机器人技术实验56题目

球员的类型-60是球员的编号

因为枚举类型从60开始

play on 模式表示 if (WM->isBallKickable()){}

23题有对所有球员的遍历

在本方角球的情况下

else if (WM->isCornerKickUs()) {}

在本方界外球的情况下

else if(WM->isOffsideUs()) { }

这里表示随机一点

```
VecPosition pos(-(ball.getX())/fabs(ball.getX()) * 5 + ball.getX(),
-(ball.getY())/fabs(ball.getY()) * 5 + ball.getY());
```

// 拿球后行为，利用已有 WorldmodelWorldmodel（21）
(1)在 playOn 模式下，拿到球 以后朝前方快速带。

```
soc=dribble(0.0,DRIBBLE_FAST);//0.0表示向前方，第二个参数是带球方式
```

(2)在 PlayOn 模式下，拿到球 以后朝门方向慢速带。

```
AngDeg ang=(VecPosition(52.5,0.0)-posAgent).getDirection();//获得球门方向
soc=dribble(ang,DRIBBLE_SLOW);
```

(3)在 playOn 模式下，拿到球以后把围绕自己身体逆时针转 。

```
soc=kickBallCloseToBody(-30);//逆时针转动
```

(4)在 playOn 模式下， 拿到球后， 有人逼抢 (自身周围 7米范围 有至少 1名对 方 球员)，则把球踢到距离对手的另外一侧，安全 带球（如对手在右侧，把球踢到左侧 ，如对手在左侧，把球踢到右侧）。

```
Circle cir(posAgent,7);//以智能体所在位置为圆心，自身范围为7创建一个圆  ，posagent表示当前执行代码的球员本身
int num=WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS,cir);//获得在这个圆内对手的数量
AngDeg ang=0;
if(num>0){
 //获得距离球员最近的对手对象
 ObjectT o=WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,posAgent);
 //获得对象所在的位置
 VecPosition p=WM->getGlobalPosition(o);
 //当>=0时说明p(对手)在球员的顺时针方向即右侧
 if((p-posAgent).getDirection()>=0){
    ang+=45;//对手在球员右侧，向左侧带球
    soc=dribble(ang,DRIBBLE_WITHBALL);
 }
 else{
    ang-=45;//对手在球员左侧，向右侧带球
    soc=dribble(ang,DRIBBLE_WITHBALL);
 }
}
else{
 //向球门方向带球
 AngDeg ang=(VecPosition(52.5,0.0)-posAgent).getDirection();
 soc=dribble(ang,DRIBBLE_WITHBALL);
}
```

(5)在 playOn 模式下，拿到球以后，有人逼抢，传球给最近的队友；否则向球门
方向快速带球 。

```
Circle cir(posAgent,7);
int num=WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS,cir);
if(num>0) {
    //有人逼抢获取离自己最近的队员进行传球,传球的
    soc=leadingPass(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),1);
    //参数1表示将球传递到距离被传球队员大约1m的位置
}
else{
    AngDeg ang=(VecPosition(52.5,0.0)-posAgent).getDirection();
    soc=dribble(ang,DRIBBLE_FAST);
}
```

(6)在 playOn 模式下，如果 有人逼抢 (自身周围 7米范围有至少 1名对方球员)， 则安全带球；否则向门方快速 带球。

```
Circle cir(posAgent,7);
//获得这个圆形内对手球员的个数
int num=WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS,cir);
AngDeg ang=(VecPosition(52.5,0.0)-posAgent).getDirection();
if(num>0)
    soc=dribble(0.0,DRIBBLE_WITHBALL); //安全带球
else
    soc=dribble(ang,DRIBBLE_FAST);
```

(7)在 PlayOn 模式下， 若前方没有对方球员，则直接以最大速度 向对方 球门 射门 （周期数为偶，球门的右侧射奇向的左侧射门）。

```
AngDeg ang = (VecPosition(52.5,0.0)-posAgent).getDirection();
//该函数计算出球门正前方100米处是否有对方球员
if(WM->isOpponentAtAngle(ang,100)==false){

VecPosition posGoal( PITCH_LENGTH/2.0, (-1 + 2*(WM->getCurrentCycle()%2)) * 0.4 *
SS->getGoalWidth() ); //交替进攻球门的下方或者上方

 soc = kickTo( posGoal, SS->getBallSpeedMax() );

}
```

(8)在 playOn 模式下， 拿到球 后，在本方半场 踢到球场中心点； 过了半场， 快速 带球到对方球门 。

```
if(WM->getBallPos().getX()<0) //如果在我方半场
    soc=kickTo(VecPosition(0,0),1.0);//将球以速度1.0踢向球场中心点
else{
 AngDeg ang=(VecPosition(52.5,0)-posAgent).getDirection();
 soc=dribble(ang,DRIBBLE_FAST);//向球门方向快速带球之前一定要知道方向因为带球的参数是一个角
度
}
```

(9)在 PlayOn 模式下，拿到球后，把传给最近的周围没人防守的队友（没有人防
守以其周围 5米范围是否有对方球员为准）脚下。

```
//以离自己最近的球员为圆心，5m为半径画圆
Circle cir(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),5);
int num=WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS,cir);
if(num==0)
 //以速度1.0将球传给这个球员
 soc=leadingPass(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),1.0);
else soc=dribble(0.0,DRIBBLE_WITHBALL);
//否则向前带球
```

(10)在 playOn 模式下，拿到球后，把球传给最靠近自己的前方的没人防守的
队友（判断身边 5米范围是否有对方防守队员）。

```
Circle cir(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),5);
int num=WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS,cir);
//获得最靠近自己球员的坐标
VecPosition diff=WM->getGlobalPosition(WM-
>getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent));
AngDeg ang=(diff-posAgent).getDirection();//获得两个球员间的角度
if(num==0&&ang>=-90&&ang<=90)//判断是否在前方且没人防守
    soc=leadingPass(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),1.0);
else
    soc=dribble(0.0,DRIBBLE_WITHBALL);
```

(11)在 playOn 模式下，拿到球以后，在本方半场传给次近的队友；在对方
半场，非 10 号球员传给 10 号球员， 10 号球员则快速向门方带球。

```
//当在我方半场
if(WM->getBallPos().getX()<0)
 //找到次近的队友
 soc=leadingPass(WM->getSecondClosestInSetTo(OBJECT_SET_TEAMMATES,WM-
>getAgentObjectType()),1.0);
else{                                  //也就是我方的10号球员
    if(WM->getPlayerNumber()!=10)  //我的类型减去60就是编号
        soc=leadingPass(OBJECT_TEAMMATE_10,1.0);
    else
        soc=dribble((VecPosition(52.5,0.0)-
posAgent).getDirection(),DRIBBLE_FAST);
}
```

(12)在 playOn 模式下，如果在本方半场，则朝前方慢速带球，如果在对方半场，则朝球 门快速带球。

```
if(WM->getBallPos().getX()<0)
    soc=dribble(0.0,DRIBBLE_SLOW);
else
    soc=dribble((VecPosition(52.5,0.0)-posAgent).getDirection(),DRIBBLE_FAST);
```

(13)在 playOn 模式下，拿到球后 ，如果是 2号，则把球踢到左侧边 线，如果是 5号，则把球踢到右侧边线，并把脖子方向转向球；其他球员则向前带球。

```
//整个球场宽度为68，左侧边界为-34，右侧边界34
if(WM->getPlayerNumber()==2)
 soc=kickTo(VecPosition(WM->getBallPos().getX(),-34),1.0);
else if(WM->getPlayerNumber()==5)
{
  soc=kickTo(VecPosition(WM->getBallPos().getX(),34),1.0);
  //把脖子转向球也就是一直看着球
  ACT->putCommandInQueue(turnNeckToObject(OBJECT_Ball,soc));
}

else soc=dribble(0.0,DRIBBLE_WITHBALL);
```

(14)在 playOn 模式下 ，拿到球后 ，如果 4号，则传球给 7号; 否则的话 ，传 球给最近的队友 ；到对方禁区后以最大速度射向空隙的球门一侧 。

```
//到达对方禁区
if(WM->isInTheirPenaltyArea(WM->getBallPos())) {
 //获得对手守门员的位置
 VecPostion posGoalie=WM->getGlobalPosition(WM->getOppGoalieType());
 //获得球员与守门员之间的角度
 AngDeg ang_goalie=(posGoalie-posAgent).getDirection();
 //获得向上的角度
 AngDeg angup=(VecPosition(52.5,6.0)-posAgent).getDirection();
 //获得向下的角度
 AngDeg angdown=(VecPosition(52.5,-6.0)-posAgent).getDirection();

 if(std::fabs(angup-ang_goalie)>std::fabs(angdown-ang_goalie))  //std中含有一些标准
函数这个函数用来计算绝对值
 //如果向上射门的角度>向下射门的角度
```

```
  soc=kickTo(VecPosition(52.5,6.0),SS->getBallSpeedMax());
 //则向上以最大速度射门
 else
  soc=kickTo(VecPosition(52.5,-6.0),SS->getBallSpeedMax());
}

if(WM->getPlayerNumber()==4)

 soc=leadingPass(OBJECT_TEAMMATE_7,1.0);

else leadingPass(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),1.0);
```

(15)在playOn模式下，拿球后垂直带球。

```
//球在下半场向上90角度带球
if(WM->getBallPos().getY()<=0) soc=dribble(90,DRIBBLE_WITHBALL);
//球在上半场向下-90度带球
else soc=dribble(-90,DRIBBLE_WITHBALL);
```

(16)在playOn模式下，拿到球后带球到达球场中心；在球场中心附近位置（中心位置周围2米半径范围内）传给最近的球员。

```
//如果拿球的队员到球场中心的位置>2.0则向球场中心跑去
if(posAgent.getDistanceTo(VecPosition(0,0)>2.0)) {
 AngDeg ang=(VecPosition(0,0)-posAgent).getDirection();
 soc=dribble(ang,DRIBBLE_WITHBALL);
}
//到球场中心范围内传给最近的球员
else
    soc=leadingPass(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),1.0);
```

(17)在playOn模式下，如果10号控球，则其带球前进，然后5号跟着10号一起前进，两球员在同一水平线上，且距离为5。

```
//如果不是10号球员将球传给10号球员

if(WM->getPlayerNumber()!=10)

 soc=leadingPass(OBJECT_TEAMMATE_10,1.0);
//如果是10号球员
else soc=dribble(0.0,DRIBBLE_SLOW);


if(WM->getPlayerNumber()==5) {

 VecPosition pos=(WM->getBallPos().getX()-5,WM->getBallPos().getY());

 soc=moveToPos(pos,20); //当这个点与球员角度之间角度<20的时候转向

}
```

(18)在playon模式下，如果是10号球员，在可踢球的状态下，如果自身的x轴坐标大于30，则直接朝着离对方球员远的球门点射门。

```
if ( WM->isBallKickable())
{
    if(WM->getPlayerNumber()==10&&posAgent.getX()>30)
    {
        VecPosition posGoalie=WM->getGlobalPosition(WM->getOppGoalieType());
        AngDeg ang_goalie=(posGoalie-posAgent).getDirection();
        AngDeg angup=(VecPosition(52.5,6.0)-posAgent).getDirection();
        AngDeg angdown=(VecPosition(52.5,-6.0)-posAgent).getDirection();
     if(std::fabs(angup-ang_goalie)>std::fabs(angdown-ang_goalie))
        soc=kickTo(VecPosition(52.5,6.0),SS->getBallSpeedMax());//朝着距离对方球员远的球门点进行射门
     else
        soc=kickTo(VecPosition(52.5,-6.0),SS->getBallSpeedMax());
    }
}
```

(19)在playOn模式下，如果是9号拿到球，则令9号和10号同时冲至敌方球门处，在球门前，9号传给10号，由10号进行射门

```
 //如果不是9号球员   也可以使用getNumber函数
if(WM->getAgentObjectType()!=OBJECT_TEANMATE_9) {
 //将球传给9号球员
  soc=leadingPass(OBJECT_TEAMMATE_9,1.0);
 //获得球门方向的角度
  AngDeg ang=(VecPositoin(52.5,0)-posAgent).  getDirection();
 //向球门快速带球
  soc=dribble(ang,DRIBBLE_FAST);

}
//同时10号球员向球门方向前进
if(WM->getAgentObjectType()==OBJECT_TEANMATE_10)
     soc=moveToPos(VecPosition(52.5,0),20)；
//到达球门正前方传球给10号球员   也就是到达禁区位置
if(WM->isInTheirPenaltyArea(WM->getBallPos())){

  soc=leadingPass(OBJECT_TEAMMATE_10,1.0);

}
//10号球员进行射门
if(WM->getAgentObjectType()==OBJECT_TEANMATE_10){

  soc=kickTo(VecPosition(52.5,0),SS->getBallSpeedMax())；
 }
```

(20)在playOn模式下，如果我是4号球员并且拿到了球，则传给7号球员，同时7号球员再传给9号球员，9号球员继续以最大速度冲到球门处射门。

```
if(WM->isBallKickable())
{
    if(WM->getPlayerNumber()==4)
    {
        soc=leadingPass(OBJECT_TEAMMATE_7,1);
    }
   if(WM->getPlayerNumber()==7) //一个表示对象，另一个表示的是球员的编号

     soc=leadingPass(OBJECT_TEAMMATE_9,1);

   if(WM->getPlayerNumber()==9)
   {
        if(WM->getBallPos().getX()!=52.5) //球门的位置
        {
            soc=dribble((VecPosition(52.5,0)-
posAgent).getDirection(),DRIBBLE_FAST);
        }
       //到达球门处进行射门
        else soc=kickTo(VecPosition(52.5,0),SS->getBallSpeedMax())
   }
}
```

(21)在playOn模式下，如果在我方半场拿到球，则向中场线以最大的速度踢，如果在敌方半场拿到球，则向敌方球门处以最大的速度踢

```
//当在我方半场
if(WM->getBallPos().getX()<=0){
 VecPosition pos(0,WM->getBallPos().getY());//计算出中场线的坐标
 soc=kickTo(pos,SS->getBallSpeedMax());//朝中场线踢
}else
    soc=kickTo(VecPosition(52.5,0),SS->getBallSpeedMax());
//在对方半场向球门处以最大速度踢球
```

// 拿球后行为,自己添加WorldModel函数

(22)在playOn模式下，拿球后传球给更靠近对方球门的最近队友。

```
if ( WM->isBallKickable())
{
    soc=leadPass(WM->getClosestInSetTo(OBJECT__SET__TEAMMATES,WM-
>getPosOpponentGoal()),1.0);//更加靠近对方球门的球员
}
```

(23)在playOn模式下，拿球后，搜索前方-30~30之间距离自己20米内是否有队友，如果有则传给该队友，否则自己带球。

```
//WorldModel.cpp
ObjectT WorldModel::getTeammateAtAngleEx(AngDeg angA, AngDeg angB, double dDist)

{

  VecPosition posAgent = getAgentGlobalPosition();
  VecPosition posTeam;

  AngDeg angTeam;

  int iIndex;
  //遍历所有的本方队员
  for(ObjectT o = iterateObjectStart(iIndex,OBJECT_SET_TEAMMATES);
     o != OBJECT_ILLEGAL;
     o = iterateObjectNext(iIndex,OBJECT_SET_TEAMMATES))
  {
   //获得当前坐标
   posTeam = getGlobalPosition(o);
   //获得我和被遍历的teammate的之间的角度
   angTeam = (posTeam - posAgent).getDirection();

   if(angA <= angTeam && angTeam <= angB && posAgent.getDistanceTo(posTeam) <
dDist)
   //当前o代表的本方队员在范围angA-angB之间且距离我的距离小于dDist
    return o;

  }

  iterateObjectDone(iIndex);
  //否则返回未找到这个对象
  return OBJECT_ILLEGAL;

}

  //Playerteam.cpp
  if ( WM->isBallKickable())

   {

    double ang = (VecPosition(52.5,0)-posAgent).getDirection();//获得我和球门之间的
夹角也就是正前方的位置

       ObjectT o = WM->getTeammateAtAngleEx(ang-30, ang, 20);//在-30  -   30之间
       //存在一个队员
       if( o != OBJECT_ILLEGAL)

       {
        soc = leadingPass(o,1);
       }

       else

       {
        soc = dribble(ang, DRIBBLE_SLOW);//自己朝着球门带球
       }
   }
```

(24)在playon模式下，求出y轴等于0的两侧的对方球员数量，将球传向对手少的一方，并且x轴值最大的队友。

```
//WorldModel.cpp下
ObjectT WorldModel::getMaxXTeammateInSide(bool isOwnSize)//isOwnSize参数表示上半场的对手球员数目多或少，为真则多
{
    int iIndex;
    ObjectT maxMate = OBJECT_ILLEGAL;
    for (ObjectT  o = iterateObjectStart(iIndex, OBJECT_SET_TEAMMATES);
                          o != OBJECT_ILLEGAL;
                          o = iterateObjectNext(iIndex, OBJECT_SET_TEAMMATES))
//遍历球场上的所有队员，将结果存放在o中
        {
            VecPosition oPos = getGlobalPosition(o);//获得o的坐标
            if((isOwnSize && oPos.getY() >= 0) ||
                 (!isOwnSize && oPos.getY() <= 0))//上半场人多需要将球传到下半场的球员脚下
            {
                if (maxMate == OBJECT_ILLEGAL || oPos.getX() > getGlobalPosition(maxMate).getX())
                     maxMate = o;
            }
        }
        iterate ObjectDone(iIndex);//遍历结束
        return maxMate;//返回这个球员对象
}
 //以下函数作用是返回哪边的球员数量多  如果为true则y>0的区域对方球员的数量在上半场多，反之在下半场对方球员多
 bool   WorldModel:: isOwnSideOpponentMost()
 {
    int ownSideCount = 0;
    int count = 0;
    int iIndex;
    ObjectT maxMate = OBJECT_ILLEGAL;
    for (ObjectT  o = iterateObjectStart(iIndex, OBJECT_SET_OPPONENTS);
                          o != OBJECT_ILLEGAL;
                          o = iterateObjectNext(iIndex, OBJECT_SET_OPPONENTS))

     {
         if (getGlobalPosition(o).getY() > 0)
             ownSideCount++;//上半场对手球员数目
         count++;//遍历的总球员数
     }
     return ownSideCount > count - ownSideCount;//上半场球员数>下半场时返回真
 }

 //PlayerTeams.cpp
else if( WM->isBallKickable())
{
    ObjectT mate = WM->getMaxXTeammateInSide(WM->isOwnSideOpponentMost());
    soc = leadingPass(mate, 1);//传给这个球员对象
    ACT->putCommandInQueue(soc);

    ACT->putCommandInQueue( turnNeckToObject(OBJECT_BALL,soc));
    return soc;

}
```

(25)在playon模式下，如果自身7米范围内有两个或两个以上的对手的话，则传球到x轴值最大的队友。

```
//在WorldModel类中添加
ObjectT WorldModel::getMaxXTeammate(){

  int iIndex;

  ObjectT maxMate = OBJECT_ILLEGAL;

  for (ObjectT o = iterateObjectStart(iIndex, OBJECT_SET_TEAMMATES);

             o != OBJECT_ILLEGAL;

             o = iterateObjectNext(iIndex, OBJECT_SET_TEAMMATES)) {

       VecPosition oPos = getGlobalPosition(o);

         if (maxMate == OBJECT_ILLEGAL || oPos.getX() >
getGlobalPosition(maxMate).getX())

           maxMate = o;     }

     iterateObjectDone(iIndex);

     return maxMate;

}
//cpp

if ( WM->isBallKickable())

{

      Circle cir(posAgent, 7);

     int num = WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS, cir);

     if (num >= 2) {

         ObjectT mate = WM->getMaxXTeammate();

          soc = leadingPass(mate, 1);
     }
}
```

***(26)在playon模式下，求出己方的越位线，存在可以踢球的队友的话，如果自身是10号。那么10球员垂直跑向越位线的x轴值-2米，y轴不变的点。

```
if(WM->isBallInOurPossesion() && WM->getPlayerNumber==10)
```

```
•     soc=moveToPos(WM->getOffsideX()-2,posAgent.getY(),20);
可能是正解考试按这个来写
//WorldModel.cpp

//：：表示作用域
double WorldModel::getLastTeammateDefenderX(double *dX)
{
    double dHighestX=0.0;
    double dSecondX=0.0,x;
    ObjectT o,oLast=OBJECT_ILLEGAL,oSecondLast=OBJECT_ILLEGAL;
    for(int i=0;i<MAX_TEAMMATES;i++)
    {
        o=Teammates[i].getType();
        if(isConfidenceGood(o))
        {
            x=Teammates[i].getGlobalPosition().getX();
            if(x>dHighestX)
            {
                dSecondX=dHightestX;
                dHightestX=x;
                oSecondLast=oLast;
                oLast=o;
            }
            else if(x>SecondX)
            {
                dSecondX=x;
                oSecondLast=o;
            }
        }
    }
    if(dHightestX<PENALTY_X&&getOwnGoalieType()==OBJECT_ILLGAL)
    {
        dSecondX=dHighestX;
        oSecondLast=oLast;
    }
    if(dX!=Null)
    {
        *dX=dSecondX;
    }
    return (WM->getGlobalPositon(oSecondLast))->getX();
}
//Playerteam.cpp
if(WM->isBallKickable())
{
    if(WM->isBallInOurPossesion() && WM->getPlayerNumber==10) //第一个参数表示判断球是否由我方队员掌控
      soc=moveToPos((WM->getLastTeammateDefenderX()-2,posAgent.getY()),20); //跑向所要求的点
}
```

(27)在playon的模式下，求出球运动方向（即求出其对应的直线方程），并且求出自己到该直线的距离，如果距离小于4的话，那么就垂直跑向该条直线。

```
if(isBallKickable())
{
    Line ballRun = Line::makeLineFromPositionAndAngle(WM->getBallPos(), WM-
>getBallDirection());//求出球的运动方向，传入参数是一个点和角度然后通过这两个值计算出直线方程
//计算出该条直线到达Agent的距离
if (ballRun.getDistanceWithPoint(WM->getAgentGlobalPosition()) < 4){
//垂直跑向这条直线，也就是找到这条线上某一点到达这个Agent的最短
 soc = moveToPos(ballRun.getPointOnLineClosestTo(WM->getAgentGlobalPosition()),
20);
}
```

(28)在playon的模式下，如果是自己可以踢球的状态下，如果自身7米内没有对方球员的话，则快速带球，带球方向是朝着点（53,0）方向。

```
Circle cir(posAgent,7);

int num=WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS,cir);

AngDeg ang=(VecPosition(53,0)-posAgent).getDirection();

if(num>0) //有对方球员
 soc=dribble(0,DRIBBLE_WITHBALL);

else //朝着53，0方向带球
 soc=dribble(ang,DRIBBLE_FAST);
```

(29)在playon的模式下，如果距离球最近的是我方队友的话，如果我的x坐标小于30的话，并且我是10号，那么我跑向球的坐标加上（0,10）的坐标位置。

```

if(WM->isBallInOurPossesion() && WM->getPlayerNumber==10 && posAgent.getX()<30)
{
    VecPosition pos=WM->getBallPos()+VecPosition(0,10);
    soc=moveToPos(pos,20);
}
```

(30)在playon的模式下，发现前方没有队友的时候，如果自身5米内有两个及两个以上的对手的话，那么将球传给最近的队友。

```
//在worldmodelHighLevel.cpp中：仿照isOppenentAtAngle

bool WorldModel::isTeammateAtAngle( AngDeg angA , AngDegB，double dDist  ){

VecPosition posAgent  = getAgentGlobalPosition();//获得自身的位置

  VecPosition posTeam;

  AngDeg   angTeam;

  int     iIndex;
  for ( ObjectT o = iterateObjectStart( iIndex, OBJECT_SET_TEAMMATES ); o != OBJECT_ILLEGAL; o = iterateObjectNext ( iIndex, BJECT_SET_TEAMMATES ) )
  {
    posTeam  = getGlobalPosition( o );//获得我这个队友的位置坐标
    angTeam = ( posTeam - posAgent ).getDirection() ;//获得队友与我的角度，ang是球门与我的角度
    if(angA<=angTeam&&angTeam<=angB&&posAgent.getDistanceTo(posTeam)<dDist)
    {
        return true;//在这个角度范围内有队友
    }
  }

  iterateObjectDone( iIndex );

  return false;//前方没有队友

}

//在playerTeams.cpp中：

if(WM->isBallKickable())
{
   Circle cir(posAgent,5);

   int num=WM->getNrInSetInCircle(OBJECT_SET_OPPONENTS,cir);

   AngDeg ang = (VecPosition(52.5,0)-posAgent).getDirection();

   if(WM->isTeammateAtAngle(ang-45,ang+45,10)==false && num>=2)//在前方
   {
      soc=leadingPass(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,posAgent),1.0);
   }

}
```

(31)在 playon 的模式下，发现前方没有对方球员的时候，则快速向球门方向带球，距离守门员5-10米时向空隙较大的一侧射门。

```
if(isBallKickable())
{
    AngDeg ang = (VecPosition(52.5,0)-posAgent).getDirection();

•       if(WM->isOpponentAtAngle(ang,20)==false){

•        soc = dribble(ang,DRIBBLE_FAST);

•       }

•       VecPosition posOppGoalie = WM->getGlobalPosition(WM->getOppGoalieType());

•       double dist = posOppGoalie.getDistanceTo(posAgent);

•       if(dist >= 5 &&dist <= 10)
•       {
•         AngDeg ang_goalie=(posOppGoalie-posAgent).getDirection();


•         AngDeg angup=(VecPosition(52.5,6.0)-posAgent).getDirection();


•         AngDeg angdown=(VecPosition(52.5,-6.0)-posAgent).getDirection();


•        if(fabs(angup-ang_goalie)>fabs(angdown-ang_goalie))
•         soc=kickTo(VecPosition(52.5,6.0),SS->getBallSpeedMax());
•        else
•         soc=kickTo(VecPosition(52.5,-6.0),SS->getBallSpeedMax());
•       }
}
```

(32)在 playon 的模式下，如果在对方禁区内就射门，否则，如果是7号球员就向球门方向带球，其他球员就将球传给7号。

```
if(isBallKickable())
{
    VecPosition opp=WM->getPosOppoentGoal();//获得对方球门的坐标

    if(WM->isInTheirPenaltyArea(WM->getBallPos())) //如果球在对方禁区中
    {
        soc=kick(opp.getDirection(),SS->getBallSpeedMax());
    }
    else
    {
        if(WM->getNumber()==7)
        {
            soc=dribble(opp.getDirection(),DRIBBLE_WITHBALL);  //安全带球
        }
        else
        {
            soc=leadingPass(OBJECT_TEAMMATE_7,1.0);
        }
    }
}
```

(33)在 playon 的模式下，（8，9，10，11）前方没有对方球员的时候，则快
速向球门方向带球，距离守门员 5-10 米时向空隙较大的一侧射门。

```
else if (WM->isBallKickable()) // if kickable // 如果球已知，而且当前球在我脚下(可踢)
 {
  int number = WM->getPlayerNumber();//获得球员的编号
  if (number >= 8 && number <= 11){
   AngDeg ang = (VecPosition(52.5, 0) - posAgent).getDirection();
   if (WM->isOpponentAtAngle(ang, 20) == false){
    soc = dribble(ang, DRIBBLE_FAST);
   }
   VecPosition posOppGoalie = WM->getGlobalPosition(WM->getOppGoalieType());
   double dist = posOppGoalie.getDistanceTo(posAgent);
   if (dist >= 5 && dist <= 10){
        AngDeg ang_goalie = (posOppGoalie - posAgent).getDirection();
        AngDeg angup = (VecPosition(52.5, 6.0) - posAgent).getDirection();
        AngDeg angdown = (VecPosition(52.5, -6.0) - posAgent).getDirection();
    if (fabs(angup - ang_goalie) > fabs(angdown - ang_goalie))
        soc = kickTo(VecPosition(52.5, 6.0), SS->getBallSpeedMax());
    else
        soc = kickTo(VecPosition(52.5, -6.0), SS->getBallSpeedMax());
   }
  }
 }
```

// 其他比赛模式下，进攻行为其他比赛模式下，进攻行为（8）

(34)在本方角球模式下，如果自己是 10 号球员，则跑向角球点，并开球（球可踢，则踢球给9号）；如果自己是9号球员，则跑向距离角球点附近（随机选一点），准备接应球，其他球员跑本位点.

```
else if (WM->isCornerKickUs())

{

   if (WM->getAgentObjectType() == OBJECT_TEAMMATE_10)

  {

•    if (WM->isBallKickable())

•      soc = leadingPass(OBJECT_TEAMMATE_9, 1);

•    else

•      soc = moveToPos(WM->getBallPos(), PS->getPlayerWhenToTurnAngle());

  }

  if (WM->getAgentObjectType() == OBJECT_TEAMMATE_9)  {

•    VecPosition ball = WM->getBallPos();

•    VecPosition pos(-(ball.getX())/fabs(ball.getX()) * 5 + ball.getX(),

•            -(ball.getY())/fabs(ball.getY()) * 5 + ball.getY());

•    soc = moveToPos(pos, PS->getPlayerWhenToTurnAngle());
```

```
  }
    ACT->putCommandInQueue( soc );

    ACT->putCommandInQueue( turnNeckToObject( OBJECT_BALL, soc ) );
}
```

(35)在本方界外球模式下，如果自己是距离球最近的队员，跑向球；并开球（球可踢，则把球传给最近的队友）。

```
else if(WM->isOffsideUs()) {

  if(WM->getFastestInSetTo( OBJECT_SET_TEAMMATES, OBJECT_BALL, &iTmp )

•         == WM->getAgentObjectType())
  {
     //球可踢
•    if(WM->isBallKickable()) {

•       ObjectT objTea =WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,WM->getAgentObjectType(),&dist);

•

•      soc=leedingpass(objTea,1.0);

•    }
     //球不可踢
•    else {

•      soc=moveToPos(WM->getBallPos(),20);

•    }

•    ACT->putCommandInQueue(soc);

  }

}
```

(36)在本方界外球模式下，在本方半场左侧，则由 2 号去发；如果是本方半场右侧，则由5号去发；球在2号或者5号脚下，则踢向距离自己最近的队友。

```
else if(WM->isOffsideUs()) {

  VecPosition posBall=WM->getBallPos();

  if((posBall.getX()<0 && posBall.getY()<0 && WM->getPlayerNumber()==2)||

•      ((posBall.getX()<0 && posBall.getY()>0 && WM->getPlayerNumber()==5)))  {

•    if(WM->isBallKickable()) {
```

```
•      ObjectT objTea =WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,WM-
>getAgentObjectType(),&dist);

•      VecPosition posTea=WM->getGlobalPosition(objTea);

•      soc=kickTo(posTea,SS->getBallSpeedMax());

•    }else {

•      soc=moveToPos(WM->getBallPos(),20);

•    }

•    ACT->putCommandInQueue(soc);

  }

}
```

(37) 在本方界外球的模式下，如果我是距离球第二近的队友的话，那么我也朝着球的位置跑，直到距离球7米的范围。

```
else if(WM->isOffsideUs()){

 ObjectT o=WM->getSecondClosestInSetTo(OBJECT_SET_TEAMMATES,OBJECT_BALL) ;

 double myPosx=WM->getAgentGlobalPosition().getX();

 double myposy=WM->getAgentGlobalPosition().getY();

 double ballposx=WM->getBallPos().getX();

 double ballposy=WM->getBallPos().getY();

 double dis=sqrt(pow(ballposx-myPosx,2)+pow(ballposy-myposy,2));

 if(WM->getAgentObjectType()==o&&dis>7) {

  soc=moveToPos( VecPosition(ballposx,ballposy ) , 20 );

  ACT->putCommandInQueue( soc );

 }

}
```

(38)在本方界外球的模式下，如果我是 4 号的话，并且距离球最近的队友不是我，那么我将跑向球的坐标加上（5,0）的位置上去。

```
else if(WM->isOffsideUs()){

if((WM->getPlayerNumber()==4)&&(WM-> getClosestInSetTo( OBJECT_SET_TEAMMATES,
OBJECT_BALL)!=OBJECT_TEAMMATE_4))

  soc=moveToPos((WM->getBallPos()+VecPosition(5,0)),20);

  ACT->putCommandInQueue( soc ); // 放入命令队列

  ACT->putCommandInQueue( turnNeckToObject( OBJECT_BALL, soc ) );

}
```

(39)在本方界外球的模式下，如果我是距离球第二近的队友，那么我跑向距离球 12 米的范围内，并且距离对方球门点（52.5,0）最近的点。

$$(1+k^2)x^2+(-2ballx+2kb1-2kbally)x+(ballx^2+b1^2-2b1bally+bally^2-144)=0$$

$$deita=(-2ballx+2kb1-2kbally)^2-4(1+k^2)(ballx^2+b1^2-2b1bally+bally^2-144)$$

```
else if ( WM->isOffsideUs() ){

  ObjectT o = WM -> getSecondClosestInSetTo(OBJECT_SET_TEAMMATES , OBJECT_BALL) ;

  if ( WM ->getAgentObjectType() == o ) {

      //由于需要跑到距离对方球门最近的点也就是球门中心点与以球为圆心12为半径的圆交点
      double ballx = WM->getBallPos().getX() ;

      double bally = WM->getBallPos().getY() ;

      double k ,b1 ,deita, x1, x2, y1, y2;;
      // 计算以球为圆心，12为半径的圆与对方球门线（x=52.5）的交点

       k=bally/( ballx-52.5);// 计算交点所在直线的斜率 设y=kx+b1

       b1=0-k*52.5;//计算出截距(代入球门的坐标(52.5,0))       也就是y=tx+m;之后联立两个
方程
        //即：(x-ballx)*2+(y-bally)*2=144 与直线 y=kx+b1

      //这个式子的结果推导如图


       deita = pow(2*k*b1 - 2*ballx - 2*k*bally, 2) - 4*(pow(k, 2) + 1)*(pow(b1,
2) + pow(ballx, 2) + pow(bally, 2) - 144);


       x1 = (-(2*k*b1 - 2*ballx - 2*k*bally) + sqrt(deita)) / (2*(pow(k, 2) +
1));
       x2 = (-(2*k*b1 - 2*ballx - 2*k*bally) - sqrt(deita)) / (2*(pow(k, 2) +
1));

       y1 = k*x1 + b1;
       y2 = k*x2 + b1;

       double y ,x;

       if( y1<y2){

        y=y1 ;x=x1;

       }

       else

         y=y2;x=x2;

       soc = moveToPos( VecPosition( x,y) ,20);

  }

  ACT->putCommandInQueue( soc ); // 放入命令队列

}
```

(40)在本方界外球的模式下，距离球最近的队友跑向球并开球，如果我不是距离球第二近的队友，那么我向球的坐标方向跑去。

```
else if ( WM->isOffsideUs( )){
 ObjectT o1 = WM->getClosestInSetTo( OBJECT_SET_TEAMMATES , OBJECT_BALL) ; //获得离球最近的对象
 if( WM->getAgentObjectType() == o1 )
 {
  if( WM->getGlobalPosition( o1) != WM->getBallPos() )
      soc = moveToPos( WM->getBallPos() ,20) ; //若距离球最近的对象o1不在球的位置
  else  {
      //在球的位置
   ObjectT o3 = WM->getClosestInSetTo( OBJECT_SET_TEAMMATES ,posAgent) ;
   soc = leadingPass( o3 , 1) ;//如果这个球员在球处传球给离球最近的队员
  }
 }
 ObjectT o2 = WM->getSecondClosestInSetTo(OBJECT_SET_TEAMMATES , OBJECT_BALL) ;
 if( WM->getAgentObjectType() != o2){
     VecPosition pos=WM->getBall();
      soc = moveToPos( pos,20) ;
 }
   ACT->putCommandInQueue( soc );
}
```

(41)在本方界外球的模式下，距离球最近的队友跑向球并开球。如果我不是距离球第二近的队友，那么求出距离球第二近的队友的坐标加上（10，5）的点；如果该点在球场内，则跑向该点，如果该点在场外，则计算距离该点最近的边界点。

```
else if(WM->isOffsideUs()){
 ObjectT o1=WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,OBJECT_BALL) ; //距离球最近
的

 ObjectT o2=WM->getSecondClosestInSetTo(OBJECT_SET_TEAMMATES,OBJECT_BALL) ; //距离
球第二近的

 double o2Posx;//距离球第二近的x坐标

 double o2posy; //距离球第二近的y坐标

 o2Posx=WM->getGlobalPosition(o2).getX();

 o2posy=WM->getGlobalPosition(o2).getY();

 if (WM->getAgentObjectType()==o1 ) {

  if(WM->getGlobalPosition(o1)!=WM->getBallPos())  {

   soc=moveToPos(WM->getBallPos(),20); //如果我不在球的位置我跑向球的位置并且开球

  }

  else {
      //我在球的位置那么我传球给距离我最近的队员

   ObjectT o3 = WM->getClosestInSetTo( OBJECT_SET_TEAMMATES , o1) ;

     soc = leadingPass(o3,1);

  }

 }

  if(WM->getAgentObjectType()!=o1&&WM->getAgentObjectType()!=o2)  {

      if(o2Posx+10<=52.5&&o2posy+5<=34)
      {
          soc=moveToPos(VecPosition(o2Posx+10,o2posy+5),20);
      }
      else
      {
          VecPosition closest=WM-
>getClosestInsetTo(VecPosition(o2Posx+10,o2posy+5));
      }



  }

 ACT->putCommandInQueue( soc );

}
```

//playOn模式下，防守行为模式，防守行为（8）

(42)在playOn模式下，5号球员与拿球的对方球员的距离始终为5。

```
if(WM->getPlayerNumber()==5) {
    VecPosition pos(WM->getBallPos().getX()-5,WM->getBallPos().getY());
    soc=moveToPos(pos,20);
}
```

(43)在playOn模式下，2号和4号一起去盯防对方拿球队员。

```
if(isBallInOurPossesion==false)//不是我方球员控球
{
    if(WM->getAgentObjectType()==OBJECT_TEAMMATE_2){
 VecPosition pos(WM->getBallPos().getX()-5,WM->getBallPos().getY());
 soc=moveToPos(pos,20);
}
if(WM->getAgentObjectType()==OBJECT_TEAMMATE_4){
 VecPosition pos(WM->getBallPos().getX(),WM->getBallPos().getY()-5)
 soc=moveToPos(pos,20);
}

}
```

(44)在playOn模式下，如果对方10号拿球，如果我是2、3、4号，则去盯防10号。

```

//球对我来说不可以踢
int num=WM->getPlayerNumber();
//如果是对方10号球员拿球说明对方10号球员离球最近
if(WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,WM-
>getBallPos())==OBJECT_OPPONENT_10&&
  (num==2||num==3||num==4)){
    //盯防球员
  VecPosition pos=WM->getMarkingPosition(OBJECT_OPPONENT_10,2.0,MARK_BALL);
  soc=moveToPos(WM->getGlobalPosition(OBJECT_OPPONENT_10),PS-
>getPlayerWhenToTurnAngle());
  ACT->putCommandInQueue( soc );
}
```

(45)在playOn模式下，如对方9号拿球，我方2、3、4号距离球最近的球员去盯防9号，其他队员盯防距离自己最近的对方球员

```
if(WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,WM-
>getBallPos())==OBJECT_OPPONENT_9){
 ObjectT closestObject=OBJECT_TEAMMATE_2;
 VecPosition team=WM->getGlobalPosition(closestObject);
 double minDis=team.getDistanceTo(WM->getGlobalPosition(OBJECT_OPPONENT_9));
 team=WM->getGlobalPosition(OBJECT_TEAMMATE_3);
 double dis=team.getDistanceTo(WM->getGlobalPosition(OBJECT_OPPONENT_9));
 if(dis<minDis) {
  minDis=dis;
  closestObject=OBJECT_TEAMMATE_3;
 }
 team=WM->getGlobalPosition(OBJECT_TEAMMATE_4);
 dis=team.getDistanceTo(WM->getGlobalPosition(OBJECT_OPPONENT_9));
 if(dis<minDis){
    closestObject=OBJECT_TEAMMATE_4;
 }
 if(WM->getAgentObjectType()==closestObject){
     //第一个参数表示想要标记的对象  第二个参数表示期望被标记对象与O也就是盯防队员之间的距离
     //第三个参数表示这种类型的盯人很难接到传球
  VecPosition pos=WM->getMarkingPosition(OBJECT_OPPONENT_9,2.0,MARK_BALL);
  soc=moveToPos(pos,PS->getPlayerWhenToTurnAngle());
 }
 else {
  ObjectT opp=WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,WM->getAgentObjectType());
  VecPosition oppPos=WM->getMarkingPosition(opp,2.0,MARK_BALL);
  soc=moveToPos(oppPos,40);
 }
}
```

(46)在playOn模式下，如果对方比我先接近球，则离球最近的队员去盯球，其他球员盯防距离自己最近的对方球员。

```
//在worldModel.cpp中添加
bool WorldModel:: isOpponentCloserToBall(){
 ObjectT opp=getFastestInSetTo(OBJECT_SET_OPPONENTS,OBJECT_BALL);
 double oppDisToBall=getGlobalPosition(opp).getDistanceTo(getBallPos());
 ObjectT teammate=getFastestInSetTo(OBJECT_SET_TEAMMATES,OBJECT_BALL);
 double teammateDisToBall=getGlobalPosition(teammate).getDistanceTo(getBallPos());
 if(oppDisToBall<teammateDisToBall)
    return true;
 else
    return false;
}
//Player.cpp
//如果对方球员更靠近球
if(WM-> isOpponentCloserToBall()){
    //获得离（对方离球最近的成员）最近的队员信息

    ObjectT teammate=WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,WM->getBallPos());
    //当我是离球最近的队员时
    if(WM->getAgentObjectType()==teammate) {
        soc=moveToPos(WM->getBallPos(),PS->getPlayerWhenToTurnAngle());
 }
 else {
  ObjectT opp=WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,WM->getAgentObjectType());
  VecPosition oppPos=WM->getMarkingPosition(opp,2.0,MARK_BALL);
  soc=moveToPos(oppPos,PS->getPlayerWhenToTurnAngle());
 }
}
```

(47)在playOn模式下，如果对方11号拿到球，则7号球员从左边去断球，8号球员从右边去断球。

```
//对方11号拿到球意味11号队员离球最近
if(WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,WM->getBallPos())==OBJECT_OPPONENT_11){
    VecPosition oppPos=WM->getGlobalPosition(OBJECT_OPPONENT_11);
 if(WM->getPlayerNumber()==7){
    VecPosition selfPos=WM->getGlobalPosition(OBJECT_TEAMMATE_7);
    double dis=selfPos.getDistanceTo(oppPos);
  if(dis>5){
      //如果和对方11号球员之间的距离大于5则去到距离4的位置断球也就是左边
   soc=moveToPos(VecPosition(oppPos.getX(),oppPos.getY(-4),40);
  }
  else{
      //否则进行断球，参数表示能否进行断球   isGoalie为true时不可断球，反之可以断球
    soc=intercept(0);
  }
 }
 else if(WM->getPlayerNumber()==8){
  VecPosition selfPos=WM->getGlobalPostion(OBJECT_TEAMMATE_8);
  double dis=selfPos.getDistanceTo(oppPos);
  if(dis>5){
   soc=moveToPos(VecPosition(oppPos.getX(),oppPos.getY()+4),40);//去往右边进行断球
  }else{
    soc=intercept(0);
  }
 }
}
```

(48)在playOn模式下，防守模式下（即球不在我方球员脚下），我方6号球员始终跟着敌方9号，我方7号始终跟着敌方10号，8号球员始终跟着敌方11号球员。

```
if(!WM->isBallInOurPossesion()){
 if(WM->getAgentObjectType()==OBJECT_TEAMMATE_6) {
      VecPosition pos=WM->getGlobalPosition(OBJECT_OPPONENT_9);
      soc=moveToPos(pos,40);
 }
 if(WM->getAgentObjectType()==OBJECT_TEAMMATE_7){
  VecPosition pos=WM->getGlobalPosition(OBJECT_OPPONENT_10);
  soc=moveToPos(pos,40);
 }
 if(WM->getAgentObjectType()==OBJECT_TEAMMATE_8){
  VecPosition pos=WM->getGlobalPosition(OBJECT_OPPONENT_11);
  soc=moveToPos(pos,40);
 }
  ACT->putCommandInQueue( soc ); // 放入命令队列
  ACT->putCommandInQueue( turnNeckToObject( OBJECT_BALL, soc ) );
}
```

(49)在playOn模式下，防守模式下（即球不在我方球员脚下），我方距离球最近的2名队员负责去盯防对方离球最近的球员，其他球员采用人球都盯的方式盯防距离自己最近的对方球员（即跑到距离自己最近的对方球员的可能接球路线的前面位置点）

```
if(!WM->isBallInOurPossesion()){
         //获得离球最近的对方球员对象
         ObjectT closestOpp = WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,
OBJECT_BALL);
         //获得我方离球最近和第二近的球员对象
         ObjectT closestTeam = WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,
OBJECT_BALL);
         ObjectT secondClosestTeam = WM-
>getSecondClosestInSetTo(OBJECT_SET_TEAMMATES, OBJECT_BALL);
         //获得一个球员对象
         ObjectT agent = WM->getAgentObjectType();
         if(agent==closestTeam || agent==secondClosestTeam){
          //如果是距离球最近的两个队员，盯防距离球最近的队员                参数可以直接拦截
传球路线
          VecPosition oppPos=WM-
>getMarkingPosition(closestOpp,2.0,MARK_BISECTOR);
          soc=moveToPos(oppPos,PS->getPlayerWhenToTurnAngle());
        }
         else{
          //剩下的球员去盯防离自己最近的球员
          ObjectT opp=WM->getClosestInSetTo(OBJECT_SET_OPPONENTS,WM-
>getAgentObjectType());
          VecPosition oppPos=WM->getMarkingPosition(opp,2.0,MARK_BALL);
          soc=moveToPos(oppPos,40);
        }
      }
```

//其他比赛模式下，防守行为其他比赛模式下，防守行为（7）

(50) 在对方界外球的模式下，如果我是 4 号，那么我就跑向距离球最近的对方球员处。

```
else if(WM->isOffSideThem()){
  if(WM->getPlayerNumber()==4){
      ObjectT opp=WM->getCloesestInsetTo(OBJECT_SET_OPPONENTS,OBJECT_BALL);
      VecPosition pos=WM->getGlobalPosition(opp);
      soc=moveToPos(pos,PS->getPlayerWhenToTurnAngle())
  }
  ACT->putCommandInQueue( soc );
 }
```

(51) 在对方界外球的模式下，如果我是距离球第二近的队员的话，那么我跑向距离球第二近的对方球员的位置。

```
else if(WM->isOffSideThem()){
  if(WM->getSecondClosestInSetTo(OBJECT_SET_TEAMMATES,OBJECT_BALL)==WM-
>getAgentObjectType()){
      ObjectT opp=WM->getSecondClosestInSetTo(OBJECT_SET_OPPONENTS,OBJECT_BALL);
      VecPosition pos=WM->getGlobalPosition(opp);
      soc=moveToPos(pos,PS->getPlayerWhenToTurnAngle());
  }
  ACT->putCommandInQueue( soc );
}
```

(52)在对方角球模式情况下，如果球在我方左侧，则2，3，6，7号平均分布在以球的位置为圆心，半径10米的圆弧上（其中2号的初始位置可以自行设定，间距也可以自行设定，但是需要保证2，3，6，7都在场内）；如果球在我方右侧，则4，5，7，8号平均分布在以球的位置为圆心，半径10米的圆弧上（其中4号的初始位置可以自行设定，间距也可以自行设定，但是需要保证4，5，7，8都在场内）。

```
if(WM->isCornerKickThem()){
             const double Pi = 3.1415926;
             int num = WM->getPlayerNumber();
             if(WM->getBallPos().getY()<0){
                if(num == 2){
                    soc = moveToPos(VecPosition(-42.5,34.0), 40);//移动到距离球门10
米的边界线上
                }
                else if(num == 3){
                    //由于球的位置为(-52.5,34)
                    soc = moveToPos(VecPosition(-52.5+10*cos(Pi/6),34.0-
10*sin(Pi/6)), 40);
                }
                else if(num == 6){
                    soc =
moveToPos(VecPosition(-52.5+10*cos(2*Pi/6),34.010*sin(2*Pi/6)), 40);
                }
                else if(num == 7){
                    soc = moveToPos(VecPosition(-52.5+10*cos(3*Pi/6),34.0-
10*sin(3*Pi/6)), 40);
                }
             }
             else if(WM->getBallPos().getY()>0){
              if(num == 4){
               soc = moveToPos(VecPosition(-42.5,-34.0), 40);
              }
              else if(num == 5){
               soc =
moveToPos(VecPosition(-52.5+10*cos(Pi/6),-34.0+10*sin(Pi/6)), 40);
              }
              else if(num == 7){
               soc =
moveToPos(VecPosition(-52.5+10*cos(2*Pi/6),-34.0+10*sin(2*Pi/6)), 40);
              }
              else if(num == 8){
               soc =
moveToPos(VecPosition(-52.5+10*cos(3*Pi/6),-34.0+10*sin(3*Pi/6)), 40);
              }
             }
             ACT->putCommandInQueue( soc ); // 放入命令队列
             ACT->putCommandInQueue ( turnBodyToObject ( OBJECT_BALL ) );
            //ACT->putCommandInQueue( turnNeckToObject( OBJECT_BALL, soc ) ); //
把脖子转向球，也就是一直看着球
            }
```

(53)在对方界外球的模式下，如果我不是距离球最近的队友，并且我的 x 轴坐标大于0的话，那么我跑向自身位置加（-10,0）的点的位置去。

```
else if(WM->isOffSideThem())
{
if(WM->getClosestInSetTo(OBJECT_SET_TEAMMATES,OBJECT_BALL)!=WM-
>getAgentObjectType()
         && WM->getAgentGlobalPosition().getX()>0){
      soc=moveToPos(VecPosition(-10,0)+WM->getAgentGlobalPosition(),20);
  }
  ACT->putCommandInQueue( soc );
  }
```

(54)在对方界外球的模式下，如果我身边 5 米内有对方球员的话，那么我跑向身边对方球员的位置，并且我的x轴坐标的值要比该对方的大2。

```
else if ( WM->isOffsideThem() ){
   Circle cir ( posAgent , 5 ) ;
   int num = 0 ;
   num = WM->getNrInSetInCircle( OBJECT_SET_OPPONENTS , cir ) ;
   if ( num != 0 ){
      ObjectT o = WM->getClosestInSetTo( OBJECT_SET_OPPONENTS , posAgent) ;
      VecPosition poso = WM->getGlobalPosition(o ) ;
      VecPosition pos( poso.getX() + 2 , poso.getY() ) ;
      soc = moveToPos( pos , 20 ) ;
   }
   ACT->putCommandInQueue( soc );
}
```

(55)在对方界外球的模式下，如果球的位置的 x 轴小于 0，如果我是 4 号或 5号的话，一起跑向距离球第二近的对方球员的位置。

```
else if ( WM->isOffsideThem()){
 if (WM->getBallPos().getX()<0&&( WM -> getPlayerNumber()==4 || WM-
>getPlayerNumber()==5 )){
      ObjectT o = WM -> getSecondClosestInSetTo( OBJECT_SET_OPPONENTS
,OBJECT_BALL ) ;
      VecPosition  posopptwo = WM ->getGlobalPosition( o ) ;
      soc = moveToPos( posopptwo , 20 ) ;
      ACT->putCommandInQueue( soc );
 }
}
```

(56)在对方角球模式情况下，如果球在我方左侧，则2，3号负责盯防对方距离球第二近的对方球员；如果球在我方右侧，则4，5号负责盯防对方距离球第二近的对方球员。

```
if(WM->isCornerKickThem()){
             ObjectT opp=WM-
>getSecondClosestInSetTo(OBJECT_SET_OPPONENTS,OBJECT_BALL);
             int num = WM->getPlayerNumber();
             if(WM->getBallPos().getY()<0 && (num == 2 || num == 3)){
              VecPosition oppPos=WM->getMarkingPosition(opp,2.0,MARK_BALL);
              soc=moveToPos(oppPos,40);
             }
             else if(WM->getBallPos().getY()>0 && (num == 4 || num == 5)){
              VecPosition oppPos=WM->getMarkingPosition(opp,2.0,MARK_BALL);
              soc=moveToPos(oppPos,40);
             }
             ACT->putCommandInQueue( soc ); // 放入命令队列
             ACT->putCommandInQueue ( turnBodyToObject ( OBJECT_BALL ) );
            //ACT->putCommandInQueue( turnNeckToObject( OBJECT_BALL, soc ) ); //
把脖子转向球，也就是一直看着球
}
```