# Final Report

J. D. Escallon Guzman, J. D. Lozano Luna, J. E. Munoz Gomez

Department of Computer Engineering

Universidad Distrital Francisco Jose de Caldas

Emails: {jdescallong, judlozanol, jemunozg}@udistrital.edu.co

## Introduction

In this report, we are going to summarize all the processes related to designing and implementing a solution for the chosen Kaggle competition by applying principles of system analysis and design.

## Competition Context

The competition organized by Google Research and Manchester City F.C. has as its main objective the development of artificial intelligence agents capable of playing football in a simulated environment. This dynamic allows for the experimentation and evaluation of game strategies through daily simulations, where each agent controls a single player and makes decisions in real time based on observations of the match state. The scoring system of the competition is based on a μ value and an initial uncertainty σ assigned based on result of simulated football matches played between the submitted agents of the competitors, where the criterion for having a better score on the leaderboard is just if the agent has a win, a loss, or a draw, so it doesn't consider goal difference. To verify that the agent is eligible to participate, it will play a first simulation against itself to prove that it does not trigger any errors. The environment where the agents play their matches is an open-sourced football simulation environment provided by Google Research, so it will be described and used in future parts of this report. The rules for this competition matches are like the rules of real football except that we don't have to consider switching sides in mid time because the environment fixes it automatically, and we will not have any extra time or penalties at the end.

Among the most relevant restrictions, we find that matches have a fixed duration of 3,000 steps with no additional time, where each step is given by each time the agent receives the match information (Observations) and give an action for the active player (the one with the ball in chase of having the ball, or the closest to the ball otherwise). Additionally, each agent controls only one player at a time, and external variables such as weather or referee errors are not considered. Competitors can submit up to five versions of agents per day. The ranking is organized through matchups between agents with similar skill levels, and the results of each match directly affect the μ values. The leaderboard displays the best performance of each team.

## A systemic approach of the competition

The competition environment can be interpreted as a complex system that includes well-defined inputs, processes, and outputs. We will have a first general approach, analyzing all the competition in the following diagram:

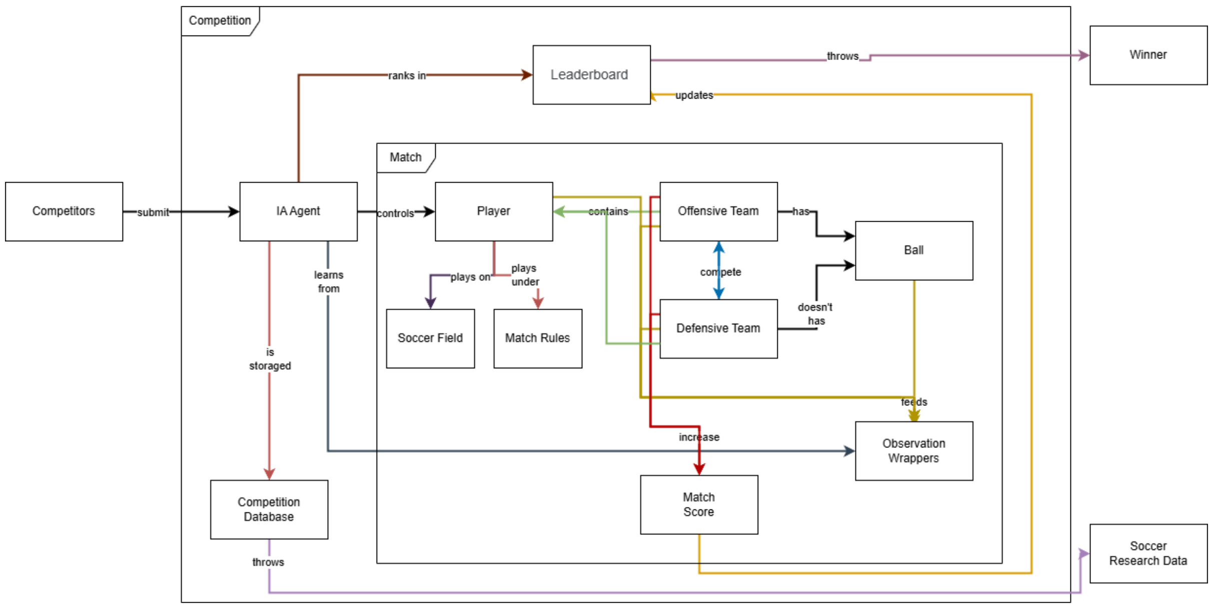

Figure 1. Full System Representation

To make the system easier to understand, we will split it into two sub systems: The classification system and the match system.

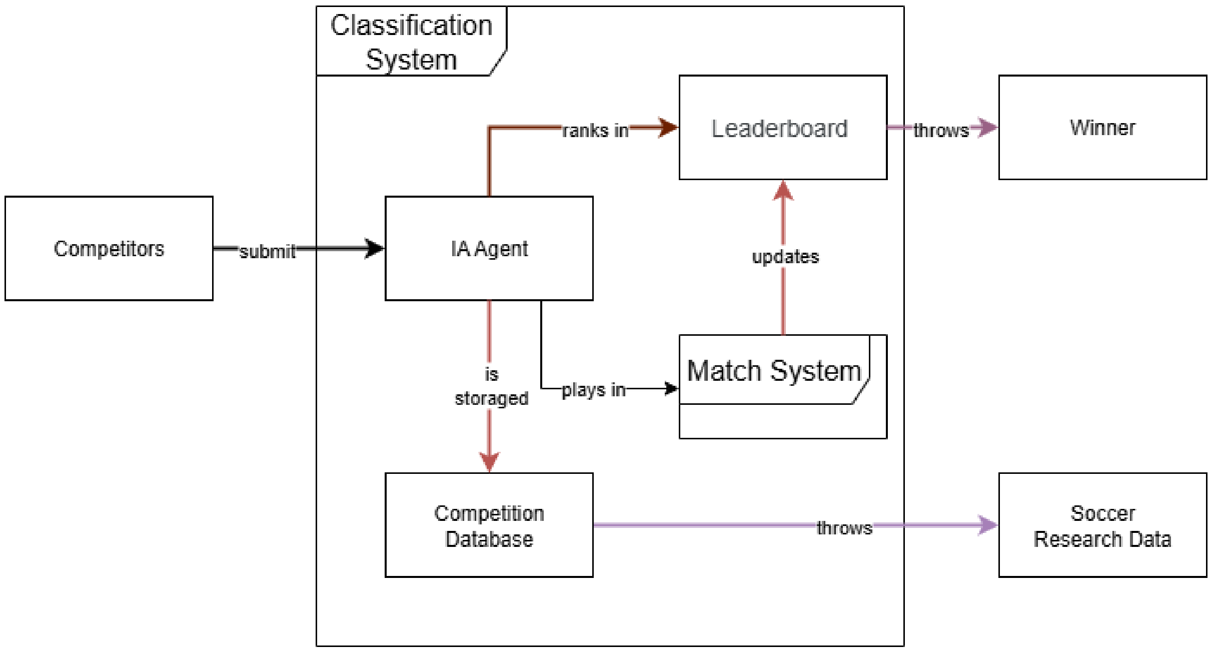

Figure 2. Classification System Representation

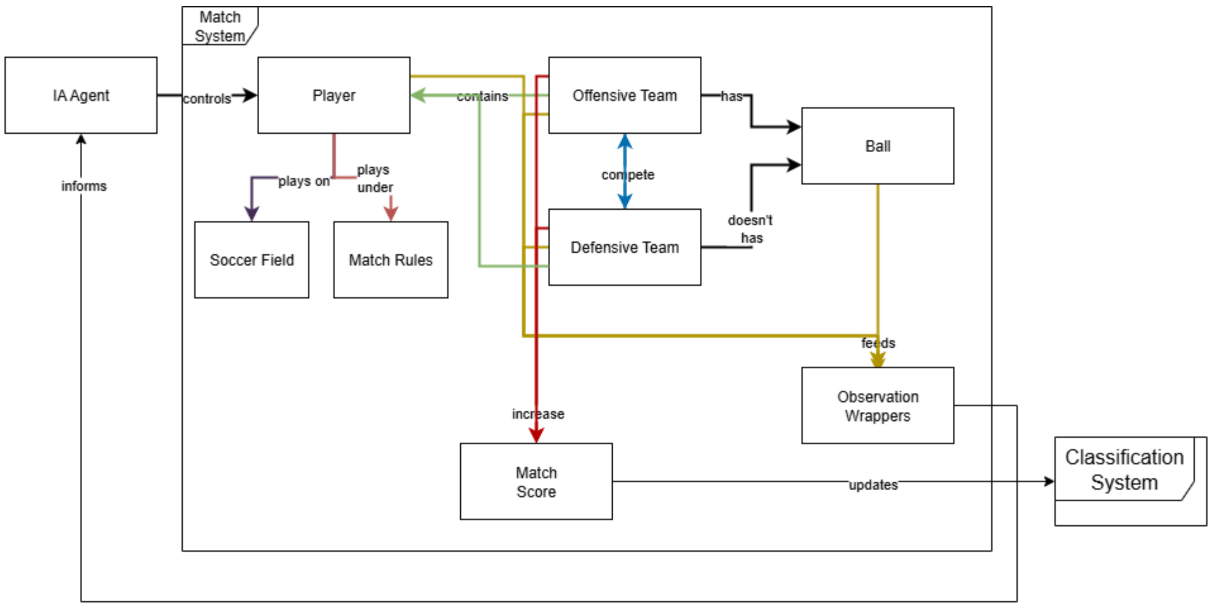

Figure 3. Match System Representation

Multiple chaotic behaviors can be observed. For example, random actions such as rebounds, defensive errors, or counterattacks can alter the course of the game. Likewise, small differences in decisions can lead to completely different outcomes, increasing the variability of the system. The chaotic nature of football also makes it difficult for agents to correctly interpret the effectiveness of their strategies. Additionally, the correct and clear definition of rewards must be considered, as during training, it can cause agents to develop emergent behaviors. While these behaviors may prove effective in certain contexts, they can also lead to undesired decisions by the system.

The critical information flows are given by the observations generated at each step of the match. These contain data about positions, roles, game states, and ball possession. This information is essential for agents, as it enables effective decision-making during the game.

**¿What does an optimal solution should do?**

The analysis of the environment allowed us to identify a series of key requirements for developing an optimal solution. The agent must be capable of correctly interpreting the data provided by the simulation environment, regardless of the format in which it is represented (raw, pixels, or simple115_v2). Based on this information, the agent must make precise and adaptive decisions, always complying with the rules of the game.

To achieve this, a properly structured reward system must be implemented. This system should encourage desirable behaviors and penalize those that are detrimental to the team. It must consider individual actions and their impact within specific match phases (attack, defense, transition, etc.), thereby promoting decision-making that is more coherent with the flow of the game.

The implementation of a backup mechanism was considered. These backups will serve as restoration points in case of potential failures, while also acting as a preventive measure against the chaos caused by erroneous learning or the repetition of undesired behaviors.

Thanks to this, it would be possible to revert to previous versions of the model that demonstrated stability and performance, preventing system degradation due to failed training sessions. In this regard, the agent must constantly strive to make the best possible decision at every moment of the match, guided by the accumulated experience from its reinforcement learning and supported by a control system that ensures its consistency and evolution.

**Designing a modular solution**

The solution was a modular system to respond to the previously described challenges. The input adapter allows translating and standardizing the different observation formats, thus facilitating the agent's compatibility with its simulation environment. The action chooser is the core of the system, responsible for taking the player's actions through reinforcement learning.

The reward system represents a fundamental part of the agent's learning. In it, the rules and objectives of the game are structured by assigning values to different actions and contexts. An example of this is when an agent loses the ball, they will receive a penalty, while effective passes and goal-scoring finishes will be rewarded. This system directly confronts the chaotic nature of the environment, providing a clear guide so the agent can better interpret the actions it should take.

To improve decision-making, we divided the game into phases that differ based on the location and possession of the ball. These phases are:

Attack: the team will have possession of the ball in the opponent's field.

Defense: the opponent will have the ball near our goal.

Build-up: the team will have possession of the ball in our own field.

Transition: these will be intermediate situations.

High press: the opponent will have the ball in their own field.

This classification helps the reward system adapt dynamically to the game, enabling more intelligent behavior.

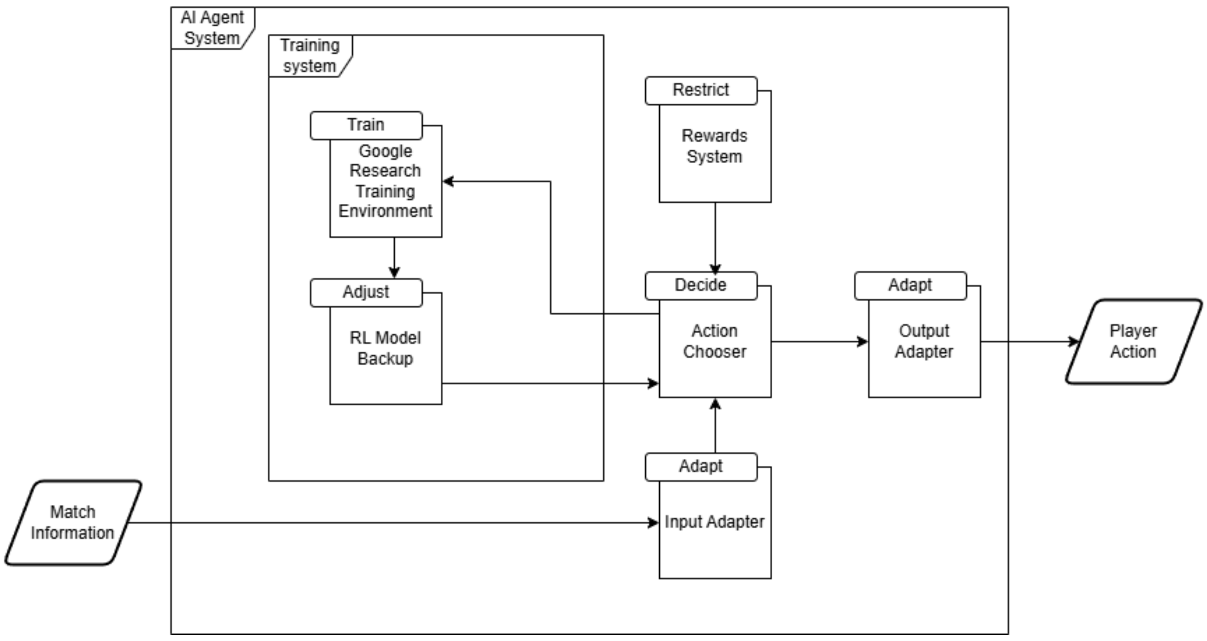

Figure 4. Modular Designed Solution

## Designing the reward system

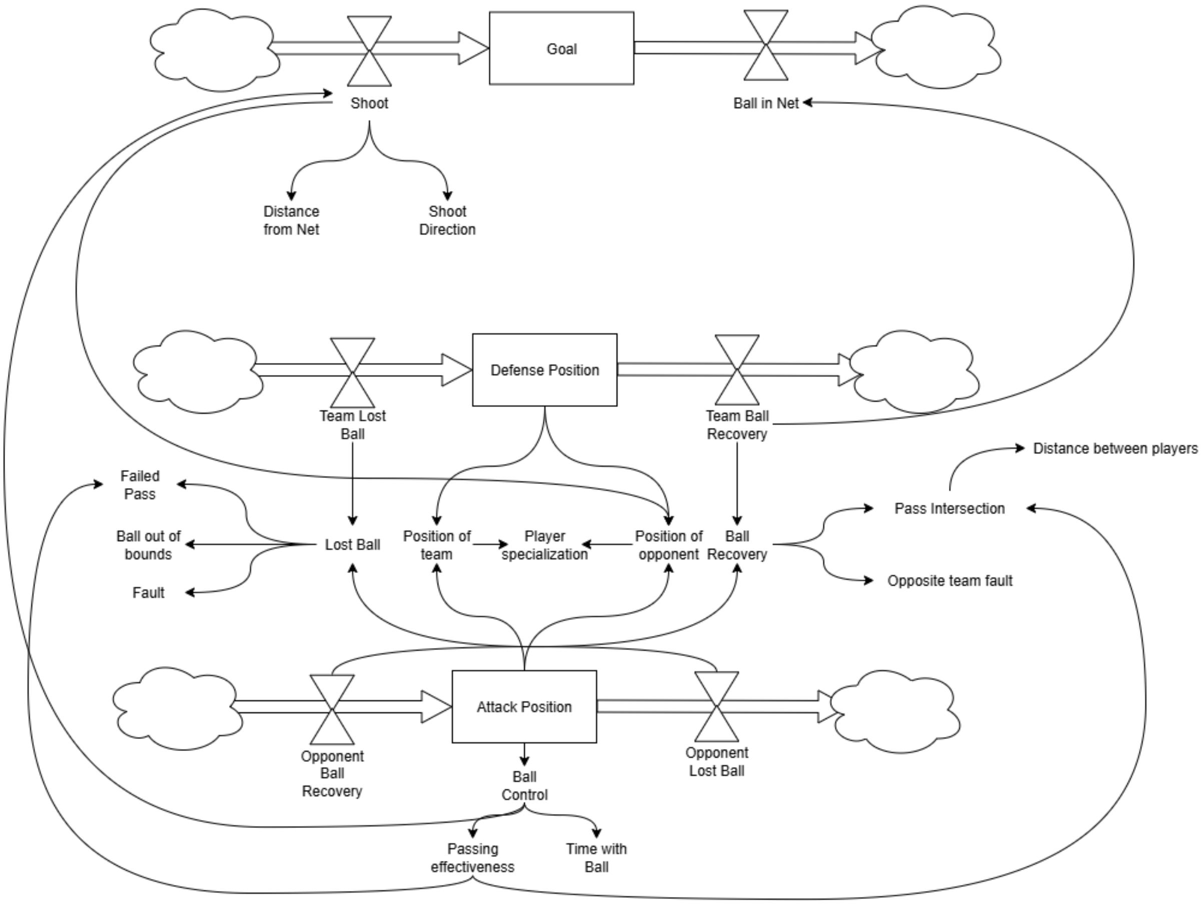

Figure 5. Stock and Flow for Rewards Design

When performing an abstraction analysis, we concluded that the most abstract point of a football match is the goal action, whether it is in favor or against the player. Based on this, we created an analysis and diagram that explores the different entities and actions related to achieving this goal.

A goal begins with a shot and ends with the ball in the net, but many actions and variables occur before and during this event. This led us to the conclusion that it is essential to reward both scoring a goal and shooting, even if the shot doesn't result in a goal. Likewise, we apply a penalty if a goal is scored against us.

For a shot to be effective, it depends on factors like the shot direction and the distance from the player to the goal. Therefore:

- If the shot is on target, the agent will receive a reward.
- If the shot goes off target, it is penalized.
- The closer the shot, the greater the reward.
- If the shot is from a very long distance, a penalty is applied.

Before being able to shoot, many factors influence the situation. These are divided into two main contexts: defensive position and attacking position.

Defensive Position:

This starts when the ball is lost and ends when it is recovered. Being in this position is negative for our agent since it increases the risk of conceding a goal.

Ball loss was associated with three factors:

- Missed passes
- Sending the ball out of bounds
- Committing fouls

These three actions are penalized.

On the other hand, ball recovery is associated with two factors:

- Intercepting opponent passes
- Opponent errors

To improve performance here, we implemented a reward for pressuring the opponent when nearby — the closer the agent is to the opponent, the higher the reward, which increases the chances of ball recovery (which is also rewarded).

Attacking Position:

This is treated as positive, as it means we have the chance to score. It focuses on ball control to eventually create a shot.

Ball control depends on two main factors:

- Time in possession
- Successful passes

Rewarding time in possession can cause the agent to hold onto the ball too long, making it easier for the opponent to take it away. Therefore:

- There is a penalty for excessive time in possession.
- There are rewards for different types of successful passes.
- Losing ball control is penalized.

These two contexts are also related to the player’s position (role), such as defender, midfielder, or attacker, since an attacker won't defend the same way a defender would. For this reason, we decided to personalize the rewards depending on the player's position.

**Some experimenting and emergent behaviors.**

When simulating a sport with such a wide range of possibilities, the initial ideas we had for the reward system were not always efficient, often leading the agent to develop emergent behaviors. During our experimentation process, we encountered many of these situations.

For example, at first we defined specific zones where a player should be depending on their position. For instance, a center-back was expected to stay within the second or fourth quarter of the field. However, this led the agent to only care about staying in that area, rather than focusing on the ball or the overall game context.

Another case was related to passing. The agent realized that making long passes granted a good reward, and that backward passes were not penalized. As a result, it started exploiting this by continuously passing the ball back and forth between the striker and the goalkeeper to gain rewards.

We also observed a lack of adaptability to different contexts due to how the rewards were structured. The environment allows us to expose the agent to various scenarios, so we trained it in a situation where it was alone in front of the goal with no goalkeeper, to help it learn how to score goals. Initially, the agent did score successfully in that simplified environment. However, when we transitioned to the full 11 vs. 11 matches, it began shooting from any position as soon as it got the ball — clearly a behavior that wouldn't work in real matches. Because of this, we realized it was necessary to introduce a penalty for shooting from long distances.

## Outcomes and Conclutions

We exposed the agent to various reward structures, each with different improvements, failures, and results. This was possible thanks to the flexibility in designing the reward system, as the model translates program information into a format understandable by a general audience, making the reward structure easier to analyze and refine.

We managed to obtain agents that showed satisfactory behaviors in specific situations depending on the type of agent — some became good at passing, others at defending or attacking. However, we faced a common issue: the agents were often too rigid in their behavior. For example, an agent that was effective at applying high pressure became disorganized when defending against a low block.

Additionally, agents exhibited moments of brilliance, making very smart decisions that showed clear progress. Yet, surprisingly, when faced with the same scenario later, they would sometimes behave erratically, choosing completely different and ineffective actions — even if a previous strategy had worked well.

Overall, we believe the agent did not meet our expectations, as it did not behave like a well-rounded player capable of adapting to any situation. Although we observed significant behavioral changes in response to continuous reward modifications, we did not see a meaningful difference in behavior based on the player's position or specialty, which was one of our main strategies. Unfortunately, this did not yield the expected results.

Finally, we believe that the model offers high scalability for improving the agent. With a better understanding and more experience in tuning behavior through rewards, we could significantly enhance its performance. Additionally, we consider that a highly efficient way to boost the agent's capabilities would be to split the model into two neural networks: one dedicated to attack and another to defense. This would allow for more specialized and context-aware decision-making, and given the flexibility of our system, such a change could be implemented smoothly.

## Bibliography :

[1] TensorFlow, *TF-Agents: A reliable, scalable, and easy-to-use TensorFlow library for Reinforcement Learning*. [Online]. Available: https://www.tensorflow.org/agents?hl=es-419

[2] J. D. Escallón Guzmán, J. D. Lozano Luna, and J. E. Muñoz Gómez, *Workshop No. 1 — Kaggle Systems Engineering Analysis*, Workshop report, 2025. [Online]. Available: https://github.com/judlozanol/Systems-Analysis/tree/main/workshop1

[3] Google, *Google Research Football with Kaggle*, 2020. [Online]. Available: https://www.kaggle.com/competitions/google-football/overview

[4] Rules of Sport, *Football Rules*. [Online]. Available: https://www.rulesofsport.com/sports/football.html

[5] Kaggle, *Football Environment JSON Config*. [Online]. Available: https://github.com/Kaggle/kaggle-environments/blob/master/kaggle_environments/envs/football/football.json

[6] Kaggle, *Football Environment Python File*. [Online]. Available: https://github.com/Kaggle/kaggle-environments/blob/master/kaggle_environments/envs/football/football.py

[7] Google Research, *Google Research Football GitHub Repository*. [Online]. Available: https://github.com/google-research/football/

[8] Google Research, *Google Football Observation Documentation*. [Online]. Available: https://github.com/google-research/football/blob/master/gfootball/doc/observation.md#raw-observations

[9] J. D. Escallón Guzmán, J. D. Lozano Luna, and J. E. Muñoz Gómez, *Workshop No. 2 — Kaggle Systems Engineering Analysis*, Workshop report, 2025. [Online]. Available: https://github.com/judlozanol/Systems-Analysis/tree/main/workshop2

[10] J. D. Escallón Guzmán, J. D. Lozano Luna, and J. E. Muñoz Gómez, *Workshop No. 3 — Kaggle Systems Engineering Analysis*, Workshop report, 2025. [Online]. Available: https://github.com/judlozanol/Systems-Analysis/tree/main/workshop _3_Simulation