Data Physicalisation for Climate Change Data Using an Interactive Globe: Exploration of Input Modes for Time Series Data

BSc Creative Technology
Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS)

María Cobo Muñoz

First supervisor: dr. Champika Epa Ranasinghe
Second supervisor: Auriol Degbelo
External supervisors: dr. Yuri von Engelhardt and dr. Paulo Raposo
Critical observer: Nacir Bouali

July 2024

UNIVERSITY OF TWENTE.

Abstract

Climate change is one of humanity's most pressing challenges, evidenced by increasingly frequent and severe extreme weather events, rising global temperatures, and widespread impacts on ecosystems. To address these challenges, innovative solutions and global collaboration are essential for improving our lives and those of future generations.

One promising approach to enhancing public understanding of climate change is data physicalisation, an emerging discipline that conveys complex information engagingly and impactfully by incorporating multiple senses through embodied interaction. In this context, globes are the physical representation of geographical data, and they offer a more accurate representation of our planet, capturing its true curvature and spatial relationships and enabling embodied interactions such as rotation, pointing, and touching.

To explore these advantages, an interactive LED globe was designed, built, and tested through an iterative design process. Temperature anomalies from 1980 to 2020 were displayed using different output modes, such as colour and animation. The globe featured three different input modes, slider, discrete knob, and continuous knob, each representing a different metaphor for year selection and time series data input.

This research investigates the impact of these different input modes on user engagement and comprehension when interacting with the globe installation. By incorporating embodied interactions through year selection and globe rotation as natural movements the aim was to create a more engaging and memorable experience, thereby raising awareness of climate change. In the test, 36 participants evaluated the user experience, mental and physical load, satisfaction, accuracy, and efficiency of the system. While no significant differences in performance between the input modes were found, users preferred the discrete knob with clicking sounds and haptic feedback. Overall, the installation successfully engaged users and sparked curiosity, receiving positive feedback.

Acknowledgements

First, I would like to share my gratitude with my supervisor Champika Epa Ranasinghe, for her support and guidance throughout the entire process of this thesis. From the beginning, she made time in her very busy schedule to facilitate this project. Thank you for your involvement and constant enthusiasm, they were great motivators in this journey.

Secondly, thanks to Auriol Degebelo, who, as a second supervisor, joined our meetings and provided very valuable insights into how to set up a compelling user study. Thank you for your guidance when we were lost in that matter and for your commitment.

Special thanks to Luka, my co-researcher and great friend. This would have literally not been possible without you, and I am very proud of the result and the great team we make. Thanks for the moral support, valuable contributions, all those coding hours, and it was just a pleasure doing my thesis with you. You made it seem easy.

I also must thank Yuri von Engelhardt and Paulo Raposo for their valuable feedback early on the project and their availability and willingness to share their knowledge and advice.

Not to forget to thank Alfred de Vries, who provided materials and tools just when needed, and many good tips on wire connections and electronic circuits with the LEDs. Thank you, Joris Köster and Jorge Davó, for checking them and ensuring things lit up but did not light on fire. Thank you, Pablo Núñez Martín, for your input on Blender and 3D modelling of the continents.

Of course, big thanks to all 36 participants who gave us very positive feedback and greatly contributed to the project by gifting us some of their time, most of them during their busy thesis days.

Finally, I can't end this without giving a heartfelt thank you to my friends. Special thanks to Suzy, who has been a cornerstone in my academic trajectory and university experience. Seeing her go through her own thesis experience motivated me to be better, and I only have admiration for you and your academic skills. And lastly, my family back in Mogro, my parents who have always supported me in my education and the leap to study abroad, even when it was hard for them to see their daughter go far away. Thank you for being the best parents I could ask for and for giving me so many opportunities in life. I hope to continue to make you proud.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

Data physicalisation is an emerging discipline which focuses on using physical representations to communicate data. It has shown potential for conveying complex information in an engaging and impactful manner [1]. Unlike its counterpart, "data visualization", physicalisation consist of interactive, tangible installations, through which the target audience can grasp nuances of the data set in ways that graphs are simply limited, by bringing more senses into the cognitive process through embodied interaction [2]. Given this potential, applying data physicalisation can be particularly useful to understand topics more easily such as climate change. General awareness is the first step towards developing concern, enabling people to take action and make the urgent changes required [3].

In this regard, globes are essentially different in how we perceive and engage with geographical information compared to maps. While flat maps have long been the conventional mode of representing Earth's surface, it has its limitations. By contrast, a globe offers a more faithful representation of our planet, capturing its true curvature and spatial relationships. Moreover, the introduction of globe installations brings forth a novel dimension to data exploration, enabling embodied interactions such as rotation, pointing, and touching [4]. These interactions have the potential to make data exploration a more intuitive and immersive experience. The direct, natural interactions with a physical object differ from the mediated interactions of a mouse, keyboard, or touch screen, enhancing user engagement and understanding. Despite their advantages, it's important to note that globes typically offer less detail compared to flat maps, making them better suited for illustrating general trends rather than precise geographical features [5]. While interactive maps have been widely developed in digital formats, the concept of physical interactive globes remains relatively novel, yet promising in its potential to promote engagement, enhance education through experiential learning and foster collaborative exploration of climate change data.

Climate change is one of the most pressing challenges of humanity today. It is manifesting through the increasingly more frequent and severe extreme weather events, rising global temperatures, and the impacts on ecosystems worldwide. Not only it has an environmental impact, but also social, affecting food security, health, and global economies, disproportionately impacting the most vulnerable populations [6]. "There is no planet B" and as the scientific consensus on the human causes of climate change solidifies, it is essential to look for innovative solutions, and global collaboration to improve our lives and the future generations'.

The primary objective of this project is to explore how time series data, a key characteristic of climate change data, can be effectively realized in data physicalisation. To achieve this, an interactive physical globe is designed, constructed, and tested as an innovative medium for communicating climate change issues in an accessible and engaging way for the general public. The data is mapped onto the globe using an LED grid, displaying temperature anomalies over time sourced from Copernicus datasets. These anomalies are visualized through changes in brightness and colour, enabling users to explore historical data interactively. My role in this collaborative project focuses on exploring time series data, its characteristics, and the meaningful interactions it allows for in data physicalisations. Additionally, I will examine how to sort the data in a way that makes it perceivable by the user, exploring various input methods such as sliders and knobs, and assessing their impact on user engagement, perception, and comprehension of the climate data presented.

Therefore, the research question is: *"What is the impact of different input modes for time series data on user engagement and comprehension in of climate change data on an interactive LED globe?"*

Chapter 2 – Background Research

2.1 Data physicalisation overview

In the field of data physicalisation, there seems to be consensus around the definition given by Jansen et al., which describes data physicalisation as "a physical artifact whose geometry or material properties encode data" [1]. These artifacts offer metaphors for expressing and experiencing data, which can potentially lead to new insights and emotional responses [7]. Jansen et al. further elaborates stating that physicalisation can help cognition, communication, learning, problem solving, and decision making [1].

2.2 Variables in data physicalisation and their effectiveness

The variables used in data physicalisation are generally used under a commonly accepted definition. Hornecker et al. defines as "explicit variables" the elements that display data with one-on-one mappings, these are usually also named "modalities" interchangeably [8]. The main categories match with the five senses: visual, haptic, sound, taste, and smell; and the first three have their own list of more concrete manifestation cases. For example, the visual category includes variables such as position, size, or colour, while the haptic category has temperature, vibration, and texture, among many others. Ranasinghe et al. [2] adopt a similar classification and extend it by including physical (materiality) and dynamic variables.

The efficacy of each modality in data physicalisation, as Hornecker claims, depends on its capacity to foster user collaboration and comprehension of abstract concepts [9]. Specific comparisons are outlined by Jansen et al. [1] who suggest that length as a visual variable conveys quantitative values, while colour hue cannot. Alexander et al. [7] highlight the particular effectiveness of gestures for conveying dynamic information. Similarly, Hogan and Hornecker [10], note that while people are used to visual representations such as graphs, these often fail to trigger emotional engagement as haptic and auditory cues do, which encourage more emotive language and a deeper concern for the implications of the data. Supporting this, Jansen et al. [11] point out the key role of physical touch as a cognitive aid, while physically manipulating data representations seem less crucial to the engagement with data. Finally, regarding smell and taste, Hogan and Hornecker [12] warn about their potential limitations in conveying precise data insights but propose that integrating these with more familiar sensory variables can offer an holistic sensorial experience with data. Which aligns with the suggestion by Alexander et al. [7] about how some representations may not always be the most efficient for specific cases, but have the potential to improve user engagement by emotions or simply enjoyment.

Specially interesting is the case of combination of modalities. The integration of multiple modalities in data physicalisation aims to enrich user engagement and understanding of data, yet not all combinations are beneficial. For instance, Alexander et al. [7] indicate that tactile information can conflict with visual information, complicating the user's ability to process data effectively. Similarly, Hogan and Hornecker [9] warn about the use of multiple modalities to represent the same data without adding new information, as this can lead to redundancy and increase cognitive load.

In conclusion, while visual perception is often highlighted as an effective modality, there remains a significant research gap regarding the impact of other modalities, whether used individually or in combination.

2.2.1 Examples of data physicalisation using different modalities/materials

Many modalities have already been explored in the past on data physicalisation. These are varied and make use of different modalities and materials to encode data in a tangible form, enhancing user interaction and understanding.

One innovative approach involves using electromagnetic field strength and friction. Dullaert et al. (2024) [13] investigated this by creating a system where resistance is realized using an electromagnetic field, with the field's strength proportional to the data. Friction is achieved through a rotating wheel with a rough surface, where the motor speed is proportional to the data, allowing users to feel data changes through physical resistance and friction.

Temperature and vibration are also modalities for data physicalisation. Van Loenhout et al. (2022) [14] demonstrated this in their research on physicalizing sustainable development goals (SDGs) data. They mapped datasets to vibration frequency and temperature, where higher data values correspond to higher vibration frequencies and warmer temperatures.

Light variables offer a versatile medium for representing data. De Kreij et al. (2024) [15] used light to engage children with air quality data. In their study, LEDs on a globe represented countries with specific air quality levels, lighting up accordingly. Additionally, they explored using marbles to fill and empty a tub, where the weight of the marbles corresponded to data values.

Similarly, Peeters et al. (2023) [16] utilized light variables in their EmoClock project, which communicated real-time emotional states through data physicalisation. Bio signals sensing heart rate and galvanic skin response were timestamped and translated into LED stripes. The LEDs displayed arousal and valence levels by changing colours according to a gradient on the clock- shaped installation.

Air flow and fan speed can also encode data physically. Houben et al. (2016) [17] explored this in their Physikit project, which involved physical ambient visualizations in the home. They visualized data through airflows produced by small and large fans. Continuous data was represented by varying the intensity of the fans, while relative changes were indicated by turning different fans on or off. When data values were stable, no fans were activated.

Surface roughness offers another tactile method for data physicalisation. Du et al. (2024) [18] investigated how varying surface textures could encode data points. They controlled the perceived level of roughness by adjusting dot diameter on a surface. Higher dot diameters corresponded to lower roughness levels. They tested mapping data values to surface roughness in both positive and negative correlations.

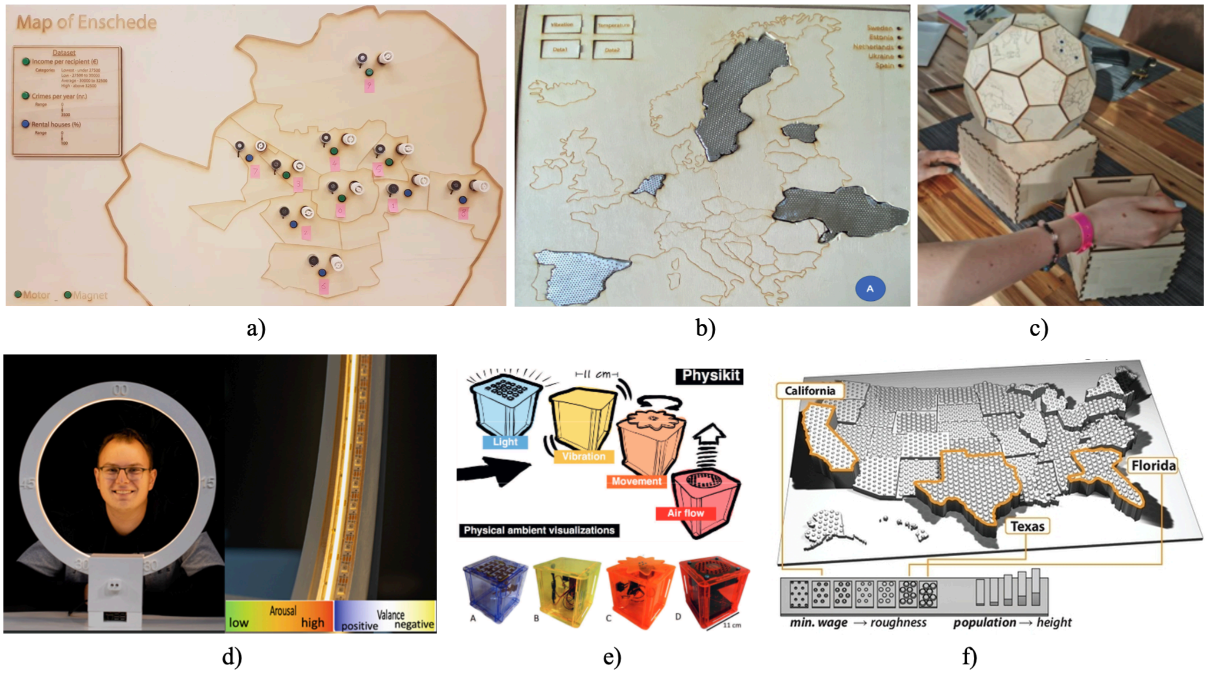

Figure 1 ***Examples of data physicalisations with different modalities to encode data.*** *a) Electromagnetic strength and friction. b) Temperature and vibration. c) Light and weight. d) Light hue. e) Air flow. d) Roughness.*

These examples illustrate the diverse range of modalities and materials used in data physicalisation. From haptic feedback and temperature changes to light variables, air flow, and surface textures, these approaches offer innovative ways to represent data physically.

2.3 Related work

In the realm of data visualization, particularly regarding climate change, there are a diverse ways of visualizing and physicalizing this data. These span from digital maps and globes to immersive VR experiences and physical interactive displays. This section explores the state of the art on data physicalisation of climate change, interactive visualizations, and interactive globes.

2.3.1 Online interactive visualizations

Many online platforms offer interactive visualizations of climate change data, of which some use maps or 3D globes to present information. Educational sites, such as those provided by NASA [19], [20] and the European Space Agency (ESA) [21], are some examples. These resources allow users to explore data over various years, zoom in on specific regions, pan, switch between different datasets, and apply filters to customize the view. They can be updated in real time and provide great amount of detail. Nevertheless, some visualizations can get complicated and might need specific technical knowledge, so they are not as accessible to everyone. They are also not tangible, and the interactions are mediated by a mouse, touch screen or keyboard.

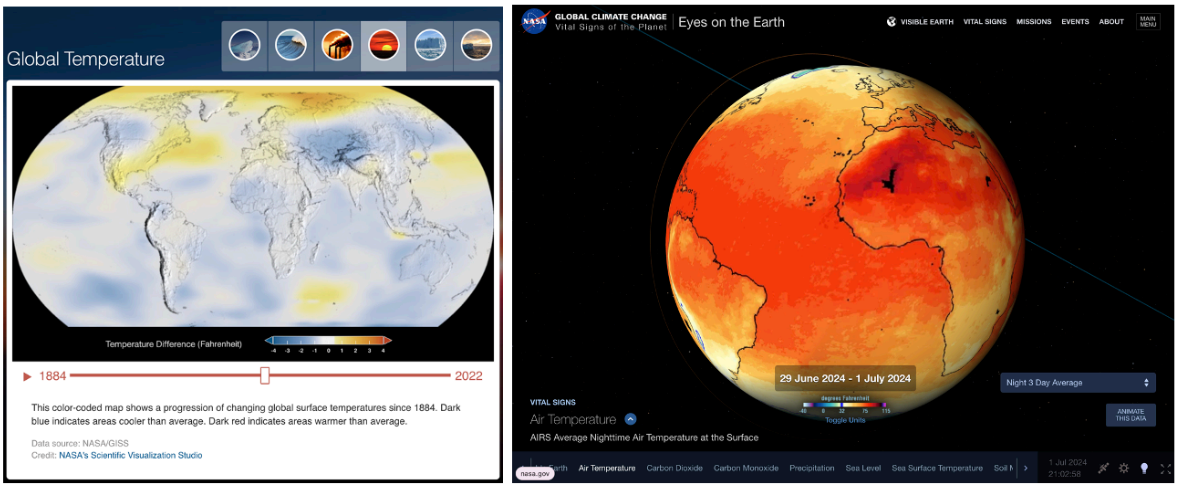

Figure 2 Online interactive visualizations of global temperature data by NASA

2.3.2 Educational interactive toy globes

Figure 3 Educational interactive toy globe with audio feedback

Another related category includes educational interactive toy globes, which are typically designed to teach geography. These popular globes often come with a pen that users can use to point at different locations. When a location is pointed at, the globe plays audio that provides the name of the location and interesting facts about it, even features some games [22]. While these globes are not specific to climate change data, they do offer interactive, tactile, and multimodal ways to engage with geographical information. The globe itself is static, but it can be rotated and manipulated directly.

2.3.3 VR experiences

Virtual reality (VR) offers an immersive way to experience data visualizations, including those related to climate change. The 360 Degree Climate Dome by Garmantis [23] is an excellent example of how VR can provide a multi-sensory experience. It was designed to educate users about climate change by immersing them in a virtual environment that showcases the impacts and data of climate change. This installation not only visualizes climate data but also incorporates olfactory, tactile, and auditory elements, making the experience highly engaging and memorable.

Figure 4 360 Degree Climate Dome VR experience

2.3.4 Data physicalisation installations

Figure 5 Weather Report data physicalisation installation

Data physicalisation involves representing data through physical means, and several installations have tackled climate change data using this approach. One example is the Weather Report installation [24], an art piece that combines human experiences with scientific data physicalisation to visualize climate change. It uses over 800 balloons as physical pixels to display weather data, providing a tactile and interactive experience. Its purpose is to make complex climate data more accessible and understandable to non-scientists by bridging the gap between objective data and subjective human experiences. Visitors can interact with the installation by touching and walking through the balloon walls, entering their weather-related memories at a multi-touch kiosk, and creating animations that represent their experiences. Additionally, they can engage with the projection beams, casting shadows and becoming part of the visual display. This interactive setup not only educates the public about climate change but also fosters a deeper connection and understanding of the data through personal and embodied engagement.

2.3.5 Spherical displays

Figure 6 GeoDome spherical display in a museum showcasing climate change data visualization

Interactive globes manufactured by companies such as The Elumenati represent another variant of data visualization. The GeoDome [25], is an example of a high-resolution spherical display that is are visually stunning and is often found in museums. They are interactive, though typically manipulated through devices like Kinect of a tablet rather than direct physical manipulation. These globes offer a very high resolution, almost like on a 2d screens. However, they are quite expensive, with costs ranging between $40,000 and $150,000, making them accessible primarily to institutions rather than individual consumers.

In summary, the landscape of climate change visualization is rich and varied, encompassing online interactive maps and globes, educational toy globes, immersive VR experiences, data physicalisation installations, and high-resolution spherical displays. Each of these approaches offers unique advantages and engages users in different ways. The current project aims to build on these foundations by creating an interactive globe that not only visualizes climate change data but also allows for direct physical interaction, enhancing both educational value, user engagement and memorability.

Chapter 3 – Methodology

The methodology followed in this project is based on the Creative Technology design process. This process guides designers through a series of phases in which exercises of diverging and converging ideas that fosters lateral thinking and new perspectives. It starts with a design question that kicks off the Ideation phase, followed by Specification, Realisation and Evaluation (see figure 1). Moreover, this process incorporates continuous feedback loops between phases to reassess and refine the designed concept based on new information acquired in past phases. [26].

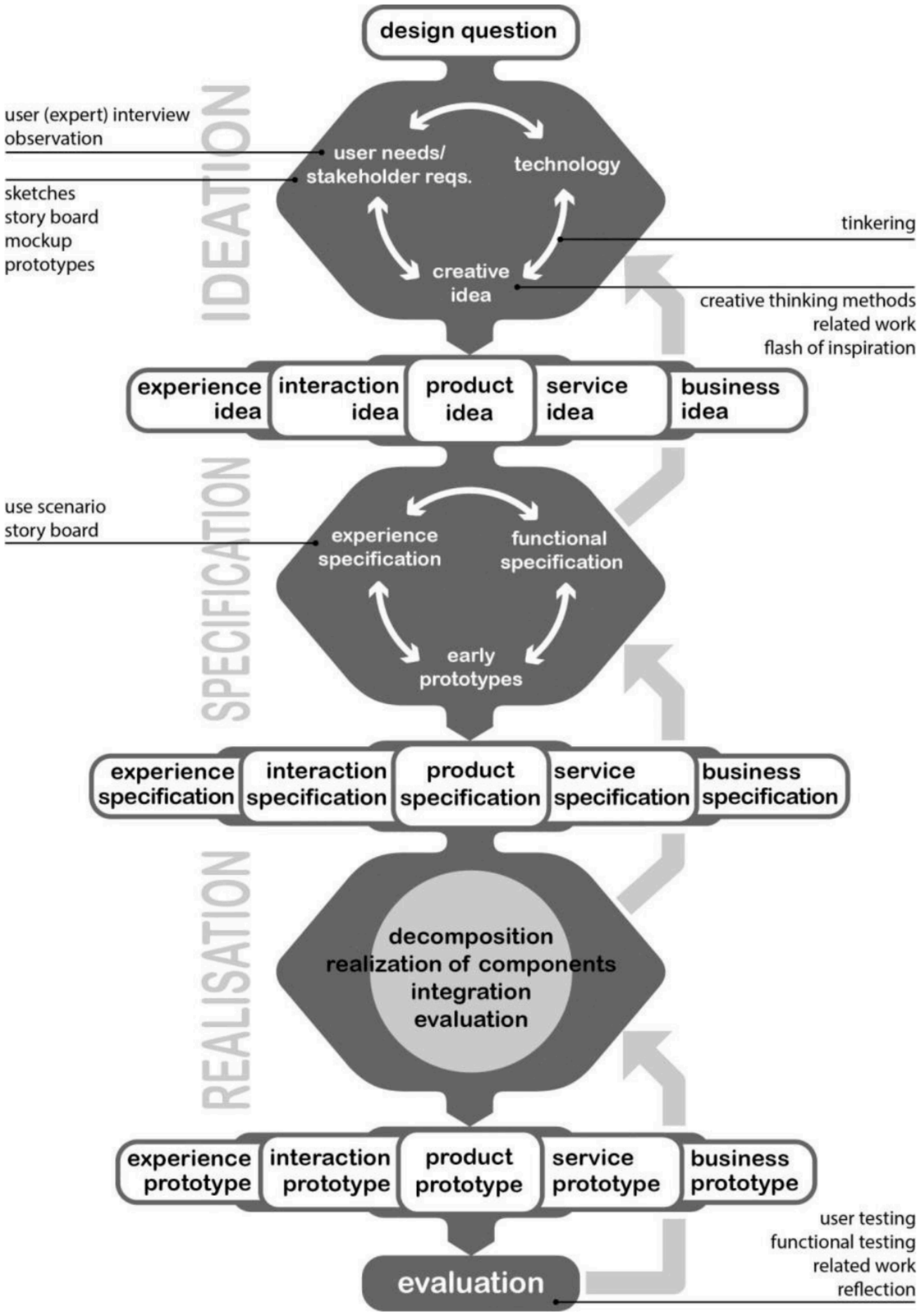

Figure 7 Creative Technology Design Process [26]

3.1 Ideation

This phase begins with a design question: How can we create a data physicalisation of climate change data using an interactive globe? This prompts further ones, such as which data to represent, what interactions are allowed, and how to build the globe. To address these questions, several brainstorming sessions are held with the researchers and discussed with the supervisor to develop a clearer project concept. Additionally, we consulted with people knowledgeable in data visualization and geographical data to get a better understanding and approach. Between these sessions, inspiration is drawn from state-of-the-art examples and relevant literature in the field of data physicalisation.

3.2 Specification

Building on the concept from the previous phase, the Specification phase focuses on defining clear requirements and testing the functionality, feasibility, and user experience of various components of the installation. Rapid prototyping and testing of system parts are crucial to validate or eliminate potential project directions. This phase ends with a well-defined design concept, informed by testing, literature, and the researchers' experience.

3.3 Realisation

In the Realization phase, the project is executed according to the design and requirements established in previous phases. The construction and subsequent integration of components are carried out systematically and in parallel, involving two researchers and conducting tests at every step. Task division was key for efficient workflow and easier troubleshooting. While in this phase unexpected issues may occur, improvisation is essential to identify workable alternatives without compromising the overall concept and its requirements.

3.4 Evaluation

Finally, the process concludes with a user evaluation where users unfamiliar with the installation interact with it to provide unbiased insights. While functionality testing was conducted in the previous phase, this evaluation focuses on other aspects of the system. The assessment checks for efficiency, effectiveness, mental and physical demands, user experience (UX), usability, and overall satisfaction. This stage is crucial for critically examining the design, ensuring that objectives were met, and obtaining research findings for future developments.

Chapter 4 – Ideation

4.1 Initial requirements

At the beginning of this phase, we considered several preliminary requirements that had been set beforehand as essential for the project. These requirements included the following:

- The physicalisation should represent climate change.
- The physicalisation should have physical input types for changing the year or modality.
- The physicalisation should be a globe and thus not be a flat map.
- The globe should be able to rotate.
- The physicalisation should facilitate different output modes.
- The physicalisation should be intuitive for the users.
- The physicalisation should not have significant delays that influence the user experience.

4.2 Brainstorming session

Considering the preliminary requirements, an initial brainstorming session was held between the researchers and the supervisor to explore the potential features of the installation. During this session, a mind map was created to outline various possibilities, including shapes, interactions, modalities, and data, as summarized in Figure 8.

The focus of this brainstorming session was on generating a wide range of ideas rather than considering feasibility. Various ways of representing data were explored, such as vertical versus horizontal stripes. Another key requirement that emerged was the need for modularity.

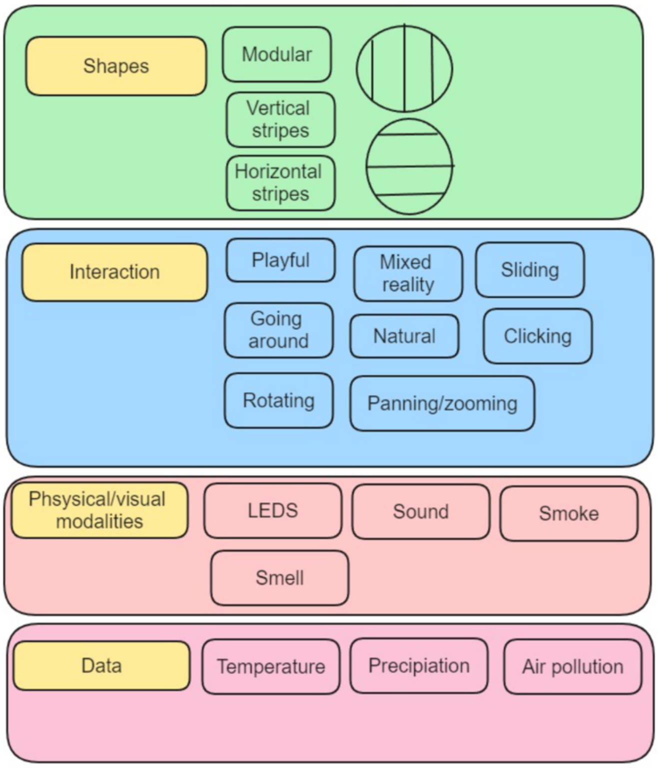

Figure 8 Brainstorm mind map

Several interactions were identified, including rotating, navigating, clicking, and zooming on the globe. The potential integration of mixed reality was also considered, with an emphasis on ensuring that the interactions would be playful and intuitive.

The modalities considered were LEDs (visual), sound and smoke (visual and smell). Lastly, we selected the most relevant data from the WMO's list of state-of-the-climate indicators for our globe: surface temperature, precipitation, and air pollution [27].

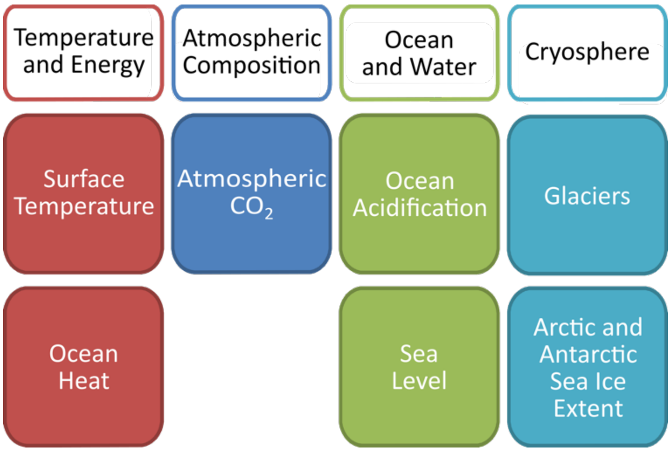

Figure 9 7 state-of-the-climate indicators by WMO [27]

4.3 Inspiration

4.3.1 Spherical displays

After the brainstorming session, the next step was to design an initial concept for the globe. To begin, inspiration was taken from existing LED globes, even not interactive ones. The aim was to study previous constructions and control methods used for the animations. This phase was crucial for determining what was achievable within the project's scope. The primary resource for this exploration was YouTube tutorials.

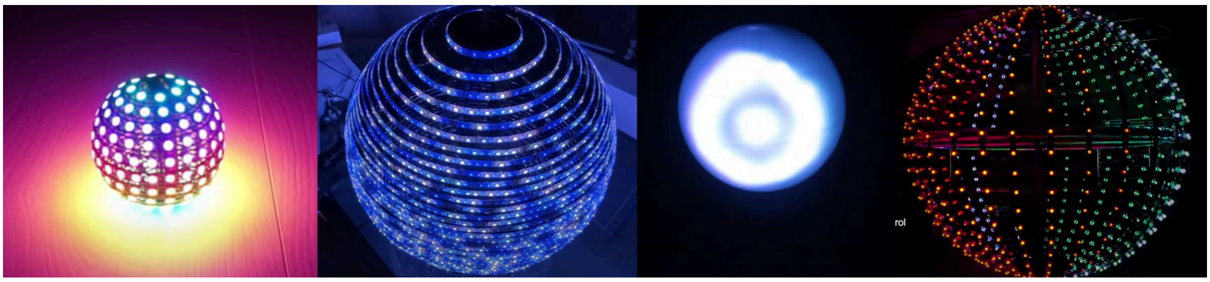

Figure 10 LED globes inspiration

Figure 10 showcases four examples of LED globes. From left to right: the first is a small ball with individually soldered LEDs, approximately 15 cm in diameter, housing batteries and controllers internally for mobility and robustness [28]. However, due to our project's size requirements and feasibility concerns with soldering numerous LEDs, this option was dismissed.

The second example adopts a latitude-based approach, employing LED strips affixed to circular forms, achieving a diameter of about 50 cm [29]. This method retains the integrity of LED strips, significantly reducing manual labour complexity.

The third example, also utilizes LED strips but connects small sections to achieve spherical curvature, contrasting with the parallel orientation of strips in the previous design. This globe, also 50 cm in diameter, incorporates a diffusing material over the LEDs for a smoother, more touchable appearance, hiding electronic components for aesthetics and safety [30]. While this design offered an optimal aesthetic and functional design, the additional components and complex soldering made it too expensive and complex.

The final example is a meridian-based design with a diameter of 180 cm, notable for its lower LED density compared to the latitude-based design [31]. This variation raises concerns about accuracy, as LEDs in the centre represent larger areas than those at higher latitudes, potentially confusing users. Additionally, this design integrates proprietary commercial software.

Of these examples, the latitude-based design (second from the left) emerged as the preferred inspiration due to its simplicity, accuracy, and feasibility for our project's goals.

4.3.2 Physical input modes

In the exploration of physical data input methods, the focus was on ways to interact with time series data, particularly for inputting years and selecting datasets. This research provided various possibilities.

Individual buttons for each year or dataset region were a possible method of selection. These buttons can be customized in shape and appearance to be more visually attractive. However, the practicality of this approach for year selection is limited, as it would require a large number of buttons. On the other hand, increment and decrement buttons offer a simpler solution, where users can increase or decrease the year value. This method is commonly seen in devices like alarm clocks and Copernicus uses it in their Climate Pulse [32], but it is not optimal for year selection as it might seem obstructive. However, buttons could be useful for switching between datasets.

Linear sliders provide a more intuitive and visually appealing way to select years. Users can drag the slider along a track to choose a year within a specified range. This method is commonly used in both physical and digital interfaces, especially in climate change representations. Online versions often allow for the selection of a year interval by setting minimum and maximum values, a feature not typically available in commercial physical sliders.

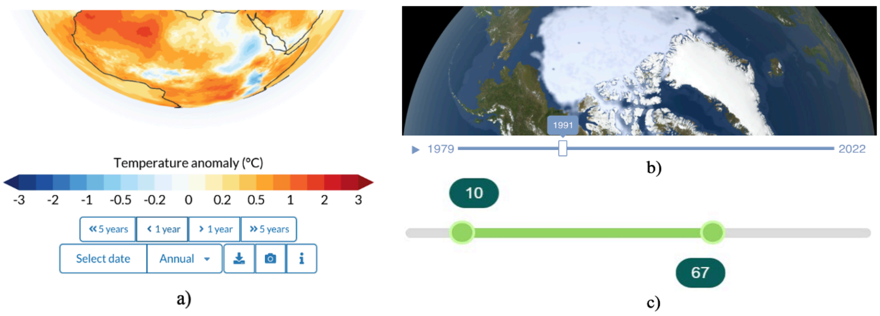

Figure 11 Year selection methods: a) Climate Pulse uses increment buttons. b)NASA visualizes climate change with a slider. c)Online sliders have the possibility of selecting ranges.

Knobs, including potentiometers and rotary encoders, offer another tactile method for year selection. Potentiometers present limits and a smooth rotation while rotary encoders can rotate infinitely and even provide tactile and auditory feedback through slight clicks. Rotary switches are also an option, with multiple positions corresponding to different years or ranges of years. These are commonly used in appliances like washing machines and microwave ovens. They provide a satisfying interaction by stronger clicks and defined positions with limits for min and max. In all cases, the selected year can be displayed either on a screen or marked on the surface.

RFID and NFC tags present another approach. For example, systems like the one used in the game Disney Infinity, works by placing a figurine on a pedestal to select characters. The same could be done to identify and select the dataset. Alternatively, LCR sensors can be used for this same interaction, to detect which figurine is out of its rest position and thus, on the pedestal.

Figure 12 Disney Infinity selection figurines

Touch interfaces, including touchpads and touchscreens, can be used to input data. Users can swipe or tap to select a year. Capacitive touch buttons and touch sensors with movement recognition can transform thin surfaces into interactive panels. This is a clean and innovative input method but less embodied that the ones above.

Figure 13 Capacitive touch sensor

Scroll wheels, like those found on computer mice, could provide a fun and less common way to select years. However, they are less practical as they do not provide a direct visual indication of the selected year.

Each of these physical inputs offers distinct ways to make year selection and dataset interaction user-friendly, depending on the specific context and user needs. In the case of this project, sliders, knobs, and figurines were selected moving forward as they prove to be straight forward, engaging and of easy implementation.

4.4 First sketches

Once insights into the possibilities of LED globes and input modes were gathered, preliminary sketches were created to visualize initial design concepts. Early ideas cantered on different methods of displaying data and various types of interactive interfaces.

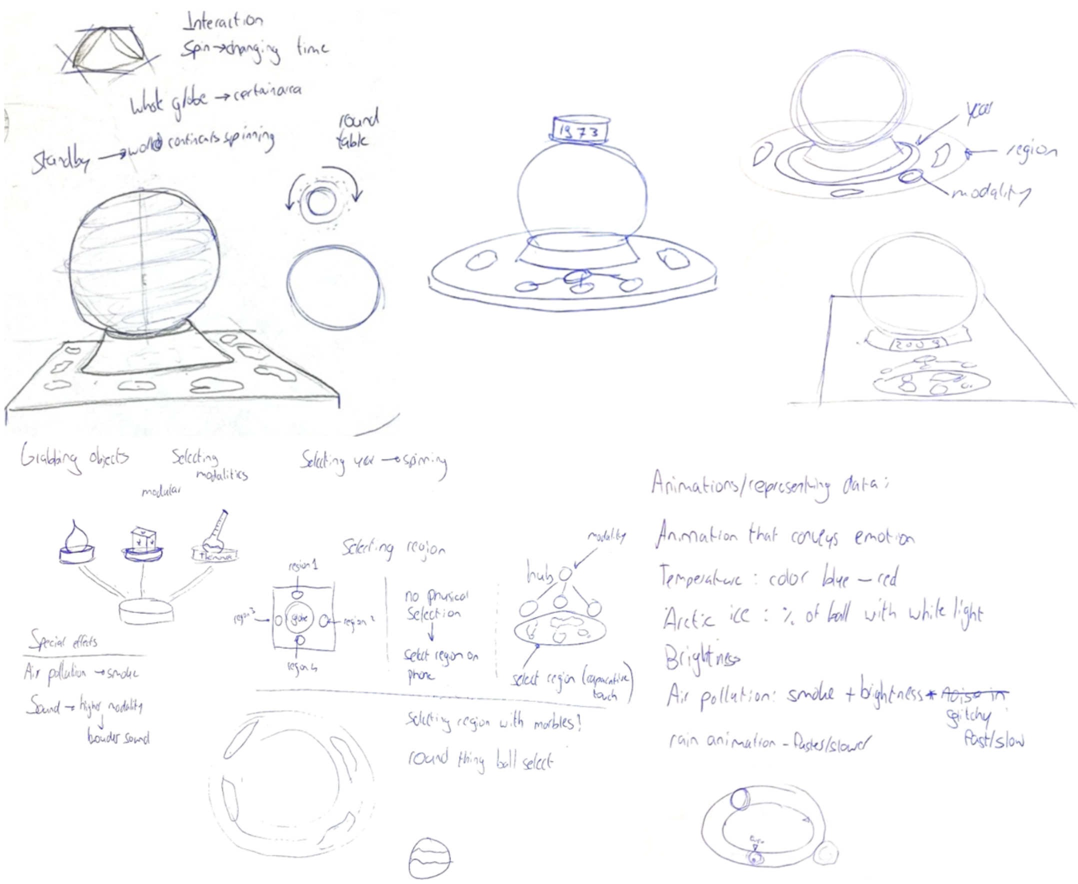

Figure 14 First sketches of the installation

For data output, a spherical abstract representation was explored. This concept would display different datasets in one region at the time by adjusting brightness, colour, rain animations, percentage of illuminated surface, and incorporating sound that would increase with higher values. Smoke effects would activate upon reaching specific thresholds.

In terms of input modes, several interactive interfaces were considered. These included round and square tables for selecting datasets and regions, using a smartphone to choose indicators, or selection with figurines, marbles, or capacitive touch inputs. These interfaces could be positioned centrally or distributed around the globe, encouraging movement and participation from multiple people. Another interactive feature involved using the globe's rotation to navigate through different years, providing a dynamic means of exploring the data. A screen was also included to visualize the current year being shown.

Figure 15 Design concept for the installation

Drawing from these ideas, the next step was to define five possible outputs (PO) and five potential input methods (PI). Building on the abstract globe concept previously ideated, where continents are depicted in abstract vertical slices and labelled accordingly, here are brief descriptions of the different input concepts:

- PO1: Different temperatures are represented by using a colour hue scale, so the redder, the warmer.
- PO2: Different temperatures are represented by using a brightness scale, so the brighter, the warmer.
- PO3: A threshold for air pollution is set, once the data point goes over this threshold, smoke is emitted from under the globe.
- PO4: Music is mapped to temperature, when the temperature becomes higher, the music starts playing louder or faster.
- PO5: Precipitation is represented using animation, the faster certain parts blink, the higher the precipitation.

And possible output mode concepts:

- PI1: The globe is able to rotate by using one's hands.
- PI2: There are miniatures of a thermometer, ice cube, a cloud, and the text "CO2". The user can pick up a miniature from the hub and place it on a central pad, the globe will now switch modalities and you will see different data related to the miniature you put on the main pad.
- PI3: A sort of touch screen slider is utilized so a user can select a year by touching the corresponding spot on the slider. The years you can select are visible above the touchable slider.

- PI4: A rotating knob is utilized so a user can scroll through the years. The current selected year is displayed on an LCD screen.
- PI5: A rotating wheel is used to show the causality between CO2 emissions and temperature. When someone rotates the wheel, a simulation will start where more CO2 is added, and the globe presents a possible future scenario in which CO2 emissions and thus temperature rise.

4.5 Feedback from experts

Once specific concepts were developed, they were presented to researchers specializing in geographical representation and data visualization for feedback. Their main critique highlighted that abstracting continents defeats the purpose of using a globe, which is to provide a scaled-down representation of Earth. Additionally, accurately representing data on a globe presents challenges due to projection distortions and grid scaling issues. They recommended to use the Quaternary Triangular Mesh (QTM) system was introduced as a method to simplify this process and do it in an accurate way. This approach involves meshing the Earth's surface into triangles, where each triangle can represent a data point that can be scaled accordingly [33].

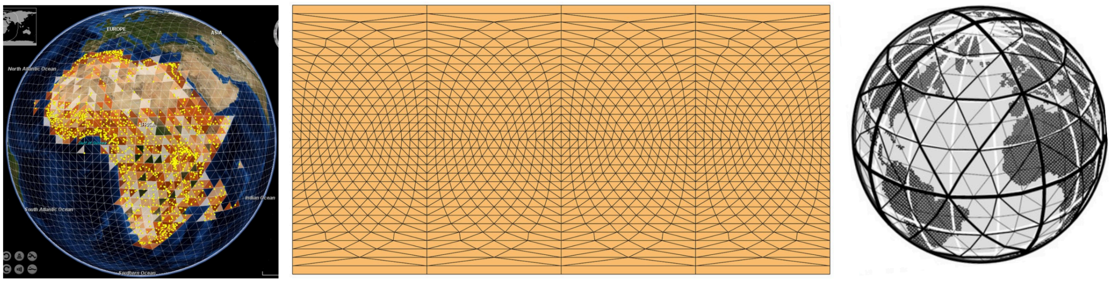

Figure 16 QTM system representation [33]

Furthermore, they suggested to use Copernicus, the Earth observation component of the European Union's Space program. Copernicus offers publicly accessible datasets, including climate change models and historical data [34]. Particularly insightful were Climate Pulse [32] and Interactive Climate Atlas [35] from Copernicus Climate Change Service (C3S), which inspired the final design of the globe.

These visualizations, such as tracking temperature anomalies over time, served as direct inspiration, given the intention to utilize the similar datasets.

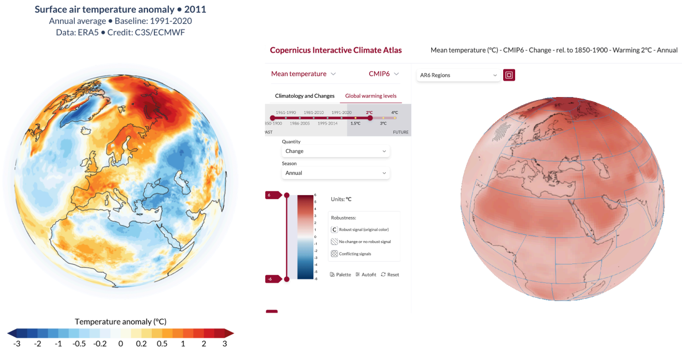

Figure 17 Copernicus' Climate Pulse and Interactive Atlas

4.6 Concept iterations

Following expert feedback, the concept incorporating QTM was refined. However, feasibility within project constraints—both monetary and temporal—became key at this point. Implementation considerations focused on a design suitable for user evaluation, limiting input to year selection via a knob or slider on a swappable dashboard. The dataset choice between total precipitation and surface temperature, could be selected by placing corresponding figures on NFC-tagged pedestals. As time became limited later in the project, the option for user-selected modalities was discarded; instead, researchers preselected them through code. Outputs were limited to only utilize LEDs for colour and animation, with sound or smoke reserved for potential implementation if time allowed. A budget accommodating 2048 LEDs, allowed for a 60 cm ball capable of rotation. Visually differentiating regions and land masses was a challenge, leading to the decision to outline continents initially with wire, later opting for 3D printing. Thought was put to a rotational system (Figure 10) allowing for continuous rotation without tangling wires, utilizing bearings on a base and a slip ring connecting globe components to the power supply.

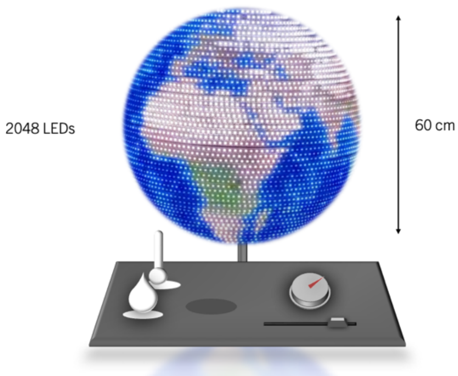

Figure 18 Specification sketch of the system

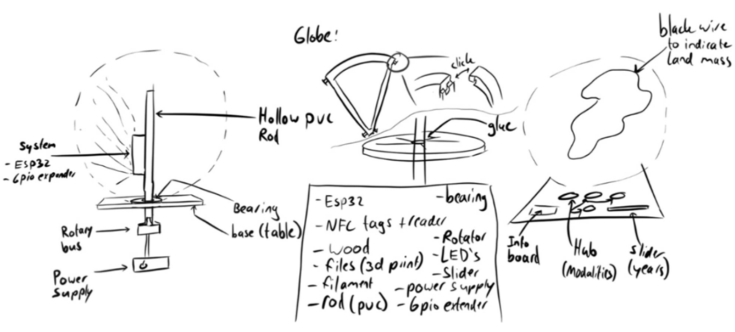

Figure 19 Iteration of the installation

Later in the process, it became clear that implementing the QTM with the LED strip ring concept (as depicted in Figure 20 was not feasible. The globe designed following the QTM approach (on the left) utilized regularly spaced concentric circles, where the distance between rings varied in height, resulting in a significantly higher density at the poles. This uneven distribution happens because the regular height of triangles used in the mesh generates different spacing for the rings viewed from the front.

Consequently, the decision was made to abandon the QTM and instead directly map the data coordinates onto a physical grid composed of equally spaced rings in height. An adjustment was necessary to maintain a consistent number of LEDs. The revised design resulted in a smaller but more densely packed ball, measuring 45 cm in diameter (on the right on Figure 20).

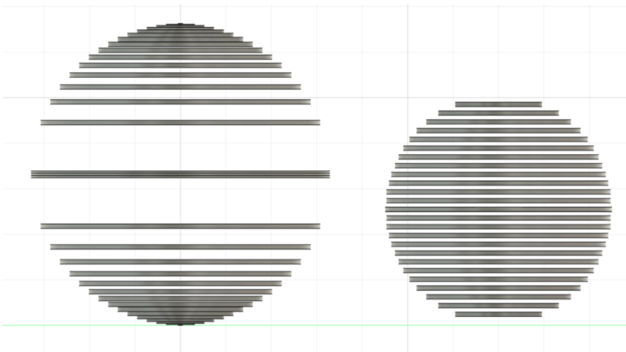

Figure 20 Globe rings with and without using QTM

4.7 Data selection

Ultimately, the decision was made to utilize Copernicus' datasets, specifically the ERA5 climate reanalysis, for its accessibility via API and customizable download options, and acknowledged as a reference organization by the World Meteorological Organization (WMO) [36]. However, we still had to decide whether we would be using absolute values or anomalies.

Anomalies refer to the difference between the actual value of a variable (like temperature) on a specific day, month, or year and the long-term average for that same time period. In climate reports, this long-term average usually refers to the 30-year period from 1991 to 2020, as set by the World Meteorological Organization [37]. Warming stripes are a well-known example of using anomalies to illustrate climate change. Created by climate scientist Ed Hawkins in 2018, these visual representations show annual temperature anomalies from around 1850 or 1880 to the present [38]. Each stripe represents a year, with colours indicating whether the temperature was above or below the long-term average, effectively highlighting trends in global warming.

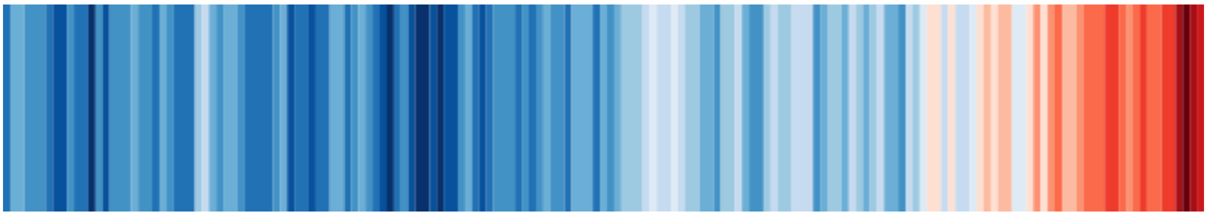

Figure 21 Warming stripes

The advantage of displaying absolute values lies in its straightforwardness, making it universally understandable as it presents raw data directly. In contrast, understanding anomalies requires some prior knowledge or explanation, as it involves interpreting data that is abstracted from its raw form.

However, showcasing absolute values for comparing data across years globally presents perception challenges. For instance, displaying -40 degrees in the poles and 40 degrees in deserts with a wide range of colours can lead to visual ambiguity. Moreover, changes over the years often amount to just a few degrees or even fractions of a degree, which are not easily perceptible at the scale of global data representation. This makes discerning differences between two years nearly imperceptible, as changes involving one degree, such as from 12 to 13.5, result in minimal colour variation within a specific region due to the subtle differences in absolute values not significantly altering the colour representation.

In contrast, by displaying anomalies, the change itself is directly perceived, as it is explicitly represented in the data visualization. This approach significantly enhances the readability and communication of climate change trends. Therefore, it was decided to opt for anomalies over absolute values given by the clarity they provide in visualizing subtle climate change variations.

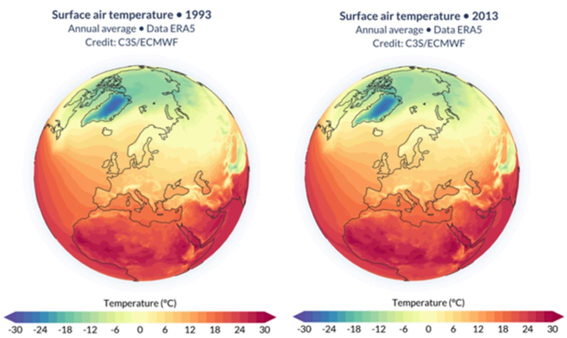

Figure 22 Surface air temperature in 1993 vs 2013 in absolute values

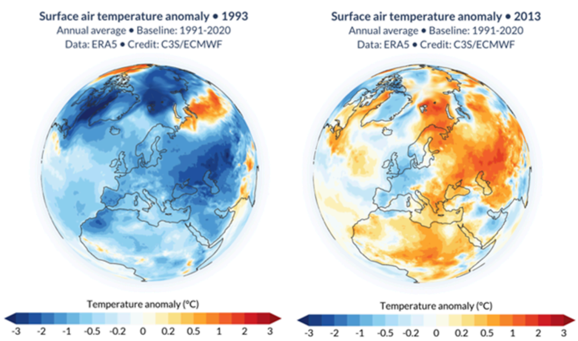

Figure 23 Surface air temperature in 1993 vs 2013 in anomalies

Chapter 5 – Specification

5.1 Requirements

Building upon the concept developed in the previous phase, a set of requirements were established to guide the construction of the installation.

5.1.1 Functional requirements

The user should be able to:

- Select the year from which the data is shown on the globe.
- Navigate through data from 1980-2023.
- Read the data from the globe.
- Distinguish different levels of temperature on the globe.
- Spin the globe.
- Know what continent they are looking at.

The system should allow for:

- Switching between output modes (Blinking, colour, blinking and colour)
- Making easy to implement changes in the way the variables are mapped and displayed.
- Fast updates of the data

5.1.2 Non-functional requirements

The system should:

- Have natural interactions.
- Have embodied control so one can physically interact with the physicalisation.
- Be intuitive and easy to use.
- Be able to be user tested to see the difference between different input and output modes.

5.2 Data set

The dataset selected is the ERA5 climate reanalysis, produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) for the Copernicus Climate Change Service (C3S) [39]. The surface air temperature anomaly was chosen to be displayed on the globe. This data is available from 1979 to 2023 across all months, with the climate reference period spanning 1991 to 2020. For reasons related to data input and display, which will be explained later, the final data presented ranges from 1980 to 2020 in five-year intervals, with the average temperature of each year. The data is provided in GRIB files, offering coordinate-based information in a regular latitude-longitude grid. Given the difficulty of visualizing data at the country level on the globe, the regions are referred to by continent, as well as the poles and the equator, for easier reference by users.

Table 1 Overview of data set variables

Variables	
Year	1980-2020 (5-year interval)
Region	Africa, America (North/South), Europe, Asia, Oceania, Antarctica, poles, Equator
Data set	Surface air temperature anomaly

5.3 Structure

The final design involves latitude-oriented rings, uniformly spaced in height. The globe will have a diameter of 45 cm and will house 1,982 LEDs. The controllers and wiring will be contained within the globe, which is composed of two hollow hemispheres. The 32 rings will be supported by arcs that complete half a meridian, and all components will be attached to a central PVC tube connected to the base. It is crucial that the globe can rotate without the cables getting tangled. To achieve this, the central pipe will rotate on bearings connected to the platform and use a slip ring to manage the cables during rotation. To visualize the landmasses, the outlines of the continents will be 3D printed and painted white to enhance visibility. The base will also house the power supply and serve as a dashboard for the input controls.

5.4 Input modes

The installation will feature three input modes for year selection:

- Slider: This provides a linear representation of time series data, closely matching how people typically perceive time as a continuous, linear progression. This is the most common representation in screen-based or paper-based information visualizations of time series data. The slider allows for a smooth, continuous linear movement.
- Knob 1(continuous knob): This offers a circular, continuous input with smooth rotation. The stepwise transitions are subtle and not easily perceptible through touch or sound, making the movement feel seamless and quiet.
- Knob 2 (discrete knob): This provides a circular, discrete input with distinct, incremental movements. Users can feel each stepwise transition through tactile feedback and hear a slight clicking sound, enhancing the sensation of moving from one level to another.

Compared to the slider, the knobs are more metaphorically distant from the concept of time series data. However, they offer unique tactile experiences that may appeal to different user preferences.

5.4.1 Time series data

These input modes are designed to allow users to intuitively navigate through time series data, which consists of a sequence of data points collected at successive, typically uniform, intervals [40]. Time series data captures temporal dynamics and is utilized across various fields to analyse trends, patterns, and relationships over time. Time series data has several key characteristics [40], [41]:

- Temporal Ordering: Data points are ordered chronologically, making the sequence crucial for analysis.
- Autocorrelation: Values can be correlated with previous values, creating a unique dependency not present in other data types.
- Seasonality: Regular patterns or cycles appear at specific intervals, such as daily, monthly, or yearly.
- Trend: Over long periods, data might show an overall increase, decrease, or stable pattern.
- Noise: Random variations or "noise" are present, complicating analysis.

Time series data is particularly relevant in climate change studies, where data representation across specific time intervals varies based on analysis goals, data type, and research questions. Common practices include [42] [43] [44] [45]:

- Decadal averages: Averaging climate data over decades to smooth out year-to-year variability and highlight long-term trends.
- 5-Year averages: Balancing detailed changes and reducing short-term variability.
- Annual Data: Used for more immediate or short-term changes, identifying specific events or anomalies.
- Monthly data: Analysed for detailed studies, especially on seasonal patterns or specific phenomena like El Niño.
- Running averages: Moving averages, such as 10-year running averages, to smooth short-term fluctuations and highlight longer-term trends.
- Climatological normals: Typically, 30-year averages defined by the World Meteorological Organization (WMO), such as the 1990-2020 reference period for many climate analyses.

Temperature anomalies are often shown as annual or decadal averages to highlight long-term trends. For the current project, the input controls' physical and computational constraints limited the number of years that could be displayed, making it impractical to show more years without sacrificing user experience. Testing on the globe revealed that even differences between two years were noticeable when showing anomalies, whereas changes in absolute temperature values over decades were less visible, due to small changes over longer periods of time. Given the datasets, technical constraints, and the intermediate term we wanted to showcase, a 5-year interval was selected. This choice balanced the need for meaningful data representation with the usability and perceptibility requirements of the project.

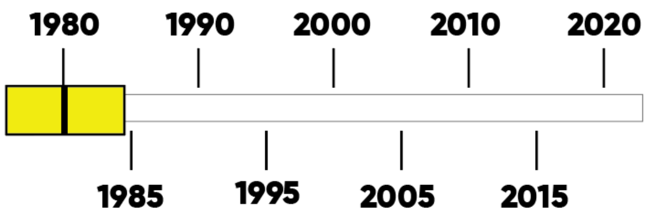

Figure 24 Slider with a timeline with 5-year intervals from 1980 to 2020

5.5 Output modes

In order to display the data three output modes regarding the LEDs were defined:

- Colour: The LED lights on the sphere should be mapped to a colour scale, utilizing data bins to represent different temperature ranges. The colour palette will be a gradient from blue to red, passing through white, to represent temperature anomalies on the colder and warmer sides.
- Animation: This mode combines blinking with colour, but in a binary fashion—red for warmer and blue for colder anomalies. The frequency of the blinking will indicate the magnitude of the anomaly in absolute terms, the faster the blinking, the larger the anomaly. The LED lights on the sphere should blink in a way that makes different blinking patterns easily distinguishable.
- Colour + Animation: This mode integrates both colour and blinking as described above. The LED lights will blink with a gradient colour scale, where the blinking frequency indicates the magnitude of the anomaly, and so the same data is encoded twice.

Chapter 6 – Realisation

6.1 System

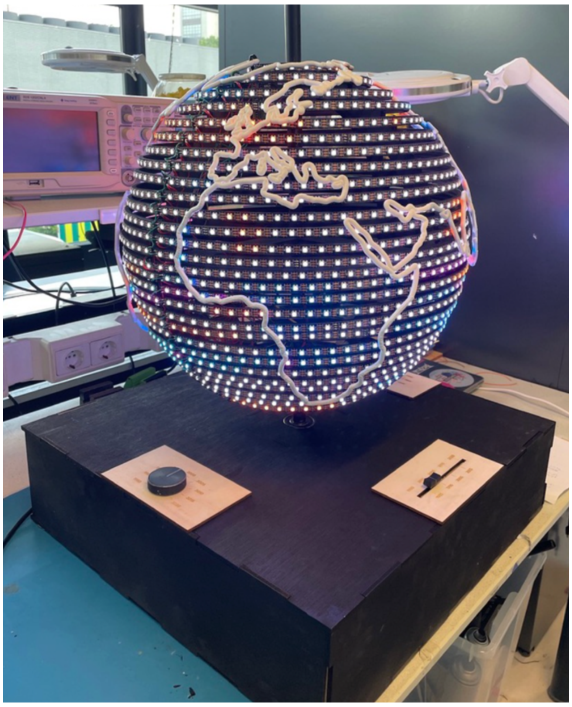

Figure 25 Final installation setup

The physicalisation system consists of four main components: the two hemispheres that form the globe, the dashboard, and the server. Both the dashboard and hemispheres are powered by a computer power supply providing 10 Amps at 5V, and they communicate wirelessly with the laptop. The laptop holds pre-processed temperature anomaly data, which can be directly outputted to the hemispheres.

The system is run by three ESP32-WROOM microcontrollers and a laptop. These microcontrollers were selected due to their built-in wireless communication capabilities. One microcontroller drives the upper hemisphere, another drives the lower hemisphere, and the third handles inputs from various knobs and the slider. Using multiple microcontrollers allows for system modularity and reduces the tasks each microcontroller have to handle, improving the speed of wireless communication. Each microcontroller communicates with the laptop but not directly with each other, ensuring efficient data transmission and processing, optimizing the overall system performance.

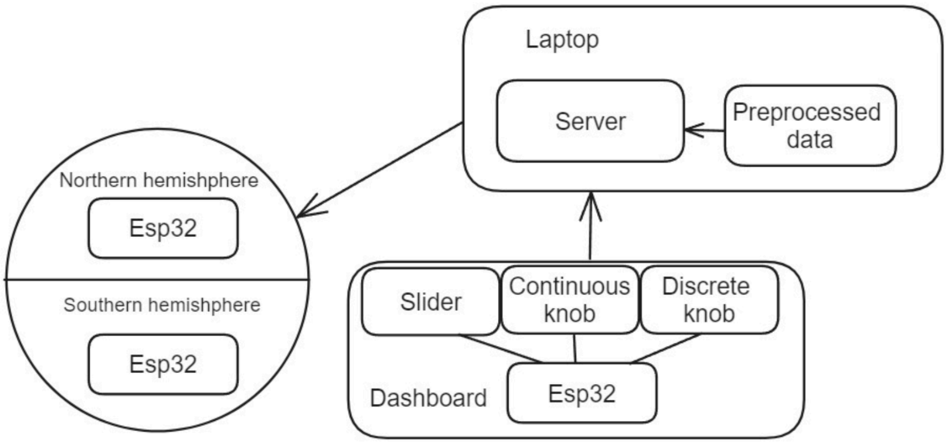

Figure 26 System diagram

6.2 Structure

The physical globe is made from a skeleton made of laser-cut wood, forming rings of 1 cm thickness arranged in a latitude-based design. The globe has 32 rings, each spaced 1 cm apart, with a total diameter of 45 cm. The calculations for the diameters and spacing of the rings taking into account number of LEDs was done with a custom Python script, this can be found in Appendix C – Code lines The model was created using Fusion 360, laser-cut, and then spray-painted black for aesthetic purposes. The LED strips are attached to the rings using the convenient adhesive backing they come with. The wooden parts were glued together, but the two hemispheres were left independent to maintain modularity and ease of work. The ESP32 microcontrollers were also attached to the interior structure, ensuring that everything is encapsulated inside.

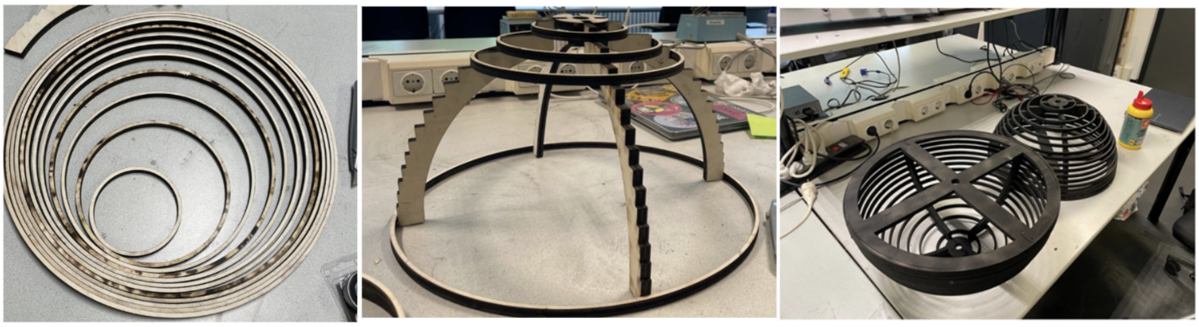

Figure 27 Globe skeleton

6.2.1 Base

The base houses the power supply, the ESP microcontroller for input controls, and the rotation mechanism. The base consists of a laser-cut wood box painted black, with precise cut-outs for the input controls and the central pole.

Serving as both a functional and aesthetic element, the base includes a dashboard made from laser-cut wood rectangles that fit around the knobs and slider, which are then glued onto the base. The years are engraved onto the dashboard for easy reference. The back of the base has ventilation holes for the power supply and a switch to turn the installation on and off.

Figure 28 Back panel of the installation

6.2.2 Rotation

The hemispheres are held together and in place by a thin PVC pipe, painted black to match the aesthetic. The wires travel inside the pipe down to the base. To enable infinite rotation without tangling the wires, a slip ring is used between the power supply and the electronics on the globe. The selected slip ring had thick wires as it had to hold all the current needed for the entire globe. This slip ring is connected to a bearing at the base and attached to the PVC pipe via a 3D-printed connector, which rests on a second bearing.

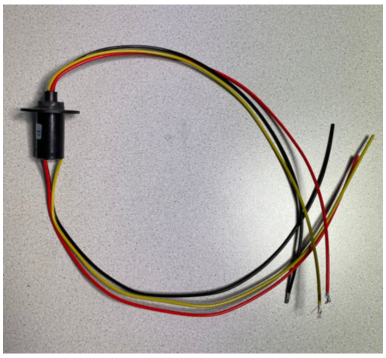

Figure 29 Slip ring

Figure 30 3D printed part for attaching the slip ring to the pole

This system requires elevation from the floor of the base to function correctly, so it is encapsulated inside a box attached to the ceiling of the base. This design ensures smooth rotation and secure wiring.

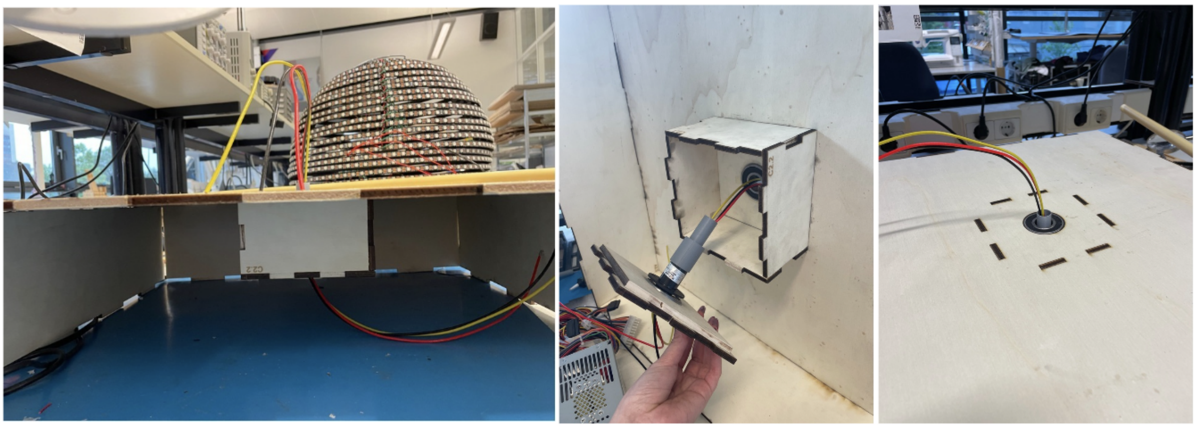

Figure 31 Rotation mechanism

6.2.3 Continents

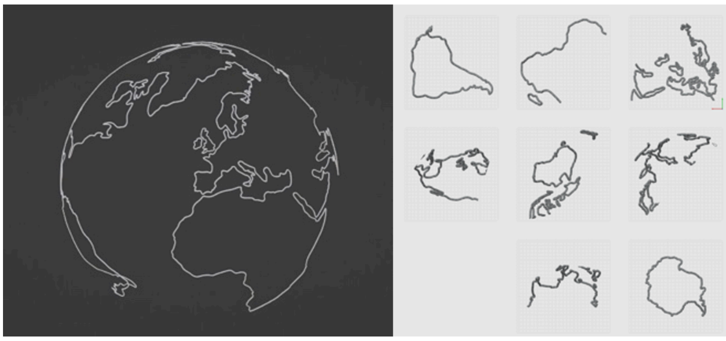

Figure 32 3d model of continent outlines and plates prepared for printing

The continents were 3D modelled in Blender to fit over a 45 cm diameter ball, ensuring the necessary curvature for precise placement. The outline of the continents is 1 cm thick and made of white spray-painted PLA to increase visibility when backlit by the LED ball. To fit the 3D printer plate, the model was cut into 8 individual prints. These prints were later glued together and placed on the globe by lighting the continents on the LEDs, ensuring accurate positioning.

6.2.4 Globey tag

A 3D-printed tag with the nickname "Globey," given by the researchers, was added to the top of the globe. This decision was made because leaving the pole slightly longer than necessary was practical for transportation and provided support for the upper hemisphere while working with the globe. The tag also served as an embellishment, adding a personalized touch to the globe.

Figure 33 Globey logo

6.3 Input

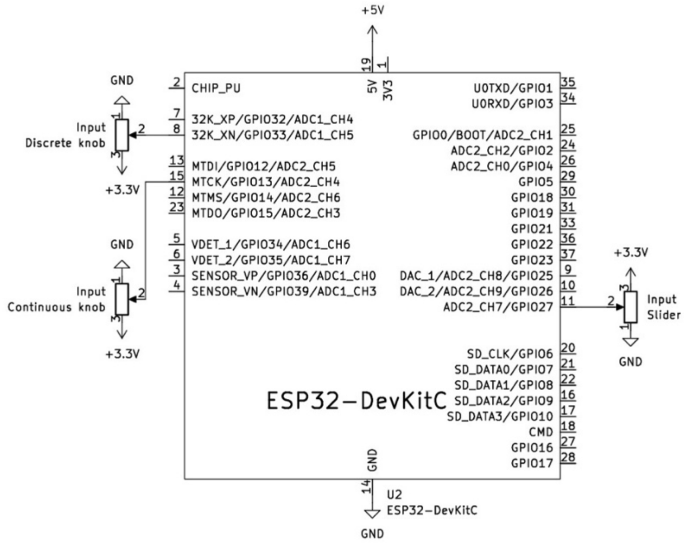

Figure 34 Dashboard ESP schematic

The different elements function as potentiometers and are powered by the 3.3V output of the power supply, as the ESP32's analog-input channels can only measure up to 3.3 volts.

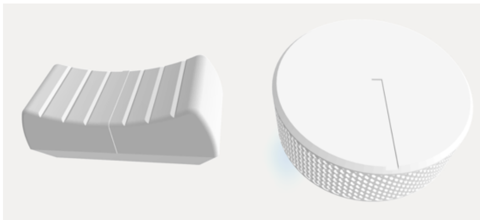

Figure 35 3d models for grip

3D-printed parts were used to improve the aesthetics and user-friendliness, adding grip and precision. These models were downloaded from Thingiverse. The years were engraved on wooden plates to view the selection.

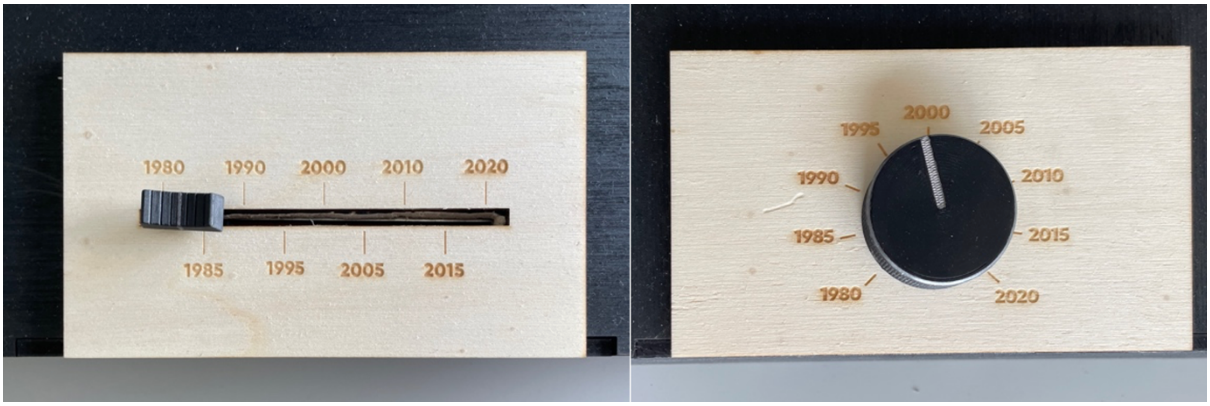

Figure 36 Engraved dashboard

6.3.1 Slider

The selected slider is a linear potentiometer with a resistance of 10kΩ and a travel length of 10mm. Longer options were difficult to find, but a longer travel length would provide more accurate year selection by increasing the margin between each year.

Figure 37 Slider

6.3.2 Continuous knob

For the continuous knob, a potentiometer was chosen. It allows for smooth and continuous movement with defined minimum and maximum stops.

Figure 38 Potentiometer

6.3.3 Discrete knob

A rotary switch with a 1KΩ resistor ladder was chosen for the discrete knob. This setup measures the voltage drop across each established position. The rotary switch has 12 possible positions, each corresponding to a specific voltage level due to the resistor ladder. It provides tactile feedback by producing a clicking sound at each position. The 12 positions could have been a limiting factor for the amount of years to be selected but only 9 of them were necessary in this installation.

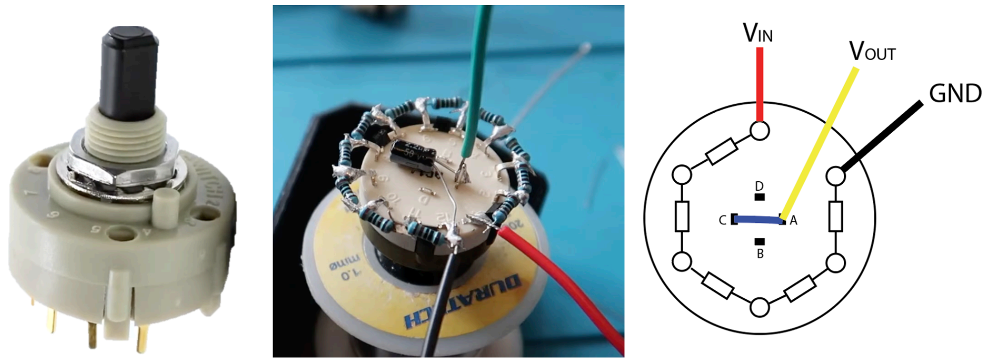

Figure 39 Rotary switch with resistor ladder

6.4 Output

6.4.1 LEDs

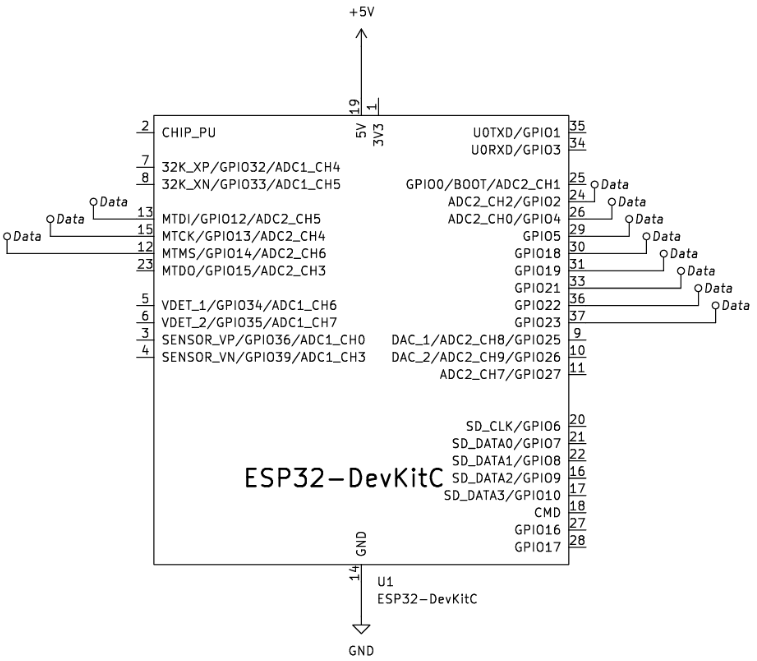

Figure 40 LEDs ESP schematic

The WS2812b LED strips were chosen for their ease of use and modularity. They can be cut to different lengths and are simple to program.

The schematic above is used for a single hemisphere and is duplicated for both hemispheres. The ground and power lines are connected to the power supply. There are 11 data wires connected to various LED strips. Each hemisphere contains 16 rows of LEDs, but only 11 data wires are used because some LED strips are daisy-chained to minimize the number of required pins.

The Adafruit Neopixel library is used to control the LEDs. Each LED strip requires its own data wire. However, the documentation does not specify the maximum number of LEDs that can be connected to a single strip while maintaining proper functionality. To ensure reliable operation, the design uses multiple shorter LED strips instead of one long strip. Some rings are connected in series, while others are connected in parallel, to distribute the current adequately across the thin wires.

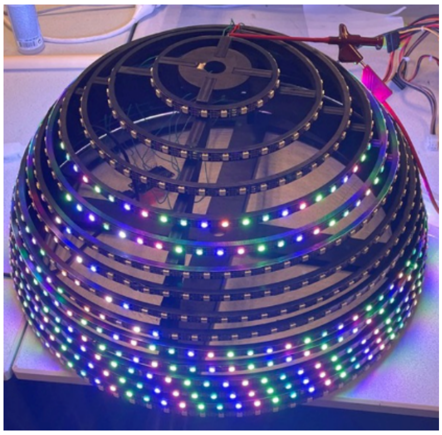

Figure 41 LEDs on one hemisphere

6.4.2 Data mapping

Animation, colour, and colour + animation modes were implemented. The data had to be set into bins in order to translate colour and frequency levels. After testing both modalities on the globe these were the appointed bins:

[0, 0.5], [0.5, 2], [2, 3]

The data's minimum and maximum values were set at -3 and 3 degrees, respectively, to enhance perceivability, aligning with the Copernicus Climate Atlas standards for displaying temperature anomalies. The temperature anomaly can be both positive and negative; however, frequencies use absolute values, as negative numbers are complicated to show by blinking. This information is encoded in binary colour data for the 'animation' mode: red for values above 0 and blue for values below 0. This allows users to distinguish between warm and cold temperatures, with reliable information conveyed through blink speed.

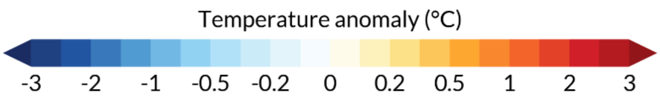

Figure 42 Copernicus' colour legend for anomalies

The blinking animation was tested by setting the sphere to a specific colour and then applying data bins and frequencies. Testing revealed that frequencies significantly different from each other are more recognizable than those that are close. Slow blink speeds require concentration and time for recognition. Therefore, only three distinct frequencies were chosen, categorized into three levels: almost no change, moderate change, and significant change. The frequency value ranges from 0-255 and is multiplied by 100 on the microcontroller, determining the time between blinks (e.g., 10000 milliseconds).

[bin_start, bin_end, frequency]
[0, 0.5,100], [0.5, 2,60], [2,3,20]

When using five colour bins, both blinking and colour can be used simultaneously. It was crucial to ensure that the combined use of colour and blinking did not mislead users due to inconsistent bin assignments. The colours range from dark blue to light blue, white, yellow, and red. Ensuring visible contrast between dark blue and light blue, and between yellow and red, was important. Orange was tested but proved hard to distinguish from red. Additionally, the red LEDs themselves have a slight orange tint. The final set of colour bins was determined as follows:

[red value, green value, blue value]
[0, 0, 255], [50, 50, 150], [50, 50, 40], [75, 75, 0], [255, 0, 0]

The server processes data values to provide RGB and frequency values to the controller. The server sends a large array of 8-bit integers, with each LED receiving four integers: one for red, green, blue, and one for frequency. The maximum brightness for colours is 25, unless otherwise specified. A dimming factor, based on the current time in milliseconds, is then applied through a sine function to ensure smooth animation.

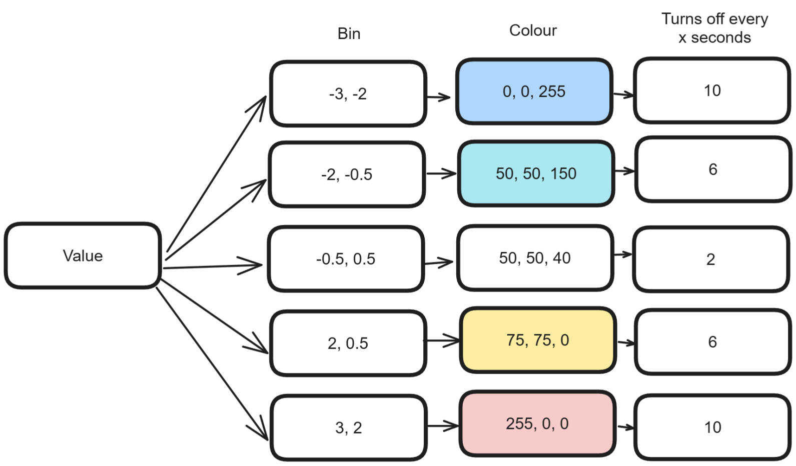

Figure 43 Mapping of data to output modes

6.5 Wireless communication and server

The communication between the microcontrollers and the laptop was handled through WebSockets, which is a communication protocol that facilitates real-time, two-way communication between a server and a client, with the laptop acting as the server in this case. The microcontroller used for the dashboard sends the selected year from all the knobs to the laptop, and the microcontrollers for the upper and lower hemispheres only receive data and directly output it to the connected LEDs, all data and visualization processing is handled on the server.

Additionally, a web interface was hosted on the server to control various parameters of the installation from the backend, essential for debugging and controlling variables during the user study. This interface allows the researchers to tweak variables affecting the physicalisation, including changing the year, enabling, and disabling the different input modes, setting data bins for frequency and colours, adjusting the colour palette, brightness, and defining max and min values.

Figure 44 Web interface for controlling the visualization variables

To detect the year selected among the inputs, the system waits for user input and checks if the input is enabled on the web interface. If enabled, it gathers data for the selected year and applies the relevant settings, such as colours and bins. The data is then converted into an array of RGB and blink frequency values, avoiding the need to send raw values to the globe. It splits data accordingly, and each packet is sent to the corresponding hemisphere.

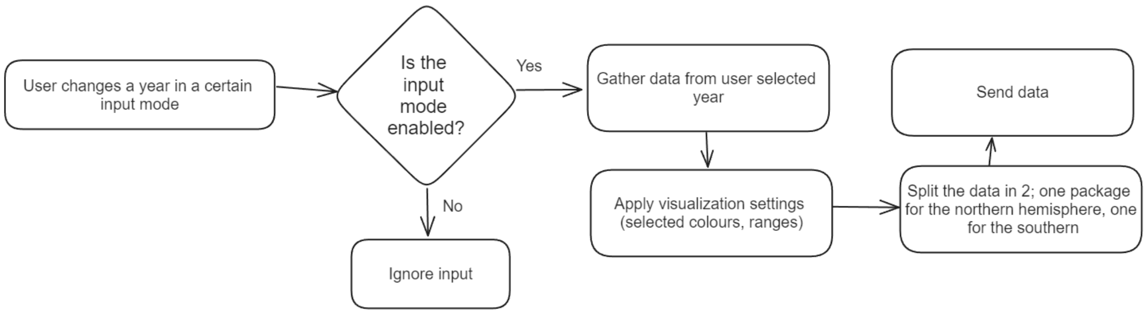

Figure 45 Server logic for detecting year input

6.6 Data conversion

The data from the ERA5 yearly mean anomaly was downloaded for each year in the database, with each year represented as a separate file in .GRIB format, a common format for weather and earth data. Each GRIB file from the CDS contains metadata and a 2D array with temperature anomaly values, with latitudes on the x-axis and longitudes on the y-axis, incremented by 0.25 degrees.

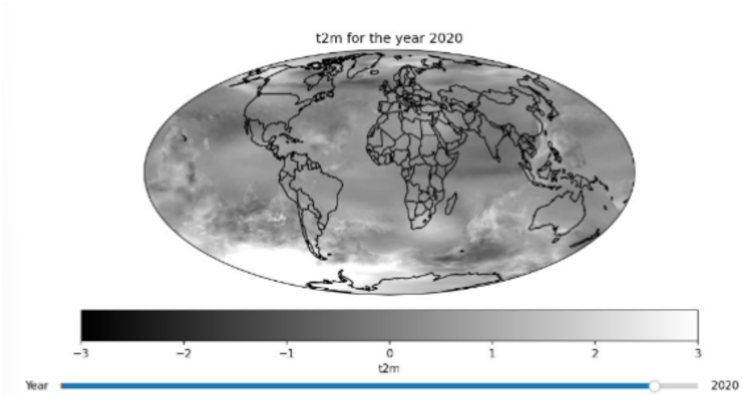

Figure 46 GRIB data visualization using matplotlib

The low resolution of the globe made near real time interaction possible, but that requires scaling down the GRIB files to the globe's resolution. The globe has 32 rings, each containing a different number of LEDs. A script was created to divide the latitude into 32 different bins, each further divided into longitude bins corresponding to the number of LEDs on each ring. This method creates 'boxes' into which the high-resolution data values fit. The values from the GRIB file were scaled down and linearly interpolated to ensure smoothness. Each square on the right side represents an LED on the physical globe. The downscaled data is then stored as a JSON file containing Python dictionaries with values. This allowed the data to be loaded very quickly, enabling real-time interaction.

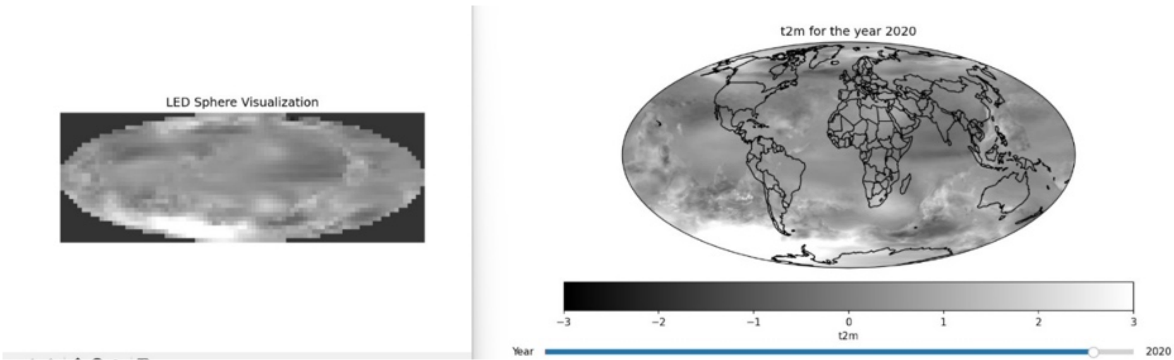

Figure 47 Scaled down data visualization

Chapter 7 – Evaluation

To evaluate how the installation conveys climate change data and engages users, various input and output modalities were tested through a user study. This evaluation focused on three potential physical interaction devices for time series input: a slider and two types of knobs. In the sections below, the goal, design, procedure, and results of the experiment are described in greater detail.

7.1 Goal of experiment

The purpose of this study is to compare the effectiveness, efficiency, user experience, and user satisfaction of the three interaction metaphors: a slider, a continuous knob, and a discrete knob. Additionally, it aims to evaluate how well the globe physicalisation conveys climate change data and promotes user engagement.

7.2 Study Design

The study was designed to allow both researchers to utilize the results for their respective analyses. The input methods were tested in combination with three different output modalities, resulting in nine possible input-output pairings. A between-subject design was used, where participants were assigned to a single combination of modalities, to minimize learning effect. The study aimed to ensure an equal number of participants tested each input method.

7.3 Variables

7.3.1 Dependent Variables

Dependent variables are the outcomes that the researchers measure to determine the effect of the independent variables. In this case, they were efficiency, effectiveness, mental demand, physical demand, UX/usability, satisfaction, and comments. The table below explains how each dependent value was measured.

Table 2 Dependent variables and their measuring techniques

Dependent variables	Measuring Technique
Efficiency	The amount of time users take to answer the questions in the questionnaire.
Effectiveness	Number of accurate answers to tasks
Mental demand	NASA-TLX using a seven point likert scale
Physical demand	NASA-TLX (use seven point likert scale)
UX/usability	UEQ S
Satisfaction	3-question After Scenario Questionnaire (ASQ) for questions 1 and 2
Subjective feedback	Analysing comments left by the participants in a free input text

7.3.2 Independent Variables

Independent variables are the factors that the researchers manipulate or control to observe their effect on the dependent variables. In this case it refers to the different input and output modes. Since one user only interacts with one combination of input-output, these are classified into different codes and assigned at random to each user. See table below for the code distribution.

Table 3 Independent variable code for participant distribution

	Output 1 (colour)	Output 2 (animation)	Output 3 (colour + animation)
Input 1 (discrete knob)	P1.1	P2.1	P3.1
	P1.2	P2.2	P3.2
	P1.3	P2.3	P3.3
	P1.4	P2.4	P3.4
Input 2 (continuous knob)	P4.1	P5.1	P6.1
	P4.2	P5.2	P6.2
	P4.3	P5.3	P6.3
	P4.4	P5.4	P6.4
Input 3 (slider)	P7.1	P8.1	P9.1
	P7.2	P8.2	P9.2
	P7.3	P8.3	P9.3
	P7.4	P8.4	P9.4

7.3.3 Subjective Variables

These are the variables referring to characteristics that vary across participants. In this study those were compiled in the demographic questions, including age, gender, experience with data physicalisation, experience with climate change data, experience reading geographical data. By compiling this experience-related information, we can later examine any correlations between these variables and the study outcomes.

7.4 Participants

36 participants were recruited through word of mouth and social media. There were no special requirements regarding participant age or experience. However, participants had to be able to understand and speak English as the study was conducted in this language.

7.5 Procedure

During the study, one participant and one or two researchers were present. The time for the study is about 15 minutes. After a short introduction about the goals of the experiment, the participant was asked to fill out the consent form. Next, the researcher asks the participant to step in front of the installation.

The researcher explains how the physicalisation works and invites the participant to start with the questionnaire on the laptop that will guide them through the experience. They first introduce demographic data and personal background in a questionnaire, a3nd then, the participant some time to play and become familiar with the installation through dummy data. The participant performs 4 tasks interacting with the globe and dashboard while simultaneously filling in the questions regarding the data they perceive. At the end there are a few questions about the usability of the globe, its physical and mental demand, and a free text input for comments. Lastly, the researcher wraps up the experiment session by giving the participant an opportunity to ask any questions or comment on the experiment and offering the participant a cookie as a reward.

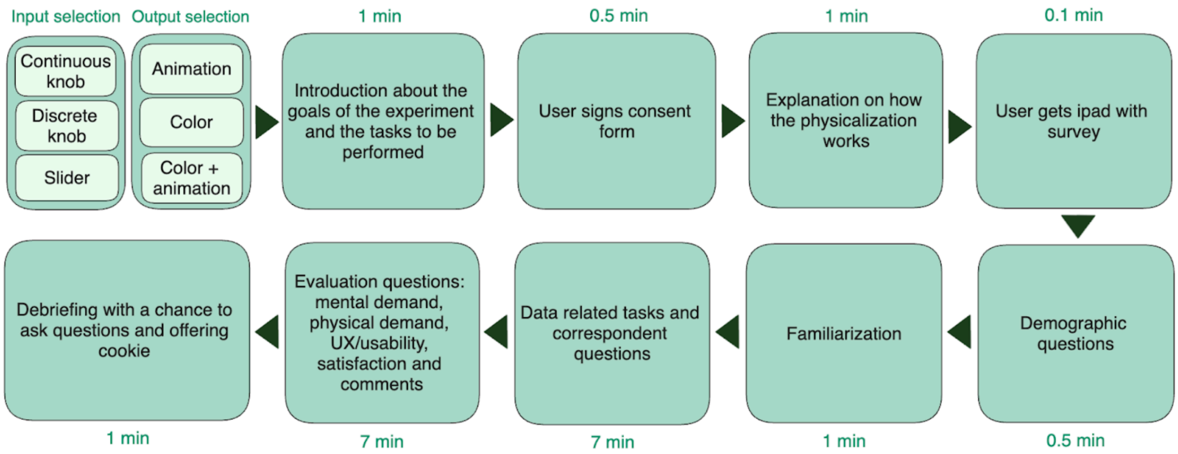

Figure 48 Procedure of the user evaluation

7.5.1 Apparatus

Participants interact with the interactive physical globe installation consisting of a globe made of LED rings and dashboard. The user is able to interact with the dashboard that the researcher indicates and touch the globe in order to point at it or rotate it. A laptop is used to fill in the survey hosted in Lime Survey. This survey tool allows for measuring the time each participant takes to answer the questions.

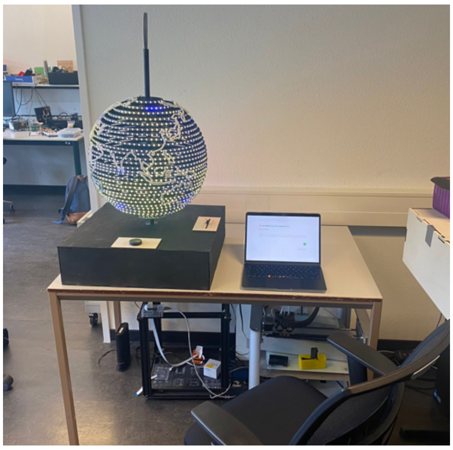

Figure 49 Experiment setup

7.6 Tasks

The user tasks in this study are categorized into four groups: preliminary tasks, familiarization, exploration, and evaluation. During the preliminary tasks, participants fill out a consent form and demographic questionnaire, which takes approximately one minute. The familiarization phase begins with an explanation of how the globe works, the meaning of the displayed data (temperature anomaly), and what the tasks they are about to perform. Following this, participants interact with a globe showing dummy data to become comfortable with the system, a process that usually takes about two minutes. Next, in the exploration phase, participants are prompted with four questions about climate change, which they answer by interacting with the globe. This phase usually takes around seven minutes. Finally, in the evaluation phase, participants complete questions regarding their mental and physical load, the user-friendliness of the system, and are given an opportunity to provide feedback. The evaluation phase also takes about seven minutes. The participant gets a laptop which has 4 questions related with one task each.

7.7 Questionnaire

The 4 questions relating to the data being shown are:

- In the last decade (2010-2020) has Africa become more warm or more cool in general?
- In 1985, what hemisphere has the biggest temperature change, southern or northern? Keep in mind that change includes warmer as well as colder temperatures.
- What area has the highest temperature anomaly in 2000? (Warmer)
- Is the equator warmer in 1990 compared to 2020?

The questions were extracted from a pool of potential identification and comparison questions (see table). These questions included various combinations of year selections (e.g., one year, comparing two years, a range of years, or the total range of years in the dataset) and region selections (e.g., the same region in two different years, or two regions in the same year). The wording of the questions had to be carefully chosen, as terms like "warmer," "colder," and "higher anomaly" can be confusing when evaluating temperature anomalies. Therefore, we provided additional explanations for questions 2 and 3 to ensure clarity.

Table 4 Classification and examples of climate change questions

Type	Question type	n. years	n. regions	Template	Example	Answer
Y, R vs R	attribute in space + identify	1	2	In YEAR, what region has the biggest temperature change compared to the mean (anomaly), REGION or REGION?	In 1980, what hemisphere has the biggest temperature change compared to the mean (anomaly), southern or northern?	Southern/ northern
Y, max	attribute in space + identify	1	all	In YEAR, which REGION experiences the most increasing temperature/lower temperatures than usual?	In 2015, which continent experiences the most increasing temperature? highest temperatures?	Continent
R, Y vs Y	Space in time + identify	2	1	Is REGION warmer/colder than usual in YEAR compared to YEAR?	Is the equator warmer in 1990 compared to 2020?	Yes/No
R, range YY	Space in time + identify	range	1	Over the past RANGE, has REGION generally become warmer or cooler?	Over the past decade, has Africa generally become warmer or cooler?	Warmer/ cooler
R, total Y	Space in time + identify	all	all	In what REGION does the mean temperature change the most?	In what area does the mean temperature change the most?	Poles/ equator

7.8 Results

7.8.1 Quantitative results

The evaluation results can be divided into qualitative and quantitative aspects. The quantitative analysis, based on questionnaire responses and task performance, was compared, and statistically analysed using SPSS to determine whether there were significant differences between the input modes regarding efficiency, effectiveness, mental and physical demand, UX/usability, and satisfaction.

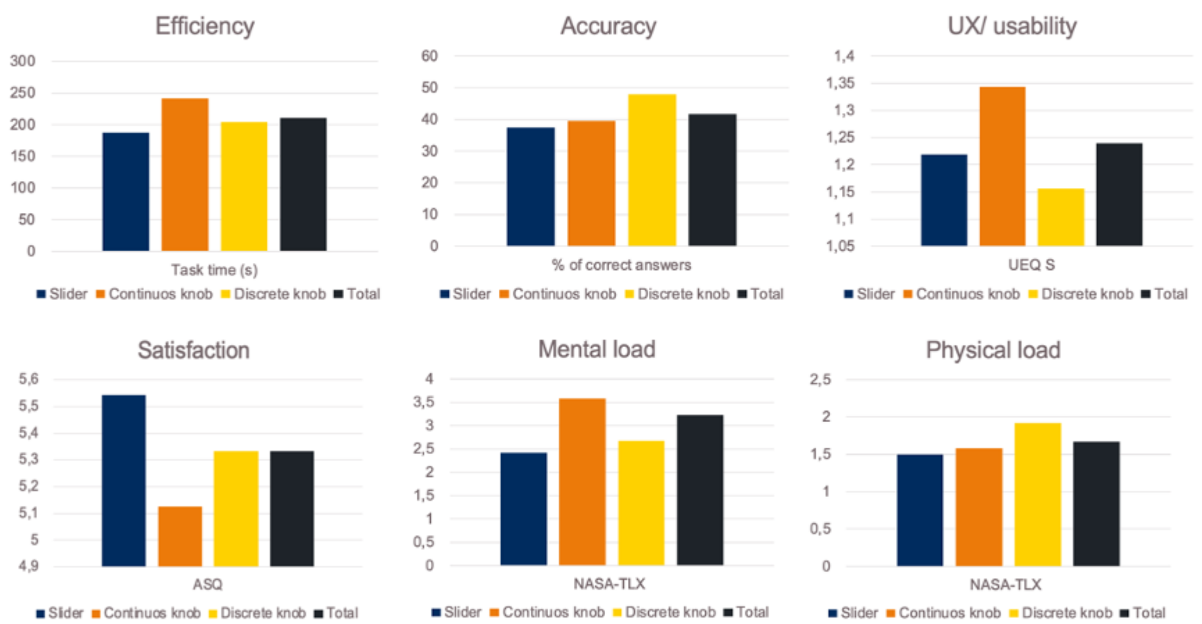

Figure 50 Graphed means of the performance of each input mode per evaluated metric

Efficiency was determined by the time it took participants to complete all four tasks. The continuous knob performed better in terms of efficiency compared to the other two input modes. For accuracy, measured by the percentage of correct answers, the discrete knob performed the best. However, accuracy values averaged below 50%, which is low given the multiple-choice format of the questions.

UX/usability was assessed using the UEQ-S questionnaire, where users rated the installation on a scale between two opposite qualities, such as complicated/easy or boring/exciting. The results, converted to a score from -3 to 3, showed all input modes scored positively, averaging around 1.2, indicating above-average performance. The continuous knob led this metric.

Satisfaction was measured with two questions from the ASQ questionnaire, rating the ease and time to complete the tasks on a scale from 1 to 7. All scores were over 5 for every input mode indicating high satisfaction. Mental and physical loads were assessed using the NASA-TLX questionnaire, where users rated their perceived mental and physical loads from very low to very high on a scale from 1 to 7. The graphs show low values for both mental and physical loads, with mental load averaging 3.2 and physical load around 1.7. The continuous knob had the highest mental load, while the slider had the lowest. For physical load, the discrete knob led, and the slider had the lowest values, though the differences were not substantial.

ANOVA was chosen to determine if there were significant differences between the input modalities, as it is the most suitable test for comparing the means of more than two groups when the data is normally distributed. The test was conducted with the following hypotheses:

H0: There is no difference among group means

H1: There is a difference among group means

ANOVA requires the assumption of normality. To check for normality, QQ plots were generated. As shown in Figure 51, the data points align closely with the diagonal line, indicating that the data is approximately normally distributed.

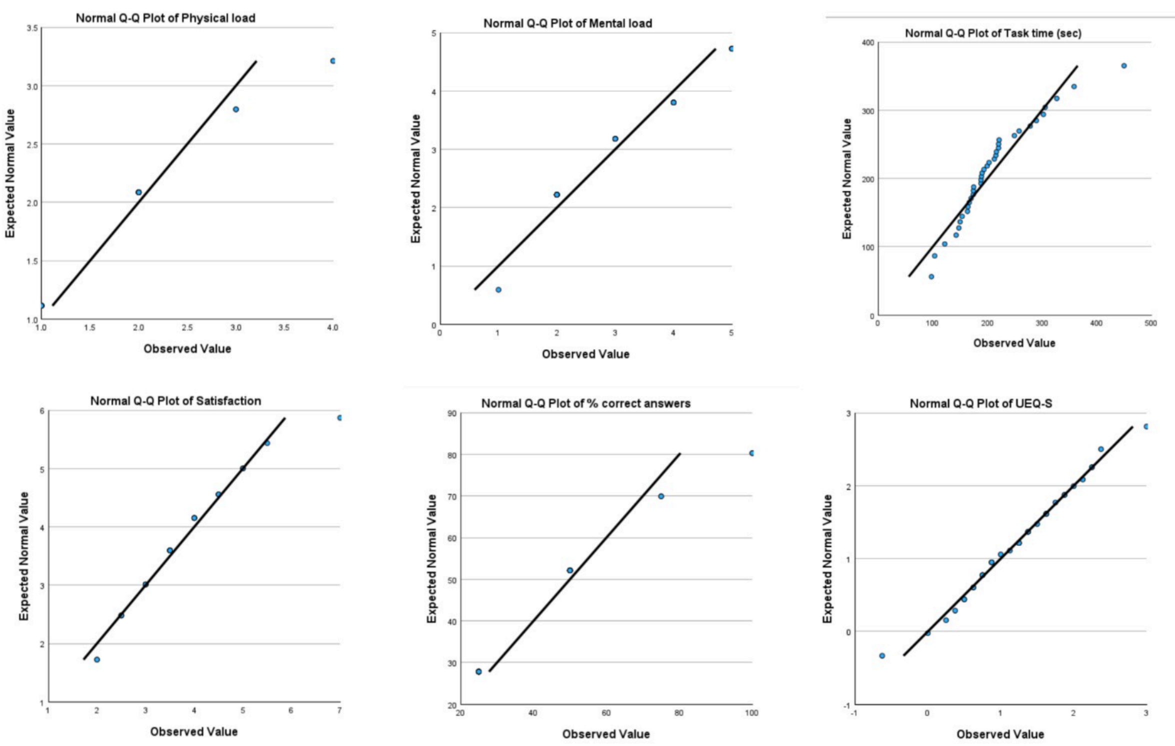

Figure 51 Normal Q-Q Plots for the Measured Metrics

The significance, or p-value, indicates whether there is a statistically significant difference between the input modes. Since the study is exploratory with a relatively small number of participants, an alpha value of 0.1 was used. This means that a p-value less than 0.1 indicates statistical significance.

As shown in Figure 2, all p-values are higher than 0.1, indicating no significant differences between input modes for any of the metrics. Consequently, the null hypothesis could not be rejected. However, efficiency (task time) and mental load are close to being significant.

ANOVA

		Sig.
UEQ-S	Between Groups	.829
Task time (sec)	Between Groups	.188
Mental load	Between Groups	.159
Physical load	Between Groups	.439
Satisfaction	Between Groups	.632
% correct answers	Between Groups	.439

Figure 52 ANOVA results table

7.8.2 Qualitative results

The final section of the questionnaire allowed users to provide free-text feedback, and researchers also took notes of comments made during the evaluation.

Regarding the input modes, some participants compared the different input modes outside the evaluation (as they only interacted with one mode during the test). The majority expressed a preference for the discrete knob, describing it as “satisfying” due to the clicking sound and haptic feedback. One participant mentioned that it would have been nice if the slider had a clicking option too. Those who tested the continuous knob also made positive comments about how smooth it felt. Some participants expected years in between the intervals engraved on the dashboard, leading to confusion, especially with the slider. There was also uncertainty about whether the 5-year intervals meant that the data shown was the average of those five years or just the selected year.

Some popular remarks regarding the whole installation were about the difficulty to distinguish between land mass and ocean with the outline of the continents:

- *“When trying to locate regions on the globe it is somewhat difficult to differentiate between bodies of water and land”.*
- *“I also was a bit disoriented at first with looking at the world map.”*
- *“It is sometimes a bit hard to see where you are with the edge outlines of the continents only.”*

Another notable comment was the need for labels for the data being displayed and a legend for the colours shown on the globe. While some were amused by how the globe spun, others felt it was unstable or fragile:

- “The globe feels quite fragile”.
- “I thought that turning the globe by hand felt a bit difficult and unstable.”

An interesting comment compared the globe to high school map reading exercises from an atlas and suggested its potential use for teaching geography-related topics to children in a more engaging way.

Data perception was sometimes challenging, especially when blinking was involved as an output mode, which many found confusing.

Overall, the installation was well-received, with users finding it fun and interesting. Even when the installation was off, people were drawn to it, curious about its purpose.

- “The concept is very innovative in the way to assess geographical map information”.
- “Impressive installation all in all!”
- “An interactive and nice experience.”
- “The physicality of the object made it very appealing and intuitive to manipulate. It felt great to navigate through the data.”
- “I love the visualisation! it works amazing too”.

7.9 Implication of the results

Overall, the globe was well-received, engaging, and curiosity-inducing. Although the quantitative results showed no significant difference between input modes, subjective feedback indicated a clear preference for the discrete knob due to its satisfying feedback. Since it makes no difference in the metrics, the discrete knob can be considered the best option.

Accuracy and efficiency in answering the questions related to the tasks could be argued that may be more influenced by the output modes, as they determine how the user receives the data. Input mode could affect the results if the year selection was wrong, but the design aimed to minimize this issue. Low accuracy can be attributed to the nature of the questions and the difficulty of reading data from the globe, particularly when comparing areas that cannot be seen simultaneously. It was also observed that participants often relied on their pre-existing knowledge or intuition on climate change when data was ambiguous or hard to read.

Despite multiple users expressing a strong liking for the discrete knob, it performed the worst in the usability test. Factors such as the between-group design, lack of comparison with other inputs, and the holistic evaluation of the installation could have influenced their preferences.

In terms of mental and physical load, the hypothesis that the slider's linear movement corresponds with the mental image of time might explain its lower load scores. However, this remains a hypothesis due to the lack of significant results. The continuous knob generally performed slightly better in most of the metrics, but again, the differences were not significant enough to give meaningful insights.

Chapter 8 – Discussion & Future Work

8.1 Limitations

When building the globe, several limitations impacted the final outcome, primarily due to monetary and time constraints that hindered a more accurate representation. For instance, the cost of LEDs constrained our budget, affecting the display resolution. The placement of the LEDs was another challenge, as the vertical orientation of the LED strips did not align with the sphere's curvature, resulting in less accurate representation. Although soldering individual LEDs or using more densely packed strips could address these issues, budget and time limitations prevented following this path.

The continents' placement was imprecise, introducing an error margin to the representation. Feedback from the evaluation indicated that the continents were difficult to perceive, complicating data reading. And the QTM system, initially considered for data mapping and conversion, proved unsuitable for our globe design, further affecting accuracy.

Working with time series data posed some challenges. For instance, slight changes in absolute values were difficult to show even in groupings of decades. Trying to display these on LEDs would have been nearly impossible. Displaying anomalies made changes more perceptible but required explanation, as user studies indicated it took one or two explanations for clarity.

Comparing two years could be done by comparing a month or year averages, and by showing increments of 5 years it could be interpreted as showing the average of those 5 years or show just the first year in the interval. The final decision was to show the average temperature of one year, as this was the easiest way to interact with the data base. However, the pros and cons of our chosen methods have not been fully evaluated, which may affect data perception or accuracy. To avoid user confusion and misinformation, it is crucial to clearly label what is being displayed.

The choice of the input mechanisms was guided through the nature of the times series data. Sliders and knobs provided a continuous (linear or rotational) range of values which correlates with the linearity of time and distribution of one year after the other in an established order. Even though the smooth options allowed for faster interactions, the discrete knob emphasizes the individual data points of the year selected. An ideal option would have been a slider with steps; however, suitable options were either unavailable or had too few steps for the project requirements.

The slider itself was limited in how many years could be labelled on the dashboard due to its length. A longer slider or a digital display for the years could have mitigated this issue. The discrete knob also presented a limitation with its 12 positions, requiring the data to be grouped by 5-year intervals to fit these constraints. Additionally, the dataset, limited to temperature anomalies from 1979 to 2023 from Copernicus, restricted the years available for visualization. Comparisons with longer-term trends, dating back to the 1800s as seen in other climate change visualizations, were not possible without switching data providers or conducting additional calculations. However, the intervals presented allowed for a mid-term comparison and trend perception, providing a detailed enough overview.

Another limitation was the lack of a display of the selected year around the installation, reducing its effectiveness as a collaborative tool. A planned solution, if time had permitted, was to add one display showing the year around the globe. Several interactive features, such as sound or smoke effects and additional datasets like precipitation, were part of the initial concept but were eventually dropped due to time constraints. Including these elements would have demonstrated the globe's capability to communicate various climate change indicators, not just temperature.

Regarding the user test, a larger sample size would have provided more robust results, as the current study with 36 participants showed no significant differences between input modes. The low scores on the four tasks suggested that the globe's clarity might have led users to rely on their intuition, revealing a weak point in the installation. The low detail level of the display made it difficult to estimate specific regions and compare their values. Regions like the poles were not visible due to the ring-based design. Including personal preference questions could have added valuable insights into subjective aspects, rather than relying on informal feedback. Specially since many of the feedback responses from the survey did not mention the input modes, likely because the most prominent interface element was the output LED, while the input devices were more in the background of the experience.

Overall, while the globe might not be the most efficient or accurate way to display this type of data, it has the potential to be engaging and informative.

8.2 Requirement check

The following section evaluates whether the project requirements set in the specification section were met.

Functional requirements:

- The user should be able to select the year from which the data is shown on the globe: This was successfully implemented.
- The user should be able to navigate through data from 1980-2023: Partially achieved. The dataset was limited to 1979-2020 due to constraints mentioned earlier.
- The user should be able to read the data from the globe: Partially achieved. Users could read the data, but the low accuracy scores (less than 50% correct answers) indicate that it was challenging.
- The user should be able to distinguish different levels of temperature on the globe: Successfully achieved with three different output modes (colour, blinking, and a combination of both).
- The user should be able to spin the globe: Successfully achieved.
- The user should be able to know what continent they are looking at: Partially achieved. Users often found this confusing, as indicated by feedback.

- The system should allow for switching between output modes (blinking, colour, blinking and colour): Partially achieved. This functionality was available but could only be controlled by researchers via the server's web interface, not by users.
- The system should allow for making easy to implement changes in the way the variables are mapped and displayed: Successfully achieved. The code is flexible and can accommodate other variables.
- The system should allow for fast updates of the data: Successfully achieved. Data updates were almost instantaneous.

Non-functional requirements:

- The system should have natural interactions: Successfully achieved.
- The system should have embodied control so one can physically interact with the physicalisation: Successfully achieved.
- The system should be intuitive and easy to use: Partially achieved. Users requested labels for the data and legends for the colours to improve intuitiveness.
- The system should be able to be user tested to see the difference between different input and output modes: Successfully achieved. The system was evaluated to compare different input and output modes.

In conclusion, the project successfully met 8 out of 13 requirements, with the remaining five partially addressed, showing progress towards an engaging and interactive climate change visualization tool.

8.3 Recommendations and future work

Based on the limitations and insights gained from the project, the following improvements are recommended to enhance the globe's functionality and user experience.

First, it is recommended to investigate the possibility of controlling both hemispheres with a single ESP to simplify the control system and reduce costs. Although this change would make the design less modular, it is a tradeoff worth considering. Additionally, the clicking mechanism should definitely be featured, as it was the preferred input method among users. This method provides satisfying tactile feedback that significantly enhances the user experience. Or in turn, a slider with detents should be considered. Additionally, the stability of the globe needs improvement. Using a more robust configuration of the bearings or adding an extra arm to hold the top of the globe could address the problems about its flimsy feel.

Improving the representation of continents is another crucial point for better clarity and accuracy in data reading. More detailed outlines or alternative display techniques should be explored to make the continents more distinguishable. Furthermore, blinking modes should either be avoided or implemented with a fixed pattern, as users found the current blinking mode confusing. Regarding the way of presenting the data, anomalies, although they introduce a learning curve, remain the optimal method for representing data with slight and progressive changes, such as climate change. This approach ensures perceivability and helps users to grasp the subtle yet significant shifts in the data. To contribute to user understanding, labels explaining the anomaly concept and legends for the colors used in the display should be added. This addition will help users interpret the data more effectively.

Incorporating additional modalities, such as smell or sound, can create a more memorable and impactful embodied experience. This multisensory interaction can enhance the engagement and educational value of the installation. Moreover, including other datasets and allowing users to switch between them can provide a broader understanding of climate data, making the globe a more versatile tool. Projection models could be incorporated to display not only historical data but also potential future scenarios. This can convey a stronger message about the urgency of addressing climate change. Furthermore, exploring the potential to target other groups, such as children, could make learning about climate change more engaging and accessible for younger audiences.

As for the user evaluation, a bigger sample group would make for more solid and conclusive results, and a better curated set of questions should be developed catered to the limitations of the globe.

By addressing these recommendations and pursuing the outlined future work, the interactive globe can be refined and enhanced to provide a more effective, engaging, and educational tool for visualizing climate change data.

Chapter 9 – Conclusion

The main goal of the research was to investigate the impact of different input modes, like sliders and knobs, on user engagement and comprehension when interacting with an interactive LED globe that visualizes time series climate change data. The research question was:

"What is the impact of different input modes for time series data on user engagement and comprehension of climate change data on an interactive LED globe?"

To achieve this, an interactive LED globe was designed, built, and tested. Temperature anomalies were selected instead of absolute values to more perceptibly show slight changes over the years. This data was displayed over a span of 40 years, from 1980 to 2020, using different output modes such as color and animation. The globe featured three different input modes, each representing a different metaphor for year selection and time series data input.

The aim was to incorporate embodied interactions through year selection and rotation of the globe as natural movements to create a more engaging and memorable experience, thereby making people more aware of climate change.

In the test, 36 participants evaluated the user experience, mental and physical load, satisfaction, accuracy, and efficiency of the system. The results showed no significant differences in performance between the input modes, but users preferred the discrete knob that provided clicking sounds and haptic feedback. Overall, the installation was successful in being engaging and curiosity-triggering, as users gave a lot of positive feedback. However, accuracy should be improved by refining the questions and redesigning the LED placement to have a denser resolution. A larger study would provide stronger conclusions.

The relevance and contribution of this research lie in its innovative approach to data visualization through physical interaction, focusing on time series data, making complex climate change data more accessible and engaging to the general public. This contributes to the fields of data physicalisation by providing new insights into effective ways to present and interact with time series data such as climate data.

In conclusion, the globe installation successfully engages users and sparks curiosity, serving as a powerful conversation starter on the pressing issue of the climate crisis. To reach its full potential, further research and improvements are essential to make it an impactful tool that can educate, engage, and drive meaningful action in the fight against climate change.

References

[1] Y. Jansen *et al.*, ‘Opportunities and Challenges for Data Physicalization’, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, Seoul Republic of Korea: ACM, Apr. 2015, pp. 3227–3236. doi: 10.1145/2702123.2702180.

[2] C. Ranasinghe and A. Degbelo, ‘Encoding Variables, Evaluation Criteria, and Evaluation Methods for Data Physicalisations: A Review’, *Multimodal Technol. Interact.*, vol. 7, no. 7, p. 73, Jul. 2023, doi: 10.3390/mti7070073.

[3] S. Venghaus, M. Henseleit, and M. Belka, ‘The impact of climate change awareness on behavioral changes in Germany: changing minds or changing behavior?’, *Energy Sustain. Soc.*, vol. 12, no. 1, p. 8, Dec. 2022, doi: 10.1186/s13705-022-00334-8.

[4] Colombus, ‘Why Buy a Globe Instead of a World Map?’, Colombus. [Online]. Available: https://www.columbusglobes.com/globes-vs-maps

[5] Mapshop, ‘Maps vs. Globes: The Info You Need to Know’. [Online]. Available: https://www.mapshop.com/your-map-guide-for-maps-vs-globes/

[6] UN, ‘Climate Change “Biggest Threat Modern Humans Have Ever Faced”, World-Renowned Naturalist Tells Security Council, Calls for Greater Global Cooperation’, United Nations Meetings Coverage and Press Releases. [Online]. Available: https://press.un.org/en/2021/sc14445.doc.htm

[7] J. Alexander, P. Isenberg, Y. Jansen, B. E. Rogowitz, and A. Vande Moere, ‘Data Physicalization (Dagstuhl Seminar 18441)’, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi: 10.4230/DAGREP.8.10.127.

[8] E. Hornecker, T. Hogan, U. Hinrichs, and R. Van Koningsbruggen, ‘A Design Vocabulary for Data Physicalization’, *ACM Trans. Comput.-Hum. Interact.*, vol. 31, no. 1, pp. 1–62, Feb. 2024, doi: 10.1145/3617366.

[9] E. Hornecker, ‘A Design Theme for Tangible Interaction: Embodied Facilitation’, in *ECSCW 2005*, H. Gellersen, K. Schmidt, M. Beaudouin-Lafon, and W. Mackay, Eds., Berlin/Heidelberg: Springer-Verlag, 2005, pp. 23–43. doi: 10.1007/1-4020-4023-7_2.

[10] T. Hogan and E. Hornecker, ‘How Does Representation Modality Affect User-Experience of Data Artifacts?’, in *Haptic and Audio Interaction Design*, vol. 7468, C. Magnusson, D. Szymczak, and S. Brewster, Eds., in Lecture Notes in Computer Science, vol. 7468. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 141–151. doi: 10.1007/978-3-642-32796-4_15.

[11] Y. Jansen, P. Dragicevic, and J.-D. Fekete, ‘Evaluating the efficiency of physical visualizations’, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris France: ACM, Apr. 2013, pp. 2593–2602. doi: 10.1145/2470654.2481359.

[12] T. Hogan and E. Hornecker, ‘Towards a Design Space for Multisensory Data Representation’, *Interact. Comput.*, p. iwc;iww015v1, May 2016, doi: 10.1093/iwc/iww015.

[13] S. Dullaert, C. Ranasinghe, A. Degbelo, and N. Bouali, ‘Data Physicalization with Haptic Variables: Exploring Resistance and Friction’, in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, Honolulu HI USA: ACM, May 2024, pp. 1–8. doi: 10.1145/3613905.3651011.

[14] R. Van Loenhout, C. Ranasinghe, A. Degbelo, and N. Bouali, ‘Physicalizing Sustainable Development Goals Data: An Example with SDG 7 (Affordable and Clean Energy)’, in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, New Orleans LA USA: ACM, Apr. 2022, pp. 1–7. doi: 10.1145/3491101.3519638.

[15] S. De Kreij, C. Ranasinghe, and A. Degbelo, ‘Data Physicalization and Tangible Manipulation for Engaging Children with Data: An Example with Air Quality Data’, in *Proceedings of the 23rd Annual ACM Interaction Design and Children Conference*, Delft Netherlands: ACM, Jun. 2024, pp. 507–516. doi: 10.1145/3628516.3655788.

[16] D. Peeters, C. Ranasinghe, A. Degbelo, and F. Ahmed, ‘EmoClock: Communicating Real-Time Emotional States Through Data Physicalizations’, in *Human-Computer Interaction – INTERACT 2023*, vol. 14142, J. Abdelnour Nocera, M. Kristín Lárusdóttir, H. Petrie, A. Piccinno, and M. Winckler, Eds., in Lecture Notes in Computer Science, vol. 14142. , Cham: Springer Nature Switzerland, 2023, pp. 416–425. doi: 10.1007/978-3-031-42280-5_26.

[17] S. Houben *et al.*, 'Physikit: Data Engagement Through Physical Ambient Visualizations in the Home', in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA: ACM, May 2016, pp. 1608–1619. doi: 10.1145/2858036.2858059.
[18] X. Du *et al.*, 'That's Rough! Encoding Data into Roughness for Physicalization', in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Honolulu HI USA: ACM, May 2024, pp. 1–16. doi: 10.1145/3613904.3641900.
[19] NASA, 'Global Climate Change', Eye on The Earth. [Online]. Available: https://climate.nasa.gov/earth-now/#/vital-signs/air-temperature/airs-infrared-surface-3day
[20] NASA, 'NASA's Scientific Visualization Studio'. [Online]. Available: https://climate.nasa.gov/interactives/climate-time-machine/?intent=021
[21] ESA, 'Earth online', Visualization tools. [Online]. Available: https://earth.esa.int/eogateway/search?text=&category=Tools+and+toolboxes&tools_type=visualisation&sortby=RELEVANCE
[22] Amazon, 'LeapFrog Interactive Kids World, Smart Globe for Kids to Learn Geography While Having Fun, Pre School Educational Toys for Kids 5, 6, 7+ Years Elderly'. [Online]. Available: https://www.amazon.nl/-/en/LeapFrog-Interactive-Geography-Educational-Elderly/dp/B07SLF2Q49
[23] Garmantis, '360 DEGREE CLIMATE DOME', Multi-sensory immersive experience installation with VR goggles for trade shows and events. [Online]. Available: https://www.garamantis.com/en/project/360-degree-climate-dome/
[24] D. F. Keefe *et al.*, 'Weather Report: A Site-Specific Artwork Interweaving Human Experiences and Scientific Data Physicalization', *IEEE Comput. Graph. Appl.*, vol. 38, no. 4, pp. 10–16, Jul. 2018, doi: 10.1109/MCG.2018.042731653.
[25] The Elumenati, 'GeoDome Globes'. [Online]. Available: https://www.elumenati.com/product/globe/
[26] A. Mader and W. Eggink, 'A design process for creative technology', *Int. Conf. Eng. Prod. Des. Educ.*, Sep. 2014.
[27] World Meteorological Organization, 'Climate Indicators', Climate Data Catalogue. [Online]. Available: https://climatedata-catalogue-wmo.org/climate_indicators#:~:text=WMO%20uses%20a%20list%20of,Arctic%20%26%20Antarctic%20sea%20ice%20extent.
[28] *LED sphere with 194 LEDs and how to build it*, (Dec. 16, 2019). [YouTube]. Available: https://www.youtube.com/watch?v=Q5d8gTppuYo
[29] *An LED Sphere YOU Can Build*, (Jan. 18, 2024). [YouTube]. Available: https://www.youtube.com/watch?v=l0WGbmc9_4Q&t=1s
[30] *Spherical LED Matrix - Sphere Shaped LED Screen From LED Pixel Strip*, (May 12, 2015). [YouTube]. Available: https://www.youtube.com/watch?v=oP9oEa2yVMQ
[31] *Giant Animated RGB LED Pixel Mega Sphere Ball Kit*, (Aug. 14, 2018). [Online Video]. Available: https://www.youtube.com/watch?v=QivlngKW6wg
[32] Copernicus, 'Climate Pulse'. [Online]. Available: https://pulse.climate.copernicus.eu/
[33] P. Raposo, 'Implementing the QTM Discrete Global Grid System (DGGS)'. ITC Mini Symposium On Sustainable Research Software Development For Geo-Information Science And Earth Observation, Nov. 17, 2022. [Online]. Available: https://ris.utwente.nl/ws/portalfiles/portal/332170331/4-6-QTM-Generator.pdf
[34] Copernicus, 'About Copernicus'. [Online]. Available: https://www.copernicus.eu/en/about-copernicus
[35] Copernicus, 'Copernicus Interactive Climate Atlas'. [Online]. Available: https://atlas.climate.copernicus.eu/atlas
[36] World Meteorological Organization, 'Global Datasets'. [Online]. Available: https://climatedata-catalogue-wmo.org/global-datasets-selector?field_type_of_data_tid=2
[37] World Meteorological Organization, 'Updated 30-year reference period reflects changing climate'. [Online]. Available: https://wmo.int/media/news/updated-30-year-reference-period-reflects-changing-climate
[38] Institute for Environmental Analytic Institute for Environmental Analytics, '#ShowYourStripes'. [Online]. Available: https://showyourstripes.info/

[39] H. Hersbach *et al.*, ‘Essential climate variables for assessment of climate variability from 1979 to present. Copernicus Climate Change Service (C3S) Data Store (CDS).’ 2018. [Online]. Available: https://cds.climate.copernicus.eu/cdsapp#!/dataset/ecv-for-climate-change?tab=form
[40] Express Analytics, ‘Everything You Need to Know About Time Series Analysis – A Primer’. [Online]. Available: https://www.expressanalytics.com/blog/time-series-analysis/
[41] R. B. Lund, C. Beaulieu, R. Killick, Q. Lu, and X. Shi, ‘Good Practices and Common Pitfalls in Climate Time Series Changepoint Techniques: A Review’, *J. Clim.*, vol. 36, no. 23, pp. 8041–8057, Dec. 2023, doi: 10.1175/JCLI-D-22-0954.1.
[42] Climate Atlas of Canada, ‘Interpreting Climate Data’, Climate Atlas. [Online]. Available: https://climateatlas.ca/atlas-guidebook/interpreting-climate-data
[43] M. Marchi, D. Castellanos-Acuña, A. Hamann, T. Wang, D. Ray, and A. Menzel, ‘ClimateEU, scale-free climate normals, historical time series, and future projections for Europe’, *Sci. Data*, vol. 7, no. 1, p. 428, Dec. 2020, doi: 10.1038/s41597-020-00763-0.
[44] NOAA, ‘U.S. Climate Normals’, National Centers for Environmetal Information. [Online]. Available: https://www.ncei.noaa.gov/products/land-based-station/us-climate-normals
[45] World Meteorological Organization, ‘WMO Global Annual to Decadal Climate Update 2024-2028’. [Online]. Available: https://library.wmo.int/viewer/68910/download?file=WMO_GADCU_2024-2028_en.pdf&type=pdf&navigator=1

Appendix

Appendix A – Consent form

Information letter for Data physicalization of climate change data using an interactive globe

Dear reader,

You have been recently approached for participation in the research project entitled "Data physicalization of climate change data using an interactive globe". By means of this information brochure, we would like to inform you about this research and kindly ask you for your participation. You are free to decide whether you wish to participate. Before you decide to participate, it is important to know more about the research project. Please read the information in this brochure carefully and, if needed, contact the researcher for any questions that you may have. If you agree to participate, the researcher will make an appointment with you for the study. The entire interview will take around thirty minutes to complete.

What is the aim of the research?

The urgency of addressing climate change demands innovative approaches to communicate its complexities effectively to the general public. People have made visualisations in many forms like 2d charts or artistic physicalizations of climate change. Visualisations in AR and VR have also been explored, but its key issue is that it is only visible for a single user. These visualizations limit the embodied interaction a user can experience in comparison to a physicalization of climate change data.

Physical spheres, a common geometric shape visible from all areas, might provide a solution for communicating climate change data in a more embodied way. The main goal will be to effectively communicate the urgency of climate change that is accessible and usable by the general public. Therefore, this paper will be focused on how climate change data can be represented in a spherical form using lights or LED's so that it has a triggering impact on the general public audience.

This user testing aims to investigate the usability, engagement, and understanding of the installation.

The research project is conducted in the context of the ongoing graduation project of the Creative Technology bachelor curriculum at the University of Twente.

How will the research be executed?

In the proposed research project, entitled "Data physicalization of climate change data using an interactive globe", you will participate in a user test. In this user test we aim to get insight from users on the usability, engagement, and understanding of the installation.

The subject will perform tasks interacting with the installation, prompted by questions on a questionnaire they will fill in regarding the data displayed. The answers on the questionnaire and observations made by the researchers will be gathered and processed afterwards. The session will not be recorded

UNIVERSITY OF TWENTE.

What are the possible benefits or risks of participation?
Participation in this research project does not provide any direct benefits for you. However, your contribution may lead to very valuable data to improve the installation. As for risks, this research project does not have risks for you, physically or psychologically. This research project has been reviewed by the Ethics Committee Information and Computer Science.

What happens if you do not wish to participate in this research?
You may voluntarily decide whether you wish to participate in this research project. If you decide not to participate, you do not need to take any action. You also do not have to provide a reason for not wanting to participate in the research project. If you do decide to participate, you still may, at any moment of the research project, decide to stop your participation. Any data gathered before the withdrawal of participation will be deleted.

What will be done with your data?
If you participate in the study, your data will be gathered. Your data may only be used for the study and will never be publicly released. You will not be identifiable in any publication or presentation about the study. Personal data, including age, gender, and profession, will only serve for demographical purposes and will not be used to identify you. You will get a copy of your answers once we have processed the information, so you can correct any interpretation mistakes. Your data and audio recordings will be deleted once the project is finished.

Where can you ask any other questions?
Before you participate in this study, you have the opportunity to ask any questions you may have.
If you have any questions about or issues with the study, please do not hesitate to contact the researchers (see contact information below). Please also find, accompanied to this study brochure, the informed consent that you will sign before the start of the experiment.

Thank you in advance for your cooperation in this study.
Yours sincerely,
María Cobo Muñoz and Luka van Hoeven

Creative Technology
Faculty of EEMCS
University of Twente

UNIVERSITY OF TWENTE.

Consent Form for Data physicalization of climate change data using an interactive globe

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated [DD/MM/YYYY], or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	◻	◻
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	◻	◻
I understand that taking part in the study involves that the interview will be audio recorded and saved. It will be deleted at the completion of the research or when requested. We will not transcribe the interview.	◻	◻
Use of the information in the study		
I understand that information I provide will be used for the user evaluation of the installation.	◻	◻
I understand that personal information collected about me that can identify me, such as [e.g. my name or where I live], will not be shared beyond the study team.	◻	◻
I agree that my information can be quoted in research outputs.	◻	◻
Future use and reuse of the information by others		
I give permission for the corresponding report to be archived in Mobility online so it can be used for future research and learning.	◻	◻
I give the researchers permission to keep my contact information and to contact me for future research projects.	◻	◻

Signatures

______________________ ______________________ _________

Name of participant Signature Date

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

_________________________ ___________________ _________

Researcher name Signature Date

UNIVERSITY OF TWENTE.

Study contact details for further information:
Luka van Hoeven (l.k.vanhoeven@student.utwente.nl),
María Cobo Muñoz (m.cobomunoz@student.utwente.nl)

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl

UNIVERSITY OF TWENTE.

Appendix B – User test questionnaire

LimeSurvey

Section A: To be filled in by the researchers

A1. Participant number

Section B: Demographics

B1. What is your age?

B2. What is your gender?

Female ☐
Male ☐
Non-binary ☐

B3. Are you colorblind?

Yes ☐
No ☐

B4. How would you rate your knowledge about climate change?

No knowledge ☐ Beginner ☐ Somewhat knowledgab le ☐ Knowledga ble ☐ Expert ☐

B5. How familiar are you with data physicalizations?

Not at all ☐ A little bit ☐ Somewhat ☐ Very ☐

B6. How familiar are you with reading geographical data? (How good you are reading maps, how often, etc.)

Not at all ☐ A little bit ☐ Somewhat ☐ Very ☐

LimeSurvey

Section C: Exploration

Feel free to expore the functionality of the globe, once you feel comfortable go to the next part of the survey where you complete tasks related to the data being shown. Use this time to familiarize yourself with the globe. Once you feel comfortable press next.

Section D: Tasks

You will be asked questions about the data that is being shown

D1. In the last decade (2010-2020) has Africa become more warm or more cool in general?

- Warmer ☐
- Cooler ☐

D2. In 1985, what hemisphere has the biggest temperature change, southern or northern? Keep in mind that change includes warmer as well as colder temperatures.

- Northern hemisphere ☐
- Southern hemisphere ☐

D3. What area has the highest temperature anomaly in 2000? (Warmer)

- North pole ☐
- North America ☐
- Africa ☐

D4. Is the equator warmer in 1990 compared to 2020?

- Yes ☐
- No ☐

Section E: Evaluation

E1. Answer the following questions about satisfaction

	Strongly disagree						Strongly agree
Overall, I am satisfied with the ease of completing these tasks.	☐	☐	☐	☐	☐	☐	☐
Overall, I am satisfied with the amount of time it took to complete these tasks.	☐	☐	☐	☐	☐	☐	☐

E2. Answer the following questions about mental and physical demand.

	Very low						Very high
How mentally demanding was these tasks?	☐	☐	☐	☐	☐	☐	☐
How physically demanding was these tasks?	☐	☐	☐	☐	☐	☐	☐

LimeSurvey

E3. **Fill in the following in regards to your experience with the physicalization**

	1	2	3	4	5	6	7
obstructive | supportive	☐	☐	☐	☐	☐	☐	☐
complicated | easy	☐	☐	☐	☐	☐	☐	☐
inefficient | efficient	☐	☐	☐	☐	☐	☐	☐
confusing | clear	☐	☐	☐	☐	☐	☐	☐
boring | exciting	☐	☐	☐	☐	☐	☐	☐
not interesting | interesting	☐	☐	☐	☐	☐	☐	☐
conventional | inventive	☐	☐	☐	☐	☐	☐	☐
usual | leading edge	☐	☐	☐	☐	☐	☐	☐
	☐	☐	☐	☐	☐	☐	☐

E4. **Please fill in any questions, remarks or feedback about the physicalization and your experience.**

Appendix C – Code lines

Server code: app

```
from flask import Flask, render_template
from flask_sock import Sock
import threading
from functions import *
import numpy as np
import time
import re

app = Flask(__name__)

sock = Sock(app)

global last_access_time, min_temp, max_temp, bins_and_frequencies, color_bins,
colors, max_brightness, slider_active, knob_1_active, knob_2_active,
sphere_animation, sphere_color, animation_color
min_temp = -3
max_temp = 3
bins_and_frequencies: List[Tuple[int, int, int]] = [(2, 3, 20),(0.5, 2, 60),(0,
0.5, 100)]
color_bins = [(-3,-2),(-2, -0.5),(-0.5, 0.5),(0.5, 2),(2,3)]
colors = [
    (0, 0, 255),
    (50, 50, 150),   # Cyan
    (50, 50, 40),  # Orange
    (75, 75, 0),
    (255, 0, 0),  # Orange
    ]
max_brightness = 25
slider_active = False
knob_1_active = False
knob_2_active = False
sphere_animation = True
sphere_color = True
animation_color = (50,50,50)

RATE_LIMIT = 0.050
last_access_time = 0
# Store active WebSocket connections
connections = {}
sphere_data = get_detail()

@app.route("/")
def hello_world():
    return render_template('index.html')

@sock.route('/north')
def send(wsSend):
    connections['north'] = wsSend
    try:
        while True:
            data = wsSend.receive()
            print(data)
    except Exception as e:
        print(f"Connection error: {e}")
```

```python
@sock.route('/south')
def sendSouth(wsSend):
    connections['south'] = wsSend
    try:
        while True:
            data = wsSend.receive()
            print(data)
    except Exception as e:
        print(f"Connection error: {e}")

@sock.route('/rgb')
def sendRGB(wsSend):
    connections['rgb'] = wsSend
    try:
        while True:
            data = wsSend.receive()
            if data:
                print(f"Received data: {data}")
                try:
                    rgb_values = re.findall(r'\d+', data)
                    r, g, b = map(int, rgb_values)

                    detail = sphere_data['2006']
                    limited = limit_data(detail, -3, 3)
                    sphere_colors = make_one_color(limited, 25, r, g, b)
                    concat_array = get_concatenated_array(sphere_colors)
                    north, south = return_hemispheres(concat_array)
                    print(north[:10])
                    print(south[:10])
                    north_bytes = bytes(north)
                    south_bytes = bytes(south)
                    threading.Thread(target=broadcast, args=(south_bytes,
'south')).start()
                    threading.Thread(target=broadcast, args=(north_bytes,
'north')).start()
                except ValueError as e:
                    print(f"Error parsing RGB values: {e}")
            else:
                break

            # print(data['r'], data['g'], data['b'])
            # recieved = wsSend.receive()
    except Exception as e:
        print(f"Connection error: {e}")

@sock.route('/settings')
def setSettings(wsSend):
    global min_temp, max_temp, bins_and_frequencies, color_bins, colors,
max_brightness, slider_active, knob_1_active, knob_2_active, sphere_animation,
sphere_color, animation_color
    connections['settings'] = wsSend
    try:
        while True:
            received = wsSend.receive()
            if received:
                try:
                    settings = json.loads(received)

                    min_temp = settings['min']
```

```
                    max_temp = settings['max']
                    bins_and_frequencies = settings['bins_and_frequencies']
                    colors = settings['colors']
                    max_brightness = settings['max_brightness']
                    color_bins = settings['color_bins']
                    slider_active = settings['slider_active']
                    knob_1_active = settings['knob_1_active']
                    knob_2_active = settings['knob_2_active']
                    sphere_animation = settings['sphere_animation']
                    sphere_color = settings['sphere_color']
                    animation_color = tuple(settings['animation_color'][0])
                    # animation_color = (animation_color, animation_color[1],
animation_color[2])

                    colors = [tuple(color) for color in settings['colors']]
                    color_bins = [tuple(bin) for bin in settings['color_bins']]
                    bins_and_frequencies = [tuple(bins) for bins in
settings['bins_and_frequencies']]

                except ValueError as e:
                    print(f"Error parsing settings: {e}")
            else:
                break
    except Exception as e:
        print(f"Connection error: {e}")

@sock.route('/input1')
def sendInput(wsSend):
    connections['input1'] = wsSend
    try:
        while True:
            recieved = wsSend.receive()
            if recieved:
                global last_access_time
                current_time = time.time()  # Convert to milliseconds
                try:
                    if current_time - last_access_time < RATE_LIMIT:
                        # print('skip')
                        continue  # Skip the rest of the loop
                    # print('not skippng')

                    last_access_time = current_time
                    sendSphereData(recieved)


                    # threading.Thread(target=broadcast, args=(south_bytes,
'south')).start()
                    # threading.Thread(target=broadcast, args=(north_bytes,
'north')).start()
                except ValueError as e:
                    print(f"Error parsing RGB values: {e}")
            else:
                break
    except Exception as e:
        print(f"Connection error: {e}")

@sock.route('/controller')
def sendYear(wsSend):
    connections['controller'] = wsSend
```

```
    try:
        while True:
            recieved = wsSend.receive()
            if recieved:
                global last_access_time, knob_1_active, knob_2_active,
slider_active
                current_time = time.time()  # Convert to milliseconds
                try:


                    last_access_time = current_time
                    parsed = json.loads(recieved)
                    slider = parsed['slider']
                    knob1 = parsed['knob1']
                    knob2 = parsed['knob2']
                    if knob_1_active:
                        sendSphereData(knob1)
                    elif knob_2_active:
                        sendSphereData(knob2)
                    elif slider_active:
                        sendSphereData(slider)

                except ValueError as e:
                    print(f"Error parsing RGB values: {e}")

            else:
                break
    except Exception as e:
        print(f"Connection error: {e}")

def broadcast(message, client):
    try:
        connections[client].send(message)
    except Exception as e:
        print(f"Error sending message: {e}")

def sendSphereData(recieved):
    detail = sphere_data[str(recieved)]
    normalize = limit_data(detail, min_temp, max_temp)
    sphere_colors = []
    if sphere_animation and sphere_color:
        sphere_colors = map_data_to_colors_and_frequency(normalize,colors,
color_bins, max_brightness, min_temp, max_temp, bins_and_frequencies)
    elif sphere_animation and not sphere_color:
        sphere_colors = map_data_to_frequency(normalize,
max_brightness,bins_and_frequencies, animation_color, min_temp, max_temp)
    elif not sphere_animation and sphere_color:
        sphere_colors = map_data_to_colors(normalize, color_bins, colors,
max_brightness)
    elif not sphere_animation and not sphere_color:
        sphere_colors = get_geography(25)

    north, south = get_hemispheres(sphere_colors)
    # print(north)
    # print("North",north[:100], "South",south[:100])
    # print("North",north[3864:], "South",south[3864:])
    # print("leng", len(north), len(south))

    north_bytes = bytes(north)
```

```
    south_bytes = bytes(south)
    broadcast(south_bytes, 'south')
    broadcast(north_bytes, 'north')
```

Server code: functions

```
import json
import numpy as np
import os
import matplotlib.pyplot as plt
from matplotlib.widgets import Slider
from matplotlib.colors import LinearSegmentedColormap
import itertools
from typing import List, Dict, Tuple
import matplotlib.colors as mcolors

def get_detail():
    all_detail = {}
    # Get the list of all JSON files in the sphere_detail_output folder
    json_files = [f for f in os.listdir('sphere_detail_output') if
f.endswith('.json')]

    # Append the contents of each JSON file to the all_detail dictionary
    for file in json_files:
        year = file.split('_')[2].split('.')[0]
        with open(os.path.join('sphere_detail_output', file), 'r') as f:
            detail = json.load(f)
            all_detail[year] = detail
    return all_detail

def get_geography(max_brightness):
    detail = {}
    with open(os.path.join('mcont', 'continents.json'), 'r') as f:
        detail = json.load(f)
    rgb_data = {}
    for key in detail:
        rgb_data[key] = []
        for value in detail[key]:
            # Convert normalized value to RGB color
            rgba = tuple(value)
            rgb = [int(max_brightness * c) for c in rgba]
            rgb.append(int(0))
            rgb_data[key].extend(rgb)
    return rgb_data


# Update function for slider
def limit_data(detail_from_year, min, max):
    for key in detail_from_year:
        for i in range(len(detail_from_year[key])):
            if detail_from_year[key][i] > max:
                detail_from_year[key][i] = max
            elif detail_from_year[key][i] < min:
                detail_from_year[key][i] = min
            # print(detail_from_year[key][i], "Normalized:",
(detail_from_year[key][i] - min) / (max - min))
```

```
            # detail_from_year[key][i] = (detail_from_year[key][i] - min) / (max -
min)
        # detail_from_year[key] = np.clip(detail_from_year, 0, 1)
    return detail_from_year

def get_frequency(value, min, max, bins_and_frequency: List[Tuple[int, int,
int]]):
    distance_from_middle = abs(value)
    if distance_from_middle > max:
        distance_from_middle = max
    if distance_from_middle < 0:
        distance_from_middle = 0
    bins = [(x[0], x[1]) for x in bins_and_frequency]
    frequency_per_bin = [x[2] for x in bins_and_frequency] #Divide by 100, since
we are multiplying with 100 in the end
    for index, (bin_start, bin_end) in enumerate(bins):
        if bin_start <= distance_from_middle <= bin_end:
            return frequency_per_bin[index]
    return frequency_per_bin[len(frequency_per_bin)-1]

def get_color(value, bins, colors):
    for index, (bin_start, bin_end) in enumerate(bins):
        if bin_start <= value < bin_end:
            return colors[index]
    return colors[-1]

def map_data_to_colors_and_frequency(clipped_data, colors, color_bins,
max_brightness, min, max, bins_and_frequency: List[Tuple[int, int, int]]):
    rgb_data = {}
    for key in clipped_data:
        rgb_data[key] = []
        for value in clipped_data[key]:
            # Convert normalized value to RGB color
            rgba = get_color(value, color_bins, colors)
            rgb = [int(max_brightness * c/255) for c in rgba]  # Take the RGB
values and scale to 0-255
            rgb.append(int(get_frequency(value, min, max, bins_and_frequency)))  #
Add the alpha value to the list (for frequency of the color)
            rgb_data[key].extend(rgb)
    return rgb_data

def map_data_to_colors(clipped_data, color_bins, color, max_brightness):
    rgb_data = {}
    for key in clipped_data:
        rgb_data[key] = []
        for value in clipped_data[key]:
            # Convert normalized value to RGB color
            rgba = get_color(value, color_bins, color)
            rgb = [int(max_brightness * c/255) for c in rgba]
            rgb.append(int(0))
            rgb_data[key].extend(rgb)
    return rgb_data

def map_data_to_frequency(clipped_data, max_brightness, bins_and_frequency:
List[Tuple[int, int, int]], animation_color: Tuple[int, int, int], min_temp,
max_temp):
    rgb_data = {}
    for key in clipped_data:
        rgb_data[key] = []
```

```
        for value in clipped_data[key]:
            # Convert normalized value to RGB color
            rgba = animation_color
            if value < 0:
                rgba = (0, 0, 255)
            elif value >= 0:
                rgba = (255, 0, 0)
            rgb = [int(max_brightness * c/255) for c in rgba]
            rgb.append(int(get_frequency(value,min_temp, max_temp,
bins_and_frequency)))
            rgb_data[key].extend(rgb)
    return rgb_data

def make_one_color(clipped_data, max_brightness, r, g, b):
    rgb_data = {}

    for key in clipped_data:
        rgb_data[key] = []
        for value in clipped_data[key]:
            # Convert normalized value to RGB color
            rgba = (r/255, g/255, b/255)
            rgb = [int(max_brightness * c) for c in rgba]  # Take the RGB values
and scale to 0-255
            distance_from_middle = abs(value - 0.5)
            # rgb.append(int(distance_from_middle * 255))  # Add the alpha value
to the list (for frequencty of the color)
            rgb.append(int(0))
            rgb_data[key].extend(rgb)

    return rgb_data



def create_red_to_blue_cmap(colors):
    colors1 = [
    (0, (5/255, 0, 213/255)),
    (0.08, (0, 40/255, 214/255)),   # Dark Blue
    (0.16, (0, 80/255, 214/255)),  # Blue
    (0.24, (0, 130/255, 214/255)),  # Cyan
    (0.32, (62/255, 142/255, 190/255)),  # White
    (0.42, (89/255, 143/255, 177/255)),  # Yellow
    (0.5, (107/255, 107/255, 107/255)),  # Orange
    (0.58, (255/255, 85/255, 13/255)),       # Red
    (0.68, (255/255, 80/255, 0)),  # Blue
    (0.76, (255/255, 50/255, 0)),  # Cyan
    (0.84, (255/255, 30/255, 0/255)),  # White
    (0.92, (255/255, 20/255, 0)),  # Yellow
    (1, (255/255, 0, 0)),  # Orange
    ]

    # colors = [
    # (0, (5/255, 0, 213/255)),
    # (0.24, (0, 130/255, 214/255)),  # Cyan
    # (0.5, (107/255, 107/255, 107/255)),  # Orange
    # (0.75, (255/255, 50/255, 0)),
    # (1, (255/255, 0, 0)),  # Orange
    # ]

# Create the colormap
```

```python
    cmap = mcolors.LinearSegmentedColormap.from_list('discrete_color', colors,
N=len(colors))


    return cmap

def get_concatenated_array(colordata):
    concatenated_array = sum(colordata.values(), [])
    return concatenated_array

def get_hemispheres(colordata):
    north_hemisphere = []
    south_hemisphere = []

    # Ensure that keys are sorted as integers to maintain order
    sorted_keys = sorted(colordata.keys(), key=lambda x: int(x))

    for key in sorted_keys:
        if 1 <= int(key) <= 15:
            north_hemisphere.extend(colordata[key])

    for key in sorted_keys[::-1]:  # Iterate over sorted keys in reverse order
        if 16 <= int(key) <= 30:
            south_hemisphere.extend(colordata[key])

    return north_hemisphere, south_hemisphere

def return_hemispheres(concat_list):
    total_rows = len(concat_list)
    mid_index = total_rows // 2  # Integer division for midpoint

    # Handle odd number of rows by adding one extra to the northern hemisphere
    if total_rows % 2 != 0:
        mid_index += 1

    south = concat_list[:mid_index]
    north = concat_list[mid_index:]

    return north, south


# detail = get_detail()['2005']
# normalize = normalize_data(detail, -3, 3)
# colors = map_data_to_colors(normalize, create_red_to_blue_cmap(), 50)
# concat_array = get_concatenated_array(colors)
# north, south = return_hemispheres(concat_array)
```

Html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Globey</title>
    <link rel="stylesheet" href="{{url_for('static', filename='global.css')}}"
type="text/css">
```

```
    <script src="{{url_for('static', filename='script.js')}}"
type="text/javascript"></script>
  </head>
  <body>
    <h1>Globey</h1>
    <span id="sliderValue">1979</span>
    <input style="display: inline-block; width: 70%;" type="range"
id="messageInput" min="1979" max="2023" step="4" value="1979">
    <input type="color" id="colorPicker" name="head" value="#e66465" />
    <div>
      <form id="dataForm">
        <label for="min">Min:</label>
        <input type="number" id="min" name="min" value="-3"><br><br>

        <label for="max">Max:</label>
        <input type="number" id="max" name="max" value="3"><br><br>

        <label for="bins_and_frequencies">Bins and Frequencies (format:
[0,1,10],[1,2,20],[2,3,30]):</label>
        <input type="text" id="bins_and_frequencies" name="bins_and_frequencies"
value="[0,0.5,100],[0.5,2,60],[2,3,20]"><br><br>

        <label for="colors">Colors (format:
[255,255,255],[255,0,0],[0,0,255]):</label>
        <input type="text" id="colors" name="colors" value="[0, 0, 255],[50, 50,
150],[50, 50, 40],[75, 75, 0],[255, 0, 0]">
        <br><br>

        <label for="bins">Bins (format: [-3,-2],[-2,-1]):</label>
        <input type="text" id="bins" name="bins" value="[-3,-2],[-2, -0.5],[-0.5,
0.5],[0.5, 2],[2,3]"><br><br>

        <label for="max_brightness">Max Brightness:</label>
        <input type="number" id="max_brightness" name="max_brightness"
value="25"><br><br>

        <label for="slider_active">Slider Active:</label>
        <input type="checkbox" id="slider_active" name="slider_active"><br><br>

        <label for="knob1_active">Knob 1 Active:</label>
        <input type="checkbox" id="knob1_active" name="knob1_active"
checked><br><br>

        <label for="knob2_active">Knob 2 Active:</label>
        <input type="checkbox" id="knob2_active" name="knob2_active"><br><br>

        <label for="animation">Animation:</label>
        <input type="checkbox" id="animation" name="animation" checked><br><br>

        <label for="animation_color">Animation color (format:
[255,255,255]):</label>
        <input type="text" id="animation_color" name="animation_color"
value="[50,50,50]"><br><br>

        <label for="color">Color:</label>
        <input type="checkbox" id="color" name="color" checked><br><br>

        <button type="button" id="sendForm">Send</button>
    </form>
```

```
    </div>
</body>
</html>
```

Javascript

```
const sendSocket = new WebSocket("ws://" + window.location.host + "/input1");
const settingsSocket = new WebSocket("ws://" + window.location.host +
"/settings");
const rgbSocket = new WebSocket("ws://" + window.location.host + "/rgb");
document.addEventListener("DOMContentLoaded", function () {
    // const sendRGB = new WebSocket("ws://" + window.location.host + "/rgb");

    sendSocket.onmessage = function(event) {
        const messageDisplay = document.getElementById("messages");
        const messageElement = document.createElement("div");
        messageElement.textContent = "Received: " + event.data;
        messageDisplay.appendChild(messageElement);
    };

    document.getElementById("colorPicker").oninput = function() {
        const hexValue = document.getElementById("colorPicker").value;
        const r = parseInt(hexValue.substring(1, 3), 16);
        const g = parseInt(hexValue.substring(3, 5), 16);
        const b = parseInt(hexValue.substring(5, 7), 16);
        const colorData = `r${r}g${g}b${b}`;
        console.log(colorData);
        rgbSocket.send(colorData);
    };


    document.getElementById("messageInput").oninput = function() {
        const message = document.getElementById("messageInput").value;
        document.getElementById("sliderValue").textContent = message;
        console.log("Slider changed to: " + message);
        sendSocket.send(message);
    };

    function combineColorsAndBins(colors, bins) {
        return colors.map((color, index) => {
            return { color: color, bin: bins[index] };
        });
    }

    document.getElementById("sendForm").onclick = function() {
        const data = {
            min: parseFloat(document.getElementById('min').value),
            max: parseFloat(document.getElementById('max').value),
            bins_and_frequencies:
parseArrayInput(document.getElementById('bins_and_frequencies').value),
            colors: parseArrayInput(document.getElementById('colors').value),
            color_bins: parseArrayInput(document.getElementById('bins').value),
            max_brightness:
parseFloat(document.getElementById('max_brightness').value),
            slider_active: document.getElementById('slider_active').checked,
            knob_1_active: document.getElementById('knob1_active').checked,
            knob_2_active: document.getElementById('knob2_active').checked,
```

```
            animation_color: 
parseArrayInput(document.getElementById('animation_color').value),
            sphere_animation: document.getElementById('animation').checked,
            sphere_color: document.getElementById('color').checked
        };

        settingsSocket.send(JSON.stringify(data));
    };

    function parseArrayInput(input) {
        return input.split('],[').map(item => item.replace(/\[|\]/g, 
'').split(',').map(Number));
    }
});
```

Json dump code

```
import numpy as np
import xarray as xr
import matplotlib.pyplot as plt
import cartopy.feature as cfeature
import cartopy.io.shapereader as shpreader
from shapely.geometry import Point
from shapely.prepared import prep
import geopandas as gpd
import os
import json



variable_name = 't2m'  # Example: 2 meter temperature
file_path = 
'C:\\Users\\lukav\\Downloads\\yearTemperature\\12month_anomaly_Global_ea_2t_202001
-202012_1981-2010_v02.grib'
ds = xr.open_dataset(file_path, engine='cfgrib')[variable_name]

sphere_detail_amount = [21, 36, 46, 54, 60, 65, 69, 73, 76, 78, 80, 82, 83, 84, 
84, 84, 84, 83, 82, 80, 78, 76, 73, 69, 65, 60, 54, 46, 36, 21]
sphere_detail = {i: [(0, 0, 0) for _ in range(sphere_detail_amount[i-1])] for i in 
range(1, len(sphere_detail_amount)+1)}

def calculate_equal_area_latitude_bins(num_rings):
    latitudes = np.linspace(90, -90, num_rings + 1)
    return [(latitudes[i], latitudes[i+1]) for i in range(num_rings)]

def calculate_longitude_bins(num_leds):
    longitudes = np.linspace(180, -180, num_leds + 1)
    return [(longitudes[i], longitudes[i+1]) for i in range(num_leds)]



def map_data(ds):
    sphere_detail = {i: [0 for _ in range(sphere_detail_amount[i-1])] for i in 
range(1, len(sphere_detail_amount)+1)}
    latitude_bins = calculate_equal_area_latitude_bins(len(sphere_detail_amount))
```

```
    shapefile = shpreader.natural_earth(resolution='50m', category='physical',
name='land')
    land_gdf = gpd.read_file(shapefile)

    def is_land(lat, lon):
        point = Point(lon, lat)
        return any(land_gdf.contains(point))

    for lat_idx, (lat_start, lat_end) in enumerate(latitude_bins):
        lat_slice = ds.sel(latitude=slice(lat_start, lat_end))
        num_leds = sphere_detail_amount[lat_idx]
        longitude_bins = calculate_longitude_bins(num_leds)
        longitude_bins = list(reversed(longitude_bins))
        # Iterate over each longitude bin for the current latitude
        for lon_idx, (lon_start, lon_end) in enumerate(longitude_bins):
            lon_slice = lat_slice.sel(longitude=slice(lon_start, lon_end))

            # interpolated_slice = lon_slice.interp(latitude=lat_slice.latitude,
longitude=lon_slice.longitude, method='cubic')

            # Check if the center of the bin is land or ocean
            center_lat = (lat_start + lat_end) / 2
            center_lon = (lon_start + lon_end) / 2
            if is_land(center_lat, center_lon):
                sphere_detail[lat_idx + 1][lon_idx] = (1, 1, 0)  # Yellow for land
            else:
                sphere_detail[lat_idx + 1][lon_idx] = (0, 0, 1)  # Blue for ocean
    return sphere_detail

max_length = max(sphere_detail_amount)

# Initialize an empty array to store the colors for the plot
output_folder =
'c:\\Users\\lukav\\Documents\\graduationScripts\\sphere_detail_output'

ds = xr.open_dataset(file_path, engine='cfgrib')[variable_name]

# Calculate the sphere_detail for the dataset
sphere_detail = map_data(ds)

# Create the output file path
output_file_path = os.path.join(output_folder, f'continents.json')

# Dump the sphere_detail to a JSON file
with open(output_file_path, 'w') as f:
    json.dump(sphere_detail, f)
```

Data conversion code

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from scipy.interpolate import griddata
import numpy as np
import xarray as xr

variable_name = 't2m'  # Example: 2 meter temperature
```

```
file_path =
'C:\\Users\\lukav\\Downloads\\yearTemperature\\12month_anomaly_Global_ea_2t_198101
-198112_1981-2010_v02.grib'
ds = xr.open_dataset(file_path, engine='cfgrib')[variable_name]

def calculate_equal_area_latitude_bins(num_rings):
    sin_latitudes = np.linspace(1, -1, num_rings + 1)
    latitudes = np.degrees(np.arcsin(sin_latitudes))
    return [(latitudes[i], latitudes[i+1]) for i in range(num_rings)]

def calculate_longitude_bins(num_leds):
    longitudes = np.linspace(0, 360, num_leds + 1)
    return [(longitudes[i], longitudes[i+1]) for i in range(num_leds)]

def interpolate_data(xarray_data, target_latitudes, target_longitudes):
    lat_vals = xarray_data.latitude.values
    lon_vals = xarray_data.longitude.values
    data_vals = xarray_data.values

    lon_grid, lat_grid = np.meshgrid(target_longitudes, target_latitudes)
    points = np.array([(lat, lon) for lat in lat_vals for lon in lon_vals])
    values = data_vals.flatten()

    grid_z = griddata(points, values, (lat_grid, lon_grid), method='linear')
    return grid_z

def map_data_to_leds(xarray_data, ring_led_counts):
    num_rings = len(ring_led_counts)
    latitude_bins = calculate_equal_area_latitude_bins(num_rings)

    target_latitudes = [(lat_start + lat_end) / 2 for lat_start, lat_end in
latitude_bins]
    target_longitudes = np.concatenate([np.linspace(0, 360, count, endpoint=False)
for count in ring_led_counts])

    interpolated_data = interpolate_data(xarray_data, target_latitudes,
np.unique(target_longitudes))

    led_mapping = {}

    for i, (lat_start, lat_end) in enumerate(latitude_bins):
        ring_led_count = ring_led_counts[i]
        longitude_bins = calculate_longitude_bins(ring_led_count)

        for j, (lon_start, lon_end) in enumerate(longitude_bins):
            # Get indices of longitudes within the current bin
            lon_indices = np.where((target_longitudes >= lon_start) &
(target_longitudes < lon_end))[0]

            # Ensure indices are within bounds
            lon_indices = lon_indices[lon_indices < interpolated_data.shape[1]]

            if len(lon_indices) > 0:
                avg_value = interpolated_data[i, lon_indices].mean()
            else:
                avg_value = np.nan

            led_mapping[(i, j)] = avg_value
```

```
    return led_mapping


def plot_led_sphere(led_mapping, ring_led_counts):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')

    for (i, j), value in led_mapping.items():
        lat_bin = calculate_equal_area_latitude_bins(len(ring_led_counts))[i]
        lon_bin = calculate_longitude_bins(ring_led_counts[i])[j]

        lat = (lat_bin[0] + lat_bin[1]) / 2
        lon = (lon_bin[0] + lon_bin[1]) / 2

        lat_rad = np.radians(lat)
        lon_rad = np.radians(lon)

        x = np.cos(lat_rad) * np.cos(lon_rad)
        y = np.cos(lat_rad) * np.sin(lon_rad)
        z = np.sin(lat_rad)

        ax.scatter(x, y, z, color=plt.cm.viridis(value))

    plt.show()

# Example usage with an xarray dataset `ds` and given ring LED counts:
ring_led_counts = [21, 36, 46, 54, 60, 65, 69, 73, 76, 78, 80, 82, 83, 84, 84]
led_mapping = map_data_to_leds(ds, ring_led_counts)
plot_led_sphere(led_mapping, ring_led_counts)
```

Plotting the raw data

```
import os
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import xarray as xr
from matplotlib.widgets import Slider
import matplotlib.colors as mcolors

# Path to the directory containing your GRIB files
directory_path = 'C:\\Users\\lukav\\Downloads\\yearTemperature'


# Variable you want to plot
variable_name = 't2m'  # Example: 2 meter temperature

# Get a sorted list of GRIB files
grib_files = sorted([f for f in os.listdir(directory_path) if
f.endswith('.grib')])

# Extract years from filenames and create a dictionary mapping
file_years = {int(f.split('_')[5][:4]): f for f in grib_files if
f.endswith('.grib')}

# Create a figure and axis for the plot
```

```python
fig, ax = plt.subplots(figsize=(10, 5), subplot_kw={'projection':
ccrs.Mollweide()})

# Dictionary to store preloaded data
preloaded_data = {}

# Preload data for all years
all_data = []
for year, file_name in file_years.items():
    file_path = os.path.join(directory_path, file_name)
    ds = xr.open_dataset(file_path, engine='cfgrib')
    preloaded_data[year] = ds[variable_name]
    all_data.append(ds[variable_name])
    ds.close()


# Concatenate all data along the time dimension
all_data = xr.concat(all_data, dim='time')
vmin = all_data.min()
vmax = all_data.max()
colors = [(0, 'red'),(0.5, 'white'), (1, 'blue')]  # Positions and colors
cmap = mcolors.LinearSegmentedColormap.from_list('blue_white_red', colors)

# Create initial plot
data = preloaded_data[min(file_years.keys())]
lats = data.latitude.values
lons = data.longitude.values
plot = ax.imshow(data, origin='lower', extent=(lons.min(), lons.max(), lats.min(),
lats.max()),
                 transform=ccrs.PlateCarree(), cmap=cmap, vmin=-3, vmax=3)
ax.coastlines()
ax.add_feature(cfeature.BORDERS)

cbar = plt.colorbar(plot, orientation='horizontal', pad=0.05, ax=ax)
cbar.set_label(variable_name)
ax.set_title(f'{variable_name} for the year {min(file_years.keys())}')

# Function to update the plot for a given year
def update_plot(val):
    year = int(slider.val)

    # Get the data for the selected year from preloaded data
    data = preloaded_data[year]
    # Update contour levels with new data
    plot.set_data(data.values)
    # Update title
    ax.set_title(f'{variable_name} for the year {year}')

    # Redraw the figure
    fig.canvas.draw()

# Create a slider
ax_slider = plt.axes([0.1, 0.02, 0.8, 0.03])
slider = Slider(ax_slider, 'Year', min(file_years.keys()), max(file_years.keys()),
valinit=min(file_years.keys()), valstep=1)

# Attach the update function to the slider
slider.on_changed(update_plot)
```

```
plt.show()
```

Dashboard code

```
#include <ArduinoHttpClient.h>
#include <WiFi.h>
#include <ArduinoJson.h>

// Wifi settings
char* ssid = "Interaction Lab HMI-435 2.4GHz";
char* pass = "h!9N8JLwo2";

char serverAddress[] = "192.168.50.168";  // server address will be given to you
int port = 5000;
char path[] = "/controller";  // if you are working on the brain component, put
/brain for instance

WiFiClient wifi;
WebSocketClient client = WebSocketClient(wifi, serverAddress, port);

int status = WL_IDLE_STATUS;

int discreteKnob = 33;
int continuousKnob = 34;
int slider = 32;

int years[] = { 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015, 2020 };

int allDiscreteKnobRanges[] = { 100, 300, 650, 1000, 1600, 1800, 2200, 2500, 2900,
3300, 3980 };
int discreteKnobRanges[] = { 2900, 2500, 2200, 1800, 1600, 1000, 650, 300 };
int continuousKnobRanges[] = { 120, 650, 1200, 1700, 2350, 2850, 3350, 3900 };
int sliderRanges[] = { 70, 600, 1150, 1650, 2260, 2750, 3400, 4030 };
int year = 0;
int year1 = 0;
int year2 = 0;
int previousYear = -1;  // Initialize to a value that won't match
int previousYear2 = -1;
int previousYear1 = -1;

void connectSocketClientIfNotConnected() {
  if (!client.connected()) {
    client.begin(path);
    Serial.println("Connected");
    delay(500);
  }
}

void sendToServer(int sliderValueThing, int knob1, int knob2) {
  // Make a JSON document so the webserver can easily do something with this.
  StaticJsonDocument<JSON_OBJECT_SIZE(100)> doc;
  doc["slider"] = sliderValueThing;
  doc["knob1"] = knob1;
  doc["knob2"] = knob2;

  String jsonMessage = "";
  serializeJson(doc, jsonMessage);  // Put the message in the string jsonMessage
```

```
  client.beginMessage(TYPE_TEXT);  // Begin sending the message in Json using
websockets
  client.print(jsonMessage);
  client.endMessage();
}

void setup() {
  // Put your setup code here, to run once:
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(WiFi.status());
    delay(1000);
  }
  // Print your WiFi IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
  connectSocketClientIfNotConnected();
}

unsigned long previousTime = 0;
void loop() {
  unsigned long currentTime = millis();
  if (currentTime - previousTime > 50) {
    previousTime = currentTime;
    int discreteKnobValue = analogRead(discreteKnob);
    for (int i = 0; i < 8; i++) {
      if (discreteKnobValue >= discreteKnobRanges[i]) {
        year = years[i];
        break;
      } else {
        year = years[8];
      }
    }

    //------------------------------continuous knob-------------------------------
---
    int continuousKnobValue = analogRead(continuousKnob);
    for (int i = 0; i < 8; i++) {
      if (continuousKnobValue < continuousKnobRanges[i]) {
        year1 = years[i];
        break;
      } else {
        year1 = years[8];
      }
    }

    //------------------------------slider---------------------------------
    int sliderValue = analogRead(slider);
    for (int i = 0; i < 8; i++) {
      if (sliderValue < sliderRanges[i]) {
        year2 = years[i];
        break;
      } else {
        year2 = years[8];
      }
    }
```

```
  }

  if (previousYear != year || previousYear2 != year2 || previousYear1 != year1) {
    // sendToServer(year2, year, year1);
    Serial.print("Discrete Year: ");
    Serial.println(year);
    Serial.print("Slider Year: ");
    Serial.println(year2);
    Serial.print("Continuous Year: ");
    Serial.println(year1);
    sendToServer(year2, year, year1);  //knob 1 is the discrete knob

    // Update previous values
    previousYear = year;
    previousYear2 = year2;
    previousYear1 = year1;
  }

  // delay(50);  // Wait for half a second before the next loop
}
```

Hemisphere code

```
#include <ArduinoHttpClient.h>
#include <WiFi.h>
#include <ArduinoJson.h>
#include <Adafruit_NeoPixel.h>

// Define the number of strips and their respective lengths
#define NUM_STRIPS 7
#define WS_TX_BUFFER_SIZE 7000

// Lengths and pins for each strip
int stripLengths[NUM_STRIPS] = { 103, 179, 142, 154, 162, 167, 84 };
int stripPins[NUM_STRIPS] = { 22, 21, 19, 5, 16, 0, 15 };
//Don't forget to install these libraries if you don't have them locally
uint8_t ledData[3964];

long previousTime = 0;
long deltaTime = 0;

int maxFrequency = 500;

// Create separate instances of Adafruit_NeoPixel for each strip
Adafruit_NeoPixel strips[] = {
  Adafruit_NeoPixel(stripLengths[0], stripPins[0], NEO_GRB + NEO_KHZ800),
  Adafruit_NeoPixel(stripLengths[1], stripPins[1], NEO_GRB + NEO_KHZ800),
  Adafruit_NeoPixel(stripLengths[2], stripPins[2], NEO_GRB + NEO_KHZ800),
  Adafruit_NeoPixel(stripLengths[3], stripPins[3], NEO_GRB + NEO_KHZ800),
  Adafruit_NeoPixel(stripLengths[4], stripPins[4], NEO_GRB + NEO_KHZ800),
  Adafruit_NeoPixel(stripLengths[5], stripPins[5], NEO_GRB + NEO_KHZ800),
  Adafruit_NeoPixel(stripLengths[6], stripPins[6], NEO_GRB + NEO_KHZ800),
};
```

```
//Wifi settings
char* ssid = "Interaction Lab HMI-435 2.4GHz";
char* pass = "h!9N8JLwo2";

char serverAddress[] = "192.168.50.168";  // server address will be given to you
int port = 5000;
char path[] = "/north";  //if you are working on the brain component, put /brain
for instance

WiFiClient wifi;
WebSocketClient client = WebSocketClient(wifi, serverAddress, port);

int status = WL_IDLE_STATUS;
int count = 0;

unsigned long currentTime;

void connectSocketClientIfNotConnected() {
  if (!client.connected()) {
    client.begin(path);
  }
}

void setup() {
  //setup serial connection and wifi
  // Serial.begin(115200);
  //WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, pass);
  // Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    // Serial.print(WiFi.status());
    delay(1000);
  }
  // print your WiFi IP address:
  IPAddress ip = WiFi.localIP();
  // Serial.print("IP Address: ");
  // Serial.println(ip);



  ///////////////////////////
  //You can start coding setup here:
  /////////////////////////////
  for (int i = 0; i < NUM_STRIPS; i++) {
    strips[i].begin();
    strips[i].show();
    delay(50);
  }
  //////////////////////////
}

void setStrip() {
  int index = 0;  // Starting index in ledData array
  unsigned long currentTime = millis();
  for (int i = 0; i < NUM_STRIPS; i++) {
    int stripLength = stripLengths[i];
    for (int j = 0; j < stripLengths[i]; j++) {
      int r = ledData[index++];
      int g = ledData[index++];
```

```
      int b = ledData[index++];
      int interval = ledData[index++] * 100;  // Blink rate from 0 (always on) to 255
(fastest blink)

      float dimmingFactor;
      if (interval == 0) {
        dimmingFactor = 1.0;
      } else {
        // Map blinkRate to a time interval for blinking
        int phase = currentTime % interval;  //j is the offset calculated based on
led

        int halfInterval = interval / 2;
        if (phase < halfInterval) {
          dimmingFactor = (sin((float)phase / halfInterval * 2 * PI) + 1) / 2;
          // dimmingFactor = 1.0 - (float)phase / halfInterval;
        } else {
          // dimmingFactor = (float)(phase - halfInterval) / halfInterval;
          dimmingFactor = (sin(((float)phase - halfInterval) / halfInterval * 2 *
PI) + 1) / 2;
        }  // Calculate the phase of the blink using modulus to prevent overflow
issues
        if (dimmingFactor > 1.0) {
          dimmingFactor = 1.0;
        }  // dimmingFactor = (sin(phase * 2 * PI) + 1) / 2;  // Creates a smooth
sinusoidal dimming
     }

      // Apply the dimming factor to the color components
      r = r * dimmingFactor;
      g = g * dimmingFactor;
      b = b * dimmingFactor;
      strips[i].setPixelColor(j, r, g, b);
    }
  }
  for (int i = 0; i < NUM_STRIPS; i++) {
    strips[i].show();
  }
}

void loop() {
  // long currentTime = millis();
  // currentTime = millis();

  connectSocketClientIfNotConnected();
  // Serial.println(client.available());
  //checking if message has been recieved
  int messageSize = client.parseMessage();
  if (messageSize > 0) {
    if (messageSize != 3964) {
      return;
    }
    int bytesRead = client.read((uint8_t*)ledData, messageSize);

    if (bytesRead != messageSize) {
      return;
    }
  }
  setStrip();
  // delay(10);
  //END OF MESSAGE BLOCK
}
```

Ring diameter calculation

```
import numpy as np
TOTALLEDSAVAILABLE = 35 * 60
LEDSPERMETER = 60

def heightFromTop(row, diameter, rowAmount):
    return ((diameter) / rowAmount) * (row) + (diameter/rowAmount)/2

def radiusBasedOnRow(row, diameter, rowAmount):
    heightFromCentre = (diameter/2) - heightFromTop(row, diameter, rowAmount)
    radius = np.sqrt((diameter/2)*2 - heightFromCentre*2)
    return radius

def detailBasedOnRowAndRadiusOfRow(rowRadius):
    circumfrence = rowRadius* 2 * np.pi
    return circumfrence * LEDSPERMETER

def roundUpDetailAndGetRadius(details):
    ceiling = np.ceil(details)
    return (ceiling/LEDSPERMETER) / (2 * np.pi)

def roundDownDetailAndGetRadius(details):
    ceiling = np.floor(details)
    return (ceiling/LEDSPERMETER) / (2 * np.pi)

def heightBasedOnRadius(radius, maxDiameter):
    maxRadius = maxDiameter / 2
    heightFromCentre = np.sqrt(maxRadius*2 - radius*2)
    return maxDiameter/2 - heightFromCentre

def radiusBasedOnHeight(height, maxDiameter):
    maxRadius = maxDiameter / 2
    heightFromCentre = maxDiameter - height
    return np.sqrt(maxRadius*2 - heightFromCentre*2)

def calcBasedOnDiameter(maxDiameter):
    details = 0
    rowAmount = int((maxDiameter) / 0.015)
    # print("rowAmount", rowAmount)
    for row in range(rowAmount):
        radiusOfRow = radiusBasedOnRow(row, maxDiameter, rowAmount)
        detailsBasedOnRow = detailBasedOnRowAndRadiusOfRow(radiusOfRow)
        nearestRadiusBasedOnDetail =
roundDownDetailAndGetRadius(detailsBasedOnRow)
        detailToHeight = heightBasedOnRadius(nearestRadiusBasedOnDetail,
maxDiameter)
        print("height", heightFromTop(row, maxDiameter, rowAmount) * 100 ,
"Diameter:",nearestRadiusBasedOnDetail * 200)
        details = details + np.ceil(detailsBasedOnRow)
    print(details)


calcBasedOnDiameter(0.45)
```