A Core Developer's insights on Gazebo's Future

Jose Luis Rivero
jrivero@honurobotics.com

honu ROBOTICS

Who am I, Why I'm here

1. Part of the Open Robotics team since 2013
   a. Open Source Robotics Foundation (OSRF)
   b. Open Robotics Corporation
   c. Open Source Robotics Alliance (OSRA)

2. Current a member of the Project Management Committee for Gazebo and Infra in the OSRA.

3. Co-founder of Honu Robotics: consulting company doing Gazebo developments and installations.

A 24 year old 3D Simulator

The beginning

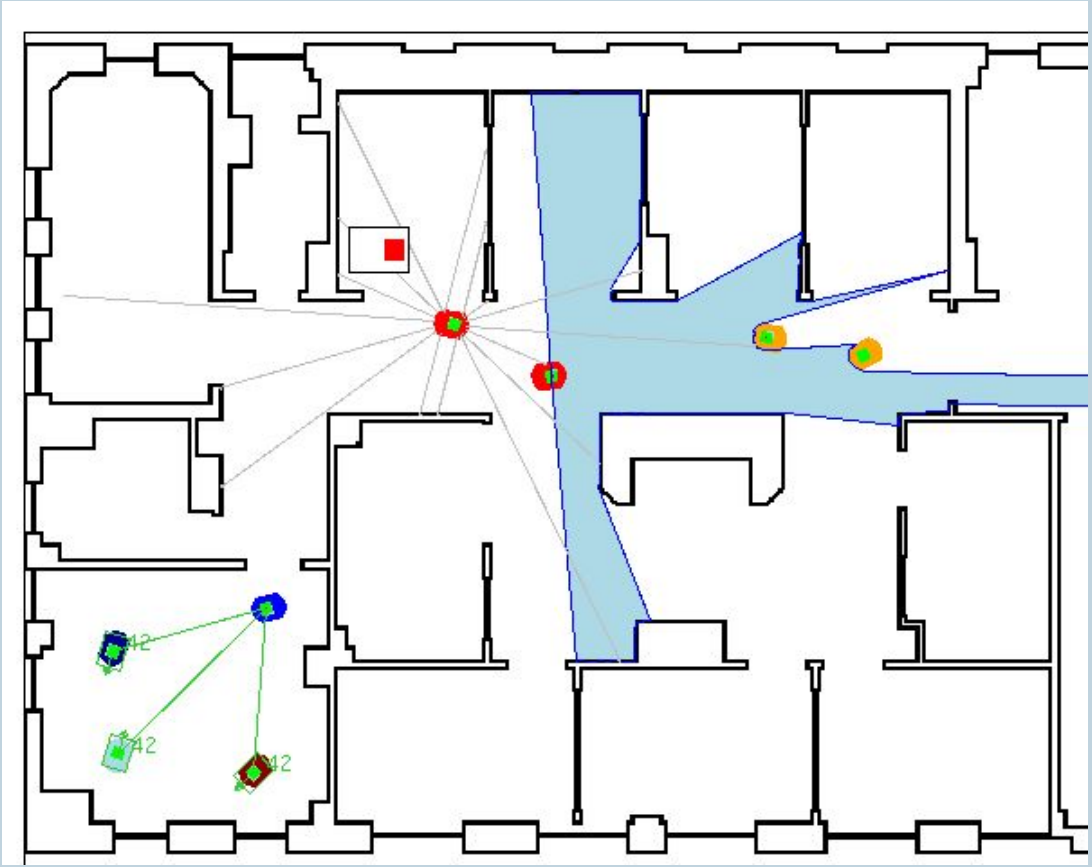
The beginning (2002-2007)

- Gazebo was originally created in 2002 at the University of Southern California by Andrew Howard and Nate Koenig.

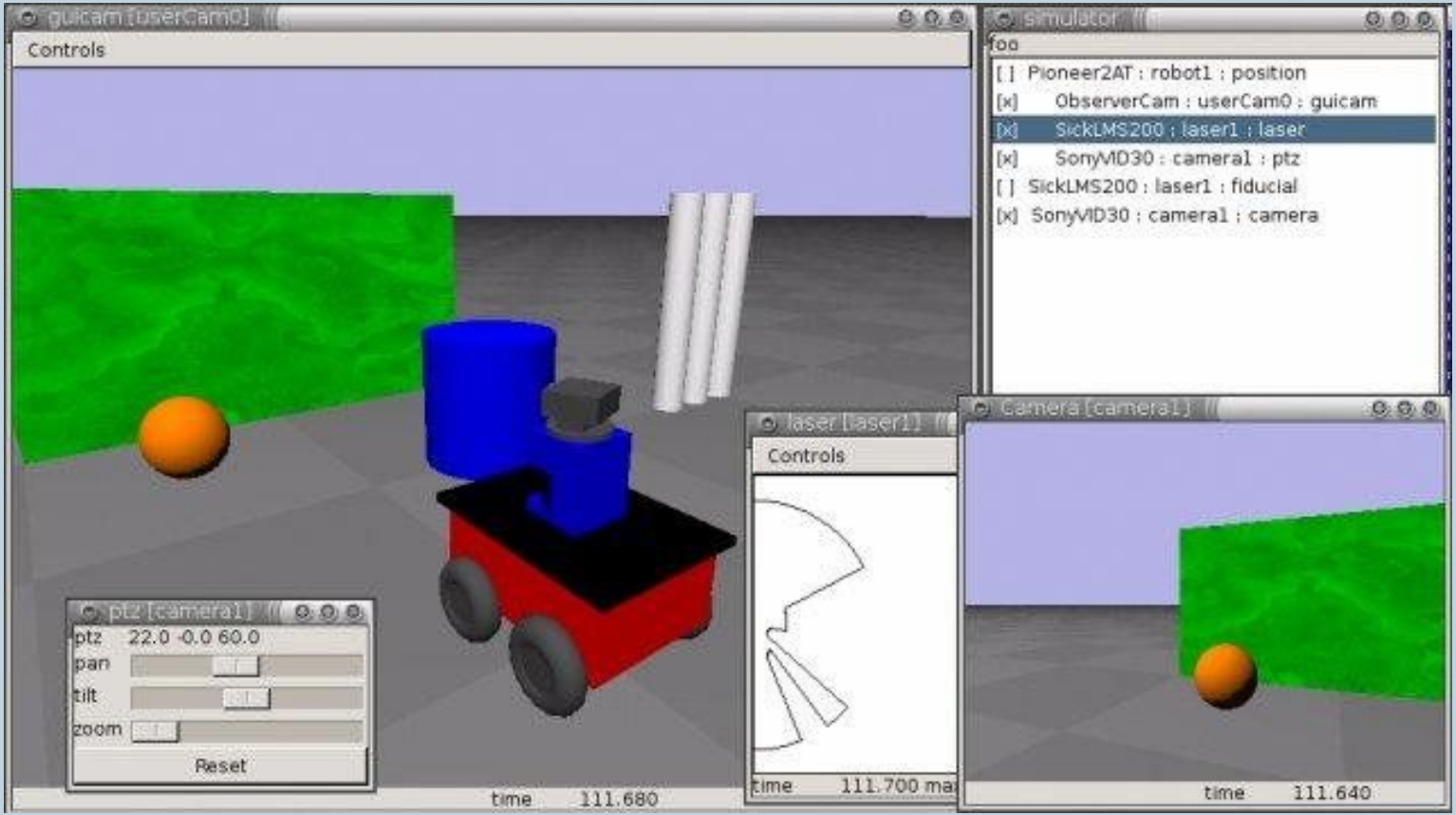
- It was built as an extension of the Stage / Player project that was basically a framework and a 2D robotics simulator.

- Physics was implemented using ODE (Open Dynamics Engine) and rendering was done directly using pure OpenGL.

Stage simulator 1.6 (2005)



Gazebo 0.5 (2004)



A 24 years old 3D Simulator

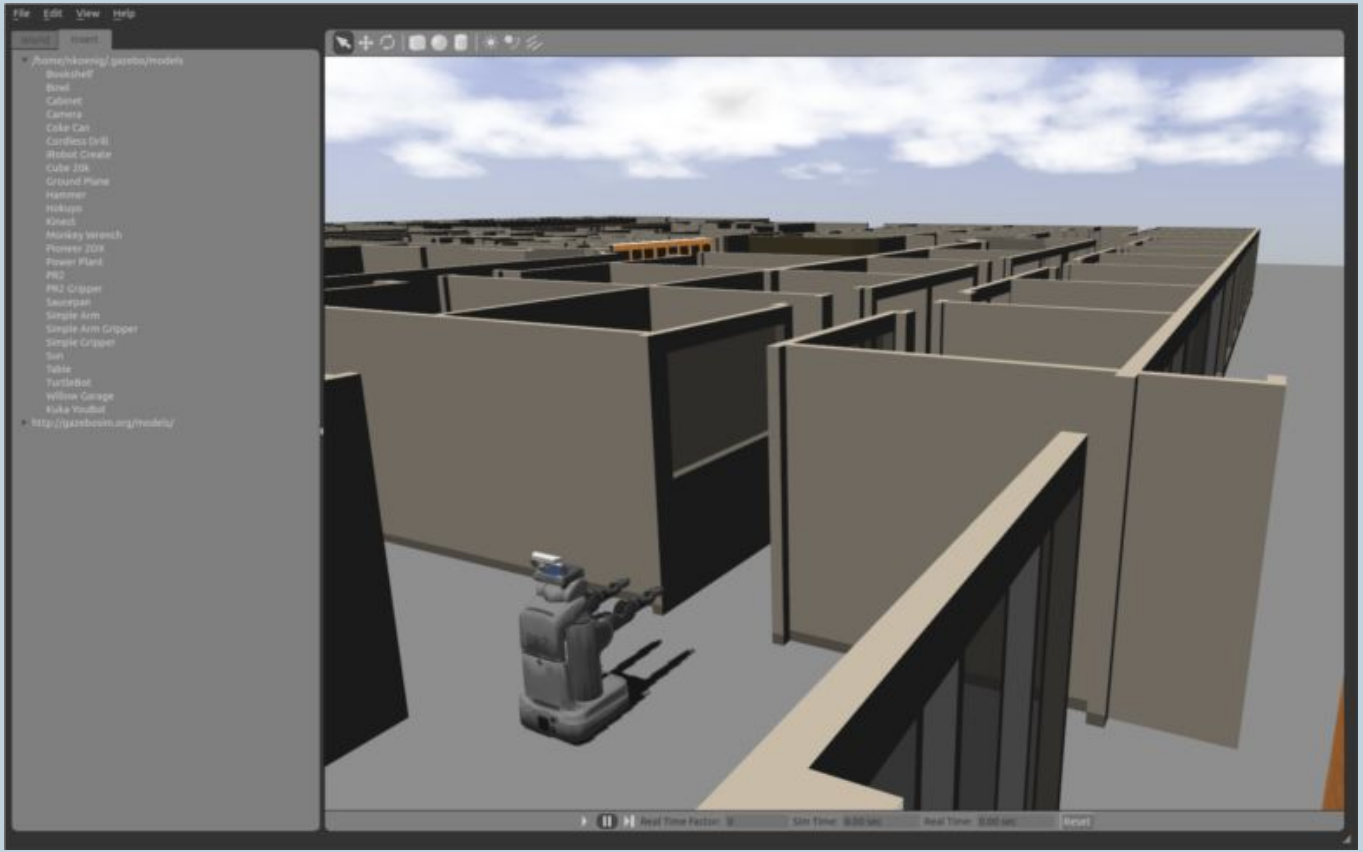
The evolution:
Willow Garage and Open Robotics

Evolution

Willow Garage marked a milestone in the Gazebo development. PR2 and ROS integration happened by 2009. OSRF continued the funding on Gazebo, started in 2012 and mostly supported by implementing external projects.

Open Source Robotics Foundation

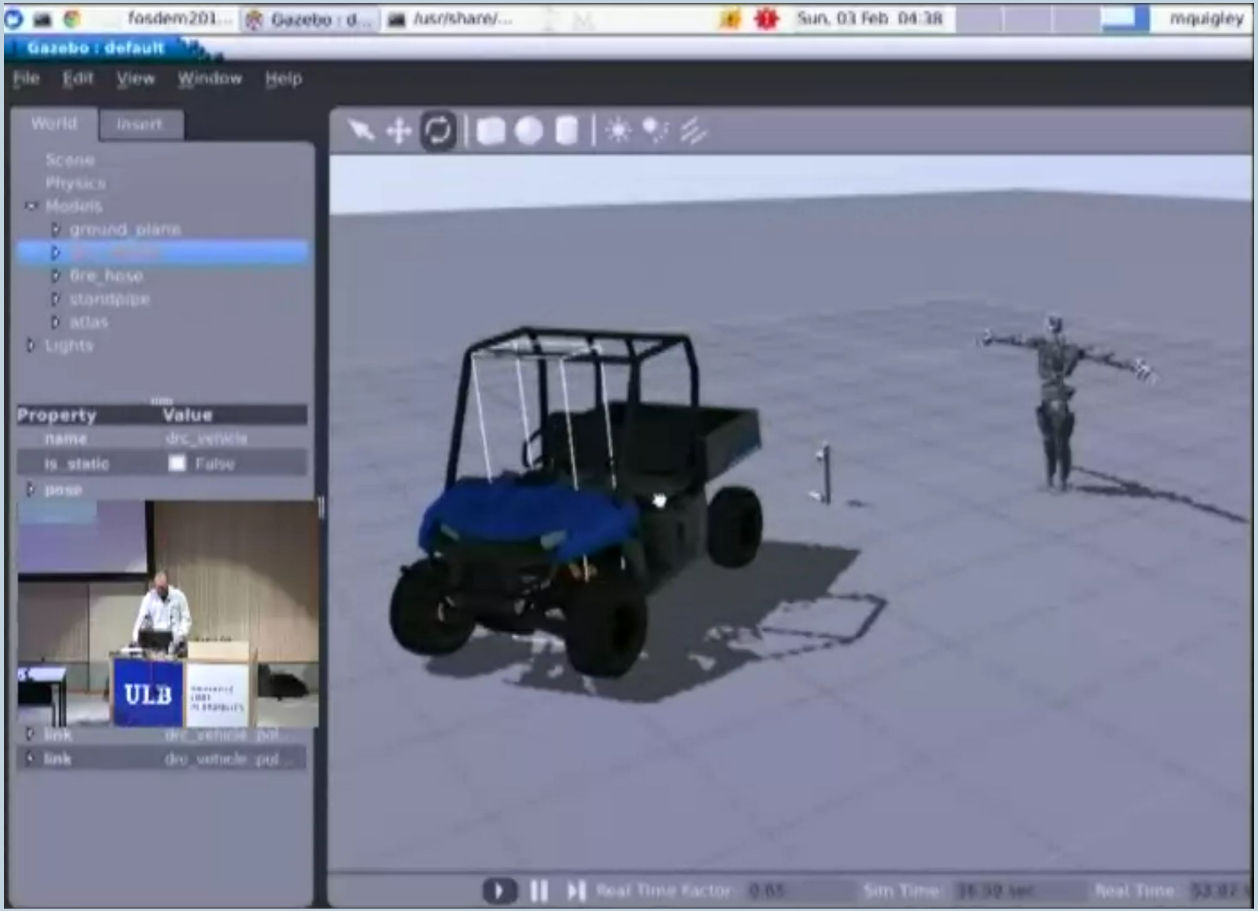
Gazebo 1.4 (2013)



Evolution

- From 2012 to 2015: DARPA Robotics Challenge competition pushed the Gazebo development into Humanoids.

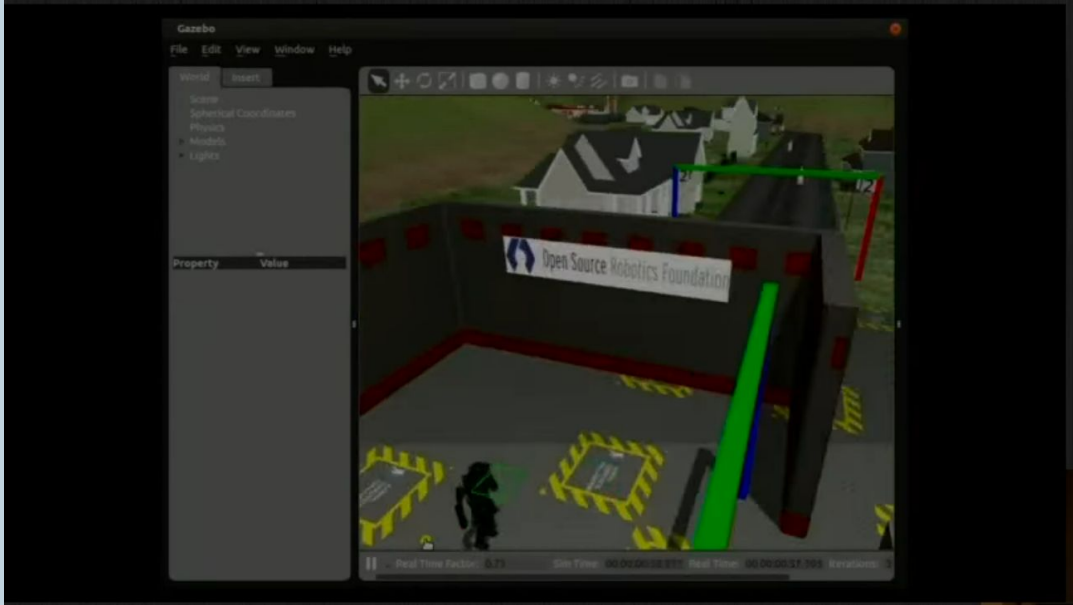
- 2014: Gazebo 3/4 implemented:
  - multi-physics engines: ODE, Simbody, Dart, Bullet.
  - DEM support.



https://archive.fosdem.org/2013/schedule/event/ros_open_sourcet_sobotics/

Simulating Humanoid Robots in the Cloud the testing behind the biggest world competition

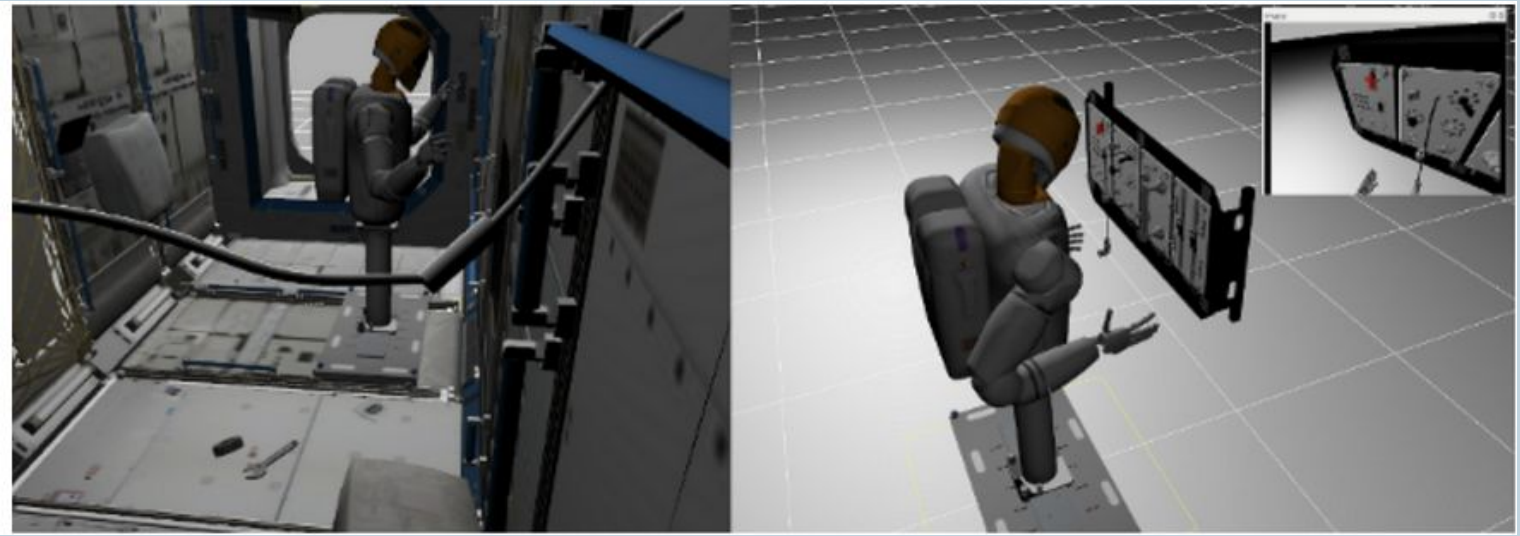
Jose Luis Rivero



FOSDEM 16.org

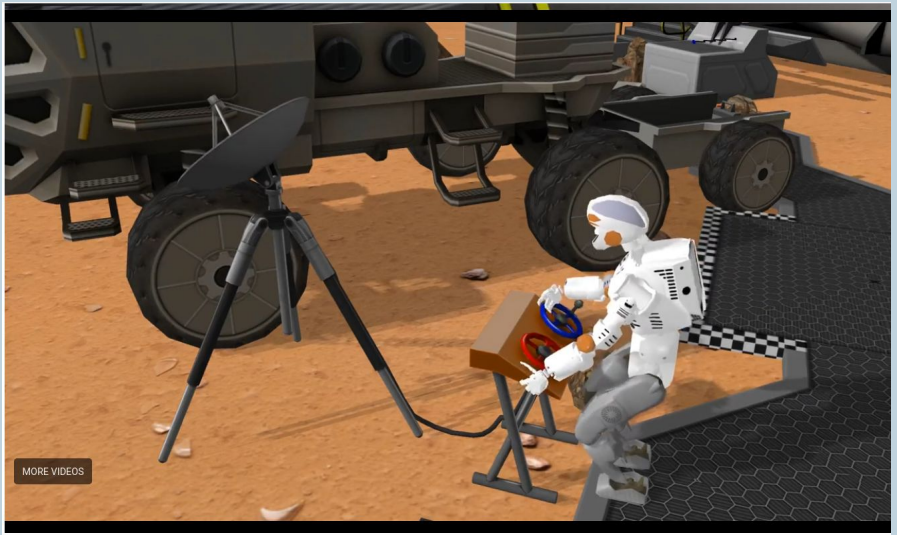
honu ROBOTICS

Evolution



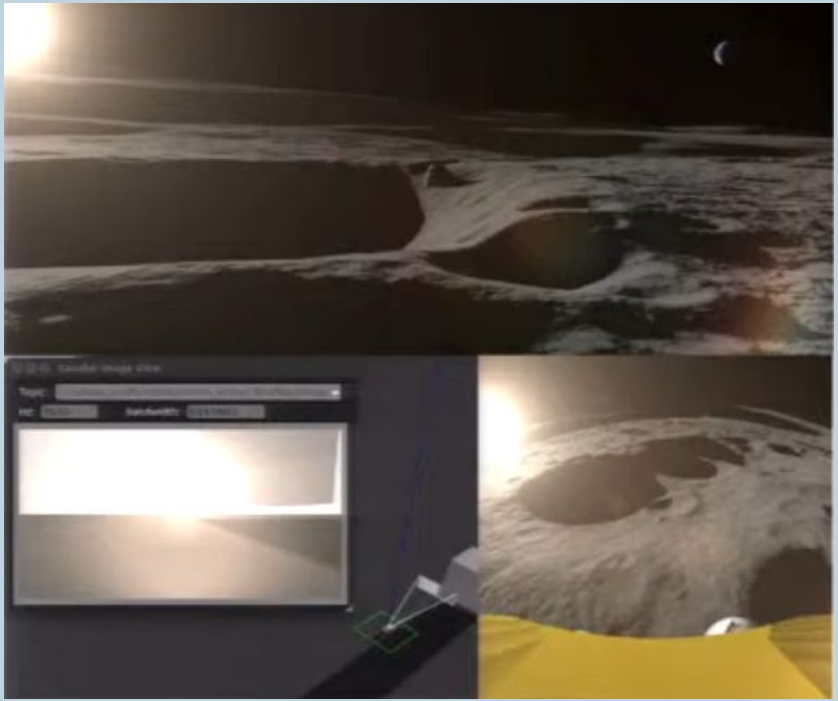
Space Robotics Challenge

- 2017 Space Robotics Challenge: teams to develope and displaying the ability of the Valkyrie (R5) robot to assist in a virtual NASA Mars mission.
- Gazebo 7/8 showcasing improved humanoid dynamics, sensor integration (Multisense, IMU), and external walking controller for bipeds.



Latest releases

- (2018) Gazebo 9
  - Improved shadows
  - Camera lens flare
  - Attach lights to links
  - Gravity compensation plugin
- (2020) Gazebo 11 (Latest)
  - SDF 7 Frame Semantics
  - BVH skeletal animations
  - Tracked vehicles with flippers
  - Spherical sonar sensor



Gazebo renders the moon
https://www.youtube.com/watch?v=bZ7TvBTrogQ

A 24 year old 3D Simulator

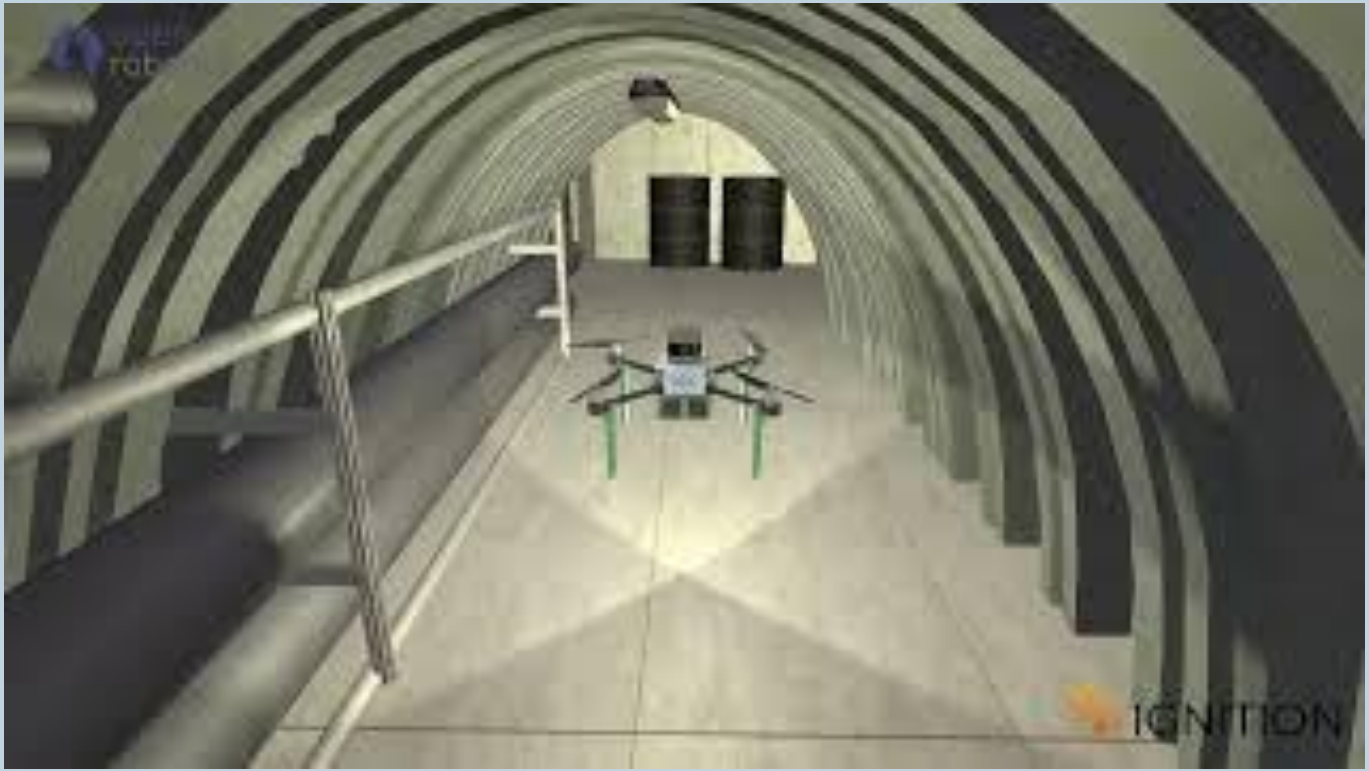
Rewriting Gazebo: the new Gazebo

(yes, previously known as Ignition)

Modular Architecture (Acropolis 2019)

- Entity Component System (ECS): new simulation engine architecture for improved performance and modularity.
- Modular library design: Separation into independent libraries (gz-physics, gz-rendering, gz-transport, gz-sim, etc.)
- Plugin-based system for rendering and physics.
- Modern versions: Ogre 2.x, DART 5.x, Protobuf 3.x, QtQuick

Rewriting Gazebo



The new Gazebo today

- Rendering (Fortress / Garden): Ogre 2.3, Vulkan support, Physically Based Rendering (PBR), Custom shaders, Particle effects, Lens flare.
- Physics: Bullet Featherstone (Garden), Trivial physics Engine (Dome), Auto inertia calculation (Harmonic).

The new Gazebo today

- ROS 2 Integration: Bidirectional bridge, Composable Nodes for performance (Ionic), Standard Simulation Interface (Jetty).
- Used in a variety of domains:
  a. Maritime: Hydrodynamics, buoyancy, ocean currents, added mass
  b. Ground vehicles: Tracked vehicles, mecanum wheels, dynamic wheel slip
  c. Aerial/Space, Manipulation, etc.

The Modern Times

What has changed in the modern times?

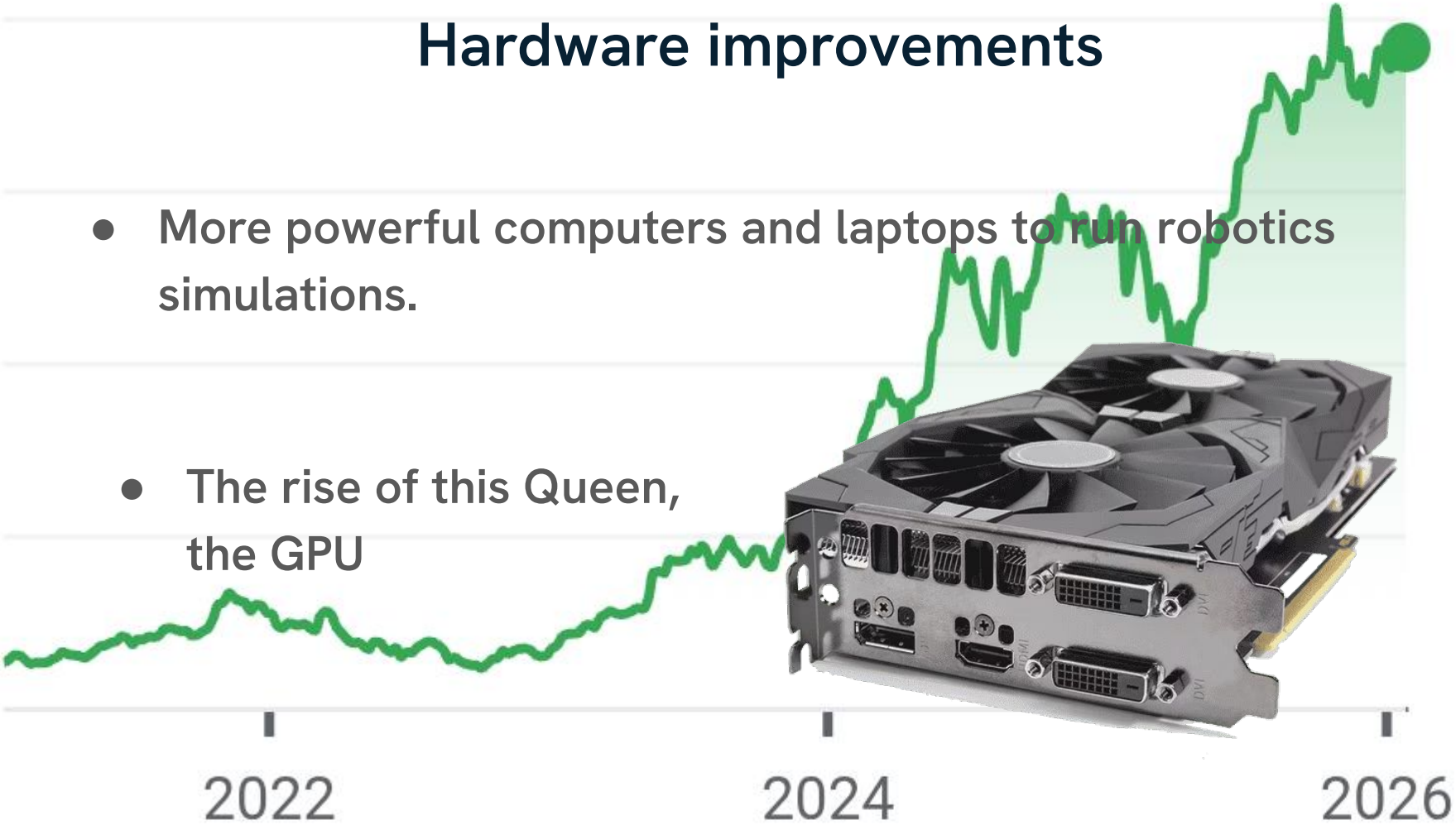
Computation power

Hardware improvements

- More powerful computers and laptops to run robotics simulations.
  - The rise of this Queen, the GPU

Hardware improvements

- More powerful computers and laptops to run robotics simulations.
  - The rise of this Queen, the GPU



Programming languages

Trends in programming

- New C++ standards, increase popularity of Go, Typescript being the most used language in 2025.
- Python: used more and more in high level robotics applications and adopted by most of the AI community.

- Rust: efficient, secure, low level language. Number of projects in all areas is growing very fast. Already present in mature projects like Linux Kernel or Firefox.

Package managers / Build systems

Evolution of package managers / build systems

- Hermetic sandboxing build systems based on rules, like Bazel.
- Package managers using functional language, acting in the border of building / deploying, like Nix.
- Environment Isolation with prebuilt binaries, like Conda. Lockfile based reproducible builds package managers, like Pixi.

Even mutations in package managers...

1. Pip / PYPI to install all kind of

Optimized data communications

Evolution and creation of new pub/subs

- DDS (Data Distribution Service) software evolutions: FastDDS, CycloneDDS + Iceroyx, etc. Implementing shared memory approaches.

“Middleware Pain? Meet Iceoryx2” by Michael at 14:25

- Zenoh (2020): written in Rust, it is a modern pub/sub/query system, supporting a variety of network topologies and bandwidth limited networks.

“ROS-Z: A Rust/Zenoh-native stack” Julien and Yuyuan at 17:00.

The robotics frameworks

New era robotics frameworks

- Dora-rs (2022)
  - Written in Rust. Zero-copy local comms using Apache arrow, Open Telemetry support, multi-AI and multi-hardware applications, Zenoh.

- Copper-rs (2024)
  - Written in Rust. Using Bevy for rendering and Avian3d physics engine.
  - Supports a declarative way of defining the system and it is fully reproducible.

New robotics simulations

Modern robotics simulations

- NVIDIA Isaac Sim™: released in 2019, built on top of NVIDIA Omniverse™ . Physics are implemented in NVIDIA® PhysX®. Integrated with NVIDIA Isaac™ Lab.

Isaac Sim 

- O3DE: initial version on 2021, it is a gaming engine with its own rendering using NVIDIA PhysX. There is an extension to connect it to ROS. Nice rendering capabilities, lots of GUI helpers.

O3DE 

Physics engines

- Mujoco: Google Deepmind physics library, first open source release in 2022. Good dynamics support.
  a. Powerful extensions to GPU computation: Mujoco-MJX and Mujoco-Warp.
  b. It has its own renderer based on OpenGL, can also render headless using EGL.

Machine learning / AI

Machine learning / AI

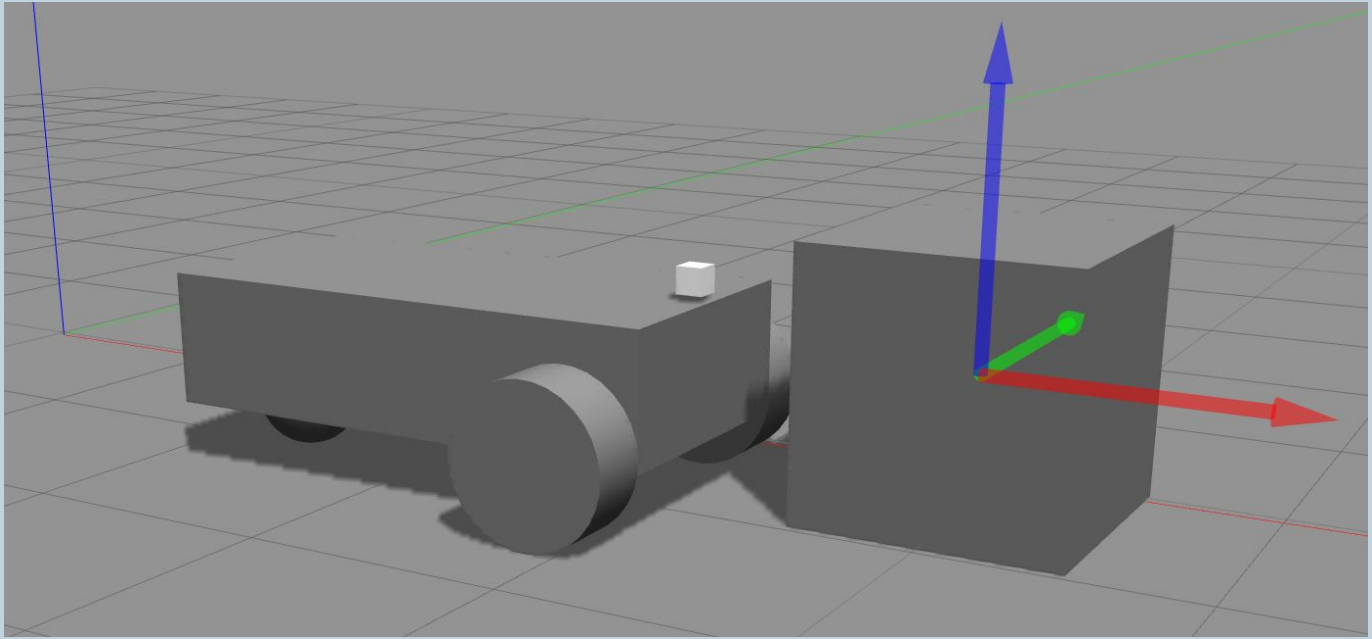
- Adoption or Deep learning frameworks: Tensorflow and Pytorch popularit. Supporting GPU / TPU acceleration, neural networks and being used through Python.

- Robotics Gyms : Isaac Lab robotics learning framework built on NVIDIA Isaac Sim, designed for reinforcement learning and imitation with GPU-accelerated physics simulation accelerated physics.

Hands on:
Gazebo current and future

“Photorealism”

“Gazebo is not photorealist”



A personal take on realistic worlds

- The rendering engine plays an important part in creating a realistic rendering. But, it is the most important part?
- Have you tried to open IsaacSim / O3DE and create a model from scratch? The work of the model designer and the rendering engineer tweaks are important too.
- Defaults rendering features plays an important role. Can you run several photorealistic simulators in your laptop?

Same script
Different simulators

O3DE

GAZEBO

Isaac Sim

Ongoing rendering works

- Expected for next release: Gazebo Kura

Roadmap | Primary
GAZEBO State: ongoing

Improve rendering quality

- Explore how Gazebo can leverage modern rendering engines that provide higher rendering fidelity.
- Potentially create a new way of integrating external rendering engines running in a separate process.
- Curate models and worlds with high rendering fidelity for reference
- POC: iche033

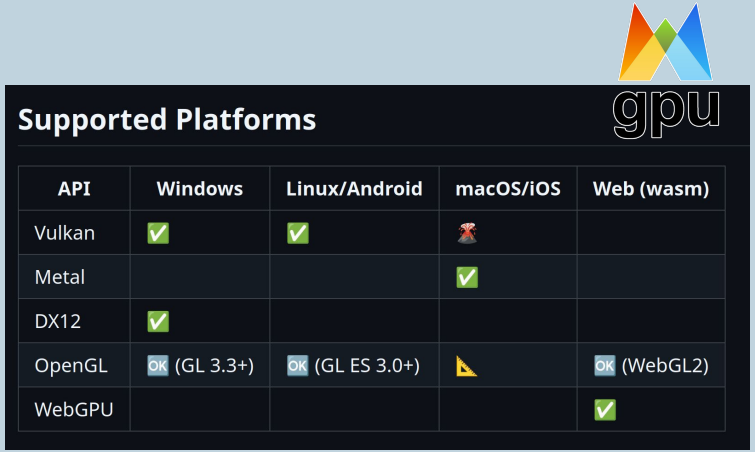
- Intrinsic Gazebo Hackathon (Dec 25)
  a. Analysis of existing open source alternatives: Godot, Google Filament, Bevy. Goal: proof of concept.

Hackathon | Intrinsic
GAZEBO State: finished

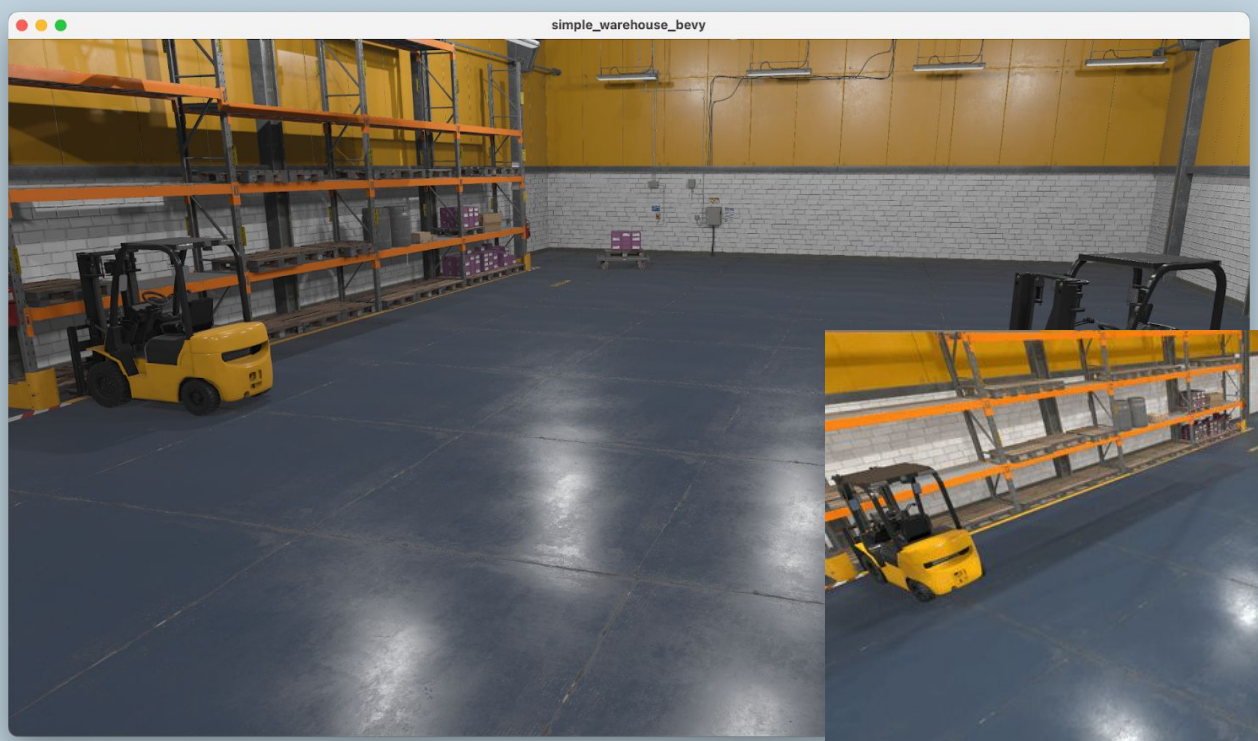
Ongoing rendering works

- Bevy appeared in August 2020, it is a gaming engine (2D/3D), written in Rust and WGPU. Cross-platform, ECS based, fast, with a 2D/3D renders and animations.

- WGPU is a multiplatform rust graphics API that calls other graphics APIs depending on the platform:



Gazebo vs Bevy



Hackathon | Intrinsic

GAZEBO State: finished

Add design document for Bevy based external renderer #10

Draft **azeey** wants to merge 2 commits into `gazebosim:main` from `azeey:external_renderer`

Conversation 5 | Commits 2 | Checks 1 | Files changed

azeey commented on Dec 15, 2025 — Member

Revie

Design | PR

GAZEBO State: draft

external_renderer.md +464 Viewed

Gazebo ↔ Bevy External Rendering Architecture

Metadata	Details
Status	Proposed
Objective	Decouple rendering from Gazebo physics to enable Bevy-based visualization and sensor simulation.
Core Stack	**Gazebo** (C++), **Bevy** (Rust), **Zenoh** (IPC/SHM), **FlatBuffers** (Serialization).
Target Latency	< 16ms (60 FPS)

---

1. Executive Summary

This document defines the architecture for moving the rendering workload out of the Gazebo simulation process into a dedicated external process running the Bevy game engine. This allows us to utilize Bevy's modern ECS and WGPU rendering pipeline while keeping the physics simulation stable in Gazebo (gz-sim).

The system uses **Zenoh** middleware over **Shared Memory** with **FlatBuffers** for high-performance, zero-copy communication. Bevy acts as a "Thin Client" for visualization and a "Render Farm" for sensors, while Gazebo retains authority over simulation and physics.

2. High-Level Architecture

“zero-copy” communications

gz-transport communications

- gz-transport is based on ZeroMQ and Protobuf. Custom code avoids intra-process serialization and avoiding more than one copy of the messages.

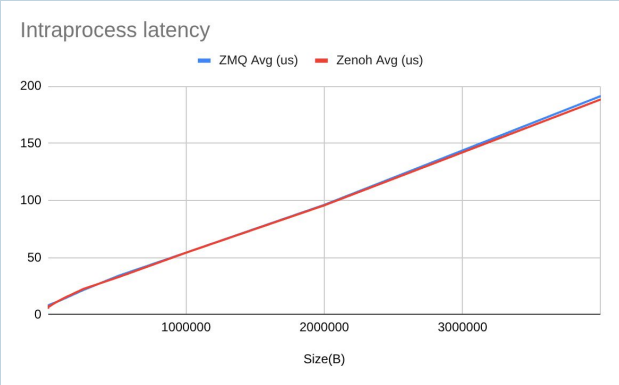
- Currently gz-transport supports the use of Zenoh since the Gazebo Jetty release as an alternative to ZeroMQ.

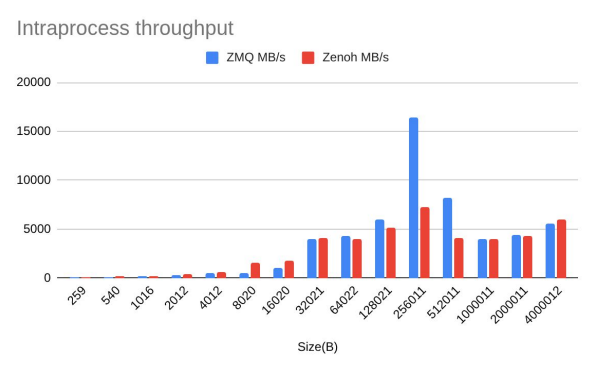
Zenoh/ZMQ initial benchmark

Worse

Latency

Better

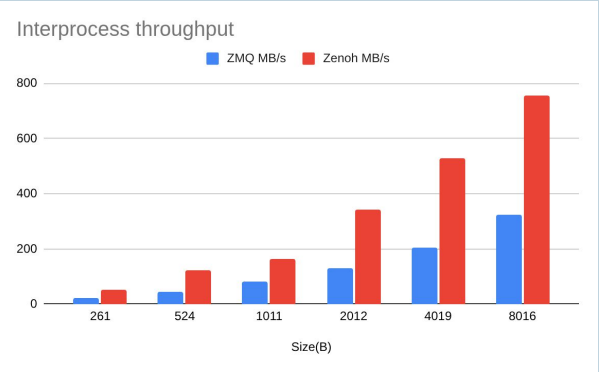




Better

Throughput

Worse



gz-transport communications

Roadmap | Primary
GAZEBO State: ongoing

- Expected for the next release

Zenoh Refinement

- Continue refining the recent Zenoh integration work in gz-transport to make it production ready
- Add support for loading Zenoh configuration from file
- Consider supporting shared memory transport over Zenoh
- POC: caguero

Implement the shared memory capabilities of Zenoh in gz-transport.

Physics Engines

Mujoco physics engine

- The Gazebo PMC has selected Mujoco physics Engine as the candidate to be supported by gz-physics, the fourth engine supported.

Roadmap | Primary
GAZEBO State: ongoing

Add MuJoCo as a new physics engine

- MuJoCo is known for its speed and superior handling of contact dynamics. In addition it is actively maintained and has a thriving community of users. Integrating MuJoCo as Gazebo's fourth physics engine will ensure that Gazebo can offer a more efficient physics engine while also ensuring the long term health of the project.
- POC: azeey

Mujoco prototype

Hackathon | Intrinsic
GAZEBO State: finished

Mujoco integration prototype #811

Draft **azeey** wants to merge 16 commits into `gz-physics9` from `mujoco`

Conversation 11 · Commits 16 · Checks 9 · File

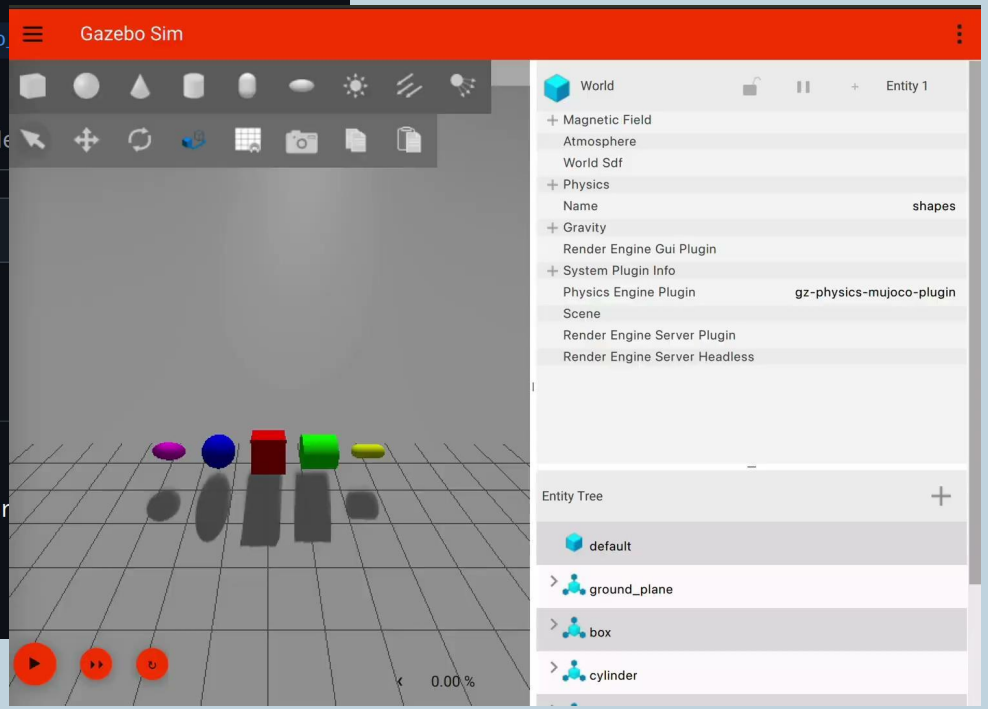
azeey commented on Nov 19, 2025 • edited

[skip ci]

Set up

1. Use a colcon workspace with the Jetty collection
2. Clone https://github.com/google-deepmind/mujoco into the wor
3. Build!

As of `fde1d15`, it is now possible use the plugin in gz-sim

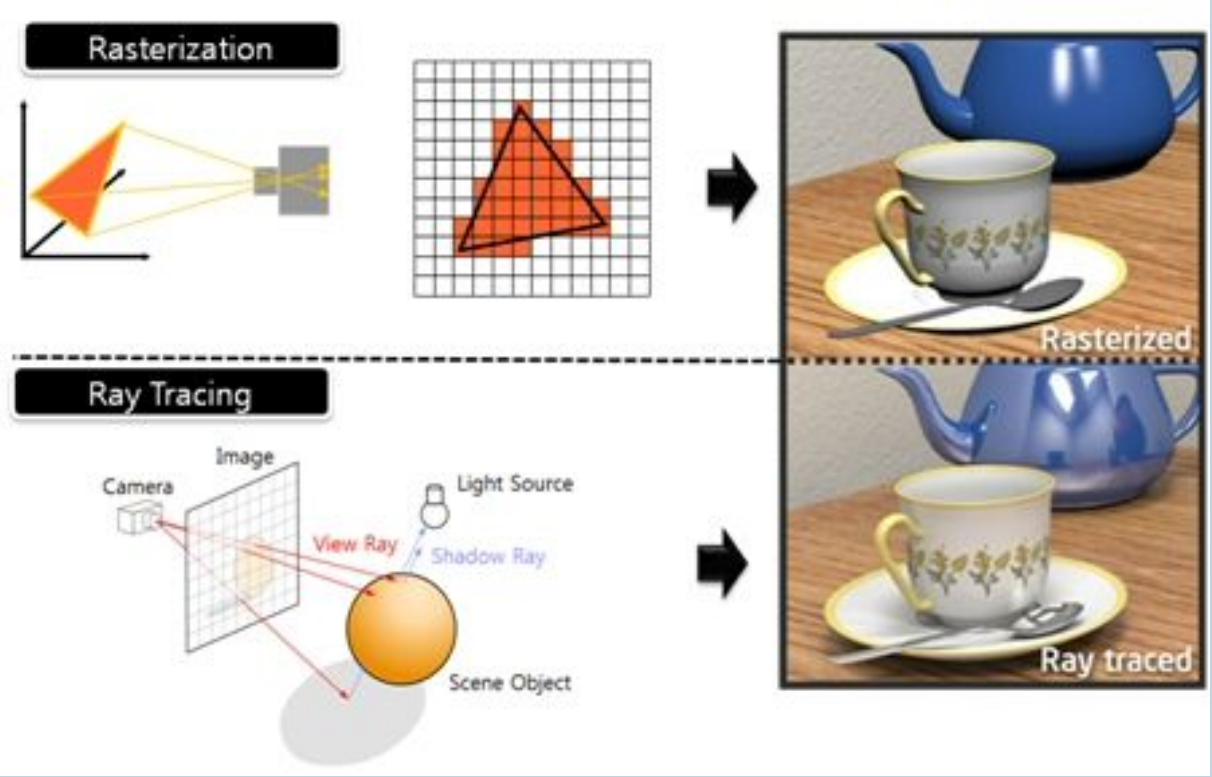


Sensor performance

Gazebo’s Rendering Speed has been something of a meme

Ray-Tracing Everywhere Vendor Agnostic Ray Tracing in Gazebo
https://vimeo.com/1136163508

Rasterization vs Ray tracing



https://www.pixelsham.com/2019/10/24/whats-the-difference-between-ray-tracing-and-rasterization/

WGPU and RayTracing

- WGPU opens the door for Gazebo to use a modern graphic API: open source, multi platform, GPU accelerated.

API	Usability	Graphics Cards Supported	OSes Supported	Ray Tracing supported
Cuda OptiX	Easy	NVidia only	Windows, Linux	Yes
Vulkan	Hard	Many	Windows, Linux	Yes
OpenGL	Easy	Many	Windows, Linux	No
Metal	Easy	Apple Only	Mac OS	Yes
DirectX	Medium	Many	Windows	Yes
WGPU	Easy	Many	Windows, Linux, Mac OS	Experimental

WGPU and RayTracing

- Experimental raytracing wgpu sensors for lidar and camera are implemented.

GSoC 25 | Google
GAZEBO State: ffinished

README Contributing Apache-2.0 license

WGPU based Raytracing Sensors for Gazebo

This project builds on the Raytracing sensors provided by the WGPU sensors library. Currently, this is a proof-of-concept stage and can render boxes, planes, and meshes, with ongoing efforts to expand its capabilities.

Current Status

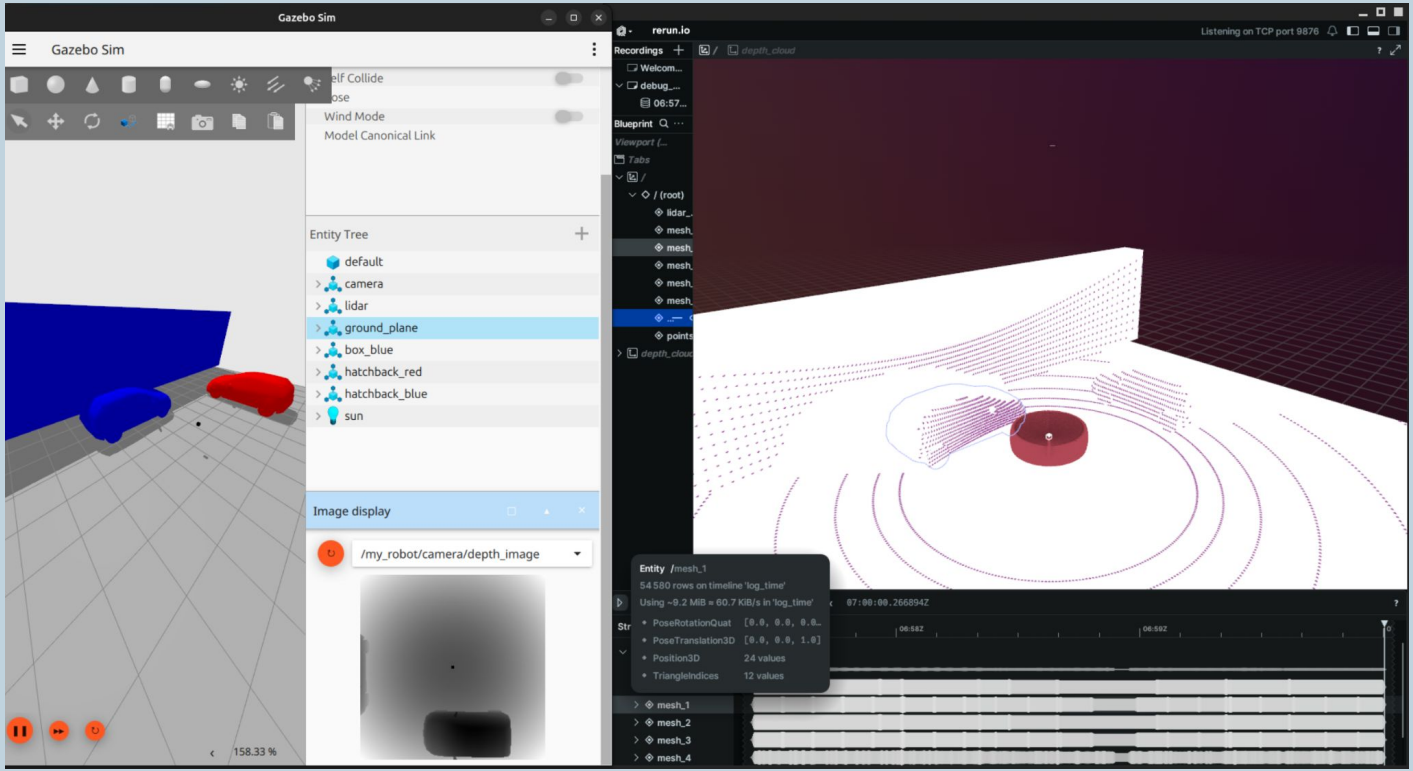
This project is in active development. The core framework is functional, supporting both LiDAR and Depth Camera sensors that can run simultaneously. It can accurately render scenes composed of meshes, boxes, and planes.

Summary of current features:

- **Raytracing Sensors**: Provides custom `rt_lidar` and `rt_camera` sensors for Gazebo.
- **Multi-Sensor Support**: Capable of running multiple raytracing sensors simultaneously.
- **Geometry Support**: Currently supports rendering of boxes, planes, and meshes.
- **Data Output**: Publishes standard Gazebo Transport messages.
  - Depth Camera: `gz::msgs::Image` is fully supported and can be visualized with Gazebo's Image Display plugin.
  - LiDAR: `gz::msgs::PointCloudPacked` is published. For the best experience, real-time visualization is provided via the Rerun viewer, which launches automatically.

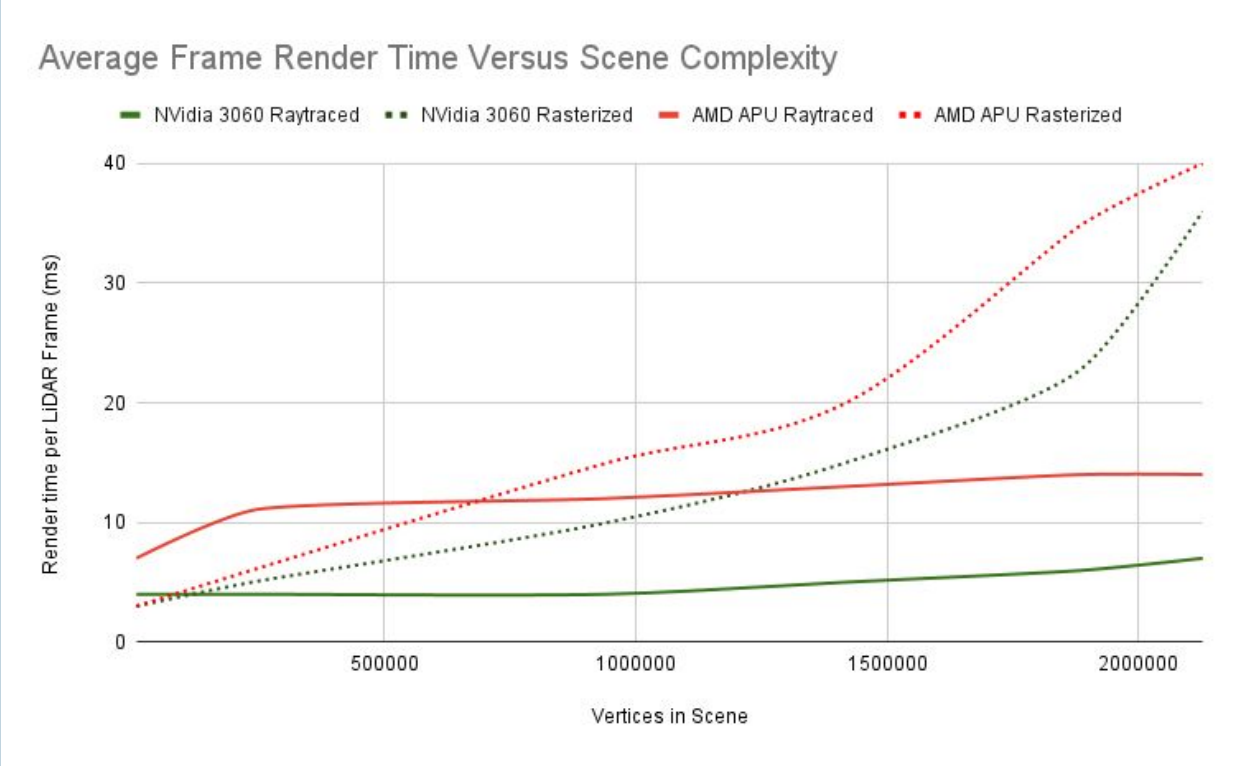
- Shashank Rao & Arjo
  https://github.com/arjo129/gz_wgpu_rt_lidar

WGPU Gazebo Ray tracing



ROSCon26 Singapore https://vimeo.com/1136163508

WGPU Gazebo Ray tracing



ROSCon26 Singapore https://vimeo.com/1136163508

OSRA Investment: 250.000$

OSRA's Technical Governance Committee Approves $250,000 Funding for Infrastructure and Documentation Enhancement

OSRA Announcements and News osra-tgc

gbiggs CTO at Open Robotics 🙃 Aug 2025

The Technical Governance Committee (TGC) of the Open Source Robotics Alliance (OSRA) has approved funding of US$250,000 to significantly enhance project infrastructure and documentation across the OSRA ecosystem. This funding will focus on maximising value across all OSRA projects through strategic improvements in two key areas.

Enable the use of Cargo-provided dependencies in the buildfarm

The Infrastructure PMC put forward a proposal to support Cargo dependency management in our buildfarm. The goal of this work is to enable packages being built by the build farm to depend on external Cargo packages. The end result will be ROS packages being able to leverage Rust libraries via Cargo-distributed packages. This work will have a broad impact across all OSRF projects.

Fundamental work to improve documentation across all OSRF projects

Improvements to our documentation is one of the most common requests from both users of OSRF projects, who often find it difficult to use, and contributors, who would prefer to be implementing new features rather than writing documentation. Documentation is also the foundation of a happy community and a healthy project, though, so it is something that we cannot ignore. Fortunately, all PMCs are aware of the need to improve their documentation and all requested funding this year to help them do so. This item combines the requests from all PMCs into a single item that will provide the fundamentals for good documentation that all projects can leverage going forward.

Funding | OSRA

open source robotics alliance State: ongoing

Funding | OSRA

open source robotics alliance State: ongoing

Packaging / Building

Pixi / Conda-forge

- Currently, there are updated conda-forge packages for all the Gazebo stable versions.
  a. Pixi is the default method of installation for Windows.

Gazebo | Release
GAZEBO State: released

- Conda-forge + Pixi is the target for internal MacOSX in the next release cycle

Roadmap | Secondary
GAZEBO State: TODO

Replace homebrew with Pixi in our macOS CI

- Due to the rolling nature of homebrew, the maintenance burden of building bottles and the associated CI disruptions have become too much. Pixi on the other hand can provide version locked binaries.
- POC: j-rivero

Bazel

- Bazel support:
  - Already in Gazebo Jetty: Bazel Bzlmod migration, BCR publishing.
  - Gazebo PMC will vote to adopt Bazel as officially supported next week.

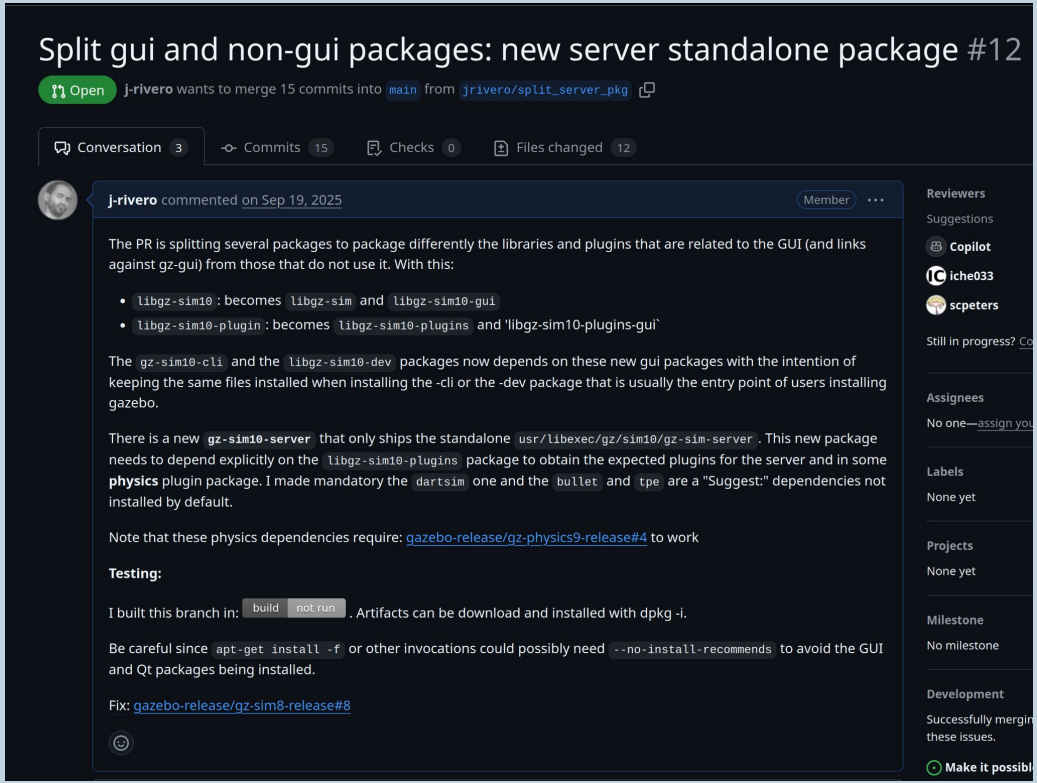
Gazebo in your CI pipeline

Gazebo as part of a CI pipeline

- Headless rendering support: EGL through Ogre-next since Gazebo Fortress.

- Setup-gazebo github action
  - https://github.com/gazebo-tooling/setup-gazebo
- Gazebo pre-built containers (OCI images)
  - https://github.com/j-rivero/gz_oci_image

Gazebo as part of a CI pipeline



- Server only installation: not the client, not the Qt stack nor the dependencies.

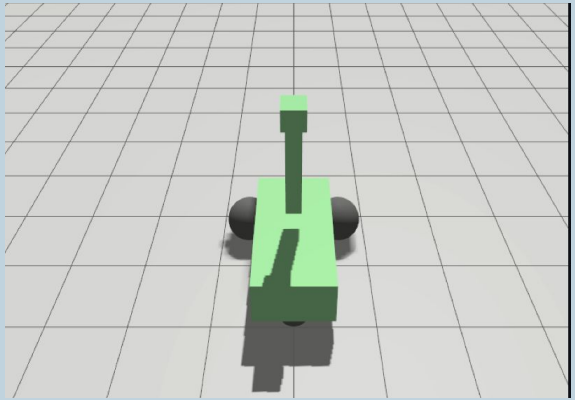
gz-sim | PR

GAZEBO State: in review

Machine learning / AI

Reinforce Learning Example

Already merged an example that uses RL (Stable BaseLines3) for implementing a cartpole example with a tutorial.



Name	Last commit message	Last commit da...
..		
README.md	Fix cpplint, codespell complaints (#2938)	7 months ago
cart_pole.sdf	Set default camera angle in cart_pole example (#3159)	3 months ago
cart_pole_env.py	Fixes the RL Demo Crash (issue #3100) (backport #316...	2 months ago

README.md

Example for Reinforcement Learning (RL) With Gazebo

This demo world shows you an example of how you can use SDFormat, Stable Baselines 3 and Gazebo to perform RL with python. We start with a very simple cart-pole world. This world is defined in our sdf file `cart_pole.sdf`. It is analogous to the cart-pole world in gymnasium.

Funding | OSRA

open source robotics alliance

State: ongoing

https://github.com/gazebosim/gz-sim/tree/main/examples/scripts/reinforcement_learning/simple_cart_pole

https://github.com/j-rivero/nightly_gz_sim_reinforce_learning/

Gazebo as niche simulator?

Should Gazebo focus in only a part of robotics simulations?

Gazebo has been and it is an open source project, now under the governance of the OSRA, for almost 25 years.

It is not a product, it is not part of a company.

It’s being pushed by public contributions that respond to a variety of social and personal interest. It is not under the direction of a product manager.