Autonomous MicroMouse Robot

Elman Steve Laguna

October 22, 2025

Project Status and Technology Stack

This project combines multiple technologies to create a simulation of a micromouse competition robot.

Core Technologies

Rust Programming Language The entire simulation is built using the Rust Programming Language.

Bevy Game Engine (v0.16.1) Bevy provides the foundational framework for the simulation.

Physics Simulation - Avian2D Avian2D provides realistic 2D physics simulation.

Supporting Libraries

bevy_ecs_tilemap Specialized tilemap rendering for maze visualization.

Knossos Procedural maze generation library.

Python Ecosystem

Python handles data analysis and visualization.

Architecture Integration

The system follows a modular pipeline architecture:

1. **Rust/Bevy**: Real-time simulation, physics, and sensor processing.
    - Maze generation (Knossos)
    - LiDAR simulation (custom raycast system)
    - Robot control (WORK IN PROGRESS)
    - Physics simulation (Avian2D)
2. **Data Export**: Sensor readings and position data serialized to JSON via `serde`.
3. **Python Analysis**: Post-processing and visualization.
    - Occupancy grid construction from LiDAR scans

Development Tools

UV Package Manager Fast, reliable Python dependency management.

Cargo Rust's build system and package manager, handling all compilation and dependencies.

Design Rationale

This technology stack was chosen to balance several competing requirements:

Performance Rust provides low-level performance with a modern syntax.

Safety Compile-time guarantees prevent logic errors that would be runtime bugs in Python/C++.

Modularity ECS enables easy experimentation with different components.

Analysis Python is super easy to use for data analysis and visualization.

Personal Growth Learning Rust and Bevy is a personal goal.

Small Maze Generation

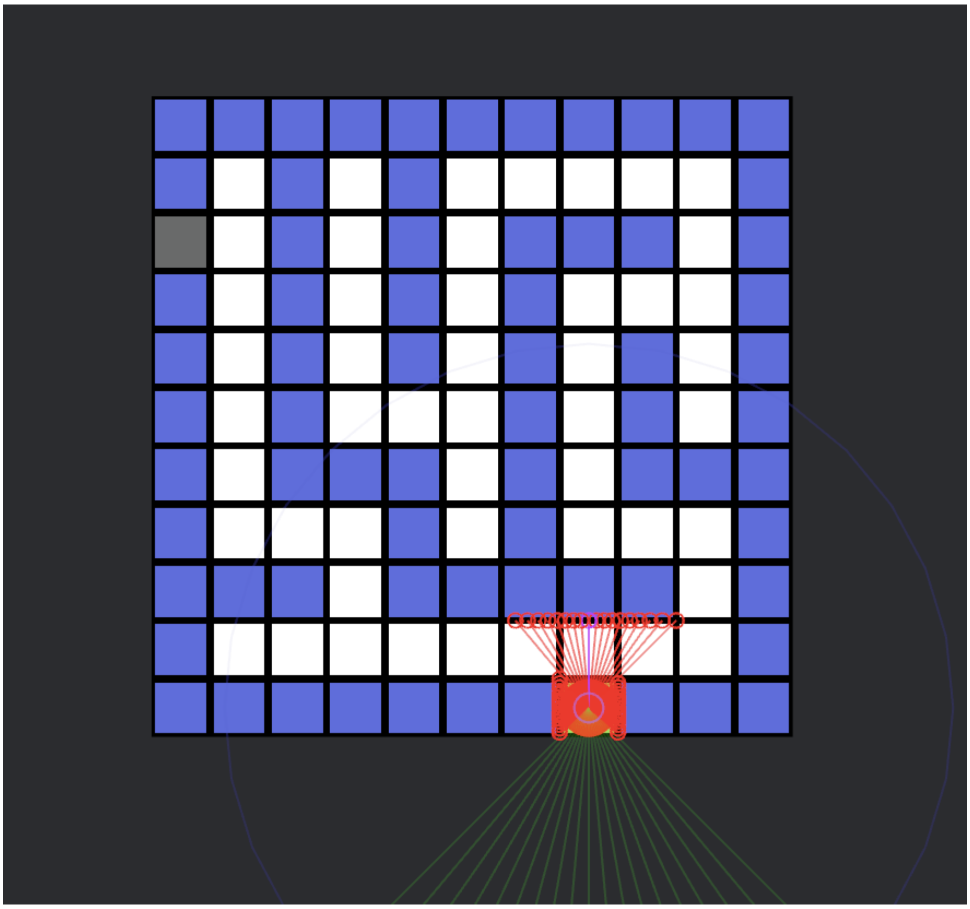


Figure 1: A 5×5 maze configuration generating an 11×11 tile grid.

Medium Maze Generation

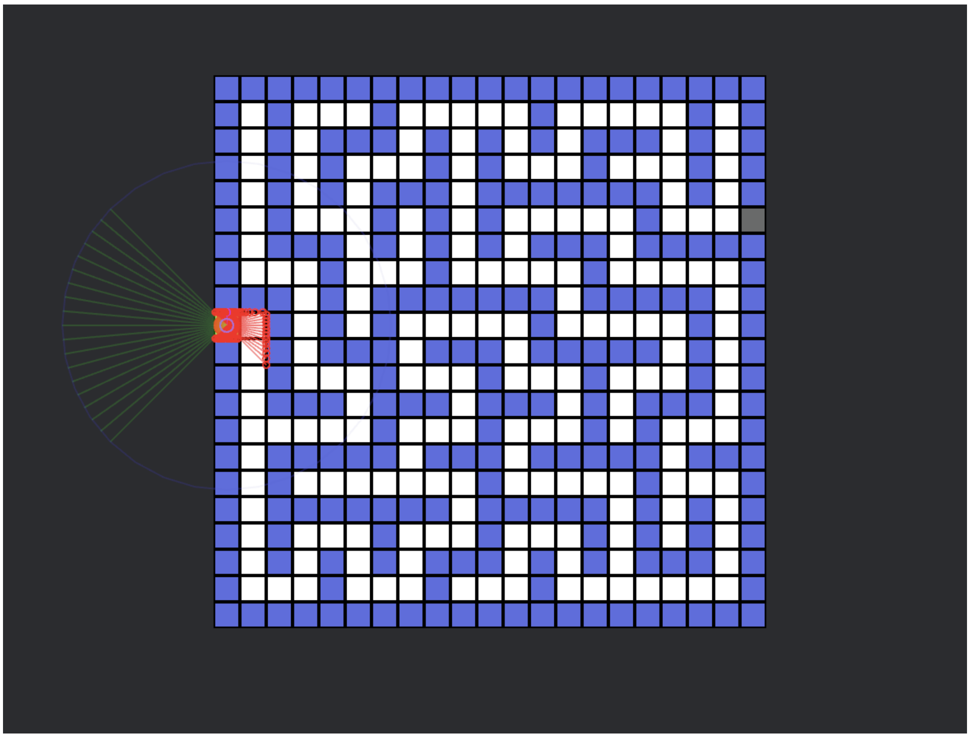


Figure 2: A 10×10 maze configuration generating a 21×21 tile grid.

Large Maze Generation

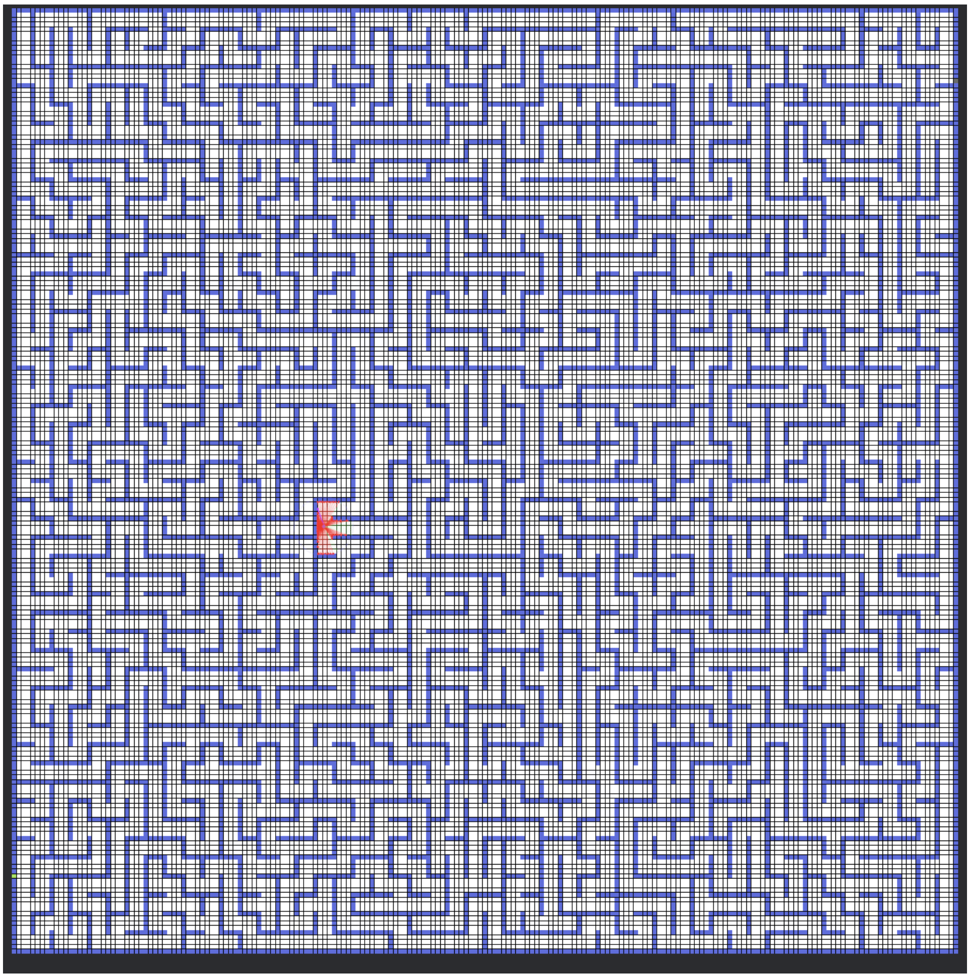


Figure 3: A 50×50 maze configuration generating a 101×101 tile grid.

Maze Generation

The maze generation system uses the `knossos` library with various algorithms to create procedurally generated mazes. The generation process follows these key steps:

1. **Cell Grid Creation**: An initial grid of `MAZE_WIDTH` $\times$ `MAZE_HEIGHT` cells is created, representing potential passages in the maze.
2. **Wall Insertion**: The Recursive Backtracking algorithm carves passages through the grid, with walls placed between cells and around the perimeter.
3. **Dimension Calculation**: The final tile count follows the formula:

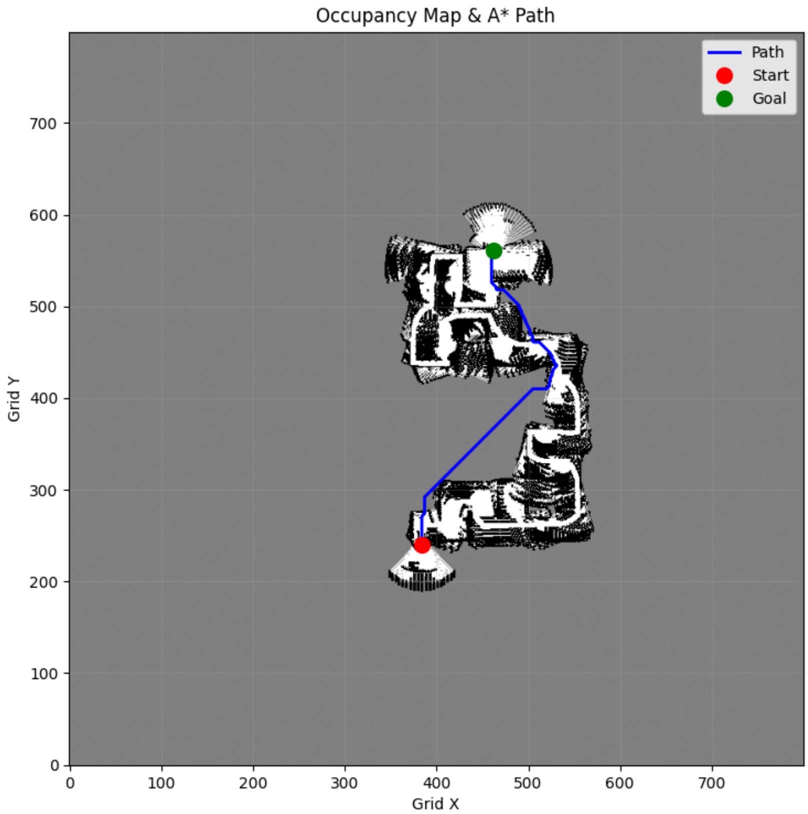
$$\text{Final Size} = (\text{MAZE\_WIDTH} \times 2 + 1) \times (\text{MAZE\_HEIGHT} \times 2 + 1) \tag{1}$$

   This accounts for the cells, walls between cells, and border walls.
4. **Start and Goal Placement**: The algorithm automatically places a start position and goal position within the generated maze using a configurable seed for reproducibility.

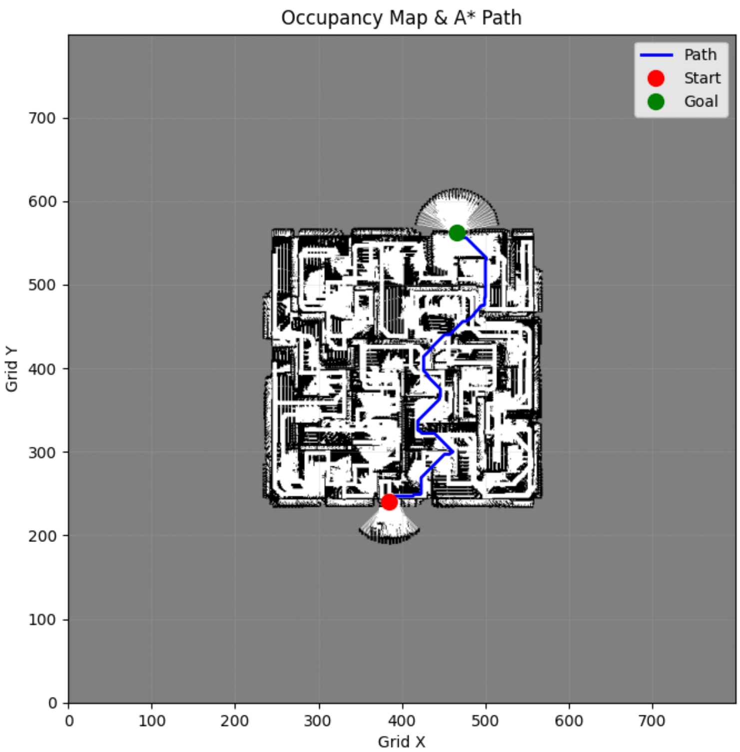
The use of a fixed seed (`SEED = 490`) ensures that the same maze configuration is generated consistently for testing and comparison purposes. The `GAME_MAP_SPAN` parameter controls the spacing between maze elements, set to 1 for standard wall thickness. Turns out that using too thick of a wall makes this a different problem. I have to start using slam techniques.

Pathfinding Results

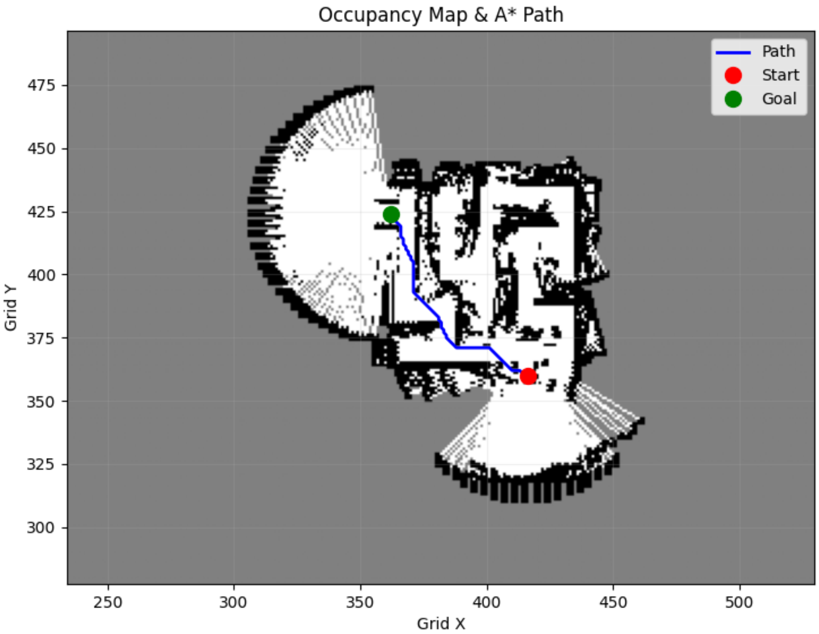
Poor Quality Data or Bad Post Processing



Improved Data or Post Processing



Small Maze (Post Processing)



Pathfinding Algorithm

The A* algorithm is used for maze solving in Python and Matplotlib to visualize.

Current Challenges and Future Work

Data Collection

- Manual robot control for LiDAR data acquisition is tedious and time-consuming
- Need to implement autonomous navigation

Technical Knowledge Gaps

- SLAM (Simultaneous Localization and Mapping) fundamentals require further study
- Mapping relationship between physical and simulated environments unclear
- Occupancy grid construction from LiDAR data needs research

Abstraction Level

- Struggling to understand and reduce infinite real-world detail to finite usable data
- Should focus on tile-to-tile navigation rather than millimeter precision

Scope Reduction

- Prioritize discrete tile-based movement over continuous positioning
- Focus on traversal between adjacent tiles as primary navigation unit

References

[1] M. Aborizky, H. A. As'ari, A. Fadlil and R. D. P. W. "Lidar-Based 2D SLAM for Mobile Robot in an Indoor Environment: A Review," *2021 9th International Conference on Information and Communication Technology (ICoICT)*, 2021, pp. 488-493. doi: 10.1109/ICoICT52021.2021.9538731.

[2] L. Vandevenne, "Raycasting," *Lode's Computer Graphics Tutorial.* [Online]. Available: https://lodev.org/cgtutor/raycasting.html

[3] NEON Science, "Lidar Basics," *National Ecological Observatory Network*, 2021. [Online]. Available: https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics

[4] Zona Land Education, "Intersection of Two Lines," *Zona Land Education.* [Online]. Available: http://zonalandeducation.com/mmts/intersections/intersectionOfTwoLines1/intersectionOfTwoLines1.html

[5] NEON Science, "How Does LiDAR Remote Sensing Work? Light Detection and Ranging," *YouTube*, Nov. 24, 2014. [Video]. Available: https://www.youtube.com/watch?v=EYbhNSUnIdU

[6] FinFET, "Raycasting with Pygame in Python! Simple 3D game tutorial Devlog," *YouTube*, Nov. 6, 2021. [Video]. Available: https://www.youtube.com/watch?v=4gqPv7A_YRY

[7] WeirdDevers, "Ray casting fully explained. Pseudo 3D game," *YouTube*, Dec. 27, 2020. [Video]. Available: https://www.youtube.com/watch?v=g8p7nAbDz6Y

[8] Veritasium, "The Fastest Maze-Solving Competition On Earth," *YouTube*, May 24, 2023. [Video]. Available: https://www.youtube.com/watch?v=ZMQbHMgK2rw

[9] mattbatwings, "What School Didn't Tell You About Mazes #SoMEpi," *YouTube*, Jun. 22, 2024. [Video]. Available: https://www.youtube.com/watch?v=uctN47p_KVk

[10] Kai Nakamura, "How to Make an Autonomous Mapping Robot Using SLAM," *YouTube*, May 14, 2024. [Video]. Available: https://www.youtube.com/watch?v=xqjVTE7QvOg

[11] Articulated Robotics, "How do we add LIDAR to a ROS robot?," *YouTube*, Jun. 2, 2022. [Video]. Available: https://www.youtube.com/watch?v=eJZXRncGaGM