Impact Terrarium

By Shareen Esmail

THIS IS DIFFERENT FROM FINAL PRESENTATION

(this also includes a conclusion and more images from the exhibition and more...)

Table of contents

01

Research

Understanding User Context

Problem

A gap emerged between what people perceived their impact on the world and its actual implications. Interviews revealed a sense of hopelessness regarding global warming. This Sense urges behaviors towards sentiments such as 'What difference does it make?'

Key User
Persona

Age:
21
Pronouns:
she/her/they/them
Family:
no kids, not married,
lives on her own
Occupation:
Part time student and part time worker in retail
Stresses:
Worried that what she does isnt important. That it doesn't make a difference in the grand scheme of things
Quotes:
*"It's just one can, it's not like the world isn't already f***ed up anyway"*

Feyre Archeron

Overview

This project focuses on creating personalized terrariums that resemble local parks or green areas, aiming to foster a deeper connection for individuals. Through these customized terrariums, I seek to address the widespread sense of hopelessness surrounding environmental issues like global warming. By illustrating the impact of individual actions within the context of a familiar green space, my goal is to empower individuals to recognize the significance of their choices.

02 Ideation

Conceptualizations

Cultural and Contextual Research

Technology Exploration

Conceptualization

Mind *Map*

My initial mindmap was just a general brainstorm of how I have seen hybrid interactions between nature and technology. I quickly ran out of ideas and started to just think about interactions that I personally experienced as hybrid interactions. One of the big ones being a project of mine from last semester.

1st empathy map

My first empathy map was just of random questions I had asked my brother. Questions with answers below:

What are you thinking and saying
what do you say about nature
like literally anything
ANYTHING PLS

I dont knoww

I am thinking about the complexity of nature

and saying "nature is cool"

anything about global warming are you saying

I am saying "the planets on fire"

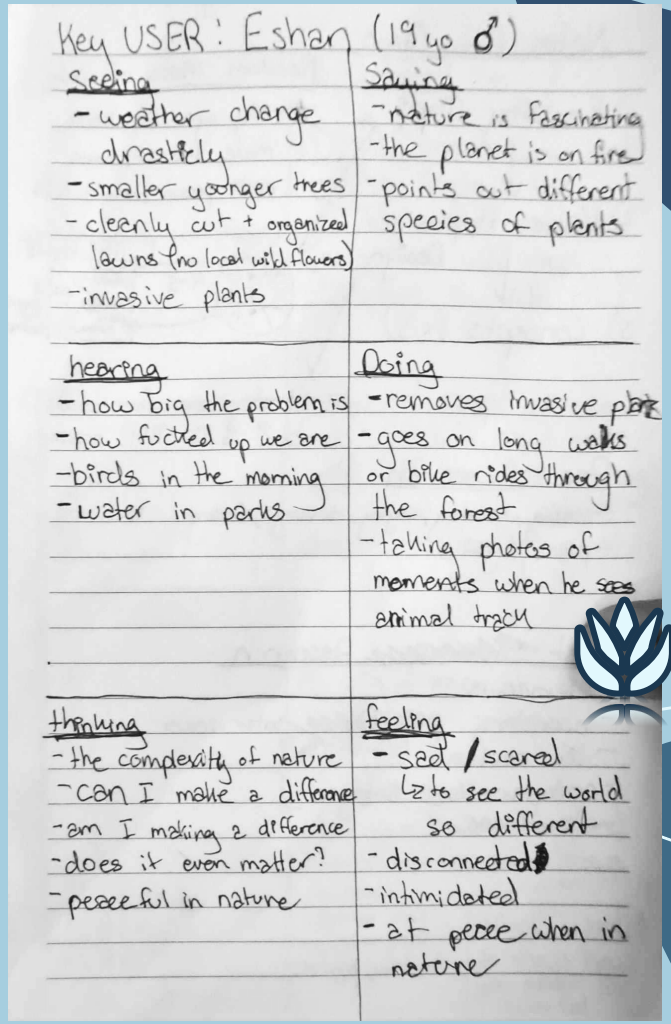
anything else you are thinking
like do you have a positive outlook in life about nature
or a negative one
when do you run into nature

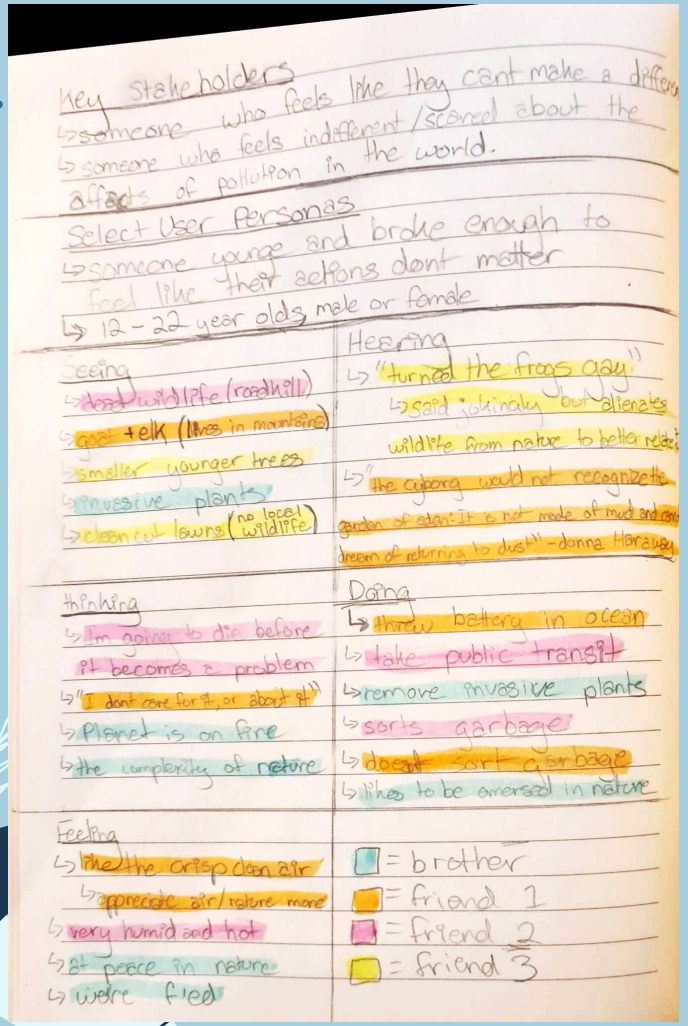
positive ig?

I see nature when I see suburban animals

like crows squirrels or raccoons

I see it at parks n stuff]





Final empathy map

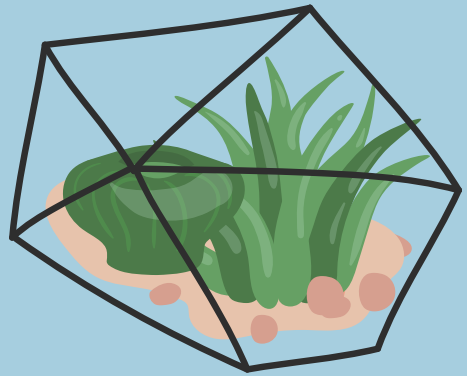
This empathy map was made while I was on call with my friends. I had tried to combine some of their answers with my brother so that I could have 1 final empathy map.

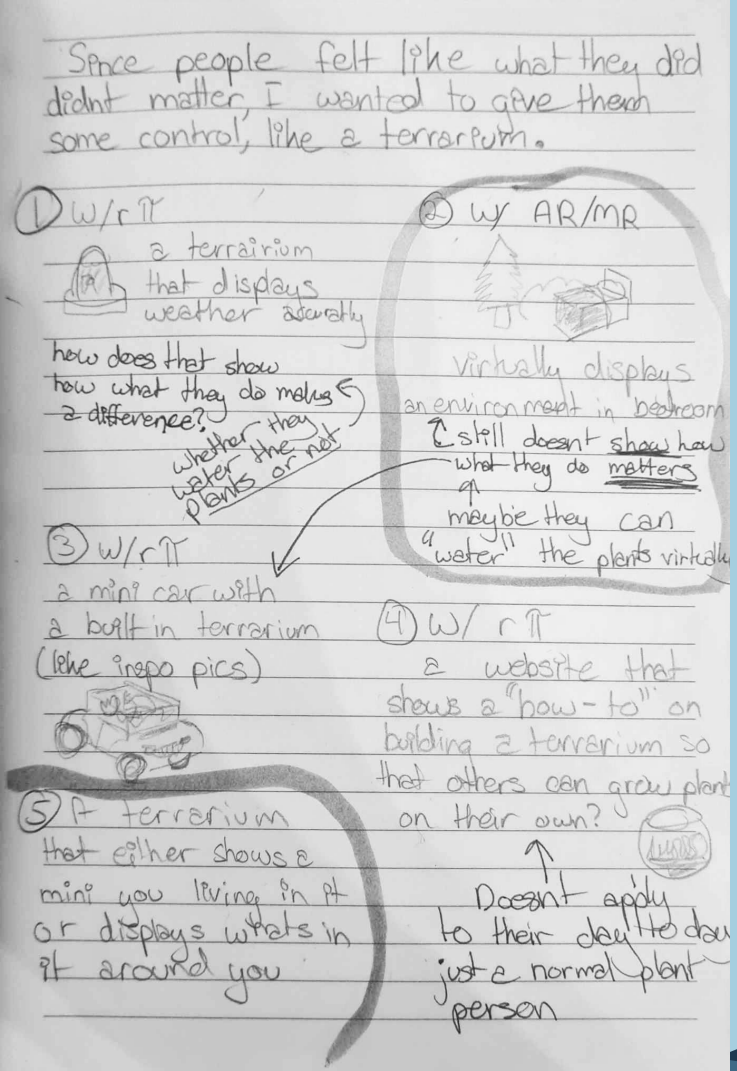
Their answers we just as sincere as my brothers. I had to go through and reword some of what they said so that it was more of an answer instead of a spewing of memes and jokes

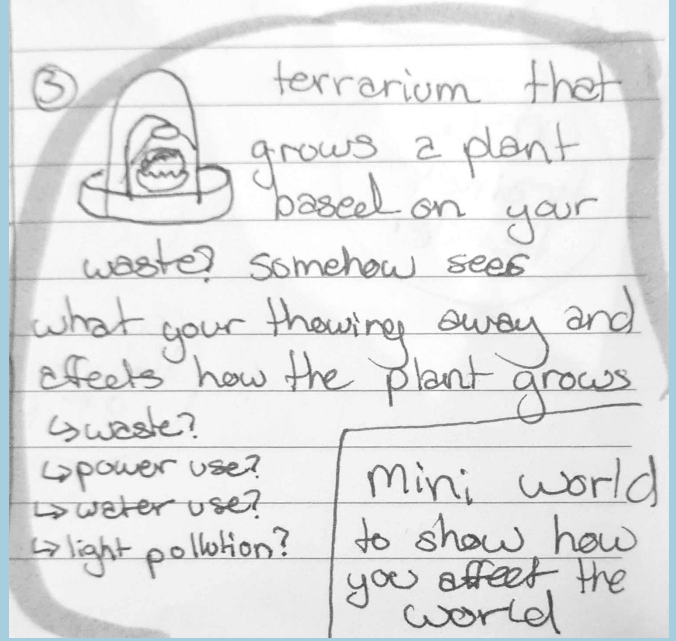
Reflecting on interviews

What I noticed is that most of who I interviewed either seemed scared or indifferent about nature and their impact on it. Indifferent, not in that way as if they didnt care what was happening, but it was in sense as if they felt what they did didn't matter. They felt as if they couldn't change it so they didnt do anything, and their fear showed itself whenever they were confronted with their feeling of helplessness. They felt they were sooner likely to die, than they would see/create a noticeable change in the world.

Yet they all had an inexplicable connection to nature. I want my project to give my friends and others hope. To not be so pessimistic and prove that what they do matters.





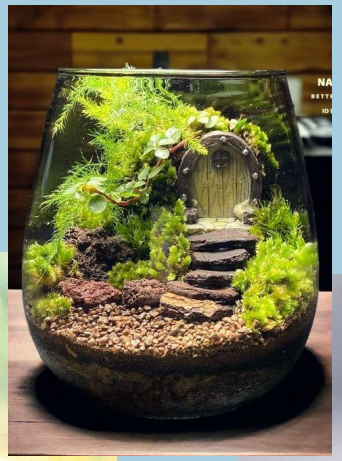


Concept Brainstorming

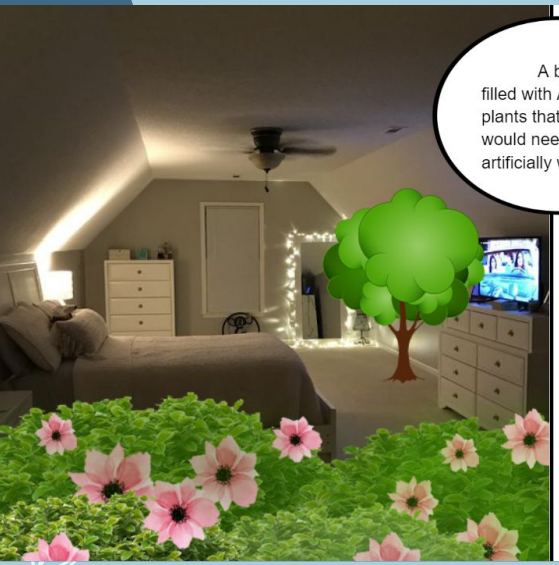
Quick sketches of ideas that related to terrariums. I tried mixing some AR/MR because I thought I could have enough time to make a big enough project that uses both.

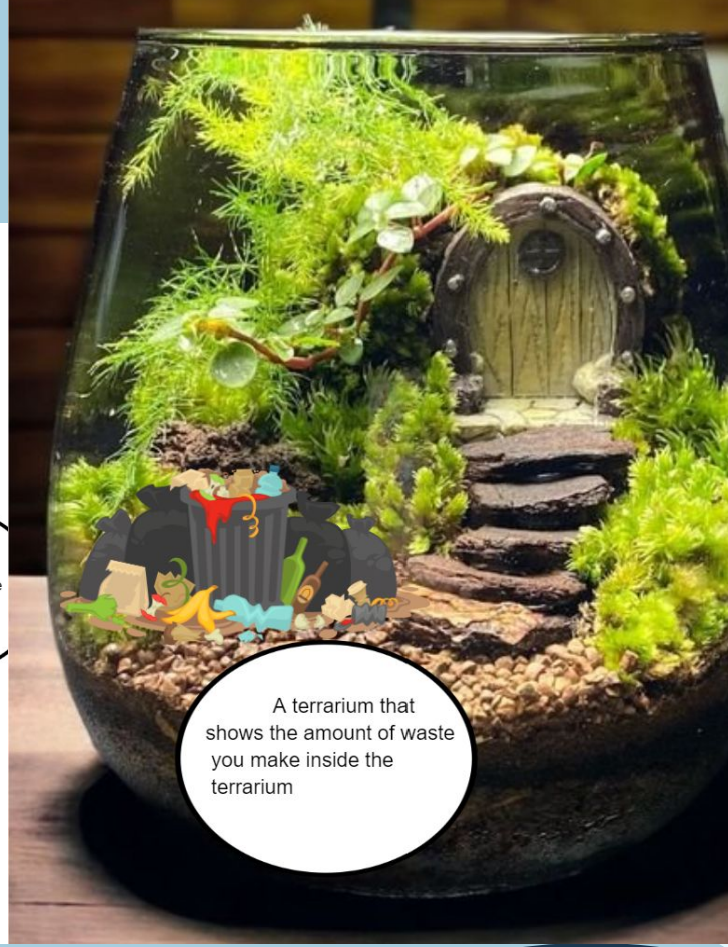
Inspiration

Inspiration

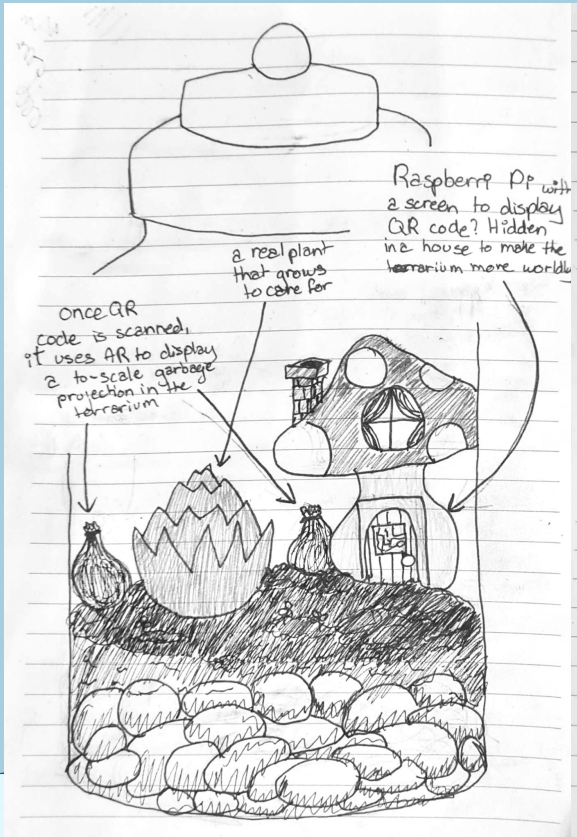


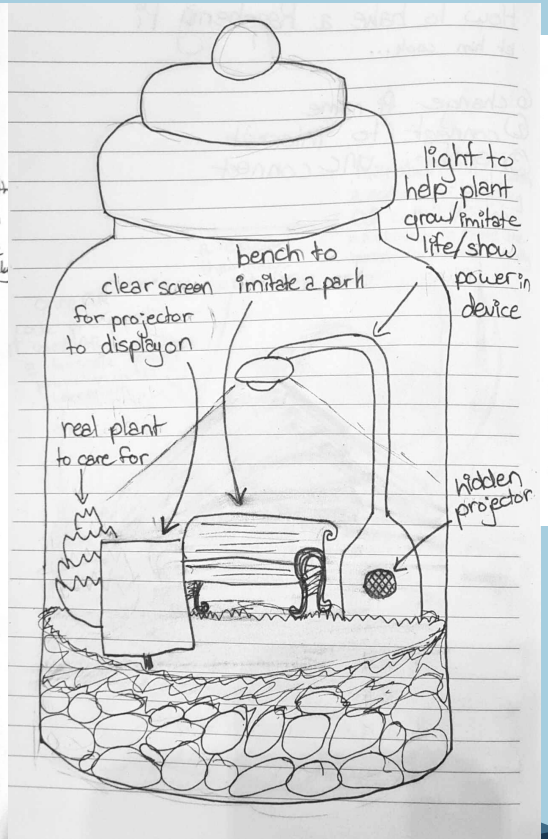
Sketching Process

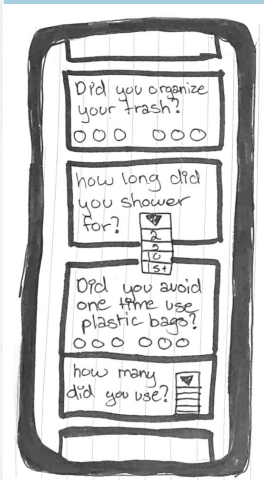


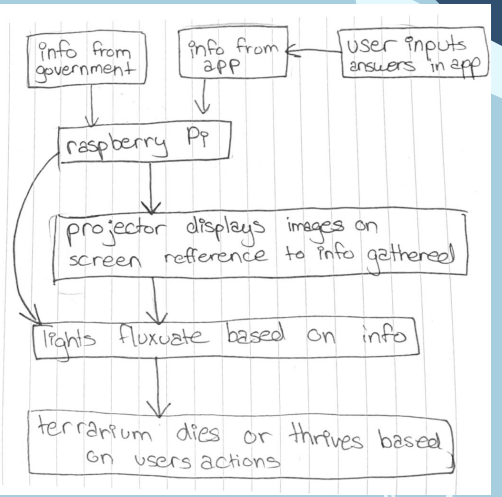


Refining Chosen Concept







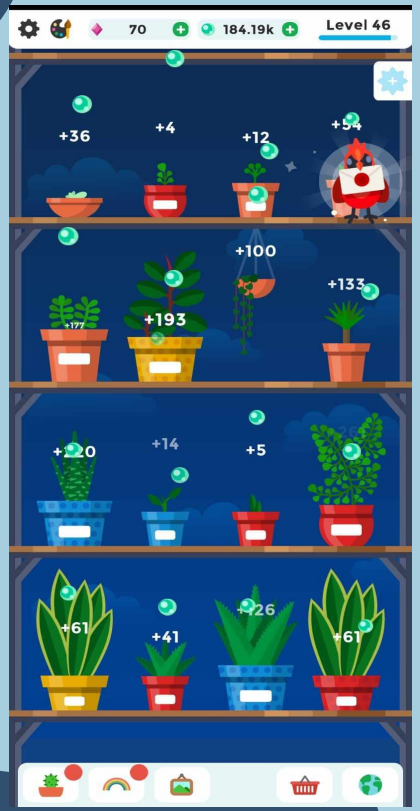


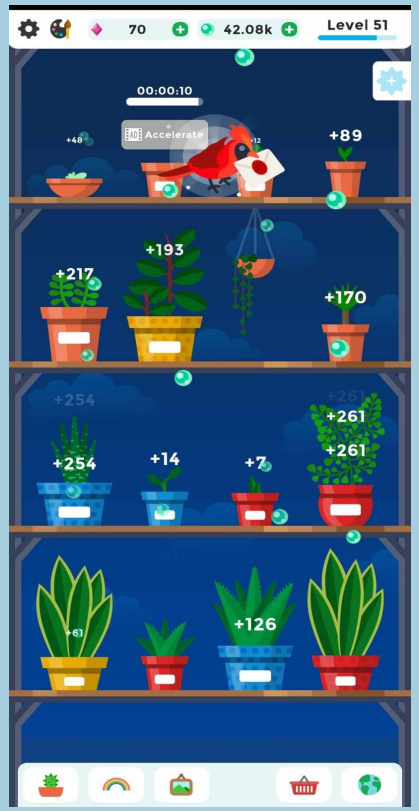
Cultural and Contextual research

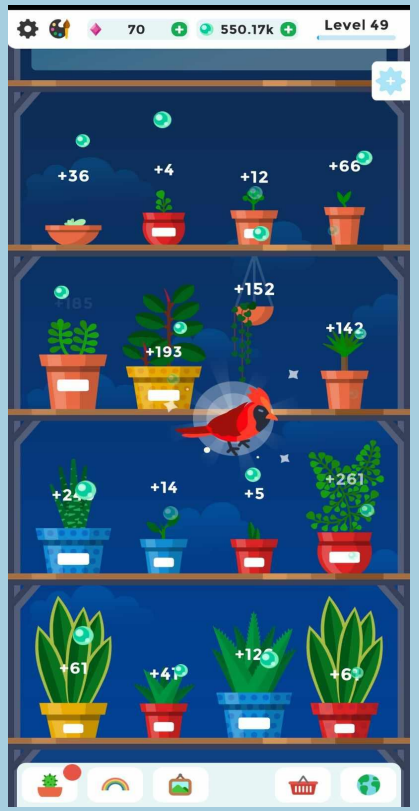
Terrarium Research

I research anything and everything related to terrariums

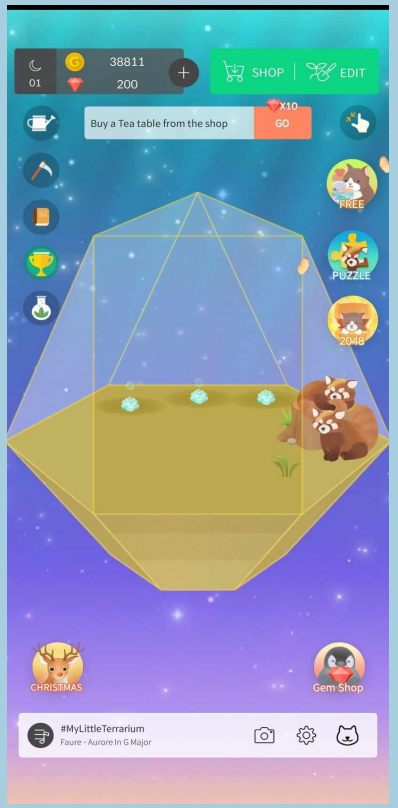
Terrarium: Garden Idle





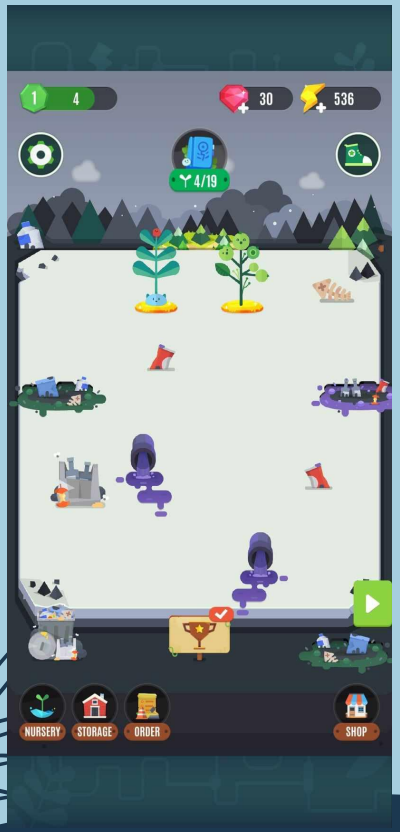


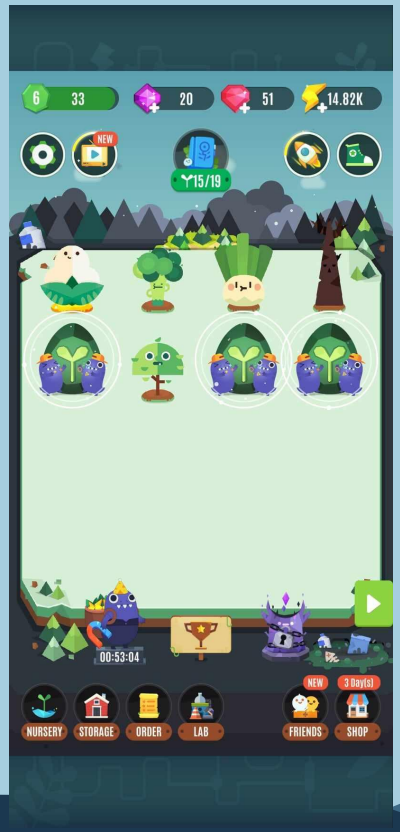
My Little Terrarium: Idle Game

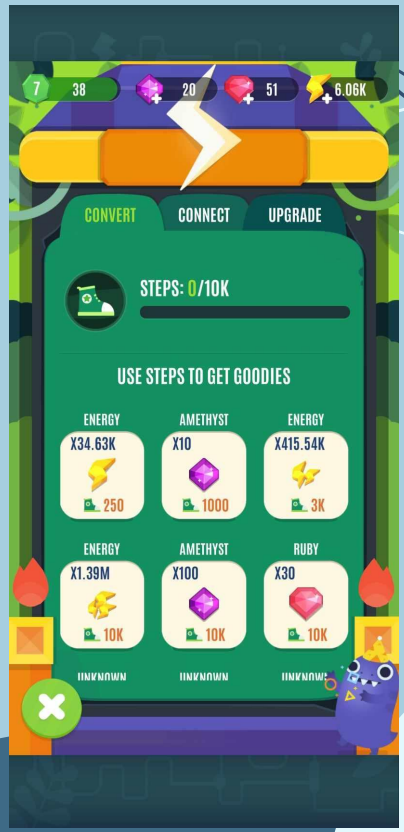


Tap Tap Fish AbyssRium (+VR)

Pocket Plants: Grow Plant Game







Making my own terrarium

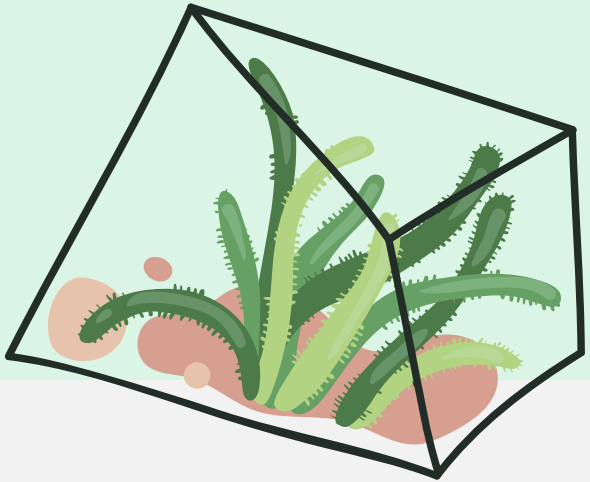
The Beginning

The End

Impact *Terrarium*

Design Brief

By Shareen Esmail

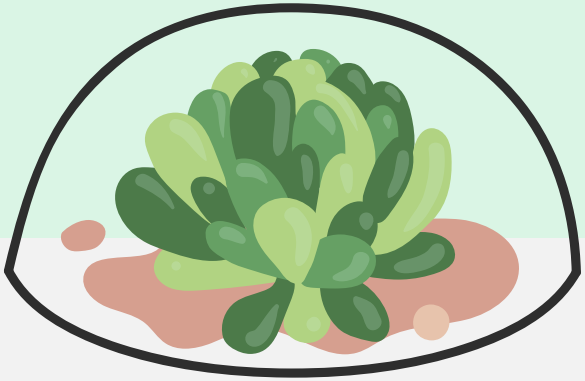


Why

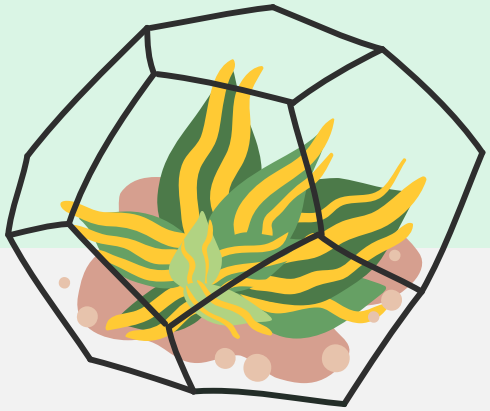
I plan to create a terrarium that depicts the users impact in relation to global warming in a more visible and relatable environment. By displaying how their actions, such as: if they recycle, how much water or power they use, impact the terrarium and the plant(s) that live in it, I hope to create a bond between the user and nature that gives a new perspective as to how detrimental their actions on the environment can be.

What

There will be a light that will either aid in the growth of the plant(s) or be detrimental to its growth. It will change depending on the information gathered from the 2 following sources: From companies in charge of natural resources (ex. BCHydro) or from information collected by the government (electricity use, garbage/recycling pick up, and from an application that surveys the user on basic information that may not be available through the above listed sources. The information gathered will not only affect the light and therefore the plant(s) growth, but will also affect what the projector installed inside the terrarium will display. What it may display ranges from tumbleweeds, powerboxes, garbage bags, ect. It must be designed in a manner to appear as something the user will recognize when they leave their house (a park setting, greenery by a city building, ect.)



Who



It should be marketed to those who are starting to feel as if their future doesn't matter. To those who believe that their actions have no impact in the world. People between the ages of 18-25 would be ideal, since they have the power to impact and make changes in the world but may not realize it.

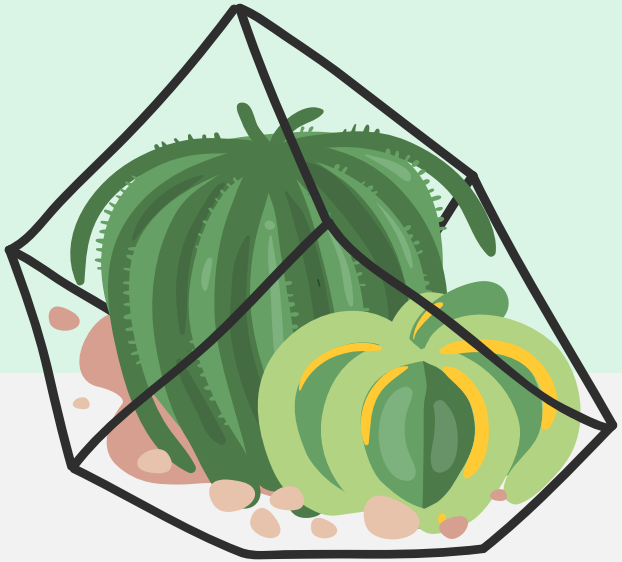
Project Timeline

Reading Week	Feb. 26th	Mar. 4th	Mar. 11th	Mar. 18th	Mar. 25th	April 3rd
Figure out how to set up projector to RPi By now I should have my projector and should be trying to get it connected to my RPi						
	Design images for projector to use					
		Get RPi connected to internet, and how to receive and use information from an online source				
				Design Figma for app/ Set up terrarium		
					Presentation and final touches	

Project Plan

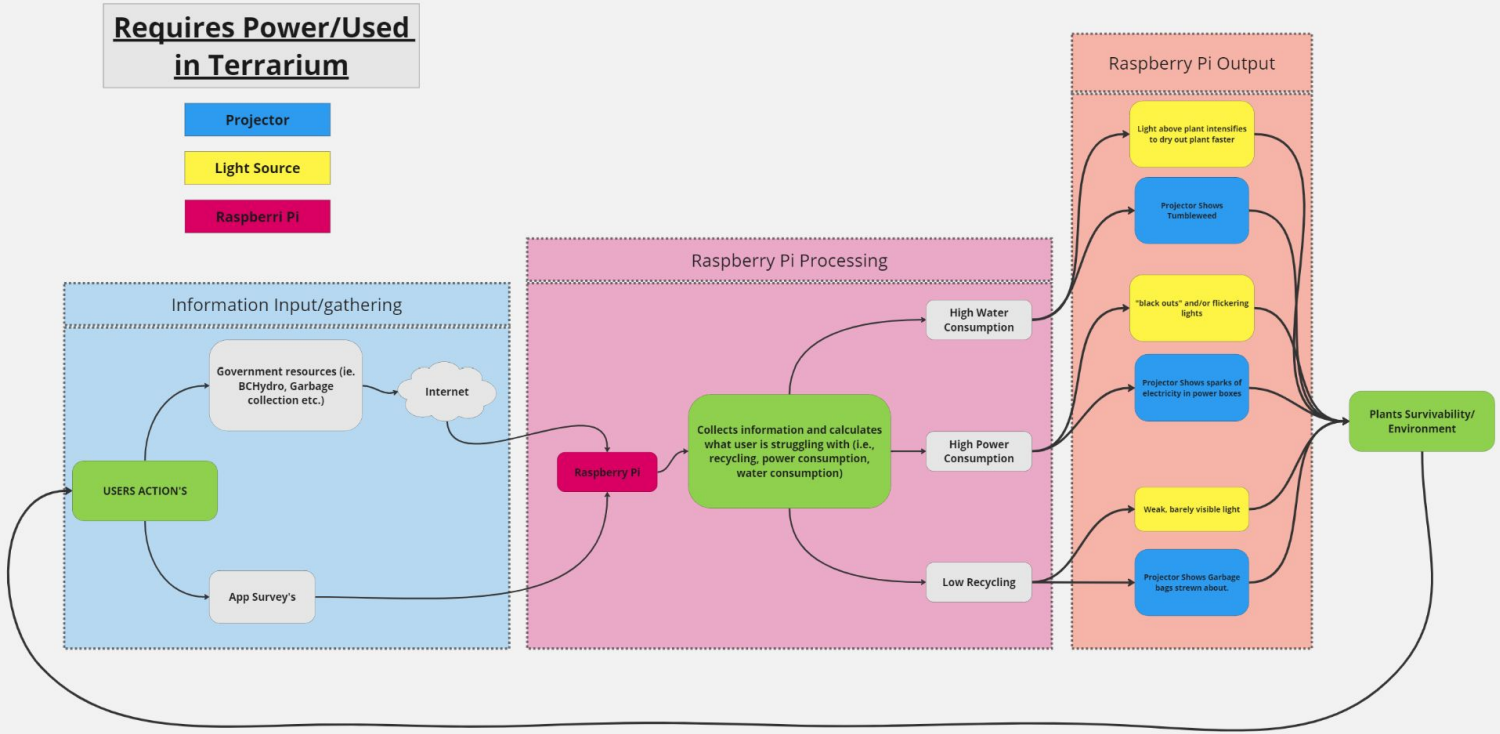
What will need to be done for prototype

- At least 3 images for projector to display
- Raspberry Pi is able to read information on it and display correct images on projector
- Raspberry Pi able to turn LED light off and on based on information received from it
- 3-5 Figma screens for accompanying app



Technology Exploration

What does it do?



Building 03

Prototyping

Digital Development

Physical Construction

Integration

Digital Construction

How does it do it?
1st iteration

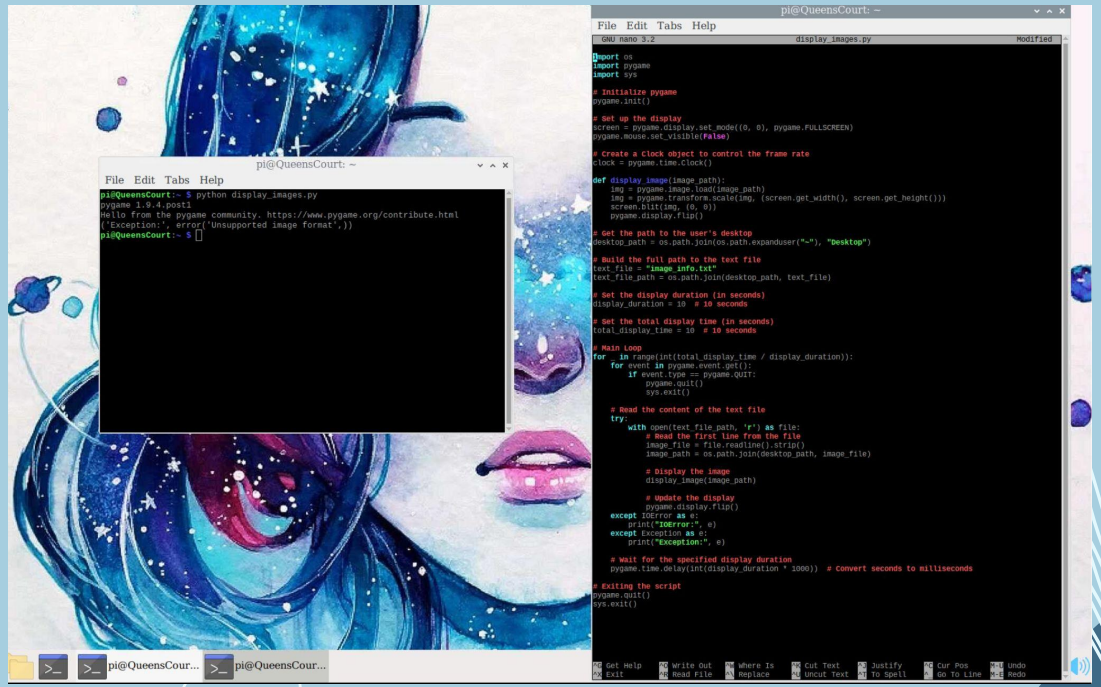
Explanation

What is it doing?

- Shows image
- Closes after 5s
- Choosing image from text file

Why?

To imitate grabbing information from an online source



Asking the questions

The set up

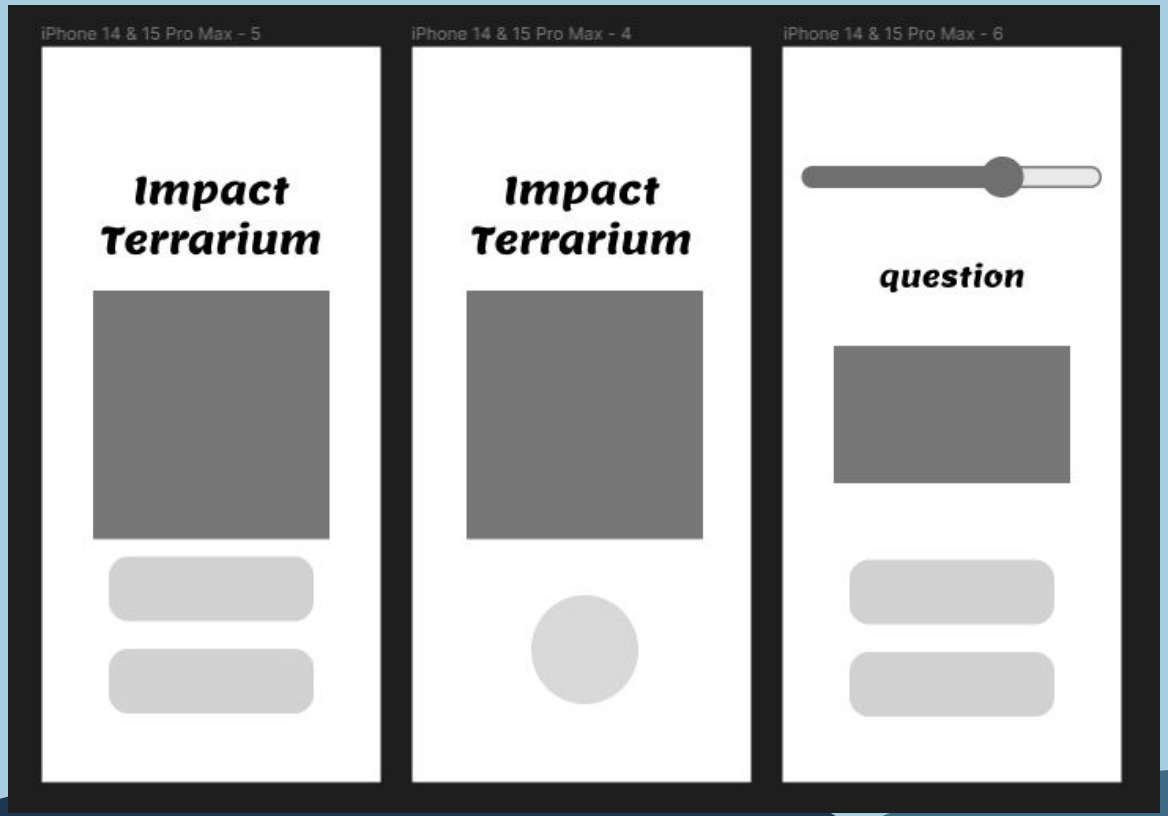
- Creating questions
- Calculating the score
- Display the image relating to the score
  - Good person
  - Medium person
  - Bad person

```
questions = [
    "Did you use a plastic bag today?",
    "Did you recycle your waste today?",
    "Did you use public transportation instead of a car today?"
]
```

```
def calculate_score(responses):

    score = 0
    for response in responses:
        if response.lower() == "no":
            score += 1
    return score
```

```
# Display image based on score
if score <= 1:
    image_filename = "good_person.jpg"
elif score <= 2:
    image_filename = "medium_person.jpg"
else:
    image_filename = "bad_person.jpg"
```

Rough Figma Wireframe



Prototyping

User testing results

Goal

To test my code and see if it worked the way i intended it too.

Result 1

The questions were to simple, and straightforward. Users could just answer no and get the "good" answer right away.

Result 2

Not enough questions. Some could be more important than others, and good person, medium person and bad person is a bad scale to judge people.

New Questions

Explanation for each decision

- Category
  - Instead of sorting people by a good, medium and bad category, it organizes people by what they struggle the most with. (i.e. conserving power, conserving water or recycling)
- Weight
  - How important is the question? How much does it affect the environment?
- "Positive": "bad", "negative": "good"
  - This way, users cant keep answering no to get points, and would need to read through each question fully. This shows whether someone answers negatively/positively (no and yes) and if that is a good thing or bad thing

```
# Define questions with their weights and scoring systems
questions = [
    # Electricity usage question
    {"category": "electricity", "question": "Do you leave the lights on when leaving a room?", "weight": 2, "positive": "bad", "negative": "good"},
    {"category": "electricity", "question": "Do you use energy-saving appliances?", "weight": 3, "positive": "good", "negative": "bad"},
    # Water usage question
    {"category": "water", "question": "Do you take shorter showers to save water?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you leave the faucet running unnecessarily?", "weight": 3, "positive": "bad", "negative": "good"},
    # Recycling question
    {"category": "recycling", "question": "Do you recycle plastic bottles regularly?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you throw recyclable items in the trash?", "weight": 3, "positive": "bad", "negative": "good"},
]
```

Questions

```python
def calculate_score(responses):
    category_scores = {"electricity": 0, "water": 0, "recycling": 0}
    for i, response in enumerate(responses):
        question = questions[i]
        category = question["category"]
        if response.lower() == "yes":
            if question["positive"] == "good":
                category_scores[category] += question["weight"]
            elif question["positive"] == "bad":
                category_scores[category] -= question["weight"]
        elif response.lower() == "no":
            if question["negative"] == "good":
                category_scores[category] += question["weight"]
            elif question["negative"] == "bad":
                category_scores[category] -= question["weight"]
    return category_scores
```

How does it calculate?

Even more questions

```python
# Define questions with their weights and scoring systems
questions = [
    {"category": "electricity", "question": "Do you leave the lights on when leaving a room?", "weight": 2, "positive": "bad", "negative": "good"},
    {"category": "electricity", "question": "Do you use energy-saving appliances?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you unplug electronics when they are not in use?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you use natural light during the day instead of turning on lights?", "weight": 1, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you turn off power strips when not in use?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you use a programmable thermostat to regulate energy usage?", "weight": 4, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you rely on natural ventilation instead of air conditioning whenever possible?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you take shorter showers to save water?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you leave the faucet running unnecessarily?", "weight": 3, "positive": "bad", "negative": "good"},
    {"category": "water", "question": "Do you fix leaky faucets promptly to conserve water?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you use a dishwasher instead of washing dishes by hand?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you collect rainwater for outdoor use?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you use a low-flow showerhead to conserve water?", "weight": 4, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you wash full loads of laundry instead of partial loads?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you recycle plastic bottles regularly?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you throw recyclable items in the trash?", "weight": 3, "positive": "bad", "negative": "good"},
    {"category": "recycling", "question": "Do you compost organic waste instead of throwing it away?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you separate recyclables from regular trash?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you participate in community recycling programs?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you purchase products made from recycled materials?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you educate others about the importance of recycling?", "weight": 5, "positive": "good", "negative": "bad"}
]
```

How to stop the black screen of death

While True

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
# Read the content of the text file
    Try:
     with open(text_file_path, 'r') as file:
            # Read the first line from the file
            image_file = file.readline().strip()
            image_path = os.path.join(desktop_path, image_file)

            # Display the image
            display_image(image_path)

            # Update the display
            pygame.display.flip()

            # Check if the display duration has elapsed
            if pygame.time.get_ticks() - timer_start >= display_duration:
                # Reset the timer
                timer_start = pygame.time.get_ticks()
    except IOError as e:
        print("IOError:", e)
    except Exception as e:
        print("Exception:", e)
```

For Loop

```
# Main Loop
for _ in range(int(total_display_time / display_duration)):
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # Read the content of the text file
    try:
        with open(text_file_path, 'r') as file:
            # Read the first line from the file
            image_file = file.readline().strip()
            image_path = os.path.join(desktop_path, image_file)

            # Display the image
            display_image(image_path)

            # Update the display
            pygame.display.flip()
    except IOError as e:
        print("IOError:", e)
    except Exception as e:
        print("Exception:", e)

    # Wait for the specified display duration
    pygame.time.delay(int(display_duration * 1000))  # Convert seconds to milliseconds
```

The evolution of input

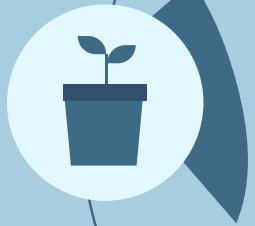
1st round

Could only read the first line from a text file

```
with open(text_file_path, 'r') as file:
        # Read the first line from the file
        image_file = file.readline().strip()
        image_path = os.path.join(desktop_path,
image_file)l
```

2nd round

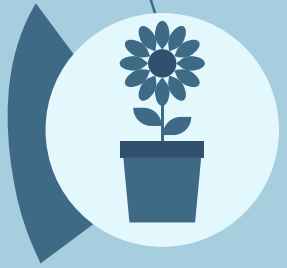
Could ask with its own input and window what file to open



```
# Display prompt to enter the image file name
   font = pygame.font.Font(None, 36)
   prompt_text = font.render("Enter the name of the image file (including
extension):", True, (255, 255, 255))
   screen.blit(prompt_text, (10, 10))
   pygame.display.flip()
```

3rd round

Could ask 1 question and display 1 image no matter the answer



```
def ask_for_response():
   font = pygame.font.Font(None, 36)
   prompt_text = font.render("Did you use a
plastic bag today? (Yes/No)", True, (255, 255,
255))
   screen.blit(prompt_text, (10, 10))
```

pi@QueensCourt: ~

File Edit Tabs Help

```
pi@QueensCourt:~ $ python display_images.py
pygame 1.9.4.post1
Hello from the pygame community. https://www.pygame.org/contribute.ht
('Exception:', error('Unsupported image format',))
pi@QueensCourt:~ $ python trail2.py
pygame 1.9.4.post1
Hello from the pygame community. https://www.pygame.org/contribute.ht
```

pygame window

Did you use a plastic bag today? (Yes/No)

GNU nano 3.2 trail2.py

```python
# Get the path to the user's desktop
desktop_path = os.path.join(os.path.expanduser("~"), "Desktop")

# Main Loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    ask_for_response()

    # Wait for user input
    user_input = ""
    input_finished = False  # Flag to indicate whether Enter has been pressed
    while not input_finished:
        for event in pygame.event.get():
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_BACKSPACE:
                    user_input = user_input[:-1]
                elif event.key == pygame.K_RETURN or event.key == pygame.K_KP_ENTER:
                    input_finished = True
                else:
                    user_input += event.unicode

        screen.fill((0, 0, 0))  # Clear the screen
        ask_for_response()  # Redraw prompt text
        font = pygame.font.Font(None, 36)
        input_text = font.render(user_input, True, (255, 255, 255))
        screen.blit(input_text, (10, 50))

        pygame.display.flip()
        clock.tick(30)

    # Check if the input is either "Yes" or "No"
    if user_input.lower() not in ("yes", "no"):
        display_error_message("Sorry, I didn't get that.")
        pygame.display.flip()
        pygame.time.delay(2000)
        continue  # If not, ask the question again

    # Build the image path based on user's response
    image_filename = "plastic_bag_" + user_input.lower() + ".jpg"
    image_path = os.path.join(desktop_path, image_filename)

    # Check if the image file exists
    if not os.path.exists(image_path):
        error_message = "Image not found."
        pygame.time.delay(2000)
        continue

    # Display the image
    display_image(image_path)
    pygame.display.flip()
    pygame.time.delay(5000)  # Display the image for 5 seconds

pygame.quit()
sys.exit()
```

Script for user testing

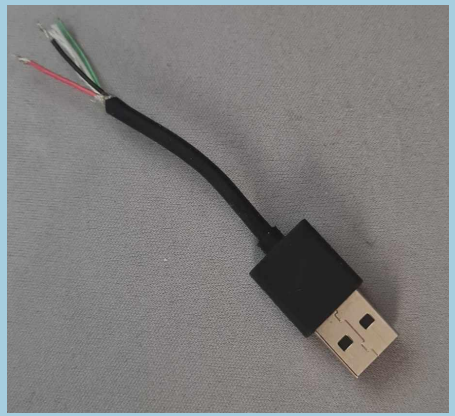
Hello, in case you don't know my name is Shareen and this is my user testing! The main process of this testing is mainly to see if my code works so this is going to be a very low stakes test and should only maybe take 15 minutes? I do also have a couple of questions to ask you while you go through this testing such as:

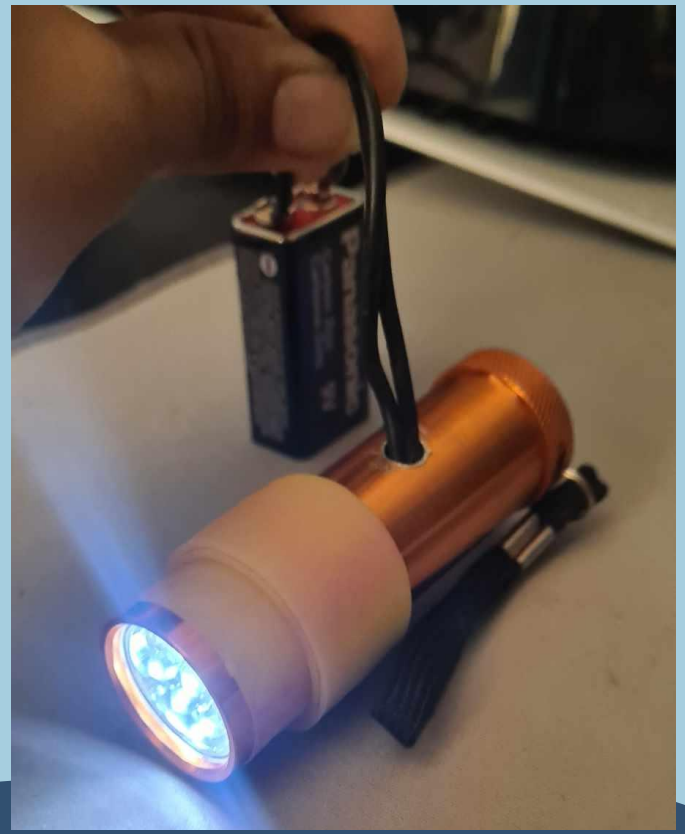
Do you understand why this program is asking these questions?
Do the questions make sense? (In the way that they are phrased)
Can you tell that you are only supposed to answer yes or no?
Is it too easy or difficult to answer these questions?
What questions would you add that you think are important?
Do the images that are displayed make sense/relate to the way that you answered these questions?

Thank you for participating in my user testing, please let me know if there is anything else i should add or do.

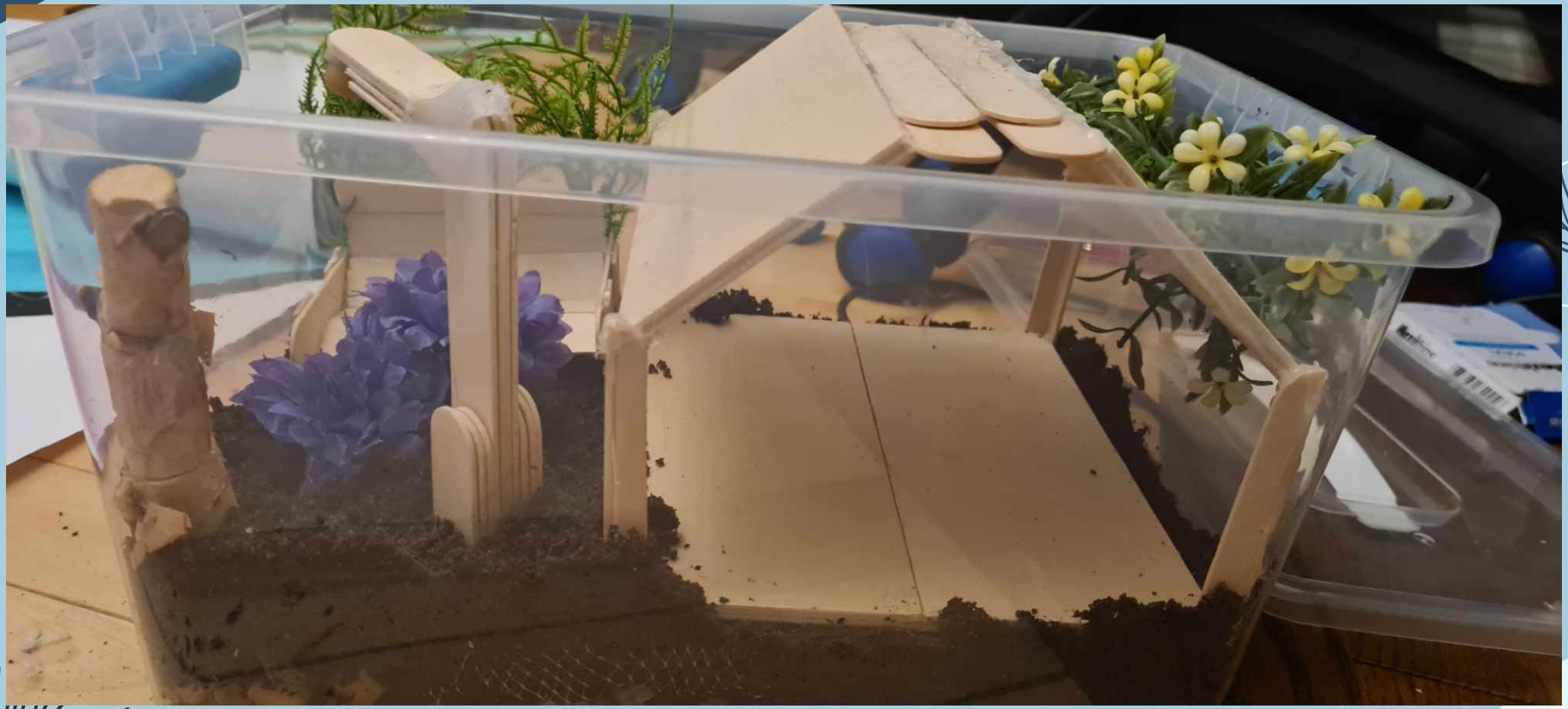
Physical Construction

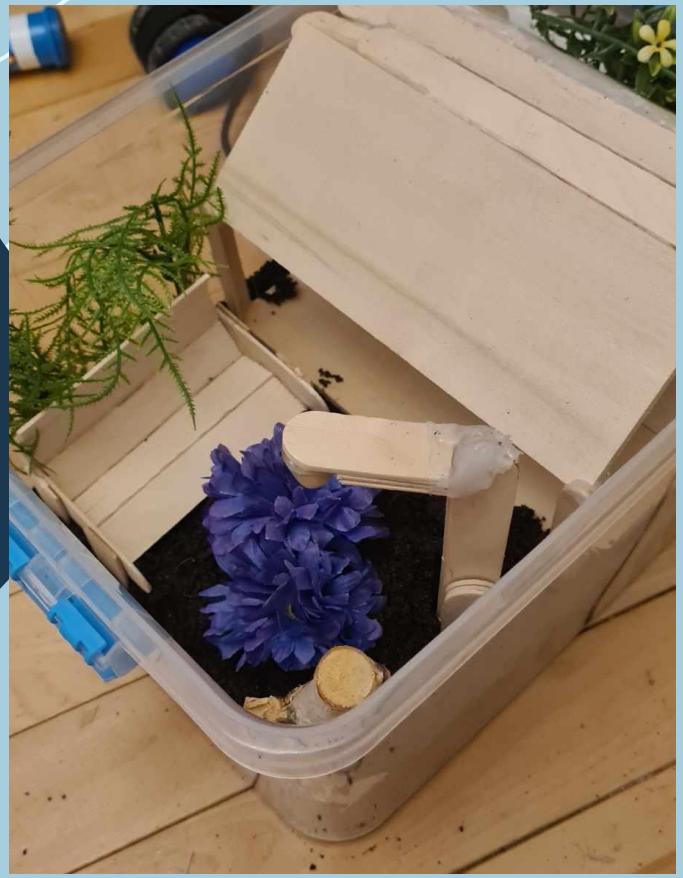
TRYING to make a light





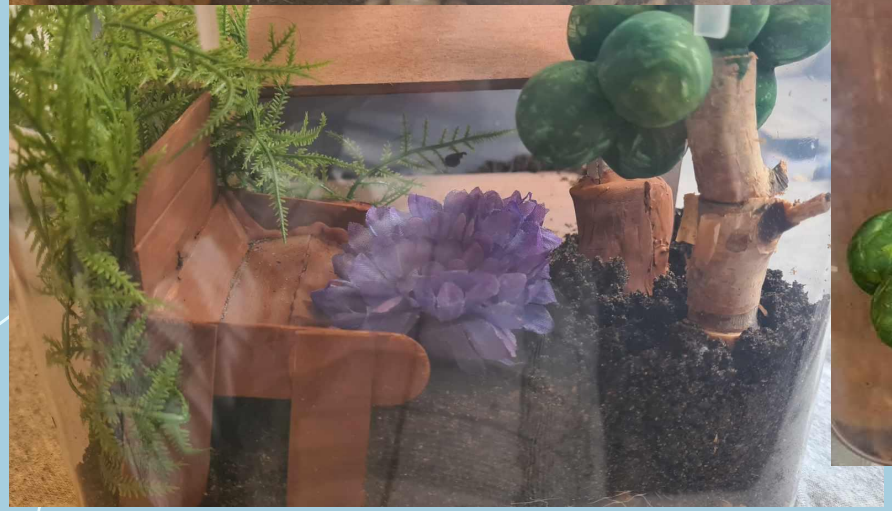
Making Terrarium



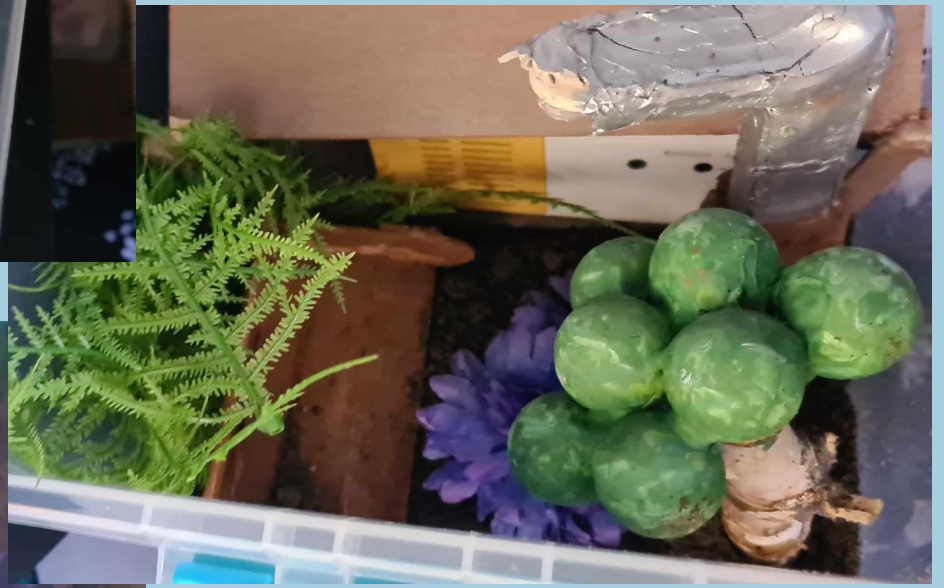


The terrarium

Terrarium



Integration



04 Refinement

Iterative Refinement

Optimization

Coding Process

A huge Thank You to Chat GPT for helping me code a large chunk of this

Problems I faced

Black Screen

Wasn't finding image
Wasn't closing program
Can't manually close program

Backspace

It simply did not work and I had not a single clue as to why

Error Message

Either it wouldn't appear, move to the next question anyway, or display the message but then not let you type

Questions

Getting a window set up to display and even ask questions

GIFS

No explanation, it just didn't work. I picked out some really cool ones too

Flashlight

Also no explanations. I don't really have one. It just didn't work and there wasn't enough time

Calculating

It was counting points wrong this whole time... had imagined it working in the first place???

terrarium

My test terrarium plant died!!!

Chat GPT

I feel a little bad about it but I did rely on chat GPT alot for this project since I had never coded on Raspberry Pi before and I had no idea where to start

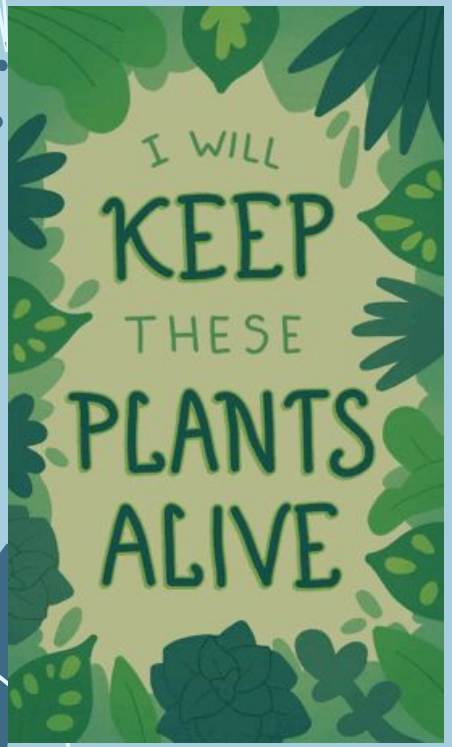

```
screen.fill((0, 0, 0))  # Clear the screen
ask_question(question)  # Redraw prompt text
font = pygame.font.Font(None, 36)
input_text = font.render(user_input, True, (255, 255, 255))
screen.blit(input_text, (10, 50))

# Display error message if response is invalid
if invalid_response:
    display_error_message("Invalid response. Please type 'yes' or 'no'.")
```

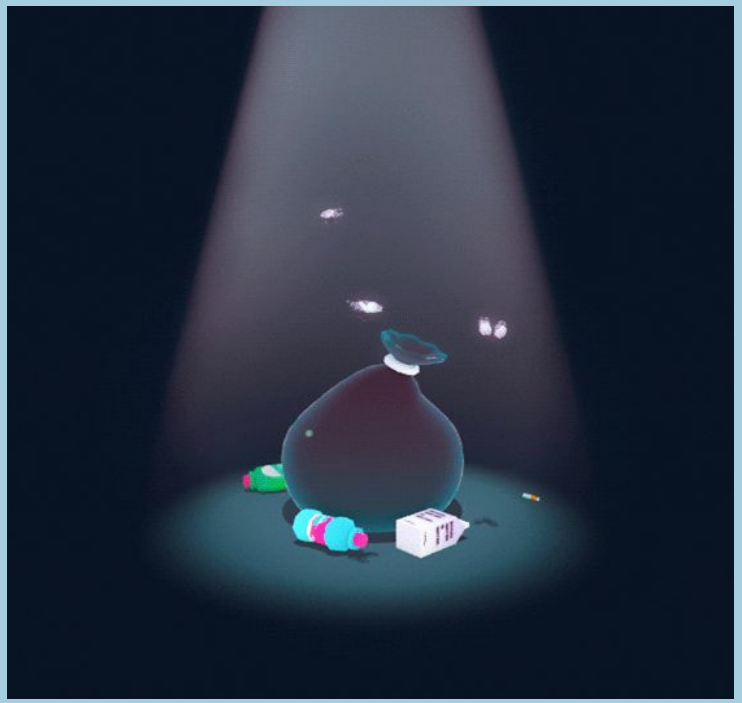
Creating a WORKING Error message

This took way more time than I would care to admit. Either it wouldn't let me do a simple backspace for when there was a typo, or it would count the non answer as an answer and just give a point based on what you previously answered, or it would show the error message but wouldn't let you type again until it was cleared

My Gif Stint

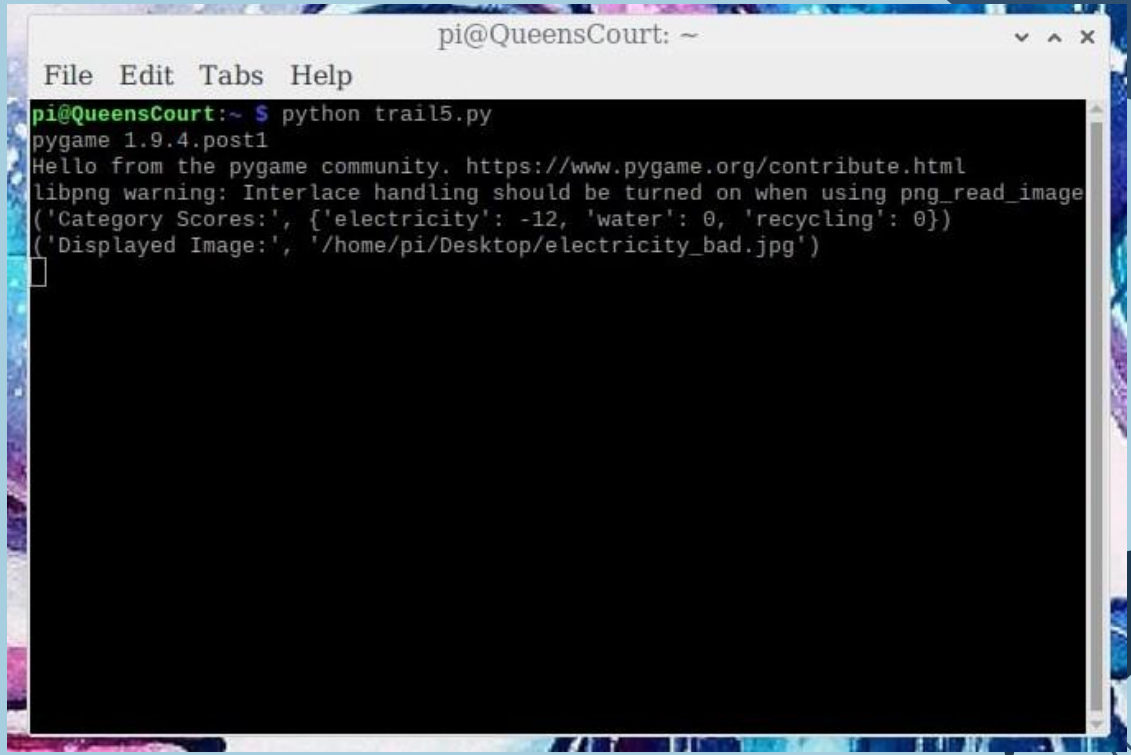
It would not, allow me to use gif, I tried turning it into a movie and having it display it frame by frame, but something about it wouldn't even read the file and just leave the screen blank





Oh no...

It would not actually distribute the points evenly. It was just categorizing everything as electricity...



So that it ACTUALLY counts points

Old

```
def calculate_score(responses):
    category_scores = {"electricity": 0, "water": 0, "recycling": 0}
    for i, response in enumerate(responses):
        question = questions[i]
        category = question["category"]
        if response.lower() == "yes":
            if question["positive"] == "good":
                category_scores[category] += question["weight"]
            elif question["positive"] == "bad":
                category_scores[category] -= question["weight"]
        elif response.lower() == "no":
            if question["negative"] == "good":
                category_scores[category] += question["weight"]
            elif question["negative"] == "bad":
                category_scores[category] -= question["weight"]
    return category_scores
```

New

```
# Function to calculate category scores
def category_score(responses, questions):
    category_scores = {"electricity": 0, "water": 0, "recycling": 0}
    for i, response in enumerate(responses):
        question = questions[i]
        category = question["category"]
        if response.lower() == "yes":
            if "positive" in question:
                category_scores[category] -= question["weight"]
            elif "negative" in question:
                category_scores[category] += question["weight"]
        elif response.lower() == "no":
            if "negative" in question:
                category_scores[category] += question["weight"]
            elif "positive" in question:
                category_scores[category] -= question["weight"]
    return category_scores
```

Explaining the reasoning behind certain decisions

Why a terrarium?

I choose a terrarium because it gives the users more control over the subject. The User currently struggling with feeling a lack of control over their life and the world around them, so the fact that this is a terrarium is supposed to help give them some of that control back. The terrarium is also made to look like an area or scene that the user can actually relate to. Like a local park or forest. This is so that the user can for a connection and potentially be more motivated to save that area.

Why only 3 categories?

Global warming is a big topic with many different factors contributing to it and many visual indicators of its presence. I summarized it to what I believe is the top 3 main factors but of course that is just my personal opinion. It is also the main 3 that a user can actually have an impact on and change in their lives.

Why ask questions?

The original idea was to not ask any questions and gather all the information from online sources like the government of canada or BC Hydro. However, not all the information I wanted to include was documented by the government, in particular the amount of 1 time use plastics being thrown away. That's when the idea of conduction a survey came in, and it grew because the act of completing a survey and seeing right away the changes it creates in the terrarium can give a sense of hope or motivate the users to take better care of their actions. It again helps the users with any potential struggles they may be facing mentally.

Review of Project Plan

I have 4 images

I have an image for each category and an image for someone who is taking care of the planet to the best of their ability

Read and display

I am able to read a txt file and display an image accordingly (at least that was my goal at first)

LED Light

I attempted to get a flashlight connected and have shown evidence above of my trails, however, this was the one thing I was not able to finish in my project plan

6 Figma Screens

I was able to make 6 screens for my figma prototype, to ask questions and display a timer before being able to take the quiz again. It also gives advice on how to improve your habits for the next time you take the quiz.

Project Plan

What will need to be done for prototype

- At least 3 images for projector to display
- Raspberry Pi is able to read information on it and display correct images on projector
- Raspberry Pi able to turn LED light off and on based on information received from it
- 3-5 Figma screens for accompanying app

The Code

First in pictures, then in words

pi@QueensCourt: ~

File Edit Tabs Help

GNU nano 3.2 trail8.py

```
import os
import pygame
import sys
import random

# Initialize pygame
pygame.init()

# Set up the display in fullscreen mode
screen = pygame.display.set_mode((0, 0), pygame.FULLSCREEN)
pygame.mouse.set_visible(False)  # Hide the mouse cursor

# Get the dimensions of the fullscreen display
screen_width, screen_height = pygame.display.get_surface().get_size()

# Create a Clock object to control the frame rate
clock = pygame.time.Clock()


# Define questions with their weights and scoring systems
questions = [
    {"category": "electricity", "question": "Do you leave the lights on when leaving a room?", "weight": 2, "positive": "bad", "negative": "good"},
    {"category": "electricity", "question": "Do you use energy-saving appliances?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you unplug electronics when they are not in use?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you use natural light during the day instead of turning on lights?", "weight": 1, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you turn off power strips when not in use?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you use a programmable thermostat to regulate energy usage?", "weight": 4, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you rely on natural ventilation instead of air conditioning whenever possible?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you take shorter showers to save water?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you leave the faucet running unnecessarily?", "weight": 3, "positive": "bad", "negative": "good"},
    {"category": "water", "question": "Do you fix leaky faucets promptly to conserve water?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you use a dishwasher instead of washing dishes by hand?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you collect rainwater for outdoor use?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you use a low-flow showerhead to conserve water?", "weight": 4, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you wash full loads of laundry instead of partial loads?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you recycle plastic bottles regularly?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you throw recyclable items in the trash?", "weight": 3, "positive": "bad", "negative": "good"},
    {"category": "recycling", "question": "Do you compost organic waste instead of throwing it away?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you separate recyclables from regular trash?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you participate in community recycling programs?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you purchase products made from recycled materials?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you educate others about the importance of recycling?", "weight": 5, "positive": "good", "negative": "bad"}
]

# Grabbing a random sample of 5 questions
random_questions = random.sample(questions, 5)

# Function to ask a question
def ask_question(question):
    font = pygame.font.Font(None, 36)
    prompt_text = font.render(question, True, (255, 255, 255))
    screen.blit(prompt_text, (10, 10))

# Function to display an error message
def display_error_message(message):
    font = pygame.font.Font(None, 24)
    error_text = font.render(message, True, (255, 0, 0))
    screen.blit(error_text, (10, 90))
```

```python
def display_image(image_path):
    try:
        image = pygame.image.load(image_path)
        image_width, image_height = image.get_size()

        # Calculate
        screen_width, screen_height = screen.get_size()
        ratio = min(screen_width / image_width, screen_height / image_height)
        scaled_width = int(image_width * ratio)
        scaled_height = int(image_height * ratio)

        # Position !
        x = (screen_width - scaled_width) // 2
        y = (screen_height - scaled_height) // 2

        # Scale the image
        scaled_image = pygame.transform.smoothscale(image, (scaled_width, scaled_height))

        # Fill the screen
        screen.fill((0, 0, 0))

        # Blit the scaled image onto the screen
        screen.blit(scaled_image, (x, y))
        pygame.display.flip()
    except FileNotFoundError:
        print("Error: File not found at", image_path)
    except pygame.error as e:
        print("Error loading image:", e)

# Function to calculate category scores
def category_score(responses, questions):
    category_scores = {"electricity": 0, "water": 0, "recycling": 0}
    for i, response in enumerate(responses):
        question = questions[i]
        category = question["category"]
        if response.lower() == "yes":
            if "positive" in question:
                category_scores[category] -= question["weight"]
            elif "negative" in question:
                category_scores[category] += question["weight"]
        elif response.lower() == "no":
            if "negative" in question:
                category_scores[category] += question["weight"]
            elif "positive" in question:
                category_scores[category] -= question["weight"]
    return category_scores

# Get the path to the user's desktop
desktop_path = os.path.join(os.path.expanduser("~"), "Desktop")

# Main Loop
running = True
while running:
    responses = []

    # Ask questions
    for question_data in random_questions:
        input_finished = False
        invalid_response = False
        question = question_data["question"]
        ask_question(question)

        while not input_finished:
            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    running = False
                    input_finished = True
                elif event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_RETURN or event.key == pygame.K_KP_ENTER:
                        if user_input.lower() in ["yes", "no"]:
                            input_finished = True
                        else:
                            invalid_response = True
                            user_input = ""
                    elif event.key == pygame.K_BACKSPACE:
                        user_input = user_input[:-1]
                    else:
                        user_input += event.unicode

            screen.fill((0, 0, 0))
            ask_question(question)
            font = pygame.font.Font(None, 36)
            input_text = font.render(user_input, True, (255, 255, 255))
            screen.blit(input_text, (10, 50))

            if invalid_response:
                display_error_message("Invalid response. Please type 'yes' or 'no'.")

            pygame.display.flip()
            clock.tick(30)

        responses.append(user_input)

    # Calculate category scores
    category_scores = category_score(responses, random_questions)

    # Find the highest score
    highest_score = max(category_scores.values())

    # Find the category with the highest score
    highest_categories = [category for category, score in category_scores.items() if score == highest_score]



    # Exiting the loop
    running = False

    # Get worst category
    worst_category = max(category_scores, key=category_scores.get)

    # Check if person is labeled as a "good person"
    good_person = all(score <= 3 for score in category_scores.values())

    # Display
    if good_person:
        image_path = os.path.join(desktop_path, "good_person.jpg")
    else:
        # Display image based on worst category
        image_filename = worst_category + "_bad.jpg"
        image_path = os.path.join(desktop_path, image_filename)



    display_image(image_path)

    print("Category Scores:", category_scores)
    print("Displayed Image:", image_path)

    pygame.time.delay(5000)  # Display the image for 5 seconds

pygame.quit()
sys.exit()
```

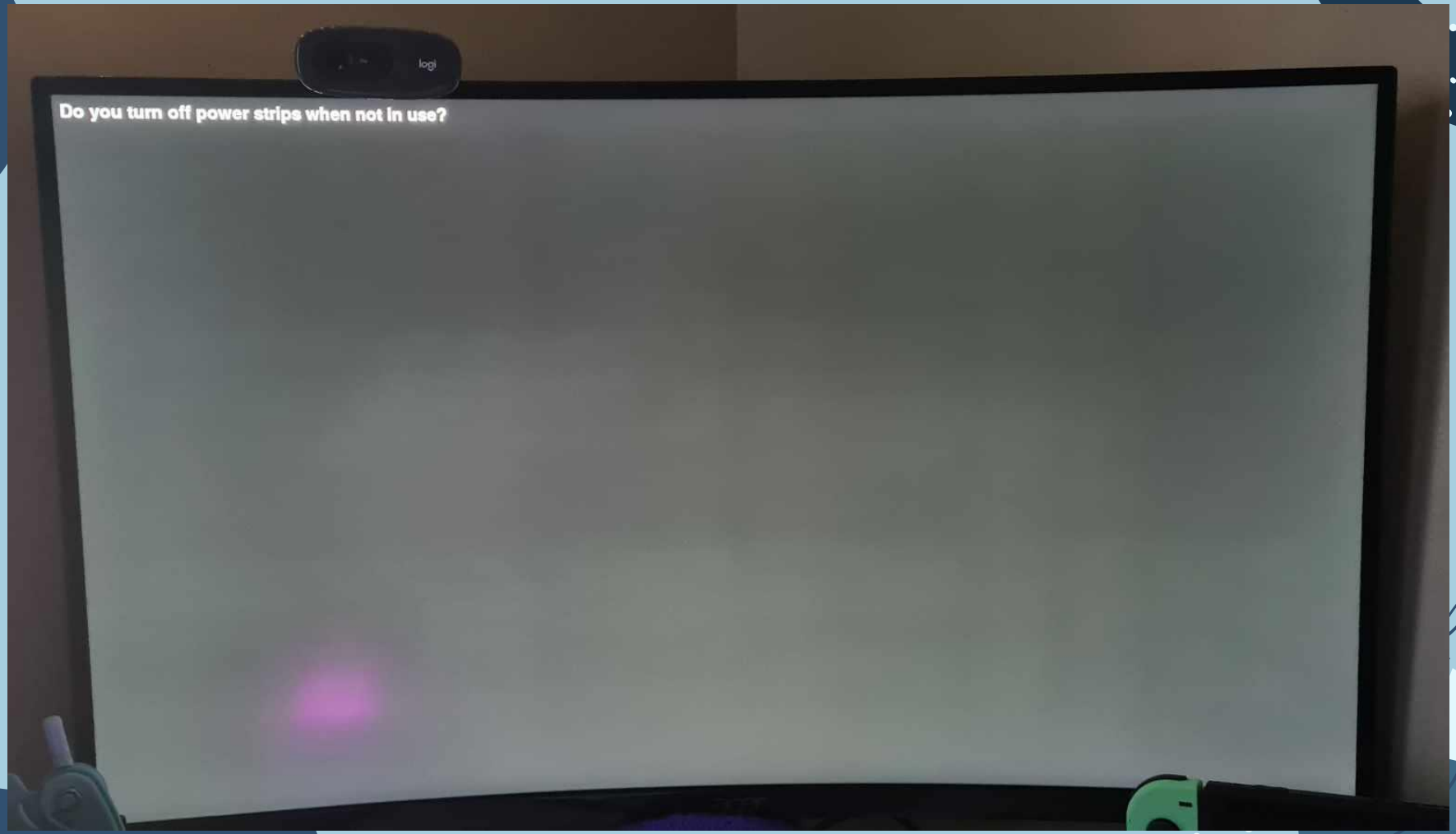
The written code

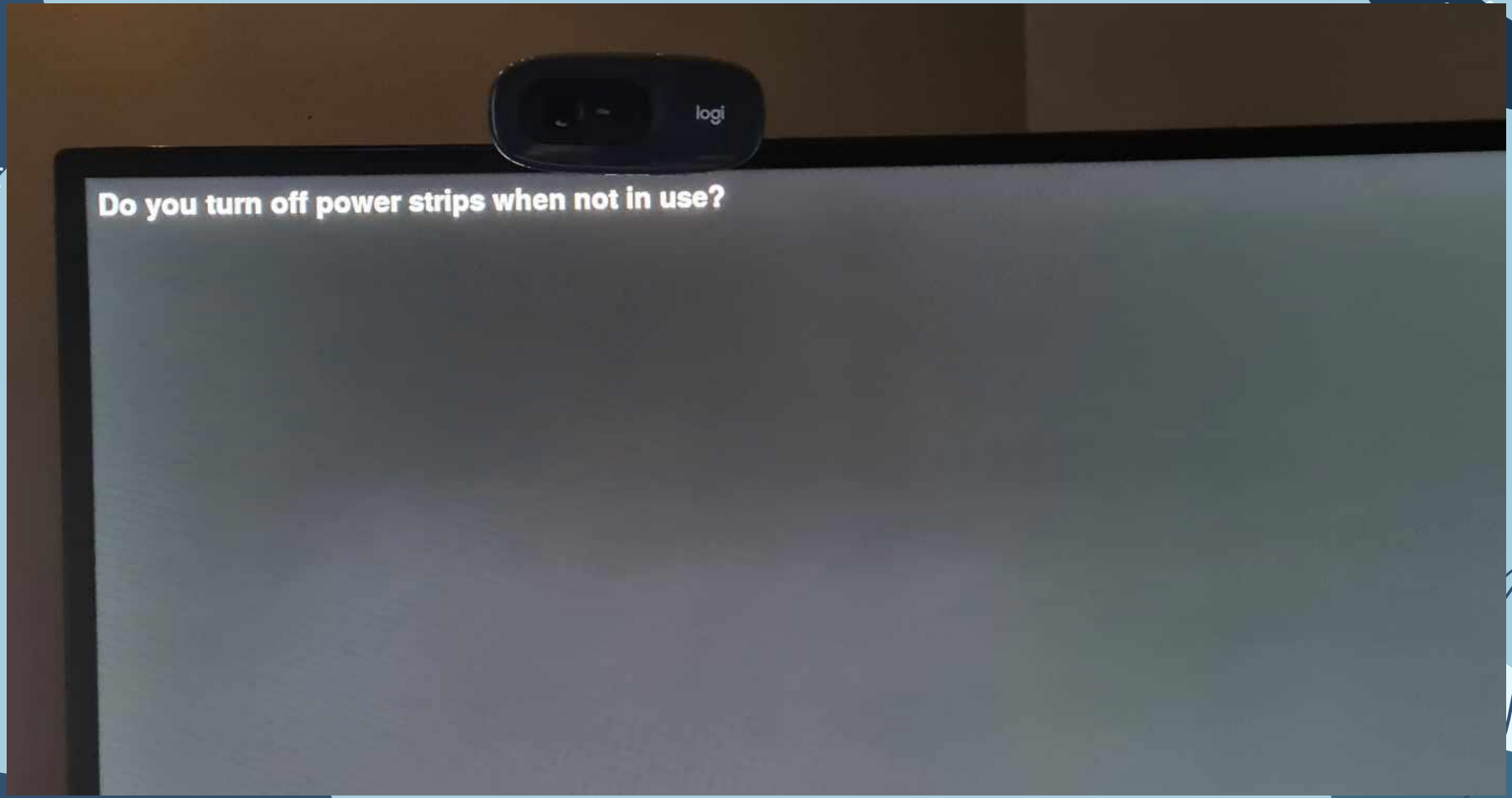
```
import os
import pygame
import sys
import random

# Initialize pygame
pygame.init()

# Set up the display in fullscreen mode
screen = pygame.display.set_mode((0, 0), pygame.FULLSCREEN)
pygame.mouse.set_visible(False)  # Hide the mouse cursor

# Get dimensions
screen_width, screen_height = pygame.display.get_surface().get_size()

# Create a Clock object to control the frame rate
clock = pygame.time.Clock()


# Define questions with their weights and scoring systems
questions = [
    {"category": "electricity", "question": "Do you leave the lights on when leaving a room?", "weight": 2, "positive": "bad", "negative": "good"},
    {"category": "electricity", "question": "Do you use energy-saving appliances?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you unplug electronics when they are not in use?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you use natural light during the day instead of turning on lights?", "weight": 1, "positive": "good", "negative": "b$
    {"category": "electricity", "question": "Do you turn off power strips when not in use?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you use a programmable thermostat to regulate energy usage?", "weight": 4, "positive": "good", "negative": "bad"},
    {"category": "electricity", "question": "Do you rely on natural ventilation instead of air conditioning whenever possible?", "weight": 3, "positive": "good", "n$
    {"category": "water", "question": "Do you take shorter showers to save water?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you leave the faucet running unnecessarily?", "weight": 3, "positive": "bad", "negative": "good"},
    {"category": "water", "question": "Do you fix leaky faucets promptly to conserve water?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you use a dishwasher instead of washing dishes by hand?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you collect rainwater for outdoor use?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you use a low-flow showerhead to conserve water?", "weight": 4, "positive": "good", "negative": "bad"},
    {"category": "water", "question": "Do you wash full loads of laundry instead of partial loads?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you recycle plastic bottles regularly?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you throw recyclable items in the trash?", "weight": 3, "positive": "bad", "negative": "good"},
    {"category": "recycling", "question": "Do you compost organic waste instead of throwing it away?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you separate recyclables from regular trash?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you participate in community recycling programs?", "weight": 3, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you purchase products made from recycled materials?", "weight": 2, "positive": "good", "negative": "bad"},
    {"category": "recycling", "question": "Do you educate others about the importance of recycling?", "weight": 5, "positive": "good", "negative": "bad"}
]

# Grabbing a random 5 questions
random_questions = random.sample(questions, 5)

# Function ask a question
def ask_question(question):
    font = pygame.font.Font(None, 36)
    prompt_text = font.render(question, True, (255, 255, 255))
    screen.blit(prompt_text, (10, 10))

# Function to display error message
def display_error_message(message):
    font = pygame.font.Font(None, 24)
    error_text = font.render(message, True, (255, 0, 0))
    screen.blit(error_text, (10, 90))
```

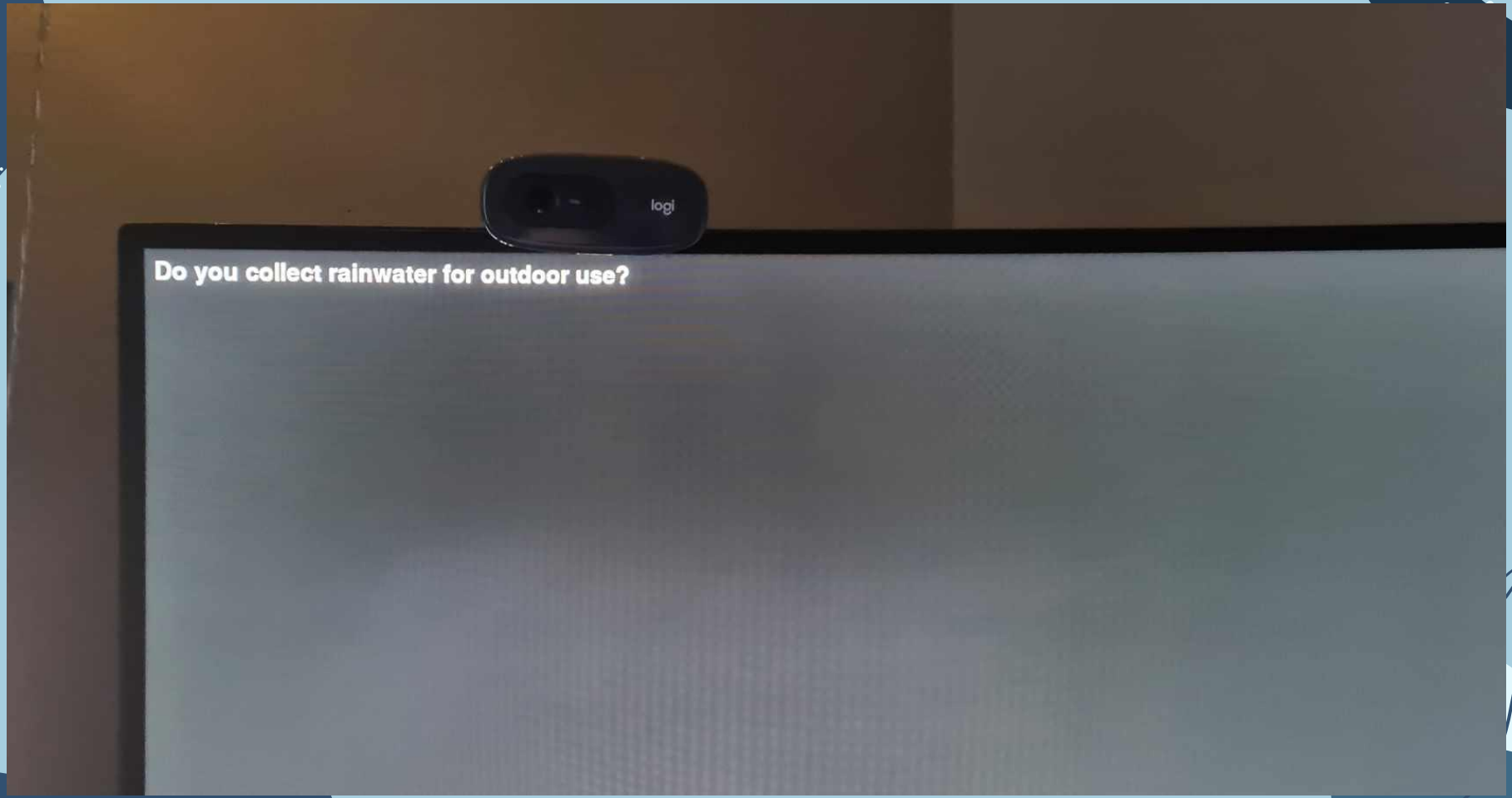
```
def display_image(image_path):
    try:
        image = pygame.image.load(image_path)
        image_width, image_height = image.get_size()

        # Calculate
        screen_width, screen_height = screen.get_size()
        ratio = min(screen_width / image_width, screen_height / image_height)
        scaled_width = int(image_width * ratio)
        scaled_height = int(image_height * ratio)

        # Position
        x = (screen_width - scaled_width) // 2
        y = (screen_height - scaled_height) // 2

        # Scale the image
        scaled_image = pygame.transform.smoothscale(image, (scaled_width, scaled_height))

        # Fill background color
        screen.fill((0, 0, 0))

        # Blit the scaled image onto the screen
        screen.blit(scaled_image, (x, y))
        pygame.display.flip()
    except FileNotFoundError:
        print("Error: File not found at", image_path)
    except pygame.error as e:
        print("Error loading image:", e)
def category_score(responses, questions):
    category_scores = {"electricity": 0, "water": 0, "recycling": 0}
    for i, response in enumerate(responses):
        question = questions[i]
        category = question["category"]
        if response.lower() == "yes":
            if "positive" in question:
                category_scores[category] -= question["weight"]
            elif "negative" in question:
                category_scores[category] += question["weight"]
        elif response.lower() == "no":
            if "negative" in question:
                category_scores[category] += question["weight"]
            elif "positive" in question:
                category_scores[category] -= question["weight"]
    return category_scores

# Get path
desktop_path = os.path.join(os.path.expanduser("~"), "Desktop")

# Main Loop
running = True
while running:
    responses = []

    # Ask questions
    for question_data in random_questions:
        input_finished = False
        invalid_response = False
        user_input = ""
```

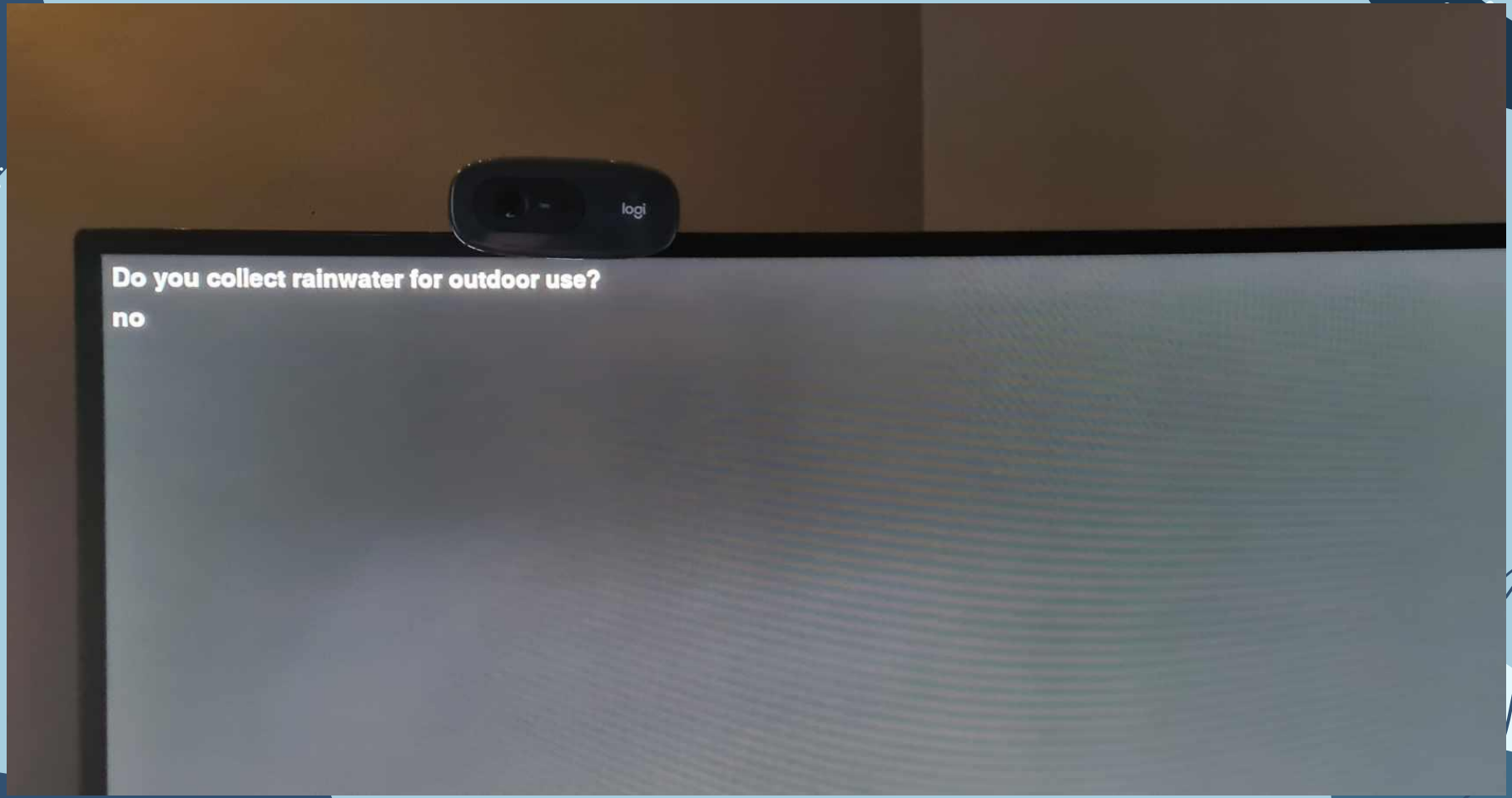
```
        question = question_data["question"]
        ask_question(question)

        while not input_finished:
            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    running = False
                    input_finished = True
                elif event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_RETURN or event.key == pygame.K_KP_ENTER:
                        if user_input.lower() in ["yes", "no"]:
                            input_finished = True
                        else:
                            invalid_response = True
                            user_input = ""
                    elif event.key == pygame.K_BACKSPACE:
                        user_input = user_input[:-1]
                    else:
                        user_input += event.unicode

            screen.fill((0, 0, 0))
            ask_question(question)
            font = pygame.font.Font(None, 36)
            input_text = font.render(user_input, True, (255, 255, 255))
            screen.blit(input_text, (10, 50))

            if invalid_response:
                display_error_message("Invalid response. Please type 'yes' or 'no'.")

            pygame.display.flip()
            clock.tick(30)

        responses.append(user_input)

    # Calculate category scores
    category_scores = category_score(responses, random_questions)

    # Find the highest score
    highest_score = max(category_scores.values())

    # Find the category with the highest score
    highest_categories = [category for category, score in category_scores.items() if score ==
highest_score]

    # Exiting the loop
    running = False  # Adjust this condition as per your requirements

    # Get worst category
    worst_category = max(category_scores, key=category_scores.get)

    # Check if person is labeled as a "good person"
    good_person = all(score <= 3 for score in category_scores.values())

    # Display image
    if good_person:
        image_path = os.path.join(desktop_path, "good_person.jpg")
    else:
        # Display image based on worst category
        image_filename = worst_category + "_bad.jpg"
        image_path = os.path.join(desktop_path, image_filename)



    display_image(image_path)

    print("Category Scores:", category_scores)
    print("Displayed Image:", image_path)

    pygame.time.delay(5000)  # Display the image for 5 seconds

pygame.quit()
sys.exit()
```

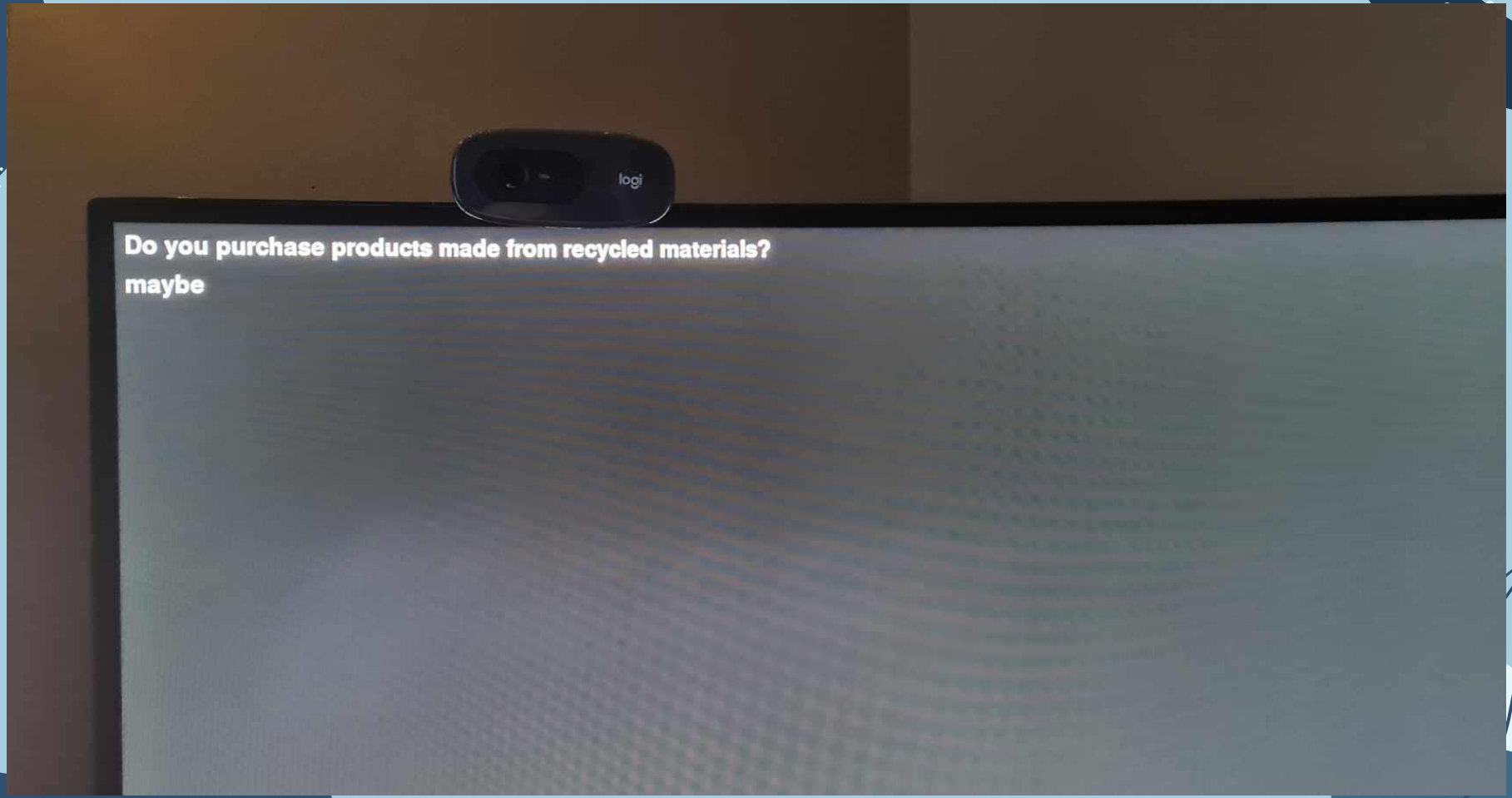
A step by step of the code in action

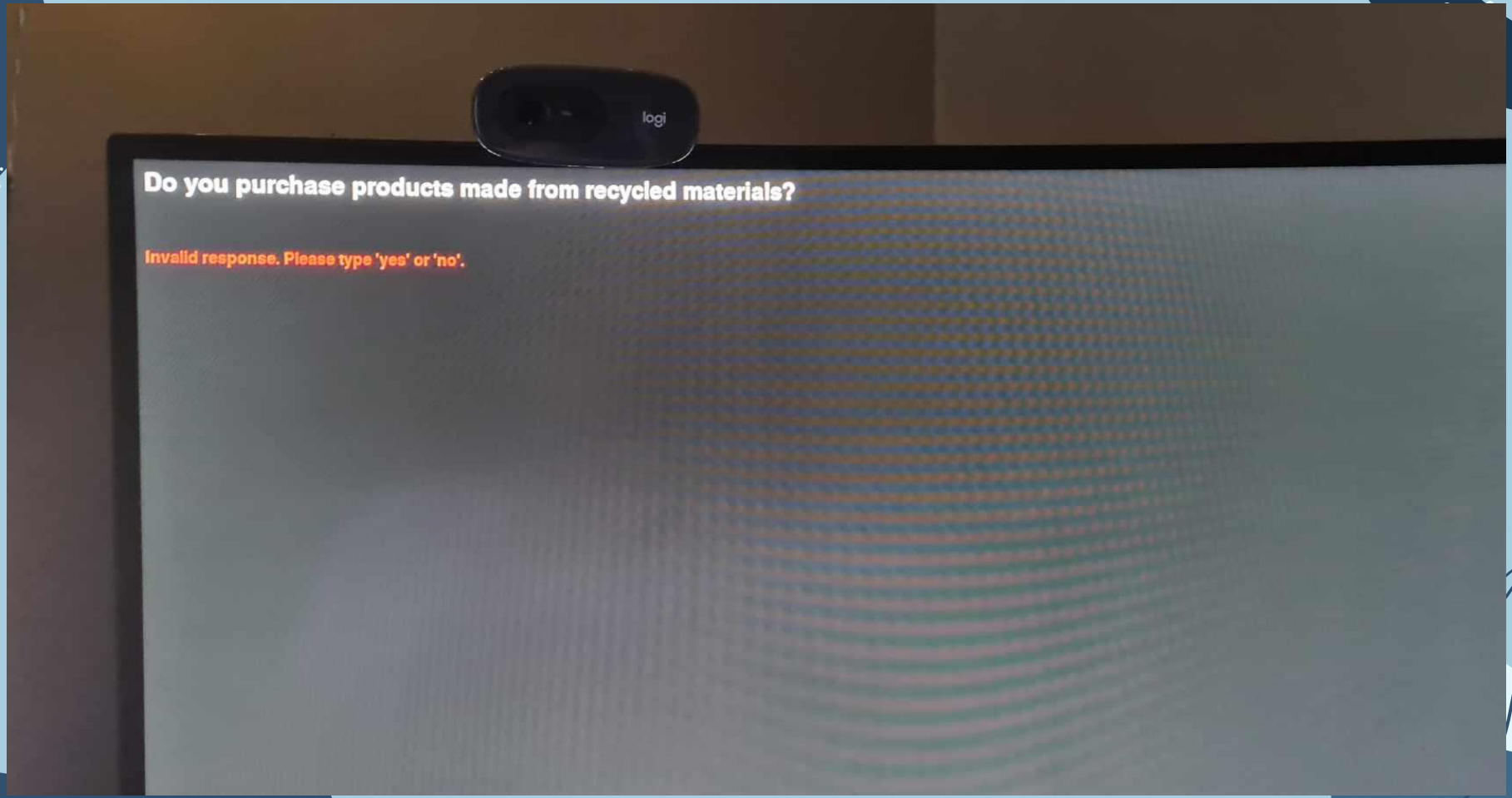

Do you turn off power strips when not in use?

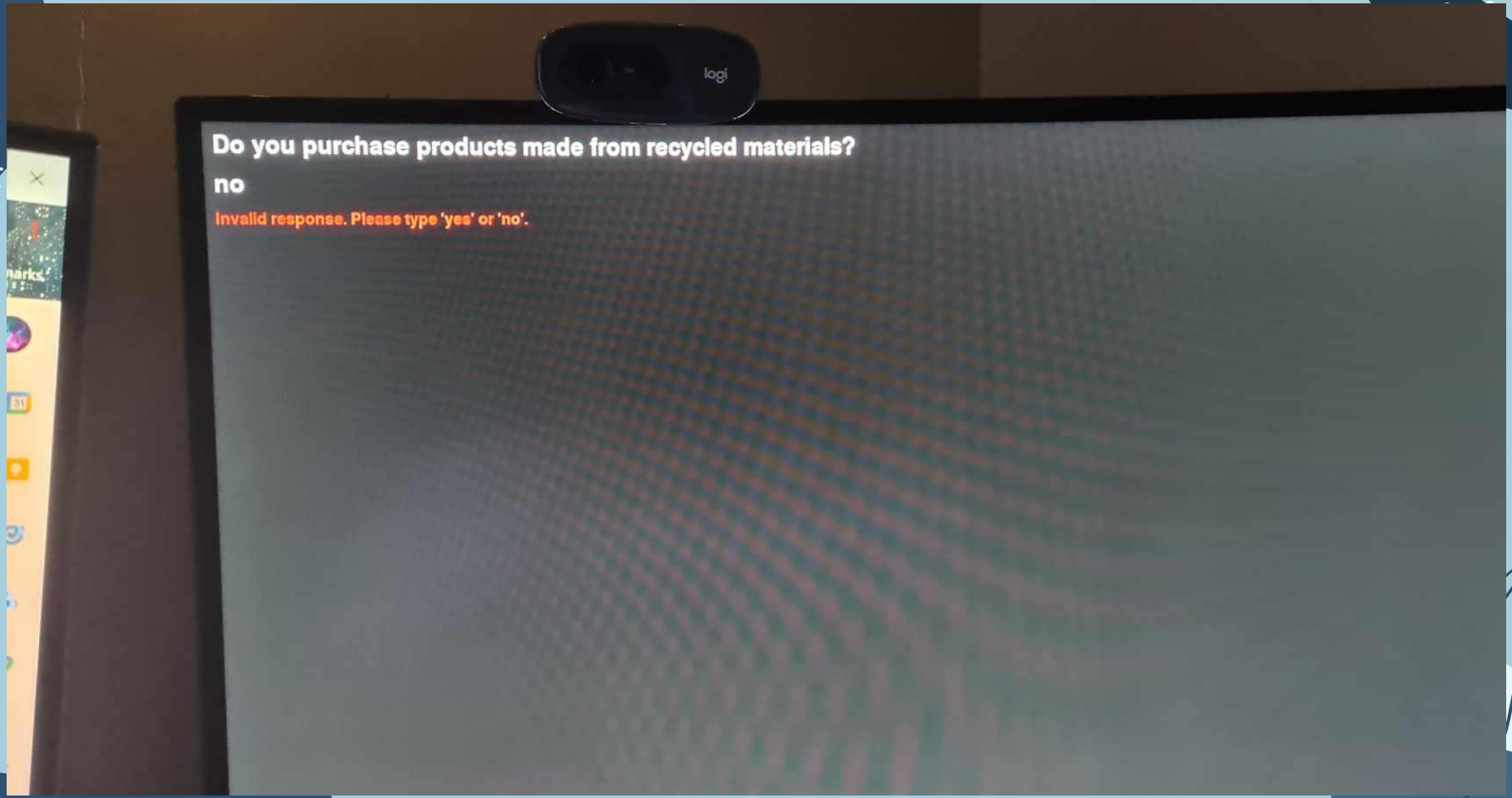

logi
Do you turn off power strips when not in use?

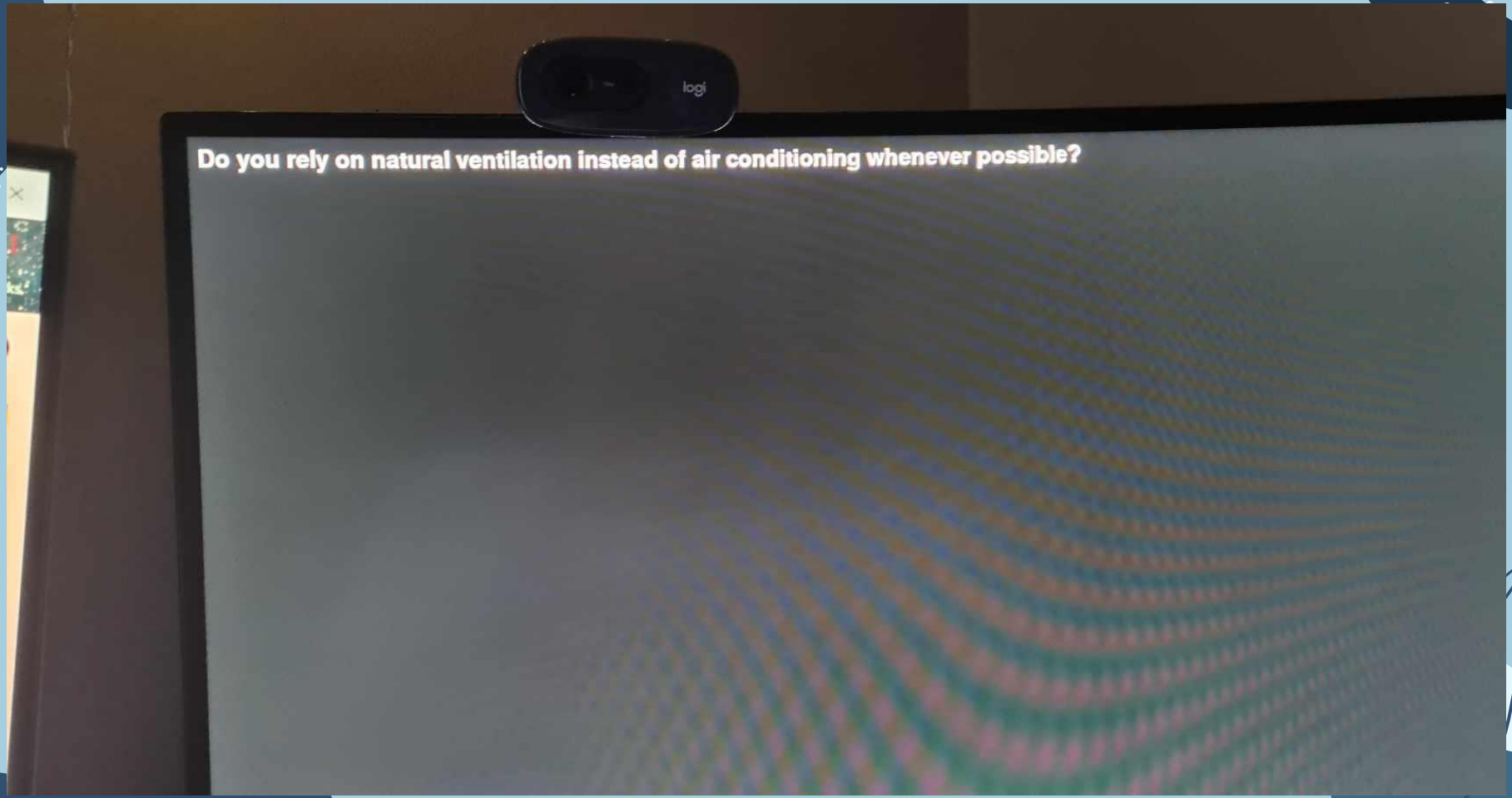

logi
Do you collect rainwater for outdoor use?

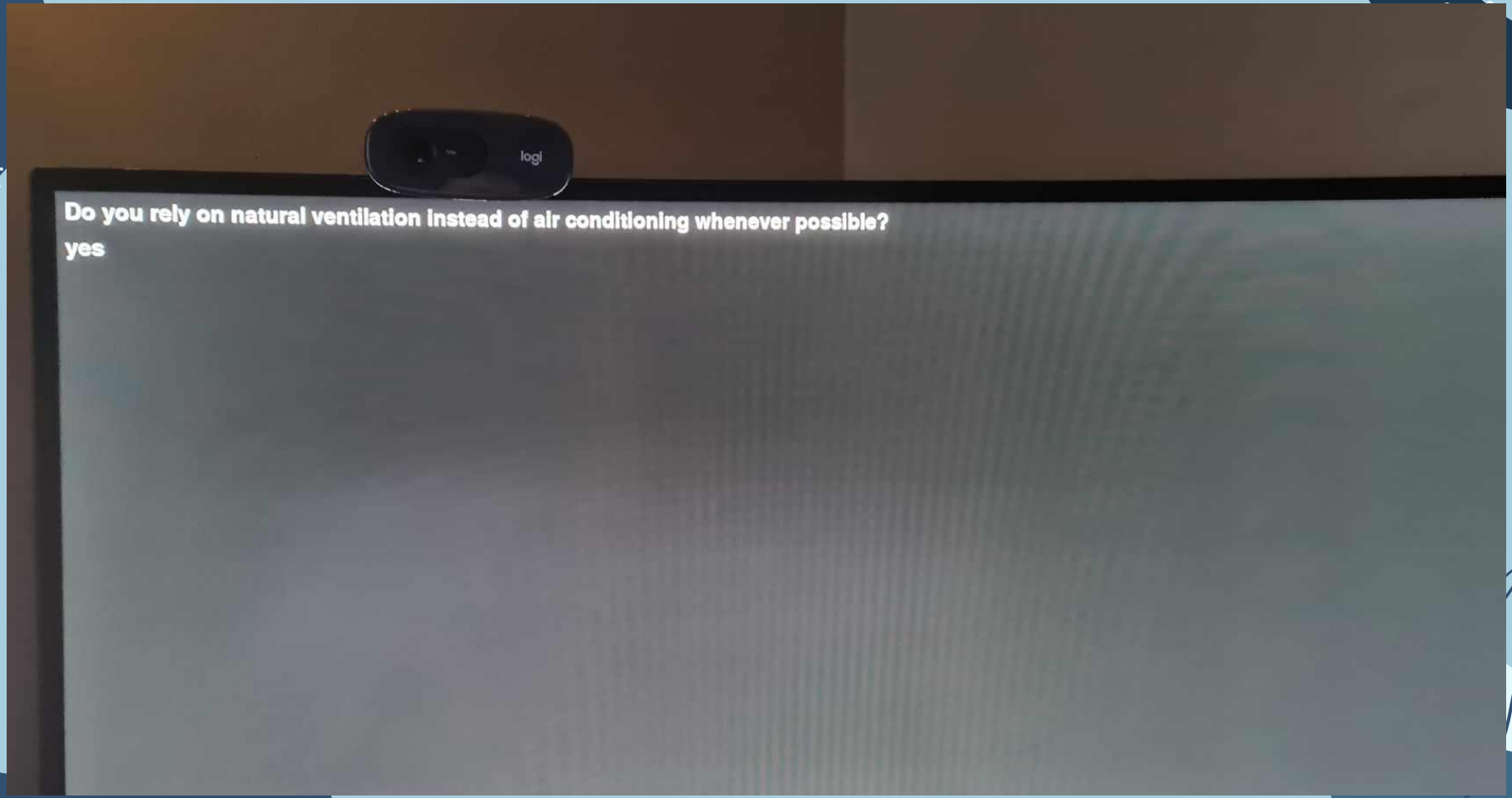

logi
Do you collect rainwater for outdoor use?
no

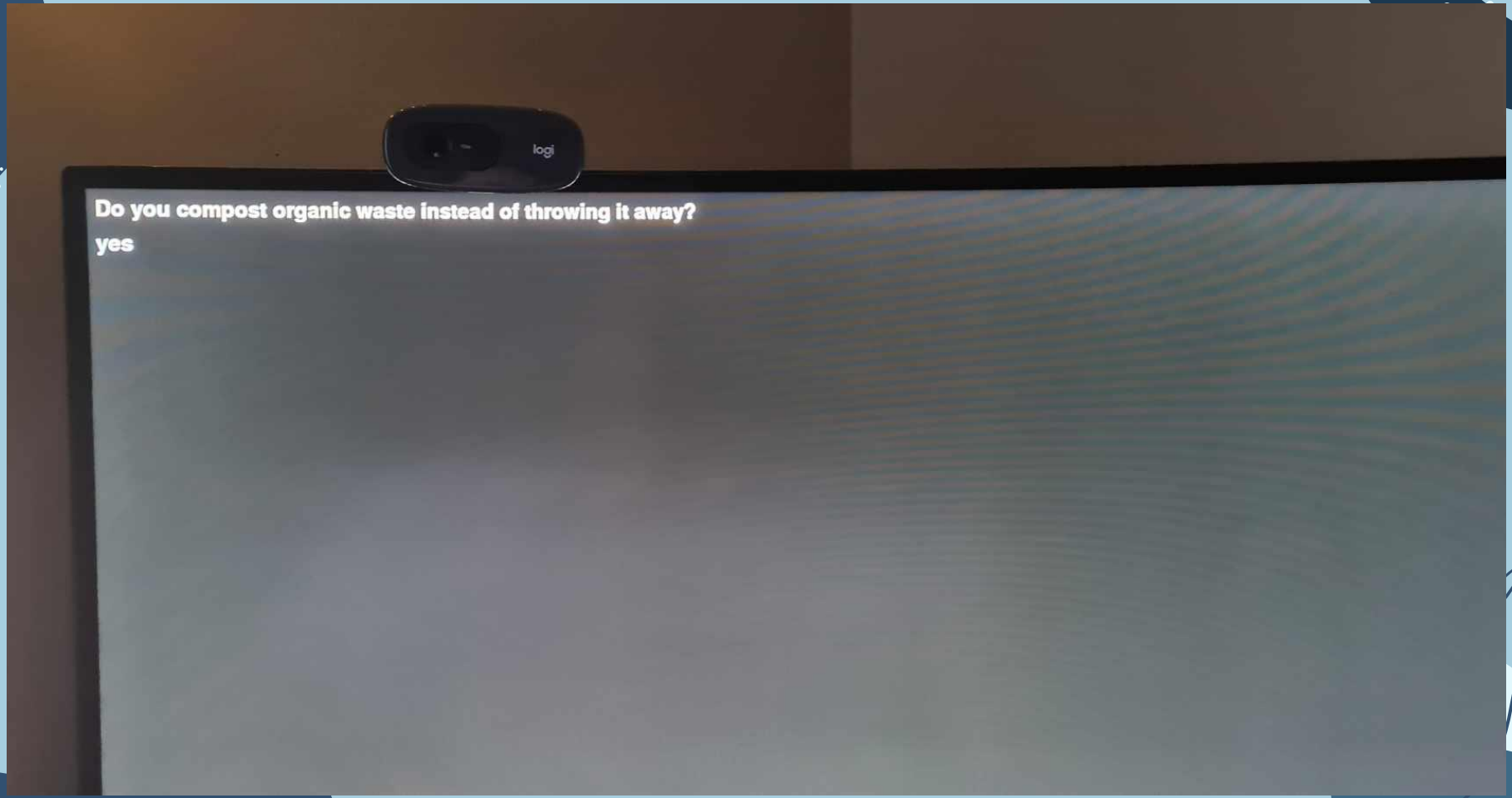

logi
Do you purchase products made from recycled materials?
maybe


logi
Do you purchase products made from recycled materials?
Invalid response. Please type 'yes' or 'no'.


logi
Do you purchase products made from recycled materials?
no
Invalid response. Please type 'yes' or 'no'.

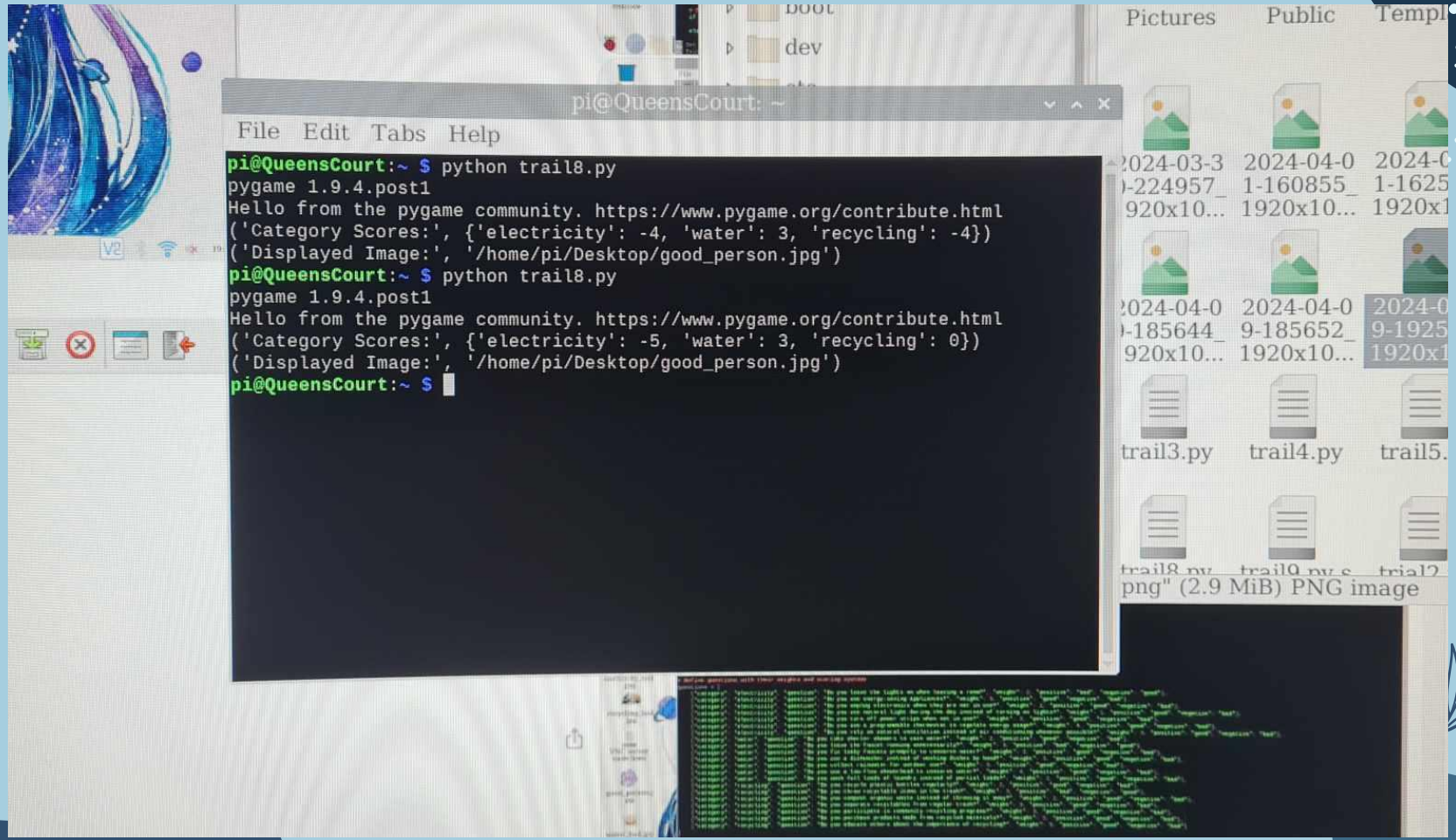

logi
Do you rely on natural ventilation instead of air conditioning whenever possible?


logi
Do you rely on natural ventilation instead of air conditioning whenever possible?
yes


logi
Do you compost organic waste instead of throwing it away?
yes

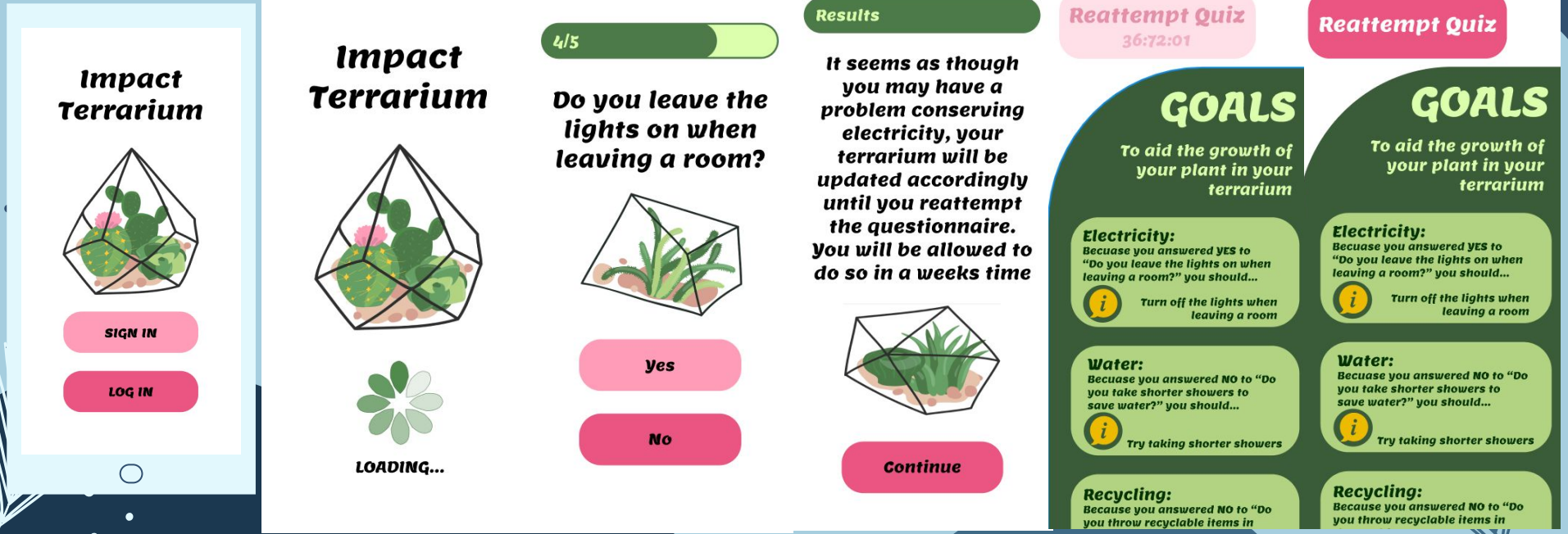

All Bookmarks

The 4 Categories


pi@QueensCourt: ~
File Edit Tabs Help
pi@QueensCourt:~ $ python trail8.py
pygame 1.9.4.post1
Hello from the pygame community. https://www.pygame.org/contribute.html
('Category Scores:', {'electricity': -4, 'water': 3, 'recycling': -4})
('Displayed Image:', '/home/pi/Desktop/good_person.jpg')
pi@QueensCourt:~ $ python trail8.py
pygame 1.9.4.post1
Hello from the pygame community. https://www.pygame.org/contribute.html
('Category Scores:', {'electricity': -5, 'water': 3, 'recycling': 0})
('Displayed Image:', '/home/pi/Desktop/good_person.jpg')
pi@QueensCourt:~ $
Pictures
Public
dev
trail3.py
trail4.py
png" (2.9 MiB) PNG image

I did not get a chance to set up my project for the exhibit, but this is what my terrarium should look like with the projector inside and on

Figma


Impact Terrarium
SIGN IN
LOG IN
Impact Terrarium
LOADING...
4/5
Do you leave the lights on when leaving a room?
Yes
No
Results
It seems as though you may have a problem conserving electricity, your terrarium will be updated accordingly until you reattempt the questionnaire. You will be allowed to do so in a weeks time
Continue
Reattempt Quiz
36:72:01
Reattempt Quiz
GOALS
To aid the growth of your plant in your terrarium
Electricity:
Because you answered YES to "Do you leave the lights on when leaving a room?" you should...
Turn off the lights when leaving a room
Water:
Because you answered NO to "Do you take shorter showers to save water?" you should...
Try taking shorter showers
Recycling:
Because you answered NO to "Do you throw recyclable items in
GOALS
To aid the growth of your plant in your terrarium
Electricity:
Because you answered YES to "Do you leave the lights on when leaving a room?" you should...
Turn off the lights when leaving a room
Water:
Because you answered NO to "Do you take shorter showers to save water?" you should...
Try taking shorter showers
Recycling:
Because you answered NO to "Do you throw recyclable items in

Hopes for the future

VR/AR projections

To either create an AR setting where a mini you or person/animal of your choice will be displayed living in the terrarium. Or a VR scenario where what is currently happening in the terrarium is projected around you in a VR way

A smaller projector and bigger terrarium

To big and bulky, would also mean I would have space to put in the raspberry pi as well and a portable battery potentially. Also more space to easily replicate the park

More than just images

More than just showing pictures, if the terrarium could somehow change its environment, not only to better deter or aid the growth of the plant inside, but also to more accurately show the effects of the users actions

A working light bulb

The whole point of the project was to show that your actions affect the plants in the terrarium, but all it really does is display an image inside of it, not really affecting it

Conclusions

Honestly, I am not 100% satisfied with how my final project turned out. I am not too sure as to why because, except for the LED light flashlight, I had met every other expectation for this project. I love my idea and believe that the problem it is tackling is impacting the world as a whole. I believe that my terrarium will address this problem space and will create positive change. The code also turned out much better than I expected, considering that it was my first time coding in a terminal in Raspberry Pi and I had a short window to learn the language and grow accustomed to the layout and style of coding on Raspberry Pi. I hadn't even considered how fleshed out the questions would need to be or how integral to the project they would become in the beginning but I am proud of the questions asked on how it calculates the score based on those questions. I guess I am proud of this project then. I am nevertheless disappointed in the final for multiple reasons. I was unable to use a real plant due to the base of the pot in question/ I was unable to find a proper terrarium plant in which that would not have mattered. I was also slightly disappointed in how little the results of the questionnaire affected the terrarium. Other than displaying the image (and if it worked, affecting the light) the results of the questionnaire did not change much. I don't think I would have been able to rectify this feeling of disappointment in the term of this class. Hopefully after working on this terrarium outside of class, and by fully completing my vision, I will be proud to show this project to others without feeling as if I am showing a half finished project.

Thanks!

CREDITS: This presentation template was created by **Slidesgo** , including icons by **Flaticon** , infographics & images by **Freepik**