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

Andrew Nguyen, Bryce George, Colton Homuth, William Ross

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

January 29, 2021

UCCS University of Colorado Colorado Springs

University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus

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.

Social Interactivity for Youth with Autism using the NAO robot

- Core SDK
  - Supports advanced social interactivity behaviors
  - Framework for building Activity Modules
  - Orchestrates Activity Module execution
- Social Interactive Drawing Module
  - PoC Activity Module
  - Guides human subject through a drawing exercise
  - Incorporates social interaction

Presentation Overview

- **System Design Analysis**
  - Hardware System Designs 1 & 2
  - Hardware System Design Selection
  - Software System Designs 1 & 2
  - Software System Design Selection
- **Updates to Specifications & Requirements**
- **System Diagrams**
  - Block Diagrams
  - Software Diagrams
- **Test Plan**
  - Overview
  - Integration Tests
  - Demonstration Plan
- **Development Strategy**
- **Project Schedule**
  - Sprint Breakdown
  - Epics Breakdown
  - Board & Gantt Chart
- **Budget**
- **Knowledge Integration**

Hardware System Design 1

Use of NAO CPU solely for processing

- CPU vs. GPU for graphics processing
- Single core leaves no room for parallel processing*
- No need for external hardware

** Without an external GPU, the NAO robot will not be capable of handling any additional peripheral video/camera inputs.*

NAO Specs

- Camera captures 960p 30fps
- Single core ATOM CPU
- 1GB RAM**
- 2GB internal Flash memory + 8GB micro SDHC (Secure Digital High Capacity)

*** 1GB of RAM is widely considered bare-minimum for image processing and would be insufficient for any of our planned extensions*

Hardware System Design 2

Use of Nvidia Jetson Nano to do all real-time image processing

- GPU does processing in parallel, so much more efficient than normal CPU when it comes to image processing

Use of LiDAR depth sensor or Xbox Kinect

- Most of the stretch scenarios we had in mind will require better vision hardware

Jetson Nano Specs

- 128 core Maxwell GPU
- 4GB LPDDR4
- Quad core ARM CPU
- Encoding:
  - 4K @ 30fps
  - 4x 1080p @ 30fps
- Decoding
  - 2x 4K @ 30fps
  - 8x 1080p @ 30fps

NAO Specs

- Camera captures 960p 30fps
- Single core ATOM CPU
- 1GB RAM
- 10GB total storage
  - 2GB internal Flash memory
  - 8GB micro SDHC

Hardware System Design Selection

Selected Design 2, with GPU supplementing NAO hardware:

- Jetson Nano is capable of decoding images as fast as NAO camera captures
- Doing image processing with the external GPU frees up NAO CPU to focus on other tasks
- Quad core CPU will allow more parallel processing to take place
- Increase in RAM will also aide in image processing
- External hardware is very small and able to be connected to the NAO seamlessly
- Depth sensor gives freedom to explore more advanced visual systems (environment mapping, proximity detection, etc.)
- GPU will be able to handle increased resolution from depth sensor

Jetson Nano along with a depth sensor provides major improvements with very few drawbacks.

Software System Design 1

Using wrapper functions to include all functionality needed within our Core SDK

- Will increase the size of our Core SDK drastically with many modules that may not be useful outside of our project
- Will allow all code used in project to be in one place
- Ability to adjust NAOqi capabilities to fit our need
- Could condense and improve readability of code inside the drawing modules

More modular design, may improve readability and condense the code inside the drawing module.

Software System Design 2

Using a mix of new functionality from our core SDK and existing functionality from NAOqi framework

- Less modular design, could lead to longer blocks of code within the drawing modules
- Utilizing NAOqi framework directly will decrease the workload
- Keeping Core SDK simple could lead to less confusion when being utilized in the future

Overall, less modular design and may hurt readability, but this will give more time to focus on essential functionalities.

Software System Design Selection

Selecting the second software system design including accessing both the Python SDK and our Core SDK in the drawing modules

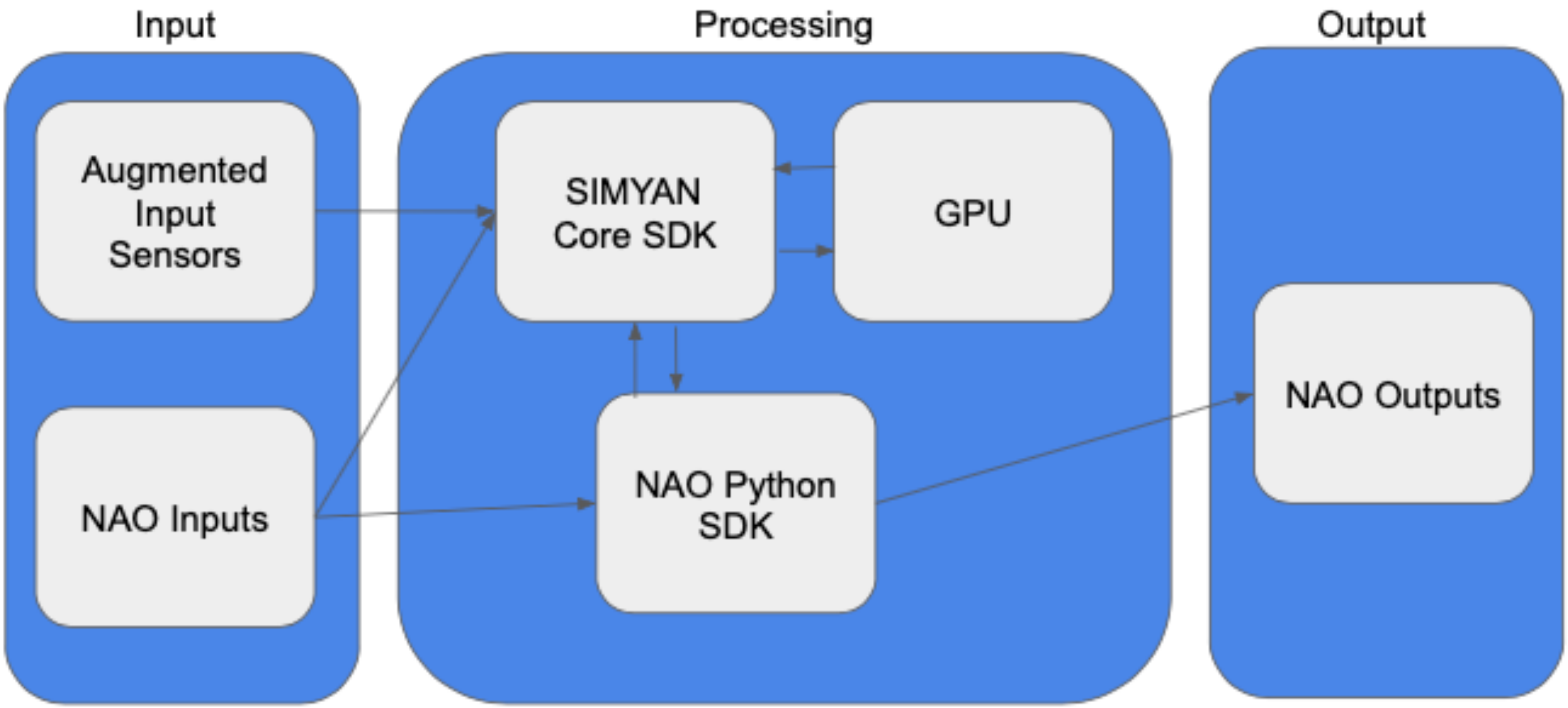
- Gives us freedom to leverage existing Python SDK when needed without adding an extra layer of abstraction
- Allows Core SDK to fully supplement Python SDK focusing on our specific contributions
- Allows more time focusing on core development rather than just wrapping existing modules

This strategy will allow us to more heavily focus on developing unique innovations for our Core SDK.

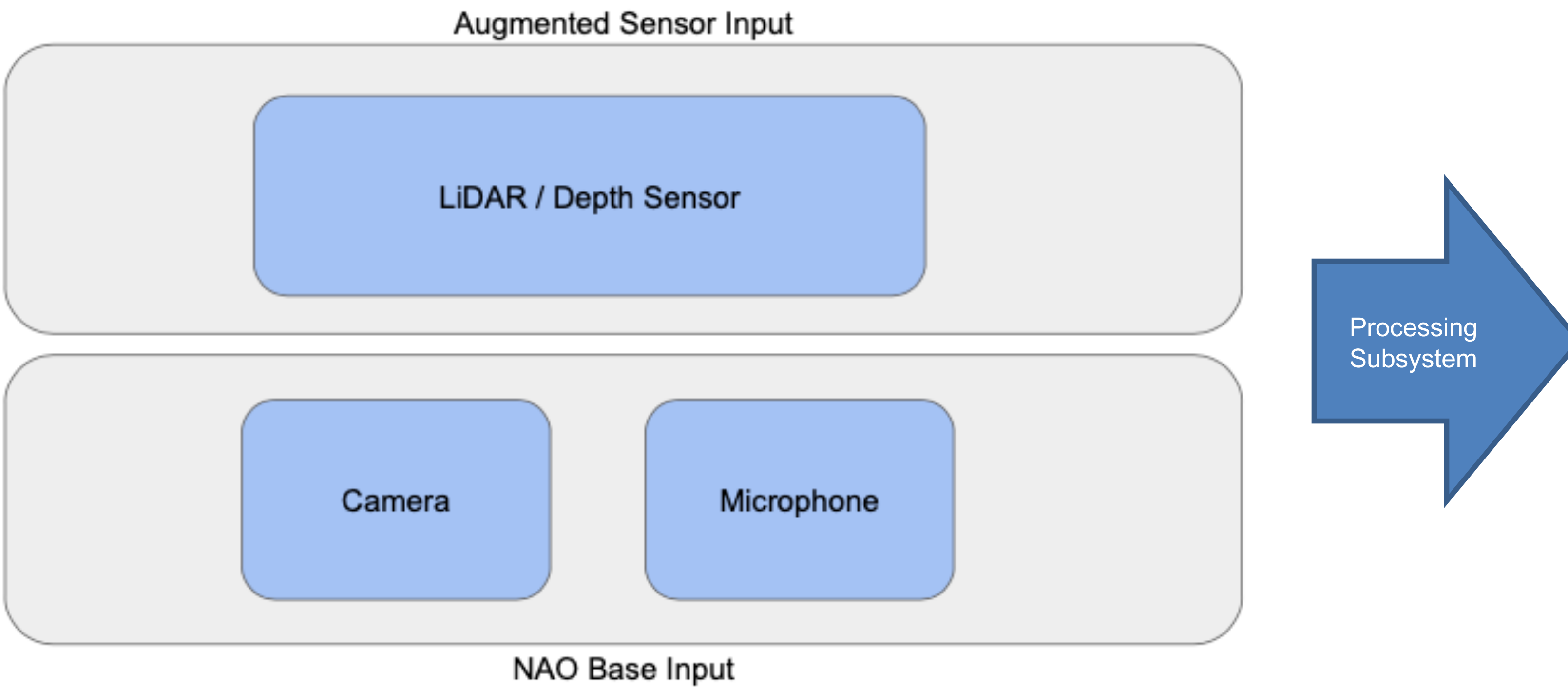
Specification & Requirement Updates

Requirements	Specifications
The robot shall be able to recognize and handle a dry erase marker of radius 0.5-0.75” and weighing no more than 3oz. The robot shall be able to recognize a drawing surface on a dry erase whiteboard. The robot shall have a mechanism for attaching the Jetson Nano GPU to it without impeding its movements.	Jetson Nano GPU Attachment Capacity Dimensions: 100mm x 80mm x 29mm

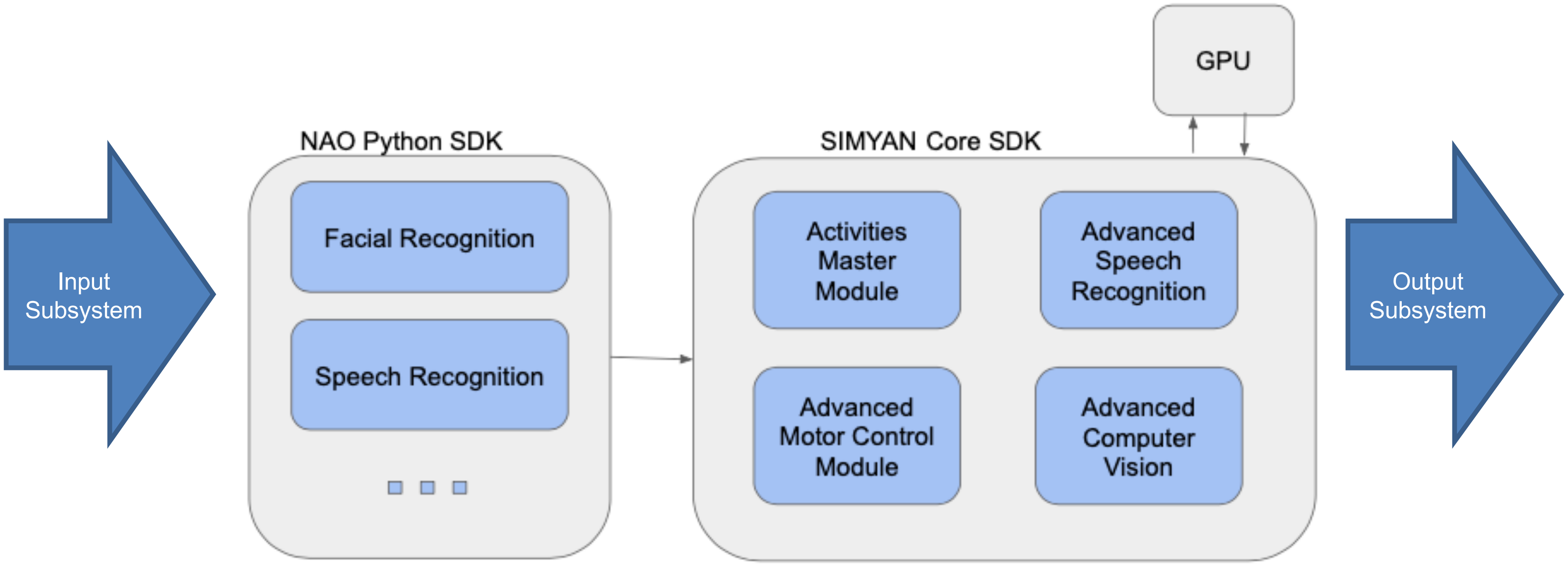
System Diagram



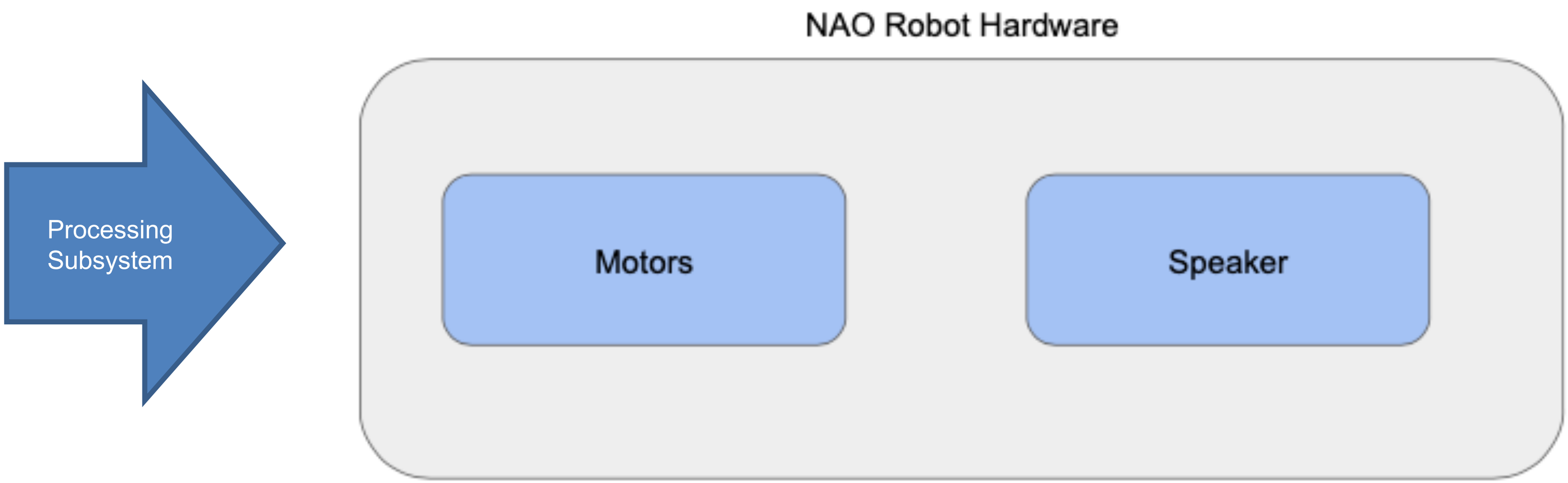
Input Subsystem



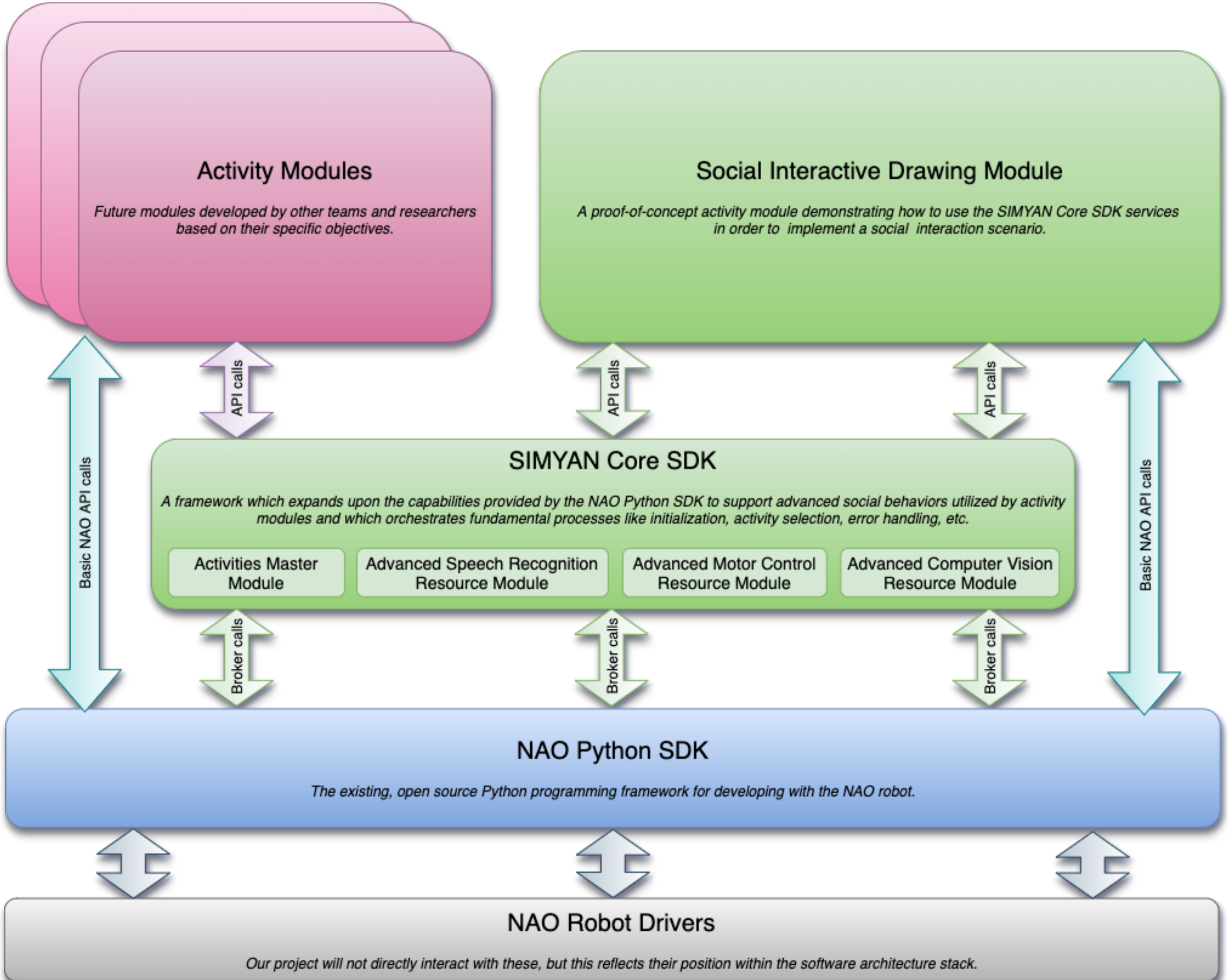
Processing Subsystem



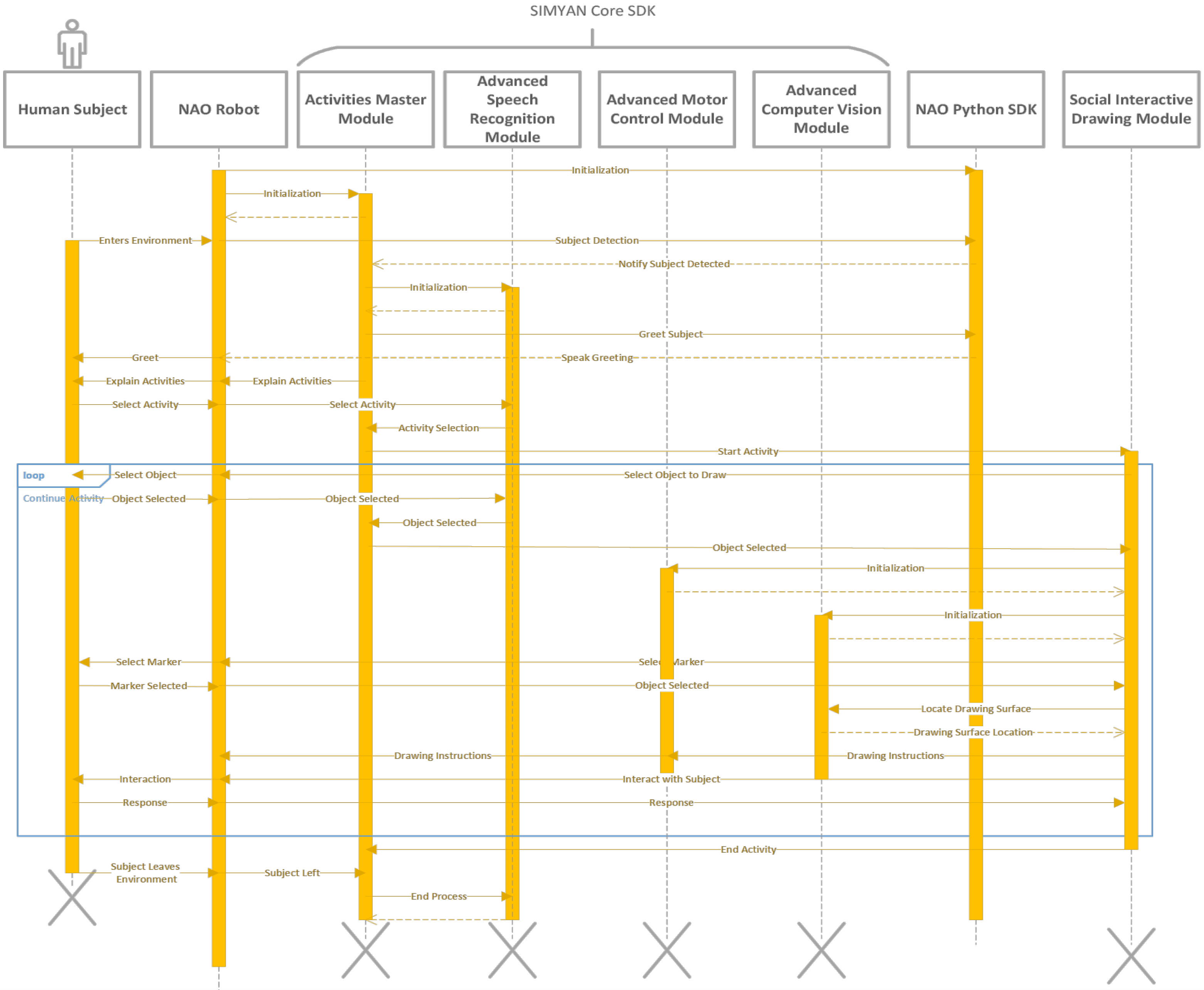
Output Subsystem



Software Architecture Stack



Sequence Diagram



Test Plan: Overview

1. **Unit Testing**

   Ensuring that unit tests are:

   - ***thorough*** *(70-100% coverage)*
   - ***passing*** *(100% passing)*
   - ***complete*** *(pass and fail test cases for sufficient range of inputs)*

2. **Integration Testing**
3. **Demonstration Plan**

Integration Tests

Test inter-module connections in the code:

- Speech Recognition to Drawing Module Integration
  - Object Selection
  - Color Selection
  - Social Interactions
- Motor Control Module to Drawing Module Integration
  - Collect Marker
  - Draw on Drawing Surface
- Vision Module to Drawing Module Integration
  - Recognize Subject
  - Recognize Drawing Surface
  - Recognize Drawing Implements

Integration Test definitions include:

- ✓ Preconditions
- ✓ Scenario
- ✓ Testing Steps
- ✓ Passing Results / Criteria

Demonstration Plan

Flow of events:

- **Default Scenario**
  - Robot draws a picture with the human subject and engages in expected social behaviors
  - Highlights the fulfillment of the MVP and any completed extensions
- **Exceptional Cases**
  - Subject gives invalid command to the robot
  - Subject selects invalid object to draw
  - Marker has a cap on
- **Delivery**
  - Pre-recorded video
  - Brief code overview
    - ***Diagrams***
    - ***Pseudocode***

Demonstrate all use cases:

- ✓ Load Master Module
- ✓ Initialize Activities Master Module
- ✓ Start Activity
- ✓ Select Object to Draw
- ✓ Obtain Writing Implement
- ✓ Draw Object
- ✓ Subject Interaction

Development Strategy

Agile Development	Test Driven Development	Paired-Programming
• Iterative development cycle ➢ Plan ➢ Develop ➢ Test ➢ Deploy ➢ Retrospective • Focus on delivering working code each sprint	• Unit testing • Define tests before writing code • Integration test cases • Write just enough code to pass the tests	• Task owner paired with assisting partner • Develop together whenever possible • Partner stays updated on development progress • Partner assists with code Review/QA

Sprint Breakdown

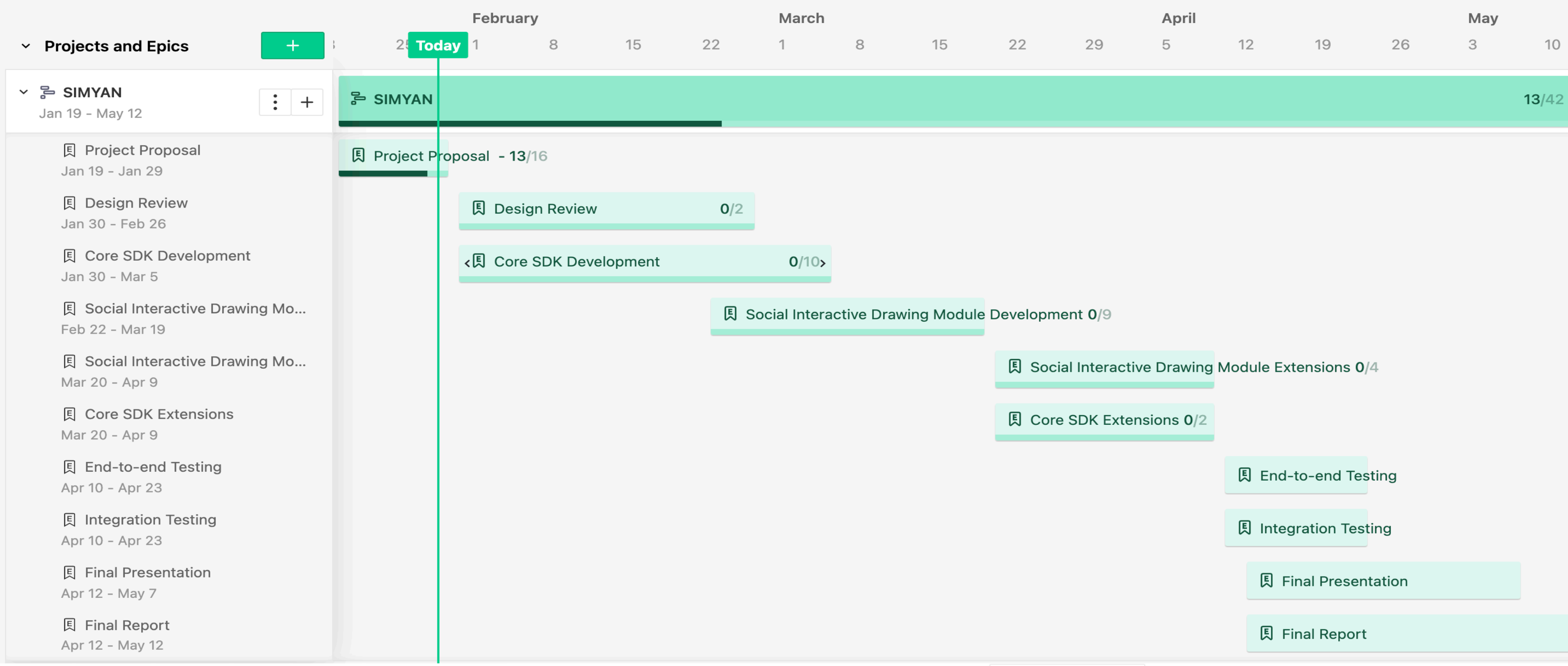
<table>
<tr><th>Sprint 1</th><th>Sprint 2</th><th>Sprint 3</th><th>Sprint 4</th><th>Sprint 5</th><th>Sprint 6</th></tr>
<tr><td>Feb 1 - Feb 16</td><td>Feb 22 - Mar 5</td><td>Mar 8 - Mar 19</td><td>Mar 24 - Apr 9</td><td>Apr 12 - Apr 23</td><td>Apr 26 - May 7</td></tr>
<tr><td colspan="2">Design Review</td><td></td><td></td><td></td><td>Final Presentation</td></tr>
<tr><td colspan="2">Core SDK Development</td><td></td><td>Core SDK Extensions</td><td></td><td>Final Report</td></tr>
<tr><td></td><td colspan="2">Social Interactive Drawing Module Development</td><td>Social Interactive Drawing Module Extensions</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>Integration Testing</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>End-to-end Testing</td><td></td></tr>
</table>

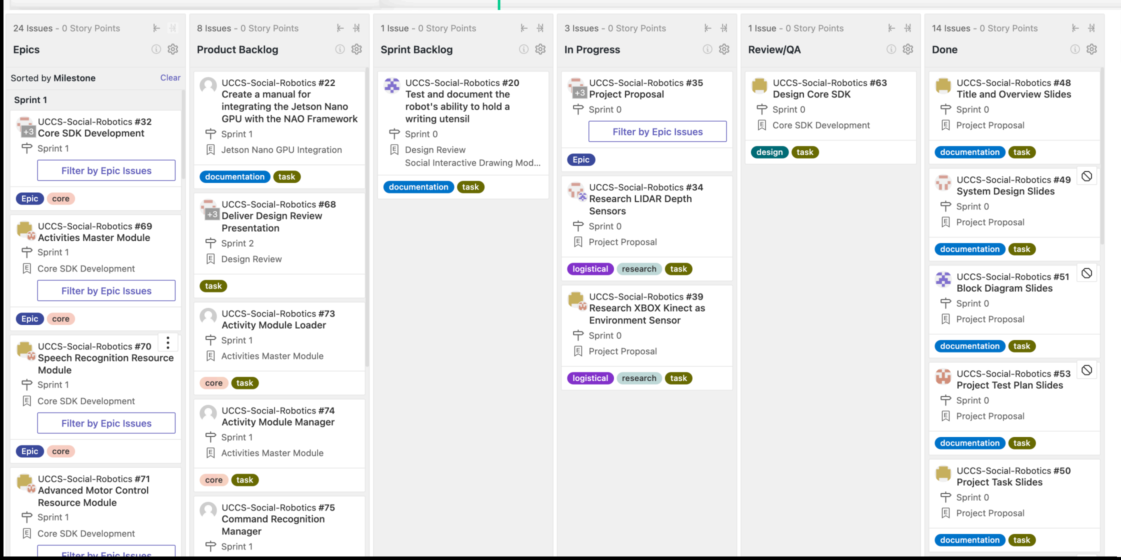
Epics Breakdown

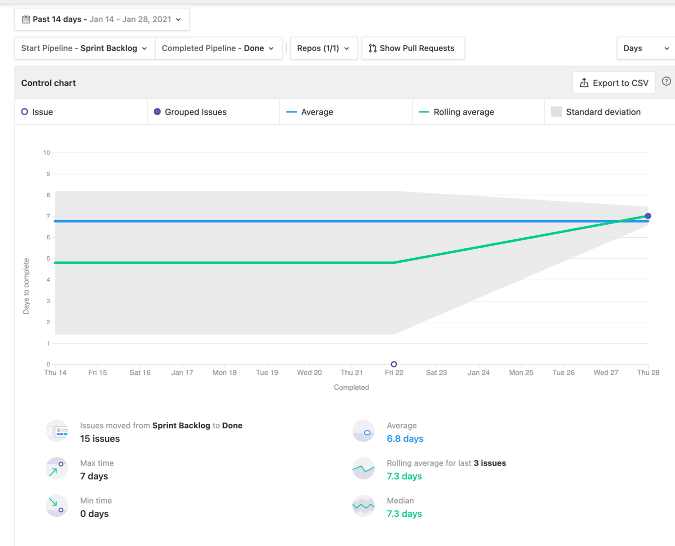
Core SDK Development	Core SDK Extensions	Social Interactive Drawing Module Development	Social Interactive Drawing Module Extensions
Activities Master Module B ➢ Activity Module Loader ➢ Activity Module Manager Command ➢ Recognition Manager **Speech Recognition Resource Module** A **Advanced Motor Control Resource Module** AB **Advanced Computer Vision Resource Module** CW ➢ Jetson Nano GPU Integration C	**Advanced Computer Vision Resource Module Extensions** CW **Advanced Motor Control Resource Module Extensions** AB	**Drawing Activity Command Recognition** A **Marker Detection** C **Marker Collection** W **Marker Handling** B **Drawing Surface Detection** C **Drawable Object Loader** B **Object Drawing Manager** CW **Social Interaction Manager** AB	**Self-Directed Marker Collection** AW **Self-Directed Marker Selection** AW **Advanced Drawing Selection** AB **Automated Object Drawing Specification Generator** BC

Integration Testing	End-to-end Testing	Design Review	Final Presentation	Final Report	LEGEND
Run Integration Test Cases * **Fix Failures** * **Re-Test Failed Cases** *	**Run End-to-end Test Cases** * **Fix Failures** * **Re-Test Failed Cases** *	**Develop System Design** * **Prepare Design Review Presentation** * **Deliver Design Review Presentation** *	**Prepare Final Presentation** * **Prepare Demonstration** * **Deliver Final Presentation** *	**Prepare Final Report** * **Deliver Final Report** *	*** – All** **A – Andrew** **B – Bryce** **C – Colton** **W – Will** *Superscript = Primary Owner(s)*

Board & Gantt Chart







Budget

Items:

- Jetson Nano GPU
- 3D printed materials: utensil holder, backpack (for GPU)
- Depth Sensor (LiDAR/Infrared)

Cost allocation:

- Jetson Nano GPU - free, borrowed from advisor Bill Michael
- Depth Sensor ($300 maximum)
- Rest of materials ($100 maximum)

***Funds will be acquired from department allowance of $100 per person**

Knowledge Integration

- **Agile Development**
- **Test Driven Development**
- **System Design and Analysis**
  - Problem Decomposition
  - Requirement Identification
  - Architecture
  - Systems Engineering
- **Machine Learning**
  - Computer Vision
  - Natural Language Processing

Conclusion

Goals:

- ✓ **To develop our own SDK for the NAO robot that will include support for advanced social behaviors and complex motor skills.**

- ✓ **Secondarily, is to create a robot guided scenario to promote social interaction with ASD children.**

- ✓ **Produce a quality product that can be built upon with further research.**

Any Questions?

University of Colorado
Colorado Springs

University of Colorado

Boulder | Colorado Springs | Denver | Anschutz Medical Campus