InternPro Weekly Progress Update

Name	Email	Project Name	NDA/ Non-NDA	InternPro Start Date	OPT
Adharsh Prasad Natesan	anatesan@asu.edu	IT-Core Foundation Suriname	Non-NDA	2024-08-05	Yes

Progress

Include an itemized list of the tasks you completed this week.

#	Action Item/ Explanation	Total Time This Week (hours)
1	Installing Ubuntu as a Dual Boot with Windows	2
2	Installing ROS 2 on Ubuntu	2
3	Installing Gazebo on Ubuntu	2
4	Searching and Converting a Farmland Model for Gazebo	3
5	Importing and Displaying Farmland in Gazebo Using a `.dae` File	3
6	Literature Review on Farmland Creation and Navigation in ROS2	3
7	Literature Review on Farmland Creation and Navigation in ROS2	3
8	Projected Work Plan for Next Week	1
9	Report writing	1
	Total hours for the week:	20

Verification Documentation:

Action Item 1: Installing Ubuntu as a Dual Boot with Windows – 2 hour(s).

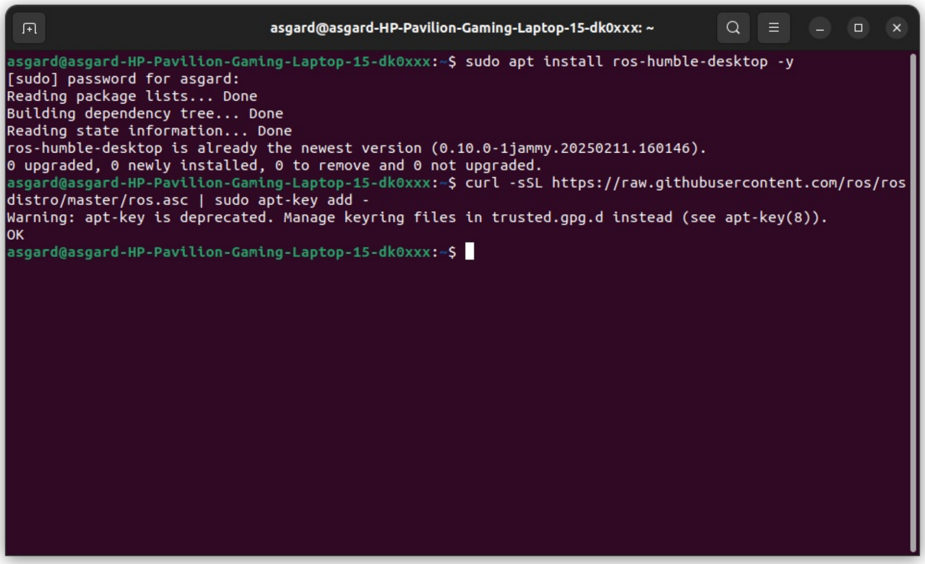
Project Work Summary

1. Backed up important files to an external drive or cloud to prevent data loss before partitioning.
2. Downloaded the Ubuntu ISO from the official website and used Rufus to create a bootable USB drive, ensuring the correct partitioning scheme (GPT) and UEFI compatibility.
3. Adjusted Windows settings by shrinking the main partition to allocate at least 20GB for Ubuntu and disabling "Fast Startup" to avoid boot issues.
4. Accessed the boot menu to start the installation, manually set up partitions, and installed GRUB as the bootloader to enable dual-booting with Windows.
5. Verified system stability post-installation.

Action Item 2: Installing ROS 2 on Ubuntu – 2 hour(s).

Project Work Summary

- Updated the system’s package index and upgraded existing packages to ensure compatibility with ROS 2.
- Added the official ROS 2 repository to the system’s sources list.
  ```bash
  sudo apt install software-properties-common
  sudo add-apt-repository universe
  sudo apt update && sudo apt install curl -y
  curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
  ```
- Installed the full desktop version of ROS 2 Humble, which includes all core libraries and tools:
  ```bash
  sudo apt install ros-humble-desktop -y
  ```
- Configured the shell environment to source ROS 2 setup files automatically:
  ```bash
  echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc
  source ~/.bashrc
  ```
- The resulting setup provides a fully functional ROS 2 environment, enabling development and testing of robotics applications with support for tools like `colcon` and `rviz`.

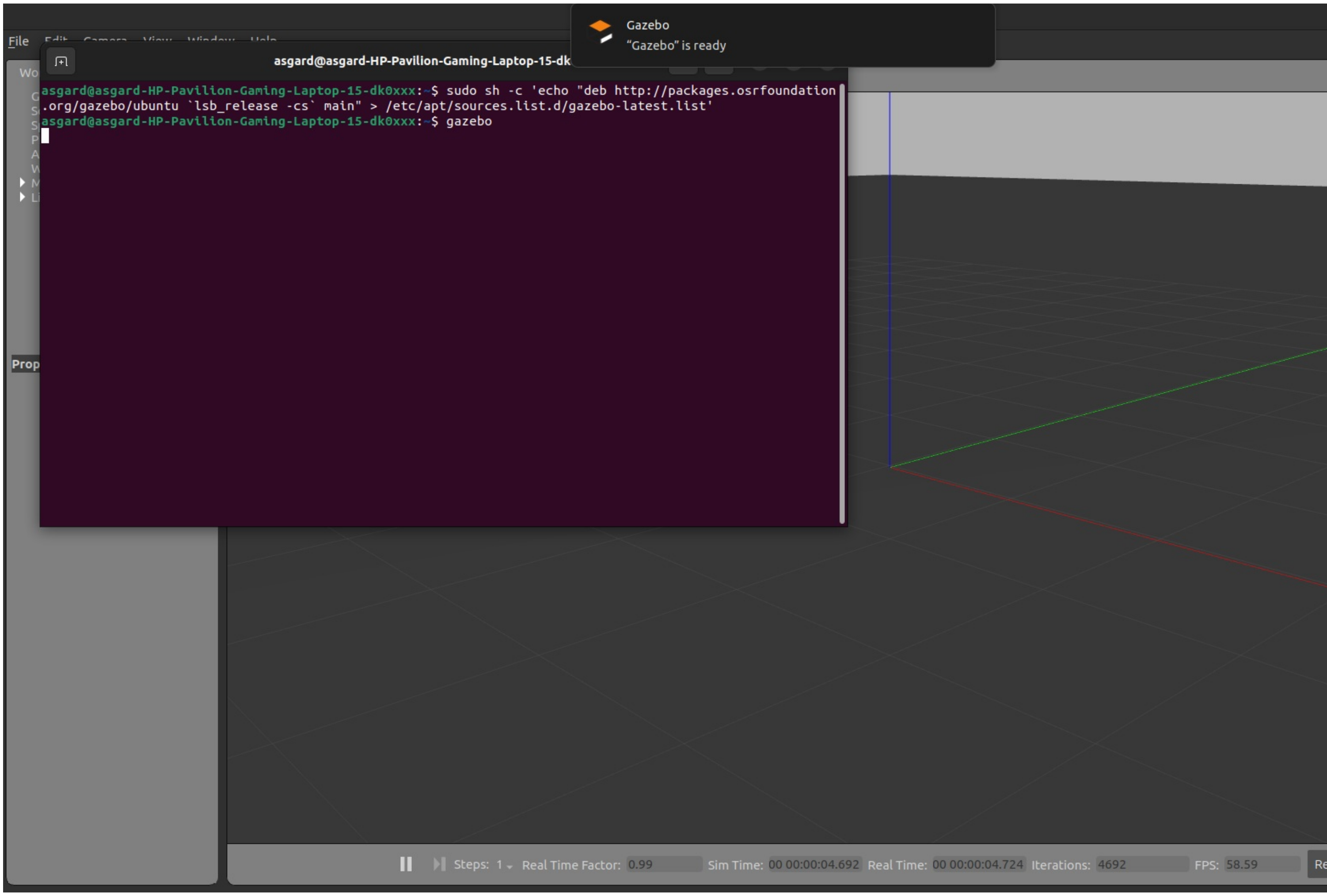


-
-

Action Item 3: Installing Gazebo on Ubuntu – 2 hour(s).

Project Work Summary

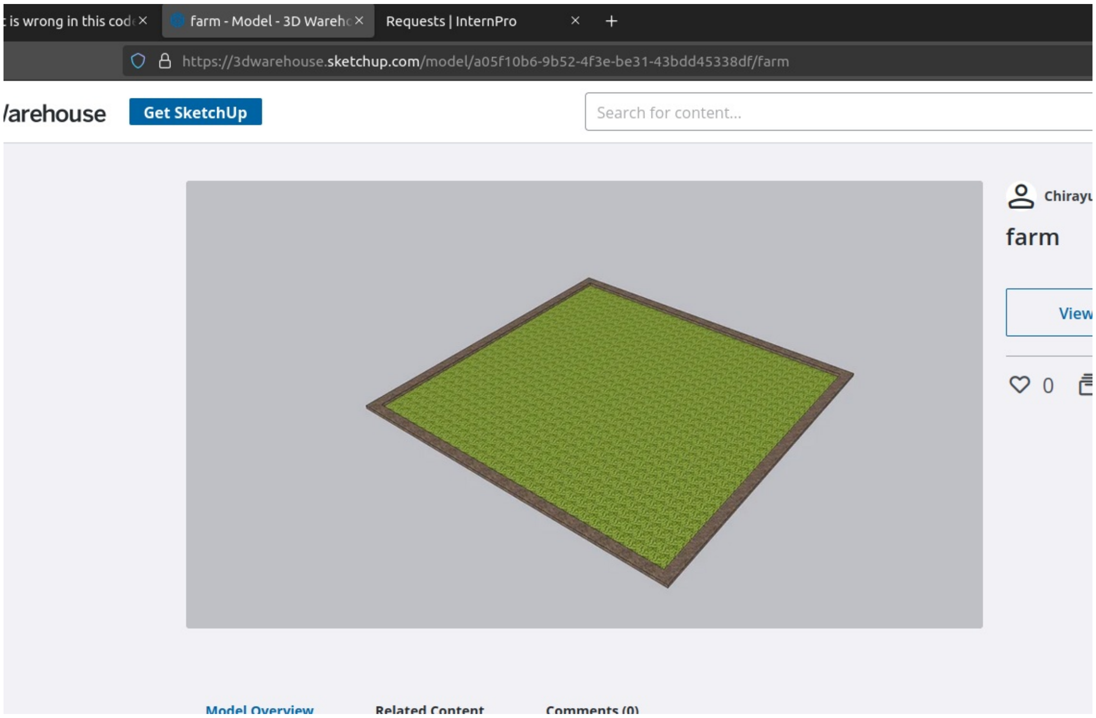
- To set up Gazebo for robotics simulation on an Ubuntu system, we followed a structured step-by-step process to ensure proper installation and integration with ROS 2. This task focused on installing the required packages, verifying the installation, and preparing the environment for use with ROS 2.
- Added the Open Source Robotics Foundation (OSRF) repository to the system's sources list to access the latest Gazebo packages:
  ```bash
  sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-latest.list'
  ```
- Imported the GPG key to verify package authenticity:
  ```bash
  wget https://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
  ```
- Updated the package list and installed Gazebo along with its development libraries:
  ```bash
  sudo apt-get update
  sudo apt-get install gazebo11 libgazebo11-dev
  ```
- Tested the installation by launching Gazebo:
  ```bash
  gazebo
  ```
  - Ensured that the Gazebo interface loaded successfully without errors.
- Installed `gazebo_ros_pkgs` to enable communication between ROS 2 and Gazebo:
  ```bash
  sudo apt-get install ros-humble-gazebo-ros-pkgs
  ```
- Verified the integration by launching a test world in Gazebo and visualizing robot data in ROS 2:
  ```bash
  ros2 launch gazebo_ros empty_world.launch.py
  ```



Action Item 4: Searching and Converting a Farmland Model for Gazebo – 3 hour(s).

Project Work Summary

- Started by searching for suitable farmland models on the 3D Warehouse website, focusing on realistic representations of agricultural fields. After browsing through multiple options, evaluated them based on visual appearance and their suitability for simulation. Eventually, selected a model with a green field and clear boundaries that met our project's requirements.
- Once finalized the model, downloaded it in .skp (SketchUp) format, as this is the default format used by 3D Warehouse. After downloading, checked the file to ensure it was intact and ready for conversion. Since Gazebo doesn't support .skp files directly, needed to convert it into a compatible format.
- To do this, used an online conversion tool to transform the .skp file into .dae (Collada) format. Carefully uploaded the file, making sure that all the geometry and textures were preserved during the conversion process. After the conversion was complete, downloaded the .dae file and verified its integrity by inspecting it for any missing elements or distortions.
- Next, organized the converted .dae file into the appropriate directory structure for Gazebo. Also made sure to include any associated textures alongside the model to prevent rendering issues. Finally, confirmed that the farmland model was correctly formatted and ready for integration into our simulation environment.
- Through this process, efficiently located, converted, and prepared a realistic farmland model for use in robotics simulations, ensuring compatibility and proper visualization in Gazebo.

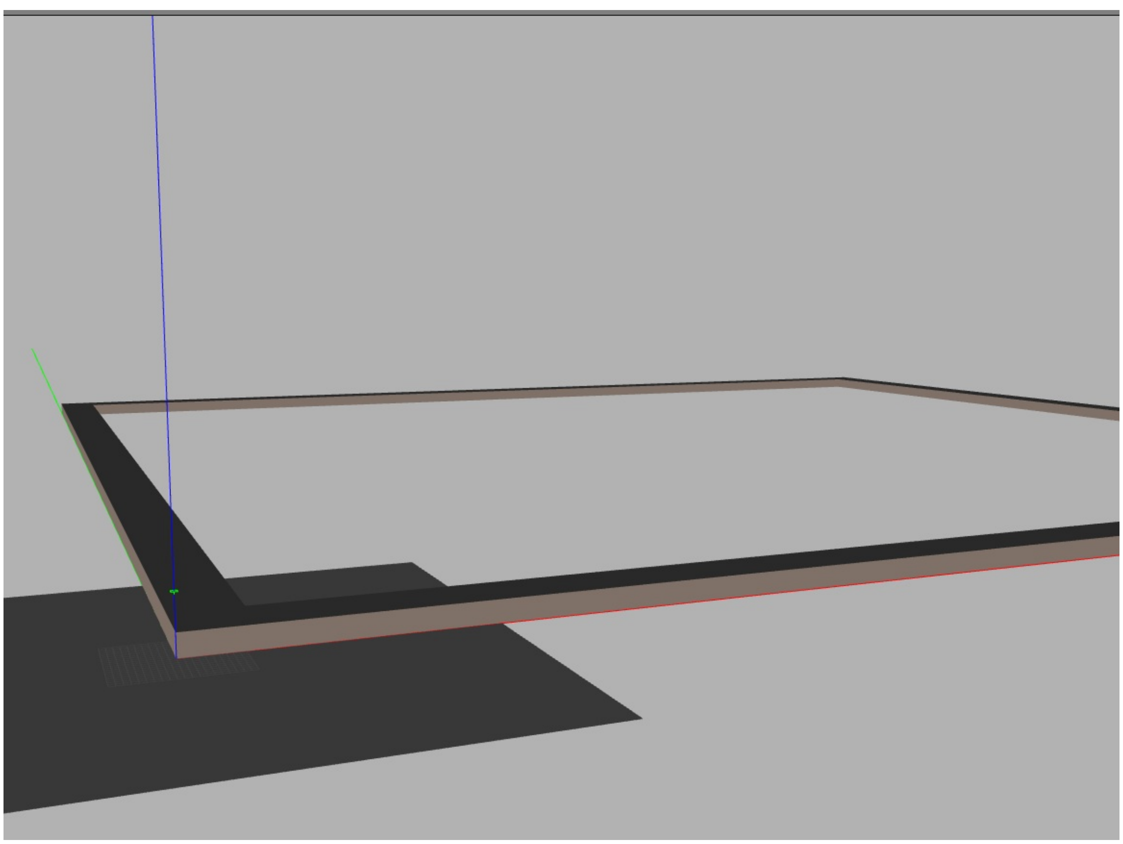


-
-

Action Item 5: Importing and Displaying Farmland in Gazebo Using a `.dae` File – 3 hour(s).

Project Work Summary

- We started by downloading a farmland model in `.dae` format from an online 3D model repository.
- The `.dae` file was checked for compatibility with Gazebo, ensuring it included basic geometry.
- A new model directory (`my_farm_model`) was created under `~/.gazebo/models/`.
- The `.dae` file was placed in a `meshes/` subdirectory to match Gazebo's expected structure.
- Supporting files, including `model.config` and `model.sdf`, were created to define metadata and describe how the model should appear in Gazebo.
- A `model.config` file was written to provide metadata about the farmland model, including its name, version, author, and description.
- A `model.sdf` file was created to describe how Gazebo should render the farmland.
- The `.sdf` file referenced the `.dae` file's location and included scaling parameters to adjust the size of the farmland.
- While there were no errors during implementation, only part of the farmland was visible in Gazebo.
- Despite these challenges, we successfully imported and displayed the farmland model in Gazebo. The next steps involve investigating and resolving visual inconsistencies to achieve a complete and realistic representation of the farmland



Action Item 6: Literature Review on Farmland Creation and Navigation in ROS2 – 3 hour(s).

Research

- 3D-Simulation Data-Making Trial to Present and Analyze Small-Sized Farmlands Fields with Car-Shaped Robot, ROS2, SLAM, and Foxy for Real Agricultural Workers
- https://www.researchgate.net/publication/356741933_3D-Simulation_Data-Making_Trial_to_present_and_analyze_Small-sized_Farmlands_Fields_with_Car-shaped_Robot_ROS2_SLAM_and_Foxy_for_Real_agricultural_workers
- Shinji KAWAKURA and Ryosuke SHIBASAKI
- Summary of the Report
  - This paper explores a simulation framework designed for small-sized farmlands, leveraging ROS2 and SLAM-based navigation techniques.
  - The study introduces a car-shaped robot (NANO-RT1) equipped with various sensors to collect 3D data from agricultural fields.
  - To evaluate mapping accuracy, five SLAM algorithms—Hector-SLAM, G-mapping, Karto-SLAM, Core-SLAM, and Lago-SLAM—were tested under different field conditions.
  - The primary goal of the system is to support agricultural workers by digitalizing farmland data and improving robotic navigation in unstructured environments. The simulation was built using ROS2 Foxy, allowing real-time sensor integration and robot control.
  - By comparing SLAM algorithm performance, the study provides insights into selecting the most effective approach for precise mapping in agricultural automation. The results include error measurements and standard deviations, demonstrating how different techniques perform across varying terrain conditions.
  - Beyond navigation, the paper highlights broader applications, such as optimizing farm operations, enhancing worker efficiency, and improving security within agricultural environments.
- Relation to Project
  - This research directly ties into farmland simulation in ROS2 by offering insights into constructing realistic digital farmland environments and optimizing SLAM-based navigation.
  - The study’s approach to generating digital farmland twins aligns well with simulation-based agricultural research. Additionally, the SLAM algorithm comparison serves as a valuable benchmark for selecting the most suitable method for agricultural tasks, depending on environmental complexity.
  - Since the project also utilizes ROS2 Foxy, this paper reinforces best practices in integrating modern robotics middleware for real-world agricultural applications.
- Motivation for Research
  - Traditional farming methods often lack efficient tools for generating accurate digital representations of fields.
  - As autonomous agricultural robots become more common, they require robust mapping techniques to navigate challenging terrain and avoid obstacles.
  - By combining SLAM with ROS2, this research presents a scalable and adaptable solution for real-time data collection and autonomous navigation in agricultural settings.

Action Item 7: Literature Review on Farmland Creation and Navigation in ROS2 – 3 hour(s).

Project Work Summary

- Comparative Analysis of Unity and Gazebo Simulators for Digital Twins of Robotic Tomato Harvesting Scenarios
- https://link.springer.com/chapter/10.1007/978-3-031-72059-8_2
- Juan Pablo Espejel Flores, Abdurrahman Yilmaz, Luis Arturo Soriano Avendaño & Grzegorz Cielniak
- Summary of the Report
  - This paper is part of the Agri-open Core (AOC) project and introduces a tomato farm simulator designed to work seamlessly with both Gazebo and Unity using ROS2. The simulator provides tools for generating realistic tomato farms or glasshouses in a parametric manner, allowing users to customize features such as plant rows, stems, leaves, and fruits.
  - One of its key functionalities is enabling researchers to test robotic harvesting scenarios by integrating navigation algorithms into simulated environments. Users can adjust farm size, plant spacing, row distances, and lighting conditions through a Jupyter Notebook interface, making the system highly flexible. The generated farm models are compatible with both Gazebo Classic and Gazebo Sim, facilitating easy integration into ROS2-based simulations.
  - For navigation, the simulator leverages ROS2 packages like nav2 for path planning and obstacle avoidance, allowing for realistic movement within the virtual farm. This makes it a valuable tool for developing and refining autonomous agricultural robots before real-world deployment.
- Relation to Project
  - This research offers practical tools for farmland simulation in Gazebo while integrating navigation capabilities in ROS2. The tomato farm generator can be adapted for different types of crops and farmland layouts, making it versatile for various agricultural research applications.
  - The use of nav2 showcases how ROS2's navigation stack can handle complex farm environments, providing insights into obstacle avoidance and path planning for autonomous farming robots.
  - Additionally, the parametric customization feature enables researchers to create diverse agricultural scenarios without needing extensive 3D modeling expertise.
- Motivation for Research
  - A key challenge in robotic farming is the need for realistic digital twins of agricultural environments to test algorithms before deploying robots in real fields.
  - This research addresses that challenge by creating a simulation framework that supports precise validation of robotic harvesting and navigation strategies.
  - By simulating different farm layouts, researchers can refine their algorithms while reducing real-world testing costs. The integration of both Unity and Gazebo provides flexibility in choosing the most suitable simulation platform based on specific project requirements, ensuring adaptability for a wide range of agricultural robotics applications.

Action Item 8: Projected Work Plan for Next Week – 1 hour(s).

Project Work Summary

- Map Creation
  - Start by creating a basic farmland map in Gazebo, either manually using built-in tools or by importing a pre-existing `.dae` model.
  - Ensure the map includes essential elements like crop rows, boundaries, and obstacles for navigation testing.
  - Save the map as a `.world` file for future use.
- Add Rover to the Environment
  - Integrate a TurtleBot3 or a custom rover model into the farmland environment.
  - Verify that the rover spawns correctly in the map and is positioned at an appropriate starting point.
- Navigation Setup
  - Set up the ROS2 `nav2` stack for navigation within the farmland.
  - Implement basic waypoint navigation to move the rover between predefined points in the map.
  - Test obstacle avoidance functionality using simple obstacles added to the environment.

Action Item 9: Report writing – 1 hour(s).

Project Work Summary

- Created word document layout to write contents of the weekly progress.
- Created relevant subsections in the epicspro website and documented 20 hours of weekly progress.
- Collected relevant documents research papers, relevant links and company's objective from their portal.