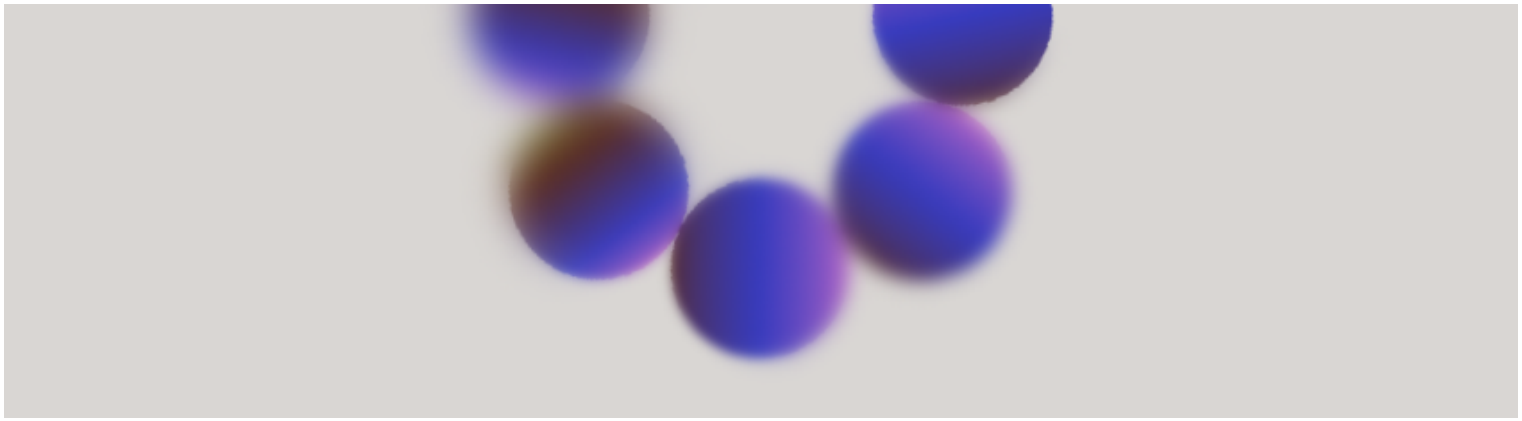

Generative Gestaltung - Semesterprojekt WiSe24/25 - Silas Hering

Installation und Ausführung

Dependencies installieren

```
npm i
```

Projekt ausführen

```
npm run dev
```

Das Project über https und Netzwek laufen lassen. (Um es auf einem Handy zu sehen z.B.)

```
npm run host
```

Zum PDF konvertieren

```
pandoc README.md -o README.pdf --from=gfm -V geometry:a4paper -V geometry:top=8mm -V geometry:b
```

Aufgabestellung

Ihr sucht Eur ein freies Thema, welches dann unter den Euch bekannten Funktionen von p5.js umgesetzt wird. Hier sollte vor allem darauf geachtet werden, dass der Aspekt des Generativen im Projekt sichtbar wird! Notion - Semesterprojekt

Über das Projekt

Das Generative des Projektes lässt sich in folgenden Aspekten wiederfinden:

1. Veränderung der Umgebungslautstärke
2. Deviceausrichtung
3. Displaygröße
4. Interaktion mit dem Display
5. Veränderung des Standorts
6. Neu-laden

Aufgrund dieser veränderbaren Werte wird ein Gradient erzeugt, der abhängig von Umgebungslautstärke oder Interaktion mit dem Display auf Spheren oder Cylindern dargestellt wird. Der Gradient wird vom jeweiligen Standort beeinflusst, sodass die Gradienten mehrerer Devices, die sich am selben Standort befinden, die selbe Farbgebung aufweisen. Ca. alle 10m^2 ändert sich die Farbgebung.

Die Deviceausrichtung beeinflusst eine perspektivische Verschiebung der dargestellten Objekte.

Vorgehensweise

> Die einzelnen Arbeitsschritte können in den Commits des Github Projekts nachvollzogen werden.

Ich begann die Aufgabe mit der Inspirationsfindung und dem Schreiben eines Konzepts Generative Gestaltung - Semesteraufgabe - Konzept - Silas Hering
Im nächsten Schritt beschäftigte ich mich damit, eine Klasse mit dem Namen ”Data” zu programmieren. Mithilfe dieser Klasse können alle Daten, die für das generative Projekten gebraucht werden, initializiert, aktualisiert und abgerufen werden. In der Inizializierung werden z.B. Eventlistener für die Device-Position oder Audio-Input erstellt. Über eine `refresh()` Funktion können alle Daten aktualisiert werden.
Auch gibt es eine ”Blur” Klasse, die für die verschiedenen unscharfen Bereiche zuständig ist, die über den Canvas wandern. In ihr können Position, Intensität und die Größe bestimmt werden.
Mit die meiste Zeit floss in die Überlegung, wie man am besten Gradienten auf den Elementen darstellen kann. Schließlich fand ich eine gute Möglichkeit für die Umsetung darin, erst ein Array aus Farben zu erstellen, dessen Auswahl und Länge an verschiedene Bedingungen geknüpft sind. Z.B. den Standort des Gerätes (`gradientColorArrayGenerator()`). Dann wird der Gradient an die Funktion `createGradientTexture()` Funktion übergeben, die aus dem Gradient eine Textur erstellt, die auf einem 3D Mesh gerendert wird.

Mit mehr Zeit für die Umsätzung des Projektes, hätte ich mich gerne tiefer mit dem Erstellen von Gradienten beschäftigt. Besonders die Erstellung eines Gradienten, der nicht nur aus vielen verschieden fabigen Linien besteht, sondern einzelnen Pixeln, die ihre Position untereinander ”verstehen”, wäre spannend gewesen.
Grundsätzlich bin ich mit dem finalen Ergebnis aber zufrieden.