# Final Report on Group Project

# LaserHoops

DZC30, Group 5


| | |
|---|---|
| Alžběta Mrkvová | 1955691 |
| Joris Raaphorst | 1685368 |
| Katarzyna Klęczek | 1911384 |
| Lisa Rabbers | 1304240 |


# Contents

# Introduction

This is the final report for the course “Design for games & play III; playful interactions”. Our team designed and created a prototype of a playful object that can be placed almost anywhere and can be used by many people at the same time. Our journey was not straightforward, but the many ideations and discussions led us to create something unique, something we are quite proud of.

In this report, we first describe our Design Process. The chapter can serve as a signpost for the report, navigating the reader throughout the report. We decided to structure it this way, since it better pictures the whole process of the creation.

# Design Process

This chapter covers the design process, in which four iterations were conducted over a span of eight weeks. Check for a complete overview of the process diagram in Figure 1.

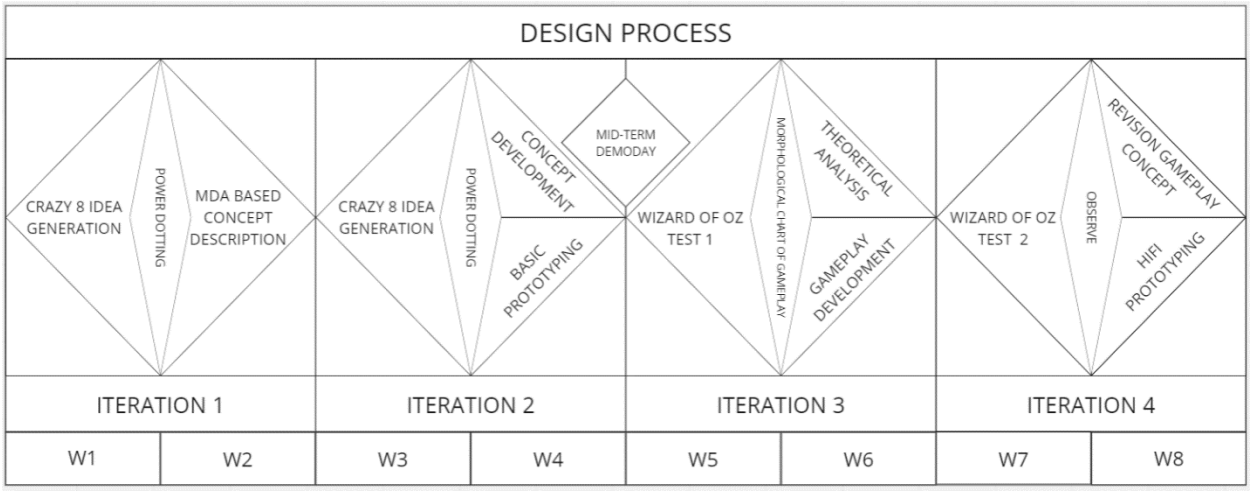



*Figure 1: Design Process Diagram*

During the first two iterations ideas were generated based on the core direction “Garden” by utilizing the Crazy 8’s method (Google, z.d). Each session was conducted with four people, generating 8 ideas each, and were narrowed down using power dotting and dialogic discussion techniques.

Iteration 1 resulted in four concept directions, namely, Stepping Stones, Stealth, Gnome, and Singing Fruit. These idea directions were further explored during a writing exercise expanding on the mechanics and dynamics of the ideas to assess feasibility and viability, allowing to narrow the idea directions down to the Stealth and Stepping Stone direction.

Iteration 2 was included to ensure proper creative coverage for these ideas, generating 16 ideas per direction. After an internal discussion on feasibility, viability and desirability, power dotting was used to select the final idea, which resulted in merging of the Lava Tiles and the Stealth Hoop concepts.

Mid-term demo: Merging the concepts resulted in the core idea of LaserHoops, which was presented as a preliminary concept during the mid-term demo-day. The presentation was well received, confirming core concept selection, mechanical set-up and our initial gameplay design.

Iteration 3 focused on the dynamics of the gameplay. By performing an early Wizard of Oz user test, we collected user insight on the gameplay, and managed to build a morphological chart covering different approaches to potential functionalities within the gameplay. Parallel to this, a thorough theoretical analysis was conducted to evaluate our design, enabling us to ground our assumptions into theory resulting in several optimizations of the gameplay design.

Iteration 4 was used to evaluate the design, since the prototype was still in development, a second Wizard of Oz test was conducted with the TAs of the course. Using observations, we managed to confirm many of our design intentions, but also allowed us to collect user insights for optimization.

For delivery a fully functional prototype was created which included almost all of the design intentions. During the presentation valuable feedback was provided, encouraging us to continue the development of the concept, which results in a promising future works chapter.

# Iteration 1

This chapter covers the first iteration of our design process and our considerations. We cover the design considerations which are supported by examples from our leading concept. The other design descriptions are attached in A.2 - Iteration 1.

## Concept Descriptions

A round of Crazy 8's was the start of the journey. We limited our ideation to the context of the backyard and lamp because we saw many potential problems with the other two contexts – lake and glove. For example, we did not want to deal with shielding electronic components from the water. We decided to put the ideas we thought of on paper in these eight minutes. We generated ideas without judgement, as such each idea was considered valuable input to build upon. From this, thirty-two ideas were born. By clustering similar ideas and performing a power dotting exercise four idea directions were selected. We named the concepts Stepping Stones, Stealth, Gnome, and Singing Fruit. We elaborated on each concept by writing out the concepts of the games. Mainly the mechanics and dynamics of the MDA model by Hunicke et al. (2004), and playful experience by Korhonen (2009) (PLEX in the text) were considered here.

In this chapter, we mention only two of the four concepts, Stepping Stones and Stealth, because we build on these concepts our final idea. More information about the other two concepts can be found in Appendix A.2 - Iteration 1.

## Stepping Stones Concept

This playable object is a tiled path or a playground that can be placed in a backyard. The tiles can glow and/or play sounds, they are translucent (or backlit as a keyboard) to show the light, and they have a pressure sensor that enables interaction with it.

One of the possible gameplays would be to step on randomly lit-up tiles as quickly as possible. Players could try to best each other's time. The following idea was to make the tile glow with more intensity or to make some increasingly louder sound if it was not stepped on in some time interval.

Another gameplay would have the players step on the unlit stones, the glowing ones would indicate dangerous places – like in the game “The Floor is Lava”. The stones could light up and down randomly, changing the playing field. The goal would be to reach the other end of the path without touching the lit-up tiles.

There could also be a setting where the tiles light up when someone steps on them, creating a memory of the person’s passage.

This concept fits well with the experience of competence – it encourages the user to gain the skill of mastering this game. Players can get better at the game if they play it more often and complete it in shorter times. The tiles would give the players a feeling of autonomy and self-expression in the non-game setting, as they are in charge of what tiles light up. This object offers challenge and competition. The players can challenge themselves to complete the path or to complete it in a shorter time.

PLEX: challenge, competition
Potential sensors: pressure sensor, piezo sensor, timer
Potential actuators: light, speaker

**Dynamics**
There are several dynamics that we can think of, featuring the light actuators, audio, and the shape of the tiles.

Players step on the tiles and run through the garden as if it were a racing game. They would want to step on all the lit-up tiles, and they would run and quicken their pace if they knew they had a limited time before the tiles dimmed.

Players would follow different lit-up paths every time they are in the garden to create different paths. They would have to explore more parts of the garden, or they would end up using different roads. Each time the number of steps and their location is initialized randomly and thus creating a new experience every time. Players would consciously look for the tiles that start interaction and try to finish it, this would happen probably if random tiles were making louder and louder sounds but not creating a path.

The emerging dynamic "The Floor is Lava" would be avoiding the tiles that are lit-up. If the colour of the light is red, they may avoid tiles, if it is green or white, they may be more encouraged to follow the path and step on the tiles.

Depending on the sounds, players may be encouraged to step on the tiles more or less. The intensity of the sound (louder or softer) will either motivate them to interact with the tile or to just ignore them.

The shape of the tiles affects the interaction – it may be either harder to step on a smaller tile, or not challenging for others to step on bigger one.

Since the interaction starts with someone stepping on a tile, there is not much variability on how it can affect the mechanics, it is the way to start the interaction. Unfortunately, they won’t know that there is an interaction unless the tiles randomly light up or fade to indicate some action of the object.

Stepping on stones is the main mechanic of this object and it allows for most of the dynamics that are mentioned above. Some of them also depend on the actuators but stepping nevertheless is the most important part, therefore the code name Stepping Stones.

## Stealth Concept

Do you think you could be a good burglar and not wake up the sleeping residents? While watching an action movie, have you ever wondered how you would fare in a room interwoven with lasers? Then you should test your "burglar skills".

There are many directions this game can take. You can try to move through a web of lasers without disturbing them. If you interrupt the light, some tone will sound. It can be an annoying sound like a buzzer – you lost the game. Or it can be musical, like walking through the strings of a harp. It is up to the player for which version they have mood – more artistic, or with more skill and various movements?

You can play a multiplayer stealth game. Burglars and the police: move quietly, softly, slowly... The system will encourage the players to follow the instructions and move through the garden. In a game with multiple players, they will feel the desire to overcome and best their opponents. Single-player version: follow the system's instructions (follow the light on a bracelet, for example) and get points accordingly. Multiplayer: compete while following the system's indications and try to survive the game longer. Burglars and the police get their actions randomly. Burglars don't want to get caught by the police. A player gets disqualified upon not following instructions (buzzer). Possible sets of instructions could be don't stay still for longer than a few seconds, don't move too fast (indicated by yellow light), stay completely still at some moments (red light), or else be on the move through the garden (green light).

How much autonomy the players have depends on the game mode. The game can be played with very restrictive instructions, or with more freedom in action. The players feel competent when they follow the instructions correctly and don't get disqualified.

PLEX (burglars and police): competition, immersion, challenge to get away from opponents, the thrill of the chase, and the possibility of failing to follow the rules.

For the laser grid version, arbitrarily set-up lasers would be installed in some more confined environment, e. g. grove. Interrupting a laser beam produces a tone. It can be played by an individual or by a team. Although a tempting idea, it is not that suitable for playing in a backyard. It would be an intricate and artistic light installation instead (in foggy weather).

PLEX (laser grid): immersion, challenge, thrill; expression, and possibly relaxation in the musical mode of the laser grid.
Potential sensors: accelerometer, sound level sensor, seismometer (for burglar & police game), lasers (for laser grid)
Potential actuators: speakers, lights, buzzer

**Mechanics**
We describe a wearable version of the concept. The wearable device contains multiple sensors that track the player's movement or other behaviours. These sensors can be an accelerometer, a sound level sensor, a seismometer, etc. The device itself is wearable – a bracelet, a waistband, or a shoulder belt.

The object also has speakers, lights, or a buzzer as actuators. The lights give instructions to the player. For example, a blue light would indicate that the user must move around, and red means to stay silent.

**Dynamics**
Players wear the device on their bodies while playing a game that can be played with multiple people. They can play hide and seek, with the device telling them when they need to move or stay still. If the players don't follow the instructions, a sound is played, giving away the player's location. This leads to feeling a thrill. The players experience time pressure to follow the instructions, while they need to simultaneously think about the game they are playing. This creates a challenge.

**Aesthetics**
These dynamics lead to a playful object that has immersion, challenge, and thrill for the user.

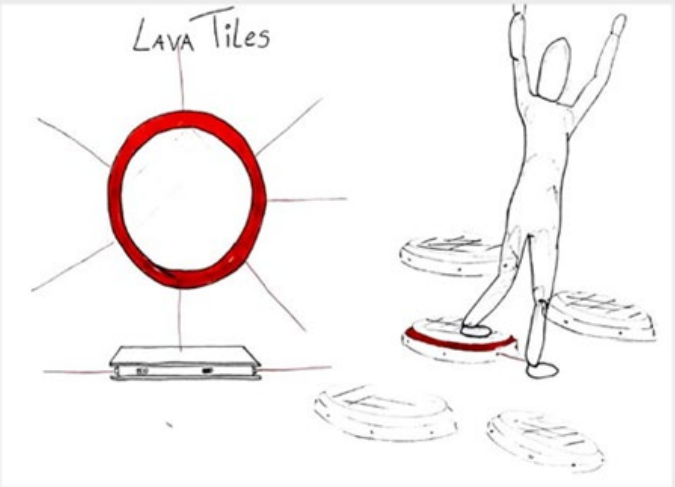
# Iteration 2

In this chapter, we elaborate on two of our initial ideas and come up with the idea of LaserHoops.

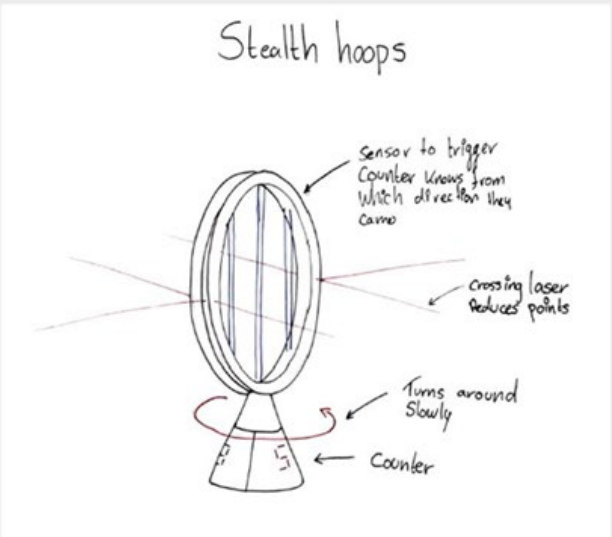
## New Concept: LaserHoops

The core ideas all contained promising directions, especially Stealth and Stepping Stones, but had not been explored fully yet. To ensure proper creative coverage for these ideas we decided to do another Crazy 8's idea generation based on those two idea directions. Sketches of the generated ideas are in Appendix A.3 - Iteration 2.

The pictures below (Figure 2) are from the second Crazy 8's. Each member had five votes to distribute to their favourite ideas – power dotting method. After a discussion on feasibility, viability and desirability, we decided to merge these two ideas into a more playful object, a hoop with sensors in the inside and outside. That was the start of LaserHoops.




**Lava Tiles Concept** - These stepping stones are your safe haven in the dangerous game of the floor is lava! Each stone lies in the middle of pool of lava and if you step into it the tile scream in horror (and signal red). The gameplay is versatile, as you can integrate it in many forms of role play, indoor and outdoor. Only one tile will already turn a playground into a lava




**Laser Hoop Concept** - Playing laser hoops is not easy! As the ring turns around to need to pass a ball, frisbee or yourself through the hoops without stepping into the lasers! Play against the game, or against each other!

*Figure 2: Lava Tiles and Laser Hoop Concept*

## LaserHoops: Concept Description According to the MDA Model

*"The content in this chapter was the state of delivering during the Midterm Demo Day."*

LaserHoops transform the environment into a playing field and can be used in many ways. To give a clear description of the design, the MDA design by Hunicke et al. (2004) will be used. To start, an overview of the mechanics is given, then the dynamics will be discussed, and lastly, the aesthetics that are created by these dynamics.

**Mechanics**
The user can jump inside and outside of the hoop. The hoop contains distance sensors on the inside, that track if the user is inside. Besides that, it also contains motion sensors on the outside. If the user moves on the ground (or the level on which the hoops are placed), these sensors will pick that up. The hoop has LEDs strip both on the inside and outside that communicate various states of the object. They show the number of points the hoop has in blue LEDs and light up red outside when detecting a player outside the hoop, or inside with green lights when the player is inside. The lights are also used for countdowns.

**Dynamics**
The hoops can be seen as save zones. If the player is inside the ring, the hoop gains a point. However, if the player stays inside the ring for too long, the ring "sinks" and loses a point, so the player needs to leave quickly. If the player gets caught by the sensors outside the ring, the ring loses a point. If the hoop stays unvisited for too long, a countdown starts and when it's done, it also loses a point. When a hoop runs out of points, it "dies", turns off its lights, and stops interacting.

This combination of dynamics ensures that the players keep moving, while also being careful not to get caught by the outside sensors.

LaserHoops transform any space into a "The Floor is Lava" game. Depending on which environment the player chooses, such as a field or a classroom, there are different factors that influence how to play the game. For example, in a classroom, many chairs and tables could be used to get from one hoop to another safely. But in a field of grass, the players need to jump from one hoop to another. It is however advised to use the hoops in combination with other objects, to increase the number of options the players have.

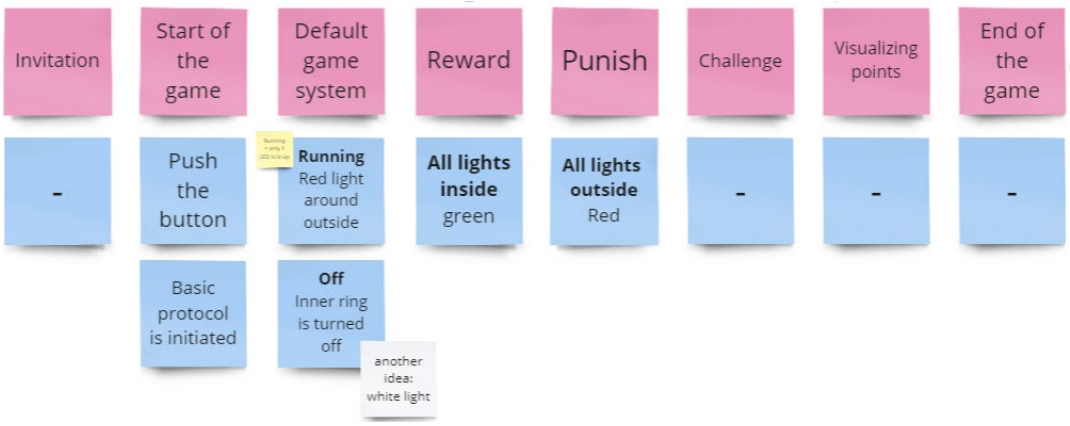



*Figure 3: Initial Gameplay Design*

**Aesthetics**
These dynamics together create multiple aesthetics, for instance a challenge. The game can be described as an obstacle course. Players can get better at it and come up with new strategies. Besides that, players can use their imagination and fantasy to see the hoops as safe islands in a lake of lava. To keep all the rings alive, and gain as many points as possible on them, players should work together, and develop different strategies or divide roles, which rely on their fellowship.

The different ways the lights are used create a sense-pleasing effect. The lights show what the player is doing wrong or right. Besides that, it also can be pleasing from a purely artistic point of view. The rainbow lights give a happy sensation that invites the players, and an entire field of lit-up rings can look magical. The game also allows for expression, to a certain extent. The users can play around with how and where they place the rings, what kind of other items they use to move above the sensors, how far apart they place the rings etc.

As for the playful experience categories, as described by Korhonen (2009), this idea would be suitable for challenge, competition, and exploration.

# Iteration 3

This chapter focuses on the gameplay of LaserHoops. We performed the first user test to collect insight of the users on the gameplay. We elaborated on the concept with a morphological chart that described various signals (light and audio) of the hoop. These signals are for the players to know in which state the hoop currently is, so they can decide what actions to take in the game of “The Floor is Lava”. This chart can be found in Appendix A.4 - Iteration 3. We then selected one signal per state for implementation and for the second user testing.

After the design decisions inspired by the user test, we introduce theoretical analysis evaluating the design of LaserHoops.

## First User Test

For the first user test, we conducted the Wizard of Oz test. We prepared four hoops from bicycles. We planned to let the testers play with the objects however they liked, without providing them with any information. We observed them during their interaction with the object.

We started with the four hoops on the ground for the user test. After a few minutes of the interaction, we put the rings upright and gave the testers a ball. Later, we told them about a simple ruleset for playing the game “The Floor is Lava”. The testers could not walk in close range to the hoops where the lasers would pick up their movement, but by jumping into the hoops they could earn points. They had to think of other ways to get from one ring to another. For example, one participant hopped from loop to loop, and another tried to climb over tables and chairs.

**Participants**
We asked another team to participate in this user test, three men in their 20s.

**Summarized Findings**
We used the PANAS questionnaire to gain insight into their moods. They were to fill it out before and after interacting with the hoops. Additionally, we asked them some open-ended questions about their

first thoughts about the objects, which interactions and features they liked the most, and if they had any other ideas on how to use the object.

The most important findings were that the participants felt more interested, more determined, more proud, more active and more inspired after playing with the object. One participant felt less irritable, and one felt less strong after playing with the object.

Without the rules, the object was really vague and not very interesting for the testers. The light strips would be a positive addition. We were advised to protect the object better and to use audio with the light indication we outlined to the participants.

**First Setup of the Technical Ruleset**
At the time of the first user test, we started working on the prototype. The mechanics of LaserHoops were envisioned and described to the testers like following:

- There are LED lights on the inner ring and on the outer hoop.
- The LEDs light up green on the inner hoop if movement is sensed on the inside of the hoop, and red on the outer hoop if there is movement sensed on the outside of the hoop.
- The amount of lit-up green LEDs on the inner hoop is proportional to the number of points that are scored.

## New Design Decisions

After the first playtest in week 4, we established more visual signals of the system. There are different kinds of visual feedback, distinct colorus indicate different states: green – something positive has happened, red – something negative has happened or is going to happen, blue – connected to a hoop's life. We chose the colours that are associated with "right" and "wrong" so the players would have some idea of what is happening. We changed the life signalization from green to blue, so the colourful reaction is separated from stepping into the hoop.

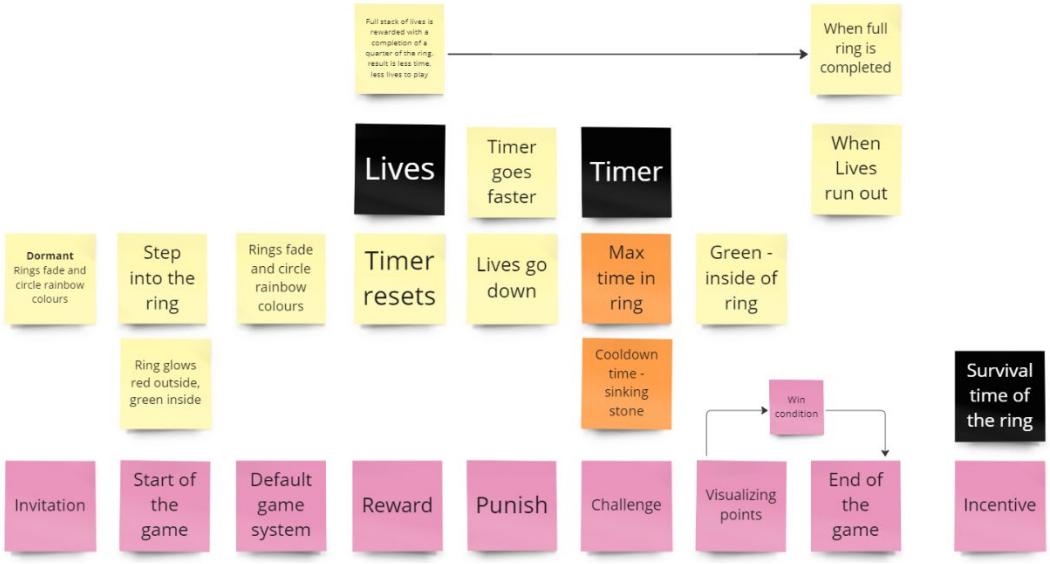



*Figure 4: Revised Gameplay Design*

We wanted to use multiple patterns for the lights. For countdowns, the LED strip would incrementally light up. The pace to which the lights are added tells the player how much time they have left, it is connected to the urgency of an action, and the player can plan his next actions accordingly. Blinking pattern of blue lives signalizes that something is about to happen to the lives if no actions is taken, in this case the hoop loses a life if the hoop is not visited in time. A fast green blink is equivalent to a cheerful "hurrah" when a correct move was played.

Our main design is the imitation of the game "The Floor is Lava", so it uses multiple hoops to cover a bigger surface. However, it is possible to use one hoop in an enjoyable and fun way. An example of it being used individually is as a basket to throw a ball through and for counting points. More options for gameplay bring the players more freedom in what type of game they can play. They can have different and various playful experiences with the same object but with different rules, hopefully never getting bored with the object and throwing it away. But for now, we decided to focus only on the main design.

## Theoretical Analysis of LaserHoops

In this chapter, we critically analyse the concept of LaserHoops from the viewpoint of several theories. We start with the Self-Determination Theory, then follow with an analysis in the context of social interaction among the players, and end with the analysis of experience of the audience, i. e. people who are not close enough to interact with the playful object but can observe the various interactions.

### Self-Determination Theory

By considering motivational frameworks we can better understand what players want to get out of an experience and optimize our designs accordingly. The Self-Determination Theory (SDT; Deci & Ryan, 1985; Ryan & Deci, 2000b), a well-researched theory of motivation, is well suited for assessing gameplay. The core of the theory revolves around three key themes which together are a recipe for intrinsic motivation. SDT suggests that the themes lie at the foundation of psychological wellness, and that Autonomy, Competence and Relatedness are primary, basic needs.

**Autonomy** is apparent since the placement of the rings is up to the players and even though the rings express behaviour, indicating gameplay, it is up to the players to deal with this behaviour. However, each game needs some constraints, to create a challenge. As such the autonomy of the players is limited by several elements. The behaviour of the rings gives the players an incentive to keep on moving. The rings are slowly dying and can only be recovered by paying them regular visits. Additionally, a 'sinking' effect has been added to the gameplay to prevent camping. As well as a cooldown effect, to prevent players from waiting next to a ring and repeatedly activating it. When considering physical constraints, one would be that rings cannot be relocated when active. Additionally, we want to prevent players from standing on top of the ring, which would break the gameplay. To prevent this, a few pressure sensors could be placed on top of the ring, this would also allow for controlling some settings with the foot.

**Competence** revolves around the balance between challenge and achievement. When a challenge is too hard, players might give up, when a challenge is too easy, players might get bored (Schell, 2008). Since the competence of players grows with time played, it is required that the challenge level of the game can be modified. When reviewing LaserHoops, it is easy to state that the modular nature of the hoops allows players to set the game up in an ever more challenging way. However, not all play areas support an ever-increasingly challenging configuration. Just as easy is stating that expanding the number of rings

would increase the challenge difficulty. And yes, when we theoretically place a hundred rings in a field, a nice challenge–achievement balance might occur as players attempt to keep as many rings alive as possible. However, we must acknowledge that increasing the number of rings would be an expensive endeavour, reducing accessibility to customers. We should therefore consider how we can accommodate players in tweaking the challenge levels of the game. Since the current gameplay incorporates a timer and a point system, it would be fairly easy to allow players to set difficulty levels on the rings. In which each level has different settings regarding points to reward and time and severity to punishment.

**Relatedness** might be the least strong of the three, which doesn't say it is not present. LaserHoops can be played by just one person, however, gameplay with others is supported and in theory, the game could be infinitely expanded. That the three core functions are represented in the gameplay is positive. LaserHoops was not initially designed for supporting relatedness, and the game facilitates individual play as well as team play because players play against the game. Within an open-play configuration, a player-versus-player (PVP) gameplay can be imagined, but this is not necessarily intuitive. It is therefore that we might find opportunity in incorporating elements for PVP gameplay. By for instance assigning players to a colour, linking the points to the activated colour of a ring, and circulating the activated colours. More players could participate with a lower number of rings, while adding an additional challenge dimension and increasing the relatedness, as player interactions increase.

## Social Interactions

Interaction is designed both to be fun and playable individually and with more people. The target group for this experience are people who are physically able and have some imagination, though this is less important. If more than one person plays the game, they will be able to freely communicate with each other vocally as well as see other people's body and face cues. It will thus be an example of local co-presence and co-location. People are going to be present at the same place and time and free to interact with each other. Ideally, they will want to cooperate and thus inform another person, for example, if a ring is blinking or to which ring they will go next. They can do that easily by both telling/shouting information to other players and pointing to a direction to specify a ring they are referring to. What they can also do is try to help each other physically to go from one ring to another. They can try to move furniture to make it easier, give someone a hand, give them some helpful instructions of what in their opinion the easiest or fastest path is to take. They can assign roles to each other, for example, one can be a watcher, two can be jumpers, or assign rings to a specific person.

There are many opportunities for others to get engaged with the game, depending on the space the interaction is settled in, either an open space, a closed garden, or even a room. If it is in a closed room, then it is highly unlikely for the number of players to increase during the gameplay, but if it is in a park then probably chances of some interaction between people playing and the audience are more likely. The rings themselves don't really aid privacy but also this experience is not designed for that. In terms of personal distance, it solely depends on the players themselves if they decide to be in the same ring together or not. It could happen that they will touch, but most of the time probably will be more than 3 meters away from each other (De Kort & IJsselsteijn, 2008).

The interaction can be placed in a large space and anyone who passes by it could start the interaction. The most probable interaction of passers-by would be the accidental triggering of outer sensors and having a negative impact on the gameplay. However, the space could be designed in a way that the

borders of interactions are visible, and people can safely stand nearby and watch the game without provoking any undesired reactions.

## Audience Experience

The experience for the audience is observing the game of LaserHoops from afar because the playing field is quite large due to the sensors (ca 3 meters from the outmost rings). It can be entertaining to watch the athletic feats of the players as they try to get into the hoops without disturbing the sensors. The audience can talk with the players and shout their suggestions and warnings, since they share the same open area.

LaserHoops are designed for public spaces. With this setting, a question arises: how will the audience perceive the interaction? In the paper by Reeves et al. (2005) they introduce a taxonomy of design strategies for public spaces. The strategies depend on how much the manipulations and effects of performers with an interface are revealed or hidden to the audience. In our case, the players' manipulations are revealed for both the players and the audience. For the players, the effects are revealed, because they need to be able to see the effects of their actions or inactions (light of the hoops) from afar to be able to react in time and relocate to a place of interest to achieve a subgoal (i. e. keeping a hoop alive). If the playing field is large, the effects can be only partially hidden from the audience, since the light from hoops far away can be hard to see. The audience will see the players' movements and the light signals from the system. The audience can learn from observing the players, gain insight into the game's rules with some solutions for how to cross the playing field without touching the forbidden floor. They can then use this knowledge if they decide to start playing the game. The mapping of the players' manipulations is linear. There are but a lot of sensors that can be triggered by anything (not just the players) so some of the triggering manipulations can go unnoticed, resulting in effects that are hard to explain for everyone involved.

Once the game starts, anyone can become a player if they come to the playing field and try to follow the game rules. The smoothness of the transition from observer to player depends on the layout of the obstacles near the border of the playing field.

The audience should be aware of the playing field because reckless or uninformed passing through the playing field could trigger some of the sensors. It could drastically change the game's progress and frustrate the players.

The audience can be divided into prospective players who are ready to cross the invisible borderline and join the game; observers standing close to the playing field and watching what is happening, maybe interacting with the players (or helping them) in some way; people passing through the field and disturbing the game; and finally, people far away from the players and the game, noticing interesting movements of the players and possibly being lured to become observers of the game.

# Iteration 4

## Second User Test

**Method**
In the second user test in week 5, we used again the Wizard of Oz method. Our prototype was not fully finished yet and was rather fragile. Furthermore, the prototype is a singular hoop and for our intended gameplay, more hoops were needed. The setup of the test was following:

Four bicycle hoops that were imitating the laser hoops were placed on the ground far away from each other. Two of the experimenters were standing in between the wheels with six different signs: paper with a red arrow making a circle (indicating a countdown until a hoop sinks), paper with a blue arrow making a circle (indicating a circle slowly dying because it was not visited for a long time), green paper (indicating correct move into a hoop), red paper (to signal a wrong move outside the hoops), paper with a blue "+1" sign (gained life for a hoop), paper with blue "-1" sign (lost life of a hoop). The relations between the cards and interactions were not explained beforehand, since we wanted to test how explanatory the light signals would be. Two other experimenters were walking around the room and observed the behaviour and actions of the testers.

The testers were invited to the play space and instructed to do whatever they wanted while looking at the signal cards. During the experiment, experimenters with cards would raise a card with the corresponding signal up and point to the hoop from which the feedback would be coming. After a few minutes of interaction, a rule was explained: if you touch the ground, you will get the red card. The interaction lasted for around 10 minutes and finished with an explanation of all the rules and intended interaction and gameplay. It was followed by an unstructured discussion with testers about their experience.

**Participants**
We asked a lecturer and a teaching assistant to be our testers, women in their 20s.

**Questions**

- Do the players understand the different colour signals on the LED strings?
- Are the rules for losing and gaining a life of a hoop (not of a player) clear?
- Would the players use the hoops in some other way than we presented them during the playtest?

**Results**
Some rules, like making some wrong move (stepping close to the outer part of the hoop) were clear and intuitively understood by participants, so red light is a good choice for the signal. The same could be said for the green light for stepping inside the hoop.

What was not entirely clear was why the hoops started the countdown when the players were not interacting with them.

The players had also a problem understanding some signals that were presented by the experimenters with the pieces of paper, leading to the players not knowing how exactly to play the game. They walked

from one hoop to the other when it changed colour, stepping in the forbidden range of the hoop in the meantime.

We have seen that the players were trying at first to figure out the rules on their own and once all of them had some understanding, they started working together to make the game easier for them. After explaining some of the rules of the system, the goal of the test skewed to testing the fun and experience of the play.

The testers found it enjoyable and fun. But they did not tell us if they would try to play a different game with the hoops than “The Floor is Lava” that was tested. They discussed the mechanics of including new players and brought up the topic of the number of hoops per player which we decided to focus on more and think through.

**Discussion of the Results**
Some of our ideas for signaling what was happening were understood and we will work with them further. Some ideas for signalization were harder to grasp (the countdowns) – the problem of their understanding was most certainly in the inconsistency of the signals displayed. The times for the response differed because no one was precisely counting the number of seconds until a change for a hoop happened. Also, the people displaying the signals sometimes did not see all the things that were happening, e. g. someone approached the hoop and triggered the signal for wrong behavior (you cannot step outside the hoops, in a very large range) – but it was not signalized, thus creating confusion for the players.

The main and seems like the only problem was the inconsistency with system feedback, as the experimenters did not use specific times for feedback and sometimes could not react immediately. This is, however, a problem that depends on the setup and thus we believe that in the real game, where timers are embedded in technology and not operated by humans this problem won't occur. Other than that, we believe that the results were as we expected, the players had fun and played the game mostly as the design intended.

We discussed the ratio of the number of hoops to the number of players, which was suggested by our testers. What would be the ideal number of the hoops and players? In conclusion, if more than one person were interacting with the hoops, the number of them should also increase accordingly. One of the ideas was for it to be around two hoops per person, so that it is not too easy to keep them all alive and not too hard. But we need to take into account also their placement and the difficulty of the game, so it is not so easy to say which would be the best solution to this problem.

# Concept Details

## Main Game Loop

After the user tests, we led discussions on the main game loop. We focused on how to ensure the players keep on moving to different parts of the playing field in the game “The Floor is Lava”. To ensure this, we needed to add some punishments: what happens when a player stays in a loop for a long time, or what happens if a player does not visit a hoop at all? Some counters needed to be added: how long can a player stay inside a hoop, how long a hoop can be left alone. We also discussed how to signal different states, by using two LED strips on the hoop. There could be different difficulty levels as well.

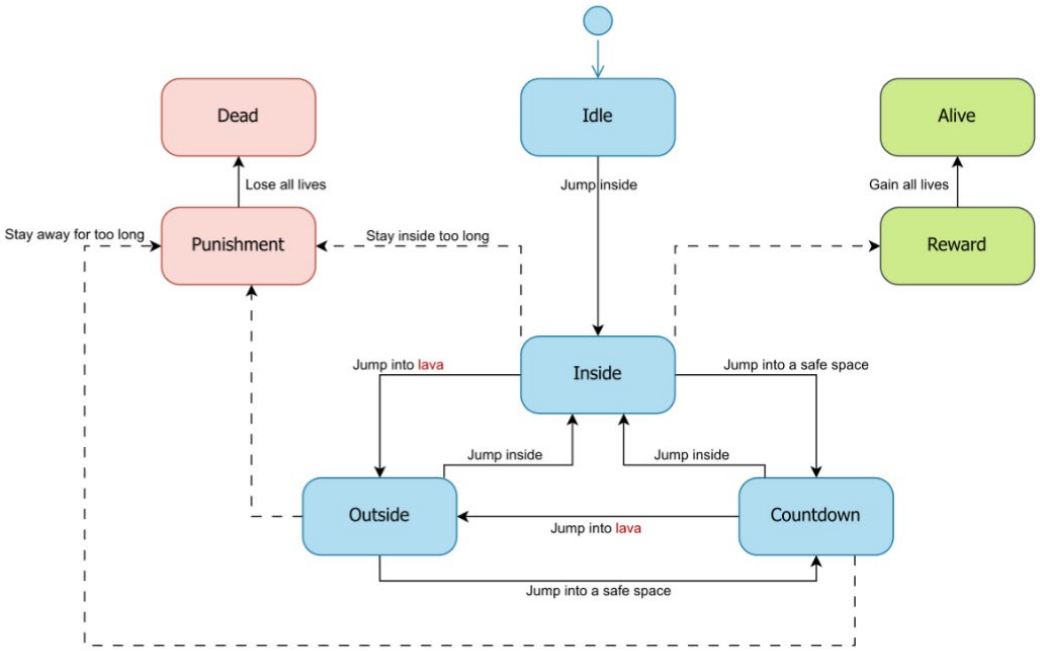



*Figure 5: Gameplay Design (simplified)*

There are a lot of variables that can be adjusted to create a game difficulty – times for the countdowns, starting lives of hoops, and severity of the punishments, to name a few. After designing more levels of difficulty, the next natural step would be to implement buttons that would allow players to switch between different levels of difficulty.

The easiest level is meant to be used by small children (4-6 years old) and the impossible is designed with an athletic, well-coordinated group in mind. Normal and hard should be fun and challenging enough for most of the users.

In Table 1 is a sketch for different difficulty levels that could be programmed into the hoop. The numbers of LEDs were picked up based on the divisibility of the total number of LEDs in one strip (30 LEDs).

| Difficulty | #LEDs for 1 life | #lives at start | Countdown in (s) | Countdown out (s) | No visit time (s) | Inside time (s) |
|---|---|---|---|---|---|---|
| Easy | 3 | 10 | 30 | 30 | 25 | 15 |
| Normal | 5 | 6 | 15 | 20 | 15 | 10 |
| Hard | 10 | 3 | 5 | 10 | 8 | 5 |
| Impossible | 30 | 1 | 3 | 5 | 3 | 2 |

*Table 1: Characteristics of Different Difficulty Levels*

To further describe the main game loop, we provide a description through different lenses of play (Schell, 2008) we worked on for week 4:

## Stages of Interaction

During the invitation stage the hoops are in some kind of idle animation mode. Its LEDs are lit up and create slow colourful waves that are hypnotizing and soothing. This invites the player to approach the rings and start the interaction. Later on, during the exploration stage during the play, it might not be entirely clear why different colours show up and what they mean, so the player may make some mistakes. The game is designed in a way that beginners can still have fun and not lose the game in this stage – the lives are set to half of the max at the beginning so there is a lot of room for experimenting. During the last stage, immersion, the players are already familiar with the mechanics and are trying to develop the most efficient strategies as well as just to have fun. Maybe they are interested in winning the game as quickly as possible or trying to obtain some particular configuration of the lights on the rings. It depends on them what they want to do and how they will achieve it (De Valk, 2015).

## Forms of Play

The interaction can be analyzed in terms of different kinds of play. First is physical play: the players must either jump or climb a lot to get from one space to another. It is very dynamic and fast paced. Secondly, there are some aspects of games with rules, mechanics that will at the end turn off the lights if the players perform or not perform certain actions in a specific time, so if they want to be able to interact with the rings, they need to keep the rules in mind. It has some social play elements as it is designed to be playable by more than one person, they can either play together and cooperate or try to make a small competition. Depending on who is playing with whom, there are different options of gameplay, and the players can choose it themselves. Lastly, the game is somehow a pretend play, it is like the game "The Floor is Lava" in which the players should avoid touching the floor. In this game stepping on the floor has bad consequences for the hoops (M. M. Bekker et al., 2014) (Bekker et al., 2014b).

## Rules of Interaction

The rules can be described in the diagram below. Each hoop has a timer. The players need to get inside the hoop without stepping into the range of the outside sensors (practically the whole floor is scorching lava). They can use obstacles above the ground for transportation. Once inside the circle, they can remain there for a limited amount of time. Once the time is up, they need to move to another destination, otherwise, the hoop's life will decrease. If it is on zero, it fully dies and stops glowing. When a player steps outside the ring, but inside the outer sensors, the hoop's life is also decreased.

The point of the game is to keep moving among the hoops (stepping only inside them) and keeping them alive.

## Parameters of Time, Place, and Social Action

If the object is located indoors, it can be used any time of the day – there would be no problem with the weather and visibility. If it is located outside, the play would take place during the day when the players can see clearly their environment, and the obstacles and not get hurt during playing (jumping to/from obstacles). The bad weather would be problematic for the object, because of its fragility (sensors, wires).

For the idea “The Floor is Lava” the players can play with it in a cooperative manner, together trying to keep all the rings alive and glowing, when they see some of the hoop is dying and they don’t have time to get there in time, they can throw something inside to gain more time, and then run to retrieve it back before another timer starts its countdown. Or they can play against each other, trying to sabotage each other’s attempts.

## Technical Setup and Implementation

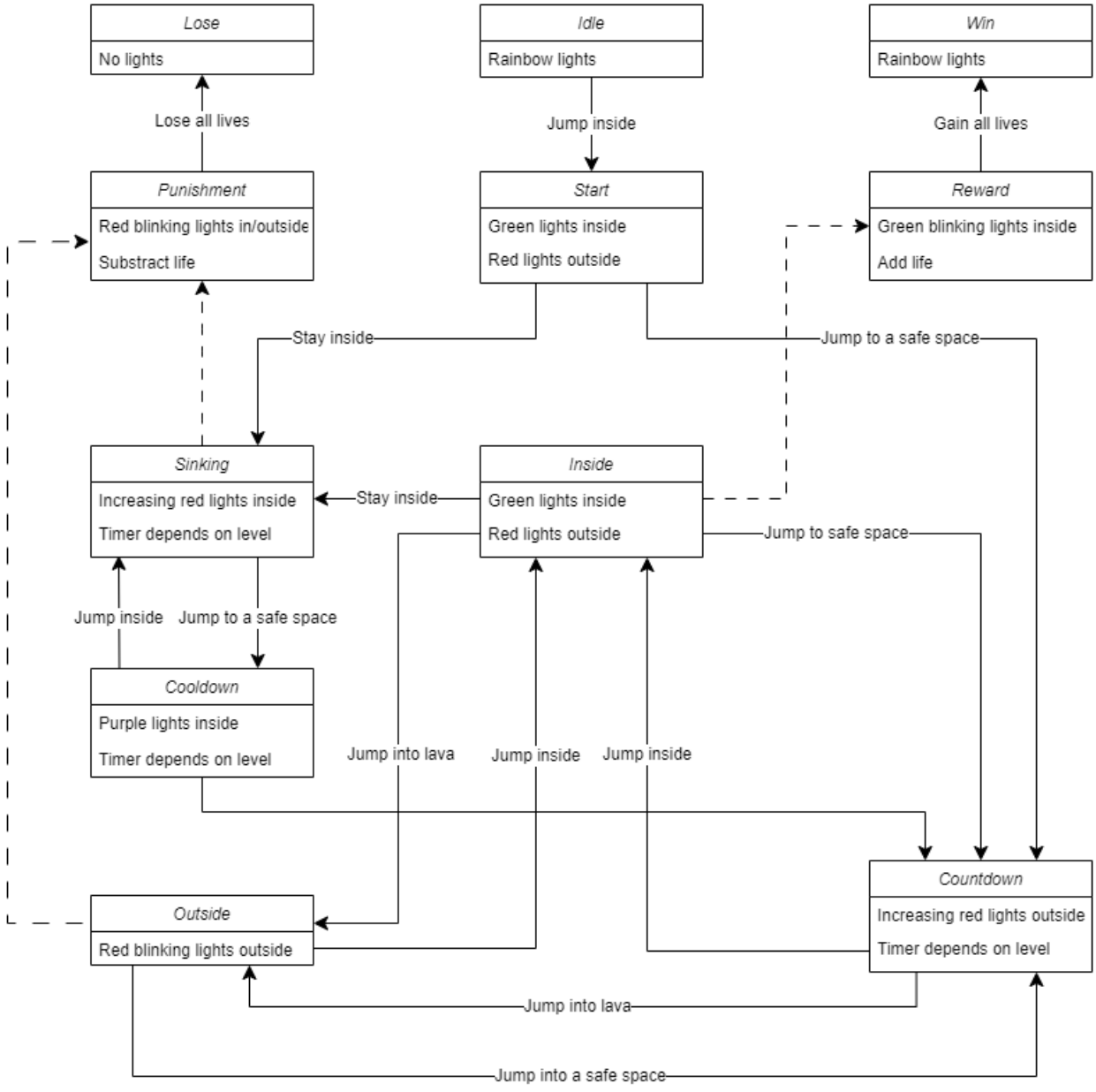



*Figure 6: System Schematics*

## Mechanical

The use case of the hoop required a firm and sturdy base, for which a rim of a bicycle wheel proved to be very successful. The rim is made from aluminum and comes in a concave shape, which makes it strong and durable. Bicycle wheels are abundant in the Netherlands and are not repurposed, only recycled, making them ideal for national scale-up ambitions.

The Hoop is equipped with two LED strips, containing 30 LEDs each and running on 5 V. On the outside of the hoop 4 motion sensors are equipped, these were selected for their detection angle of 100 degrees. Their output is binary and has a detection cycle of approximately 2 seconds. As immediate feedback is required it is recommended to source other motion sensors with a lower cycle time for future developments. The detection area is shaped as a cone, by placing an extension of the housing above the sensor their detection area was flattened.

The inner ring contains an ultrasonic distance sensor which has a very fast response time, but an angle of only 15 degrees. For the inside of the ring only one sensor is required, as all users step into the center of the ring, which is the detection area.

For the prototype, an Arduino microcontroller was used to control the whole system. This was housed in a box, part of the ring. For future development a smaller Microcontroller will be required, as it reduces the risk of anyone jumping on it.

In order to connect everything flexible split core 3 mm wires were used, which were connected using terminal blocks, only limited soldering was required which is generally prevented in prototyping for durability and sustainability reasons. The soldering that wad needed connected the sensor pins to some wires, these connections were isolated using shrink sleeves, as the aluminum ring is conductive.

## Software

The creation of the code was the majority of the work, as it had to execute the intended gameplay. The Arduino code contains parts which allow it to communicate with the sensors and control the actuators. So firstly, the functional code parts were written to make sure all sensors were properly read. Next to this the FastLED library was utilized to control the LED strips. Based on experiments, several animations were created to communicate the game states to the players. A few examples are rainbow lights for winning, fading in for warnings and blinking for punishment and reward.

After the outline of the gameplay was decided, a state diagram (Figure 6) was created, which in turn was written as a pseudo code. This consists of several while loops in which counters were built based on the millis() function. We prevented the use of delays as it shuts down the sensors, making the ring unresponsive. The code is built-up modularly by using voids, but no classes were required for this code. The voids allowed us to create more complex subfunctions like instance addPoint() and punishment() but also allowed us to start other game loops during the while functions.

Contrary to many Arduino codes, this code was built in a way that the void loop() only runs one time. After this, the game keeps cycling between while loops, in whatever state it is. Even when the ring dies, the sensors keep running, so that the ring can be revived.

The main part of the code is attached in Appendix A6. - Code, this includes the set-up(), void loop() and the void main() functions. We left out the sensor read out code and the animation code, since these do

not contribute to the understanding of the concept. The total code consisted of 1.192 lines and could be improved upon. However, a more modular approach was chosen in the construction of the code, therefore the code is longer, but it is easier to adjust parts to finetune the gameplay after testing.

## Conclusion

To summarize, the process of designing and creating LaserHoops was not straightforward from the beginning. From the idea of a talking garden gnome, through falling fruits from the trees and tile paths we arrived at the hoops through a creative idea generative process and a lot of iterations of previous ideas. There are still some rules and functionalities that we should think of and improve the final product, but we believe that the current version is already sufficient to provide playful interaction. The rules are important, however since we were designing with the concept of free play in mind, they are not the most crucial in our belief. Nevertheless, it will be good to implement functionalities that would focus more on social aspects of the interaction, such as including the audience, new players joining, or a passer-by stepping into the range of the play. Those directions need to be explored further and tested to find the most optimal way of implementation for future features.

Players are given enough autonomy via LaserHoops while yet receiving the necessary game structure for comprehension. There may need to be an extra restriction. By placing pressure sensors on top of the hoop, undesirable behavior may be stopped, and settings control could be added. Allowing players to modify the pre-set difficulty levels may boost autonomy. Enhancing the challenge-achievement balance, changing the timeframe, point system, and sensor sensitivity will help to broaden the scope of competence that can be created. Allowing PVP gameplay would make this even better because it would make the player experience more relatable and encourage social gameplay.

No specific number of hoops per player would be ideal to keep players in the flow of the game because, as was already mentioned, there are many variables at play (Csikszentmihalyi, M., 1975). The game might be disrupted if an audience member joins in the middle of it without increasing the number of hoops. The potential players would have to express their interest in playing the game in some way (e. g. pushing a button or going through some gate). Hoop deactivation may occur in conjunction with exiting the game (pushed button, exiting via the gate) to maintain a similar level of difficulty.

According to this analysis, LaserHoops' concept has some real potential, but there is still room for development. If it is possible, the design will be evaluated to see if the essay's insights may be used. The hoops are made to resemble the game “The Floor Is Lava,” but they may also be used as basketball hoops, obstacle courses, and a variety of other activities because they are made with open play in mind. This article just examines one of the various interactions that can happen when utilizing the hoops (De Valk et al., 2014).

## Future Works

That the LaserHoops concept has potential for further development stands to reason, however the highly positive responses we have received from students and university staff was unexpected. It shows that the nature of this project holds serious potential for further development.

Further development can be perceived from several angles, conceptually we could improve and expand the gameplay on the hoop. Since the hoop offers an open platform, its usability can be wide and diverse. A few examples would be to add the element of team colours, allowing more players to use only a few rings. Another example could be to use the ring for a twister like game, to place the rings vertically in space, or on top of something. By creating a modular software platform for the rings, the versatility of the hoops could be explored in a playful manner. This would require some buttons to navigate between the modes of gameplay. This would also allow us to play with different hoops in different modes of gameplay. Imagine game modes like a watcher ring, a goal ring, sinking rings, rescue rings, guardian rings and so forth.

Another angle with which the future works can be considered is the business model and production of the product. For production a first design, which would be low threshold in development, would be to integrate the electronics into a double-sided Velcro strip. This not only allows us to transform bicycle rims into play objects but opens the field of application as they can be attached to any shape. This way we can get creative and transform playgrounds, furniture, and whatever else you can imagine!

From a business point of view this production strategy would allow us to start low-cost small-scale production and explore a multitude of market introductions. Interesting starting points would include schools, student associations, conferences and gameshows. Through slow market introduction and brand building, larger scale production could be considered which would make the product available for the consumer market. By offering the user access to the gameplay modules, users can “hack” the rings and explore their own gameplay. Allowing us to crowdsource development and explore use cases.

A technical challenge which would amplify the hoops potential is the ability to sell the Velcro strip in long strips which can be cut to size based on the use case. Several technical issues arise when considering this option. However, when this becomes an option, the already wide range of use cases for this platform would be even further expanded.

A potentially interesting side-track is the potential of the platform to contribute to games and play research. By offering the platform to game design scholars we can further explore physical gameplay development.

So concretely, future steps would be to safeguard IP, source production partners, shape a business plan, find capital, design the v2 hardware, build the modular software platform. After reaching the milestone of a scalable proof of concept, the focus shifts to branding and field testing on larger scale with school and on events, generating user insight for parallel improvements and gameplay building. After redesigning and engineering, a scalable market introduction might become financially viable. Looking further ahead we can start to consider hiring personnel, setting up distribution channels, switching to a product as a service business model, subscription models, licensing IP and all that good stuff.

But for now, let's just state that we believe that there is a future for LaserHoops!

# References


Bekker, M. M., Schouten, B., & De Graaf, M. (2014). Designing Interactive Tangible Games for Diverse Forms of Play. *John Wiley & Sons, Inc. EBooks*, 710–729. https://doi.org/10.1002/9781118796443.ch27

Bekker, T., De Valk, L., & Eggen, B. (2014a). A toolkit for designing playful interactions: The four lenses of play. *Journal of Ambient Intelligence and Smart Environments, 6*(3), 263–276. https://doi.org/10.3233/ais-140259

Csikszentmihalyi M. (1975). *Beyond Boredom and Anxiety.* San Francisco, CA: Jossey- Bass Publishers.

De Kort, Y. A., & IJsselsteijn, W. W. (2008). People, places, and play. *Computers in Entertainment, 6*(2), 1–11. https://doi.org/10.1145/1371216.1371221

Deci, E. L., & Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior.*

De Valk, L., Bekker, T., & Eggen, B. (2014). Drawing up the rules: Encouraging children's rule creation in interactive open-ended play. *International Journal of Child-Computer Interaction.* https://doi.org/10.1016/j.ijcci.2015.07.002

De Valk, L. de, Bekker, T. and Eggen, B. (2015). Designing for social interaction in open-ended play environments. *International Journal of Design, 9* (1), 107-120.

Hunicke, R., LeBlanc, M., & Zubek, R. (2004, July). MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI* (Vol. 4, No. 1, p. 1722).

Korhonen, H., Montola, M., & Arrasvuori, J. (2009, October). Understanding playful user experience through digital games. In *International Conference on Designing Pleasurable Products and Interfaces* (Vol. 2009, pp. 13-16).

Reeves, S., Benford, S., O'Malley, C., & Fraser, M. (2005). Designing the spectator experience. *Human Factors in Computing Systems*. https://doi.org/10.1145/1054972.1055074

Ryan, R. M., Rigby, C. S., & Przybylski, A. K. (2006). The Motivational Pull of Video Games: A Self-Determination Theory Approach. *Motivation and Emotion, 30*(4), 344–360. https://doi.org/10.1007/s11031-006-9051-8

Schell, J. (2008). *The Art of Game Design: A book of lenses.* CRC press.

*Share and engage with the Design Sprint Community.* (z.d.). https://designsprintkit.withgoogle.com/methodology/phase3-sketch/crazy-8s

# Appendix

## A.1 - Individual Reflections

**Kasia**

In my opinion the course was nicely designed, and all the parts fit nicely together. After participating in the first two courses from the Design for Games and Play package I was expecting again to have to build a game and so I was not prepared to build a physical thing that would have to provide some playful interactions. I think that this course very well shows what a process of designing a real product looks like and how to get through the process so that it is the most successful at the end. Although I feel like knowing how to make physical products will not help me in my future career as a game designer, theories introduced through the course will come in handy in game design. Also, knowledge how to analyze existing projects is something that I learnt in the course and I think will be useful in the future, either for self-assessment or by analyzing some other existing projects and gaining insight and inspirations for my own.

In terms of a group work I think it was nice, rather good division of tasks although the person who had to have the object at their place and tinker it did more than the rest. There was rather clear communication and the fact that we had (mostly) Thursday timeslots reserved to work on this project helped with organization and schedule planning. I feel like not many things here could be improved. I feel like we were wasting some time at the beginning trying to develop ideas that we abandoned later but I know it is a part of design process and it would be impossible to have a perfect idea from the start.

**Joris**

I selected this course because I wanted to extent my knowledge of integrating Game Play theory into product design because Play has the amazing capability to nudge behaviour and make thing highly educational while reducing cognitive load. I did get what I aimed for, and much of the theory I was looking for was presented to me. Next to this, being an industrial designer, I was enabled to work with game designers from a more computer science background. All my peers had already studied much of the games and play theory, which allowed us to incorporated this theory into the projects quickly, but also prevented me from getting more deeply acquainted with these theories.

Next to this I was the only one acquainted with the design process, which proved challenging as the process of my peers was more linear compared to the dynamic design processes. As such I acted often from a more guiding or leading role in the team, improving my skills of going through a design process with peers from other disciplines. I attempted to offer everyone an equal amount of creative space, but experienced that idea generation is not as natural for others as for me, I wish to learn how to facilitate this better, because now many of our discussions revolved around my ideas. This is probably due to my sketching skills and experience in idea generation, but I know others have probably even more valuable ideas, as they more knowledgeable than me in their field knowledge

Being the designer I naturally was tasked with the responsibility of building the physical prototype. Since this also incorporated the alignment of the sensors, I turned to coding, which after I played around with it, also turned into my task, however happily supported by Lisa. Due to these circumstances, I took upon myself quite a workload, and in the process, limited the access to the prototyping skills for the others. This is also a learning for me, I should be able to led go of control, to give others the opportunity to learn.

All in all we had a great project with amazing results. But looking back I should have distributed the work better. Either being able to let go of control, or to push more of the work towards others. Next to this I should have pushed harder to get out of my comfort zone. I did learn a lot more about coding, which I was relatively unfamiliar with, but I wished I had absorbed more of the game theory.

**Abi**

I attended several different courses on game development in Charles University in Prague. This course was quite different. There was both some introduction to theory on player experience and praxis. I was used to creating parts of video games for school, but I never tried to create something tangible. It was great that we managed to build a working prototype in eight weeks, including a thorough ideation process. I enjoyed creating the ideas and thinking about the various pitfalls of each design. It was surprising for me how many great ideas we came up with.

I was quite looking forward to learning with Arduino and sensors, but in the end, I didn't really get to it. The prototype was at one member in Amsterdam and sometimes getting to Eindhoven wasn't doable for him. This brought some problems with the code writing and debugging, so unfortunately most of this work was left to him. In the end, I took upon myself reformulating our weekly assignments into this final report.

The team members were friendly, hardworking and they were open to discussion. I really liked working on the project and I think that I learned some new approaches to how to design a product.

**Lisa**

I took this course after DGPI and DGPII, and I expected it to be similar to these courses. I did not foresee that we had to make a playful object, using sensors and actuators. However, I enjoy learning about this, and I immediately set goals for myself to learn how to make a nice prototype and write code for Arduino, since these are both things that I was very unexperienced in. Besides that, I haven’t done much user testing for courses before.

I really enjoyed that the course offered a lot of time to work on the project with our group during the lectures. This made it easy to ask for feedback, and made the course interactive. Besides that, we had a lot of interaction with the other groups, which automatically led to feedback and new ideas.

Next to applying a lot of relevant theories from sources provided by the course, I think I also achieved my goal of learning to code for Arduino. I wish I would have been a bit more involved with building the prototype and figuring out how the sensors work, but that was also a logistical challenge. I do feel that working together with Joris on the code was very useful. As for the user testing, we got offered quite some information on that, which we applied for our own user testing to gain insights in our project.

## A.2 - Iteration 1

Compilation of concept descriptions from weeks 1 and 2.

### Singing Fruit Concept

Small balls or light bulbs are attached to tree branches (on strings or magnets). Let's call them fruit. The fruit can randomly light up or produce sounds like bird chirping. Each fruit has a random timer that determines how long the fruit can stay on the tree without falling down and getting bad. The fruit ripeness is determined by either its light colour, or the light's intensity. Once the timer reaches zero the fruit descends or falls down. The fruit is attached on something that also indicates the place where the fruit was at the first place and can be reattached. The player tries to catch the fruit before it completely falls to the ground and put it back on its initial position. Once the fruit is touched, it lights up again – restart of its life cycle. The player can also touch fruit on the tree branches to make it sing, and to restart the process of growing (before it starts falling) or squeeze the fruit to deactivate it. There could be also a display in the middle of the garden counting how many fruits one was able to catch successfully.

This object can make the player feel competent and let them achieve mastery in this reflex game. The impact of the user actions will be clearly visible in the environment and make them feel in control. This experience can touch upon the feeling of competence and challenge. It also allows the players to express themselves by playing sounds through the fruits.

Potential sensors: touch sensor, force sensor, timer
Potential actuators: light, speakers, magnets

**Mechanics**
Light bulbs in the shape of fruit are hanging on strings from tree branches at various heights. There is a game version of the object, and an interactive and more artistic version.

For the game version, each fruit randomly starts glowing and humming, according to its timer. As the time goes on, the urgency level increases – the singing/humming is louder, the light is more intense, and it changes colours, from green to yellow, orange, and then red. After ripening, the fruit starts falling. When a player touches the ripe fruit, its timer stops, and the fruit stop falling. It plays a tune for a moment, and also maybe returns to its original height (or it is up to the player). If the fruit is not caught in time, it stops its fall at a low height, so it doesn't break, remains in alert colour, and after some time starts returning to its original height while fading out its light.

In the artistic version, the fruit is silently waiting for touch. The touch wakes it up and it starts glowing and singing a tune for a few moments. The fruit glows while the tone plays: both will fade in and after some time fade out - the speed of fade-in can be proportional to the touch or size of the affected area. There is a touch sensor on each fruit. Each fruit has an assigned tune and colour of light. For a quieter version, the tone fades out much sooner than the light goes out.

**Dynamics**
At the game version, players would try to touch the fruit before it starts falling. If the fruit is at an unreachable height, they would try to catch it as it falls or jump up to get to it. The players also can imagine the falling fruit being missiles and try to dodge them.

At the nongame version, the fruit starts glowing after human touch. Because of that, people would gather under the tree branches after dark and touch the fruit to light the space. The fruit would always produce the same tones, so people could interact with it to create music.

**Aesthetics**
The light bulbs are of the shape of one type of fruit (e. g. fig, peach, pear, or apple). It could also be something more artistic and imaginative. There is but an important limitation – it should be without sharp edges because of the aspect of catching the falling fruit. Each fruit could be a bit different in size and shape, but it should fit nicely into a palm. The colour palette of the fruit lights would be more natural: green, yellow, and red, maybe purple. The tones of the singing would be clear and ethereal, like the sound of a harp or flutes.

We want to encourage people to experiment with music and give them a moment for relaxation. They can relax by walking around and touching the fruit or playing the quick reflexes game while forgetting their worries for a few minutes.

## Gnome Concept

Remember Tamagotchi? The toy was a worldwide spread trend teaching us skills about caretaking. With the technology of today the principles of that simple but captivating concept can be used to build autonomy in the competence of caretaking.

The caretaking skill the Tamagotchi concept teaches us perfectly aligns with the skills required to take care of plants. Real life plants with their need for periodic attention are a challenge for many, but the reward of success is a house or garden full of flourishing plant life. To learn and grow our competence, we need to discover how to nurture and improve the wellbeing of our plants. But plants can't talk so how can we learn about their needs in a fun and interactive way?

We introduce a new friend with green fingers, your personal garden gnome, a funny little creature originating in folklore. What if your ceramic little pal would keep an eye out for you? He will help you to water your plants and to discover how to nurture your plants.

Imagine that he can inform you about the rainfall and sunlight received, he can tell you about what your garden needs in which season and how you can deal with pests, and when a plant needs to get a larger pot. And he has a serious temperament, and his pointy hat turns red if you don't take proper care of his home. But if he is happy, you might catch him humming songs. The gnome is accompanied by a water reservoir which he will use to water the plants if they need it when you are away for the weekend.

PLEX: challenge, discovery, humour, nurture
Self Determination Theory: competence
Potential sensors: microphone, moisture sensor, light sensor, timer, infrared proximity sensor
Potential actuators: speaker, light, water pump

**Mechanics**
For this concept we imagine a ceramic or polymer gnome in which a microcontroller is positioned. The gnome is accompanied by a water holder which is connected to a water pump. The water reservoir can be easily filled with water and the gnome is waterproof.

A light sensor is positioned in the Gnome's eyes, and out of his feet a pin intended to be buried in the ground sticks out to collect data about the moisture levels. The pointy red hat contains a light which starts to glow if the gnome is in need of attention. A water pump will transfer water from the reservoir containing water to the plant if the moist levels are low.

The intelligence of the gnome could be regulated with the integration of GPT3. By using a prompt, engineering a personality will be created for the gnome, and the AI would also be restricted in the output the prompt gives. As such, only garden-related topics can be discussed, and personal interaction will be structured but influenced by the "mood" of the gnome. His mood is based on the values from the sensors. With low moisture levels, his mood is grumpy, but when the moisture levels are good, the gnome is friendly. The data from the light sensor also influences the gnome's behavior - sleepy, awake, or demanding shade in too much sunlight.

The Gnome activates each time the proximity sensor is activated. When activated, he will try to get attention by whistling or calling out. By pushing on his hat, he will give an update on the situation, then he finishes his comment by asking if he can give you any advice. By pushing twice on the hat, the gnome will skip his comment and directly move to the question. This is the only moment the microphone is activated and records.

**Dynamics**

The interaction with the Gnome consists of being alerted by asking for your attention when in need of water. He will attempt to start a conversation with you, and it is up to you how you engage with him.

The more attention you give him, the friendlier he gets. Failing to meet his needs will result in a grumpier character.

**Aesthetics**

Visually the gnome looks like a traditional garden gnome, a funny playful look which is common in the regular garden gnome.

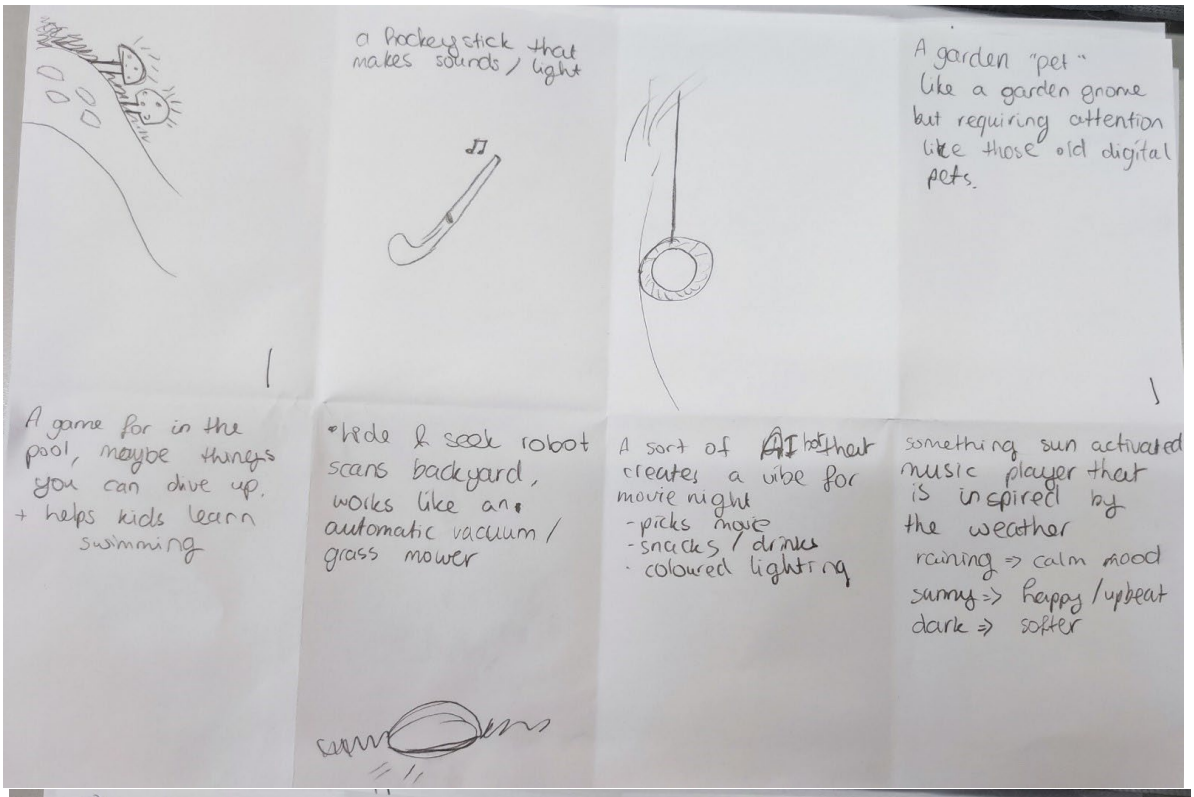
The character of the gnome is the most essential, the interaction with him gives him personality and is shaped in a way that is funny and motivating. The friendly side of the gnome is one in which the gnome shows how content he is with life. He shares motivational comments while using garden metaphors and makes garden and gnome jokes.

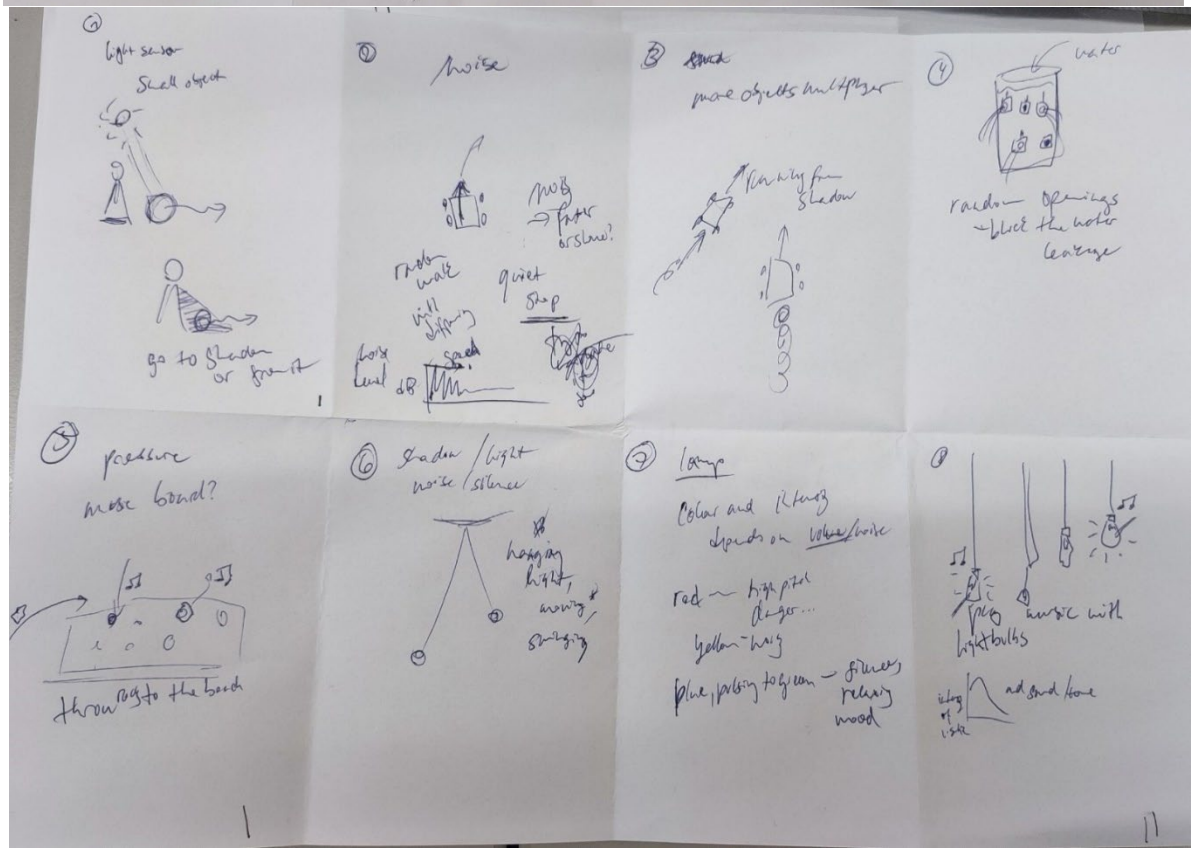
The Gnome acts like an old wise gnome who is sometimes forgetful. This trait can be used to feed the system data about the plants he is taking care of, and the surroundings he is in (e. g. indoors, outdoors, greenhouse).

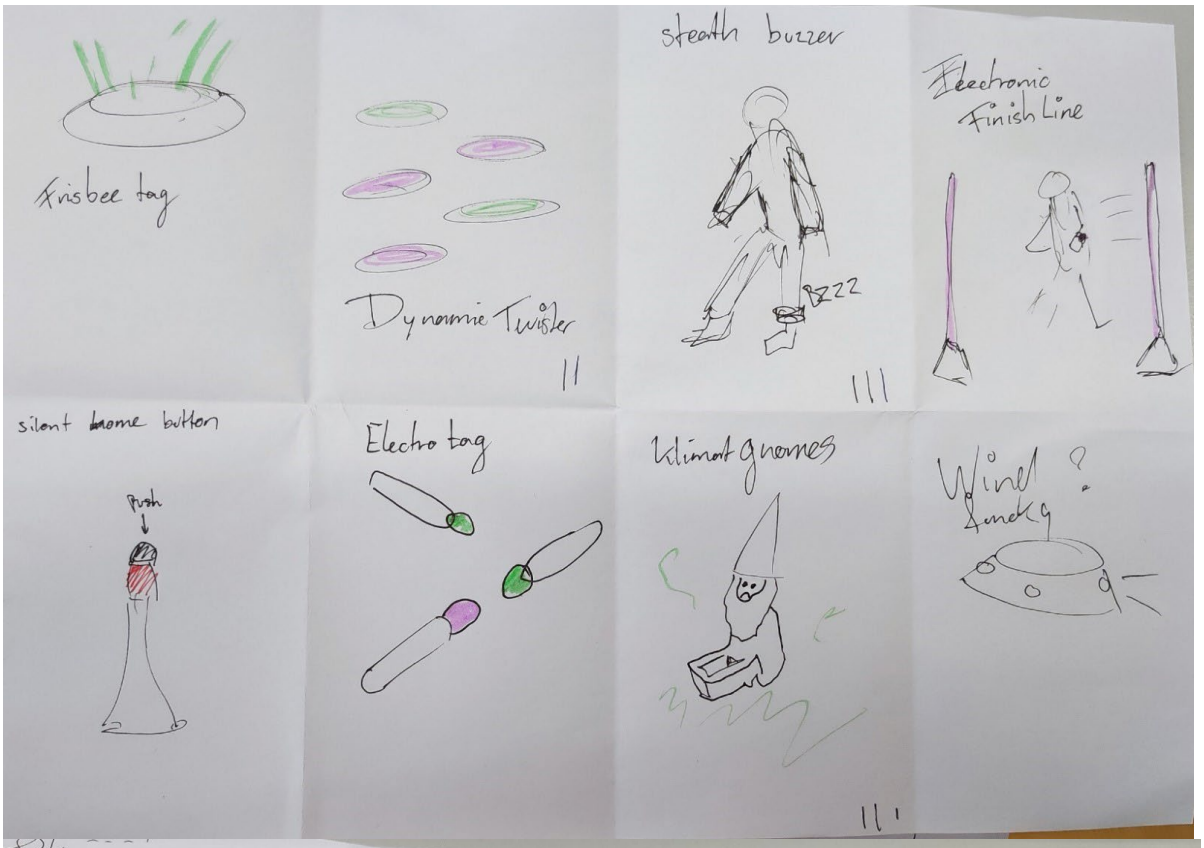
The grumpy side of the gnome is respectful but clearly reflects his discontent with the situation. He shares suggestions about how to improve his situation with references on how it was done back in the day and that suggests the superiority of gnomes compared to humans.


*Sketch of the Gnome*

## Sketches from Crazy 8's






escaping ball
- distance sensor
- wheels / ball

you have to catch

buttons that light up some shapes on the ground when you step on them like tiles (maybe in specific shapes)

I

the same as the previous one but playing sounds instead of lightning. (similar to piano stairs but maybe here is birds' songs)

lightning up a path you need to walk through (& set up a timer)
↳ racing game

II

1
2
3

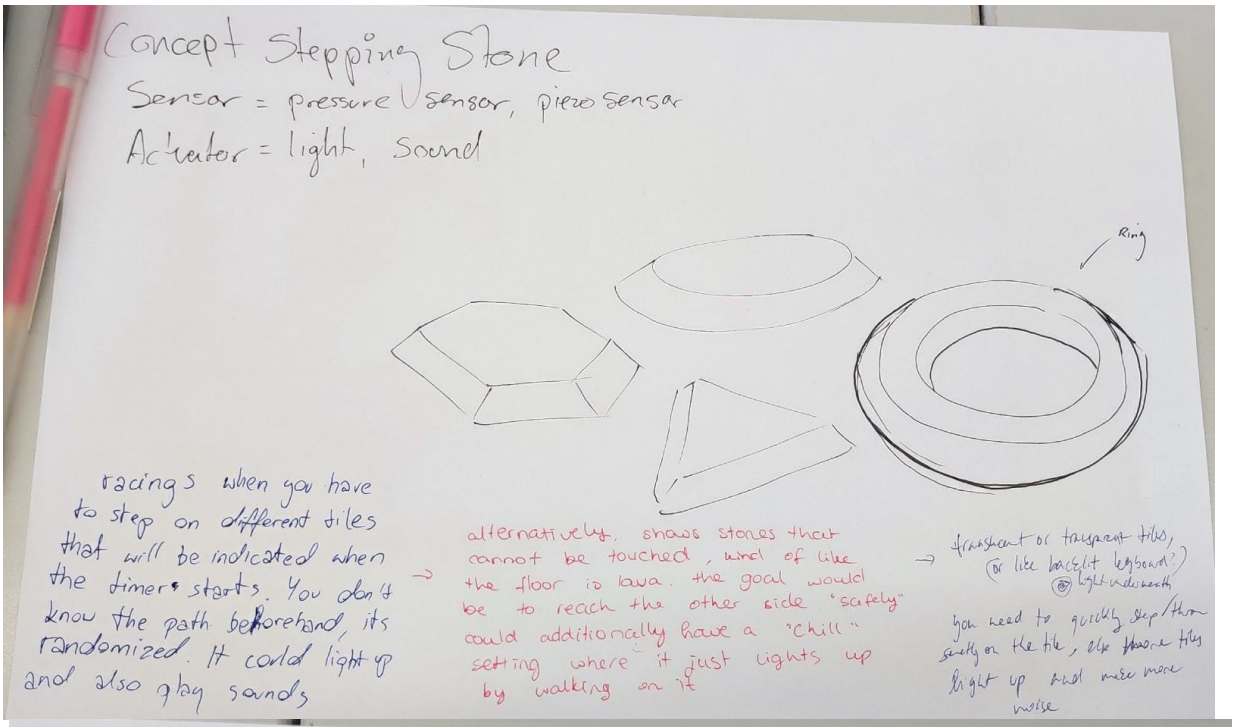
falling things on a wire and you can predict which one will fall because then will fade
(gasnare sviatca = daisy chain)

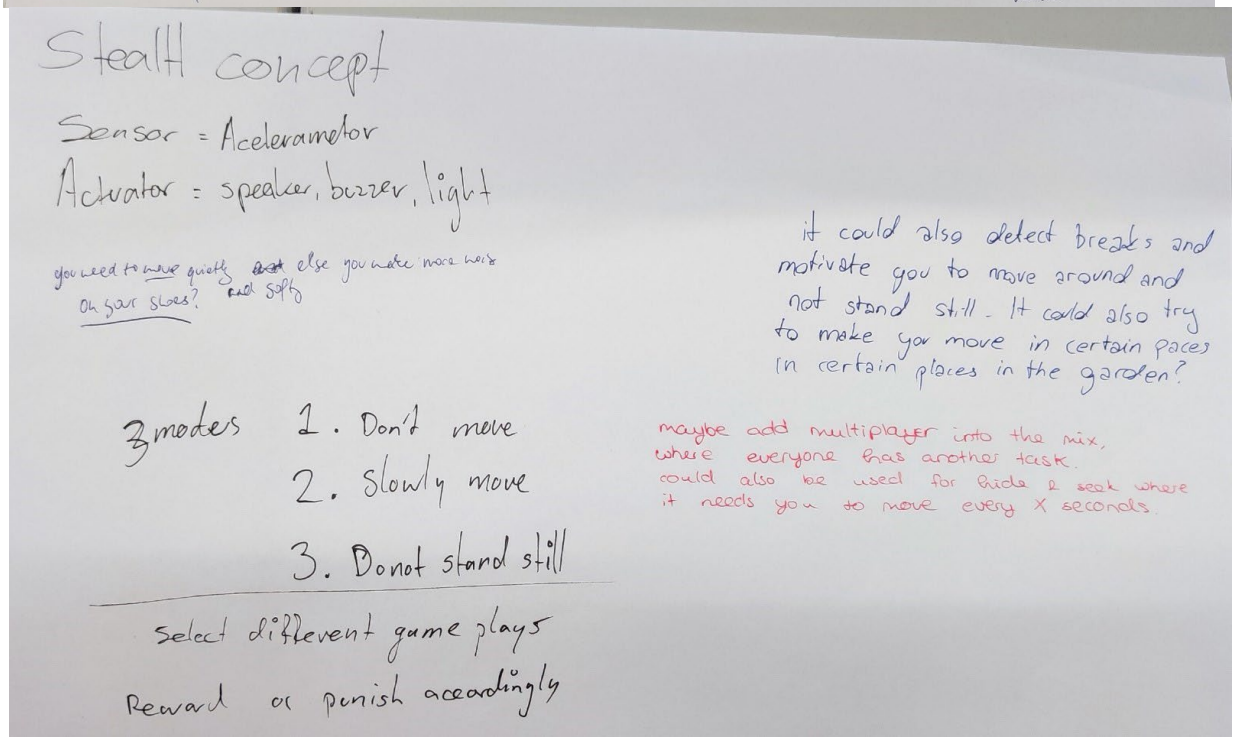
birds singing on the while you walk forward the path and cross lasers

I

like a small robot that follows you (circle around) or a ball

## Sketches and notes to the Four Initial Concepts



Concept - Singing Fruit

Sensor =

Actuator =

- they drop down when they stop singing / lights go off
- Idea is to catch ~~them~~ as many as you can
- 2 modes: • touch the fruit lamp → it ~~lights~~ lights up for short moment + makes a tone — and fades away
  • party strobing — lights up and you need to touch them to turn them

1. Squeeze or hit the fruit when they start singing, Drop it activated
2. Touch the fruit to make them sing

they go up and down on springs ~~so that~~.
also they can sometimes randomly start singing or light up without interaction (so sometimes they reset automatically)

Concept Gnome

Sensor = proximity sensor

Actuator = speaker

Playfull gardning

sense drought

garden keeper - tells you about water level in the garden
it also is happy when you're walking around and miss you when you're not going to garden frequently (could be days but for testing its seconds/minutes)

has a "personality" that is affected by multiple things, some need user interaction and other are based on environment
could have music / lights reflecting its mood. uses a small display

Grumpy Gnome - can have a little conversation with you ~~[illegible]~~
→ more social contact :)
- reacts on pat on his head ~~[illegible]~~

## A.3 - Iteration 2

Pictures of our ideas from the second Crazy 8's. The stars on some ideas are the residue of the power dotting method.

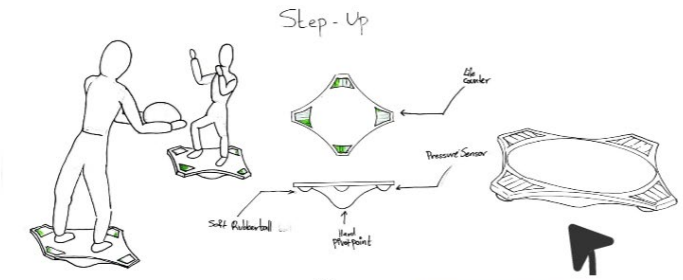
### Ideations on the Stepping Stones Concept

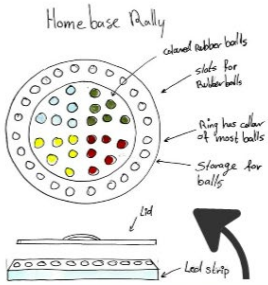
memory tiles - tiles light up in a pattern that dissapears after 20/30 sec. The player has to go over the tiles that were light up as if it tries to recreate the pattern, if they win, the tiles light up green and afterwards, next pattern is created, otherwise they light up red. (the order matters)
1.5) alternate mechanism is to replace or add sounds to the tiles, and the player has to recreate the melody

racing game - certain tiles light up, creating a path, the path is slowly fade away. The player has to run through the light up path faster then it fades away. here one can also step on different tiles, what matters is to touch all the tiles that were light up before the end of the time

"catch me if you can" - one tile light up and stays in that state until it's pressed, after that the next tile is light up and the player has to wander through the whole garden to find them and click ( could be also time restricted) that you have only 10 sec to press it, otherwise the light "moves" to a different tile

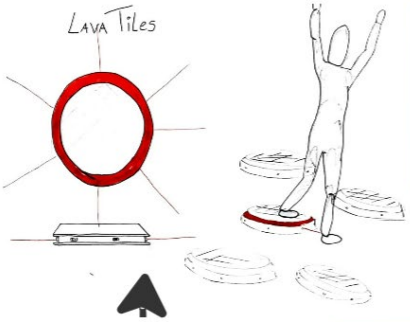
floor is lava - most tiles are light up and you can move only on the ones that are faded out



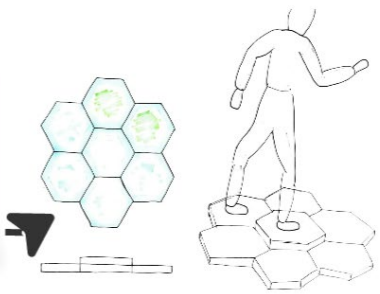

Step-up all your normal games and take on the challange of balance. The balance board gives you 5 lives on each side of the board, each time you lean into one side to much, you loose a life. The board can be used for games, but is also suitable to gamify sports.

Homebase Rally - an induvidual or team has to collect and bring back all the balls of one colour. But how you do it is up to you, search and find? Catch me if you can? Or is it going to be a race? Let your fantasy go wild with homebase rally and discover new gameplay every time!




Lava Tiles - These stepping stones are your safe haven in the dangerous game of the floor is lava! Each stone lies in the middle of pool of lava and if you step into it the tile scream in horror (and signal red). The gameplay is versatile, as you can integrate it in many forms of role play, indoor and outdoor. Only one tile will allready turn a playground into a lava pit, but many will increase the fun more and more!

Sinking Lands - this watery floor will quickly become very challanging, as the islands you stand on sinks when you stand on them. The longer you play the quicker the sink. How long can you last?



Use the stones as home bases. A player needs to return to during a game. Spread them out over the backyard / playing field. The stone will lit up when it is time to return, and fade within x seconds. If the light is of, you are too late.

Make them interactive, the stones can be connected, so if one person walks on them in their own backyard, they also light up in the same pattern in de connected backyard.

Walk as fast as possible over a path of stepping stones, without touching the ones that are lit. This could be done by multiple players. Saves the high score.

Combine them: stepping stones are the home bases. There are two teams, that play capture the flag, they have bands on that indicate when they can move, stay still, or need to return home (if they have been around someone who has a higher level, or if they cheated).

on the floor, there are stepping stones set in a rather arbitrary setting (not a grid). you hear a sequence of tones (or sounds). when you step on a stone, it produces a particular tone. if you step on the same tone from the sequence it lights up a happy colour. else it glows for a few seconds, then all stones that you already had glowing turn off and the sequence plays again. when a sequence is completed, another random tone sequence can start playing. each session has different tunes assigned to different stones. the players will learn the step tones in the first shorter levels and further in the game the sequences will be longer.

the stones are placed not in a grid. players can record and repeat sequences. there are two repeating rounds:
1) initialization: step on the stones in some order
- as the last step, step on a special finish stone to change the turn
2) repeat sequence: the stones will quickly light up in the same recorded order. players have to pay attention, remember it, and step on the stones in the same order

twister. lot of stones in a grid - or not (more natural placing).
1) select the number of players -> number of colours (e.g. R, G, Y).
2) a stone lights up with a colour of the player on turn. when the stone is stepped on, another stone lights up with a player on turn's colour.
3) the system needs to keep lighted up only up to 4 stones with the same colour (or teams?). time to time, a stone will dim and a player is allowed to remove their limb/weight from it

mine field. tiles in a grid.
try to get from one end of the grid to the other side. the intensity of the light from the stones around you (in 8-neighbourhood) indicate a probability of a mine under it. players can move however they like but in the rest of the grid under the fog of war for them.

tiles, several colours for number of players. random stones light up - players have to run to their own tile, after stepping on it, their light moves to another tile.
race: who will complete X tiles?

## Ideations on the Stealth Concept

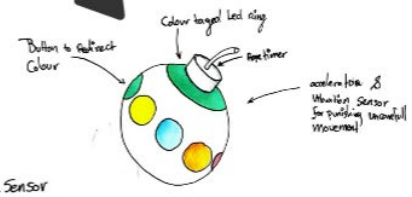
multiplayer game of "green/ red light" and the band will indicate when you can or can not move. It will be indicated by the band vibrating and/or change of some small light in the band or in the garden

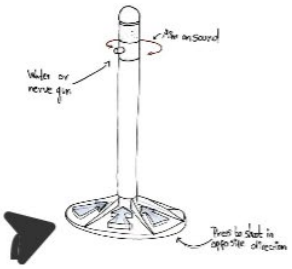
certain areas in the garden are so called "zones" and depending on current color of your band you need to be in the same color zones "so if you have a green light band currently that means you need to run to the green base", color bands change randomly so maybe if it takes too long someone can't manage to reach the correct zone in time, then the band will fade and somoene has to wait for a few minutes to be able to join the game again

an invisible maze - you don't know where you can go, but there are invisible walls in the garden, you have to walk from one side to the other and the buzzer informs you if you are following the correct path or if you are standing in front of a wall

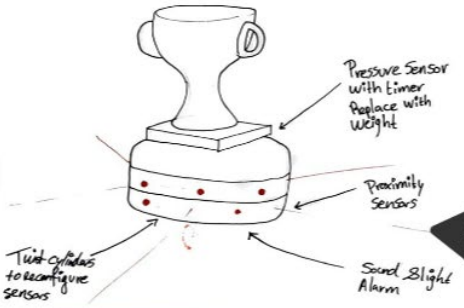
hot and cold game, similar to the idea above, but now you don't necessary know where you are going and looking for pathway, but you are looking for the destination, the band is lightning up every few seconds/minutes and based on where you were previously and where youare not, it will either show you that you are closer to the desitnation or more far away, that way you are slowly moving through the garden trying to find "a treasure"

Every player presses one colour button, then pull out the timer chord. The game starts. Every time the bomb shows a colour the person who owns that colour will have to push his button. The count down is then paused for a few seconds, each time a little shorter. If the timer runs out the system will make the sound of an explosion and the last colour shown loses.



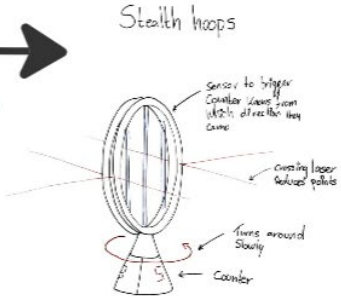




Better be quiet, because the machine might shoot you down! This turret is self aiming on sounds but if you approach it silently enough you can also use it to shoot by pressing on the pedals on the floor. But remember to remain quiet even when you hit someone else, because it might turn his back on you!




Each game needs a treasure, but every game has a different treasure right? When you find the treasure, you will have to slowly and carefully approach the price. But you can't just grab the price, you quickly have to switch it out with something that has comparable weight!

Playing laser hoops is not easy! As the ring turns around to need to pass a ball, frisbee or yourself through the hoops without stepping into the lasers! Play against the game, or against eachother!




Stealth Tag, each player wears a harnas, and each time the button on your back gets hit, you have too slow down a little. When you move to fast, the system will let you know by making noise. The last one walking obviously wins!

Hide and seek, but the seeker can see who is moving, but not where. They have a device that shows them the person that is moving, making it easier to spot them.

Capture the Flag: people have different levels, and the ones with the higher level are allowed to move more, they can more easily take out players with a lower level, but are not allowed to leave their own field.

You can only move when your bracelet is making music, and you need to be completely still when it is silent. Players can race who reaches the finish the fastest.

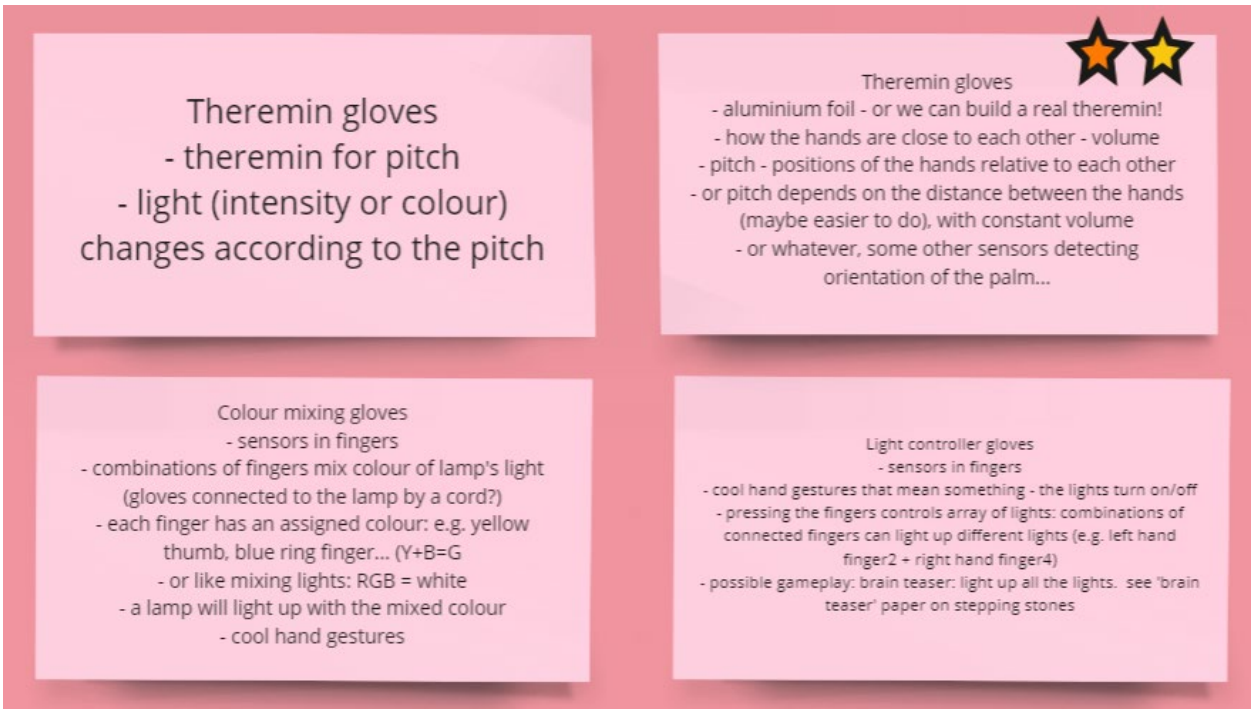
catch me if you can - burglars and police
- each player has a bracelet that tells them how to move through the colour it emits
- some grace period: few seconds to reorient/stop themselves (blinking light)
- if the player does not move according to the light (after the few seconds when it glows steadily), the bracelet's light turns off -> not an active player
- burglars try to escape the police who are chasing them

brain teaser: translucent stones in a garden - with aluminium foil :). when you get near to it it starts glowing. but some other stones can also light up. and some on the other hand stop (bit flip). try to light up all the stones - or turn them all off. - find the code to turn the lights off!

multiplayer mode: catch me if you can while not getting disqualified. you can move only just some speed, do not go faster, and do not do sudden moves. group of people is trying to get away, the rest is trying to catch them

translucent stones in the garden as guards. you can move them to form different patterns, to different parts of the garden. try to sneak around them, quickly run around them without disturbing them. be on the lookout. if you are near them for longer time, they will start glowing.

### Ideations on the Glove Concept




## A.4 - Iteration 3

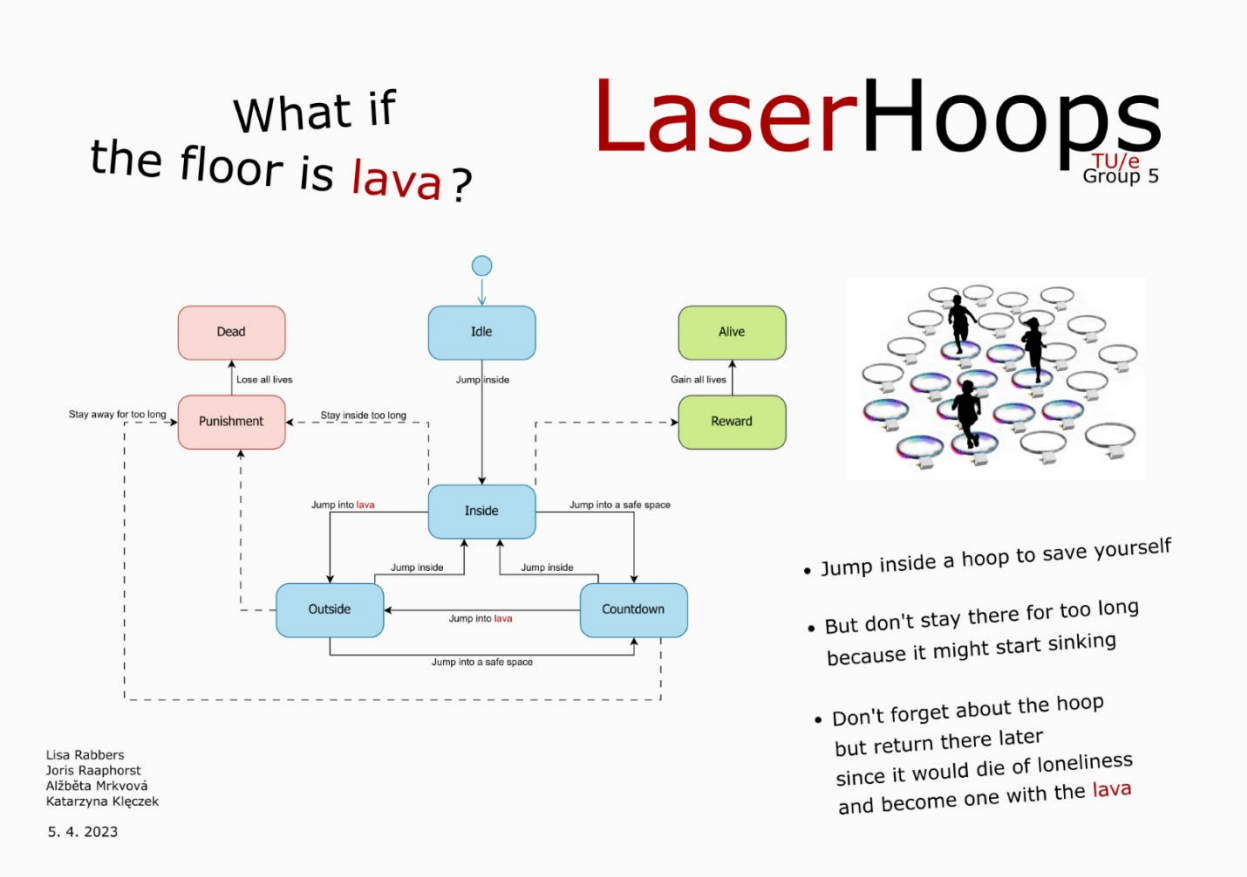
### Morphological Chart

This chart was used for generating ideas for signaling different states and actions of LaserHoops. Some of the signals were chosen for implementation and testing.

| Function | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Invitation | Pulsing light outside | **Running light outside** | Running light inside | Pulsing light inside | |
| Starting game | **Push a button on the ring** | Automatic countdown | | | |
| Reward | **Pulse of green light in the inner circle** | Pulse of green light inside AND outside | Pulse of green light on the outside ring | *Happy sound* | Deactivating lasers for a few seconds |
| Punish | Pulse of red light on the outer ring | Pulse of red light on the inside AND outside | **Pulse of red light on the inside ring** | *Buzzer* | |
| Visualizing points | Intensity of green light | Amount of green Led's = amount of points | Pulsing green light number of times = number of points | Lives: green lights taken away at every punishment | |
| Ending game | Game over: pulsing red lights | Winning: pulsing green light | Pushing a button | | |

## A.5 - Poster for the Final Presentation

For the poster, we used simplified state diagram of LaserHoops. We thought that it would be easier to explain the rules using this diagram.




## A6. - Code

The pages below contain the main part of the code we used to control the ring. The shared part is 631 lines of code, whereas the total code was constructed with 1192 lines of code. The code that was left out is the code we used for reading out the sensors and the light animations.

The code contains several out commented Serial print functions, these were used to trouble shoot the code. Also, not all parts are efficiently written, however, this approach was chosen in order to build a robust and modular code which was easily adapted after and during experiments.

```
//SET-UP LED STRIPS
#include <FastLED.h>
#define STRIP1_PIN     6 //inner
#define STRIP2_PIN     7 //outer
#define NUM_LEDS    30
#define BRIGHTNESS  255

CRGB ledsIN[NUM_LEDS];
CRGB ledsOUT[NUM_LEDS];

//SET_UP TIME MANAGEMENT
long a; // interval storage for: Ultrasonic
long b; // interval storage for: Rainbow
long c; // interval storage for: Rainbow
long d; // interval storage for: Countdown sensor check
long e; // interval storage for: Countdown timer
long f; // interval storage for: PreCountdown Interval
long g; // interval storage for: Sinking steps
long h; // interval storage for: Sinking total time
long j; // interval storage for: Cooldown
long k; // interval storage for: Sinking Delay
long l; // interval storage for: cooldown intervals
long m; // interval storage for: Fader
long n; // interval storage for: Revive timer

int SensorInterval = 20;
int CountdownInterval = 2000;
uint8_t Countdownfade;

//DESIGNATE CONNECTION PINS & STARTING VALUES
//int ProxPin = 0; int ProxValue = 0; int distance = 0; float threshold_prox = 0;

int echoPin = 4;
int trigPin = 5; int duration = 2000; float threshold_ultra = 900;
int PIR1pin = 10; int PIR1value = 0;
int PIR2pin = 11; int PIR2value = 0;
int PIR3pin = 2; int PIR3value = 0;
int PIR4pin = 3; int PIR4value = 0;

int switchPin1 = 8; int switch1Value;
int switchPin2 = 9; int switch2Value;

//LEVEL VARIABLES
bool GameOn = false;
bool outside = false;
bool inside = false;
bool motion = false;
bool Dead = false;
```

```
bool Win = false;
bool cool = false;
bool Sink = false;
bool AntiCheat = false;
bool REVIVE = false;
bool revivetoggle = false;

int revivecounter = 0;
int points = 15;         // depends on level
int pointsInterval = 0;      // Variable

int XS = 3;    // depends on level - smallest punishment or reward
int S = 3;      // depends on level - small punishment or reward
int M = 3;      // depends on level - big punishment or reward
int L = 10;      // depends on level - mega punishment or reward
int XL = 15;      // depends on level - mega punishment or reward

int timeBeforeCountdown = 10000;   // depends on level
int CountdownTime = 10000;
//int timeBeforeSinking = 10000;     // depends on level
int SinkingTimer = 10;
int TimeDifferential;
int CCounter;
int SCounter;

unsigned long markTime;
bool timerOn1 = false;
bool timerOn2 = false;

void setup() {
  Serial.begin(9600);
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(PIR1pin, INPUT);
  pinMode(PIR2pin, INPUT);
  pinMode(PIR3pin, INPUT);
  pinMode(PIR4pin, INPUT);
  pinMode(switchPin1, INPUT);
  pinMode(switchPin2, INPUT);

  FastLED.addLeds<WS2812, STRIP1_PIN, GRB>(ledsIN, NUM_LEDS);
  FastLED.addLeds<WS2812, STRIP2_PIN, GRB>(ledsOUT, NUM_LEDS);
  FastLED.setBrightness(BRIGHTNESS);

  Serial.println (";");
  Serial.println ("Run Set-up");
  Serial.println ("Idle");
}
```

```
void loop() {
  // Sensor read out
  while (Dead == false) {
    ultrasonic (200);
    motionDetect ();
    //if (GameOn == false) {
    //Start game
    idle(50);              // we start with the ring in an idle state
    //}
  }
}

void idle(int wait) {
  rainbowIN (wait);     // LED's have a rainbow pattern to invite user
  PointCountIn ();      // show the amount of points to start with

  //Serial.print("Idle");

  if (duration < threshold_ultra) { // the player enters the ring
    Serial.println(";");
    Serial.print("TRIGGER:");
    Serial.println(duration);
    GameOn = true;
    inside = true;
    start();         // start the game
  }
}

// States
void start() {           // first state after entering ring
  Serial.println("Start");
  markTime = millis();
  while (millis() - markTime < 5000) {
    InsideGreenFadeIn (5000);
    OutsideRedFadeIn (5000);
  }

  InGreen();
  OutRed();

  if (duration < threshold_ultra) { //when still inside the ring
    sinking();
  }
 if (duration > threshold_ultra) { //when allready outside the ring
    countdown();
  }
}
```

```
void sinking() {         // the ring is sinking

 //Reset lights
 OutBlack ();
 InBlack ();

 Serial.println("Sinking start");

 SensorCheck();
 if (Dead == false) {
  if (Sink == false) {
   markTime = millis();
   Sink = true;
   //Serial.print("Mark time: ");
   //Serial.print(markTime);
   //Serial.println(";");
  }
 }

 while (Dead == true) {
  Sink = false;
  OutBlack ();
  InBlack ();
 }

 while (Sink == true) {
  SensorCheck ();              // checks values

  while (Dead == true) {
   Sink = false;
   OutBlack ();
   InBlack ();
  }

  while (duration > threshold_ultra) { // IF WE STEP OUT OF THE RING
   SensorCheck ();              // checks values
   Serial.println(";");
   Serial.println("LEFT THE RING");
   AntiCheat = true;
   k = millis() + 3000;
   //Serial.print("t= ");
   //Serial.println(millis());

   while (AntiCheat == true) {
    while (Dead == true) {
     Sink = false;
     OutBlack ();
     InBlack ();
```

```
      }
      SensorCheck ();              // checks values
      Purple ();
      // Serial.print("AntiCheat == true");
      if (millis () > k) {         //Reroute to countdown
       Serial.print("k= ");
       Serial.println(k);
       Serial.println(";");
       Serial.println("STAYED OUT LONG ENOUGH - RECOVER");
       Sink = false;
       AntiCheat = false;
       Serial.println("Sink = False");
       countdown();
      }
      if (duration < threshold_ultra) { // if we step back in the ring, continue sinking
       Serial.println(";");
       Serial.println("RE-ENTRY in RING");

       //SinkingResetShort ();
       //pointsInterval = L;
       //punishment(outside);  // punish

       AntiCheat = false;
       //Sink = false;
      }
      /*/
            if (duration < threshold_ultra) { // if we step back in the ring during cooldown, RE-ENTRY
PUNISHMENT
              Serial.println(";");
              Serial.println("CHEAT PUNISH");
              SinkingResetShort ();
              pointsInterval = L;
              punishment(outside);  // punish
              AntiCheat = false;
              Sink = false;
             }
       /*/
    }
   }




while (duration < threshold_ultra) {   // player stays inside
    while (Dead == true) {
      Sink = false;
      OutBlack ();
      InBlack ();
```

```
    }

    SensorCheck ();             // checks values

    InsideRedFadeIn (1000); //red fade in every second

    if (millis() > h) {
      h = millis() + 1000; // set read-out interval
      Serial.println(";");
      Countdownfade = 0; // reset colour fade
      SinkingTimer -= 1;
      Serial.print("Sinking: ");
      Serial.print(SinkingTimer);
    }

    if (SinkingTimer <= 0) {
      Serial.println("Sink Timer Short Reset");
      SinkingResetShort ();
      pointsInterval = S;
      punishment(outside);  // punish
    }
   }
  }

  inside = false;
  if (motion == true) {  // player moved outside
    Serial.println("Motion Punishment");
    outside = true;
    pointsInterval = M;
    punishment(outside);    // punish
    outsideState();         // enter outside state
  }
  else {          // player moved to save zone
    if (Dead == false) {
      Serial.println("Sink Timer Full Reset");
      SinkingResetLong ();
      countdown();       // enter countdown state
    }
    while (Dead == true) {
      Sink = false;
      OutBlack ();
      InBlack ();
    }
  }
}

void countdown() {         // the ring is in countdown state
  if (Dead == false) {
```

```
Serial.println("COUNTDOWN START");

//Reset lights
OutBlack ();
InBlack ();

//Set-lights
PointCountIn();


if (timerOn1 == false) {
  markTime = millis();
  Serial.print("Mark time: ");
  Serial.print(markTime);

  //Serial.print("-> Actual Time: ");
  //Serial.println(millis ());

  //f = millis() + 1000;
  timerOn1 = true;
  //timerOn2 = true;
  //Serial.println(timerOn1);
  //Serial.println(timerOn2);
}
while (timerOn1 == true) {
  while (millis () - markTime < 10000) {
    //In Precountdown      //Serial.print("$");
    SensorCheck ();
    InsideCheck();
    OutsideCheck ();
    CountdownIntervalCheck();
  }
  while (millis () - markTime > 10000) {
    //Beyond Precountdown - PLACE NOTHING ELSE IN THIS STATEMENT!

    while (millis () - markTime < 20000) {
      //During Countdown //Serial.print("@");
      SensorCheck ();
      InsideCheck();
      OutsideCheck ();
      CountdownIntervalCheck();
    }

    while (millis () - markTime > 20000 && Dead == false) {
      //After Countdown //Serial.print("!");

      SensorCheck ();
      InsideCheck();
```

```
        OutsideCheck ();

        Serial.println(";");
        Serial.println("COUNTDOWN IS DONE");

        CountdownReset ();
        pointsInterval = M;
        punishment(outside);  // punish
        countdown();      // enter coutntdown state again
      }
     }
    }
  }
}

void Cooldown(int wait) {
 Serial.print("CoolDown: ");
 Serial.println(wait);

 PointCountIn();
 rainbowCool(wait);
}

void outsideState() {         // the ring is in outside state
 Serial.println("outsideState");
 PointCountIn ();      // show lives inside
 while (motion == true) {      // if the player stays outside
  // maybe other way around
  pointsInterval = M;
  punishment(outside);    // punish
  OutRed ();
  OutRedFadeOUT(2000);
  delay (3000);
  SensorCheck();

  if (PIR1value == 0 && PIR2value == 0 && PIR3value == 0 && PIR4value == 0) {
   motion = false;
   //Serial.println ("No motion detected");
  }
 }

 if (duration < threshold_ultra) { // the player moves inside
  inside = true;
  outside = false;
  reward();        // reward
  sinking();         // go into sinking state
 }
```

```
else {          // the player moves to safe zone
   countdown();       // enter countdown
  }
}


void dead() {
  Serial.println("THE RING IS DEAD!!!!!");
  Dead = true;
  OutBlack ();
  InBlack ();

  while (Dead == true) { //REVIVE
    SensorCheck();
    if (REVIVE == false) {
      if (duration < threshold_ultra) { // Jump in
        REVIVE = true;
        revivecounter = 0;
        n = millis() + 4000; // set revival timer
      }
    }

    while (REVIVE == true) {
      SensorCheck();

      while (millis () < n) {
        SensorCheck();
        while (revivetoggle == true) {
          SensorCheck();
          if (duration > threshold_ultra) { // Jump up
            revivetoggle = false;
            Serial.println(";");
            Serial.println("UP");
          }
        }

        while (revivetoggle == false) {
          SensorCheck();
          if (duration < threshold_ultra) { // Jump in
            Serial.println(";");
            Serial.println("IN");
            revivetoggle = true;
            revivecounter += 1;
            Serial.print("revivecounter= ");
            Serial.print(revivecounter);
```

```
          if (revivecounter == 4) {
            Serial.println("REVIVE");
            REVIVE = false;
            Dead = false;
            points = 15;
            revivecounter = 0;
            SinkingResetLong ();
            countdown();
           }
          }
         }
        }
        if (millis () > n) {
          REVIVE = false;
        }
      }
     }
    }

void win() {
  Serial.println("THE RING IS SAVED!!! YOU WIN!!!");

  while (Win == true) {
    rainbowIN (10);
    rainbowOUT (10);
    // didn't specify behaviour yet
  }
}

// Punishments & Rewards
void punishment(bool outside) {

  if (outside == true) {
    Serial.println("punishment outside");
    //RedBlinkOUT(1000);
    //RedBlinkIN (1000);
    RedBlinkBoth(1000);
  }
  else {
    Serial.println("punishment inside");
    //RedBlinkIN (1000);
    //RedBlinkOUT(1000);
    RedBlinkBoth(1000);
  }
  substractPoint();
}
```

```
void reward() {
  Serial.println("reward");
  //GreenBlinkIN (2000);
  GreenBlinkBoth(2000);
  CountdownReset ();
  addPoint();

  if (Win == false) {
    Cooldown(10000);
  }
}

// Adding and substracting points

void substractPoint() {
  Serial.print(points);
  Serial.print(" - ");
  points -= pointsInterval;
  Serial.print(pointsInterval);
  Serial.print(" = ");
  Serial.println(points);

  if (points <= 0) {
    dead();
  }
}

void addPoint() {
  Serial.print(points);
  Serial.print(" + ");
  points += pointsInterval;
  Serial.print(pointsInterval);
  Serial.print(" = ");
  Serial.println(points);

  if (points >= 30) {
    Win = true;
    win();
  }
}
```

```
void PrecountdownIntervalCheck () {
  if (millis () > f) {
    CCounter += 1;
    Serial.println(";");
    Serial.print("Precountdown interval: ");
    Serial.println(CCounter);
    //Serial.print("Time: ");
    //Serial.println(millis () - markTime);

    //Serial.print("Actual Time: ");
    //Serial.println(millis ());

    f = millis() + 1000; // set read-out interval
  }
}

void CountdownIntervalCheck() {
  if (millis() > e) {
    Serial.println(";");
    CCounter += 1;
    Serial.print("Warning: ");
    Serial.print(CCounter);
    Serial.print("   ");
    //Serial.println("= Light response");

    //Serial.print("Time: ");
    //Serial.print(millis () - markTime);
    //Serial.print("-> Actual Time: ");
    //Serial.println(millis ());

    e = millis() + 1000; // set read-out interval
    //Serial.print("e= ");
    //Serial.println(e);
    Countdownfade += 13;

    uint8_t hue = 0;
    uint8_t saturation = 255;

    CRGB color = CHSV(hue, saturation, Countdownfade);
    fill_solid(ledsOUT, NUM_LEDS, color);

    //fadeToBlackBy(ledsOUT, NUM_LEDS, 10);
    FastLED.show();
  }
}
```

```
void CountdownReset () {
 CCounter = 0;
 Countdownfade = 0;
 e = 0;
 timerOn1 = false;
}

void SinkingIntervalCheck() {
 uint8_t hue = 0;
 uint8_t saturation = 255;
 Sink = true;
 markTime = millis();
 Serial.print("Mark time: ");
 Serial.println(markTime);

 // Fading in
 while (Sink == true) {
  while (millis () - markTime <= 10000) {
   SensorCheck();

   if (millis () - markTime >= 10000) { // IF TIME RUNS OUT
    Serial.println ("SINKING IS OVER");
    Serial.print ("Time past: ");
    Serial.println (millis () - markTime);
    Sink = false;
    SinkingResetShort ();
    sinking();
   }

   if (millis () > g) {         // Short intervals for increase in RED light
    g = millis() + 1000 / 255;
    Countdownfade += 1;
    CRGB color = CHSV(hue, saturation, Countdownfade);
    fill_solid(ledsIN, NUM_LEDS, color);
    FastLED.show();
   }

   if (millis() > h) {
    h = millis() + 1000; // set read-out interval
    Serial.println(";");
    Countdownfade = 0; // reset colout fade
    SinkingTimer -= 1;
    Serial.print("Sinking: ");
    Serial.print(SinkingTimer);
   }
  }
 }
}
```

```
void SinkingResetShort () {
  SinkingTimer = 3;
  h = 0;
  g = 0;
  Countdownfade = 0;
}


void SinkingResetLong () {
  SinkingTimer = 10;
  h = 0;
  g = 0;
  Countdownfade = 0;
  Sink = false;
}
```