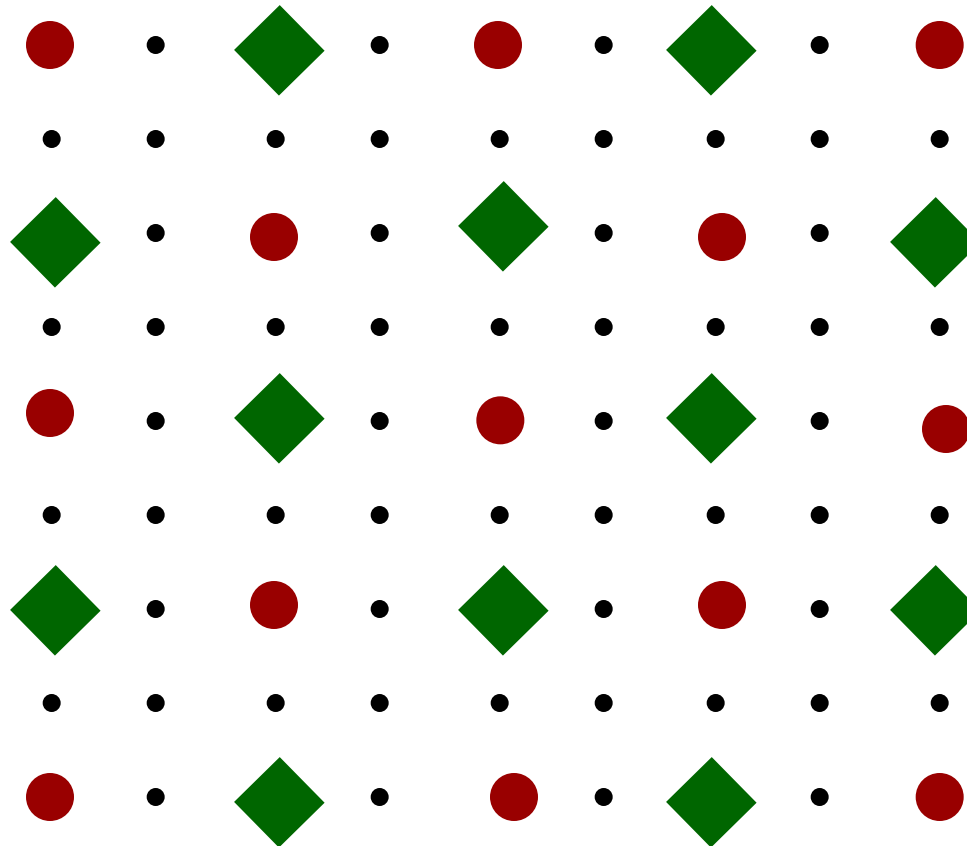
# Squares & Diamonds

- Square Schritt, grün wird neu berechnet



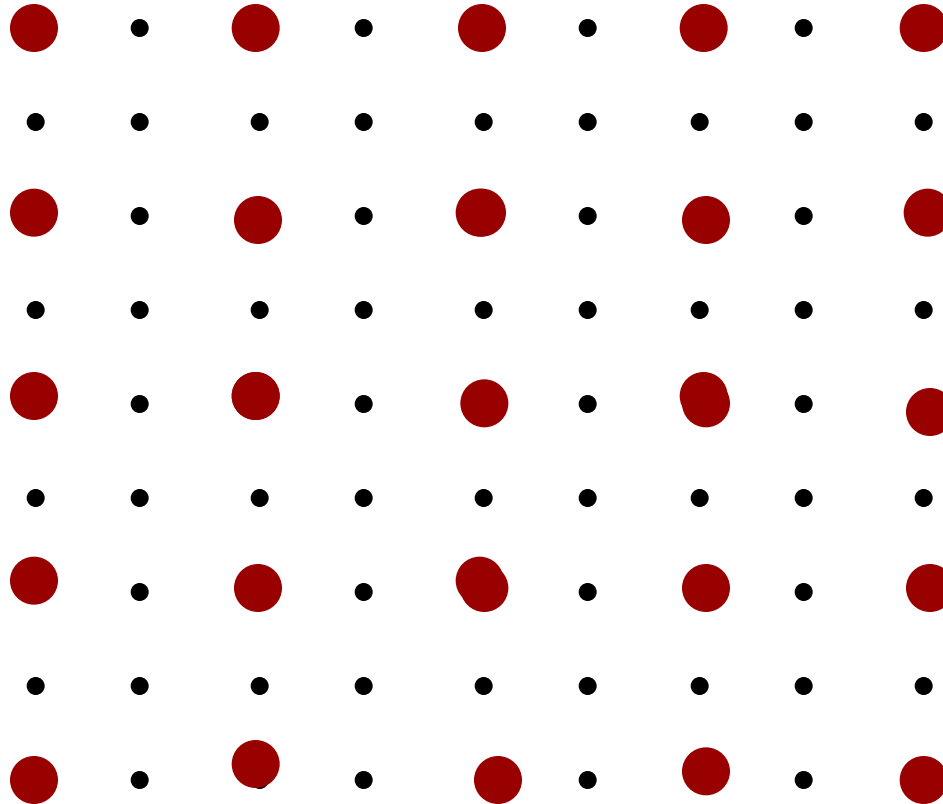
# Squares & Diamonds

- Diamond Schritt, grün wird neu berechnet



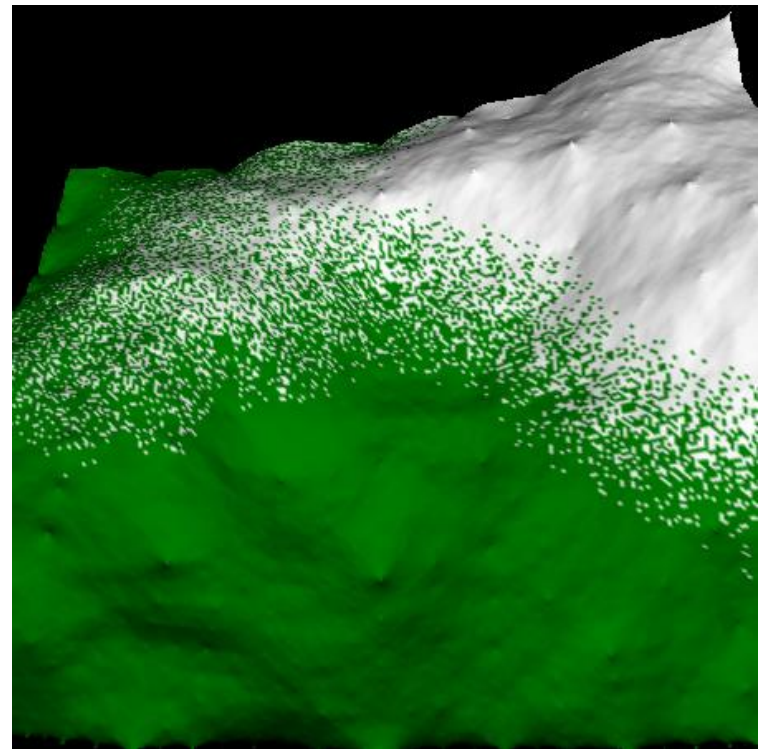
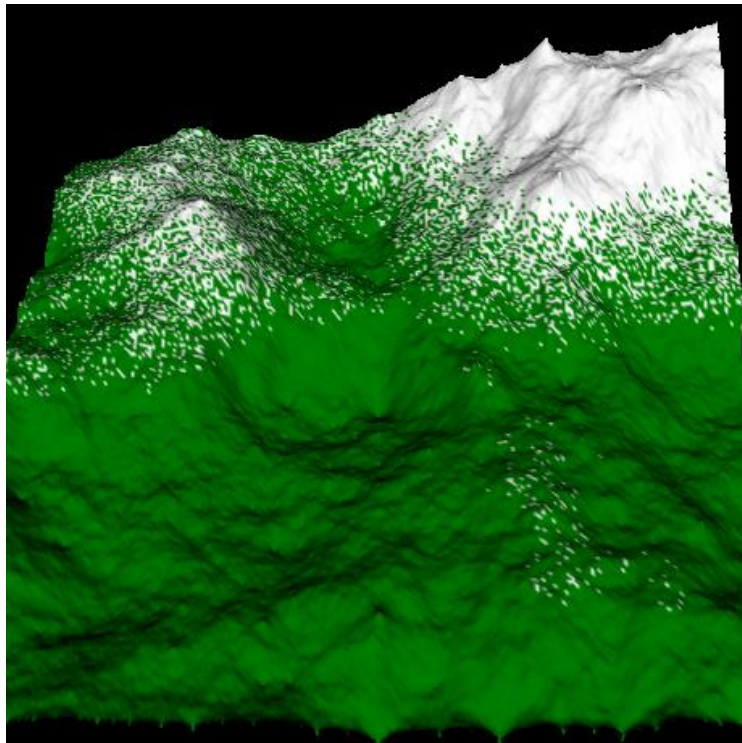
# Squares & Diamonds

- Ausgangslage nächster „grob nach fein“ Schritt



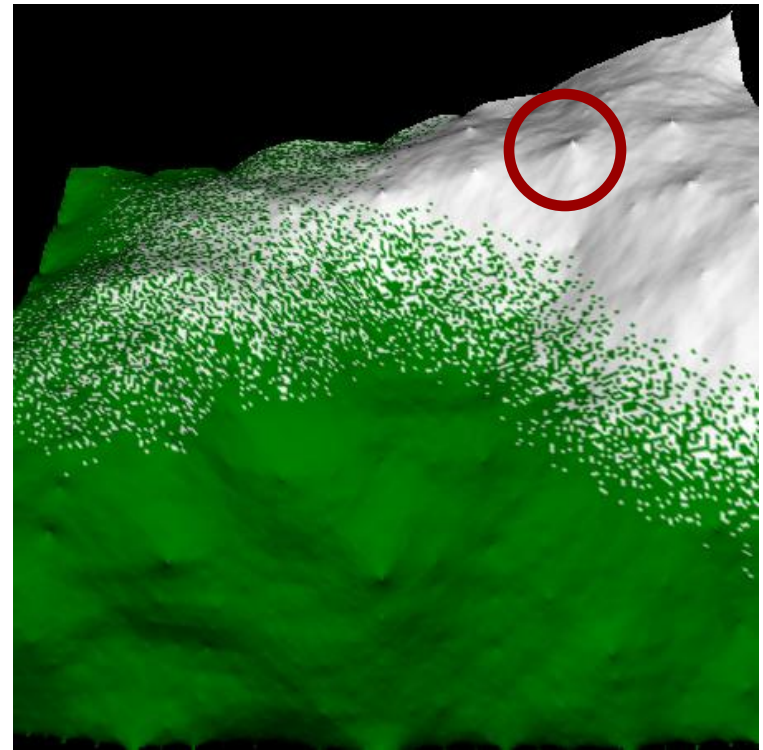
# Squares & Diamonds

- Iterieren bis alle Werte berechnet
- Zufällige Abweichungen sollen kleiner werden je feiner die Stufe
  - Freier Parameter, der Aussehen des Geländes bestimmt



# Bemerkungen

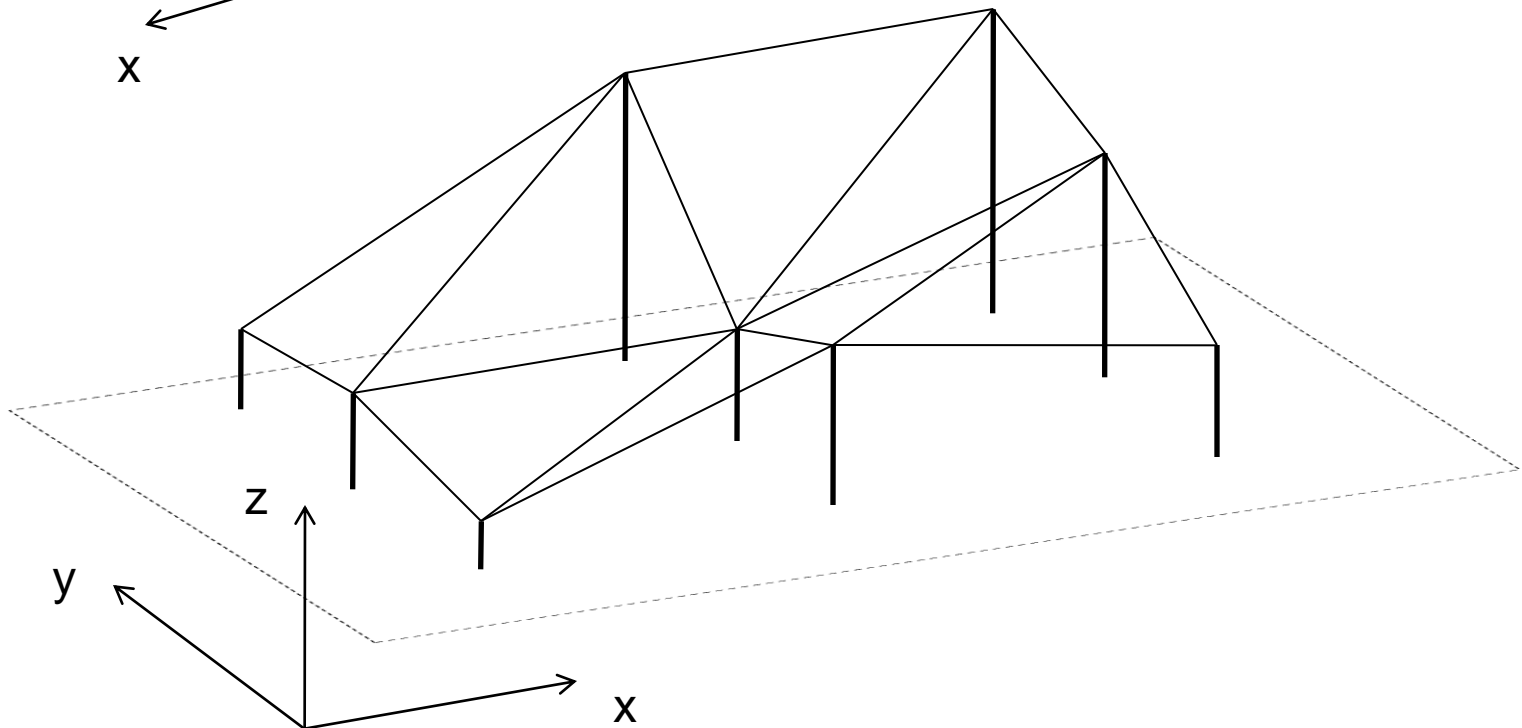
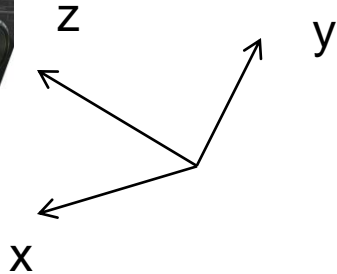
- Einfärbung des Terrains je nach Höhe und andere eigene Erweiterungen fakultativ
- „Spitzen“ im Terrain sind Konsequenzen (Artefakte) des Squares & Diamonds Algorithmus
- Normalen an jedem Eckpunkt berechnen





# 4. Interaktive Kamerabewegung

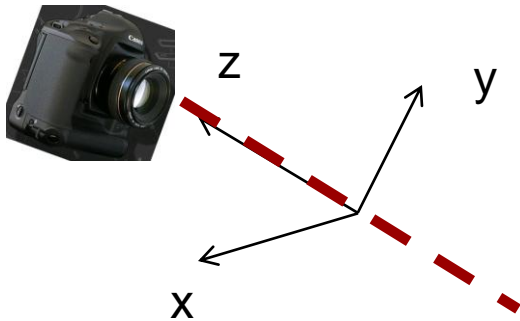
Kamerakoordinaten



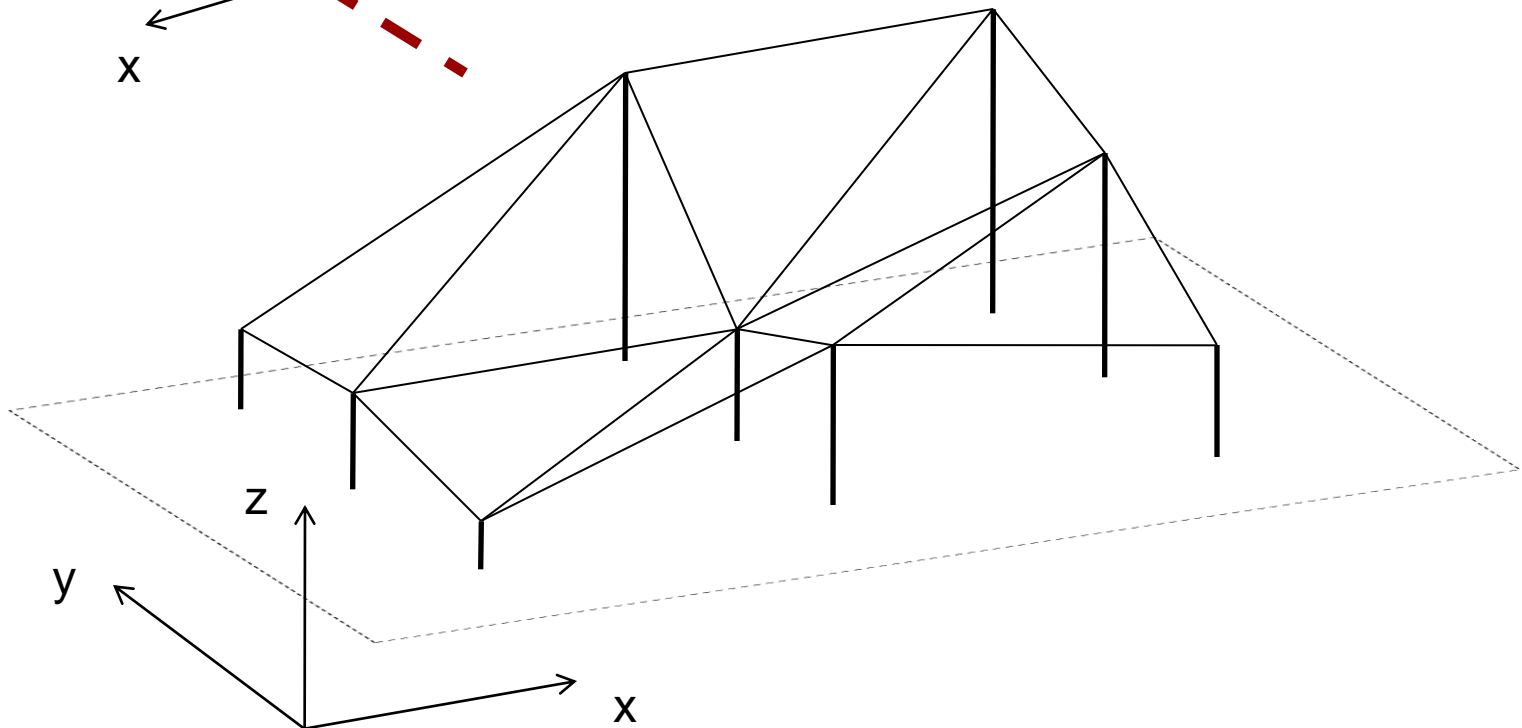
Weltkoordinaten

# 4. Interaktive Kamerabewegung

Kamerakoordinaten



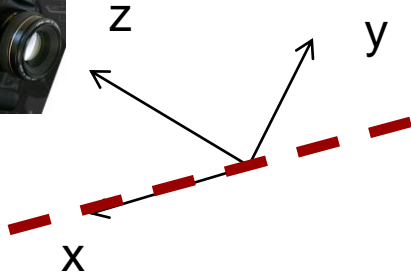
**Tasten W & S: vor & zurück**



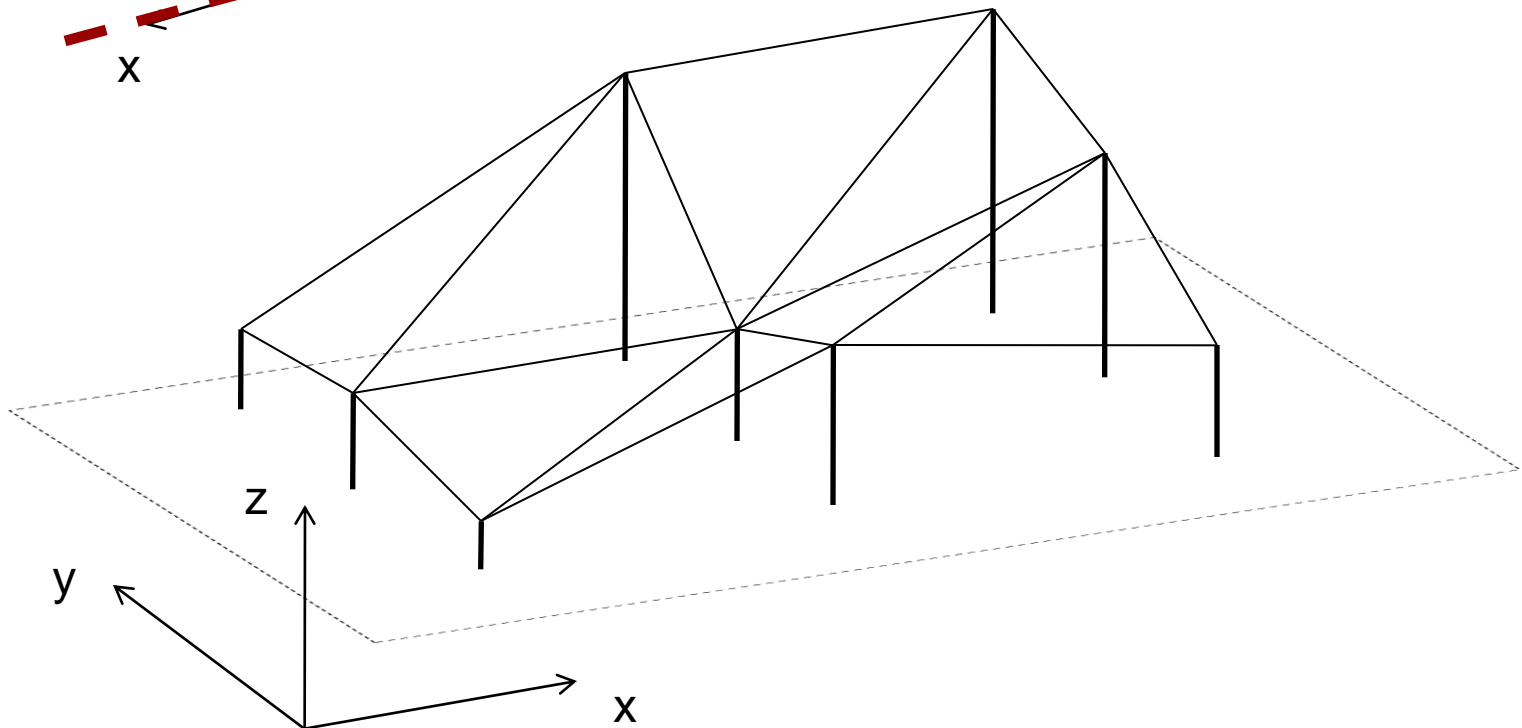
Weltkoordinaten

# 4. Interaktive Kamerabewegung

Kamerakoordinaten



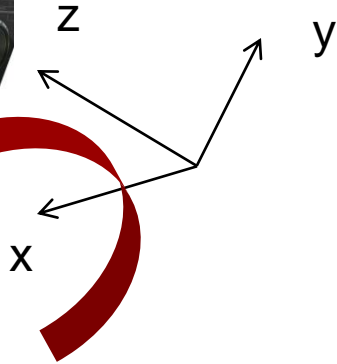
**Tasten A & D: links & rechts**



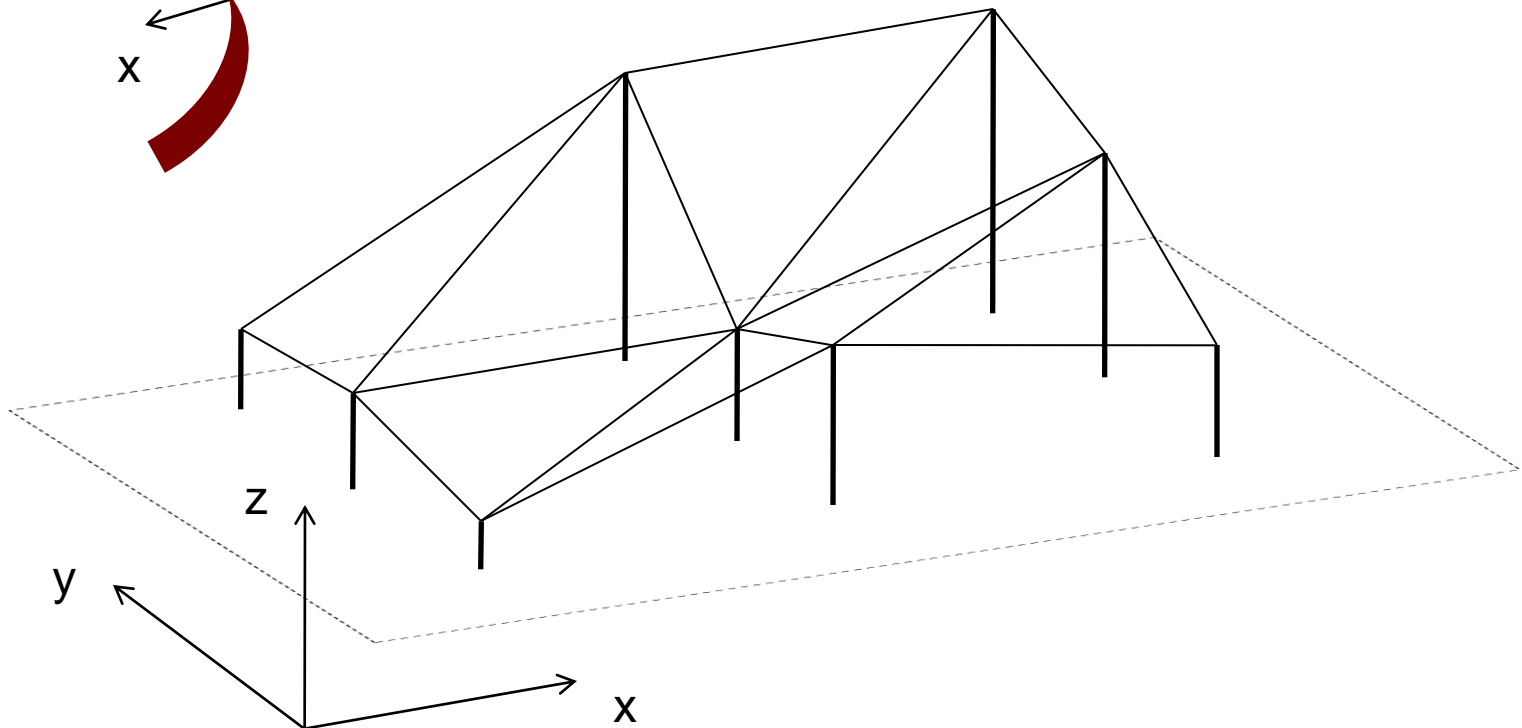
Weltkoordinaten

# 4. Interaktive Kamerabewegung

Kamerakoordinaten



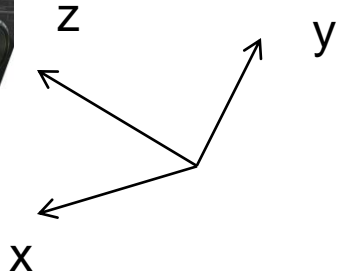
**Maus auf/ab: Rotation um Kamera x**



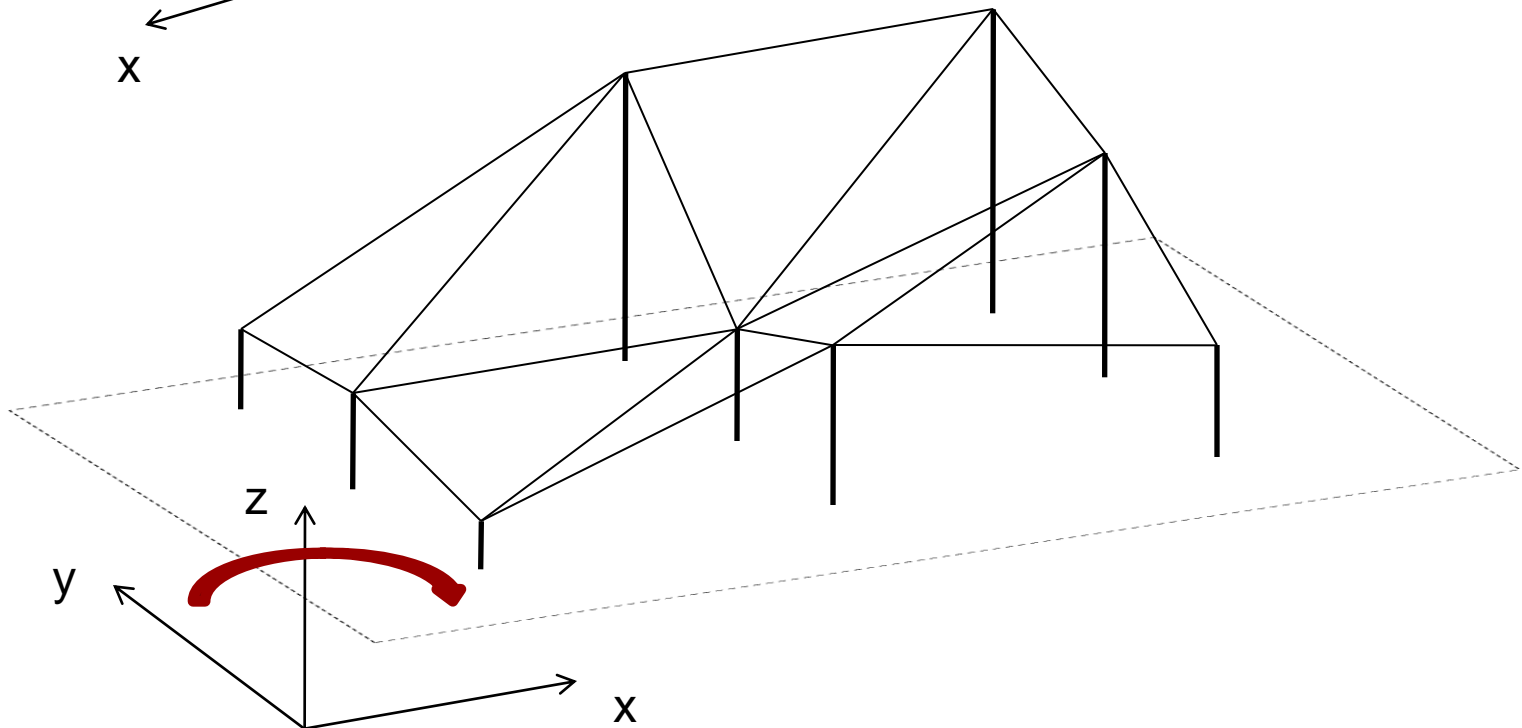
Weltkoordinaten

# 4. Interaktive Kamerabewegung

Kamerakoordinaten



**Maus links/rechts: Rotation um Welt z**



Weltkoordinaten

# Tastatureingabe in Java

```
public static class MyKeyListener implements KeyListener {  
    public void keyPressed(KeyEvent e) {  
        switch(e.getKeyChar()) {  
            case 's': ...; break;  
            case 'w': ...;break;  
            ...  
        }  
    }  
    public void keyReleased(KeyEvent e) {  
    }  
    public void keyTyped(KeyEvent e) {  
    }  
}
```

**In der main Methode:**

```
jframe.addKeyListener(new MyKeyListener());
```

# 4. Interaktive Kamerabewegung

- Demonstration