

# ThunderLeague: Team Description Paper 2021

Caio de Souza Barbosa Costa, Giovanni Cabral Morales, Antonio Lago Araújo Seixas, Gabriel Cosme, Lucas Pavan, Maria Fernanda Fernandes Rezende, and Ricardo Tamay Honda

University of São Paulo,  
São Paulo, São Paulo, Brazil

{caio.costa, giovanni.morales, antonio.seixas, gabriel.cosme,  
lucas.pavan, maria.rezende, ricardo.honda}@thunderatz.org

**Abstract.** The ThundeRatz robotics team has created a RoboCup 2D Soccer Simulation team in 2018 and has been improving since then. This paper brings what we developed until now that made us achieve the position of one of the main 2D Soccer teams in Brazil. Most of the advances described here were inspired in the Gliders2D source code, that are: the pressing and blocking behavior, the evaluator algorithm for defensive and attack purposes and the implementation of better player’s positioning behaviour. Besides that, we also modified and created new formations using the Fedit2 editor. All those changes combined allowed the team to achieve better defensive results in comparison with Gliders2D v1.5.

## 1 Introduction

ThundeRatz is the robotics team of University of São Paulo. It was founded in 2001, and since then, has been participating in many robotics competitions worldwide. Recently, the team is increasingly engaging in autonomous robotics and has developed a Very Small Size Soccer team and a RoboCup 2D Soccer Simulation team. The latter is called ThunderLeague.

ThunderLeague was created in 2018, using the source code of Agent2D [1, 2] and, without any modifications, has participated in its first competition: Winter Challenge 2018, the biggest robotics competition in Latin America. In the next year, with big changes in formations, the team was able to win the first prize on that same competition and, since then, has become one of the main Soccer 2D teams in Brazil.

In the last year, ThunderLeague has been following the steps of Gliders2D v1 [4] line of development and has been working on specific changes that would boost its performance in the short-term. Those modifications have allowed the team to win another national competition in Brazil this year, the Iron Cup 2021.

In this paper, we describe the changes that have been implemented since ThunderLeague was identical to Agent2D [1]. As previously stated, all those changes were guided by Gliders2D work in 2018 (from version 1.1 to 1.5) [4], so that many of our original changes are small adjustments to increase the team’s performance even more.

## 2 Changes to Agent2D

### 2.1 Formations

In 2019, after using the Agent2D [1] team without changes for a year and taking part in two competitions, we realized the importance the formations have in the team's performance and started working on replacing Agent2D default ones.

We focused on modifying and creating new formations, using the Fedit2 editor [3], and tested them against other Brazilian teams that took part in past competitions. Multiple versions of some formations were made and the chosen were the ones with better goal difference. Then, the better ranked formations were tested against each other, in a process of elimination, until there were only one remaining.

This change alone improved our performance significantly and was the reason we won our first competition.

### 2.2 Pressing and Blocking

One of the implementations was the addition of the *pressing* variable. By basing itself on the Gliders2D [4], the *pressing* variable modifies the basic move behavior by changing the necessary conditions for the team to try to intercept the ball. Those changes are made by adding a variable that considers the player's role and the field area, adding a constant to balance the comparison between the player's minimal ball gettable cycle and the fastest opponent minimal ball gettable cycle. To be able to intercept the ball, this comparison must return that it's easier for the player to reach the ball than the fastest opponent added with the pressing variable.

The Blocking feature implements several conditions to execute the Marlik [6] team's known blocking action. The blocking action is divided into three functions:

1. The function used to get the blocking target point considers the player's role, the player's position, and the ball position. It uses the ball's and the player's position to define the estimated opponent dribble speed and, by using the roles, ball positions, and the estimated opponent's position, it defines a blocking target point.
2. The execution of the block move assigns a blocking point and checks some of the opponent's characteristics, like future position, current position, velocity, and creates a target point with the function defined above. With this target point, it calculates the opponent's target point and, with the use of intercept cycle's number, roles, and ball distances, evaluates what blocking move should be done.
3. The ball interception action is much simpler and checks the ball reach cycles of the player, teammates and opponents, and then check if the ball interception is possible.

With both of these additions, the team was able to improve the intercepting actions and execute correctly the blocking.

### 2.3 Evaluator

By basing itself on the Gliders2D [4] implementation of an *evaluator*, the implemented algorithm allows the use of the new *evaluator* for defensive and attack purposes. That became possible by dividing the field into sectors, which helps the analysis of the field situation. First of all, a variable that counts the number of directed opponents, based on the player with the ball possession vision, is implemented. This part of the algorithm is similar to implemented in Gliders2D in 2018. In order to make it viable for defensive purposes, another variable that contains a sector in the ally midfield area and is capable of identifying opponents that reside there was created.

With these two variables, the *evaluator* is divided into three conditions:

1. When there are no opponents forward and in the ally midfield area, the best point is the opponent's goal;
2. When there are more opponents in the ally midfield area than in front of the player with ball possession, the algorithm assumes a defensive mode, which finds the best point through Voronoi diagrams. It considers the forward line and the defensive line and counts the number of opponents between these two;
3. If none of the above conditions is satisfied, the *evaluator* suggests an attack move, finding the best point through Voronoi diagrams. It considers the forward line and the offside line and counts the number of opponents between these two.

By using this algorithm, the new team was capable of increase its goals and victories rate in comparison with the original team.

### 2.4 Positions

Another field evolved from Gliders2D [4] implementation was the players' positioning. Once the team's defense performance was solid against other teams in the national competition, we've focused on trying to improve the strikers' positioning in order to have them in promising places so that they could create more opportunities.

The first modification was made using dynamic Delaunay Triangulation based formations [5], modifying the forwards' locations on defense formation so that they no long need to return near our goal area, thus been able to stay in a good place to start a counterattack once defenders regain ball possession. Other change in defense situations was to use a small ball attraction parameter in some game modes (e.g. opponent goal kick) to make the strikers maintain their positions and don't lose stamina aimlessly.

Having less defensive duty, strikers could accumulate more stamina and use it to reach their offensive positions faster. Those places were selected according to Voronoi Diagram, which was based on Gliders2D [4]. The basic concept of it is to select intersection points between offside line and Voronoi Diagram edges

and assign some of them to the 3 forward players. Some changes were made in the depth value from which this method is applied, in order to get better results. Maximum dash power could be used by strikers more often (because of the stamina increase due to a more passive positioning in defensive situations), and therefore some positional-based constraints were created to decide when to make the strikers run faster to those points.

## 2.5 Results

After all these changes, ThunderLeague has achieved similar results to Gliders2D v1.5 [4] with better defensive performance. The table 1 shows that, after 1000 games against Agent2D [1], our team has 22% less conceded goals and it has allowed us to win more games overall.

Team	Points for	Points against	Goals scored	Goals conceded	Goal diff.
Gliders2D v1.5	2.785	0.154	3.995	0.785	3.210
ThunderLeague	2.889	0.093	3.869	0.612	3.257

**Table 1.** Performance comparison after 1000 games

## 3 Conclusions and future works

The focus of this paper is to describe the implementations done until the beginning of 2021 and it's noticeable how much the team has improved during this time comparing its results in games against Agent2D [1]. Many of our current work, which encompass the implementation of an enemy-repulsive behaviour based on Gliders2015 work [7], the creation of small formation adjustments to allow blocking opponents (similar to LeftEagle work in 2016 [8]) and the improvements of the passing behaviours (which includes the risky passes created on Gliders2d v1.6 [4]), is not described here.

Another interesting implementation we are currently working on is the creation of a team strategy evaluator, which will allow the team to analyze the game and change its formation and parameters, allowing the players to be more or less aggressive during the match.

## References

1. Akiyama H., Nakashima T. (2014) HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation. In: Behnke S., Veloso M., Visser A., Xiong R. (eds) RoboCup 2013: Robot World Cup XVII. RoboCup 2013. Lecture Notes in Computer Science, vol 8371. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-662-44468-9\\_46](https://doi.org/10.1007/978-3-662-44468-9_46)

2. agent2d-3.1.1, online, available at: <https://pt.osdn.net/projects/rctools/downloads/55186/agent2d-3.1.1.tar.gz/>. Consulted on April 2021.
3. fedit2-0.0.1, online, available at: <https://osdn.net/projects/rctools/downloads/68531/fedit2-0.0.1.tar.gz/>. Consulted on April 2021.
4. Prokopenko, M., Wang P.: Gliders2d: Source Code Base for RoboCup 2D Soccer Simulation League (2018).
5. Akiyama, H., Shimora, H.: HELIOS2010 Team Description 2010.
6. Tavafi, A., Nozari, N., Vatani, R., Yousefi, M.R., Rahmatinia, S., Pirdir, P.: MarliK 2012 soccer 2D simulation team description paper. In: RoboCup 2012 Symposium and Competitions, Mexico City, Mexico (2012).
7. Propopenko, M., Wang, P.: Gliders2015: Opponent avoidance with bio-inspired flocking behaviour.
8. Abdelhafiez, A., Ehab, A., Kamel, A., Breaka, H., Abdelrahman, H., Soliman, N., Ali, O.: LeftEagle Soccer 2D simulation Description Paper 2016.