# Ideation and Design

## Local Interactions

I do not find the exhibit for local visitors to be coherent. My understanding of the interactions is shown in Figure 1.

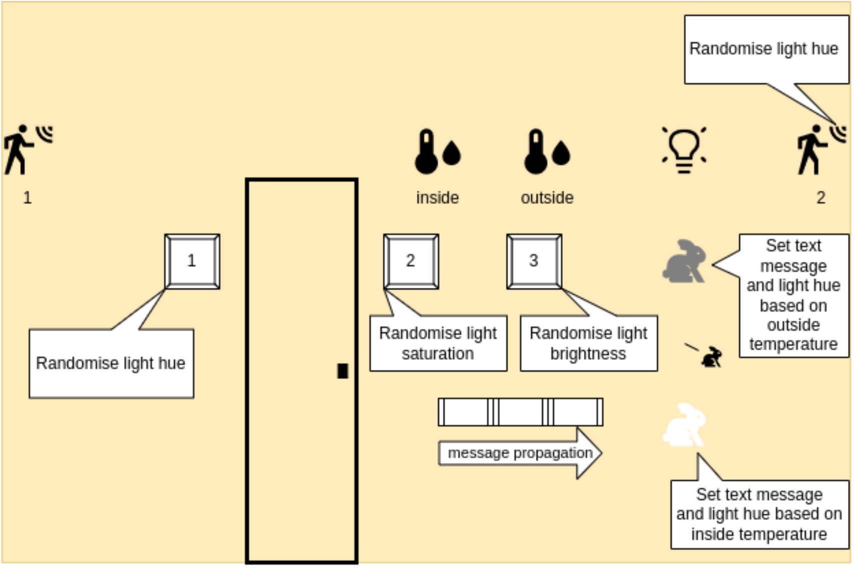
Randomise light hue
1
inside
outside
2
1
2
3
Set text message and light hue based on outside temperature
Randomise light hue
Randomise light saturation
Randomise light brightness
message propagation
Set text message and light hue based on inside temperature

Figure 1: Local exhibit interactions

The interactions with the buttons and motion detector(s) are semantically divorced from the interactions with the rabbits, but they both affect the light – I do not find that to be logical, nor conducive to meaningful interaction. Also, the local interactions do not disambiguate between different types of button press.

## Inter-user Interactions

The motion detectors should make it relatively simple to let remote users know that local users are next to the display.

It is more difficult to get the attention of local users – the OLEDs are not the most attention-grabbing, and the light only updates once per second, so it is not possible to quickly flash it on and off/between different colours. We will essentially have to assume that local users are paying attention to the OLEDs.

The fact that the local input devices all affect the light will make it impossible to allow local users to communicate with remote users without side effects. The buttons could be used for: allowing local users to answer multiple-choice questions posed via the OLEDs; binary or tertiary input; or some variation of Morse code. The rabbits could allow answering yes or no questions but will always prompt ChatGPT-generated OLED output, which would interrupt the communication.

## Remote Interactions

Due to not understanding the intention behind the local interactions, I find it difficult to directly build on them for the remote interactions. The data available is:

- Motion detectors 1 and 2: motion or no motion

- The indoor and outdoor temperature and humidity: raw integer values
- The local button presses: 3 buttons and 3 types of press
- The rabbits' contact: white, grey or none
- OLED text: raw strings
- The light: hue, saturation and brightness

A very simple interaction method would be to create an interface like Figure 1 that reflected the state of the physical display in real time, using the buttons on two BBC micro:bits as proxies for buttons 1-3 and the rabbits (one button could encode single press as white, long press as grey and double press as removed from contact). The groupIds of the inputs are not writable in the database, but it would be quite simple to replicate code in the app to spoof local inputs (provided the museum's codebase is not expected to change). That is not a very creative idea, however.

Relying on the motion detectors for input aside from alerting remote visitors to local ones will mean that some remote interactions are potentially impossible without local visitors, which would be undesirable. Using the temperature and humidity lends itself to natural history but does have the issue that on a day with stable weather it will not be very dynamically interactive. As it stands, the light values will be mostly random. We can also use the current time of day as an input.

Essentially, to achieve m*eaningful and logical* interactions will require creating interaction semantics from scratch. To help generate an idea for the exhibit, I make the following assumptions about its use:

- The exhibit is part of a natural history display and should be in some way educational.
- The exhibit is intended primarily for children (and light-hearted adults): it is not solemn in tone.
- The exhibit will be used locally during business hours but could be accessed at any time by remote visitors.

## Modelling Rabbit Behaviour

We could model the behaviour of a rabbit based on the time of day and weather conditions. This allows visitors to learn about rabbits in an interactive and hopefully engaging way. The activities of a happy and healthy rabbit include the following[1]:

- rest and sleep in comfort
- eat and drink undisturbed
- exercise and explore safely
- hide when afraid or feeling insecure
- shelter from the weather including cold, heat, and rain
- play with companions

What activity they choose to do can be informed by the following:

- Time of day: rabbits are crepuscular, meaning they are mostly active at dawn, dusk and certain periods during the night[2].
- Weather: rabbits do not like to get wet[3] and will usually seek shelter during rain.
- Outside temperature: rabbits (at least those native to temperate parts of the world) are more likely to be endangered by high temperatures than low ones, but will typically seek cool spaces at either extreme[4].
- The presence of other animals: rabbits are prey animals and hide from (perceived) predators.

A model of a rabbit's daily life is not inherently interactive. We can allow visitors to control either the rabbit or the rabbit's environment. In the former case, we could have some sort of scoring metric based on how closely the visitor mimics natural behaviour, gamifying the experience, but given the only strong predictions we can make from our simple environment model are two binary ones (inside/outside and resting/active), there would be limited scope for engaging gameplay. Consequently, the latter seems to be a better choice, and certainly has room to be educational and engaging.

In summary, the concept for the exhibit is to simulate the behaviour of a rabbit based on its surrounding environment. Visitors can affect the environment through both the local and remote exhibits and then watch how the rabbit responds, hopefully learning something about rabbits along the way. The rabbit's actions will be reflected by the Expo/React interface for remote visitors, and in strings displayed on the OLED screens for local visitors. The remote exhibit will have a graphical representation of the environment, and the local exhibit will rely on the existing devices.

# Implementation

## Rabbit Simulation

Our environment will be modelled by the following features:

| | Data type | Local input |
|---|---|---|
| **Temperature** | Raw (integer) value | Light hue (button 1, motion detector 2 and grey/white contact) |
| **Humidity** | Raw (integer) value | Outside humidity sensor |
| **Time of day** | Integer (1440 minutes per day) | Light brightness (button 3) |
| **Season** | Integer (1-4) | Button 1 long |
| **(Perceived) predator present** | Boolean | Motion detector 1 |
| **Random choice variable** | Integer | Light saturation (button 2) |
| **Raining** | Boolean | **Humidity and temperature**[1] |
| **Rabbit location** | Boolean | Rabbit contact *and simulated rabbit's decisions* |
| **Rabbit activity** | String | *Any change in the above* |

Table 1: Environment features

Our little simulated rabbit will decide what to do and where to do it (inside or outside) based on the above environment features. See the codebase for the complete logic, but examples of decisions include: sleeping for randomised amounts of time during the day; guaranteed wakefulness around sunrise and sunset; hiding from (new) visitors outside the warren; sheltering from bad weather; eating and playing if not hiding, sheltering or sleeping.

[1] To my understanding, predicting if it is raining based on current humidity and temperature alone is unlikely to be accurate (both because other factors like pressure and wind direction should be accounted for, and because moment-to-moment weather prediction is less accurate than more long-term prediction of trends). However, it will do for the sake of our interactive museum exhibit.

## Local Exhibit

The IoT devices impact the environment as described in Table 1. The impact of the local interactions may not be obvious to visitors, so some sort of plaque explaining the exhibit should be added. The rabbit's activities and observations (e.g. if it is raining) are represented by messages sent the OLED screens. To prevent messages being repeated too frequently, if the simulation would send the same message twice in a minute, then it sends a random rabbit fact between the repeated messages. This should increase both how engaging and how educational the exhibit is.

If remote users set static values for the environment features, then by default this is not directly reflected by the local exhibit (e.g. changing temperature does not affect light hue), which is consistent with the lack of feedback for motion detector 1 etc.

## Remote Exhibit

### P5 and Expo

The remote exhibit is a single webpage built with React, with an embedded P5 canvas that offers a graphical representation of the environment and simulated rabbit. Due to this only being a prototype, the graphical display is minimalistic: the little rabbit is modelled as a square, the grey and white rabbits are static rectangles, and the presence of a visitor is denoted by a large and imposing triangle appearing outside the warren (see Figure 2). The little rabbit teleports based on their current activity, which is represented by the text above their head. In Figure 2, the little rabbit is sleeping in the warren because it is near noon (note the position of the sun), and rabbits are not typically active during the day. The presence of a visitor does not wake the rabbit.

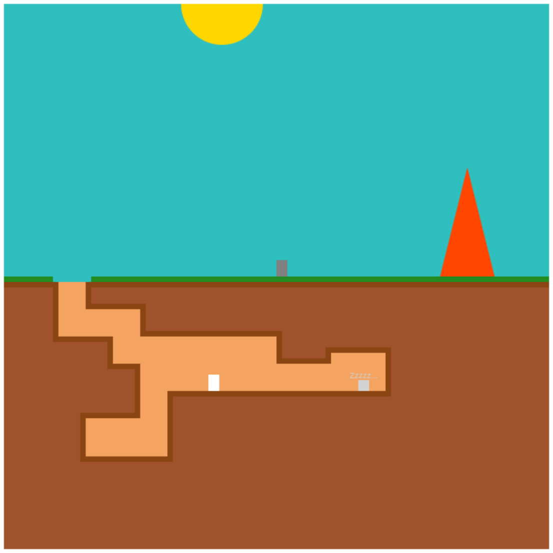

Figure 2: P5 interface

Beneath the P5 interface there is a table showing the values of the various environment variables, with sliders to set them to static values instead of updating from the local exhibit, which is a finer degree of control than can be offered with the local exhibit's interactions. The last 10 messages sent to the OLED screens are also displayed on the page, as well as a brief explanation of the exhibit. The rabbit icon next to the "Request micro:bit port" button reflects the colour of the light on the local display, just for fun.

## BBC micro:bit

A connected BBC micro:bit can act as a proxy for the buttons of the local exhibit, affecting the light values (and by extension the simulation) just like the button presses would. It also has the additional functionality of making the light hue and brightness match the temperature and time of day of the simulation, as opposed to the other way around. See Table 2 for the key bindings.

| Action | Effect |
|---|---|
| Short press A | Randomise hue |
| Long press A | Set hue to match simulation temperature |
| Short press B | Randomise brightness |
| Long press B | Set brightness to match simulation time of day |
| Shake | Randomise saturation |

Table 2: BBC micro:bit bindings

## Difficulties

Other students sending updates to the shared database made debugging difficult.

Other students sending lots of values to the database in quick succession (100+ in the space of a second) would cause my app to crash, so I limited state updates from the database to 1 per 100ms. This means that some updates might be missed, but that is both rare in normal use and preferable to crashing.

The existing codebase limits temperatures between 0 and 20 for the sake of conversion to light hue. I maintained this limit for consistency but did have to fudge the temperatures at which rabbits seek shelter inside, as they can usually withstand temperatures beyond that range.

Wrapping my screens in Tab and Stack navigators kept setting their width to 0, which is part of why the website is only a single (slightly crowded) page. I spent a while trying to fix this without success and have no real idea what went wrong. If page navigation were working then I would have included the DebugScreen on a separate page, which simply shows the live values of both the database and the simulation.

react-native-paper.DataTable.Cell.onPress does not work – the cell does not become clickable. Contained components (e.g. Button) are also not clickable. My workaround is using material-ui Grid components for non-header table rows.

The grey rabbit contact of the physical exhibit forces the rabbit outside, but it might immediately hide/shelter if the environment dictates it; I think this is appropriate, as if you forced a rabbit into the open it probably would hide from you, but it can seem strange.

## References

1. ‘Suitable environment for rabbits | nidirect’, *NI Direct*. https://www.nidirect.gov.uk/articles/suitable-environment-rabbits (accessed May 02, 2023).
2. ‘Facts about rabbits you probably didn’t know! | Blue Cross’, *Blue Cross*. https://www.bluecross.org.uk/advice/rabbit/wellbeing-and-care/facts-about-rabbits (accessed May 02, 2023).
3. ‘Bathing Bunnies | Rabbit Welfare Association & Fund (RWAF)’, *Rabbit Welfare Association*. https://rabbitwelfare.co.uk/bathing-bunnies/ (accessed May 02, 2023).
4. ‘Will My Rabbits Be Okay Outside in Winter? - Omlet Blog UK’, *Omlet.co.uk*. https://blog.omlet.co.uk/2020/10/30/will-my-rabbits-be-okay-outside-in-winter/ (accessed May 02, 2023).