m(A*)rs

Logan Pickell

December 2, 2024

Abstract

The mA*rs project leverages distributed computing and advanced pathfinding techniques to simulate autonomous bot navigation on a simulated harsh martian landscape. Using Unity game engine and Aron Granberg A* Pathfinding Project, the system explores the challenges of spawning and coordinating multiple agents in real-time while optimizing resource efficiency. By implementing optimization techniques such as object pooling and isometric grids, the system significantly improved performance, achieving a 116 percent increase in frame rates.

Introduction

As humanity moves closer to Mars exploration, autonomous systems for navigating and interacting with unfamiliar terrains are becoming increasingly critical. One of the key challenges is the ability of agents to navigate unfamiliar and complex terrains efficiently. Not only is it challenging to navigate an unknown terrain, it is also computationally expensive to compute a path through the terrain.

The mA*rs project addresses this challenge by simulating agent navigation using Aron Granberg's A* pathfinding project. The primary goal of this project is to create a system capable of efficiently managing a number of autonomous agents navigating complex environments, with an emphasis on performance optimizations through distributed computing techniques. By leveraging the power of distributed systems, mA*rs aims to overcome the inherent bottlenecks associated with spawning and controlling multiple agents.

Distributed computing not only helps in parallelizing workload distribution but also ensures scalability, enabling the system to handle larger and more complex terrains without compromising performance.

Implementation and Optimization

- Project Structure
  - Game Objects
    - Supply Crate (as seen in Figure 1)
    - Player (as seen in Figure 2)

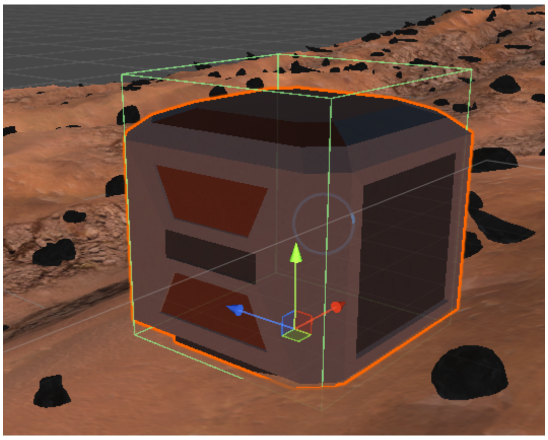

Figure 1: Supply Crate

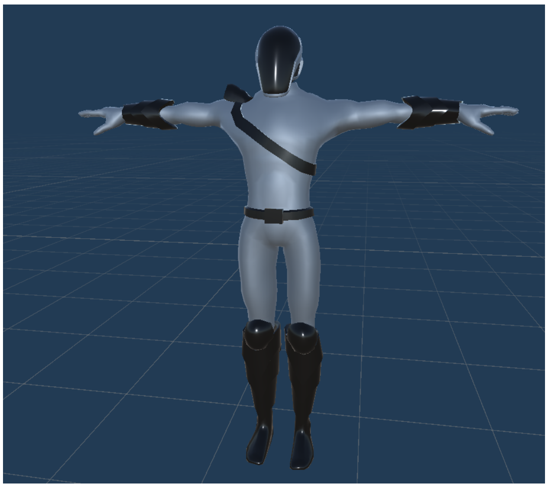

Figure 2: Player Prefab

- ∗ Mars Terrain (as seen in Figure 3)
- ∗ Mars Station
- ∗ Agent (as seen in Figure 4)

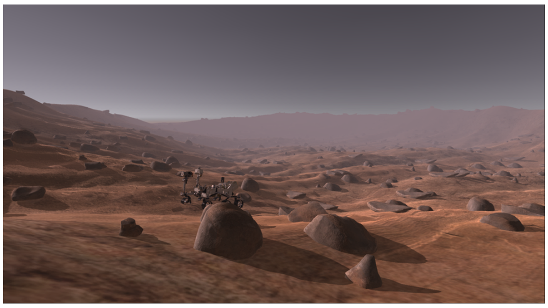

Figure 3: Mars Terrain

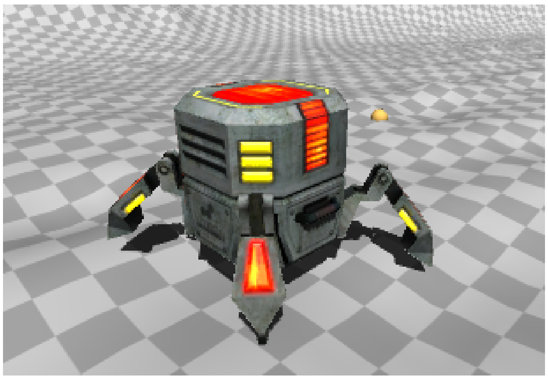

Figure 4: Agent

- Pathfinding Integration with A* Algorithm

  Using the free version of Aron Granberg's A* pathfinding project. A* algorithm is used for real-time navigation updates.Agents dynamically calculate paths based on their spawn location and target position.

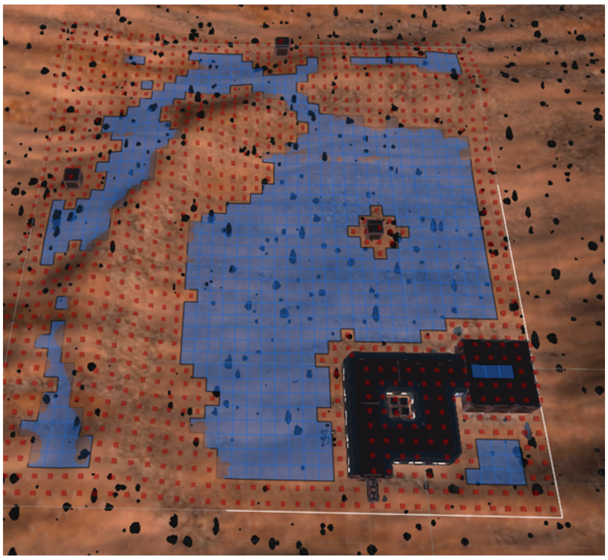

Figure 5: Terrain Scan

- Spawning agents

  Spawning of agents occurs at the computer desk, which is equipped with a trigger collider. When the player enters the collider, a boolean variable, playerInRange, is set to True. If the player is in range and presses the interaction key, an agent is spawned, up to a maximum of six agents. Additionally, the camera is linked to one of the six monitors at the station, allowing the player to track the agents and observe their progress as they navigate to their destination.

- Optimization Strategies

  Agents are pre-instantiated at the start of the program and stored in a pool. When the player presses a designated key, agents are activated and placed at the correct location. By pooling agents, the system eliminates runtime instantiation overhead, resulting in smoother performance even during peak activity.

  The grid system was changed from a regular square grid to an isometric grid. In an isometric grid, movement between adjacent tiles is represented as consistent steps, eliminating the need to calculate or approximate diagonal distances. This uniformity simplifies cost calculations, improves decision-making, and reduces computational complexity.

  The grid was scaled to exclude unreachable areas of the terrain, minimizing unnecessary pathfinding calculations. By focusing only on navigable regions, the system achieves better performance and avoids wasting computational resources on irrelevant nodes.

Results and Performance Evaluation

- Performance Metrics

  I used the Unity statistics during run time to evaluate how well the program was running. I noticed an increase in the frames per second and the CPU main thread, even though there were 2 more animator components playing. This is because we had 5 bots in the unoptimized code, and 6 bots plus a player game object in the optimized version. Even with these additional components, the optimized version performed much better.

- Optimization Impact

  The optimization process significantly improved the performance of the scene, as evidenced by an increase in frame rate from 11.7 frames per second to 25.3 frames per second, reflecting smoother rendering and reduced frame times. The CPU main thread time dropped substantially from 85.5ms to 39.5ms, indicating efficient resource utilization.

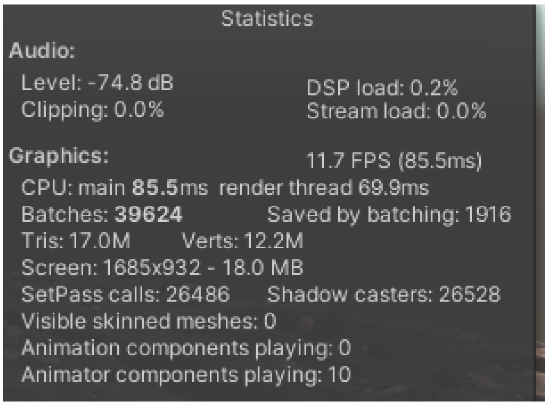

Figure 6: Unoptimized

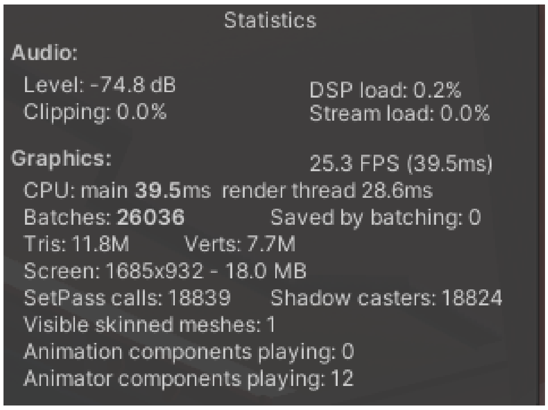

Figure 7: Optimized

Conclusion and Future Work

The mA*rs project successfully demonstrates the potential of distributed computing and advanced pathfinding techniques to enable efficient navigation for autonomous agents in a simulated Martian terrain. By leveraging optimizations such as object pooling, isometric grids, and terrain-focused navigation, the system achieved significant performance gains, increasing frame rates from 11.7 FPS to 25.3 FPS and reducing CPU main thread time from 85.5 ms to 39.5 ms. These improvements highlight the effectiveness of resource management strategies in handling the computational challenges of real-time multi-agent systems.

Future iterations of mArs will explore Unity's Job System and Burst Compiler to harness multi-threading for offloading computationally heavy tasks, such as the A pathfinding algorithm, from the main thread. Additionally, a distributed system simulation for a collaborative 'escape room' scenario will be developed, enabling agents to work together and adapt dynamically in solving navigation puzzles under resource constraints.

References

- NASA. (n.d.). Mars Perseverance Rover, 3D model - NASA science. NASA. Retrieved from `https://science.nasa.gov/resource/mars-perseverance-rover-3d-model/`
- OpenAI. (2024). ChatGPT [Large language model]. Retrieved from `https://openai.com`
- Siccity. (n.d.). GLTFUtility [GitHub repository]. Retrieved from `https://github.com/Siccity/GLTFUtility`
- Granberg, A. (n.d.). A* Pathfinding Project. Retrieved from `https://arongranberg.com/astar/`
- Unity Asset Store. (n.d.). Sci-fi styled modular pack [3D asset]. Retrieved from `https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913`
- Unity Asset Store. (n.d.). EasyStart third person controller [Game toolkit]. Retrieved from `https://assetstore.unity.com/packages/tools/game-toolkits/easystart-third-person-controller-278977`
- Unity Asset Store. (n.d.). Mars landscape 3D [Environment asset]. Retrieved from `https://assetstore.unity.com/packages/3d/environments/landscapes/mars-landscape-3d-175814`