Social Interactivity Mentor for Youth with Autism using the NAO Robot (SIMYAN)

Final Presentation

Andrew Nguyen, Bryce George, Colton Homuth, William Ross

Sponsor: Dr. Adham Atyabi
Faculty Sponsor: Mr. Bill Michael

May 7, 2021

Project Recap

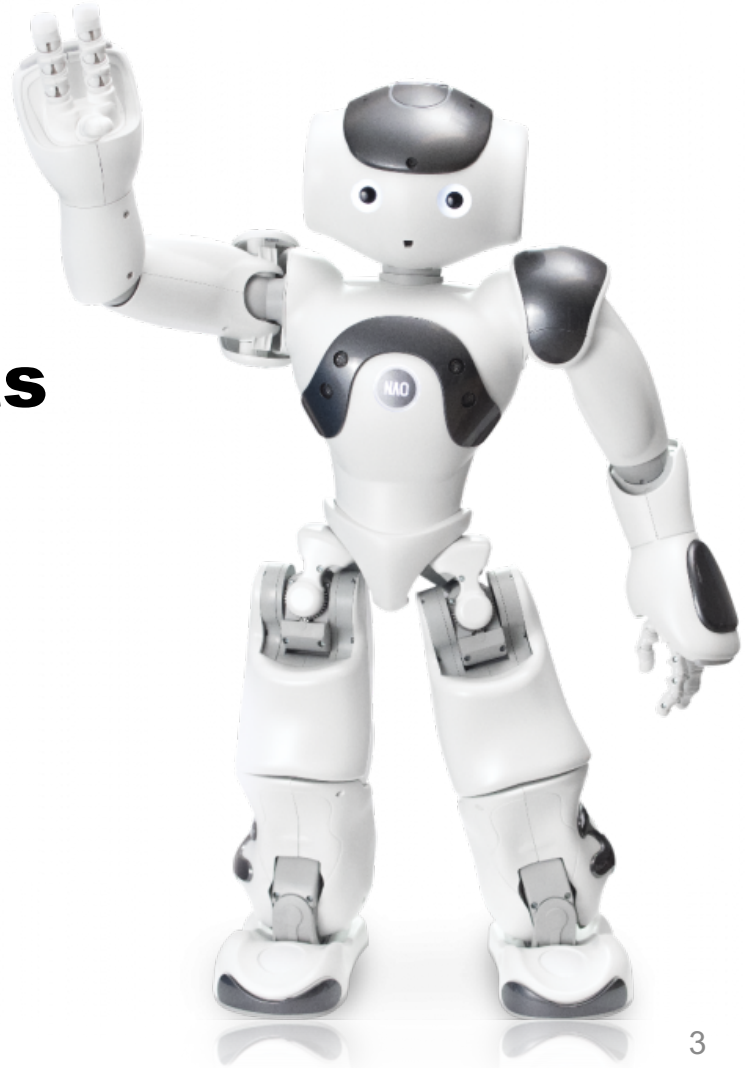
Objective:

Create a framework for the NAO robot to support future implementation of ASD treatment/intervention activities designed by medical professionals and researchers.

- **Core SDK**
  - Supports implementation of advanced social interactivity behaviors
  - Framework for building interactive activities
  - Orchestrates provided services
  - Extensible, documented, and easy-to-use
- **Drawing Demo**
  - A PoC activity
  - Guides human subject through a drawing exercise
  - Demonstrates a social interaction between the robot and the subject

Presentation Overview

- **Problem Review**
- **Project Requirements**
- **Solution Overview**
- **Test Strategy & Results**
- **Project Status**
  - Goals
  - Schedule
- **Budget**
- **Lessons Learned**
- **Conclusion**



Problem Review

- **ASD can impair social development which is vital for success is today's society**
- **Robotics can be used to help develop these skills in children with ASD**
- **NAO robot is designed for social interactivity**
- **Robot comes with good functionality but still requires further development**

Requirements

- **Integrate Jetson Nano GPU wirelessly**
- **Integrate Kinect to Jetson Nano**
- **Create, Document and Test SIMYAN Framework**
- **Drawing Activity Proof of Concept**
- **Machine Learning for additional autonomy**

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
1	Load Master Module	1. Allocate all necessary resources and load the Activities Master Module 2. Detect subject within 5m of starting position	1. Performed on robot startup 2. Verbal notification of completion
2	Initialize Activities Master Module	1. Verbally greet subject 2. Orient face to look at subject (+/- 5°) 3. Allocate/initialize all master level subroutines	1. Greet subject < 1s after detection

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
3	Start Activity	1. Verbally explain operation instructions 2. Verify valid verbal activity selections 3. Notify of invalid activity selections and re-prompt 4. Discover all available activity modules 5. Load and initialize selected activity module	1. Verify and acknowledge selections <= 1s after heard

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
4	Select Object to Draw	1. Verbally explain drawing activity 2. Verbally list drawable objects 3. Prompt for object selection 4. Verify object selection 5. Notify of invalid object selections and re-prompt	1. Verify and acknowledge selections <= 1s after heard

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
5	Draw Object	1. Locate/detect drawing surface within 5m of position 2. Notify of no drawing surface found and prompt for next action 3. Detect drawing surface boundaries 4. Notify inaccessible drawing surface conditions and instruct how to position correctly	1. Detect drawing surface in <= 3s 2. Identify drawing surface boundaries in <= 1s

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
5	Draw Object (continued)	6. Load necessary hand/arm motor control module(s) 7. Locate object drawing instructions 8. Execute drawing instructions to draw the object on the drawing surface 9. Able to draw square a. Four angles within +/- 1° of 90° b. Four sides same length to within +/- 5mm	

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
5	Draw Object (continued)	10. Able to draw rectangle a. Four angles within +/- 1° of 90° b. Two sets of sides same length to within +/- 5mm 11. Able to draw circle a. No segment of circumference varies by more than +/- 5% of radius length in distance from the center	

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
5	Draw Object (continued)	13. Able to draw triangle a. One angle +/-1° of 90° b. Two angles +/-1° of 45° c. Length of sides adjacent to 90° angle are same length within +/- 5mm	

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
6	Interact with Subject	1. Select an appropriate user interaction statement 2. Speak interaction statement to subject 3. Listen for verbal response to an interrogative statement 4. Evaluate response statement	1. Select or generate an interaction statement in <= 3s 2. Evaluate a response statement in <= 3s

REQUIREMENTS SPECIFICATIONS

#	Use Case	Functional Requirements	Quality of Service Requirements
7	Complete Activity	1. Evaluate whether activity can be repeated 2. Ask whether to repeat activity or exit activity 3. Verify and verbally acknowledge choice to repeat or exit the activity 4. Re-initialize activity module 5. Exit and unload activity module	1. Verify and acknowledge selections <= 1s after heard

Standards

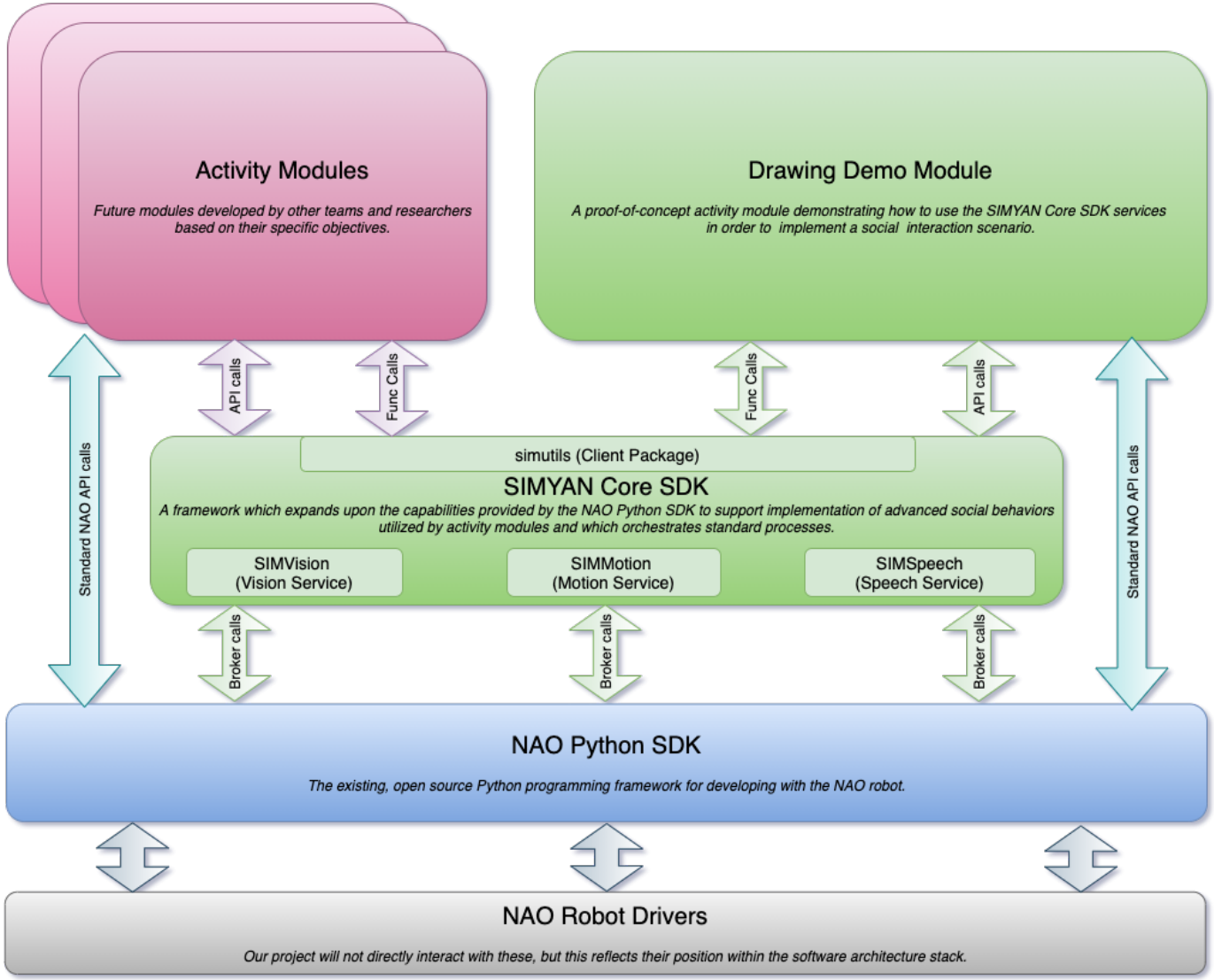
Coding Standards	• Python PEP 8 formatting style
	• Use 4 spaces instead of tabs
	• Comments
	  • Describe all functions and their parameters
	  • References/links to supporting documentation
	• (Optional) Custom utilities should provide a command line or GUI interface

Documentation Standards (GitHub)	• Committed to UCCS-Social-Robotics/docs • Must include a Wiki page or markdown for each module • Include author(s) GitHub username and date information at the top of documentation files and wiki pages
Safety Standards	• Subscribe to IEC 60601 technical standards • Covid-19 Safety Protocol

Solution Overview

- **Team was able to fully integrate Jetson Nano GPU wirelessly**
- **Team was able to expand upon software architecture of NAO using Python and document/test these updates**
- **Team was able to create successful drawing activity with multiple shapes with fixed variables**

Solution Architecture



Impediments

- **Covid-19 pandemic caused many issues**
  - Limited time with the robot early in the semester
- **NAO robot has very limited documentation**
  - There is very little support for the NAO, so a lot of functionalities are either documented poorly or not at all.
  - Project turned from SDK development to R&D
    - ***Much of our time was spent figuring out how to interface with NAO instead of extending existing capabilities***

Integration Test Strategy & Results

- **Test inter-module connections in the code:**
  - **Speech Recognition to Drawing Module interaction**
    - **Object Selection**
    - **Social Interaction**
  - **Motor Control Module to Drawing Module Integration**
    - **Draw on Drawing Surface**

Integration Test Strategy & Results

- **Results:**
  - **Speech Recognition to Drawing Module interaction**
    - **Recognize verbal commands to draw a picture**
    - **Invoke speech events to draw a particular shape**
  - **Motor Control Module to Drawing Module Integration**
    - **Draw different types of shapes on the drawing surface**
      - Circle, Triangle, Square, House (Square + Triangle)

Incomplete/Remaining Test

- **Incomplete Tests**
  - **Did not test for expected interaction flow from invalid picture selection**
    - ***Not testable with revised speech service/events implementation***
- **Remaining Tests**
  - **Testing Computer Vision component of checking drawing surface boundaries**
  - **Unit testing of SDK**
    - **PyTest**
  - **End-to-End Testing for error cases**

Goals Overview

PRIMARY

- **Development of Core SDK**
  - Speech Recognition Support Service
  - Motion Support Service
  - Vision Support Service
- **Development of Drawing Activity PoC**
  - Marker Detection & Collection
  - Marker Handling
  - Drawing Surface Detection
  - Drawing Specification Generator + Loader
- **Integrate Jetson Nano GPU with the NAO system**
- **Support Documentation**
  - qi/NAOqi Framework
  - Jetson Nano Integration
  - SIMYAN SDK

STRETCH

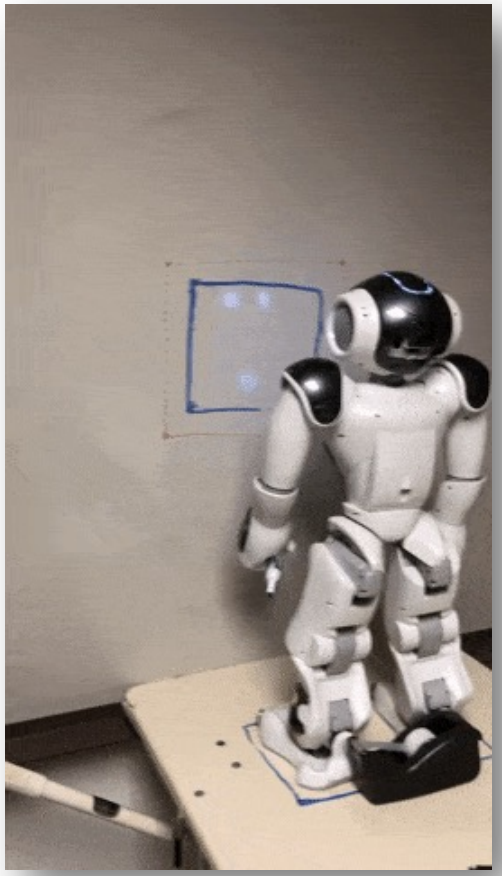
- **Machine Learning**
  - Computer Vision
  - Natural Language Processing

Development of Core SDK

- **SIMSpeech Service Module**
  - Manages speech events
  - Allows multiple simultaneous subscribers
- **SIMMotion Service Module**
  - Manages execution of motion sequences
  - Provides simplifying abstractions for generating motion sequences
- **SIMVision**
  - Allows the robot to detect a whiteboard and its boundaries
- **simutils Package**
  - Provides client-side support for utilizing SIMYAN services
  - Provides utilities for managing frequent NAO operations more simply and concisely

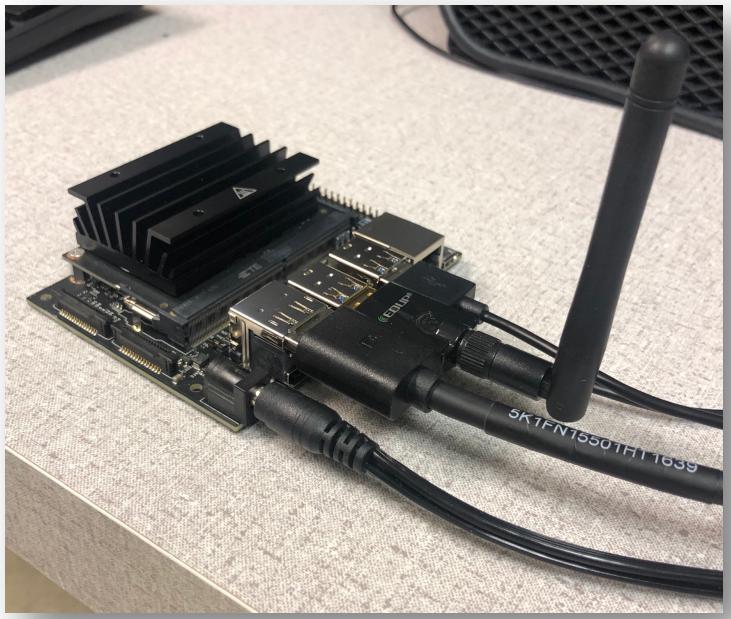
Development of Drawing Activity PoC

- **Marker Detection + Collection**
  - Insufficient time to implement
  - Chose to focus on more generally useful capabilities
- **Marker Handling**
  - Drawing demo exhibits NAO's ability to handle a marker for drawing on a whiteboard
- **Drawing Surface Detection**
  - Implemented the capability
  - Insufficient time to work into the drawing demo
- **Drawing Specification Generator + Loader**
  - Generator mostly complete - needs tuning of Hough transform parameters to achieve morphological closing
  - Loader for absolute motion sequences implemented for drawing demo



Integrate Jetson Nano with NAO

- **Jetson Nano setup**
  - OS install
  - Wireless Access Point for NAO to connect to
- **Jetson Nano running qi Framework**
  - Compiled following libraries from source
    - libqi
    - qilang
    - almath
  - All SIMYAN code can run entirely from the Nano
    - Takes workload away from NAO's CPU
    - Can utilize Jetson Nano hardware for ML or CPU-intensive apps
    - Much faster development turnaround - no need to deploy to robot
  - Opportunity to add peripherals
    - XBOX Kinect



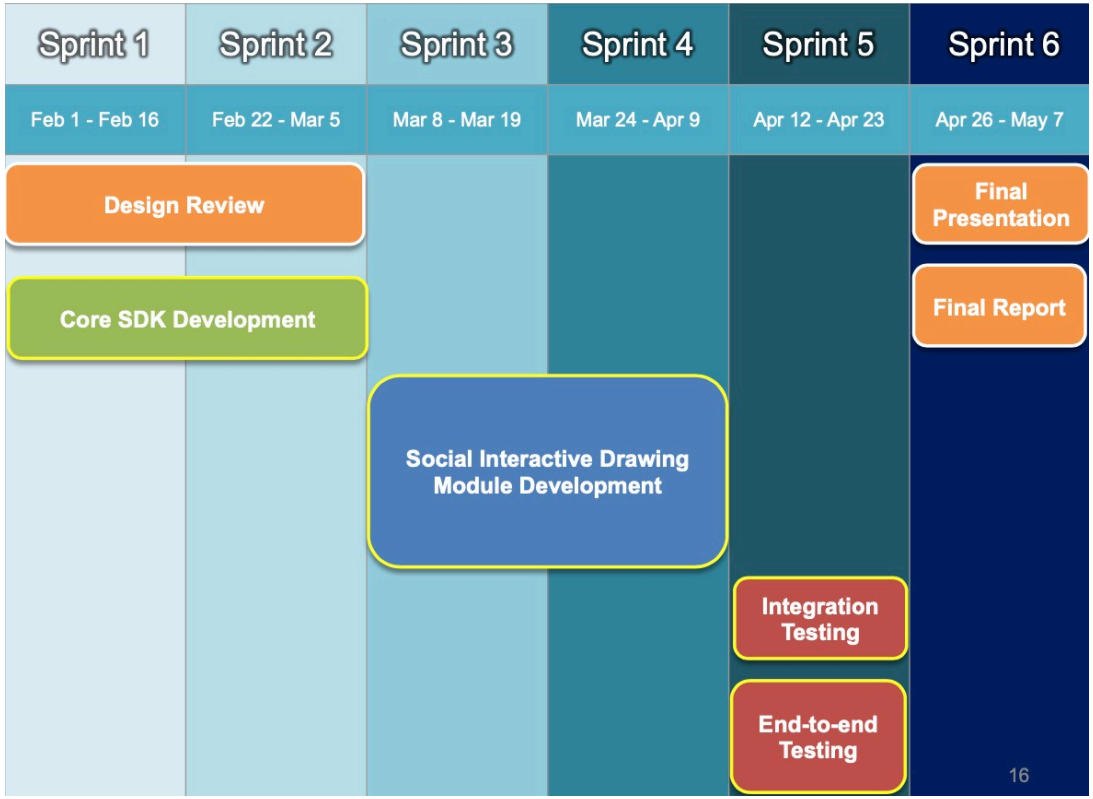
Support Documentation

- **Thorough in-code and auxiliary documentation for SIMYAN SDK**
- **Valuable documentation for understanding and working with NAO**
  - qi & NAOqi Framework(s)
  - NAO API Summaries
  - Tutorials
    - Python SDK Installation on Windows, Linux, Mac
    - Jetson Nano setup
    - Utilities: qibuild, qicli, robot-jumpstarter, studiotoolkit
  - Tips and Tricks
  - Code examples
  - Resource Links
  - Suggested Applications, Extensions, & Future Work
- **Easy-to-access GitHub Wiki**

Machine Learning

- **Computer Vision**
  - Had hoped to integrate XBOX Kinect and incorporate CV utilities but ran out of time
  - Utilizing OpenCV for Drawing Specification Generator
- **Natural Language Processing**
  - Not able to implement due to impediments
- **Future Opportunities**
  - Above capabilities would be fantastic opportunities to use or extend upon SIMYAN
  - Overhead of developing such activities significantly reduced by using SIMYAN
  - Jetson Nano provides excellent platform for hosting such applications

Project Schedule



- Frequently fell behind sprint schedule due to unforeseen impediments
- Elements of Core SDK and Activity Development pushed back multiple times
- Still able to adapt, be flexible, and deliver value because of sprint approach

Lessons Learned

- **Frequent team meetings are key**
- **Remote collaboration is challenging**
  - Best when kept short and focused
  - Work to ensure engagement
  - Clarify frequently to confirm understanding
- **When project risk increases unexpectedly, immediately begin evaluating the schedule to compensate**
- **Set personal and team SMART goals for every session**

Budget

Item	Quantity	Cost per Unit ($)
Jetson Nano GPU	1	0 (borrowed from advisor Bill Michael)
Realtek RTL8187L Chipset 2000mW Wireless USB Wifi Adapter 54Mbps Card	1	24
Xbox Kinect	1	50
Power Cord	1	7
Kinect Adapter	1	22
Rubber Washers	2	2.50
Total	**105.50**	
Allowance	**400**	

Conclusion

- **Our primary goal:**
  - Deliver a solution that is extensible and will expand the existing vision, speech recognition, and motion capabilities of the NAO robot
- **Our secondary goal:**
  - Use the SDK that we developed to enact a predefined drawing scenario
- **Objectives met:**
  - We have met our original goals to create an SDK to expand original capabilities of NAO and use the SDK to enact a drawing scenario

Future Opportunities

- **Integrate computer vision components**
  - Edge detection can be used to detect where to draw the edges of any object
    - ***Instruction generator based on edges***
- **Integrate feed from Xbox Kinect**
  - Use Xbox Kinect to get a higher quality view of what the NAO can see.
- **Using SIMYAN SDK in future efforts**
  - Using SIMYAN vision, speech recognition, and motion modules to perform additional tasks

Project Code and Documentation available on GitHub:
https://github.com/ancient-sentinel/UCCS-Social-Robotics

https://www.eduporium.com/media/catalog/product/cache/c687aa7517cf01e65c009f6943c2b1e9/n/a/nao-print_03_4.png

Any Questions?

University of Colorado
Colorado Springs

University of Colorado

Boulder | Colorado Springs | Denver | Anschutz Medical Campus