# 3v3 Soccer Simulation

Paper # 1 Mirko Michovich and Segun Madarikan

## ABSTRACT

This paper addresses artificial intelligence performance in the development of soccer simulation software. A 3 vs 3 soccer simulation was created using python and the pygame module. An explanation of how this was done is detailed in the paper. This paper is suitable for computer science students who are knowledgeable about the basics of Artificial intelligence and are interested soccer simulation games and software. This paper analyzes the performance of artificial intelligence soccer teams and how performance differs when teams provided with different sets of algorithms are pitched against one another. It reveals that for this soccer simulation at least, the more offensive set of algorithms proved to be more effective at winning games and performed better in some other benchmarks detailed in the paper.

## KEYWORDS

Soccer;artificial intelligence;game software;

## 1 INTRODUCTION

Soccer is the most popular sport in the world. And as a result, soccer simulation games have been one of the most successful video game categories, with the industry generating billions of dollars in revenue every year. For our project, we decided to develop a 2D 3v3 soccer simulation game in python. Considering that both of us are knowledgeable about the sport, we thought that this would be a good area of research. Of course there is already a great deal of research that has gone into soccer simulation. However, even in the most sophisticated gaming engines, there is still room for improvement from a tactical/teamwork perspective for AI in soccer games. With our game, we tried to create two different tactics for our AI teams and compare them, shedding light on the kind of tactics that lead to greater performance.

Our primary goal was to develop a team of agents to be able to play against another team of agents and for them to simulate intelligence in play. Meaning that the agents play with teamwork and the intention to win. By creating decision tree algorithms for each agent, we attempted to ascribe different behaviours to agents based on the roles that they were assigned at the beginning of a match. After creating a working algorithm for each agent/player, we created additional algorithms for each agent to create two different tactics for a team to be able to use; defensive and offensive. With this, we were able to create two different teams that would use different tactics(groups of algorithms) and compare the two tactics we created.The core objective of our project is to analyze the performance of the teams when playing against each other with different tactics. Therefore, certain nuances/rules in soccer (ie offside and throw-in) were dropped for the sake of time.

In this project we will use our research to make an informed decision on which algorithms to test/incorporate into our game to make the agents effectively play alongside each other. By assigning different algorithms to opposing teams, we can see which algorithms simulate teamwork the best and stick to those. We will then observe how this impacts the behaviour of different types of agents.

One of our goals from the beginning was the intention to have an agent that is capable of playing soccer. Also, the agent should be aware of scoring a goal and if possible to develop a strategy to score the most efficient way. Another goal we set was to see the interaction that the AI has when playing with a human. This interaction should be able to change and develop a way to score a goal.

In order to achieve these goals, we used the research detailed in the related works to understand the logic behind such a game. Competitions such as RoboCup (an international scientific initiative with the goal to advance the state of the art of intelligent robots) and video games such EA's FIFA have already put a lot of research into this topic. So we looked through their algorithmic logic to get an idea for how to create our own.

## 2 DISCUSSION

### 2.1 Related Works

In order to achieve the desired goals we had to find related works to have a better background. After reading, we chose some works we thought are useful for the purpose of this paper. Consequently, we started to look at the different stages of the project. Once, the stages of the project were achieved and we had enough background we initiated the development of the game. We thought that having different backgrounds would help us to make this project better since the area of Artificial Intelligence is new we wanted to make sure we had good sources.

Considering the work of Moemeng, we analyzed the possible algorithm in our desired AI. RoboCup was a good start to understand the way agents should behave when playing soccer. This paper addressed certain obstacles that once faced, made the soccer simulation games able to play cohesively and play alongside a human (that will have different/more complicated behaviour). This is insightful information we had before programming our agents. In addition to this, a guideline to the development of basic soccer simulation was also given in the paper.

*Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. Andre, S. Koenig (eds.), July 2018, Stockholm, Sweden*



https://doi.org/doi

Furthermore, we got the ideas for the class diagram on fig. 5 [4] from the decision tree displayed in the paper. This class diagram helped us to get a better sense particularly of the number of players, team, the side the players should face, marker id, velocity of the ball, its behaviour and collision reaction based on the coordinates of the borders of the field. Nevertheless, we wanted to make a simple game, yet capable of making its own decisions. That is why we thought that having a good example of a class would make things better. It helped us to understand the objects we wanted to create in the program. As well the behaviour of each object. We implemented this classes with our methods, where every method had a specific task.

In addition, we wanted to make the User Interface of the program simple as well. As a result, we had the field and the rules set in order to start coding the players and the ball. As Mozgovoy and Umarov showed in their paper in fig.1[5] we were able to have the same layout with the modification of only 3 players. To go back to the idea of simplification, we used squares of 40x40 pixels to make the players, we added color to the players making them red and blue. We also added some assumptions, for example since the players do not have faces, we assumed that the face of the square looking at the opponent's goal was considered the face. This paper explained how Maxim Mozgovoy and Iskander Umarov created a workable AI for 2D soccer simulation games. We used the concept of believability as a good goal to achieve, since it implied the agent would be aware of the environment they were in. Something that we liked is the capacity of the agents for behaving like human players.

Furthermore, we came up with the coordinates of the players based on the position. Since there are only 3, we wanted to make sure that every single player had the similar amount of area that can cover if the ball approached. Considering Buckland's ch. 4 we got a lot of ideas from his work. In Figure 4.1[1] we had a similar arrangement of the field as Mozgovoy and Umarov. We intended to keep 3 players per side. However, what we really found useful was the way Buckland divided the field into 17 equal spaces. Since he had 5 players each player could have almost 4 spaces covered. Consequently, we had only 3 players and we decided to divide our field into 8 squares of the same area. This allowed each player to have two spaces.

Under those conditions, the players were able to have an area that was going to be adapted based on the strategy. Firstly, we coded the players static. This allowed us to adjust UI conditions based on the coordinates of the ball and the player. We called each player from 0 to 5. The idea of the coordinates we also got from Buckland. However, we made it simpler, since our players were only squares. Once, our players had the reaction and the basic physics with the ball. We had trouble making the ball stick to the players and then being released. In order to achieve this, we used a random measurement of timer, where the player would hold the ball for a certain amount of time and then it would be shot. The direction of the shot was also thought to be random. Regarding Verhoeven's [7] work we were able to understand the behaviour of the ball. This made the program less likely to have bias for one team or another. This journal article served us to have information regarding games and Artificial Intelligence. We were using sections of this book to get information for our project. For example section 4.1 Ball Game Analysis. We used the model to see what we could improve in the development of the ball. In addition, we used section 4.4 Artificial Intelligence Bot Development and also we used section 5 Rocket Bot Tournament. What we liked of this section is that we were able to see and analyze how the bots change based on the game they are put to play. Therefore, the goal and objective of an agent could be the same, but because of the rules and the change on environment the behaviour changed. That is what we took into consideration for our game. Since, we had two strategies and those could make the bots change the objective.

Furthermore, after realizing all the details of the players and the ball, we gave movement to the players. Firstly, we decided to get the players bouncing at first. Then we had to implement the shooting and the most important part was the AI. For the purpose of achieving the AI we decided to make three main strategies. Since the goal of the AI is to score the greatest amount of goals. We used the help of Hausknecht to get strategy of Half field offense [3], and we made adaptations on defensive strategy. The offense used the four areas on the top of the field for two players. In addition the defensive strategy uses the four middle areas to have two players there and one at the back. Finally the defense puts all the players in the first 4 areas at the bottom of the field. Finally we also get inspiration on Fu's work to understand the use of Machine Learning for games, this paper showed some algorithms that were useful to understand the function of Deep Reinforcement Learning in multi-agent problems. The authors used other algorithms, we were keeping the idea to make a simple AI agent and that is why we prefered to stick to our idea. However, it was good to have the influence and expositiont to those algorithms like, P-DQN, Deep MAPQN, Deep MAH-HQN. Using these two approaches the authors saw which one better suits the 3v3 games. They used different games and different modes like Ghost Mozgovoy Story and they analyze the win rate using the algorithms mentioned. The authors also showed that the algorithms played different outcomes when they are used 1v2 Defense and 2v1 Offense. In conclusion the way the algorithms perform for 3v3 are based on the strategy and the goals of the agents. Which for us was really important since we were going to coduct similar experiments. [2].

Finally, we made the players to be aware of their teammates coordinates and for that we used Scheepers work [6]. In contrast to the other papers, Machine learning was addressed as much as this paper. However, we took from this paper the sense of a team since his work helped us to determine possible behaviours on our experimentation. We were more aware of some of the possible outcomes the agents could do. What we got from this article was that it's not like Rocket League which is in 3d. This one focused on a 2d soccer game which is what suited us best. We were able to understand the importance of having a good influence of papers before starting a research since in a way they are the influence and the ones that give you ideas to think about the most convenient method to carry out the desired idea.

## 2.2 Methodology

The main problem we were looking into was what tactic or group of algorithms leads to better results in a game. Our 3v3 soccer game was made up of two teams: team red and team blue. These two teams would play against each other, one using the defensive tactics we created and the other offensive.

### 2.2.1 *Creation of the game*

. Our game was created using the Pygame module in the python programming language. This enabled us to create an interface where we can spectate the AI and simulate a soccer match. Firstly, we created a 2 dimensional soccer field. Our players were represented as colored rectangles(color depending on which team they were on), while the ball was represented by a black circle as shown in Figure 1. A classic game of soccer would normally involve rules such as throw-ins if the ball was to go out of the field on the sides, and goal kicks or corner kicks if it went out by the goals. However, for the sake of simplicity, we removed these rules and simply made the ball bounce off the walls. This made programming significantly simpler. Inspired by Mat Buckland's work in his simple 5v5 soccer game [1], we split the field into 8 equal sections and used the location of the ball to determine the decisions made by each individual player. As mentioned earlier, there were two primary tactics created for each team: defensive and offensive. To keep track of the score, when the ball made contact with one of the goals, the other team's score would go up by one and the players would be reset to the default positions as shown in Figure 1. The ball would then be shot to a random location from the middle of the field to kickoff the game again.

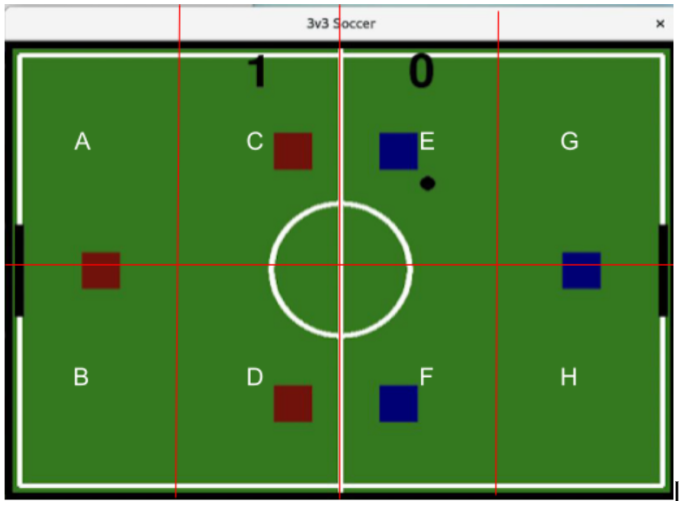


**Figure 1: A snapshot of the field with grid-lines indicating regions of the field.**

### 2.2.2 *Teams Tactics*

. Defensive Tactics:
The defensive tactic was used by our red team in experimentation. For this tactic, the furthest back player (which starts between section A and B as shown in Figure 1) acts as a goalkeeper or last form of defense. This player never leaves regions A and B. When the ball enters this region, the player tries to retrieve it and clear/pass it to the right. In this sense, the player acts as a reflex agent, because it is completely aware of its environment and acts accordingly.
For the next player, in region C, the player is in charge of region C and D. It does not leave these regions and retrieves the ball and passes/shoots it to the right when it has the ball. Similarly to the furthest back player, it is a reflex agent.
The last player(found in section D) acts as the sole attacker. Once the game begins, it moves into sections E,F,G and H. This player tries to retrieve the ball if the ball is in any of those regions. When it has the ball, it either passes it to a random location to the right, or it shoots for the goal. We made it have a 33.33 percent chance of shooting at the goal and 66.67 percent chance of shooting randomly. This was to create the illusions of the player trying to shoot but the shot not being on target on occasion.
Offensive Tactics:
The offensive tactic was used by our blue team in experimentation. For this tactic, the furthest back player (which starts between section G and H as shown in Figure 1) acts as a goalkeeper or last form of defense. This player never leaves regions G and H. When the ball enters this region, the player tries to retrieve it and clear/pass it to the left. In this sense, the player acts as a reflex agent, because it is completely aware of its environment and acts accordingly.
For the next blue player, in region E, the player is in charge of regions C and E. It does not leave these regions and retrieves the ball and passes/shoots it to the left when it has the ball. Similarly to the attacker on team red, it has a 33.33 percent chance of shooting at the goal and 66.67 percent chance of shooting randomly.
For the last blue player, in region F, the player is in charge of regions D and F. It does not leave these regions and retrieves the ball and passes/shoots it to the left when it has the ball. Similarly to the last player described, it has a 33.33 percent chance of shooting at the goal and 66.67 percent chance of shooting randomly.

One notable difference between Defensive and Offensive is that whereas the red team has a player covering every region, the blue team does not have anyone covering regions E and F. Meaning that if a red player has the ball in those regions, no one can get the ball from them. However, the offensive/blue team has two players in the red team's half so they are much more attacking.

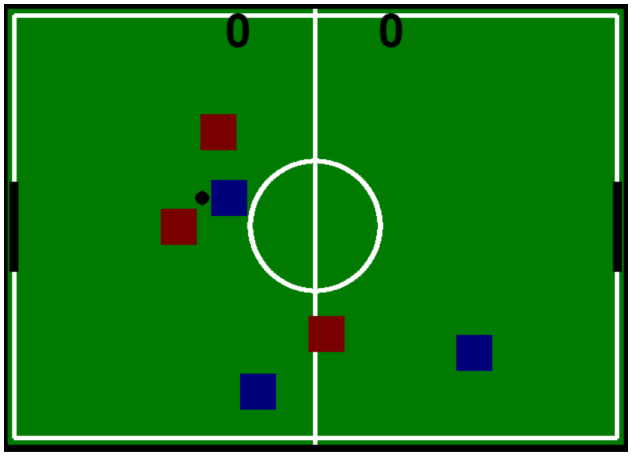


**Figure 2: Snapshot of game in action**

#### 2.2.3 *Code*

. As mentioned before, this program made use of the python programming language and the pygame module. The program made use of three classes: a class for the players/agents, a class for the ball, and a class for the goal posts.

The ball was a set to be a pygame sprite. This is an object type that enabled us to control its velocity and implement basic game physics for the ball. The ball starts off with a random velocity at the start of every game or after every goal. To imitate the state of a player having possession of the ball, if a player were to collide with a ball, the balls velocity is set to be the velocity of the player. Afterwards, when the ball is released by the player, the velocity of the ball changes depending on the situation.

The players were also set to be sprites. However, with the use of loops and if and else statements, we were able to use decision tree algorithms for each player. Figure 3 shows the basic logic behind the decision making of the average player. A player/agent first checks if their team has the ball, if not they go to retrieve it if it is in their allocated region. If their team has the ball, if they have the ball, they have will shoot/pass the ball after having the ball for about 0-5 seconds.

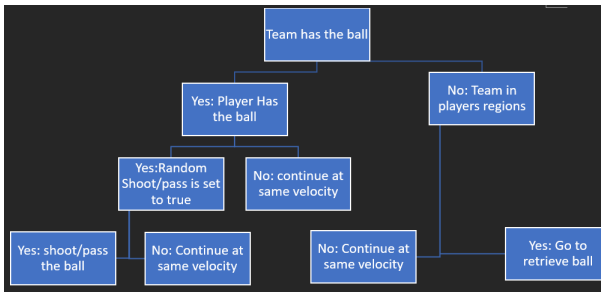


**Figure 3: Decision Tree for Players**

#### 2.2.4 *Performance Benchmarks*

. Performance Benchmarks: To determine the performance of the team, we ran the program 10 times, each being 3 minute games we measured with a timer. Our three benchmarks for performance were number of games won, number of goals scored, and possession.

Games Won: This, as expected, is our most important benchmark. The team that has scored more goals by the end of the game is declared winner of that trial. The number of wins for each team were then tallied and the team with more wins is superior in this benchmark.

Goals Scored: We recorded the number of goals for each team in each game/trial. Afterwards, we averaged our results to get the goals per game for each team. The team with more goals will win this benchmark. It is likely that the winner of the games won will also win this but it gives us insight into the defensive performance of the teams

Possession: Due to issues with measuring time in game, we could not record the time each team had possession of the ball. However, by incrementing the magnitude of an integer variable each time the red team was on the ball and incrementing the magnitude of another integer variable each time the blue team was on the ball, we were able to determine which team had more possession. So we recorded which team had more possession for each team. The team with more games was the winner of this benchmark.

## 2.3 Experiment

*2.3.1 Games Won*
. After 10 trials, the table 1 below details the winner of each trial and the total number of wins for each trial. The number 1 indicates that that team won and 0 represents a loss.

**Table 1: Games Won**

| Game No. | Red | Blue |
|---|---|---|
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 0 |
| 4 | 0 | 1 |
| 5 | 1 | 0 |
| 6 | 0 | 1 |
| 7 | 0 | 1 |
| 8 | 1 | 0 |
| 9 | 1 | 0 |
| 10 | 0 | 1 |
| Total | 3 | 7 |

As shown above, the blue team 70 percent of the games. This implies that the offensive tactic is superior to the defensive tactic at winning games. This is our most benchmark which means that our offensive tactic is highly favored to be the better performer overall. It could also mean that based on the rules of our game, strong offense is more important than strong defense.

*2.3.2 Average goals per game*
. After running our tests for 10 trials. A tabular as well as visual representation of our results can be found below.

**Table 2: Goals scored in Game**

| Game No. | Red | Blue |
|---|---|---|
| 1 | 1 | 7 |
| 2 | 1 | 5 |
| 3 | 0 | 7 |
| 4 | 4 | 3 |
| 5 | 2 | 8 |
| 6 | 1 | 8 |
| 7 | 2 | 9 |
| 8 | 6 | 5 |
| 9 | 4 | 3 |
| 10 | 1 | 8 |
| Average | 2.2 | 6.3 |

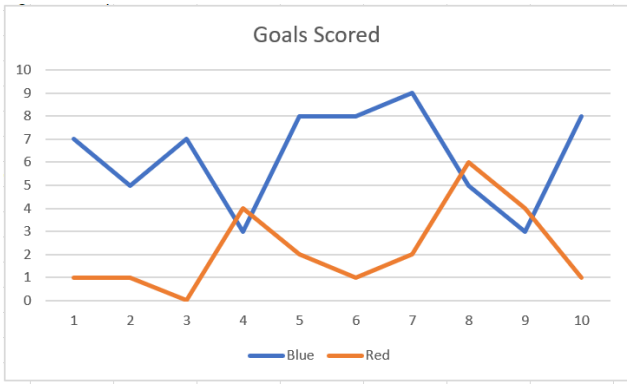


**Figure 4: Visual representation of goals scored**

As shown in Table 2, the average number of goals scored by the blue team were significantly higher than that of the red team. This could be due to the fact that blue team had two agents in the red teams half. It can be concluded that the attacking ability of the offensive tactic is more valuable than the defensive advantage of the defensive tactic.

*2.3.3 Possession*
. In the table 3 below, we can find the team that had more possession for each game. The number 1 means that that team had more possession whereas 0 means that that team had less possession.

**Table 3: Possession winner**

| Game No. | Red | Blue |
|---|---|---|
| 1 | 1 | 0 |
| 2 | 0 | 1 |
| 3 | 0 | 1 |
| 4 | 0 | 1 |
| 5 | 1 | 0 |
| 6 | 0 | 1 |
| 7 | 0 | 1 |
| 8 | 1 | 0 |
| 9 | 1 | 0 |
| 10 | 0 | 1 |
| Total | 4 | 6 |

The possession stats show that the blue team had more possession than the red team. However, it was a close call and leads us to believe we cannot conclude that the blue team is better at retaining possession. The fact that it was this close leads us to believe that possession may not have a correlation to number of goals scored.

## 3 CONCLUSIONS

After experimenting with the two different strategies in the game; offensive and defensive. We have come to the conclusion that the offensive strategy is the most suitable and the most accurate to meet the agent's goal. We have been able to observe that the behavior of the agents changed with respect to the strategy and that even if there are two different strategies, the less defensive one will always won. For example, if team 1 (red) had a defensive strategy and team 2 (blue) had an offensive strategy. As you could imagine, the offensive strategy won. This showed us that the agent for whatever objective always tried to optimize the outcome for that objective.

Finally, we were able to realize the complexity involved in making a decision tree, since the one we made was not that big but it covered the necessary rules. We were able to realize that taking into account all the rules would take much more time and also there is always the possibility that not all the rules are covered. That is why we proposed to make the system from goal based to learning based. It could use your data and input to figure out what the goal is and based on the goal given by the user, the system could learn the most efficient way to play the game.

On the other hand, we were able to observe that correlation between ball possession and offensive strategy is positive. This implies that the ball stays longer in the offensive team since not only can they retain more, but it also implies that they make more satisfactory passes to their teammates. The correlation was proportionally positive as well for possession of the ball and offensive strategy with games won. Consequently, the correlation between lost games and defensive strategy is not how we thought. Since the defensive strategy went to support and prevented some goals from being scored, the algorithm was able to score more goals than the defensive team could retain. It should be noted that the defensive strategy did not have as much advantage as the defensive, since we did not train the goalkeepers in any other way or we did not implement more advantage so that they could easily catch the ball. We just used the position of the players and moved them back where they were already determined to go for the ball and still trying to score goals.

#### 3.0.1 *Improvements*

. A possible improvement that we have considered is to have a defensive strategy that can change with respect to the position of the ball. On the other hand, it would be interesting to implement an offensive strategy that would also be dynamic and could change with respect to the position of the players and the ball.

Furthermore, we could observe the agents always tried to score goals, the problem is when two players wanted to take the ball away from each other and passed the ball between them. Although we had the algorithm that decides to pass arbitrarily and with a certain angle. It would be an improvement to implement a strategy that would take into account the area where the player is and where his teammates are. This algorithm can recognize and is aware of the position of the players and the ball at all times. That is why the players move towards the ball. We have yet to implement a strategy to recognize which player is the friend or foe and to make a more accurate decision on where to pass the ball.

In addition, we have analyzed possible improvements for this simulator. Knowing what was in the experiment and in the rules of soccer compared to real life. One improvement would be to have more rules for the agents, this would imply having a much larger decision tree that could take more cases and agent's states. The fact that it was oversimplified helped us to get an idea of what the outcome and behavior of the agents was. At the same time, we have analyzed that the passing between players is not the most suitable, an algorithm that takes into account many more features of the game would help the passing to be much more realistic and thus have a more fluid game play. For the moment, all passes are made randomly.

#### 3.0.2 *Further Applications*

. After the improvements considered above, we saw that the implemented technology could be used to test real game strategies, and the adaptability of the game could serve for other sports. For example the game could be used in other sports like Basketball, American Football, etc. In order to be able to adapt to those sports, there would have to be a set of global variables that are able to change to set of universal rules hat most of the sports have. Once that is accomplished the values for the algorithm of strategies, decision tree and the rules of the game as long it is rectangular would change. Furthermore, it could be possible to have the dimensions of the field modified in order to reach a major number of sports. The possibilities are endless, since with a good adaptation and improvements of the algorithm, the system would be able to show the best strategies and plays within the game.

These would still be very difficult to transfer to real life since there are more things to take into consideration that would make the decision tree even bigger. However, it could serve to give an idea of a strategy. There would be the possibility of transferring the strategy game to tasks that are not only sports, but could be used to solve basic problems that require a common goal. If sufficiently adapted, we believe that the game/simulator could be able to solve positioning problems. If the game is adapted and becomes more dynamic, the possibility of using it with other forms of players is much greater. That would imply having a larger number of players and therefore changing all the rules of the game.

At the same time, another application that could be used is with the enabling of data inputs based on an opponent's strategy and the agent could see the best alternatives with respect to that strategy. Once it could be used in games and sports, the applicability goes beyond that, as the system is designed to reach a goal. It could be implemented in flash games and applications can go further with more adaptations. But since our goal was to maintain a soccer game most implementations with a rule change would be sports.

## REFERENCES


[1] Mat Buckland. 2009. *Programming game AI by example.* Wordware Publ.

[2] Haotian Fu, Hongyao Tang, Jianye Hao, Zihan Lei, Yingfeng Chen, and Changjie Fan. 2019. Deep Multi-Agent Reinforcement Learning with Discrete-Continuous Hybrid Action Spaces. (2019). arXiv:cs.LG/1903.04959

[3] Matthew Hausknecht, Prannoy Mupparaju, Sandeep Subramanian, Shivaram Kalyanakrishnan, and Peter Stone. 2016. Half Field Offense: An Environment for Multiagent Learning and Ad Hoc Teamwork. In *AAMAS Adaptive Learning Agents (ALA) Workshop*. Singapore. http://www.cs.utexas.edu/users/ai-lab?hausknecht:aamasws16

[4] Peerapol Moemeng. 2004. Issues in Soccer Simulation Software Development. 7 (05 2004), 175–179.

[5] Maxim Mozgovoy and Iskander Umarov. 2011. Believable Team Behavior: Towards Behavior Capture AI for the Game of Soccer. (01 2011).

[6] Christiaan Scheepers and Andries P. Engelbrecht. 2014. Training multi-agent teams from zero knowledge with the competitive coevolutionary team-based particle swarm optimiser. *Soft Computing* 20, 2 (2014), 607–620. https://doi.org/10.1007/s00500-014-1525-0

[7] Yannick Verhoeven and Mike Preuss. 2020. On the Potential of Rocket League for Driving Team AI Development. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2335–2342. https://doi.org/10.1109/SSCI47803.2020.9308248