

KI-Workshop 2
WORK IN PROGRESS - XX.XX.2026

Table of contents

Vorstellung Erfindergeist Jülich e.V	3
Spenden	3
Wie läuft es heute ab	5
Wie läuft es heute ab: Theorie	5
Wie läuft es heute ab: Lokal-Demos	5
Wie läuft es heute ab: Cloud-Demos	5
Wie läuft es heute ab: HuggingFace	6
Wie läuft es heute ab: Unternehmens-Showcase	6
Wie läuft es heute ab: Fragen & Diskussion	6
Was ist KI?	7
Was ist KI?: Was ist KI nicht	7
Token	8
Token: Beispiele	8
Token: Kosten ChatGPT	8
Token: Beispiel Rechnung Buch	8
Token: Was sind Parameter?	9
Token: Präzision	9
Token: RAM Anforderungen	9
Token: Tokens pro Sekunde	9
Token: Was sagt uns das nun?	10
3D-Modellierung: OpenSCAD + Claude Code	11
3D-Modellierung: Was ist OpenSCAD?	11
3D-Modellierung: KI als Programmierpartner	11
3D-Modellierung: Beispiel	12
3D-Modellierung: Ergebnis — Hexenturm Jülich	12
3D-Modellierung: Modularer Aufbau	13
3D-Modellierung: Modularer Aufbau prompts	13
3D-Modellierung: Details im Code	15
3D-Modellierung: Detail prompts	15
Lokal: Ollama und OpenWeb UI	17
Lokal: OpenCode	18

Lokal: ConfyUI	19
Cloud: Meshy	20
HuggingFace	21
HuggingFace: Die Plattform	21
HuggingFace: FLUX.1 — Text zu Bild	22
HuggingFace: Was ist FLUX.1?	22
HuggingFace: Wie funktioniert Text-to-Image?	22
HuggingFace: Tipps für gute Prompts	22
HuggingFace: Wan2.2 — Text zu Video	24
HuggingFace: Was ist Wan2.2?	24
HuggingFace: FP8 — Warum läuft das im Browser?	24
HuggingFace: Whisper — Sprache zu Text	25
HuggingFace: Was kann Whisper?	25
HuggingFace: Wo wird Whisper eingesetzt?	25
HuggingFace: Kokoro TTS — Text zu Sprache	26
HuggingFace: Was kann Kokoro TTS?	26
HuggingFace: Whisper + Kokoro = Sprach-Pipeline	26
HuggingFace: Pixal3D — Bild zu 3D-Modell	27
HuggingFace: Was ist Pixal3D?	27
HuggingFace: Pixal3D vs. OpenSCAD + KI	27
Microsoft	28
Cloud: Azure Showcases	29
Termine-Webseite	30
Plan erstellen	30
Reicht ein Plan?	30
Prompt-Beispiel: Herzanimation 1/2	30
Prompt-Beispiel: Herzanimation 2/2	30
Claude.md	30
Prompt-Beispiel: Archivments	31
Referenzen	32
Spenden	33
Ende	35

Vorstellung Erfindergeist Jülich e.V

- Gemeinnütziger Verein mit Sitz in Jülich, gegründet 2021
- **Ziel:** Förderung von Kreativität und Innovation durch praktisches Lernen und Zusammenarbeit
- Angebote
 - Workshops und Kreativ-Tage (Robotik-, Künstliche Intelligenz-, Podcast-Workshops, etc.)
 - Repair Cafe
 - Offene Werkstatt (3D-Drucker, Lasergravur, Holzverarbeitung, Textilveredelung, etc.)

Speaker notes go here.

Spenden

- Erfindergeist Jülich e.V. finanziert sich ausschließlich durch Mitgliedsbeiträge und Spenden. Wir Danken allen Unterstützern, die unsere Arbeit ermöglichen!
- Spenden Konto und PayPal auf unserer Webseite <https://linktree.erfindergeist.org/>



Wie läuft es heute ab

Heute führen wir Sie Schritt für Schritt durch die Welt der KI — von den Grundlagen bis zu echten Unternehmensanwendungen.

Wir zeigen, demonstrieren und beantworten Ihre Fragen.

Wie läuft es heute ab: Theorie

Was wir erklären:

- Was ist KI — und was ist sie *nicht*?
- Wie funktionieren große Sprachmodelle?
- Was sind Token und warum sind sie wichtig?

Ziel: gemeinsames Grundverständnis für alles, was danach kommt.

Wie läuft es heute ab: Lokal-Demos

Was wir live vorführen — auf unserer eigenen Hardware:

- **Ollama + OpenWeb UI** — KI-Chatbot ohne Internet und Cloud
- **OpenCode** — KI unterstützt beim Programmieren
- **ComfyUI** — KI generiert Bilder aus Text

Sie sehen, wie leistungsfähig KI auf lokalem Equipment läuft.

Wie läuft es heute ab: Cloud-Demos

Was wir über Cloud-Dienste zeigen:

- **3D-Modellierung: OpenSCAD + Claude Code** — 3D-Modelle aus Text erstellen (technisch)
- **Meshy** — 3D-Modelle aus Text oder Skizzen erstellen (Kreativ)
- **Azure KI-Dienste** — Unternehmensgrade KI-Integration von Microsoft

Sie sehen den Unterschied zwischen lokalem Betrieb und Cloud-Skalierung.

Wie läuft es heute ab: HuggingFace

Was wir über HuggingFace zeigen:

- **FLUX.1** — Text-zu-Bild: aus einer Beschreibung wird ein Bild generiert
- **Wan2.2** — Text-zu-Video: aus einer Beschreibung wird ein kurzes Video generiert
- **Whisper** — Sprache-zu-Text: gesprochene Sprache wird automatisch transkribiert
- **Kokoro TTS** — Text-zu-Sprache: Texte werden in natürliche Stimmen umgewandelt
- **Pixal3D** — Bild-zu-3D: ein Foto wird in ein 3D-Modell verwandelt

Offene Modelle, die jeder kostenlos ausprobieren kann — direkt im Browser.

Wie läuft es heute ab: Unternehmens-Showcase

Echte KI-Anwendungen aus der Praxis:

- Eine Firma stellt mehrere konkrete Fallbeispiele vor
- Von der Problemstellung zur KI-Lösung
- Was funktioniert gut — und wo sind die Grenzen?

Praxisbezug für alle, die KI im Berufs- oder Unternehmenskontext einsetzen wollen.

Wie läuft es heute ab: Fragen & Diskussion

Sie sind nicht nur Zuschauer:

- Nach jeder Phase: Zeit für Fragen
- Am Ende: offene Diskussionsrunde

Kein Vorwissen nötig — jede Frage ist willkommen.

Was ist KI?

- **Künstliche Intelligenz (KI)** bezeichnet Computersysteme, die menschliche Fähigkeiten wie Lernen, Denken oder Entscheiden nachahmen. Indem es Mathematische Modelle und Algorithmen verwendet, kann KI Muster erkennen, Vorhersagen treffen und Probleme lösen.
- KI umfasst viele Bereiche: Bilderkennung, Sprachverarbeitung, Empfehlungssysteme und mehr.
- **Large Language Models (LLM)** sind eine spezielle Form der KI, trainiert auf riesigen Textmengen — sie verstehen und erzeugen Sprache, beantworten Fragen und helfen bei der Analyse von Informationen.

Was ist KI?: Was ist KI nicht

- Experiment. besuchen Sie <https://chatgpt.com/> und fragen Sie “Nenne mir eine zufällige Zahl.”
- KI generiert Antworten, indem es die wahrscheinlichsten nächsten Wörter vorhersagt, basierend auf den Mustern, die es in den Trainingsdaten gelernt hat. Daher ist die Antwort nicht wirklich zufällig, sondern eine Zahl, die in den Trainingsdaten häufig vorkommt oder als wahrscheinlich angesehen wird. In diesem Fall könnte es sein, dass die Zahl 73 oft in den Daten vorkommt.

Token

- Token sind die kleinsten Texteinheiten, in die ein KI-Modell Text zerlegt. Gleichzeitig dienen sie als Abrechnungseinheit bei Cloud-Diensten.
- Jedes Wort und Satzzeichen zählt meist als 1 Token. Leerzeichen werden typischerweise dem nächsten Token vorangestellt, nicht als eigenes Token gezählt.
- Ein Token ist nicht immer ein ganzes Wort. Das Wort „Apfel“ ist vielleicht ein Token, „Verständnisfragen“ hingegen werden in mehrere Tokens zerlegt.

Token: Beispiele

- Beispiel: „Hallo Welt!“ → 3 Token („Hallo“, „ Welt“, „!“).
- Faustregel: 100 Token entsprechen etwa 75 Wörtern.
- Interaktiv ausprobieren: platform.openai.com/tokenizer

Token: Kosten ChatGPT

<p>GPT-5.4</p> <p>Unser leistungsfähigstes Modell für professionelle Arbeit.</p> <p>Preis</p> <p>Eingabe: 2,50 \$ / 1 Million Tokens</p> <p>Zwischengespeicherte Eingabe: 0,25 \$ / 1 Million Tokens</p> <p>Output: 15,00 \$ / 1 Million Tokens</p>	<p>GPT-5.4 mini</p> <p>Unser bisher stärkstes Mini-Modell für Programmierung, Computernutzung und Subagenten.</p> <p>Preis</p> <p>Eingabe: 0,750 \$ / 1 Million Tokens</p> <p>Zwischengespeicherte Eingabe: 0,075 \$ / 1 Million Tokens</p> <p>Output: 4,500 \$ / 1 Million Tokens</p>	<p>GPT-5.4 nano</p> <p>Unser günstigstes Modell der GPT-5.4-Klasse für einfache Aufgaben mit hohem Volumen.</p> <p>Preis</p> <p>Eingabe: 0,20 \$ / 1 Million Tokens</p> <p>Zwischengespeicherte Eingabe: 0,02 \$ / 1 Million Tokens</p> <p>Output: 1,25 \$ / 1 Million Tokens</p>
---	--	---

Token: Beispiel Rechnung Buch

- Moderne Krimis haben etwa 100.000 Wörter, das wären ca. 133.000 Token.

Auszug
<https://openai.com/de-DE/api/pricing/>
vom 09.04.2026
— Preise ändern sich häufig, bitte aktuell prüfen.

- Bei ChatGPT (GPT-4o) würde dies ca. **0,33 USD** kosten — bei teureren Modellen entsprechend mehr (vgl. nächste Folie).
- Achtung: Jedes KI-Modell hat ein begrenztes **Kontextfenster** — die maximale Anzahl an Token, die es auf einmal verarbeiten kann (z.B. 128.000 Token bei GPT-4o). Ein ganzes Buch übersteigt dieses Limit und müsste in kleinere Abschnitte aufgeteilt werden, was die Kosten erhöhen könnte. Das Kontextfenster erklärt auch, warum eine KI frühere Gesprächsteile “vergessen” kann.

Token: Was sind Parameter?

Wer KI lokal betreiben will, muss verstehen: wie viel RAM braucht ein Modell? Das hängt direkt von den Parametern ab.

- Parameter sind die gelernten Zahlenwerte eines Modells. Ein **7B-Modell** enthält **7.000.000.000 solcher Zahlen** — sie bestimmen gemeinsam, wie das Modell auf Eingaben reagiert.
- Je mehr Parameter, desto leistungsfähiger — aber auch desto mehr RAM wird benötigt.

Token: Präzision

Präzision beschreibt, wie viele Bits eine Zahl im Modell belegt. Mehr Bits = mehr RAM, aber nicht zwingend bessere Alltagsqualität.

RAM-Bedarf bei voller Präzision (FP16 = 16 Bit pro Zahl, 2 Bytes/Parameter)

Token: RAM Anforderungen

- 3B (3 Milliarden Parameter): ~6 GB
- 7B (7 Milliarden Parameter): ~14 GB
- 13B (13 Milliarden Parameter): ~26 GB
- 70B (70 Milliarden Parameter): ~140 GB

In der Praxis verwendet Ollama quantisierte Modelle (4-bit) — jede Zahl belegt nur noch 4 statt 16 Bit, der RAM-Bedarf sinkt auf **25 %**, bei kaum merklichem Qualitätsverlust — z.B. ~4–5 GB für ein 7B-Modell.

Token: Tokens pro Sekunde

- **Tokens pro Sekunde (TPS)** ist das lokale Pendant zur Cloud-Abrechnung: statt Kosten pro Token misst man hier die Geschwindigkeit — wie viele Token das Modell pro Sekunde ausgibt.
- Beispiel: eine Nvidia RTX 4090 (24 GB VRAM) erreicht mit Ollama typischerweise:
 - ~80–120 TPS für ein 7B-Modell (Q4) — flüssig lesbar
 - ~40–60 TPS für ein 13B-Modell (Q4) — ebenfalls flüssig
 - 70B-Modelle (Q4 ~35 GB) passen nicht mehr in den VRAM einer einzelnen Karte

Token: Was sagt uns das nun?

	Cloud	Lokal
Token =	Abrechnungseinheit	Geschwindigkeitsmaß (TPS)
Typische Modelle	500B+ Parameter	7B–13B Parameter (quantisiert)
Kosten	Laufend, pro Token	Einmalig (Hardware)
Kontextfenster	128K+ Token	4K–128K Token (modellabhängig)
RAM-Bedarf	Kein eigener	4–26 GB je nach Modell

3D-Modellierung: OpenSCAD + Claude Code

OpenSCAD ist eine Programmiersprache für 3D-Modelle — statt mit der Maus zu modellieren, beschreibt man das Modell als Code.

Das Prinzip: Prompt an Claude Code → SCAD-Code → fertiges 3D-Modell

Beispiel: Der **Hexenturm Jülich** — ein historisches Wahrzeichen, nachgebaut aus Code.

3D-Modellierung: Was ist OpenSCAD?

- OpenSCAD beschreibt 3D-Objekte als **parametrischen Code**
- Grundformen wie Zylinder, Quader und Kugeln werden kombiniert, subtrahiert und transformiert
- Parameterisierung ermöglicht einfache Änderung auch ohne Kenntnisse
- Ideal für technische und architektonische Modelle

```
// Beispiel: einfacher Turm mit Kuppel
cylinder(h=50, r=10);
translate([0,0,50])
  sphere(r=5);
```

```
// Beispiel: parameter
// Breite
roof_w      = 100;
//Tiefe
roof_d      = 90;
...
```

3D-Modellierung: KI als Programmierpartner

Wie Claude Code beim Modellieren hilft:

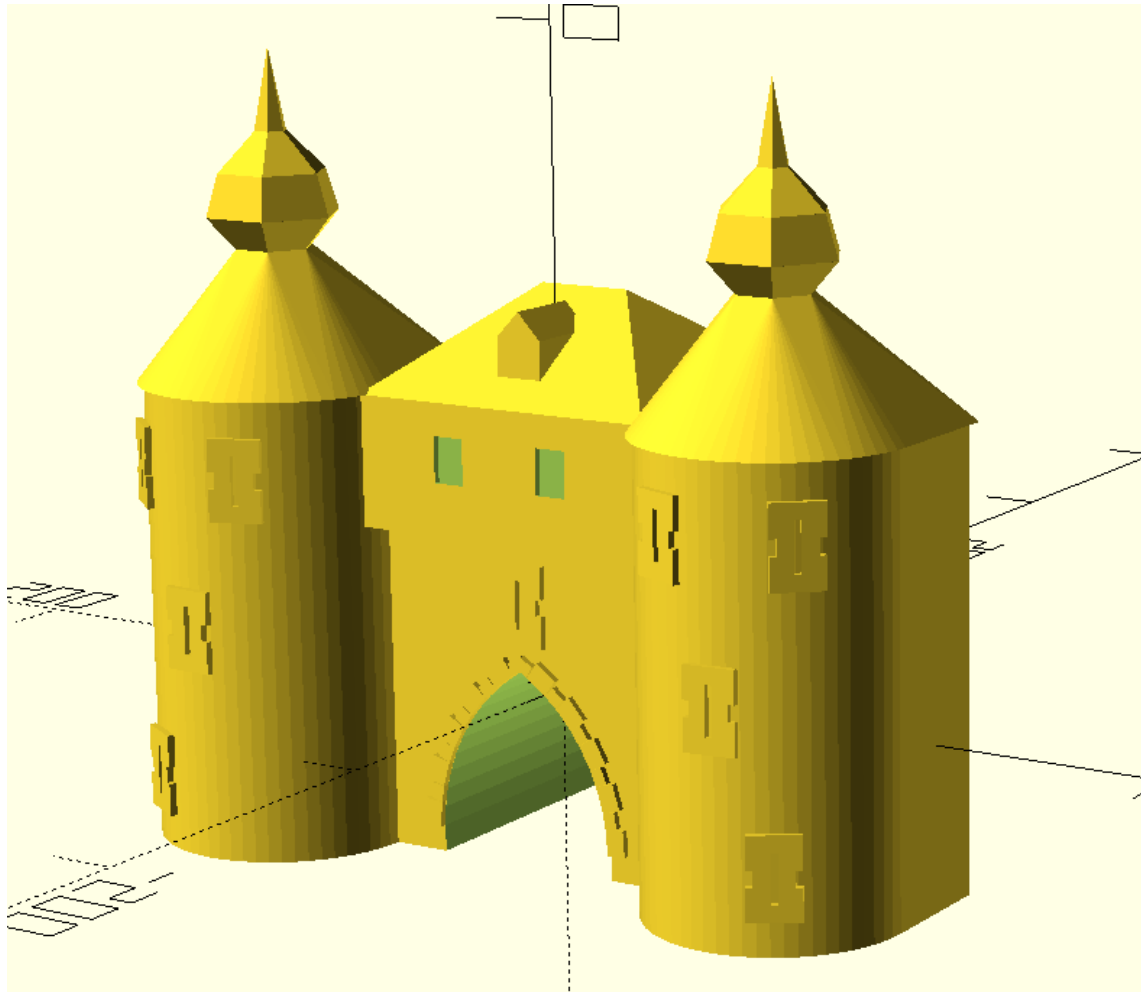
1. Beschreibung des gewünschten Objekts in natürlicher Sprache
2. Claude Code generiert den OpenSCAD-Code
3. Modell wird geprüft und verfeinert — durch weitere Prompts
4. Iterativer Prozess: Feedback → Anpassung → Verbesserung

3D-Modellierung: Beispiel

“Erstelle einen zylindrischen Turm mit konischem Dach und einer spitzen Turmspitze”

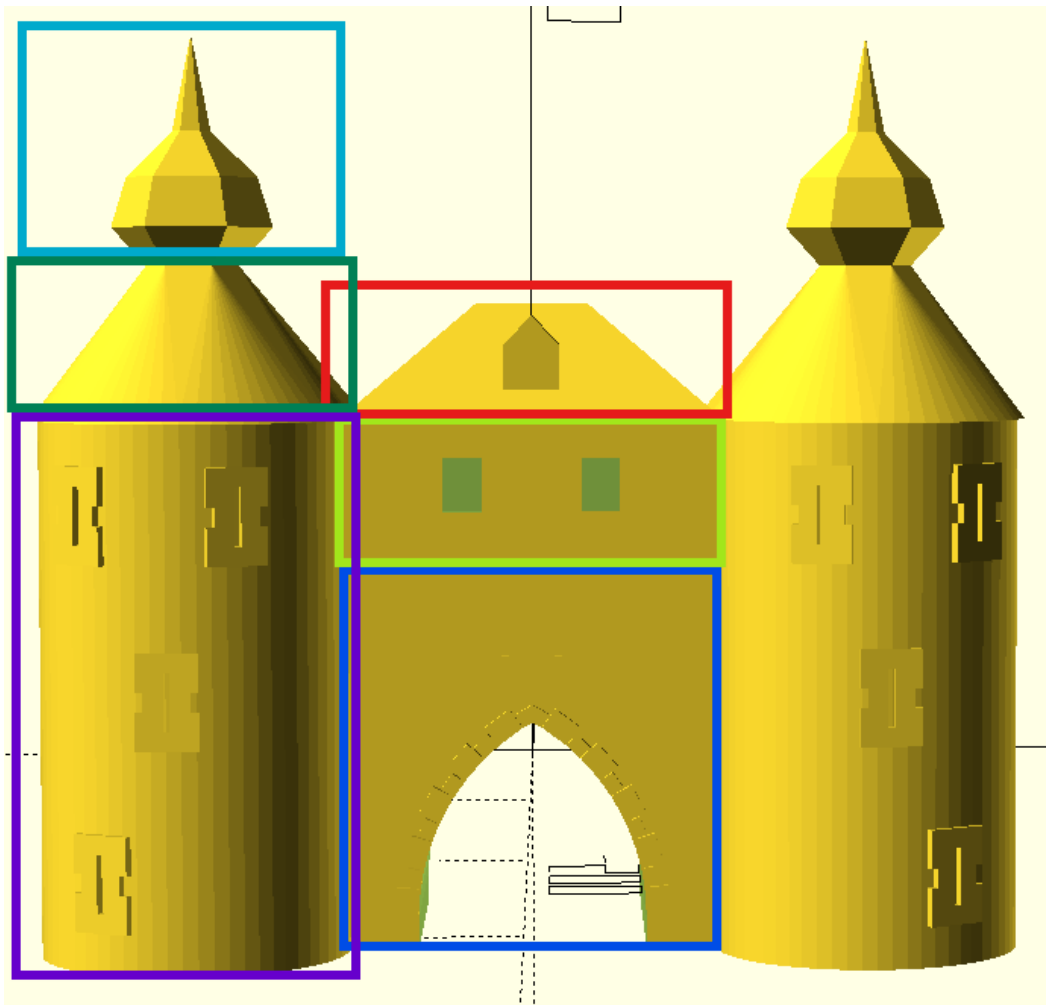
KI ersetzt hier keine CAD-Kenntnisse — sie macht den Einstieg drastisch einfacher.

3D-Modellierung: Ergebnis — Hexenturm Jülich



Der fertige Hexenturm — vollständig aus OpenSCAD-Code generiert, mit Hilfe von Claude Code.

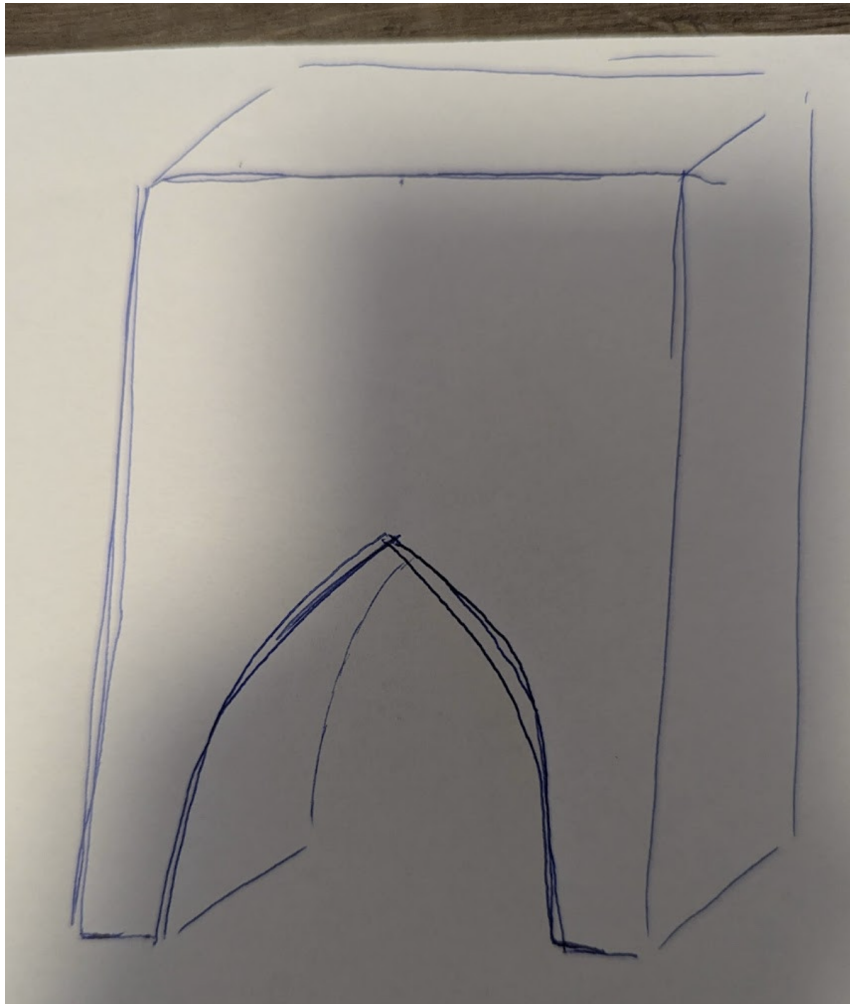
3D-Modellierung: Modularer Aufbau



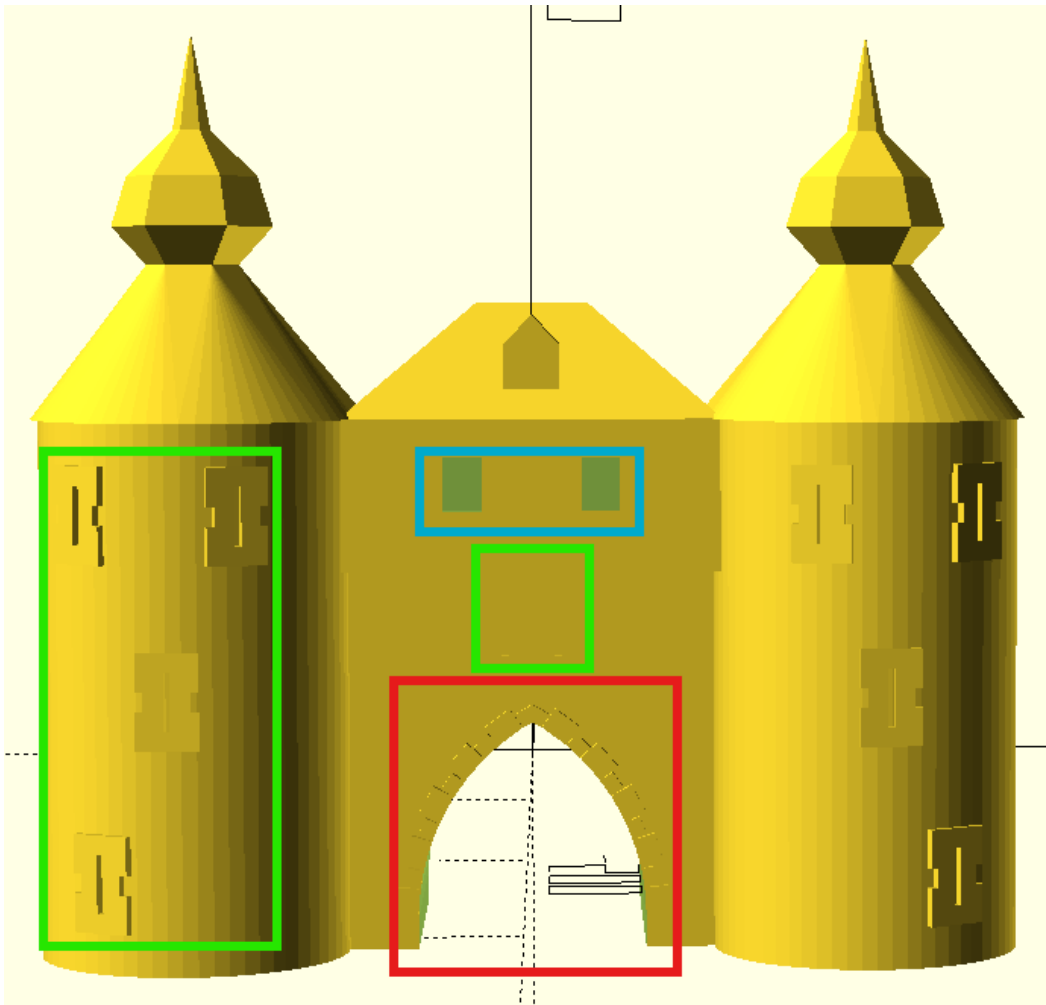
3D-Modellierung: Modularer Aufbau prompts

” Erstelle ein Rechteck wo ein Tunnel durchgeht. anbei ein Bild wie ich es mir vorstelle. Generiere Parameter für x,y,z sowie breite und höhe des quaders und Tunnels. zentriere das Objekt.”

Das Modell besteht aus einzelnen Modulen:
Turmspitze,
Kegelhelm,
Turmschaft,
Mittelgebäude
und
Toreinfahrt —
jedes separat
im Code
definiert und
kombiniert.



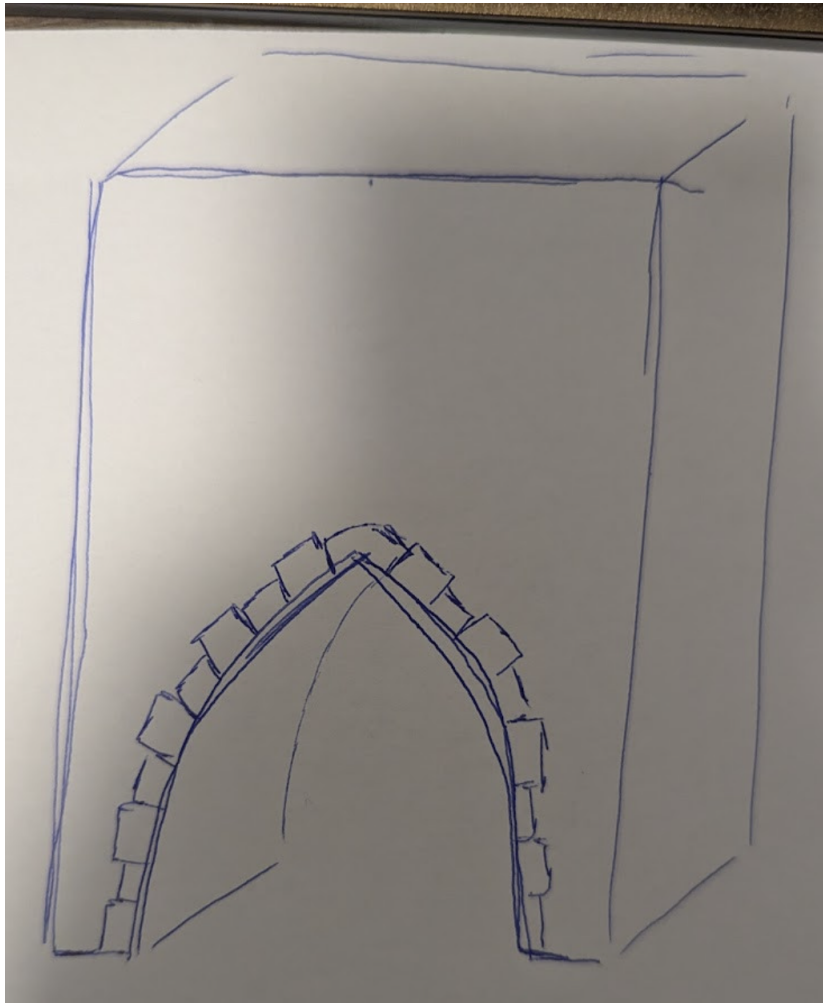
3D-Modellierung: Details im Code



3D-Modellierung: Detail prompts

“Entlang des Tunnels hätte ich gerne an nur einer Seite eine Verzierung. Und zwar sollen das Steine darstellen, die abwechselnd größer und kleiner werden. Oben mittig ist ein Spezialstein. Anbei ein Bild, wie ich es mir etwa vorstelle.”

Details wie Fensteröffnungen, Torbogen und Wandstruktur wurden später hinzugefügt.



Lokal: Ollama und OpenWeb UI

- Ollama ist ein lokal bereitgestellter KI-Modell-Runner, der es Benutzern ermöglicht, große Sprachmodelle (LLMs) direkt auf ihrem PC auszuführen, ohne auf Cloud-Dienste angewiesen zu sein. Es bietet eine benutzerfreundliche Oberfläche und unterstützt verschiedene KI-Modelle, die lokal installiert werden können.
- OpenWeb UI ist eine Open-Source-Webanwendung, die es Benutzern ermöglicht, KI-Modelle über eine benutzerfreundliche Weboberfläche zu nutzen. Es bietet Funktionen wie Textgenerierung, Chatbot-Interaktionen und mehr, und kann sowohl lokal als auch in der Cloud betrieben werden.

Beispiele in OpenWeb UI zeigen.

Lokal: OpenCode

- OpenCode ist eine KI-gestützte Programmierhilfe, die Entwicklern dabei hilft, Code schneller und effizienter zu schreiben. Es bietet Funktionen wie Code-Vervollständigung, Fehlererkennung und Vorschläge für Verbesserungen, um den Entwicklungsprozess zu optimieren.

Beispiele in OpenCode zeigen.

Lokal: ConfyUI

- ConfyUI ist eine KI-gestützte Anwendung zur Bilderzeugung, die es Benutzern ermöglicht, durch die Eingabe von Textbeschreibungen oder anderen Anweisungen automatisch Bilder zu generieren. Es nutzt fortschrittliche KI-Modelle, um kreative und realistische Bilder basierend auf den gegebenen Eingaben zu erstellen.

Cloud: Meshy

- Meshy ist eine KI-gestützte 3D-Modellierungsplattform, die es Benutzern ermöglicht, komplexe 3D-Modelle durch einfache Textbeschreibungen oder andere Anweisungen zu erstellen. Es nutzt fortschrittliche KI-Technologien, um realistische und detaillierte 3D-Modelle zu generieren, die in verschiedenen Anwendungen wie Spieleentwicklung, Animation oder Produktdesign verwendet werden können.

HuggingFace

HuggingFace ist die größte Plattform für offene KI-Modelle — vergleichbar mit GitHub, aber für KI.

Heute zeigen wir fünf Demos aus verschiedenen Bereichen: Bild, Video, Sprache und 3D.

HuggingFace: Die Plattform

- Über **1 Million Modelle** frei verfügbar: Sprache, Bild, Audio, Video
- **Spaces**: Interaktive Demos, direkt im Browser — keine Installation nötig
- Modelle von großen Laboren (Meta, Google, OpenAI) und Community-Entwicklern
- Mit einem **Pro-Account** lassen sich leistungsfähigere GPU-Ressourcen nutzen

huggingface.co
— kostenlose
Basisnutzung,
Pro-Account
für erweiterte
Ressourcen

HuggingFace: FLUX.1 — Text zu Bild

FLUX.1 [schnell] von **Black Forest Labs** erzeugt hochwertige Bilder aus Textbeschreibungen — in wenigen Sekunden.

„*A red panda sitting on a rooftop at sunset, cinematic*” → fertiges Bild

Demo: huggingface.co/spaces/black-forest-labs/FLUX.1-schnell

HuggingFace: Was ist FLUX.1?

- Entwickelt von **Black Forest Labs** — gegründet von den Autoren von Stable Diffusion
- **[schnell]** bedeutet: optimiert für Geschwindigkeit — 1–4 Schritte statt 20–50
- Qualität vergleichbar mit Midjourney oder DALL·E 3
- Architektur: **Diffusion** — Start aus Rauschen, schrittweise Verfeinerung

Eigenschaft	Wert
Typ	Text-to-Image
Schritte	1–4
Lizenz	Offen
Laufzeit	~3–8 s/Bild

HuggingFace: Wie funktioniert Text-to-Image?

Diffusion — vereinfacht in drei Schritten:

1. **Start:** Das Modell beginnt mit einem Bild aus purem Rauschen
2. **Entrauschen:** Schritt für Schritt wird das Rauschen reduziert — geleitet vom Text-Prompt
3. **Ergebnis:** Nach wenigen Schritten entsteht ein kohärentes Bild

FLUX.1 [schnell] erreicht gute Qualität schon in **4 Schritten** — klassische Modelle brauchen 20–50.

HuggingFace: Tipps für gute Prompts

Was funktioniert gut:

- Stil nennen: „*watercolor painting*“, „*photorealistic*“, „*oil on canvas*“
- Licht beschreiben: „*golden hour*“, „*soft studio lighting*“, „*dramatic shadows*“
- Komposition angeben: „*close-up portrait*“, „*wide angle*“, „*bird’s eye view*“

Was FLUX.1 nicht zuverlässig kann:

- Lesbaren Text im Bild erzeugen
- Exakte Personenähnlichkeit reproduzieren

HuggingFace: Wan2.2 — Text zu Video

Wan2.2 ist ein offenes Video-Generierungsmodell von **Wan-AI** — es erzeugt aus einer Textbeschreibung ein kurzes Video.

„A cat walking through a sunlit garden” → fertiges Video

Demo: huggingface.co/spaces/r3gm/wan2-2-fp8da-aoti-preview-2

HuggingFace: Was ist Wan2.2?

- Entwickelt von **Wan-AI** (wan.video), veröffentlicht als offenes Modell
- Modellgröße: **14 Milliarden Parameter**
- Aufgabe: **Text-to-Video** — aus einer Beschreibung wird ein kurzes Video synthetisiert
- Architektur: **Diffusion** — dasselbe Prinzip wie FLUX.1, aber für Videosequenzen

Eigenschaft	Wert
Parameter	14B
Typ	Text-to-Video
Lizenz	Offen
Laufzeit	~30–60 s/Video

HuggingFace: FP8 — Warum läuft das im Browser?

Das Modell hat 14B Parameter — normalerweise ~**28 GB RAM** nötig (FP16).

Die Demo-Version nutzt **FP8-Quantisierung** (8 Bit statt 16 Bit pro Zahl):

- RAM-Bedarf sinkt auf ~**14 GB** — passt auf eine GPU-Instanz
- Qualitätsverlust kaum wahrnehmbar
- **AOTI** (Ahead-of-Time Compilation) beschleunigt die Inferenz zusätzlich

HuggingFace stellt kostenlose GPU-Rechenzeit bereit (*Running on Zero*) — daher kann jeder die Demo gratis nutzen.

HuggingFace: Whisper — Sprache zu Text

Whisper von **OpenAI** wandelt gesprochene Sprache automatisch in Text um — in über 90 Sprachen.

Demo: Mikrophon aufnehmen oder Audio hochladen → Text erscheint automatisch.

Demo: huggingface.co/spaces/hf-audio/whisper-large-v3

HuggingFace: Was kann Whisper?

- **Spracherkennung** in über 90 Sprachen
- **Übersetzung**: direkt ins Englische ohne Zwischenschritt
- Funktioniert mit Mikrophon oder hochgeladenen Audio-Dateien
- Sehr robust gegenüber Dialekten, Akzenten und Hintergrundgeräuschen

Eigenschaft	Wert
Typ	Speech-to-Text
Sprachen	90+
Lizenz	Offen (MIT)
Laufzeit	Echtzeit

HuggingFace: Wo wird Whisper eingesetzt?

- Automatische Untertitel für Videos
- Transkription von Meetings und Interviews
- Barrierefreiheit: Gesprochenes für Hörgeschädigte in Text umwandeln
- Basis für Sprachassistenten und Sprachsteuerung

Whisper ist das „**Ohr**“ der KI — Kokoro TTS ist die „**Stimme**“.

HuggingFace: Kokoro TTS — Text zu Sprache

Kokoro erzeugt aus geschriebenem Text natürlich klingende Sprache — die Umkehrung von Whisper.

Whisper: Sprache → Text · Kokoro: Text → Sprache

Demo: huggingface.co/spaces/hexgrad/Kokoro-TTS

HuggingFace: Was kann Kokoro TTS?

- Über **50 Stimmen** — verschiedene Altersgruppen, Akzente, Geschlechter
- Mehrsprachig: Englisch, Deutsch, Französisch, Japanisch u.v.m.
- Sehr natürlicher Klang — kaum von menschlicher Stimme zu unterscheiden
- Läuft vollständig im Browser, keine Installation nötig

Eigenschaft	Wert
Typ	Text-to-Speech
Stimmen	50+
Lizenz	Offen
Laufzeit	~2-5 s

HuggingFace: Whisper + Kokoro = Sprach-Pipeline

Aus zwei Bausteinen wird ein vollständiger Sprachassistent:

Mikrofon → [Whisper] → Text → [KI-Modell] → Text → [Kokoro] → Lautsprecher

- **Whisper**: Sprache → Text (Eingabe verstehen)
- **KI-Modell**: Text → Text (antworten)
- **Kokoro**: Text → Sprache (vorlesen)

So entstehen Sprachassistenten — aus frei verfügbaren, offenen Bausteinen.

HuggingFace: Pixal3D — Bild zu 3D-Modell

Pixal3D von Tencent ARC wandelt ein einzelnes Foto in ein vollständiges 3D-Modell um — mit Textur, direkt im Browser rotierbar.

Demo: huggingface.co/spaces/TencentARC/Pixal3D

HuggingFace: Was ist Pixal3D?

- Eingang: ein einzelnes Foto (Produkt, Gegenstand, Figur)
- Ausgang: vollständiges 3D-Modell mit Textur als `.glb`-Datei
- Basiert auf **Multi-View Diffusion** — das Modell ergänzt nicht sichtbare Seiten
- Ergebnis direkt im Browser rotierbar und herunterladbar

Eigenschaft	Wert
Typ	Image-to-3D
Eingang	1 Foto
Ausgang	<code>.glb</code> / <code>.obj</code>
Laufzeit	~30–90 s

HuggingFace: Pixal3D vs. OpenSCAD + KI

Zwei Wege zum 3D-Modell — beide mit KI:

OpenSCAD + Claude Code

- Eingang: Textbeschreibung
- Ausgang: parametrischer Code → 3D
- Stärke: präzise Maße, jederzeit änderbar
- Geeignet für: technische Bauteile

Pixal3D

- Eingang: ein Foto
- Ausgang: fotorealistisches 3D-Modell
- Stärke: organische Formen, Texturen
- Geeignet für: kreative Assets, Spieleentwicklung

Microsoft

<https://adoption.microsoft.com/de-de/scenario-library/>

Cloud: Azure Showcases

xxx love xxx

Termine-Webseite

KI-generierte Webseite um den Termine und erklärungen wie die Termine öffentlich zur verfügung gestellt werden.

Plan erstellen

Der Plan ist das Herzstück um die KI in die richtige richtung zu lenken. Es ist wichtig hier so genau wie möglich zu sein, damit die KI auch wirklich das generiert was man möchte.

- Technologie einschränken
- Zielgruppe definieren und Inhaltliche Schwerpunkte setzen
- Alles in Bereiche aufbrechen (header, Footer, Bereiche definieren)
- CI Definieren (Farben, Schriften, Logo, etc.)
- Code Quality und Sicherheit definieren

Reicht ein Plan?

Natürlich reicht ein guter Plan nicht aus. Verfeinerungen folgen.

Prompt-Beispiel: Herzanimation 1/2

- Um der KI zu erklären, welche Animation gerne hätte, wurde auf eine example Seite verlinkt gsap.com (fig. 1)
- folgend wurde gepromptet, dass die Animation mehr wie das eigentliche Herz aussehen soll (fig. 2).

Prompt-Beispiel: Herzanimation 2/2

Claude.md

im Internet manchmal als Token-Fresser gebrandmarkt kann die Claude.md sehr hilfreich sein. Man kann Claude immer wieder sagen ergänze die Claude.md mit unseren letzten erkenntnissen.

Prompt-Beispiel: Archivments

Die Minispiele wurden so angepasst dass gelber text oder anomationen diese an teaser. Es gab nur einen Prompt um den Archivment bereich zu erstellen:

Finde alle klickbaren spiele mit gelben text. Erstelle eine Archivment bereich. Archivments sind zuerst grau.

Referenzen

- openai.com
- ollama.com
- openwebui.com
- opencode.ai
- confyui.com
- meshy.ai
- Azure KI

Spenden

- Erfindergeist Jülich e.V. finanziert sich ausschließlich durch Mitgliedsbeiträge und Spenden. Wir Danken allen Unterstützern, die unsere Arbeit ermöglichen!
- Spenden Konto und PayPal auf unserer Webseite <https://linktree.erfindergeist.org/>



Ende

- Vielen Dank für die Teilnahme!
- [Präsentation Online anschauen](#)
- [Download Präsentation als PDF](#)
- erfindergeist.org
- Haben Sie Fragen oder Anmerkungen?