Ramiah Curry

Computer Science Major
First Year ARCS Scholar
Graves Award

MOREHOUSE COLLEGE

Unity and Isaac Sim: Bridging the gap between simulation and reality

Background

→ JetBot is an open-source robot based around NVIDIA's Jetson Nano

→ **Why use Unity?**
- → A quickly modifiable 3D game engine equipped with a relatively easy to learn machine learning package used to train agents

→ **Why use Omniverse?**
- → NVIDIA's own virtual environment software
- → Highly realistic physics based simulations focused on robotics

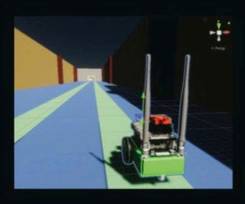
Research (with pictures)

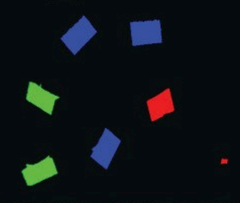
JetBot, imported into Unity

Corridor, modeled in Unity

Raw data, photo taken with in game camera

Labeled signs, parsed from the data





Red = Left Turn, Blue = Right Turn, Green = UTurn

→ **Unity's ML-Agents Package**
- → Allows us to train our JetBots in simulated environments that mimic the real corridor

→ **Synthetic Data**
- → Utilized by scripting variations in background, lighting, and object placements
- → The scene included "distractions" in the background with target road signs scattered throughout

What's been done?

→ **Unity**
- → Trimmed scanned 3D sign models using Blender and imported them into Unity
- → Generated synthetic data for object detection using the Perception Package
- → Trained model to detect road signs using training script from jetson-inference

→ **Omniverse**
- → Scanned and imported 3D sign models into Omniverse using 3D MagiScan
- → Created Replicator environment script and custom writer script for saving synthetic image data to the computer
- → Trained model to detect road signs using training script from jetson-inference
- → Developed a reinforcement learning script for JetBot to identify and follow a red cube

Conclusions (Unity)

Unity offers an assortment of tools and packages for creating simulations and applications

→ **Pros**
- → Able to make realistic, synthetic data to train a real-world object detection model within a virtual environment
- → Reinforcement learning works very well in Unity, and is very scalable

→ **Cons**
- → Needs works with porting reinforcement learning to the JetBot due to lacking a native connection

Conclusions (Isaac Sim)

→ **Isaac Sim is a toolkit with a lot of potential, however, has a fair amount of drawbacks**
- → Pros
  - → Variety of tools available for robotics development, simulation, and deployment
  - → Machine learning capabilities
  - → Focus on closing the sim2real gap in virtual reality
- → Cons
  - → The cost to use is high
  - → Very new so code gets deprecated really quickly
  - → May run into errors without a documented solution

References and Acknowledgments

[1] A. Juliani et al., "Unity: A general platform for intelligent agents," arXiv.org, https://arxiv.org/abs/1809.02627 (accessed Jun. 30, 2023).

[2] U. Technologies, "Digital Twins," Unity, https://unity.com/solutions/digital-twins (accessed Jun. 30, 2023).

[3] "Nvidia Isaac Ros," GitHub, https://github.com/NVIDIA-ISAAC-ROS (accessed Jun. 30, 2023).

[4] "Isaac Sim Introduction," What Is Isaac Sim? - Omniverse Robotics documentation, https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/overview.html (accessed Jun. 30, 2023).

[5] NVIDIA-Omniverse, "Nvidia-Omniverse/omniisaacgymenvs: Reinforcement learning environments for omniverse isaac gym," GitHub, https://github.com/NVIDIA-Omniverse/OmniIsaacGymEnvs/tree/main (accessed Jun. 30, 2023).

[6] Photogrammetry in Blender and Meshroom - Blender Tutorial. YouTube, 2021. https://www.youtube.com/watch?v=L_SdlR57NtU (accessed Jun. 30, 2023).

Thank you to my mentors and additional help from: Dr. Wong, NSF, REU, UTK, RECSEM, Patrick, Steven and Franklin.

Scholar Awards Celebration
November 15, 2023

SCHOLAR AWARDS CELEBRATION · ARCS FOUNDATION ATLANTA · ARCS

Igniting Innovation in Georgia