

チーム戦術を考慮したサッカー競技のオブジェクト指向シミュレーション

Object-Oriented Simulation of Soccer-Play in Consideration of Team Tactics

We design a model of soccer-play based on the object-oriented modeling. In particular, team tactics is considered in this model, by which we can perform real simulation of the match with 11 on 11 players. Moreover, some results of analysis obtained by the simulation are presented.

1. はじめに

サッカー競技では各選手は自分で判断して行動している。この様子を忠実に表現したシミュレーションを行うために、オブジェクト指向モデリングを用いることにした。オブジェクト指向とは常に物(オブジェクト)を中心に考え、物が何かをするというパターンで動くプログラムを作るという考え方である。

本研究では、チームとしての戦術を実行できるサッカー選手のモデル化を行う。まず、すべての選手に共通する動作や判断をモデル化したサッカー選手のクラスを設計する。このクラスを継承することで、それぞれのポジションに固有の動作や判断などを細かくモデル化したポジションクラス(FW, MF, DF, GK)を作成することが可能となる。このようにして、最終的には戦術をもったチームによる、11人対11人のサッカー競技をシミュレーションで実現する。その際に、数パターンの代表的な戦術を考慮したサッカー競技のシミュレーションを試みることで、チームを構成する選手の能力や戦術に対する分析を行う。なお、実装にはJava言語を用いる。

2. モデル化とクラスの設計

2.1 クラスの設計

オブジェクト指向モデリングにおけるクラスとは、同じ性質を持ったオブジェクトの集まりを意味する。サッカー選手の場合、チームとしての戦術を考慮する必要がない場合には、基本的に全選手が同じ行動をとると見なせるので、1つのクラスを設計すれば十分である^[1]。

しかしながら、戦術面を考慮した場合には、ポジションによってチーム内での役割が異なってくる。そこで今回は、オブジェクト指向モデリングの特徴の1つである継承機能を用いることで、上述のポジションの違いをモデル化することを考える。継承とはあるクラスの性質を引き継ぐ新しいクラスを生成すること、あるいはそのような関係があることをいう^[2]。具体的には、選手クラスを継承してFW, MF, DFおよびGKのポジションに応じたクラスを設計することができる。

これ以外にもフィールドクラス(GROUND)、ボールクラス(BALL)、初期設定クラス(FORMATIONSET)、能力表示クラス(MYCANVAS)を定義した。今回考えたサッカー競技のクラス間の関係をFig. 1にまとめる。なお、選手クラス(SOCCERMAN)は抽象クラスであり、自らはデータや振る舞いを持たないことに注意する。

このように定義したクラスから生成される複数のオ

ブジェクトは、オブジェクト間の協調動作を一連のメッセージ送信で実現することが可能^[3]であり、今回定義したオブジェクトが相互に情報をやり取りすることで、サッカー競技をシミュレーションすることができると。このようなオブジェクト間での情報の流れをFig. 2にまとめる。

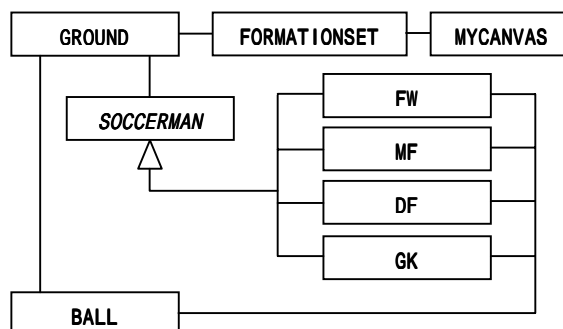


Fig. 1 サッカー競技のクラス図

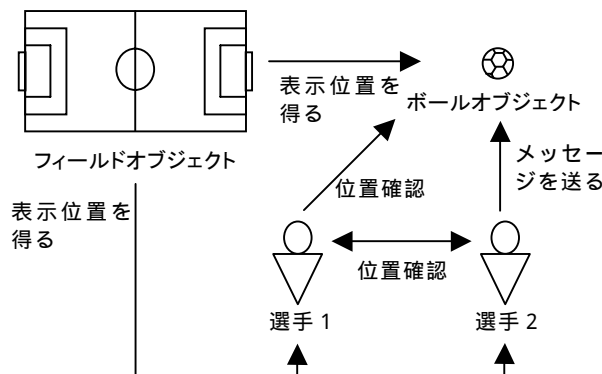


Fig. 2 各オブジェクト間での情報の流れ

2.2 フィールドのモデル化

フィールドオブジェクトはサッカー競技を行うための場所であり、役割として各選手とボールの現在位置、ボール所持率、得点など現在の情報を統括し画面に表示する。その他にも各選手やボールなど必要なオブジェクトを全て生成し、得点判定、オフサイド判定などの審判としての役割も果たす。

2.3 ボールのモデル化

ボールを持っている選手がボールに対してメッセージを送ることによって、競技時にボールが動くと考えた。選手がボールに送るメッセージは、選手が行う行動によって異なる。どの種の行動の場合も必要となる、目的位置、移動距離、移動時間、背番号の基本4種類と、行動別に必要な幾つかのメッセージを送る。また、ボールが選手に対して情報を提供できる多くのメソッドを用意し、選手間での連携を助ける役割も果たす。

2.4 選手のモデル化

各選手は自分の位置，他の選手の位置およびボールの位置から次の行動を判断する．選手は個々の状況に応じた判断をして行動するのだが，チームの戦術を実行する上で味方との連携を持つような行動をとらなくてはならない．それらを実現するために設計した各ポジションオブジェクトによる固有の動作と連携を必要とする動作を Table 1 に示す．

Table 1 ポジション別による動作

動作	ポジション
ゾーンディフェンス	FW, MF
ラインディフェンス	DF
センタリング	FW, MF, DF
スルーパス	FW, MF, DF
フォーメーションを維持する	FW, MF, DF
ディフェンスとの距離を取る	GK
シュートコースに入る	GK

3. マルチスレッドの適用

実際のサッカー選手はそれぞれが非同期的に行動しており，この点が上で述べた選手のオブジェクト指向モデルによって表現される．ところが1台のコンピュータでは同時に複数の処理は行えないため，サッカー競技の非同期性を実現することは難しい．そこで，Java言語の機能のひとつであるマルチスレッドの適用を考えた．マルチスレッドとはそれぞれの処理をごく短時間で切り替えながら処理を進めることで，人間の目にはあたかも複数の処理を同時に行っているように見せかける方法である^[4]．今回，各選手オブジェクト，ボールおよびフィールドに対してそれぞれをスレッドとして定義した．これによってサッカー競技の非同期性を擬似的ながら実現することができた．

4. シミュレーション結果

戦術の違いによるボールの動き方の変化をシミュレーションで求める．

まず，両チームとも同じ条件(選手の能力と戦術)とした場合の結果を Table 2 と Fig. 3 に示す(以下，この条件を case1 と呼ぶ)．Table 2 には各チームのボール所持率，オフサイド回数，得点をまとめた．また，フィールドにおける，ボールの存在時間の割合を Fig. 3 に示す．図中の等値線はボールの存在時間の長い場所を基準に10分割した値を示している．これらの結果から，両チームのボール所持率などがほぼ互角であり，等値線の分布も左右対称になっていることがわかる．

一方，左チームは case1 と同じ条件とし，右チームはゾーンおよびラインディフェンスを用いていない場合(case2)の結果を Table 3 と Fig. 4 に示す．これらの結果から，左チームのボール所持率が高くなり，等値線の分布も case1 に比べて右に寄っていることがわかる．この事実は，左チームでは相手が攻め込む前にボールを奪い，その結果，攻撃の時間が多くなっていることを示している．このことは，左チームがディフェンスに戦術を用いた結果であると考えられる．ま

た，ラインディフェンスを使用することで，オフサイドトラップも多く成功していることもわかる．

Table 2 両チームの結果(case1)

	左チーム	右チーム
ボール所持率(%)	51	49
オフサイド回数(回)	6	8
得点(点)	1	0

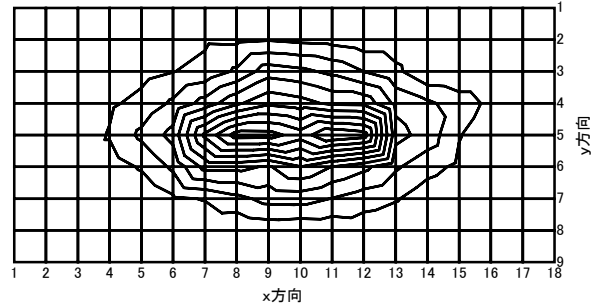


Fig. 3 ボールの存在時間の割合(case1)

Table 3 両チームの結果(case2)

	左チーム	右チーム
ボール所持率(%)	55	45
オフサイド回数(回)	7	21
得点(点)	2	2

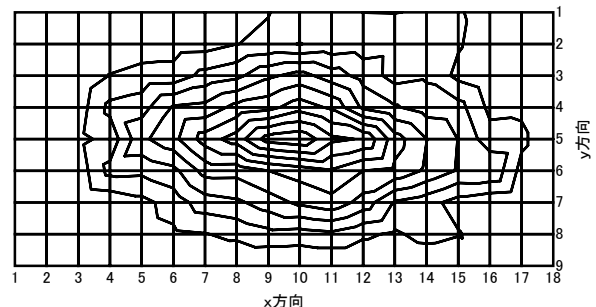


Fig. 4 ボールの存在時間の割合(case2)

5. まとめ

ポジション別の選手オブジェクトをモデル化することで，戦術を実行できる選手による11人对11人のサッカー競技を実現できた．また，戦術が機能することを検証するため，いくつかのパターンによるシミュレーションを行い，結果からモデルの有効性を確かめた．これにより，チームを構成する選手の能力や戦術に対する分析を行うことが可能となる．

参考文献

- [1] 阿部光男，オブジェクト指向モデルによるサッカー競技のシミュレーション，平成12年度茨城大学卒業論文(2000)
- [2] 磯田定宏，オブジェクト指向モデリング，コロナ社，pp.35-47(1998)
- [3] M.C.Chan, S.W.Griffith, A.F.Iasi, Javaプログラミング 1001Tips, オーム社, pp.624-650(1997)
- [4] 夏目 大，図解解説よくわかる Java, 翔泳社, pp.25-27, pp.30-45(2000)