PLAYFULLY TEACH ABOUT PATTERNS OF **COLLECTIVE BEHAVIOUR** THROUGH AN **INTERACTIVE DESIGN**

Tessa Tan

Master thesis Integrated Product Design

TU Delft

Playfully teach about patterns of collective behaviour through an interactive design

Master Thesis
Integrated Product Design
Faculty of Industrial Design Engineering
Delft University of Technology

Author
Tessa Tan

Supervisory Team
Ing. Aadjan van der Helm
Dr. Jan Jaap van Assen

Date
17 February 2022

ACKNOWLEDGEMENT

First of all, I want to thank everyone who helped and supported me throughout my graduation project.

I would like to express my thanks to Aadjan van der Helm and Jan Jaap van Assen for their support and guidance at every stage of the project. Thank you for all the feedback that supported me throughout the design process and for the learning opportunities that helped me improve my design and results.

I want to thank StudioLab and Applied Labs at the Faculty of Industrial Design Engineering for their guidance on the digital hardware of my prototype, and Don, Willem and Carlo for supporting me in creating the physical part of my prototype at the Model Making and Machine Lab. I also would like to thank Arnold for providing me with knowledge about museum interaction design and Mathieu for giving me insights about designing for children.

I am extremely grateful to Rob, Daphne and Gavin who made it possible for me to evaluate and validate my prototype in Naturalis, which certainly helped me improve my design.

Lastly, I want to express my thanks to all participants, my friends and my family who helped in evaluating and giving feedback on all my ideas and concepts. Without them, I could not have come to the final result of my graduation project.

SUMMARY

Collective behaviour can be seen everywhere. It is the phenomenon where many individuals form a group together and act as a whole. These groups are mesmerising to watch, but we can also learn from them. The way these groups behave depend on how the individuals react to their local surroundings. Obtaining knowledge about this concept brings knowledge in different areas like animal groups, human behaviour and complex design systems.
The goal of the project was to make a wide audience aware of collective behaviour, by teaching them in a playful interactive way, through a museum installation. The target group are museum visitors of age 8 and older. The question was what the user experience would look like and how the design would take shape.

The project started with understanding what collective behaviour is, to find out which elements are most important to convey in the design. Designing for a museum context and different technologies that can be used in the installation are explored. This phase concluded with requirements to implement in the museum installation.
In the following phase, the user interaction and the way visitors will manipulate the installation are ideated, and the goal for the user is determined. This phase included brainstorming, ideation by sketching and eventually lead to the design of the installation embodiment.

In the last phase of the project, the final design was determined and detailed: Control the Collective. By interacting with the designed installation, users can manipulate the size of three zones, in which the agents behave in a certain way. These are the zone of repulsion, the zone of orientation and the zone of attraction. Museum visitors can change the size of these zones by sliding disks along a line on the floor and with the use of corresponding buttons. These disks and buttons are placed on top of a circular platform, which users can stand on top of. The size of these zones and the influence on the composition of an individual within the group are visualised on this platform. Simultaneously, on a wall curved around the platform, the accompanying visualisation of the moving group is displayed.
The museum installation was evaluated and validated in Naturalis on user experience and the level of understanding of users. Lastly, recommendations are provided for future research and implementation of the design.

TABLE OF CONTENTS

1. INTRODUCTION

This chapter introduces the concept of collective behaviour: what is it, where can we see it and what can we learn from it? It will conclude with the scope of the graduation project.

1.1 WHAT IS COLLECTIVE BEHAVIOUR?

Collective behaviour can be seen everywhere around us. It is the phenomenon where many individuals form a group together and act as a whole (Sumpter, 2005, 2010; Giardina, 2008). However, not all groups behave collectively. When groups display collective behaviour, the members of the group, also called agents, move and behave cohesively in a self-organized way.
We can distinguish different kinds of patterns in these groups, but they all have something in common: the behaviour of every agent in the group is influenced locally by other agents and their environment. This means that every agent is aware of its local surroundings, which it bases its actions on. Having everyone in the group respond to their own surroundings, results in the collective movements that we can see (Couzin et al., 2002; Reynolds, 2007). It is interesting to see that if only a small part of the group obtains new information, this new knowledge will transfer through the local interactions, and eventually reach the whole group. An example of this information is a change of direction to avoid a predator attack (Hemelrijk & Kunz, 2004).

Animal groups
Collective behaviour can be seen in many animal groups. Take for example a school of fish (Figure 1). They can be found swimming in the ocean in large organized groups. Being in a group brings many advantages. One of them is safety in numbers, together they stand a better chance against predators. Schooling can visually confuse predators, and being in a group makes it more difficult for predators to single out one individual. It is also easier to find food in a group since more eyes are looking. Being in a school together has been helping fish in their evolutionary success (Cabos, 2018; Larsson, 2012).

Human crowds
Collective behaviour is also apparent in human behaviour. Take for instance a crowded train station or a busy zebra crossing (Figure 2), where we all move in the same direction. This happens while we are also trying to not bump into each other (Vilardo & Wepprecht, 2016).

Bacteria fluids
Collective behaviour can also be found on a microscopic level. An example is the behaviour of bacteria fluids. Microswimmers like the Escherichia coli and the Bacillus subtilis move around while being influenced by their neighbours. This results in self-organizing patterns, which is shown in Figure 3 (Lushi et al., 2014; Xu et al., 2019; Zhang et al., 2010).

Figure 1: School of fish (Cowell, 2019)

Figure 2: Shibuya Crossing in Tokyo (Suwannaphoom, 2019)

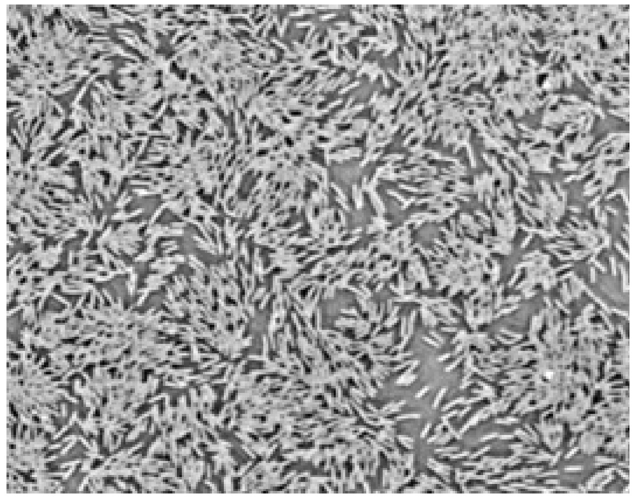


Figure 3: Collective motion patterns of Escherichia coli bacteria (Max Planck Institute for terrestrial Microbiology, 2019)

1.2 WHY IS COLLECTIVE BEHAVIOUR INTERESTING?

While collective behaviour is mesmerising to watch, we can also learn from it. The different compositions and patterns, ranging from swirling starling murmurations to an aligned group of swimming manta rays, all depend on the difference in how individuals of the group react to their local surroundings.
Since this concept plays a role in various areas, teaching people about this concept brings them valuable knowledge: obtaining an understanding where these movements and patterns come from. We as humans show collective behaviour in our daily life. Having knowledge of how collective behaviour is formed, can help when being in a crowd, by becoming more aware of our surroundings, recognizing signals and anticipating how the mass will move. This can be of great help when being in a high-density crowd, for example at festivals. It is interesting to see why animals behave in a certain way, and how it is of importance for their ecology and evolution. The awareness how animals and other organisms are able to form self-organized groups, also gives us knowledge for designing complex systems, such as software for automatic robots. Adapting the movements, patterns and underlying behaviour rules of collective organisms to technology, brings many possibilities.

1.3 SCOPE OF THE PROJECT

Collective behaviour is a very broad concept and consists of many elements. The goal of the project is to make a wide audience aware of collective behaviour, and explain that by being influenced by local surroundings, collective patterns can be formed. The intention is not to teach them the full ins and outs of collective behaviour, but that with simple underlying rules, complex collective motion patterns can emerge. Understanding how these simple rules impact the behaviour, gives the user of the design a better insight in what is behind collective behaviour.
The project includes requirements which should be taken into account when designing and developing the concept. An overview of these requirements can be found in Section 3.4.1 Requirement list.
To reach a wide audience to fill this knowledge gap, the context of placement of the design will be a museum: a museum installation is going to be designed (Requirement 1).
The particular museum that is going to be designed for is Naturalis, a national museum of natural history and a research center on biodiversity, located in Leiden (Naturalis, 2022). The museum installation should be designed in a way to include all its visitors, however, the complexity of the educative part is targeted for visitors of 8-year-olds and older (Requirement 2).
The challenge is how to translate current research about collective behaviour, into an easily understandable design.
Collective behaviour is visible in the resulting moving patterns that the individuals grouped together display. To show these patterns, the installation will mainly consist of a visual element (Requirement 5). With the use of visualisation, insights of the changes in the composition of patterns and direction of movement can be conveyed.

Stakeholders

Naturalis museum
Educate and attract new visitors

Researchers
Reach wider audience

Museum-goers
Teach them something new and boost their curiosity

The way the visitors will use the design will be interactively, to opt for better comprehension: the ability to directly manipulate factors and see the changing result will lead to a more in-depth understanding of the concept (Requirement 6). This interaction should be intuitive, for museum visitors to quickly understand how to interact with it, so they do not spend their time figuring out how the installation itself works (Requirement 7).
The users are key, the objective is that they feel entertained while learning, and stay engaged to spend enough time interacting with the design to obtain an understanding of collective behaviour (Requirement 8). However, they should not feel overwhelmed, so a balance should be found between being educational and enjoying the design (Requirement 11).
The user interaction experience is important. The question is how this will take shape, what kind of interaction it will include and how the overall experience will look like. The goal is to create a design that is engaging, memorable and educational. The resulting interactive installation will playfully bring museum visitors awareness of the underlying rules of the complexity of collective behaviour and its emergent and cohesive properties.

Key questions

- *Which parts of collective behaviour are going to be explained in the design?*
- *What will the interactive user experience look like?*
- *How will the embodiment of the installation take shape?*

2. APPROACH

This chapter shows the design process and explains the three main phases of the project.

2.1 DESIGN PROCESS

The project approach can be divided into three main phases: Discover, Ideate & Develop, and Evaluate & Deliver, shown in Figure 5.

Discover

The discover phase focuses on understanding collective behaviour (Chapter 3). Research is carried out about the underlying rules leading of collective behaviour. The goal is to find out which elements are important to create different kinds of patterns. This phase also explores how to design for a museum context and explores different technologies which can be used for the installation. The phase concludes with a list of requirements.

Ideate & Develop

This phase implements the knowledge obtained from the discover phase. Here, the way users can manipulate the collective behaviour (Chapter 4), the goal for the users (Chapter 5) and the design of the museum installation embodiment (Chapter 6) are explored.

This phase is all about the user interaction experience and is highly iterative and uses the build, measure, learn approach (Rogers, 2019). This is a cycle approach (see Figure 4), where ideas are built, evaluated, learned from and improved. This phase explores how users will interact with the design and what it will look like. It concludes with having decided on the composition of all elements present in the museum installation and the user interaction scenario.

Final design & Evaluation

In the last phase the final design is created and is going to be evaluated and validated (Chapter 7). A prototype is built which validates (part of) the user experience and the level of understanding of the users. Lastly, recommendations will be provided for future research and implementation of the design (Chapter 8).

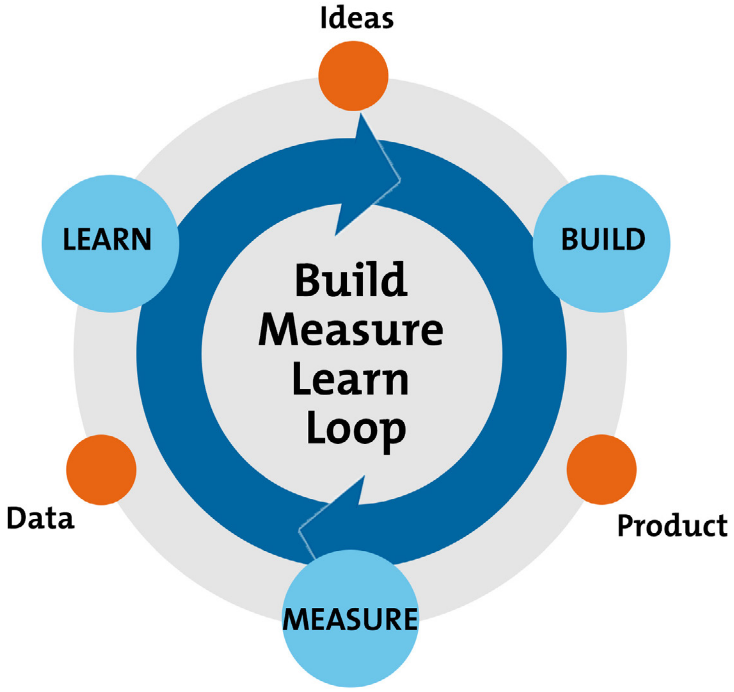


Figure 4: Build Measure Learn Loop (Mind Tools, n.d.)

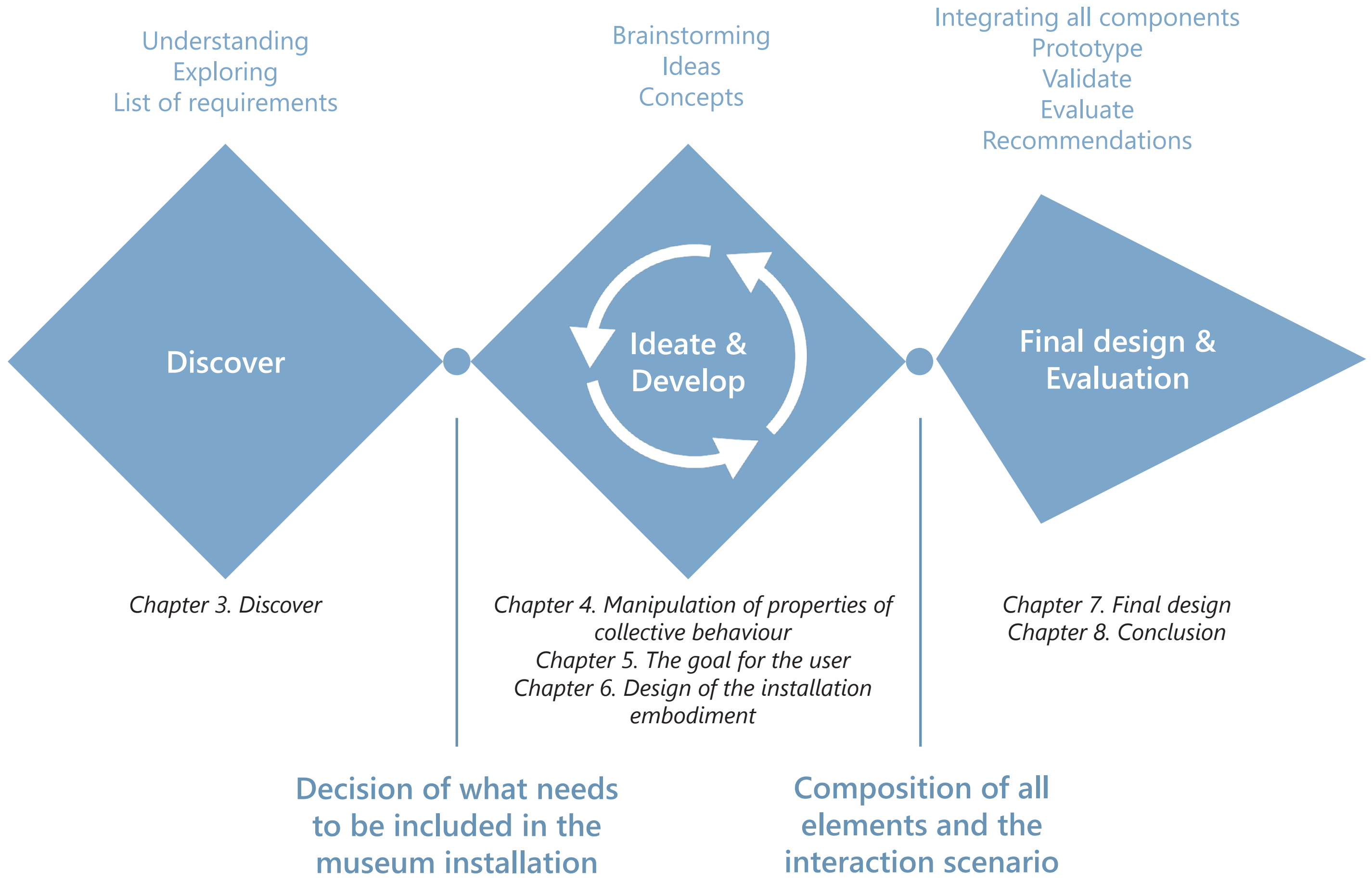


Figure 5: The three project phases

3. DISCOVER

This chapter provides an understanding of the parameters behind collective behaviour, to gain insights into what information to translate in the interactive installation. Technology to use for this installation is explored and the chapter also dives deeper into the question how to design for a museum. To conclude, a list of all requirements is assembled to implement in the design.

3.1 THE RULES BEHIND COLLECTIVE BEHAVIOUR

To understand how the compositions, movements and patterns of collective behaviour are achieved, theories and models have been developed (Rahman et al., 2020). At the moment, one of the most common models is based on metric interaction. In this scenario, individuals interact with all neighbours within a certain distance (Couzin et al., 2002). In another model, which uses topological interaction, every individual interacts with a fixed number of its nearest neighbours (Ballerini et al., 2008; Kumar & De, 2021). Since the metric model serves as a foundation of many current models and it can replicate characteristics of collective behaviour similar to those of natural groups, this graduation project will focus on the metric model.

Repulsion, alignment and attraction

When animals move in a collective, they form an organized group. It has been found that this behaviour of individuals can be explained by three simple behaviours: repulsion, alignment and attraction behaviour (Couzin et al., 2002; Reynolds, 2007). These behavioural rules are based on the position and orientation of individuals relative to each other and are based on visual local information (Couzin, 2021).

When individuals notice others in the zone of attraction, they will move towards their average position, as seen in the bottom image of Figure 6. This attraction zone is modelled as a sphere with the individual in its centre (Figure 7). This behaviour represents cohesion: individuals wanting to join groups and avoid being on the periphery of the group (Couzin et al., 2002).
Whenever an agent spots neighbours in the zone of orientation, a zone which is smaller than the zone of attraction (see Figure 7), it will try to align itself in the same direction as them, shown in the middle image of Figure 6. Lastly, when another agent comes too close and is find within the zone of repulsion (see Figure 7), the individual will steer away and thus always try to maintain a minimum distance from others, shown in the top image of Figure 6.

These steering behaviours combined, lead to the collective patterns seen in groups. For most animals, these are influenced by the evolutionary driving force to minimize the risk of being eaten. For example, fish do not want to be at the rear of the group, because it increases the chance to get eaten. This is why the group moves.

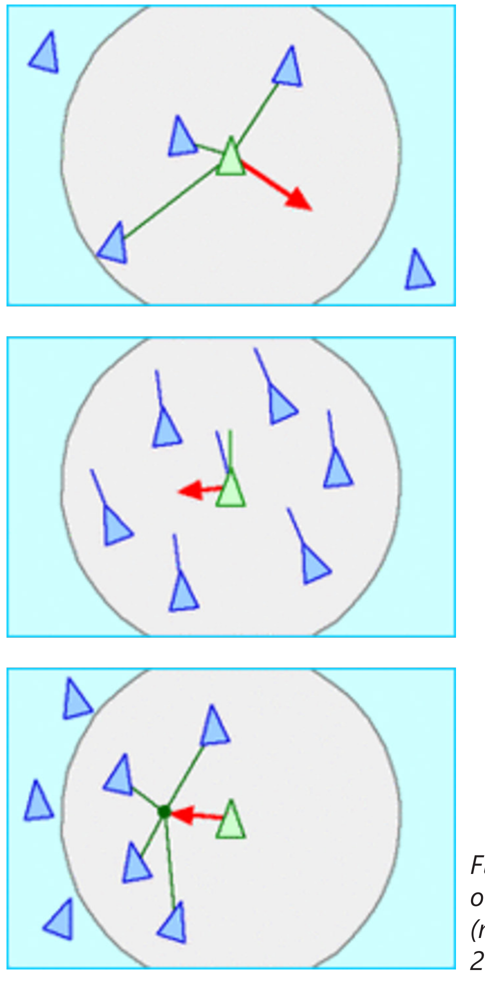


Figure 6: The desired direction when noticing others in the repulsion zone (top), orientation zone (middle) and attraction zone (bottom) (Reynolds, 2007)

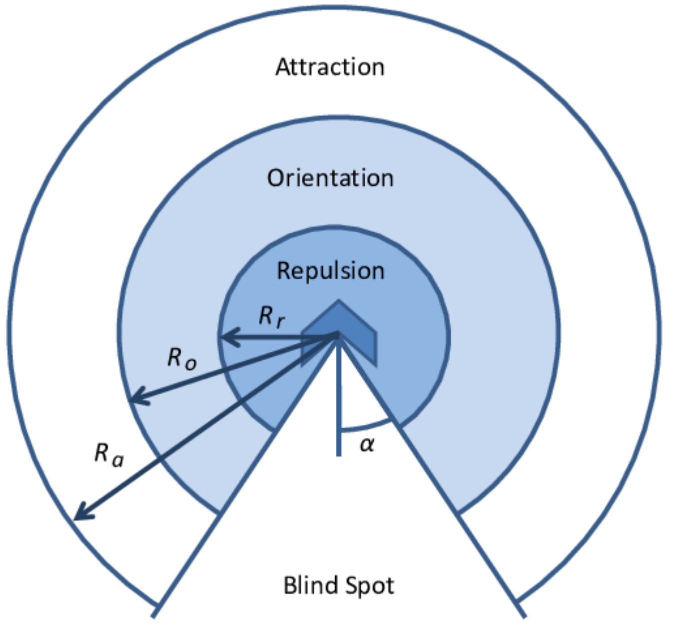


Figure 7: The attraction, orientation and repulsion zone around a centred individual (Pendleton & Goodrich, 2013)

Swarming, polarized, milling

Adjusting the radius of these zones results in three different collective behaviours (Couzin et al., 2002; Tunstrøm et al., 2013), which are visualised in Figure 8.

A swarm has a low group polarization and low angular momentum. Group polarization stands for the alignment between individuals, whereas the group angular momentum represents the sum of the angular momenta of the individuals relative to the centre of the group. In a swarm there is little to no alignment: the individuals are disordered.

In a polarized group, the polarization is high, but the angular momentum is low, since the agents all face the same average direction.

Lastly, in a milling group, the angular momentum is high while the overall polarization is low, but locally present. The group rotates around an empty core. The zone of attraction is large, but the zone of orientation is small.

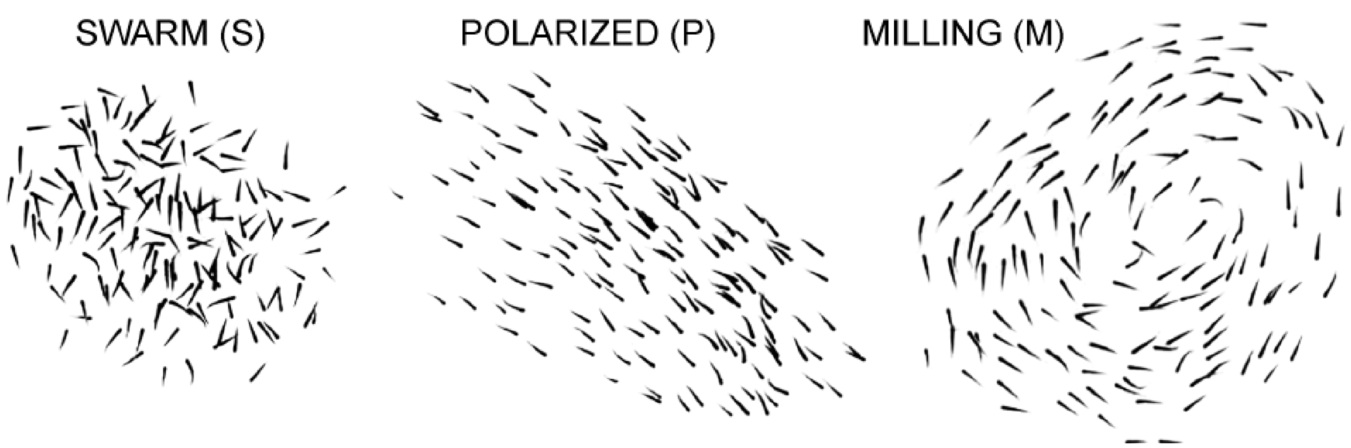


Figure 8: Swarming, polarized and milling behaviour (Tunstrøm et al., 2013)

Transition between states

Experiments have been carried out on this phenomenon and it is found that schools of fish do not stay in one state all the time, but transition between these states (Tunstrøm et al., 2013). This can happen in response to external and internal factors, for instance because of a predator, food, an obstacle or a change in motion of others.

In Figure 9, density plots are visualised showing the proportion of time a group of fish spent in the different states. The x-axis shows the group's angular momentum while the y-axis shows the group's polarization. The time spent in the different states is influenced by the group size, for example, a large group of 300 fish spent most of their time milling, whereas a group size of 70 fish goes through all three states.

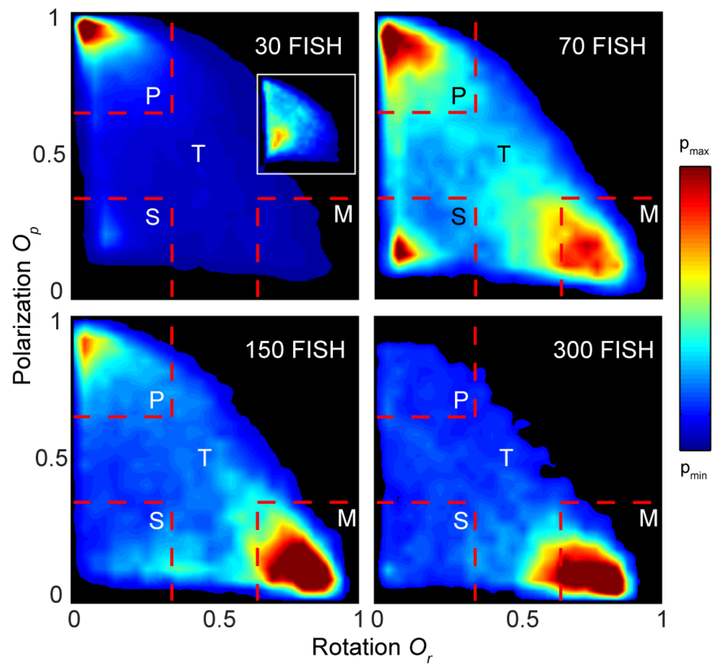


Figure 9: Density plots of the proportion of time groups of fish spend in the different states: polarized (P), swarm (S), milling (M) and transition (T). The colour red represents a longer amount of time spent in the state, whereas blue represents a shorter amount of time (Tunstrøm et al., 2013).

However, the transition between states depends on the previous structure of the group (Couzin et al., 2002). The group has a memory of what state they were in before, while their individuals do not since they have no indication of what the whole group looks like.

For instance, when the radius of the zone of orientation is increased from 1 to 1.5, the group transitions from a swarm (low polarization and low angular moment) to a milling behaviour (low polarization, high angular moment), as seen by the continuous line in Figure 10. If the zone of orientation is increased even more, the group transitions into a polarized group (high polarization, low angular moment).

In contrast, if the group's radius of orientation is decreased from a high value (dashed line in Figure 10), the polarization of the group stays high, and the angular moment stays low, keeping the group in the polarized state. Only when the radius of the zone of orientation drops below 1.5 will the group polarization suddenly decrease, making the group transition to the swarm state. Therefore, even though the size of the parameters might be the same values at two different times, the group state can differ depending on its previous state.

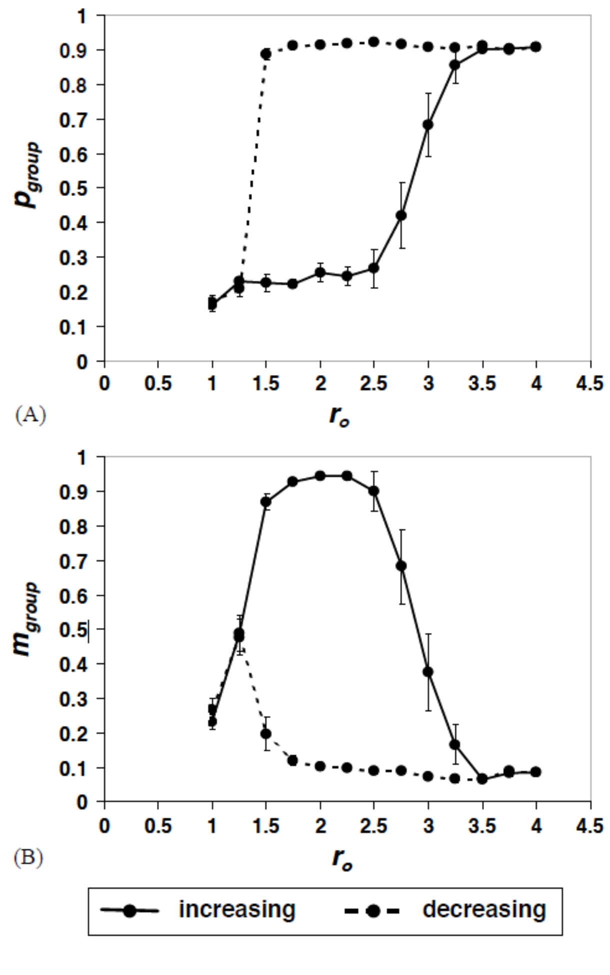


Figure 10: The polarization and angular moment of the group when increasing (continuous line) and decreasing (dashed line) the radius of the zone of orientation. The state that the group goes in depends on the previous state of the group (Couzin et al., 2002).

Field of view

Animals are not able to see all around them. At the front, they can see with both eyes in the binocular field. On the sides, they can see with only one eye: the monocular field. A blind spot, shaped like a cone, is found behind the individual where it can not see, shown in Figure 11 (Pita et al., 2015).
The differences in the size of these fields result in different patterns being formed by the group. An example can be found in Figure 11 and Figure 12. A difference between the size of the blind spot of a zebrafish and a European starling can be seen. With a larger blind spot, parallel groups form a longer shaped group in the direction of movement (Couzin et al., 2002). When the blind spot becomes too large, the group has the possibility to lose its cohesiveness.

Turning rate, speed and noise

The turning rate and speed of individuals are inversely proportional to each other. With a low speed, a higher turning rate is achieved. But, this consequently increases the angular noise, which can lead to a disordered group. High speeds, with low turning rates, will in contrast dampen noise and facilitate order, since there are restrictions to turning (Klamser et al., 2021).
With little noise, parallel groups can already form at a relatively low zone of orientation. When the noise gets too high, swarms will form, or eventually the group will fragment. Larger zones of orientation and attraction are then needed for stable groups (Couzin et al., 2002).

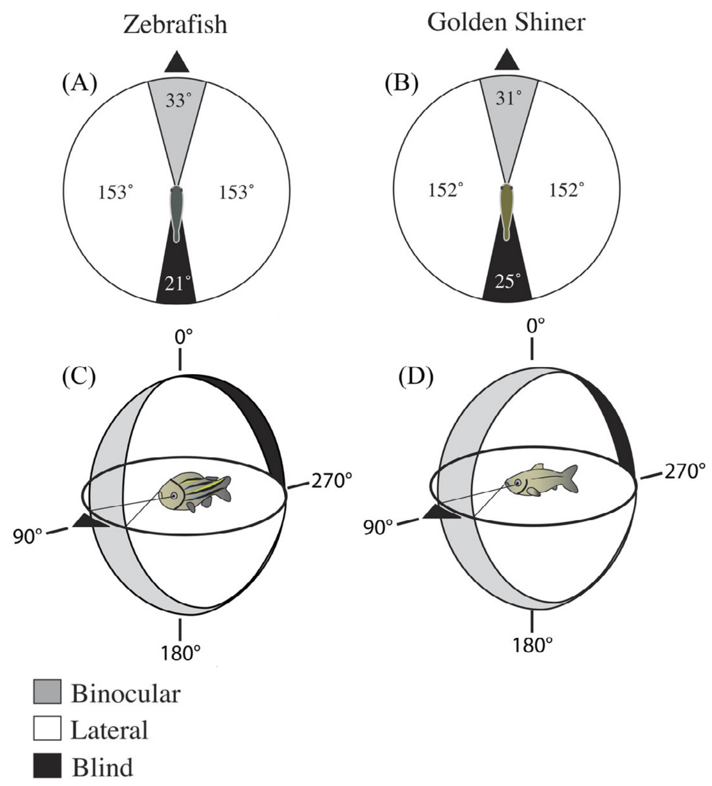


Figure 11: The angles of vision for Zebrafish and Golden Shiners (Pita et al., 2015)

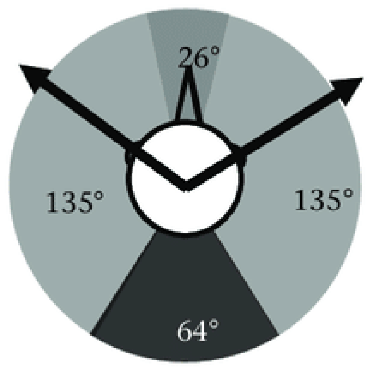


Figure 12: The visual field of a European starling (Butler et al., 2016)

Information transfer

Alignment is important to minimize collisions, but it is also important for information transfer. Accurate information transfer has its advantages for foraging and avoiding predators.
For example, when an individual decides to suddenly turn, its neighbours will also orientate in the same direction, without having detected the stimulus to change direction themselves. In a swarming type of group, this information will be transferred less efficiently compared to a polarized one (Couzin et al., 2002).
However, there can be a conflict in the preferred direction between group members. If the number of individuals for both preferences is the same, the group will move in the average direction. If the degree of conflict increases, the group will move randomly to one of the directions.
However, when there's a majority of preference in a certain direction, the group will favour that direction (Couzin et al., 2005; Sumpter, 2008).

A study has found that for three-spined sticklebacks which are looking for food, the first fish will find food faster when the group is in a swarm state. Since the group is not facing the same general direction, their visual fields cover more of their environment. However, once the information of the location of food comes available, the time it takes for the rest of the group members to find this food source is faster in ordered polarized groups. So, for the whole group to find food, a polarized state is more advantageous (MacGregor et al., 2020).

Other differences between individuals

Besides conflict in the preferred direction of the group, individuals also differ from each other between the other parameters. These differences influence their position in the group (see Figure 13). If an individual has a faster speed, it will be in the front of the group, further from the centre. The lower the turning rate, the more the individual will be at the front of the group and slightly further away from the centre. Individuals with a higher error will take positions towards the rear of the group. As for the three zones of steering behaviours, only the zone of repulsion and the zone of orientation influence the individuals' position in the group. The smaller the zone of repulsion, the closer the individual is to the front of the group or the centre. The size of the zone of orientation is slightly negatively correlated with being in front of the group, but strongly positive as variance increases (Couzin et al., 2002).

Because there are many parameters at play behind the motions of collective behaviour. A balance should be found in not making it too complex for users to understand the concept.

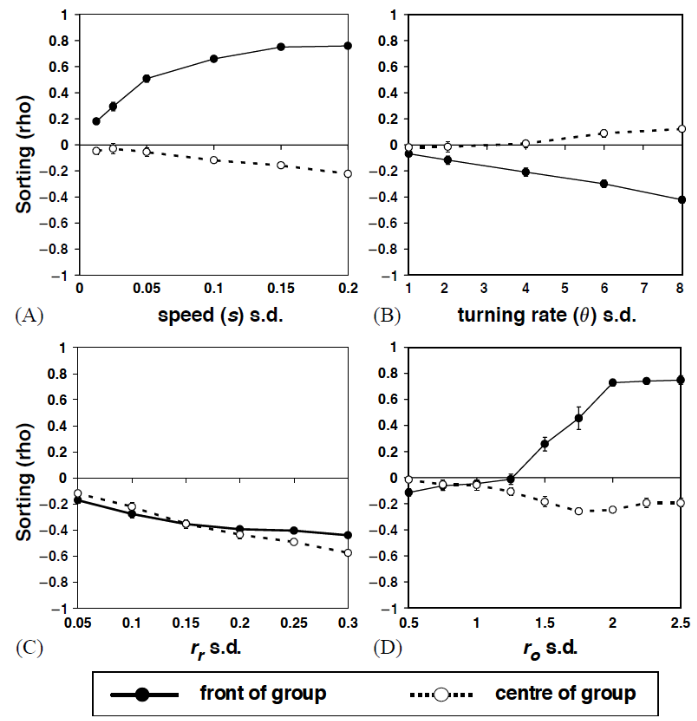


Figure 13: Factors influencing being at the front of the group and centre of the group (Couzin et al., 2002).

3.2 EXPLORING TECHNOLOGY

3.2.1 Arduino

Since the installation is going to be interactive, the user should be able to provide input. This can be achieved through different uses of technology. One of these options, which is going to be used is Arduino.

Arduino is an open-source platform, which consists of hardware and software. The Arduino boards, which are microcontrollers, can read inputs from sensors (e.g. if a button is pressed) and turn them into outputs (e.g. turn on a LED). This will be a useful way for users to control the installation. With the use of the Integrated Development Environment software, scripts can be written to let the Arduino board know what to do. Many different boards are available where sensors and actuators can be plugged into. These boards can be connected to a computer via USB to upload the code and read values, while some can also send data via Wifi or Bluetooth (Arduino, 2018; Ben, 2013).

Grove is a ready-to-use toolset, which aids the ease of connecting sensors (Seeed Technology, 2022). With a grove interface, sensors and actuators can be plugged straight into the board (Figure 14), without requiring to use a breadboard which is usually needed when using an Arduino board (Figure 15).

The movements and patterns of collective behaviour need to be visualised in the installation (Requirement 5). An algorithm, which uses the underlying behaviour rules to replicate collective behaviour in a visual simulation is needed. This algorithm, combined with the input from Arduino will serve as the base of the installation.

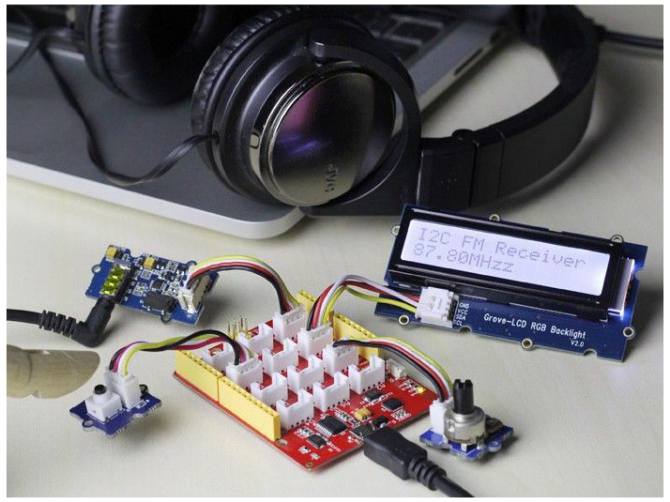


Figure 14: Seeeduino Lotus board with grove interface (Seeed Technology, 2022)

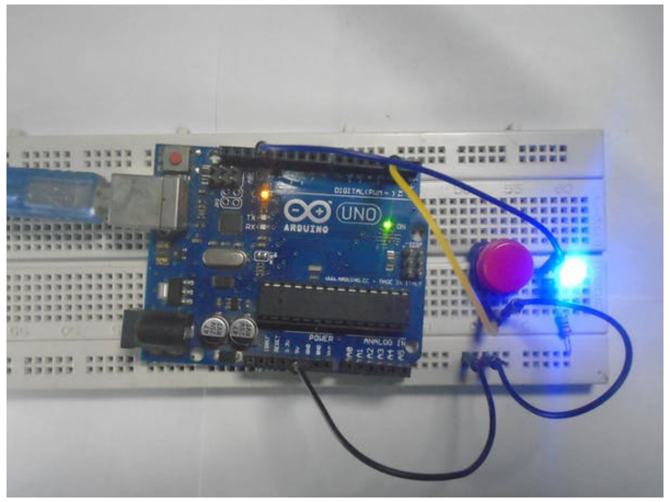


Figure 15: Arduino project with a button and LED (Aqib, 2018)

3.2.2 Algorithms

To show the patterns and behaviours which are based on the input from users in a moving visualisation, an algorithm is needed to create this visual simulation.

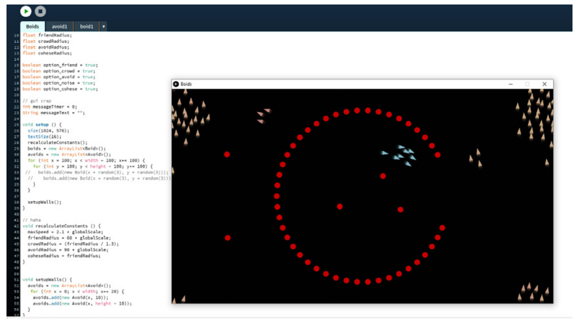
Many algorithms are based on the boid algorithm (Reynolds, 2007). This is the bird-like algorithm that implements the three rules of repulsion, alignment and attraction.
The models work in the following way (Abreu, 2021):

- Repulsion: when another agent comes within a certain distance, move in the other direction
- Alignment: move in the average direction of the other agents
- Attraction: move to the average position of nearby agents

When the three rules are combined for all individuals, it will result into the collective patterns. In the next section, scripts based on this boid algorithm are explored. They were analysed on:

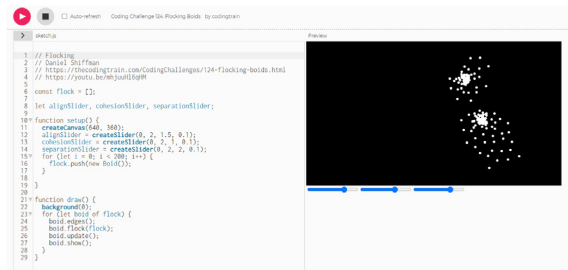
- Ease of making adjustments to the algorithm to tweak it for use in the installation
- Type of program and coding language used
- Look of the simulation

Processing (Jackaperkins, 2020)



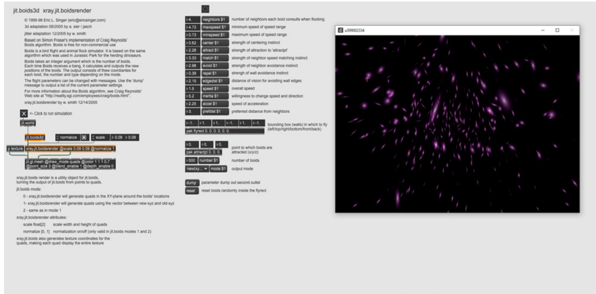
- The orientation, alignment and attraction rules could be turned on and off
- Ability to add boids and add obstacles
- Java language
- Code is simply written, adjusting the code is easy
- Agents are visualised as triangles, which shows the direction of the movement
- Processing is a free program

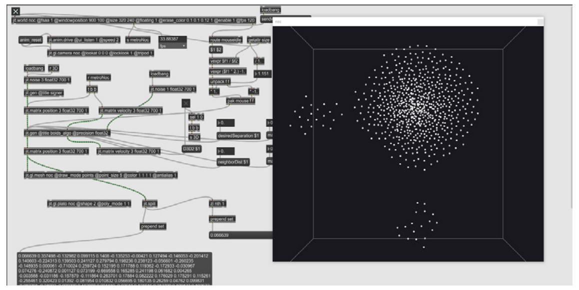
P5.js (The Coding Train, 2018)



- The strength of the repulsion, alignment and attraction behaviour could be adjusted with sliders
- Agents are visualised as circles, the direction of movement could not easily be interpreted
- Javascript language
- Code is more complex, but logically written
- Can run in the web browser: easily accessible and free

Max/MSP



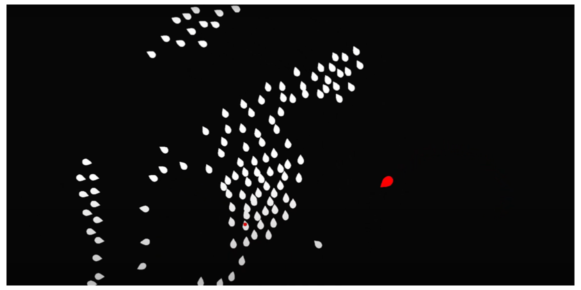


- A visual programming language: uses a data flow system where objects are connected
- Ease of use of implementing Arduino sensors for input
- Program is not free

- Algorithm 1: Patterns of individuals were not clearly visible (Freeman, 2014)
- Looked like agents were colliding with each other in the 3D space
- Difficult to distinguish between agents

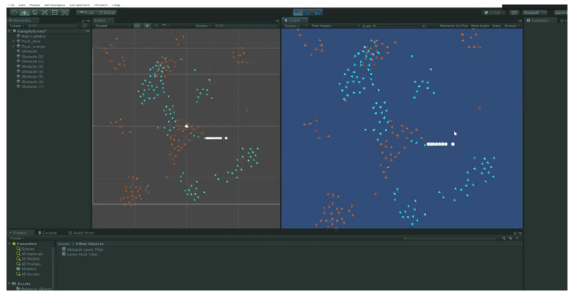
- Algorithm 2: Agents are visualised as spheres, the direction of movement of agents is not shown (Foderaro, 2019)
- The code is built in a complicated way, which is difficult to adjust

Godot (Sergioabreu, 2021)



- Open source video game engine
- GDsript language
- Godot is a free program
- More complicated written code
- Predator is included in the code

Unity and visual studio (Boardtobits, 2019)



- C# language
- Difficult to alter, complicated code with multiple scripts
- Visual studio is not free
- Includes obstacles, but the behaviour around these did not look natural, since the obstacles were coded as points in space

3.2.3 Decisions for the simulation

Basics of collective behaviour

Something to note is that all these algorithms used the repulsion, orientation and attraction rules, but they did not implement a parameter for a maximum angle of rotation. Secondly, the blind spot was also not taken into account. The agents moved as if they could detect neighbours all around them.
However, by manipulating the repulsion, orientation and attraction rules, the simulation did showcase different kind of compositions and patterns that closely replicate real-life swarm and flocks. This implies that with just these three steering behaviours, collective behaviour can already be formed, even though other parameters also have an influence on the behaviour.
This lead to the decision to let users only manipulate the three steering behaviours, while keeping the other parameters static. Since these repulsion, orientation and attraction behaviours are the foundation of many models and can generate a wide range of different collective patterns, these will be used to explain the basics of collective behaviour, without making it too complex for users (Requirement 11).

2D versus 3D

Models for simulations in two dimensions and three dimensions were analysed. While manipulating the collective behaviour, it was found that in 2D it was easier to perceive the influence of the repulsion behaviour. It was clearly visible the agents tried to keep a minimum distance between each other. However, when agents moved behind each other in the third dimension, it made it more difficult to comprehend they indeed stayed at a distance from each other and did not collide. The position of agents in relation to one another was more difficult to perceive in 3D. Comparing these simulations to collective behaviour seen in nature, flying birds (Figure 16) and swimming fish will move around in three dimensions, while a herd of wildebeests and a flock of surf scoters on the water surface (Figure 17) collectively behave

Figure 16: A flock of starlings (Langston, 2016)

Figure 17: A flock of surf scoters moving on the water surface, seen from above (Lukeman et al., 2010)

in two dimensions. This shows, that collective behaviour is apparent in both two dimensions and three dimensions. To not overcomplicate the design (Requirement 11), it was decided that for this graduation project, the third dimension will be kept out of the scope, and the focus will lie on the movements in two dimensions.

Direction of movement

Something else to note is that some simulations did not show the direction of movement in the simulation of the agents. At a glance, the alignment of the members in the group could not be perceived. This makes it difficult to see the influence of the orientation rule. When the agents were visualised as triangles, the direction of movement was indicated by this shape and the alignment between members could be perceived. This is something important to implement in the installation.

3.3 HOW TO DESIGN FOR A MUSEUM

3.3.1 Museum inspiration

To reach a wide audience with the educative installation, it will be designed for in a museum, in particular Naturalis, a museum on biodiversity (Naturalis, 2022).
Naturalis has been visited to get an impression of the museum to be able to design an installation that fits in. Two other museums have also been observed to find out what is important to take into account and implement in the museum installation.

Naturalis Biodiversity Center, Leiden

Naturalis makes use of dark rooms (Figure 18 and Figure 19) and spotlights high on the ceiling. This helps highlight important features, making the visitors' attention go to the right place. Naturalis mostly makes use of screens, projections, skeletons of animals and stuffed animals (Figure 18 and Figure 19). Their target audience are families with children. The installations are mainly targeted for 8-year-olds and older (D. Suijker, personal communication, January 20, 2022). At some installations, extra information is provided by someone from the museum. Some of their installations have a clear ending, so users move on to the next thing. However, some interactive installations do not have an ending, giving the user more time to explore. On average visitors spend about 2.5 hours in the museum.

Figure 18: Spotlights on object of interest, printed backgrounds with projections in Naturalis

Figure 19: Room with a large screen in Naturalis

NEMO Science Museum, Amsterdam

NEMO has many educative installations, where you can learn something small in a fun way, an example is shown in Figure 20. NEMO's sections are themed, which helps visitors relate to the different installations, and have something to remember it by.

Figure 20: Learn what light consists of by mixing multiple colors (NEMO, n.d.)

REMASTERED experience, Rotterdam

REMASTERED is a museum with many visual effects and interactive rooms (Figure 21). This museum mostly focuses on aesthetics, the experience does not teach visitors about the shown concept. When many visitors interacted with an interactive touch installation, it was found that it was not always working smoothly (Figure 22). This shows that it is important to design for multiple users, or make sure only a small number of people can interact with it. The installation should not malfunction when it receives too much input, however the way a limit is implemented should not confuse the user (Requirement 6).

Figure 21: Fish moved away from touch

Figure 22: The interactivity was not always working

Literature

Research has found that museum visitors need to stay longer than 3 minutes in one room, to later recall that information. So, a minimum amount of time is needed to go from an impression to a consumption, and make visitors remember what they have learned (Requirement 8) (Pierdicca et al., 2019). However, this research was based on a room, and not one installation in particular. Other research has stated that we remember our experience in a series of snapshots, rather than remembering all events (Kane, 2018). The emotional impact of an event and the end of an experience are remembered the most easily. Designing for positive emotions and the end of the interaction are therefore very important (Requirement 9).

3.3.2 Experts on design

Experts on museum experience design and design for children were asked for their expertise, to gain insights in translating research of collective behaviour into an educative interactive museum installation which is targeted for people of age 8 and older.

Expert museum experience design
(A. Vermeeren, personal communication, November 25, 2021)

- It is important users can relate to the design: make it familiar and recognizable.
- Make the whole experience part of the story.
    - Creating a story for the user to interact with, gives them something to relate to. A story also stimulates the imagination and helps while learning, since it aids people to better process and remember new information (Schep et al., 2015).
- A framework to use for open-ended play: invite, explore, immerse.
    - Make sure people are invited to the installation and that they want to try it out (Requirement 10). An exploring stage is implemented to figure out how it works. Once this is achieved, the users can immerse themselves in the installation.
- A method to design for a 'wow moment' is the Body, mind, heart and soul framework (Bär et al., 2018).

Expert design for children
(M. Gielen, personal communication, November 29, 2021)

- Find the different layers of understanding, which can go from easy to understand to a more into depth understanding. This can help create something that fits everyone's abilities (Requirement 2). Younger children do not have to understand the most difficult layer, but will still learn from interacting with the design.
- Interacting with the installation should not feel right or wrong, because this can lead to associating the interaction with negative feelings.
    - Users should feel encouraged (Requirement 9).
- The goal is to make learning fun while being in the museum and motivate every user to discover more about the world: create a unique learning environment.

Body, Mind, Heart and Soul framework
This method comes from the book World of Wonder: Experience design for curious people (Bär et al., 2018) recommended by Arnold Vermeeren, an expert in museum experience design research. This method is about creating a 'wow moment', which is a great way to transfer information which will be remembered, in a short amount of time. For this moment to be created, the design should focus on these four elements: Body, mind, heart and soul.

Body: How can the subject appeal to the senses?
This element focuses on stimulating the senses to get users invested.

Mind: What is interesting about the subject?
This element focuses on what it is about: the content or story. The mind should be challenged in a way to spark interest. With the use of multiple layers everyone can approach it in a different way.

Heart: What moves them and how can we make it feelable?
This element focuses on the emotion. When the right emotions are evoked, the experience will be more (positively) remembered. It helps if users can relate themselves to it.

Soul: Why is this topic meaningful?
This is about the relevance of the design, and about making visitors realize that the experience is part of a larger whole.

3.3.3 How to design a learning environment for all museum visitors?

One of the main goals of the museum installation is to teach the visitors something new. However, not everyone learns in the same way. David Kolb defines a cycle of four stages of learning: concrete experience, reflective observation, abstract conceptualization and active experimentation, as seen in Figure 23. This cycle can be entered at any stage but should be completed for the full learning process (Kurt, 2020).

In the concrete experience, new knowledge is acquired by actively engaging in the task or activity. During the reflective observation, the experience will be reflected. This can be done by asking questions or discussing and watching the experience of others. In the abstract conceptualization phase, the learner tries to make sense of the events that happened and form conclusions. In the last stage, the active experimentation, conclusions to the new experiences will be applied. This is done in a test setting (Andersen, n.d.; Cherry, 2020).

Since everyone has a preferred way of learning, the following learning styles can be derived from the cycle (Chapman, 2017; Cherry, 2020):

Diverger (feel and watch) - Why?
- Assesses experiences from various perspectives
- Works collaboratively
- Likes brainstorming and exploring
- Prefers to watch and gather information
- Is imaginative

Assimilator (think and watch) - What is there to know?
- Uses reasoning and a logical approach
- Independent
- Interested in abstract ideas
- Requires a good and clear explanation
- Good at understanding information
- Time to think it through

The converger (think and do) - How?
- Problem-solving approach
- Focuses on solutions
- Interactive instead of passive
- Prefers practical tasks

Accommodating (do and feel) - What would happen if I do this?
- Doers, hands-on
- Adaptable and intuitive
- Changes their plans accordingly to new information
- Trial and error approach
- Actively engaged

Designing for all four types of learners helps to cater to a learning experience for all visitors (Requirement 2).

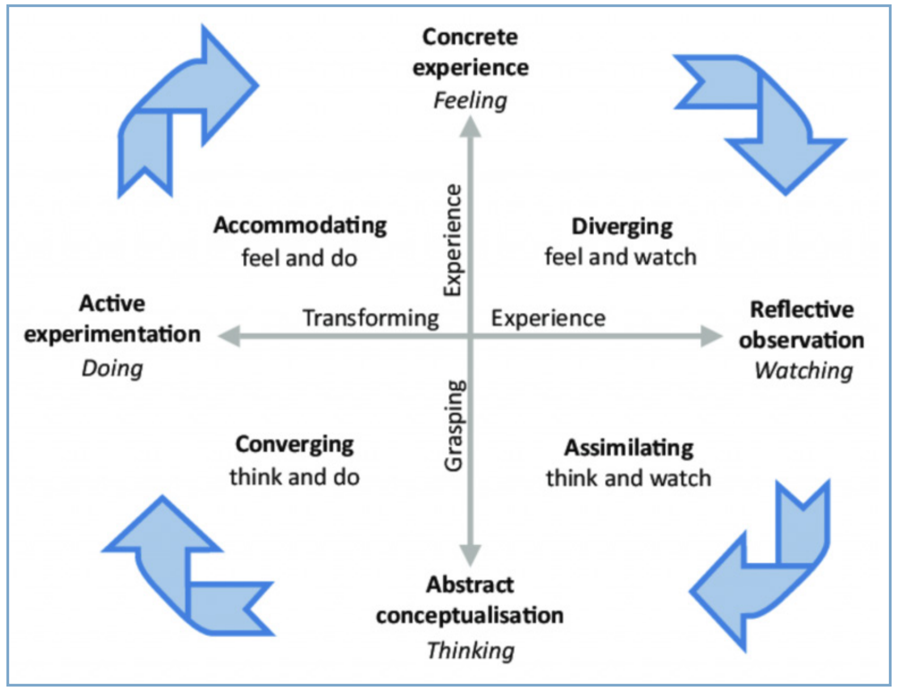


Figure 23: Kolb's learning stages with its four learning styles (Kolb, 2020)

3.4 WHAT TO INCLUDE IN THE DESIGN

Concluding from the previous chapters, the parameters that the users will be able to manipulate are those for the repulsion, orientation and attraction behaviours (Requirement 3). The complexity of collective behaviour will this way be explained via simple underlying rules.
An important aspect for users to understand here is that every individual in the group takes notice of their local surroundings, and responds accordingly (Requirement 4). By adjusting this response, the composition and movement of the group will change. If users can directly manipulate this behaviour and note the change in the group, they can see the link between them and thus obtain insights into what matters in the shaping of patterns of collective behaviour.
Since the museum installation should fit the abilities of all its visitors (Requirement 2), multiple levels of understanding will be implemented in the museum installation. These levels go from easy to understand to having a deeper understanding, making the design educative for everyone.

1. If I make an adjustment, I change the desired direction of every individual in the group
2. The thing I change is the response of an individual's behaviour to their local surroundings
3. If I change this variable to a certain size, it will result in a certain composition and pattern change in the whole group
4. I comprehend how the combination of all variables lead to a certain result

3.4.1 Requirement list

To summarize what to implement in the museum installation, the following requirement list is assembled.

Requirement 1: The design fits in a museum context, in particular the Naturalis museum.

Requirement 2: The design should be educative in a way for all museum visitors of age 8 and older.

Requirement 3: The design teaches about the underlying rules of collective behaviour in two dimensions: repulsion, alignment and attraction behaviour.

Requirement 4: The design makes people understand that the group patterns can be derived from the local interactions of every individual.

Requirement 5: The design includes visual elements where the moving collective behaviour patterns are displayed.

Requirement 6: The design should be interactive and provide feedback. The interactivity should be designed in a way that it can handle all inputs from the users and not get overloaded.

Requirement 7: The design interaction should be intuitive, so users quickly understand how to interact with the installation.

Requirement 8: Users should stay engaged and interact with the design for a minimum of 2 minutes to obtain understanding, and no longer than 5 minutes to not disrupt the flow of the museum.

Requirement 9: The users should end the interaction with a positive feeling and feel encouraged while interacting with the design.

Requirement 10: Users should relate in a way to the design, to spark their interest and feel invited to try it out.

Requirement 11: Users should not feel overwhelmed by the design.

4. MANIPULATION OF PROPERTIES OF COLLECTIVE BEHAVIOUR

Users will be able to influence the collective behaviour by manipulating the behaviours of repulsion, orientation and attraction. In this chapter, evaluations with users have been carried out to find out in what way they will manipulate these. It was looked into if the users understood what part of the behaviour they were adjusting and if they could comprehend how it influenced the composition and patterns of the whole group.

4.1 MANIPULATION OF THE REPULSION BEHAVIOUR

This evaluation was carried out to decide in what way the repulsion behaviour would be manipulated by users.
To show the change in the moving group, a simulation of moving triangles using the boid (bird-like) algorithm was used (Jackaperkins, 2020). This algorithm was altered so the size of the repulsion zone could be adjusted by the users. They could do this by rotating a button that was connected to an Arduino board (see Figure 24).
What was noticed here is that users quickly found the connection between the difference of the repulsion size and the influence on the group dynamic. They anticipated correctly what would happen (understanding level 1 and 3). The resulting behaviour was being compared to everyone having their own personal space, which they could relate to. It was therefore clear to the users that every individual responded to their own surroundings, not wanting to collide with others (level 2). The expected change was also clearly visible to the users in the moving simulation, which made the users feel like they were in control.

Test setting

- ◊ 3 participants aged between 23-25 years old
- ◊ 2 participants aged between 55-57 years old
- ◊ Laptop to show moving simulation in Processing
- ◊ Seeeduino Lotus with a rotary potentiometer, connected to a laptop via USB

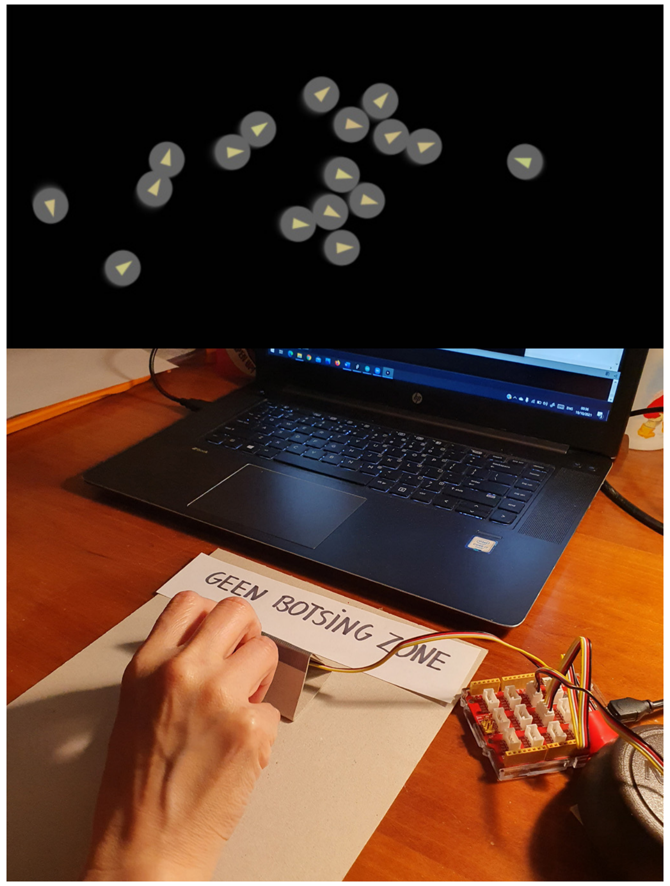


Figure 24: Changing the size of the repulsion zone

4.2 MANIPULATION OF THE ORIENTATION BEHAVIOUR

In this evaluation, the way to manipulate the orientation behaviour was explored.
The algorithm was altered so users could adjust the strength of the orientation behaviour, by moving a slider back and forth. Changing this value on its own made users realize that with a high strength value the movement of the group was streamlined and aligned with their neighbours (Figure 25). With a low value the group seemed to be moving in all different directions (levels 1 , 2 and 3). Nevertheless, when changing the strength alignment variable in combination with the repulsion variable, the algorithm did not simulate the correct patterns which were supposed to happen with this combination of variables. This confused the participants. Therefore, finding out of level 4, a combination of all variables, was comprehended, could not be concluded in this evaluation.

A different algorithm was explored where users were able to change the size of both the repulsion and orientation zone (de Weerd, 2016). This is shown in Figure 26. For users, the distinction in what they were adjusting became more apparent this way.
However, while adjusting the size of the orientation zone, a complex aspect happens which depended on the composition of the group. A change in the group was easily seen by users when the orientation zone was changed from small to large and back (from a swarming state, to a polarized state, back to the swarming state). But when adjusting the size between medium values, it was difficult for users to note a change in the moving simulation. The group would stay in the polarized group state and not change much. In comparison to a group of fish, when increasing the zone of orientation, the group should transition to a milling state. However,

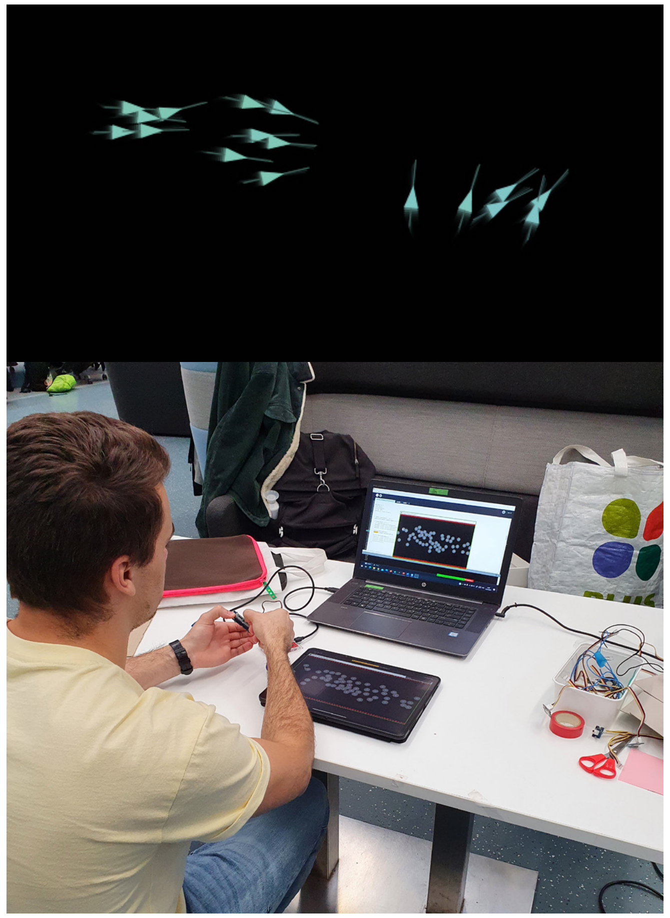


Figure 25: Changing the strength of the alignment steering behaviour

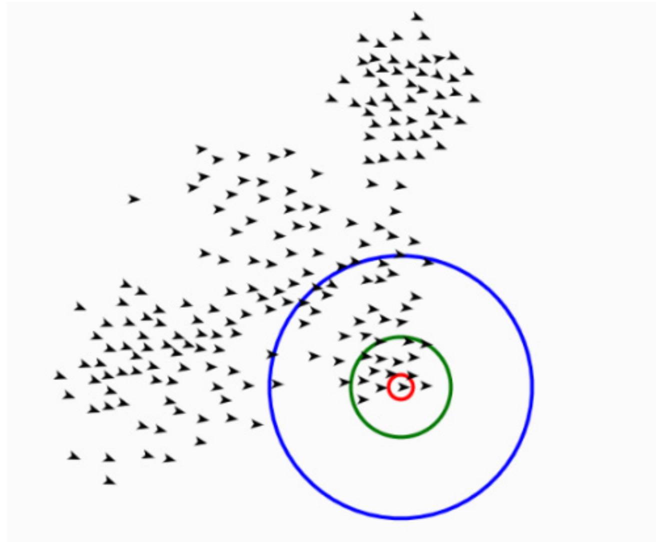


Figure 26: Visualisation of the size of the zone of repulsion (red), alignment (green) and attraction (blue)

this milling behaviour was difficult to mimic with the algorithm to always happen when increasing the orientation zone. This could be due to the fact that the field of view of the agents is 360 degrees, and they are influenced by all neighbours around them, which is not true for a group of fish.
To make it more clear how the orientation behaviour has an influence on the group, it was decided to implement the manipulation by being able to switch between a small and large orientation zone. This way, in combination with the other values (level 4), the orientation steering behaviour is easy to comprehend, by seeing the difference of agents aligning themselves to their surrounding neighbours and not aligning themselves.

Test setting
◊ 3 participants aged between 23-25 years old
◊ 2 participants aged between 55-57 years old
◊ Laptop and iPad to show moving simulation in Processing (algorithm 1)
◊ Laptop to show moving simulation in p5.js (algorithm 2)
◊ Seeeduino Lotus with a slide potentiometer meter, connected to a laptop via USB (algorithm 1)
◊ Seeeduino Lotus with 3 rotary potentiometers, connected to a laptop via USB (algorithm 2)

4.3 MANIPULATION OF THE ATTRACTION BEHAVIOUR

This evaluation was conducted to find out how users would manipulate the attraction behaviour.
In the first evaluation, users were able to manipulate the attraction steering behaviour by turning this behaviour on and off, which they could do by pressing a button. The resulting behavioural change of the group was clearly seen and understood (see Figure 27). They noticed that when the attraction rule was turned on, agents moved into groups, and when it was turned off, agents stayed further apart from each other (layer 1 and 2). To bring the explanation of the attraction steering to a higher degree, by not only turning it on and off, the second algorithm was used (Figure 28), so users could manipulate the size of the zone where the attraction behaviour takes place.
They could do this by rotating the corresponding button (Figure 29). Users understood that with a small zone of attraction, the groups formed small groups, and when the zone of attraction was increased, bigger groups were formed (level 1). However, here users had more difficulty understanding level 2. Understanding that the attraction steering behaviour applies to everyone in the group, so also to the agents on the periphery of the group, was more difficult to grasp. This needs to be understood first, before the user can understand the third and fourth level.

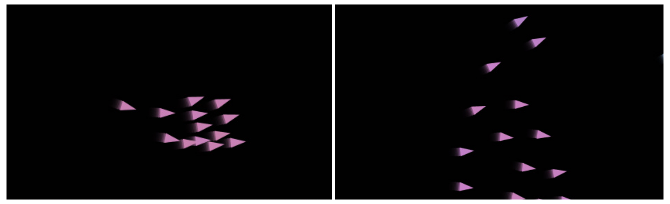


Figure 27: Attraction behaviour rule turned on (left) and attraction turned off (right)

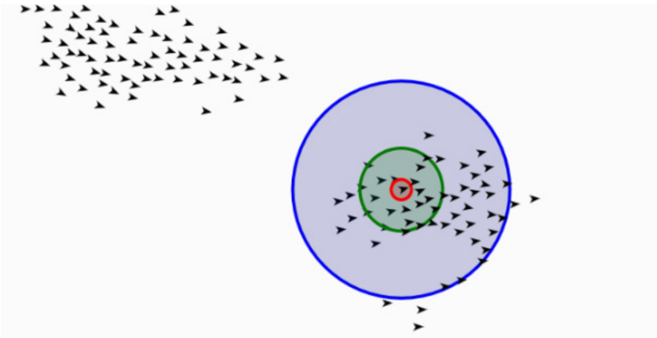


Figure 28: Simulation where the sizes of the repulsion, attraction and alignment zone can be changed

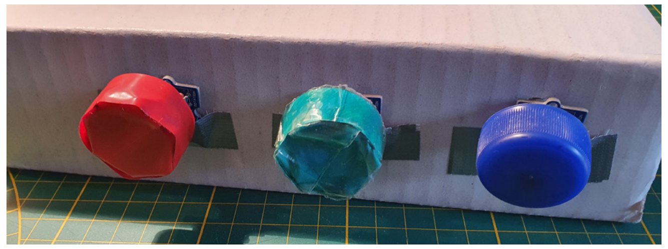


Figure 29: Rotatary buttons to adjust the size of the zones

Test setting
◊ 3 participants aged between 23-25 years old
◊ 2 participants aged between 55-57 years old
◊ Laptop and iPad to show moving simulation in Processing (algorithm 1)
◊ Laptop to show moving simulation in p5.js (algorithm 2)
◊ Seeeduino Lotus with a button sensor, connected to a laptop via USB (algorithm 1)
◊ Seeeduino Lotus with 3 rotary potentiometers, connected to a laptop via USB (algorithm 2)

Takeaways
- Repulsion: change the size of the zone of repulsion
- Orientation: choose between a small size or large size of the zone of orientation
- Attraction: change the size of the zone of attraction
- Make sure users realize this behaviour applies to every individual in the group

5. THE GOAL FOR THE USER

Next up, users should have a goal of why to adjust the variables and seeing the corresponding patterns of the collective movement. While interacting with the design, users should feel engaged in a way to become curious and obtain a deeper understanding of how, why and when certain patterns take shape.
Users should stay engaged long enough for them to have time to learn something new. Creating a story for the user to interact with, gives them something to relate to and remember it by. This storyline exploration went through different iteration cycles to come to the right one.

5.1 STORYLINE EXPLORATION

After brainstorming many different storylines and exploring different gameplays (see Appendix 1), the following directions were chosen.

- The group has to move away from something. Here the user will notice that if only a few of the group notice a threat and turn in a different direction, information will transfer through the group, making the whole group change their direction. However, the way this information is spread depends on the composition of the group and the way every agent responds to its surroundings. For instance, if the polarization of the group is high, the agents will adapt their direction quickly to their neighbours.
- The group has to move towards something. Here it is about letting the user know that being part of a group makes it easier to find something since more eyes are looking. Something to find out here is what kind of group formation makes it easier and faster for this to happen.

These directions were translated into the five contexts shown in Figure 30.

These ideas were accessed with a group of 19 people, to find out which storyline sparked interest and made the most sense for the context of learning about collective behaviour. This evaluation was carried out through an online questionnaire with students ranging between the age of 20 and 30 years old.

When assessing these first ideas, the main feedback of the evaluation was that users were interested in the whole group and their movements. Gameplays where it was possible for just individuals of the group to collect something, did not spark their interest since they did not see how changing the variables and the group formation would benefit the group. The composition and movements should be important to the whole group since that is the point of them being in a collective.

Takeaways

- Users are interested in the whole group, not the individuals on their own
- Composition and movement should benefit the whole group

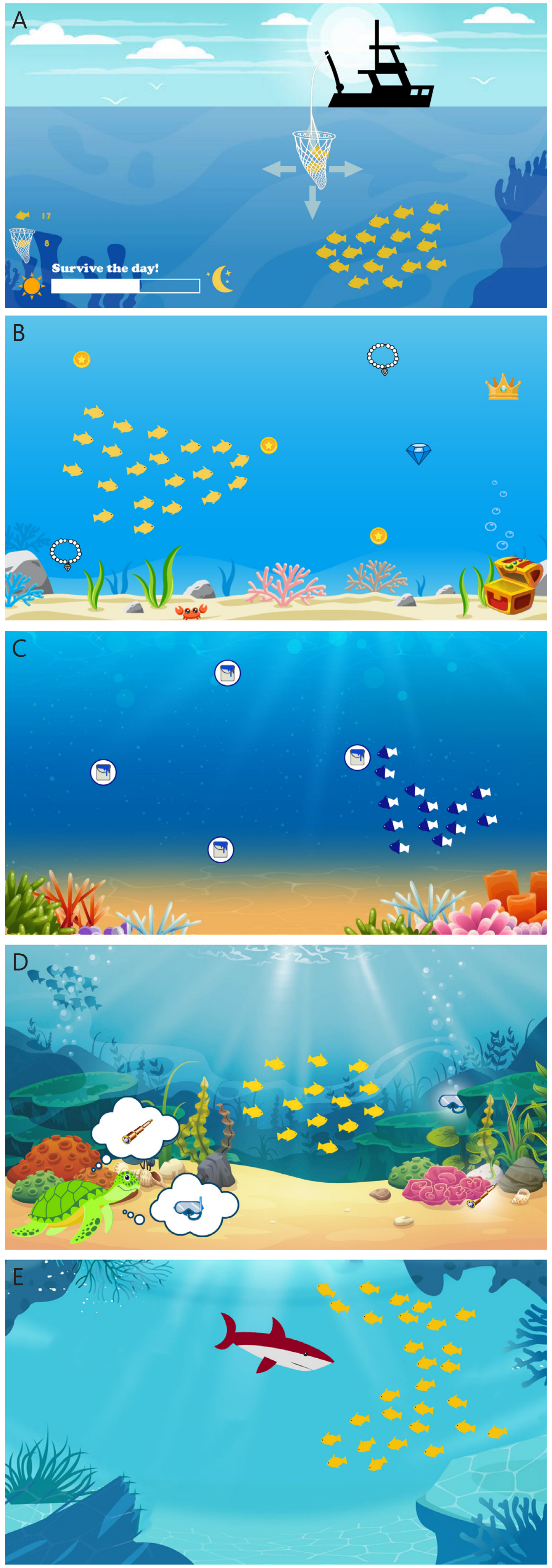


Figure 30: (A) Don't get caught in the net and survive the day. (B) Treasure has been lost. Explore the ocean and find the items. (C) The fish have lost their colour, help them find their colour back. (D) Quest: here a search factor is added to the gameplay. (E) Protect the school of fish against a predator! Here the threat is constantly moving.

5.2 FINDING THE RIGHT THEME

New ideas were created, now keeping in mind the benefit for the whole group. Taking into account why animals form groups in the first place, it all comes down to survival. With safety in numbers, it is safer against predators, easier to find food, and less energy is needed (see Figure 31).
The newly created ideas were evaluated with children in the library. Something that became apparent is that most participants related to the fish storylines since this was familiar to them. However, making the story only apply to fish, or just one animal group, does not cover the wide scope of collective behaviour. Therefore, instead of creating the design around one animal theme, no animal group is going to be highlighted. Since the goal of the interaction is creating and seeing different compositions and patterns, the approach would be making users imagine themselves what these patterns visualise. This way every user can relate to it in their own way.
The new ideas were evaluated, based on the criteria in Table 1.

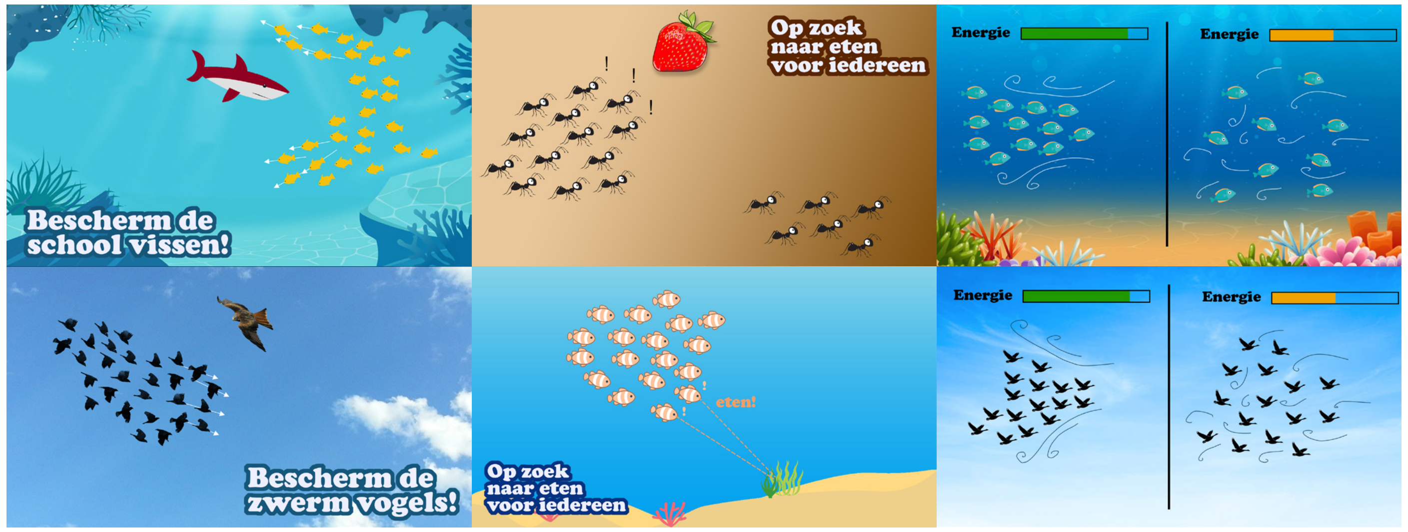


Figure 31: Storylines with the theme survival

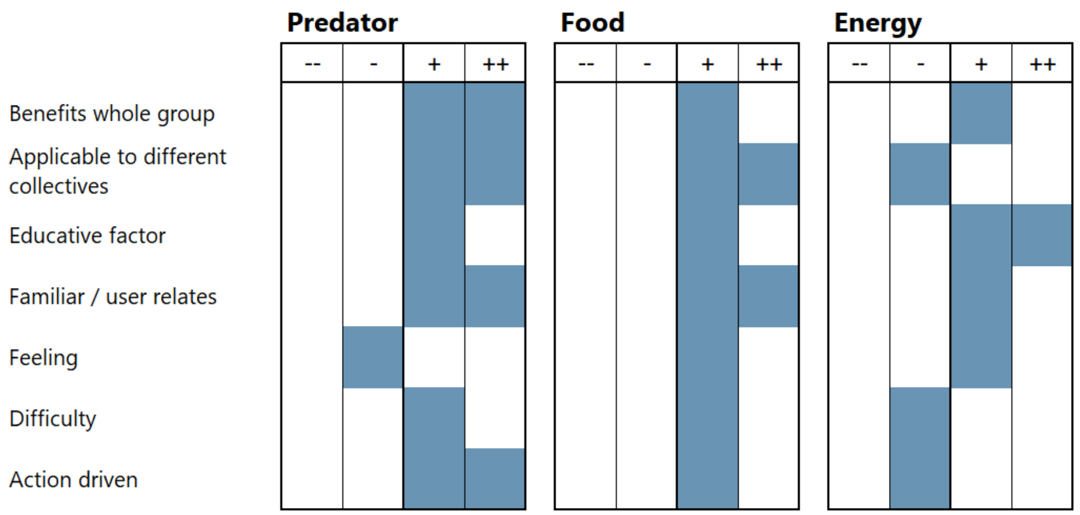


Figure 32: Harris profile for the three storylines

Benefits whole group	Changing the group's composition and movements should benefit the whole group, simulating what happens in reality.
Applicable to different collectives	A storyline that can be imaginable for various collective groups.
Educative factor	This refers to the interaction being educative, instead of being just pleasing to look at.
Familiar	This refers to users understanding the goal quickly, without it requiring an in depth explanation.
Feeling	The user should feel positive while playing, and not feel frustrated.
Difficulty	This refers to how difficult it is to reach the goal. If it is too easy, users will get bored quickly without getting insights in how collective behaviour works. If it is too difficult, users could get discouraged and stop playing too early.
Action-driven	This refers to the interaction being action-driven, so users stay engaged.

Table 1: Design criteria for the storyline

A Harris profile, ranking the three storyline themes, was used to make a decision of which one to continue with (seen in Figure 32).

The predator storyline scores highest among the three. An important factor is that staying in a group benefits everyone greatly. The composition of the group can lead to easily being eaten or surviving. Staying clear of predators was ranked more important than finding food quickly. Something else that is important, is that including a predator is applicable in various collective groups: a shark for fish, a hawk for starlings or a cheetah hunting wildebeests. However, the energy efficiency that birds and fish create by moving in a group is not applicable for wildebeests. Staying clear of a predator was also a concept that felt familiar and the goal was easy to understand, which was more difficult to grasp for the energy storyline.

This predator storyline gives reason to the user to find a suiting group composition that survives the longest against the predator. The focus stays on understanding the underlying rules of collective behaviour.

Takeaways

- Storyline theme: Try out different patterns and group compositions to find out how to survive the longest against a predator

5.3 EVALUATION OF CHOSEN DIRECTION

Evaluation with added predator

A quick evaluation was carried out, where participants could experience the moving simulation with the added predator, to see how they would interact with it.
The algorithm written in p5.js was used to create this simulation, which includes a predator where agents move away from, shown in Figure 33 (de Weerd, 2016). By adjusting the steering behaviour variables the behaviour of the group can be changed.
The main takeaway that came out of this session is that the participants expected the agents to be removed when touched by the predator, instead of them just merely moving away. They said adding this feature would add to feeling more engaged in saving the group.

How to end

A consideration should be made if a clear ending is necessary or if open-ended play is better suitable. A clear ending can take shape in the interaction as a timer that runs out, or when agents get below a certain amount. However, it has been found that this does not lead to the desired positive emotions for the users. The intended interaction is having fun while learning, and not ending up with negative emotions when you get a bad score or the timer runs out before you were finished exploring. Users should have the opportunity to take their own time in this immersive phase and should not feel rushed. Therefore, an open-ended interaction style will be used.

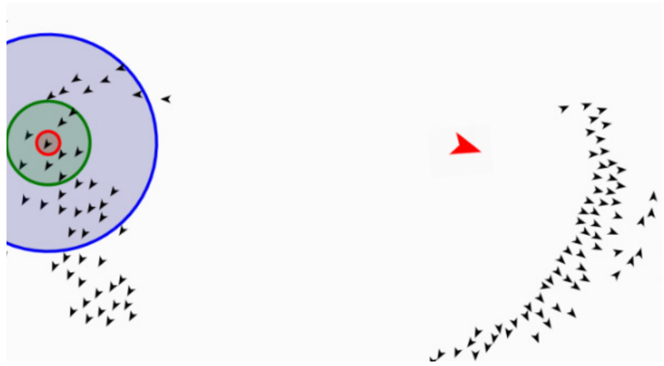


Figure 33: Simulation of collective behaviour (200 agents) including a predator

Evaluation with children

The previously used algorithm of the simulation was adjusted so when a predator touched an agent, the agent disappears from the simulation.
A new evaluation was carried out, this time at an primary school in a class with children aged between 9 and 10 years old, seen in Figure 34.
The evaluation goal was to find out how the flow goes from one user to the other.
It was found that the flow from one user to the next varied. Some gave up their spot for a new user when the group of agents had become relatively small, while others were still exploring the different variables and requested for more agents to be added to the mix again. The average time spent per user was between 2 and 4 minutes. However, this time was influenced by the context where classmates who where watching, were also interacting by thinking out loud and requesting the user to try different things. This is different compared to a museum context, where visitors do not know each other.
Since the predator now removes agents, it became apparent that a reset button should be implemented so every new user can restart the installation. The standard composition that the first participants started with consisted of 120 agents and 1 predator. But, after seeing others play with the installation first, new participants started requesting to begin their turn with a different composition. They wanted to see what happened with a different combination of agents and predators. Creating different compositions to start with, adds to users making the storyline their own, and having something to recall afterwards. This also gives the option for applying multiple difficulty levels, giving users a challenge when needed.

Takeaways

- When a predator collides with an agent, the agent should be removed
- Use open-ended play
- Implement a start button where users can choose between different starting compositions of the simulation

Figure 34: Evaluation with predator at a primary school

5.4 CONCLUSION ON THE STORYLINE

Concluding, the theme of predator avoidance is going to be used, in an open-ended play interaction. Since these predators will decrease the numbers of the agents, users should be able to start the interaction themselves, where they can choose between different compositions. These different options will be categorized on difficulty. This way, everyone can find something that fits their ability.

Creating patterns (easy)

0 predators, 150 agents
Here users can clearly see the patterns emerging, without a predator disrupting these. It can give more clarity to the users if they first want to see what the different steering behaviours do.

Protect the group (standard)

1 predator, 300 agents
Here it is about finding which composition and patterns works best for predator avoidance.

Protect the group (hard)

2 predator, 500 agents, faster predator
Here a challenge is added by having two predators which are faster.

6. DESIGN OF THE INSTALLATION EMBODIMENT

What users will see and do during the interaction experience is discussed in this chapter: how will users get feedback on their input and their influence on collective behaviour. The look of the installation is decided, implementing how the user is going to interact with the design.
The chapter concludes with the composition of all elements of the installation and the corresponding interaction scenario.

6.1 WHAT DO THE USERS SEE?

While interacting with the installation, feedback should be given to the user for them to know what is happening. This feedback should include:

- Seeing the change in the resulting patterns and behaviour of the group
- Indication which parameter they are changing
- The change in value compared to what it was before

6.1.1 Visualisation of the moving simulation

In the simulation which shows the collective behaviour, the following points are implemented.
The agents with their direction of movement are shown by visualising them as triangles, to show their (non)alignment with others (see Figure 35). With the use of this abstract triangular shape, users can envision themselves what they see in the patterns. To be able to focus on the moving compositions and patterns, a non distracting solid coloured background is chosen, with a contrasting colour for the agents, to be able to easily see them. The predator is displayed in another contrasting colour.

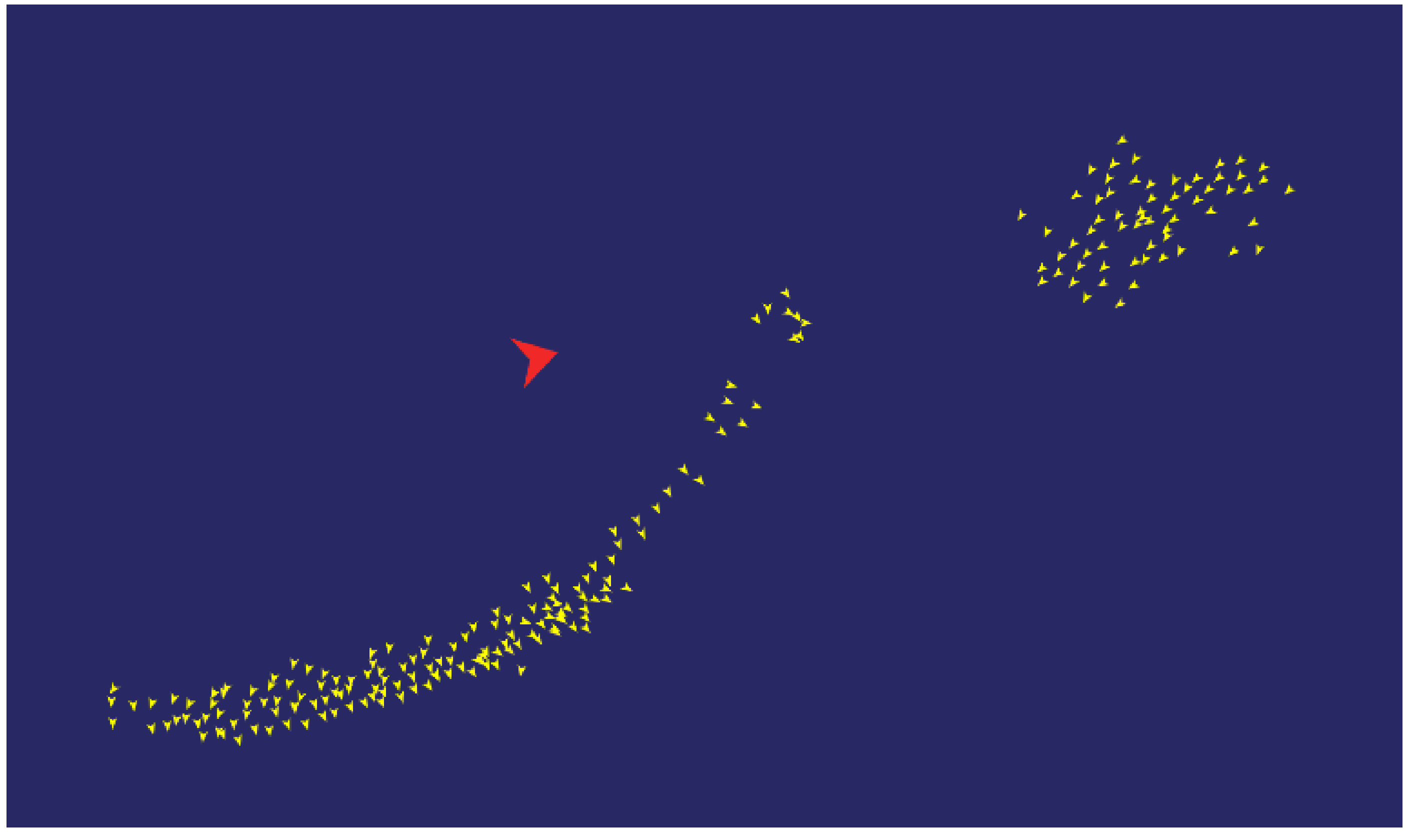


Figure 35: Visualisation of the moving agents and predator

6.1.2 Visualisation of the zone of repulsion and attraction

For the zone of repulsion and the zone of attraction the user will be able to manipulate its size. These zones can both be modelled as circles around every individual, with a radius that the user can increase or decrease. It would make sense to visualise the current size of these zones as circles, so users can compare the values while adjusting. The design of this circle should be considered, with the intention that users link them to repulsion and attraction.

Fifteen participants were asked which kind of visualisation they would choose for the repulsion and attraction zone (see Appendix 2). Inspiration they could choose from is shown in Figure 36.
It was found that the continuous circle (r1 and r6) was most often chosen to visualise the repulsion zone. However, the chosen options for the attraction zones varied, indicating that there is no significant association between a certain visualisation and attraction behaviour.

Since the repulsion and attraction circle will both be shown and changed in size at the same time, they must look different to each other for the user to easily distinguish between them. A way to easily show this difference is using a different colour. However, to take into account colour blindness it is decided to also visualise them in two different styles. What was pointed out by the participants is that since other agents should not cross the border of the repulsion zone, it can be visualised as a continuous line, acting like a 'shield'. On the contrary, agents can enter the attraction zone of others, which will therefore be visualised with a dashed line. This dashed line indicates the size of the zone where the agent looks for others to move towards to. Two colours that add to this concept are red and green. Red belongs to the repulsion, where no one can enter, and green to the attraction zone, where the agent is looking for others.

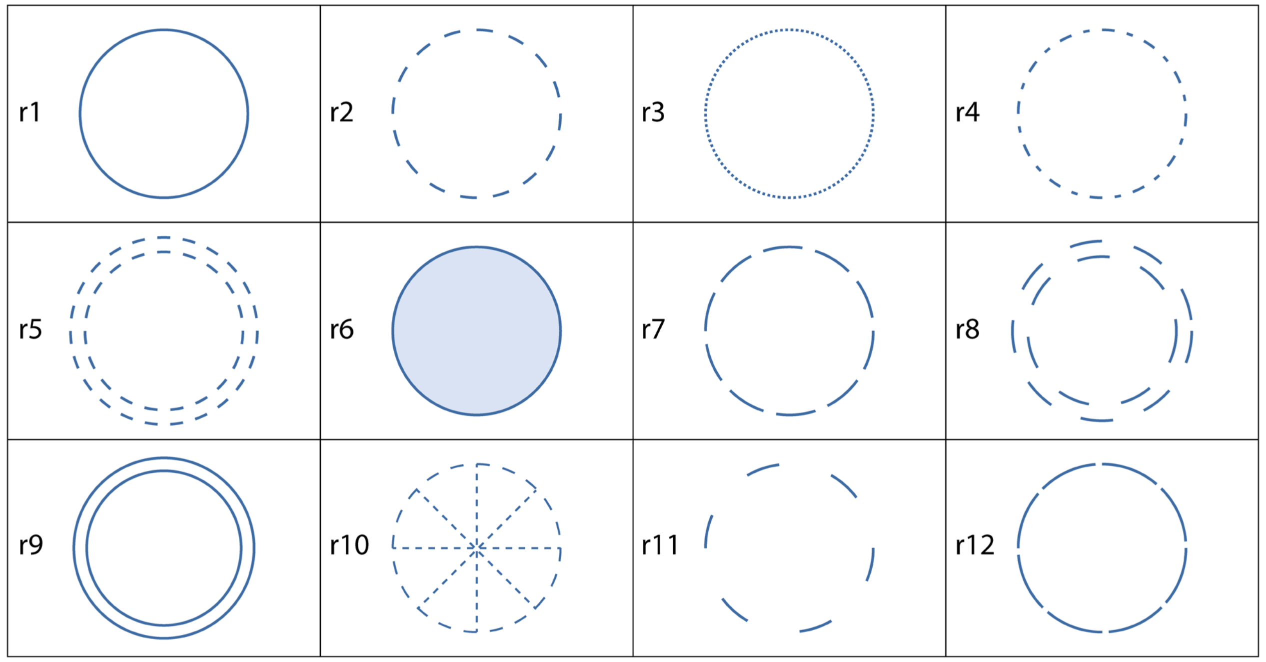


Figure 36: Inspiration for visualisation style

Repulsion and orientation circles in the moving simulation

This circles are going to be implemented in the moving group, to show users in real-time the adjustments they are making and seeing the change happening in the group. Since these zones apply to every single agent in the group, a decision should be made on how many agents will show these zones. This was evaluated with users.

When showing the repulsion and attraction zone for only one individual in the group (Figure 37), some users had the initial impression that only the behaviour of that certain individual was being changed by them. However, the design changes the difference in local response for everyone in the group. Furthermore, whenever the highlighted agent was in the middle of the group, users thought that only the highlighted agent attracted others towards itself, instead of realizing that every agent (also the ones on the edge) has their own attraction zone, making them all move towards each other.

Hence, the simulation was altered to show the zones for all agents (Figure 38). However, this looks too cluttered and overwhelming. The moving agents in the simulation are difficult to see, and the zones take too much attention away from the patterns and behaviour of the agents themselves.

Consequently, the zones were made visible for 5-10% of the group (Figure 39). This way, for multiple agents scattered in a group the zones are shown. The concept that all agents have their own local attraction zone and move towards others was better comprehended this way. However, to some, it still gave an overwhelming impression.

Not being satisfied with any of the three options, a different approach was taken to make clear that these zones apply to everyone in the group.

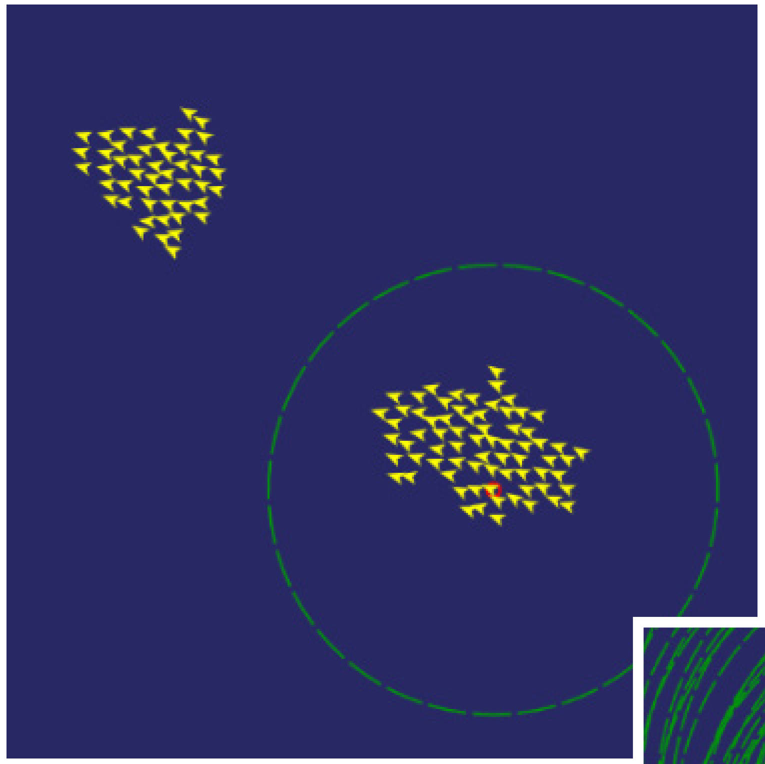


Figure 37: Zones visualised around one agent

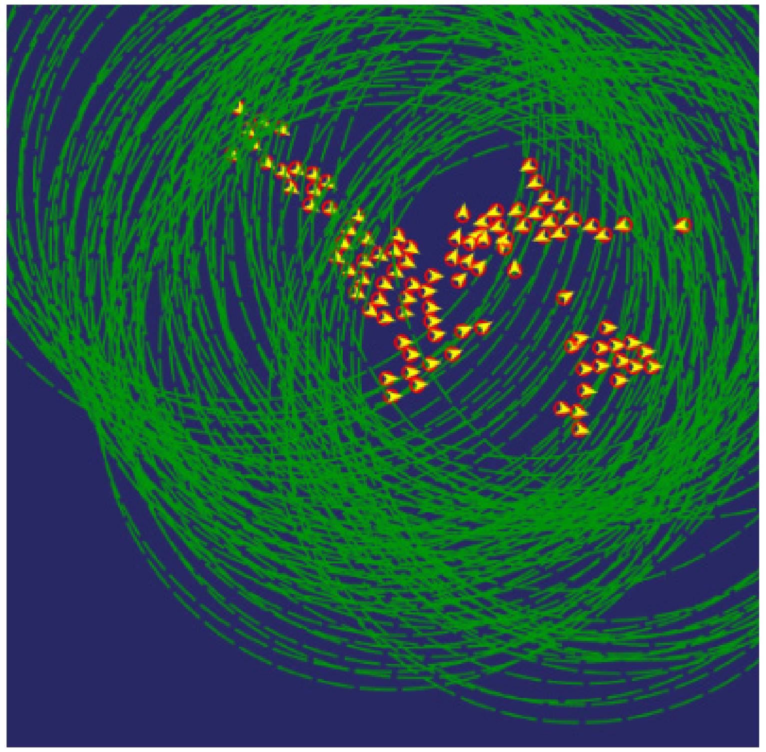


Figure 38: Zones visualised for all agents

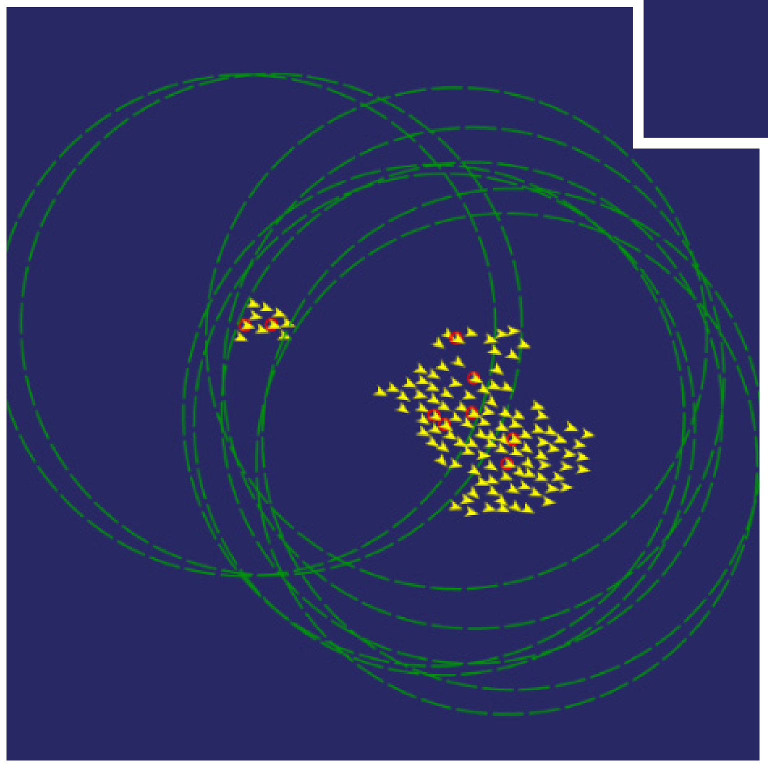


Figure 39: Zones visualised for 8% of the agents

6.1.3 Splitting the two visualisations: local surroundings and moving simulation

In this new concept, both the local surroundings and the moving group are set in the spotlight. Two separate visualisations will be shown to the user: one shows the local surroundings of an agent in a static way. The second visualisation shows the moving group, without any agent highlighted. This way, users have a clear insight into the local surroundings of each agent and can also see the corresponding moving simulation.

Local surroundings of individual
The static visualisation will show the size of the repulsion and attraction zone around the centred agent and its corresponding composition within the group (see Figure 40). For example, when the repulsion zone is increased, the static visual will show agents that are further distanced from each other. The zone of attraction shows the area of other agents that are taken into account to steer towards. When this is increased, they will be able to see more agents.

Moving group showing collective behaviour
The moving simulation will display the group moving in a collective, without the distraction of the zones being shown (see Figure 41). This group is influenced and adaptable by the characteristics chosen by the user. None of the agents will have a special look to them, making clear to the user that the characteristics of the static visualisation apply to all agents.
Something that has been added to this moving simulation, is showing how information is spread through the group. Agents that have spotted the predator, and therefore have more knowledge, are displayed in an orange colour. It can be seen that other agents without this information, already move away from the predator as well (Figure 42).

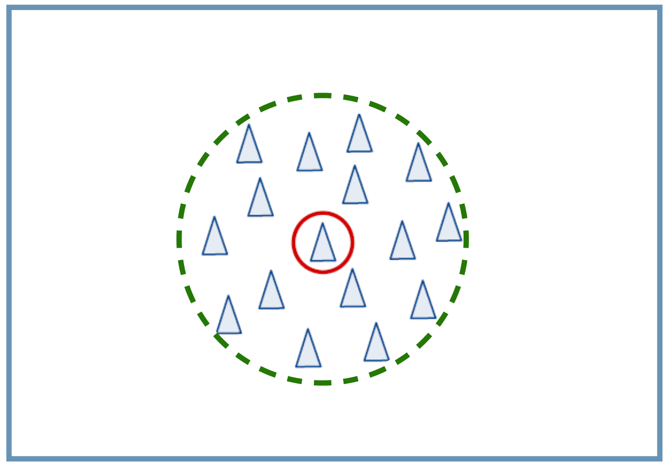


Figure 40: Static visualisation showing the local surroundings

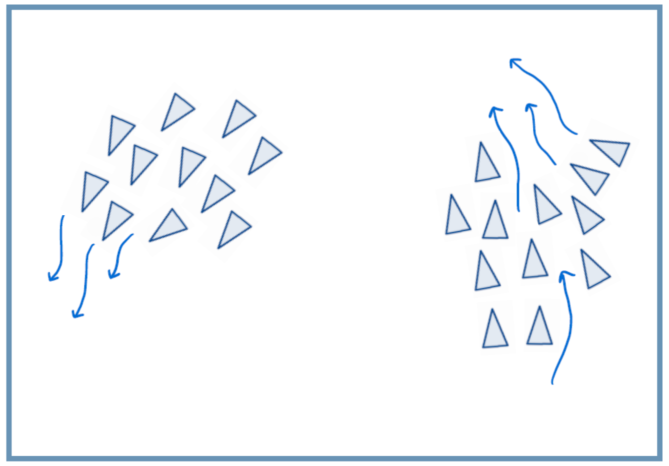


Figure 41: Moving simulation of the collective behaviour

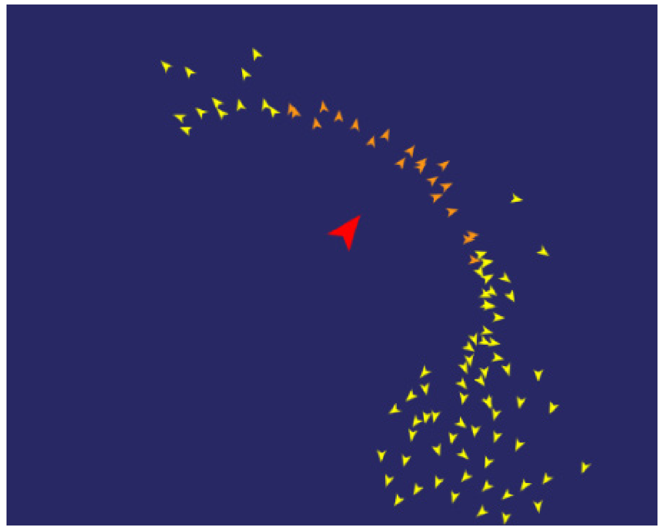


Figure 42: Visualisation of how information is spread through the group

6.1.4 Visualisation of the orientation

Users are able to switch between either a large or small size of the orientation zone. What this difference does is switching between alignment among the members in the group to (little to) no alignment between agents. The intuitive way of showing this in the static visualisation, is showing the agents all pointing in one direction (high alignment) versus showing them not pointing in the same direction (low alignment), as seen in Figure 43.

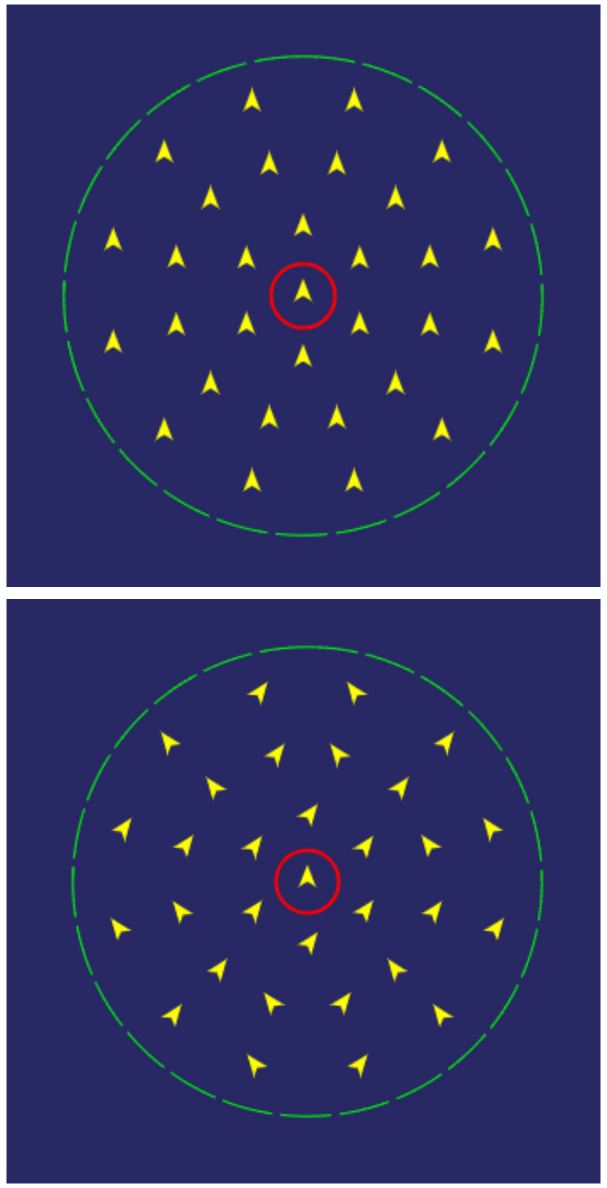


Figure 43: Static visualisation focusing on the local surroundings, showing a high value (top) and low value (bottom) of the size of the orientation zone

6.2 THE LOOK OF THE INSTALLATION AT A GLANCE

Learning styles design criteria
Both the moving and static visualisation should be visible for users and fit in a museum context. One of the main objectives of the museum installation is to teach museum visitors something new. However, not everyone learns in the same way. The different learning styles defined by Kolb (see Section 3.3.3 How to design a learning environment for all museum visitors?), come into play here (see Figure 44).

The diverger prefers to watch and gather information beforehand. The installation should therefore take shape in a way museum visitors can watch others, before interacting with it themselves. The assimilator acts on reasoning and a logical approach and therefore requires a clear explanation. These are people who would like to obtain background information first, before diving into the interaction experience. The convergers, who think and do, use a problem-solving approach. A practical interactive way of learning, opposing to a theoretical passive learning style, is what works best for them. The last learning style is the accommodator. Accommodators are people who will dive straight in and are hands-on doers. They change their plans according to new information they obtain and use a trial and error approach. For them, instant feedback is important so they can adjust their actions.

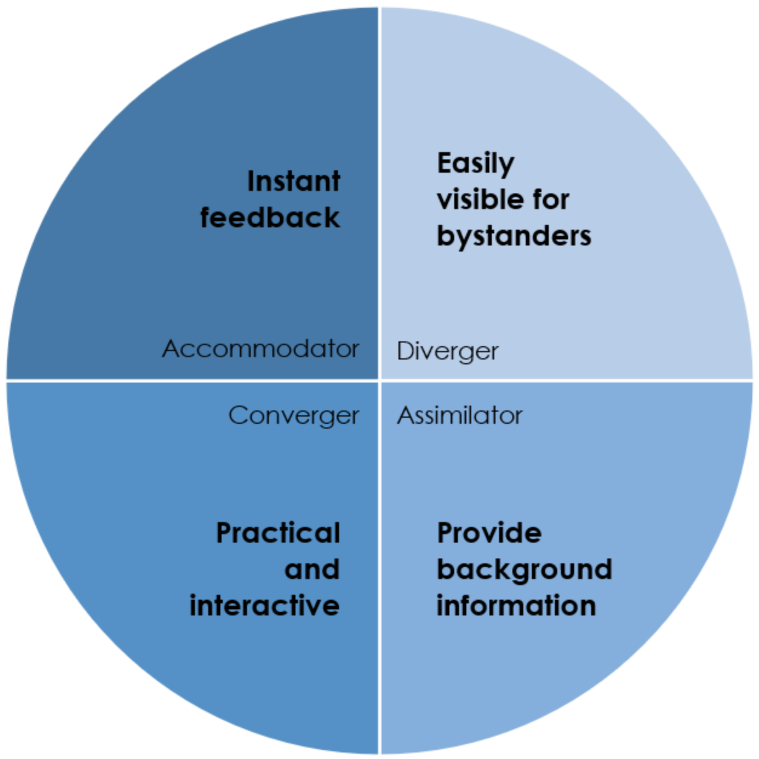


Figure 44: Design criteria based on the four learning styles of Kolb

Shape of the installation
After exploring many different options for the design and shape of the installation through sketching (see Appendix 3), the two options shown in Figure 45 were most fitting to show both the moving simulation and the static local surroundings.

They were chosen on their high visibility for the user and bystanders, and fit in a museum context. Since two different aspects are going to be shown to the user, the two mediums will be combined, shown in Figure 46.
The moving simulation will be displayed on a curved wall, which will give the user an immersive experience, while it is still visible for others to see. Putting the second visualisation on the floor was chosen so the user could stand on top of it, and see the surroundings changing by being in the middle of it themselves.

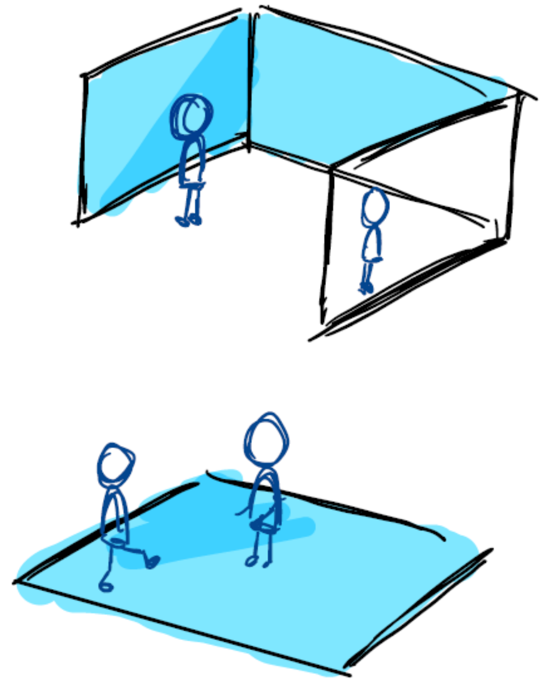


Figure 45: Chosen options from the design and shape exploration: visualise on the wall and visualise on the floor

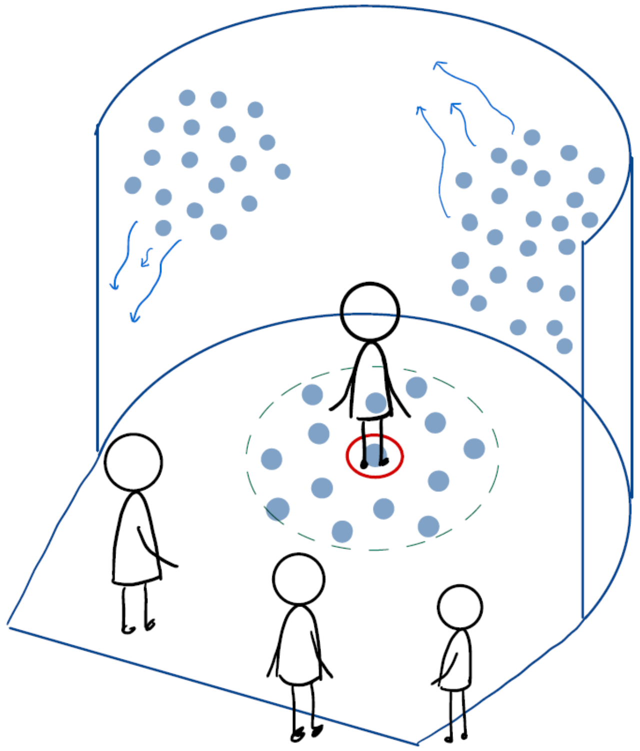


Figure 46: The look of the installation at a glance

6.3 DESIGNING THE USER INTERACTION

An important aspect of the user experience is the way they control the collective behaviour. For the change in repulsion and attraction zone, the adjustment is their size. For the orientation zone, users can choose between the two options: a small size or a large size.

6.3.1 Using rings to adjust the repulsion and attraction zone

After coming up with ideas by using brainstorming, how to's, ideation by sketching and co-creation sessions (see Appendix 4), the following concept was created.
The user stands in the middle of the floor visualisation, in between two rings, shown in Figure 47. By expanding and contracting the rings, the user can control the size of the two zones. By changing the size this way, the movement feels connected to the design. The user feels like they are in control and can see the change happening around their feet and on the screen in front of them.

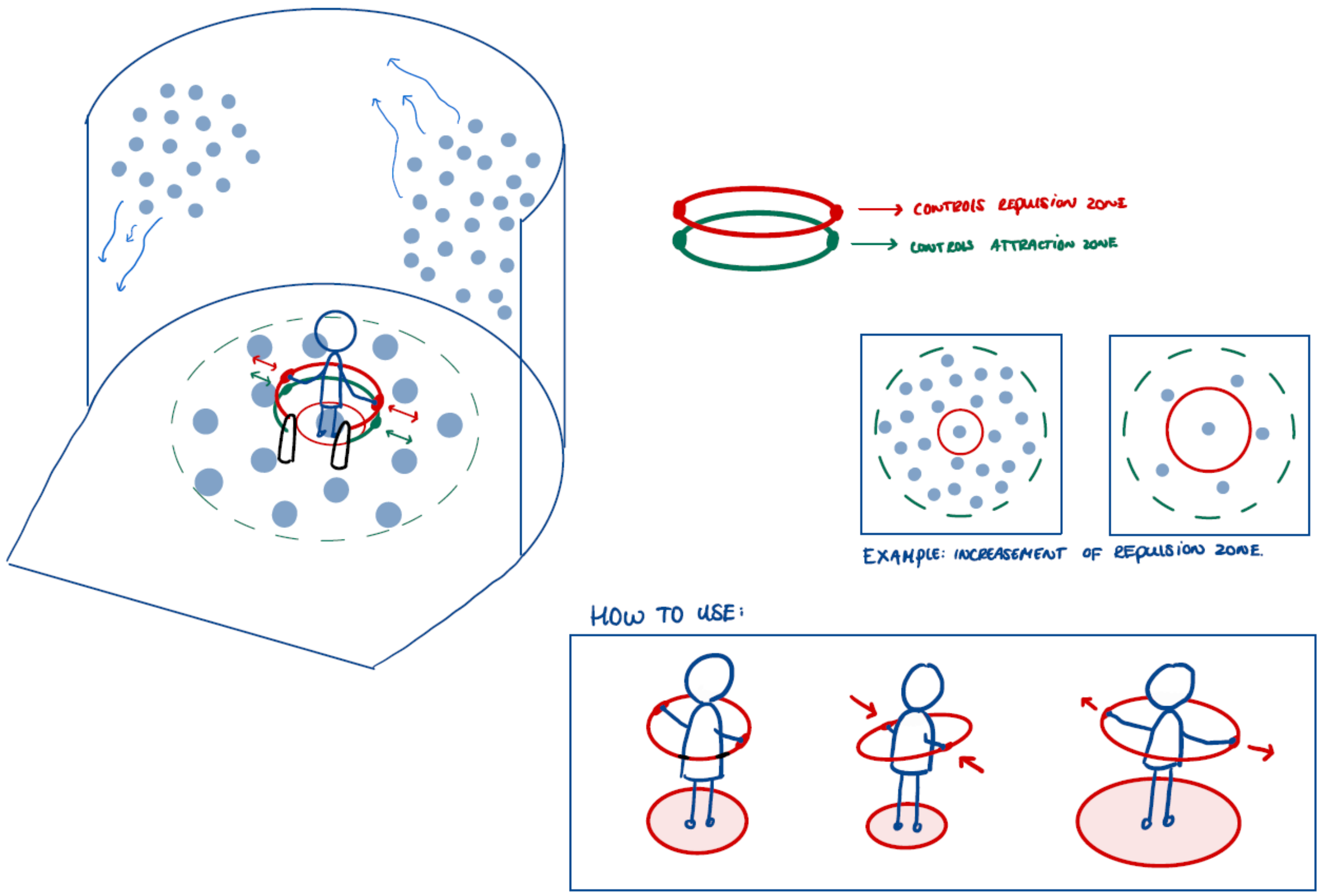


Figure 47: Concept: change the size of the repulsion and attraction zones by expanding and contracting the rings

Operational prototype

An operational model was created to see how this would look in reality. The rings were materialized by two elastic bands, so users could mimic the movement of expanding and contracting the circle. A Wizard of Oz type of test was used for this evaluation. With a projector, the visualisation was shown on the floor.

The first thing that came to attention, was that when the user stood on top of the middle agent, it was difficult to see what was happening underneath (see Figure 48). The projection's size was 1.5 meters by 1.5 meters, since going any bigger would not suffice the museum context. Standing on top of the visualisation was not convenient when trying to see what is going on, especially when the repulsion zone is small and all agents are gathered close together. This was a surprise, and not considered when coming up with this concept idea.

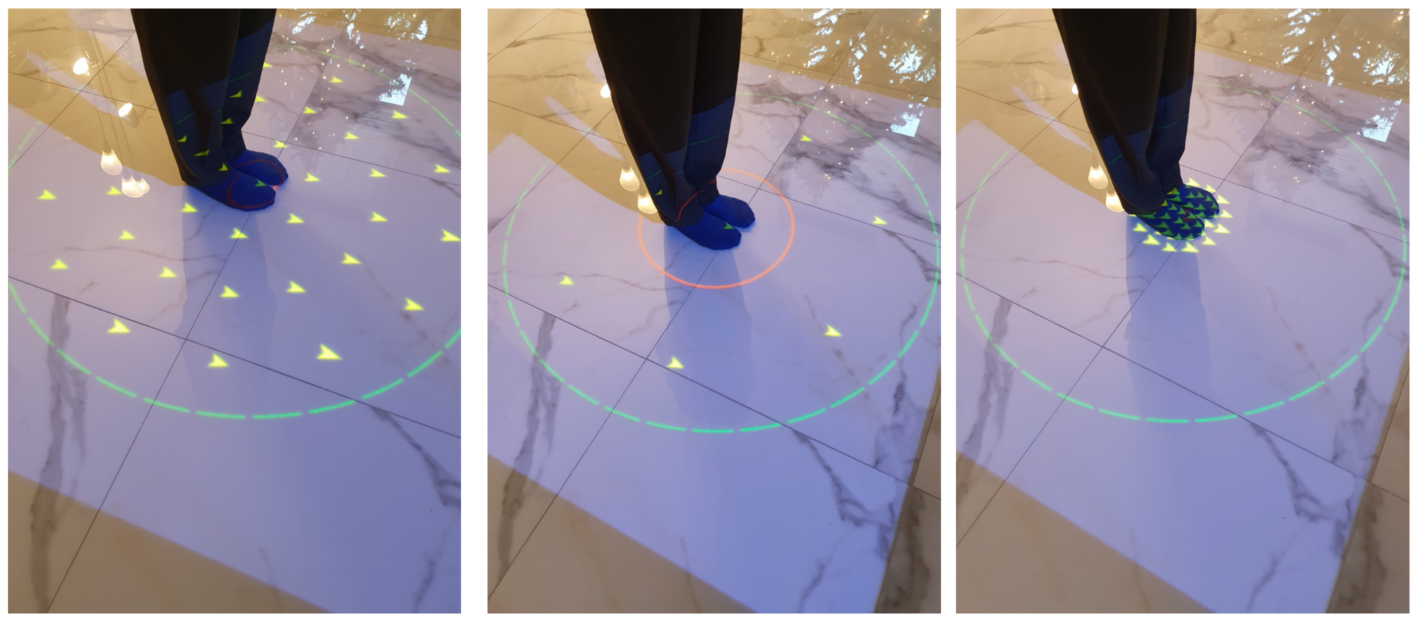


Figure 48: Projection on the floor of 1.5m by 1.5m, with a user standing in the middle. Different sizes of the repulsion zone are shown.

6.3.2 Using linear movement to adjust the repulsion and attraction zone

A different idea was created with the following key aspects in mind: an immersive experience where the way of changing feels connected to the whole experience, while using the body and mind actively. The circles on the floor will be adjusted with an accompanying handle that can be moved linearly. The position of this handle defines the size of the radius of the circle, see figure Figure 49.

Operational prototype

A quick prototype was created with the use of an ultra-sonic distance sensor and an Arduino, shown in Figure 50. The size of the circle corresponded to the plastic handle, which could be moved back and forth. Since the user does not have to stand in the middle of the floor anymore, they can now walk around while still be able to control the installation, which results in the floor being clearly visible to the user.

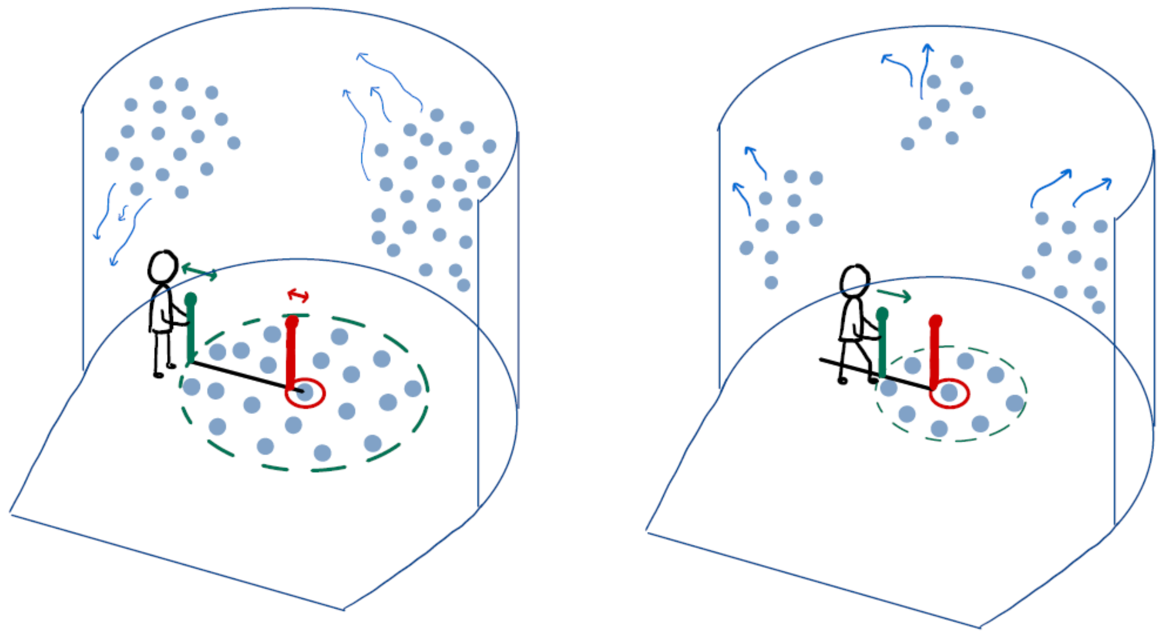


Figure 49: Concept: change the size of the repulsion and attraction zones with the use of corresponding handles

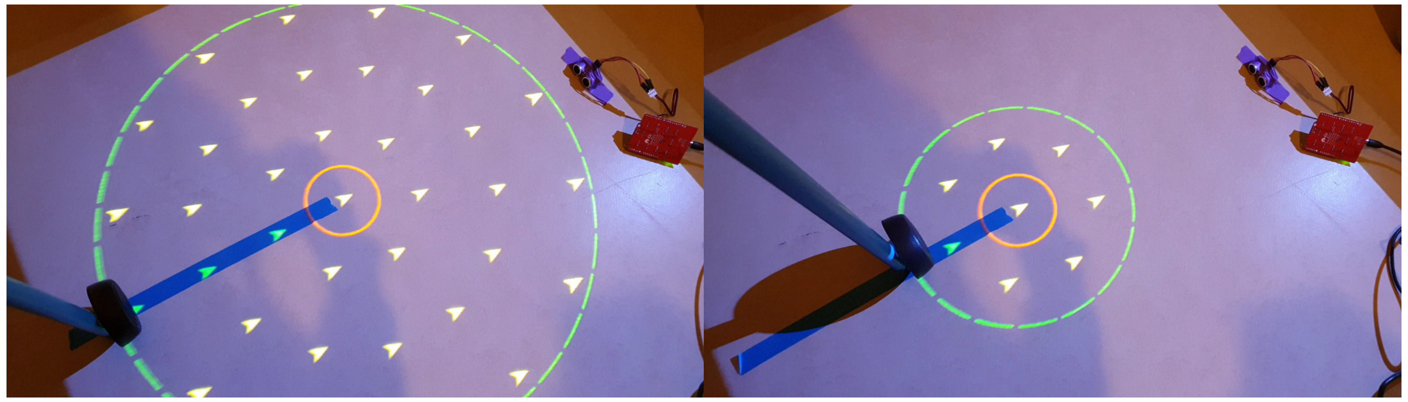


Figure 50: Operational prototype: move handle back and forth and see circles change size accordingly

6.3.3 Adjust the orientation

Looking at the exploration for different options of changing the orientation zone (see Appendix 4), it was looked into what would fit the current design.
Something to point out is that for this orientation parameter, users have the ability to choose between two options. The way it is going to be manipulated should not imply that there is a range in between. Therefore two buttons will be used, where one stands for a small sized orientation zone and the other for a large sized orientation zone. Only one of the two options can be in use at a time: when one of the buttons is pushed, the corresponding button will light up while the other will turn off (see Figure 51).
The placing of this button will be in the same line as the handles for the other two variables, to let the user know these three can be changed and are part of the interaction.
Since the colours red and green are already chosen for the other two zones, a different colour is chosen for the orientation buttons so users can easily refer to them.

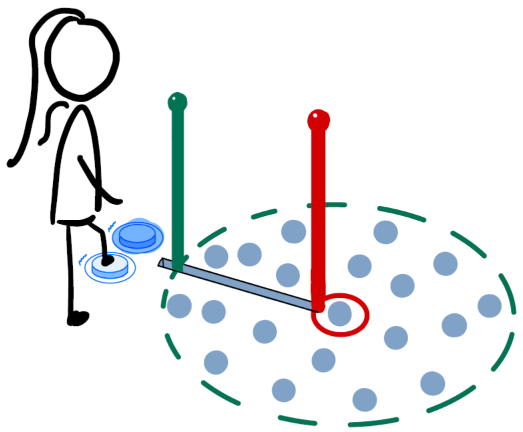


Figure 51: (In orange) The way to manipulate the orientation zone

6.4 WHAT DOES THE USER NEED TO KNOW BEFOREHAND?

Feel invited

Two reasons are present why some background information is needed before museum visitors start using the designs.
First, it is important to invite visitors and make it appealing for them to decide to start interacting with the installation by giving them an indication of what it is about. An important feature of the overall shape of the design is that it is visible, even when you are not interacting with it closely. Visitors can already see the moving simulation, and see others interacting. This will hopefully spark the interest of others, to make them want to try it themselves. This should be evaluated with the final prototype.

Background information

Secondly, some users are not comfortable jumping right into something new (see Figure 44), when they have no idea what it is about and what they can do with it. Providing an explanation on what the design is about, and what you can do as a user, is desired here. Although, this brings the question of what should already be revealed to the museum visitors, and what should be kept hidden for them to explore themselves.
During the many evaluations throughout this design process, it was found that the concept of the different steering behaviours was not understood by everyone without some prior knowledge. Because the users did not know beforehand what the different behaviours were in the three zones, they spent their time figuring out what they were adjusting, instead of learning how these rules together create collective behaviour.
Therefore, letting the user know beforehand that the agents do not want to collide, want to be part of a group and align themselves with others, is important to give users the opportunity to dive deeper and let them explore how a combination of these behaviours influence the different patterns in collective behaviour.

Video explanation

A video was created to explain in a short period of time about the three steering behaviours and how to control the size of the corresponding zones, screenshots are shown in Figure 52. This test was done in conjunction with the prototype of the resizeable rings.
It became apparent when this video was shown at the beginning of the first encounter with the installation, users did not follow the tutorial in sync with the explanation. Some were still figuring out the controls, while the tutorial video already went to the next part, which made them miss the next explanation. As predicted, some participants did not even follow the tutorial and wanted to begin instantly. It can therefore be concluded that the tutorial should not be something the users are mandatory to follow before using the interaction, and should therefore not be part of the user interaction for everyone.

It turned out that people wanted to go through their explore stage at their own pace, with the information available to refer to when needed. They preferred to see all three rules together at the same time.
Therefore a new informational video graphic (Figure 53) was created.

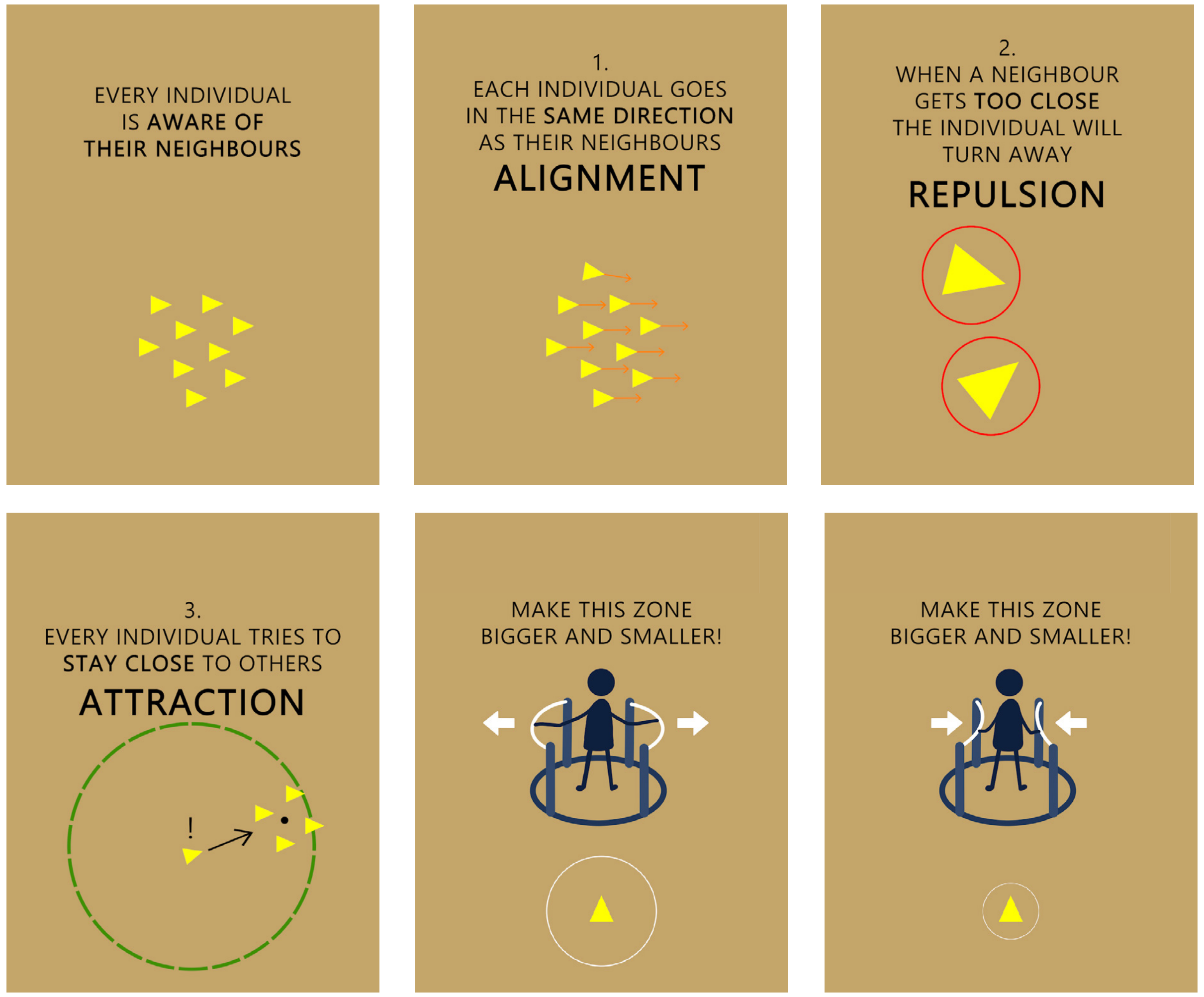


Figure 52: Screenshots from the tutorial video

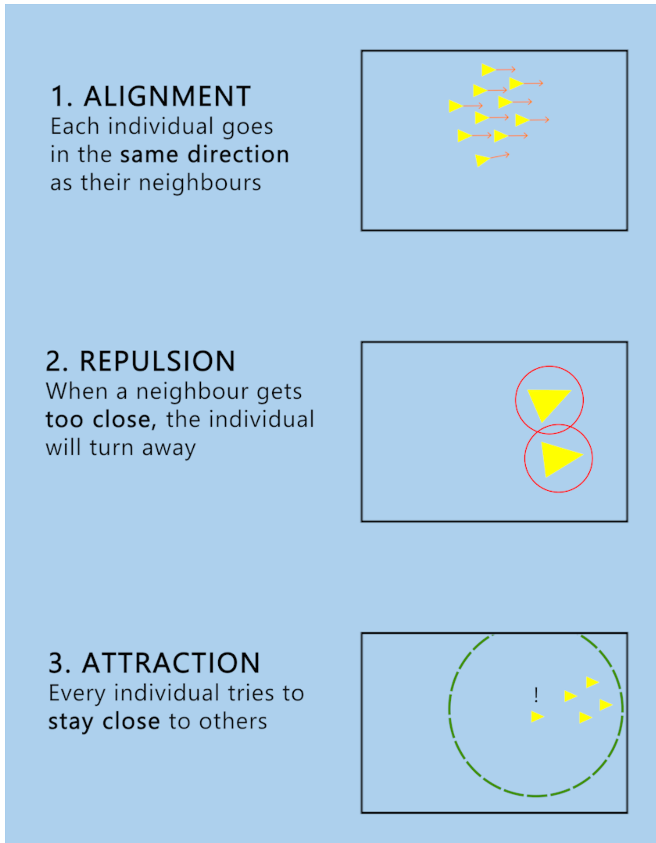
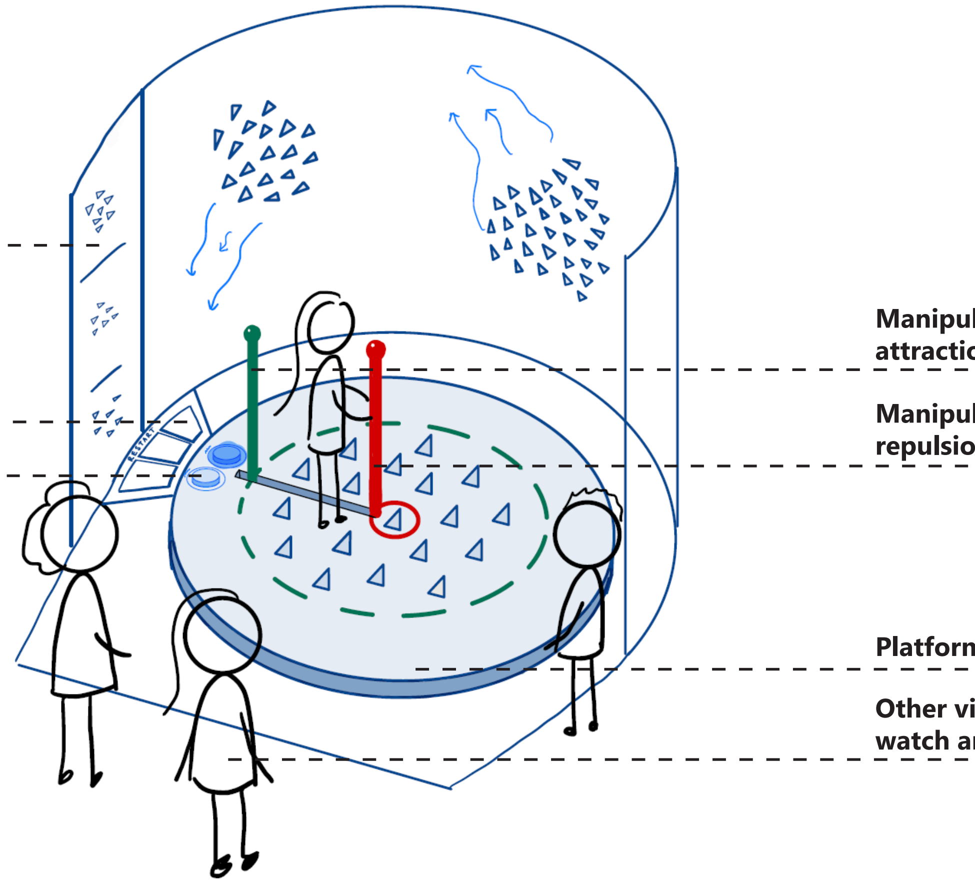


Figure 53: Screenshot from the new tutorial video

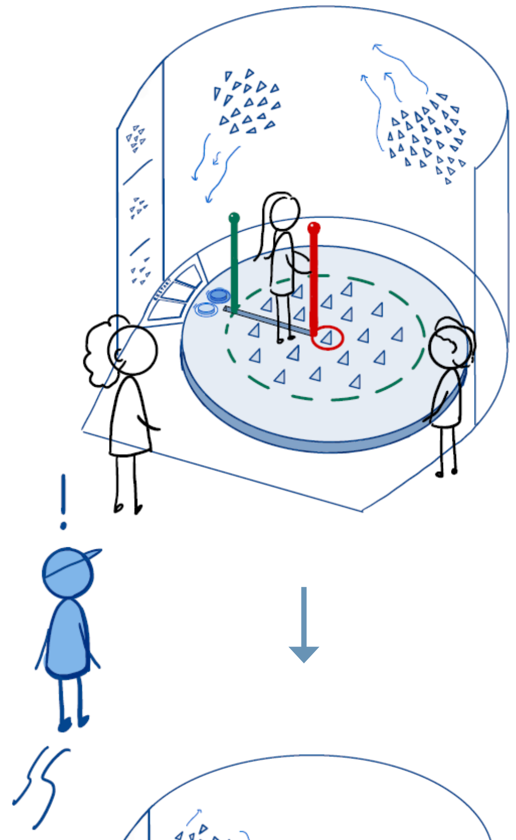
6.5 PLACEMENT OF ALL ELEMENTS IN THE INSTALLATION

Now knowing all elements, the placement of them together in the installation can be decided.
A curved wall will be used to show the moving simulation of the collective behaviour. In front of this wall, the interactive environment takes place on top of a platform. The static visualisation of the local surroundings of individuals will be displayed on the floor.
On this floor visualisation, users are able to change the radius of the repulsion and attraction zone with the use of corresponding handles. The orientation zone is manipulated by two buttons leading to a small or large sized orientation zone.
Museum visitors can watch when others are interacting with the installation. The placement of the restart buttons should be taken into consideration. Bystanders should not be able to easily reach these buttons, disrupting the interaction of the current user. They are placed in a way so only the current user can operate them. Lastly, the background information should be in view of the user to refer to, but also easily visible for bystanders, to be able to already see what it is about.



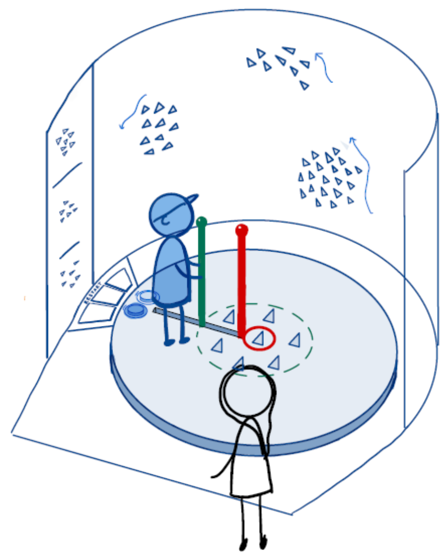
6.6 INTERACTION SCENARIO

The interaction scenario shows the intended use of the museum installation. In this experience, the Body, mind, heart and soul elements are taken into consideration.
To make adjustments to the collective behaviour, users interact actively with the installation by using the handles. The interaction is designed in a playful manner to evoke positive emotions from its users, while their mind is challenged by figuring out the connection of the size of the behavioural zones and the influence on the moving group. Lastly, the learning experience is intended for users to become aware of collective behaviour and realize how this is apparent in animal groups, but also plays a role in many other areas.



Museum visitor sees the design from afar
What is this about? I see multiple visualisations and patterns moving on the screen! It seems like I can play with this installation by moving those handles. Let's check it out!

Visitor waits for their turn
Visitor watches the current player, while looking at what is happening and taking in the whole installation. The visitor will notice the handles, buttons and explanation on the wall.



Visitor interacts with the design
Where do I start? Oh, I see a restart button! Let's start easy. I'm going to move the handles and see what will happen. What will these two buttons do?
The visitor starts to anticipate what will happen and realizes how the different patterns are formed.
Let's try a more difficult level.
User restarts on the standard level
Tries to find a formation where the group survives the longest
Makes room for next person
Remembers and talks about the installation

7. FINAL DESIGN

In this chapter, the final decisions of the installation are decided and the final design is presented.
A prototype is built to validate part of the user experience and the levels of understanding of the users. The final design is accessed on its desirability, feasibility and viability. Lastly recommendations for future research will be provided.

7.1 HOW THE INSTALLATION IS GOING TO BE MADE

This section explains how the installation is going to be created in terms of the visualisations, mechanics and electronics. The programming behind the installation is discussed and the way of creating the mechanism is determined.

7.1.1 Visualisation

Agents in the moving simulation

The simulation of the algorithm is going to be run in p5.js, which uses JavaScript as coding language. The code is based on the algorithm created by De Weerd (2016). Features have been adjusted and added to this algorithm. For an overview, see Figure 58. Values from the sensors can be sent as input to the algorithm, thus making the computer do the computing power, instead of the microcontroller used for the sensors.
The algorithm works by implementing the avoidance, alignment and attraction steering behaviours. The size of the surrounding area that each agent takes into account, depends on the input from the sensors from the interactive environment.

For every agent, the algorithm does the following:

- When it sees other agents within the repulsion range, its preferred direction will be the opposite direction. The closer an agent, the higher the need is to move away from that direction.
- It wants to go in the same average direction as other agents who are present the orientation zone.
- It wants to move to the average position of the other agents in the attraction zone.
- If it notices a predator, it will steer away from it and change in colour

By constantly looping through all agents, finding their position related to others and predators, the preferred direction of movement is found. The algorithm outputs this as a visualisation of moving agents. In Figure 54 to Figure 57 the visualisation of the moving simulation is shown for different values of the zones of repulsion, orientation and attraction.

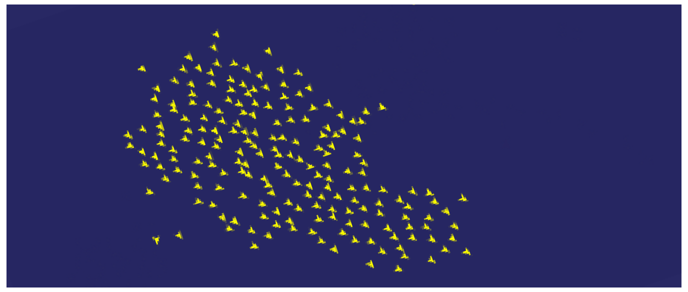


Figure 54: Low repulsion, high orientation, high attraction

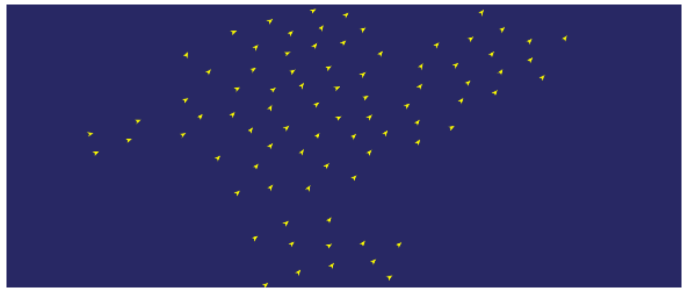


Figure 55: High repulsion, high orientation, high attraction

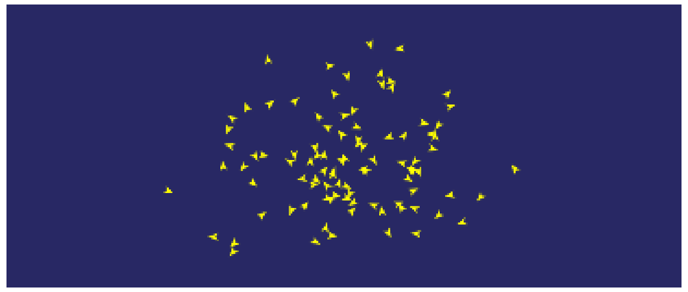


Figure 56: Low repulsion, low orientation, high attraction

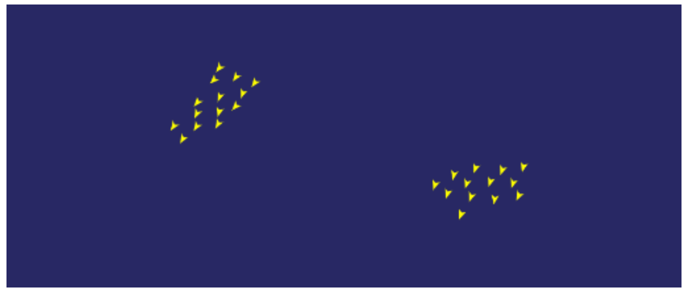


Figure 57: Medium repulsion, high orientation, low attraction

Script - Moving simulation (see Appendix 5)

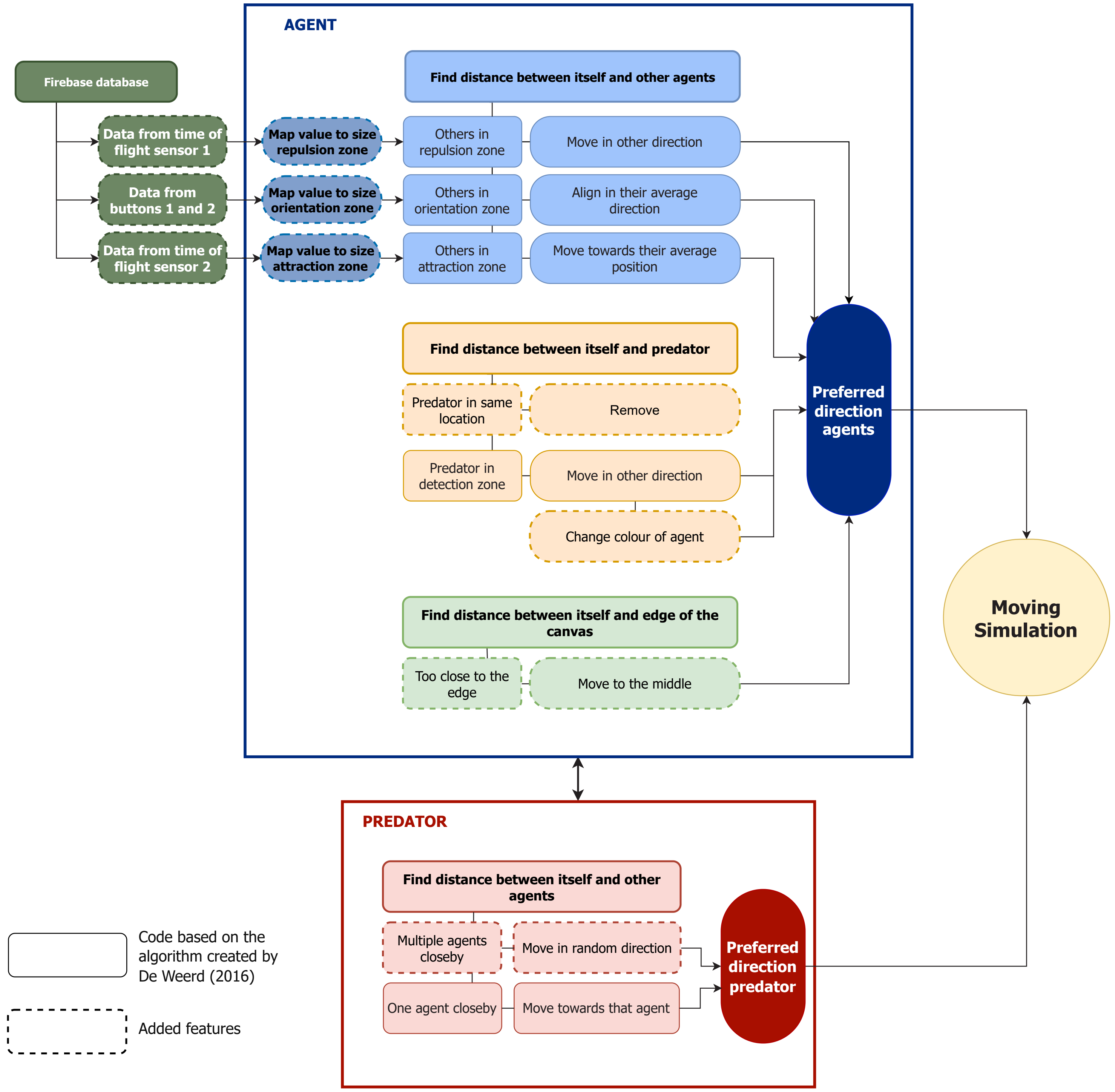


Figure 58: Overview of the script used for the simulation of the moving group. The blocks outlined with a continuous line are code based on the algorithm created by De Weerd (2016). Blocks outlined with a dashed line are added features to the script.

In the algorithm written by De Weerd (2016), when agents moved outside of the frame they would appear back in frame at the opposite side. This, however, lead to non-natural behaviour since agents suddenly appeared at the edges. The group within the frame could not detect and anticipate them beforehand. To account for this, an imaginable border is created around the screen. When agents reach this, their preferred direction is aimed towards the centre of the canvas.

The predator in the moving simulation

The behaviour of the predator is simulated as followed (also shown in Figure 58):
Every predator continuously calculates the distance between itself and all other agents. The predator was coded so that it moved towards the closest agent. However, this resulted in a predator behaviour that did not distinguish between an agent in a group and an agent on its own. A change was made to the algorithm: whenever the predator sees multiple agents in a close range, it will get confused and is not able to pinpoint a single agent.
The speed of the predator is set slightly higher than the agents. Whenever the coordinates of the predator and an agent are the same, the agent is removed from the simulation.

Local surroundings of an individual

To create the visualisation of the surroundings of one individual, the moving algorithm was rewritten so the agents all stayed in one place, depending on the size of the repulsion, orientation and attraction zones. See Figure 63 for an overview. This script uses the same values for the input as the moving simulation, so they correspond to each other.
One agent is always shown in the middle, with the size of the repulsion and attraction zone around it. To show which neighbours the centred agent considers to move towards to, only those within the zone of attraction are visible. Secondly, the way the agents are spaced out depends on the size of the repulsion zone. When this is larger, the agents are more distanced from each other.

At first, this visualisation was coded by projecting circles of agents around the central one. The distance between those circles of agents corresponds to the size of the zone of repulsion. While this seemed visually correct for the first circles around the middle one (see Figure 59A), it turns out that the larger the circle is, the more spaced out the agents become (see Figure 59B).
Since this is not the desired visualisation, a new way of positioning the agents is used. The script translates all agents into a raster, so they are equally distanced from each other. Within the code, the agents are translated into 4 quadrants, to be able to position them along the x- and y-axis. The distance between the individuals is the size of the repulsion zone. When the distance of an agent to the centre is larger than the zone of attraction, it will not be displayed. This resulted in the visualisation shown in Figure 60A.

To show when the orientation zone is small, and the agents do not align with each other, they will be pointing into random directions. However, since the position of the agents is

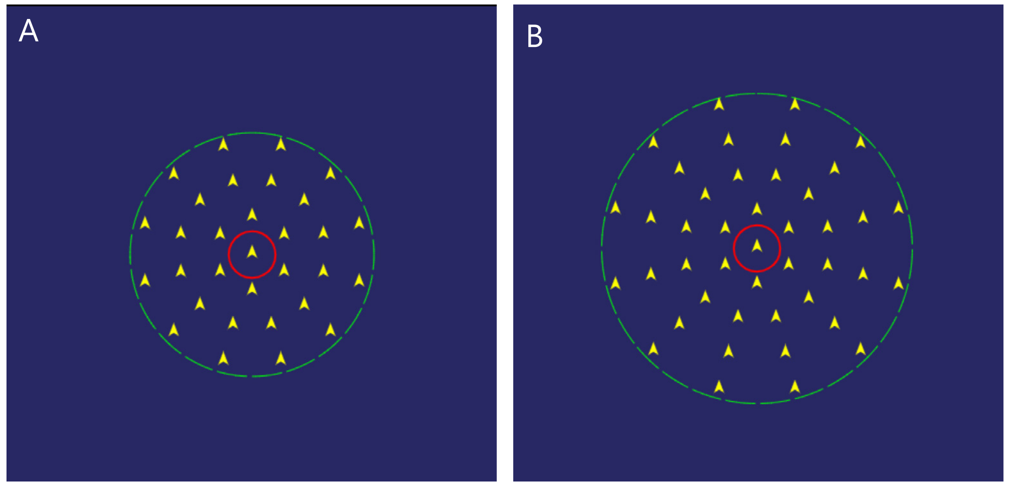


Figure 59: Agents are spaced out in circles around the middle agent

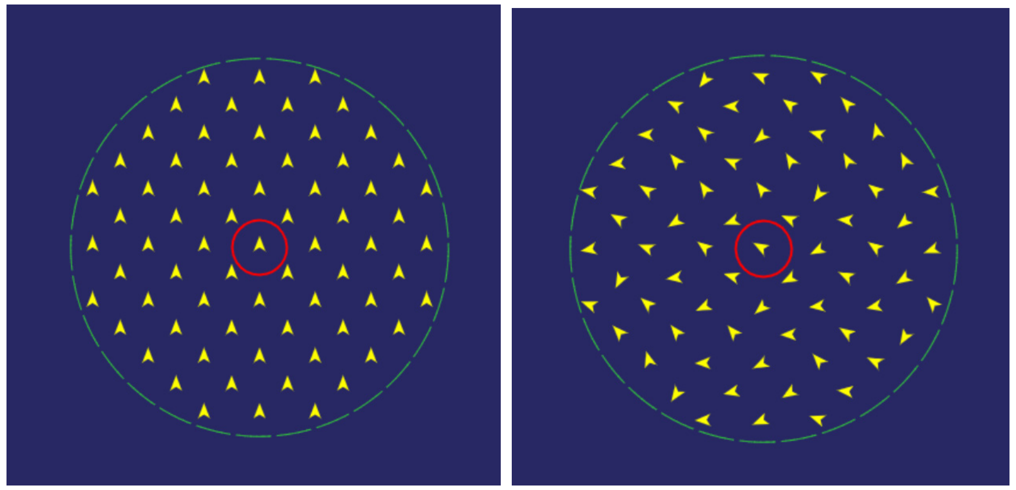


Figure 60: Agents are spaced out in a raster around the middle agent

continuously being updated whenever there is a change in one of the zones, by using a randomized angle of direction for every agent, the visualisation would keep randomizing directions instead of being static. To tackle this problem, a randomSeed is used. This makes the randomization of the directions always the same, creating a static output for the visualisation (see Figure 60B).

The user is able to change between a small and large orientation zone. But, when testing this out with three participants (aged 25, 55 and 57 years old), they did not understand that the alignment steering behaviour is also based on a zone, since this zone was not displayed. To show users all three steering behaviours are applied within the zones, the decision was made to also show the orientation zone. This orientation zone would always be bigger than the repulsion zone and smaller than the attraction zone, shown in Figure 61.

How to display

The moving simulation will be projected on a curved projection screen, to create the immersive experience. The use of a projector is chosen, since it will be cheaper than using a curved display. The local surroundings visualisation will be projected on the floor. Even though this surface is flat, again a display will not be used. Since users will walk on top of it, this can also become expensive. Secondly, if a display is used, users can get the expression that it is there to look at, and not suitable to step on top of. Therefore, a projector will be used, which needs to be hung directly above to minimize the shadows.

Colours and shapes

The colours for the agents and background were tested with a projector on different surfaces. It was found that the colour yellow was highly visible on every surface. To be able to highlight this colour even better, the colour blue is chosen since this colour is the opposite of yellow on the colour wheel. Because the repulsion zone is already displayed in red, the colour for the predator is changed from red to orange. Its shape is also adjusted to look differently from the agents (see Figure 62).

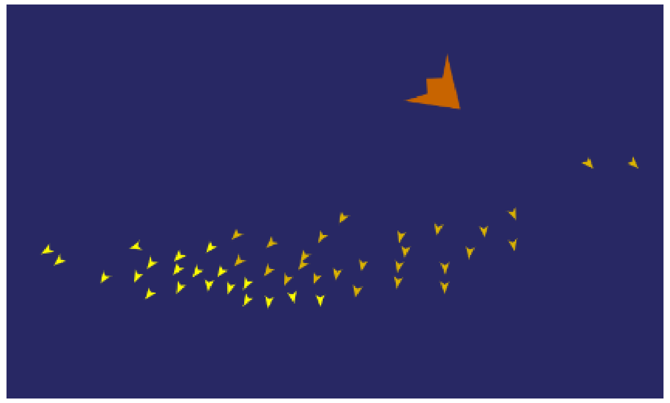


Figure 62: Colours of the agents and predator

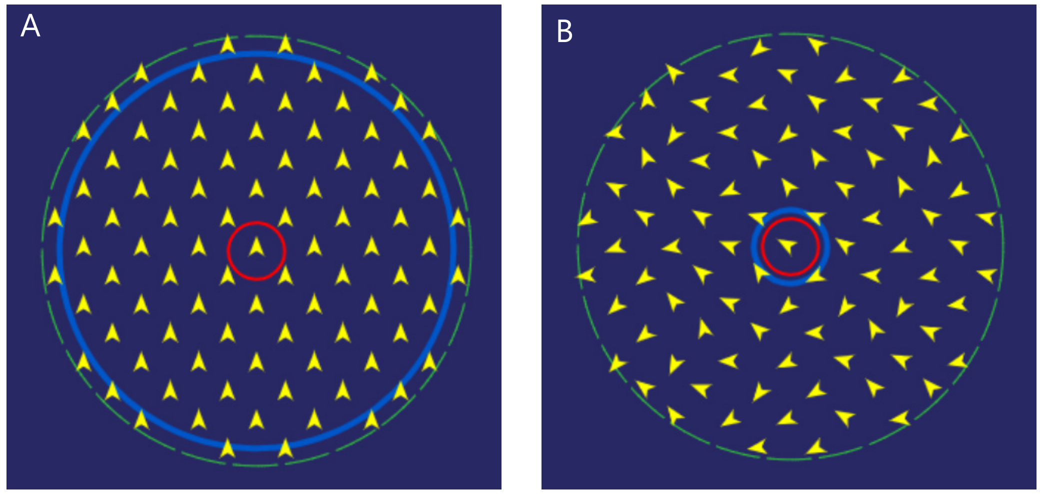


Figure 61: Showing the big and small orientation zone (in blue)

Script - Floor visualisation (see Appendix 6)

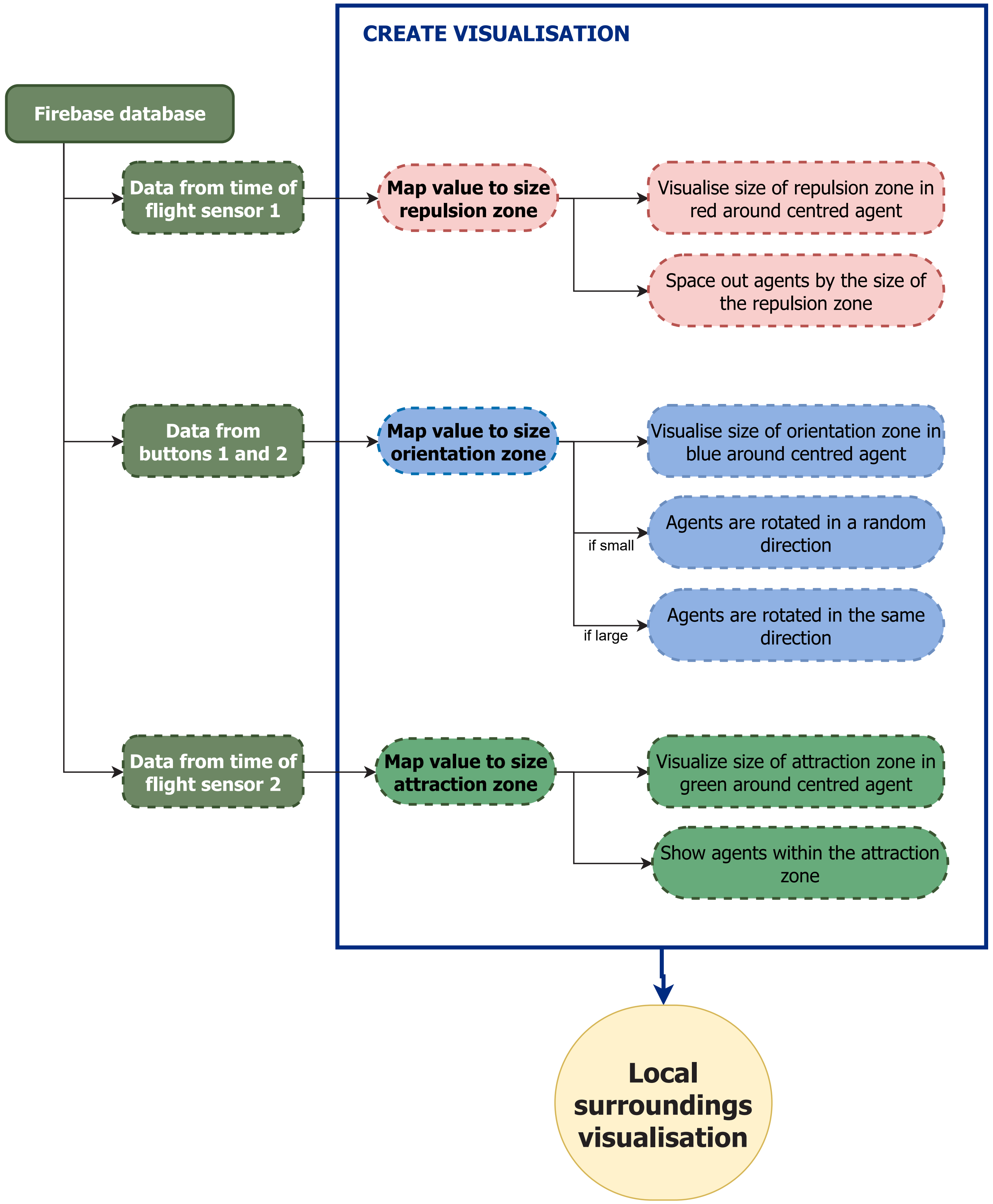


Figure 63: Overview of the script used for the visualisation of the local surroundings around an individual.

7.1.2 Mechanism

An important aspect of the installation is the sliding mechanism. The way this will be implemented is discussed in this section.

Dimensions

The installation includes a linear movement along the radius of the circles on the floor. To find out the dimensions of this sliding movement in the interactive space, a rough prototype was created, shown in Figure 64. Keeping in mind a balance between the visibility of the circles, still visible when they are small, while also having a good overview of the whole picture when large, the sliding distance will be 60 centimetres. The maximum size of the outer circle will therefore be 120 centimeters.

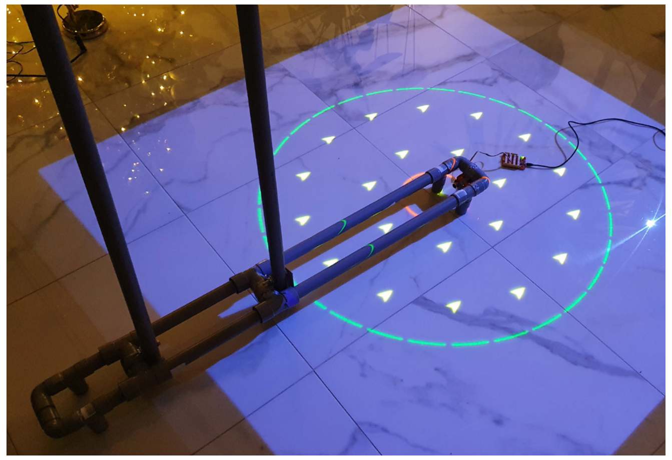


Figure 64: Finding the right dimensions with a prototype made out of PVC tubes and two T-pieces that could slide back and forth.

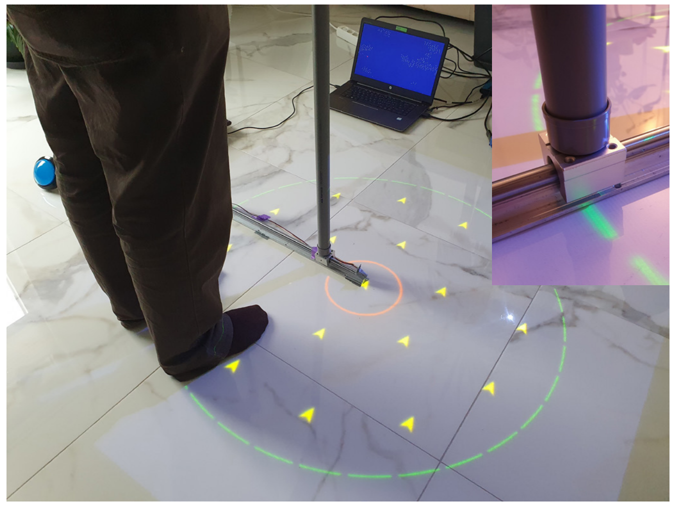


Figure 65: Linear slider with bearing block and a close-up of the bearing block

Sliding mechanism

To find out if the sliding mechanism could work when manufacturing the installation, a look was taken at a linear rail combined with a bearing block (see Figure 65). It was found that if the handle was hold at a high height while starting the sliding movement, a torque is created at the sliding block, which makes the block tilt and prevent the handle from sliding, see Figure 66.
A way to fix this problem is by implementing a hinge in the connection of the handle with the block, so less friction would occur. However, since the block was placed on a circular rail, it could also hinge to the sides, making the handle fall over.
To get rid of the problem of the handle falling over, two rails were used, shown in Figure 67. However, friction still occurred when holding the handle high and starting the movement. When consulting experts at the Model Making and Machine Lab at the Industrial Design faculty at the TU Delft, it was found out that a linear guide rail with a rectangular shaped rail would work better in the desired design. It was also discovered that the linear bearing rails that were purchased had too much clearance than should be expected from these rails. Without this clearance, or with the rectangular-shaped rail, they expect a smooth linear movement to be possible, when starting the movement.

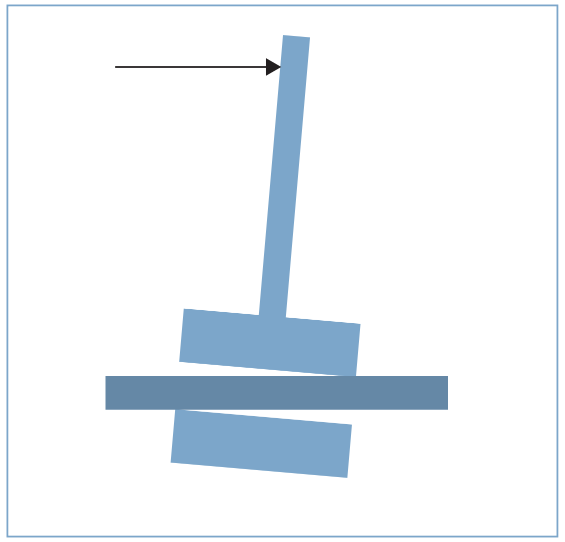


Figure 66: By pushing at a high height, the torque makes the block tilt and create friction

Controlling close to the ground

Moving the handles at a high height while standing will always have the disadvantage of the possibility of creating a torque resulting in friction. Because this movement is one of the main interactions of the installation, a smooth experience is important for the users. Since the context of the installation is a museum, where it will be repeatedly used by visitors, who also include children, the mechanism should be robust and not break down easily.
Therefore, if the movement is controlled closer to the ground, friction will be reduced, making the movement go smoothly and having less chance of breaking apart. To be able to still use the installation while standing, the sliding mechanism is created in a way to be used by feet (shown in Figure 68).

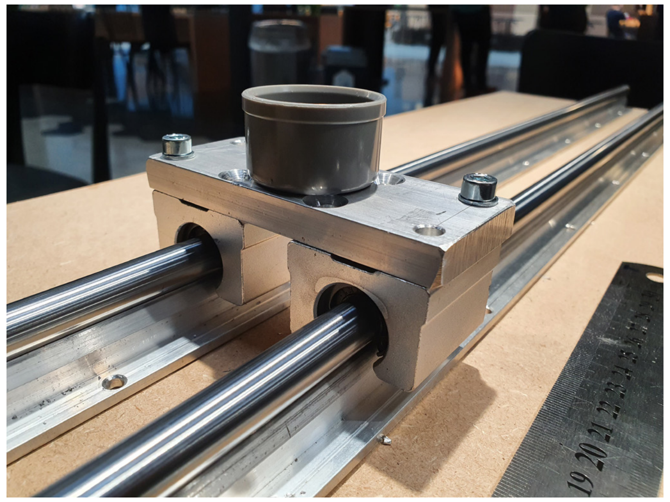


Figure 67: Using two rails and connecting the blocks

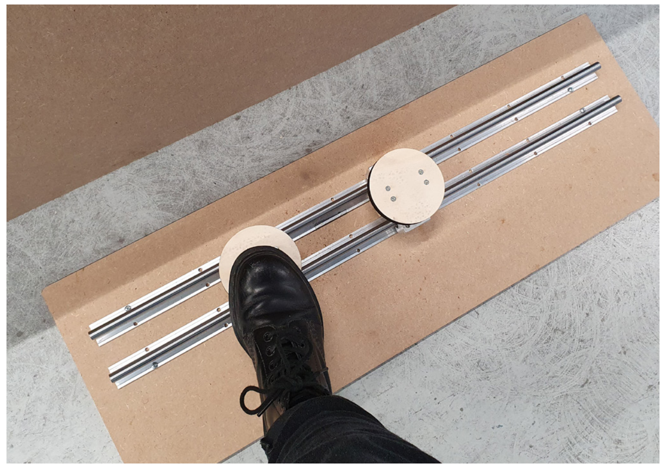


Figure 68: Moving the disks by foot

7.1.3 Electronics

Keeping track of position

Since the position on the sliding mechanism decides the size of the repulsion and attraction zones, this position should be kept track of. This can be done in different ways: with the use of potentiometers, distance sensors, touch sensors or making use of a camera. These all have their pros and cons, listed in Table 2.
By looking at the pros and cons, and asking for advice at the Applied Labs at the faculty of Industrial Design Engineering at the TU Delft, the decision was made to use distance sensors. Two different types of distance sensors were analysed: an ultrasonic distance sensor and a time of flight sensor. Since the ultrasonic sensor makes use of sound, its response rate was slower than the time of flight sensor. Moreover, the small size of the time of flight sensor came with great advantages of implementing it in the design, and picking up less interference from surroundings. The time of flight sensors are therefore chosen.

Buttons

For the restart and orientation buttons, large buttons will be used which are placed on the platform. These can be operated by either users' feet or hands. Lights within the buttons will give feedback to the user. The restart button should light up whenever it is pushed, letting the user know the button is working. The lights within the orientation buttons show which one is currently turned on.

Data transfer

The collected data from these time of flight sensors and buttons should be communicated to both the moving and floor visualisation scripts. This is done wirelessly, so no extra cables will disrupt the interactive environment and the computer does not have to be close to the interactive space. The data from the sensors is uploaded to an online database, while the visualisation algorithms pull the data from this database. The database that is used is Firebase, which is developed by Google and can store values which are synced in real-time.

Type of sensor	Pros	Cons
Slide potentiometer Keeps track of the current position along a linear line. A potentiometer works through a variable resistor which determines the output voltage.	Accurate to sense position.	Since the linear movement in the installation is on a large scale, either a large linear potentiometer is needed, or a construction should be created to translate the large distance to a small potentiometer.
Distance sensor Keeps track of the distance	Accurate to sense position. Only one needed.	Be careful that users do not interfere with the measurement of the distance. Too much noise can give inaccurate data.
Ultrasonic sensor This sensor measures distance by emitting and receiving ultrasonic sound waves.	Not affected by colours and transparency of detected object. Cheaper than a time of flight sensor.	Lower response rate compared to a time of flight sensor. Wide field of view.
Time of flight sensor This sensor measures distance by pulsing and receiving laser light.	Small in size. Fast response rate and a narrow field of view.	More expensive than an ultrasonic sensor. Affected by transparent objects.
Touch sensors When a sensor is touched, the position is recorded	Less sensitive to noise in data	Multiple touch sensors would be needed, for a more accurate position.
Camera Keep track of the position in real time	Always keep track of the position	The camera needs to keep track of the right object, chance of interference from users. Noisy in low light environment.

Table 2: Pros and cons of ways to measure the position along the sliding mechanism

Arduino

In the installation two Arduino nano 33 IoT boards will be used. On both boards scripts are running to receive input from the sensors and send this data to the online database (see Appendix 7 and Appendix 8 for the scripts). One board will receive and send input of the attraction and orientation zone, while a second board will receive and send input of the repulsion zone and restart button, see Figure 69 for an overview.
In both scripts, the Arduino listens to values from the time of flight sensor and buttons. Since updating a value to the online database requires time, only when a value is different from the previous value, will it be updated to the database. While this resulted in clean data from the buttons, it was found that the output of time of flight sensors was not stable but fluctuated a little, when the position of the sliding mechanism was not being changed. To filter out this noise, only when a value is different to the previous value, and bigger or smaller than the range of noise, would the value be updated to the database.

The way the Arduino boards and the visualisation algorithms communicate with each other, can be found in the overview in Figure 70.

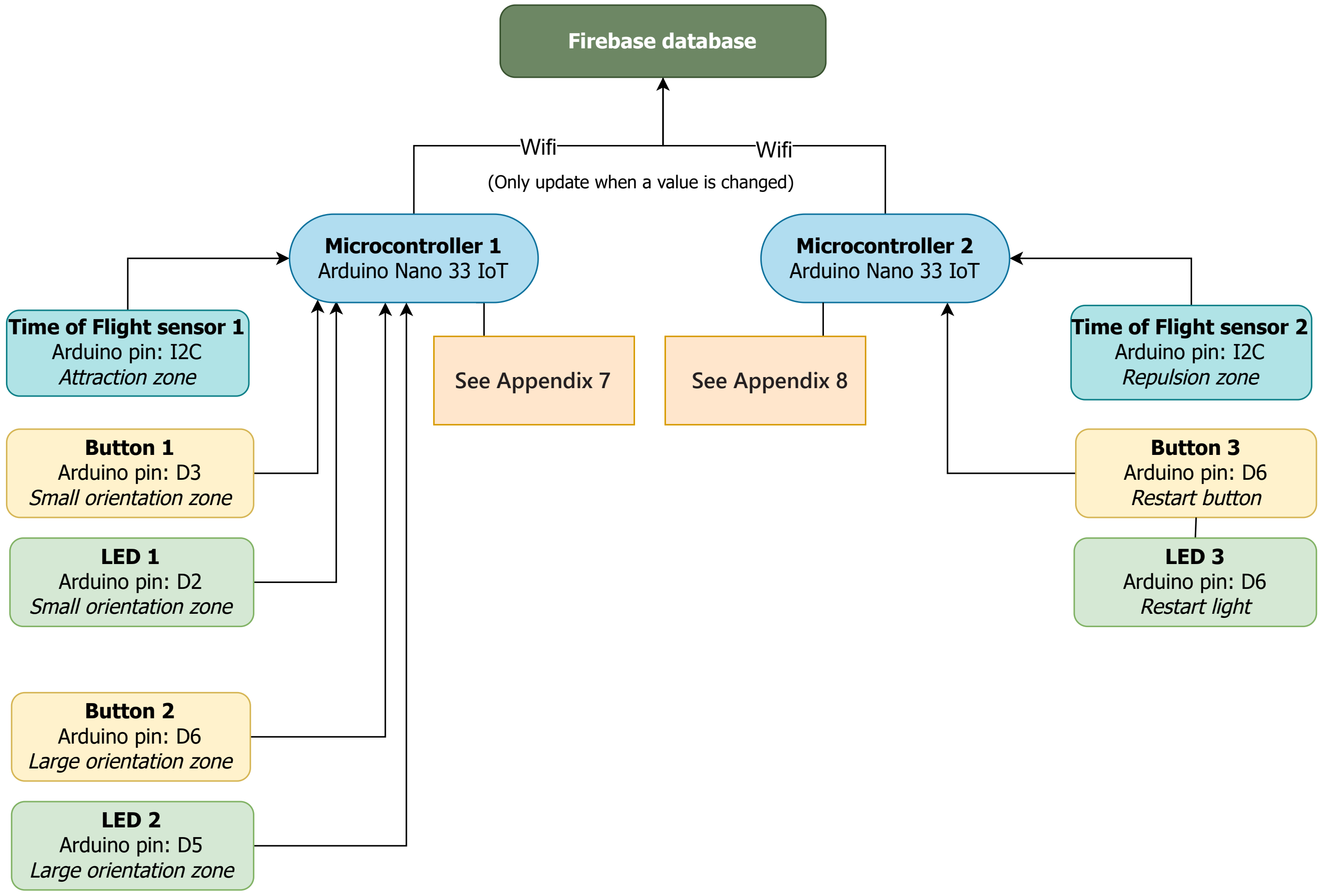


Figure 69: System diagram of both Arduino boards.

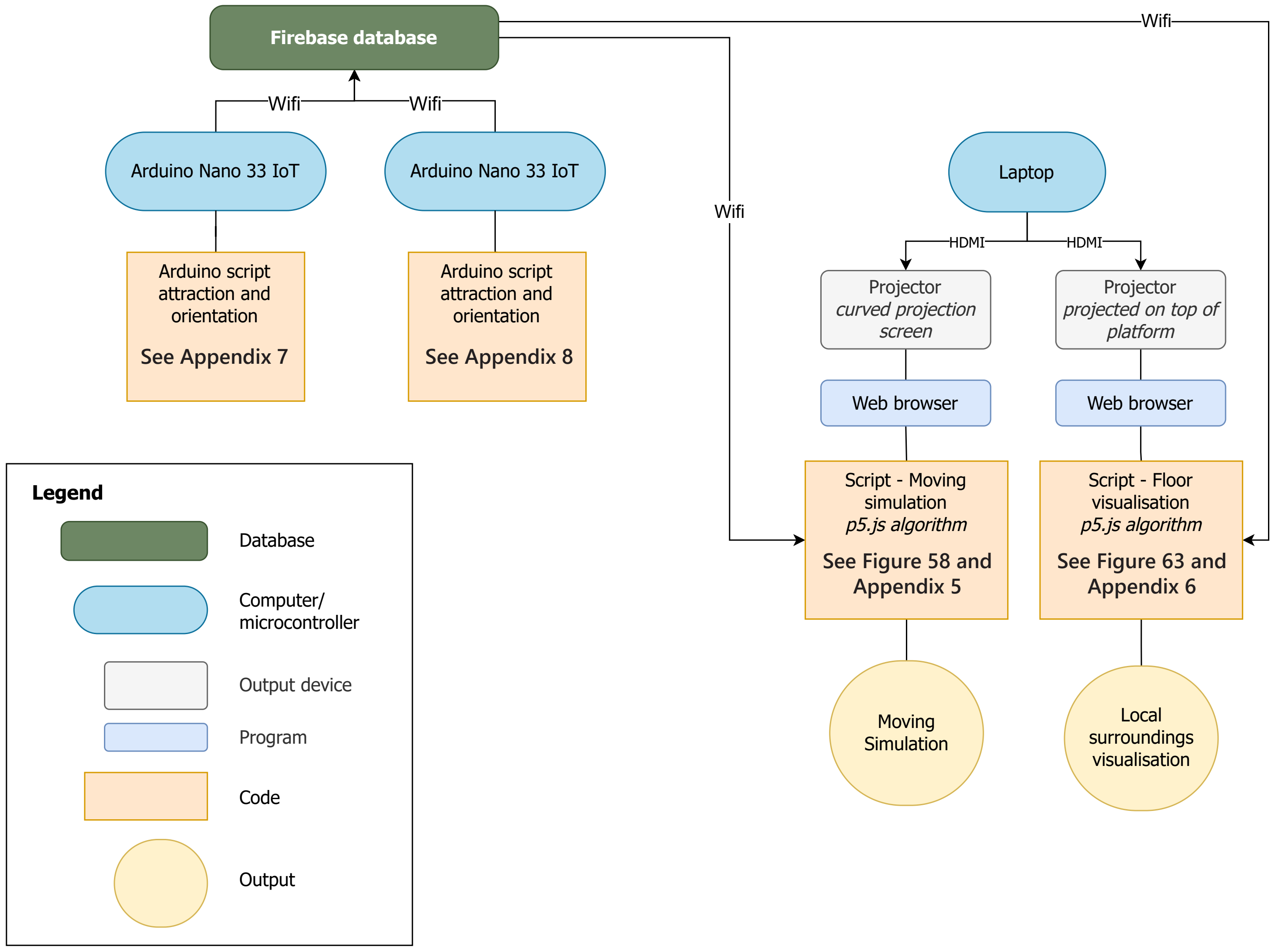


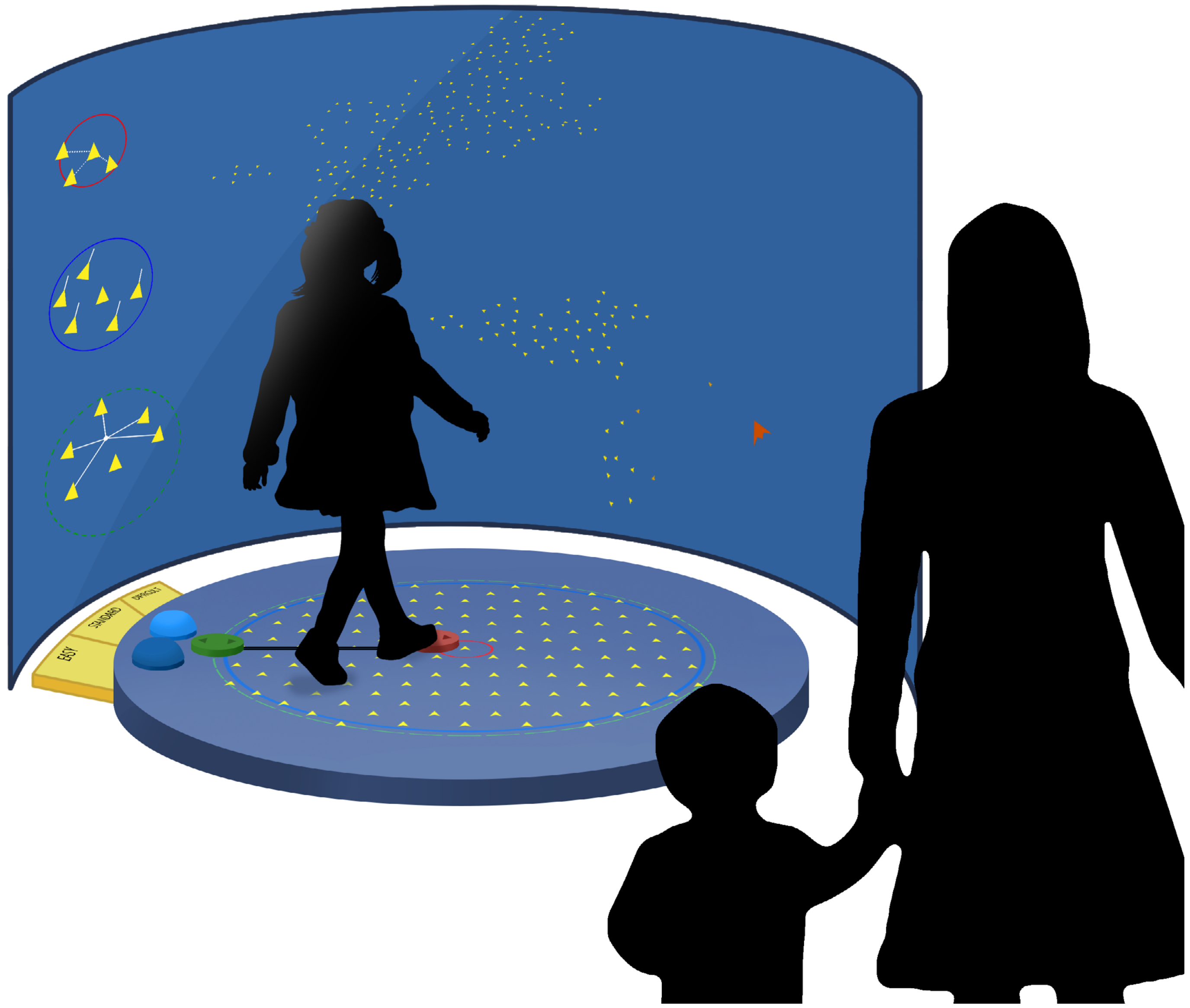
Figure 70: Overview of the communication between the Arduino boards and the two visualisation algorithms.

7.2 FINAL LOOK OF THE INSTALLATION

The final look of the installation is shown to the right. Museum visitors can learn about collective behaviour in a playful immersive experience. The museum installation is called: Control the Collective.

Learn in a playful interactive way how local interactions shape collective behaviour.

CONTROL THE COLLECTIVE



7.3 VALIDATION

The following parts of the Control the Collective installation will be validated. They are listed below in order of the user interaction with the installation.

- Do users feel invited and want to interact with the installation, when they walk past it?
- Do users understand how to control the installation?
- Do museum visitors stay engaged long enough to learn something?
- Do museum visitors learn something new?

To be able to validate these, some preparations have to be made.

7.3.1 Location to validate

The preferred place to validate this installation would be in a museum, in particular Naturalis, since it is designed for a museum. However, some difficulties were encountered when asking different museums for the possibility to validate a prototype with their visitors. First of all, the installation should fit within the theme and values of the museum. Secondly, because of Covid-19, health measures were implemented in museums which made it difficult to find an opportunity to validate my prototype.
A backup plan was created if it was not possible to validate the installation in a museum. This would take place in the university of the TU Delft. A downside to this location is that finding out how long users stay engaged and if users feel invited would not be tested in the right setting of being in a museum.
Eventually, Naturalis made it possible to place a prototype in the LiveScience section of their museum (Figure 71 and Figure 72). This was the perfect kind of space to test if users feel invited and see how long they stay engaged, while evaluating with the actual target group.

Figure 71: Entrance to LiveScience in Naturalis

Figure 72: Space to set up the prototype. This is part of the LiveScience room people can roam through.

7.3.2 What to include in the prototype

A prototype is needed to validate Control the Collective in real life with the target group.
The following parts should be included in the prototype:

- Moving simulation visualisation
- Local surroundings visualisation
- Two linear movements to manipulate the repulsion and attraction zone.
- Two buttons to manipulate the orientation zone.
- A restart button.

These elements are chosen so the user can see the overall look of the installation, can interact with it, and see the change in the moving simulation and the visualisation of the local surroundings.
Some changes have been made since only part of the installation is being tested. Instead of using a curved projection screen, the moving simulation will be visualised on a large screen. For the visualisation on the floor a projector is placed on a table, so it does not need to be hung from somewhere high.
For the buttons in the prototype, large arcade buttons will be used. These are sturdy and have a LED inside. One button is used to restart the simulation, for every new user to bring all agents back. The simulation will start without a predator, which will be added after 30 seconds. This way, in the test set-up users will experience the interaction with and without a predator, to see what they prefer.
The background explanation videos will not be included in the prototype, instead static explanation visuals will be used.

7.3.3 Building the prototype

Platform

Since this prototype is validated in Naturalis, they have asked to make sure it is safe for visitors to use. Therefore, just like in the design for the installation, a raised round platform will be made. Here, the sliding mechanism can be placed underneath, so only the movable part of the rails will be reachable to the visitors (see Figure 76). All electronics can be hidden underneath and a smooth surface is created for the projection to project on. The circular platform is supported by beams and can be split in two for ease of transportability.

Electronics

The time of flight sensors should be put in a place on the linear rail, to measure the distance of the disks on the sliding mechanism. Different designs were 3D modelled in Solidworks and 3D printed (shown in Figure 73). Important aspects were a tight fit so the sensor could not move around, and staying underneath the surface of the circular platform so the user does not interfere with the measurement.
The buttons are implemented in the design by placing them along the top surface of the platform.
During the evaluation the Arduino boards will be powered by two battery packs, while they are connected via WiFi to the database via a phone hotspot.

Background information

To give users context of the three different zones, three visuals will be available for the users to look at (see Figure 75).

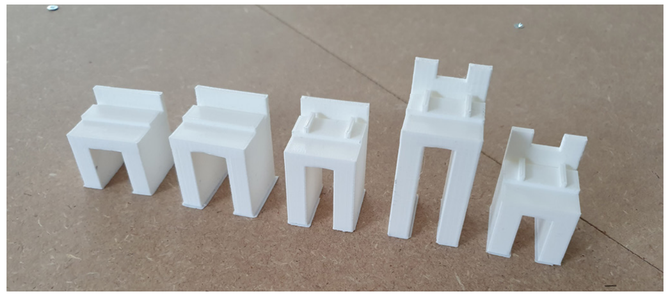


Figure 73: Several 3D printed iterations to find the right design for the time of flight sensor

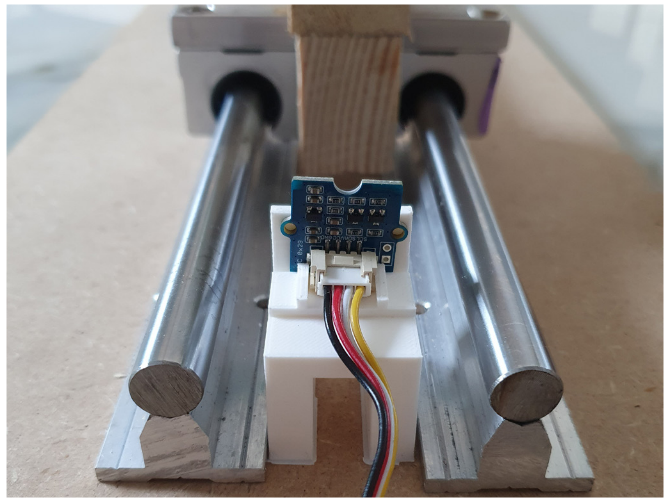


Figure 74: Chosen 3D print for time of flight sensor holder

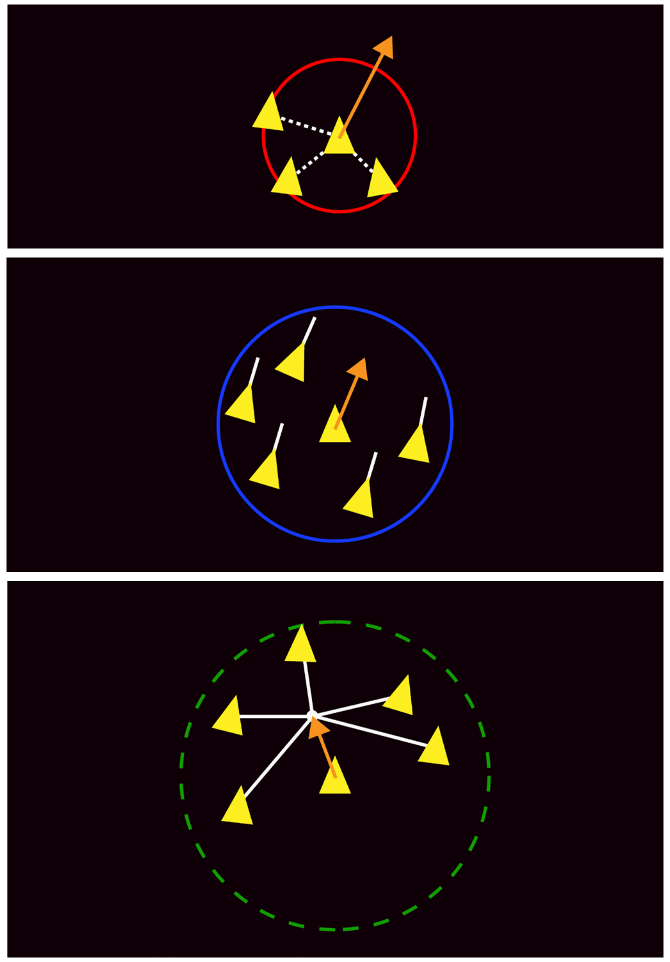


Figure 75: Explanation for the zone of repulsion, orientation and attraction

Figure 76: Prototype that is used for the validation in Naturalis. The two disks to move the sliding mechanism are painted in the colour of the corresponding circle, to let the user know which disk corresponds to which circle. Arrows are added on top to give users the use-cue that they are movable. The buttons are annotated with the words 'small' and 'large' for orientation and 'restart' for the restart button.

7.3.4 Day at Naturalis - results

The prototype of the installation was placed for a full day in the LiveScience room in Naturalis for evaluation and validation (see Figure 79 to Figure 81 and Appendix 9). LiveScience is located on the ground floor of Naturalis and functions as a shop window. The museum installation could be seen when entering this room. Museum visitors were observed and interviewed (with the help of assistants).

Observation results

In a span of two hours, groups of people that entered the room were counted (see Figure 77). It was found that almost all these groups were families, consisting of parents with their children. Five of these groups came to the installation and could immediately interact with it. 30 groups stopped at the installation but had to wait since the installation was in use, of which 20 groups waited for their turn to play. The last 22 groups that were counted walked past the design and did not stop.

The time that the first twelve groups spent at the installation were recorded (see Figure 78). On average, the installation was used for 3 minutes and 10 seconds. In some groups, children wanted to continue playing, but the parents wanted to move on to the other parts of the museum, while in other groups children were walking on their own and spent a shorter time at the installation. In the groups that stayed relatively longer at the design, the parents were involved and interacted together with their children to help them understand quicker. Since the installation was continuously in use, some groups stopped their interaction when the parents decided to make room for other visitors.

Use results

The restart button was used by all users at the start of their interaction. Due to the placement of the button at the front of the design, it was the first thing they saw. The repulsion and attraction zones sliders were explored after. The orientation buttons were used by

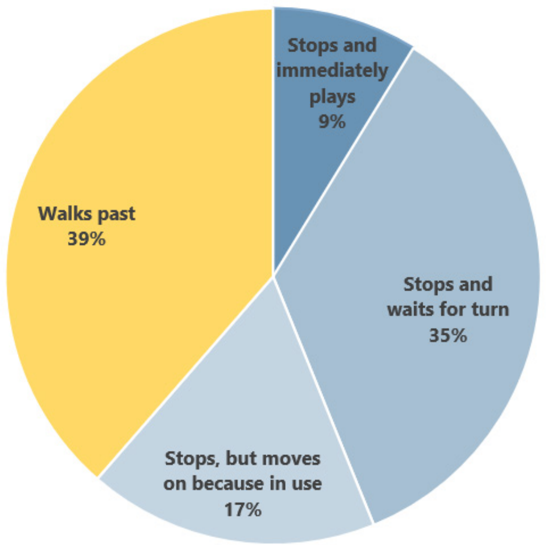


Figure 77: 57 groups of people were counted and categorised into: stops and immediately plays, stops and waits for turn, stops but eventually moves on, and walks past

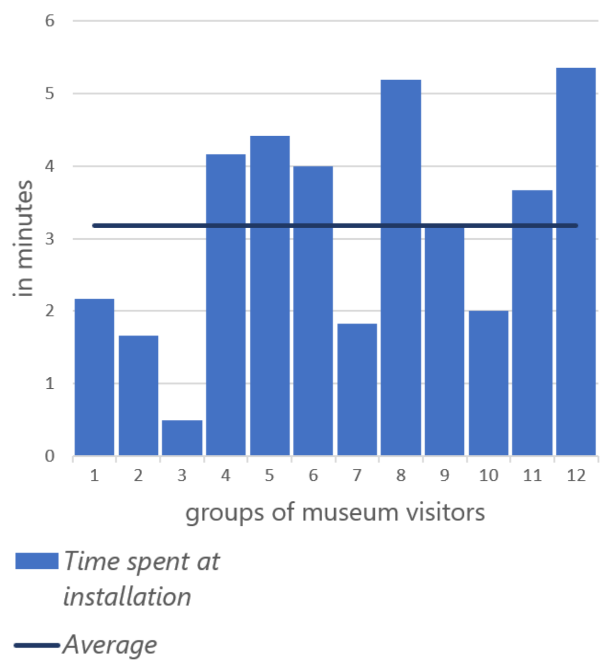


Figure 78: Time visitors spent interacting with the installation

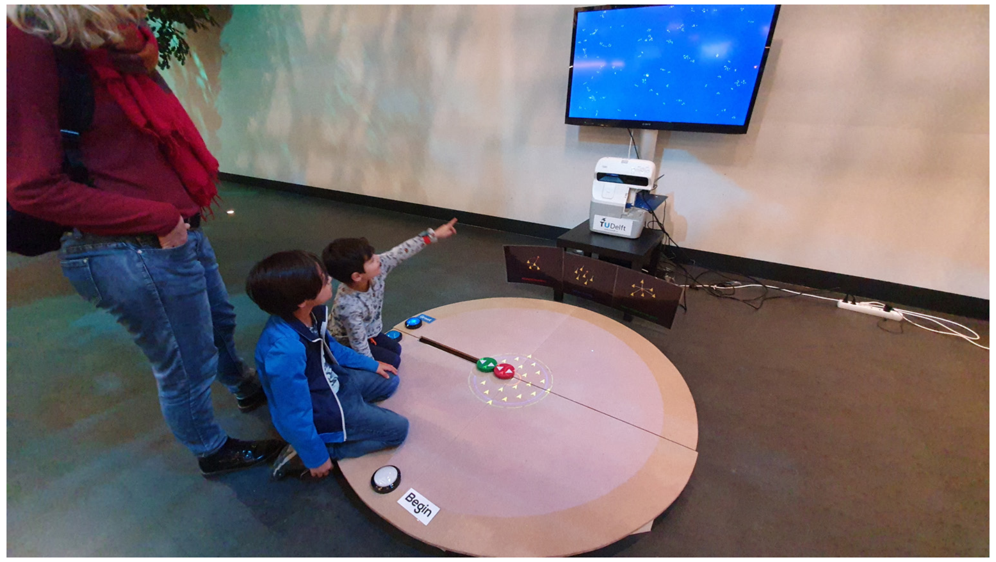


Figure 79: Prototype of the installation in Naturalis museum in use by museum visitors

Figure 80: Prototype of the installation in Naturalis museum in use by museum visitors

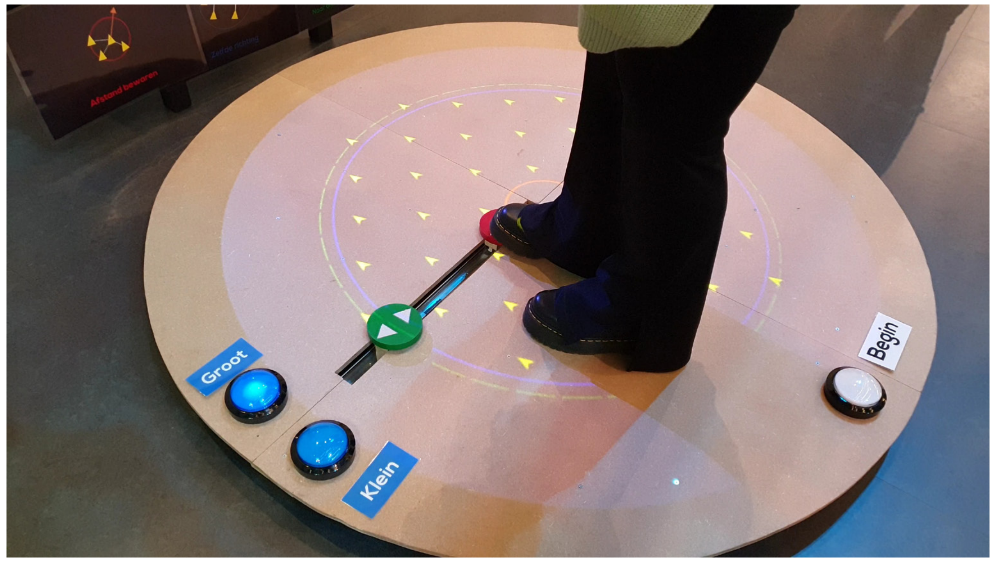


Figure 81: Close up of the sliding disks and the buttons of the prototype

everyone, except for the group of children that interacted with the installation for only 30 seconds (visitor group 3 in Figure 78).

While the adjustments of the size of the repulsion and orientation zones showed a difference in the moving patterns quickly, it took more time to see this change when adjusting the size of the attraction zone. It was shown that (especially young) children did not have patience, and slid the disks back and forth without stopping in between. Since they did it so fast, only the visualisation on the floor changed, while the visualisation on the screen did not show an effect as fast. When this happened while the parents were also involved, they told them to slow down and see what happens first.

The people that read the explanation sheets, tried to adjust the different zones accordingly. The influence of the change of the repulsion and orientation zones was understood correctly. However, the change in the size of the attraction was sometimes not fully understood at once. When it was explained verbally, while also showing them the resulting change on the floor and on the screen, they did comprehend it. This shows that only a static explanation is not enough for everyone to quickly understand how it works.

The parents that did not notice or read the explanation sheets, asked what the installation was about. When they were told how the three behavioural zones influenced the group, they started to explain it to their children and tried out what happens with the change of size of the zones.

Interview results

Users were interviewed after playing with the design. They liked that they could see both locally what was happening on the floor and how it influenced the behaviour of the moving group on the screen. Parents said that while learning in a playful manner, it was easier for their children to remember what they had learned. Doing something instead of just reading or looking at something, makes it more fun and therefore makes them more curious to learn something new. Something else that was mentioned was the use of the different colours. This made it easy to link which circle belonged to which disk or button. Stepping on top of the platform was associated with feeling like stepping inside a game, like feeling immersed. However, some visitors said that they were very focused on what was happening on the floor at the beginning, and did not immediately take notice of what was happening on the screen. Nonetheless, what was happening on the floor was clear to them, because they could immediately notice the change in the visualisation whenever they adjusted something. The issue where people did not immediately notice the screen, could be due to the test set-up (see Figure 82). The screen was relatively smaller than the platform and users had to look up from the floor to see the screen. So, the visibility of the screen is of importance.

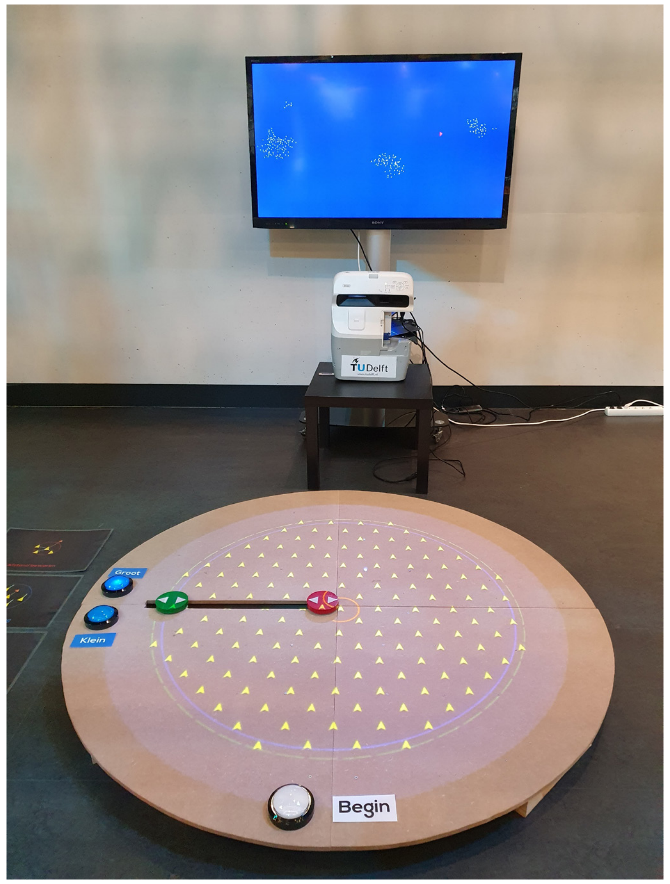


Figure 82: Prototype of the installation in Naturalis museum

7.3.5 Evaluation of the desirability, feasibility and viability of the installation

Looking at the developed museum installation, it can be evaluated on three main themes: its desirability, feasibility and viability. This framework brings a balance in seeing if it brings value to users (people), if it can be done (technology) and if it is viable to be created (business).

Desirability – Does it bring value?

The evaluation at Naturalis has shown that many museum visitors were interested in the installation and wanted to interact with it. More than 60% of the visitors that saw the design, came up to it wanting to interact with the installation. Visitors enjoyed the experience since they actively had to do something. While the installation is not only designed to be fun to play with, it also educates its users about the concept of collective behaviour. Parents were interested as well in what the design was about and they were involved in learning with their children how the different patterns of collective behaviour take shape.

An employee of Naturalis, who guides and gives workshops, said that especially children enjoyed interacting with the installation since they are easily stimulated by playing games. They are in control of the installation, while it also shows them the effect. He mentioned he can imagine this installation being part of Naturalis, since they also have other interactive installations, which visitors can play with and learn from.

Since the aim of this design is to teach the general public about the concepts of collective behaviour, it can also be used to support researchers explaining collective behaviour, by aiding in giving a visual interactive explanation. For example, it can support communicating Jan Jaap's research to the general public (van Assen, 2021).

Concluding, the installation is desirable for multiple stakeholders: museum visitors, museums and researchers.

Feasibility - Can it be done?

The installation consists of physical, electronic and software elements. The prototype used in Naturalis showed that the current software and electronics are feasible to create the installation. The software used for the visualisation is written in JavaScript, which can run in several programs, and even in a web browser. The electronics that provide the input from sensors run on an Arduino. Naturalis uses Arduinos for other installations in their museum as well and is thus a good fit in their museum infrastructure.

For the physical parts of the installation, it is important that they are museum proof. In the prototype, wood is used for the platform and plastic buttons were implemented. Those materials were robust enough in a museum setting, where children could use force and jump on top. For the sliding mechanism, two disks on top of two linear guide rails were used. In the prototype this worked smoothly. However, all elements in the prototype were created rather separately, so it was easily transportable. In the actual manufacturing of the installation all physical elements should be created in a way so they can all be connected together. Nevertheless, knowing what needs to be implemented in the design, this seems feasible to do.

Viability – Will it survive on a longer term?

The evaluation showed that not everyone that wanted to interact with the installation had the opportunity to do so since it was constantly in use. Naturalis has said that most of the time, visitors are not able to see everything they want to see in one visit, which makes them come back. Having another design in their museum that visitors want to go back for, fits in their goals for the museum.

The aim of the installation is to teach visitors new knowledge, which they will remember and use in their everyday life. When understanding in a better way how collective behaviour works, they will have a different view of how every around them is connected. For example, they can use this knowledge when trying to go faster through a crowded place.

8. CONCLUSION

8.1 CONCLUSION

The goal of this project was to find a way to make the general public aware of collective behaviour. The question was how the user experience would look like and how this would take shape in a design. Since collective behaviour is a complex phenomenon, the challenge was to find a way to create a design that would be able to explain this concept in a simple way.

The design that has been created is a museum installation designed for Naturalis, a national museum of natural history and a research center on biodiversity. The target group of this installation are visitors of age 8 and older. By interacting with this museum installation, Control the Collective, users can manipulate how every agent reacts to neighbours and their surroundings, and thus influence the behaviour of the whole group. They can do this by adjusting the size of three zones, in which the agents behave a certain way. These are the zone of repulsion, the zone of orientation and the zone of attraction. The size of these zones and the corresponding composition of an individual within the group are visualised on the floor. Simultaneously, on a curved wall, the accompanying visualisation of the moving group is displayed.
Users can manipulate the size of the repulsion and attraction zones, by sliding a disk along a line on the floor, increasing or decreasing the radius of the circle. The zone of orientation can be made either small or large, by pressing the corresponding button. These disks and buttons are placed on top of a circular platform, which users can stand on top of, while the moving simulation wall curves around it.

The Body, mind, heart and soul framework is taken into account while designing this user experience. The body is used actively to control the installation. The mind is challenged by learning about collective behaviour and finding the connection between what is being adjusted and its effect on the group. The experience is designed to be playful and focuses on positive emotions. Lastly, learning about this topic is meaningful since it relates to many different things in our daily life, from being in a crowded hallway, to how we are mesmerized by flocks of birds.

Looking back at the requirements that were set up at the beginning of the project, many have been achieved, while some can be further explored in future research. The installation was designed to fit in a museum, which was validated by placing a prototype of the installation in Naturalis. With this prototype, museum visitors could learn about the local interactions of repulsion, orientation and attraction behaviour, which shape collective behavioural patterns. It was found that the effect it had on the composition of an individual in the group, was understood more quickly than its effect on the moving group. This could be due, however, to a limitation of the study, where the screen of the prototype was small compared to the designed version. The screen, which showed the moving simulation, was sometimes overlooked by users when interacting for the first time. The presence of the restart button helped users in finding a starting point when seeing the installation for the first time. The ability to slide the disks and press the buttons was intuitively understood, however, the way the users did this varied. Some users used their feet, while others, mainly children, sat on the platform while using their hands. Visitors stayed engaged for an average of 3 minutes and could recognize flocks of birds, schools of fish and other animal groups in the patterns they created.

Concluding with the four levels of understanding that were aimed to be understood by the visitors, it was found that for the first three levels comprehension was achieved for most users. Due to the feedback of the visualisation that was shown on the floor, users could easily make out the change of behaviour of an individual towards their local surroundings. On the screen, they were able to see the effect it had on the whole group. However, understanding level 4, how the combination of all variables led to a certain result in the group, was not accomplished yet for everyone. To lead everyone to this deeper level of understanding, extra feedback should be given to the users to guide them through the full learning process.

8.2 RECOMMENDATIONS

The final concept of the Control the Collective installation can serve as a starting point for further development and implementation of a visually educative design to explain collective behaviour. This section will highlight recommendations for future research.

- Since some users moved the disks back and forth at a high rate, the moving simulation did not always have time to update. To be able to give users better feedback, so they can understand what they are adjusting in a more streamlined way, the disks should be moved at a slower rate. This can be achieved by for example making the disks heavier.
- Since users obtained a better understanding of what they were manipulating when they asked about it and received a verbal explanation, some extra feedback in text shown on the screen could help aid their understanding. Cues like 'the agents now form a bigger group' or 'the agents now keep more distance between each other', can lead to better comprehension. The goal is that the users understand the behaviours in the different zones, in order to obtain an understanding of the influence of a combination of them.
- The way the platform and the screen will be embodied should be taken into account. Users should immediately know that they can step on top of it, and should not get the impression it is not how it is intended to be used. The curved screen should be visible, even when users are looking at what is happening on the floor. The screen should therefore reach to the floor. The way the projections and electronics are powered, should be considered.
- Further explorations and tests should be carried out to find out which types of restart buttons are desired in the design. For example, do users understand what is happening when a predator is already present at the beginning, or should every new user start without a predator first. The difference in difficulty levels between the restart buttons should also be explored.

8.3 PERSONAL REFLECTION

During the project, I had some struggles with finding a balance between exploring enough different directions and deciding to choose something. I also realized that I expected to fit too many of my goals in a tight planning. I wanted to do a lot things and create high fidelity prototypes early on. However, this took more time than expected and resulted in detailing things too early. First having a solid basis before going into detail is something I learned along the way. I had to let go of the feeling of thinking I did not receive a result, when I created something that could not be 'seen or touched'.
My original planning started with a research phase, continued by an exploration phase which would result in a chosen concept that would be finalized. However, this path was not fully followed. The main change was that instead of having a chosen concept that was finalized, the final concept kept being adjusted and improved. During the exploration phase, I also went back to the research phase. By being immersed in the project for some while, this actually helped me understand and implement the literature better.

During my graduation project I explored various things I still wanted to do during my time as student. One of these was the way of creating the prototype. Up until my graduation project I mainly used digital designs or 3D prints to showcase my designs, but this time I wanted to go out of my comfort zone and use new ways of making a prototype. In terms of using code in my designs, I was used to programming in Arduino. During my graduation project, I started with the use of an algorithm written in Processing, which is similar to Arduino. However, I found another algorithm that implemented other features I needed. This one used JavaScript and html, two coding languages I had never used before and explored for the first time during my graduation project. I learned many new things about coding, which I think is very valuable knowledge as an industrial designer. Since I also wanted to improve my visualisation skills, I used ideation by sketching during the first phases of the project.

In the end, I am happy with what I achieved during my graduation project. I learned many new things that I can take with me to the future.

9. REFERENCES

9.1 REFERENCE LIST

Abreu, S. (2021, January 21). *Boids - the emergence of flocks*. YouTube. Retrieved January 14, 2022, from https://www.youtube.com/watch?v=pXQCasmBhY4

Anderson, M. (n.d.). *Concrete/Reflective/Abstract/Active – David Kolb*. Retrieved January 10, 2022, from https://web.cortland.edu/andersmd/learning/kolb.htm

Aqib, M. (2018, December 26). *Arduino Button Tutorial Using Arduino DigitalRead Function*. Arduino Project Hub. Retrieved January 14, 2022, from https://create.arduino.cc/projecthub/muhammad-aqib/arduino-button-tutorial-using-arduino-digitalread-function-08adb5

Arduino. (2018, February 5). *What is Arduino?* Arduino. Retrieved January 14, 2022, from https://www.arduino.cc/en/Guide/Introduction

Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., Viale, M., & Zdravkovic, V. (2008). Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences*, *105*(4), 1232–1237. https://doi.org/10.1073/pnas.0711437105

Ben. (2013, February 26). *What is an Arduino?* SparkFun. Retrieved January 14, 2022, from https://learn.sparkfun.com/tutorials/what-is-an-arduino/all

Boardtobits. (2019, March 14). *Scripts from the flocking algorithm tutorial from Board to Bits.* GitHub. Retrieved January 14, 2022, from https://github.com/boardtobits/flocking-algorithm

Bulmer, A. (2017, February 1). *The science of stalking fish*. Active Angling New Zealand. Retrieved November 1, 2021, from https://activeanglingnz.com/2017/02/01/the-science-of-stalking-fish/

Butler, S. R., Hosinski, E. C., Lucas, J. R., & Fernández-Juricic, E. (2016). Social birds copy each other's lateral scans while monitoring group mates with low-acuity vision. *Animal Behaviour*, *121*, 21–31. https://doi.org/10.1016/j.anbehav.2016.08.002

Bär, E., & Boshouwers, S. (2018). The wow moment. In *WOW Worlds of wonder: Experience design for curious people* (pp. 51–57). essay, BIS Publishers.

Cabos, L. (2018, January 8). *3 reasons why fish swim in schools*. Retrieved November 23, 2021, from https://www.visitloscabos.travel/blog/post/3-reasons-fish-swim-schools/

Chapman, A. (2017, June 7). *Kolb's Learning Styles*. Retrieved November 23, 2021, from https://www.businessballs.com/self-awareness/kolbs-learning-styles/

Cherry, K. (2020, May 15). *Kolb's Theory of Learning Styles*. Retrieved November 23, 2021, from https://www.verywellmind.com/kolbs-learning-styles-2795155

Couzin, I. (2021, April 27). *Iain Couzin on Collective sensing and decision-making in animal groups*. YouTube. Retrieved November 1, 2021, from https://www.youtube.com/watch?v=y9HYtUeB2ms.

Couzin, I. D., Krause, J., Franks, N. R., & Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, *433*(7025), 513–516. https://doi.org/10.1038/nature03236

Couzin, I. D., Krause, J., James, R., Ruxton, G. D., & Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, *218*(1), 1–11. https://doi.org/10.1006/jtbi.2002.3065

Cowell, P. (2019, January 10). *Fish school*. Retrieved January 10, 2022, from https://photos.com/featured/fish-school-paul-cowell-photography.html

de Weerd, H. (2016, October 26). *Boids: Flocking made simple*. Harmen de Weerd. Retrieved November 23, 2021, from https://www.harmendeweerd.nl/boids/

DigiDaan. (n.d.). *Beeldmateriaal*. NEMO Science Museum. Retrieved January 14, 2022, from https://www.nemosciencemuseum.nl/nl/over-nemo/pers/beeldmateriaal/

Foderaro, F. (2019, March 29). *Translate nature of code to Max - flocking and Boids part 2 / end*. YouTube. Retrieved January 14, 2022, from https://www.youtube.com/watch?v=bfo5XPJ_3Ho

Freeman, N. (2014, January 7). *Nick Freeman's Boids Synthesizer*. GitHub. Retrieved January 14, 2022, from https://github.com/nickfreeman/Boids-Synthesizer

Giardina, I. (2008). Collective behavior in animal groups: Theoretical models and empirical studies. *HFSP Journal*, *2*(4), 205–219. https://doi.org/10.2976/1.2961038

Hemelrijk, C. K., & Kunz, H. (2004). Density distribution and size sorting in fish schools: An individual-based model. *Behavioral Ecology*, *16*(1), 178–187. https://doi.org/10.1093/beheco/arh149

Jackaperkins. (2020, November 28). *Jackaperkins/Boids: The age old boids project!* GitHub. Retrieved January 14, 2022, from https://github.com/jackaperkins/boids

Kane, L. (2018, December 30). *The peak–end rule: How impressions become memories*. Nielsen Norman Group. Retrieved January 14, 2022, from https://www.nngroup.com/articles/peak-end-rule/

Klamser, P. P., Gómez-Nava, L., Landgraf, T., Jolles, J. W., Bierbach, D., & Romanczuk, P. (2021). Impact of variable speed on collective movement of animal groups. *Frontiers in Physics*, *9*. https://doi.org/10.3389/fphy.2021.715996

Kumar, V., & De, R. (2021). Efficient flocking: Metric versus topological interactions. *Royal Society Open Science*, *8*(9). https://doi.org/10.1098/rsos.202158

Kurt, S. (2020, December 28). *Kolb's experiential learning theory & learning styles*. Retrieved December 23, 2021, from https://educationaltechnology.net/kolbs-experiential-learning-theory-learning-styles/

Langston, E. (2016, January 22). *Audubon's Zachary Slavin explains why starling flocks captivate*. Audubon. Retrieved January 14, 2022, from https://www.audubon.org/news/audubons-zachary-slavin-explains-why-starling-flocks-captivate

Larsson, M. (2012). Why do fish school? *Current Zoology*, *58*(1), 116–128. https://doi.org/10.1093/czoolo/58.1.116

Lukeman, R., Li, Y.-X., & Edelstein-Keshet, L. (2010). Inferring individual rules from Collective Behavior. *Proceedings of the National Academy of Sciences*, *107*(28), 12576–12580. https://doi.org/10.1073/pnas.1001763107

Lushi, E., Wioland, H., & Goldstein, R. E. (2014). Fluid flows created by swimming bacteria drive self-organization in confined suspensions. *Proceedings of the National Academy of Sciences*, *111*(27), 9733–9738. https://doi.org/10.1073/pnas.1405698111

MacGregor, H. E., Herbert-Read, J. E., & Ioannou, C. C. (2020). Information can explain the dynamics of group order in Animal Collective Behaviour. *Nature Communications*, *11*(1). https://doi.org/10.1038/s41467-020-16578-x

Max Planck Institute for terrestrial Microbiology. (2019, December 18). *Collective motion of Escherichia coli at high cell density*. Retrieved November 23, 2021, from https://www.mpi-marburg.mpg.de/797385/2019-12-b1

Mind Tools. (n.d.). *The build-measure-learn feedback loop: Creating real value by testing ideas*. Retrieved January 10, 2022, from https://www.mindtools.com/pages/article/build-measure-learn.htm

Naturalis. (2022). *Naturalis Biodiversity Center*. Retrieved January 10, 2022, from https://www.naturalis.nl/en

NEMO. (n.d.). *Wonderlijke wetenschap - Tentoonstellingen*. NEMO Science Museum. Retrieved January 10, 2022, from https://www.nemosciencemuseum.nl/nl/wat-is-er-te-doen/tentoonstellingen/wonderlijke-wetenschap/

Pendleton, B., & Goodrich, M. (2013). Scalable human interaction with robotic swarms. *AIAA Conference Paper*. https://doi.org/10.2514/6.2013-4731

Pierdicca, R., Marques-Pita, M., Paolanti, M., & Malinverni, E. S. (2019). IOT and engagement in the ubiquitous museum. *Sensors*, *19*(6), 1387. https://doi.org/10.3390/s19061387

Pita, D., Moore, B. A., Tyrrell, L. P., & Fernández-Juricic, E. (2015). Vision in two cyprinid fish: Implications for collective behavior. *PeerJ*, *3*. https://doi.org/10.7717/peerj.1113

Rahman, S. R., Sajjad, I., Mansoor, M. M., Belden, J., Murphy, C. T., & Truscott, T. T. (2020). School formation characteristics and stimuli based modeling of Tetra Fish. *Bioinspiration & Biomimetics*, *15*(6), 065002. https://doi.org/10.1088/1748-3190/aba2f6

Reynolds, C. (2007, July 30). *Boids*. Boids (flocks, herds, and schools: A distributed behavioral model). Retrieved November 1, 2021, from http://www.red3d.com/cwr/boids/

Rogers, D. (2019, November 10). *Build, measure, learn cycle*. Medium. Retrieved January 10, 2022, from https://medium.com/@dominic_11011/build-measure-learn-cycle-ace388a13b4d

Schep, M., van Boxtel, C., & Noordegraaf, J. (2015). *Wat je kunt leren van een educatieve museumrondleiding*. Landelijk Kennisinstituut Cultuureducatie en Amateurkunst. Retrieved January 10, 2022, from https://lkca.nl/wp-content/uploads/2020/02/ce42_2015_wat_je_kunt_leren_van_een_educatieve_museumrondleiding.pdf

Seeed Technology. (2022). *Grove*. Grove - Seeed Studio Electronics. Retrieved January 14, 2022, from https://www.seeedstudio.com/category/Grove-c-1003.html

Sergioabreu. (2021, January 21). *The Boids Algorithm implemented in Godot*. GitHub. Retrieved January 14, 2022, from https://github.com/sergioabreu-g/boids

Sumpter, D. J. T. (2005). The principles of collective animal behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *361*(1465), 5–22. https://doi.org/10.1098/rstb.2005.1733

Sumpter, D. J. T. (2010). *Collective animal behavior*. Princeton University Press.

Sumpter, D., Buhl, J., Biro, D., & Couzin, I. (2008). Information transfer in moving animal groups. *Theory in Biosciences*, *127*(2), 177–186. https://doi.org/10.1007/s12064-008-0040-1

Suwannaphoom, A. (2019, January 19). *Shibuya crossing from top view*. Retrieved January 10, 2022, from https://pixels.com/featured/shibuya-crossing-from-top-view-anek-suwannaphoom.html

The Coding Train. (2018, December 11). *Flocking simulation*. The Coding Train. Retrieved January 14, 2022, from https://thecodingtrain.com/CodingChallenges/124-flocking-boids

Tunstrøm, K., Katz, Y., Ioannou, C. C., Huepe, C., Lutz, M. J., & Couzin, I. D. (2013). Collective states, multistability and transitional behavior in schooling fish. *PLoS Computational Biology*, *9*(2). https://doi.org/10.1371/journal.pcbi.1002915

van Assen, J. J. R. (2021). *Collective Flow*. Jan Jaap. Retrieved February 11, 2022, from http://www.janjaap.info/

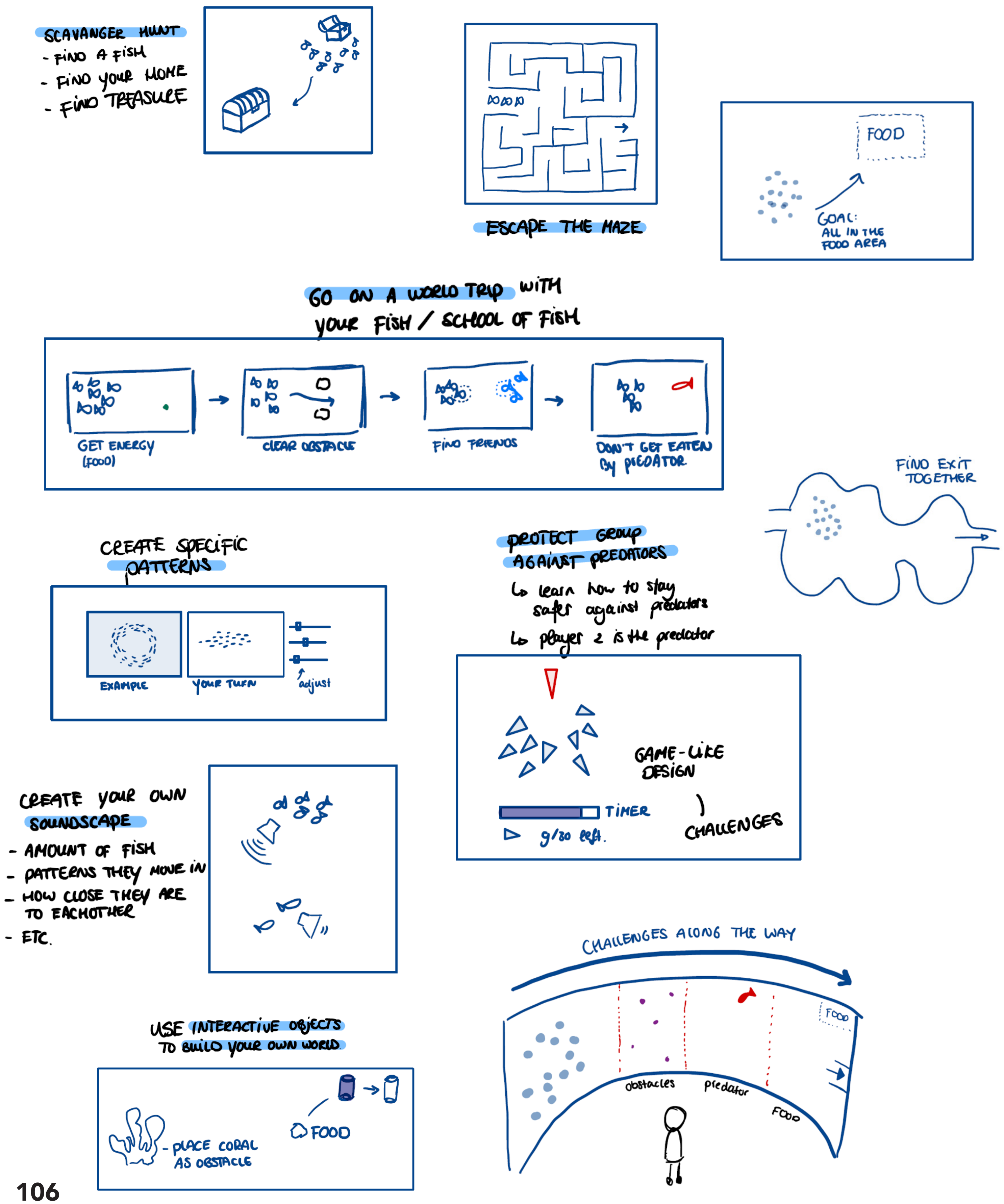
Vilardo, P., & Wepprecht, M. (2016, April 8). *Types of collective behavior*. Retrieved November 23, 2021, from https://pressbooks.howardcc.edu/soci101/chapter/21-1-types-of-collective-behavior/

Xu, H., Dauparas, J., Das, D., Lauga, E., & Wu, Y. (2019). Self-organization of swimmers drives long-range fluid transport in bacterial colonies. *Nature Communications*, *10*(1). https://doi.org/10.1038/s41467-019-09818-2

Zhang, H. P., Be'er, A., Florin, E.- L., & Swinney, H. L. (2010). Collective motion and density fluctuations in bacterial colonies. *Proceedings of the National Academy of Sciences*, *107*(31), 13626–13630. https://doi.org/10.1073/pnas.1001651107

10. APPENDIX

APPENDIX 1. STORYLINE EXPLORATION



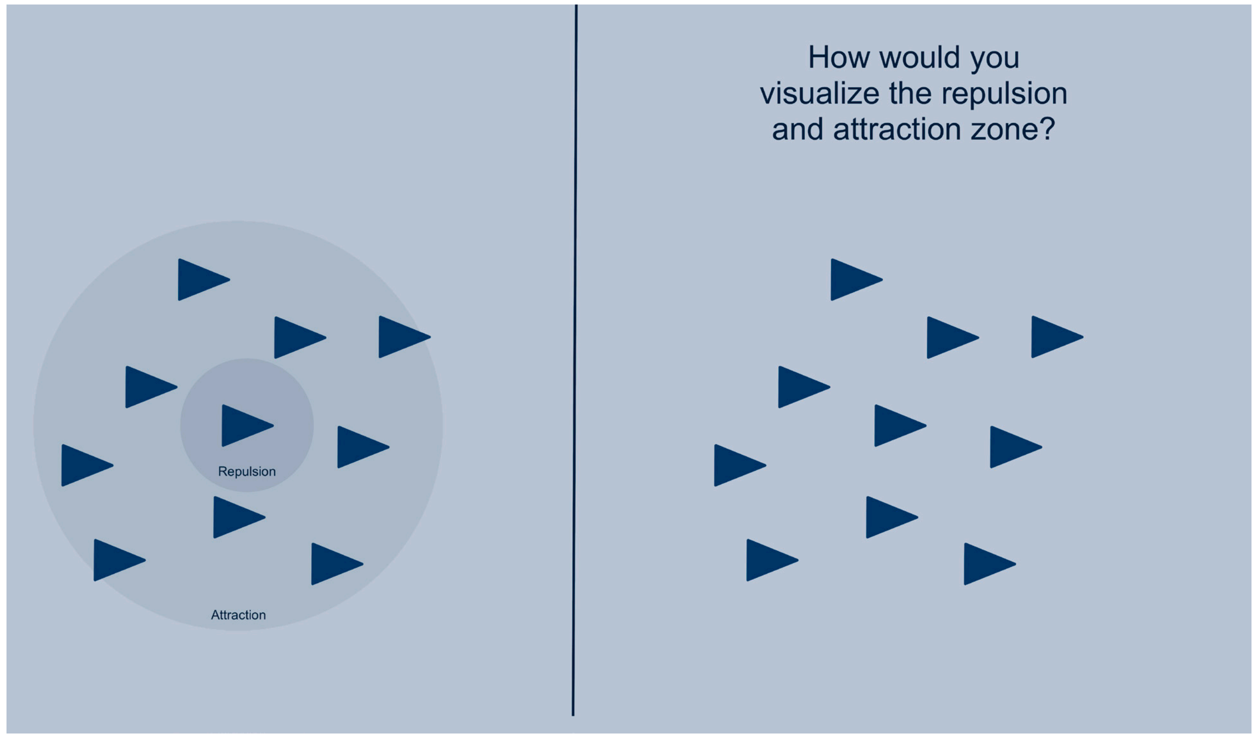
APPENDIX 2. VISUALISATION FOR REPULSION AND ATTRACTION ZONES

The initial set-up for this test was the following:
1. Explanation what the behaviour of agents is like in the repulsion and attraction zone.
2. Let the users draw the zones how they envision them
3. Show the options of the inspiration table
4. Let users change their initial answer

However, during the pilot test, everyone just drew a circle for both the repulsion and attraction zone, with no distinction between the two. The circles that were drawn were not the correct size of the zones. Therefore, the set-up was being changed to:

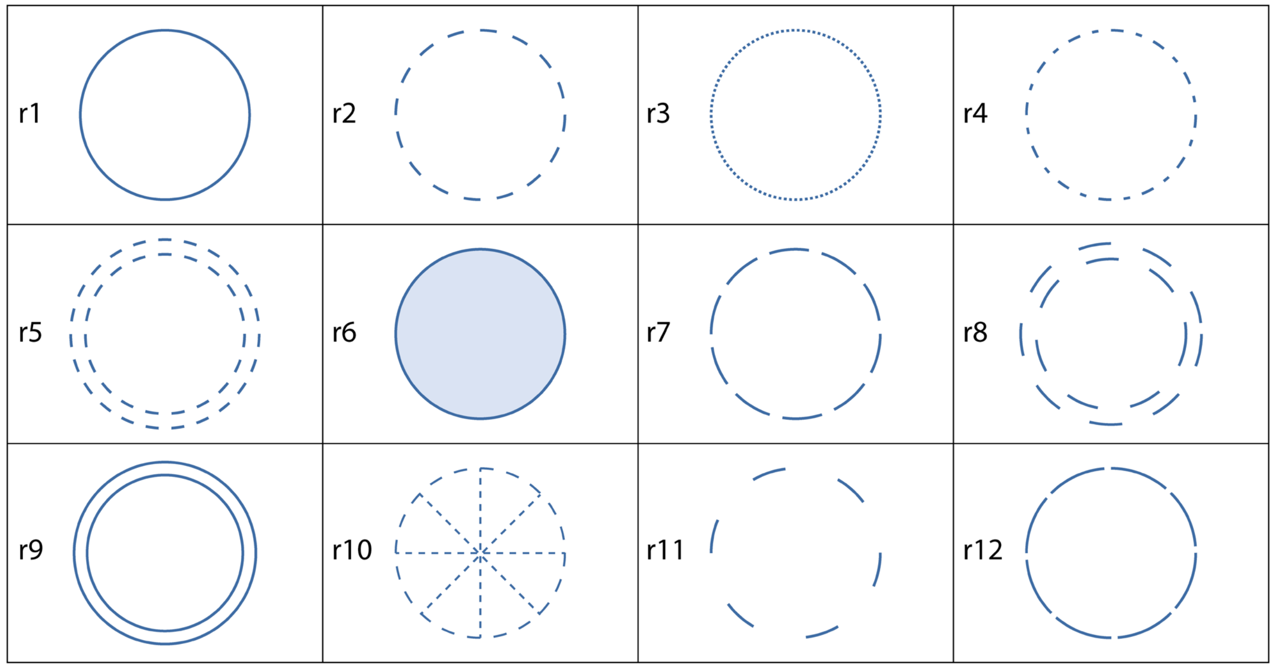
1. Explanation about the repulsion and attraction zone.
2. Show the options of the inspiration table
3. Let the users choose the visualisation of the zones in the figure as seen below

Explanation:
Imagine a school of fish or a flock of birds.
Every individual in the group does not want to be alone, so everyone looks around. When they see someone in the attraction zone they will move towards them.
But they also don't want to collide with others, so if they detect someone in the repulsion zone, they will move away and keep some distance.
How would you visualise these 2 zones? You can get inspiration from the table, or if you have another idea you can tell me.

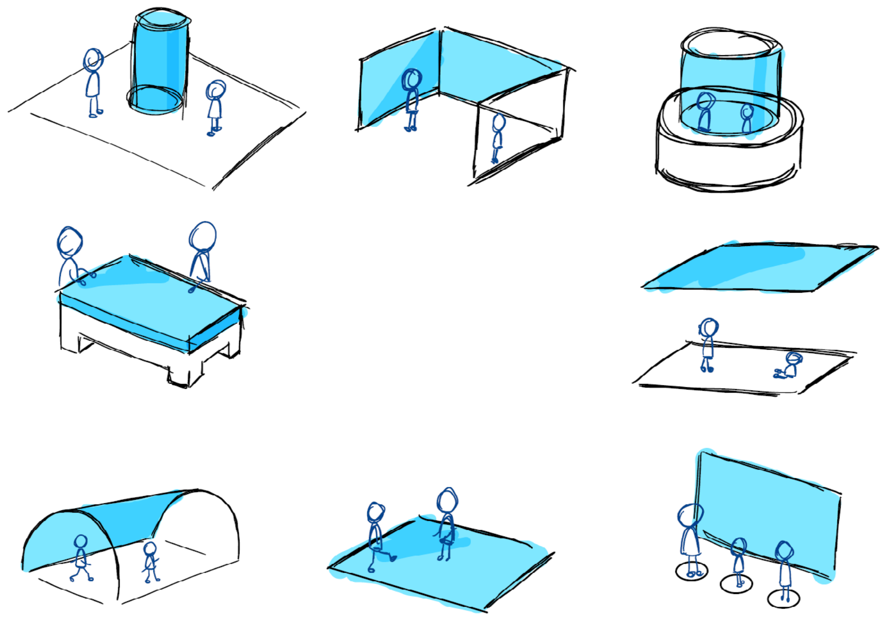


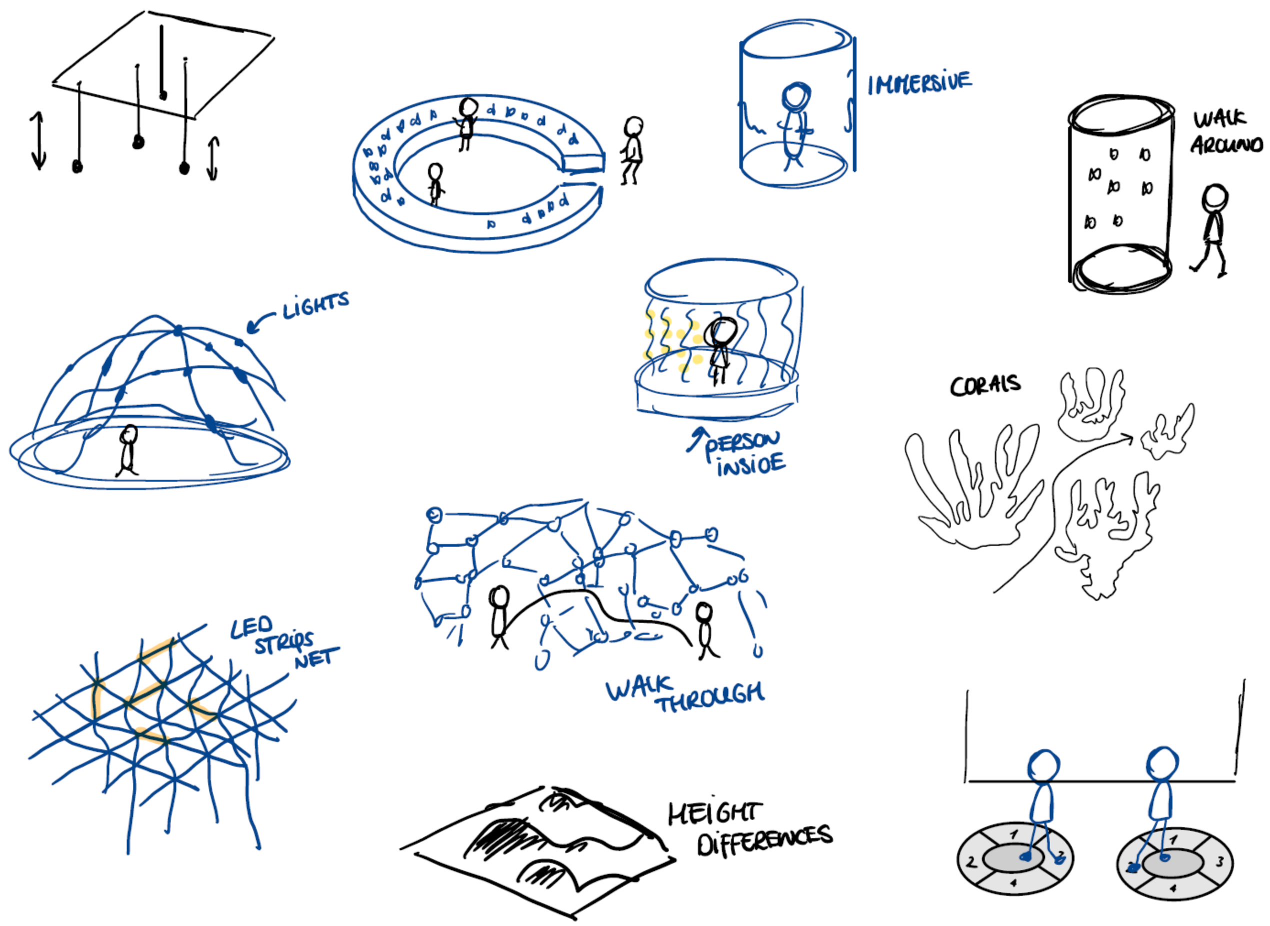
Results:

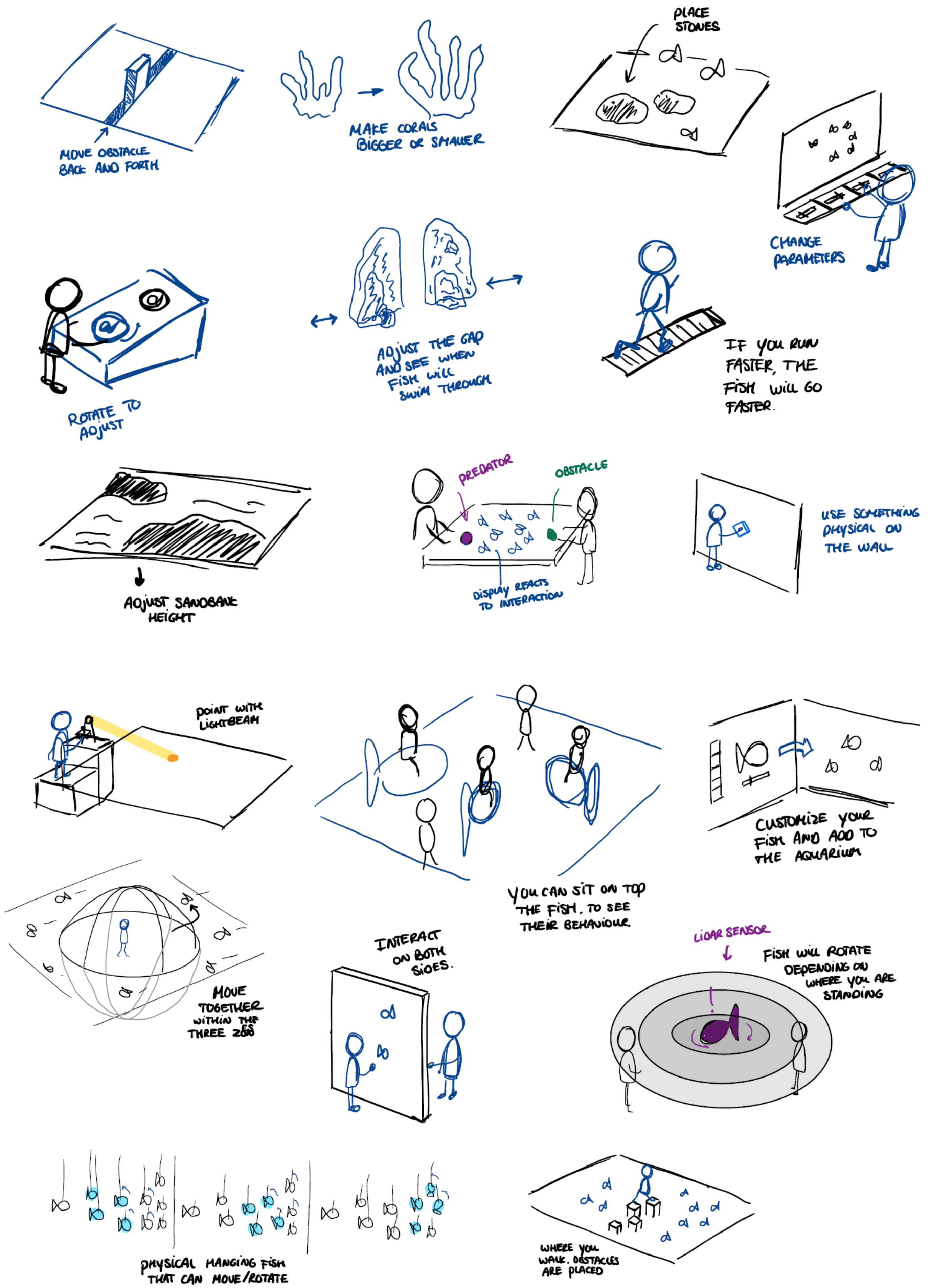
1. r1, r10. “The lines in r10 indicate where it looks for others”
2. “Arrows pointing outwards, arrows pointing inwards”
3. r1, r2
4. r4, and between r2-r11. “Both should be dashed lines, where the repulsion dashes are closer together.”
5. r1, r2. “Others can get past the attraction line, therefore dashed. But the repulsion zone should be a continuous line, because they should not collide.”
6. r1, r11. “Double circle like r8 is going to be overwhelming.”
7. r1, r1, “red and green”
8. “Gradient colour, r5/r8 double dash line for attraction”
9. r6, r2/r11. “Repulsion should be closed off, others can’t enter. For attraction, others can pass that threshold, dashed line.”
10. r6, r1. “R10 combined with outside arrows also possibility for repulsion”
11. r1, r1, “but coloured differently”
12. r5, r8
13. r6, r6, “but different colours. Like a shield around.”
14. r1, r1

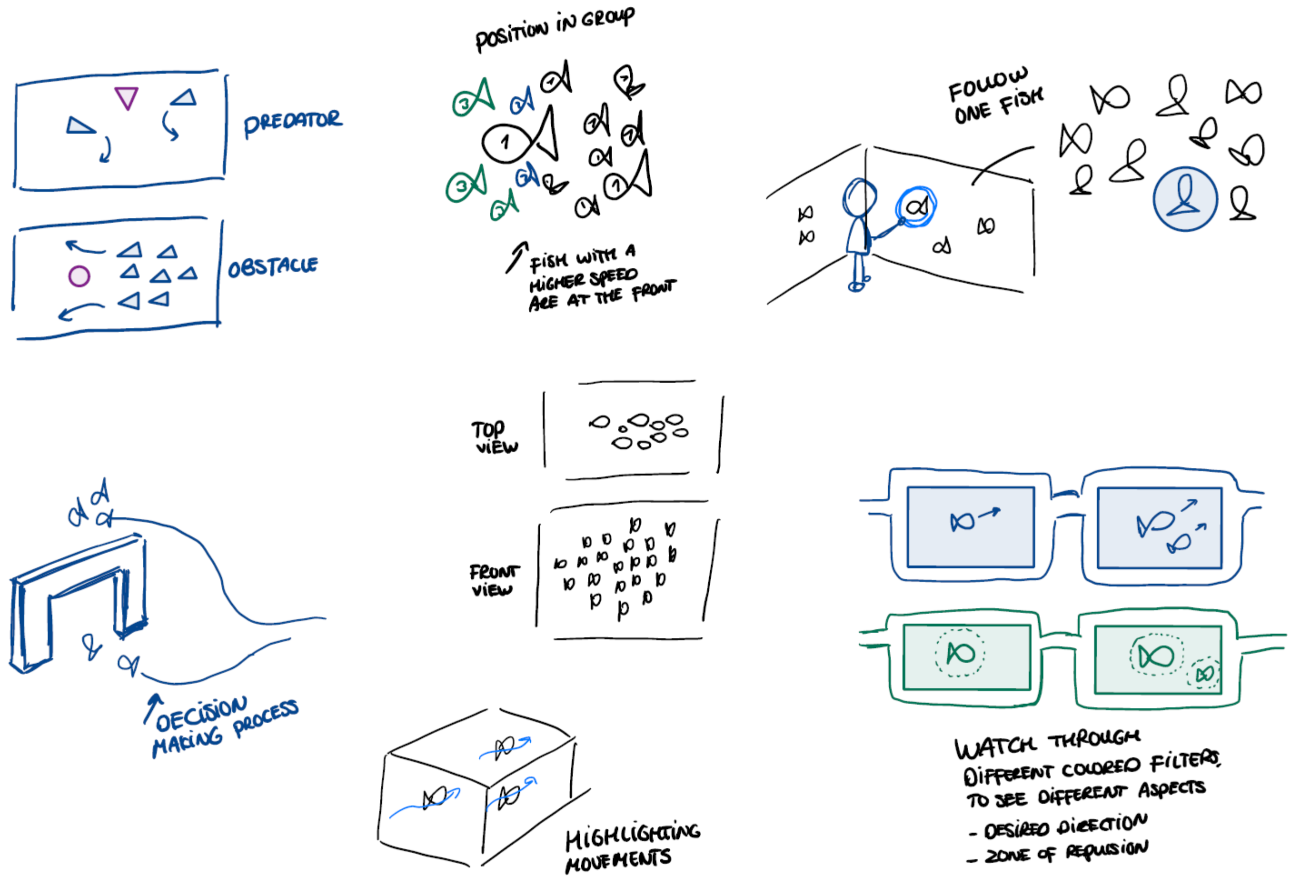


APPENDIX 3. DESIGN AND SHAPE EXPLORATIONS



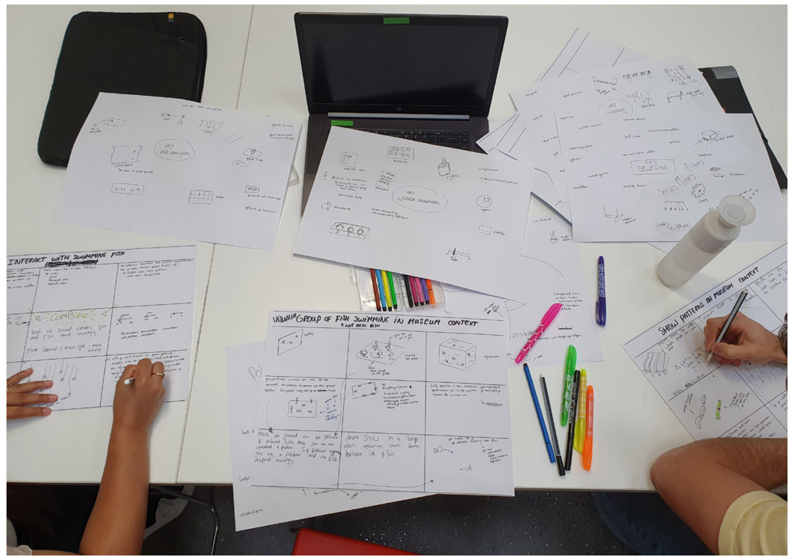



MOVE OBSTACLE BACK AND FORTH
MAKE CORALS BIGGER OR SMALLER
PLACE STONES
CHANGE PARAMETERS
ROTATE TO ADJUST
ADJUST THE GAP AND SEE WHEN FISH WILL SWIM THROUGH
IF YOU RUN FASTER, THE FISH WILL GO FASTER.
ADJUST SANDBANK HEIGHT
PREDATOR
OBSTACLE
DISPLAY REACTS TO INTERACTION
USE SOMETHING PHYSICAL ON THE WALL
POINT WITH LIGHTBEAM
YOU CAN SIT ON TOP THE FISH, TO SEE THEIR BEHAVIOUR
CUSTOMIZE YOUR FISH AND ADD TO THE AQUARIUM
MOVE TOGETHER WITHIN THE THREE ZONES
INTERACT ON BOTH SIDES.
LIDAR SENSOR
FISH WILL ROTATE DEPENDING ON WHERE YOU ARE STANDING
PHYSICAL HANGING FISH THAT CAN MOVE/ROTATE
WHERE YOU WALK, OBSTACLES ARE PLACED

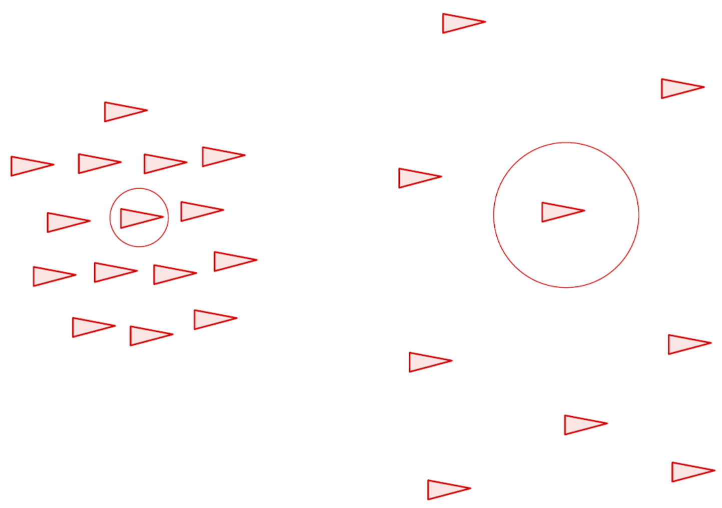

PREDATOR
OBSTACLE
POSITION IN GROUP
FISH WITH A HIGHER SPEED ARE AT THE FRONT
FOLLOW ONE FISH
DECISION MAKING PROCESS
TOP VIEW
FRONT VIEW
HIGHLIGHTING MOVEMENTS
WATCH THROUGH DIFFERENT COLORED FILTERS, TO SEE DIFFERENT ASPECTS
- DESIRED DIRECTION
- ZONE OF REPULSION

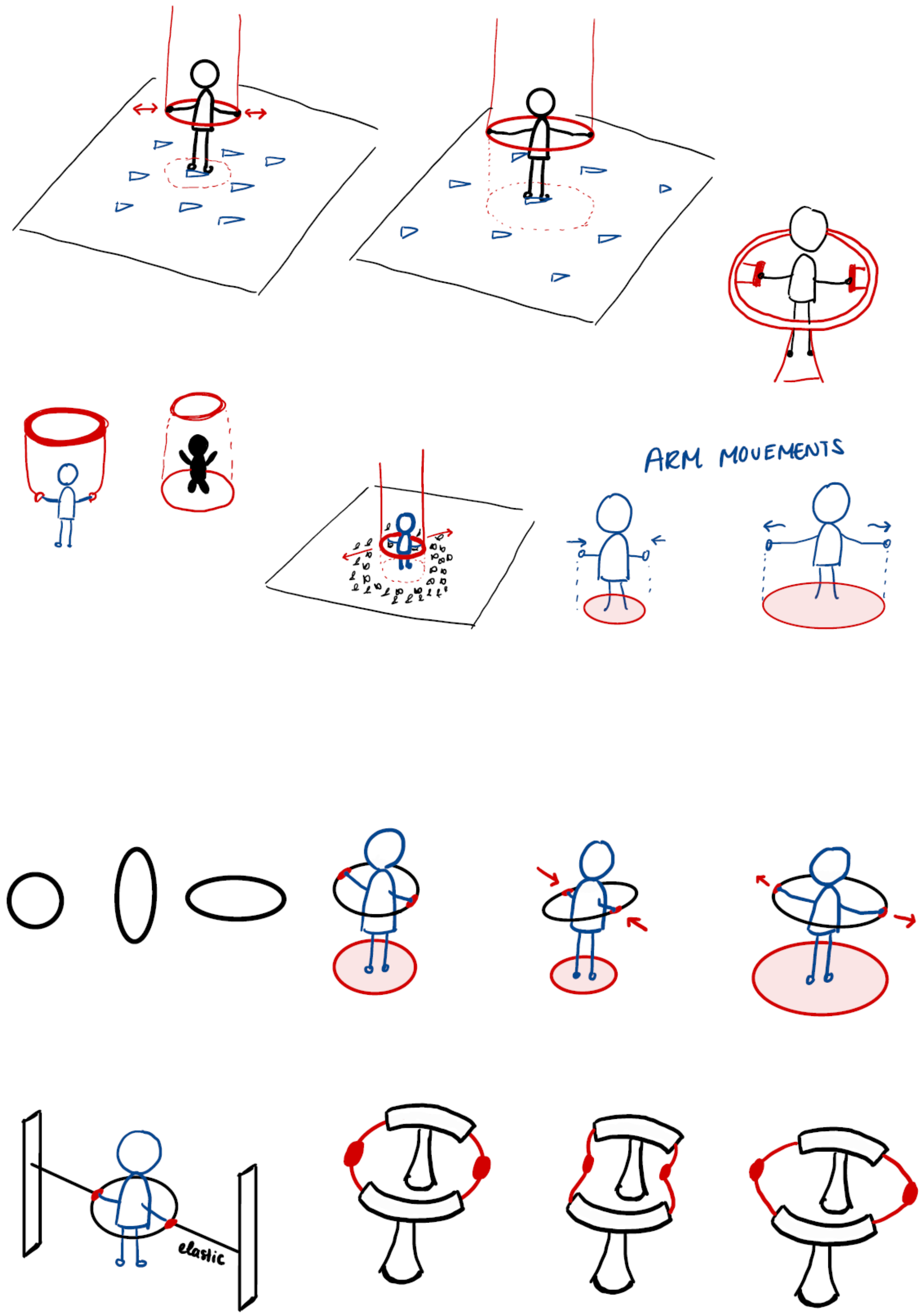
APPENDIX 4. IDEATION USER INTERACTION

Co-creation session

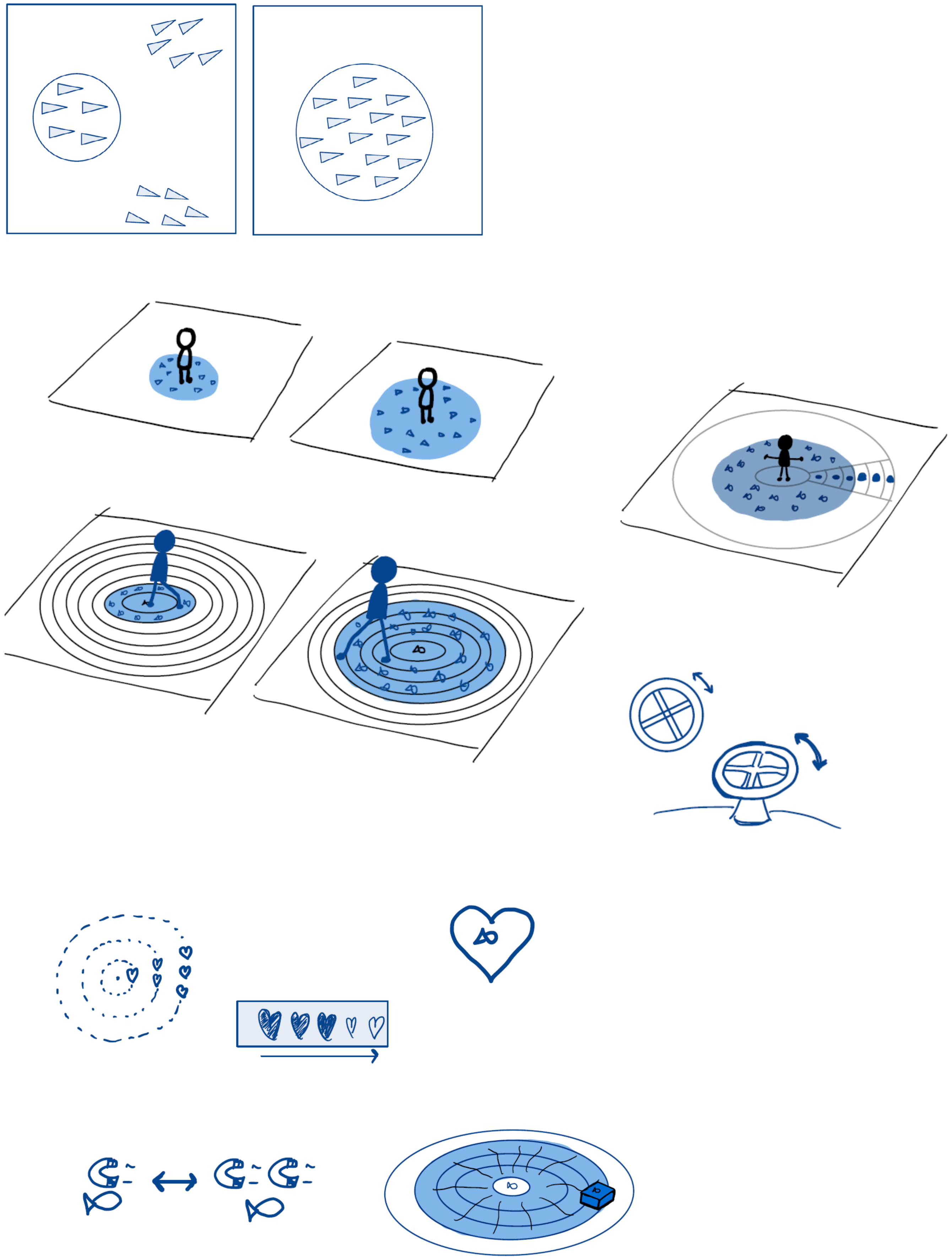


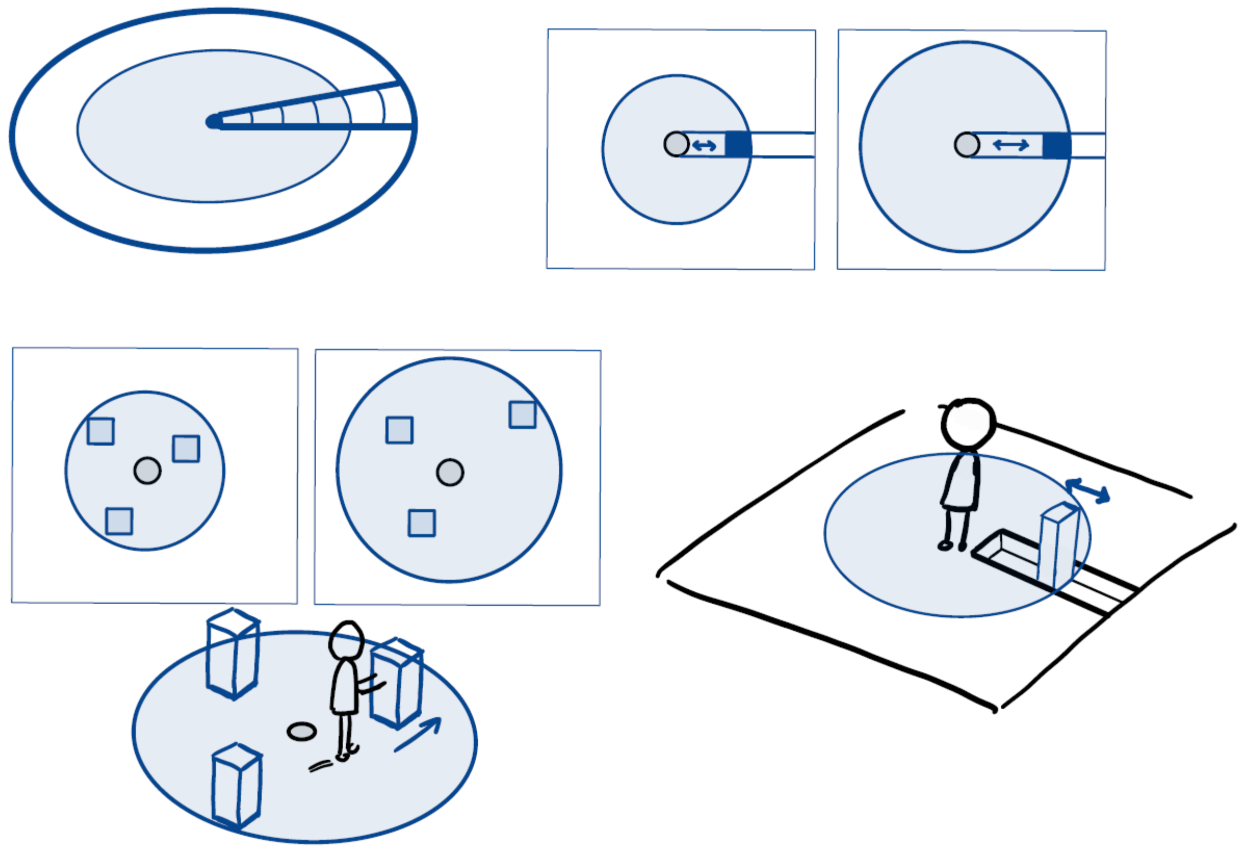
Manipulate size repulsion zone



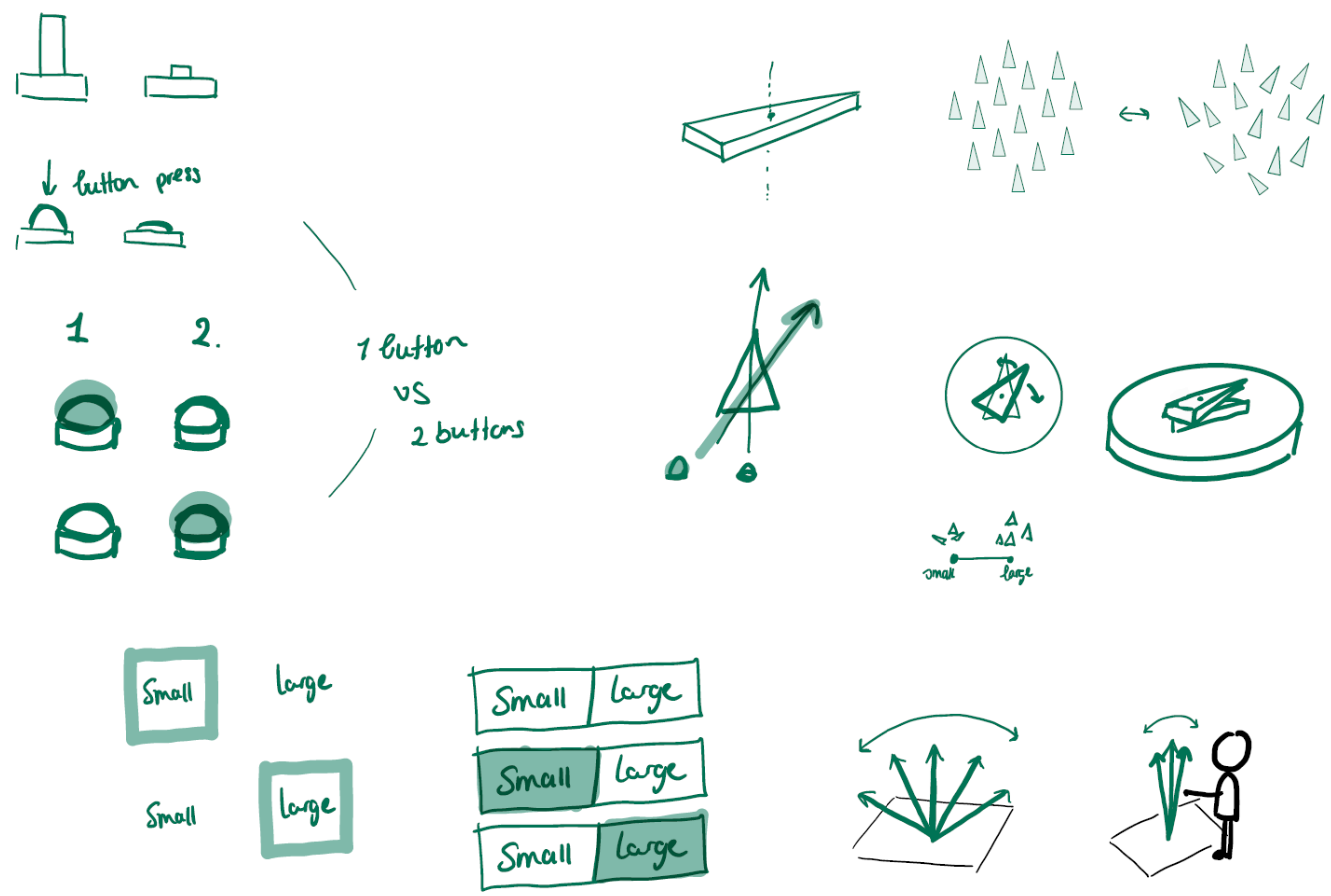


Manipulate size attraction zone





Manipulate small or large orientation zone



APPENDIX 5. SCRIPT - MOVING SIMULATION

```
// [Boids_sim.js]

//Input data from Arduino
let sensors0;
let sensors01;
let sensors012;
let sensors0_old = 0;
let sensors012_old = 0;

let range = 10;
let green = 0;
let red = 0;
let blue = 0;
let buttonvalue = 1;

let pX, pY;

//predator
let predatorPositionX;
let predatorPositionY;
let removeR = 0.015;

//restart
let restarteasy;
let restartdifficult;
let restart;

//Timers
let timer1 = 0;
let timer2 = 0;
let writetext = 1;
let waitforpredator = 20000;

//canvas range
let outsiderangelower = 0.1;
let outsiderangeupper = 0.9;

function setup() {

  //NOTE: Firebase is turned off, so personal key and ID information is not shared. To manually change the size of the z
ones, search for 'NO FIREBASE' within the code

  //Firebase info

  /*
  var firebaseConfig = {
    apiKey: "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
    authDomain: "xxxxx.firebaseapp.com",
    databaseURL:
      "https://xxxxxx.firebaseio.com",
    projectId: "xxxxxx",
    storageBucket: "cxxxxx.appspot.com",
    messagingSenderId: "123456789",
    appId: "x:xxxx:web:xxxxx",
    measurementId: "xxxxxx",
  };

  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  database = firebase.database();

  var ref1 = database.ref("circlevalues/uptodate/circlevalue1");
  ref1.on("value", gotData1, errData1);
  var ref2 = database.ref("circlevalues/uptodate/circlevalue2");
  ref2.on("value", gotData2, errData1);
  var ref3 = database.ref("circlevalues/uptodate/circlevalue3");
  ref3.on("value", gotData3, errData1);

  var ref4 = database.ref("circlevalues/uptodate/nrsvalue");
  ref4.on("value", gotData4, errData1);

  var ref5 = database.ref("circlevalues/uptodate/starteasy");
  ref5.on("value", gotData5, errData1);
```



```
  var ref6 = database.ref("circlevalues/uptodate/startdifficult");
  ref6.on("value", gotData6, errData1);

*/

  createCanvas(512, 512);
  background(255);

  pX = width / 2;
  pY = height / 2;
}

//get data from firebase
function gotData1(data) {
  //REPULSION DATA FROM FIREBASE
  sensors0 = data.val();

  if (sensors0 > 1000) {
    sensors0 = 1000;
  }
}

function gotData2(data) {
  //ORIENTATION DATA FROM FIREBASE
  buttonvalue = data.val();
}

function gotData3(data) {
  //ATTRACTION DATA FROM FIREBASE
  sensors012 = data.val();
  if (sensors012 > 1000) {
    sensors012 = 1000;
  }
}

function gotData4(data) {}

function gotData5(data) {
  //RESTART EASY DATA FROM FIREBASE
  restarteasy = data.val();

  if (restarteasy == 1) {
    //Set following new composition
    setBoidNumber(300);
    setPredatorNumber(0);
    restart = 1;
    writetext = 1;
    timer1 = millis();
    timer2 = millis();
  }
}

function gotData6(data) {
  restartdifficult = data.val();

  if (restartdifficult == 1) {
    //Set following new composition
    setBoidNumber(200);
    setPredatorNumber(2);
  }
}

function errData1(err) {
  console.log("Error!");
  console.log(err);
}

function draw() {
  setAvoidRange();
  setAlignRange();
  setAttractRange();
```

```
}

simulationInfo = {
  roost: { x: 0.5, y: 0.5, z: 0 },
  boids: [],
  predators: [],

  boardSizeX: 1920 * 3,
  boardSizeY: 1080 * 3,
  nBoids: 1000,
  nPredators: 0,
  birdSpeed: 0.001,
  predatorSpeed: 0.006,
  avoidanceRange: 0.0001,
  alignmentRange: 0.00109,
  attractionRange: 0.00123,
  avoidanceStrength: 0.3,
  alignmentStrength: 0.3,
  attractionStrength: 0.3,
  roostActive: true,
  roostRange: 0.1,
  roostStrength: 0.3,
  blindArc: 0.5,
  predatorAvoidance: 0.01,
  predatorAvoidanceStrength: 2,
  zoom: 40,
  handle: 0,
  tick: 0,
  show: {
    avoid: false,
    align: false,
    attract: false,
    flee: false,
    roost: true,
  },
};

//ADD NEW BOIDS
function getNewBoid() {
  var retVal = {
    position: { x: Math.random(), y: Math.random(), z: 0 },
    speed: [
      { x: Math.random() - 0.5, y: Math.random() - 0.5, z: 0 },
      { x: Math.random() - 0.5, y: Math.random() - 0.5, z: 0 },
    ],
    highlight: false,
    color: "rgb(255,255, 0)",
    highlight2: false,
  };
  normalize(retVal.speed[0]);
  normalize(retVal.speed[1]);

  return retVal;
}

function init() {
  var i;
  simulationInfo.board = document.createElement("canvas");
  simulationInfo.board.width = simulationInfo.boardSizeX;
  simulationInfo.board.height = simulationInfo.boardSizeY;
  addMouseTracker(document.getElementById("viewport"));
  for (i = 0; i < simulationInfo.nBoids; ++i) {
    simulationInfo.boids[i] = getNewBoid();
  }
  for (i = 0; i < simulationInfo.nPredators; ++i) {
    simulationInfo.predators[i] = {
      position: { x: Math.random(), y: Math.random(), z: 0 },
      speed: { x: Math.random() - 0.5, y: Math.random() - 0.5, z: 0 },
    };
    normalize(simulationInfo.predators[i].speed);
  }
  simulationInfo.boids[0].highlight = true;
  simulationInfo.boids[1].highlight2 = true;
```



```
  toggleAvoid(document.getElementById("showAvoid").checked);
  toggleAlign(document.getElementById("showAlign").checked);
  toggleAttract(document.getElementById("showAttract").checked);
  toggleFleeing(document.getElementById("showFleeing").checked);
  toggleRoost(document.getElementById("showRoost").checked);
  setAvoidStrength(document.getElementById("avoidStrength").value);
  setAlignStrength(document.getElementById("alignStrength").value);
  setAttractStrength(document.getElementById("attractStrength").value);
  setFleeStrength(document.getElementById("fleeStrength").value);
  setFleeRange(document.getElementById("fleeingRange").value);
  setBoidNumber(document.getElementById("boidDial").value);
  setPredatorNumber(document.getElementById("predatorDial").value);
  setRoostSize(document.getElementById("roostDial").value);
  setBoidSpeed(document.getElementById("speedDial").value);
  setPredatorSpeed(document.getElementById("predatorSpeedDial").value);
  setZoom(document.getElementById("zoomDial").value);
  step();
}

function mousePressed() {
  //WHEN MOUSE IS PRESSED IN CANVAS --> FULL SCREEN
  if (
    mouseX > 0 &&
    mouseX < windowWidth &&
    mouseY + 1350 > 0 &&
    mouseY + 1350 < windowHeight
  ) {
    let fs = fullscreen();
    fullscreen(!fs);
  }
}

function windowResized() {}

function normalize(vec) {
  var len = Math.pow(vec.x, 2) + Math.pow(vec.y, 2) + Math.pow(vec.z, 2);
  if (len > 0) {
    len = Math.sqrt(len);
    vec.x /= len;
    vec.y /= len;
    vec.z /= len;
  }
  return vec;
}

function getLength(vec) {
  return Math.sqrt(vec.x * vec.x + vec.y * vec.y + vec.z * vec.z);
}

function getDistance(firstVec, secondVec) {
  var dist =
    Math.pow(firstVec.x - secondVec.x, 2) +
    Math.pow(firstVec.y - secondVec.y, 2) +
    Math.pow(firstVec.z - secondVec.z, 2);
  return dist;
}

function multiplyVector(vec, scalar) {
  return { x: vec.x * scalar, y: vec.y * scalar, z: vec.z * scalar };
}

function addVector(firstVec, secondVec) {
  return {
    x: firstVec.x + secondVec.x,
    y: firstVec.y + secondVec.y,
    z: firstVec.z + secondVec.z,
  };
}

function getAngle(firstVec, secondVec) {
  return Math.acos(
    (firstVec.x * secondVec.x +
```

```
      firstVec.y * secondVec.y +
      firstVec.z * secondVec.z) /
      (getLength(firstVec) * getLength(secondVec))
  );
}

function updateAvoidAlignAttract(boidActor, boidObject, avoid, align, attract) {
  var dist = getDistance(boidActor.position, boidObject.position);
  if (dist < simulationInfo.avoidanceRange) {
    avoid.x +=
      (boidActor.position.x - boidObject.position.x) *
      Math.pow(
        (simulationInfo.avoidanceRange - dist) / simulationInfo.avoidanceRange,
        2
      );
    avoid.y +=
      (boidActor.position.y - boidObject.position.y) *
      Math.pow(
        (simulationInfo.avoidanceRange - dist) / simulationInfo.avoidanceRange,
        2
      );
    avoid.z +=
      (boidActor.position.z - boidObject.position.z) *
      Math.pow(
        (simulationInfo.avoidanceRange - dist) / simulationInfo.avoidanceRange,
        2
      );
  }
  if (dist < simulationInfo.alignmentRange) {
    align.x += boidObject.speed[simulationInfo.tick].x;
    align.y += boidObject.speed[simulationInfo.tick].y;
    align.z += boidObject.speed[simulationInfo.tick].z;
  }
  if (dist < simulationInfo.attractionRange) {
    attract.x += boidObject.position.x - boidActor.position.x;
    attract.y += boidObject.position.y - boidActor.position.y;
    attract.z += boidObject.position.z - boidActor.position.z;
  }
}

function recalculateSpeeds() {
  var i,
    j,
    avoid,
    align,
    attract,
    maxRange,
    angle,
    minPoint,
    countEffectors,
    skippedEffectors;
  minPoint = 0;
  maxRange = Math.max(
    Math.sqrt(simulationInfo.attractionRange),
    Math.sqrt(simulationInfo.alignmentRange),
    Math.sqrt(simulationInfo.avoidanceRange)
  );
  for (i = 0; i < simulationInfo.boids.length; ++i) {
    avoid = { x: 0, y: 0, z: 0 };
    align = { x: 0, y: 0, z: 0 };
    attract = { x: 0, y: 0, z: 0 };
    for (j = minPoint; j < i; ++j) {
      if (
        simulationInfo.boids[i].position.x -
          simulationInfo.boids[j].position.x >
        maxRange
      ) {
        minPoint++;
        continue;
      }
      if (
        Math.pow(
          simulationInfo.boids[i].position.y -
            simulationInfo.boids[j].position.y,
          2
        ) > simulationInfo.attractionRange
      )
        continue;
      updateAvoidAlignAttract(
        simulationInfo.boids[i],
        simulationInfo.boids[j],
        avoid,
        align,
        attract
      );
    }
    for (j = i + 1; j < simulationInfo.boids.length; ++j) {
      if (
        simulationInfo.boids[j].position.x -
          simulationInfo.boids[i].position.x >
        maxRange
      )
        break;
      if (
        Math.pow(
          simulationInfo.boids[i].position.y -
            simulationInfo.boids[j].position.y,
          2
        ) > simulationInfo.attractionRange
      )
        continue;
      updateAvoidAlignAttract(
        simulationInfo.boids[i],
        simulationInfo.boids[j],
        avoid,
        align,
        attract
      );
    }
    if (avoid.x != 0 || avoid.y != 0 || avoid.z != 0) {
      avoid = multiplyVector(
        normalize(avoid),
        simulationInfo.avoidanceStrength
      );
    } else {
      avoid = addVector(
        multiplyVector(normalize(align), simulationInfo.alignmentStrength),
        multiplyVector(normalize(attract), simulationInfo.attractionStrength)
      );
    }
    simulationInfo.boids[i].speed[1 - simulationInfo.tick] = {
      x: simulationInfo.boids[i].speed[simulationInfo.tick].x,
      y: simulationInfo.boids[i].speed[simulationInfo.tick].y,
      z: simulationInfo.boids[i].speed[simulationInfo.tick].z,
    };

    if (
      simulationInfo.boids[i].position.x < outsiderangelower ||
      simulationInfo.boids[i].position.x > outsiderangeupper ||
      simulationInfo.boids[i].position.y < outsiderangelower ||
      simulationInfo.boids[i].position.y > outsiderangeupper
    ) {
      simulationInfo.boids[i].speed[1 - simulationInfo.tick] = normalize(
        addVector(
          simulationInfo.boids[i].speed[1 - simulationInfo.tick],
          multiplyVector(
            normalize({
              x: 0.5 - simulationInfo.boids[i].position.x,
              y: 0.5 - simulationInfo.boids[i].position.y,
              z: -simulationInfo.boids[i].position.z,
            }),
            simulationInfo.roostStrength
          )
        )
      );
    }
```

```
    simulationInfo.boids[i].speed[1 - simulationInfo.tick] = normalize(
      addVector(simulationInfo.boids[i].speed[1 - simulationInfo.tick], avoid)
    );
    avoid = getPredatorAvoidance(simulationInfo.boids[i]);
    if (avoid.x != 0 || avoid.y != 0 || avoid.z != 0) {
      simulationInfo.boids[i].speed[1 - simulationInfo.tick] = normalize(
        addVector(
          simulationInfo.boids[i].speed[1 - simulationInfo.tick],
          multiplyVector(
            normalize(avoid),
            simulationInfo.predatorAvoidanceStrength
          )
        )
      );
      simulationInfo.boids[i].highlight = true; //make orange,  when it sees predator
    } else {
      simulationInfo.boids[i].highlight = false;
    }

    if (
      document.getElementById("viewport").mouse.hasFocus &&
      getDistance(simulationInfo.boids[i].position, getMousePosition()) <
        simulationInfo.predatorAvoidance
    ) {
      simulationInfo.boids[i].speed[1 - simulationInfo.tick] = multiplyVector(
        normalize(
          addVector(
            simulationInfo.boids[i].speed[1 - simulationInfo.tick],
            multiplyVector(
              normalize(getMouseAvoidance(simulationInfo.boids[i])),
              simulationInfo.predatorAvoidanceStrength
            )
          )
        ),
        2
      );
    }
  }
  reorientPredators();
  simulationInfo.tick = 1 - simulationInfo.tick;
}

function step() {
  var i,
    j,
    avoid,
    align,
    attract,
    maxRange,
    angle,
    minPoint,
    countEffectors,
    skippedEffectors;
  recalculateSpeeds();
  for (i = 0; i < simulationInfo.predators.length; ++i) {
    simulationInfo.predators[i].position.x =
      (simulationInfo.predators[i].position.x +
        1 +
        simulationInfo.predatorSpeed * simulationInfo.predators[i].speed.x) %
      1;
    simulationInfo.predators[i].position.y =
      (simulationInfo.predators[i].position.y +
        1 +
        simulationInfo.predatorSpeed * simulationInfo.predators[i].speed.y) %
      1;

    simulationInfo.predators[i].position.z =
      simulationInfo.predators[i].position.z +
      simulationInfo.predatorSpeed * simulationInfo.predators[i].speed.z;
  }

  for (i = 0; i < simulationInfo.boids.length; ++i) {
    simulationInfo.boids[i].position.x =
```



```
      (simulationInfo.boids[i].position.x +
        1 +
        simulationInfo.birdSpeed *
          simulationInfo.boids[i].speed[simulationInfo.tick].x) %
      1;
    simulationInfo.boids[i].position.y =
      (simulationInfo.boids[i].position.y +
        1 +
        simulationInfo.birdSpeed *
          simulationInfo.boids[i].speed[simulationInfo.tick].y) %
      1;
    simulationInfo.boids[i].position.z =
      simulationInfo.boids[i].position.z +
      simulationInfo.birdSpeed *
        simulationInfo.boids[i].speed[simulationInfo.tick].z;
  }
  requestRepaint();
  simulationInfo.boids.sort(function (a, b) {
    return a.position.x - b.position.x;
  });
  setTimeout("step();", 20);
}

function reorientPredators() {
  var i, j, minDist, curDist, targetBoid;
  for (i = 0; i < simulationInfo.predators.length; ++i) {
    minDist = getDistance(
      simulationInfo.predators[i].position,
      simulationInfo.boids[0].position
    );
//make predator confused when seeing many agents closeby
    var minuss = 0.25 - red * 0.01;
    if (minuss < 0.002) {
      minuss = 0.002;
    }
    if (buttonvalue == 0) {
      minuss = 0.002;
    }

    targetBoid = 0;

    for (j = 1; j < simulationInfo.boids.length; ++j) {
      curDist = getDistance(
        simulationInfo.predators[i].position,
        simulationInfo.boids[j].position
      );

      if (curDist < minDist + minuss) {
        minDist = curDist;

        targetBoid = j;
      }
    }

    simulationInfo.predators[i].speed = normalize(
      addVector(
        simulationInfo.predators[i].speed,
        multiplyVector(
          normalize({
            x:
              simulationInfo.boids[targetBoid].position.x -
              simulationInfo.predators[i].position.x,
            y:
              simulationInfo.boids[targetBoid].position.y -
              simulationInfo.predators[i].position.y,
            z:
              simulationInfo.boids[targetBoid].position.z -
              simulationInfo.predators[i].position.z,
          }),
          0.2
        )
      )
    );
```

```
    }
}

function getPredatorAvoidance(boidActor) {
  var j, avoid, dist;
  avoid = { x: 0, y: 0, z: 0 };
  for (j = 0; j < simulationInfo.predators.length; ++j) {
    dist = getDistance(
      boidActor.position,
      simulationInfo.predators[j].position
    );
    if (dist < simulationInfo.predatorAvoidance) {
      avoid.x +=
        (boidActor.position.x - simulationInfo.predators[j].position.x) *
        Math.pow(
          (simulationInfo.predatorAvoidance - dist) /
            simulationInfo.predatorAvoidance,
          2
        );
      avoid.y +=
        (boidActor.position.y - simulationInfo.predators[j].position.y) *
        Math.pow(
          (simulationInfo.predatorAvoidance - dist) /
            simulationInfo.predatorAvoidance,
          2
        );
      avoid.z +=
        (boidActor.position.z - simulationInfo.predators[j].position.z) *
        Math.pow(
          (simulationInfo.predatorAvoidance - dist) /
            simulationInfo.predatorAvoidance,
          2
        );
    }
  }
  return avoid;
}

function getMouseAvoidance(boidActor) {
  var mousePos = getMousePosition();
  return {
    x: boidActor.position.x - mousePos.x,
    y: boidActor.position.y - mousePos.y,
    z: 0,
  };
}

function requestRepaint() {
  if (simulationInfo.handle == 0) {
    simulationInfo.handle = window.requestAnimationFrame(repaint);
  }
}

function getMousePosition() {
  return {
    x:
      document.getElementById("viewport").mouse.x /
        (document.getElementById("viewport").width * simulationInfo.zoom) +
      (1 - 1 / simulationInfo.zoom) / 2,
    y:
      document.getElementById("viewport").mouse.y /
        (document.getElementById("viewport").height * simulationInfo.zoom) +
      (1 - 1 / simulationInfo.zoom) / 2,
    z: 0,
  };
}

function repaint() {
  var i, ctx;
  ctx = simulationInfo.board.getContext("2d");

  //BACKGROUND COLOUR BLUE
  ctx.fillStyle = "rgb(40,40,100)";
```



```
  ctx.fillRect(0, 0, simulationInfo.boardSizeX, simulationInfo.boardSizeY);

  //SHOW TEXT WHEN RESTART BUTTON IS PRESSED
  if (restart == 1) {
    ctx.font = "300px Fantasy";
    ctx.fillStyle = "rgb(255,255,255)";
    ctx.fill();
    ctx.fillText(
      "Begin",
      simulationInfo.boardSizeX * 0.45,
      simulationInfo.boardSizeY * 0.85
    );
    ctx.restore();
  }

  //REMOVE RESTART TEXT WHEN TIME HAS PASSED
  if (millis() > timer1 + 5000) {
    restart = 0;

  }

  //PREDATOR COMES IN AFTER CERTAIN AMOUNT OF TIME (waitforpredator)
  //WRITE TEXT
  if (millis() > timer2 + waitforpredator) {
    setPredatorNumber(1);

    if (writetext == 1) {
      ctx.font = "200px Fantasy";
      ctx.fillStyle = "rgb(200,100,0)";

      ctx.fillText(
        "Kijk uit voor het roofdier!",
        simulationInfo.boardSizeX * 0.35,
        simulationInfo.boardSizeY * 0.85
      );
      ctx.restore();
    }
  }

  //REMOVE PREDATOR TEXT AFTER TIME HAS PASSED
  if (millis() > timer2 + waitforpredator + 10000) {
    writetext = 0;
  }

  for (i = 0; i < simulationInfo.predators.length; ++i) {
    ctx.save();
    ctx.translate(
      simulationInfo.predators[i].position.x * simulationInfo.boardSizeX,
      simulationInfo.predators[i].position.y * simulationInfo.boardSizeY
    );
    if (simulationInfo.predators[i].speed.y < 0) {
      ctx.rotate(2 * Math.PI - Math.acos(simulationInfo.predators[i].speed.x));
    } else {
      ctx.rotate(Math.acos(simulationInfo.predators[i].speed.x));
    }

    //PREDATOR SHAPE AND COLOUR
    ctx.beginPath();
    ctx.moveTo(7, 12);
    ctx.lineTo(-7, 35);
    ctx.lineTo(45, 0);
    ctx.lineTo(-7, -35);
    ctx.lineTo(7, -12);
    ctx.lineTo(-5, 0);
    ctx.closePath();
    ctx.fillStyle = "rgb(200,100,0)";
    ctx.fill();
    ctx.restore();
  }

  // BOIDS LOOP
  for (i = 0; i < simulationInfo.boids.length; ++i) {
    // loop through all boids
```

```
    ctx.save();
    ctx.translate(
      simulationInfo.boids[i].position.x * simulationInfo.boardSizeX,
      simulationInfo.boids[i].position.y * simulationInfo.boardSizeY
    );
    if (simulationInfo.boids[i].speed[simulationInfo.tick].y < 0) {
      ctx.rotate(
        2 * Math.PI -
          Math.acos(simulationInfo.boids[i].speed[simulationInfo.tick].x)
      );
    } else {
      ctx.rotate(
        Math.acos(simulationInfo.boids[i].speed[simulationInfo.tick].x)
      );
    }

    //If boid sees predator, make ORANGE
    if (simulationInfo.boids[i].highlight == true) {
      simulationInfo.boids[i].color = "rgb(220,180, 0)"; //orange
      ctx.lineWidth = 4;
    } else {
      //if boid does not see predator, make YELLOW again
      simulationInfo.boids[i].color = "rgb(255,255, 0)";
    }

    ctx.beginPath();
    ctx.moveTo(0, 0);
    ctx.lineTo(-3, 5);
    ctx.lineTo(10, 0);
    ctx.lineTo(-3, -5);
    ctx.closePath();
    ctx.fillStyle = simulationInfo.boids[i].color;
    ctx.fill();

    ctx.restore();

    //DELETE BOIDS THAT ARE TOUCHED BY PREDATOR

    //PREDATOR REMOVES
    for (k = 0; k < simulationInfo.predators.length; ++k) {
      predatorPositionX = simulationInfo.predators[k].position.x;
      predatorPositionY = simulationInfo.predators[k].position.y;

      if (
        simulationInfo.boids[i].position.x > predatorPositionX - removeR &&
        simulationInfo.boids[i].position.x < predatorPositionX + removeR &&
        simulationInfo.boids[i].position.y > predatorPositionY - removeR &&
        simulationInfo.boids[i].position.y < predatorPositionY + removeR
      ) {
        if (
          simulationInfo.boids.length > 1 &&
          simulationInfo.predators.length >= 1
        ) {
          simulationInfo.boids.splice(i, 1);
          document.getElementById("boidLabel").innerHTML =
            document.getElementById("boidLabel").innerHTML - 1;
        }
      }
    }
  }

  ctx = document.getElementById("viewport").getContext("2d");
  ctx.clearRect(
    0,
    0,
    document.getElementById("viewport").width,
    document.getElementById("viewport").height
  );
  ctx.drawImage(
    simulationInfo.board,
    (simulationInfo.boardSizeX * (1 - 1 / simulationInfo.zoom)) / 2,
    (simulationInfo.boardSizeY * (1 - 1 / simulationInfo.zoom)) / 2,
    simulationInfo.boardSizeX / simulationInfo.zoom,
```



```
    simulationInfo.boardSizeY / simulationInfo.zoom,
    0,
    0,
    document.getElementById("viewport").width,
    document.getElementById("viewport").height
  );
  simulationInfo.handle = 0;
}

function toggleFleeing(newVal) {
  simulationInfo.show.flee = newVal;
}

function toggleRoost(newVal) {
  simulationInfo.show.roost = newVal;
}

function toggleAvoid(newVal) {
  simulationInfo.show.avoid = newVal;
}

function toggleAlign(newVal) {
  simulationInfo.show.align = newVal;
}

function toggleAttract(newVal) {
  simulationInfo.show.attract = newVal;
}

function setFleeRange(newVal) {
  simulationInfo.predatorAvoidance = Math.pow(newVal / 1000, 2);
  document.getElementById("fleeingRange").value = newVal;
}

function setAttractRange(newVal) {
  //WHEN ATTRACTION HAS CHANGED (MORE THAN CERTAIN RANGE TO REMOVE NOISY VALUES)
  if (
    sensors012 > sensors012_old + range ||
    sensors012 < sensors012_old - range
  ) {
    green = map(sensors012, 440, 60, 13, 45);   //MAP TIME OF FLIGHT INPUT TO ZONE VALUE

    sensors012_old = sensors012;
  }


  //IF FIREBASE IS TURNED OFF
  //NO FIREBASE
  green=35; //changable between 1-50
  //

  newVal = green;

  simulationInfo.attractionRange = Math.pow(newVal / 1000, 2);
  document.getElementById("attractRange").value = newVal;
}

function setAlignRange(newVal) {

  //IF FIREBASE IS TURNED OFF
  //NO FIREBASE
  buttonvalue=1; //changable between 0-1
  //

  blue = map(buttonvalue, 0, 1, red, green); //MAP ORIENTATION INPUT TO ZONE VALUE (SMALL OR BIG)
  newVal = blue;

  simulationInfo.alignmentRange = Math.pow(newVal / 1000, 2);
  document.getElementById("alignRange").value = newVal;
}

function setAvoidRange(newVal) {
   //WHEN REPULSION HAS CHANGED (MORE THAN CERTAIN RANGE TO REMOVE NOISY VALUES)
```

```
  if (sensors0 > sensors0_old + range || sensors0 < sensors0_old - range) {
    red = map(sensors0, 104, 310, 8, 44); //MAP TIME OF FLIGHT INPUT TO ZONE VALUE
    sensors0_old = sensors0;
  }

  //IF FIREBASE IS TURNED OFF
  //NO FIREBASE
  red=10; //changable between 1-50
  //

  newVal = red;
  simulationInfo.avoidanceRange = Math.pow(newVal / 1000, 2);
  document.getElementById("avoidRange").value = newVal;

}

function setFleeStrength(newVal) {
  simulationInfo.predatorAvoidanceStrength = Math.pow(newVal / 100, 2);
  document.getElementById("fleeStrength").value = newVal;
}

function setAttractStrength(newVal) {
  simulationInfo.attractionStrength = Math.pow(newVal / 100, 2);
  document.getElementById("attractStrength").value = newVal;
}

function setAlignStrength(newVal) {
  simulationInfo.alignStrength = Math.pow(newVal / 100, 2);
  document.getElementById("alignStrength").value = newVal;
}

function setAvoidStrength(newVal) {
  simulationInfo.avoidanceStrength = Math.pow(newVal / 100, 2);
  document.getElementById("avoidStrength").value = newVal;
}

function setRoostStrength(newVal) {
  simulationInfo.roostStrength = Math.pow(newVal / 100, 2);
  document.getElementById("roostStrength").value = newVal;
}

function setBoidSpeed(newVal) {
  simulationInfo.birdSpeed = newVal / 10000;
}

function setPredatorSpeed(newVal) {
  simulationInfo.predatorSpeed = newVal / 10000;
}

function setZoom(newVal) {
  simulationInfo.zoom = newVal / 10;
}

function setRoostSize(newVal) {
  simulationInfo.roostRange = Math.pow(newVal / 100, 2);
  simulationInfo.roostActive = simulationInfo.roostRange > 0;
}

function setBoidNumber(newVal) {
  var i, highlightFound;
  simulationInfo.nBoids = 200;

  while (simulationInfo.boids.length > newVal) {
    //delete 1 at a time
    simulationInfo.boids.splice(
      Math.floor(Math.random() * simulationInfo.boids.length),
      1
    );
  }
  while (simulationInfo.boids.length < newVal) {
    i = simulationInfo.boids.length;

    simulationInfo.boids[i] = getNewBoid();
```



```
  }
  highlightFound = false;

  if (!highlightFound) {
    simulationInfo.boids[0].highlight = true;

    // HERE  AMOUNT OF BOIDS WITH VISUALS (percentage)
    for (i = 2; i < 0.0 * simulationInfo.boids.length; ++i) {
      simulationInfo.boids[i].highlight = true;
    }
  }
  document.getElementById("boidLabel").innerHTML = newVal;
}

function setPredatorNumber(newVal) {
  var i, highlightFound;
  simulationInfo.nPredators = newVal;
  if (simulationInfo.predators.length > newVal) {
    simulationInfo.predators.length = newVal;
  }
  while (simulationInfo.predators.length < newVal) {
    i = simulationInfo.predators.length;
    simulationInfo.predators[i] = {
      position: { x: Math.random(), y: Math.random(), z: 0 },
      speed: { x: Math.random() - 0.5, y: Math.random() - 0.5, z: 0 },
    };
    normalize(simulationInfo.predators[i].speed);
  }
  document.getElementById("predatorLabel").innerHTML = newVal;
}
```

```
// [index.html]

<html><head>
<title>Boids2</title>

  <script src="firebase.js"></script>



  <script src="p5.js"></script>
    <script src="p5.sound.min.js"></script>
    <script language="javascript" type="text/javascript" src="p5.serialport.js"></script>

<!--    <script src="serial.js"></script> -->
  <script language="JavaScript" src="mouse.js"></script>
<script language="JavaScript" src="boids_sim.js"></script>



   <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8">

   <script src="sketch.js"></script>

</head>
<body onload="init();">

<canvas id="viewport" width="1600" height="900" style="border:1px solid black;"></canvas><table border="0">


<tbody><tr><td>Zoom: </td><td align="right"><input type="range" id="zoomDial" min="10" max="40" value="12" oninput="setZ
oom(this.value);"></td></tr>

<tr><th></th><th>Number of agents</th><th>Movement speed</th></tr>

<tr><td>Boids: <span id="boidLabel"></span></td><td align="right"><input type="range" id="boidDial" min="2" max="2500" v
alue="300" oninput="setBoidNumber(this.value);"></td><td align="right"><input type="range" id="speedDial" min="10" max="
100" value="34" oninput="setBoidSpeed(this.value);"></td><td></td></tr>

<tr><td>Predators: <span id="predatorLabel"></span></td><td align="right"><input type="range" id="predatorDial" min="0"
max="5" value="0" oninput="setPredatorNumber(this.value);"></td><td align="right"><input type="range" id="predatorSpeedD
ial" min="10" max="100" value="44" oninput="setPredatorSpeed(this.value);"></td><td></td></tr>

<tr><th>Action</th><th>Distance</th><th>Strength</th><th>Show</th></tr>
```

```
<tr><td>Avoid collisions</td><td align="right"><input type="range" id="avoidRange" min="5" max="250" value="7" oninput="
setAvoidRange(this.value);"></td><td align="right"><input type="range" id="avoidStrength" min="0" max="150" value="35" o
ninput="setAvoidStrength(this.value);"></td><td align="center"><input type="checkbox" id="showAvoid" onclick="toggleAvoi
d(this.checked);"></td></tr>

<tr><td>Align with flockmates</td><td align="right"><input type="range" id="alignRange" min="5" max="250" value="28" oni
nput="setAlignRange(this.value);"></td><td align="right"><input type="range" id="alignStrength" min="0" max="150" value=
"36" oninput="setAlignStrength(this.value);"></td><td align="center"><input type="checkbox" id="showAlign" onclick="togg
leAlign(this.checked);"></td></tr>

<tr><td>Approach flockmates</td><td align="right"><input type="range" id="attractRange" min="5" max="250" value="71" oni
nput="setAttractRange(this.value);"></td><td align="right"><input type="range" id="attractStrength" min="0" max="150" va
lue="46" oninput="setAttractStrength(this.value);"></td><td align="center"><input type="checkbox" id="showAttract" oncli
ck="toggleAttract(this.checked);"></td></tr>

<tr><td>Predator avoidance</td><td align="right"><input type="range" id="fleeingRange" min="5" max="250" value="80" onin
put="setFleeRange(this.value);"></td><td align="right"><input type="range" id="fleeStrength" min="0" max="150" value="80
" oninput="setFleeStrength(this.value);"></td><td align="center"><input type="checkbox" id="showFleeing" onclick="toggle
Fleeing(this.checked);"></td></tr>

<tr><td>Stay close to the center</td><td align="right"><input type="range" id="roostDial" min="0" max="50" value="0" oni
nput="setRoostSize(this.value);"></td><td align="right"><input type="range" id="roostStrength" min="0" max="150" value="
140" oninput="setRoostStrength(this.value);"></td><td align="center"><input type="checkbox" id="showRoost" onclick="togg
leRoost(this.checked);"></td></tr>
</tbody></table>

</body></html>
```

APPENDIX 6. SCRIPT - FLOOR VISUALISATION

```
// [boids_sim.js]

//Input data from Arduino
let sensors0;
let sensors01;
let sensors012;
let sensors0_old = 0;
let sensors012_old = 0;

let range = 10;
let green;
let red;
let blue;
let buttonvalue = 1;

let orientation;
let attractcircle;
let avoidradius;
let centerboard;

let pX, pY;
let currentboidnr;

//raster
let xrow;
let yrow;
let yy = [];
let xx = [];
var positions = [];
let xcoordinate;
let ycoordinate;
let distancetomiddle;

function setup() {

  //NOTE: Firebase is turned off, so personal key and ID information is not shared. To manually change the size of the z
ones, search for 'NO FIREBASE' within the code

  //Firebase info
  /*
  var firebaseConfig = {
    apiKey: "xxxxxxxxxxxxxxxxxxxxxxxxxxx",
    authDomain: "xxxxx.firebaseapp.com",
    databaseURL:
      "https://xxxxxx.firebaseio.com",
    projectId: "xxxxxx",
    storageBucket: "cxxxxx.appspot.com",
    messagingSenderId: "123456789",
    appId: "x:xxxx:web:xxxxx",
    measurementId: "xxxxxx",
  };

  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);

  var database = firebase.database();

  var ref1 = database.ref("circlevalues/uptodate/circlevalue1");
  ref1.on("value", gotData1, errData1);
  var ref2 = database.ref("circlevalues/uptodate/circlevalue2");
  ref2.on("value", gotData2, errData1);
  var ref3 = database.ref("circlevalues/uptodate/circlevalue3");
  ref3.on("value", gotData3, errData1);

  var ref4 = database.ref("circlevalues/uptodate/nrsvalue");
  ref4.on("value", gotData4, errData1);

 */


  createCanvas(windowWidth, windowHeight);
  background(255);
```



```
  pX = width / 2;
  pY = height / 2;
}

//get data from firebase
function gotData1(data) {
  //REPULSION DATA FROM FIREBASE
  sensors0 = data.val();

  if (sensors0 > 1000) {
    sensors0 = 1000;
  }
}

function gotData2(data) {
  //ORIENTATION DATA FROM FIREBASE
  buttonvalue = data.val();
}

function gotData3(data) {
  //ATTRACTION DATA FROM FIREBASE
  sensors012 = data.val();

  if (sensors012 > 1000) {
    sensors012 = 1000;
  }

}

function gotData4(data) {}

function errData1(err) {
  console.log("Error!");
  console.log(err);
}



simulationInfo = {
  roost: { x: 0.5, y: 0.5, z: 0 },
  boids: [],
  predators: [],

  boardSize: 1920 * 3,
  nBoids: 200,
  nPredators: 0,
  birdSpeed: 0.001,
  predatorSpeed: 0.006,
  avoidanceRange: 0.0001,
  alignmentRange: 0.00109,
  attractionRange: 0.00123,
  avoidanceStrength: 0.3,
  alignmentStrength: 0.3,
  attractionStrength: 0.3,
  roosctive: true,
  roostRange: 0.1,
  roostStrength: 0.3,
  blindArc: 0.5,
  predatorAvoidance: 0.01,
  predatorAvoidanceStrength: 2,
  zoom: 40,
  handle: 0,
  tick: 0,
  show: {
    avoid: true,
    align: true,
    attract: true,
    flee: false,
    roost: true,
  },
};

//ADD NEW BOIDS
```

```
function getNewBoid() {
  var retVal = {
    position: { x: 0.5, y: 0.5, z: 0 },
    speed: [
      { x: 0, y: 0, z: 0 },
      { x: 0, y: 0, z: 0 },
    ],

    highlight: false,
    color: "rgb(255,255, 0)",
    highlight2: false,
  };

  return retVal;
}

function init() {
  var i;
  simulationInfo.board = document.createElement("canvas");
  simulationInfo.board.width = simulationInfo.boardSize;
  simulationInfo.board.height = simulationInfo.boardSize;
  for (i = 0; i < simulationInfo.nBoids; ++i) {
    simulationInfo.boids[i] = getNewBoid();
  }

  simulationInfo.boids[0].highlight = true;
  toggleAvoid(document.getElementById("showAvoid").checked);
  toggleAttract(document.getElementById("showAttract").checked);
  setAvoidStrength(document.getElementById("avoidStrength").value);
  setAttractStrength(document.getElementById("attractStrength").value);
  setZoom(document.getElementById("zoomDial").value);
  step();
}


function mousePressed() {
  //WHEN MOUSE IS PRESSED IN CANVAS --> FULL SCREEN
  if (
    mouseX > 0 &&
    mouseX < windowWidth &&
    mouseY + 1100 > 0 &&
    mouseY + 1100 < windowHeight
  ) {
    let fs = fullscreen();
    fullscreen(!fs);
  }
}

function windowResized() {
  resizeCanvas(windowWidth, windowHeight);
}

function setZoom(newVal) {
  simulationInfo.zoom = newVal / 10;

}

function toggleAvoid(newVal) {
  simulationInfo.show.avoid = newVal;
}


function setAvoidRange(newVal) {
     //WHEN REPULSION HAS CHANGED (MORE THAN CERTAIN RANGE TO REMOVE NOISY VALUES)
  if (sensors0 > sensors0_old + range || sensors0 < sensors0_old - range) {
    red = map(sensors0, 104, 310, 8, 44); //MAP TIME OF FLIGHT INPUT TO ZONE VALUE
    sensors0_old = sensors0;
  }

  //IF FIREBASE IS TURNED OFF
  //NO FIREBASE
  red=10; //changable between 1-50
  //
```



```
  newVal = red;
  simulationInfo.avoidanceRange = Math.pow(newVal / 1000, 2);
  document.getElementById("avoidRange").value = newVal;
}
function setAvoidStrength(newVal) {
  simulationInfo.avoidanceStrength = Math.pow(newVal / 100, 2);
  document.getElementById("avoidStrength").value = newVal;
}

function setAttractRange(newVal) {
  //WHEN ATTRACTION HAS CHANGED (MORE THAN CERTAIN RANGE TO REMOVE NOISY VALUES)
  if (
    sensors012 > sensors012_old + range ||
    sensors012 < sensors012_old - range
  ) {
    green = map(sensors012, 440, 60, 13, 45); //MAP TIME OF FLIGHT INPUT TO ZONE VALUE

    sensors012_old = sensors012;
  }

  //IF FIREBASE IS TURNED OFF
  //NO FIREBASE
  green=35; //changable between 1-50
  //

  newVal = green;
  simulationInfo.attractionRange = Math.pow(newVal / 1000, 2);
  document.getElementById("attractRange").value = newVal;
}

function setAlignRange(newVal) {

  //IF FIREBASE IS TURNED OFF
  //NO FIREBASE
  buttonvalue=1; //changable between 0-1
  //

  blue = map(buttonvalue, 0, 1, red * 0.45 + 1, green - 2); //MAP ORIENTATION INPUT TO ZONE VALUE (SMALL OR BIG)
  newVal = blue;

  simulationInfo.alignmentRange = Math.pow(newVal / 1000, 2);
  document.getElementById("alignRange").value = newVal;
}

function setAlignStrength(newVal) {
  simulationInfo.alignStrength = Math.pow(newVal / 100, 2);
  document.getElementById("alignStrength").value = newVal;
}

function setAttractStrength(newVal) {
  simulationInfo.attractionStrength = Math.pow(newVal / 100, 2);
  document.getElementById("attractStrength").value = newVal;
}

function toggleFleeing(newVal) {
  simulationInfo.show.flee = newVal;
}

function toggleAttract(newVal) {
  simulationInfo.show.attract = newVal;
}

function toggleAlign(newVal) {
  simulationInfo.show.align = newVal;
}

function setBoidNumber(newVal) {
  // newVal = currentboidnr;
  newVal = xrow * yrow * 2;
  var i, highlightFound;
  simulationInfo.nBoids = 200;
```

```
  while (simulationInfo.boids.length > newVal) {
    //delete 1 at a time
    simulationInfo.boids.splice(
      Math.floor(Math.random() * simulationInfo.boids.length),
      1
    );
  }
  while (simulationInfo.boids.length < newVal) {
    i = simulationInfo.boids.length;

    simulationInfo.boids[i] = getNewBoid();
  }
  highlightFound = false;

  if (!highlightFound) {
    simulationInfo.boids[0].highlight = true;
  }
  document.getElementById("boidLabel").innerHTML = newVal;
}

function step() {
  var i,
    j,
    avoid,
    align,
    attract,
    maxRange,
    angle,
    minPoint,
    countEffectors,
    skippedEffectors;
  // recalculateSpeeds();

  for (i = 0; i < simulationInfo.boids.length; ++i) {
    // MAKE THEM STATIC
    if (simulationInfo.boids[i].highlight == true) {
      simulationInfo.boids[0].position.x = 0.5 % 1;
      simulationInfo.boids[0].position.y = 0.5 % 1;
    }
  }

  requestRepaint();
  simulationInfo.boids.sort(function (a, b) {
    return a.position.x - b.position.x;
  });
  setTimeout("step();", 20);
}

function draw() {
  setAvoidRange();
  setAttractRange();
  setAlignRange();
  setBoidNumber();
}

function requestRepaint() {
  if (simulationInfo.handle == 0) {
    simulationInfo.handle = window.requestAnimationFrame(repaint);
  }
}

function repaint() {
  var i, ctx;
  ctx = simulationInfo.board.getContext("2d");

  //background black
  ctx.fillStyle = "rgb(0,0,0)";
  ctx.fillRect(0, 0, simulationInfo.boardSize, simulationInfo.boardSize);

  //BACKGROUND BLUE FILLED CIRCLE
  ctx.beginPath();
  ctx.arc(
    0.5 * simulationInfo.boardSize,
```



```
    0.5 * simulationInfo.boardSize,
    (0.45 * simulationInfo.boardSize) / simulationInfo.zoom,
    (0.45 * simulationInfo.boardSize) / simulationInfo.zoom,
    0,
    2 * Math.PI
  );
  ctx.closePath();

  ctx.fillStyle = "rgb(40,40,100)";
  ctx.fill();

  //SIZE REPULSION RADIUS CIRCLE (DEPENDS ON INPUT)
  avoidradius =
    0.5 *
    (Math.sqrt(simulationInfo.avoidanceRange / 0.3) * simulationInfo.boardSize);

//SIZE ATTRACTION RADIUS CIRCLE (DEPENDS ON INPUT)
  attractcircle =
    Math.sqrt(simulationInfo.attractionRange) * simulationInfo.boardSize;

  //CENTER OF BOARD
  centerboard = 0.5 * simulationInfo.boardSize;

  //CREATE RASTER FOR ALL BOIDS

  //CREATE ARRAY GRID POINTS RASTER
  xx = [];
  yy = [];
  var distanceboid = 0.5;

  //HOW MANY NEEDED IN EACH ROW
  xrow = (2.4 * attractcircle) / avoidradius;
  yrow = xrow;

  // First create horizontal and vertical line (cross shape), so theyre not created multiple times
  m = 0;
  for (n = 0; n < yrow; ++n) {
    if (n % 2 == 0) {
      xx.push(avoidradius * distanceboid * 2 * m);
      yy.push(avoidradius * distanceboid * n);
      xx.push(avoidradius * distanceboid * 2 * m);
      yy.push(avoidradius * distanceboid * (-1 * n));
    }

    if (n % 2 == 1) {
      xx.push(avoidradius * distanceboid * 2 * m + avoidradius * distanceboid);
      yy.push(avoidradius * distanceboid * n);
      xx.push(avoidradius * distanceboid * 2 * m + avoidradius * distanceboid);
      yy.push(avoidradius * distanceboid * (-1 * n));
    }
  }

  n = 0;
  for (m = 1; m < yrow; ++m) {
    if (n % 2 == 0) {
      xx.push(avoidradius * distanceboid * 2 * m);
      yy.push(avoidradius * distanceboid * n);
      xx.push(avoidradius * distanceboid * 2 * (-1 * m));
      yy.push(avoidradius * distanceboid * (-1 * n));
    }

    if (n % 2 == 1) {
      xx.push(avoidradius * distanceboid * 2 * m + avoidradius * distanceboid);
      yy.push(avoidradius * distanceboid * n);
      xx.push(
        avoidradius * distanceboid * 2 * (-1 * m) + avoidradius * distanceboid
      );
      yy.push(avoidradius * distanceboid * (-1 * n));
    }
  }

  //CREATE REASTER GRID, ALL 4 QUADRANTS
  for (m = 1; m < xrow; ++m) {
```

```
    for (n = 1; n < yrow; ++n) {
      if (n % 2 == 0) {
        xx.push(avoidradius * distanceboid * 2 * m);
        yy.push(avoidradius * distanceboid * n);
        xx.push(avoidradius * distanceboid * 2 * (-1 * m));
        yy.push(avoidradius * distanceboid * (-1 * n));
        xx.push(avoidradius * distanceboid * 2 * m);
        yy.push(avoidradius * distanceboid * (-1 * n));
        xx.push(avoidradius * distanceboid * 2 * (-1 * m));
        yy.push(avoidradius * distanceboid * n);
      }

      if (n % 2 == 1) {
        xx.push(
          avoidradius * distanceboid * 2 * m + avoidradius * distanceboid
        );
        yy.push(avoidradius * distanceboid * n);
        xx.push(
          avoidradius * distanceboid * 2 * (-1 * m) + avoidradius * distanceboid
        );
        yy.push(avoidradius * distanceboid * (-1 * n));
        xx.push(
          avoidradius * distanceboid * 2 * m + avoidradius * distanceboid
        );
        yy.push(avoidradius * distanceboid * (-1 * n));
        xx.push(
          avoidradius * distanceboid * 2 * (-1 * m) + avoidradius * distanceboid
        );
        yy.push(avoidradius * distanceboid * n);
      }
    }
  }

  randomSeed(112233); //randomseed for the randomization of orientation when not aligned

  //BOIDS LOOP
  for (i = 0; i < simulationInfo.boids.length; ++i) {

    ctx.save();

    if (simulationInfo.boids[i].highlight != true) {
      //calculate the distance to middle for each boid
      xcoordinate = xx[i] * xx[i];
      ycoordinate = yy[i] * yy[i];
      distancetomiddle = Math.sqrt(xcoordinate + ycoordinate);

      //if it is smaller than attractcircle, show the boid
      //translate the boid to the right position in the raster grid
      if (distancetomiddle < attractcircle) {
        ctx.translate(xx[i] + centerboard, yy[i] + centerboard);
        //rotate to front
        ctx.rotate(
          2 * Math.PI -
            Math.acos(simulationInfo.boids[i].speed[simulationInfo.tick].x)
        );

        //if alignment is turned on, show them all aligned
        //if alignment is turned off, give them a random direction
        if (buttonvalue == 1) {
          orientation = 1;
        } else {
          orientation = random(-0.9, 1);
        }
        ctx.rotate(2 * Math.PI - Math.acos(orientation));
      }

      ctx.fillStyle = simulationInfo.boids[i].color;
    }


    //Boid that will show the zones, the one in the middle
    if (simulationInfo.boids[i].highlight == true) {
      ctx.translate(
```



```
        simulationInfo.boids[0].position.x * simulationInfo.boardSize,
        simulationInfo.boids[0].position.y * simulationInfo.boardSize
      );


      // DRAW THE ZONES
      //ATTRACTION ZONE CIRCLE
      if (simulationInfo.show.attract) {
        ctx.beginPath();
        ctx.arc(
          0,
          0,
          Math.sqrt(simulationInfo.attractionRange) * simulationInfo.boardSize,
          0,
          2 * Math.PI
        );
        ctx.closePath();
        ctx.strokeStyle = "rgba(50,205,60)"; //green
        ctx.setLineDash([30, 5]);
        ctx.stroke();
      }

      //REPULSION ZONE CIRCLE
      if (simulationInfo.show.avoid) {
        ctx.beginPath();
        ctx.arc(
          0,
          0,
          Math.sqrt(simulationInfo.avoidanceRange * 0.2) *
            simulationInfo.boardSize,
          0,
          2 * Math.PI
        );
        ctx.closePath();
        ctx.strokeStyle = "#FF0000"; //red
        ctx.setLineDash([30, 0]);
        ctx.lineWidth = 2;
        ctx.stroke();

      }

      //ORIENTATION ZONE CIRCLE
      if (simulationInfo.show.align) {
        ctx.beginPath();
        ctx.arc(
          0,
          0,
          Math.sqrt(simulationInfo.alignmentRange) * simulationInfo.boardSize,
          0,
          2 * Math.PI
        );
        ctx.closePath();
        ctx.strokeStyle = "rgba(0,0,255)"; //blue
        ctx.setLineDash([30, 0]);
        ctx.stroke();
      }


      ctx.restore();


    }

    // SHAPE OF BOID
    ctx.beginPath();
    ctx.moveTo(0, 0);
    ctx.lineTo(-3, 5);
    ctx.lineTo(10, 0);
    ctx.lineTo(-3, -5);
    ctx.closePath();
    ctx.fill();
    ctx.restore();
```

```
  }

  ctx = document.getElementById("viewport").getContext("2d");

  ctx.clearRect(
    0,
    0,
    document.getElementById("viewport").width,
    document.getElementById("viewport").height
  );


  ctx.drawImage(
    simulationInfo.board,
    (simulationInfo.boardSize * (1 - 1 / simulationInfo.zoom)) / 2,
    (simulationInfo.boardSize * (1 - 1 / simulationInfo.zoom)) / 2,
    simulationInfo.boardSize / simulationInfo.zoom,
    simulationInfo.boardSize / simulationInfo.zoom,
    0,
    0,
    document.getElementById("viewport").width,
    document.getElementById("viewport").height
  );
  simulationInfo.handle = 0;
}
```



```
// [index.html]

<html><head>
<title>Boids2</title>


    <script src="firebase.js"></script>

  <script src="p5.js"></script>
    <script src="p5.sound.min.js"></script>
    <script language="javascript" type="text/javascript" src="p5.serialport.js"></script>

<!--    <script src="serial.js"></script> -->
  <script language="JavaScript" src="mouse.js"></script>
<script language="JavaScript" src="boids_sim.js"></script>


   <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8">

   <script src="sketch.js"></script>

</head>
<body onload="init();">

<table border="0" align="center">
<tbody><tr><td colspan="4" align="center">
<canvas id="viewport" width="1005" height="1005" style="border:1px solid black;"></canvas>


</td></tr>
<tr><td>Zoom: </td><td align="right"><input type="range" id="zoomDial" min="10" max="200" value="85" oninput="setZoom(this.value);"></td></tr>

<tr><th></th><th>Number of agents</th><th>Movement speed</th></tr>

<tr><td>Boids: <span id="boidLabel"></span></td><td align="right"><input type="range" id="boidDial" min="2" max="2500" value="300" oninput="setBoidNumber(this.value);"></td><td align="right"><input type="range" id="speedDial" min="10" max="100" value="40" oninput="setBoidSpeed(this.value);"></td><td></td></tr>

<tr><td>Predators: <span id="predatorLabel"></span></td><td align="right"><input type="range" id="predatorDial" min="0" max="5" value="0" oninput="setPredatorNumber(this.value);"></td><td align="right"><input type="range" id="predatorSpeedDial" min="10" max="100" value="60" oninput="setPredatorSpeed(this.value);"></td><td></td></tr>

<tr><th>Action</th><th>Distance</th><th>Strength</th><th>Show</th></tr>

<tr><td>Avoid collisions</td><td align="right"><input type="range" id="avoidRange" min="5" max="250" value="40" oninput="setAvoidRange(this.value);"></td><td align="right"><input type="range" id="avoidStrength" min="0" max="150" value="30" oninput="setAvoidStrength(this.value);"></td><td align="center"><input type="checkbox" id="showAvoid" checked="" onclick="toggleAvoid(this.checked);"></td></tr>

<tr><td>Align with flockmates</td><td align="right"><input type="range" id="alignRange" min="5" max="250" value="28" oninput="setAlignRange(this.value);"></td><td align="right"><input type="range" id="alignStrength" min="0" max="150" value="30" oninput="setAlignStrength(this.value);"></td><td align="center"><input type="checkbox" id="showAlign" checked="" onclick="toggleAlign(this.checked);"></td></tr>

<tr><td>Approach flockmates</td><td align="right"><input type="range" id="attractRange" min="5" max="250" value="71" oninput="setAttractRange(this.value);"></td><td align="right"><input type="range" id="attractStrength" min="0" max="150" value="30" oninput="setAttractStrength(this.value);"></td><td align="center"><input type="checkbox" id="showAttract" checked="" onclick="toggleAttract(this.checked);"></td></tr>

<tr><td>Predator avoidance</td><td align="right"><input type="range" id="fleeingRange" min="5" max="250" value="50" oninput="setFleeRange(this.value);"></td><td align="right"><input type="range" id="fleeStrength" min="0" max="150" value="130" oninput="setFleeStrength(this.value);"></td><td align="center"><input type="checkbox" id="showFleeing" onclick="toggleFleeing(this.checked);"></td></tr>

<tr><td>Stay close to the center</td><td align="right"><input type="range" id="roostDial" min="0" max="50" value="0" oninput="setRoostSize(this.value);"></td><td align="right"><input type="range" id="roostStrength" min="0" max="150" value="100" oninput="setRoostStrength(this.value);"></td><td align="center"><input type="checkbox" id="showRoost" onclick="toggleRoost(this.checked);"></td></tr>
</tbody></table>

</body></html>
```

APPENDIX 7. ARDUINO CODE - ATTRACTION AND ORIENTATION ZONE

```
#include "Adafruit_VL53L0X.h"

#include <Firebase_Arduino_WiFiNINA.h>

#include <SPI.h>

//Firebase info and WiFi information here
#define FIREBASE_HOST "xxxxx"

#define FIREBASE_AUTH "XXXX"
#define WIFI_SSID "xxx"
#define WIFI_PASSWORD "xxx123"

FirebaseData firebaseData;

String path = "/circlevalues";

//button
int buttonpin1 = 3;
int buttonpin2 = 6;

//LED
int ledpin1 = 2;
int ledpin2 = 5;

int orientationvalue1, orientationvalue2, circlevalue3;
int circlevalue3_old;
int orientationzone = 1;
int orientationzone_old;

//TOF
Adafruit_VL53L0X lox = Adafruit_VL53L0X();
int range = 3;

void setup()
{
  Serial.begin(9600);
  delay(1000);
  Serial.println();

  //Connecting to WiFi
  Serial.print("Connecting to WiFi...");
  int status = WL_IDLE_STATUS;
  while (status != WL_CONNECTED) {
    status = WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print(".");
    delay(300);
  }
  Serial.print(" IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH, WIFI_SSID, WIFI_PASSWORD);
  Firebase.reconnectWiFi(true);

  //buttons
  pinMode(buttonpin1, INPUT_PULLDOWN);
  pinMode(buttonpin2, INPUT_PULLDOWN);

  //LED
  pinMode(ledpin1,OUTPUT);
  pinMode(ledpin2,OUTPUT);

  //TOF sensor
  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin()) {
    Serial.println(F("Failed to boot VL53L0X"));
    while (1);
  }
  // power
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));

}
```

```
void loop()
{

timeofflight();

//read value buttons
  orientationvalue1 = digitalRead(buttonpin1);
  orientationvalue2 = digitalRead(buttonpin2);

//if button1 is pressed, orientation is small 0
  if (orientationvalue1 == 1) {
    orientationzone = 0;
  }

//if button2 is pressed, orientation is large 1
  if (orientationvalue2 == 1) {
    orientationzone = 1;


   }

 //if button 1 is pressed, turn led1 on, turn led2 off
 if (orientationzone == 0) {
   digitalWrite(ledpin1, HIGH);
   digitalWrite(ledpin2, LOW);
 }

 //if button2 is pressed, turn led2 on, turn led 1 off
 if (orientationzone == 1) {
   digitalWrite(ledpin2, HIGH);
   digitalWrite(ledpin1, LOW);
 }

 //if orientation zone value is changed, update Firebase
   if (orientationzone != orientationzone_old) {
     Firebase.setFloat(firebaseData, path + "/uptodate/circlevalue2", orientationzone); //set orientation to firebase
     orientationzone_old = orientationzone;
   }

 }


 void timeofflight() {
   VL53L0X_RangingMeasurementData_t measure;
   lox.rangingTest(&measure, false); // pass in 'true' to get debug data printout!

   if (measure.RangeStatus != 4) {  // phase failures have incorrect data
     //measure distance with time of flight sensor
     circlevalue3 = (measure.RangeMilliMeter);
   }

  //if distance value (attraction) is different from previous value, update Firebase
   if (circlevalue3 > circlevalue3_old + range || circlevalue3 < circlevalue3_old - range && circlevalue3 < 5000) {

     Firebase.setFloat(firebaseData, path + "/uptodate/circlevalue3", circlevalue3);
     circlevalue3_old = circlevalue3;

   }

 }
```

APPENDIX 8. ARDUINO CODE - REPULSION ZONE AND RESTART BUTTON

```
#include "Adafruit_VL53L0X.h"
#include <Firebase_Arduino_WiFiNINA.h>
#include <SPI.h>

//Firebase info and WiFi information here
#define FIREBASE_HOST "xxxxx"
#define FIREBASE_AUTH "XXXX"
#define WIFI_SSID "xxx"
#define WIFI_PASSWORD "xxx123"

FirebaseData firebaseData;

String path = "/circlevalues";
String jsonStr;

//restart button
int buttonpin = 6;

int starteasy, starteasy_old;
float circlevalue1, circlevalue1_old;

//TOF
Adafruit_VL53L0X lox = Adafruit_VL53L0X();
int range = 3;

void setup()
{
  Serial.begin(9600);
  delay(1000);
  Serial.println();

  //Connecting to WiFi
  Serial.print("Connecting to WiFi...");
  int status = WL_IDLE_STATUS;
  while (status != WL_CONNECTED) {
    status = WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print(".");
    delay(300);
  }
  Serial.print(" IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH, WIFI_SSID, WIFI_PASSWORD);
  Firebase.reconnectWiFi(true);

  //TOF sensor
  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin()) {
    Serial.println(F("Failed to boot VL53L0X"));
    while (1);
  }
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));


  //restart button
  pinMode(buttonpin, INPUT_PULLDOWN);
}

void loop()
{
  timeofflight();

//read value restart button
  starteasy = digitalRead(buttonpin);

//if restart button value is changed, update Firebase
  if (starteasy != starteasy_old) {
    Firebase.setFloat(firebaseData, path + "/uptodate/starteasy", starteasy); //set starteasy to firebase
    starteasy_old = starteasy;
  }

}
```

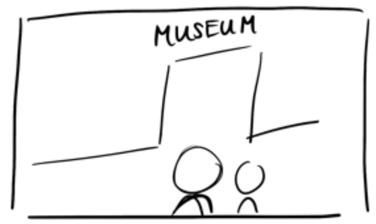
```
void timeofflight() {
  VL53L0X_RangingMeasurementData_t measure;

  lox.rangingTest(&measure, false); // pass in 'true' to get debug data printout!

  if (measure.RangeStatus != 4) {  // phase failures have incorrect data
    //measure distance with time of flight sensor
    circlevalue1 = (measure.RangeMilliMeter);
  }

 //if distance value (repulsion) is different from previous value, update Firebase
  if (circlevalue1 > circlevalue1_old + range || circlevalue1 < circlevalue1_old - range && circlevalue1 < 5000) {

    Firebase.setFloat(firebaseData, path + "/uptodate/circlevalue1", circlevalue1);
    circlevalue1_old = circlevalue1;

  }
}
```

APPENDIX 9. NATURALIS DAY CHECKLIST

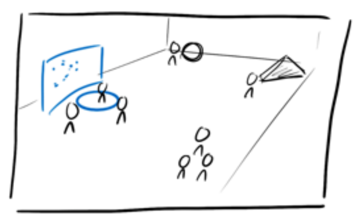
Observation:
- Count how many visitors stop and interact with the installation, and how many visitors walk past.
- Do the visitors interact with all parts of the installation (the zone of repulsion, orientation, attraction and restart button)?
- How long do museum visitors stay engaged?
- Why/when do users end the interaction with the installation?
-

Interview
- Interview users during/after interacting
- Interview Naturalis employee

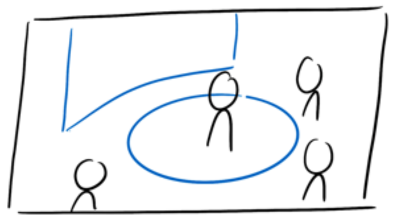
Storyboard video



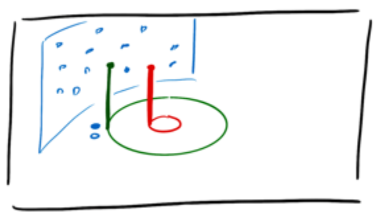
Visitors entering the museum



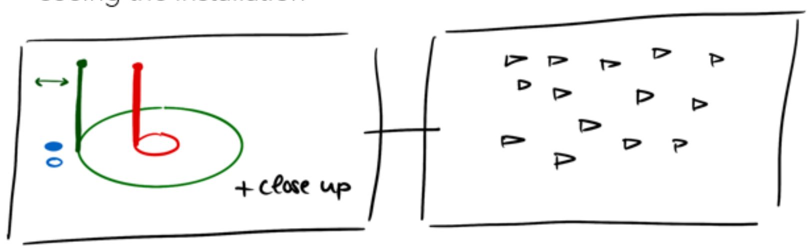
Overview of the room and seeing the installation



Zoom in on installation

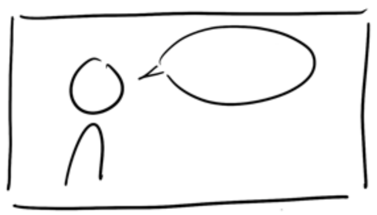


Quick overview of whole installation

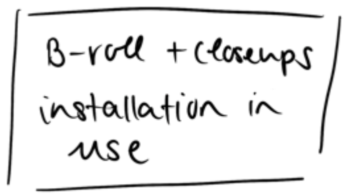


Show the interaction with the installation (floor + screen)

Interview fragments visitors



Interview fragments museum employee



Close up of users interacting etc.

APPENDIX 10. PROJECT BRIEF

TUDelft

Personal Project Brief - IDE Master Graduation

Playfully teach collective patterns of fish through an interactive design — project title

Please state the title of your graduation project (above) and the start date and end date (below). Keep the title compact and simple. Do not use abbreviations. The remainder of this document allows you to define and clarify your graduation project.

start date 02 - 09 - 2021 — 25 - 02 - 2021 end date

INTRODUCTION **

Please describe, the context of your project, and address the main stakeholders (interests) within this context in a concise yet complete manner. Who are involved, what do they value and how do they currently operate within the given context? What are the main opportunities and limitations you are currently aware of (cultural- and social norms, resources (time, money,...), technology, ...).

Many collective motion patterns can be found in nature, take for example a flock of birds or a school of fish. These groups and their individuals move and behave in a certain type of way. Research has been carried out on these collective motion patterns, but this phenomenon can be complex to understand and is hard to explain in words.

We see this collective behaviour everywhere around us in animal groups in nature, but most people do not know how and why these animals coordinate this way. Here, a knowledge gap is found, which can be solved by making people aware of the concept and explaining in an easy way how this collective motion works.
A visual design that translates this research into something which will be easy to understand, will be very helpful to teach this collective behaviour concept to others. To reach a wide audience to fill this knowledge gap, the context of the design will be a museum, showing and explaining its visitors these collective motion patterns. The design will be used and viewed in an interactive way to achieve a better comprehension: the viewer will be able to adjust factors, see the changing result and get a more in depth understanding of the concept.

Stakeholders involved will be:
- Museum goers: while enjoying their museum visit, they will learn about the collective patterns in nature and get a deeper understanding of these motions. New insights will be added to their general knowledge.
- Museum (like Naturalis and NEMO): their value is to educate its visitors, with this design they can teach about collective flow patterns in animal groups and make its visitors aware of what we actually perceive in our daily life. A new design added to the museum will also attract new visitors.
- Researchers of collective motion patterns: translate research into something understandable to be able to show a wider audience.

At the moment, there are many technological opportunities that can be used to create this design. With the use of sensors, actuators, lights, moving parts etc. collective motion patterns can be shown and be more easily understandable. A great number of programs and tools can be used for the interactivity part, but computing time, reactiveness and ease of use are aspects to take into consideration.
The key point of the resulting interactive design is to playfully teach the museum visitors, in an engaging way, for them to get an understanding and awareness of collective motion patterns in nature.

space available for images / figures on next page

IDE TU Delft - E&SA Department /// Graduation project brief & study overview /// 2018-01 v30 — Page 3 of 7

Initials & Name T.F. Tan — Student number 4562232

Title of Project Playfully teach collective patterns of fish through an interactive design introduction (continued): space for images

image / figure 1: A school of blue striped snappers

image / figure 2: A milling school of fish

PROBLEM DEFINITION **

Limit and define the scope and solution space of your project to one that is manageable within one Master Graduation Project of 30 EC (= 20 full time weeks or 100 working days) and clearly indicate what issue(s) should be addressed in this project.

The opportunity arises to teach people something they see everywhere around them, that already has existing research about how the phenomenon works, but is a knowledge gap for most people. Collective motion follows three basic rules: repulsion (avoid collision with others), alignment (move in the same direction), and attraction (steer towards the center/stay close to others). The challenge is to translate this current research, into an interactive design that will be easily understandable.
Since this concept of collective motion patterns is still very broad, I will focus on the motion of schools of fish. I find it fascinating to see how small a fish is compared to the ocean, but it still swims and coordinates together with others in a group. I think it is very interesting for many people to achieve an understanding about this.

I will focus on the design part of the interaction experience. I will make use of existing algorithms of schools of fish motions, and brainstorm, ideate and create a design that will teach the basic rules of collective behaviour in a interactive way which is easily understandable and aesthetically pleasing. The question here is how this will take shape, what kind of interaction it will include and how the overall experience would look like.
The users are key here, the museum visitors should be entertained while learning and getting aware of what is actually happening in the movements. However, they should should not feel overwhelmed by all of it, so a balance should be found between being educational and enjoying the design. Important is that the users feel engaged, and spend enough time interacting with the design to get an understanding of collective motion patterns in schools of fish.

ASSIGNMENT **

State in 2 or 3 sentences what you are going to research, design, create and / or generate, that will solve (part of) the issue(s) pointed out in "problem definition". Then illustrate this assignment by indicating what kind of solution you expect and / or aim to deliver, for instance: a product, a product-service combination, a strategy illustrated through product or product-service combination ideas, ... In case of a Specialisation and/or Annotation, make sure the assignment reflects this/these.

I am going to research and ideate how algorithms of collective motions of schools of fish can be translated in the most engaging way into something that is visible, easily understandable and interactive. This design, with as context a museum, will playfully teach its visitors (8 years and older) about these motion patterns in schools of fish, with as objective creating something that is engaging, memorable and educational.

I aim to deliver a working experienceable concept design at the end of my graduation project. This design will focus on the user experience: through interactivity learning about collective motion patterns in schools of fish. The intended design will be designed to be placed in a museum.

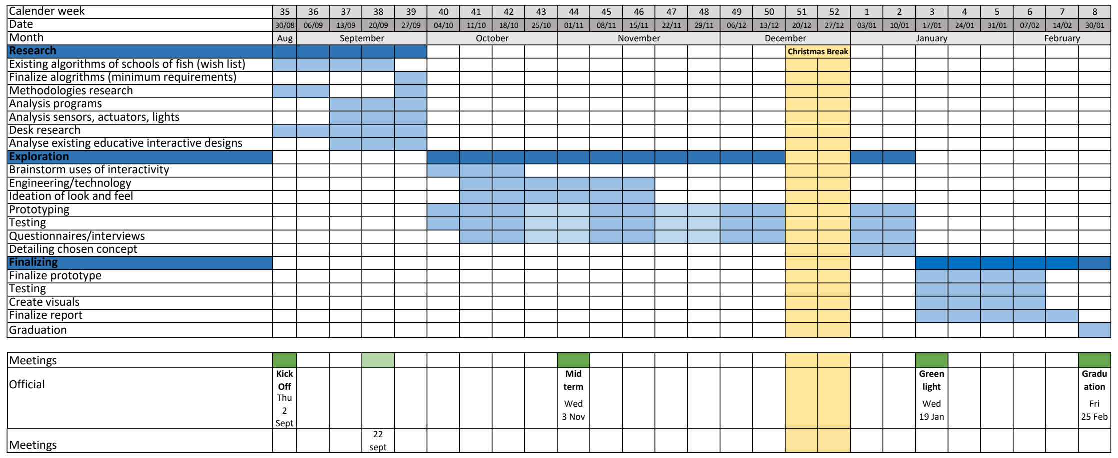
During this graduation project I will focus on the following things:
- Finding existing algorithms of collective motions of schools of fish.
- Research into different programs and technology that exist, to be able to translate the collective behaviour rules into something visible. This includes software for computing, but also sensors, actuators and other hardware.
- Exploration about different uses of interactivity (experience). What will be intuitive, easy to understand, but also be fun to use to stay engaged. Here, I will validate my findings with questionnaires and tests with the target audience.
- Ideation of different shapes and concepts of this design, with the use of sketching, drawing, programming, 3d modelling and prototypes. With the Build, Measure, Learn method I will go through multiple loops to validate my ideas.
- Ideation how to expand the design, with for example: predators, obstacles, etc.
- Detailing the final design, concluding on its use, interactivity, parts and dimensions
- Finishing the final prototype that can be experienced by the actual target group (works-like)

PLANNING AND APPROACH **

Include a Gantt Chart (replace the example below - more examples can be found in Manual 2) that shows the different phases of your project, deliverables you have in mind, meetings, and how you plan to spend your time. Please note that all activities should fit within the given net time of 30 EC = 20 full time weeks or 100 working days, and your planning should include a kick-off meeting, mid-term meeting, green light meeting and graduation ceremony. Illustrate your Gantt Chart by, for instance, explaining your approach, and please indicate periods of part-time activities and/or periods of not spending time on your graduation project, if any, for instance because of holidays or parallel activities.

start date 2 - 9 - 2021

25 - 2 - 2021 end date



(Due to have all people present, the kick off meeting took place one week earlier (week 34) than the graduation project is going to start (week 35).

In the research phase I will first look into existing algorithms and see if it is possible to include all the aspects from my 'wish list'. If this is not possible within 3.5 weeks, I will continue with the algorithms I found that cover my minimum requirements. In the exploration phase I will go through loops to iterate my ideas, findings and insights. Lastly, in the finalizing phase I will finalize my experiencable prototype and create visuals and detailing for the created design.

Minimum requirements
- Algorithms behaviour of collective patterns in schools of fish
- Space boundary of a box
- Being able to follow an individual in the group
- Change in behaviour because of an obstacle

Wish list
- Different algorithms for different species of fish (different behaviours)
- Change in behaviour because of predators and food
- Adjustable space boundary

MOTIVATION AND PERSONAL AMBITIONS

Explain why you set up this project, what competences you want to prove and learn. For example: acquired competences from your MSc programme, the elective semester, extra-curricular activities (etc.) and point out the competences you have yet developed. Optionally, describe which personal learning ambitions you explicitly want to address in this project, on top of the learning objectives of the Graduation Project, such as: in depth knowledge a on specific subject, broadening your competences or experimenting with a specific tool and/or methodology, ... Stick to no more than five ambitions.

-Engaging interactive design
Something I have always been interested in, is creating an engaging design. I want people that interact with my design to feel entertained and engaged by it, and leave them with something to remember or give some meaning to them. That is why I set up this graduation project. I think as a designer it is important to design with the user in mind, to be able to optimize this intended outcome through testing, adjusting and detailing.

-Integrated Product Design and prototyping
I chose the IPD master since I like building prototypes to test my ideas. This is something I missed a bit during my Master (pandemic). In the coming semester I hope to be able to carry out many user tests to validate my findings. Currently, I have mostly used Arduino for programming, and 3D printing for prototypes. I am motivated to try and learn new things, therefore during this graduation project I want to take the time to explore other directions of creating things and broaden my skills here.

-Elective courses knowledge
• Design Drawing
During my elective semester I took drawing courses (analog and digital) to improve my visualization techniques. I think this is a skill important for designers, but at the moment I feel like I still have to improve a lot here. Therefore, I want to use design drawing in my graduation project during the multiple phases to grow this visualization skill, so I can communicate my ideas in a better way.

• Lighting Design
I am also very inspired by the lighting design elective I followed last year. I obtained a lot of new knowledge: theory, hands on, but also how to show lights in the digital media. I would like to put this knowledge to use during my graduation project.

FINAL COMMENTS

In case your project brief needs final comments, please add any information you think is relevant.

At the moment, my supervisory team and I are in contact with museums (Naturalis, NEMO, Natuurhistorisch Museum etc.), to see if they want to collaborate on this project.