Artificial Intelligence

Presented at

Indian Institute of Technology, Bhilai

Link for our GitHub Repository:
Project GitHub Link

If the above link does not work, then use below link:
https://github.com/karankumbhar47/Robotic-Simulation

Block-Insertion Using Robo Simulation

Udan Vedant Ghanshyam	12141690
Karan Sunil Kumbhar	12140860
Dandge Sanket Rajendra	12140540
Ahire Sandesh Naval	12140110

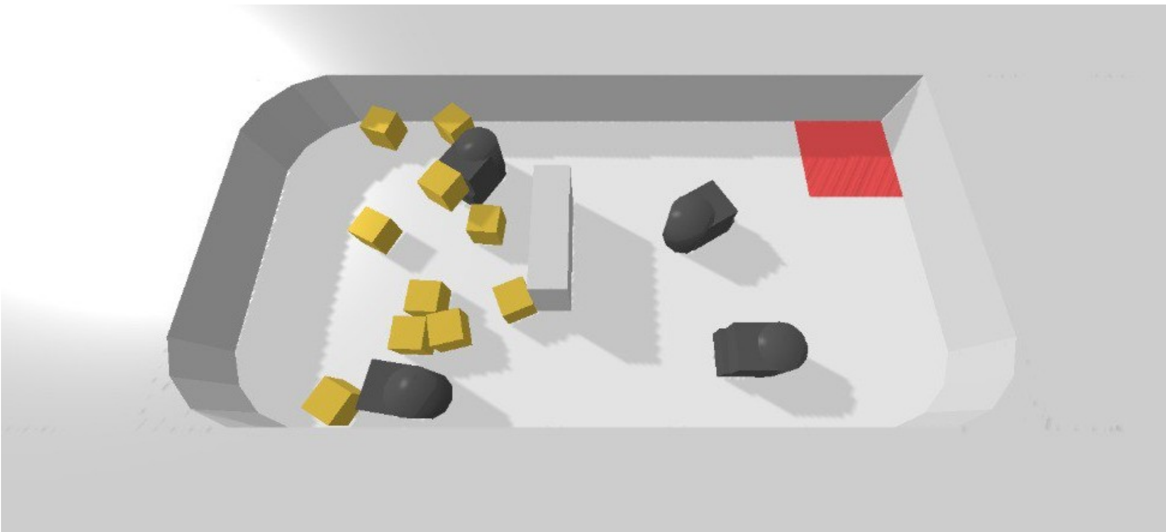
TABLE OF CONTENTS

GENERAL INTRODUCTION

The ability for multiple agents to collaborate effectively in complex and dynamic environments is crucial for the advancement of artificial intelligence. Reinforcement learning (RL) is a type of machine learning that enables agents to learn optimal behavior through trial and error interactions with their environment.

Multi-agent reinforcement learning (MARL) extends reinforcement learning (RL) to scenarios involving multiple agents, offering a powerful framework for training teams of agents to act collaboratively and achieve common goals.

Our project explores the application of MARL to develop a collaborative system of robo cars that can efficiently manipulate blocks and place them in designated areas. We envision a system comprising multiple robo cars working together to pick up blocks, navigate a simulated environment, and precisely position the blocks in their targeted locations.



Chapter

1 PROJECT CHALLENGES

Developing a successful multi-agent reinforcement learning (MARL) system for collaborative block manipulation presents several challenges which we aim to tackle in the project:

1. **Coordinating Multi-Agent Actions**:

   The robo cars must effectively coordinate their actions to avoid collisions, prevent block interference, and ensure efficient block manipulation. This requires the agents to consider the actions of others and plan their own actions accordingly.
2. **Handling Dynamic Environments**:

   The simulated environment may introduce obstacles, changing conditions, and unforeseen events, requiring the robots to adapt their strategies and actions in real-time. The agents must be able to perceive and respond to changes in the environment effectively.
3. **Scalability to Multiple Robots**:

   As the number of robots increases, the complexity of coordinating their actions and handling dynamic environments also grows. The MARL algorithms must be scalable to handle larger teams of robots without compromising performance.
4. **Handling Imperfect Information**:

   The robots may not have complete or perfect information about the environment, such as the exact location of obstacles or the precise shape of the blocks. They must be able to make decisions based on the available information and handle uncertainty effectively.

Chapter 2

PROJECT GOALS

The goals that we want to achieve from this project are as follows:

1. **Training robots to handle dynamic and uncertain environments**:

   Robo-cars are exposed to diverse and unpredictable environment, preparing them to handle the dynamic and uncertain nature of real-world environments. This includes exposure to obstacles, changing conditions, and unforeseen events, allowing the robots to develop robust strategies and adapt their behavior.

2. **Developing MARL algorithms in a controlled environment**:

   We aim to use Deep Q-Learning algorithm to train the robo-cars in order for them to get accustomed to the environment and complete the task after learning the environment.

3. **Developing Environment**:

   We desire to develop a 3D models of cubes and multiple robo-cars in a simulated environment. As making a 3D environment is somewhat a tedious task, we have taken reference from the internet to make our environment.

Chapter

3 PROJECT METHODOLOGY

3.1 Problem Formulation and Objectives:

In this section, we define the mapping from states to actions that dictates an agent's behavior in a given environment. It will then serve as a decision-making process that the agent must follow to select actions based on the current state of the environment. Rewards and Penalties are also defined.

3.2 Spatial Intention Map

Spatial intention maps is a representation wherein each agent communicates its intention to others. This communication is achieved by rendering intentions into overhead 2D maps aligned with visual observations. This representation facilitates a higher level of coordination and cooperation among decentralized mobile manipulators.

The spatial intention maps are integrated with the spatial action maps framework. This framework aligns state and action representations spatially, introducing inductive biases that encourage emergent cooperative behaviors. The spatial alignment enhances the ability of agents to perform tasks requiring spatial coordination, such as passing objects or avoiding collisions.

3.3 DQN Policy

The DQN policy network is crucial for decision-making in the multi-agent system. It employs a DataParallel architecture and utilizes Fully Convolutional Networks (FCN) with the capability to handle varying input channels. The spatial intention maps play a key role in influencing the decisions made by each agent, enhancing their ability to collaborate on tasks.

The DQN Intention Policy extends the functionality of the DQN Policy by incorporating intention networks. These networks, also utilizing FCN, predict intention maps, which are then seamlessly integrated into the decision-making process. The intention maps provide a predictive understanding of each agent's goals, contributing to more informed and collaborative decision-making.

3.4 Training and Learning Paradigms

3.4.1 Training Data

The training process involves exposing the agents to diverse multi-agent environments. The environments consist of heterogeneous robot teams with varying capabilities, including lifting, pushing, or throwing tasks. The agents learn to adapt and collaborate in response to these dynamic scenarios.

3.4.2 Learning Objectives

The learning objectives for both the DQN policy and DQN Intention policy networks include improving task performance and enhancing cooperative behaviors. The training paradigm aims to equip the agents with the ability to effectively communicate and understand the intentions of their peers, leading to more coordinated and successful task execution.

CONCLUSION

In conclusion, we made use of the reinforcement learning algorithm which is Deep Q-Learning and made sure that the robo-cars interact without interfering with each other in order to successfully execute our project.

Also, by using Spatial Intention Map, we make sure where the blocks and other objects in the environment are positioned with respect to robo-cars.