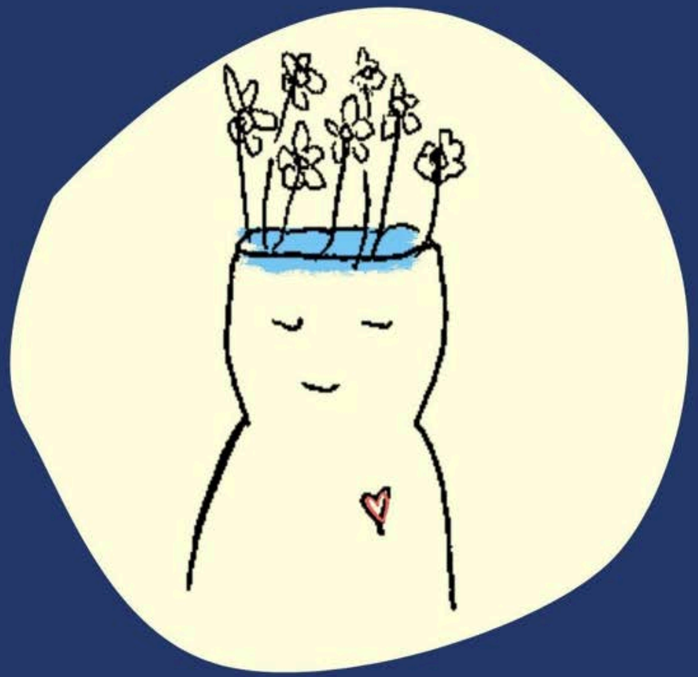
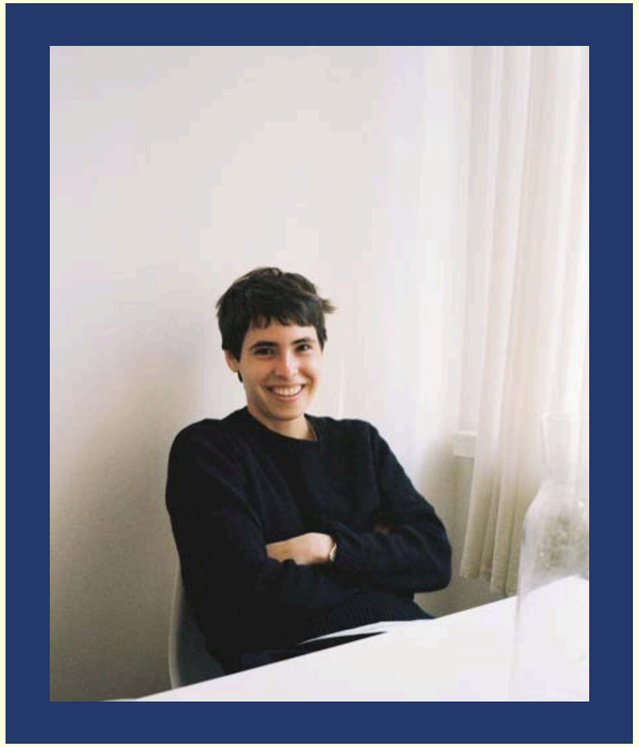


Hello, I'm B!

I am a 23-year-old Italian student continuing my studies in Data Science and Artificial Intelligence at TU Delft. I hold a Bachelor's degree from Maastricht University and have gained industry experience as a Junior Software Engineer. Passionate about problem-solving and innovation, I am a curious, determined, and open-minded individual. My main interests lie in sustainability, healthcare, and education systems, where I strive to apply technology for positive societal impact.



Tech Stack

JAVA · PYTHON · R · LATEX · ASSEMBLY SCRIPT · POSTMAN · JAVAFX · QT · STREAMLIT · MYSQL · OPENCV · ELECTRON · SQLITE · KERAS · SCIPY · SPRING · PANDAS · TENSORFLOW · MATPLOTLIB · NUMPY · PYTORCH · SCIKIT-LEARN

Soft Tools

Intellij, PyCharm, Eclipse, Visual Studio Code, MATLAB, RStudio, GitHub, GitLab

Interests

Video Editing, Football, Music, Cooking

Languages

Italian, English, Spanish

Contact

bianca.cdchiusano@gmail.com

+39 328 385 0381

https://github.com/bchiusano

https://www.linkedin.com/in/bianca-caissotti-di-chiusano/

Group Projects

Graph Colouring **01**

Dice Chess **02**

Gesture Detection **03**

NetNote **04**

Individual Projects

05 B -Present

06 Freedom of Expression

07 Your Mood, Your Music

08 AI Tool

01

Maastricht University

Graph Colouring

Project Description

The aim of this project is to improve the calculation speed of the chromatic number of a graph by applying a **multi-algorithmic approach**

Definitions

Graph or Vertex colouring is the assignment of labels, or colours, to parts of a graph subject to bound restrictions. Finding the chromatic number of a graph is an **NP-complete** problem. For that reason this research focuses on finding the smallest interval of the chromatic number with **lower** and **upper bounds**

Tech Stack

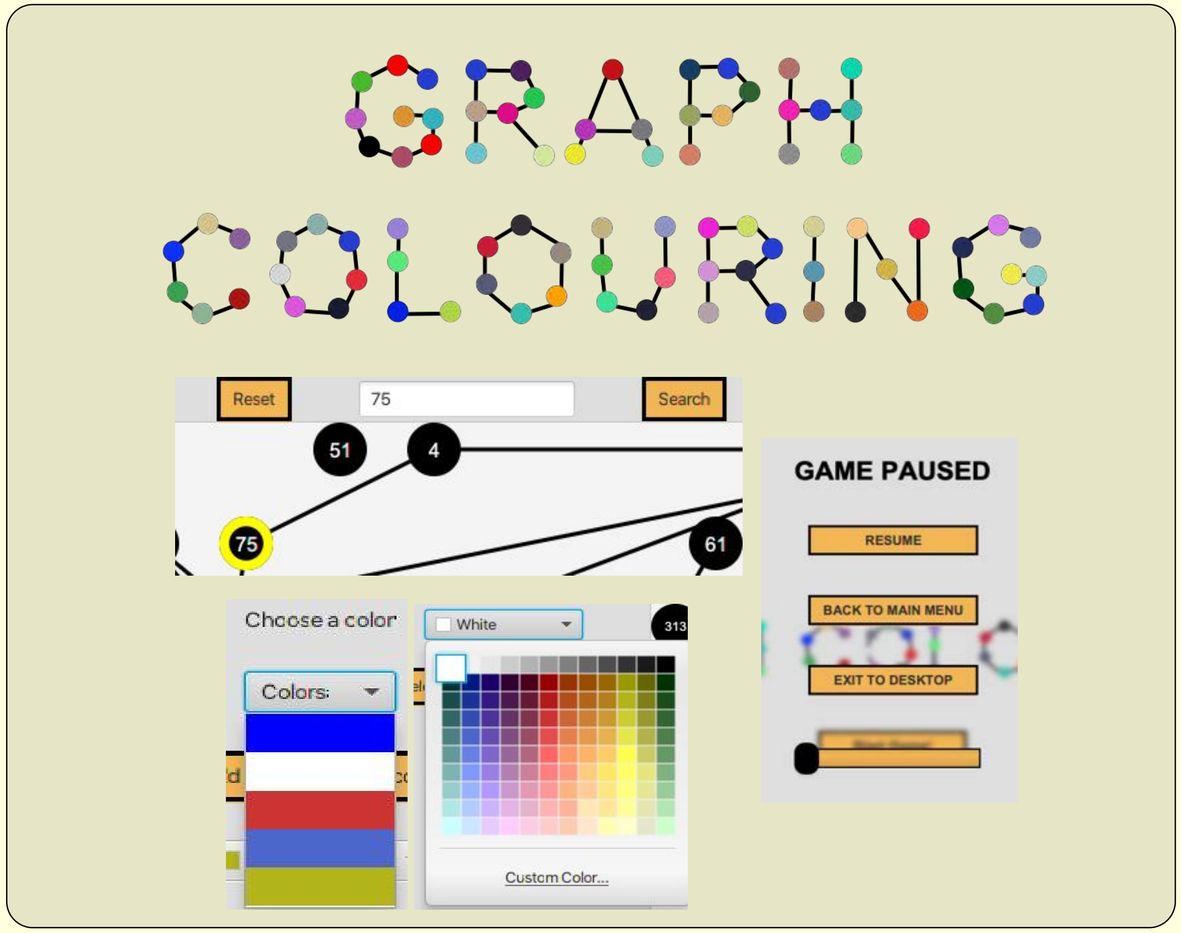
JAVA JAVAFX

Links

Research Paper

State of the Arts

Exact algorithms have been implemented, and generally fall into four categories: **vertex sequential**, **colour sequential**, **dichotomous search**, and **integer linear programming**. Vertex sequential algorithms that dynamically reorder vertices have been shown to perform better.



Approach and Implementation

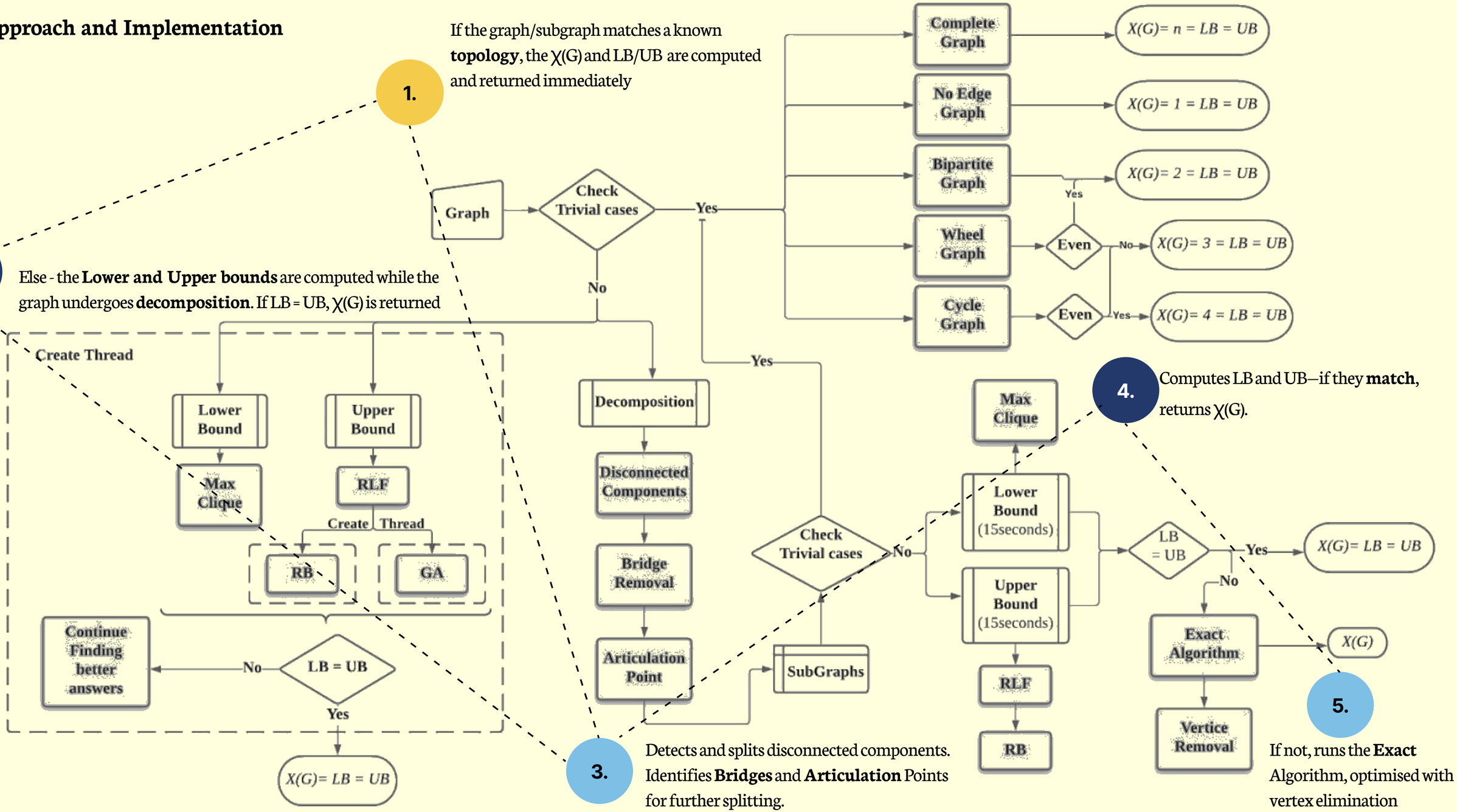
1. If the graph/subgraph matches a known **topology**, the χ(G) and LB/UB are computed and returned immediately

2. Else - the **Lower and Upper bounds** are computed while the graph undergoes **decomposition**. If LB = UB, χ(G) is returned

3. Detects and splits disconnected components. Identifies **Bridges** and **Articulation** Points for further splitting.

4. Computes LB and UB—if they **match**, returns χ(G).

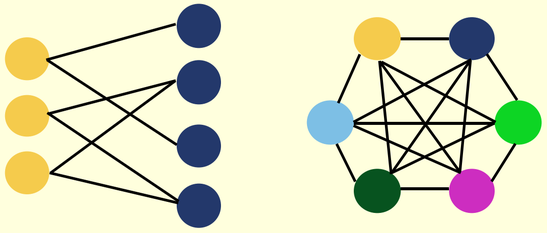
5. If not, runs the **Exact** Algorithm, optimised with vertex elimination



Less Trivial Graph Topologies

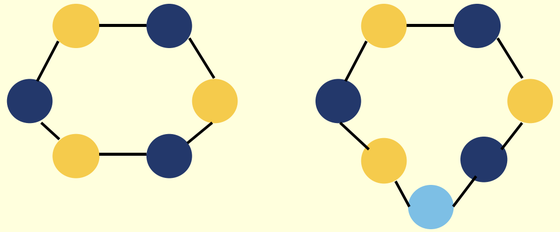
Bipartite and

Complete graphs



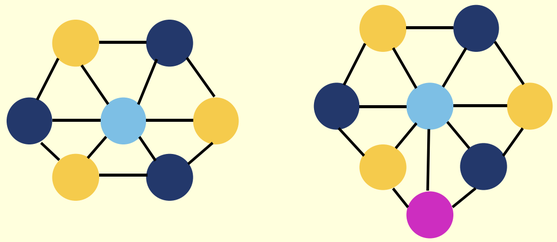
$\chi(G) = 2$ $\chi(G) = \#V$

Cycle graphs



Even $= \chi(G) = 2$ Odd $= \chi(G) = 3$

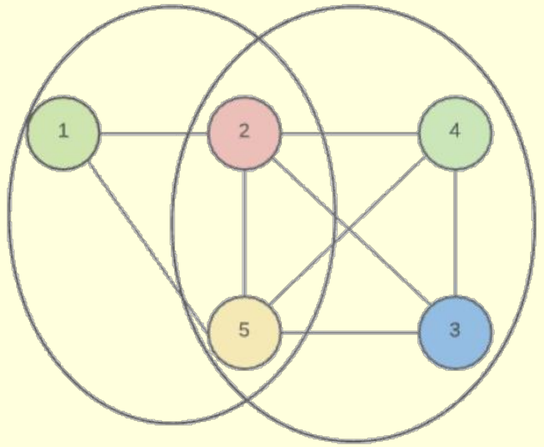
Wheel graphs



Even $= \chi(G) = 3$ Odd $= \chi(G) = 4$

Lower Bound

Maximum Clique Problem



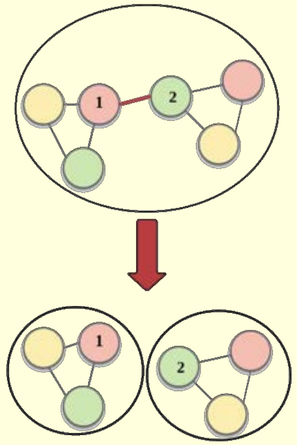
Clique = 3 Clique = 4

Upper Bound

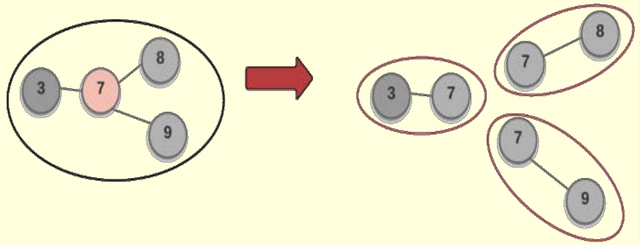
1. Greedy Algorithm
2. RLF Algorithm
3. Genetic Algorithm
4. Reverse Backtracking Algorithm

Optimization

Graph Decomposition



Articulation Point



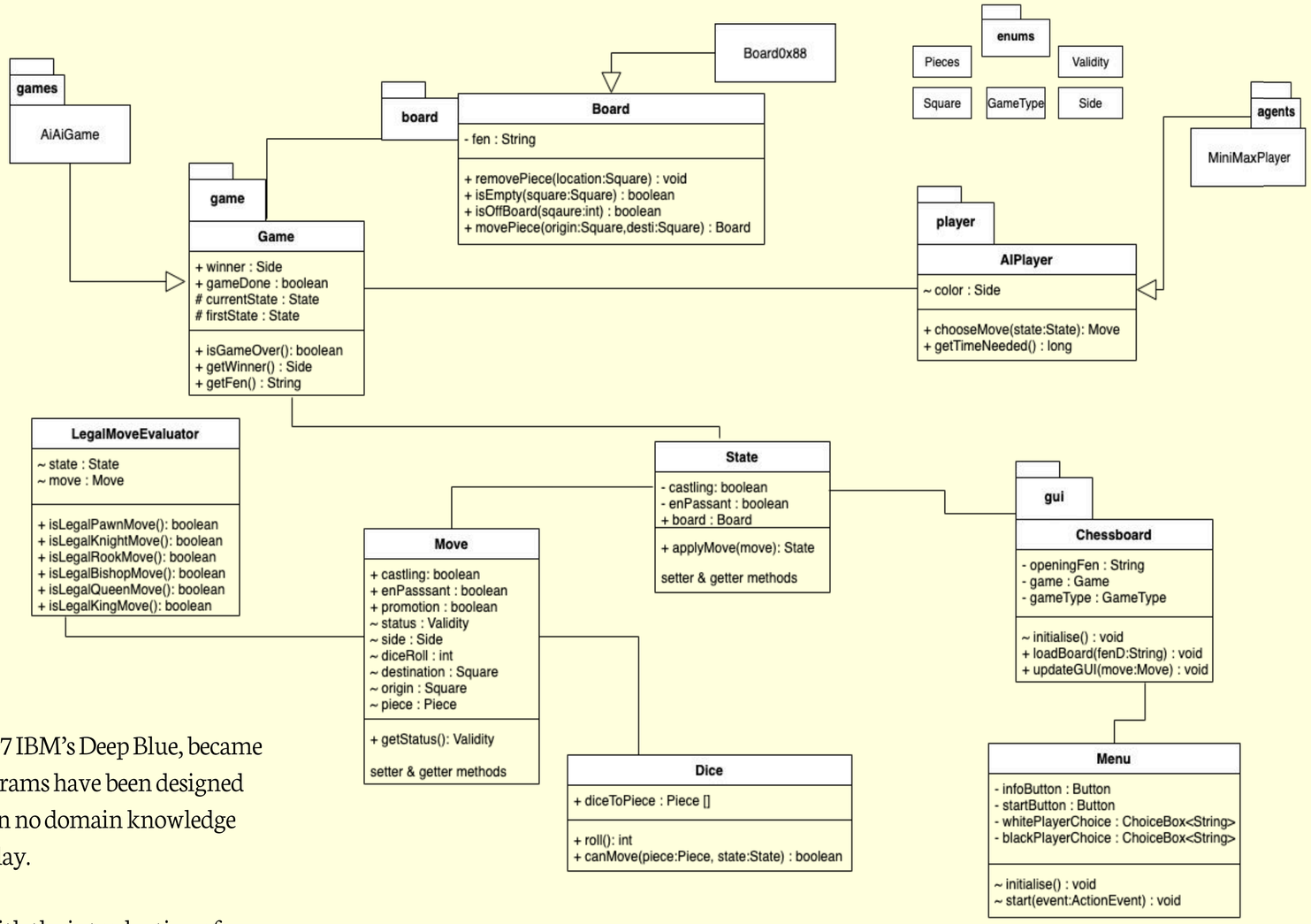
02

Maastricht University

Dice Chess

Project Description

The aim of this project is to research and implement **Adversarial Search** and **Machine Learning** algorithms in Dice Chess and compare their performance against two baseline agents.



State of the Arts

Traditionally, Chess agents were implemented based on **Adversarial Search**. In 1997 IBM's Deep Blue, became the first computer program to defeat a World Chess Champion. Successful chess programs have been designed such as DeepMind's AlphaZero using **deep convolutional neural networks**, given no domain knowledge except game rules and trained solely by reinforcement learning from games of self-play.

Dice Chess changes the deterministic environment of Chess into a **stochastic** one, with the introduction of a dice roll. This project aims to explore approaches for Chess agents to perform well in stochastic environments.

Tech Stack

JAVA JAVAFX LATEX

Links

Research Paper Github

The Evaluation Function

The primary ingredient in a chess-playing program is the board evaluation function. It's **inputs** are the **state of the board** and the colour of the **player** whose turn it is. This is used to inform to what degree one state is more or less **favourable** when compared to another state for a given side.

Agents

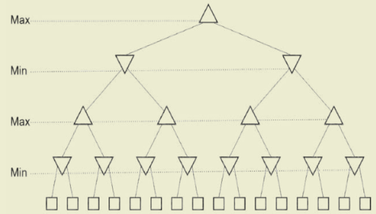
Baseline Agents

Random agent: receives an ordered list of legal moves for the piece selected and randomly chooses one to play using a random number generator.

Basic agent: Chooses the first available capture move. Else it will move to a favourable location on the board

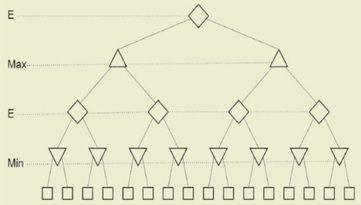
Minimax Agent

Recursive AS algorithm. Generates all possible valid moves for a given board state at each layer, then backtracks to evaluate each state at each node and chooses the move which **minimizes the maximum** possible loss



ExpectiMinimax Agent

Accounts for the stochastic environment. Contains chance nodes that alternate between the max. player and the min. player. Each node contains a list of possible states for a given piece that can be reached from the best state in the parent node.



Q Learning Agent

Initializes a Q-table to estimate the best action for each state, then iteratively updates it using the Bellman equation based on rewards from actions taken. At each step, the agent chooses either a random or previously learned action, applies it, receives a reward, and updates the Q-value accordingly to improve future decisions

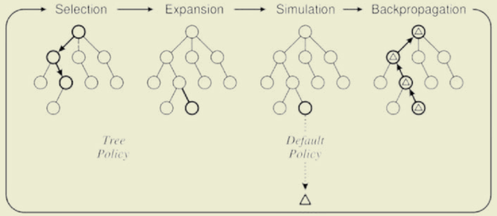
Hybrid Q Learning Agent

A combination of ExpectiMiniMax (AS) and Q-Learning (ML). Here, Q-Learning replaces the normal evaluation function used for each node of ExpectiMiniMax by selecting (as the new heuristic value) the average value for each state-action pair outputted by the generated Q-Table.

Monte Carlo Tree Search

MCTS uses random simulations to gather statistics and make informed decisions, balancing exploration and exploitation. Decision nodes represent (state, roll) pairs, and chance nodes represent (state', action) outcomes.

Selection, expansion, and back-propagation are modified to handle chance nodes.



Game, Experiments and Conclusions

Configuring the game

1. Human vs Human 2. Human vs AI 3. AI vs AI

AI vs AI

Features

En Passant

Pawn Promotion

Overall the MCTS agent had the highest win rate against both baseline agents and against the basic agent it would take the least time per turn, however, MCTS agent scored second highest for turns taken to win.

All Agents vs Random: h(piece sum + point values)

Agent	win rate	time per turn	turns to win
ExpectiMiniMax	0.9396333333	45283389.2	42.23136316
MiniMax	0.9686	450516.0283	42.74446547
QL	0.743	69653317	58.98194805
MCTS	1	275368202.3	44.79
Hybrid	0.2	10364788242	91.28

All Agents vs Basic: h(piece sum + point values)

Agent	win rate	time per turn	turns to win
ExpectiMiniMax	0.7257333333	2.19E23	36.77363333
MiniMax	0.6425333333	1.13E21	40.85153333
QL	0.304	1.67E23	44.53
MCTS	0.8756	142720869.5	41.88729787
Hybrid	0.054	1.86304E25	14.04

The Q-Learning agent had an average performance on all criteria tested since it is a model-free and off-policy algorithm, thus designed to able to perform well enough at any given kind of environment. Nonetheless QL and Hybrid QL were the worst performing agents.

03 Maastricht University

Gesture Detection

Project Description

As part of a simulation of the **CoRoSect** project, the Gesture Detection group contributed to the development of a miniaturised autonomous insect farm. This collaborative project involved several specialised teams, including groups focused on automated guided vehicles (AGVs) for transportation, computer vision for navigation, and a robotic arm for physical manipulation.

The Gesture Detection group was responsible for implementing **custom hand and limb gestures** to serve as an intuitive control interface for three core components of the system: the robotic **arm**, **AGVs**, and a programming **interface**. For our group, the aim of this project was to allow younger or inexperienced users to interact with complex robotic systems by developing a user-friendly gesture-based interface.

Tech Stack

LATEX POSTMAN PYTHON OPENCV

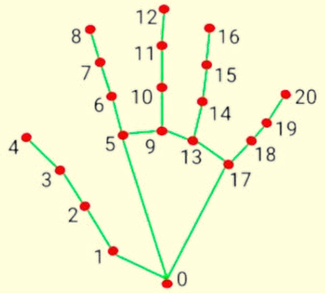
Links

Research Paper | Github | CoRoSect

CoRoSect

"CoRoSect is a three-year project funded under Horizon 2020, the EU research and innovation programme. Led by **Maastricht University**, the consortium is a diverse and dynamic network of partners, representing key players in edible insect technology, farming automation, agricultural robotics, **sustainability**-oriented innovation and collaborative business models."

CVZone



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Custom gestures were implemented using **CVZone**, a computer vision package that uses **OpenCV** and **Mediapipe** libraries, specifically "MediaPipe Hand"s. Mediapipe is a framework that offers customizable ML solutions for processing time-series data and works with a palm detection model and a hand landmark model which provides **21 hand landmarks** from the wrist to each fingertip

Implementation of Custom Gestures

Robot Arm Gestures

The first mode is for the Robot Arm control. Custom gestures are needed to perform certain actions such as **moving or replacing crates** and/or the AGVs. For this mode 6 custom gestures were implemented.

Pickup	Drop	Move Right
Closed Fist	Open Hand	Index+Middle

Move Left	Move Forward	Move Backwards
Index+Middle+Ring	Thumb	Index

AGV Gestures

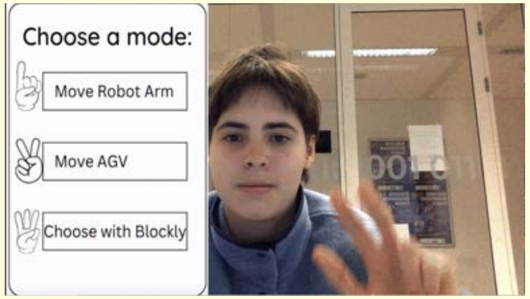
Two approaches were considered for AGV control:

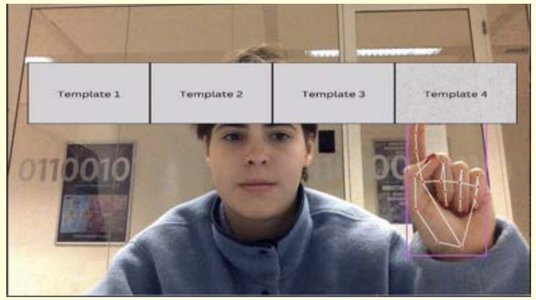
1. **Path Drawing**: allows the users to draw a path starting from one location and ending at an- other location on a custom-generated map of the insect farm. The AGV receives and moves along this path.
2. **Custom Gestures**:

Move Right	Move Left	Move Forward	Move Backwards
Index+Middle	Index+Middle+Ring	Thumb	Index

Programming Gestures

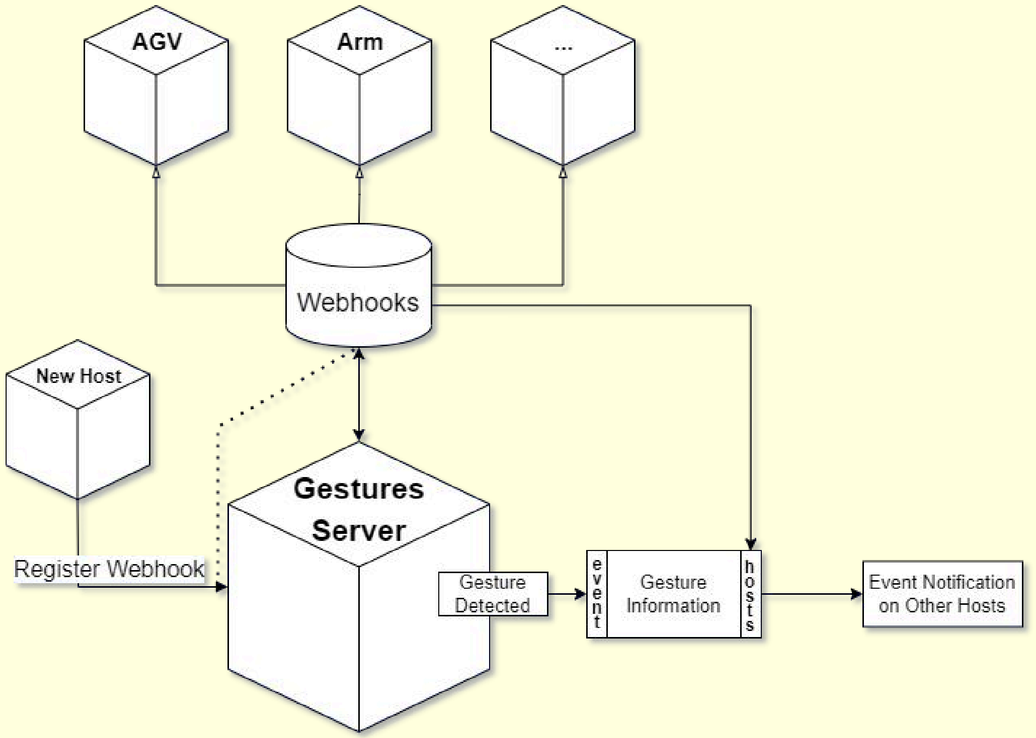
Three custom gestures were used to select, change or run the selected **template**. A template consists of ready-to-run code called "**Blockly**" provided by the programming interface group. CVZone's Hand Tracking Module was used for keeping track of the position of the user's index finger, in order to check if a template had been selected. The chosen Blockly template would be displayed on screen and executed.



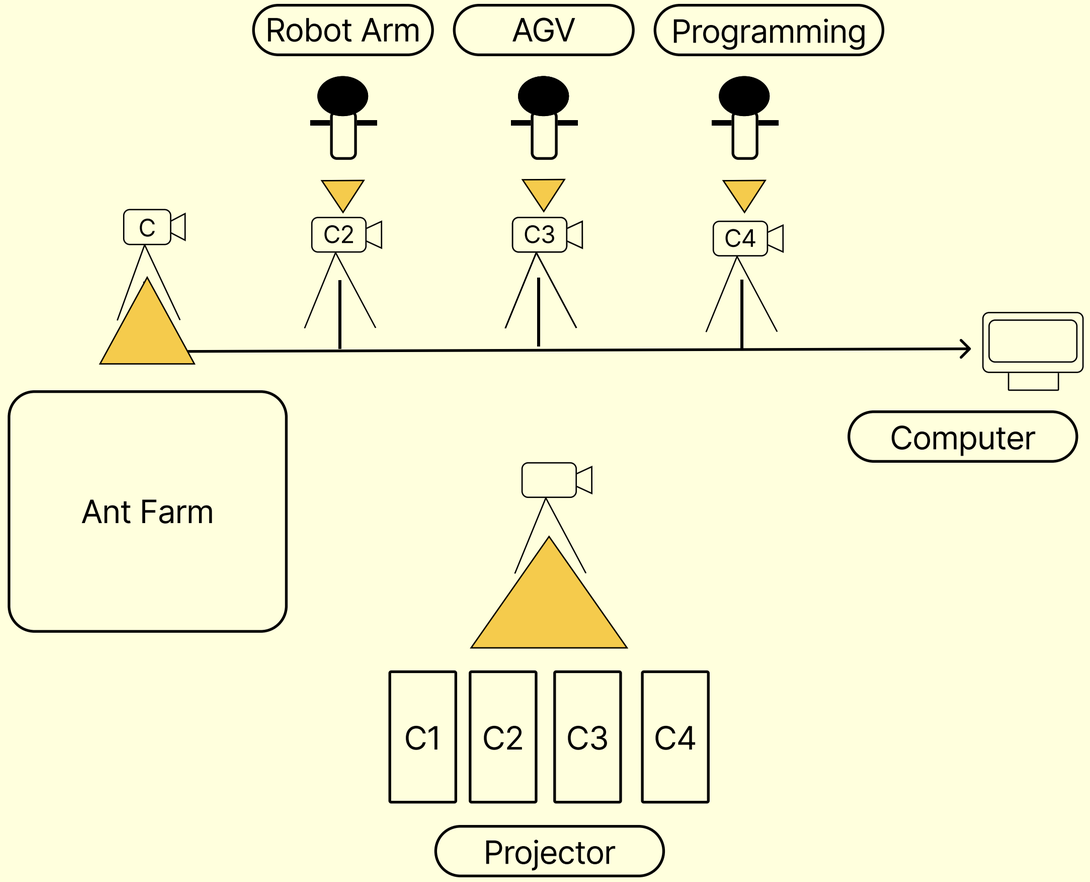


Communication and Interaction

The group developing the programming interface for the simulated ant farm provided API **endpoints** which allowed for other groups to manipulate various variables, subscribe a webhook, add and retrieve a list of gestures. When a gesture detection event occurs, our application sends an HTTP POST request to all the subscribed **webhook** endpoints previously configured to receive JSON gesture **meta-data** related to the detected gesture. The receiving application then processes the event and takes appropriate action, such as triggering a template.



The Physical Setup



During the presentation and interaction with our final product, 4 **cameras** were used and connected to a **projector** to enable users to view themselves on the screen, as well as the actual ant farm setup. Three cameras were used to detect gestures. C1 is used to display an alternative **view** of the farm, C2 was for gestures controlling the robot arm, C3 to move the AGV and finally, C4 was for the programming interface modules.

04 TU Delft Delft University of Technology

NetNote

Project Description

The aim of this project was to design and implement a **distributed** note-taking application that runs in a **client/server** setting. NetNote was the result of a collection of functional and non-functional user requirements that were provided to us. Overall this project aimed to develop a better sense of collaborative work in a professional setting through a real-life situation.

Stakeholders:

User: A user of the client application that uses NetNote to organize notes.

Admin: An operator of a NetNote server application that allows others to host their notes on a central system

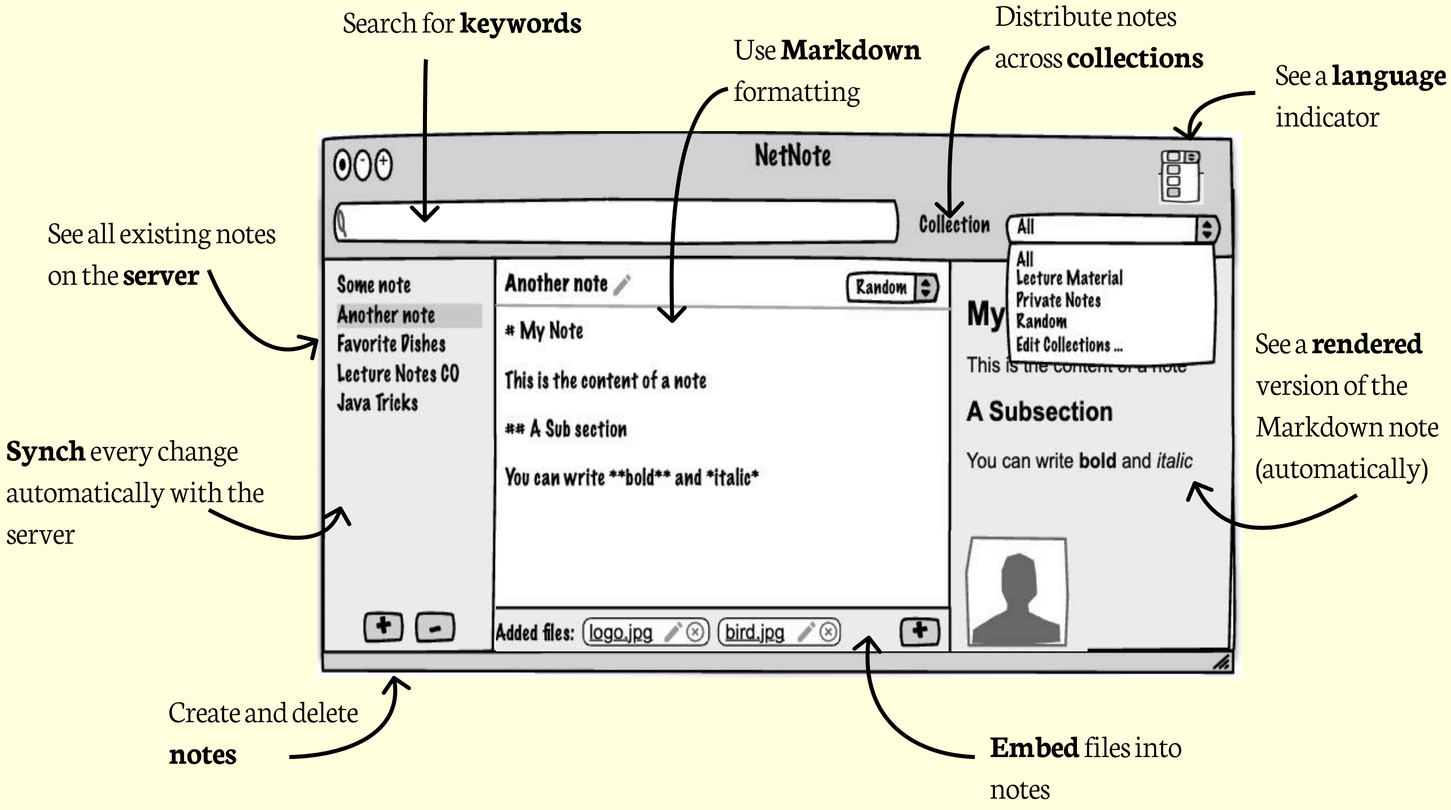
Tech Stack

JAVA JAVAFX SPRING

Technology

The application's backend is implemented with **Spring Boot**, which facilitates the use of Dependency Injection and Repository abstraction. The frontend is built using JavaFX and Scene Builder, which allows for the creation of easy and modern interfaces. Communication between client and server is handled using JSON, with **Jackson**, and automated synchronisation between clients is achieved through the use of **web sockets**.

Epics and User Stories: **As a user, I want to ...**



SELECT * FROM NOTE;

ID	CONTENT	TITLE	DATE_TIME
2905		Coffee Places	2025-04-08 16:48:57
3656	# Portfolio ## TODO - All links - Tech stack stickers of the same size - Uni logo in the same locations - Font and Font sizes - Page Numbers - Link in the table of contents	School Work	2025-04-08 16:54:29
3702	# Favourite Colours Red, Blue, Green	Favourite Colours	2025-04-08 16:51:16
4302	London, Rome	Cities to visit	2025-04-08 16:50:15
4352	Batman	Superheros	2025-04-08 16:50:31

5 rows, 26 ms)

Changes are propagated across all clients and no manual **refresh** is required when a note is edited

Markdown formatting and rendering

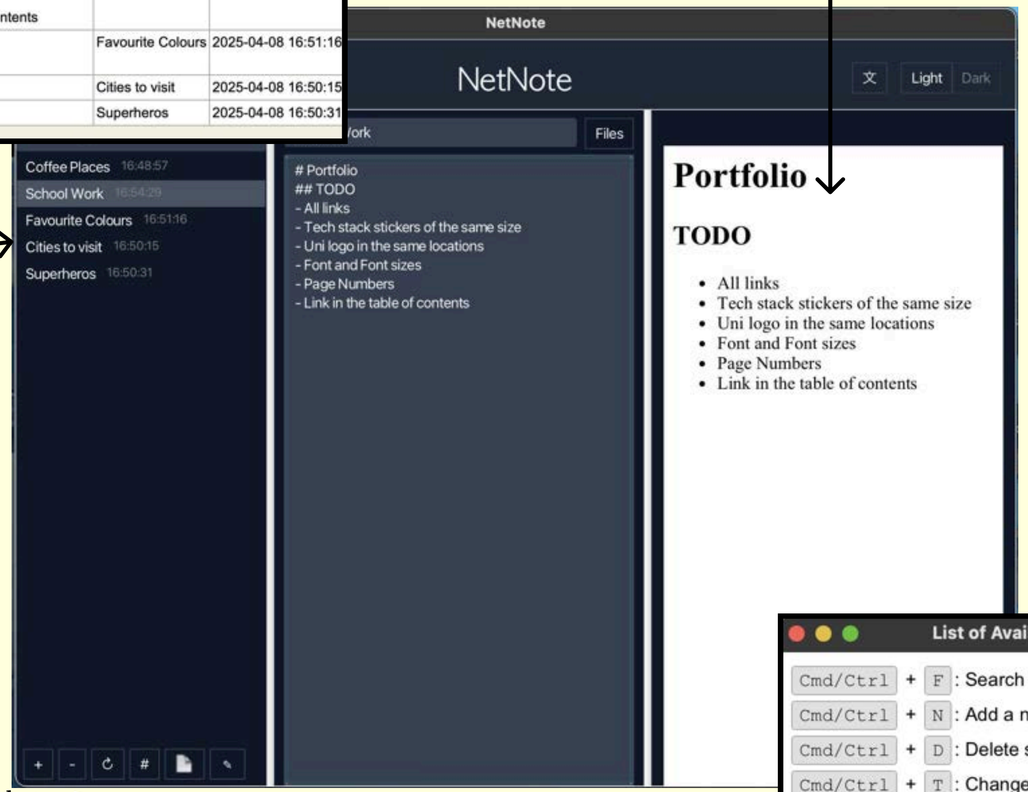
All change operations can be **undone**

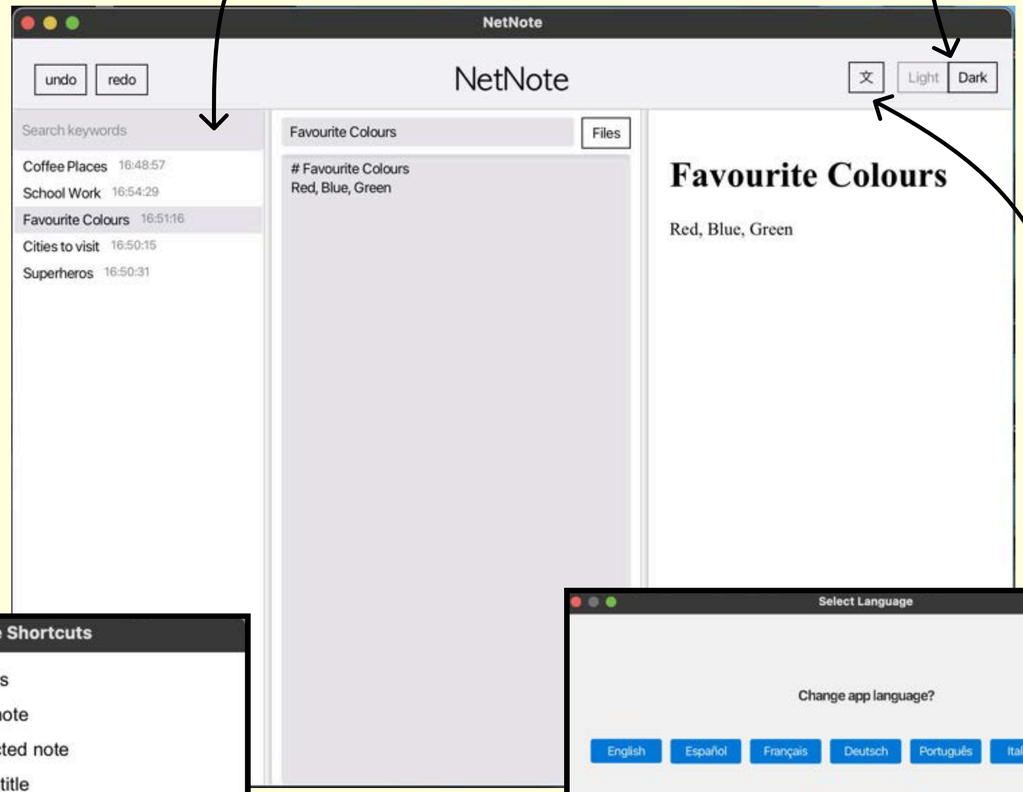
Keyword Search

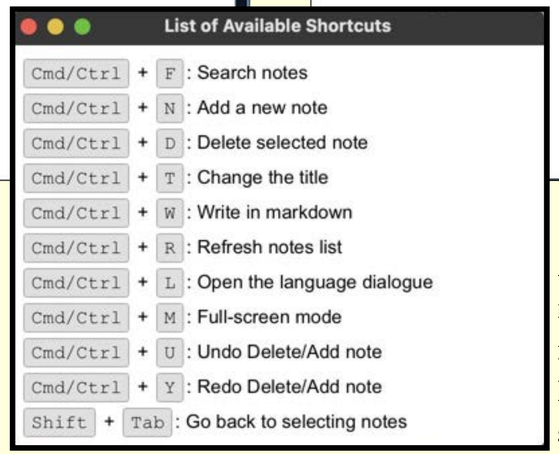
Cof

Coffee Places 16:48:57

Scenes consider a sufficient **colour contrast** for visually impaired users, and UI can switch between Light and Dark modes



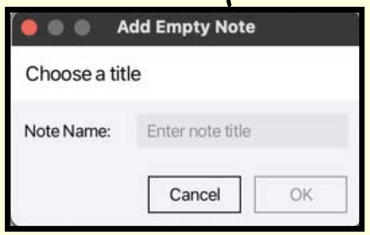




All essential functions are reachable via keyboard **shortcuts**

The user's **language** preference can be switched during runtime and **saved** for each client

All notes have a **unique** title, and all actions need **confirmation**



05

Maastricht University

B - Present

Project Description

This project was developed for a **Human-Computer Interaction and Affective Computing** class. The goal was to design interactive products that are easy to learn, effective to use and ensure an overall enjoyable experience for the user.

B-Present is a prototype of a **mental health** application that merges "Just-in-time adaptive interventions" (**JITAIs**) with a space that can be personalised based on a user's needs. This application aims to support users of **all ages** during tough times, when friends and family members don't have the time and/or space to listen.

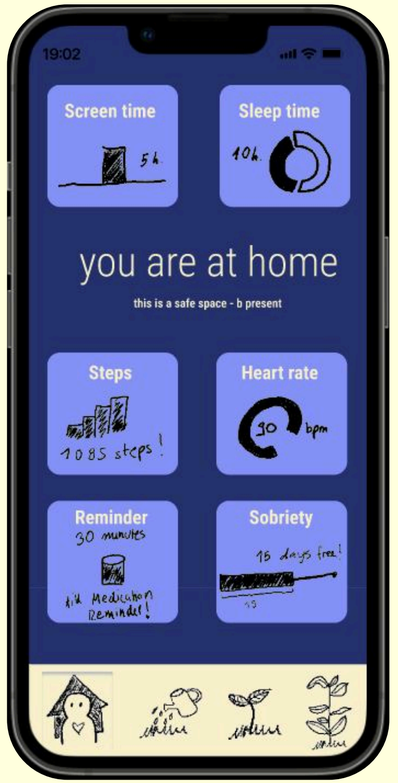
Links

Research Paper

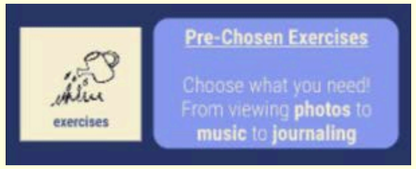
Figma

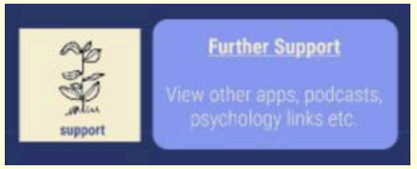
Story Board

Homepage



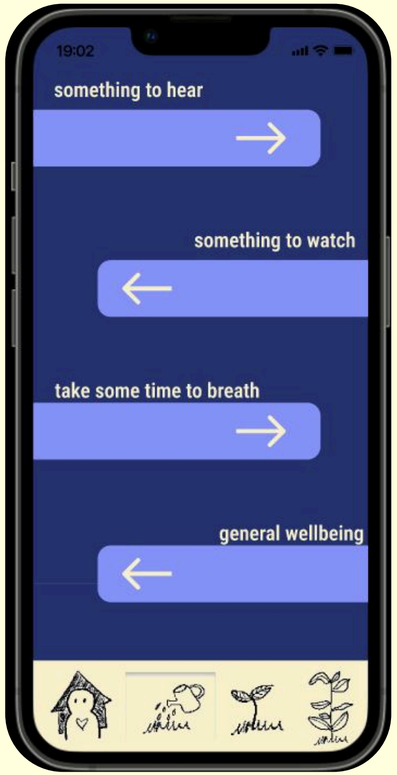
26 **questions** concerning mental health and relationships were sent to prospective users. **Answers** showed that, when dealing with anxiety, many users rely on music, walking and breathing exercises. Blue, Green, White and Purple were colours often associated with the words Calm and Safe. Considering these responses, B-Present opted for a consistent, informal and friendly design.





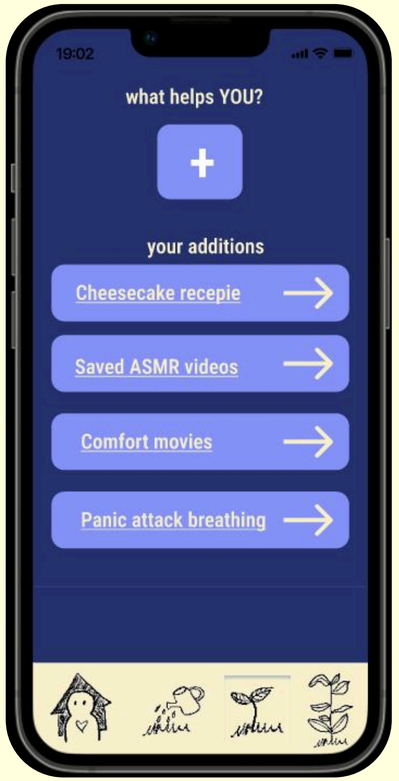
Save **music**, **podcasts**, voice messages from loved people and **audiobooks**

Several **meditation** and breathing exercises



Users can access their saved **pictures** and videos

Space to **journal** or block websites and/or contacts that could worsen their mental health.



B-Present can be **personalised** by creating new sections within the app that can consist of links, drawings, videos ect.

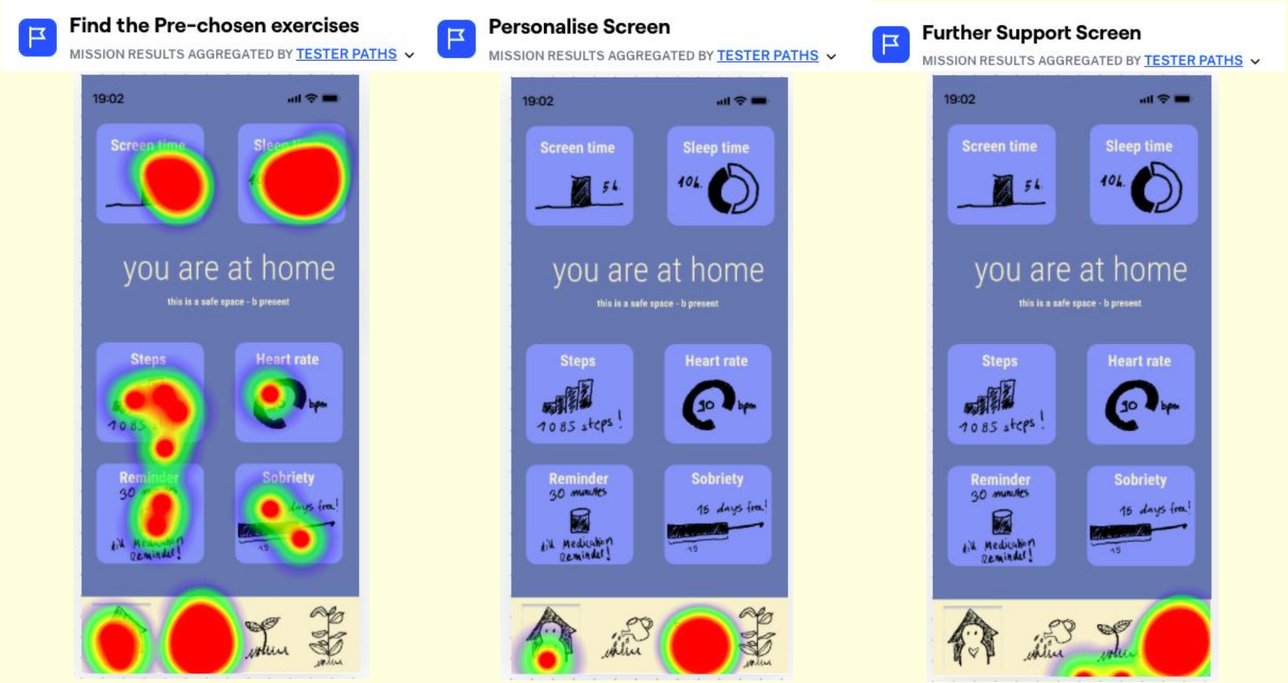
To ensure as much support as possible, users can access the **"Further Support"** section in order to discover similar applications and Psychology or LGBTQ+ pages

User Testing

Using the Navigation Bar

Through a use of a software called "**Userberry**" I was able to observe how users of different age groups interacted with the prototype.

Firstly I focused on evaluating the usability of the application in terms of **learnability** and **robustness**. Learnability was tested by asking the users to execute tasks using the navigation bar. A user would always start on the home screen and was asked to move, first to the "Exercise", then "Personalise", then "Support" screen, as three separate tasks. The principle of learnability can be observed by the decrease in **misclick** rate after each task.



30% Misclick rate

8.3% Misclick rate

5.6% Misclick rate

Confirmation Button Location

Several design choices were tested, including the positioning of **buttons** to **confirm** or cancel an action.

A component that was found to not reflect familiarity was the **help button**. Locating the button resulted in being the task that took users the most time, thus is a feature that should be improved in the future.

The **Mann-Whitney U-Test** (two tailed) was used to analyse the results gathered from two populations: 1. Users of 18-25 years of age, 2. Users above 35. Results show that users from **both** age groups were capable of navigating through B-Present, **without** increasing frustration levels.

06

Maastricht University

Freedom of Expression

NLP-BASED FEATURE EXTRACTION AND DOCUMENT CLUSTERING OF ECHR CASE VIOLATIONS REGARDING ARTICLE 10

Project Description

My Bachelor's **thesis** was brought out in collaboration with professors from Maastricht University **Law** Faculty. The aim of the thesis is to explore the **factors** contributing to the **violation** of Article 10 by utilizing Natural Language Processing (**NLP**) and Machine Learning (**ML**) techniques, specifically KMeans document clustering and LDA topic modelling.

State of the Arts

Most of the published work focuses on multiple ECHR articles and attempt to build a framework to **predict future judicial decisions**. However there is a **lack** of meaningful explanations regarding factors leading to these decisions. Moreover the models are **not** counter-factually robust and there are **lower** accuracy rates when predicting violations of Article 10.

Tech Stack

PYTHON SCIPY LATEX NUMPY PANDAS SCIKIT-LEARN

Links

Research Paper

Github

Definition of Freedom of Expression

- **Everyone** has the right to freedom of expression - includes freedom to hold opinions and to receive and impart information and ideas **without** interference by public authority and regardless of frontiers.
- Does **not** prevent from requiring **licensing** of broadcasting, television or cinema **enterprises**.
- May be subject to **restrictions** or penalties for the prevention of disorder or crime, protection of health or morals, reputation or rights of others.

Retrieving the Documents

HUDOC is a web repository that provides access to the case law of the European Court of Human Rights. The dataset was created by web scraping and consists of **281** cases published in English between **2013** and **2023**

Article	***Violation***	***Non-Violation***	***Total***
10	207	74	281

Text Pre-Processing

Text pre-processing consists of tokenisation, normalisation, stop word removal and stemming or lemmatization. The facts section was extracted using text **segmentation**. Regular expressions are then used to remove all non-alphanumeric characters and **digits**. All words considered normal and **legal** stop words, not in English or with less than 3 letters were removed, the rest were turned to lowercase.

Technique	Description
N-Grams	n = 1,2
Stop Word Removal	True, False
Lemmatization	True, False
Case Type	Violation, Non-Violation

Feature Engineering

Selecting important textual features to enhance the understanding of cases related to freedom of speech. **N-Grams** can be used to identify sequences of words are more probable than others. Moreover, **TF-IDF** was used as the word embedding technique to assign frequencies to terms in the provided documents.

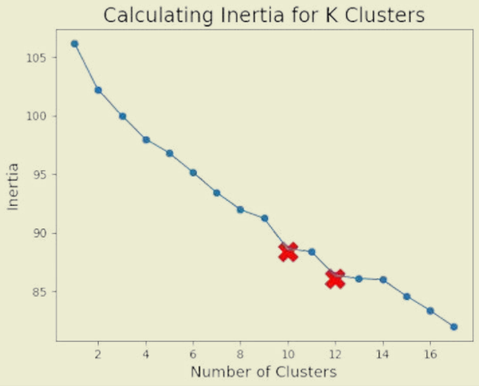
$$\text{TF} = \frac{\text{Number of occurrences of a term in a document}}{\text{Total number of terms in the document}}$$

$$\text{IDF} = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the term}}\right)$$

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$

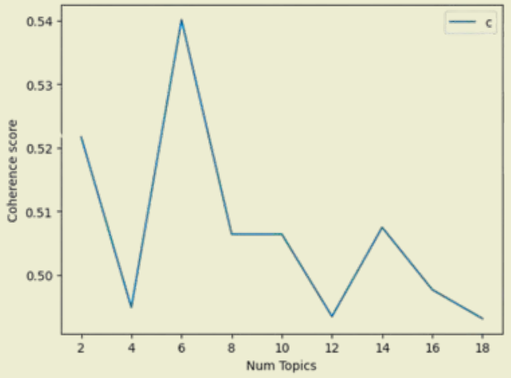
Document Clustering

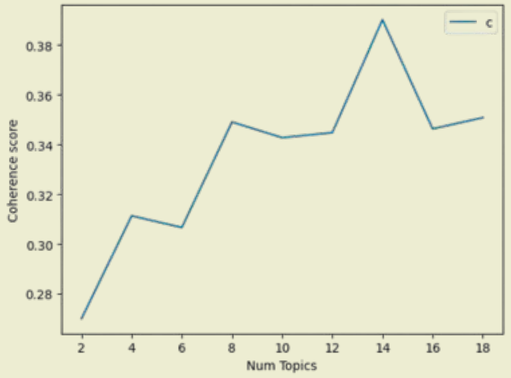
Determining the optimal number of clusters by calculating **inertia**. Singular Value Decomposition (**SVD**) is used to perform dimensionality reduction. SVD on tf-idf matrices is known as Latent Semantic Analysis (**LSA**). The optimal number of clusters is 10 or 12.



Topic Modelling

To determine the optimal number of topics, multiple LDA models were trained with varying topic numbers, and the **coherence score** was recorded. Six topics were used to represent violation of freedom of expression, and 15 for non-violation.



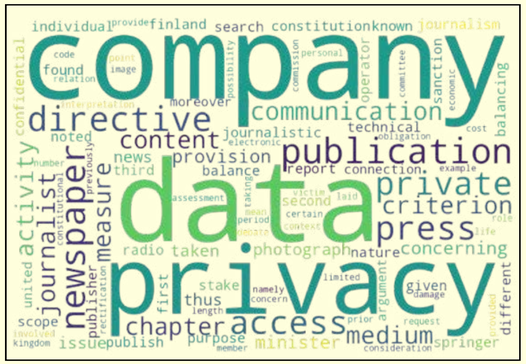


Non-Violation of Freedom of Expression

Main topics: privacy and data, medical procedures and death, hate speech and defamation

The cluster on **abortion** is very connected to the cluster of **religion**. Medical practitioners fired by the catholic hospital for expressing views against the Catholic Church. Individuals in religious organisations are subject to more limitations when practising their right to freedom of expression.

The topic of **religion** seems to be more prevalent in the violation clusters as freedom of expression is violated when racial discrimination occurs in relation to religion, whereas there is **no violation** in cases involving individuals under religious organisations



Violation of Freedom of Expression

Main topics: religion, racial discrimination, academia, politics and violence

Statements regarding **racial discrimination** are often violating Article 10. No matter the platform, any statement regarding racial discrimination and hatred is not protected by Article 10. This cluster **connects** to the cluster representing elections and **politics**.

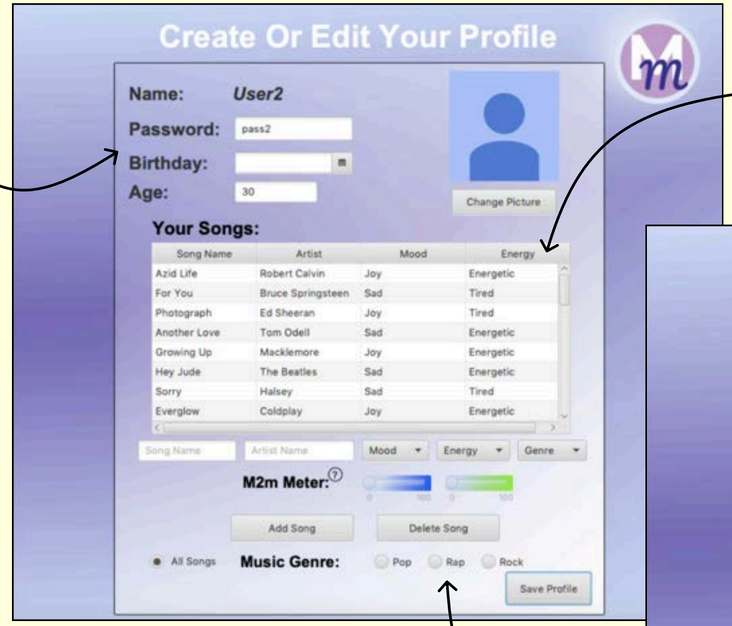
There are several cases regarding convictions of a **politician** for disseminating xenophobic comments on their website as well as elected representatives posting comments inciting **discrimination**.

"Academic freedom" allows individuals employed by academic institutions to freely express opinions on the institution in which they work. If a professor is **fired** for expressing views that go **against** the university, it would be considered a **violation** of their freedom of expression.

07

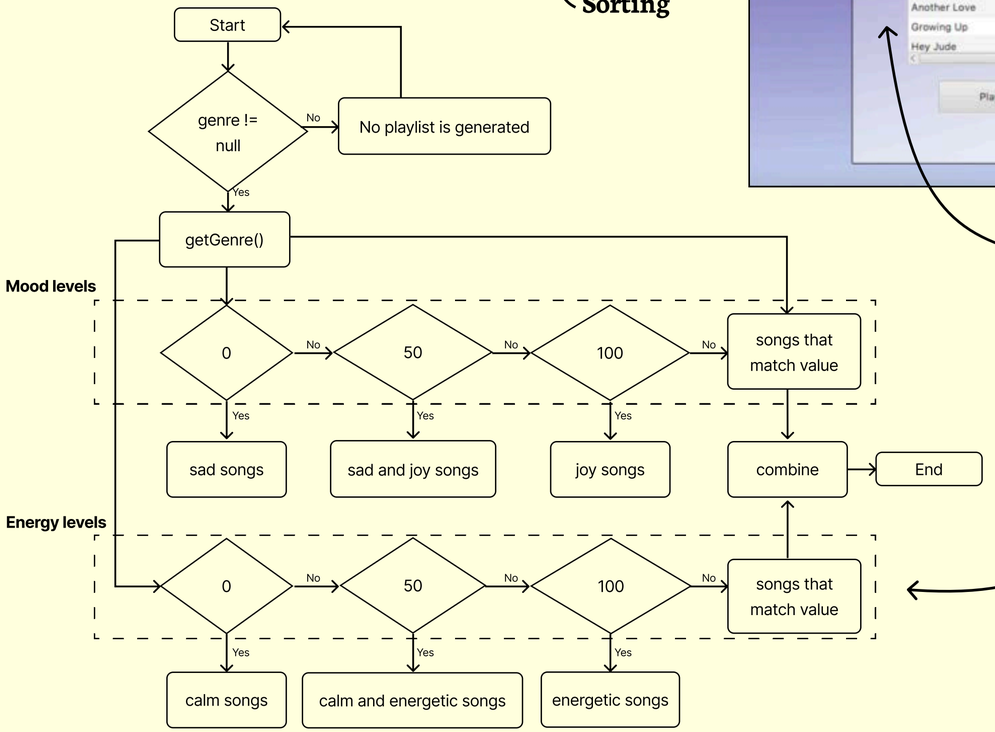
Your Mood, Your Music

User **details**: name and password saved in a **BIN** file



Displays all of the user's added songs and ratings: which **mood** and **energy** level they feel when listening to the song, and what genre it feels for them. All of the ratings are **subjective** and based on preferences

Sorting



The "**Generate Playlist Button**" takes how the user is feeling and which genre they want to listen to and creates a customised playlist. The song can be **played**, paused and stopped.

Project Description

"Your Mood, Your Music" is a project developed for the Internal Assessment of Computer Science during the **International Baccalaureate**. It is a JavaFX program that allows users to **generate** different playlists with their own songs based on their **mood, energy**, and preferred musical **genre**. More specifically it allows the creation of over **340** different playlists. The code implements the **MVC** (Model-View-Controller) design pattern.

This project is almost **5 years old**, however the idea behind it is a concept that I would like to continue to explore by developing a more refined application with the skills I gained throughout the years

Tech Stack

JAVA JAVAFX

Links

Github

08 TXT AI Tool

Project Description

While working for TXT E-Tech I proposed to develop a tool to **automate** several repetitive tasks occurring during the **debugging** phase. Moreover I developed a prototype of a **chatbot** trained on the codebase that would suggest improvements and generate **new** code.

During my time in TXT E-Tech I worked as a **software engineer** consultant for a client in the **Aerospace** sector. For this reason some of the images used in this portfolio are **blurred** to censor sensitive information.

Tech Stack

LangChain · LangGraph · W&B · Chroma · Hugging Face · OpenAI · Streamlit · Qt

Problem Statement

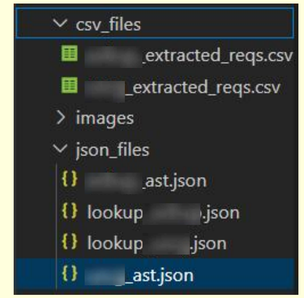
The programming language used for aircraft software is **Ada95** and the IDE used was **GNAT**. This project aimed to decrease the time spent on:

- **Debugging** a large codebase
- Refactoring code for it to comply to a specific **Coding Standard**
- Writing **System Requirements** that follow a pattern

AdaAnalyzer

AdaAnalyzer was a project developed by some colleagues which scraped the aircraft's codebase storing all important information in a **JSON** file.

I used AdaAnalyzer to create my **database** by extracting all existing requirements and the corresponding code



App Description

1. The chatbot part of the application consisted of 4 modes - **Normal Chat** was included to avoid switching between applications and have a chatbot ready to answer any question. The **open-source** LLM is chosen by the user

Structured Output

```
class Finder(BaseModel):
    package_name : Optional[str] = Field(..., description="the name of the package")
    variables: Optional[str] = Field(..., description="the name of the variable")
    functions: Optional[str] = Field(..., description="the name of the function or procedure")
```

Question + Prompt + Chat History + LLM

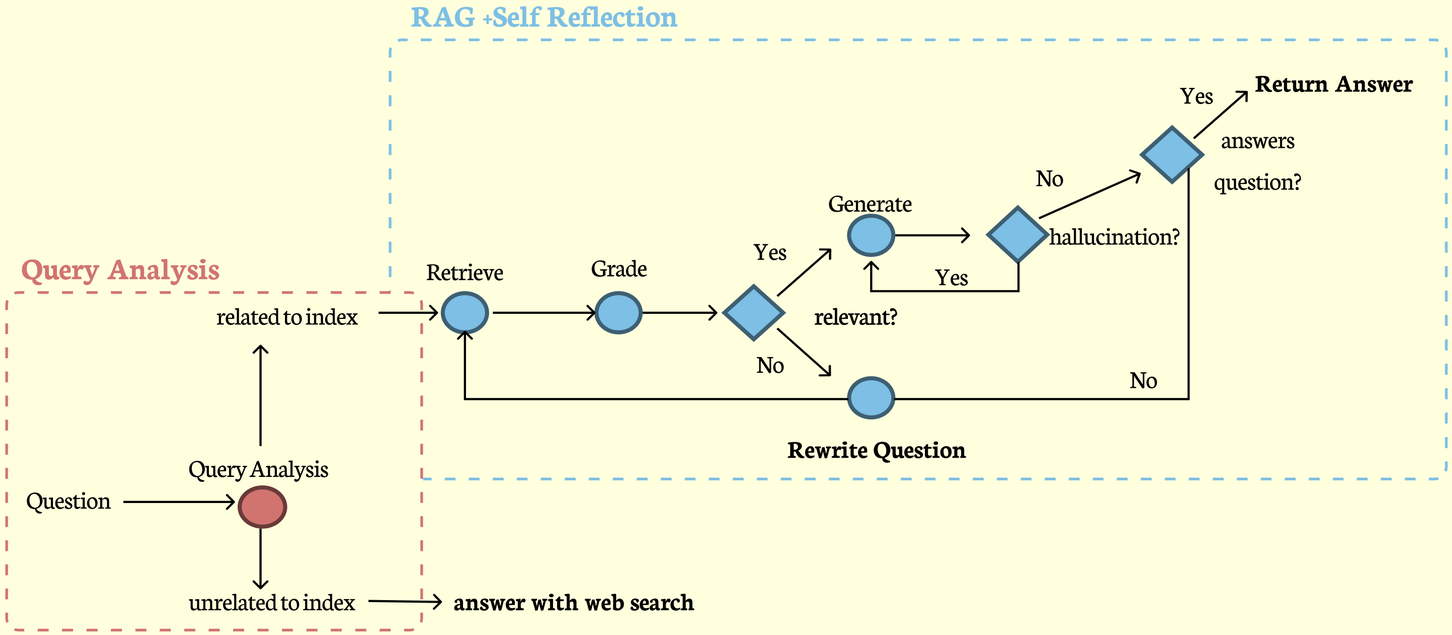
AdaAnalyzer + Structured Analysis

3. Similar to the Normal Chat but would perform **RAG** on the **Codebase** and return **structured** answers to facilitate debugging

2. Similar to the Normal Chat but would perform **RAG** on the **Coding Standard PDF**

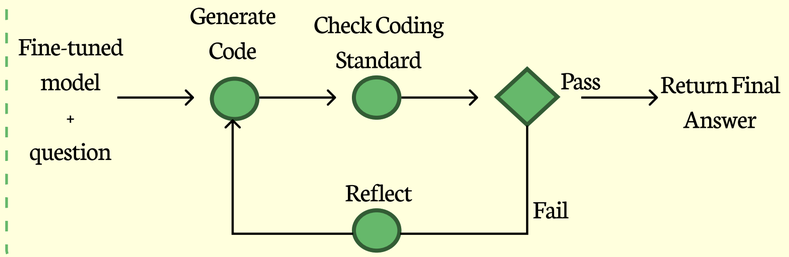
Question + RAG + Chat History + LLM

Adaptive RAG + Self Corrective Code Generation



4. Based on the user question the LM would decide to perform an **Adaptive RAG** or to generate completely **new** code.

Self Corrective Code Generation



Chatbot

Choose the aircraft configuration **codebase** you want to use

Update database when changes have been committed to the company's repository

Store **Secrets** for the chosen open-source LLM's

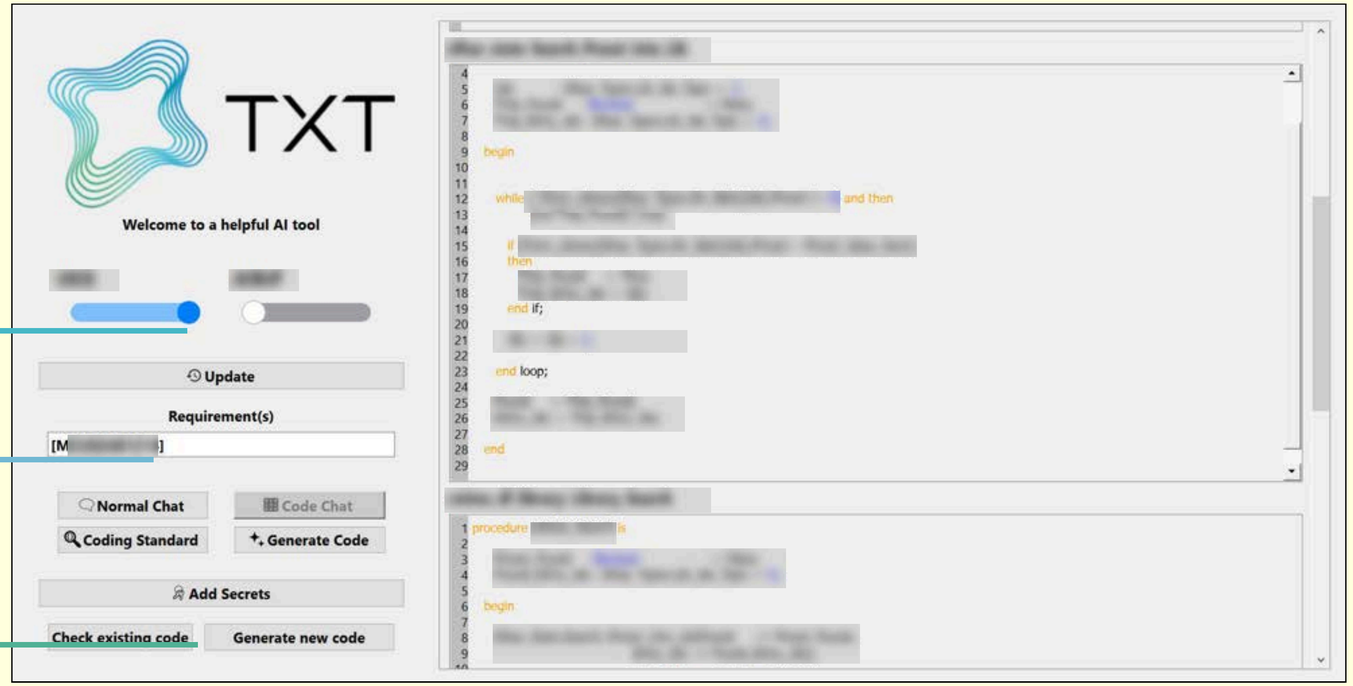
Generate New Code Button - shows the **Chat** UI

Check Existing Code

Choose the aircraft configuration **codebase** you want to use

Paste one or more **requirements** for the chosen configuration

Displays all of the code that matches the **searched** requirement(s). Simplifies debugging by displaying everything together



thank you!