DyNAMO Project Structure

1. Project Methodology

The DyNAMO project operates on an agile learning framework designed to foster rapid technical exploration.

- **Sprint Cycle:** 2 Weeks.
- **Meeting Structure:** Sub-teams function as independent learning units with their own stand-ups and technical syncs. The entire cohort convenes once per sprint for a **Knowledge Share**.
- **Knowledge Share Format:** Instead of a strictly integrated product demo, teams present their research progress, prototypes, and "lessons learned" (e.g., simulation experiments, model training results, or subsystem proofs-of-concept).
- **Primary Domain:** Development and validation occur within a **Simulation-First** workflow using **NVIDIA Isaac Lab** to allow for safe, rapid iteration.

2. Grading & Evaluation Philosophy

Context: DyNAMO is an ambitious, high-complexity research initiative. Because we are tackling undefined problems rather than following a fixed tutorial, our assessment strategy focuses on the **quality of the research journey** rather than just the final product.

This project brings together a diverse set of **Bachelor’s students, Master’s students, and Working Students**. Consequently, assessment is not one-size-fits-all; expectations are scaled according to academic level and role.

The Core Principle: Process > Product While a functional demo is the objective, the primary grade driver is the scientific method: formulating hypotheses, experimenting, failing, documenting *why* it failed, and iterating.

Evaluation Criteria:

- **Visible Effort:** Regular demos, active experimentation, and a clear trail of research. A broken robot with excellent "failure analysis" documentation is worth more than a working robot with no explanation.

- **AI Integrity & Understanding:** Usage of AI tools (ChatGPT, Claude, Gemini) is permitted and encouraged. However, usage will be spot-checked. If it is determined that a student does not understand "their own code" or design concepts during a review, it will be **heavily penalized**. We grade your understanding, not the AI's.
- **Ambition:** Willingness to tackle hard, undefined problems rather than choosing the path of least resistance.
- **Activity:** Consistent participation in stand-ups, technical syncs, and knowledge shares.
- **Team Spirit:** Unblocking teammates, sharing useful scripts, and contributing to the shared knowledge base.

3. Architecture & Decoupling Strategy

To maximize individual learning velocity, we have decoupled the development streams. This allows students to focus on deep-diving into their specific domains without being blocked by dependencies in other areas.

- **Physical Decoupling:** The system divides into the **Mobile Base** (Logistics) and the **Manipulator** (Dexterity).
- **Interface Contracts:** Teams design against agreed-upon interfaces. This teaches the importance of defining clear system boundaries and API contracts early in the engineering process.
- **Shared Reality:** Team C provides the unifying element, the Simulation Environment. All teams learn to operate within the same USD (Universal Scene Description) stages, ensuring their independent work is compatible by design.

4. Team Composition & Learning Goals

Team A: Mobile Navigation & Logistics

- **Headcount:** 2-3 Students
- **Core Focus:** Mastering autonomous mobility concepts, path planning, and mobile robot kinematics within the **ROS 2** ecosystem.
- **Learning Goal:** Explore the complexities of navigating a mobile robot through dynamic environments. The team will investigate how to map a space and reliably position a robot base, treating the interaction with the humanoid as a dynamic docking problem.
- **Key Deliverables:**

    - Implementation of the **ROS 2 Navigation (Nav2)** stack for production-grade navigation.
    - Exploration of visual servoing algorithms for precision alignment.
    - Development of dynamic obstacle avoidance logic.
- **Capstone Goal:** Develop **Human-Aware Pathing** logic. Instead of treating humans merely as static obstacles to be avoided, the system should navigate socially (e.g., respecting personal space), ensuring safe and polite interaction in a shared workspace.

Relevant Links:

- Nav2 Documentation
- Clearpath Ridgeback Manual

Team B: Advanced Manipulation

- **Headcount:** Nx2 Students
- **Core Focus:** Investigating and comparing state-of-the-art AI approaches for robotic manipulation.
- **Learning Goal:** Conduct a comparative study of modern AI paradigms to solve complex manipulation tasks (e.g., "Grasp the drill and place it in the crate"). This team operates as an internal research lab to understand the strengths and weaknesses of different policy learning methods.
- **Research Tracks:**
    - **Foundation Models (NVIDIA):** Exploring large-scale pre-trained models, specifically **Project GR00T**, for zero-shot instruction following and generalization.
    - **Imitation Learning (Open Source):** Investigating data-driven approaches using frameworks like **LeRobot**. The team will study algorithms like **ACT** (Action Chunking Transformer) and Diffusion Policy, potentially utilizing **XR Teleoperation** for collecting human demonstration data.
    - **Vision-Language-Action (VLA):** Studying how models can connect natural language processing directly to robotic actions, utilizing open-source architectures like **OpenVLA**.
    - **Embodied Reasoning (Google):** Utilizing the **Gemini API** for high-level logic and planning. This track explores **"Code-as-Policies"**, where the multimodal model analyzes the scene and writes executable Python code to coordinate actions, effectively treating the robot as an API endpoint.

- **Capstone Goal:** Achieve a robust **Multi-Object Pick and Place** sequence. The G1 humanoid must successfully identify, grasp, and place multiple distinct objects onto the Ridgeback mobile platform.

Relevant Links:

- Unitree HuggingFace
- Project GR00T
- LeRobot
- Google DeepMind Robotics

Team C: Infrastructure & Sim

- **Headcount:** 2 Developers
- **Core Focus:** Understanding simulation architecture, USD workflows, and synthetic data generation.
- **Learning Goal:** Build a deep technical understanding of the "Matrix" in which the robots operate. This team explores how to architect scalable simulation environments and leverages procedural generation to create training data, rather than relying on manual effort.
- **Key Deliverables:**
    - **USD Architecture:** Designing modular environments using Universal Scene Description concepts (Layers, References, Variants) within the **Isaac Sim/Lab** ecosystem.
    - **Synthetic Data Generation:** Utilizing tools like **Isaac Replicator** to programmatically generate and randomize training datasets.
    - **Physics Integration:** Learning to import and configure digital twins with accurate collision and mass properties for realistic simulation.
- **Capstone Goal:** Construct the **Integrated Test Env.** This final environment must feature the Ridgeback and G1 robots operating simultaneously amidst multiple tables and animated human characters, serving as the unified stage for the project's final demo.

Relevant Links:

- NVIDIA Isaac Sim 5.1 Documentation
- Universal Scene Description (USD) Guide
- Isaac Replicator (Synthetic Data Generation)
- NVIDIA Omniverse Platform

More links which may become relevant as the project progresses:

- **Third-Party SimReady USD Assets**
  - Lists third-party open-source assets compatible with Isaac Sim, utilizing the Universal Scene Description (USD) format for seamless integration into simulation workflows.
- **NVIDIA Isaac Lab-Arena Blog**
  - Introduces Isaac Lab-Arena, an open-source framework co-developed with Lightwheel for scalable robot policy evaluation.
  - Highlights features for simplified task curation, automated diversification, and large-scale parallel benchmarking to assess generalist robot policies.
- **NVIDIA Isaac Lab-Arena Documentation**
  - Technical documentation for the Isaac Lab-Arena framework (v0.1.1), detailing installation, usage, and APIs for creating and evaluating diverse robotic tasks in simulation.
- **LightWheel AI**
  - An Embodied AI infrastructure company providing SimReady assets ("The World Layer"), synthetic and egocentric human data via EgoSuite ("The Behavior Layer"), and evaluation platforms like RoboFinals ("The Evaluation Layer").
- **Building Synthetic Data Pipelines with Isaac Sim**
  - Describes how to build and orchestrate end-to-end Synthetic Data Generation (SDG) workflows using NVIDIA Isaac Sim and NVIDIA OSMO.
  - Covers creating 3D digital twins with NuRec, populating scenes with SimReady assets, and generating data with MobilityGen.
- **NVIDIA NuRec Documentation**
  - Documentation for NVIDIA Neural Reconstruction (NuRec), a set of APIs and tools for creating high-fidelity 3D digital twins from real-world sensor data (camera and LiDAR) for use in simulation.
- **Isaac Automator GitHub**
  - A tool for deploying Isaac Sim and Isaac Lab to public clouds (AWS, Azure, GCP, Alibaba Cloud).
  - Provides scripts and container build instructions to set up fully configured remote desktop cloud workstations for robotics development.
- **NVIDIA OSMO**

    - An open-source, cloud-native orchestration framework for defining, scheduling, and monitoring multi-stage physical AI workflows (data generation, training, reinforcement learning) across heterogeneous compute environments.
- **NVIDIA TensorRT Edge-LLM Blog**
    - Introduces TensorRT Edge-LLM, an optimized open-source C++ framework for running Large Language Models (LLMs) and Vision Language Models (VLMs) on embedded robotics and automotive platforms (e.g., NVIDIA Jetson Thor, DRIVE AGX Thor).
- https://github.com/nvidia-isaac/WBC-AGILE