Sensing &
Internet of Things

Project video
Github
Are.na

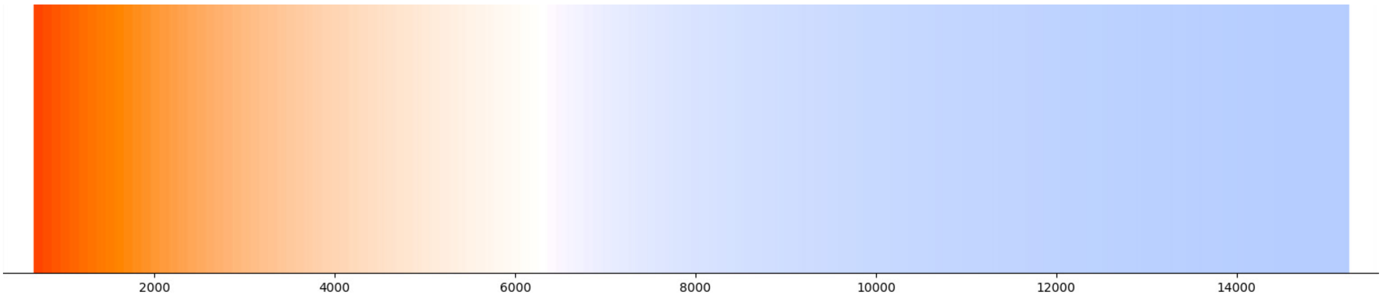
"Human-centred Office Lighting"

Kiran Scott de Martinville
Dyson School of Design Engineering

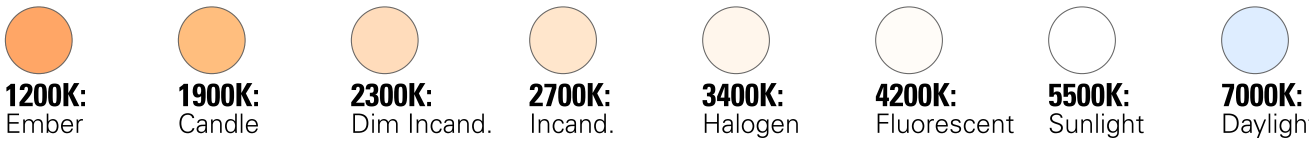
The rhythm of seasons, sunrises and sunsets have played an integral part in the million-year journey of human evolution. As a result, our bodies work on a 24h cycle regulated by a photosensitive circadian clock. This cycle has a major impact on our health both physically and behaviourally; performance patterns, immunity, alertness, hormone production, appetite and most importantly, our sleep-wake cycle. All conducted by light[1][2].

Today, we spend over half of our waking hours indoors in modern offices and workplaces flooded by artificial glare. These spaces fail to consider all characteristics of light except one - the illuminance level, which still doesn’t get close to natural daylight levels even when compliant to relevant standards. Recent surveys have concluded that a quarter or employees are dissatisfied with their lighting environment with a further four out of five having no control over it, despite the correlations between ergonomic light and job satisfaction or productivity[3].

Research on the relationship between light and our wellbeing is a recent development, well after office lighting guidelines were developed[4]. As a result they focus mainly on visibility rather than a light that keeps employees comfortable and alert. This project pilots an ergonomic lighting system that follows the characteristics of natural daylight as closely as possible, complementing the human environment.



A sample output from 1000K to 15,000K. Notice the minimal banding.



The colour temperature (Kelvin) of daylight varies throughout the day from bluish cools at noon to warmer tones at dusk or dawn. Due to recently discovered photoreceptor cells in the eye, the need for dynamic lighting has become clearer. These blue-light sensitive cells regulate our circadian (daily) and circannual (seasonal) rhythms.[5]

The first aspect of this project focuses on developing lights with variable temperature that sync to the suns height in the sky. White LED pairs are often 'set' to a limited temperature range tuned by their phosphor, so RGB LEDs were chosen to maximise temperature range.[6]

Most temperature to RGB conversion algorithms work by converting first to a XYZ colour space, followed by a later RGB transformation. A number of such algorithms exist, such as the Hernandez-Andres (1999)[7] and Kang Moon (2002)[8] methods, however;

- These aren't mathematical formula and act as look-up tables with noticeable interpolation and stepping artefacts.
- They are only accurate in a limited temperature range (2000K - 5000K for Hernandez) before returning nonsensical values
- Coupled with a XYZ->RGB transformation, the colour-science python library lags when performing real time adjustments.

As a result, I used an algorithm that reverse-engineered Mitchell Charity's raw blackbody data file developed with the CIE 1964 10-degree colour matching function[9][10]. Plotting the values reveal the following floors/ceilings:

- Red values below 6600K are always 255
- Blue values below 2000K are always 0
- Blue values above 6500K are always 255

■ *figure 1*

Natural log curve fitting was used to calculate continuous red, blue and green components, with the latter being split into two functions at the white point of 6600K. The result was then implemented into a python function:

■ *figure 2*

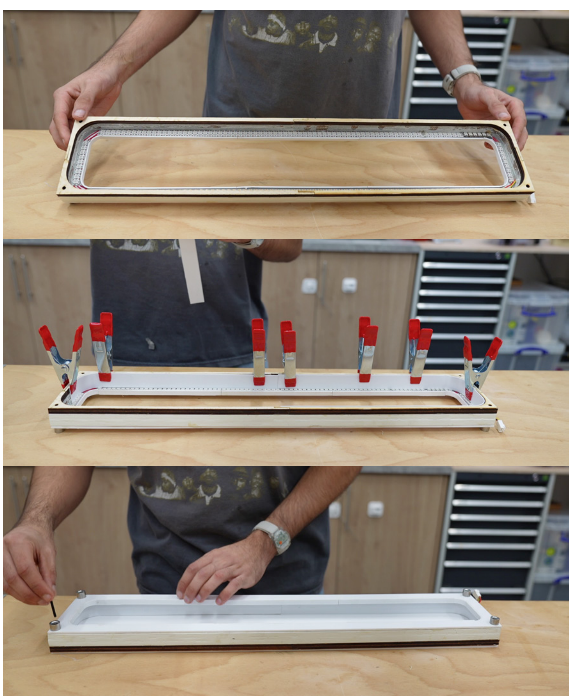
Current temperature can then be estimated by mapping the upper and lower colour ranges to the suns elevation in the sky. Since solar days are around 4 minutes longer than sidereal days, solar noon gradually shifts throughout the seasons. The illuminance level would also lower accordingly to mimic natural daylight.

The industrial design of the light panels combines the form of traditional linear light fixtures commonly found in offices with the visual characteristics of rooftop windows, hence ‘skylight’.

■ *figure 3*

The depth effect is created by having the light reflect off an inner surface before illuminating the room, creating a gradated effect. Light is then projected and subtly softened through a Perspex diffuser. The result a calmer quality of light compared to harsher direct LED illumination, keeping users comfortable.

The skylight’s inner volume constructed by layering multiple sheets of routed plywood, sandwiched between cast acrylic. The top acrylic panel features a laser-cut opening, with a inner offset making space for the LED strip to be mounted. A 3D printed inner lining ensures efficient reflection. The stack is held by 4 brushed steel bolts allowing for easy repair and maintenance. A production iteration would use a folded sheet metal chassis, powder coated white, to reduce production complexity and cost.



Skylight prototype

The skylight panels were installed in a coworking space in my department building to conduct a field study. A focus group of 8 students used the space for a week and were interviewed during use in a contextual inquiry. Additionally, a qr code was installed so that people outside the test group could learn about the project and leave feedback. Insights include:

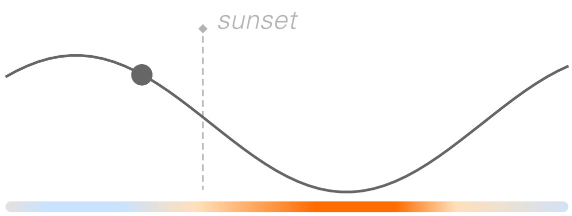
■ *figure 4*

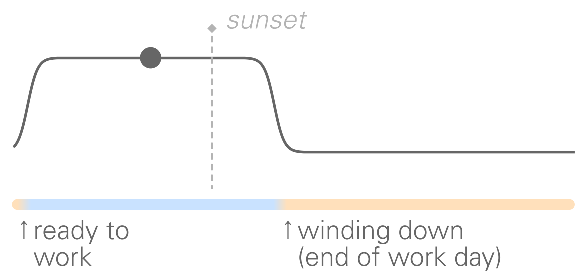
- Due to the early sunset times of London winters, large periods of time was spent in warmer temperature, lower illuminance light. Users mentioned that this didn't align with their working habits or scheduled meetings.

> " I typically found myself working way past sunset, so although the warmer light was relaxing, it made it harder to stay as productive. "

- Although users mentioned the warmer lighting near the end of their work day was soothing, multiple instances were bought up where it made tasks that required high colour-accuracy (video editing, design work etc) difficult.

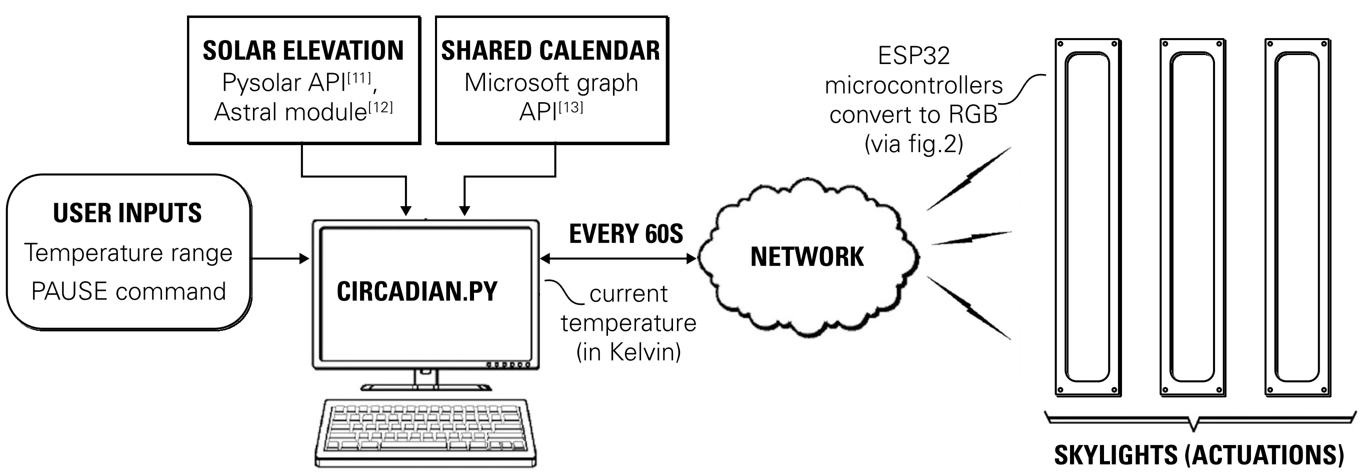
Initial vs user requested illuminance (curve) and temperature (gradient) behaviour





As a result, the behaviour of the lighting environment was changed to better reflect user needs. Since meetings and room bookings were managed using Microsoft outlook, the Graph API was used to query the latest meeting times in the room's shared calendar[13]. This would then set the period when the lights 'wind-down' to a warmer temperature stage, helping users remain productive during the work day regardless of the light levels outside, before ending the day in a natural and calming manner.

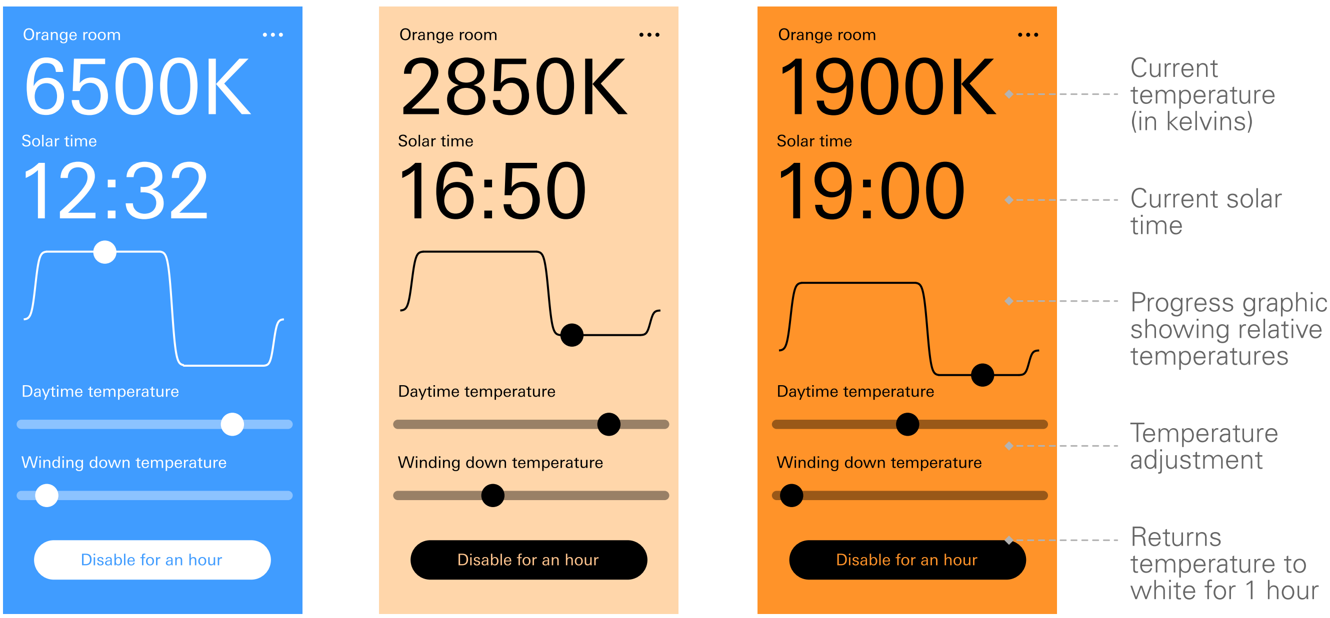
Additionally, a function was added to 'pause' the warm lighting back up to a neutral white for the duration of an hour, ensuring colour sensitive tasks could still be completed without disrupting the circadian rhythm for too long. In practice, the system fit the schedule and routine of users, without requiring them to precisely control the characteristics of each minute of the day. However, user inputs were still being done through a python terminal, so the next step was to create a dedicated UI.



Current system architecture

An initial mobile app prototype was developed using ProtoPie[15] and Socket.IO[16], allowing for temperature customisation of the work and wind-down stages simultaneously, however users reported setting lower and upper bounds of colour temperature was a confusing process.

This was due to the fact that the Kelvin scale used is counterintuitively lower at warmer, yellowish colours and higher for cooler, bluish colours, which is the reverse of more familiar heat temperature scales. Furthermore, the logarithmic behaviour at higher ranges made it difficult to for users to gage their desired lighting from the kelvin figure alone, leading them to use the web app multiple times in both stages to dial in their preferred settings.

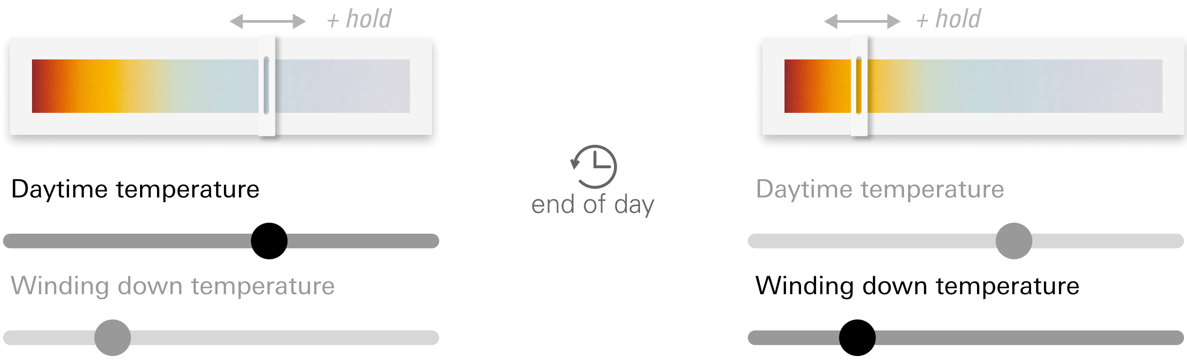


ProtoPie interaction at various stages throughout the day

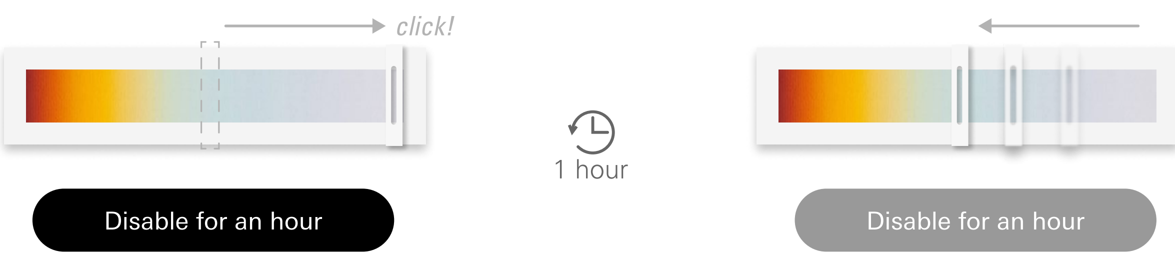
These insights, combined with the fact that adjusting a lighting environment meant to disappear into the background required a phone that took users out of their flow prompted an alternate approach: a less intrusive, less demanding spatial interface that didn't require active attention, accomplishing the same ideas with lower mental cost. The UI consists of a marker that slides across a gradient abstracting the skylight temperature.

The layout of the interface meant users intuitively realised how to adjust temperature.

As a result, the interface only changes the temperature of the current work stage. This 'one-at-a-time' approach meant users progressively tailored the lighting behaviour to what felt right to them without ever coming across a Kelvin figure. Moving the marker to a new position and holding it there causes a motor to jitter indicating to the user that the configuration had been updated through a haptic feedback.



Pushing the marker to the rightmost section of the gradient clicks a limit switch activating the pause, causing the marker to become 'stuck' at the white point for an hour before slowly returning back to its schedule. Making this a physical action requiring users to walk to the interface ensured the pause was only went to used needed, minimising negative effects on circadian rhythm.



The interface uses repurposed 3D printer components: A stepper motor and timing belt move the marker gantry along a v-slot profile.

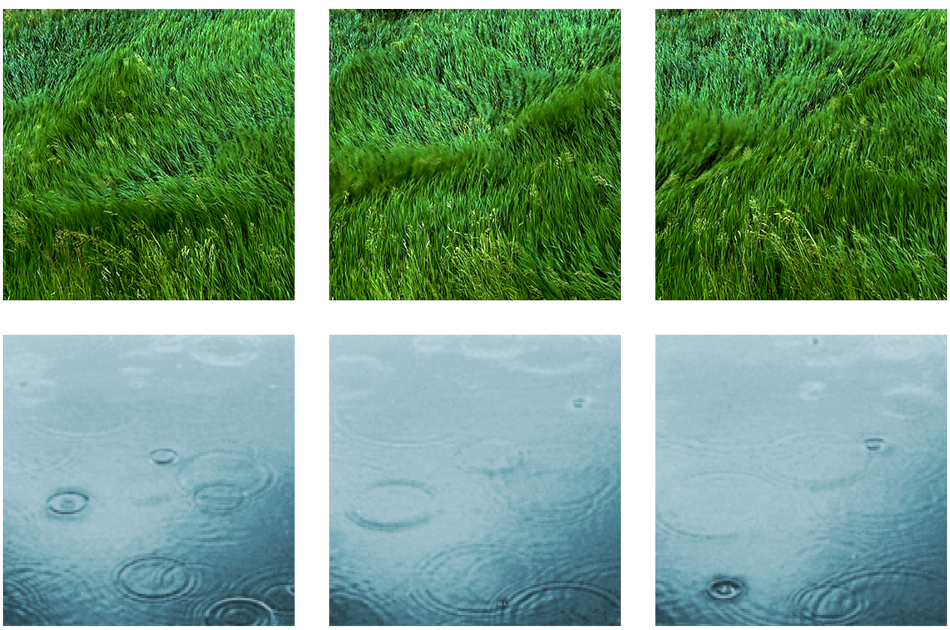
■ *figure 5*

This allowed the marker position to gently move as the work day progresses. The entire assembly is then wall mounted, acting as a light switch with an extra dimension of adjustment. The focus group enjoyed using the interface, mentioning how the subtle movement would notify them that the work day was ending without interrupting their primary task.

Once the core functionality of the lighting system had been established, explorations into additional features that leveraged the connectivity and potential of the skylights began.

This focused on calm status indicators that used peripheral awareness to load information into the environment without overwhelming the user or requiring constant attention. Mapping common smartphone utilities with the focus group led to the decision to indicate weather changes through the lighting environment. The goal was to create a calmer experience by removing the need to use an app on a productivity robbing smartphone every time users checked the weather before heading outside.

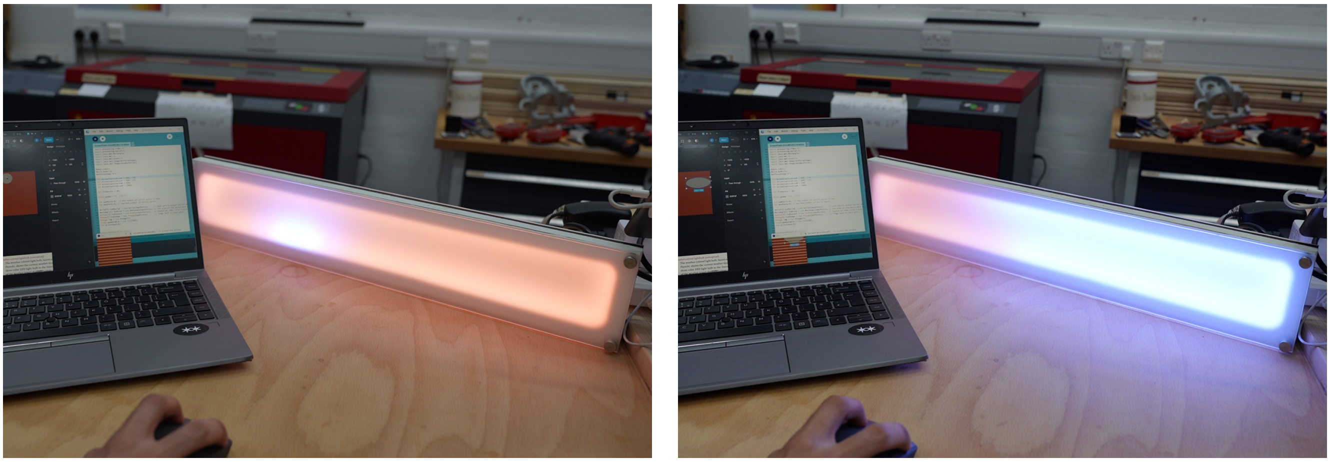
Motion cues introduced by the motorised ambient interface marker were well received by users and good at providing information without interrupting their primary task. As a result, animated patterns were prototyped to indicate significant weather changes, such as the start of rainfall or high winds.



Initial motion inspiration for weather pattern animations

The motion patterns developed took inspiration from the semiotics of the weather status they represented: exaggerated ripples from heavy rain and organic flowing grass for high winds. The idea was not to have users stare directly into the light panels, but that the subtle variance in reflections and shadows would calmly notify of the weather change.

A Processing script was used to grab screen pixels and send them to the individually addressable RGB WS2812B LEDs via a Teensy 4 microcontroller. This allowed for rapid prototyping and adjustment of the motion patterns as the skylights effectively mirrored whatever was on screen, from Figma design files to after effect projects. As a result, motion patterns could be quickly tuned without the need for any code. Multiple skylights could also be chained in series to mirror a larger area of the screen.



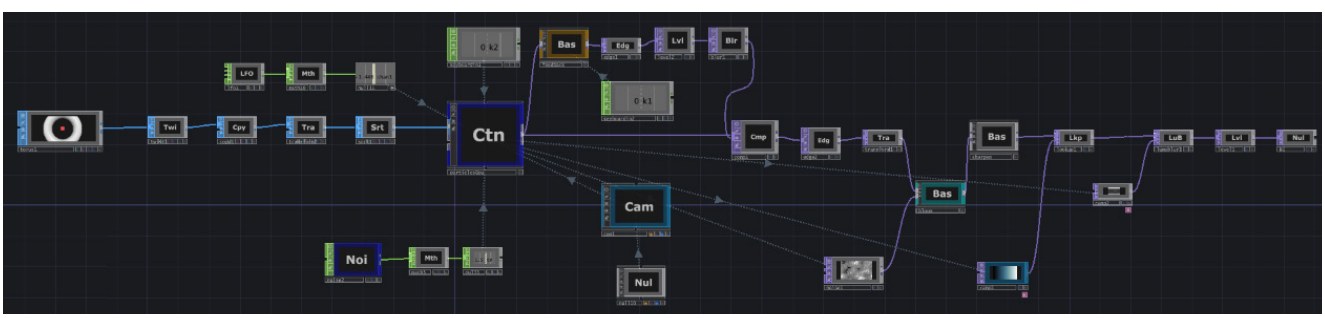
Using the pixel grabber script to mirror a Figma file onto a skylight

To generate the actual animation patterns, node based visual programming language TouchDesigner[17] was used. The script consisted of a particle system with parameters that were changed based on the current weather conditions and wind speed provided through the OpenWeatherMap API[14].

■ *figure 6*

The condition identity values are as follows:

■ *figure 7*

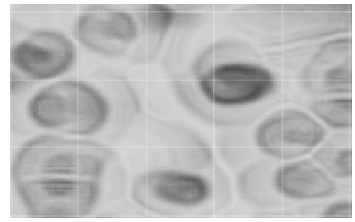


Touchdesigner script used to prototype weather animations

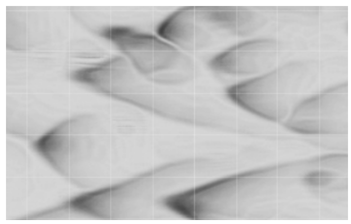
Once the condition identity changed from a value of 8xx (clear) to a 5xx (raining), particles were generated that grew and faded with intensity related to the returned ID. The result created the impression of raindrops falling on a window. Similarly, when the API returned a wind speed higher than 10m/s, force magnitude and turbulence parameters were also increased proportionally causing the particles to form streaks.

ID:800 *clear sky*

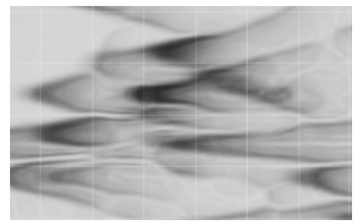
ID:500 *light rain*



ID:503 *very heavy rain*

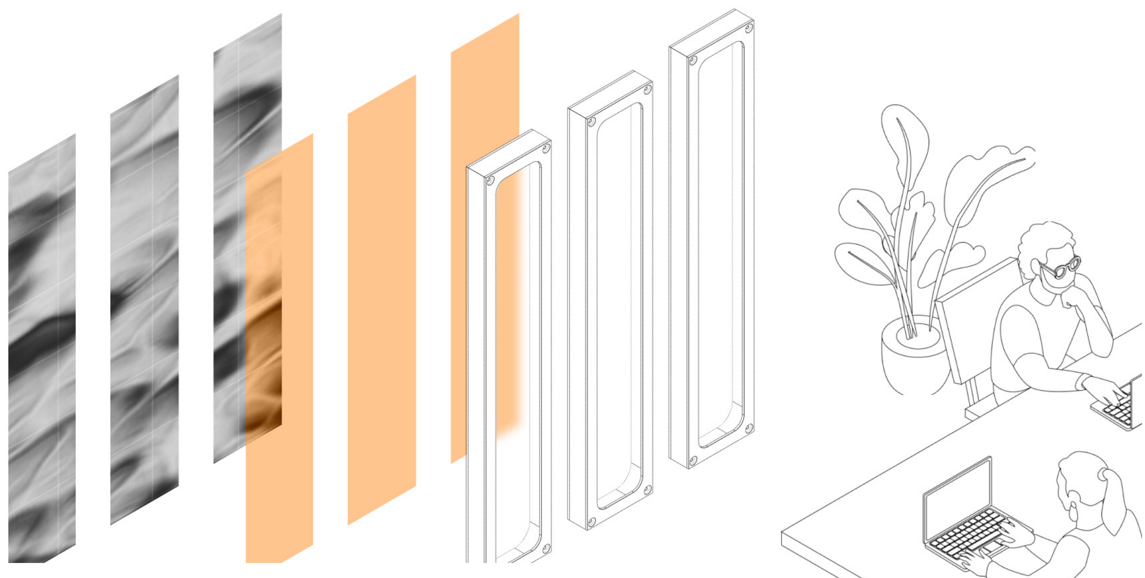


wind_speed:10.8



wind_speed:14.9

Finally, combining the reactive brightness map with the colour temperature using an overlay blend mode resulted in the enhanced dynamic lighting behaviour. The intensity and contrast of the motion elements was adjusted to ensure the effect remained subtle yet noticeable when displayed through the skylights.



MOTION LAYER
OpenWeatherMap API → TouchDesigner particles

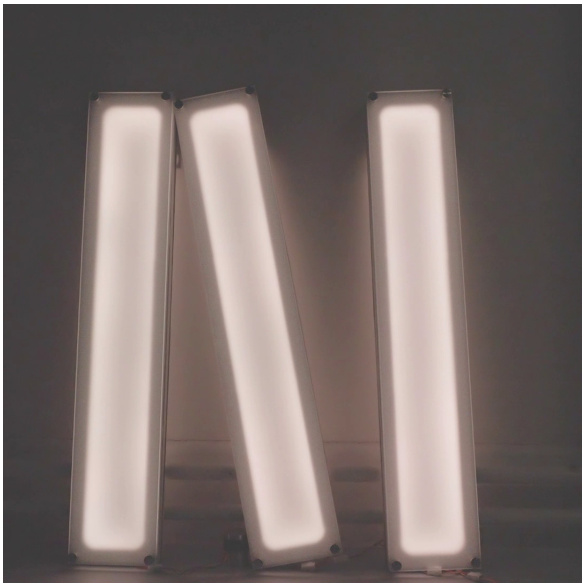
COLOUR LAYER
Solar elevation, shared calendar + ambient interface

SKYLIGHT PANELS
Teensy screengrab script → RGB LEDs

In practice, the rain animations created slow ripples of lower brightness across the skylights, creating the impression that oversized raindrops were falling onto the diffuser's surface. The wind animations were comparatively quicker, and created bright peaks along the edge of the diffuser that flickered from light to light. The difference between a calm, rainy and windy day were easily distinguishable due to the variance in motion, brightness behaviour and animation speed.

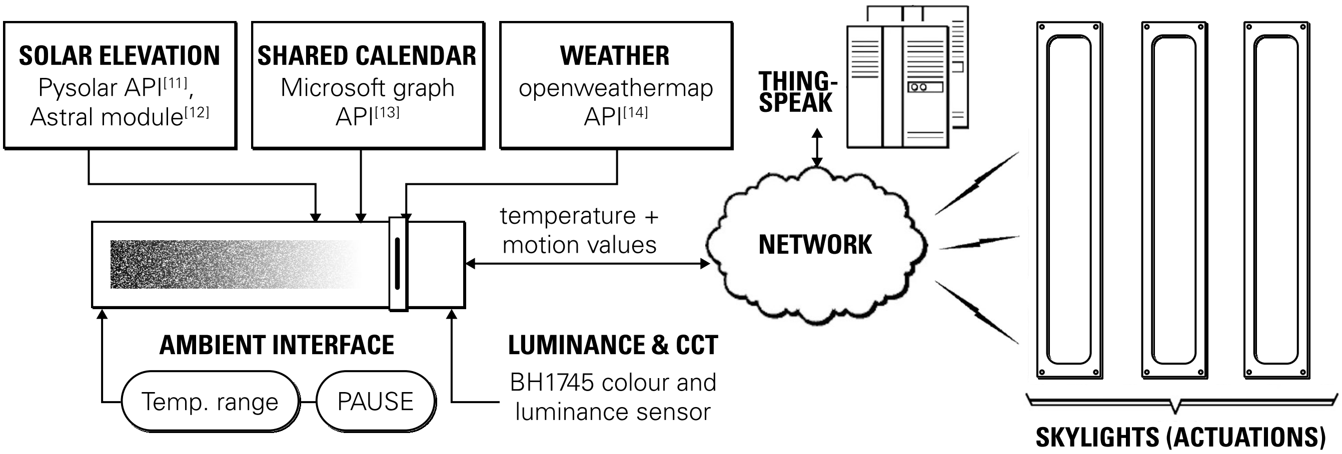
*note: these results are better communicated through the project video

Time: 12:20 (Work phase)
Colour temperature: 7000K (Overcast sky)
Weather id: 501 (Moderate rain)
Wind speed: 6.2m/s (below threshold)



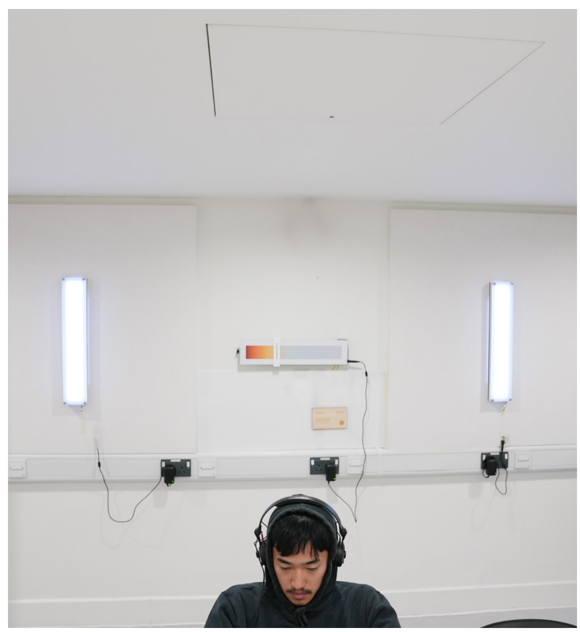
Time: 18:00 (Wind-down phase)
Colour temperature: 2850K (Incandescent)
Weather id: 803 (Broken clouds)
Wind speed: 11.0m/s (above threshold)

In order to automate the IoT system, a Raspberry pi Zero was embedded into the ambient interface. This allowed direct control of the interface marker motor through the GPIO pins. Screen recordings of the particle system at each weather ID were converted into csv files containing 512 brightness values per line, with each line corresponding to a frame. This was done as the pi wasn't able to process a real time particle system, and instead interpolated between the animation data according to the weather and wind speed provided by the OpenWeatherMap API. Furthermore, a BH1745 colour and luminance sensor was installed on the ambient interface to compare measured lighting conditions with the ideal ones. The resulting values were logged to a thingspeak channel allowing remote monitoring and system history.



Final system architecure

In conclusion, the pilot ergonomic lighting environment successfully balanced the users optimal circadian rhythm with their work schedule, leveraging APIs , wireless sensing and an ambient interface to blend into the office space, operating imperceptibly in the background and allowing all focus to be on the primary task. The industrial design of the skylights helped create a softer, pleasant lighting experience that served as a platform for additional calm weather indicators that were able to communicate useful information while minimising their presence and preserving a productive work environment.



By creating office lighting systems according to a set of intentional design principles built on optimising both user wellbeing and productive output, we can affirm the role of these spaces as places for living as well as working.

1. Weinzaepflen, C. & Spitschan, M. (Ed.) (2021). Enlighten your clock: How your body tells time. (C. Weinzaepflen, Illus.). DOI:10.17605/OSF.IO/ZQXVH
2. Koppel, T. (2012). Dynamic Lighting System for Workplaces at Northern Latitudes.
3. HSE. (2019). Human factors – Lighting. Hse.gov.uk. https://www.hse.gov.uk/humanfactors/topics/lighting.htm
4. Johannson, N. (2014). Press release New Zumtobel research on lighting quality perceived in offices.
5. Pickard, G. E., & Sollars, P. J. (2012). Intrinsically photosensitive retinal ganglion cells. Reviews of Physiology, Biochemistry and Pharmacology, 162, 59–90. https://doi.org/10.1007/112_2011_4
6. Group, S. M. (2014, August 1). LED Phosphor Application. Www.techbriefs.com. https://www.techbriefs.com/component/content/article/tb/supplements/lt/features/articles/20166
7. Hernández-Andrés, J., Lee, R. L., & Romero, J. (1999). Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities. Applied Optics, 38(27), 5703. https://doi.org/10.1364/ao.38.005703
8. Kang, B., Moon, O., Hong, C., Lee, H., Cho, B., & Kim, Y. (2002). Design of Advanced Color -Temperature Control System for HDTV Applications. Journal of the Korean Physical Society, 41(6), 865–871.
9. Blackbody colors - choosing parameters. (n.d.). Www.vendian.org. Retrieved December 6, 2023, from http://www.vendian.org/mncharity/dir3/blackbody/parameters.html
10. CIE 1964 10-deg CMFs. (n.d.). Cvrl.ioo.ucl.ac.uk. Retrieved December 6, 2023, from http://cvrl.ioo.ucl.ac.uk/database/text/cmfs/ciexyz64.htm
11. Kennedy, S. (n.d.). astral: Calculations for the position of the sun and moon. PyPI. Retrieved December 7, 2023, from https://pypi.org/project/astral/
12. Pysolar: staring directly at the sun since 2007 — Pysolar 0.8rc2 documentation. (n.d.). Pysolar.readthedocs.io. https://pysolar.readthedocs.io/en/latest/
13. CelesteDG. (n.d.). Microsoft Graph documentation. Learn.microsoft.com. https://learn.microsoft.com/en-us/graph/
14. OpenWeatherMap.org. (2012). Current weather and forecast - OpenWeatherMap. Openweathermap.org. https://openweathermap.org/
15. ProtoPie | High-Fidelity Prototyping for Mobile, Desktop, Web & IoT. (n.d.). ProtoPie. https://www.protopie.io/
16. Socket.IO. (2019, October 20). Socket.IO. Socket.IO. https://socket.io/
17. Application Building. (2017, June 28). Derivative. https://derivative.ca/feature/application-building

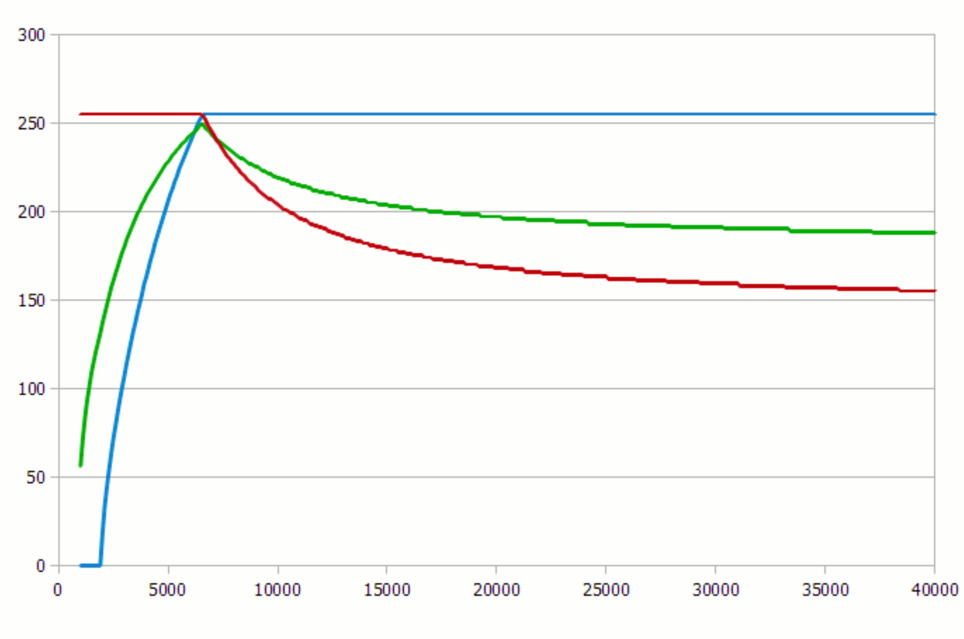


fig 1 Plot of raw Temperature vs RGB values (CIE 1964 10 degree CMFs). Notice white point at 6500K

```python
def convert_K_to_RGB(colour_temperature):
    #range check
    if colour_temperature < 1000:
        colour_temperature = 1000
    elif colour_temperature > 40000:
        colour_temperature = 40000

    tmp_internal = colour_temperature / 100.0

    # red
    if tmp_internal <= 66:
        red = 255
    else:
        tmp_red = 329.698727446 * math.pow(tmp_internal - 60, -0.1332047592)
        if tmp_red < 0:
            red = 0
        elif tmp_red > 255:
            red = 255
        else:
            red = tmp_red

    # green
    if tmp_internal <=66:
        tmp_green = 99.4708025861 * math.log(tmp_internal) - 161.1195681661
        if tmp_green < 0:
            green = 0
        elif tmp_green > 255:
            green = 255
        else:
            green = tmp_green
    else:
        tmp_green = 288.1221695283 * math.pow(tmp_internal - 60, -0.0755148492)
        if tmp_green < 0:
            green = 0
        elif tmp_green > 255:
            green = 255
        else:
            green = tmp_green

    # blue
    if tmp_internal >=66:
        blue = 255
    elif tmp_internal <= 19:
        blue = 0
    else:
        tmp_blue = 138.5177312231 * math.log(tmp_internal - 10) - 305.0447927307
        if tmp_blue < 0:
            blue = 0
        elif tmp_blue > 255:
            blue = 255
        else:
            blue = tmp_blue

    return red, green, blue
```

fig 2 Temperature (K) to Colour (RGB) function based on curve fitting fig 1

fig 3 Initial skylight moodboard

fig 4 Information and feedback plaque in testing environment

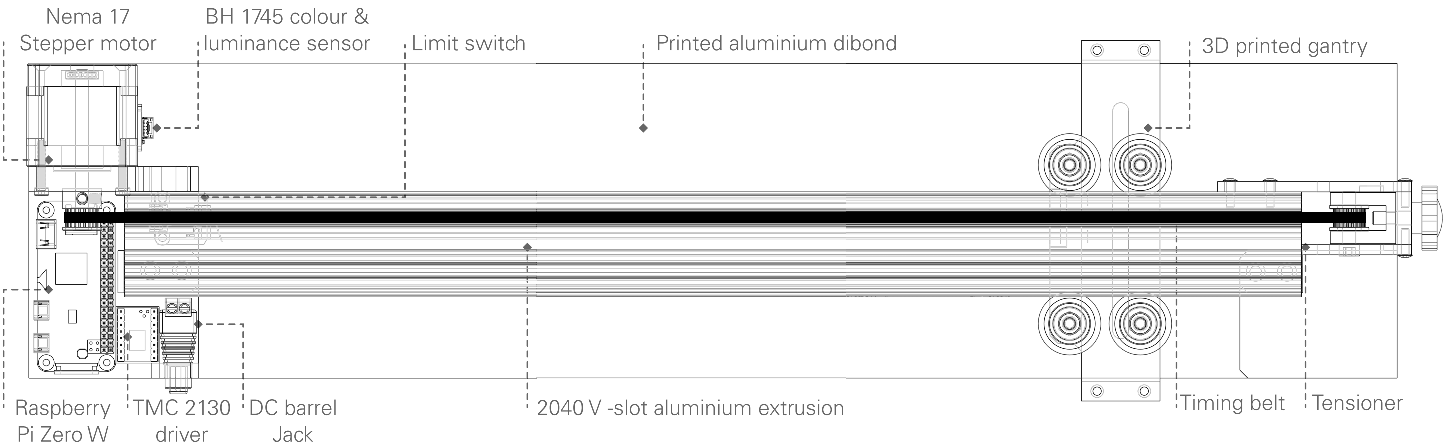


fig 5 Ambient interface

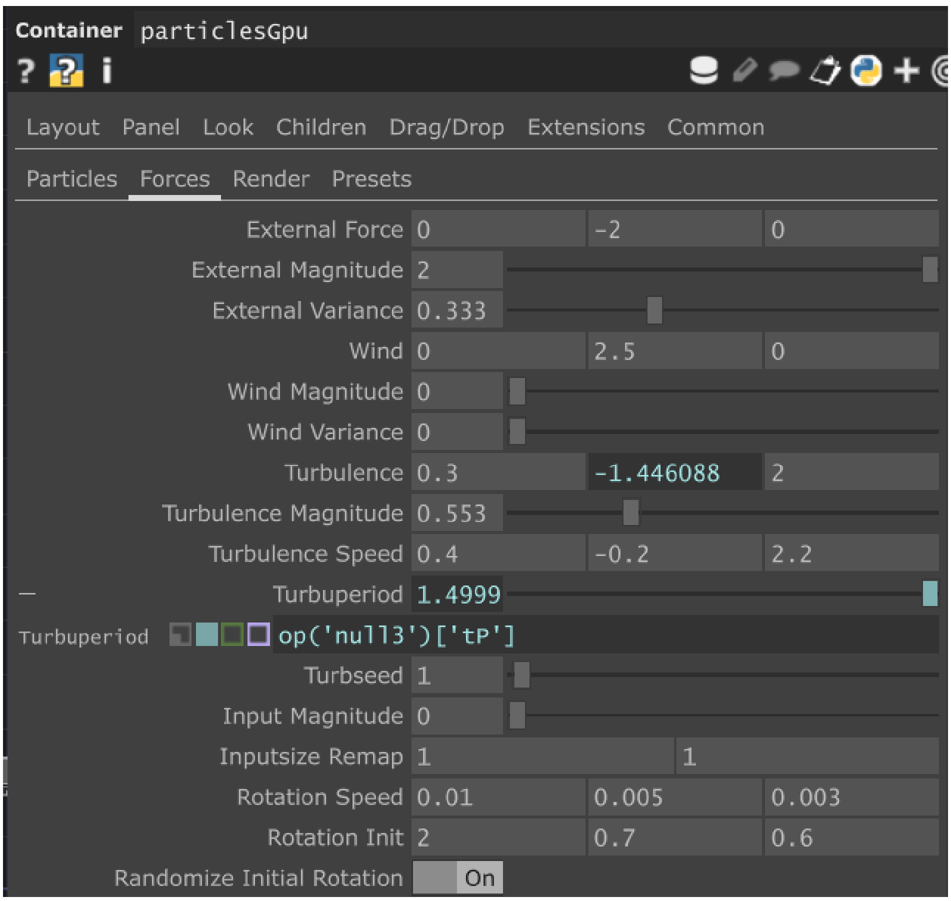


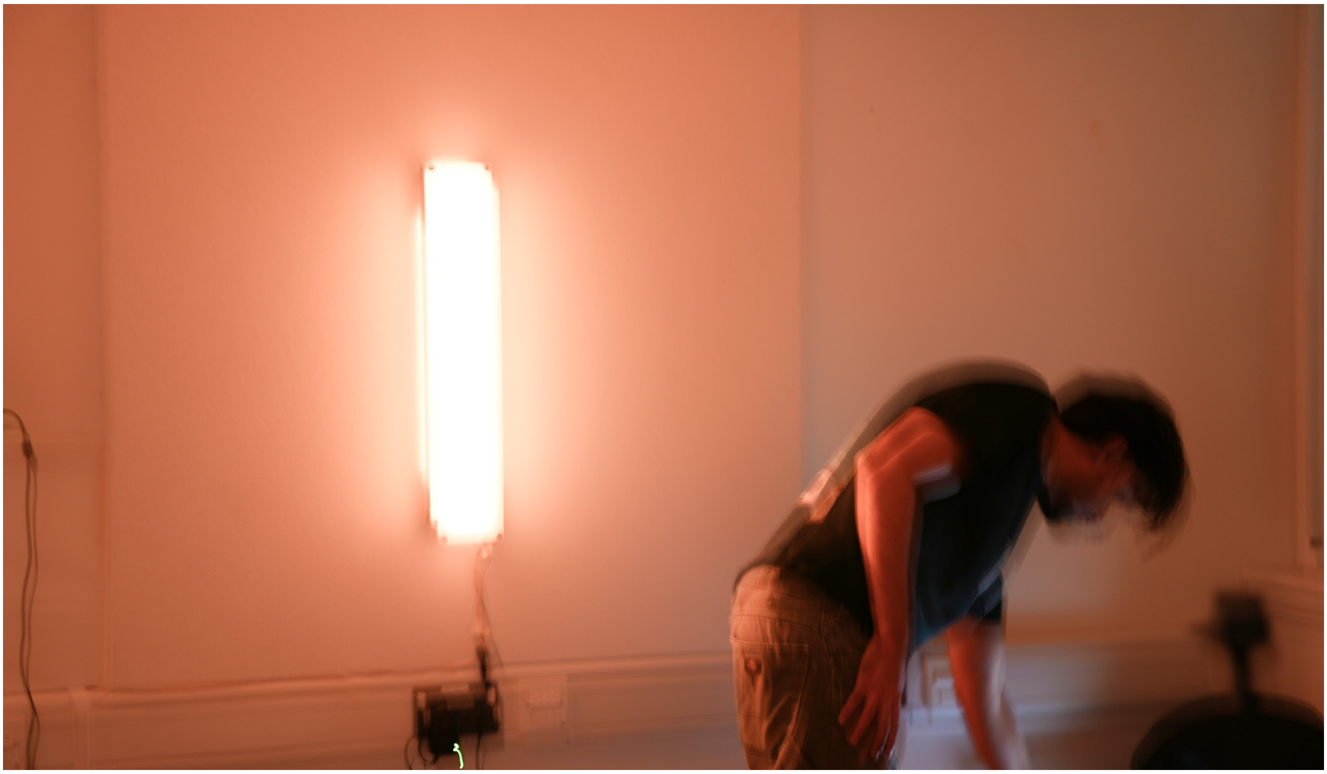
fig 6 TouchDesigner particle system parameters

ID	MAIN	DESCRIPTION
300	Drizzle	light intensity drizzle
301	Drizzle	drizzle
302	Drizzle	heavy intensity drizzle
310	Drizzle	light intensity drizzle rain
311	Drizzle	drizzle rain
312	Drizzle	light intensity drizzle rain
313	Drizzle	shower rain and drizzle
314	Drizzle	heavy shower rain and drizzle
321	Drizzle	shower drizzle

ID	MAIN	DESCRIPTION
500	Rain	light rain
501	Rain	moderate rain
502	Rain	heavy intensity rain
503	Rain	very heavy rain
504	Rain	extreme rain
511	Rain	freezing rain
520	Rain	light intensity shower rain
521	Rain	shower rain
522	Rain	heavy intensity shower rain
531	Rain	ragged shower rain

ID	MAIN	DESCRIPTION
800	Clear	clear sky
801	Clouds	few clouds: 11-25%
802	Clouds	scattered clouds: 25-50%
803	Clouds	broken clouds: 51-84%
804	Clouds	overcast clouds: 85-100%

fig 7 select OpenWeatherMap weather conditions IDs



Cover image - Love time, Lewis Khan (2017)

Photography - DVD (David Chen)

An additional thanks goes to the Ideas Lab, a student run maker space in Imperial College London. Without access to the tools and expertise curated by the team, I wouldn't of been able to fully execute the vision I had for this project.

Thank you