SoccerSim

Gabriel Yuseff

May 31, 2024

1 Introduction

I want to practice my coding skills. So I will create a Soccer simulator, based one of my first homeworks in Advanced Programming; which was a console Quiddich match simulator.
For the project I want to try and use several of the things I learned afterwards in the course.
I'll do it in C# as I used that language in the course, however I could also try doing it in Python or C++.
PS: This will also help me remember how to write TEX files

2 Project Definitions

2.1 What I want to do

Here are some ideas that I want to try:

- Simulate a Soccer game with most of its rules
- Each player behaves independently following certain logic:
  - What is the role of the player (Goalkeeper, Defender, Midfield, Attacker)
  - The team is attacking or defending
  - What is the position of the player
  - What are the positions of the other player
- Make it chance-based for decisions and outcomes
  - A player could try to make a difficult shot to the net, instead of an easy (and more intelligent) pass
  - The player may miss a pass to a teammate that is alone
  - A Goalkeeper may make score a goal by mere chance
- Add stats to the players, which will increase their success chances for the plays
- Show all in a simple graphical way
- Maybe add user intervention by creating players of its own
- Maybe add a tournament mode

2.2 What I want to remember

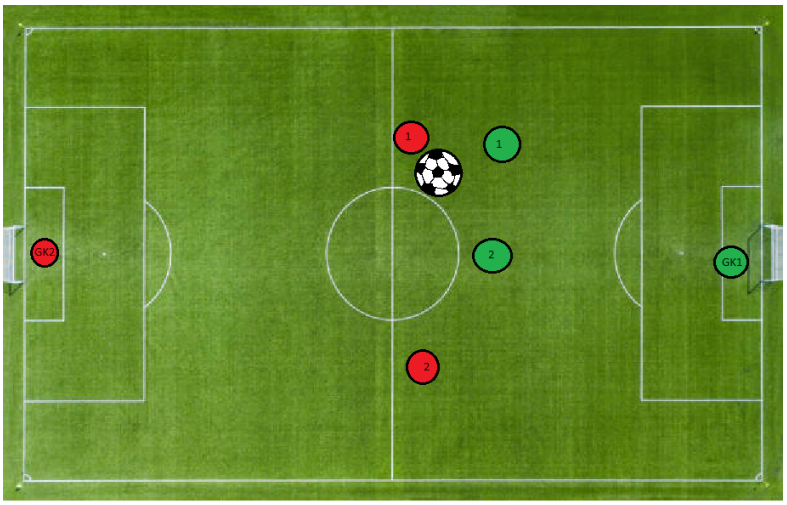
For This project I want to use these concepts (at least):

- OOP
- Graphical Interfases → .NET: Windows Presentation Foundation (WPF)
- MultiThreading
- Polymorphism

2.3 What I want to show

Use a very basic way to show the game:

- Have a fixed background
- The ball and players represented with simple sprites (circles)
- Have the score and time showing in a corner



3 Class definitions

3.1 Players

Abstract class: Will be used as a parent class for all the different players. It will contain the base attributes and methods, that will be overwritten by each specific player, class.

3.1.1 Attributes

- ID: Unique ID for each player
- Name
- Role: What is the role of this player
- Pass: Stat that determines how good they are at passes
- Shot: Stat that determines how good they are at shooting at the net
- Tackle: Stat that determines how good they are at stealing the ball from an opponent
- Dribble: Stat that determines how good they are at keeping the ball when an opponent is trying to take it from theorem

- Save: Stat that determines how good they are at catching a ball thrown to the net
- Speed: Stat that determines how fast can they move
- Position: Current position of the player in the field

3.1.2 Methods

- Constructor: Will take all the attributes to create the player
- Update: Main method of the Player class. It determines how the player will act in each iteration. It will be dependant on the role, the ball position, the other players positions, who has the ball, etc
- Move: Method to translate the player in the field. It depends on the current position and the player velocity
-