Team NA-Boxjelly

Project ChatGPT and Nao Robot

+ NAO6 + ROS

Team members, Client and Supervisor

Team members

Jun Li Chen	Product Owner
Joel Towell	Scrum Master
Yanting Mu	Quality Assurance
Chang Shen	Development Leader
Tianyang Wang	Test Leader

Masters of IT Students from different specializations, in AI, Cybersecurity and Distributed Computing.

Client

Wafa Johal

Supervisor

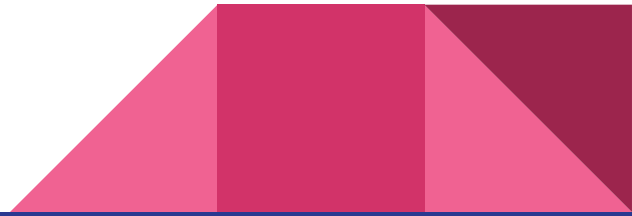
Max Plumley

Project Background

- CoWriter is a program where kids with handwriting difficulties practice handwriting by teaching Nao robot how to write
- Originally developed 7-9 years ago, fallen out of use and not maintained
- Written in Python 2.7 and using older version of Robot Operating System
- Hard-coded dialogue in the original program
- Release of ChatGPT has unlocked potential for more fluid human-robot interactions
- Renewed interest in CoWriter; integrating ChatGPT can provide a more engaging and natural interaction

Intended Users

- **Children**
  - who are actively interacting with the robot
- **Educators**
  - who facilitate the interaction between the children and robot
- **Developer/Maintainer**
  - code is a work in progress, work will be done to maintain and extend the code; we have approached the task with this user in mind too



Delivery Goals

1. Working port of original functionality from Python 2.7 to 3.8.10, and using ROS Noetic
2. Incorporation of ChatGPT into the dialogue manager
3. Speech to text to convert spoken sentences into ChatGPT inputs
4. New states to allow for interaction between child and robot (speech to text and ChatGPT)
5. Extensive unit testing to demonstrate robust functionality and usage documentation
6. Documentation of every class, method, and function, to allow for ease of maintenance and extensibility

Development Goals: Educator

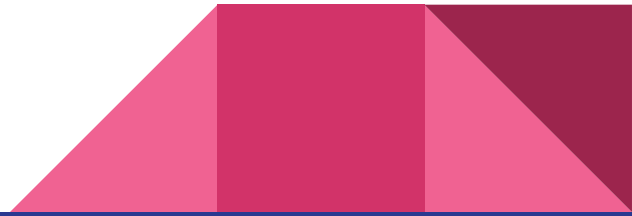
- "As an Educator, I want the robot to use ChatGPT as a dialogue manager, so that it can engage in more natural conversations with children and reduce the need for my intervention."
- Educator is mainly concerned with delivery goals 1 through 4

Development Goals: Child

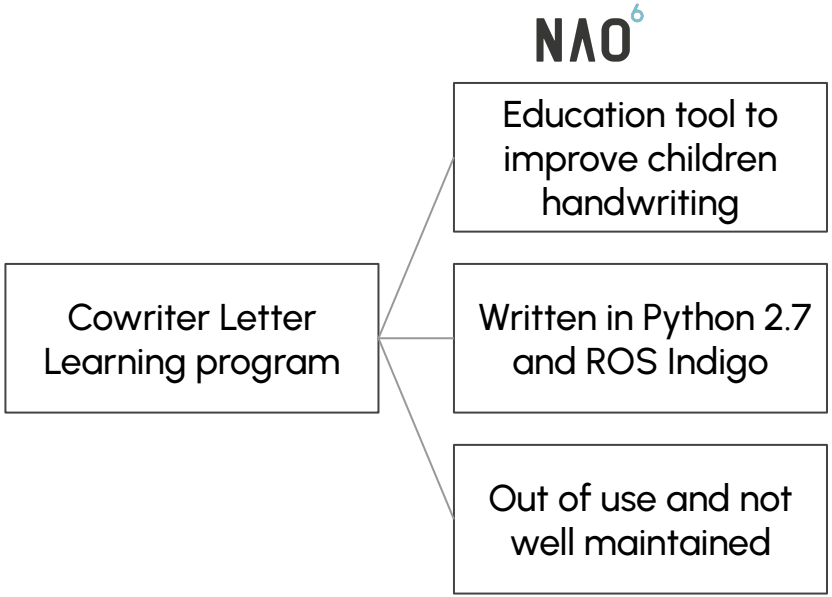
- "As a Younger Child, I want the robot to accurately parse audio into text, so that it can understand my spoken words and provide appropriate feedback and guidance."
- Child is mainly concerned with delivery goals 1 through 4

Development Goals: Developer

- "As a Developer, I want to be able to read documentation for every class, method, and function used in the CoWriter software."
- Developer is mainly concerned with delivery goals 5 and 6



Project Summary



Goals:

- Maintain compatibility with modern systems
- Improve program capabilities and performance
- Facilitate easy software maintenance and updates
- Enhance the user experience
- Personalize the learning experience

Live Demo

Robot Demo in Simulator

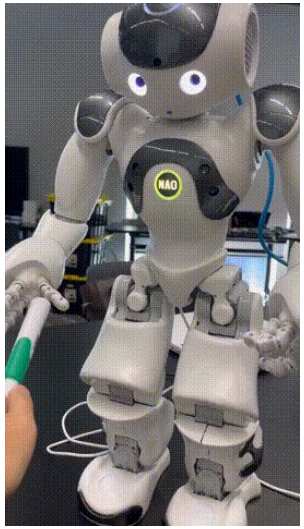
Scenario:

- Educator sending text to ChatGPT
- Educator giving letter to write
- Child giving shape as feedback
- Educator giving word to write
- Child using voice as feedback

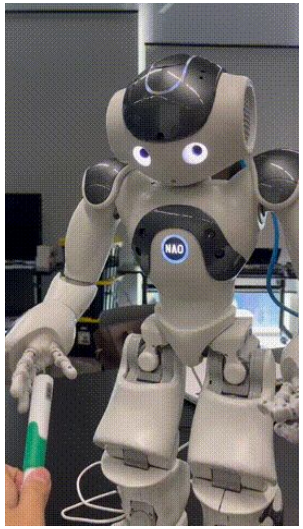
Real Robot Testing

Writing different words

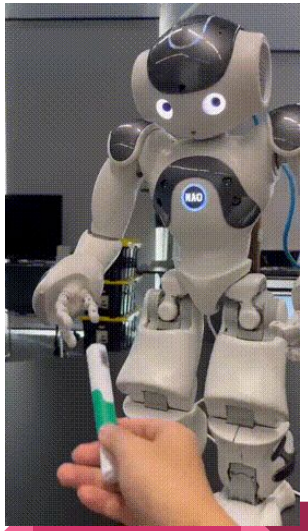
child



hello



puppy



test

Real Robot Testing

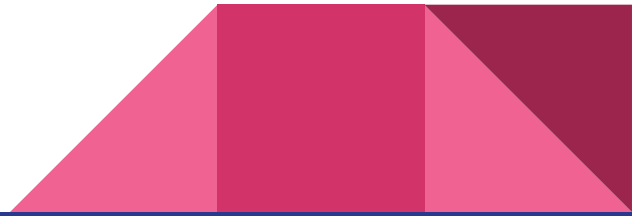
Learning from the user

Real Robot Testing

Talking with the user

Team Achievements

- Seen a complex project from start to finish with a working product at the end
- Meticulous approach to documentation and testing
  - next set of developers to take this project forward should have an easier time than we did.
- Took on the spirit of the subject and focused on *process*
  - we made mistakes along the way, but put considerable effort into following Agile
- Maintained a respectful and cooperative attitude, even in the face of deadline stress and differing viewpoints

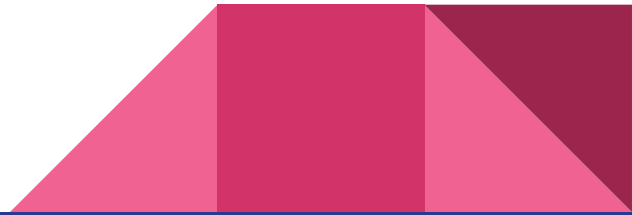


Team Challenges

- Lack of experience programming with ROS
- Sparse documentation on legacy code
- Sparse documentation on programming Nao in Python 3
- Varied availability of team members created difficulties staying on schedule
- Siloed approach to work early on led to difficulties integrating different packages

Team reflection

- It is a challenging project that allows people to accumulate a lot of practical development experience.
- Sprint retrospectives were really valuable; often at uni, you finish a piece of work and move on, but these forced us to be reflective.
    - You can see a progression on Confluence where our processes became tighter and workflow better managed through each sprint, often due to changes proposed in retrospective
- Felt the applicability of Tuckman's model of team development
    - Towards end of project, much more collaboration on individual tasks and this sped up progress
- Gave us a deeper understanding of version control
    - All had used Git, but mostly in isolation and on small projects. Using it on a large project with a team is more involved, and we can all take lessons from this into future roles



Delivery summarised

- Successfully ported the original program to Python 3.8 and ROS Noetic
- Integrated audio input and speech recognition using locally-run OpenAI Whisper language model
- Used OpenAI API to integrate ChatGPT to handle parts of the interaction
- Added new conversational states to facilitate conversation with the robot, using microphone input, Whisper and ChatGPT
- Extended existing UI
- Extensive, clear and concise documentation of the code
- In-depth installation/setup guide (& auto–installer script) for quick and easy dev. environment setup

Conclusion

- Overall, we feel this project was a success
- We analysed user requirements, negotiated feasible scope with the client, and delivered what we promised
- Meticulous approach to testing and documentation meant we only delivered core functionality, but makes it easier for future developers to extend this work
- Thanks for listening to our presentation!