SPRING 2023 SENIOR PROJECT

Quadruped Robot

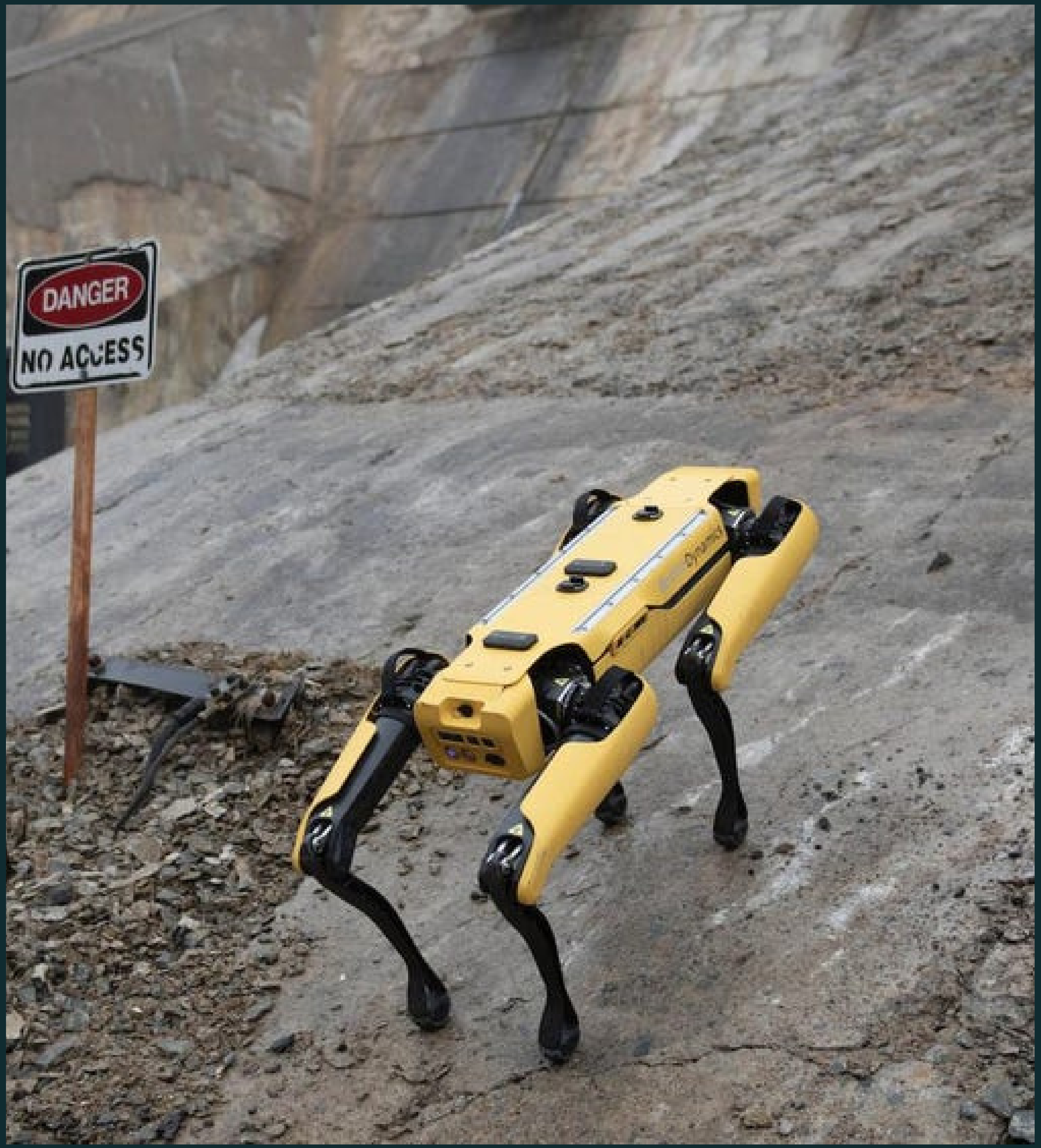
Ryunghoon Ahn, Christian Valdes, Samuel Lee

University of Texas Rio Grande Valley
Computer Science Department
Artificial Intelligence Lab
Supervised by: Dr. Dongchul Kim
December 2022 - May 2023

Quadruped Use Cases

Quadruped robots have versatile applications due to their inherent stability, ability to traverse various terrains, and adaptability to different tasks. Some use cases for quadruped employment include:

- Search and rescue
- Military and surveillance
- Inspection and maintenance



Core Elements of the Project

Hardware: Quadruped Robot

What is a quadruped?

- Quadrupedal locomotion is an important characteristic that has evolved in many animal species to provide stability, speed, and adaptability in various environments.

Why legs over wheels?

- Biomimicry
  - Terrain Adaptability
  - Balance and Stability
- Obstacle Negotiation

Software: Reinforcement Learning

What is reinforcement learning?

- Type of machine learning where an agent learns to make decisions by interacting with an environment. The goal of the agent is to learn a policy or strategy that maximizes a cumulative reward over time.

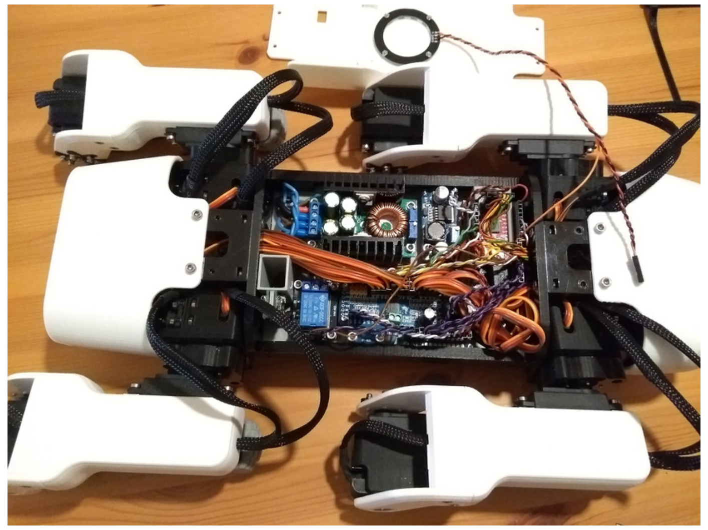
How does RL training benefit the quadruped?

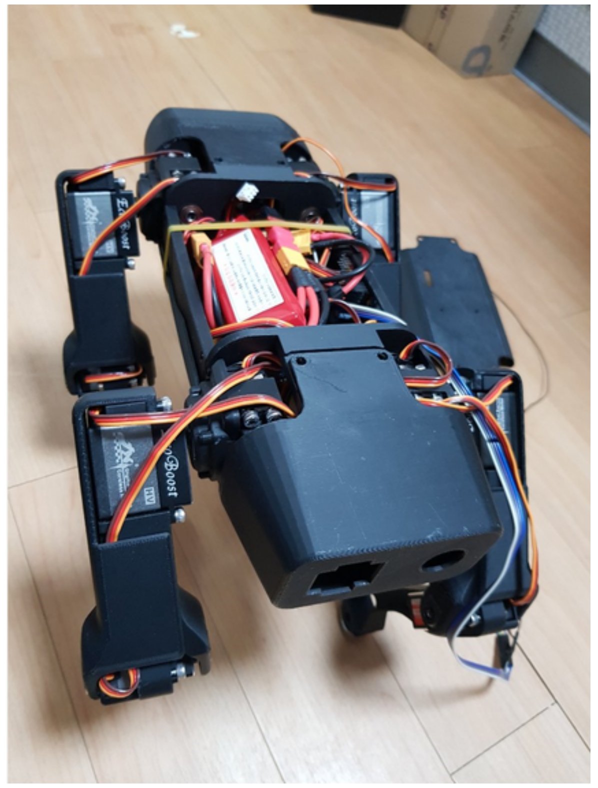
- Biomimicry
- Obstacle Negotiation

How we got here

How it
all began

As of
today





Where it all began

DIY Hobby Servos Quadruped Robot

By: Miguel Ayuso Parrilla

The project presents an affordable and accessible way for hobbyists and students to build and experiment with a custom quadruped robot using 3D printed parts, hobby servos, and a Raspberry Pi based control system.

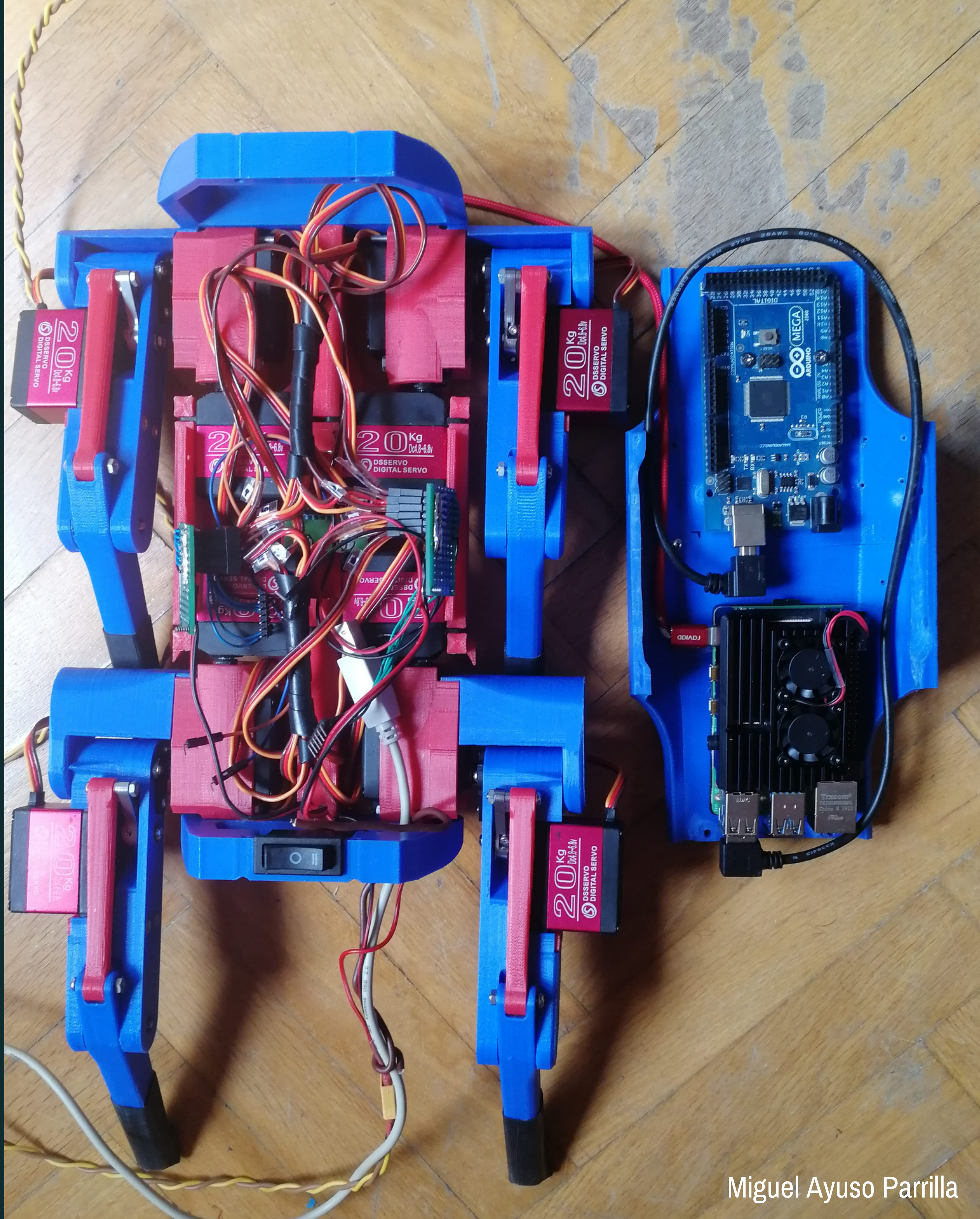
Project Highlights

- Custom Design
- Inverse Kinematics
- Pybullet simulation environment

Why we didn't pursue further

- Documentation
- Space restricted

https://hackaday.io/project/171456-diy-hobby-servos-quadruped-robot




FlorianWilk


Knosterpitter


Deok-Yeon Kim

A new starting line

SpotMicroAI

By: Deok-Yeon Kim

SpotMicroAI is an open-source, affordable, and accessible quadruped robot project inspired by Boston Dynamics' Spot robot. Utilizes 3D printed parts, hobby servos, a Raspberry Pi, and custom software. The project offers slightly more intuitive documentation and a community to guide through the building and programming process.

Project Highlights

- Open-source design
- Community-Driven
- Pybullet simulation environment

Why we didn't pursue further

- Documentation

https://spotmicroai.readthedocs.io/en/latest/

The reference

SpotMicroESP32
By: Michael Kubina

The SpotMicro project uses ESP32 microcontroller for onboard computing. This is an alternative to Raspberry Pi and the Arduino. This was used as a reference for the physical component due to the documentation on assembly and electronic schematics.

Project Highlights

- Modular Design
- Active community

Why we didn't pursue further

- ESP32-based

https://github.com/michaelkubina/SpotMicroESP32



Desirable outcome

Spot Mini Mini OpenAI Gym Environment

By: Marucie Rahme, Ian Abraham, Matthew Elwin, and Todd Murphey

This version of the SpotMicro project redesigns the chassis and is the closest in AI-implementation.

Project Highlights

- Modular Gait
- OpenAI Gym Environment

Why we didn't pursue further

- Processing done off-board
- Documentation

https://github.com/michaelkubina/SpotMicroESP32

Taking a step back

Rex: an open-source quadruped robot

By: Nicola Russo

Open-source project focused on developing an advanced quadruped robot simulation environment based on OpenAI Gym, PyBullet, and TensorFlow. The project aims to facilitate the development of advanced control policies and behaviors for quadruped robots and offers extensive documentation and examples.

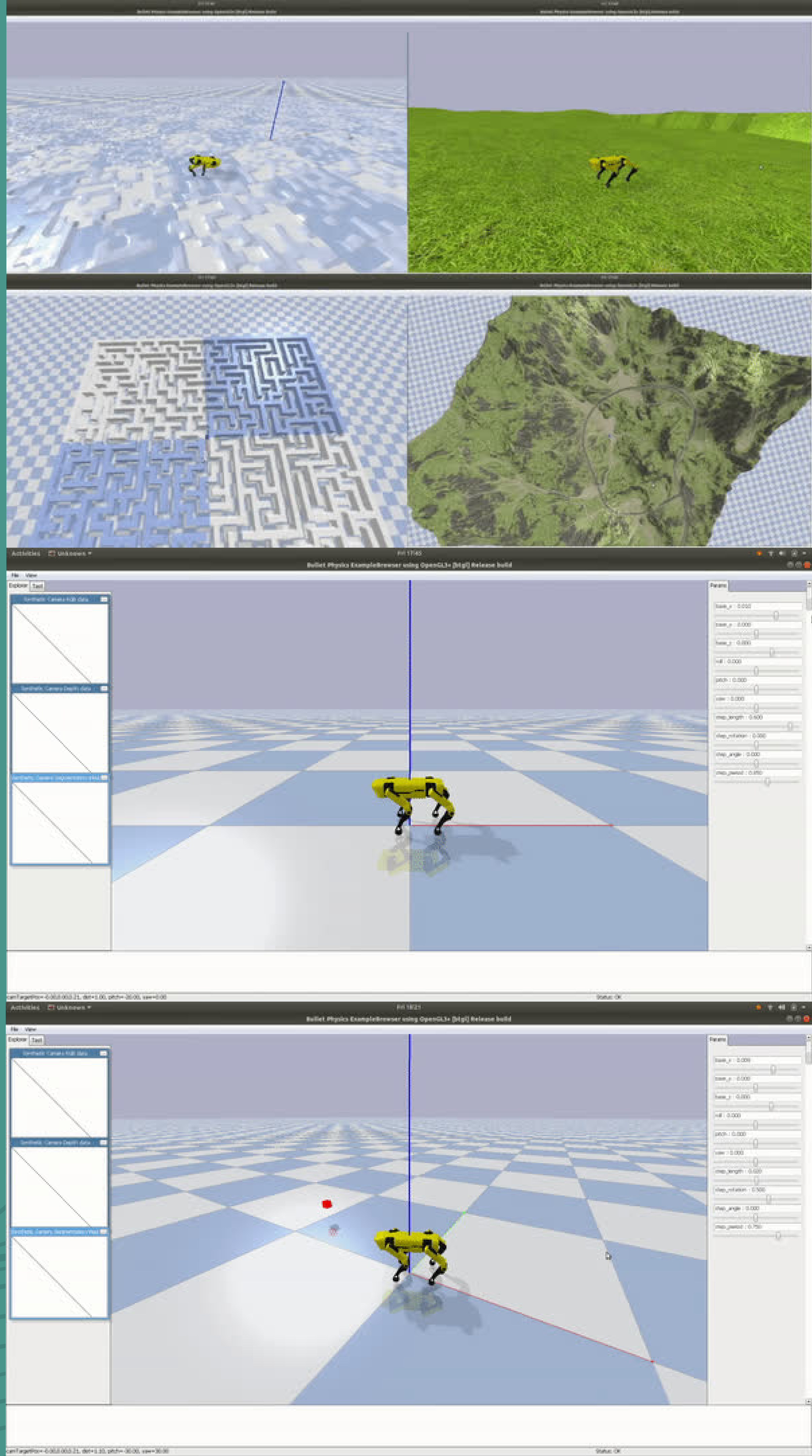
Project Highlights

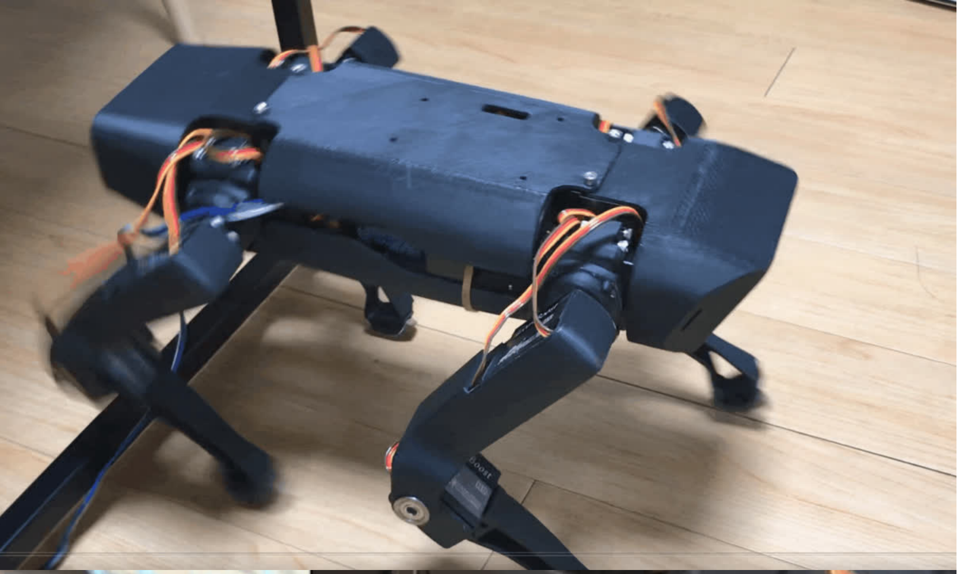
- Pybullet simulation environment
- Proximal Policy Optimization (PPO)

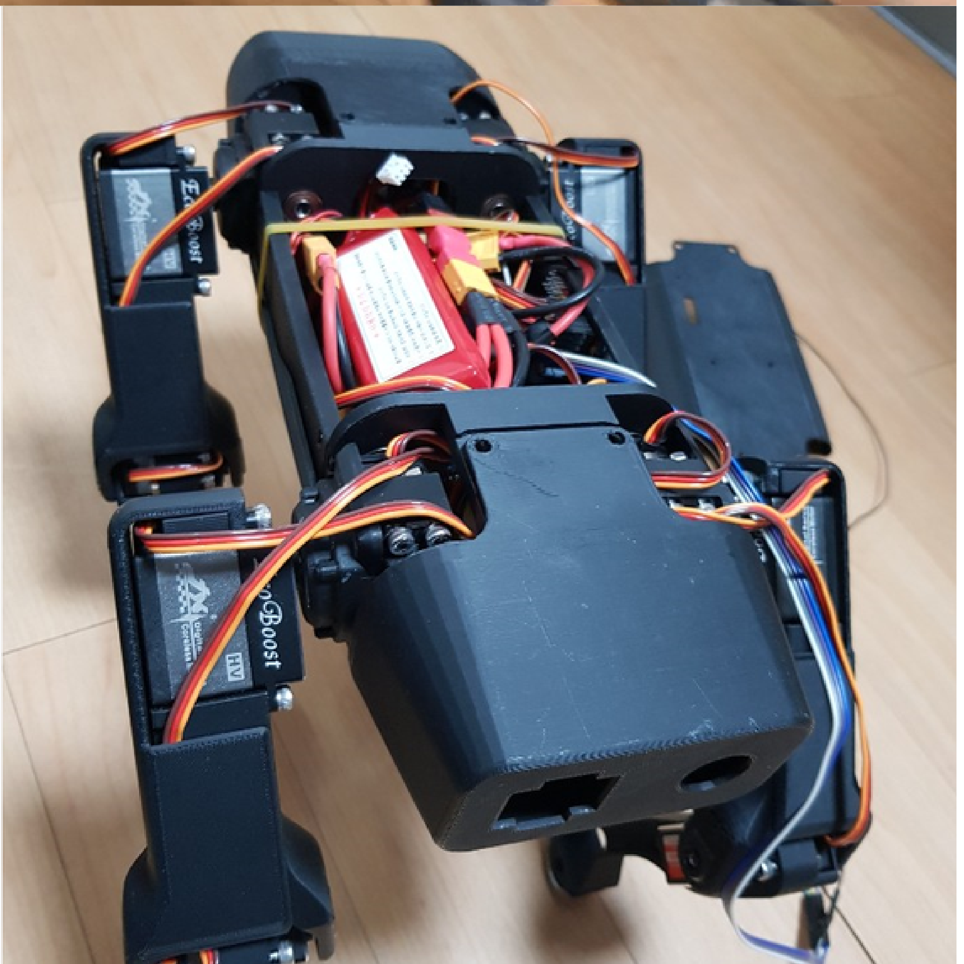
Why we didn't pursue further

- Documentation
- No ROS

https://github.com/nicrusso7/rex-gym







Full of Promise

SpotMicroAI - Road Balance version

By: Hanyang University Road Balance Team

This SpotMicro project uses NVIDIA Jetson Nano, a powerful and compact GPU-accelerated computing platform, for onboard computing and control. The use of the NVIDIA Jetson Nano enables more advanced tasks, such as machine learning and computer vision, while ROS integration allows for additional flexibility and compatibility with a wide range of robotic tools and libraries.

Project Highlights

- Utilizes Jetson Nano
- Extensive Documentation

Problems we ran into

- Documentation written in Korean
- Most processing run offboard

https://github.com/nicrusso7/rex-gym

Basic List of Materials

Compared to other quadrupeds currently available on the market, the model we selected was specifically engineered to be cost-effective while still maintaining the necessary processing power for simulating and executing movements.

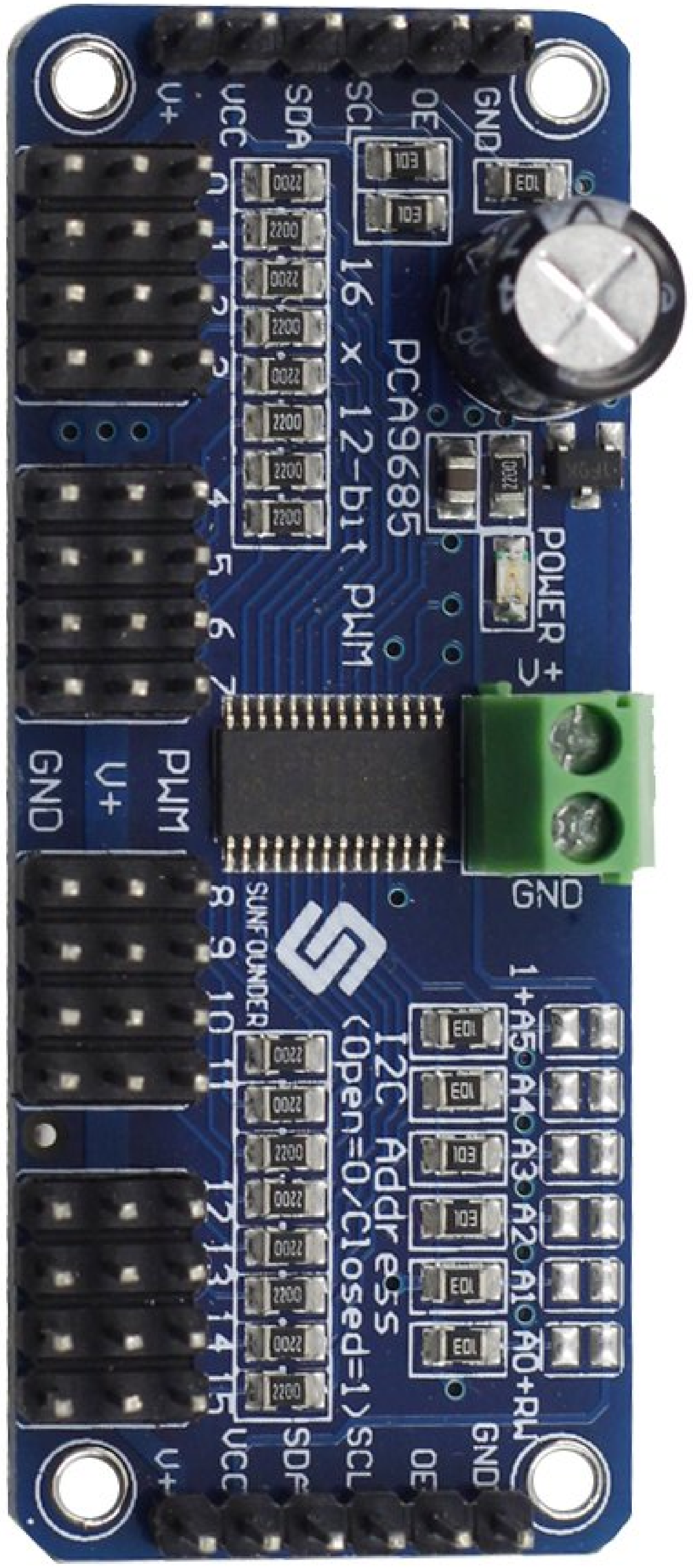
Jetson Nano
CLS6336HV Servos
PCA9685
7.4 6200mAh Lipo Battery
PLA Plastic

Jetson Nano

- Single-Board Computer with GPU-CPU Combo
- Capable of real-time object recognition, path planning, and adaptive motion control
- Highly capable and versatile robotic platform

CLS6336HV Servos

- High torque output. Capable of up to 36kg/cm of torque at 8.4V
- Precision metal gear train
- High quality bearings
- Coreless motor
- Reduced noise and improved efficiency compared to other servos on the market



PCA9685

- 16 Channel 12-bit PWM Controller
- Controls multiple servo motors simultaneously
- Compatible with microcontrollers such as the Jetson Nano using the I2C communication protocol
- Efficiently manages servo motor positions and movements, reducing the workload on the main microcontroller.

7.4 6200mAh Lipo Battery

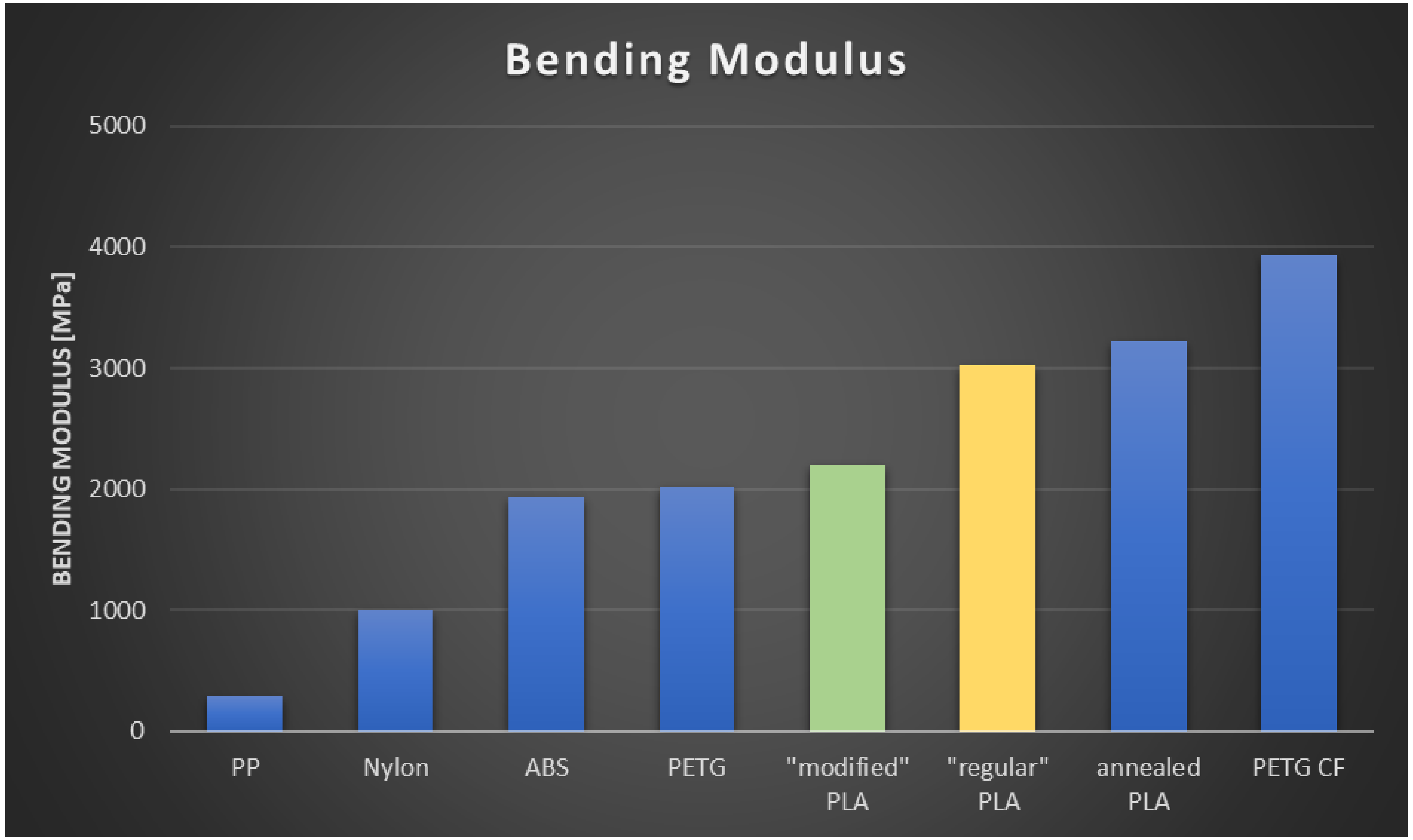
- Rechargeable battery that is commonly used in RC (radio-controlled) vehicles, drones, and other similar applications that require a high-capacity, lightweight power source.

- thinner, lighter, and more flexible than traditional rechargeable batteries.

- High discharge rates, which makes it capable of delivering large amounts of power to motors and other high-current devices.

- Should be able to power all 12 servo motors for a range of 15-20 minutes

PLA Plastic



PLA plastic has a number of properties that make it attractive for a variety of applications, including its high strength and stiffness, low toxicity, and ability to be composted in industrial facilities. It can also be colored and printed with a variety of textures and finishes, making it a versatile material for creating a wide range of products.

Bending modulus - how much a material can resist bending.

Software

Compared various simulation environments such as Isaac Gym, Pybullet and Gazebo.

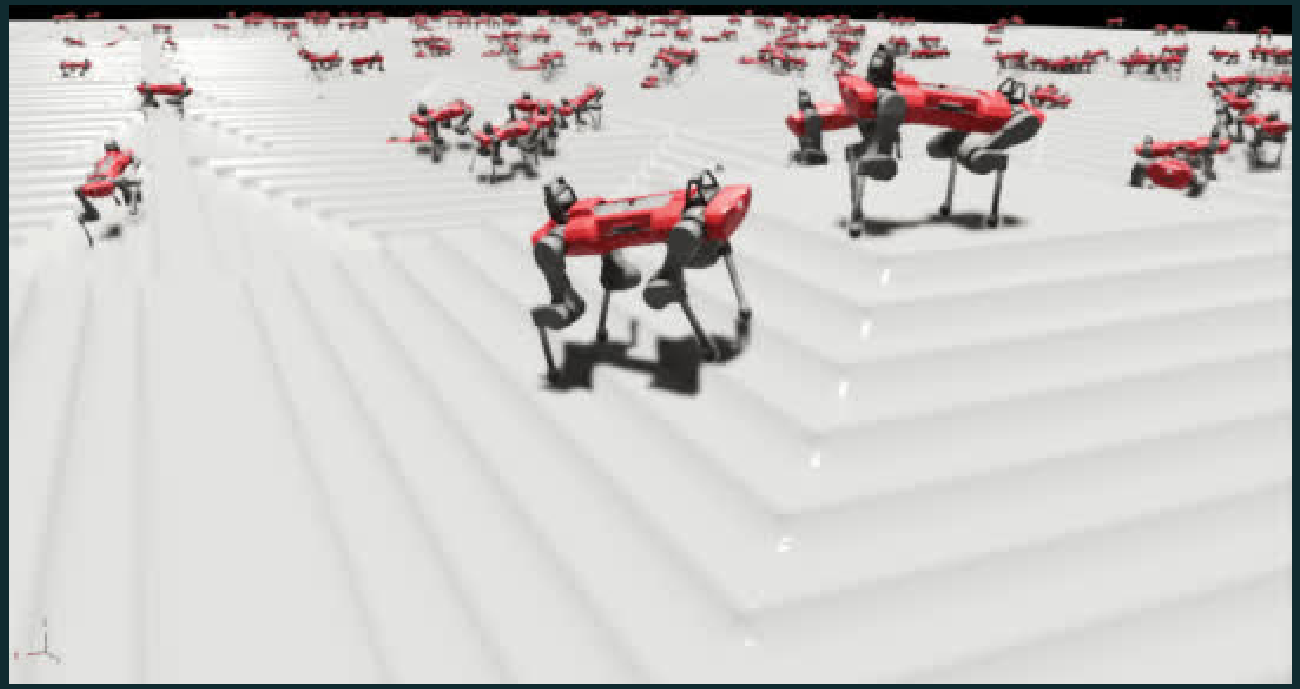
Isaac Gym

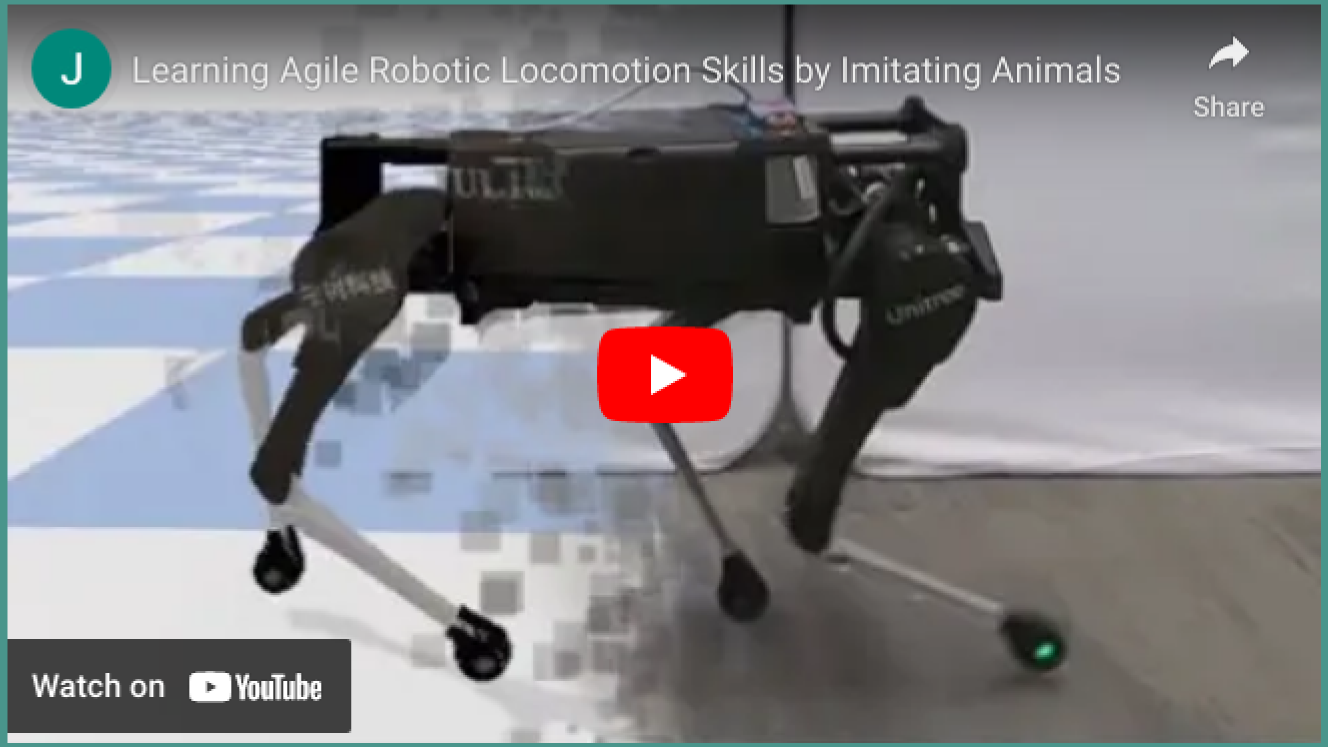
This was our first approach in tackling the simulation aspect of the project

- Growing community with increasing documentation
- Proven algorithms on similar models to ours.
- Incredibly powerful

Why we didn't stick with this approach

- Transferring of simulation to robot could not be supported because Jetson Nano is not capable of running Isaac Gym





PyBullet

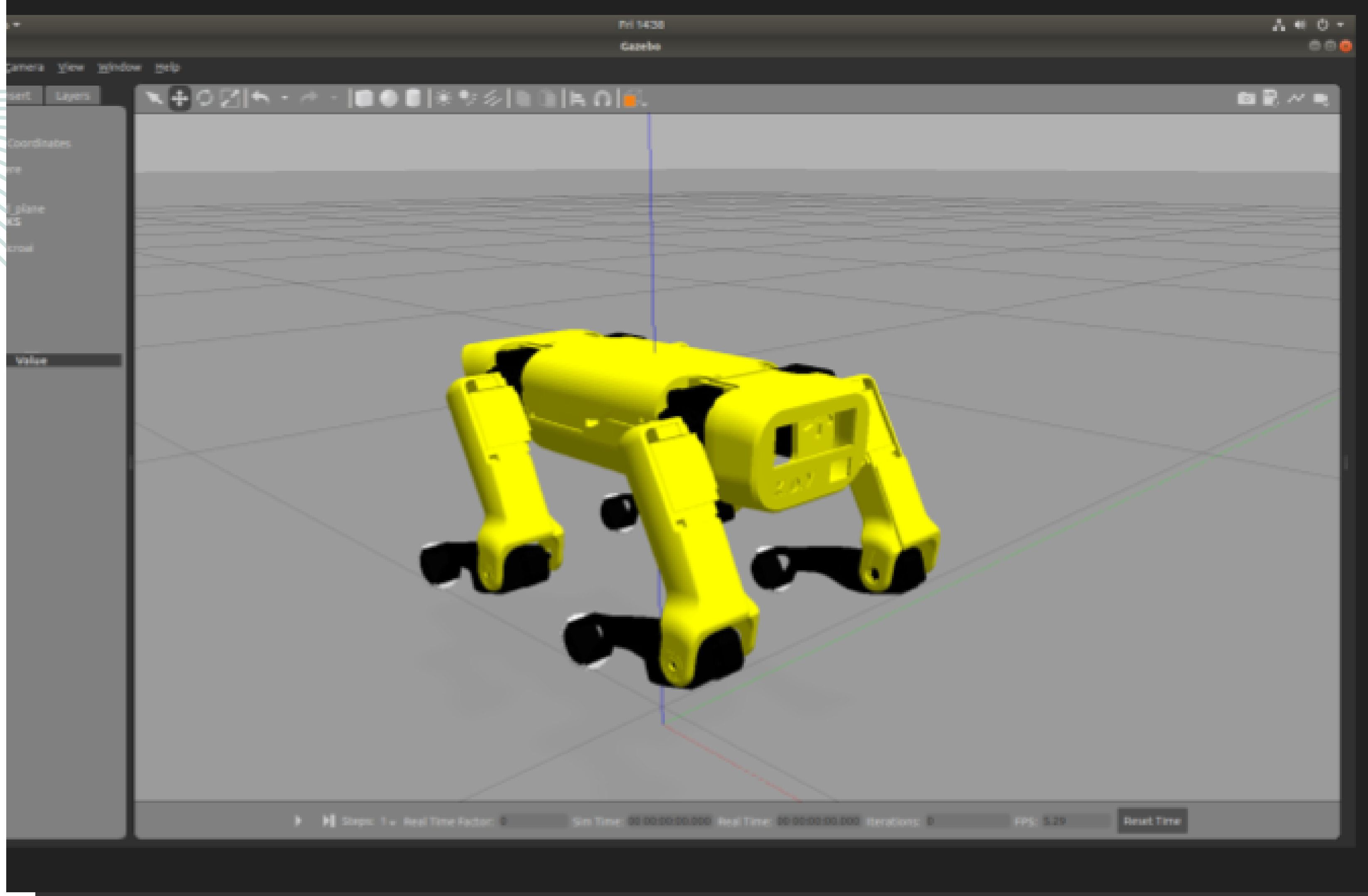
PyBullet is a physics engine integration library that provides a Python interface to the Bullet Physics Engine. It allows users to simulate and interact with physics-based systems and environments, making it useful for applications such as robotics, computer graphics, and machine learning.

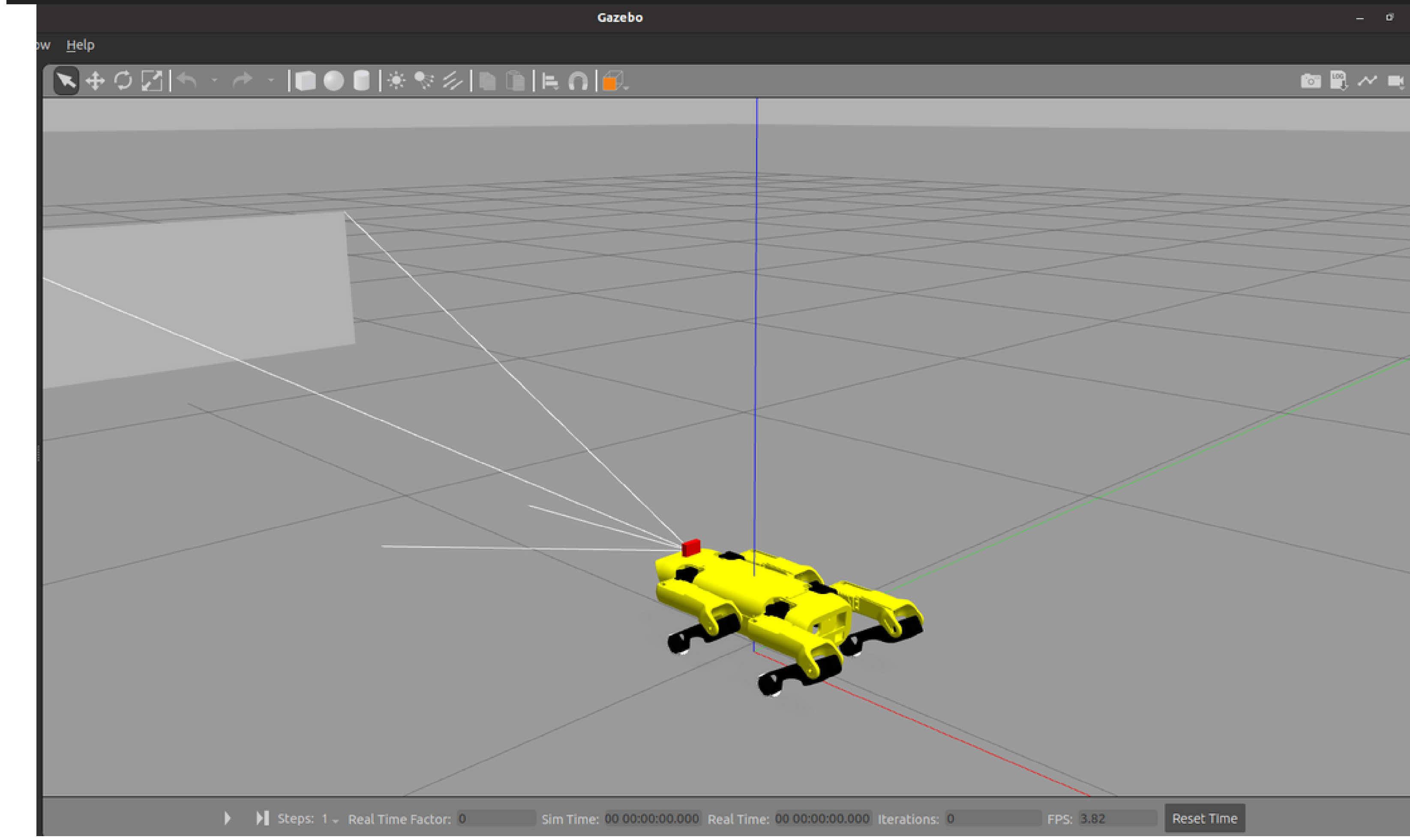
Overall thoughts

- Great simulation environment with extensive resources as it is quite popular
- Uncertain about adaptability to physical robot initially

Why ROS & Gazebo?

SpotMicroAI Model
By: Joel Martinez





- The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications.

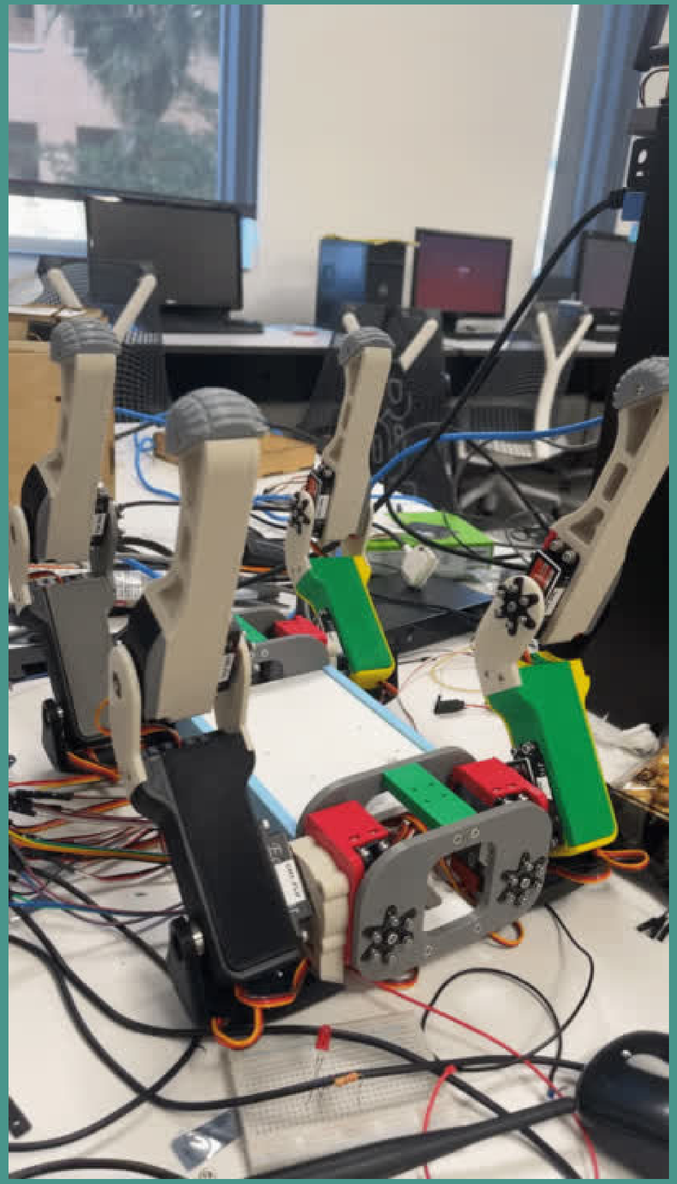
- Gazebo is the Simulation software where we can visualize all of the moving parts that ROS is responsible for.

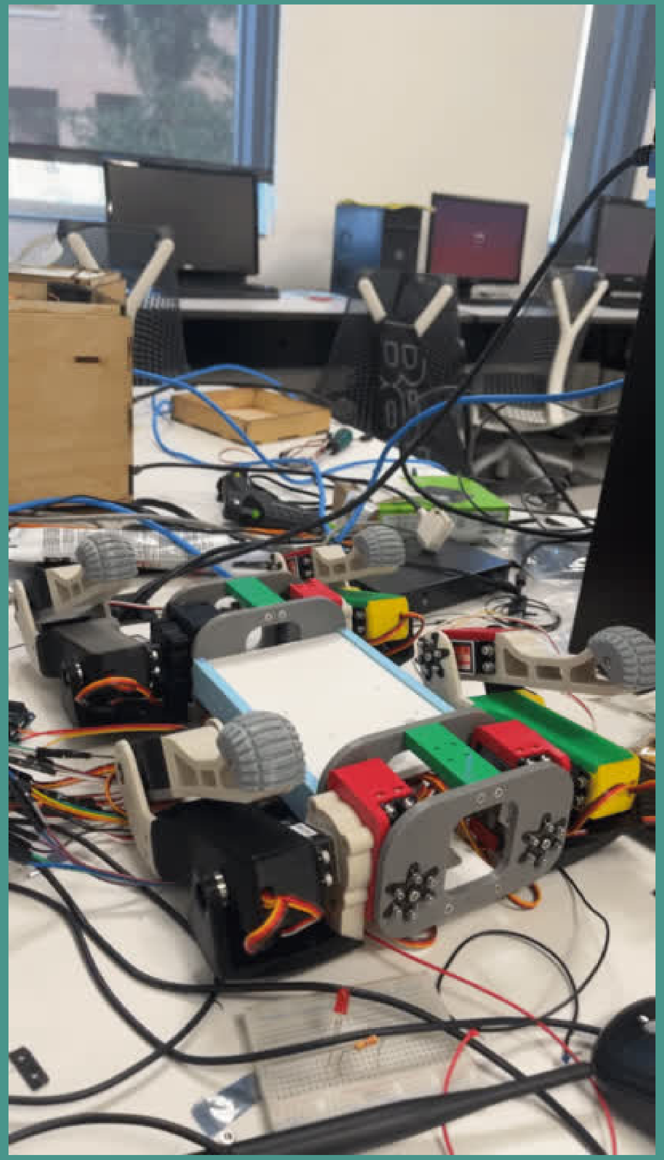
Because ROS is known for its smooth transition from simulation to real (SIM2REAL), we opted for this approach to conduct our RL simulation.

Project Highlights
- Same design and therefore smooth transition

Lacking
- IMU sensor
- Camera

https://gitlab.com/public-open-source/spotmicroai/simulation/-/tree/master/UserJo%C3%ABl%20Martinez





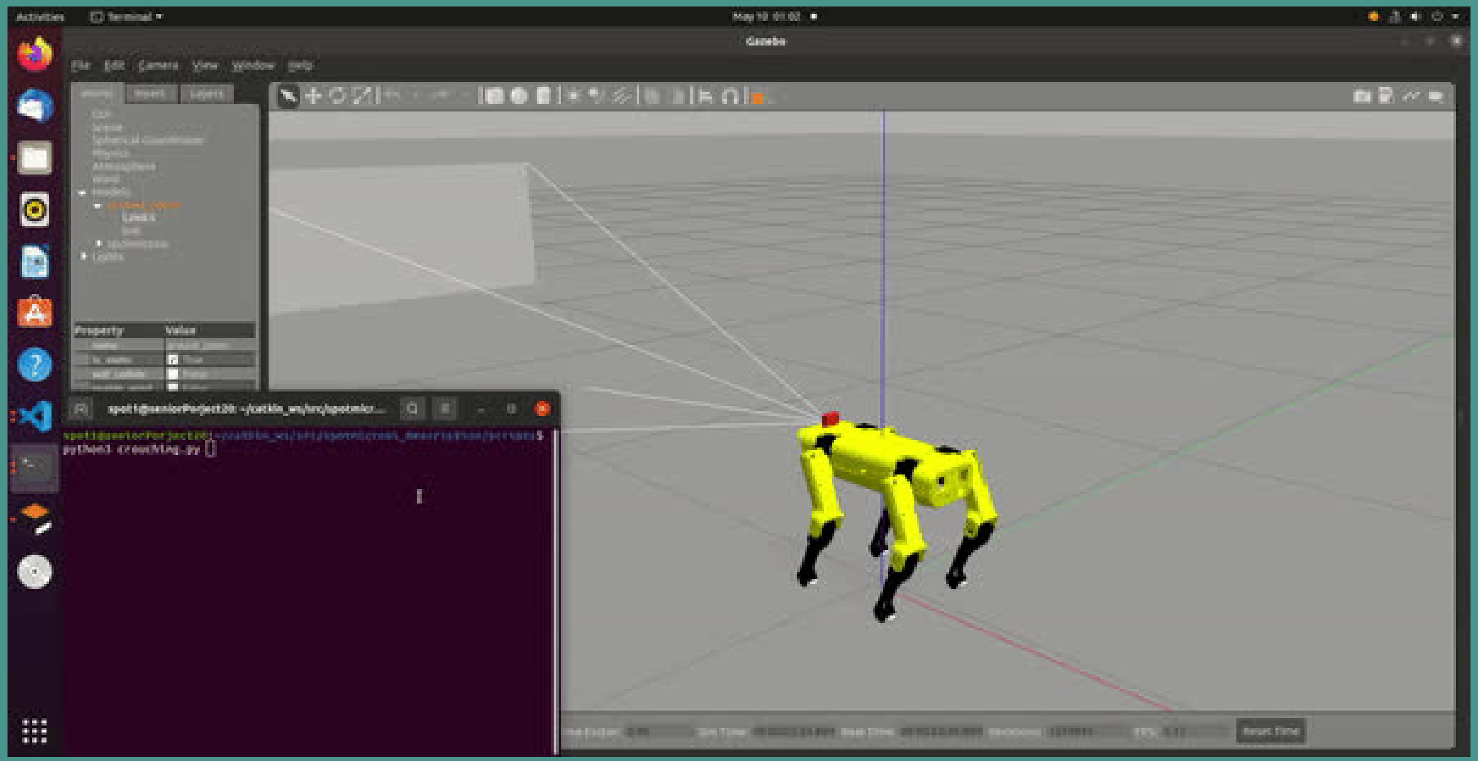
Standing/Crouching Script

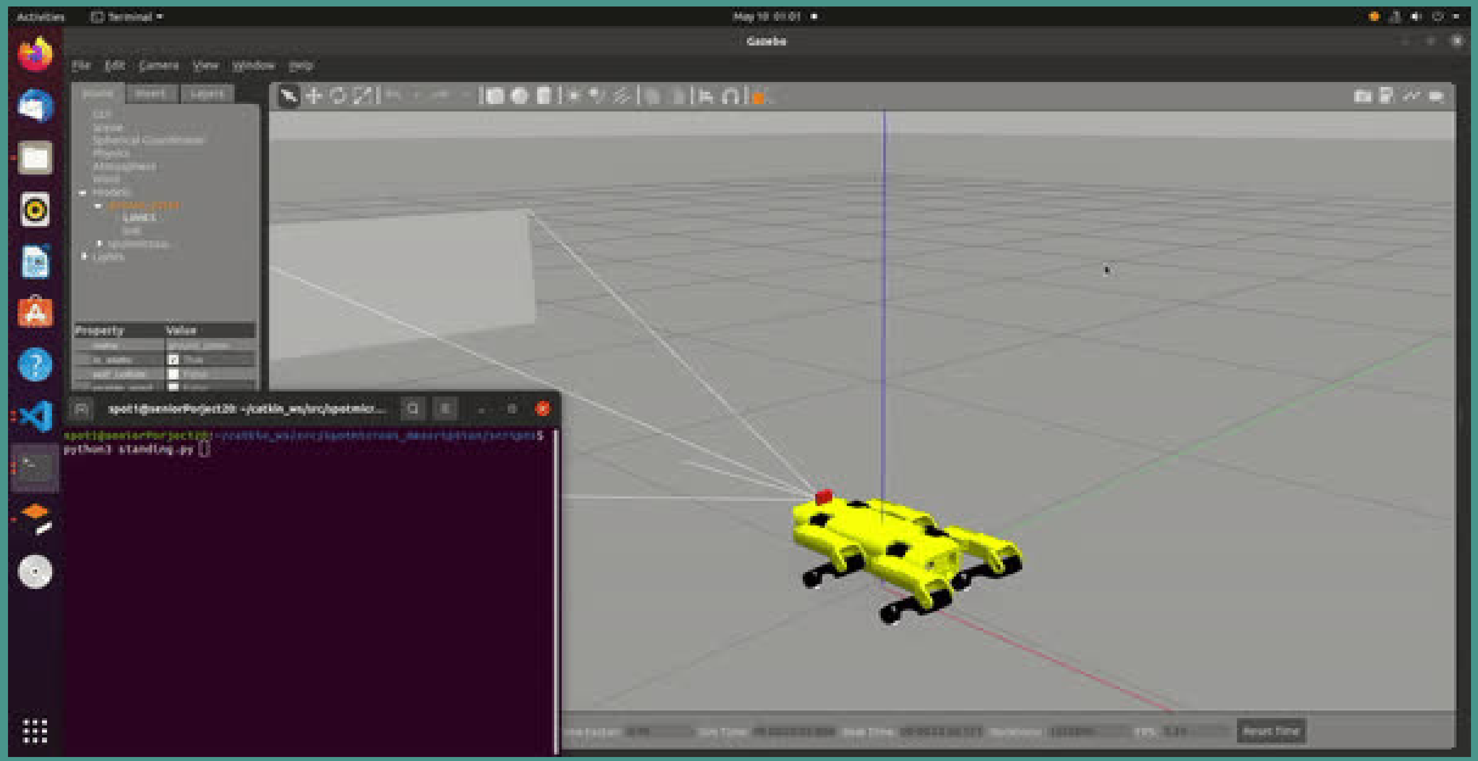
To understand ROS and its intricacies, we developed a walking script.

What's important regarding ROS is its "subscribing" and "publishing" approach "topics"

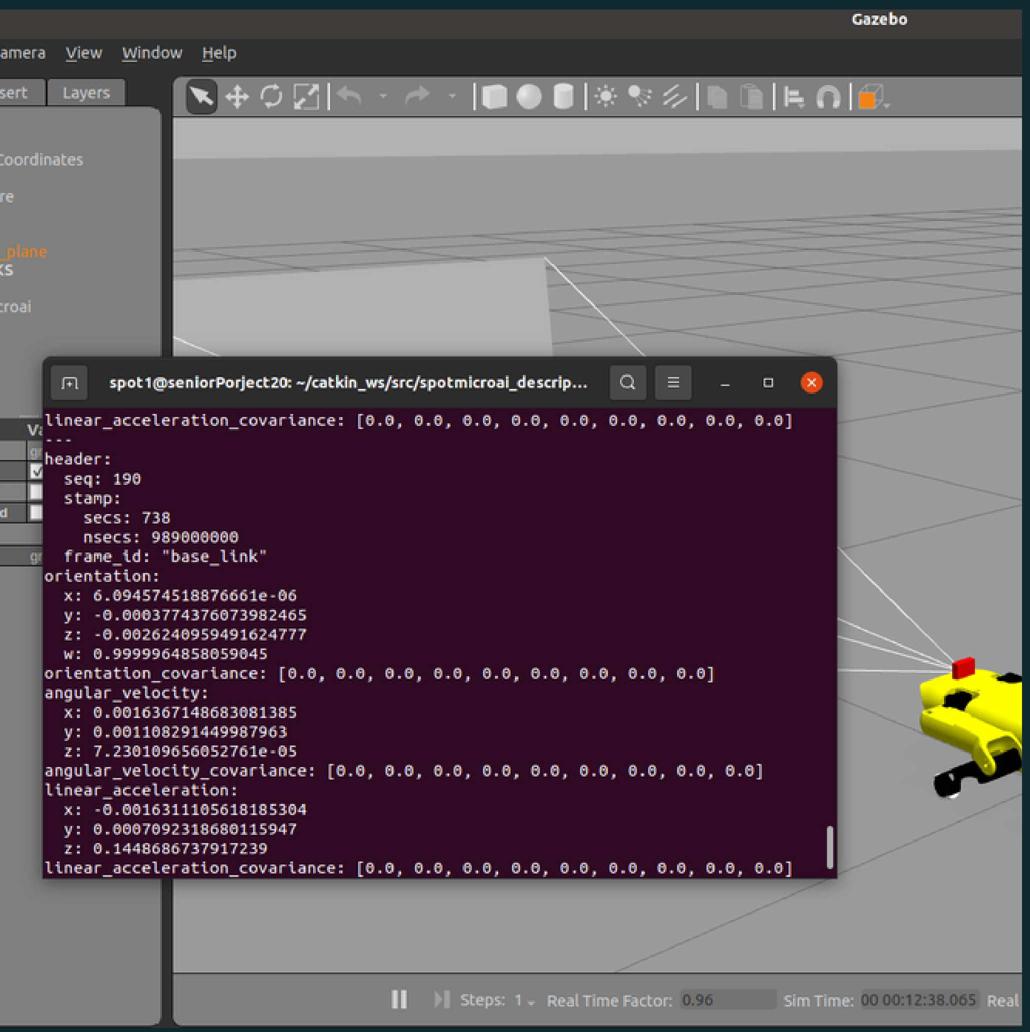
- Subscribing - can gather data from that specific "topic" or node
- Publishing - can publish data or actions to that specific "topic" or node

Important to have "topics" or nodes created when developing the model





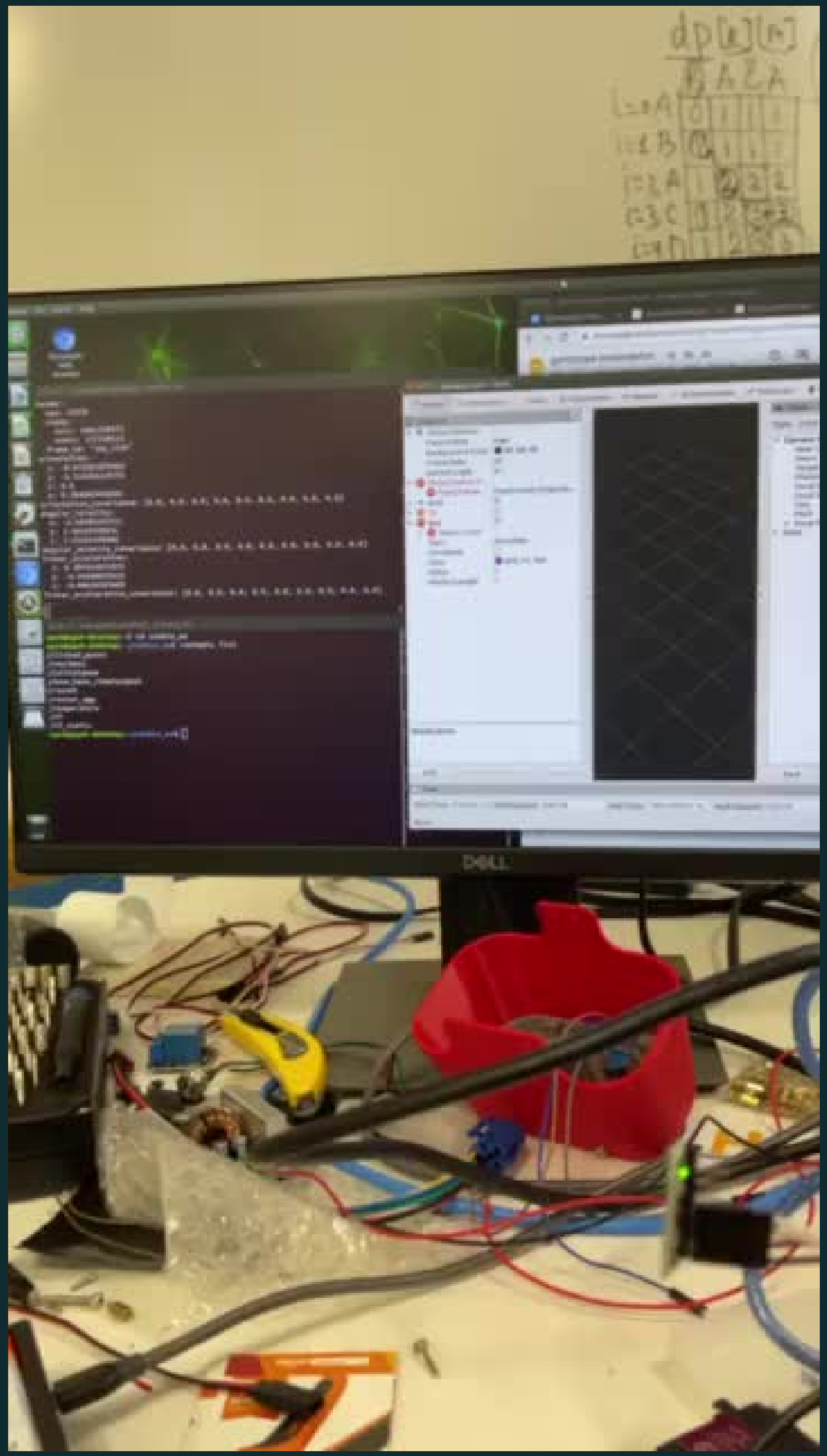
Peripheral Devices

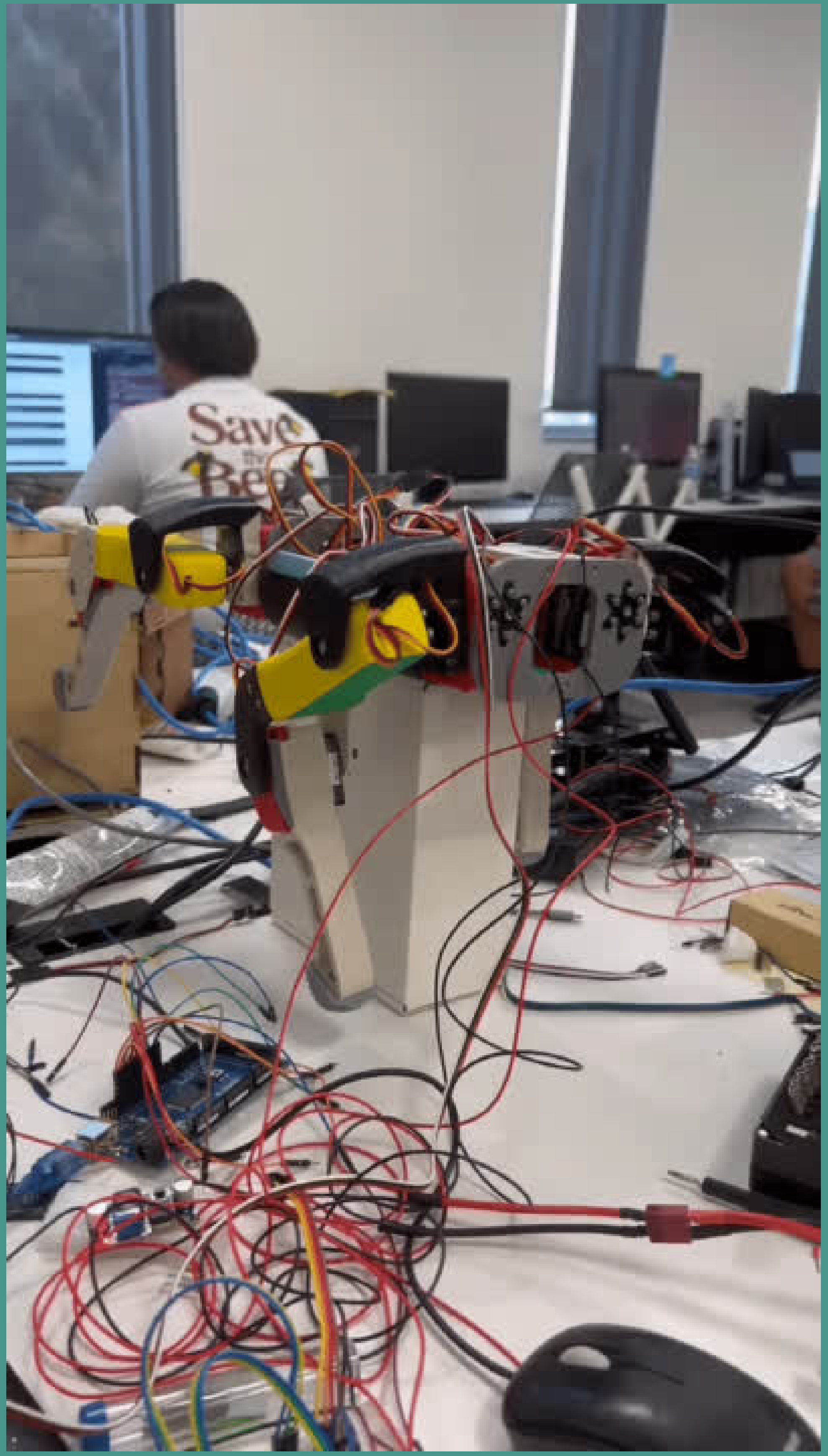


Additional devices such as OLED screens, gyroscopes, cameras can be added to the Jetson Nano, although more SDA/SCL ports are needed.

In the attached video, numbers update in real time according to gyroscope movement.

In the picture you can see how the same values are obtained from IMU 'topic' in sumulation.





Walking ROS

Able to implement a smooth walking gait to robot.

Power issues with voltage to servos which do not allow it to support weight of robot.

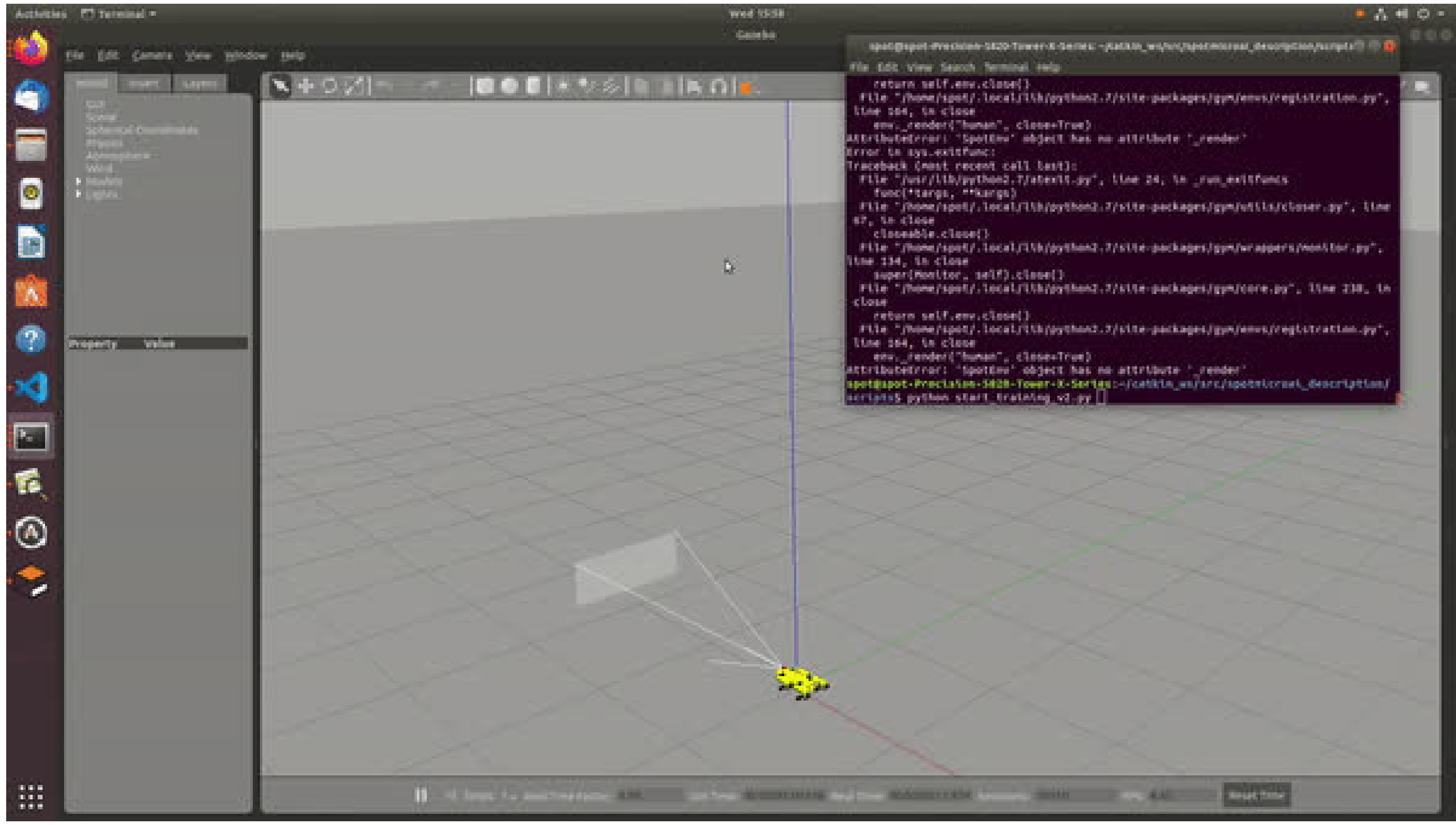
RL ROS Attempt

Various RL simulations were attempted but unsuccessful.

- First trial using Q-learn (bad approach)
- Second approach using DQN: encountered errors with ROS such as faulty connection to controllers to publish data and "unavailable data" when retrieving data from 'topics'

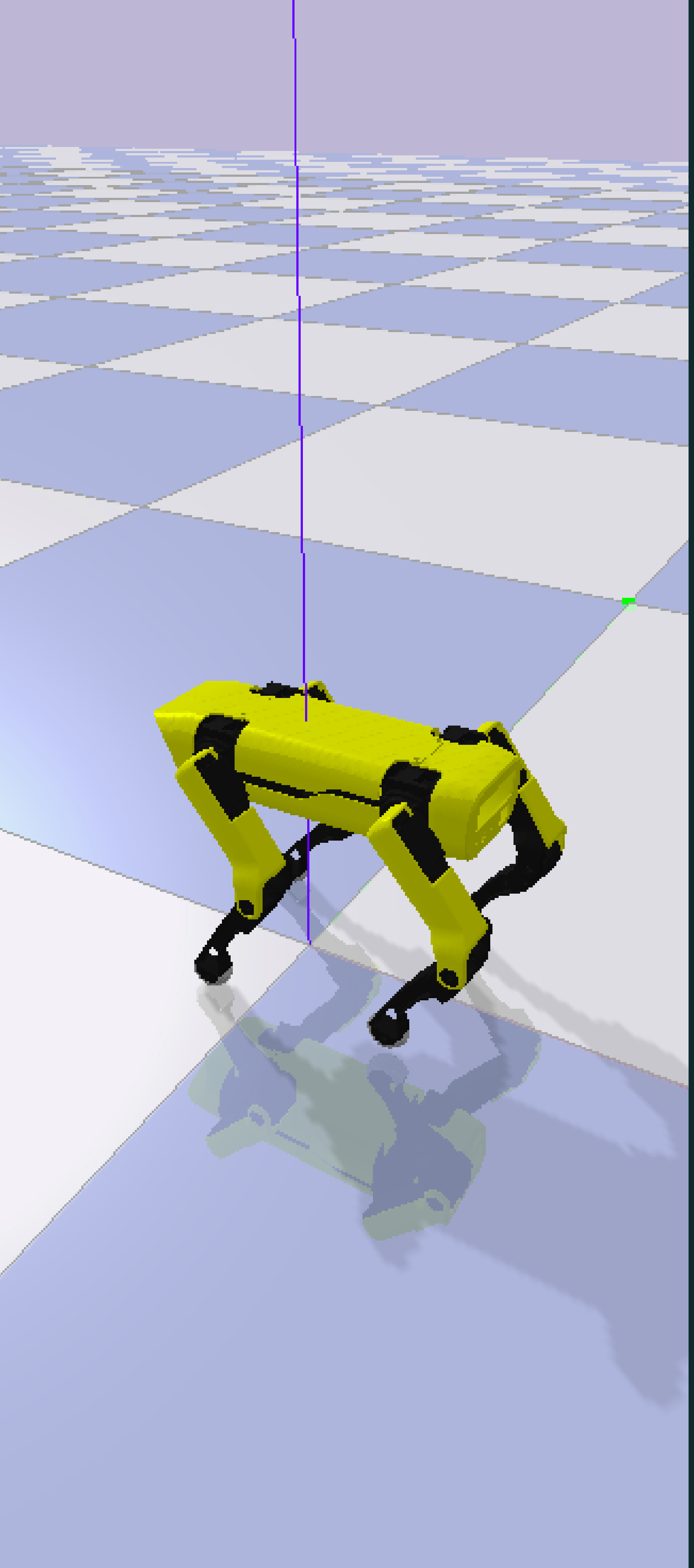
What is Q-learn: table based algorithm that calculates 'quality' of state. Task is too complex for a table based algorithm.

What is DQN: similar to Q-learn but avoids tabular representation by using deep neural network(s) to approximate 'quality' of state



```
 # Algortihm Parameters
alpha: 0.1
gamma: 0.8
epsilon: 0.9
epsilon_discount: 0.999 # 1098 eps to reach 0.1
nepisodes: 10000
nsteps: 1000

# Environment Parameters
desired_pose:
    x: 0.0
    y: 0.0
    z: 1.0
desired_yaw: 0.0 # Desired yaw in radians for the spot to stay
max_height: 3.0    # in meters
min_height: 0.5    # in meters
max_incl: 1.57        # in rads
running_step: 0.001    # in seconds
joint_increment_value: 0.05  # in radians
done_reward: -1000.0 # reward
alive_reward: 100.0 # reward
```



RL Pybullet

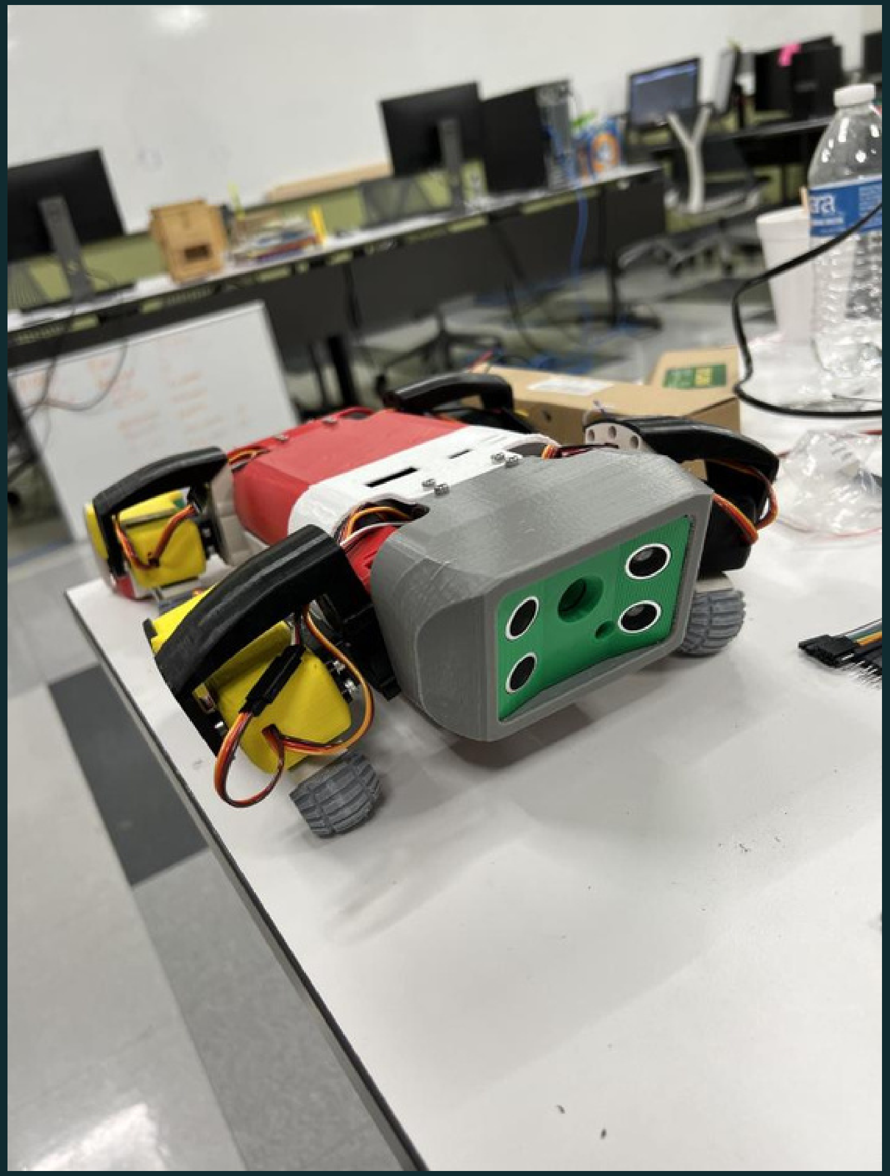
Key Features:

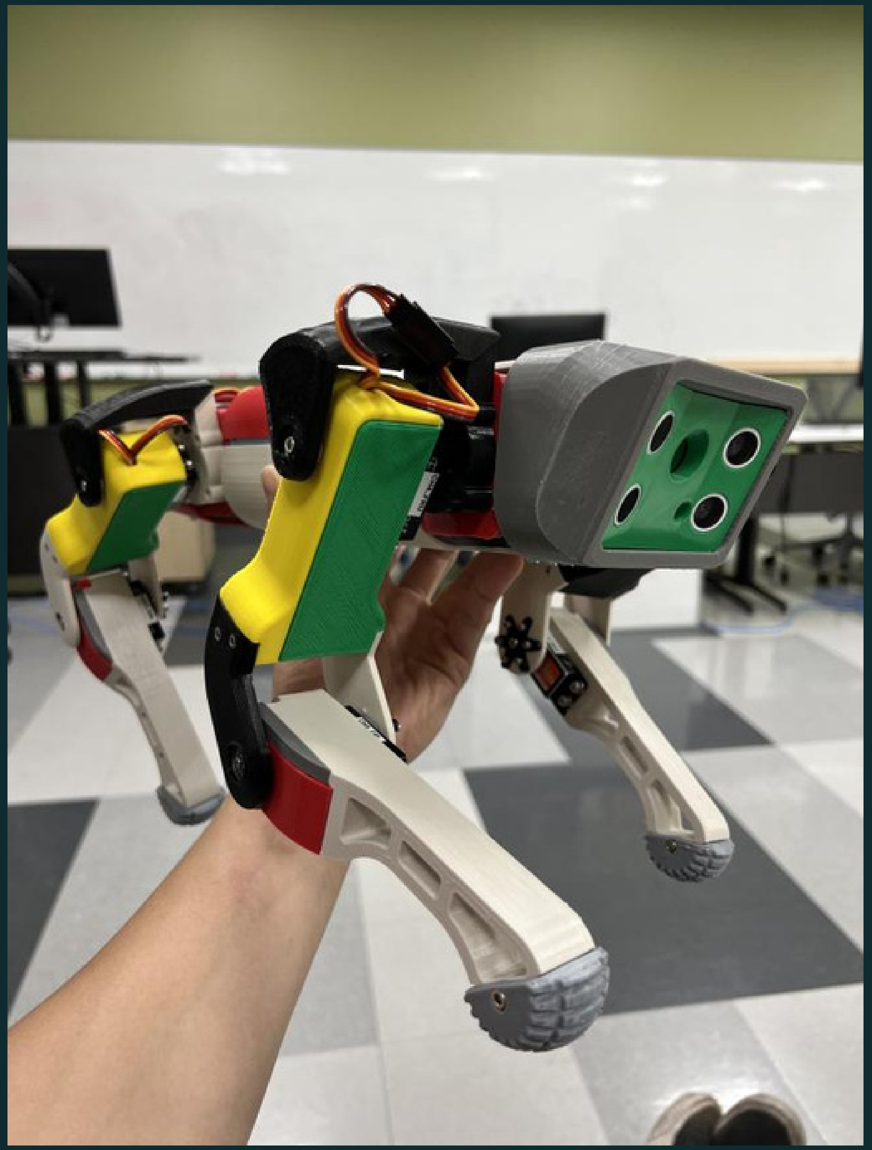
- Rigid Body Dynamics
- Collision Detection
- Kinematic and Dynamic Simulation
- Constraint Solvers
- Integration with Machine Learning Frameworks

Evaluation and lessons learned

Evaluation - Physical

Physical component of project was able to be completed during spring break.
Photo shown to Research Lab on March 14, 2023:





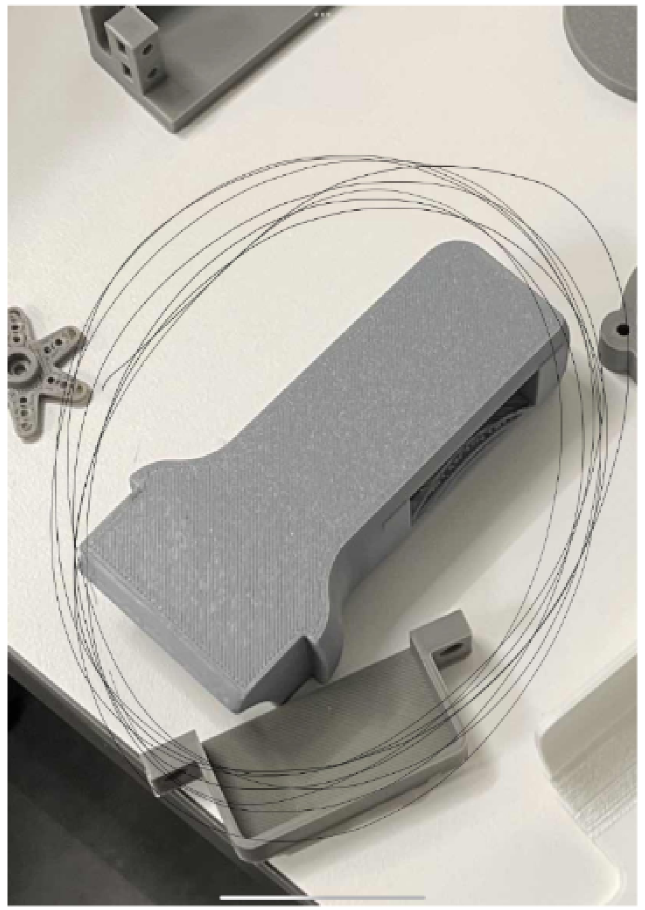
Further Evaluation / Lessons Learned

Difference between digital files and how the 3D parts come out.

Difference between youtube tutorial and code needed for our specific robot.

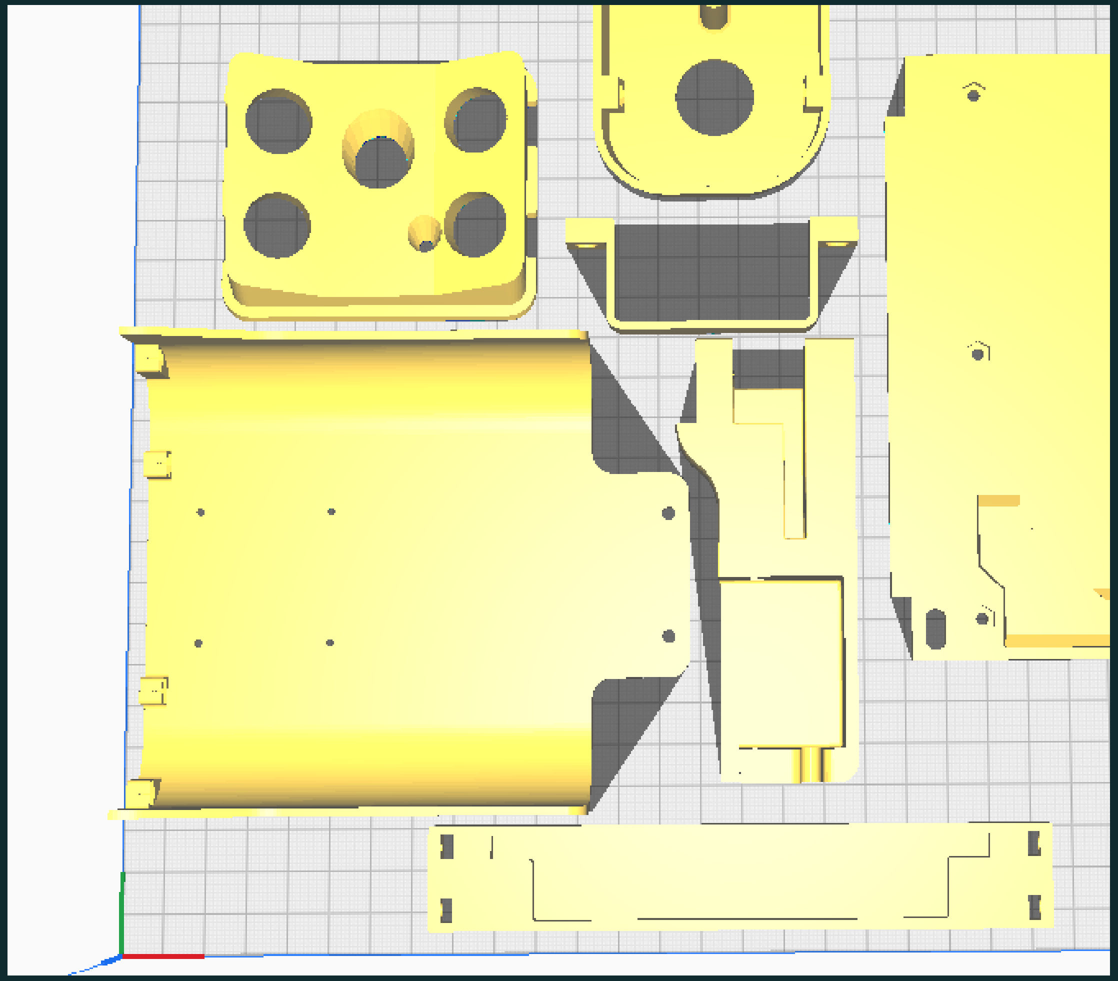
Difference between material other research projects used and our own (different weights, revised stl files).

What not to do:

- Print a part that used the wrong material.
    - 
    - Printed using TPU not PLA.

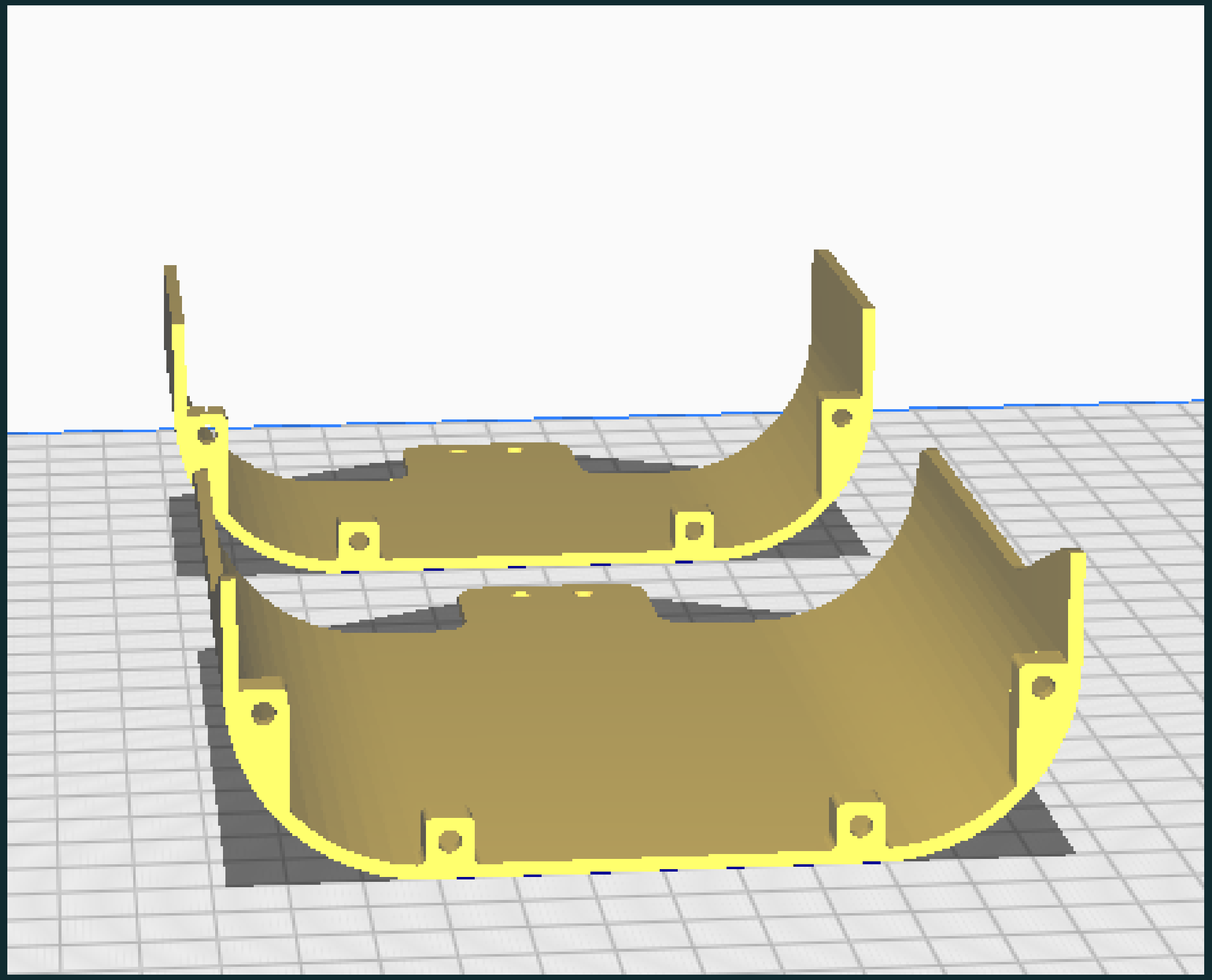
Lessons Learned

Bad stl file



wrong material

When printing more complex parts, less parts in a single batch.



Documentation

Now, we should mention our github:

https://github.com/samuellee101/Quadrupedal-Robot-Senior-Design-Project

Where we will be uploading helpful files / guides for future students.
For example, a pdf on 3D-printing and what to do, what not to do.

Special Thanks

Dr. Noe Vargas - Makerspace

Christian Narcia (AI/ML LAB) - revised stl files

Dr. Erik Enriquez - help with simulation

Dr. Dongchul Kim - project supervisor