

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

1. はじめに

近代サッカーではフィールドにある“スペース”が狭いため、選手はフィールド全体を広く使って攻める必要がある。そのためには選手が個々に役割を持ち、チームとしての戦術が機能するように動かなくてはならない。また、選手の戦術理解度が高ければ個々の能力だけで動くよりもチームとして機能する。監督の役割としては、選手の能力にあった戦術を用いることや、逆に戦術にあった選手を起用することである。このことが勝敗を大きく左右する[1]。

サッカー競技において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。言葉を変えればオブジェクトとはブラックボックスであるといえる。すなわち、サッカー選手は他の選手が判断する過程を知る必要はなく、その選手がとった具体的な行動だけがわかれば良い。また、オブジェクト指向モデリングの利点としてプログラムを大まかなところから順に作っていきけるということもあげられる。既存のプログラムに対して性能を良くするために、メソッドやクラスを追加していくことが簡単にできるからである[2]。

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。このクラスを基に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を作成していく。そして、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ことを目指す。敵や味方の人数が増えることによって、選手の動作や判断が増えてくが、上述のオブジェクト指向の利点をいかして、選手の人数を段階的に増やしながら機能を追加していくことができる。現段階では味方選手7人によるシミュレーションが可能であり、今後、7人対7人 10人対10人 11人対11人を計画している。

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面を考慮しないならば、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[3]。この観点に立ったサッカー選手のモデル化については次節にまとめる。

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[4]。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。

現段階ではサッカー選手の基本となる選手クラス(SOCCERMAN)の設計を行っている。これ以外にもフィールドクラス(GROUND)とボールクラス(BALL)を定義する。今後の予定を含めた、サッカー競技におけるクラス間の関係を Fig. 1 に、オブジェクト間での情報の流れを Fig. 2 にまとめる。

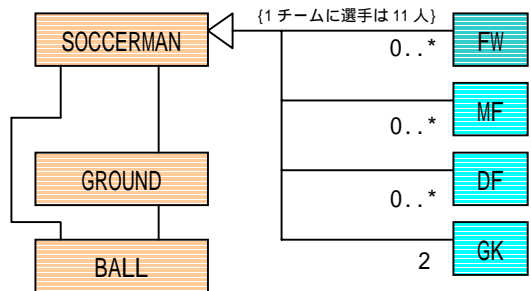


Fig. 1 サッカー競技のクラス図

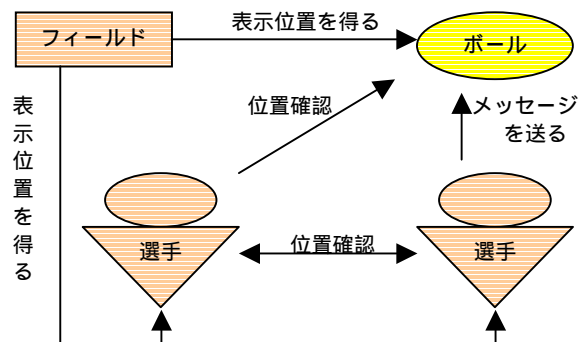


Fig. 2 各オブジェクト間での情報の流れ

2.2 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、競技中における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目的位置、移動速度、移動時間、目的選手の4種類とした。特に、ボールにパスの目的選手という情報を持たせることにより、選手は移動中のボールを見ることで、誰に対するパスかを判断することができる。かつ、周りの選手はパスを受ける選手をサポートするような行動が可能となる。

3. サッカー選手のモデル化

3.1 選手モデル

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。選手は個々に状況判断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。また、各選手の能力として、走力、ドリブルの速さ、ドリブル率、ボールキープ値、各選手に対するパス率、他の選手との距離感を与えた。個々の選手は、これらの情報を総合して、それぞれの行動を決定する。

味方選手7人(攻撃時)における選手の行動を Fig. 3 に示す.各選手はボールの所持・非所持に応じていずれか1つの行動をとる.なお,ボールを追うという行動の最後に,選手は必ずボールの動きを止めるトラップという行動をとる.したがって,現段階ではトラップをせずにダイレクトなプレーを行うことはできない.

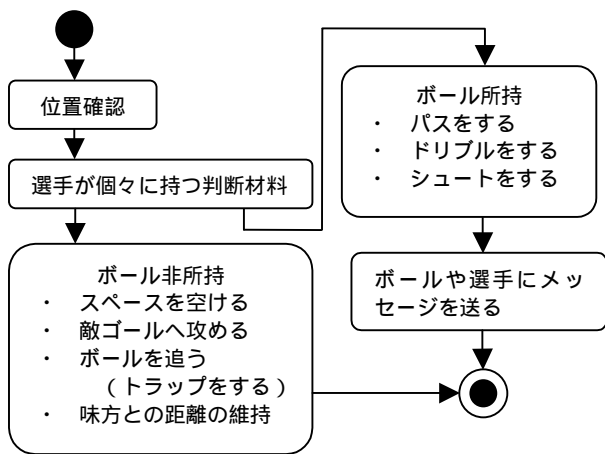


Fig. 3 選手の行動を示す状態図

3.2 選手の動き

すでに述べたように,近代サッカーでは選手がフィールド全体を広く使って攻める必要がある.また,自ら動いてスペースを作るという動きが重要視され,空いたスペースにはすばやく反応しチャンスを作るのである.このように連携を重視し,選手が意図をもって行動するためには,選手個々の動き方が重要である.そこで,攻撃時の行動として(1)スペースを空ける動き,(2)味方との距離を維持し全体のバランスを保つ動きを考えた.これらの行動によって,各選手が味方との連携を取ることによって,チームとしての戦術が機能する.

これらの行動を実現するために,選手は自分に近い味方選手2人の位置関係から,以下の規則にしたがって,次に動く方向を決定するとした.

(1) 近い選手がボールを所持する場合 (Fig. 4)
自分に近い味方選手2人のどちらかがボールを持っているのであれば,選手はスペースを空ける方向へ動く.

(2) 近い選手がボール非所持の場合 (Fig. 5)
今回,フィールドを広く使うため,各選手に他の選手との距離感を持たせた.自分に近い味方選手2人のいずれかとの距離が保たれている場合には,この選手は敵ゴール方向へ攻める動きをとる.一方,2人の選手との距離がともに保たれていない場合には味方との距離を維持する方向に動く.

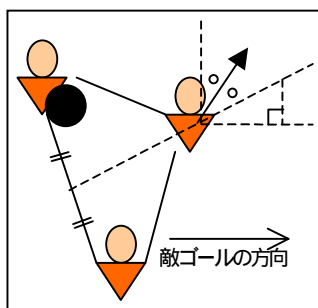


Fig. 4 スペースを空ける動き

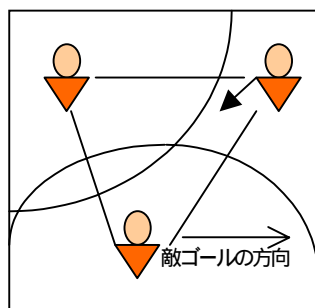


Fig. 5 味方選手との距離を維持する動き

4. Java 言語による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[5].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.選手が非同期的に行動する様子を Fig. 6 に模式的に示す.なお,フィールドオブジェクトでは,一定の時間間隔で各選手とボールの位置を表示する.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できた.

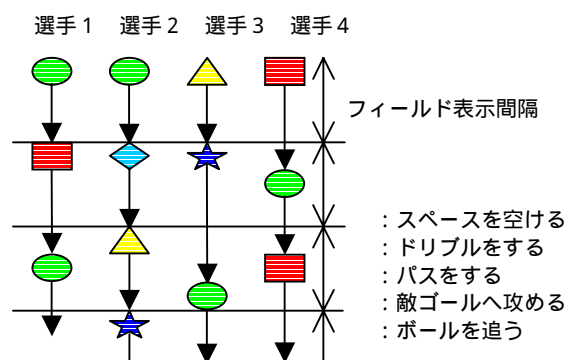


Fig. 6 選手の非同期的行動の模式図

5. まとめと今後の課題

現段階では,7人のチームによる攻撃が行える選手クラスが設計できた.各選手はパスやドリブルを行いながら敵ゴールへ移動し,最終的にシュートを打つ.さらに,スペースを空ける動きやフォーメーションを維持する動きなどを行い,チームとしての戦術を再現することができる.

今後の課題として,敵選手がいる場合の行動を追加する.敵が入ることによってスペースが狭くなるので敵を引きつける動きやトラップをせずにダイレクトなパスで味方との連携をとる行動の追加も視野に入れている.その上で,この選手クラスを継承したポジションクラス的设计を行う.それによって,守備の戦術として味方との連携を必要とするゾンプレスとオフサイドトラップの再現も目的としている.人数が増えることによって連携が複雑になっていくので,選手の動きを細かくモデル化することが重要となる.

参考文献

- [1] サッカーの戦術とフォーメーション
<http://www.rock.sannet.ne.jp/hana/soccer.htm>
- [2] 藤田一郎:最新 Java がわかる,技術評論社,pp.82-87 (1999)
- [3] 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)
- [4] 磯田定宏:オブジェクト指向モデリング,コロナ社,(1998)
- [5] 夏目 大:図解解説よくわかる Java,翔泳社,pp.25-27,pp.30-45(2000)