

# PPI 2022 - Projektarbeit

---

## Auswahl des Projekts

---

- Projekt A: Heatmap

## Aufbau des Programms

---

### 1. Eingabe

- Speicher allocieren (Anhand der ersten beiden Zeilen des Dokuments)
- Eingabe überprüfen auf Sinnhaftigkeit
- Überführen der restlichen Daten in ein `datenpunkt_array` (`data_points[i * WIDTH_DATA + HEIGHT_DATA]` )
  - Speichern im Heap
  - z-Werte werden gespeichert, x und y durch Position im Array dargestellt (ohne Schrittweite)
  - Größter und Kleinster z-Wert wird beim Einlesen getrackt
- `argv[]` Zielgröße in `WIDTH_MAP` und `HEIGHT_MAP` überführen
- Belegung des Speichers für die `z_map[]`
  - mit `[WIDTH_MAP], [HEIGHT_MAP]`

### 2. Verarbeitung

- Funktion: `color_cluster(z_min, z_max, *color_starting_values[])`
  - mathematische Ermittlung der Unterteilung der 10 Farben
  - Funktion erhält einen Zeiger auf ein Array, in das von der Funktion die Startwerte für die 10 Farben geschrieben werden

- Funktion: `interpolation(* z_map[])`
  - überführt `datenpunkte[]` in interpolierte `z_map[]`
    - Falls die Anzahl der Datenpunkte mit der gewünschten Pixelgröße der Ausgabe übereinstimmt, werden die Z-Werte nur in das neue Array kopiert
    - Falls die Anzahl der Datenpunkte geringer ist, als die gewünschte Größe der Ausgabe in Pixeln, werden neue Z-Werte ermittelt, die gemeinsam mit den Messdaten in das neue Array ( `z_map[]` ) kopiert werden
  
- Funktion: `colore_value(* z_map[], * color_starting_values[])`
  - Diese Funktion ermittelt anhand der `color_starting_values[]` die Farben der einzelnen Pixel in `z_map[]`
  - Die ermittelten Werte werden für die `bmp_create` -Funktion passend verpackt

### 3. Ausgabe

- Aufruf der `bmp_create` -Funktion

#### Aufteilung

Paul	Nico
Eingabe	color-cluster Funktion
color-value	interpolationFunktion