Mood Picture

Project Report

Master in Creative Computing & A.I.

Project I

Author: Alan Nunes Caetano.

Project available in: <https://github.com/alannunescaetano/MoodPicture>

# Introduction and goals

The "Guideline for community noise" from the World Health Organization describes community noise as noise emitted from all sources except industrial workplaces, including outdoor sources, like airports and road traffic, as indoor sources, office machines, home appliances and ventilation systems.

This guideline document also mentions a list of health and cognitive problems that could be caused when this environmental noise reach some patterns of pressure level, frequency and continuity/intermittency.

Said that, the main goal of this project is to make people reflect about the mood of their environment perceived through sounds and how this environmental noise can affect them on the long term.

Although it was used a theoretical basis to find the sound patterns, it’s important to mention that this system (on the present version) only aims to explore the feelings of the user looking at the generative art to reflect about how the environment affects its mood, without any intention to diagnose or infer health effects.

# Functionalities

The main functionalities of this solution will be:

### Gather data that could represent the user’s mood:

The first type of data to be explored will be the environment sound, captured as amplitude of sound through time.

### Create a meaningful output for the user’s reflection:

Use of generative art techniques to create an output based on the input data that represents the environment mood on that period of time.

### Compare input data over time searching for patterns:

Use A.I. to search for patterns on the input data to try to determine the importance of each pattern found.

### Show a good mood streak:

Streak indication to the user based on the percentage of periods classified as stressful.

# Units Application on the Project

### **Creative programming:**

**Use of p5.js library to create a generative output on a web page.**

### Physical computing and IoT:

Implementation of Arduino project to gather environment data with sensors.

### Applied A.I.:

Use of algorithms to find correlations between the sensor’s patterns.

### Gamification in I.S.:

Creation of a streak to incentive changes on the environment and behavior of people inside the environment.

# Use Cases

## 4.1 Initial Use Case

The initial idea for a use case is:

**Step 1:** The user starts the sensors recording when is beginning some demanding activity (work, meetings, academic tasks);

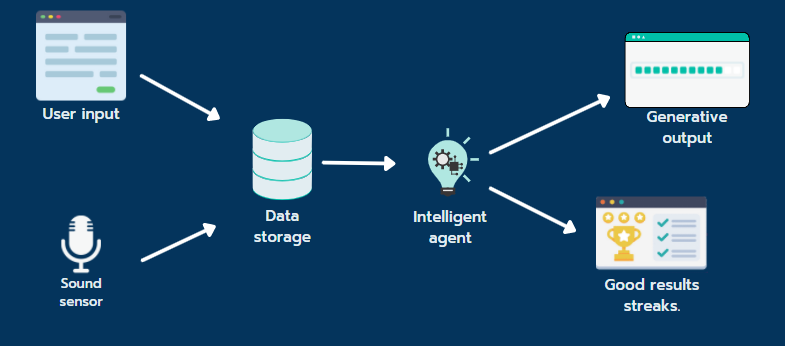
**Step 2:** After some time, the user stops the sensor;

**Step 3:** The user inform how are he/she feeling (angry/calm);

**Step 4:** The application server processes the input data and make the generative output available to the user.

# Implementation

For the implementation of this project, there was a preference for JavaScript libraries because of easier integration and quick prototyping. For the same reason, it was decided for a web UI.



## 5.1 Sound sensor

To capture the sound, it was used a Loudness Sensor LM2904 and to make the wireless communication, a ESP32 module with Wi-Fi.

The implementation made on Arduino consisted in grouping the amplitude readings in periods of time and send to a server on an HTTP POST in JSON format.

## 5.2 Artificial Intelligence

The smart agent was implemented using the Tau-Prolog library.

The rules to classify stressful periods are:

* Sound intermittency (Max / Average amplitude);
* Calibration of sensibility with the user input;
* Residual stress calculated with 5 last periods:
  + on the 3 consecutive last periods being stressful;
  + on the 5 last periods being intermittent;
  + on the 5 last periods having 4 stress periods.

## 5.3 Creative Programming

The generative art is a particle system with some specific behaviors to look like an aura and to make the impression of loudness and stress.

To achieve that, it was implemented the following behaviors on the particles:

* The particles start bigger and shrink getting away from the start point;
* The temperature of the color is raised when receives a stress indicator and cool down with time;
* The particle dispersion speed is raised when receives a high amplitude indicator and slows down with time;
* Blend mode is changed to LIGHTEST to create interaction between the colors to remember a flame;
* Physics concepts of acceleration and friction were implemented in order to mimic the flame losing force when getting away from the source.

The generative art is shown on a web page and the data is recovered on a GET request from the server.

## 5.4 Gamification

It was decided that competitions mechanisms were not suitable for this kind of goal, so it was preferred to implement an individual streak indicator for good percentage of stress.

For this prototype, it’s being used reference values above 40% of stressful events for not desirable days and below that for desirable days. This should be reviewed by an expert.

The general idea of this module is to incentive the improving and maintenance of good behavior and improvements on the environment.

## 5.5 Integration

For the integration, it was implemented a node.js server with the following methods:

* POST: to the IoT module be able to send the readings in JSON format;
* GET: to the gamification module retrieve the sessions data;
* GET/session\_id&user\_perception: to the generative art retrieve the readings and the smart agent results.

The data is stored into a SQLite database within 2 tables: AMPLITUDE\_READINGS and CAPTURE\_SESSIONS.

The smart agent is called using a node.js implementation of the Tau-Prolog library.

# Project folder structure

The CreativeAnimation folder compose the FRONT-END part of the application. The index.html file runs the generative art, while the streak.html is the implementation of the gamification module. This pages need the node.js server running localhost to run.

The IoT folder contains the AmplitudeReader.ino, which is the implementation of the sensor and ESP32 board.

The Server/smart\_agent have the implementation of the smart agent using Tau-Prolog.

The server can be started using the command *node server.js*.

# Conclusions and future work

The choice to use node.js and javascript libraries showed to be advantages on the implementation of this structure, both in terms of speed of prototyping and the easy on integrating them with libraries and JSON data.

This project shows that generative art can be used for more than educational purposes and aesthetics effects, being used in this case to represent the sensor data in a way that could affect the viewer sentiments and cause reflections.

For the future work, a suggestion could be incrementing the smart agent with more types of data: physiological sensors and health APIs data, to determine with more accuracy the patterns of sounds that can cause more stress.

# References

World Health Organization. (1999, April). "Guideline for community noise". Retrieved from <https://www.who.int/docstore/peh/noise/Comnoise-1.pdf>.