

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

Simulation of soccer-play with object oriented model

() ()

Soccer-play is modeled with the object oriented approach, in which a player is expressed as an object. Behavior of the player is characterized by the judgement and the movement. Based on this analysis, several classes are designed and implemented by Java. We simulate successfully soccer-play of 3 on 3.

1. はじめに

サッカーとは、「11人を超えない競技者からなる2つのチームによって行われ、競技中に得点の多かったチームを勝ちとする競技」である[1]。チームが機能するためには、1人1人がポジションに応じてその役割を果たす必要がある。特にスペースの少ない近代サッカーにおいては、このような戦術が勝敗を大きく左右する。しかし、戦術的な面を考慮しないならば、各選手はパスやドリブルなどの基本的な動作を行うという点で全選手が同じ行動をとるとみなすことができる。

本研究では、サッカー選手をひとつのオブジェクトとしてモデル化する。行動の点で全選手を同じとみなしうるので、選手クラスが1つあれば、あとは必要な人数分のインスタンスを生成すればよいことになる。したがって、サッカー選手クラスの設計が本研究の大部分を占める。

しかしながら、競技時の人数、さらにはその中の敵味方の人数によって、サッカー選手の動作や判断の数が変わってくる。そのため、11人対11人での競技ではなく、3人対3人での競技をシミュレーションすることを目標とした。そして、選手2人(味方同士)から選手6人(3人対3人)へと1人づつ段階的に人数を増やしていき、それに伴って動作と判断も追加していくという手順をとった。

以下では、選手6人(3人対3人でゴールキーパーを除く)の場合におけるサッカー選手のモデルについて述べる。さらに、Java言語を用いて実装し、シミュレーションした結果についても述べる。

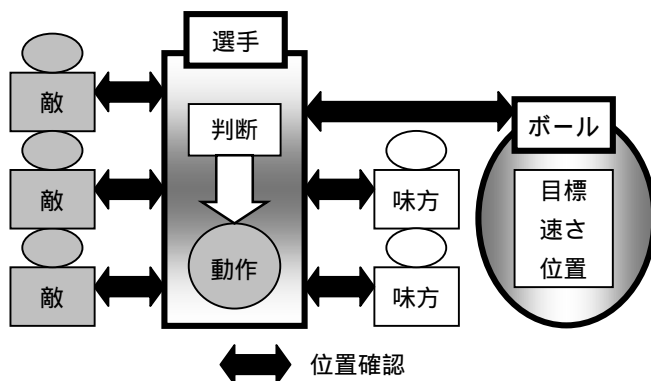


Fig. 1 選手モデルとボールモデル

2. モデル化

2.1 サッカー選手のモデル化

選手6人(3人対3人)の場合に考えられる各選手の行動を動作とそれにいたる判断という観点からまとめた。このうちすべての基本となる行動は、他の選手の位置とボールの位置の確認である。選手は競技の中で、常に他の選手の位置とボールの位置を確認することによって各判断をくだしている(Fig. 1 参照)。今回考えた判断と動作を具体的に Fig. 2 に示す。

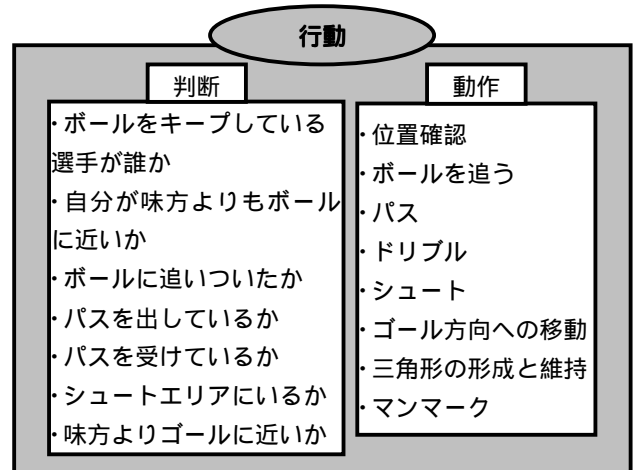


Fig. 2 判断と動作

各選手は、このような判断に基づき1つの動作を選択し実行する。また、行動は攻撃と守備に分けることができる。競技中は一方のチームの各選手は攻撃側となり、もう一方は守備側となっている。

2.2 攻撃時の行動

攻撃時における各選手の行動は、以下の5種類が考えられる。

- (1) パス
- (2) ドリブル
- (3) シュート
- (4) ゴール方向への移動
- (5) 三角形の形成と維持

このうち(1)から(3)はボールを持った選手の行動であり、(4)、(5)はボールを持っていない選手の行動である。攻撃側のチームはこれらの行動を組み合わせ、シュートをすることを目標としている。

2.3 守備時の行動

守備時における各選手の行動は、以下の2種類が考えられる。

- (1) ボールを追う
- (2) 敵をマークする

その際、(1)はボールに一番近い選手が行い、それ以外の選手2人は(2)の行動をとることにした。

2.4 攻撃と守備の切り替え

ボールを持ったチームは攻撃側になり、ボールを持っていないチームは守備側になる。ボールの所持が入れ替わると、攻撃と守備も切り替わる。各選手は自分が攻撃側なのか守備側なのかを判断しなければならない。攻撃と守備はボールを持った選手が自分と同じチームかどうかで判断することができる。ボールを持った選手を見分けるために、各選手にカウンターを設置した。カウンターはボールに触った瞬間に0になり、常に増えつづける。つまり、6人の選手の中で一番低い値を持つ選手がボールを所持していることになり、選手から離れた場所にあるボールについても最後に触った選手をボール所持とみなすことができる。このカウンターを設置したことにより、各選手は全選手のカウンターの値で攻撃か守備かを判断できるようになった(Fig. 3 参照)。

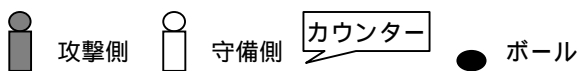
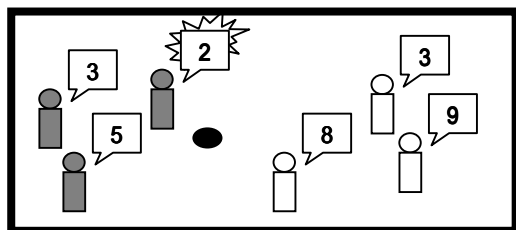


Fig. 3 攻撃と守備の切り替え

2.5 ボールのモデル化

本研究では、選手のほかにボールもモデル化している [2]。この点についても Fig. 1 の右側に示した。

3. Java による実装

前節でモデル化した選手とボールに対して、オブジェクト指向言語の1つである Java を用いて実装した。特に、各選手とボールに対して並列的に処理可能なスレッドを適用した。このことにより、各選手が独立に、さらには非同期的に行動することを表現できる。

今回、定義したクラスは以下の3種類である。

- ・グラウンドクラス(Ground)
- ・選手クラス(Soccerman)
- ・ボールクラス(Ball)

グラウンドクラスではスレッドの生成を行い、その後、選手やボール等の表示も行う[3]。今回のモデルでは選手6人(3人対3人)での競技を行うため、選手ク

ラスのインスタンス6つとボールクラスのインスタンスが1つ生成される。

また、競技シミュレーション時にボールの取り合いが起き、デッドロックになることがある。その際は、乱数を用いてカウンターの値を変更して攻撃と守備を決定しボールを取られた選手はスレッドを一時停止 [4]にする(Fig. 4 参照)ことでデッドロックを回避している。

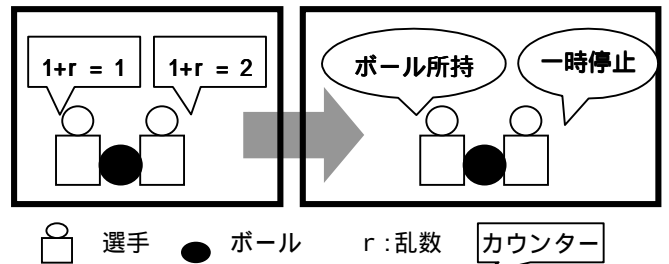
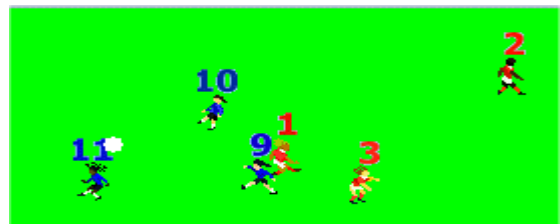


Fig. 4 デッドロックの状態の回避

4. 競技シミュレーション結果

競技シミュレーションを行う際に、各選手の移動の速さとパスを出す確率を変えていくつかの競技シミュレーションを行った。一例を Fig. 5 に示す。結果の詳細は発表時に説明する。



Aチーム: 1, 2, 3 Bチーム: 9, 10, 11

Fig. 5 競技シミュレーションの様子

5. まとめ

サッカー選手を1つのオブジェクトとしてモデル化することにより、選手6人(3人対3人)のサッカー競技シミュレーションを行うことが可能になった。今回のモデルにおいては、競技シミュレーション時の選手の移動の速さやパスの確率を変えることで競技の様子が変わることが再現された。

参考文献

- [1] サッカー競技規則
<http://www2.biglobe.ne.jp/~yanap/soccer/kisoku/kisoku.html>, (1997).
- [2] 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年度卒業研究中間発表レジュメ集, (2000).
- [3] 結城 浩: Java 言語プログラミングレッスン下, ソフトバンクパブリッシング, 第16章, (1999).
- [4] 桑原恒夫: 3日でわかる Java 共立出版株式会社, 第17章, (1999).