Improving waste separation quality at the UT Campus

Creative Technology Bachelor Thesis

Younghun Rhee

University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

Supervisors: Ir. Ing. R.G.A. Bults & Dr. K. Zalewska
Client: University of Twente Campus & Facility Management

August 2022

Abstract

A growing number of waste is a significant environmental problem in society, and the amount of waste is expected to increase dramatically. Proper waste separation is a key factor to be able to recycle waste. However, the University of Twente Campus & Facility Management investigated that waste is not well separated at the University of Twente. To solve this problem, the interactive waste island that can educate waste separation knowledge for the UT community member was developed. The promising result revealed that the prototype successfully improved the waste separation quality by increasing the number of correct waste disposals and influenced future waste separation behavior positively. The developed interactive waste island showed the potential to enhance the waste separation quality at the UT campus by educating the UT community members. This research can contribute to not only raising recycling quality at the UT campus but also solving environmental problems ultimately.

Acknowledgments

First of all, I would like to express my deepest appreciation to my supervisors, Richard Bults and Katarzyna Zalewska for all the support and guidance during the graduation project. Moreover, I am also thankful to the client, the University of Twente CFM for the waste island. Lastly, I had the pleasure of collaborating with my co-researcher, Senna Claes.

Table of Contents

List of Figures

1. Introduction

1.1 Context

Industrialization and economic development lead to the global consumption of earth resources. After products have reached their original purpose, they are ended up in waste. Recycling is an important factor to use finite resources and save the environment. Thus UT-CFM launched the plan to achieve a waste-free campus by 2030 and a circular campus by 2050. The ultimate objective of this plan is to prevent, reduce and improve waste recycling (UT-CFM, 2021). The recycling rate of municipal waste in Europe is 47.8% in 2020, still below 50%, which is a low number (Eurostat, 2022). This means the majority of wastes are not properly recycled. The crucial starting point of recycling is waste separation. Waste separation is a significant component to facilitate successful recycling management and minimize unrecyclable waste.

1.2 Problem Statement

The University of Twente is collecting waste separately in 4 streams: Paper, PMD, Organic and Residual. The UT-campus deployed waste islands separate these 4 streams and are installed in central locations in the UT buildings. However, waste separation has not been done well in the UT community. According to a waste analysis conducted by UT-CFM in January 2020, only 32% of waste in the residual stream is actually residual waste, meaning that the other 68% should be in a different stream to dispose of in a sustainable way (UT-CFM, 2021). Hypothetically, UT community members don’t have enough awareness of what waste should be disposed of in which stream. Alternatively, they are just too busy or not enough interested to care about separation thus throwing all their waste in one stream, usually the residual stream. Moreover, it often happens that waste collected during the lecture, meeting, lunch, or seminar is disposed of all together in the residual stream without separation. Another reason behind this could be the lack of information on the waste islands. Only a few examples of the stream are displayed on the waste islands, and people may be confused. A common misconception that all the waste in four streams will be merged anyway is the other cause. The research conducted at Maastricht university argues that students express positive and high intention to waste separation, however, knowledge about correct waste separation is not sufficient for some wastes such as coffee cups and napkins (Árnadóttir et al., 2018). Students believe that they separate waste well, but actually they don’t.

The challenge is to make the UT community more aware of proper waste separation on the UT campus. Individual behavior is important as the general policy of the university. This means that the current behavior of the community members should be influenced in such a way, that the trash they throw away is disposed of in the correct bins of the waste islands. This awareness is also for the people who think they separate wastes well, but they are actually disposing of wastes in the wrong streams.

1.3 Research Question

The goal of this project is to design a behavior intervention to improve the quality of waste separation. In this research, the following question will be answered:

How can a physical interactive installation be used to influence the UT community members' behavior of waste separation at the UT campus?

1.4 Thesis structure

Chapter 1 is an introduction. In Chapter 2, the background research including the state of art will be analyzed. Then in Chapter 3, the method and techniques will be discussed. Chapter 4 will identify and analyze stakeholders and introduce the concept from the idea generation. Chapter 5 explains the specifications of the system. In Chapter 6, the specified system is realized into a prototype. Chapter 7 evaluates the performance of the prototype. In Chapter 8, discussion and future work are suggested. Finally, in Chapter 9, the conclusion will be built.

2. Background Research

In this chapter, the background research is present. First, the literature review on the influencing waste separation behavior is conducted. Then, the state of art is given. The chapter will conclude with the overall discussion.

2.1 Literature Review

First, the definition of pro-environmental behavior is explored to understand its concept. After defining, the framework, Theory of Planned Behavior, will be introduced to understand waste separation behavior. Then based on the framework, the relation between students' intention and waste separation behavior will be researched since intention is the key factor that influences the behavior according to the framework. Then the category of methods to change the behavior is researched to get an insight into how to change the waste separation behavior. After that, the factors that affect the waste separation behavior on the university campus will be researched. Lastly, currently developed interventions are evaluated.

2.1.1 Pro-environmental behavior

Pro-environmental behavior is defined as behavior that has minimized harmful effects or is beneficial to the environment (Steg & Vlek, 2009). Kurisu (2016) has a similar definition that pro-environmental behavior is the behavior that contributes to reducing the negative impact or promoting positive impacts on environmental issues.

Stern (2000) distinguished three types of pro-environmental behaviors: environmental activist behavior, public non-activist behavior, and private action. Environmental activist behavior includes actively participating in protesting or communicating with government representatives on environmental policies. Public non-activist behavior is behavior that supports environmental organizations or is ready to accept public policies such as regulations or higher taxes for the purpose of protecting the environment. Private action can be explained as the behavior that purchasing eco-friendly products, using public transport instead of driving, or contributing to waste separation (Stern, 2000).

Waste separation behavior is a subcategory of pro-environmental behavior. Yuan et al. (2016) explain that waste separation behavior is an initiating moment of efficient recycling that enhance reducing and reusing of waste. Chen et al. (2019) defined waste separation behavior as the process of waste management that collects and classifies waste by categories and disposes of the waste at the appropriate place to diminish complications of waste disposal.

2.1.2 Theory of planned behavior

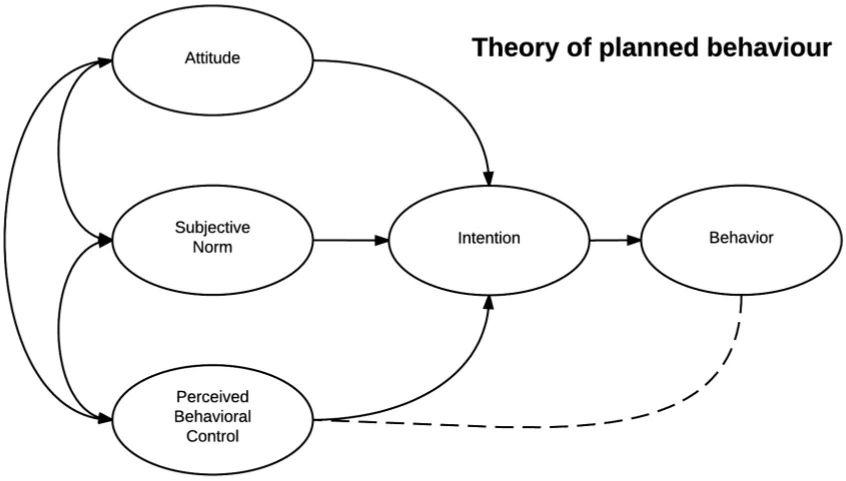

Figure 1 Theory of planned behavior

The theory of planned behavior is a theory proposed by Ajzen (1991) that explains how an individual's beliefs and attitudes lead to their behaviors. According to the theory of planned behavior, an individual's behavioral intention which is a desire to perform the behavior is a key factor that shapes and influences the behavior.

This behavioral intention is determined by three major constructs (Ajzen, 1991). Figure 1 shows the relationship between these three constructs, intention and behavior. The first construct is *an individual's attitude towards behavior*, which refers to the individual's evaluation of the behavior. This is an individual's expectation that whether a certain behavior contributes positively or negatively to their life. The second construct is *the subjective norm*, which can be described as the social pressure that is perceived by whom are close to the individual to perform the behavior. This is an opinion from the others in the individual's social circle who influences one's decision-making. The last construct is *perceived behavioral control*, which is the individual's recognition of how easy or difficult to execute the behavior. This is a belief from individuals about whether they have enough tools or self-efficacy required to implement the behavior. This construct can predict the behavior together with behavioral intention (Ajzen, 1991). In general, strong behavioral intention drive better behavior performances, and the intention is stimulated by these three constructs.

The theory of planned behavior can be used as a framework to describe pro-environmental behavior, but there is a limitation to adapting it to waste separation behavior. Blok et al. (2015) claimed that the theory of planned behavior can clearly explain pro-environmental behavior, the most important element is the intention to act that directly affects pro-environmental behavior as well as three constructs to behavioral intention. They also argued that perceived behavioral control is the best construct that explains the intention among the three constructs.

Lucarelli et al. (2020) support that those three antecedents in the theory of planned behavior significantly contribute to forecast intention to predict pro-environmental behavior. However, some research shows that the theory of planned behavior is not the perfect framework for waste separation behavior. Greaves et al. (2013) found that the theory of planned behavior can explain 53% of the variance in behavioral intention to waste recycling in the workplace. This outcome indicates that the theory of planned behavior cannot explain 100% of the intention of recycling behavior.

2.1.3 Relation between students' intention and behavior of waste separation

According to the theory of planned behavior, one's intention is the key elements that determine the behavior. To discover how this intention contributes to students' waste separation behavior extra research is conducted, and it turns out that the intention of waste separation alone does not always contribute to waste separation behavior.

Zhang et al. (2017) found that students who are in favor of waste separation do not end up with proper waste separation behavior. No or little correlation between students' intention and their actual participation in waste separation was found. The other research examines that 92% of college students recognize waste separation as proper pro-environmental behavior, however, only 17.8% of students put waste separation into practice (Hao et al., 2020). Similar outcomes were discovered in another study that the gap exists between the intention of separate waste and students' behavior to implement it in real life (Árnadóttir et al., 2018). University students theoretically understand the importance of waste separation and have a positive attitude, but it does not associate with actively participating in waste separation. Árnadóttir et al. (2018) explain that this is because lack of knowledge or skills to execute behavior accurately.

2.1.4 Methods to change waste separation behavior

Choosing the proper method to influence the behavior is important. The method should be effective and efficient. To get an insight into the proper method, currently adopted methods are researched.

Wallen and Daut (2018) introduced four categories of behavior change methods. The first method is *Education & awareness*, which is the method that delivers information by using mediums such as promotional campaigns, posters, and booklets. The second method is *Outreach, relationship building and trust*, which contains actions designed to offer services or goods that advance well-being or sustainability. The third method is *Social influences*, which interact with the behaviors, thoughts or feelings of other individuals or groups to prompt a change in behaviors, thoughts or feelings of individuals. The last method is *Behavioral insights and nudges*, which derives the predictive behavior change by understanding and implying various cognitive disciplines (Wallen & Daut, 2018). On top of these 4 elements, Grilli and Curtis (2021) suggest another category of the method, *Incentives*. It is the method that offers monetary rewards or compensations to encourage the desired behavior.

Grilli and Curtis (2021) investigated the success rate of these methods. *Outreach, relationship building and trust* and *Incentives* have the highest success rate, more than 80%. Next to that, *Social influences* is 78% effective *Behavioral insights and nudges* is 75% effective. *Education & awareness* obtained the least success rate among these five methods, 66%.

Then Grilli and Curtis (2021) also explored how many of these five types of methods are applied especially in waste separation behavior. They counted only successful interventions that affect behavioral change, a total of 34 cases are discovered. *Education and awareness* method and the *Social influence* method are the two most generally used, respectively 12 and 10 applications. Outreach, relationship building and trust, and incentives are used 6 and 5 times, while there is a single case of usage of behavioral insight and nudges method (Grilli & Curtis, 2021). Education and Social influence methods are frequent methods for changing waste separation behavior against other methods introduced, nudging is the least favorable.

2.1.5 Factors that influence waste separation behavior in the university

Discovering the factors that are directly related to waste separation behavior in the university can acquire a better perception specifically in the university setting. Thus, three factors are discovered that can influence waste separation behavior.

First, students' *knowledge and information* about waste separation relate to their waste separation behavior. This factor is related to *attitude* and *perceived behavioral control* from the theory of planned behavior since the knowledge and information can encourage confidence in waste separation behavior. The research from Dai et al. (2017) shows that willingness to separate waste is affected by knowledge and the major obstacle to waste separation is a lack of knowledge about how to classify the waste. Dai et al. claim that appropriate education can stimulate effective waste separation. Torres-Pereda et al. (2020) support this argument that environmental education intervention about the consequences of inappropriate waste separation can increase attention to waste management. Without a clear understanding of waste categorization and the outcome of the behavior, it's hard to recognize the necessity of waste separation. It appears that lack of knowledge and information has a link to the poor quality of waste separation.

Secondly, the number of waste bins and accessibility is also significant factor. This factor can influence *attitude* from the theory of planned behavior. Sheau-Ting et al. (2016) found that the most important attribute of waste separation behavior in the university community is the accessibility to recycling bins. They suggest that the optimal distance between waste bins is 100 to 500 meters, more than 500 meters distance is not favorable and convenient to waste separation behavior. They also claim that recycling bins should be located near the place where waste is often generated. Because students often do not keep the waste and throw all waste together in the ordinary residual waste bin if they couldn't find the proper recycling bins (Malakahmad, 2010). Malakahmad also supports that students have been discouraged from participating in waste separation by the unavailability of suitable and enough recycling bins. While these studies are about absolute distance, another study claims that perceived distance to the waste disposal facility is a more significant predictor of waste separation behavior (Lange et al., 2014). The research found that recycling behavior and perceived distance have a larger negative correlation than recycling behavior and actual distance. This is because individual psychology works on waste separation behavior (Chen et al., 2019). Easy to access recycling bins and close perceived or psychological distance to waste bins can enhance waste separation behavior. This factor is related to

Finally, the hygiene of the waste bin is an important element. Lee et al. (2017) argue the inconvenience cost of waste separation at disposal spots with poor hygiene is high. It is even larger than the inconvenience cost of the behavior of sorting waste. This means that people are more reluctant to use unclean and contaminated waste separation spots than doing the actual separation. They also report that the inconvenience cost is higher in the young generation than in the older generation. Another interview carried out amongst university students in Thailand shows that students hesitate to touch waste bins because they are not always kept clean and look dirty (Supakata, 2018). Unpleasant hygiene levels of waste disposal spots or bins negatively affect waste separation behaviors, and the young generation is more influenced. This is also a factor that can affect *attitude*, bringing negative impact and memory can hinder positive attitude.

2.1.6 Evaluating interventions to improve waste separation behavior in the university

Three interventions attempt to raise the quality of waste separation behavior are evaluated. The first intervention is adopting the mandatory policy. College students in Zhengzhou are more actively participating in waste separation when they are forced to do it (Hao et al., 2020). The participation rate raised from 10.1% to 17.7% after implementing the mandatory measures in Zhengzhou. Hao et al. (2020) suggest that mandatory policy affects students despite the low absolute participation rate. The obligation successfully increases the waste separation behavior.

The second intervention is the friendly appearance of the waste bin. Supakata (2018) developed a waste bin that looks like a monster to make bins more attractive. Each stream of waste is color-coded to distinguish the categories of the waste. Each monster has a story that explains the basic rules of waste separation to appeal waste separation to undergraduate students. However, the researcher revealed that this method was not successful to change the waste separation behavior, no difference was found between before and after the system is implied.

The last intervention is an information campaign for community members in the university. Árnadóttir et al. (2018) placed the informational triangles about the proper classification of wastes and university guidelines of separation on the cafeteria tables at Maastricht University to intervene in students' waste separation behavior and observed students. The result shows that the informational triangles did not improve waste separation behavior. Students do recognize triangles on the table but did not often read them. Researchers argue that this is because of the ineffectiveness of the intervention. Overall, motivating and encouraging students is more challenging than forcing them to follow strict regulations. Passive interventions such as user-friendly appearance and information exposure are not practical.

2.2 State of the art

State of the art research is conducted to get extensive insight from the existing products to improve waste separating behavior. Total of five examples are explored.

2.2.1 Scrapp

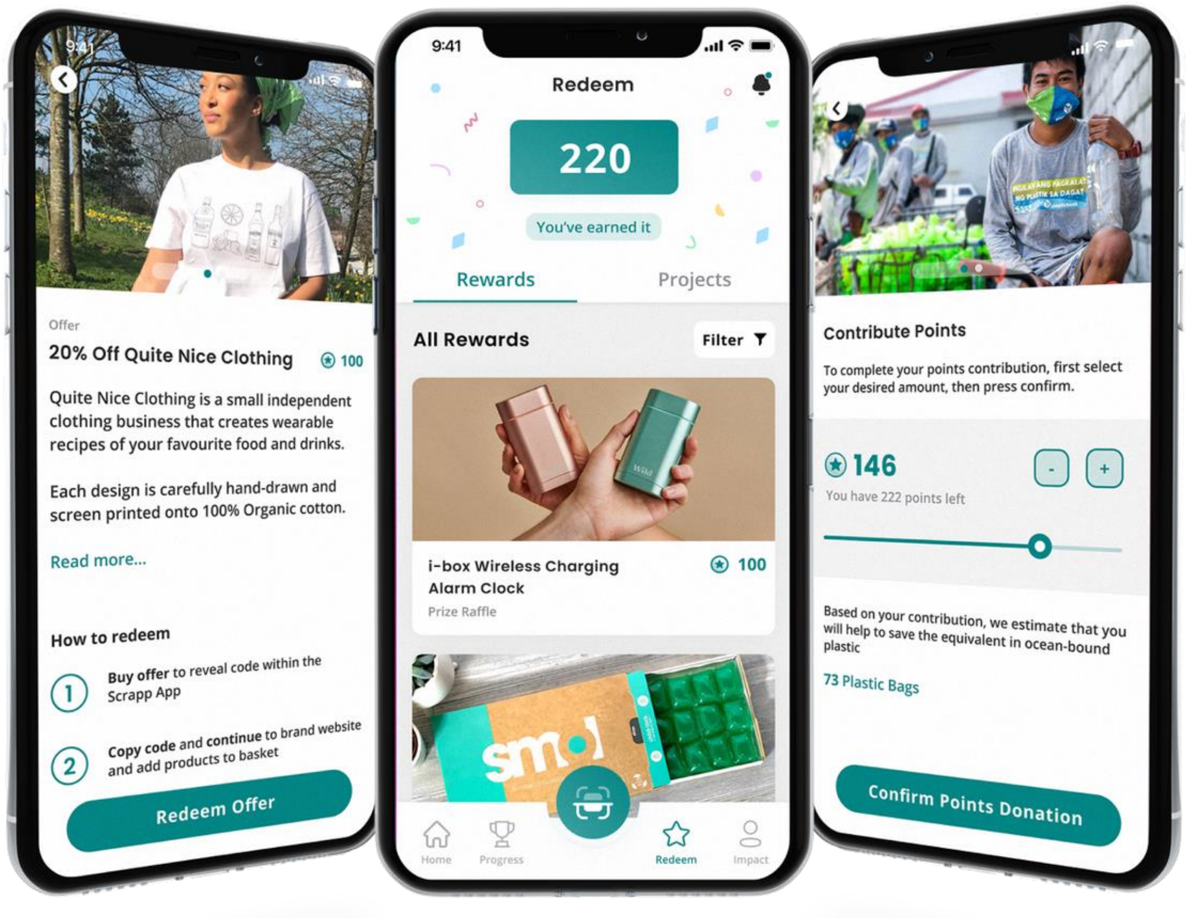

Figure 2 Scrapp

Scrapp is a mobile platform that provides a waste sorting guide and rewards (*Scrapp Recycling*, n.d.). By scanning the barcode on the product, the mobile app checks whether the item is recyclable and shows the right category of the waste to dispose of according to the local waste separating rules. The waste behavior is trackable, the dashboard displays how many items are disposed of and how many of them are recycled. Scrapp also has a reward system, users can collect Scrapp points, which can be exchanged for eco-friendly products or services via the app or sustainable marketplace. These points also can be donated to environmental projects. Scrapp trying to help the *attitude* of the waste separating behavior. This app can minimize personal efforts to decide on waste categories and gives rewards based on their action. This can affect the *perceived behavior control* of waste separation, making the user believes that waste separation is easy.

2.2.2 BIN-E

Figure 3. BIN-E

BIN-E is an IoT-based smart waste bin that automatically recognizes, separates and compresses wastes (*Bin-e Smart Waste Bin*, n.d.). This bin can be installed inside office buildings. It identifies wastes with an AI-based algorithm and sorts them into the waste category they belong to. Users don’t need to think about what type of waste they have, they just need to put the waste into the system. After the waste is sorted, the system compresses the waste to maximize the storage space left in the bin. Then it checks the fill level of each stream and shows information via the touchscreen on the top of the system. These data also can be accessed via the app on users’ smartphones. BIN-E is also can influence the *perceived behavior control* of waste separation, as similar reason as Scrapp.

2.2.3 SuperBin

Figure 4. SuperBin

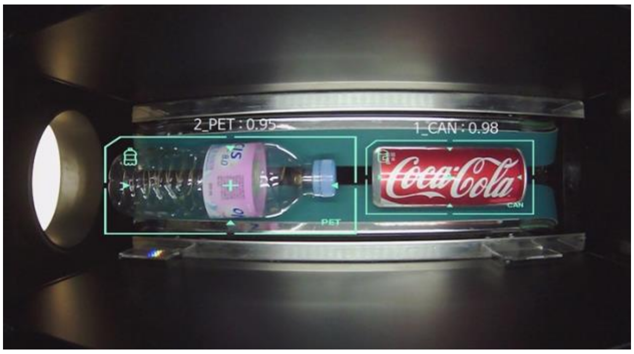

Figure 5. AI waste recognition inside SuperBin

SuperBin is a recyclable waste-collecting machine that automatically detects the type of waste using AI and offers a reward (*SuperBin Product*, n.d.). The system recognizes the waste using AI-based algorithm and accepts plastic bottles, cans, and glass bottles. This machine can be installed in public spaces or educational institutions. Users can collect cash rewards or points that can be used at the educational institution where the machine is placed. The data from the waste collection are processed to build big data. SuperBin can influence the *social norm* of waste separation since this is usually installed in the public space.

2.2.4 Today's Recycle

Figure 6. Today's Recycle

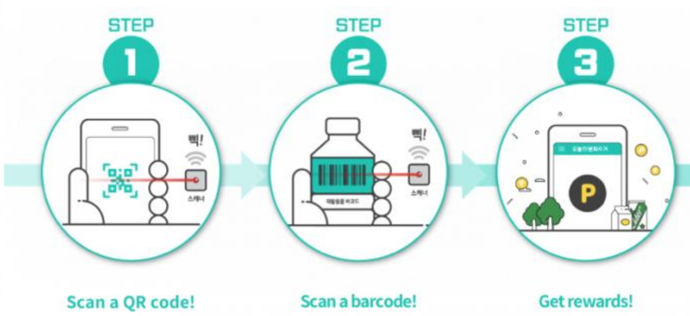

Figure 7. How to use Today's Recycle

Today's Recycle is a smart IoT installation for collecting and separating recyclable resources (*Oysterable*, n.d.). The users need to register via the app to have their personal QR code. When disposal of waste, users first need to scan a personal QR code and scan the barcode of the waste. If the waste is successfully ended up in the right category users will get point rewards that can be used at an online shopping mall in the app. The points can be donated to charity too. The system tracks weight and volume data to check the collection time and monitors user participation. The mobile app shows all types of waste that users put into the system. Today's Recycle can encourage the *social norm* of waste separation since this product is also placed in a crowded area and the *perceived behavior control* by having a reward system.

2.2.5 Charopy

Figure 8. Charopy

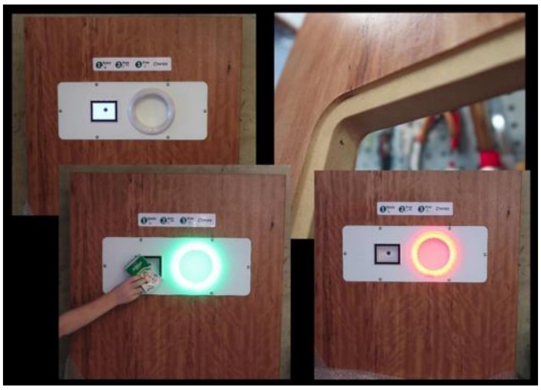

Figure 9. Charopy distinguishing correct and wrong waste

Charopy is an IoT smart bin that is developed to prevent contamination of the recycling bin by accepting only suitable wastes for recycling (*Charopy*, n.d.). Users bring their waste to the Charopy and it scans the surface of the waste. Then Charopy identifies the material of the waste and checks whether it is recyclable. If the waste is not acceptable then the LED displays red color and the flap is locked, preventing the wrong type of waste in the recycling bin. The system is connected to the internet therefore the amount of waste collected and its types are available online in real-time. Charopy can influence the *social norm* too as this is installed public spaces with crowds.

2.3 Discussion and Conclusion

The literature review discovered first the definition of pro-environmental behavior and waste separation behavior. Then the framework, the theory of planned behavior is introduced. However, the theory of planned behavior cannot explain the waste separation behavior perfectly, and also students' intention to separate waste is not always connected to behavior. From this result, the other constructs become important, *attitude*, *social norm* and *percied behavior control*. Especially, *perceived behavior control* is the most significant factor since it can directly influence behavior. Thus the intervention method should be able to influence *perceived behavior control* positively. Moreover, students' *knowledge and information are* also a significant factor, the intervention should be designed in a way that increases students' waste separation knowledge. To enhance the knowledge, *Education and awareness* method can be considered when building the intervention, but the success rate of this method is relatively low. Thus, it's better to combine other methods such as incentives and social influences which have a high success rate.

State of the art explored technology-based intervention to promote waste separation behavior and related constructs from the theory of planned behavior. Interestingly, every product has a common process. First, the system classifies the waste as recyclable and non-recyclable by using AI, scanning barcodes, or detecting surfaces. Then it guides the users to dispose of waste in the appropriate category. After wastes are accepted or collected by the system, there is a chance to collect rewards. The waste data generated from the user is shared via the internet. The intervention process can be designed based on this process. Most of the products influence *social norms* and *perceived behavior control*. The intervention should contain some techniques to affect *social norms* and *perceived behavior control.*

3. Method and Techniques

In this chapter, the method and techniques used in this research are described. The main design method is the Creative Technology Design Method. This method is selected since this project should involve the actual users at the center of the design process to give them a better experience.

3.1 Design Method

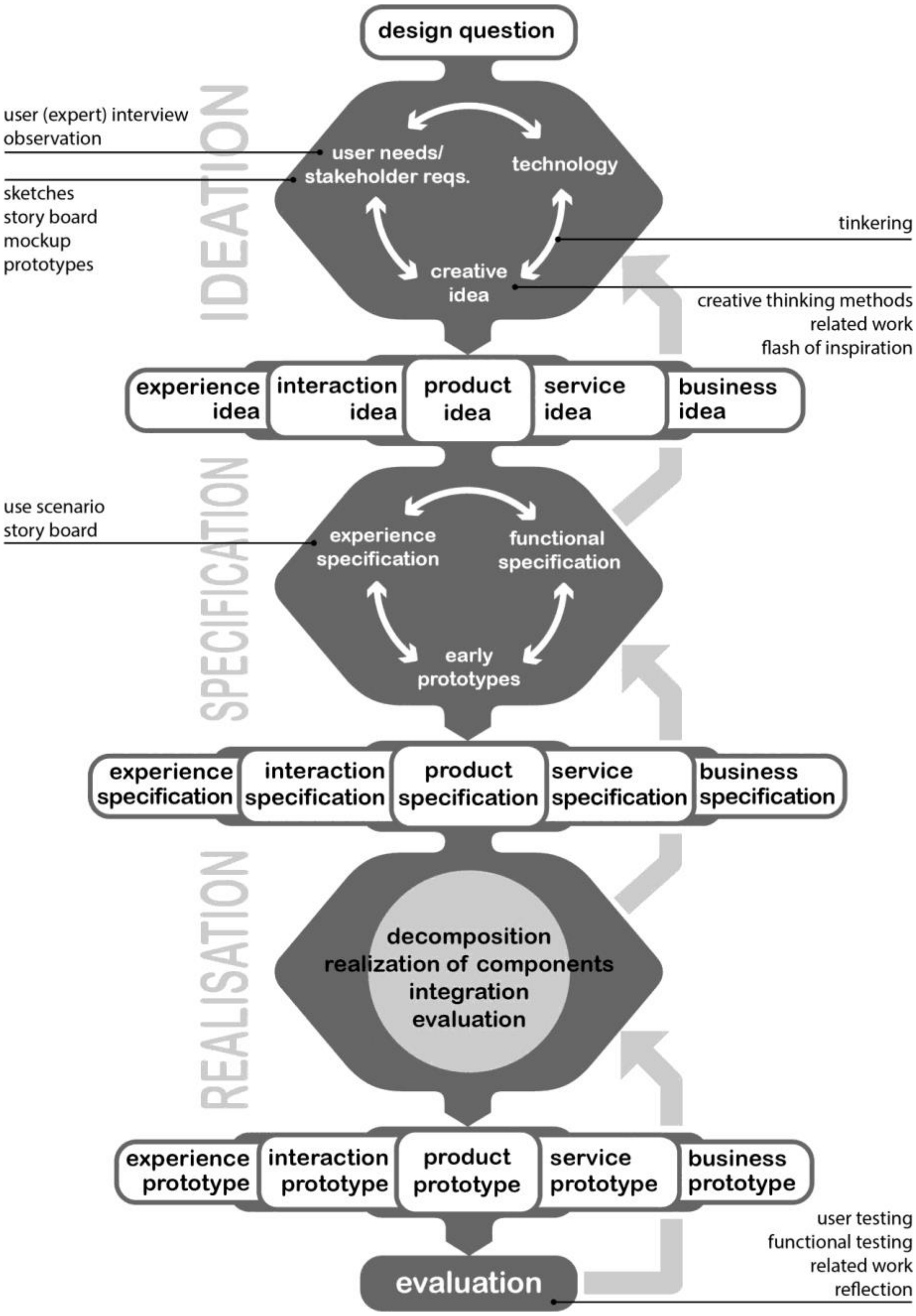

Figure 10 Creative Technology Design Method

For the design method, the Creative Technology Design Method developed by Mader and Eggink (2014) will be used. This method consists of four phases: Ideation, Specification, Realization, and Evaluation. The ideation phase shapes the product idea from the design question to solving the main problem. The ideation phase can be initiated from the three possible starting points: *user needs/stakeholder requirements*, *creative idea* and *technology*. Any of these points can be selected as a starting point and move in any direction. The next phase is the specification. In this phase, based on antecedent concepts, the functionality of the system and user interaction or experiences are specified further with details. Implementation of the short evaluation and the feedback loop also gives the opportunity to go back to the previous phase if the current process is not qualified. During the realization phase, the final prototype of the product will be developed given all the different elements that were built in the previous phases. Evaluation is the last phase, user testing and evaluation of the prototype will be conducted during this phase. All phases after the ideation have the backward path, the possibility of an iterative procedure is opened by returning to earlier phases if required.

3.2 Techniques

3.2.1 Ideation Phase

The ideation phase starts with the *user needs/stakeholder requirements*. Stakeholders will be identified and analyzed and stakeholder requirements will be defined. Then the next step will be *creative idea*. Brainstorming technique will be used to generate initial concepts. After that *technology* will be considered and the final concept will be chosen.

3.2.1.1 Stakeholder identification and analyzation

To analyze stakeholders, a power versus interest matrix will be used. The power/interest grid gives an overview of stakeholders based on their power, which is how influential stakeholders are to the project and their interests (Olander & Landin, 2005). The vertical axis shows the influences of the stakeholder and the horizontal axis shows the level of interest of the stakeholder. Stakeholders will be categorized into one of the four groups: High power-High interest, High power-Low interest, Low power-High interest, and Low power-Low interest. This technique helps to classify the impact of key stakeholders in the design process.

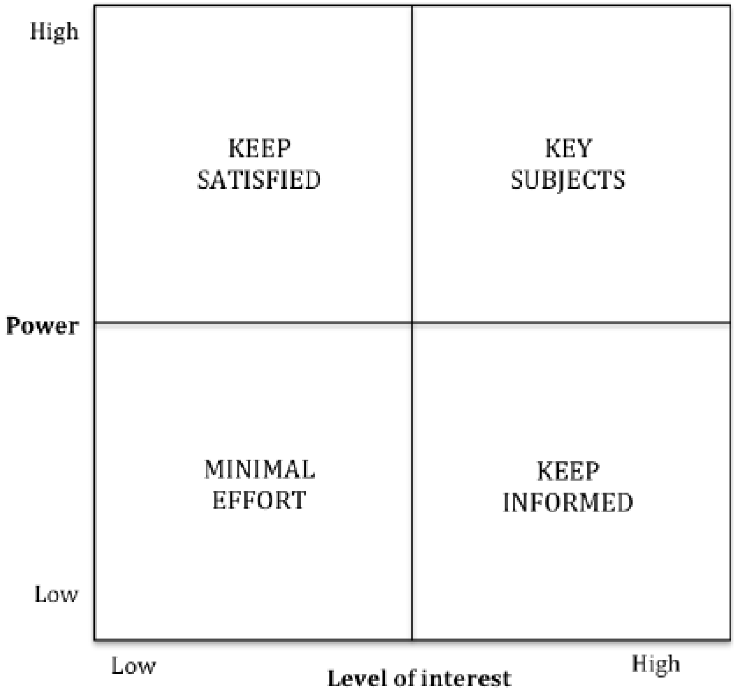

Figure 11 Power/Interest Grid

3.2.1.2 Stakeholder requirement identification

To explore stakeholder requirements, a semi-structured interview with the client to explore the client's opinion qualitatively and the MoSCoW will be used (Haughey, 2021). By interviewing the client, preliminary requirements from the client can be clearly discovered. MoSCoW method is used to categorize the requirements. MoSCoW stands for Must have, Should have, Could have, and Won't have this time. Must have category is the requirements necessary for the project's success. Should have category is the requirements are not necessary but important for the project. Could have category is the requirements that don't have a big impact on the project, but nice to have it. Won't have this time category is the requirements that have been determined as not a priority and is least significant on the project.

3.2.1.3 User-Centered Design

To involve users, UT community members, during the design process, a user-centered design will be used. The user-centered design is the approach that involves the user at the center of the design process to provide a better experience for the user (Mao et al., 2005). The context of use will be first introduced and the user requirements will be specified. Then the solution will be developed and evaluated against the requirements.

3.2.1.4 Brainstorming

To develop initial concepts brainstorming will be used. Brainstorming is a group creative technique of generating ideas and solutions (Bernstein, 2017). During the brainstorming everyone is given a pen and paper and 5-10 minutes of time to write as many ideas as possible. Since everyone can see the others' ideas, participants can come up with ideas pushing the limits. After that, everyone picks one or two favorite ideas in their lists. Then among the selected ideas, the best idea will be picked by voting.

3.2.2 Specification Phase

The start point of the specification phase will be *experience specification*. User scenarios will be used Then the next step is *functional specification*, system requirements will be developed and the system will be decomposed into two levels. The last step will be *early prototypes*.

3.2.2.1 User interaction scenario

User interaction scenario is a brief story about a user and the user's interaction when the user completes a specific task or goal (Ghazaryan, 2019). The user interaction scenario has two main elements: the role of the user and the task. To think about the role of the user, the level of knowledge and motivations or biases that the user has can be considered. The task is what the user is looking to accomplish.

3.2.2.2 Functional Decomposition

Function decomposition is breaking down the broad parts from complex levels to smaller and similar parts so that each part can be easily analyzed separately (Hayes, 2021). This helps manage complexity and reduce the uncertainty of the system.

3.2.3 Realization Phase

During the realization phase, the idea will be realized into the prototype. The ideal goal of the prototype is to meet all the functional and non-functional requirements.

3.2.4 Evaluation Phase

After the prototype is fully developed, user test will be conducted to evaluate the prototype. The sample user group will be divided in to two groups. One sample group will interact with the developed prototype to test the requirements and structured questionnaires will be given to get an insight into the user experiences of the prototype. The other sample group will interact original waste island and also structured questionnaires will be given. These two group will be compared to evaluate the prototype performance.

4. Ideation

In this chapter, the main stakeholders are identified and analyzed. Then the preliminary requirements are discussed and prioritized. Based on these outcomes, initial concepts are designed. Lastly, the final concept is developed and introduced from the initial concepts.

4.1 Stakeholder Identification

Table 1 shows the four main stakeholders and their roles and contacts of stakeholders. In this project, several stakeholders are included.

Stakeholder	Role	Contact
UT-CFM (Client)	Decision-Maker	Birgit Dragtstra
Supervisors	Decision-Maker	Richard Bults & Kasia Zalewska
UT Community Members (Students, employees, visitors)	User	-
Developers	Developer, students working on this project	Younghun Rhee & Senna Claes

Table 1 Stakeholders

4.2 Stakeholder Analysis

The power-interest matrix (Ch 3.2.1) is used to analyze four stakeholders. Figure 12 shows stakeholders are placed on the power-interest matrix based on their power and interest in this project.

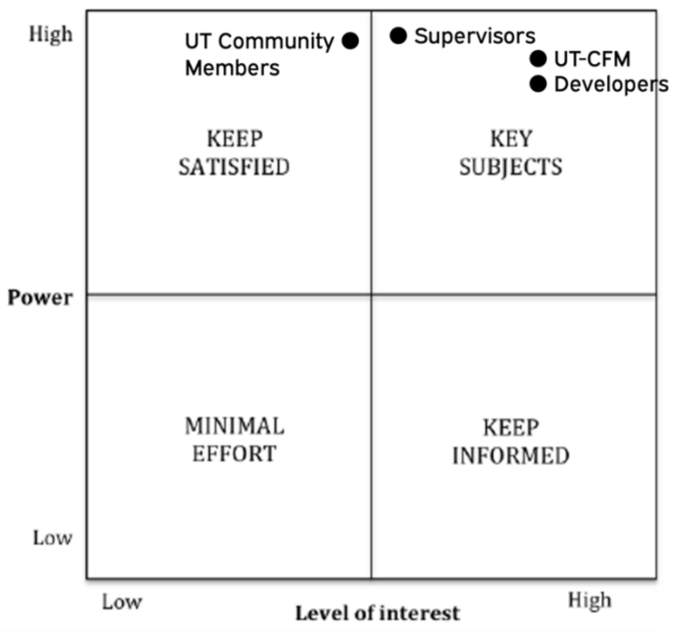

Figure 12. Power-Interest Matrix with Stakeholders

4.2.1 UT-CFM

University of Twente Campus & Facility Management department is the client of this project. CFM is a decision-maker and implements the overall environmental policies on campus to reach the sustainability goals, thus their power is significantly high. Their interest is also high as their power since the intervention that will be developed in this project can improve waste separation behavior and this will increase the quality of waste separation, which CFM would like to achieve as a sustainability goal. CFM is an important stakeholder and needs to be managed closely.

4.2.2 Developers

Developers are students who are working on this project, Younghun Rhee and Senna Claes. We design and develop the actual installation and system required for the project. I will be working on the smart technology part of the project, all the electronics, hardware and software of the whole system. Senna will focus on the interactive media part of the project, contents for the screen. Thus, our power is high but slightly lower than CFM because developers should follow preconditions from their client, CFM. We are also highly interested in this project as CFM since we are concerned about the problems CFM faces now and developing the solution for CFM. Developers are significant stakeholders.

.

4.2.3 Supervisors

The supervisors of this project are Richard Bults and Kasia Zalewska. They are decision-makers and guide the students working on this project. Because they are closely involved in the decision-making process, and their approval of the initial concept and final product is essential to continue the graduation project, they have the highest power among the stakeholders. Moreover, they would like to supervise the project to success, their interest is high but not much as CFM and developers. Supervisors are considered key stakeholders.

4.2.4 UT Community Members

UT Community Members are the people involved in the University of Twente, using the university's facilities. Students, employees, and supporting staff are UT Community Members. They study or work in university buildings and generate waste during their studying or working hours. They decide the category of the waste they have based on their own knowledge or principle and throw it away into one of the four streams on the waste island. They are the actual users using the waste island in the building to dispose of the waste, therefore their power is high. This means that the UT community members should be involved during the design process to achieve their needs. However, their level of interest is low because their short goal or object is to get rid of the waste they are having at the moment thus they don't have an awareness of how the waste is processed after the waste is left their hands. UT Community Members is the stakeholder that should be kept satisfied.

4.3 Stakeholder Requirements

Table 2 presents a total of 12 stakeholder requirements. These requirements are prioritized using a MoSCoW method from chapter 3.2.2.

Category	Requirement	Explanation
Must	Give awareness of waste separation in a limited timeframe	Users usually don't spend much time on the waste island. The intervention should be able to provide enough impact or awareness of waste separation while they are throwing the waste, which is a really short time.
	Take limited actions or steps to disposed of wastes	Users are not that patient and motivated to discover multiple steps to get information about the category of waste. The steps and their actions should be minimized.
	Be easy to interact with	Users should know how to interact with the system without specific knowledge, even if it's their first time using it.
	Educate users by showing the correct category of the waste	The system should educate users on where a waste item belongs, therefore they will separate waste in the correct way next time.
	Be placed near existing waste islands	The installation should be installed close to existing waste island so that users are able to interact with it when they want to throw the waste.
	Not modify the features such as the size, holes, or stickers of the original waste island.	UT-CFM requests to keep the waste island in its original condition. Changing the
	Not change the current location of the waste island	UT-CFM requests not to move the waste island to different places.
Should	Show benefits of recycling.	Informing the benefits of recycling can awake the awareness of the users.
	Check if waste is disposed of in the right stream.	The system is able to monitor if the waste is ended up in the correct category after the intervention.
	Prevent wastes is disposed of in the wrong category	The installation can prevent users if they would like to throw the

		waste into the wrong stream even after the intervention.
Could	Prevent technological discrimination	The installation could treat every user equally and avoid technological discrimination. Regardless of users' personal characteristics such as gender, height, or age, they could be able to access the installation and system.
	Imply positive reinforcement instead negative.	Positive reinforcement can be more effective than negative one.

Table 2 MoSCoW Analysis Stakeholders

4.4 Ideas

Together with Senna, a short brainstorming is conducted to influence waste separation behavior. In the end, researchers came up with 7 initial ideas.

4.4.1 Waste Separation App

Figure 13. Waste Separation App

The waste separation app is an app that can be installed on smartphones. Users can scan the barcode on the product package using a smartphone's camera and the app will identify the waste category and inform users where the waste should be disposed of in the waste bin. This app not only guides users but also educates them if they don't have enough knowledge of which category of waste belongs. However, not every waste has a barcode, which means that the app cannot recognize the waste without a barcode. Using the app is limited to the waste that is a package with a barcode.

4.4.2 NFC Explanation and Reward System

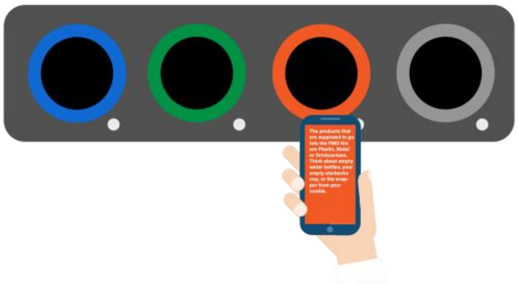

Figure 14. NFC Chips on the waste island

Small NFC chips are placed next to each category's entrance of the waste island. These NFC chips could contain explanations of each waste stream and information on the representative items for that waste stream. If users are confused about what each stream is, then they can simply tag their smartphone near these chips and these chips will trigger their smartphone to show the relative information on the screen of the smartphone. This can also be used as collecting rewards after successful disposed of the waste. If the system checks and confirms waste is collected in the appropriate category, users can claim rewards by tagging the NFC chip next to the category of the waste they throw away.

4.4.3 Interactive Waste Selection Board

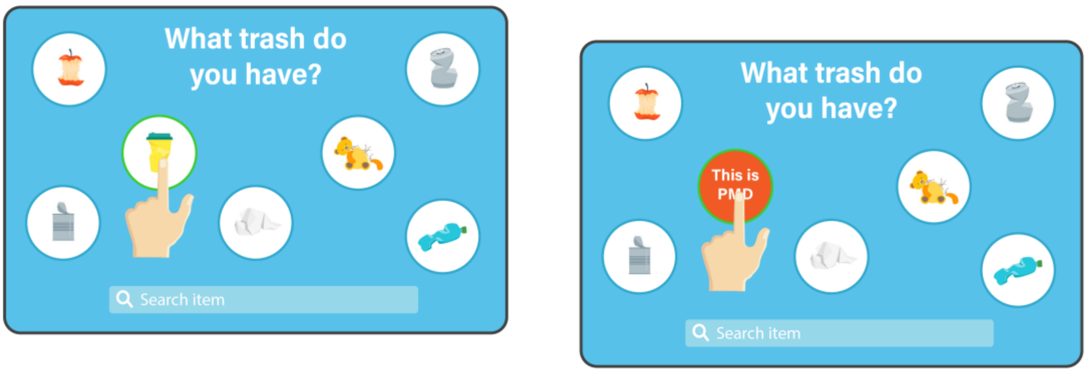

Figure 15. Interactive Waste Selection Board

The interactive waste selection board presents waste items that are frequently disposed of on the UT campus. Each item will be displayed in terms of a simple graphic thus everyone could easily understand and recognize the item without further explanation. If the user selects the item on the interactive touch board, then the board will show what stream that specific waste belongs to. This idea focuses on educating users to give better waste separation knowledge.

4.4.4 Displaying Impact of Recycling

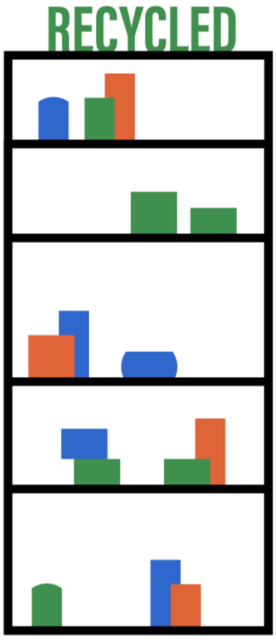

Figure 16. Shelf with recycled products

The various types of recycled products can be displayed on the shelf, showing the outcome of waste recycling. By displaying the result of the recycling, this shelf could give awareness to the public and provide the positive aspects of the recycling. This shelf can be deployed in the crowded UT building or spaces to make it visible to all UT community members. While they are passing by this shelf, they can notice the huge diversity of the products and the actual impact of the recycling and their waste separation behavior.

4.4.5 Poster

Figure 17. Poster promoting recycling

The poster is the traditional method to publicize the way of recycling and its importance. They could contain the key factors of why waste separation is important and representative wastes for each category in the waste stream. These are the formal way of delivering information with texts and graphics for a better understanding of waste separation. These media also can include statistical data on the quality of waste separation to awaken the acknowledgment of the waste separation.

4.4.6 Information Board

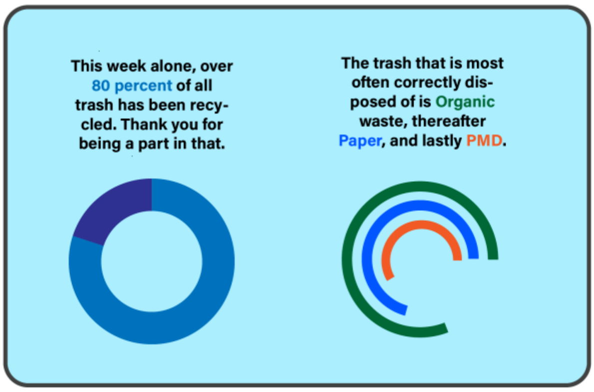

Figure 18. Information board

The information board displays the current state of waste separation on campus on the big screen. The correct waste disposal ratio and the frequencies of each waste category are displayed. If the system detects the waste is ended up in the correct category, then this data will be changed and therefore there will be an update for it.

4.4.7 Light Indicator

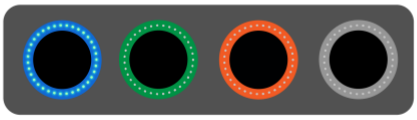

Figure 19. LED Indicators on the waste island

If the system detects where a specific waste should go, then the light around the entrance of the waste bin can blink to visually indicate to users. The light can guide users to put waste in the correct stream and prevent the wrong way of separating waste. Light is visually attractive to direct users in the right way of waste separation.

4.5 Concept

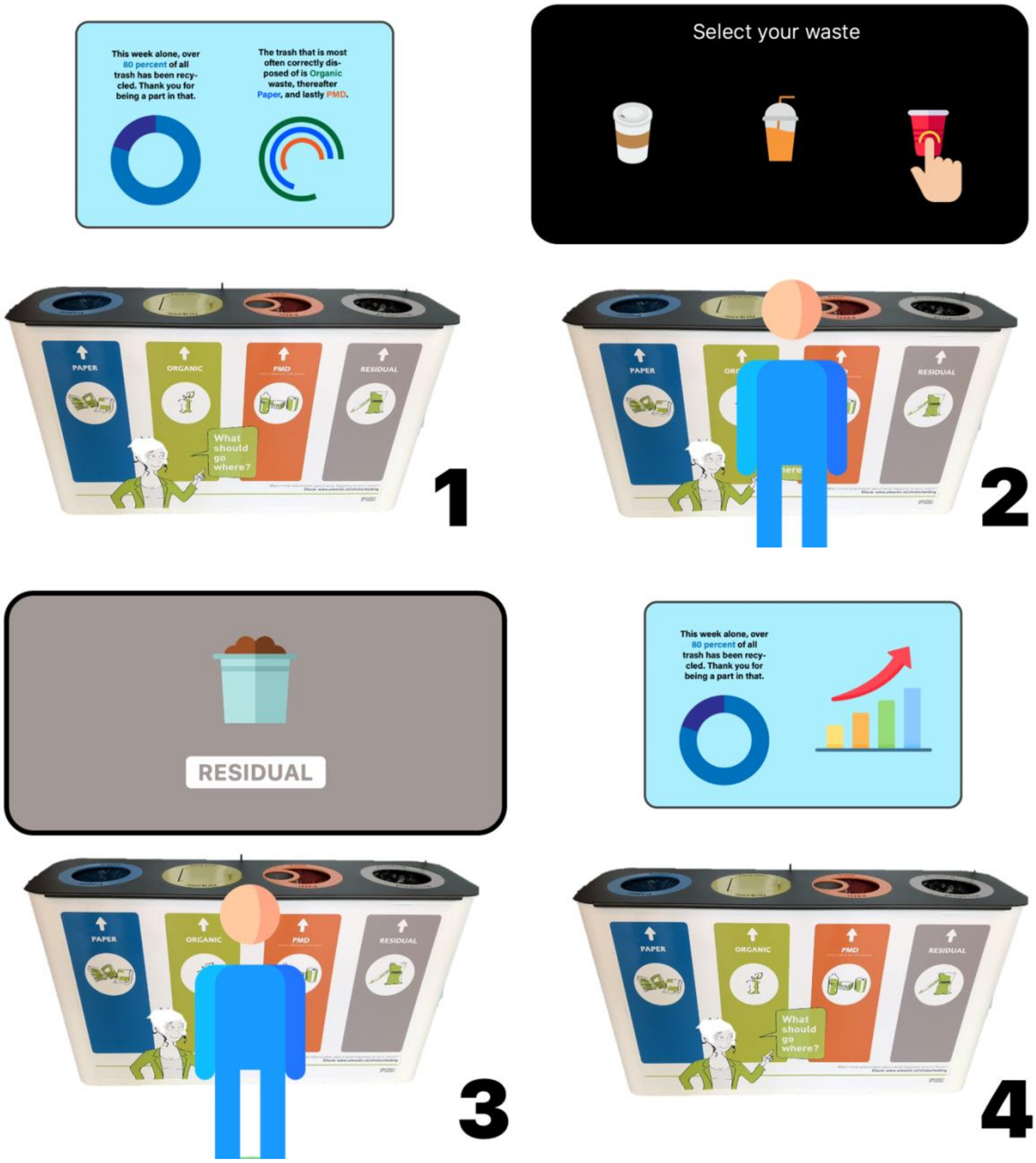

Figure 20. The final concept

The final concept is developed from the initial ideas, interactive waste separation board, information board, and light indicators are adopted into the final concept with more details. This concept works based on different situations. Figure 20 shows the concept step by step. The large interactive touch screen is placed behind the waste island. Firstly, if there is no one around the waste island, the information screen displays the statistical data that shows the current state of the quality of the waste separation of that waste island to give some awareness to the UT community members. This function is the information board from the initial idea

Secondly, the user approaches the installation and the waste island detects the distance between the user and the installation. If that user is located within a certain distance then the interactive waste selection screen is activated, which presents the most frequent waste that can be found on the UT campus. The displayed wastes are sorted based on the hierarchy, higher hierarchy categories are shown first, and then sub-categories of that category are visible. For example, waste categories such as paper, cups, and product wrappers are displayed first, and if users select the category cups then all kinds of cups such as paper cups, plastic cups, or reusable cups are displayed. This function is the separation board from the initial concept.

Thirdly, the user chooses their waste on the selection screen. After the selection, the interactive selection screen indicates the correct stream of waste. The user will dispose of waste in the category that is shown on the selection screen. Then the system monitors whether the waste is disposed of accurately following the guidance of the screen or the waste is still ended up in the wrong category. If it is verified that the waste is separated in the right direction, then the screen tells the user thank you for correct disposal. If it is wrong, then the screen shows the representative examples of wastes that belong to that category for educational purposes.

Lastly, if the user leaves the waste island, the selection screen again turns into an information screen. The system updates data from the last waste disposal, then the user can notice that the information screen is updated, for instance, the ratio of recycling is increased after their participation. This can be a small reward, the user can feel he/she is actually contributing to the recycling by separating waste correctly.

5. Specification

In this chapter, the final concept is specified. First personas are developed and the interaction scenario is presented based on personas. Next, the system requirements are given. Then the functional architecture is described and lastly, the time sequence diagram is introduced.

5.1 Personas

Three personas are developed to create the interaction scenario, one is just a normal student who is either interested or not interested in the environment. The other is an environmentalist and the last one is environmental ignorance.

5.1.1 James

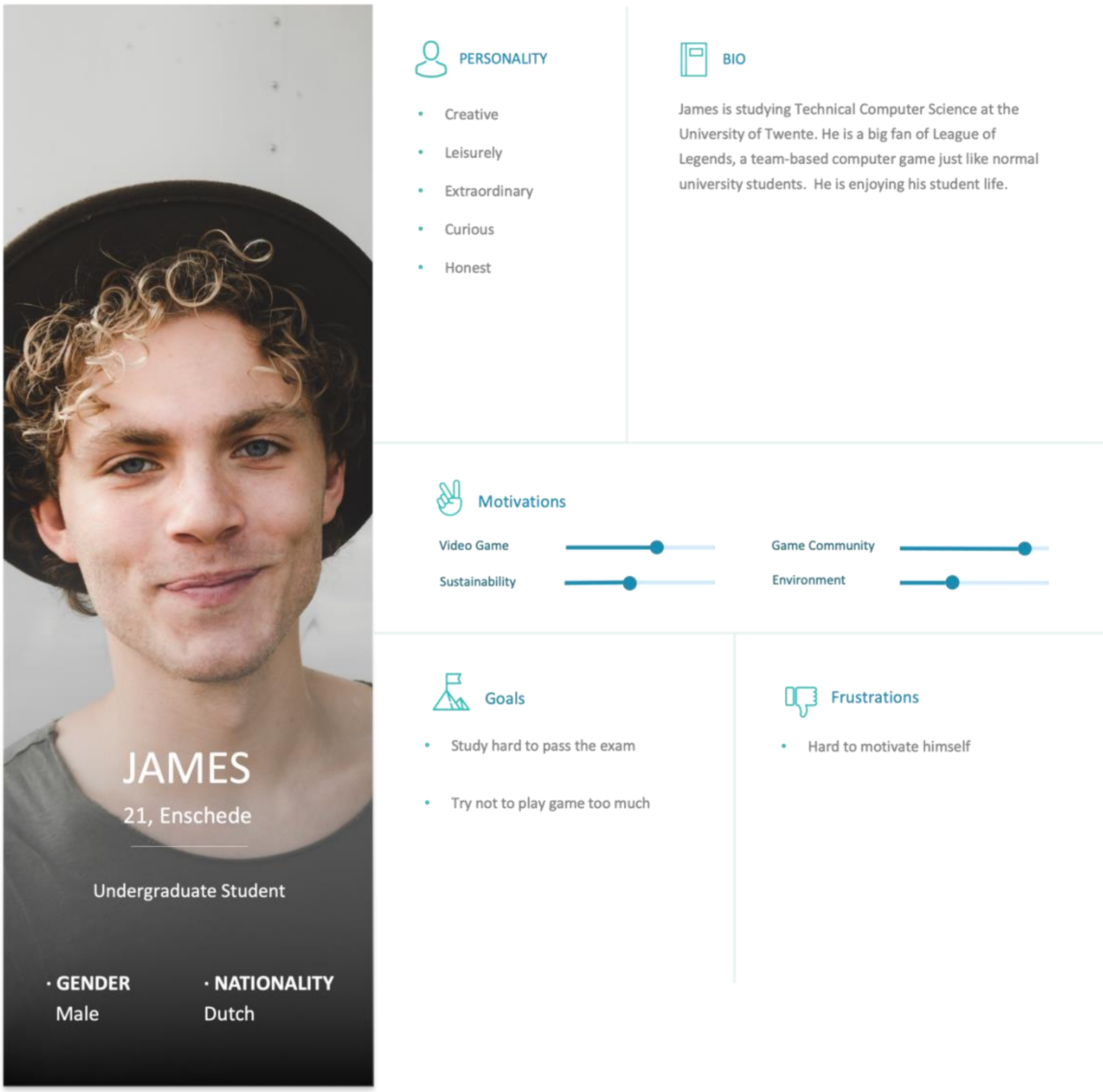

Figure 21 James

5.1.2 Mia

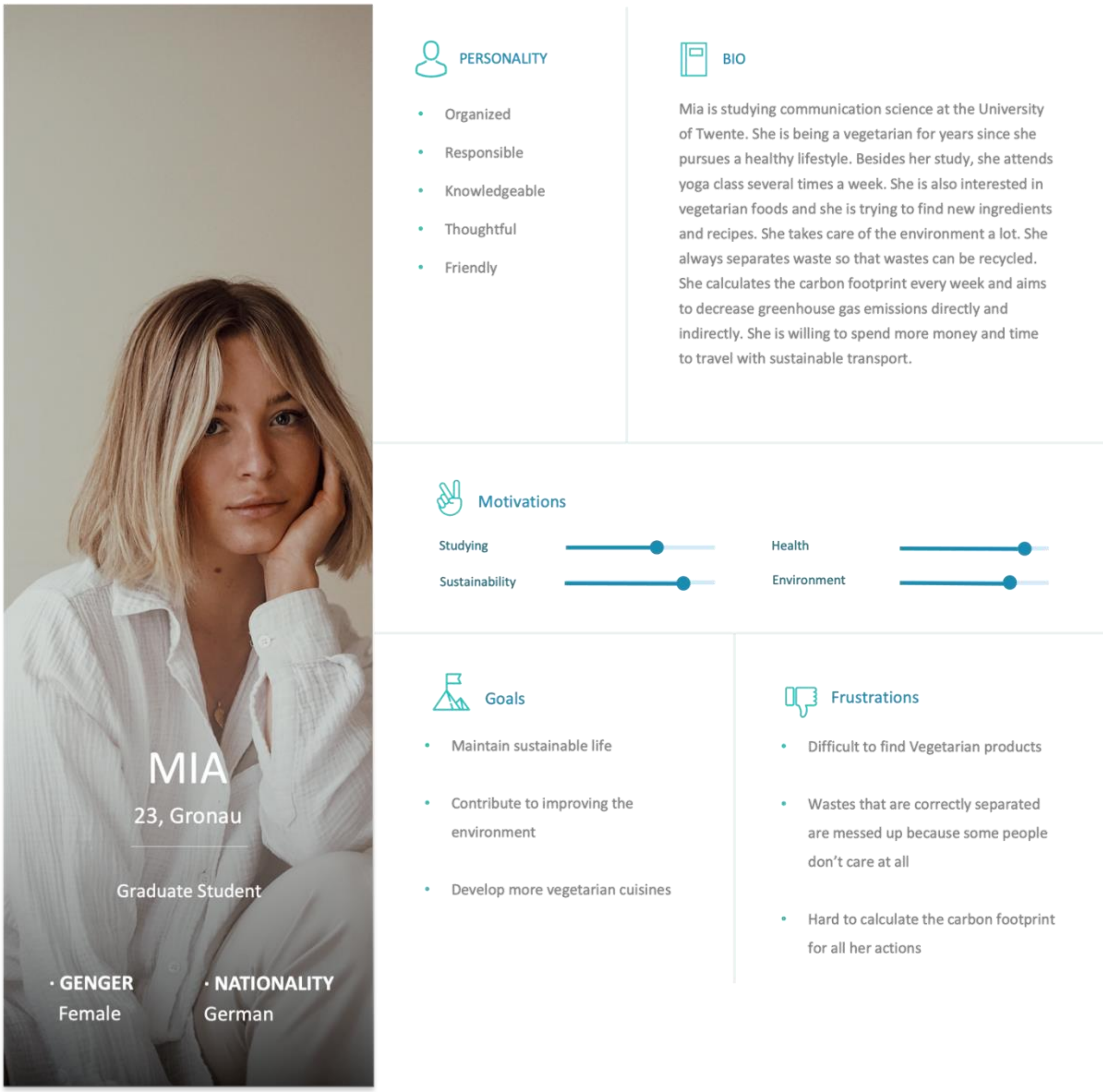

Figure 22 Mia

5.1.3 Nick

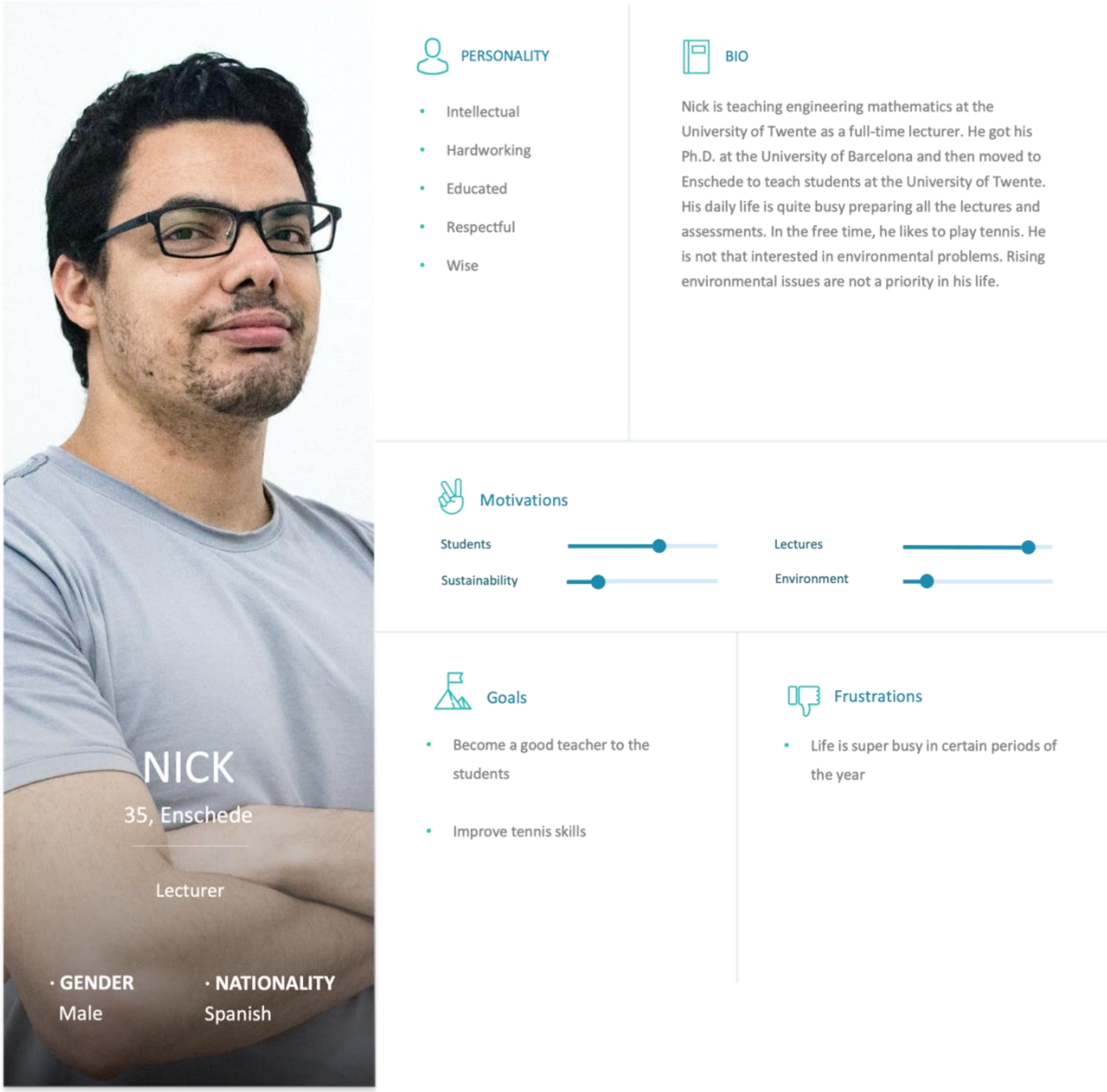

Figure 23 Nick

5.2 Interaction Scenario

To describe the possible interaction with the system, an interaction scenario is present. Every five scenarios are based on the personas.

5.2.1. User passing by

It is 10:37, and Nick is moving from his previous lecture to the next one, which starts at 10:45. He is in hurry to get to the next lecture room. While walking, he looks around the hallway. He spots the UT waste island with a big screen hanging above. From a distance of 15 meters, it is hard to see for him what is presented on the screen, but as he gets closer he notices it represents information about waste disposal at this specific waste island. Since he is busy now and he has no waste to dispose of he just walks past it while glancing at the information presented on the screen.

5.2.2. User interested in the waste island

It is 10:30, and James is moving from his previous lecture to the next one, which starts at 10:45. While walking, he looks around the hallway. He spots the UT waste island with a big screen hanging above. This time James wonders how it works, so he moves closer, and when he is a 0.5 meters distance from the waste island, the screen suddenly changes to the selection screen, and the light blinks under the cover of the waste island. He is curious about how to use this screen and he selected a random waste type, a cup. He reaches for the screen and touches the circle in which the cup is presented. After clicking the cup, the screen changes again, this time showing different types of cups. A paper cup is presented between the several cups on the screen. James looks at the different cups presented on the screen and decides to choose a paper cup randomly. He, again, reaches his hand to the screen and touches the circle in which the paper cup is presented. After touching, the whole circle changes color. James notices that it turns gray. The paper cup in the circle has changed into text: “This is Residual” it says. In addition, the lights around the residual waste bin of the waste island have a white color now, whereas the other waste bins don’t have lights. However, James doesn’t have any waste to dispose of, he is just interested in how the system works, therefore he doesn’t throw away anything. After 5 seconds, the screen goes back to the first step of selecting the waste type. Then he decides to leave the waste island. When he is further than 0.5 meters away from the screen, it changes to the information screen.

5.2.3. User disposes of waste *correctly* using the screen

It is 12:45, and Mia is finished with her lecture. Her lunch break is starting, which means she can finally relax with her friends for a bit. They decided to meet up at their standard lunch spot in the cafeteria, and Mia is excited to meet them. She holds an empty paper coffee cup from the previous lecture, which she would like to dispose of. While walking, she looks around the hallway to find a waste island to throw away her empty paper coffee cup. She spots a waste island with a big screen hanging above it. From a distance of 15 meters, it is hard to see for her what is presented on the screen, but as she gets closer she notices it represents information about waste disposal at this waste island. As she moves closer, the information becomes more clear, and from a distance of about 5-10 meters, the information is readable. She notices that it describes the waste correctly disposed of, and how much waste gets recycled. She moves even closer, and when he is a 0.5 meters distance from the waste island, the screen suddenly changes, and the light under the cover of the waste island blinks. It surprises Mia, as she did not expect the changing of screens to happen. Instead of the information screen, a selection screen is visible now. The selection screen shows her different types of waste, and the text "touch to select your waste type". Since Mia is not sure where to dispose of the paper cup, she would like to use the screen. A cup is one of the different types of waste presented on the screen. Mia connects the cup in her hand to the cup on the screen, as they look most similar, and decides to select the cup image. She reaches for the screen and touches the circle in which the cup is presented. After clicking the cup, the screen changes again, this time showing different types of cups. A paper cup is presented between the several other cups on the screen. Mia looks at the different cups presented on the screen, and can easily find the paper cup she holds in her hands at this moment. She, again, reaches her hand to the screen and touches the circle in which the paper cup is presented. After touching, the whole circle changes color. Mia notices that it turns gray. The paper cup in the circle has changed into text: "This is Residual" it says. In addition, the lights around the residual waste bin have a white color now, whereas the other waste bins are not lit up. Mia now knows she has to dispose of the empty paper coffee cup into the residual waste stream, which surprises her. She thought paper cups needed to be disposed of in the paper waste stream. She takes her cup and throws it into the residual waste stream. The screen changes again, showing a checkmark with the text "Good job! Thank you for disposing of your waste correctly". She feels proud of herself due to the feedback. As she walks more than 0.5 meters away from the waste island, she turns to look back at the screen and notices that the information screen changed. The percentage of correctly separated waste presented on the screen has increased. She realizes that she has influence over the information presented on the screen. She is glad she did dispose of her waste correctly because she cares for the environment and did not want to ruin the score on the screen. Mia continues to walk towards the cafeteria to meet his friends, she is excited to tell them about her experience with the waste island.

5.2.4. User disposes of waste *incorrectly* using the screen

It is 12:45, and Nick is finished with his second lecture. His lunch break is starting, and he holds an empty paper coffee cup from the previous lecture, which he would like to dispose of. He spots a waste island with a big screen above it. From a distance of 15 meters, it is hard to see for him what is presented on the screen, but as he gets closer he notices it represents information about waste disposal at this waste island. As he moves closer, the information becomes more clear, and from a distance of about 5-10 meters, the information is readable. He notices that it describes how waste is correctly disposed of, and how much waste gets recycled. He moves even closer, and when he is a 0.5-meter distance from the waste island, the screen suddenly changes, and the light under the cover of the waste island blinks. It surprises Nick, as he did not expect the changing of screens to happen. Instead of the information screen, a selection screen is visible now. The selection screen shows him different types of waste, and the text "touch to select your waste type". A cup is one of the different types of waste presented on the screen. Nick connects the cup in his hand to the cup on the screen, as they look most similar, and decides to select the cup image. He reaches for the screen and touches the circle in which the cup is presented. After clicking the cup, the screen changes again, this time showing different types of cups. A paper cup is presented between the several cups on the screen. Nick looks at the different cups presented on the screen, and can easily find the paper cup he holds in his hands at this moment. He, again, reaches his hand to the screen and touches the circle in which the paper cup is presented. After touching, the whole circle changes color. Nick notices that it turns gray. The paper cup in the circle has changed into text: "This is Residual" it says. In addition, the lights around the residual waste bin have a white color now, whereas the other waste streams are not lit up. However, Nick thought paper cups needed to be disposed of in the paper waste stream based on his knowledge. Therefore, the screen must be wrong, he thinks. He takes his cup and throws it into the paper waste bin. The screen changes again, showing a cross with the text "Your waste was disposed of incorrectly". He is disappointed. He really thought he had to dispose of his empty paper coffee cup in the paper waste bin, but now he doubts his knowledge. As he walks more than 0.5 meters away from the waste island, he turns to look back at the screen and notices that the information screen changed. The percentage of incorrectly separated waste presented on the screen has increased. He realizes that he has influence over the information presented on the screen. He is disappointed in himself that he did not listen to the information from the screen, as he now ruined the score on the screen.

5.2.5. User disposes of waste without using the screen

It is 10:40, and Mia is moving from his previous lecture to the next one, which starts at 10:45. She is in a slightly stressful mood since she does not know if she will make it to her lecture on time. She holds an empty plastic coffee cup that she got from Starbucks. While walking, she looks around the hallway to find a waste island to throw away her empty plastic coffee cup. She spots a waste island with a big screen above it. From a distance of 15 meters, it is hard to see for her what is presented on the screen, but as she gets closer she notices it represents information about waste disposal at this specific waste island. As she moves closer, the information becomes more clear, and from a distance of about 5-10 meters, the information is readable. She does not read the information presented, as she is stressed about being late for her lecture. She moves closer to the waste island to throw away her waste, and when she is a 0.5 meters distance away from the waste island, the screen suddenly changes. Lights around the waste island stream openings blink. It surprises Mia, as she did not expect the changing of screens to happen. Instead of the information screen, a selection screen is visible now. The selection screen shows her different types of waste, and the text “touch to select your waste type”. Since Mia already knows where to dispose of the empty plastic cup and has limited time to get to the lecture, she does not interact with the screen. She does not know how long interacting with the screen will take, and does not want to waste her time. Therefore she takes her empty plastic coffee cup and therefore throws it away in the PMD waste stream. After her waste was disposed of in the PMD waste stream, Mia notices that the screen changed again. Now it shows the text ‘Your waste is disposed of in the PMD stream. Did you separate correctly?’. She reads it and touches the “YES” button since she is confident with her decision, and walks away quickly to go to her lecture. When she is further than 0.5 meters away from the screen, it changes to the information screen which shows waste statistic of this waste island. As Mia is quickly walking away from the waste island, she does not notice the change on the screen.

5.3 System Requirements

The system requirements, including functional and non-functional requirements of the interactive waste island are defined.

5.3.1 Functional Requirements

Table 3 shows the functional requirements of the system. The requirements are categorized with the MoSCoW method.

Category	Requirement
Must	The waste island must sense the user in front of the waste island if the user is less than 0.5m away from the waste island.
	The waste island must recognize which bin the user disposed of the waste.
	The system must return to the initial state if the user leaves, more than 0.5m away from the waste island.
	The system must return to the initial state if there is no new input within 30 seconds.
	The system must return to the initial state if there is no answer from users within 5 seconds when they disposed of the waste without using the interactive screen.
	The system must update the number of wastes disposed of in each waste bin for the information screen.
	The system must update the correct waste disposal ratio, which can be calculated by the number of correct waste disposal divided by the total number of wastes for the information screen.
	The screen must detect the touch interaction of the interactive selection screen.
	The waste island should detect any size of waste disposed in each of the waste bins.
Should	The system should check if waste is disposed of in the correct waste bin.
	The system should have clearly visible lights under the cover of the waste island.
	The color of the lights for each waste bin should be similar the waste bin color.

Table 3 Functional Requirements

5.3.2 Non-functional Requirements

Table 4 shows the non-functional requirements of the system. The requirements are also categorized with the MoSCoW method.

Category	Requirement
Must	The system must be easy to interact.
	The system must generate relevant feedback to the interactive screen.
	The system must give awareness of the waste separation.
	The system must be near the waste island.
Should	The system should be interesting to use.
	The system should be easy to install in the existing UT waste island.
	The system should not interfere with the existing features of the original UT waste island, such as size, holes, or stickers.

Table 4 Non-Functional Requirements

5.4 Functional Architecture

The system has two sub-systems, the interactive waste island and the interactive screen. The brain of the system, which controls everything in the system, is the interactive waste island. I develop the interactive waste island and all data exchange protocols and my co-researcher, Senna, develops the interactive screen and contents for the screen.

5.4.1 Level 0 Decomposition

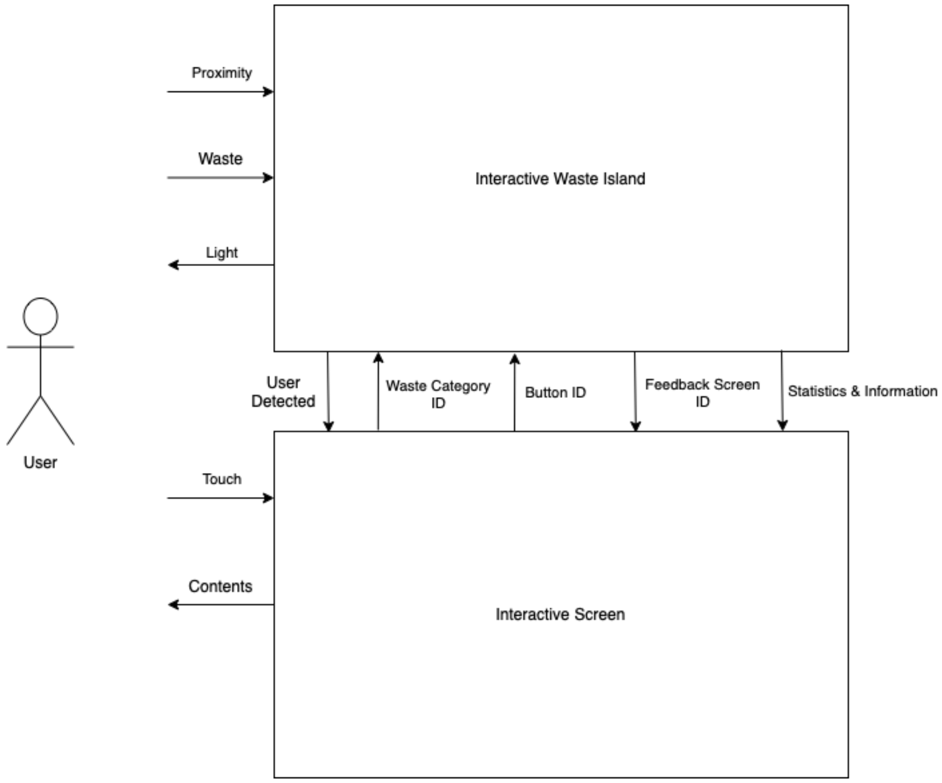

Figure 24 Level 0 Decomposition

Figure 24 shows the black-box model of the waste island and the screen. The input from the outside world to the waste island is proximity and waste and the output for the user is light. The waste island is connected to the screen and they both exchange necessary data. The interactive screen has one touch input from the user and the contents of the screen as output. The waste island sends data, which is the state of the user detection, feedback screen ID and statistic & information. The screen provides waste category ID and bin ID to the waste island. More details will be in level 1 decomposition.

5.4.2 Level 1 Decomposition

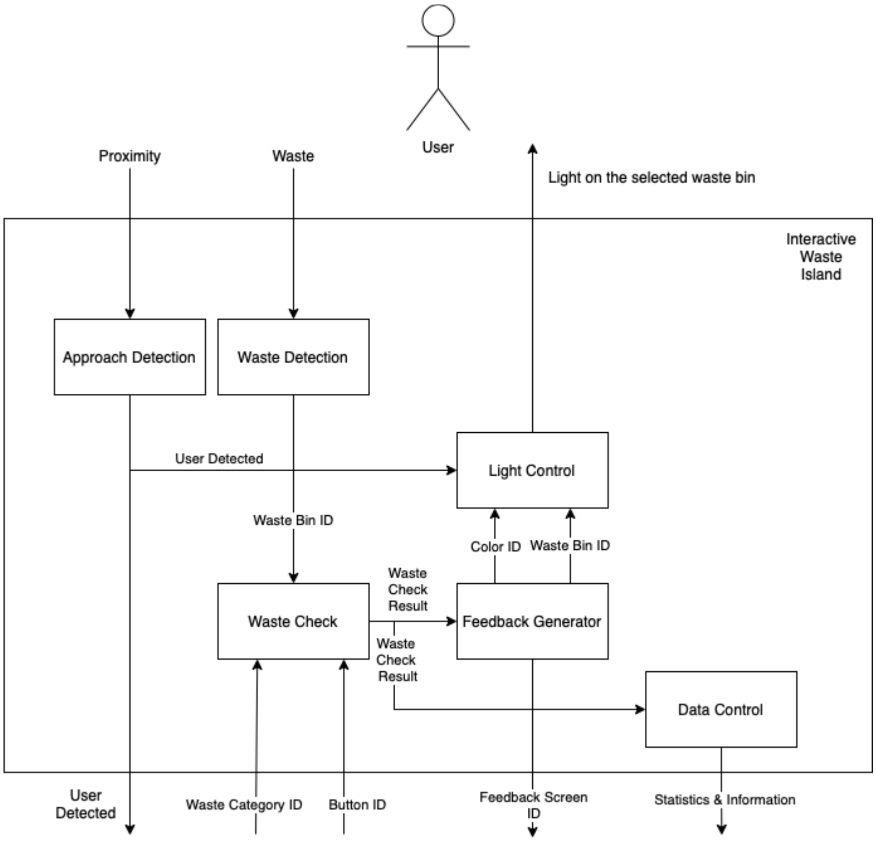

Figure 25 Level 1 Decomposition

Figure 25 shows the level 1 decomposition of the interactive waste island. Once the system has power, the approach detection function will keep performing measurements. If the distance between the waste island and the user is less than 0.5m, it will notify the interactive screen and the light control function that the user is detected. Then the screen will change to the selection screen and the light control will blink the lights under the cover of the waste island. If the user disposed of the waste, then the waste detection function will recognize the waste bin where the waste is disposed of and send its bin ID to the waste check function. The waste check function will get the bin ID from the waste detection function and waste category ID from the selected waste item and the button ID if the user disposed of the waste without using the selection screen from the interactive screen. Afterward, this function will check whether the waste bin ID and selected waste ID are matched or not and the result from the button ID. It will send the result to the feedback generator and the data control functions. Once the feedback generator gets the result, it will generate appropriate feedback based on the result data and will send a feedback screen ID which contains feedback contents provided by Senna to the screen and a color ID & waste bin ID to the light control. Light control blinks the light on the selected waste bin. After the data control function gets the result data, it will make statistical data and information based on the results and update it to the information screen.

5.4.3 Level 2 Decomposition

Level 2 decomposition describes how each function works specifically and explains the details of the input and output data at each function.

5.4.3.1 Approach Detection

Figure 26 shows the level 2 decomposition of the Approach Detection function. Approach Detection gets the distance data from the get distance function. After that, it checks the distance whether it is less than 0.5m. Finally, it sends the output whether the user is detected or not.

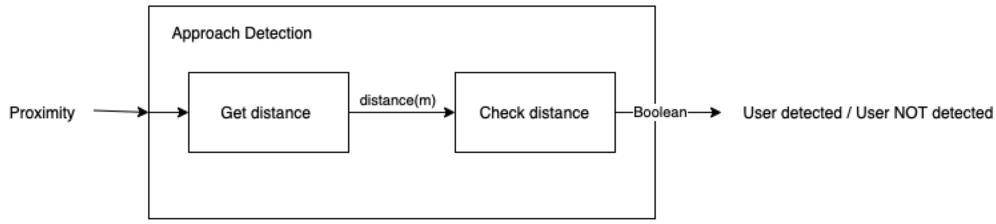

Figure 26 Approach Detection

5.4.3.2 Waste Detection

Figure 27 shows the level 2 decomposition of the Waste Detection function. This function gets sensor data from all four waste bins. If the waste is disposed of in one of the bins then it sends its waste bin ID as an output. Waste Bin ID can be “residual”, “pmd”, “organic” or “paper” since there are four waste bins in the waste island. For instance, if waste is detected in the PMD waste bin, then the waste bin ID will be “pmd”.

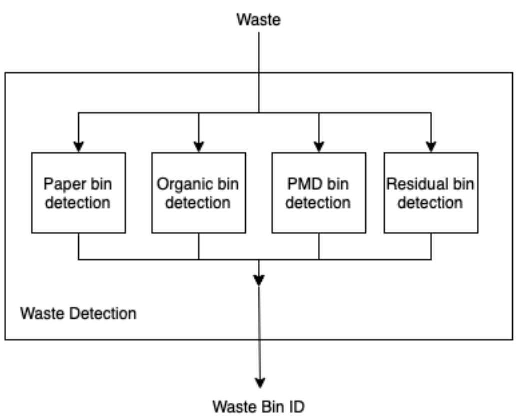

Figure 27 Waste Detection

5.4.3.3 Waste Check

Figure 28 shows the level 2 decomposition of the waste check function. This function gets the waste bin ID from the waste detection function, the waste category ID or selected button ID from the screen, and adds the timestamp when those data arrive. The waste category ID can be "residual", "pmd", "organic" or "paper" because the waste should be classified as one of these four categories at UT campus. The selected button ID is collected when the user disposed of the waste without using the selection screen. The interactive screen asks users about the correct waste disposal, then the user can choose one of these three answers: "yes", "no" or "not sure" and these three can be waste bin ID. "not sure" option is given to get an insight of how confident users are with their decision. Then it checks if the waste bin ID matches the selected waste category ID if the selection screen is used or just gets the selected button ID if the waste is disposed of but the selection screen is not used. It also checks if the system is time out. If more than 30 seconds passed after the waste category ID arrived or more than 5 seconds passed after the button ID arrived then time out. Time out is checked because the system should return to the initial state if there is no extra input. The output of this function is the waste check result, which can be one of these: waste not selected on the screen, waste is correctly or incorrectly disposed of, user not sure about the waste disposal, or too long time passed since the first input. Please see table 5 for how this output is determined based on waste category ID, waste bin ID and the condition.

Waste Category ID	Waste Bin ID	Condition	Result
O	X	Time > 30s	Time out
O	X	Time ≤ 30s	Waiting
O	O	Waste Bin ID = Waste Category ID	Correct, Waste Bin ID
O	O	Waste Bin ID ≠ Waste Category ID	Incorrect, Waste Bin ID
X	O	Time > 5s	Time out
X	O	Button ID = None	Not selected, Waste Bin ID
X	O	Button ID = "yes"	Not selected & correct, Waste Bin ID
X	O	Button ID = "no"	Not selected & incorrect, Waste Bin ID
X	O	Button ID = "not sure"	Not selected & not sure, Waste Bin ID

Table 5 Waste Check output

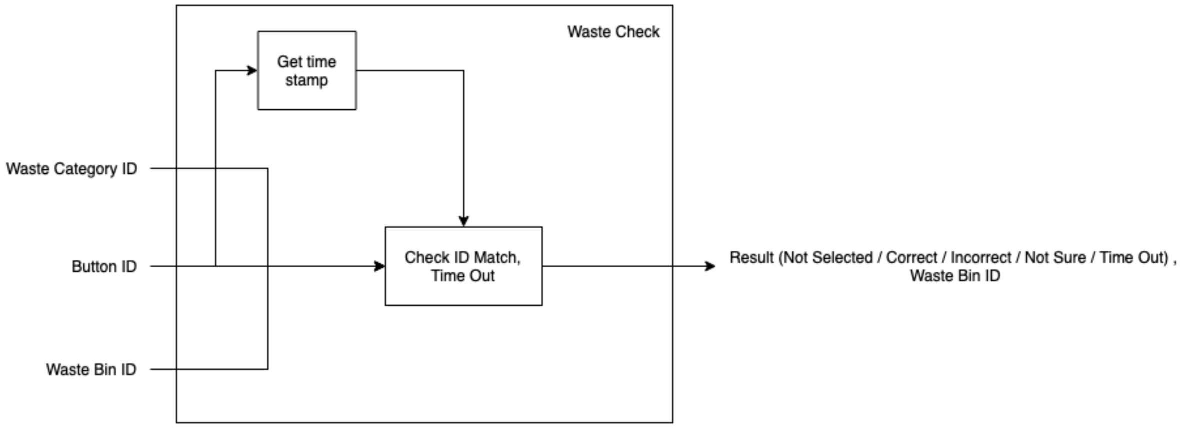

Figure 28 Waste Check

5.4.3.4 Feedback Generator

Figure 29 shows the level 2 decomposition of the feedback generator function. This function gets the result from the waste check function. Then checks the result of waste disposal and creates feedback based on the result. This feedback is delivered to the screen as a feedback screen ID and to the light control as a color ID and waste bin ID. Please see table 6 for how it works based on the result. The feedback screen ID is the unique screen ID of the premade feedback screen. This can be “correct disposal”, “incorrect disposal”, “questions”, “thank you”, “education” or “reset”. For example, if feedback screen ID “correct disposal” is delivered to the interactive screen, then the screen displays “the waste is correctly disposed of”. If feedback screen ID “education” is delivered to the screen, then the screen shows all the waste items belongs that waste bin category to educate users. If the feedback screen ID is “reset” then the system goes back to the initial state. Color ID is the color of the lights, this can be “white”, “red”, “green” and “blue”, and this is generated if the result is “Waiting”.

Result	Feedback Screen ID	Waste Bin ID	Color ID
Time out	“reset”	X	X
Waiting	X	O	O
Correct, Waste Bin ID	“correct disposal”	X	X
Incorrect, Waste Bin ID	“incorrect disposal”	X	X
Not selected, Waste Bin ID	“question”	O	X
Not selected & correct, Waste Bin ID	“thank you”	O	X
Not selected & incorrect, Waste Bin ID	“education”	O	X
Not selected & not sure, Waste Bin ID	“education”	O	X

Table 6 Feedback Generator output

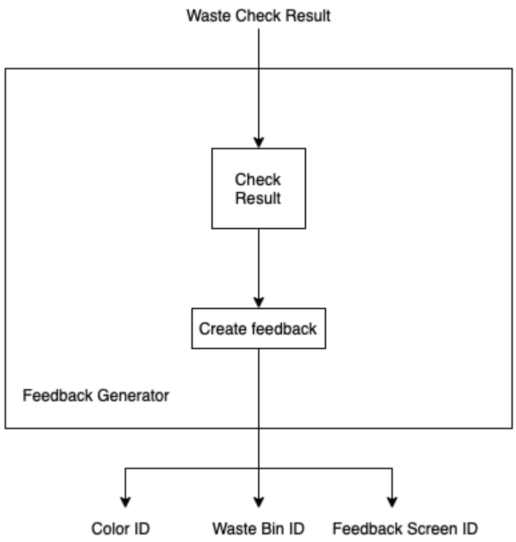

Figure 29 Feedback Generator

5.4.3.5 Data Control

Figure 30 shows the level 2 decomposition of the data control function. Data control also gets waste check results and generates statistical data based on the "result" from the waste check function and waste bin ID. It sends and updates this statistical data and information to the information screen. The statistical data includes the number and ratio of waste disposed of in each waste bin, and the number and ratio of correct, incorrect and unsure waste disposal.

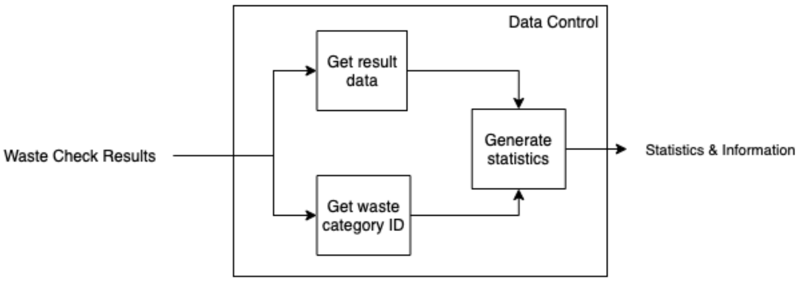

Figure 30 Data Control

5.4.3.6 Light Control

Figure 31 shows the level 2 decomposition of the light control function. This function gets data on whether the user is detected or not and color ID from the feedback generator. If the user is detected, it generates lights with pre-defined colors. If it gets a color ID and waste bin ID then generate lights only on the specific waste bin with a specific color. For example, if the color ID is "white" and the waste bin ID is "residual", the white light on the residual waste bin will be generated.

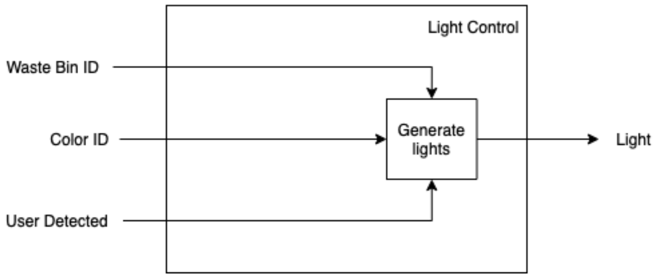

Figure 31 Light Control

5.5 Time Sequence Diagram

Figure 32 shows the time sequence diagram of the whole system, including the interactive waste island and the interactive screen. The system displays an information screen that includes the statistical data of the waste island as an initial state. If the user is detected within the 0.5m range in front of the waste island, then the selection screen is activated. The user now can see the selection screen and there are two options for the user, user or don't use the selection screen. Based on the user's decision, two cases exist. The first case is when the user is using the screen. In this case, the user selects the waste on the screen and the system lets the user know which category the waste belongs to. Then the user disposes of the waste and the system checks if the waste is ended up in the correct waste bin. Based on the result, the screen again displays the result of whether the waste is correctly or incorrectly disposed of. After 5 seconds later, the system returns to the initial state, the information screen. The other case is when the user doesn't use the screen. In this case, the user throws the waste without making any selection, and the system cannot check whether if correctly disposed of or not, however, the system can ask users whether their choice is right or not. If the user thinks the waste is disposed of correctly, then the "Thank you" screen will visible. If the user doesn't think the waste is correctly disposed of or is not sure about his/her decision, then the representative items that belong to that waste category will be displayed. After the process is finished and the user leaves the waste island, the system waits for 5 seconds and returns to its initial state, the information screen, and the statistics will be updated. If there was no input from the user, then the system waits for 30 seconds and returns to the information screen.

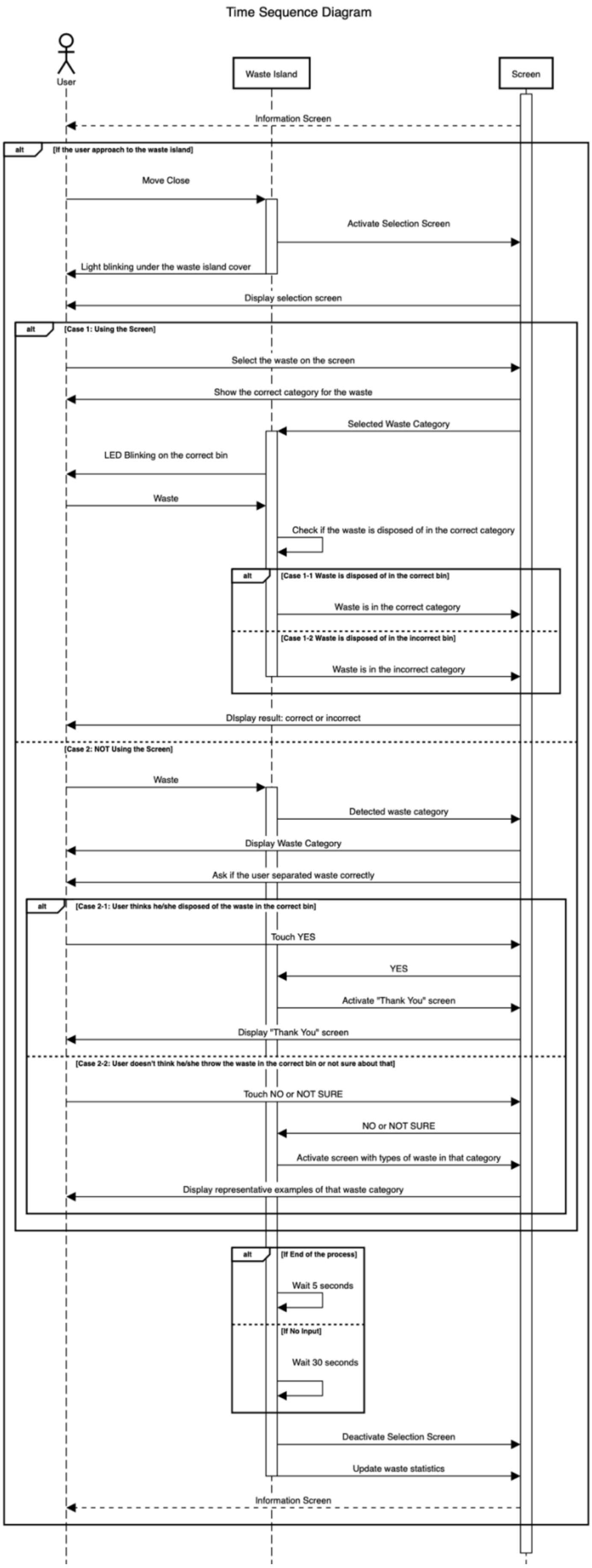

Figure 32 Time Sequence Diagram

6. Realization

In this chapter, the realization part of the project will be explained. From the level 1 decomposition in chapter 5, the key 6 functions are approach detection, waste detection, waste check, feedback generator, light control, and data control. Approach detection, waste detection, and light control need physical electric elements. Other functions can be programmed with the software.

6.1 Hardware

Hardware is an essential part of the system. To realize all required functions, electric components, touch screen, waste island and laptop is needed.

6.1.1 Electric Components

To make approach detection and waste detection and light control functional, electric components such as a microcontroller, proximity sensor, waste detector and light are required on the prototype.

6.1.1.1 Microcontroller

Figure 33 Arduino UNO R3

The Arduino UNO R3 (Figure 33) is used to control all sensors. Arduino is a microcontroller that has a number of digital pins and analog pins. Digital pins can read and write on or off state, and analog pins can read the range of analog values. The Arduino is selected since the microcontroller should be able to read and write both digital and analog values for the prototype.

6.1.1.2 User Proximity Detection

Figure 34 HC-SR04

To sense the proximity, the ultrasonic distance sensor HC-SR04 (Figure 34) is used. This sensor can measure a minimum distance of 2cm to a maximum of 400cm with an accuracy of 3mm. The effectual angle of the sensor is less than 15°. First, the transmitter (trigger pin) emits an ultrasonic sound, 40 kHz. This sound travels through the air and is reflected back if the wave meets an object. The receiver (echo pin) receives the reflected waves. To calculate the distance, equation 1 can be used. The speed is the speed of the sound in the air at 20°C, which is 343m/s. The time is the duration of time between the transmission and reception of the sound.

$$Distance = Speed \times Time$$

Equation 1 Distance calculation

This sensor is selected for the proximity sensor since it has a long range. Corridors in the UT building are mostly wide, thus it is a better idea to secure enough range for detection because the trigger point can be set within a wider range of the distance.

6.1.1.3 Waste Detection

Figure 35 GP2Y0A21YK0F

To detect the waste at each waste bin, an infrared distance measuring sensor GP2Y0A21YK0F (Figure 35) is used. It has a 10 to 80cm distance measuring range. The infrared light is emitted and reflected back from the objects. Then the analog voltage is returned and the distance can be determined based on this analogy voltage as in figure 36.

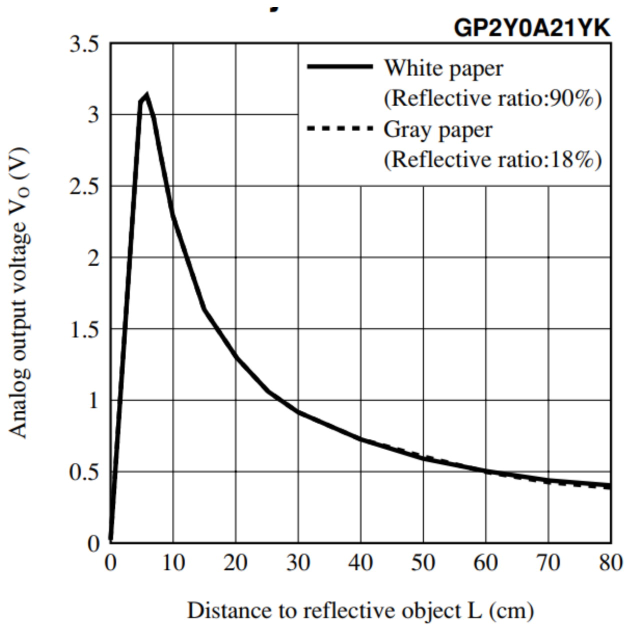

Figure 36 Distance measuring characteristics

This sensor is selected for waste detection because the size of the sensor is compact and short-range. The distance between the cover of the waste island and the waste bin is less than 2cm. The sensor should fit into that small space, therefore need to be compact. The diameter of the waste bin is 23cm so the long-range sensor is not necessary.

6.1.1.4 Light

Figure 37 LED Strip NeoPixel RGB

For the light, LED Strip NeoPixel RGB (Figure 37) is used with 29 LEDs in each strip. These LEDs have a WS2812B light driver thus all RGB LED lights on the strip are individually addressable. Usually, RGB LED needs three pins for each color, red, green and blue. However, NeoPxiel RGB LED only needs a single pin to control all RGB lights. This is a reason why it is selected for this prototype. All the wires should be placed inside the waste island, therefore using fewer wires is the better option otherwise organizing wires will complicate.

6.1.2 Touch Screen

Figure 38 CTOUCH

For the touch screen, CTOUCH is used (Figure 38). CTOUCH is an interactive touch screen that can display the computer screen. CTOUCH is attached to the stand, this screen can easily be placed behind the waste island. The touch screen is also easy for the users to navigate the contents.

6.1.3 Waste Island

Figure 39 Waste Island

The waste island is provided by the UT-CFM (Figure 39). It is the exact same waste island that is placed on the campus and that the UT community members are using. This waste island has four waste bins for each waste category.

6.1.4 Laptop

Figure 40 Laptop

To operate the software and process the screen image, the laptop is used (Figure 40). The laptop should have enough performance to process all the required software.

6.2 System overview

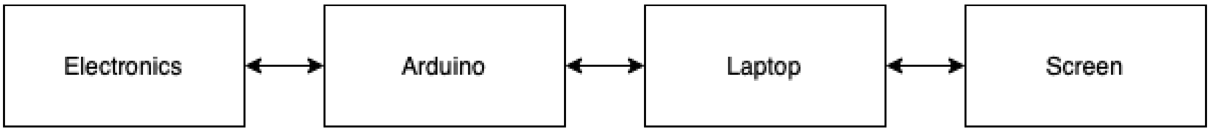

Figure 41 Hardware connection scheme

Figure 41 shows the connections of the hardware. All electric components are connected to Arduino and Arduino exchanges data with the Laptop via serial communication. The laptop is connected to the screen with the HDMI cable and touch sensor cable, the screen displays duplicated screen of the laptop.

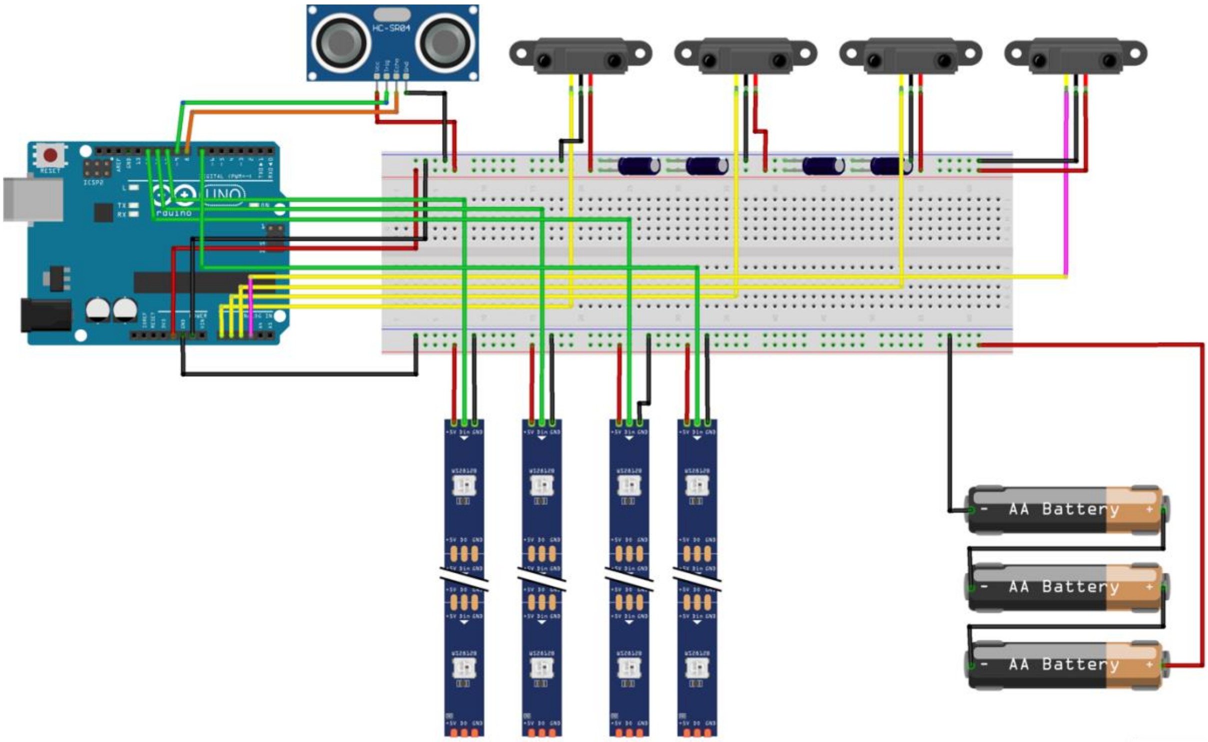

Figure 42 Wiring scheme of the electronics

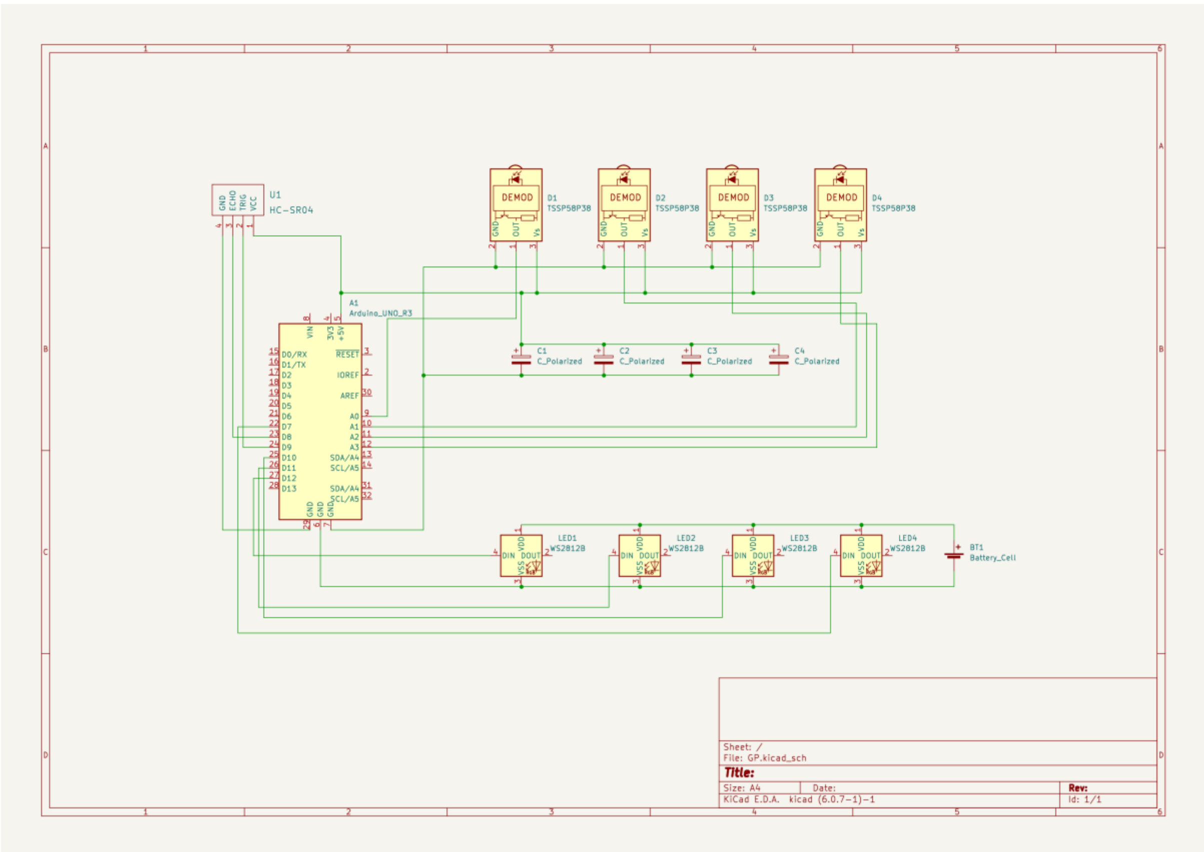

Figure 43 Circuit Diagram of the system

Figure 42 shows the electronics wiring scheme. Each electric component is connected to Arduino pins. Figure 43 shows the detailed circuit connection of the electric components.

Ultrasonic distance sensor HC-SR04

VCC is connected to the 5V power from the Arduino, and GND is connected to the ground pin of the Arduino. Trig is the trigger pin and Echo is the receiver pin which is connected to D9 and D8 on the Arduino.

Infrared distance sensors GP2Y0A21YK0F

VCC is connected to the 5V from the Arduino, and GND is connected to the GND pin of the Arduino. The by-pass capacitor of 10uF is inserted between VCC and GND of the infrared distance sensors to stabilize the power supply. The data lines are connected to A0, A1, A2 and A3 since it returns the analog output. A0 reads the value from the residual waste bin, A1 reads the value from the PMD waste bin, A2 reads the value from the organic waste bin and A3 read the value from the paper waste bin.

Neopixel LED Strips

To power the LEDs external 4.5V power is supplied by three 1.5 AA batteries. Each Neopixel LED can draw a maximum of 60mA at the maximum brightness, thus each strip can draw a maximum of $60 \times 29 = 1740mA$. The maximum current that can be drawn safely from the 5V pin of the Arduino is 500mA so LEDs can be powered by 5V pin from the Arduino, extra power supply is required to protect Arduino board. VCC of the LEDs are connected to the positive side of the battery pack and GND is connected to the negative side. Data lines are connected to the digital pins of the Arduino. The LED for the residual bin is connected to D10, the LED for the PMD bin is connected to D11, the LED for the organic bin is connected to D12, and the LED for the paper bin is connected to D7.

Arduino is connected to the laptop with a USB cable and powered by the laptop.

6.3 Software

Software is used to control hardware, two software are used in this system.

6.3.1 Arduino IDE

All the sensors connected to the Arduino are controlled by Arduino IDE. This is open-source software that is a simplified version of C++. The Arduino software performs functions of approach detection, waste detection, and light control (See Appendix A).

6.2.2 Python

For the screen software and connecting electronics and screen, Python is used. The screen is built using Pygame. Pygame is an open-source python library that is designed for programming video games. Since this library is video game friendly, it’s easy to build an interactive screen using this library. Python performs functions of the waste check, feedback generator, and data control (See Appendix B).

6.4 Final Prototype

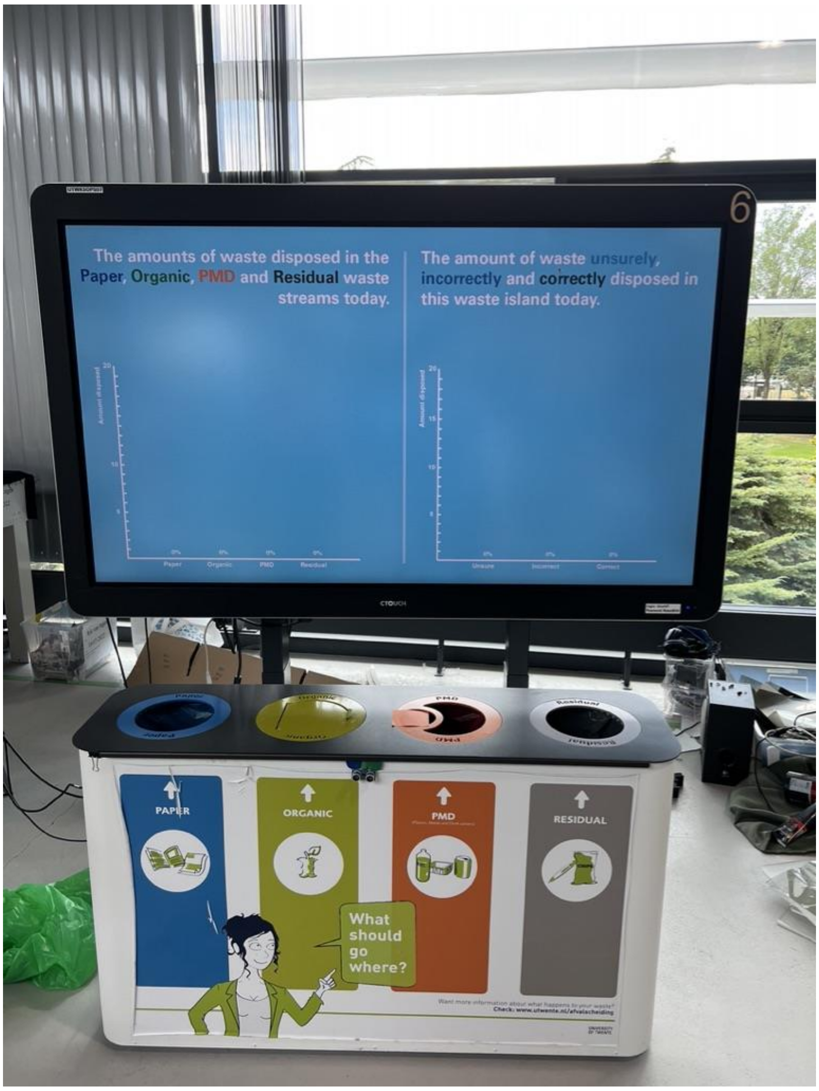

Figure 44 Prototype exterior

Figure 44 shows the final prototype. Only the ultrasonic sensor is visible in the front of the waste island since all other electrical components are installed inside the waste island. The touch screen is standing behind the waste island.

Figure 45 Inside the waste island

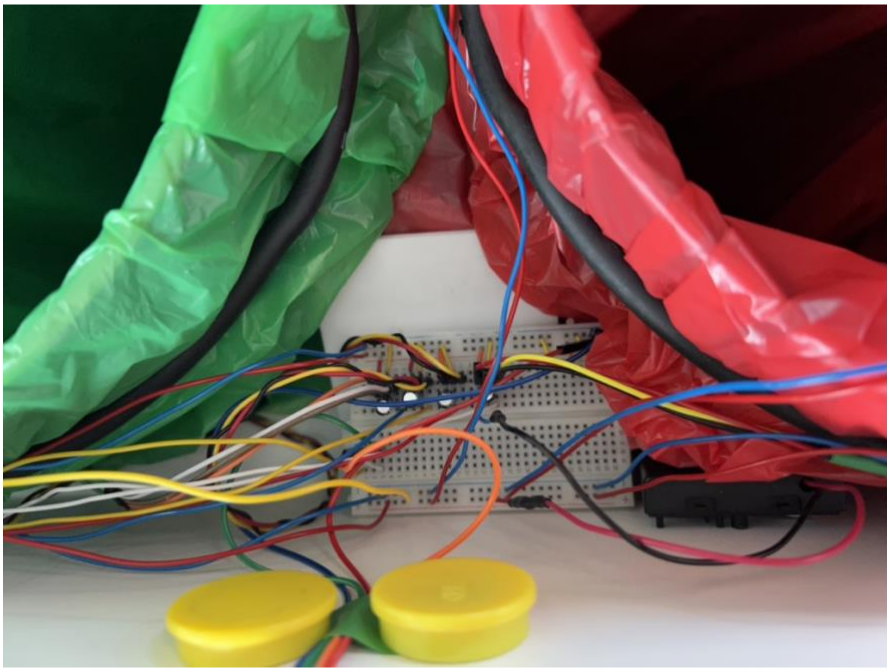

Figure 46 Breadboard in the waste island

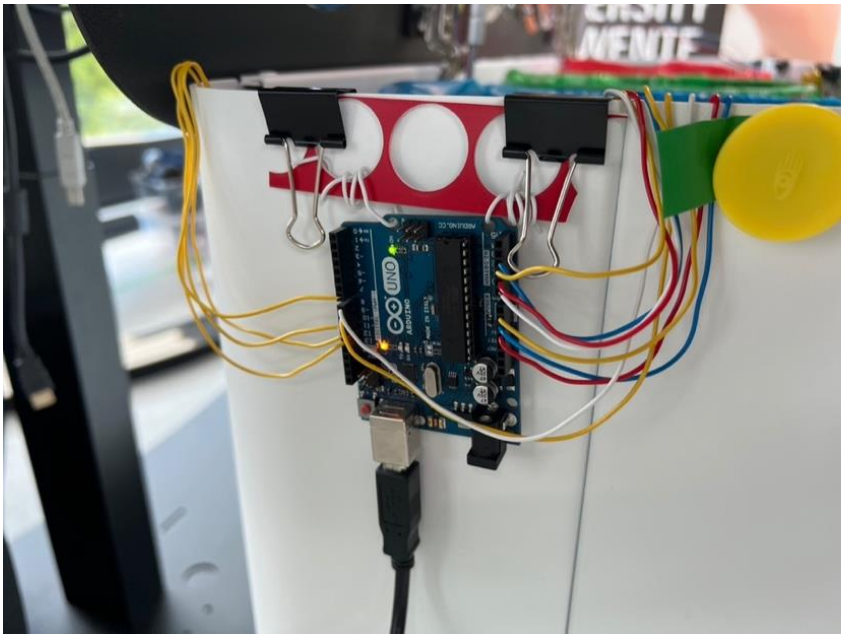

Figure 47 Arduino on the left side of the waste island

Figure 45 shows the inside of the waste island. The ultrasonic sensor, LED strips and infrared distance sensors are connected to the breadboard in the middle of the waste island with wires (Figure 46). This breadboard is connected to the Arduino. Figure 47 shows the Arduino attached to the left wall of the waste island.

Figure 48 Ultrasonic sensor on the front side of the waste island

Figure 48 shows the ultrasonic sensor is installed in the center of the front side of the waste island.

Figure 49 Infrared sensors above waste bins

Figure 49 shows where and how the infrared sensors are installed. The infrared distance sensor is installed on top of the front wall of the waste island toward another side of the wall. They are located in the middle of the waste bin thus they can detect the waste easily because most of the waste will pass the center of the waste bin.

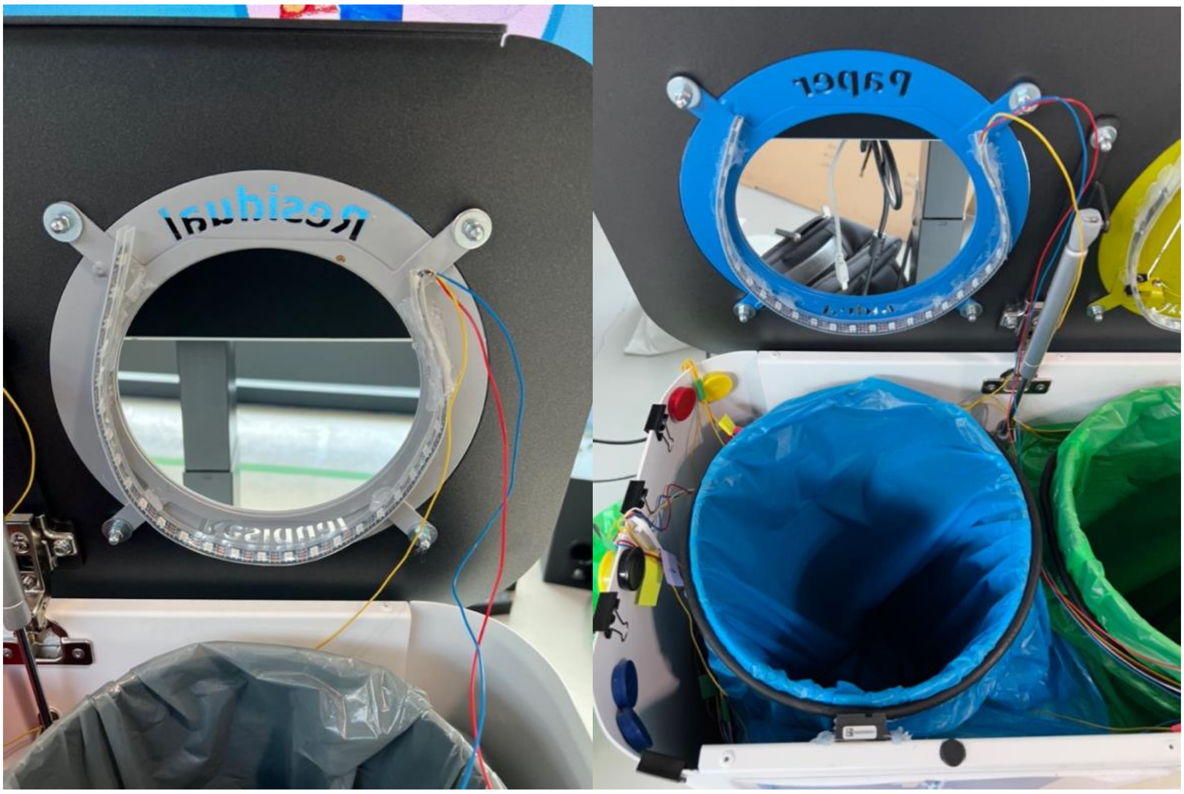

Figure 50 LED strips under the entrance of the waste bin

Figure 50 shows the LED strips installed under the cover of the waste island, under the entrance of the waste bin with a U shape. The color of the light is matched with a similar color to the waste bin. LED for the residual bin emits white light, LED for the PMD bin emits red light, LED for the organic bin emits green light, and LED for the paper bin emits blue light.

Practical Problems

There was ADC(Analog to digital converter) noise from the infrared sensor when first time reading the value since this sensor reads analog input, and these sensors could not be functional with noise. To reduce the noise the low pass filter is applied in the software. After applying the low pass filter, the noises are decreased and the values can be used at a stable level.

For the touch screen interaction, the mouse click event is used for the touch event. However, python and pygame require more recourses to process the mouse click event since the program should be ready to catch that specific event. The python becomes really slow and it takes a while to recognize the mouse click event. Also on the CTOUCH, the mouse click event is not always initiated with a single touch, it needs multiple touches to trigger the mouse click event.

To solve this, the touch action is triggered right after the mouse position is moved to the inside of the item circle and not waiting for the mouse click event. Then the problem is if the mouse position is keep staying in the same location, then touch action is triggered every second. To prevent this, the touch action is only triggered if the mouse position is moved to another coordinate. If the mouse point stays in the same position then it triggers only single touch action. By using this method the touch interaction became more smooth than before.

6.5 Performance

The performance of the system includes all possible outputs that the prototype can give to users. It has an information screen, a selection screen, and generated feedback based on the waste separation results.

6.5.1 Information Screen

Figure 51 Information Screen

The information screen shows the statistics of the waste data from the waste island. This is the default screen of the prototype. How many wastes are disposed of in each waste bin and the correct or wrong disposal proportion is displayed on the screen.

6.5.2 Selection Screen

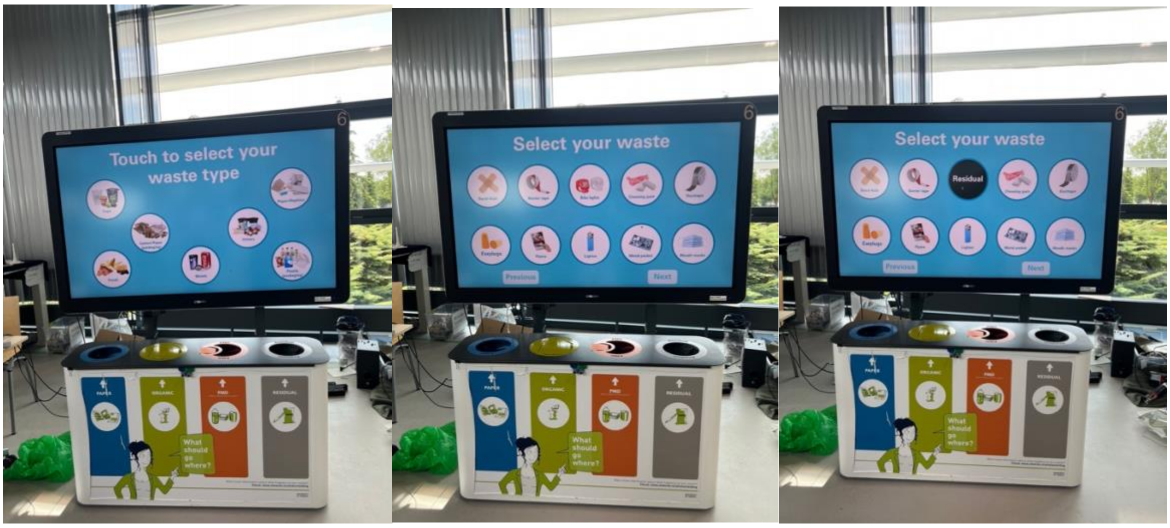

Figure 52 Selection Screen

Figure 53 Welcome lights

Figure 54 LED light under the PMD

Figure 52 shows the selection screen of the prototype. If the user is less than 50cm away from the ultrasonic distance sensor, then the selection screen is activated with welcome lights blinking

(Figure 53). The contents on the screen are developed by the co-researcher, Senna. Users can touch the screen to choose the waste they would like to dispose of. After the selection, the corresponding waste category of the waste is displayed and the LED on that waste category bin is blinking (Figure 54).

6.5.3 Correct or Incorrect disposal

Figure 55 User is disposing of the waste in the selected category

Figure 56 Correct waste disposal

Figure 55 shows the user is disposing of the waste in the selected category. After users throw away their waste, waste is checked and based on that the screen displays the result. Figure 56 shows the correct waste disposal, this will be displayed if the user puts the waste in the correct waste bin.

6.5.4 Waste disposal without using the screen

Figure 57 Screen displays the waste category

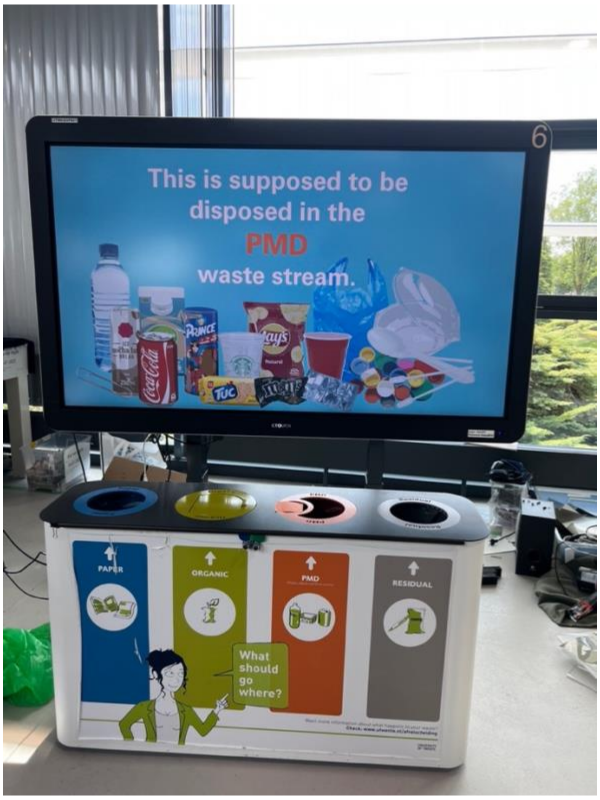

Figure 58 Education Screen

Figure 59 Thank you screen

Figure 57 shows the screen displays the waste category of the bin where the waste is disposed of when the user disposed of the waste without interacting with the screen. The screen asks if the user did a correct waste separation. Based on the user's answer either the education screen or thank you screen is displayed (Figure 58 and 59).

Category	Requirement	Results
Must	The waste island must sense the user in front of the waste island if the user is less than 0.5m away from the waste island.	△
	The waste island must recognize which bin the user disposed of the waste.	O
	The system must return to the initial state if the user leaves, more than 0.5m away from the waste island.	O
	The system must return to the initial state if there is no new input within 30 seconds.	O
	The system must return to the initial state if there is no answer from users within 5 seconds when they disposed of the waste without using the interactive screen.	O

	The system must update the number of wastes disposed of in each waste bin for the information screen.	O
	The system must update the correct waste disposal ratio, which can be calculated by the number of correct waste disposal divided by the total number of wastes for the information screen.	O
	The screen must detect the touch interaction of the interactive selection screen.	△
	The waste island should detect any size of waste disposed in each of the waste bins.	X
Should	The system should check if waste is disposed of in the correct waste bin.	△
	The system should have clearly visible lights under the cover of the waste island.	△
	The color of the lights for each waste bin should be similar the waste bin color.	O

Table 7 Functional requirements achievement

Table 7 shows whether the system satisfies function requirements. O means the requirements are fully met, △ means the requirements are partially met, and X means failed to satisfy the requirements.

7. Evaluation

In this chapter, the evaluation part of the project will be described. The goal of the evaluation is to answer the research question, finding whether this prototype can improve the waste separation quality and the user experience of the prototype. The evaluation is conducted in June 2022 in the Design Lab at the University of Twente.

7.1 User Testing

7.1.1 Setup

7.1.1.1 Samples, Assumption and Variables

The user group of this prototype is the UT community members. From the user group, 22 participants are randomly and independently recruited for the user evaluation. These participants are divided into two groups with equal numbers, thus 11 participants in each group. The first group will perform the task without the prototype being present and the second group will do the task while the prototype is presented in front of them. Every condition is controlled except the existence of the prototype. For the evaluation, it is assumed that the waste separation performance is normally distributed within each of the two groups, and the variance in each of the groups is equal. The independent variable in this research is the existence of the prototype when the participant is disposing of the waste. Then the dependent variable is the number of correct waste disposal, difficulty and confidence of the waste separation.

7.1.1.2 Experiment Environment

Figure 60 shows the experiment environment and figure 61 shows the simplified diagram. Two researchers are sitting behind the table and the prototype is standing right side of the table. All 15 waste is placed on the table. The participant is free to perform the task while the researchers are observing their performance. The prototype always stays in the same location, but the whole system is turned off (Figure 62) without any power supply when the first group is testing, thus the prototype is not interactable and functional, like a normal waste island. The prototype is working and functional when the second group is invited (Figure 63).

Figure 60 Experiment Environment

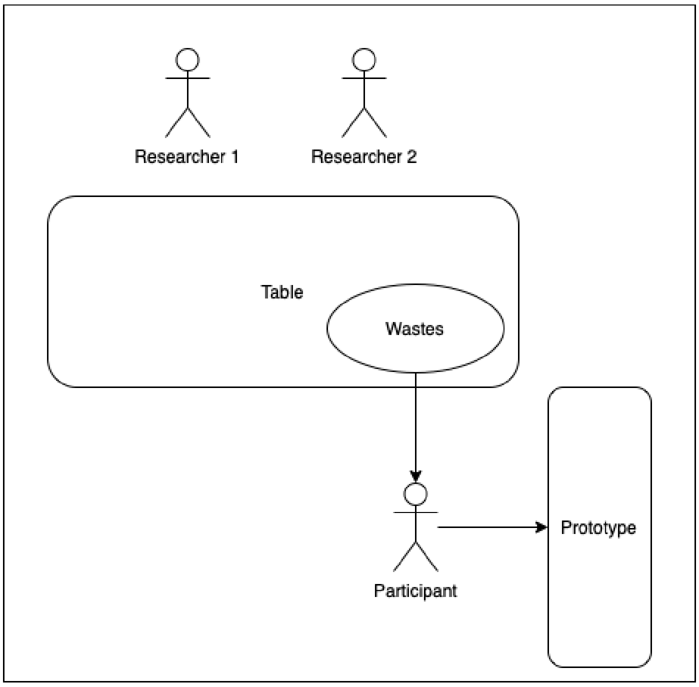

Figure 61 Simplified diagram of the experiment environment

Figure 62 Without the prototype

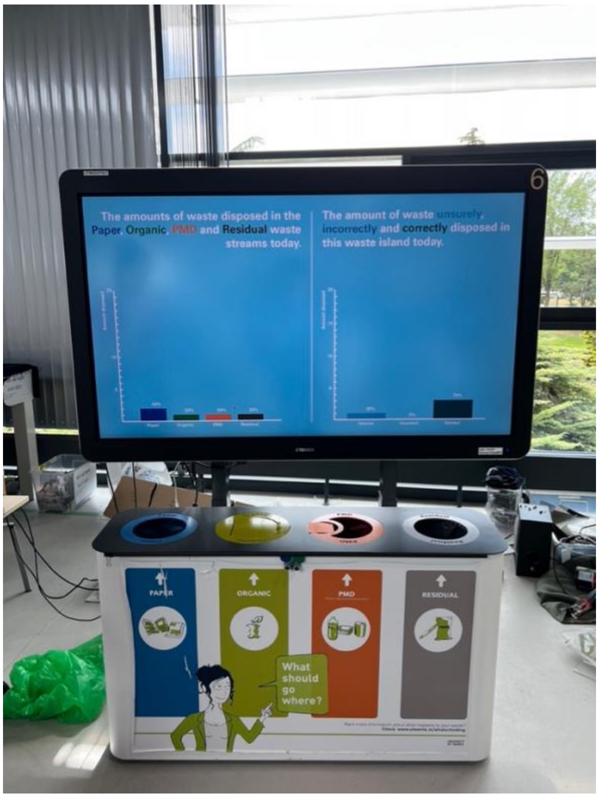

Figure 63 With the prototype

7.1.1.3 Task

The task for the participants in each group is separating 15 different wastes into the waste island in front of them. Table 8 shows the full list of wastes they need to separate and the correct category of the waste. These wastes are the waste that can be easily found on the UT campus. Figure 64 shows the wastes that participants need to dispose of. Wastes are all mixed and provided on the table. The participants need to take these wastes and dispose of them in the waste bin where they think or believe the correct category of the waste.

Figure 64 Wastes given to participants

Waste	Category
Smoothie Carton	PMD
Can	PMD
Paper Cup	RESIDUAL
Used Paper Towel	RESIDUAL
Bread Bag	PMD
Used Paper Bag	RESIDUAL
Starbucks Plastic Cup	PMD
Peanuts Wrapper	RESIDUAL
Plastic Bottle	PMD
A4 Paper	PAPER
Cookie Carton	PAPER
Bread Wrap	RESIDUAL
Brochure	RESIDUAL
Plastic Cookie Container	PMD
Banana Peel	ORGANIC

Table 8 List of wastes given to participants

7.1.1.4 Survey questionnaire

The survey questionnaire is prepared for both groups. The following two questions are asked to both groups.

Q1: How difficult did you find separating the waste?

Q2: I'm confident that I separated my waste into the proper categories.

Then other questions are asked to only group 2, who did the task with the prototype present to evaluate the usability of the prototype. Please see Appendix C to find the full questions.

7.1.2 Procedure

First, the participants are invited to the testing space. Then the researchers handed out the information brochure and gave them general information about the study such as the research object and overall procedure. Then the participants are requested to sign the consent form. After that, the researchers inform about the task and the participants performed the task freely without any interruption from the researchers. Finally, the survey questionnaire is provided to the participants after they finished the task. The study took around 5 minutes for the first group and 10 minutes for the second group since the second group has more questions to answer in the survey.

7.2 Hypothesis

To evaluate dependent variables, the hypotheses are formulated for the t-test.

First, to compare the mean number of correct waste disposal, the following null hypothesis and the alternative hypothesis are formulated:

H_0*: There is* ***no significant difference*** *in the mean* ***number of correct waste disposal*** *between the condition prototype presented and did not present.*

H_A*: There is* ***a significant difference*** *in the mean* ***number of correct waste disposal*** *between the condition prototype presented and did not present.*

Second, for the waste separation difficulty comparison, the following null hypothesis and the alternative hypothesis are formulated:

H_0*: There is* ***no significant difference*** *in the mean score of* ***waste separation difficulty*** *between the condition prototype presented and did not present.*

H_A*: There is* ***a significant difference*** *in the mean score of* ***waste separation difficulty*** *between the condition prototype presented and did not present.*

Lastly, for the waste separation confidence comparison, the following null hypothesis and the alternative hypothesis are formulated:

H_0*: There is* ***no significant difference*** *in the mean score of* ***waste separation confidence*** *between the condition prototype presented and did not present.*

H_A*: There is* ***a significant difference*** *in the mean score of* ***waste separation confidence*** *between the condition prototype presented and did not present.*

7.3 Result

7.3.1 Impact of the prototype

7.3.1.1 Number of correct waste disposal

The number of correct waste disposal is the number of correctly disposed waste in the correspondent waste bin category out of the total 15 wastes. Table 9 and Table 10 show the result of the number of correct waste disposal from group 1 and group 2. Figure 65 shows the group statistic for both groups.

Participant	Correct	Wrong	Correct Rate (%)
1	11	4	73,33
2	10	5	66.67
3	8	7	53.33
4	10	5	66.67
5	9	6	60
6	9	6	60
7	10	5	66.67
8	11	4	73.33
9	12	3	80
10	13	2	86.67
11	10	5	66.67

Table 9 Result from the group 1 (Without the prototype)

Participant	Correct	Wrong	Correct Rate (%)
1	15	0	100
2	15	0	100
3	10	5	66.67
4	14	1	93.33
5	14	1	93.33
6	9	6	60
7	10	5	66.67
8	12	3	80
9	10	5	66.67
10	11	4	73.33
11	14	1	93.33

Table 10 Result from the group 2 (With the prototype)

Group Statistics

	SystemPresent	N	Mean	Std. Deviation	Std. Error Mean
Correct	No	11	10.2727	1.42063	.42834
	Yes	11	12.1818	2.27236	.68514

Figure 65 Group statistics of the two groups

The mean number of correct waste disposal is 10.2727 for group 1 and 12.1818 for group 2. Then based on this data, two samples independent t-test is conducted to compare the mean values from both groups (Figure 66). The average correct rate of the first group and the second group are 68.48% and 81.21%.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Correct	Equal variances assumed	6.785	.017	-2.363	20	.028	-1.90909	.80802	-3.59459	-.22360
	Equal variances not assumed			-2.363	16.781	.030	-1.90909	.80802	-3.61556	-.20263

Figure 66 Independent samples t-test result

The independent samples t-test requires the assumption of homogeneity of variance, this is checked by Levene's test. For this test, the following hypotheses are formulated:

H_0: The population variance of the first group and the second group are **equal**

H_A: The population variance of the first group and the second group are **not equal**

Since the significance is less than 0.05, reject the null hypothesis and accept the alternate hypothesis, concluding that variances are not equal.

Figure 66 shows the result of the t-test. The p-value of this t-test is 0.03, which is less than 0.05, so reject the null hypothesis and accept the alternative hypothesis. Therefore, this study found that the mean number of correct waste disposal is statistically significantly different between the two groups, the group who tested with the prototype present had a higher mean number of correct waste disposal than the group who tested without the prototype at the 5% significance level.

The correct rate of waste separation is increased from 68.48% to 81.21%, with a 12.73 percent point difference. This t-test shows that this difference is statistically significant.

7.3.1.2 Waste separation difficulty

Figure 67 and figure 68 show the frequency of responses from the participant regarding how they think about the difficulty of waste separation from both groups.

Figure 67 Frequency chart of the difficulty (Group 1)

Figure 68 Frequency chart of the difficulty (Group 2)

1	Extremely difficult
2	Somewhat difficult
3	Neither easy nor difficult
4	Some what easy
5	Extremely easy

Table 11 Score assigned to each response

Group Statistics

	SystemPresent	N	Mean	Std. Deviation	Std. Error Mean
Easy	No	11	3.2727	.90453	.27273
	Yes	11	3.1818	1.07872	.32525

Figure 69 Group statistics of two groups

The score is assigned to each response as in Table 11. Then the mean score of the difficulty is calculated. The mean score of the waste separation difficulty is 3.2727 for group 1 and 3.1818 for group 2. Then based on this data, two samples independent t-test is conducted to compare the mean score from both groups (Figure 69).

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Easy	Equal variances assumed	.554	.465	.214	20	.833	.09091	.42446	-.79450	.97631
	Equal variances not assumed			.214	19.410	.833	.09091	.42446	-.79622	.97804

Figure 70 Independent samples t-test result

From Levene's test equal variances are assumed since significance is larger than 0.05 thus it fails to reject the null hypothesis of Levene's test. Figure 70 shows the result of the t-test. The p-value of this t-test is 0.833, which is larger than 0.05, thus fails to reject the null hypothesis. Therefore, this study found that the mean score of waste separation difficulty is not statistically significantly different between the two groups at the 5% significance level.

7.3.1.3 Waste separation confidence

Figure 71 and figure 72 show the frequency of responses from the participant about how they are confident of waste separation from both groups.

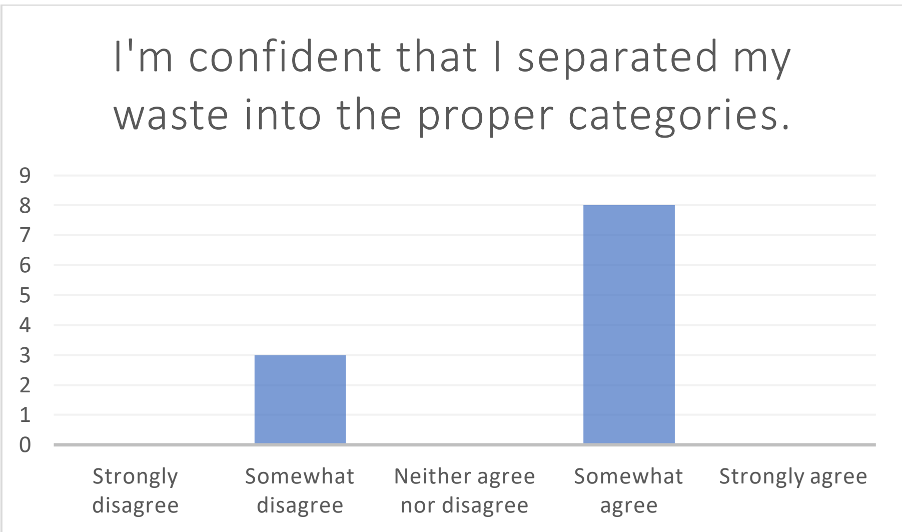

Figure 71 Frequency chart of the confidence (Group 1)

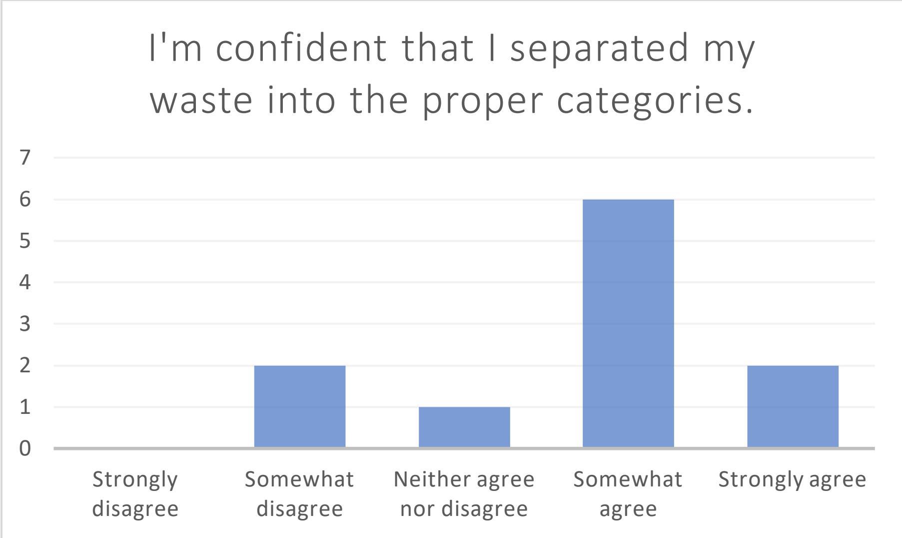

Figure 72 Frequency chart of the confidence (Group 2)

1	Strongly disagree
2	Somewhat disagree
3	Neither agree nor disagree
4	Somewhat agree
5	Strongly agree

Table 12 Score assigned to each response

Group Statistics

	SystemPresent	N	Mean	Std. Deviation	Std. Error Mean
Confident	No	11	3.4545	.93420	.28167
	Yes	11	3.7273	1.00905	.30424

Figure 73 Group statistics of the confidence

The score is assigned to each response as in Table 12. Then the mean score of the confidence is calculated. The mean score of the waste separation confidence is 3.4545 for group 1 and 3.7273 for group 2 (Figure 73). Then based on this data, two samples independent t-test is conducted to compare the mean score from both groups (Figure 74).

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Confident	Equal variances assumed	.021	.885	-.658	20	.518	-.27273	.41461	-1.13759	.59213
	Equal variances not assumed			-.658	19.882	.518	-.27273	.41461	-1.13792	.59246

Figure 74 Independent samples t-test result of confidence

From Levene's test equal variances are assumed since significance is larger than 0.05 thus it fails to reject the null hypothesis of Levene's test. The p-value of this t-test is 0.518, which is larger than 0.05, so fails to reject the null hypothesis. Therefore, this study found that the mean score of waste separation confidence is not statistically significantly different between the two groups at the 5% significance level.

7.3.2 Functionality

7.3.2.1 User Detection

Participant	Detection
1	O
2	O
3	O
4	O
5	O
6	X
7	X
8	O
9	X
10	O
11	O

Table 13 Detected and non-detected participants in group 2

Table 13 shows whether the prototype detected the participants when they first approached the front of the waste island. 8 out of 11 participants were detected, and 3 were not. Therefore the detection rate is 72.73%. These three participants who were not detected were standing on the left side of the waste island while doing their task. The participants did not approach the middle of the waste island always.

Figure 75 shows how the participants think about user detection. The average score of the user detection is 3.27 with the same scoring using Table 12. The participants neither agreed nor disagreed with the smooth user detection of the prototype.

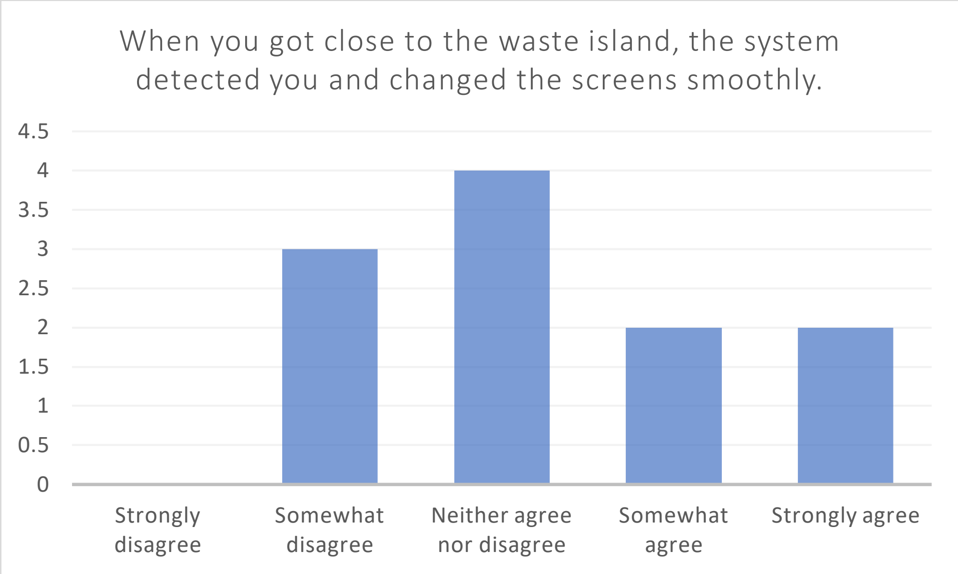

Figure 75 Frequency of responses about the user detection

7.3.2.2 Waste Detection

Participant	Success	Fail	Successful Detection Rate (%)
1	15	0	100
2	15	0	100
3	14	1	93.33
4	15	0	100
5	15	0	100
6	13	2	86.67
7	13	2	86.67
8	14	1	93.33
9	13	2	86.67
10	15	0	100
11	15	0	100

Table 14 Successful waste detection from group 2

Table 14 shows the successful waste detection from the group 2 participants. The average successful detection is 14.27, and the overall success rate is 95.15%. The prototype detected most of the waste except in some cases. When participants didn't use the screen and waste was disposed of in the waste bin, the prototype asked the participants whether it is the correct disposal. This screen remained for 5 seconds, and if participants disposed of another waste during these 5 seconds, the waste was not detected. The other case is when the participants put the waste into the bin too fast or they put the waste not in the side of the entrance of the waste bin, not in the middle. The last case is the noise from the sensor caused the malfunction of the sensor, detecting the waste even though the participant did not put anything inside the bin.

7.3.2.3 LED

Figure 76 shows how participants think about LED visibility. The average score of the visibility of the LED lights is 4, participants somewhat agreed with the clear visibility of the LED lights.

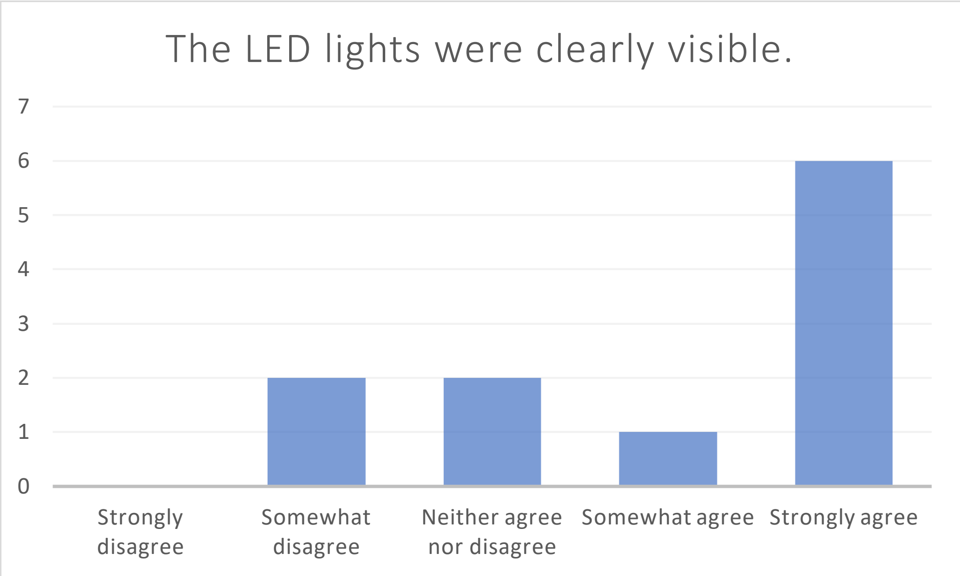

Figure 76 Frequency of responses about LED visibility

Figure 77 shows how the user thinks LED lights were helpful. The average score is 3.64, majority participants agreed with positive helpfulness of LED lights.

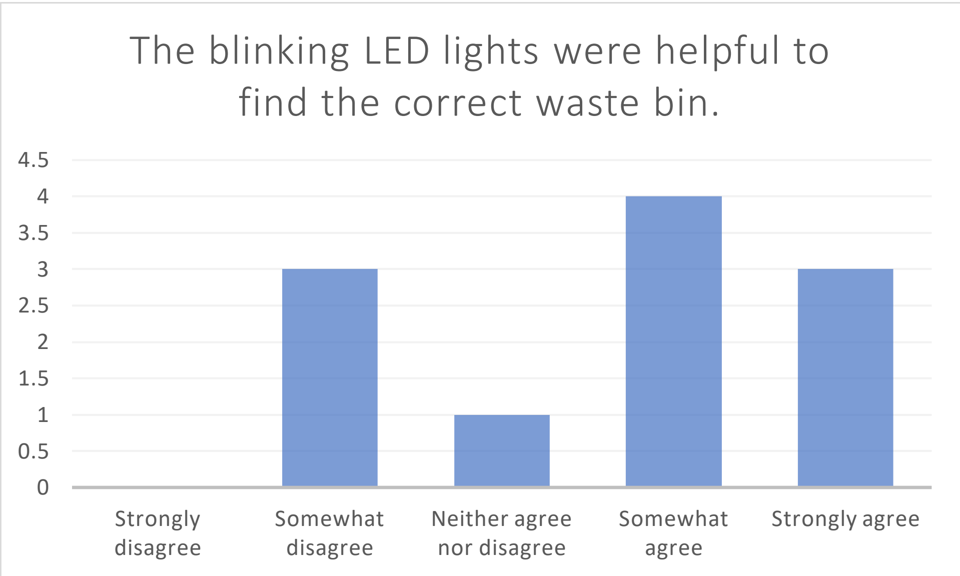

Figure 77 Frequency of responses about LED helpfulness

7.3.2.4 Behavior

Figure 78 shows how participants experience the system feedback. The average score is 3.82, meaning participants somewhat agreed with the system gives informative feedback.

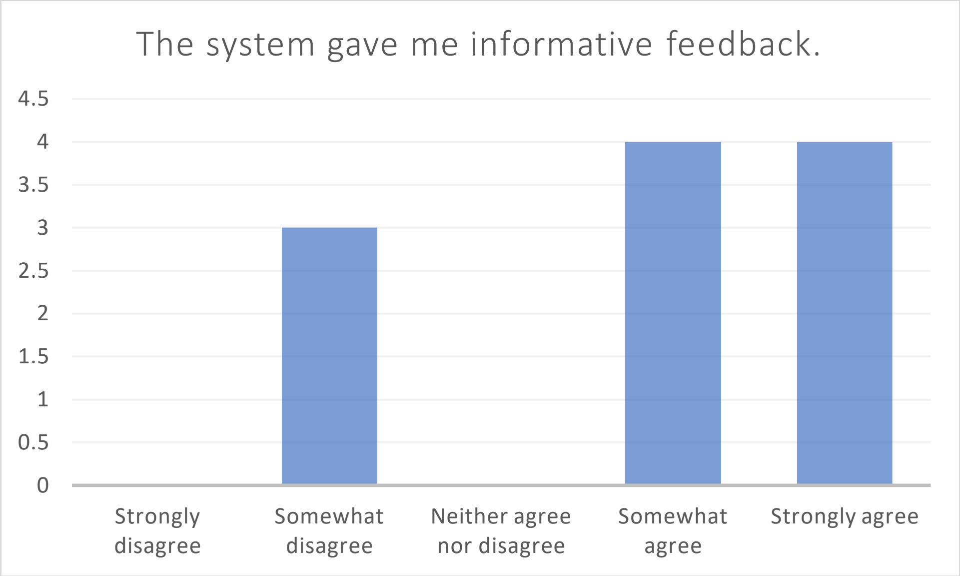

Figure 78 Frequency of responses about system feedback experience

Figure 79 presents how participants' behavior is influenced by the system. The average score is 4.18, meaning participants think this system changed their behavior positively.

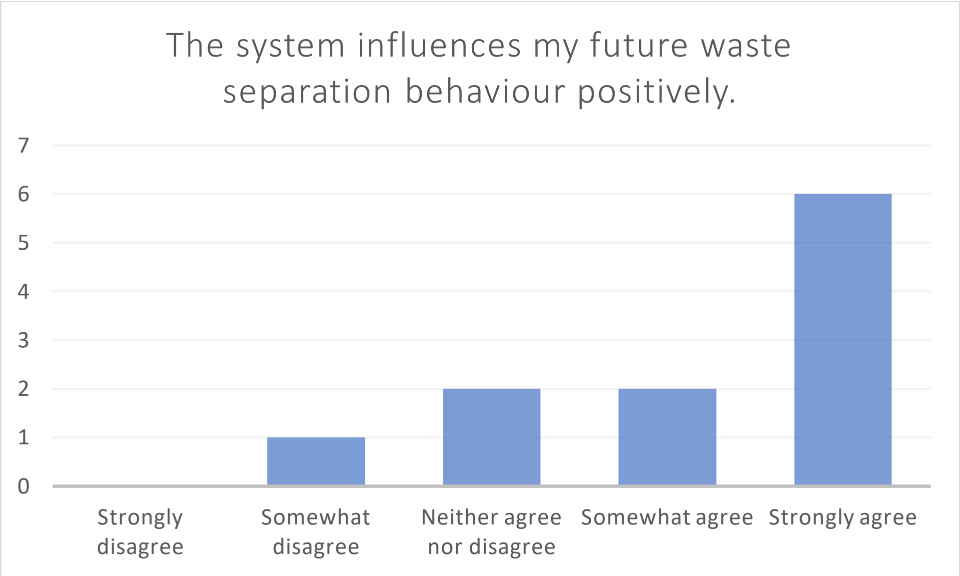

Figure 79 Frequency of responses about behavior influence

7.4 Conclusion

The evaluation results demonstrate that the prototype successfully improved the waste separation quality by increasing the number of correct waste disposal, which is a positive impact. However, the prototype couldn't make an improvement of the waste separation difficulty and confidence. The overall functionality of the prototype is also acceptable. The waste islands recognized the participants when they approach to dispose of the waste, and the LED light effectively indicates the right waste bin category. The prototype gave the participants informative feedback and positively influenced their future waste separation behavior.

8. Discussion

This chapter presents a discussion of the evaluation result of the prototype performance, limitations and recommendations.

8.1 The prototype performance

The client of this project, UT-CFM, expects an interactive system that can improve the waste separation quality at the UT campus. The results from the evaluation chapter show that the prototype enhanced the waste separation quality. The prototype educated and guided the users on how to separate wastes properly by using interactions before they dispose of waste. The knowledge from the education will remain in their mind since they think the prototype will positively influence their future waste separation behavior and when they are trying to dispose of waste later, there is a higher possibility that they will recall the experience with the interactive waste island and put the waste in the appropriate waste category. In this way, the waste separation quality can be further improved if the prototype can educate more people on campus and influence their behavior in a positive way.

8.2 Limitations & Recommendations

Even if all scores are higher than the middle score of 3, user detection has the lowest evaluation score. For user detection, the ultrasonic sensor attached to the center of the front wall of the waste island senses the proximity. Since this sensor is only placed in the center, if the user is approaching and standing side of the waste island, for example, left side or right side, the system could not detect the user even though they are standing less than 50cm from the waste island. This is the reason why 3 participants could not be detected by the sensor. To expand the detection range, two more ultrasonic distance sensors can be installed left and right sides in front of the waste island to expand the detection range of the prototype.

The waste detection rate itself is 95.15%, which is a high number, the prototype successfully detected 157 wastes out of a total of 165 wastes during the user evaluation. The infrared sensors that are used to detect the waste missed 8 wastes. The location of the sensor is considered one of the reasons. If the waste does not pass the center of the entrance of the waste bin, then the sensor cannot detect the waste because it's out of range. Also, this sensor cannot recognize waste that is smaller than 4.5cm since the waste should be bigger than the distance between the emitter and receiver of the infrared sensor. Moreover, this sensor couldn't detect the change in the distance if the waste passes the sensor too fast. These can be improved by installing more sensors to maximize the area that sensors cover and optimizing the angle of the sensor. Sometimes noise interrupts sensing. The current solution to reduce this noise is adopting the low pass filter in the software, the electric low pass filter with resistor and capacitor can have a better performance.

7 out of 11 participants think that LED lights are clearly visible, also 7 out of 11 participants think that the LED lights are helpful to find the appropriate waste bin which is shown on the selection screen. However, the LED light under the organic waste bin entrance is not visible clearly because of the cover on the organic waste bin. Moreover, the brightness of the LEDs is not at the maximum level since they are powered by 4.5V, instead of the maximum 5V. To enhance the LED light's visibility, the location of the LED can be adjusted. Now it's under the waste island cover but can be placed on the waste island cover if the LEDs don't block the entrance of the waste bins.

8.3 Future work

The client of this project, UT-CFM, is surprised by the prototype and they would like to conduct a further test with this prototype. They are satisfied with all the functionality of the prototype and result. However, this system is a prototype now, thus an additional supplement is required to make it a fully functional product without any errors. To do this, the hardware should be more stable. All the electronics are connected to the breadboard in the prototype, but a PCB board can be built to replace the breadboard, and also a stable wire connection is required. Arduino UNO can be replaced with Arduino Nano, which is a smaller type of the Arduino and the laptop can be replaced with a single-board computer such as Raspberry Pi. Using small electronics and hardware brings benefits that all these electronics and hardware can fit inside of the waste island, meaning that these can be protected by the waste island. It reduces the change of breakage from external interruptions.

9. Conclusion

This graduation project presents the development of an interactive waste island that can improve the waste separation quality at the UT campus. UT-CFM analyzed that the current waste separation quality at the UT campus is not optimal, because of the lack of knowledge, information and awareness of waste separation. The research goal is to develop a physical interactive installation that can influence the UT community members' waste separation behavior so that increase the waste separation quality at the UT campus.

To achieve the research object, an interactive waste island is developed. The waste island has an interactive screen that displays the statistics of the waste separation result from that waste island. The selection screen is activated and welcomes users by blinking lights when users are detected near the waste island. Users can choose their waste on the selection screen and the screen shows the correct category of the waste and the waste island indicates the appropriate waste bin by blinking lights. After the waste disposal, the waste island checks whether the waste is correctly or incorrectly disposed of and gives feedback based on the results.

The result shows that the prototype increased waste separation quality, gave users informative feedback, and influenced their attitude, however, the usability can be improved. The developed prototype, the interactive waste island, shows the potential to change the waste separation quality at the UT campus, and if this is further developed, this system can be implemented in public spaces at the UT. Back to the research question, the interactive waste island can educate the UT community members in a way that increases waste separation knowledge.

This interactive waste island can be the very first step to contributing to sustainability on the UT campus.

References

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, *50*(2), 179–211. https://doi.org/10.1016/0749-5978(91)90020-t

Árnadóttir, G., Kok, G., van Gils, S., & ten Hoor, G. (2018). Waste Separation in Cafeterias: A Study among University Students in the Netherlands. *International Journal of Environmental Research and Public Health*, *16*(1), 93. https://doi.org/10.3390/ijerph16010093

Bernstein, C. (2017, June 28). *brainstorming*. WhatIs.Com. https://www.techtarget.com/whatis/definition/brainstorming

Bin-e Smart Waste Bin. (n.d.). Bin-e Smart Waste Bin. Retrieved April 7, 2022, from https://www.bine.world

Blok, V., Wesselink, R., Studynka, O., & Kemp, R. (2015). Encouraging sustainability in the workplace: a survey on the pro-environmental behaviour of university employees. *Journal of Cleaner Production*, *106*, 55–67. https://doi.org/10.1016/j.jclepro.2014.07.063

Charopy. (n.d.). Charopy. Retrieved April 7, 2022, from https://www.charopy.com.au

Chen, F., Chen, H., Wu, M., Li, S., & Long, R. (2019). Research on the Driving Mechanism of Waste Separation Behavior: Based on Qualitative Analysis of Chinese Urban Residents. *International Journal of Environmental Research and Public Health*, *16*(10), 1859. https://doi.org/10.3390/ijerph16101859

Dai, X., Han, Y., Zhang, X., Hu, W., Huang, L., Duan, W., Li, S., Liu, X., & Wang, Q. (2017). Comparison between students and residents on determinants of willingness to separate waste and waste separation behaviour in Zhengzhou, China. *Waste Management & Research: The Journal for a Sustainable Circular Economy*, *35*(9), 949–957. https://doi.org/10.1177/0734242x17715096

Dragtstra, B., Marechal, B., Hilgeholt, C., Drewes, M., van der Wal, A., de Brouwer, A., van der Kuil, C., Kuiper, H., Boevink, M., Klijnstra, A., & Bijker, I. (2021). *2021 UT WASTE PLAN*. Universiteit Twente. https://www.utwente.nl/.uc/f8d3db556010288541d018563e40140aecb91b7bb8f0200/2021%20University%20of%20Twente%20Waste%20Plan.pdf

EUROSTAT. (2021). *Municipal waste statistics - Statistics Explained*. Retrieved April 14, 2022, from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Municipal_waste_statistics#Municipal_waste_generation

Ghazaryan, A. (2019, April 10). *User Stories and Scenarios in UX Design*. Designmodo. https://designmodo.com/user-stories-ux/

Greaves, M., Zibarras, L. D., & Stride, C. (2013). Using the theory of planned behavior to explore environmental behavioral intentions in the workplace. *Journal of Environmental Psychology*, *34*, 109–120. https://doi.org/10.1016/j.jenvp.2013.02.003

Grilli, G., & Curtis, J. (2021). Encouraging pro-environmental behaviours: A review of methods and approaches. *Renewable and Sustainable Energy Reviews*, *135*, 110039. https://doi.org/10.1016/j.rser.2020.110039

Hao, M., Zhang, D., & Morse, S. (2020). Waste Separation Behaviour of College Students under a Mandatory Policy in China: A Case Study of Zhengzhou City. *International*

Journal of Environmental Research and Public Health, *17*(21), 8190. https://doi.org/10.3390/ijerph17218190
Haughey, D. (2021, October 15). *MoSCoW Prioritisation Method*. Project Smart. https://www.projectsmart.co.uk/tools/moscow-method.php
Hayes, A. (2021, November 30). *What Is Functional Decomposition?* Investopedia. https://www.investopedia.com/terms/f/functional-decomposition.asp
Kurisu, K. (2016). *Pro-environmental Behaviors*. Springer Publishing.
Lange, F., Brückner, C., Kröger, B., Beller, J., & Eggert, F. (2014). Wasting ways: Perceived distance to the recycling facilities predicts pro-environmental behavior. *Resources, Conservation and Recycling*, *92*, 246–254. https://doi.org/10.1016/j.resconrec.2014.07.008
Lee, M., Choi, H., & Koo, Y. (2017). Inconvenience cost of waste disposal behavior in South Korea. *Ecological Economics*, *140*, 58–65. https://doi.org/10.1016/j.ecolecon.2017.04.031
Lucarelli, C., Mazzoli, C., & Severini, S. (2020). Applying the Theory of Planned Behavior to Examine Pro-Environmental Behavior: The Moderating Effect of COVID-19 Beliefs. *Sustainability*, *12*(24), 10556. https://doi.org/10.3390/su122410556
Mader, A., & Eggink, W. (2014). A Design Process for Creative Technology. *16th International Conference on Engineering and Product Design, E&PDE 2014*, 568–573. https://www.researchgate.net/publication/265755092_A_DESIGN_PROCESS_FOR_CREATIVE_TECHNOLOGY
Malakahmad. (2010). Solid Waste Characterization and Recycling Potential for University Technology PETRONAS Academic Buildings. *American Journal of Environmental Sciences*, *6*(5), 422–427. https://doi.org/10.3844/ajessp.2010.422.427
Mao, J. Y., Vredenburg, K., Smith, P. W., & Carey, T. (2005). The state of user-centered design practice. *Communications of the ACM*, *48*(3), 105–109. https://doi.org/10.1145/1047671.1047677
Olander, S., & Landin, A. (2005). Evaluation of stakeholder influence in the implementation of construction projects. *International Journal of Project Management*, *23*(4), 321–328. https://doi.org/10.1016/j.ijproman.2005.02.002
Oysterable. (n.d.). Oysterable. Retrieved April 7, 2022, from https://www.oysterable.com/5
Scrapp Recycling. (n.d.). Scrapp Recycling. Retrieved April 7, 2022, from https://www.scrapprecycling.com
Sheau-Ting, L., Sin-Yee, T., & Weng-Wai, C. (2016). Preferred Attributes of Waste Separation Behaviour: An Empirical Study. *Procedia Engineering*, *145*, 738–745. https://doi.org/10.1016/j.proeng.2016.04.094
Steg, L., & Vlek, C. (2009). Encouraging pro-environmental behaviour: An integrative review and research agenda. *Journal of Environmental Psychology*, *29*(3), 309–317. https://doi.org/10.1016/j.jenvp.2008.10.004
Stern, P. C. (2000). New Environmental Theories: Toward a Coherent Theory of Environmentally Significant Behavior. *Journal of Social Issues*, *56*(3), 407–424. https://doi.org/10.1111/0022-4537.00175

Supakata, N. (2018). Bin monsters for promoting waste separation behavior. *Applied Environmental Education & Communication*, *17*(4), 310–322. https://doi.org/10.1080/1533015x.2017.1415774

SuperBin Product. (n.d.). SuperBin. Retrieved April 7, 2022, from https://www.superbin.co.kr/en/contents/product.php

Torres-Pereda, P., Parra-Tapia, E., Rodríguez, M., Félix-Arellano, E., & Riojas-Rodríguez, H. (2020). Impact of an intervention for reducing waste through educational strategy: A Mexican case study, what works, and why? *Waste Management*, *114*, 183–195. https://doi.org/10.1016/j.wasman.2020.06.027

Wallen, K. E., & Daut, E. (2018). The challenge and opportunity of behaviour change methods and frameworks to reduce demand for illegal wildlife. *Nature Conservation*, *26*, 55–75. https://doi.org/10.3897/natureconservation.26.22725

Yuan, Y., Nomura, H., Takahashi, Y., & Yabe, M. (2016). Model of Chinese Household Kitchen Waste Separation Behavior: A Case Study in Beijing City. *Sustainability*, *8*(10), 1083. https://doi.org/10.3390/su8101083

Zhang, H., Liu, J., Wen, Z. G., & Chen, Y. X. (2017). College students' municipal solid waste source separation behavior and its influential factors: A case study in Beijing, China. *Journal of Cleaner Production*, *164*, 444–454. https://doi.org/10.1016/j.jclepro.2017.06.224

Appendix A. Arduino code

```
#include <FastLED.h>

unsigned long last_time = 0;
unsigned long last_timeLED = 0;
unsigned long ltime = 0;
boolean on = false;

boolean LED_on = false;
boolean active = false;
boolean ledActive = true;

int activeCounter = 0;
int ledCounter = 1;
int run_limited = 0;

boolean ledResidual = false;

#define TRIG 9 //TRIG Pin
#define ECHO 8 //ECHO Pin

#define NUM_LEDS 29

#define NUM_STRIPS 2
#define NUM_LEDS_PER_STRIP 29
CRGB leds[NUM_STRIPS][NUM_LEDS_PER_STRIP];

CRGBArray<NUM_LEDS> leds1;
CRGBArray<NUM_LEDS> leds2;
CRGBArray<NUM_LEDS> leds3;
CRGBArray<NUM_LEDS> leds4;

String ledData;
#define UPDATES_PER_SECOND 100


#define BRIGHTNESS  100
#define LED_TYPE    WS2811
#define COLOR_ORDER GRB


int getNum = -1;

float volts1;
float volts2;
float volts3;
float volts4;

float filteredVolts1;
float filteredVolts2;
float filteredVolts3;
float filteredVolts4;

float sensitivity = 0.15;

boolean S1 = false;
boolean S2 = false;

void setup() {
  Serial.begin(9600); // start the serial port
  volts1 = analogRead(A0);
  volts2 = analogRead(A1);
  filteredVolts1 = volts1;
  filteredVolts2 = volts2;

  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

```

```
  // Residual LED
  FastLED.addLeds<NEOPIXEL, 10>(leds1, NUM_LEDS_PER_STRIP);
  // PMD LED
  FastLED.addLeds<NEOPIXEL, 11>(leds2, NUM_LEDS_PER_STRIP);
  // Organic LED
  FastLED.addLeds<NEOPIXEL, 12>(leds3, NUM_LEDS_PER_STRIP);
  // Paper LED
  FastLED.addLeds<NEOPIXEL, 7>(leds4, NUM_LEDS_PER_STRIP);
  FastLED.setBrightness( 255 );
}
void loop() {
  if (activeCounter < 1) {
    ledSet();
    checkDistance();
  }
  else if (activeCounter >= 1) {
    sensing();
    getCode();
    if (ledActive == true) {
      if (ledData == "R") {
        ledControl(1);
        //ledResidual = true;
      }
      else if (ledData == "P") {
        ledControl(2);
      }
      else if (ledData == "O") {
        ledControl(3);
      }
      else if (ledData == "A") {
        ledControl(4);
      }
      else if (ledData == "N") {
        ledOff();
      }
      else if (ledData == "S" and run_limited < 10) {
        ledControl(5);
      }
      else if (ledData == "S" and run_limited >= 10) {
        ledOff();
      }
    }
    else {
      ledOff();
    }
    if (ledData == "0") {
      reset_system();
    }
  }
}
void reset_system() {
  last_time = 0;
  last_timeLED = 0;
```

```
  ltime = 0;
  on = false;
  LED_on = false;
  active = false;
  ledActive = true;
  activeCounter = 0;
  ledCounter = 1;
  run_limited = 0;
  ledData = "None";
}

void ledSet() {

  for (int i = 0; i < NUM_LEDS; i++) {
    // let's set an led value
    leds1[i] = CRGB::Black;
    leds2[i] = CRGB::Black;
    FastLED.show();
  }
}

void ledOn(int cat) {
  for (int i = 0; i < NUM_LEDS; i++) {
    // fade everything out
    switch (cat) {
      // Residual LED
      case 1:
        leds1.fadeToBlackBy(40);
        leds1[i] = CRGB::White;
        leds2[i] = CRGB::Black;
        leds3[i] = CRGB::Black;
        leds4[i] = CRGB::Black;
        break;
      // PMD LED
      case 2:
        leds2.fadeToBlackBy(40);
        leds1[i] = CRGB::Black;
        leds2[i] = CRGB::Red;
        leds3[i] = CRGB::Black;
        leds4[i] = CRGB::Black;
        break;

      // Organic LED
      case 3:
        leds3.fadeToBlackBy(40);
        leds1[i] = CRGB::Black;
        leds2[i] = CRGB::Black;
        leds3[i] = CRGB::Green;
        leds4[i] = CRGB::Black;
        break;

      // Paper LED
      case 4:
        leds3.fadeToBlackBy(40);
        leds1[i] = CRGB::Black;
        leds2[i] = CRGB::Black;
        leds3[i] = CRGB::Black;
        leds4[i] = CRGB::Blue;
        break;

      // Welcome LED
      case 5:
        leds3.fadeToBlackBy(40);
        leds1[i] = CRGB::White;
        leds2[i] = CRGB::Red;
        leds3[i] = CRGB::Green;
        leds4[i] = CRGB::Blue;
        break;

    }
    FastLED.show();
```

```
  }
}

void ledOff() {
  for (int i = 0; i < NUM_LEDS; i++) {
    // fade everything out

    leds1.fadeToBlackBy(40);
    leds2.fadeToBlackBy(40);
    leds3.fadeToBlackBy(40);
    leds4.fadeToBlackBy(40);

    leds1[i] = CRGB::Black;
    leds2[i] = CRGB::Black;
    leds3[i] = CRGB::Black;
    leds4[i] = CRGB::Black;

  }
  FastLED.show();

}

void ledControl(int cat) {
  static uint8_t hue;
  unsigned long nowLED = millis();

  if (ledCounter >= 0 && nowLED - last_timeLED > 100) {
    ledCounter -= 1;
    run_limited += 1;
    if (ledCounter < 0) {
      ledCounter = 1;

      if (LED_on == false) {
        ledOff();
        LED_on = true;
      }
      else {
        ledOn(cat);
        LED_on = false;
      }
    }
    last_timeLED = nowLED;
  }
}

void getCode() {
  if (Serial.available() > 0) {

    char data = Serial.read();
    char str[2];
    str[0] = data;
    str[1] = '\0';
    ledData = str;
  }
}

void checkDistance() {
  if (getDistance() <= 50) {
    Serial.println("0");
    activeCounter += 1;
  }
}

float getDistance() {
  float duration, distance;
  unsigned long now = millis();


  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);

  duration = pulseIn (ECHO, HIGH);

  distance = duration * 17 / 1000;


  return distance;
}


void sensing() {
  // 5v
  volts1 = analogRead(A0);
  volts2 = analogRead(A1);
  volts3 = analogRead(A2);
  volts4 = analogRead(A3);

  // Low Pass Filter
  filteredVolts1 = filteredVolts1 * (1 - sensitivity) + volts1 * sensitivity;
  filteredVolts2 = filteredVolts2 * (1 - sensitivity) + volts2 * sensitivity;
  filteredVolts3 = filteredVolts3 * (1 - sensitivity) + volts3 * sensitivity;
  filteredVolts4 = filteredVolts4 * (1 - sensitivity) + volts4 * sensitivity;


  float dist1F = (6762 / (filteredVolts1 - 9)) - 4;

  float dist2F = (6762 / (filteredVolts2 - 9)) - 4;

  float dist3F = (6762 / (filteredVolts3 - 9)) - 4;

  float dist4F = (6762 / (filteredVolts4 - 9)) - 4;

  // Residual Sensor
  if (dist1F < 14.5) {
    Serial.println("1");
    ledActive = false;
    delay(300);
  }

  // PMD Sensor
  else if (dist2F < 14.5) {
    Serial.println("2");
    ledActive = false;
    delay(300);
  }

  // Organic Sensor
  else if (dist3F < 17.5) {
    Serial.println("3");
    ledActive = false;
    delay(300);
  }

  // Paper Sensor
  else if (dist4F < 15.5) {
    Serial.println("4");
    ledActive = false;
    delay(300);
  }

}
```

Appendix B. Python Code

main.py

```
import pygame

from control import Control
pygame.init()



if __name__ == "__main__":
    c = Control()

    while c.running:
        c.loop()

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                c.running = False
```

sensor.py

```
import serial

class Sensor:
    def __init__(self):
        self.data = -1
        self.arduino = serial.Serial('/dev/cu.usbmodem1444201', 9600,
timeout=.1)

    def serialOutput(self, text):
        text = text.encode()
        self.arduino.write(text)

    def serialInput(self):
        if self.arduino.inWaiting() > 0:
            data = self.arduino.readline() #the last bit gets rid of the
new-line chars
            if data:
                data = data.decode()
                data = data.strip()

                self.data = int(data)
                print(self.data)
```

control.py

```
from screen import Screen
from sensor import Sensor

class Control:
    def __init__(self):
        self.running = True

        self.run_once = 0

        self.screen = Screen()
        self.sensor = Sensor()
        self.active = False
```

```
        self.detected = None
        self.counter = 0
        self.lastSelected = None
        self.ledCounter = 0

        self.oldData = -1
        self.run_once2 = 0
        self.run_once3 = 0

        self.ledResidual = False

    def getCategory(self):
        if self.oldData != self.sensor.data:
            self.run_once2 = 0
        if self.run_once2 == 0:
            if self.sensor.data == 1:
                self.detected = "residual"
                print("RES Disposed")
            elif self.sensor.data == 2:
                self.detected = "pmd"
                print("PMD Disposed")
            elif self.sensor.data == 3:
                self.detected = "organic"
                print("Organic Disposed")
            elif self.sensor.data == 4:
                self.detected = "paper"
                print("Paper Disposed")
            self.oldData = self.sensor.data
            self.run_once2 = 1


    def reset(self):
        self.active = False

        self.detected = None
        self.screen.counter = 0
        self.screen.systemCounter = 0

        self.screen.resetMain = False
        self.screen.disposed = False

        self.screen.selectedIcon = ""
        self.screen.menu = "main"
        self.sensor.serialOutput("0")

        self.run_once = 0
        self.run_once2 = 0
        self.run_once3 = 0

        self.screen.posControl = True
        self.screen.posControlButton = True
        self.screen.run_once = 0
        self.screen.run_once_button = 0
        self.screen.run_once_counter = True

        self.screen.isSelected = False
        self.screen.isSubSelected = False
        self.screen.screenUsed = False

        self.screen.selectedCategory = None
```

```
        self.screen.lastPos = ()
        self.screen.lastPosMain = ()

        self.oldData = -1
        self.sensor.data = -1

        if self.screen.button is None:
            self.screen.subtract += 1

        self.screen.button = None


    def ledReset(self):
        if self.screen.resetMain:
            self.screen.resetMain = False
            self.sensor.serialOutput("N")
            print("LED N")
            self.lastSelected = None

    def ledControl(self):
        if self.screen.isSubSelected:
            if self.lastSelected != self.screen.selectedCategory:
                self.run_once = 0
            if self.run_once == 0:
                if self.screen.selectedCategory == "residual":
                    self.sensor.serialOutput("R")
                    self.lastSelected = self.screen.selectedCategory
                    print("LED R")

                if self.screen.selectedCategory == "pmd":
                    self.sensor.serialOutput("P")
                    self.lastSelected = self.screen.selectedCategory
                    print("LED P")

                if self.screen.selectedCategory == "organic":
                    self.sensor.serialOutput("O")
                    self.lastSelected = self.screen.selectedCategory
                    print("LED O")

                if self.screen.selectedCategory == "paper":
                    self.sensor.serialOutput("A")
                    self.lastSelected = self.screen.selectedCategory
                    print("LED A")

                self.run_once = 1

    def loop(self):
        self.screen.loop(self.active, self.detected)
        self.sensor.serialInput()
        if not self.active and self.sensor.data == 0:
            self.active = True

        if self.active:
            if self.run_once3 == 0:
                self.sensor.serialOutput("S")
                print("S")
                self.run_once3 = 1

            if not self.screen.disposed:
                self.getCategory()
            self.ledReset()
```

```
            self.ledControl()

            if self.screen.counter > 3:
                self.reset()
            elif self.screen.systemCounter >= 30 and not
self.screen.screenUsed:
                self.reset()
```

screen.py

```
import pygame
import math

class Screen:

    def __init__(self):
        self.running = True

        self.subtract = 0

        self.correctCounter = 0
        self.incorrectCounter = 0
        self.notsureCounter = 0

        self.paperCounter = 0
        self.organicCounter = 0
        self.pmdCounter = 0
        self.residualCounter = 0

        self.paperPercent = 0
        self.organicPercent = 0
        self.pmdPercent = 0
        self.residualPercent = 0
        self.correctPercent = 0
        self.incorrectPercent = 0
        self.notsurePercent = 0

        self.width = 1920
        self.height = 1080
        self.screen = pygame.display.set_mode((self.width, self.height),
pygame.FULLSCREEN)
        self.background =
pygame.image.load("images/selection/mainselection.bmp")
        self.subBackground =
pygame.image.load("images/selection/wastetype.bmp")
        self.background_main =
pygame.image.load("images/selection/informationscreen.bmp")

        pygame.display.set_caption("TEST")

        self.isSelected = False
        self.isSubSelected = False
        self.screenUsed = False
        self.disposed = False

        self.run_once = 0
        self.run_once_button = 0
        self.run_once_counter = True

        self.posControl = True
        self.posControlButton = True
        self.lastPos = ()
```

```
        self.lastPosMain = ()

        self.button = None

        self.selectedCategory = None

        self.resetMain = False

        self.counter = 0
        self.systemCounter = 0

        self.last_time = pygame.time.get_ticks()
        self.last_systemTime = pygame.time.get_ticks()

        self.font = pygame.font.Font(None, 20)
        self.selectedIcon = ""
        self.menu = "main"

        ### Senna's Program starts here ###
        # dictionary variable that contains all the information about all
the icons
        self.iconControl: dict = {
            # The main selection screen: all the icons include what sub-
selection screen they link to
            "main": {
                "Cups": {
                    "imgXPos": 150,
                    "imgYPos": 200,
                    "imageID": "images/Maingroups/cups.bmp",
                    "linksTo": "CUPS"
                },
                "CartonPaper": {
                    "imgXPos": 450,
                    "imgYPos": 450,
                    "imageID": "images/Maingroups/cartonpaper.bmp",
                    "linksTo": "CARTONPAPER"
                },
                "Metals": {
                    "imgXPos": 800,
                    "imgYPos": 700,
                    "imageID": "images/Maingroups/metals.bmp",
                    "linksTo": "METALS"
                },
                "Organic": {
                    "imgXPos": 150,
                    "imgYPos": 700,
                    "imageID": "images/Maingroups/organic.bmp",
                    "linksTo": "ORGANIC"
                },
                "Others": {
                    "imgXPos": 1150,
                    "imgYPos": 450,
                    "imageID": "images/Maingroups/others.bmp",
                    "linksTo": "OTHERS"
                },
                "paperNapkin": {
                    "imgXPos": 1450,
                    "imgYPos": 200,
                    "imageID": "images/Maingroups/papernapkin.bmp",
                    "linksTo": "PAPERNAPKIN"
                },
```

```
                "Plastic": {
                    "imgXPos": 1450,
                    "imgYPos": 700,
                    "imageID": "images/Maingroups/plastic.bmp",
                    "linksTo": "PLASTIC"
                }
            },

            # The sub-selection screens: all the icons include the sort
waste they are
            # cups
            "CUPS": {
                "foodCup": {
                    "imgXPos": 75,
                    "imgYPos": 200,
                    "imageID": "images/cups/foodcup.bmp",
                    "sortWaste": "residual"
                },
                "noodlesCups": {
                    "imgXPos": 425,
                    "imgYPos": 200,
                    "imageID": "images/cups/noodlescup.bmp",
                    "sortWaste": "residual"
                },
                "paperCup1": {
                    "imgXPos": 775,
                    "imgYPos": 200,
                    "imageID": "images/cups/papercup1.bmp",
                    "sortWaste": "residual"
                },
                "paperCup2": {
                    "imgXPos": 1125,
                    "imgYPos": 200,
                    "imageID": "images/cups/papercup2.bmp",
                    "sortWaste": "residual"
                },
                "paperCup3": {
                    "imgXPos": 1475,
                    "imgYPos": 200,
                    "imageID": "images/cups/papercup3.bmp",
                    "sortWaste": "residual"
                },
                "pastaCup": {
                    "imgXPos": 75,
                    "imgYPos": 600,
                    "imageID": "images/cups/pastacup.bmp",
                    "sortWaste": "residual"
                },
                "plasticCup": {
                    "imgXPos": 425,
                    "imgYPos": 600,
                    "imageID": "images/cups/plasticcup.bmp",
                    "sortWaste": "pmd"
                },
                "starbuckPaperCup": {
                    "imgXPos": 775,
                    "imgYPos": 600,
                    "imageID": "images/cups/starbuckspaper.bmp",
                    "sortWaste": "residual"
                },
                "coffeecups": {
```

```
            "imgXPos": 1125,
            "imgYPos": 600,
            "imageID": "images/cups/coffeecups.bmp",
            "sortWaste": "residual"
        },
        "starbucksPlasticCup": {
            "imgXPos": 1475,
            "imgYPos": 600,
            "imageID": "images/cups/starbucksplastic.bmp",
            "sortWaste": "pmd"
        },
        "PreviousButton": {
            "imgXPos": 775,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "main"
        }
    },

    # carton and paper
    "CARTONPAPER": {
        "cartonBasket": {
            "imgXPos": 75,
            "imgYPos": 200,
            "imageID": "images/cartonpaper/cartonbasket.bmp",
            "sortWaste": "residual"
        },
        "cartonBox": {
            "imgXPos": 425,
            "imgYPos": 200,
            "imageID": "images/cartonpaper/cartonbox.bmp",
            "sortWaste": "paper"
        },
        "cartonHolder": {
            "imgXPos": 775,
            "imgYPos": 200,
            "imageID": "images/cartonpaper/cartonholder.bmp",
            "sortWaste": "paper"
        },
        "coasters": {
            "imgXPos": 1125,
            "imgYPos": 200,
            "imageID": "images/cartonpaper/coasters.bmp",
            "sortWaste": "residual"
        },
        "drinkCarton": {
            "imgXPos": 1475,
            "imgYPos": 200,
            "imageID": "images/cartonpaper/drinkcarton.bmp",
            "sortWaste": "pmd"
        },
        "eggCarton": {
            "imgXPos": 75,
            "imgYPos": 600,
            "imageID": "images/cartonpaper/eggcarton.bmp",
            "sortWaste": "paper"
        },
        "paperBags": {
            "imgXPos": 425,
            "imgYPos": 600,
            "imageID": "images/cartonpaper/paperbags.bmp",
```

```
            "sortWaste": "residual"
        },
        "paperBreadBags": {
            "imgXPos": 775,
            "imgYPos": 600,
            "imageID": "images/cartonpaper/paperbreadbag.bmp",
            "sortWaste": "residual"
        },
        "pizzaBox": {
            "imgXPos": 1125,
            "imgYPos": 600,
            "imageID": "images/cartonpaper/pizzabox.bmp",
            "sortWaste": "residual"
        },
        "teaBagPaper": {
            "imgXPos": 1475,
            "imgYPos": 600,
            "imageID": "images/cartonpaper/teabagpaper.bmp",
            "sortWaste": "residual"
        },
        "PreviousButton": {
            "imgXPos": 775,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "main"
        }
    },

    # metals
    "METALS": {
        "bottleCaps": {
            "imgXPos": 250,
            "imgYPos": 200,
            "imageID": "images/metals/bottlecaps.bmp",
            "sortWaste": "pmd"
        },
        "energyDrink": {
            "imgXPos": 600,
            "imgYPos": 200,
            "imageID": "images/metals/energydrink.bmp",
            "sortWaste": "pmd"
        },
        "icedCoffeeCan": {
            "imgXPos": 950,
            "imgYPos": 200,
            "imageID": "images/metals/icedcoffeecan.bmp",
            "sortWaste": "pmd"
        },
        "metalBox": {
            "imgXPos": 1300,
            "imgYPos": 200,
            "imageID": "images/metals/metalbox.bmp",
            "sortWaste": "pmd"
        },
        "paperclip": {
            "imgXPos": 250,
            "imgYPos": 600,
            "imageID": "images/metals/paperclip.bmp",
            "sortWaste": "residual"
        },
        "sodaCan": {
```

```
            "imgXPos": 600,
            "imgYPos": 600,
            "imageID": "images/metals/sodacan.bmp",
            "sortWaste": "pmd"
        },
        "tunaCan": {
            "imgXPos": 950,
            "imgYPos": 600,
            "imageID": "images/metals/tunacan.bmp",
            "sortWaste": "pmd"
        },
        "PreviousButton": {
            "imgXPos": 775,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "main"
        }
    },

    # organics
    "ORGANIC": {
        "appleCore": {
            "imgXPos": 425,
            "imgYPos": 200,
            "imageID": "images/organic/applecore.bmp",
            "sortWaste": "organic"
        },
        "bananaPeel": {
            "imgXPos": 775,
            "imgYPos": 200,
            "imageID": "images/organic/bananapeel.bmp",
            "sortWaste": "organic"
        },
        "pizza": {
            "imgXPos": 1125,
            "imgYPos": 200,
            "imageID": "images/organic/pizza.bmp",
            "sortWaste": "organic"
        },
        "teaBag": {
            "imgXPos": 425,
            "imgYPos": 600,
            "imageID": "images/organic/teabag.bmp",
            "sortWaste": "organic"
        },
        "tosti": {
            "imgXPos": 775,
            "imgYPos": 600,
            "imageID": "images/organic/tosti.bmp",
            "sortWaste": "organic"
        },
        "PreviousButton": {
            "imgXPos": 275,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "main"
        }
    },

    # others
    "OTHERS": {
```

```
"bandAids": {
    "imgXPos": 75,
    "imgYPos": 200,
    "imageID": "images/others/bandaids.bmp",
    "sortWaste": "residual"
},
"barrierTape": {
    "imgXPos": 425,
    "imgYPos": 200,
    "imageID": "images/others/barriertape.bmp",
    "sortWaste": "residual"
},
"bikeLights": {
    "imgXPos": 775,
    "imgYPos": 200,
    "imageID": "images/others/bikelights.bmp",
    "sortWaste": "residual"
},
"chewingGum": {
    "imgXPos": 1125,
    "imgYPos": 200,
    "imageID": "images/others/chewinggum.bmp",
    "sortWaste": "residual"
},
"ductTape": {
    "imgXPos": 1475,
    "imgYPos": 200,
    "imageID": "images/others/ducttape.bmp",
    "sortWaste": "residual"
},
"earPlugs": {
    "imgXPos": 75,
    "imgYPos": 600,
    "imageID": "images/others/earplugs.bmp",
    "sortWaste": "residual"
},
"flyers": {
    "imgXPos": 425,
    "imgYPos": 600,
    "imageID": "images/others/flyers.bmp",
    "sortWaste": "residual"
},
"lighter": {
    "imgXPos": 775,
    "imgYPos": 600,
    "imageID": "images/others/ligher.bmp",
    "sortWaste": "residual"
},
"metalPacket": {
    "imgXPos": 1125,
    "imgYPos": 600,
    "imageID": "images/others/metalpacket.bmp",
    "sortWaste": "pmd"
},
"mouthMasks": {
    "imgXPos": 1475,
    "imgYPos": 600,
    "imageID": "images/others/mouthmasks.bmp",
    "sortWaste": "residual"
},
"NextButton": {
```

```
            "imgXPos": 1275,
            "imgYPos": 830,
            "imageID": "images/selection/nextbutton.bmp",
            "linksTo": "NextOthers"
        },
        "PreviousButton": {
            "imgXPos": 275,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "main"
        }
    },

    # next others
    "NextOthers": {
        "mustardPacket": {
            "imgXPos": 75,
            "imgYPos": 200,
            "imageID": "images/others/mustardpacket.bmp",
            "sortWaste": "residual"
        },
        "pens": {
            "imgXPos": 425,
            "imgYPos": 200,
            "imageID": "images/others/pens.bmp",
            "sortWaste": "residual"
        },
        "plasticCaps": {
            "imgXPos": 775,
            "imgYPos": 200,
            "imageID": "images/others/plasticcaps.bmp",
            "sortWaste": "pmd"
        },
        "strawpackaging": {
            "imgXPos": 1125,
            "imgYPos": 200,
            "imageID": "images/others/strawpackaging.bmp",
            "sortWaste": "residual"
        },
        "sunglasses": {
            "imgXPos": 1475,
            "imgYPos": 200,
            "imageID": "images/others/sunglasses.bmp",
            "sortWaste": "residual"
        },
        "tape": {
            "imgXPos": 250,
            "imgYPos": 600,
            "imageID": "images/others/tape.bmp",
            "sortWaste": "residual"
        },
        "woodenCutlery": {
            "imgXPos": 600,
            "imgYPos": 600,
            "imageID": "images/others/woodencutlery.bmp",
            "sortWaste": "residual"
        },
        "woodenMixers": {
            "imgXPos": 950,
            "imgYPos": 600,
            "imageID": "images/others/woodenmixsticks.bmp",
```

```
            "sortWaste": "residual"
        },
        "PreviousButton": {
            "imgXPos": 775,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "OTHERS"
        }
    },

    # paper and napkins
    "PAPERNAPKIN": {
        "cleanPaper": {
            "imgXPos": 250,
            "imgYPos": 200,
            "imageID": "images/papernapkin/cleanpaper.bmp",
            "sortWaste": "paper"
        },
        "coffeeFilter": {
            "imgXPos": 600,
            "imgYPos": 200,
            "imageID": "images/papernapkin/coffeefilter.bmp",
            "sortWaste": "residual"
        },
        "dirtyNapkin": {
            "imgXPos": 950,
            "imgYPos": 200,
            "imageID": "images/papernapkin/dirtynapkin.bmp",
            "sortWaste": "residual"
        },
        "handkerchiefs": {
            "imgXPos": 1300,
            "imgYPos": 200,
            "imageID": "images/papernapkin/handkerchiefs.bmp",
            "sortWaste": "residual"
        },
        "napkins": {
            "imgXPos": 250,
            "imgYPos": 600,
            "imageID": "images/papernapkin/napkins.bmp",
            "sortWaste": "residual"
        },
        "newsPaper": {
            "imgXPos": 600,
            "imgYPos": 600,
            "imageID": "images/papernapkin/newspaper.bmp",
            "sortWaste": "paper"
        },
        "noteBooks": {
            "imgXPos": 950,
            "imgYPos": 600,
            "imageID": "images/papernapkin/notebooks.bmp",
            "sortWaste": "residual"
        },
        "paperStraws": {
            "imgXPos": 1300,
            "imgYPos": 200,
            "imageID": "images/papernapkin/paperstraws.bmp",
            "sortWaste": "residual"
        },
        "PreviousButton": {
```

```
            "imgXPos": 775,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "main"
        }
    },

    # plastics
    "PLASTIC": {
        "candyWrappers": {
            "imgXPos": 75,
            "imgYPos": 200,
            "imageID": "images/plastic/candywrappers.bmp",
            "sortWaste": "residual"
        },
        "chipsPackaging": {
            "imgXPos": 425,
            "imgYPos": 200,
            "imageID": "images/plastic/chipspackaging.bmp",
            "sortWaste": "residual"
        },
        "cookieWrapper": {
            "imgXPos": 775,
            "imgYPos": 200,
            "imageID": "images/plastic/cookiewrapper.bmp",
            "sortWaste": "residual"
        },
        "crackerPacket": {
            "imgXPos": 1125,
            "imgYPos": 200,
            "imageID": "images/plastic/crackerpackaging.bmp",
            "sortWaste": "residual"
        },
        "milkCarton": {
            "imgXPos": 1475,
            "imgYPos": 200,
            "imageID": "images/plastic/milkCarton.bmp",
            "sortWaste": "pmd"
        },
        "plasticBags": {
            "imgXPos": 75,
            "imgYPos": 600,
            "imageID": "images/plastic/plasticbags.bmp",
            "sortWaste": "pmd"
        },
        "plasticBottle1": {
            "imgXPos": 425,
            "imgYPos": 600,
            "imageID": "images/plastic/plasticbottle1.bmp",
            "sortWaste": "pmd"
        },
        "plasticBottles": {
            "imgXPos": 775,
            "imgYPos": 600,
            "imageID": "images/plastic/plasticbottles.bmp",
            "sortWaste": "pmd"
        },
        "snacks": {
            "imgXPos": 1125,
            "imgYPos": 600,
            "imageID": "images/plastic/snacks.bmp",
```

```
            "sortWaste": "pmd"
        },
        "plasticGloves": {
            "imgXPos": 1475,
            "imgYPos": 600,
            "imageID": "images/plastic/plasticgloves.bmp",
            "sortWaste": "residual"
        },
        "NextButton": {
            "imgXPos": 1275,
            "imgYPos": 830,
            "imageID": "images/selection/nextbutton.bmp",
            "linksTo": "NextPlastic"
        },
        "PreviousButton": {
            "imgXPos": 275,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "main"
        }
    },

    "NextPlastic": {
        "plasticSpoons": {
            "imgXPos": 75,
            "imgYPos": 200,
            "imageID": "images/plastic/plasticspoons.bmp",
            "sortWaste": "residual"
        },
        "plasticSticks": {
            "imgXPos": 425,
            "imgYPos": 200,
            "imageID": "images/plastic/plasticsticks.bmp",
            "sortWaste": "residual"
        },
        "saladBox": {
            "imgXPos": 775,
            "imgYPos": 200,
            "imageID": "images/plastic/saladbox.bmp",
            "sortWaste": "pmd"
        },
        "saucePackets": {
            "imgXPos": 1125,
            "imgYPos": 200,
            "imageID": "images/plastic/sauchepackets.bmp",
            "sortWaste": "residual"
        },
        "PreviousButton": {
            "imgXPos": 775,
            "imgYPos": 830,
            "imageID": "images/selection/previousbutton.bmp",
            "linksTo": "PLASTIC"
        }
    }
}

# The size of all the icons
self.sz = 350;
self.iconSize = (self.sz, self.sz)

self.cx = 475
```

```
self.cy = 700
self.r = 300

# Calculate the angle in degrees
self.val = 360
self.total = 2
self.angle = 360 #self.val * 360 / self.total

# Start list of polygon points
self.p = [(self.cx, self.cy)]

# Get points on arc
for n in range(0, int(self.angle)):
    x = self.cx + int(self.r * math.cos(n * math.pi / 160))
    y = self.cy + int(self.r * math.sin(n * math.pi / 160))
    self.p.append((x, y))
self.p.append((self.cx, self.cy))

# Font text waste types LEFT
#display_surface = pygame.display.set_mode()
pygame.display.set_caption('Show Text')
font = pygame.font.Font('freesansbold.ttf', 20)

self.textPaper = font.render('Paper', True, (255, 255, 255))
self.textRectPaper = self.textPaper.get_rect()
self.textRectPaper.center = (250, 1020)

self.textOrganic = font.render('Organic', True, (255, 255, 255))
self.textRectOrganic = self.textOrganic.get_rect()
self.textRectOrganic.center = (400, 1020)

self.textPMD = font.render('PMD', True, (255, 255, 255))
self.textRectPMD = self.textPMD.get_rect()
self.textRectPMD.center = (550, 1020)

self.textResidual = font.render('Residual', True, (255, 255, 255))
self.textRectResidual = self.textResidual.get_rect()
self.textRectResidual.center = (700, 1020)

self.text1 = font.render('Residual', True, (255, 255, 255))
self.textRect1 = self.text1.get_rect()
self.textRect1.center = (700, 1020)



# Font text waste types LEFT
self.textAmount = font.render('Amount disposed', True, (255, 255,
255))
self.textAmount = pygame.transform.rotate(self.textAmount, 90)
self.textRectAmount = self.textAmount.get_rect()
self.textRectAmount.center = (60, 490)

self.text5a = font.render('5', True, (255, 255, 255))
self.textRect5a = self.text5a.get_rect()
self.textRect5a.center = (1080, 850)

self.text10a = font.render('10', True, (255, 255, 255))
self.textRect10a = self.text10a.get_rect()
self.textRect10a.center = (1080, 700)

self.text15a = font.render('15', True, (255, 255, 255))
```

```
        self.textRect15a = self.text15a.get_rect()
        self.textRect15a.center = (1080, 550)

        self.text20a = font.render('20', True, (255, 255, 255))
        self.textRect20a = self.text20a.get_rect()
        self.textRect20a.center = (1080, 400)

        # Font text amount disposed RIGHT
        self.textUnsure = font.render('Unsure', True, (255, 255, 255))
        self.textRectUnsure = self.textUnsure.get_rect()
        self.textRectUnsure.center = (1250, 1020)

        self.textIncorrect = font.render('Incorrect', True, (255, 255,
255))
        self.textRectIncorrect = self.textIncorrect.get_rect()
        self.textRectIncorrect.center = (1450, 1020)

        self.textCorrect = font.render('Correct', True, (255, 255, 255))
        self.textRectCorrect = self.textCorrect.get_rect()
        self.textRectCorrect.center = (1650, 1020)


        # Font text amount disposed RIGHT
        self.textAmount2 = font.render('Amount disposed', True, (255, 255,
255))
        self.textAmount2 = pygame.transform.rotate(self.textAmount2, 90)
        self.textRectAmount2 = self.textAmount2.get_rect()
        self.textRectAmount2.center = (1050, 490)

        self.text5b = font.render('5', True, (255, 255, 255))
        self.textRect5b = self.text5b.get_rect()
        self.textRect5b.center = (90, 850)

        self.text10b = font.render('10', True, (255, 255, 255))
        self.textRect10b = self.text10b.get_rect()
        self.textRect10b.center = (90, 700)

        self.text15b = font.render('15', True, (255, 255, 255))
        self.textRect15b = self.text15b.get_rect()
        self.textRect15b.center = (90, 550)

        self.text20b = font.render('20', True, (255, 255, 255))
        self.textRect20b = self.text20b.get_rect()
        self.textRect20b.center = (90, 400)

        ### Senna's Program ends here ###

    def calculateDistance(self, x1, y1, x2, y2):
        dist = math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
        return dist

    def drawInformationScreen(self, paperY, organicY, pmdY, residualY,
correctY, incorrectY, notsureY):
        ### Senna's Program starts here ###

        self.screen.blit(self.background_main, (0,0))

        font = pygame.font.Font('freesansbold.ttf', 20)

        self.textPercPaper = font.render(str(self.paperPercent) + '%',
True, (255, 255, 255))
```

```
        self.textRectPercPaper = self.textPercPaper.get_rect()
        self.textRectPercPaper.center = (250, 800)

        self.textPercOrganic = font.render(str(self.organicPercent) + '%',
True, (255, 255, 255))
        self.textRectPercOrganic = self.textPercOrganic.get_rect()
        self.textRectPercOrganic.center = (400, 800)

        self.textPercPMD = font.render(str(self.pmdPercent) + '%', True,
(255, 255, 255))
        self.textRectPercPMD = self.textPercPMD.get_rect()
        self.textRectPercPMD.center = (550, 800)

        self.textPercResidual = font.render(str(self.residualPercent) +
'%', True, (255, 255, 255))
        self.textRectPercResidual = self.textPercResidual.get_rect()
        self.textRectPercResidual.center = (700, 800)

        self.textPercUnsure = font.render(str(self.notsurePercent) + '%',
True, (255, 255, 255))
        self.textRectPercUnsure = self.textPercUnsure.get_rect()
        self.textRectPercUnsure.center = (1250, 800)

        self.textPercIncorrect = font.render(str(self.incorrectPercent) +
'%', True, (255, 255, 255))
        self.textRectPercIncorrect = self.textPercIncorrect.get_rect()
        self.textRectPercIncorrect.center = (1450, 800)

        self.textPercCorrect = font.render(str(self.correctPercent) + '%',
True, (255, 255, 255))
        self.textRectPercCorrect = self.textPercCorrect.get_rect()
        self.textRectPercCorrect.center = (1650, 800)

        # Draw the bar charts on the LEFT
        # Draw the bar charts
        # Paper
        pygame.draw.rect(self.screen, (0, 84, 166), pygame.Rect(195, 1000 -
paperY, 120, paperY))
        # Organic
        pygame.draw.rect(self.screen, (0, 114, 54), pygame.Rect(345, 1000 -
organicY, 120, organicY))
        # PMD
        pygame.draw.rect(self.screen, (242, 101, 34), pygame.Rect(495, 1000
- pmdY, 120, pmdY))
        # Residual
        pygame.draw.rect(self.screen, (54, 54, 54), pygame.Rect(645, 1000 -
residualY, 120, residualY))

        # Bar Chart lines
        pygame.draw.line(self.screen, (255, 255, 255), (110, 1000), (110,
400), 5)
        pygame.draw.line(self.screen, (255, 255, 255), (110, 1000), (850,
1000), 5)

        # number lines
        for i in range(1,20):
            pygame.draw.line(self.screen, (255, 255, 255), (110, 1000-
i*30), (120, 1000-i*30), 5)

        # Draw the bar charts on the RIGHT
        # not sure
```

```
        pygame.draw.rect(self.screen, (0, 156, 195), pygame.Rect(1160, 1000
- notsureY, 180, notsureY))
        # incorrect
        pygame.draw.rect(self.screen, (0, 118, 165), pygame.Rect(1360, 1000
- incorrectY , 180, incorrectY))
        # correct
        pygame.draw.rect(self.screen, (0, 78, 96), pygame.Rect(1560, 1000 -
correctY, 180, correctY))

        # Bar Chart lines
        pygame.draw.line(self.screen, (255, 255, 255), (1100, 1000), (1100,
400), 5)
        pygame.draw.line(self.screen, (255, 255, 255), (1100, 1000), (1800,
1000), 5)
        # number lines
        for i in range(1,20):
            pygame.draw.line(self.screen, (255, 255, 255), (1100, 1000-
i*30), (1110, 1000-i*30), 5)


        # The text
        self.screen.blit(self.textPaper, self.textRectPaper)
        self.screen.blit(self.textOrganic, self.textRectOrganic)
        self.screen.blit(self.textPMD, self.textRectPMD)
        self.screen.blit(self.textResidual, self.textRectResidual)
        self.screen.blit(self.textAmount, self.textRectAmount)
        self.screen.blit(self.textAmount2, self.textRectAmount2)
        self.screen.blit(self.textUnsure, self.textRectUnsure)
        self.screen.blit(self.textIncorrect, self.textRectIncorrect)
        self.screen.blit(self.textCorrect, self.textRectCorrect)
        self.screen.blit(self.text5a, self.textRect5a)
        self.screen.blit(self.text10a, self.textRect10a)
        self.screen.blit(self.text15a, self.textRect15a)
        self.screen.blit(self.text20a, self.textRect20a)
        self.screen.blit(self.text5b, self.textRect5b)
        self.screen.blit(self.text10b, self.textRect10b)
        self.screen.blit(self.text20b, self.textRect20b)
        self.screen.blit(self.textPercPaper, (250, 970 - paperY))
        self.screen.blit(self.textPercOrganic, (400, 970 - organicY))
        self.screen.blit(self.textPercPMD, (550, 970 - pmdY))
        self.screen.blit(self.textPercResidual, (700, 970 - residualY))
        self.screen.blit(self.textPercUnsure, (1250, 970 - notsureY))
        self.screen.blit(self.textPercIncorrect, (1450, 970 - incorrectY))
        self.screen.blit(self.textPercCorrect, (1650, 970 - correctY))

        ### Senna's Program ends here ###

    def drawCorrect(self):

        img = pygame.image.load("images/selection/disposedCorrect.bmp")
        self.screen.blit(img, (25, 0))

    def drawWrong(self):

        img = pygame.image.load("images/selection/disposedIncorrect.bmp")
        self.screen.blit(img, (25, 0))

    def drawNotSelected(self, cat):

        if cat == "paper":
            img = pygame.image.load("images/selection/disposedPaper.bmp")
```

```
        elif cat == "organic":
            img = pygame.image.load("images/selection/disposedOrganic.bmp")

        elif cat == "pmd":
            img = pygame.image.load("images/selection/disposedPMD.bmp")

        elif cat == "residual":
            img =
pygame.image.load("images/selection/disposedResidual.bmp")

        self.screen.blit(img, (25, 0))

    def drawNotSelectedSecondScreen (self, cat):
        if cat == "paper":
            img = pygame.image.load("images/selection/papertypes.bmp")

        elif cat == "organic":
            img = pygame.image.load("images/selection/organictypes.bmp")

        elif cat == "pmd":
            img = pygame.image.load("images/selection/pmdtypes.bmp")

        elif cat == "residual":
            img = pygame.image.load("images/selection/residualtypes.bmp")

        self.screen.blit(img, (25, 0))

    def drawNotSelectedSecondScreenYES(self, cat):

        if cat == "paper":
            img = pygame.image.load("images/selection/ThankYouPaper.bmp")
            self.screen.blit(img, (25, 0))
        elif cat == "organic":
            img = pygame.image.load("images/selection/ThankYouOrganic.bmp")
            self.screen.blit(img, (25, 0))
        elif cat == "pmd":
            img = pygame.image.load("images/selection/ThankYouPlastic.bmp")
            self.screen.blit(img, (25, 0))
        elif cat == "residual":
            img =
pygame.image.load("images/selection/ThankYouResidual.bmp")
            self.screen.blit(img, (25, 0))

    # The method that draws all the selection screens
    def draw(self):
        ### Senna's Program starts here ###
        # Draw a background
        if self.menu == "main":
            background = pygame.transform.scale(self.background,
(self.width, self.height))

        else:
            background = pygame.transform.scale(self.subBackground,
(self.width, self.height))

        self.screen.blit(background, (0, 0))

        # Loop through all the icons in the current menu (self.menu)
        for currentIcon in self.iconControl.get(self.menu):
            # Load the image of the current icon
```

```
            img =
pygame.image.load(self.iconControl.get(self.menu).get(currentIcon).get("ima
geID"))
            # Scale the current icon
            img = pygame.transform.scale(img, self.iconSize)
            # Display the current icon in the position specified in the
iconControl
            self.screen.blit(img,
(self.iconControl.get(self.menu).get(currentIcon).get("imgXPos"),

self.iconControl.get(self.menu).get(currentIcon).get("imgYPos")))

            # If a (sub-selection screen) icon has been selected
            if self.selectedIcon != "":
                # Load the image of the icon of the sort waste selected
                img = pygame.image.load(
                    "images/selection/" +
self.iconControl.get(self.menu).get(self.selectedIcon).get(
                        "sortWaste") + ".bmp")
                # Scale the waste type icon
                img = pygame.transform.scale(img, self.iconSize)
                # Display the waste type icon on the position of the
selected icon
                self.screen.blit(img,
(self.iconControl.get(self.menu).get(self.selectedIcon).get("imgXPos"),

self.iconControl.get(self.menu).get(self.selectedIcon).get("imgYPos")))
        ### Senna's Program ends here ###

    def categoryCounter(self, type):
        if type == "paper":
            self.paperCounter += 1
        elif type == "organic":
            self.organicCounter += 1
        elif type == "pmd":
            self.pmdCounter += 1
        elif type == "residual":
            self.residualCounter += 1


    def categoryCheck(self, type):
        if type == "paper":
            self.selectedCategory = "paper"
            #print("paper")
        elif type == "organic":
            self.selectedCategory = "organic"
            #print("organic")
        elif type == "pmd":
            self.selectedCategory = "pmd"
            #print("pmd")
        elif type == "residual":
            self.selectedCategory = "residual"
            #print("res")

    def percentCaculator(self):
        total = self.paperCounter + self.organicCounter + self.pmdCounter +
self.residualCounter
        total2 = self.correctCounter + self.incorrectCounter +
self.notsureCounter

        if total is not 0:
```

```
            self.paperPercent = int(100 * self.paperCounter / total)
            self.organicPercent = int(100 * self.organicCounter / total)
            self.pmdPercent = int(100 * self.pmdCounter / total)
            self.residualPercent = int(100 * self.residualCounter / total)

        if total2 is not 0:
            self.correctPercent = int(100 * self.correctCounter / total2)
            self.incorrectPercent = int(100 * self.incorrectCounter /
total2)
            self.notsurePercent = int(100 * self.notsureCounter / total2)

    def touch(self, x, y):
        # Loop through all the icons in the current menu (self.menu)
        if True:
            ### Senna's Program starts here ###
            for currentIcon in self.iconControl.get(self.menu):
                # Check if the current icon has been clicked
                if self.calculateDistance(x, y,
self.iconControl[self.menu][currentIcon]["imgXPos"] + self.sz / 2,

self.iconControl[self.menu][currentIcon]["imgYPos"] + self.sz / 2) <
self.sz / 2:

                    # If the current menu is the main selection screen
                    if self.menu == "main" or currentIcon == "NextButton"
or currentIcon == "PreviousButton":
                        # Change the current menu to the menu the selected
icon links to
                        self.selectedIcon = ""
                        self.menu =
self.iconControl.get(self.menu).get(currentIcon).get("linksTo")
                        self.isSelected = True
                        self.screenUsed = True

                    # If the current menu is a sub-selection screen
                    else:
                        # Set the selected icon to the current icon
                        #print("SUB SELECTED")
                        self.selectedIcon = currentIcon
                        self.isSubSelected = True

self.categoryCheck(self.iconControl.get(self.menu).get(self.selectedIcon).g
et("sortWaste"))
                    break
            ### Senna's Program ends here ###

    def systemTimer(self):
        systemTime = pygame.time.get_ticks()
        # Every one second, counter -1
        if systemTime - self.last_systemTime > 1000:
            self.systemCounter += 1
            self.last_systemTime = systemTime
            print(self.systemCounter)

    def timer(self):
        time = pygame.time.get_ticks()
        # Every one second, counter -1
        if time - self.last_time > 1000:
            self.counter += 1
            self.last_time = time
            #print(self.counter)
```

```
def resetScreen(self):
    if self.menu == "main" and self.isSelected:
        self.isSelected = False
        self.isSubSelected = False
        self.run_once = 0
        self.lastPos = ()
        self.resetMain = True


def touchControl(self, x, y, active):
    if active and not self.isSelected:
        if self.run_once == 0:
            self.lastPosMain = (x, y)
            self.run_once += 1
            self.posControl = False
        if self.lastPosMain != (x, y):
            self.posControl = True

    if active and self.isSelected and not self.isSubSelected:
        if self.run_once == 1:
            self.lastPos = (x, y)
            self.run_once += 1
            self.posControl = False

        if self.lastPos != (x, y):
            self.posControl = True

    if self.posControl:
        self.touch(x, y)

def buttonControl(self,x ,y):

    if self.run_once_button == 0:
        self.lastPos = (x, y)
        self.run_once_button += 1
        self.posControlButton = False

    if self.lastPos != (x, y):
        self.posControlButton = True

    if self.posControlButton:
        if x <= 730 and x >= 515 and y <= 975 and y >= 875:
            self.button = "yes"
        elif x <= 1000 and x >= 800 and y <= 975 and y >= 875:
            self.button = "no"
        elif x <= 1380 and x >= 1080 and y <= 975 and y >= 875:
            self.button = "notSure"

def loop(self, active, category):
    if not active:
        paperY = self.paperCounter * 30
        organicY = self.organicCounter * 30
        pmdY = self.pmdCounter * 30
        residualY = self.residualCounter * 30
        correctY = self.correctCounter * 30
        incorrectY = self.incorrectCounter * 30
        notsureY = self.notsureCounter * 30
        self.percentCaculator()
        self.drawInformationScreen(paperY, organicY, pmdY, residualY,
```

```
correctY, incorrectY, notsureY)

        if active:
            self.systemTimer()

            mouseX = pygame.mouse.get_pos()[0]
            mouseY = pygame.mouse.get_pos()[1]

            self.draw()
            self.resetScreen()

            if category:
                self.disposed = True

            if self.isSubSelected:
                if self.selectedCategory == category:
                    self.drawCorrect()
                    self.timer()
                    self.button = "disposed"
                    if self.run_once_counter:
                        self.categoryCounter(category)
                        self.correctCounter += 1
                        self.run_once_counter = False
                elif category is not None:
                    self.drawWrong()
                    self.timer()
                    self.button = "disposed"
                    if self.run_once_counter:
                        self.categoryCounter(category)
                        self.incorrectCounter += 1
                        self.run_once_counter = False

            elif category and not self.isSubSelected:
                self.drawNotSelected(category)
                self.selectedCategory = None

                if self.run_once_counter:
                    self.categoryCounter(category)
                    self.run_once_counter = False

                if self.button is None:
                    self.buttonControl(mouseX, mouseY)

                if self.button == "notSure":
                    self.drawNotSelectedSecondScreen(category)

                    if self.run_once_button == 1:
                        self.notsureCounter += 1
                        self.counter = 0
                        self.run_once_button += 1
                    self.timer()

                elif self.button == "no":
                    self.drawNotSelectedSecondScreen(category)

                    if self.run_once_button == 1:
                        self.incorrectCounter += 1
                        self.counter = 0
                        self.run_once_button += 1
                    self.timer()
```

```
        elif self.button == "yes":
            self.drawNotSelectedSecondScreenYES(category)

            if self.run_once_button == 1:
                self.correctCounter += 1
                self.counter = 0
                self.run_once_button += 1
            self.timer()
        else:
            self.timer()

    if not self.disposed:
        self.touchControl(mouseX, mouseY, active)

pygame.display.flip()
```

Appendix C. Survey Questions

General Questions

Q1. How difficult did you find separating the waste?

Q2. Why would you say it was easy/difficult?

Q3. I'm confident that I separated my waste into the proper categories.

Q4. I felt satisfied with the accomplishment I got from separating my waste.

Q5. The system was convenient to use.

Q6. The system gave me informative feedback.

Q7. The system influences my future waste separation behaviour positively.

Questions only asked to group 2, related to the waste island.

Q1. The LED lights were clearly visible.

Q2. The blinking LED lights were helpful to find the correct waste bin.

Q3. When you got close to the waste island, the system detected you and changed the screens smoothly.

Q4. The screen-touch interaction was smooth.