

AIsoccer 使用方法説明

RyosukeIMAI

平成 27 年 9 月 1 日

1 概要説明

このプログラムは「openGL」を使用して書かれたプログラムの戦略の部分を使用者それぞれが書き勝敗を競うことを目的として書きました。

2 仕様

2.1 画面説明

このプログラムでは図 1 のような座標系をとり、よく使用されと思われる座標は、「#define」により以下の文字に同等の値を持たせました。
赤が A チーム青が B チームです。

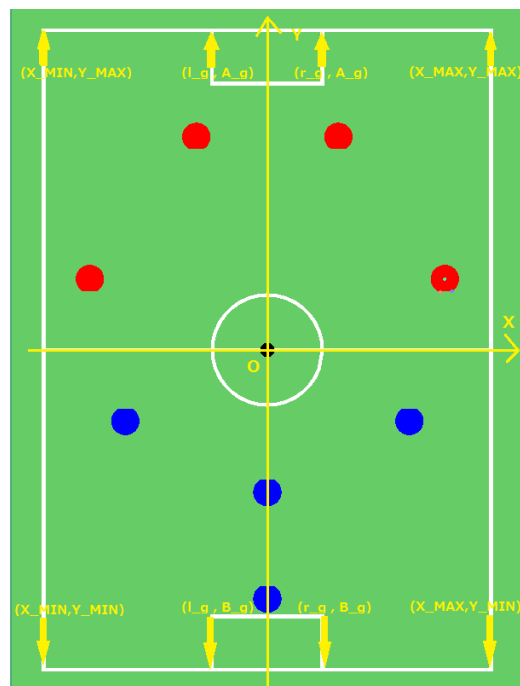


図 1: 画面説明

2.2 P_data 構造体内の変数説明

P_data 構造体はプレイヤーの状態を表す構造体です。

double : X,Y,ang

X,Y はそれぞれプレイヤーの x 座標、y 座標です。また、ang は x 軸からの時計回りにはかった向きの角度で、範囲は「 -180° ~ 180° 」です。

double : vx,vy,vang

それぞれ x の速度、y の速度、角速度です。この値は基本いじらなくても後述の関数で自動で与えられます。

int : mode,cnd,have,re

mode,cnd は戦略を立てるうえで皆さんに自由に使っていただける変数です。チーム内での分担や、行動の進み具合を表す等自由にお使いください。

have はそのプレイヤーがボールを持っている際は 1 それ以外は 0 となります。have は自動的に設定されますのでいじらないでください。

re は後述の行動する関数が完了した際に 1 となって返されます。行動終了の合図としてお使いください。

また、行動終了後に 1 となるため、必要のない場合でも `if(A[1].re == 1){A[1].re = 0}` と記述してください。でないと予期せぬバグを起こすことがあります。

2.3 B_data 構造体内の変数説明

B_data 構造体はボールの情報を表す構造体です。この値は自動的に設定されますのでいじらないでください

double : X,Y

それぞれボールの x 座標 y 座標をあらわしています。

int : have

ボールの所持者の情報です。この値が正の値なら A チームのプレイヤーが所持、負の値なら B チームのプレイヤーが所持しています。

int : state

試合の状態です。正の値なら A チーム、負の値なら B チームに所持権があります。

0 : 試合中

-1,1 : スローイン

-2,2 : ゴールキック

-3,3 : コーナーキック

3 主要関数説明

3.1 double dist(double x1 , double y1 , double x2 , double y2)

この関数は引数に二点間の座標 (x1 , y1) , (x2 , y2) を入力するとその間の距離を返します。

3.2 P_data turn(P_data play, double ang)

この関数は引数にプレイヤー構造体と目標角度 (deg) を入力することで目標角度に向かって回転し始めます。

返り値は回転するプレイヤーの P_data 構造体です。

目標角度まで回転した際には play.re が 1 になって返されますので 0 にする操作を入れてください。

使用例

```
A[1] = turn(A[1] , 45); //A[1] を 45° の向きに向かせる。
if(A[1].re == 1){      //A[1] が 45° になり A[1].re が 1 になって返ってきたら
    A[1].cnd++;        //コマンドを次へ
    A[1].re = 0;        //A[1].re を 0 に戻す。
}
```

P_data turn_G(P_data play)

この関数は turn 関数の特殊版で自分たちが攻めるゴールへ向きます。
使用方法は turn 関数と同じです。

P_data turn_P(P_data turner , P_data waiter)

この関数は turn 関数の特殊版で turner が waiter を向きます。
使用方法は turn 関数と同じです。

P_data turn_B(P_data play)

この関数は turn 関数の特殊版でボールのある方向に向きます。
使用方法は turn 関数と同じです。

3.3 P_data go(P_data play)

この関数はただ前に進む関数です。現在の向きに速度を持たせます。
返り値はそのプレイヤーの P_data 構造体です。
この関数では play.re が 1 になることはありません。

P_data G_go(P_data play , int meter)

この関数は go 関数の特殊版で自分たちが攻めるゴールとの距離が meter になるまで進みます。

go 関数と違い、行動が完了すると play.re が 1 になって返されますので 0 にする操作を入れてください。。

3.4 P_data move(P_data play , double xx , double yy)

この関数は (xx , yy) を目標地点にして移動する関数です。

返り値はそのプレイヤーの P_data 構造体です。

この関数では移動が完了したら play.re が 1 になって返されますので 0 にする操作を入れてください。

使用例

```
A[1] = move(A[1] , 100 , 400);    //A[1] を (100 , 400) へ移動させる。
if(A[1].re == 1){                  //A[1] が移動を完了して A[1].re が 1 になって返ってきたら
    A[1].cnd++;                    //コマンドを次へ
    A[1].re = 0;                   //A[1].re を 0 に戻す。
}
```

P_data get(P_data play)

この関数は move 関数の特殊版でボールを取りに行く関数です。

使用方法は move 関数と同じです。

3.5 P_data stop(P_data play)

この関数はただ play の速度を 0 にする関数です。返り値はそのプレイヤーの P_data 構造体です。

この関数では play.re が 1 になることはありません。

void stop(char side)

この関数は stop 関数のオーバーロード版で引数に 'A'もしくは'B' を持たせることでそのチーム全体の速度を 0 にします。

3.6 P_data pass(P_data passer , P_data getter)

この関数は passer が getter へパスする関数です。
passer の前方 $\pm 30^\circ$ 以内に getter が存在するまで回転します。その後ボールを getter へ向かってけりだします。

パスをし終わった際には play.re が 1 になって返されますので 0 にする操作を入れてください。

この関数は passer がボールを所持していないと機能しません。

使用例

```
A[1] = pass(A[1] , A[2]); //A[1] から A[2] へパスさせる。  
if(A[1].re == 1){        //A[1] がパスをして A[1].re が 1 になって返ってきたら  
    A[1].cnd++;           //コマンドを次へ  
    A[1].re = 0;          //A[1].re を 0 に戻す。  
}
```

3.7 P_data shoot(P_data player)

この関数は player がボールを前方へ蹴る関数です
蹴り終わった際には play.re が 1 になって返されますので 0 にする操作を入れてください。
この関数は player がボールを所持していないと機能しません。

3.8 double calc_passang(P_data passer, P_data getter)

4 戦略の書き方と書く関数

ここからは戦略の何をどの関数に書いていけばいいのかを説明します。

4.1 A_(B_)init

この関数内には初期の位置と角度を記述してください。また、プレイヤー構造体内の変数リセットも記述できます。

書き方例

```
void A_init(void){
    A[1].X = 500;
    A[1].Y = 200;
    A[1].ang = -90;
    for(int i = 0 , i <= PLAYER , i++){//諸コマンド初期化
        A[i].v = 0;
        A[i].vang = 0;
        A[i].cnd = 0;
        A[i].mode = 0;
    }
}
```

4.2 A_(B_)offence_init

この関数には自分のチームが攻撃スタートの時の位置等を記述してください。
記述方法は init と同じです。

4.3 A_(B_)strategy

この関数が戦略を記述する関数です。
3章で出てきた関数たちを駆使する等をして戦略を組んでください。
例はサンプルプログラムをご覧ください。

4.4 A_(B_)lineover_init

これはボールが線を割った時の位置等を記述する関数です。
この関数内にはあらかじめ switch 文が書いてあり横にコメントでスローインなどを書いてあります。
それぞれの場合について位置等の情報を記述してください。

5 操作方法

このプログラムはデバック用にキーボード等で開始一時停止ができます。
「左クリック」：ボールをその位置へ移動させ開始
「s」：スタート
「a」：ストップ

6 注意点とお願い

このプログラムはまだ作成途中です。
したがって予期せぬ不具合が起こるかもしれません。その際は制作者へフィードバックをお願いします。

さらにプログラム内のパラメータ等をいじることでありえない動き等が可能ですが、良識の範囲内をお願いします。

また、手続き型で書かれている本プログラムですが、そのうちオブジェクト指向型のプログラムへ変更予定です。その際は少し戦略の書き方が変わりますがご了承ください。

最後にこのプログラムを制作者を介さずに他の人に公開しないでください。よろしくお願いいたします。

現在このマニュアルも試作品なので、不備だらけだと思うので、不明点は遠慮なく制作者へ問い合わせてください。

戦略の書き方のご相談等もわかる範囲で受け付けます。