

philip gerdes

berlin

gerdes.philip@gmail.com

+49 151 28997947

Kurzportfolio

lebenslauf

		Okt 23	BA Kommunikationsdesign
	Dez 22	Praktikum	
	Sep 22	Frontend-Development bei Gorm Labenz Berlin	
	Okt 23	Studium	
	Okt 19	Kommunikationsdesign an der HTW Berlin	
	Jun 19	Auslandsaufenthalt	
	May 19	Einmonatiger Besuch Japans Tokyo	
	Dez 18	Vorpraktikum	
	Aug 18	Design-Studio & Werkstatt Kleiner König Eutin	
Jan 18		Deutsche Post	
Okt 17		Brief- und Paketzustellung Eutin	
Jun 17		Abitur	
		Carl-Maria-von-Weber-Schule Eutin	

Skills

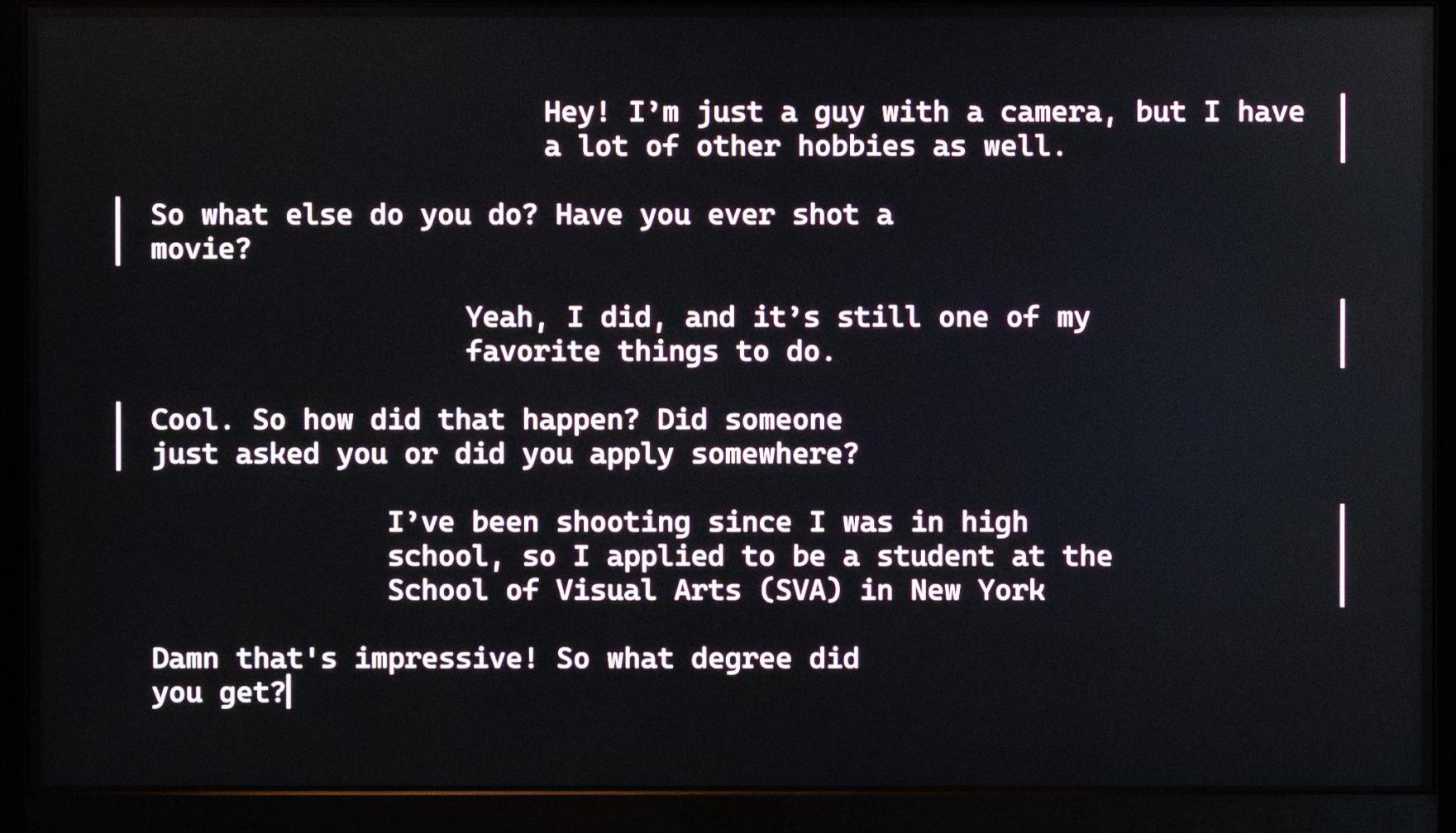
- JS | TS (Electron, Vue)
- Python (HF Transformers)
- C# (Unity Game Engine)
- Adobe Creative Cloud

Interessen

- Generative KI
- Interaktives ML

Sprachen

- Deutsch
- Englisch



Hey! I'm just a guy with a camera, but I have
a lot of other hobbies as well.

So what else do you do? Have you ever shot a
movie?

Yeah, I did, and it's still one of my
favorite things to do.

Cool. So how did that happen? Did someone
just asked you or did you apply somewhere?

I've been shooting since I was in high
school, so I applied to be a student at the
School of Visual Arts (SVA) in New York

Damn that's impressive! So what degree did
you get?



Meine Abschlussarbeit »Inter« implementiert eine Art autonomes interaktives maschinelles Lernen. Ein Klassifikations-Algorithmus wird fortlaufend trainiert und mit jeder Interaktion werden die Trainingsdaten ergänzt.

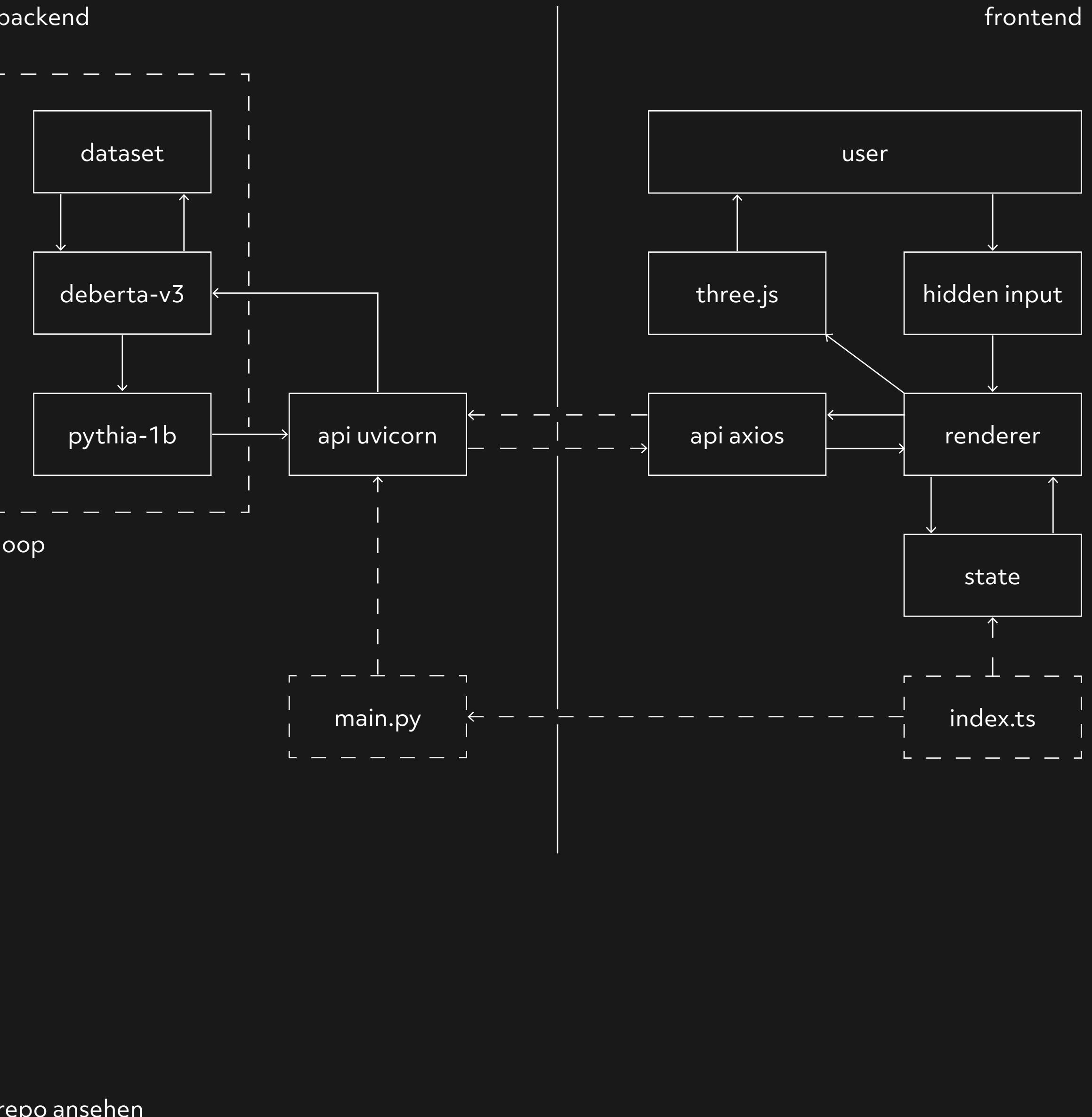
Alle notwendigen Berechnungen, sowohl für das Training, als auch für die Inferenz, werden lokal auf dem vorhandenen Rechner ausgeführt.



Beim Programmstart dient »index.ts« als Einstiegspunkt für das Frontend (Electron App). Dabei wird das Backend als separater Prozess gestartet und ein freier Port als Startparameter übergeben. Im Backend wird dann mithilfe von »FastAPI« ein »Uvicorn« Server gestartet und auf dem angegebenen Port sichtbar gemacht. In der Zwischenzeit versucht eine »Axios« Instanz im Frontend eine Verbindung zu der über den Server verfügbaren API herzustellen. Sobald dies erfolgt ist, übergibt das Frontend dem Backend den Befehl, in den eigentlichen Programm-Loop einzusteigen. Erst jetzt befindet sich die Installation im »idle« Modus und im Backend beginnt das Training der Klassifikation (deBERTa-v3). Währenddessen ruft das Frontend Datenpunkte von vergangenen Interaktionen über die API aus dem Datensatz im Backend ab und zeigt diese an.

Durch Drücken einer beliebigen Taste wird eine Request an das Backend gesendet, die das Training unterbricht. Da Huggingfaces Transformers Library PyTorch nutzt, können die Modelle ohne erneut geladen werden zu müssen in den Inferenz-Modus versetzt werden. Jetzt sehen Nutzer:innen zwar das in »three.js« umgesetzte Interface, interagieren aber eigentlich mit einem versteckten HTML Input Element. Um Eingabe und Display synchron zu halten, laufen alle Daten im Frontend durch den »state«. Sendet jemand nun eine Nachricht, wird diese als Inferenz-Request an das Backend geschickt. Hier klassifiziert »deBERTa« den Input als »truth«, »doubt«, »lie« oder »forlang«. Letzteres wird genutzt, um fremdsprachige Inputs oder sinnlose Eingaben zu filtern. Mit der Klassifizierung wird dann ein neuer Datenpunkt erstellt, welcher anschließend in das »dataset« geschrieben wird. Außerdem dient die Klassifikation als Grundlage für einen Kontextparameter, der dem generativen Modell (pythia-1b) zusammen mit der vorangegangenen Konversation übergeben wird. Dort wird eine Antwort erzeugt, die wiederum über die API an das Frontend übergeben und schließlich angezeigt wird.

Findet für einen gewissen Zeitraum keine Interaktion statt, sendet das Frontend erneut eine Request, welche das Backend wieder in den Trainingsmodus versetzt.



[repo ansehen](#)



foto: Rosi Pernthaller

»Inter« wurde bereits im Rahmen eines Testlaufes auf der Ars Electronica 2023 in Linz ausgestellt.

entity

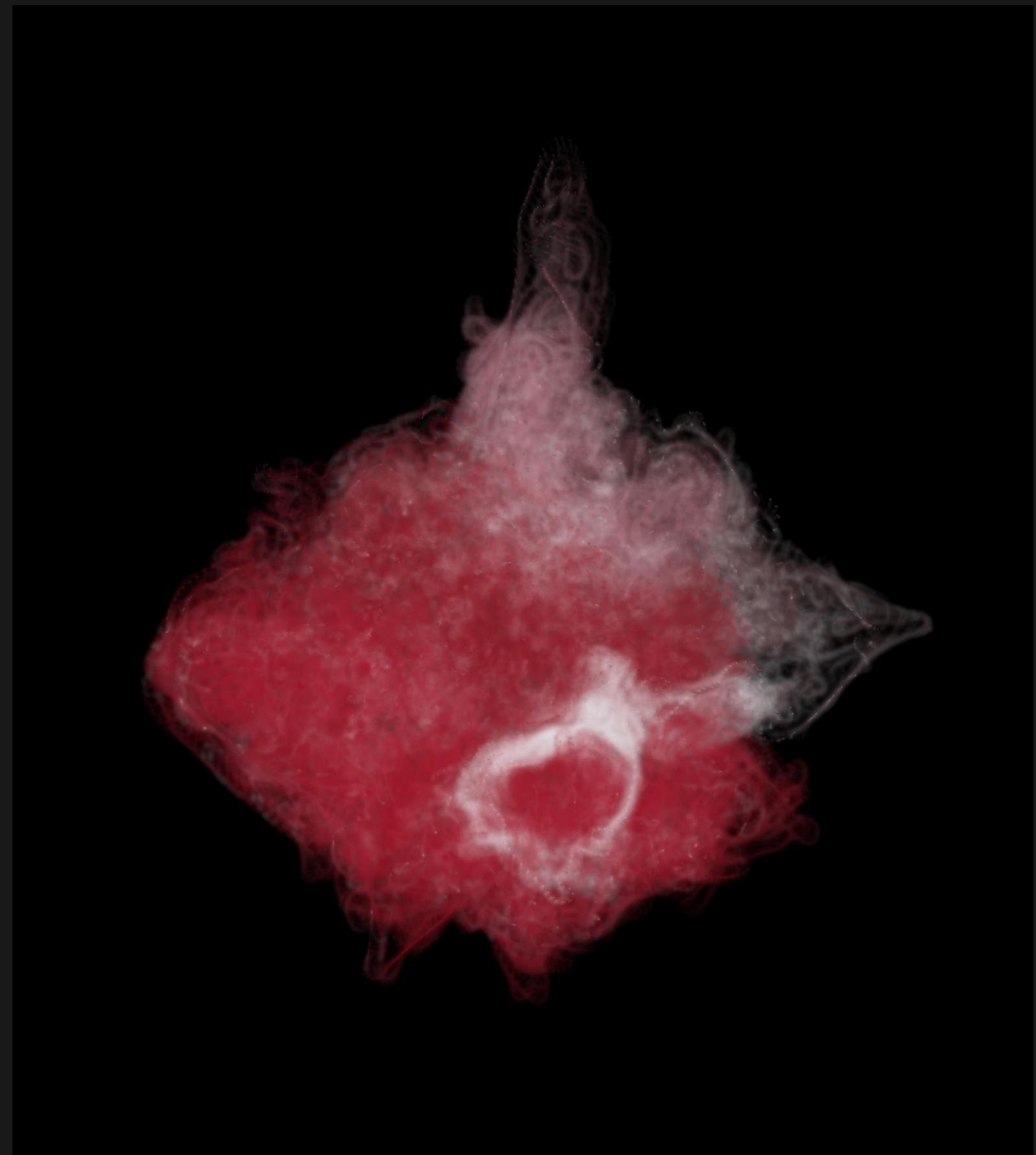
differenzielles wachstum

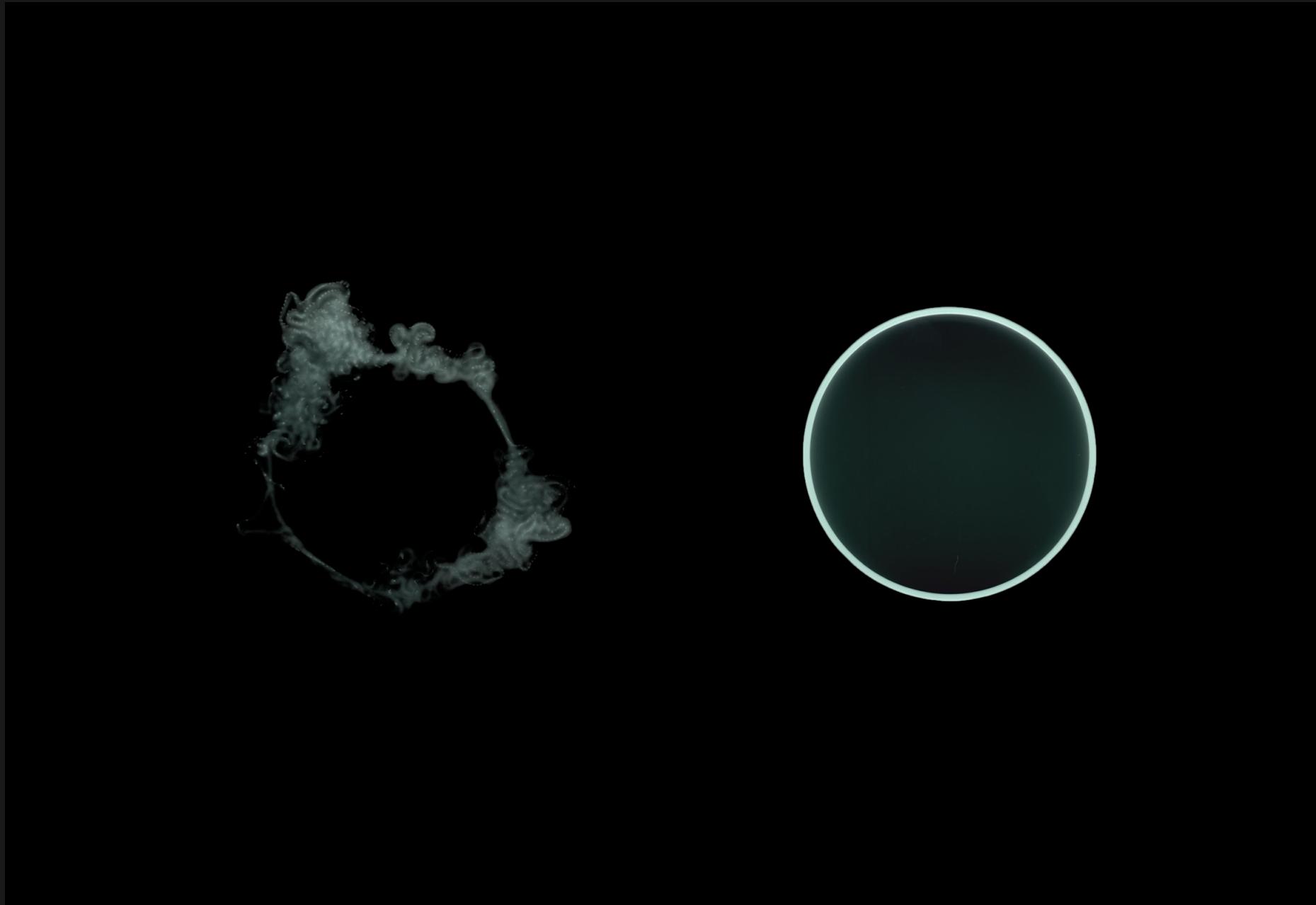


unity

»Entity« greift die von Betrachtenden mitgebrachten Gegenstände als abstrakte Visualisierungen auf. Farbe und Form werden als Grundlage für eine Echtzeit-Wachstumssimulation genutzt. Der Prozess wird von einem Projektor an der Decke sichtbar gemacht.

[video ansehen](#)





Der Wachstumsalgorithmus (Differential Line Growth) ist in drei getrennte Hauptfunktionen unterteilt, die jeweils einen Verschiebungsvektor für jede »Node« ausgeben. Die neue Position wird dann in einen »k-d-Baum« eingespeist, um die Suche nach nahen Positionen effizienter zu machen.

Bei der Analyse der eingescannten Objekte werden dominante Farben mithilfe eines »k-means-Clusters« extrahiert. Das resultierende Farbschema wird anschließend den einzelnen Nodes zugewiesen.

Um eine grobe Silhouette zu ermitteln, werden die Objektgrenzen anhand eines aus dem Durchschnitt des Scans ermittelten Luminanz-Schwellenwertes gesucht. Diese Form bildet anschließend die Grundlage für den Wachstumsprozess.

```
// Update is called once per frame
void Update()
{
    if (analyzeInput.startSim == true)
    {
        InitKDTree(analyzeIn.initPoints);
    }

    if (simRunning == true)
    {
        // Node management loop
        for (int i = 0; i < nodes.Count; i++)
        {
            nodes.Points[i] += AttractionForceOnPoint(i, desiredDistance, attractionForce);
            nodes.Points[i] += RepulsionForceOnPoint(i, repulsionForce);
            nodes.Points[i] += AlignmentForceOnPoint(i, alignmentForce);
            SplitEdges(i);
            //if (pruneNodes == true) PruneNodes(i);
        }
        nodes.Rebuild();
    }

    // Debug area
    if ((debug == true) && (Input.GetKeyDown("space")))
    {
        Debug.Log("KDTree //nodes contains " + nodes.Count + " (" + nodes.Points.Length +
    }

    if (debug == true)
    {
        for (int i = 0; i < nodes.Count; i++)
        {
            Vector3 centerOnTex = new Vector3 (canvasResolution / 2, canvasResolution /2,
            Debug.DrawLine((nodes.Points[i] - centerOnTex) * debugScale, (nodes.Points[(i + 1) % nodes.Count] - centerOnTex) * debugScale, Color.red);
        }
    }
}
```

[repo ansehen](#)

kontakt

Mobil +49 151 28997947
Mail gerdes.philip@gmail.com
Github github.com/whatphilipcodes
LinkedIn linkedin.com/in/gerdes-philip