RugbyMAS: Autonomous Agents and Multi-Agent Systems

Implementation of an Autonomous Multi-Agent System in Rugby

David Huber, 105692  
 Computer Science and Engineering  
 Instituto Superior Técnico Lisboa  
 david.huber@ulisboa.tecnico.pt

Francisco Silva, 97433  
 Computer Science and Engineering  
 Instituto Superior Técnico  
 Lisboa  
ist97433@tecnico.ulisbo.pt

ABSTRACT

In this project we aim to develop a multi-agent system for simulating a rugby match. Our approach includes defining rules for player positioning, passing and goal scoring, and developing intelligent agents that can search for gaps in the opponent's defense and play passes to other agents. Our system architecture will enable communication and coordination between agents and we will evaluate our system using metrics such as successful attempts, number of passes and average time of possession.

INTRODUCTION

Rugby is a complex team sport that requires coordination, strategy and individual skills. With the advent of artificial intelligence and multi-agent systems, there is growing interest in using these technologies to simulate sports games and improve team performance. In this project, we focus on Rugby Sevens, a fast-paced and dynamic variation of rugby involving two teams of seven players each. The problem we address is how to design and implement a multiagent system that can simulate a realistic rugby sevens game. We aim to create intelligent agents that can make decisions based on the current state of the game, the position of the opponents, and the location of the ball. Our system will allow for communication and coordination between agents, and we will explore different strategies for cooperation and negotiation.

RELATED WORK

There are several works on sports simulations, but few have focused on Rugby Sevens. Most of them have used either rule-based or machine learning approaches to model player behavior and decision-making. One of the biggest challenges in sports simulations is to create agents that can act autonomously, considering the game situation and the actions of the opponent. This project aims to overcome this challenge by developing agents that can adapt to different scenarios and make decisions based on the current state of the game

PROBLEM DEFINITION AND RELEVANCE

The problem that this project aims to address is the lack of tools to investigate and test different strategies for rugby sevens games. By developing a multi-agent system that can simulate rugby sevens matches, we can provide coaches, players and analysts with a platform to test different strategies and tactics. This is particularly important given the growing popularity of rugby sevens, which has become an Olympic sport in recent years.

Objectives

The main objective of this project is to develop a multi-agent system for simulating rugby sevens matches. This system should be able to:

● Design and implementation of a multi-agent system to simulate a rugby sevens match.

● Evaluate the performance of the multi-agent system in terms of game dynamics, team coordination and strategy effectiveness.

● Analyze the behavior of the agents and the system under different conditions, such as different team compositions, field divisions and opposing strategies.

● Develop and test communication and coordination mechanisms between agents to improve teamwork and cooperation, such as passing, support and tackling.

● Comparing the performance of the multi-agent system with other existing approaches or systems in the literature, such as other multi-agent game simulations or sports analysis tools

APPROACH

The environment is a simulated rugby field with defined rules for player positioning, ball possession, passing, and scoring depending on the strategies adopted. The multiagent system consists of intelligent agents that can perceive the environment through sensors and act on it through actuators. The system architecture provides a framework for communication and coordination between agents, as well as for decision-making and planning. To address the problem of designing intelligent agents, we will implement several features. First, agents will have conflicting goals, such as the need to score a try versus the need to protect the try area. Second, agents will have a variety of sensors, including vision, position, and ball possession. Third, agents will have communication and coordination mechanisms, such as passing and moving towards the ball carrier. The actions performed by the agents are [*DOWN, UP, RIGHT, LEFT, PASS*], note that just the agent that has the ball can perform the [*PASS*]. The *tackle* is an environment rule, that means, when a player from one of the teams has the ball and the opponent performs an action that would put himself in the ball carrier square, the *tackle* is performed, and the ball possession changes to the opponent team.

The goal of the agents it to score a rugby rehearsal as quick as possible, that means, to reach the adversary yellow area with the ball carrier, as may be seen at figure 2.

The strategies depend on if each team is either attacking or defending and are:

* Random Agent:
  + Has no strategy behind the actions of the agent, as the action is chosen randomly.
* Dummy Greedy Agents:
  + When attacking tries to go straight forward, and when it finds an opponent at a Manhattan distance ≤ 2, passes to the nearest colleague either beside or behind him.
  + When defending assumes a random action excluding the *PASS*.
* Greedy Agents:
  + When attacking tries to go straight forward, and when finds an opponent at Manhattan distance ≤ 2, passes to the colleague.

If it needs to pass:

1. Each agent identifies the 2 nearest opponents in order to measure the angle between them.
2. Each agent checks if it hasn’t opponents in Manhattan distance ≤ 2. If they haven’t, they may receive the ball.
3. The agent that has the ball after having the information above, will choose the agent that has 1. The biggest angle between teammate agent and the 2 opponents; 2. Hasn’t opponents in a range of Manhattan distance ≤ 2.
   * When defending the strategy is as simple as having all the teammates going straight forward to the ball carrier.

* Role Agents:
  + When attacking, the team assumes a diamond position as may be seen at figure 1. The roles are BALL\_CARRIER (BC), SUPPORTED\_RIGHT (S\_R), SUPPORTER\_LEFT (S\_L), SUB\_SUPPORTER\_RIGHT (SS\_R), SUB\_SUPPORTER\_LEFT (SS\_L), WINGER\_RIGHT (W\_R) and WINGER\_LEFT (W\_L).

The diamond will be formed using the roles above and assigning a position to each one. To assign the roles we propose the following algorithm:

1. Create an array with tuples in the form (Agent, DM), where the DM corresponds to the distance of Manhattan from the Agent to our teammate ball carrier.
2. Calculate the positions of the diamond, the refence should be the *BALL\_CARRIER*, for instance image it’s position as (x,y). The other roles position should be:
   * + - *S\_L*→ (x-1, y-1)
       - *S\_R*→ (x-1, y+1)
       - *SS\_L*→ (x-2, y-2)
       - *SS\_R*→ (x-2, y+2)
       - *W\_L*→ (x-3, y-3)
       - *W\_R*→ (x-3, y+3)

In the case the coordinates calculated go outside from the [21x11] grid, we do a floor and ceiling functions in order maintain the coordinates possible to be reached.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| y\x | 1 | 2 | 3 | 4 |
| 1 | W\_L |  |  |  |
| 2 |  | SS\_L |  |  |
| 3 |  |  | S\_L |  |
| 4 |  |  |  | BC |
| 5 |  |  | S\_R |  |
| 6 |  | SS\_R |  |  |
| 7 | W\_R |  |  |  |

Figure 1 - Diamond attack strategy

1. Sort the array from 1. And attribute the roles *BC ≽ S\_R ≽ S\_L ≽ SS\_R ≽ SS\_L ≽W\_R ≽ W\_L*.

After assigning all the roles the *BALL\_CARRIER* must go straight forward until it has opponents in a range of Manhattan distance ≤ 2. If any opponent enters in a range of Manhattan distance ≤ 2, the *BALL\_CARRIER* must pass the ball following the Greedy Agent criteria.

* + When defending we have as roles, *FORWARD\_DEFENDER* and *BACKWARD\_DEFENDER* absorbing respectively 4 and 3 agents from the 7 team agent. The algorithm proposed to defend is:

1. There will be 4 *FORWARD\_DEFENDER*, that the main goal will be chase the opponent’s player ball.
2. There will be 3 *BACKWARD\_DEFENDER*, that the main goal is to be 4 Manhattan squares behind the *BALL\_CARRIER.*

We also have created an User Interface to better understand the behavior of the agents, that may be seen at figure 2. When the ball possession is under the blue team, the ball carrier color will be turquoise. When the ball possession is under the red team, the ball carrier color will be orange.

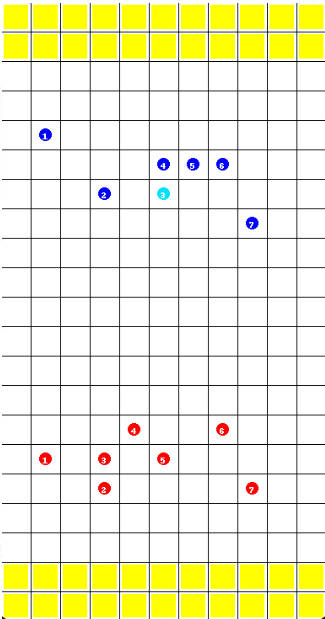


Figure 2 – RugbyMAS: Graphical User Interface

EMPIRICAL EVALUATION

* To evaluate our system, we have defined a set of metrics to check our goals. These metrics are calculated by episode, an episode is the defined either by reaching the goal or reaching the max number of steps allowed. In our testing the maximum number of steps allowed are 1000.

This number is justified due our community of agents is large, and so a few steps like 100 in some scenarios would never reach the goal, in other words, it gives us an unbiased context to all approaches as all strategies have opportunity to reach the goal.

These include the:

1. Average number of steps to have a successful attempt by episode, that says how much effort is put from the team to have rehearsal.
2. The average number of tackles by episode, that tells us how competitive were both teams trying to reach their goal. Furthermore, it also tells us how good a team is either attacking or defending.
3. The standard deviation from the metrics above, as they give us:
   1. Notion if there is convergence or not.
   2. If the results are trustful.
4. the number of tackles between agents, and the effectiveness of various cooperation and negotiation strategies. We used a combination of pure simulation calculus and real-world testing throw the GUI to validate our system and refine our design decisions. Overall, our project aims to contribute to the field of multi-agent systems by demonstrating the potential of these technologies to simulate complex team sports and improve team performance. We believe that our approach can be extended to other sports and games, offering insights into human decision-making and strategy.

REFERENCES

[1] Patricia S. Abril and Robert Plant, 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan, 2007), 36-44. DOI: <https://doi.org/>10.1145/1188913.1188915.

[2] Sten Andler. 1979. Predicate path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226-236. DOI:https://doi.org/10.1145/567752.567774

[3] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:https://doi.org/10.1007/3-540-09237-4.

[4] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY..

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00