**Project Moab End-to-end Demo**

**Setup:**

Power on the bot. Ensure your Moab is running the latest software and is connected to the local network. Run the ball color calibration. SSH into the bot in preparation for brain deployment. Full technical instructions can be found here: [Home · microsoft/moabian Wiki (github.com)](https://github.com/microsoft/moabian/wiki)

**Introduction:**

Project Moab is a desktop ball balancing robot designed to demonstrate the end-to-end Project Bonsai Platform. Microsoft Project Bonsai is a low-code AI platform for building autonomous systems. Autonomous systems are the next step in the advancement of industrial automation. These intelligent systems leverage the power of AI to sense, plan, and act in a dynamic environment.

**Hardware Overview:**

Project Moab is a simple toy problem we use to quickly demonstrate how to design, train, and deploy a high-level AI agent, called a Bonsai Brain. Moab uses an upward facing camera to detect the position and velocity of a ball on the plate. The position and velocity are described in terms of x and y coordinates on the plate. Moab then uses three actuators to change the angle of the plate. The angle of the plate is described in terms of pitch and roll.

**Brain Design:**

To design a brain to balance the ball on the center of the plate, we start by breaking the problem down into states and actions. The states are what we can sense – the x and y position and velocity of the ball. The actions are the actuators – the pitch and roll of the plate. To teach an AI to balance the ball we define two high level objectives:

1. Avoid falling off the plate
2. Drive to the center of the plate

This is the concept of **Machine Teaching** – breaking a complex problem down into one or more high level objectives or skills. To define the brain and codify the machine teaching, we use the single purpose language, **Inkling**. A good explanation of the different goal types with code samples can be found here: [Explore Autonomous Systems - Microsoft Innovation](https://innovation.microsoft.com/en-us/exploring-autonomous-systems)

**Train a brain:**

Using the sample Moab Inkling in the [**Bonsai Web UI**](https://preview.bons.ai/), show how the states and actions are defined, then show where the two goals are defined. Click train. Show how the simulators connect and the brain practices. Wait 10 minutes for the brain to train to 100% goal satisfaction.

**Export and Deploy a brain:**

Once the brain has reached 100%, stop training and click export. We are exporting to a RaspberryPi so choose Arm32V7 as the processor architecture. [Complete docker deployment](https://github.com/microsoft/moabian/blob/main/docs/deploy-brain.md) and run the brain on the hardware via SSH on the terminal.

**Additional talking points:**

* The demo brain will likely perform poorly. This is expected since the brain was designed to train quickly. Follow [Moab tutorial 2](https://microsoft.github.io/moab/tutorials/2-robustness/index.html) to show how to use domain randomization to train a more robust brain.
* The Moab demo uses a [python based simulator](https://github.com/microsoft/moabsim-py/tree/main). We also have a [MathWorks Simulink simulator.](https://www.mathworks.com/matlabcentral/fileexchange/75512-microsoft-project-bonsai-simulink-toolbox) This can be shown in a supplemental demo