

Umsetzungsbeispiel 3D-Modellierung

Beschreibung des Artefakts

3D-Modellierung ist heutzutage aus industriellen Produktionsverfahren nicht mehr wegzudenken. Sie bilden die Grundlage überall dort, wo Objekte oder Systeme digital entworfen, analysiert oder hergestellt werden. Sei es im Maschinen- und Fahrzeugbau mit CNC-Verfahren, in der Architektur für die Gebäudeplanung oder in der Medizin, um Prothesen und Implantate zu erstellen. Auch im Hobbybereich findet 3D-Modellierung ihren Platz, um Modelle für 3D-Drucker zu erstellen. In gängigen Modellierungsumgebungen können Objekte durch „händische“ Aktionen, wie z. B. das Erstellen von Punkten, Geraden, Flächen und vordefinierten Körpern, durch die Nutzung der Computermouse erstellt werden. Eine Präzisierung ist durch das Hinzufügen von mathematischen Eigenschaften wie Zentrik, Tangentialität oder durch Bemaßung von Abständen möglich. Ebenso ist es möglich, die Beschreibung der Modelle mithilfe von Programmcode durchzuführen. Hier werden Objekte durch vordefinierte Klassen vorgegeben und durch Parameter präzisiert. Dadurch können die Vorteile der Programmierung genutzt werden, um effektiv Modelle zu beschreiben.

Beschreibung des Szenarios

Die Schüler:innen setzen sich mithilfe der Modellierungssoftware OpenScad mit der analytischen Geometrie auseinander. Dabei werden erst grundlegende Befehle und Objekte der Konstruktionssoftware vermittelt und anschließend dazu genutzt, mathematische Objekte der Vektorgeometrie zu beschreiben und zu visualisieren. So werden bereits bekannte Inhalte aus der Sicht der 3D-Modellierung betrachtet und vertieft. Ebenso werden algorithmische Grundstrukturen verwendet, um die Visualisierung von Ebenen und Geraden zu implementieren.

Basisangaben

Fach

Mathematik

Klassenstufe

11 Grundkurs



Schulform	Gymnasium	Lernbereich	LB 3	
Zeitbedarf	90 min	Kompetenzrahmen	5.2 5.4	5.5

Artefaktsteckbrief 3D-Modellierung

Wofür wird es genutzt?

Einsatzszenarien für 3D-Modellierung sind unter anderem:

1. Ingenieurwesen

- Maschinenbau (Zahnräder, Gehäuse, Motorenteile)
- Fahrzeugbau (Autos, Flugzeuge, Schiffe)
- Elektrotechnik (Gehäuse, Bauteilhalterungen)

2. Architektur und Bauwesen

- Gebäudeentwürfe
- Stadtplanung

3. Medizin

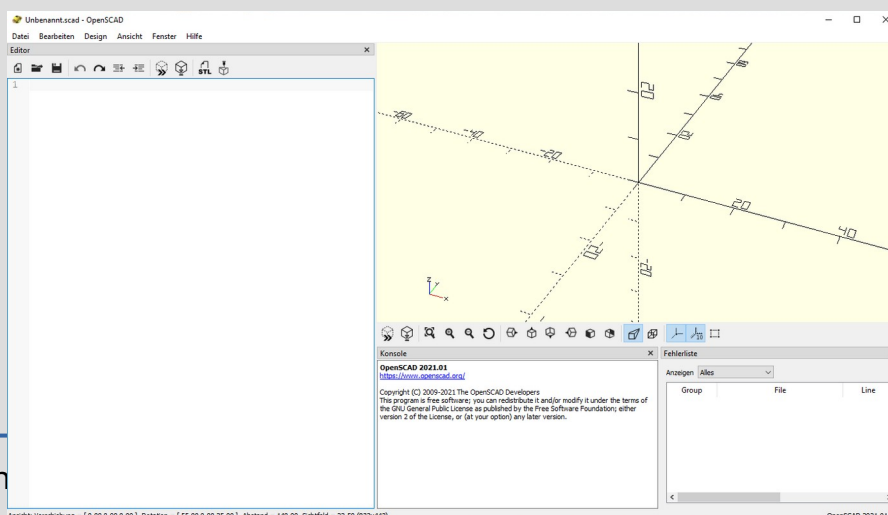
- Prothesen und Implantate
- Anatomische Modelle

4. 3D-Druck

- Prototypenbau
- Ersatzteile

Wie funktioniert es?

OpenScad ist eine Software um 3D-Modelle zu erzeugen. Dabei verhält sich das Programm wie ein 3D-Compiler, welcher Befehle und Beschreibungen aus einem Script ausliest, um aus diesen 3D-Objekte zu erzeugen.

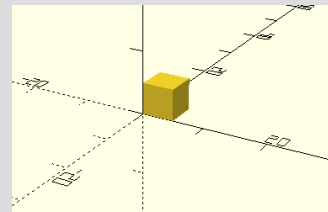
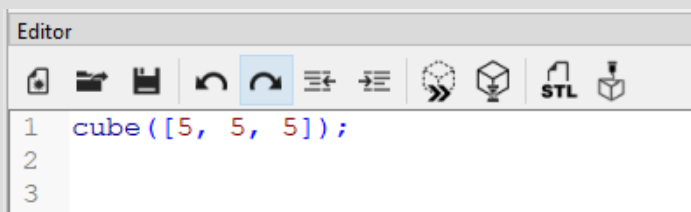


Im linken Teil können die Befehle eingegeben werden. Im rechten Teil werden aus diesen dann Objekte erzeugt und angezeigt.

@OpenScad

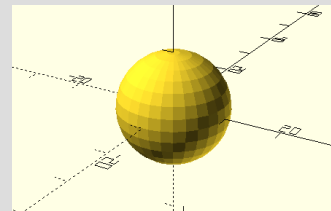
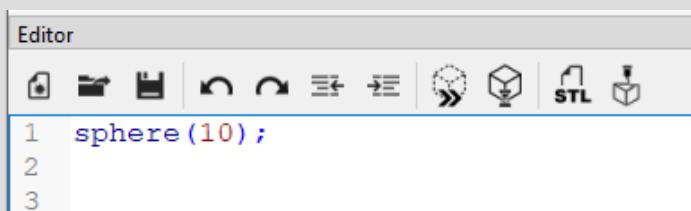
Objekte:

In OpenScad können durch einfache Befehle grundlegende Körper erschaffen werden. Zur Auswahl stehen Quader, Kugeln, Zylinder und Kegel. Die Eigenschaften der Körper, wie Höhen, Breiten oder Radien, werden mit Parametern beschrieben.



@OpenScad

@OpenScad

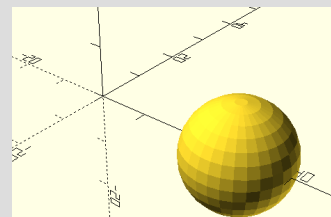
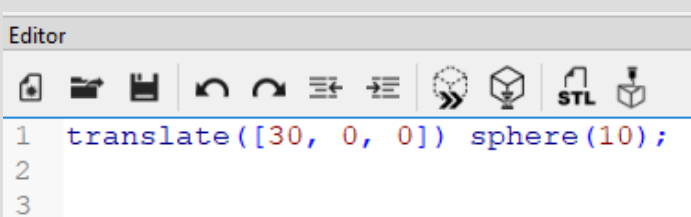


@OpenScad

@OpenScad

Transformationen:

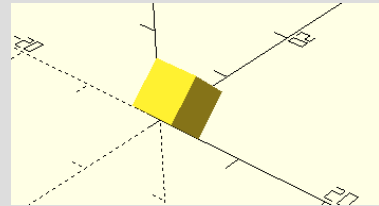
Um 3D-Objekte zu verschieben, nutzt man den Befehl `translate()` und für Drehungen `rotate()`. Hier lassen sich bereits erste Verbindungen zur Vektorgeometrie finden, da der Befehl `translate()` Objekte in x, y und z Richtung verschiebt und diese Verschiebungen als 3-Tupel, also Vektoren im Raum, interpretiert werden können.



@OpenScad

@OpenScad

```
Editor
1 rotate([45, 0, 0]) cube([5, 5, 5]);
2
3
```



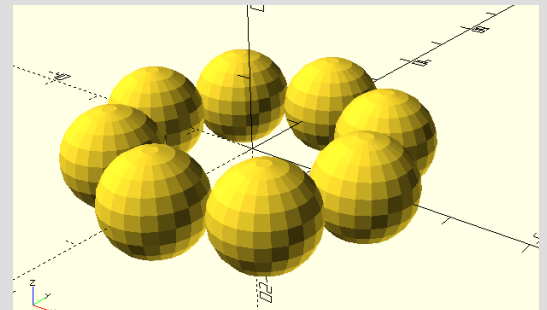
@OpenScad

@OpenScad

Programmierlogik:

In OpenScad können Elemente der Programmierung genutzt werden. So besteht die Möglichkeit Variablen und algorithmische Grundstrukturen zu verwenden, um 3D-Modelle zu erzeugen. So z. B. die klassische for-Schleife, um ein Objekt einmal grundlegend zu beschreiben und dessen Parameter durch die Iterationen der Schleife systematisch zu ändern.

```
Editor
1 for(i=[0:45:360]){
2     rotate([0, 0, i])
3     translate([18, 0, 0])
4     sphere(7);
5 }
```



@OpenScad

@OpenScad

Umsetzung im Unterricht

Technisch-/ infrastrukturelle Voraussetzungen

Für die Unterrichtseinheit benötigen alle Schüler: innen einen Computer oder Laptop, auf welchem die Software OpenScad installiert ist. Außerdem ist eine elektronische Tafel von Vorteil, um die Vorstellung der Software durch die Lehrkraft zu ermöglichen.

Weitere technische oder infrastrukturelle Voraussetzungen sind nicht nötig.

Curriculare Analyse

Fachlehrplan

Lernbereich 3: Vektoren, Geraden und Ebenen		32 Ustd.
Beherrschen der Darstellung geometrischer Objekte im räumlichen kartesischen Koordinatensystem		
Beherrschen der Addition, Subtraktion und Vielfachenbildung von Vektoren sowie der Rechengesetze für Vektoren	geometrische Veranschaulichungen, DGS Hinweis auf vektorielle Größen → PH, Lk 11, LB 3	
Anwenden von Vektoren beim Arbeiten mit geometrischen Objekten		
- Gleichungen von Geraden und Ebenen in Parameterform und in parameterfreier Form - Untersuchen der Lagebeziehungen einschließlich des Bestimmens von Schnittpunkten für die Fälle: · Gerade – Gerade · Gerade – Ebene	geometrische Veranschaulichung der linearen Abhängigkeit und der linearen Unabhängigkeit	

@Lehrplan Sachsen – Mathematik Gymnasium

Die Unterrichtseinheit ordnet sich thematisch am Ende des Lernbereichs 3 „Vektoren, Geraden und Ebenen“ der Klassenstufe 11 ein. Es wurden also schon die grundlegenden Objekte der Vektorgeometrie, Punkte im dreidimensionalen Raum, Vektoren, Geraden und Ebenen sowie die mathematischen Grundlagen, also die Addition, Subtraktion und Vervielfachung von Vektoren, das Skalarprodukt und das Vektorprodukt behandelt. Ebenso kann durch den Informatikunterricht erwartet werden, dass die Schüler: innen mit einfachen algorithmischen



Grundstrukturen arbeiten können und erste Programmiererfahrungen gesammelt haben. Vorwissen zu OpenScad ist allerdings nicht notwendig.

digitalisierungsbezogene Lernziele

#	MIA	K#	Qualifikationsebene Kompetenzrahmen	Handlungsebene
D1	A	5.2.1	Die Schüler:innen kennen und verwenden eine Vielzahl von digitalen Werkzeugen,	indem sie Objekte der Vektorgeometrie in OpenScad erstellen.
D2	M	5.2.2	Die Schüler: innen formulieren Anforderungen an digitale Werkzeuge formulieren,	indem Sie die Möglichkeiten von OpenScad auflisten und mit den gewünschten Anforderungen vergleichen.
D3	M	5.2.3	Die Schüler: innen identifizieren passende Werkzeuge zur Lösung identifiziere von Problemen,	indem Sie die Darstellungsmöglichkeiten von OpenScad mit denen dynamischer Geometriesoftware vergleichen.
D4	I/M	5.5.1	Die Schüler: innen kennen und verstehen Funktionsweisen und grundlegende Prinzipien der digitalen Welt kennen und verstehen,	indem Sie Darstellungsmöglichkeiten von Geraden und Ebenen in digitalen Umgebungen mit den mathematischen Vorstellungen vergleichen.
D5	I	5.5.2	Die Schüler: innen erkennen und formulieren algorithmische Strukturen in genutzten digitalen Tools erkennen und formulieren,	indem Sie die geläufige Darstellung von Ebenen und Geraden in Geometriesoftware in OpenScad algorithmisch implementieren.
D6	I	5.5.3	Die Schüler: innen erstellen und verwenden algorithmische Strukturen zur Lösung eines Problems planen und verwenden,	indem Sie Geraden und Ebenen durch algorithmische Grundstrukturen implementieren.
D7	M		Die Schüler: innen werten 3D-Modellierung als wichtigen technischen Beitrag zur Gesellschaft,	indem Sie Möglichkeiten und Vorteile der 3D-Modellierung aufzählen und mit analogen Mitteln vergleichen.

Unterrichtsverlaufsplanung

Legende		Unterrichtsmedium	Unterrichtswerkzeug	Unterrichtsgegenstand	
Zeit	Phase	Inhalt	SF/M ¹	Medien/Material	LZ Nr.
10 min	Einstieg und Motivation	Kurze Wiederholung zum Geraden und Ebenen. Grenzen der Visualisierung mit analogen Mitteln besprechen. → Brücke zur digitaler Darstellung	FU/UG	digitale Tafel	(D7)
15 min	Erarbeitung OpenScad	OpenScad: Benutzeroberfläche, grundlegende Befehle, Punkte darstellen, Verschiebungen durch translate() umsetzen	UG/EA	Computer, OpenScad, digitale Tafel, CheatSheet	D1
30 min	Geraden/Ebenen im Raum	Geradengleichung und Ebenengleichung in Software umsetzen → Geraden/Ebenen visualisieren Randfälle abfangen → drei Punkte kollinear (Ebene) → if-Struktur	UG/EA	Computer, OpenScad, CheatSheet	D6

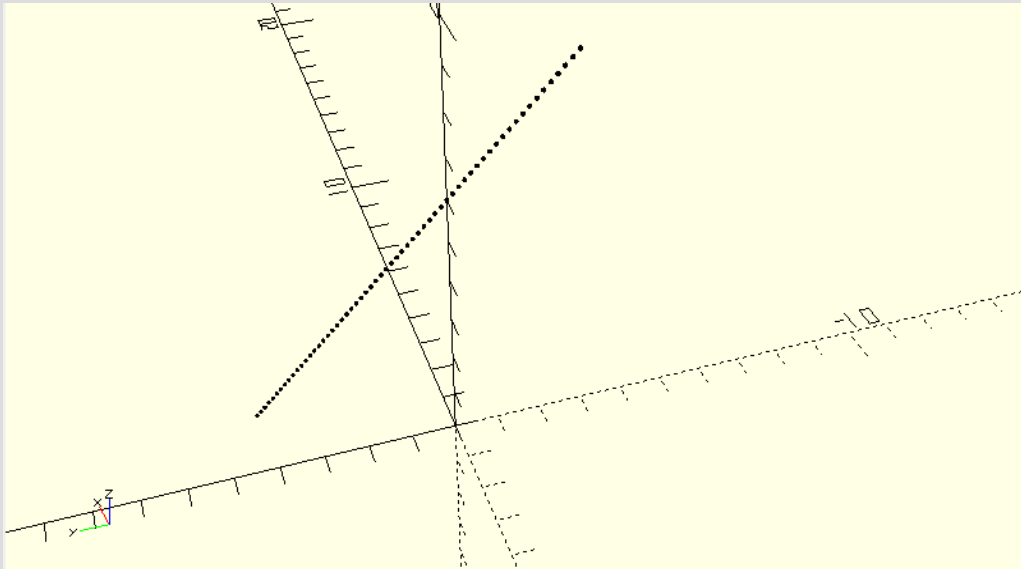
¹ SF – Sozialform (EA: Einzelarbeit, PA: Partnerarbeit, GA: Gruppenarbeit; FU: Frontalunterricht); M - Methode

Zeit	Phase	Inhalt	SF/M	Medien/Material	LZ Nr.
20 min	Diskrepanz zw. Darstellung und mathematischer Vorstellung	Diskussion der Unterschiede zw. digitaler Darstellung und der idealen mathematischen Vorstellung mathematischer Objekte: • z. B. Gerade als Menge von Punkten, Punkte in OpenScad als Kugeln darstellen → Punkte aus math. Sicht ohne Dimension. • z. B. Ebenen durch geschachtelte Schleifen nur lokal bzw. endlich darstellbar • Vergleich mit bereits bekannten Tools (GeoGebra) und deren Umsetzungsvarianten • Alternativen Erkunden und diskutieren Geraden/Ebenen durch Bearbeitung der Schleifenparameter verändern	UG	Computer, OpenScad, CheatSheet	D2, D3, D4, D5
15 min	Abschluss, Reflexion	Diskussion und Auswertung von: • Möglichkeiten ... • Vorteilen ... • Grenzen ... • mathematischer Genauigkeit ... • weiteren Einsatzgebieten ... der 3D-Modellierung	UG		D7

Arbeitsmaterialien

1 – Code für Gerade/Ebene und Bemerkungen für Unterrichtsverlauf

```
$fn = 20;  
  
// ----- Gerade -----  
p = [1,1,4]; // Stützpunkt  
v = [1,1,-2]; // Richtungsvektor  
  
// ----- Visualisierung Gerade -----  
color("black")  
for (t=[-3:0.1:3])  
    translate(p + t*v) sphere(0.05); //Geradengleichung
```



@OpenScad

Bemerkungen:

- Gerade als Menge der Punkte welche Geradengleichung erfüllen
- In OpenScad Punkte durch Kugeln (sphere) darstellen → haben dann Dimensionen → Unterschied zur mathematischen Vorstellungen
- Gerade besteht aus unendlich vielen Punkten → in OpenScad nur ein Teilausschnitt visualisierbar → Darstellungsgenauigkeit (Anzahl der Punkte) durch Anpassung der Schleife änderbar
- Vergleich mit bekannter Geometriesoftware → dort Darstellung von Geraden als „kontinuierliches“ Objekt

- Erkundung von alternativen Darstellungsmöglichkeiten → Gerade als Zylinder in OpenScad quasi kontinuierlich → Vorteile und Nachteile diskutieren

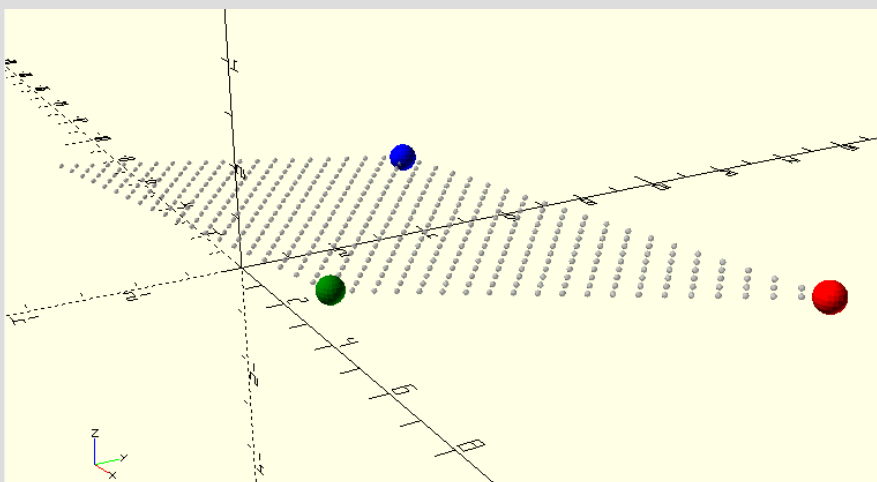
```
$fn = 20;

// ----- Punkte -----
A = [10,5,2];
B = [4,0,1];
C = [1,3,2];

// ----- Spannvektoren -----
u = B - A;
v = C - A;

// ----- Hervorhebung Punkte -----
color("red")   translate(A) sphere(0.25);
color("green") translate(B) sphere(0.25);
color("blue")  translate(C) sphere(0.25);

// ----- Ebene darstellen -----
color([0.8,0.8,0.8,0.6])
for (s=[0:0.05:1])
for (t=[0:0.05:1]) {
    translate(A + s*u + t*v)
        sphere(0.05);
}
```



@OpenScad

Bemerkungen:

- Ebene als Menge an Punkten welche Ebenengleichung erfüllen
- Thematik der Punkte und der Endlichkeit wie bei Gerade
- Hier begrenzen die Spannvektoren die Ebene → Schleifenparameter bestimmen die Ausdehnung der Ebene → Parameter ändern, um Ausdehnung zu verändern
- Vergleich mit bekannter Geometriesoftware → dort Darstellung von Ebenen als kontinuierliches Objekt
- Erkundung von alternativen Darstellungsmöglichkeiten → Ebene als flachen Quader in OpenScad quasi kontinuierlich → Vorteile und Nachteile diskutieren

Ergänzung zur Ebene → Kollinearität abfangen → if-Struktur

```
$fn = 20;

// ----- Hilfsfunktionen -----
function kreuzp(a,b) = [
    a[1]*b[2]-a[2]*b[1],
    a[2]*b[0]-a[0]*b[2],
    a[0]*b[1]-a[1]*b[0]
];

function norm(v) = sqrt(v*v);

// ----- Punkte -----
A = [1,1,1];
B = [2,2,2];
C = [3,3,3];

// ----- Spannvektoren -----
u = B - A;
v = C - A;

// ----- Kollinearität prüfen -----
kollinear = (norm(kreuzp(u,v)) == 0);

if(kollinear){
    //Meldung als Text in Anzeigefenster
    color("red")
        translate([0,0,0])
            text("Punkte sind kollinear!", size=5, halign="center",
                valign="center");
} else {
    //Punkte und Ebene wie oben im Code
}
}
```



@OpenScad

2 CheatSheet OpenScad

Das CheatSheet ist im Anhang platziert.

3 Quellen

Sächsisches Staatsministerium für Kultus. (2021). *Lehrplan Gymnasium – Mathematik* [Lehrplandatenbank]. Schulportal Sachsen.
<https://www.schulportal.sachsen.de/lplandb/index.php?lplanid=461&lplansc=SPos4aG4oIhjKYDcHVVW8&token=106de253fd303e81bb3a04f4022d82d8> (Zugriff am 11. Februar 2026).

OpenSCAD. (2026). Version 2026.02 [Computer-Software]. <https://openscad.org> (Zugriff am 11.02.2026).