PlayerTeams.cpp

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*exam questions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*---------------------------start-------------------------------\*/

/\*1.在 playOn 模式下，拿到球以后朝前方快速带球\*/

if( WM->isBallKickable() )

{

AngDeg ang = 0.0;

soc = dribble( ang, DRIBBLE\_FAST );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*2.在 PlayOn 模式下，拿到球以后朝球门方向慢速带球。\*/

if( WM->isBallKickable() )

{

VecPosition vec = VecPosition(52.5, 0) - WM->getAgentGlobalPosition();//我的位置到球门位置的向量

AngDeg ang = vec.getDirection(); //获得角度

soc = dribble( ang, DRIBBLE\_SLOW );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*3.在 playOn 模式下，拿到球以后把球围绕自己身体逆时针转。\*/

if( WM->isBallKickable() )

{

soc = kickBallCloseToBody(-180);//逆时针旋转角度为负

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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，如对手仔左侧，把球踢到右侧）。\*/

if( WM->isBallKickable() )

{

Circle cir( posAgent, 7.0 ); //建立以自身位置 为圆心，7 为半径的圆

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir ); //获得圆内 对方球员的个数

if( oppNum >= 1 )

{

ObjectT opp = WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS, posAgent);//获得 离我最近的对方球员

VecPosition oppPos = WM->getGlobalPosition(opp); //获得他的 位置

VecPosition meOpp = oppPos - posAgent; // 我与他连线的向量

AngDeg ang = meOpp.getDirection();

if( ang >= 0 )

{

ang -= 60;

}

else

{

ang += 60;

}

soc = dribble( ang, DRIBBLE\_ILLEGAL );

}

else

{

soc = dribble( 0.0, DRIBBLE\_FAST );//否则向前快速 带球，可写可不写

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

if( WM->isBallKickable() )

{

Circle cir( posAgent, 7.0 );

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

if( oppNum >= 1)

{

ObjectT teammate = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, posAgent );//获得离我最 近的队友

VecPosition vec = WM->getGlobalPosition( teammate ) - posAgent;

soc = kickTo( vec, PS->getPassEndSpeed() );

}

else

{

soc = dribble( (VecPosition(52.5, 0) - posAgent).getDirection(), DRIBBLE\_FAST );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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；否则向球门方向快速带球。\*/

if( WM->isBallKickable() )

{

Circle cir( posAgent, 7.0 );

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

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

if( oppNum >= 1)

{

soc = dribble( ang, DRIBBLE\_ILLEGAL );

}

else

{

soc = dribble( ang, DRIBBLE\_FAST );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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（周期数为偶数，球门的右侧射门，

周期为奇数，向球门的左侧射门）。\*/

if( WM->isBallKickable() )

{

if( WM->isOpponentAtAngle(-30.0, 30.0) )//如果 我前方60度范围内没有 对方球员

{

VecPosition posGoal( PITCH\_LENGTH/2.0,

(-1 + 2\*(WM->getCurrentCycle()%2)) \* 0.4 \* SS->getGoalWidth() );//根据当前的周期数 交替的 踢向球门的 两个死角

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

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

}

/\*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->isBallKickable() )

{

if( posBall.getX() < 0)// 球在我方半场

{

soc = kickTo( VecPosition(0.0, 0.0), PS->getPassEndSpeed() );

}

else

{

soc = dribble( (VecPosition(52.5, 0) - posAgent).getDirection(), DRIBBLE\_FAST );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

if( WM->isBallKickable() )

{

ObjectT teammate = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, posAgent );

Circle cir( WM->getGlobalPosition(teammate), 5.0 );

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

if( oppNum <= 0 )

{

soc = leadingPass( teammate, PS->getPassEndSpeed() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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（判断队友身边 5 米范围是否有对方防守队员）。\*/

if( WM->isBallKickable() )

{

ObjectT teammate = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, posAgent );

VecPosition teammatePos = WM->getGlobalPosition( teammate );

Circle cir( teammatePos, 5.0 );

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

AngDeg ang = ( teammatePos - posAgent ).getDirection();

if( oppNum<=0 && (ang>=-90 && ang<=90) )//判断 该 队友是否有人 防守 是否 在我 前方

{

soc = leadingPass( teammate, PS->getPassEndSpeed() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*11.在 playOn 模式下，拿到球后以后，在本方半场传给次近的队友；在对方半场，非 10 号球员传球给 10 号球员，

10 号球员则快速向球门方向带球。\*/

if( WM->isBallKickable() )

{

if( posBall.getX() < 0 )

{

soc = leadingPass( WM->getSecondClosestInSetTo( OBJECT\_SET\_TEAMMATES, WM->getAgentObjectType() ), PS->getPassEndSpeed() );

}

else

{

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_10 )//如果 我是 10号

soc = dribble( VecPosition(52.5, 0).getDirection(), DRIBBLE\_FAST );

else

soc = leadingPass( OBJECT\_TEAMMATE\_10, PS->getPassEndSpeed() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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->isBallKickable() )

{

if( posBall.getX() < 0)

soc = dribble( 0.0, DRIBBLE\_SLOW );

else

soc = dribble( (VecPosition(52.5, 0)- posAgent).getDirection(), DRIBBLE\_FAST );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

if( WM->isBallKickable() )

{

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_2 )

{

soc = kickTo( VecPosition( posBall.getX(), -34 ), PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

}

else if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_5 )

{

soc = kickTo( VecPosition( posBall.getX(), 34 ), PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

else

{

soc = dribble( 0.0, DRIBBLE\_ILLEGAL );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

}

/\*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大的球门一侧。\*/

if( WM->isBallKickable() )

{

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_4 )

soc = leadingPass( OBJECT\_TEAMMATE\_7, PS->getPassEndSpeed() );

else

soc = leadingPass( WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, posAgent), PS->getPassEndSpeed() );

if( WM->isInTheirPenaltyArea( posBall ))//如果在 对方禁区

{

VecPosition posGoalie = WM->getGlobalPosition( OBJECT\_OPPONENT\_GOALIE );//对方守门员位置

AngDeg meGoalie = (posGoalie - posAgent).getDirection();

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

AngDeg angBottom = ( VecPosition(52.5 -6.0) - posAgent ).getDirection();

// 由于编译器不支持我使用fabs()函数，所以这里采用这种形式求绝对值

AngDeg ang1 = (angTop-meGoalie)>=0 ? (angTop-meGoalie) : (-1)\*(angTop-meGoalie);

AngDeg ang2 = (angBottom-meGoalie)>=0 ? (angBottom-meGoalie) : (-1)\*(angBottom-meGoalie);

if( ang1 >= ang2 )

soc = kickTo( VecPosition(52.5, 6.0) - posAgent, SS->getBallSpeedMax() );

else

soc = kickTo( VecPosition(52.5, -6.0) - posAgent, SS->getBallSpeedMax() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

}

/\*15.在 playOn 模式下，拿球后垂直带球。\*/

if( WM->isBallKickable() )

{

// 沿着y轴 跑

if( posBall.getY() <= 0)

soc = dribble( -90, DRIBBLE\_SLOW );

else

soc = dribble( 90, DRIBBLE\_SLOW );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

if( WM->isBallKickable() )

{

soc = dribble( (VecPosition(0,0)-posAgent).getDirection(), DRIBBLE\_WITHBALL );//拿到球后带球到达球场中心

ObjectT teammate = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, posAgent );

if( posAgent.getDistanceTo( teammate ) <= 2.0)

soc = leadingPass( teammate, PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

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

{

soc = dribble( 0.0, DRIBBLE\_WITHBALL );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

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

{

VecPosition pos( posBall.getX() - 5, posBall.getY() ); //保持五米

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

ACT->putCommandInQueue( soc );

}

/\*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->getPlayerNumber()==10 && WM->isBallKickable() && posAgent.getX()>30 )

{

VecPosition posGoalie = WM->getGlobalPosition( OBJECT\_OPPONENT\_GOALIE );

AngDeg meGoalie = (posGoalie - posAgent).getDirection();

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

AngDeg angBottom = ( VecPosition(52.5 -6.0) - posAgent ).getDirection();

// 由于编译器不支持我使用fabs()函数，所以这里采用这种形式求绝对值

//AngDeg ang1 = (angTop-meGoalie)>=0 ? (angTop-meGoalie) : (-1)\*(angTop-meGoalie);

//AngDeg ang2 = (angBottom-meGoalie)>=0 ? (angBottom-meGoalie) : (-1)\*(angBottom-meGoalie);

if( fabs(angTop-meGoalie) >= fabs(angBottom-meGoalie) )

soc = kickTo( VecPosition(52.5, 6.0) - posAgent, SS->getBallSpeedMax() );

else

soc = kickTo( VecPosition(52.5, -6.0) - posAgent, SS->getBallSpeedMax() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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\*/

if( WM->getPlayerNumber()==9 && WM->isBallKickable() )

{

soc = dribble( (VecPosition(52.5, 0)-posAgent).getDirection(), DRIBBLE\_FAST );//9号快速带球 到球门 处

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_10 )

soc = dashToPoint( VecPosition(52.5, 0), WM->getCurrentCycle() );//10号跑向球门

if( WM->isInTheirPenaltyArea( posBall ) )

{

soc = leadingPass( OBJECT\_TEAMMATE\_10, PS->getPassEndSpeed() );//传球给 10号

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_10 )

soc = kickTo( (VecPosition(52.5, 0) - posAgent).getDirection(), SS->getBallSpeedMax() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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->getPlayerNumber()==4 && WM->isBallKickable() )

{

soc = leadingPass( OBJECT\_TEAMMATE\_7, PS->getPassEndSpeed() );

if( WM->getAgentObjectType()==OBJECT\_TEAMMATE\_7 && WM->isBallKickable() )//7号拿球 可踢

soc = leadingPass( OBJECT\_TEAMMATE\_9, PS->getPassEndSpeed() ); // 传给9号

if( WM->getAgentObjectType()==OBJECT\_TEAMMATE\_9 && WM->isBallKickable() )//9号拿球 可踢

{

if( WM->getRelativeDistance( WM->getOppGoalieType() ) >= 20 )//距离对方球门较远时

soc = dribble( (VecPosition(52.5, 0) - posAgent).getDirection(), DRIBBLE\_FAST );//快速向球门带球

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

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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\*/

if( WM->isBallKickable() && posBall.getX()<0 )

soc = kickTo( VecPosition( 0, posBall.getY() ), SS->getBallSpeedMax() );

else if( WM->isBallKickable() && posBall.getX()>0)

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

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

//拿球以后，需要自己添加 WorldModel 函数（12）

/\*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 = leadingPass( WM->getTeammateClosestToGoalAndAgent(), PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

if( WM->isBallKickable() )

{

ObjectT o = WM->getTeammateAtAngleDist(-30, 30, 20.0);

if( o != OBJECT\_UNKNOWN )

soc = leadingPass( o, PS->getPassEndSpeed() );

else

soc = dribble( (VecPosition(52.5, 0) - posAgent).getDirection(), DRIBBLE\_ILLEGAL );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

if( WM->isBallKickable() )

{

ObjectT maxX;

if( WM->getOpponentNumSideY( -1 ) > WM->getOpponentNumSideY( 1 ))// 我方半场 对方球员 较多

maxX = WM->getXMaxTeammateBySide( 1 );

else

maxX = WM->getXMaxTeammateBySide( -1 );

soc = leadingPass( maxX, PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, 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。\*/

if( WM->isBallKickable() )

{

Circle cir( posAgent, 7.0 );

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

if( oppNum >= 2)

{

ObjectT mate = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, VecPosition( 52.0, 0) );// x 最大的队友 离对方 球门最近

soc = leadingPass( mate, PS->getPassEndSpeed() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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->getPlayerNumber()==10 && WM->isBallInOurPossesion() )

soc = moveToPos( VecPosition(WM->getOffsideX()-2, posAgent.getY() ), 30);

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

/\*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。\*/

Line ballLine = Line::makeLineFromPositionAndAngle( WM->getBallPos(), WM->getBallDirection() );// 根据球的位置和方向 角 创建直线

if( ballLine.getDistanceWithPoint( posAgent ) < 4)

soc = moveToPos( ballLine.getPointOnLineClosestTo( posAgent ), 30);

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

/\*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。\*/

if( WM->isBallKickable() )

{

Circle cir( posAgent, 7.0 );

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

if( oppNum <= 0 )

soc = dribble( (VecPosition(53.0, 0)-posAgent).getDirection(), DRIBBLE\_FAST );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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() && posAgent.getX()<30 && WM->getPlayerNumber()==10 )

{

soc = moveToPos( (VecPosition(0, 10)+posAgent), 30);

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

if( WM->isBallKickable() )

{

Circle cir( posAgent, 5.0 );

int oppNum = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

if( oppNum>=2 && WM->isTeammateAtAngle(-30, 30) )

soc = leadingPass( WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, posAgent), PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

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

if( WM->isBallKickable() )

{

if( !WM->isOpponentAtAngle( -30, 30 ) )

soc = dribble( (VecPosition(52.5, 0)-posAgent).getDirection(), DRIBBLE\_FAST );

double dist = WM->getAgentDistFromGoalie();

if( dist>=5 && dist<=10 )

{

VecPosition posGoalie = WM->getGlobalPosition( OBJECT\_OPPONENT\_GOALIE );//对方守门员位置

AngDeg meGoalie = (posGoalie - posAgent).getDirection();

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

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

// 由于编译器不支持我使用fabs()函数，所以这里采用这种形式求绝对值

//AngDeg ang1 = (angTop-meGoalie)>=0 ? (angTop-meGoalie) : (-1)\*(angTop-meGoalie);

//AngDeg ang2 = (angBottom-meGoalie)>=0 ? (angBottom-meGoalie) : (-1)\*(angBottom-meGoalie);

if( fabs(angTop-meGoalie) >= fabs(angBottom-meGoalie) )

soc = kickTo( VecPosition(52.5, 6.0) - posAgent, SS->getBallSpeedMax() );

else

soc = kickTo( VecPosition(52.5 -6.0) - posAgent, SS->getBallSpeedMax() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*32.在 playon 的模式下，发现前方没有对方球员的时候，则快速向球门放向带球，距离守门员 5-10 米时向空隙较大的一侧射门。\*/

if(WM->isBallKickable())

{

if(!WM->isOpponentAtAngle(-30, 30))

soc = dirbble((VecPosition(52.5, 0)-posAgent).getDirection(), ERIBBLE\_FAST);

double dist = WM->getAgentDistFormGoalie();

if( dist>=5 && dist<=10 )

{

VecPosition posGoalie = WM->getGlobalPosition( OBJECT\_OPPONENT\_GOALIE );//对方守门员位置

AngDeg meGoalie = (posGoalie - posAgent).getDirection();

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

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

// 由于编译器不支持我使用fabs()函数，所以这里采用这种形式求绝对值

//AngDeg ang1 = (angTop-meGoalie)>=0 ? (angTop-meGoalie) : (-1)\*(angTop-meGoalie);

//AngDeg ang2 = (angBottom-meGoalie)>=0 ? (angBottom-meGoalie) : (-1)\*(angBottom-meGoalie);

if( fabs(angTop-meGoalie) >= fabs(angBottom-meGoalie) )

soc = kickTo( VecPosition(52.5, 6.0) - posAgent, SS->getBallSpeedMax() );

else

soc = kickTo( VecPosition(52.5 -6.0) - posAgent, SS->getBallSpeedMax() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

}

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

int playNum = WM->getPlayerNumber();

if( (playNum==8 || playNum==9 || playNum==10 || playNum==11) && WM->isOpponentAtAngle( -30, 30 ) )

{

soc = dribble( (VecPosition(52.5, 0)-posAgent).getDirection(), DRIBBLE\_FAST );

double dist = WM->getAgentDistFromGoalie();

if( dist>=5 && dist<=10 )

{

VecPosition posGoalie = WM->getGlobalPosition( OBJECT\_OPPONENT\_GOALIE );//对方守门员位置

AngDeg meGoalie = (posGoalie - posAgent).getDirection();

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

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

// 由于编译器不支持我使用fabs()函数，所以这里采用这种形式求绝对值

//AngDeg ang1 = (angTop-meGoalie)>=0 ? (angTop-meGoalie) : (-1)\*(angTop-meGoalie);

//AngDeg ang2 = (angBottom-meGoalie)>=0 ? (angBottom-meGoalie) : (-1)\*(angBottom-meGoalie);

if( fabs(angTop-meGoalie) >= fabs(angBottom-meGoalie) )

soc = kickTo( VecPosition(52.5, 6.0) - posAgent, SS->getBallSpeedMax() );

else

soc = kickTo( VecPosition(52.5 -6.0) - posAgent, SS->getBallSpeedMax() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

//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.\*/

if( WM->isCornerKickUs() )

{

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_10 )

{

if( WM->isBallKickable() )

soc = leadingPass( OBJECT\_TEAMMATE\_9, PS->getPassEndSpeed() );

else

soc = moveToPos( posBall, 30);

}

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_9 )

{

VecPosition pos;

pos.setX( posBall.getX() );

if( posBall.getY() > 0 )

pos.setY( posBall.getY() - 10 );

else

pos.setY( posBall.getY() + 10 );

soc = moveToPos( pos, 30 );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*35.在本方边线球模式下，如果自己是距离球最近的队员，跑向球；并开球（球在自己脚下则把球传给最近的队友）\*/

if( WM->isOffsideUs() )

{

if( WM->getFastestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL ) == WM->getAgentObjectType() )

soc = moveToPos( posBall, PS->getPlayerWhenToTurnAngle() );

if( WM->isBallKickable() )

soc = leadingPass( WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, WM->getAgentObjectType() ), PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*36.在本方边线球模式下，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。\*/

if( WM->isOffsideUs() )

{

if( posBall.getX()<0 && posBall.getY()<0 && WM->getAgentObjectType()==OBJECT\_TEAMMATE\_2 )

soc = moveToPos( posBall, PS->getPlayerWhenToTurnAngle() );

else if( posBall.getX()<0 && posBall.getY()>0 && WM->getAgentObjectType()==OBJECT\_TEAMMATE\_5 )

soc = moveToPos( posBall, PS->getPlayerWhenToTurnAngle() );

if( WM->isBallKickable() && (WM->getAgentObjectType()==OBJECT\_TEAMMATE\_2 || WM->getAgentObjectType()==OBJECT\_TEAMMATE\_5) )

soc = leadingPass( WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, WM->getAgentObjectType() ), PS->getPassEndSpeed() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*37.在本方边线球的模式下，如果我是距离第二近的队友的话，那么我也跑朝着球的位置跑，直到距离球 7 米的范围。\*/

if( WM->isOffsideUs() )

{

double dist = sqrt((posAgent.getX()-posBall.getX())\*(posAgent.getX()-posBall.getX()) + (posAgent.getY()-posBall.getY())\*(posAgent.getY()-posBall.getY()));

if( WM->getSecondClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL)==WM->getAgentObjectType() && dist>=7 )

{

VecPosition pos = posBall - posAgent;

double d = posBall.getDistanceTo( posAgent );

VecPosition target( (pos.getX()/d)\*7.0, (pos.getY()/d)\*7.0 );

soc = moveToPos( posAgent+target, PS->getPlayerWhenToTurnAngle() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*38.在本方边线球的模式下，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。\*/

if( WM->isOffsideUs() )

{

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

{

if( WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL ) != WM->getAgentObjectType() )

soc = moveToPos( posBall+VecPosition( 5, 0 ), PS->getPlayerWhenToTurnAngle() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

}

/\*39.在本方边线球的模式下，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（53,0）最近的点。\*/

if( WM->isOffsideUs() )

{

if( WM->getSecondClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL ) == WM->getAgentObjectType() )

{

VecPosition posGoal( 53, 0 );

double dist = posBall.getDistanceTo( posGoal );// 球和 球门间的 距离

if( dist > 12 )

{

VecPosition pos = posGoal - posBall;

VecPosition target( (pos.getX()/dist)\*12, (pos.getY()/dist)\*12 );

soc = moveToPos( posBall+target, PS->getPlayerWhenToTurnAngle() );// 球员在 球与球门 连线上， 距离球12

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

}

}

/\*40.在本方边线球的模式下，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，那么我向自身向量（5,0）的方向跑去。\*/

if( WM->isOffsideUs() )

{

ObjectT nearestMate = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL );

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

soc = moveToPos( posBall, PS->getPlayerWhenToTurnAngle() );

if( WM->getSecondClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL ) != WM->getAgentObjectType() )

soc = moveToPos( posAgent+VecPosition( 5, 0 ), PS->getPlayerWhenToTurnAngle() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*41.在本方边线球的模式下，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。\*/

if( WM->isOffsideUs() )

{

if( WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL ) == WM->getAgentObjectType() )

soc = moveToPos( posBall, PS->getPlayerWhenToTurnAngle() );

ObjectT secNear = WM->getSecondClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL );

if( secNear != WM->getAgentObjectType() )

{

VecPosition pos = WM->getGlobalPosition( secNear ) + VecPosition( 10, 5 );

if( WM->isInField( pos ) )

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

else

{

ObjectT o = WM->getClosestFlagToPos( pos );

soc = moveToPos( WM->getGlobalPosition( o ), PS ->getPlayerWhenToTurnAngle() );

}

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

//playOn 模式下，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)

{

ObjectT oppBallKeeper = WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

VecPosition posOpp = WM->getGlobalPosition( oppBallKeeper );

VecPosition pos = posAgent - posOpp;

double d = posOpp.getDistanceTo( posAgent );

VecPosition target( (pos.getX()/d)\*5, (pos.getY()/d)\*5 );

soc = moveToPos( posOpp+target, PS->getPlayerWhenToTurnAngle() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*43.在 playOn 模式下，2 号和 4 号一起去盯防对方拿球队员。\*/

if( WM->getPlayerNumber()==2 || WM->getPlayerNumber()==4)

{

ObjectT oppBallKeeper = WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

soc = mark( oppBallKeeper, 0.0, MARK\_BALL );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

ObjectT oppBallKeeper = WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

if( oppBallKeeper == OBJECT\_OPPONENT\_10 )

{

if( WM->getAgentObjectType()==OBJECT\_TEAMMATE\_2

|| WM->getAgentObjectType()==OBJECT\_TEAMMATE\_3

|| WM->getAgentObjectType()==OBJECT\_TEAMMATE\_4 )

{

soc = mark( oppBallKeeper, 0.0, MARK\_BISECTOR );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, 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\*/

ObjectT oppBallKeeper = WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

if( oppBallKeeper == OBJECT\_OPPONENT\_9 )

{

ObjectT nearestBall = WM->getNearestBallObj( OBJECT\_TEAMMATE\_2, WM->getNearestBallObj( OBJECT\_TEAMMATE\_3, OBJECT\_TEAMMATE\_4 ) );

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

soc = mark( oppBallKeeper, 0.0, MARK\_BISECTOR );

else

{

ObjectT nearestOpp = WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, WM->getAgentObjectType() );

soc = mark( nearestOpp, 0.0, MARK\_ILLEGAL );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

double \*dist1;

double \*dist2;

WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL, dist1 );

WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL, dist2 );

if( \*dist1 < \*dist2 )

{

ObjectT nearestBall = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL );

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

soc = mark( nearestBall, 0.0, MARK\_BISECTOR );

else

soc = mark( WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, WM->getAgentObjectType() ), 5.0, MARK\_ILLEGAL );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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。\*/

//这个所谓的左右，你就当自己是 站在球场上的球员，来判断左右。 注意 球场上y轴正方向 朝下

if( !WM->isBallKickable() && WM->getAgentObjectType()==OBJECT\_OPPONENT\_11 )

{

if( WM->getPlayerNumber() == 7)

{

soc = moveToPos( WM->getGlobalPosition( OBJECT\_OPPONENT\_11 )+VecPosition( 0, -5 ), PS->getPlayerWhenToTurnAngle() );

soc = intercept( true );

}

else if( WM->getPlayerNumber() == 8 )

{

soc = moveToPos( WM->getGlobalPosition( OBJECT\_OPPONENT\_11 )+VecPosition( 0, 5 ), PS->getPlayerWhenToTurnAngle() );

soc = intercept( true );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*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->getPlayerNumber() == 6 )

soc = moveToPos( WM->getGlobalPosition( OBJECT\_OPPONENT\_9 ), PS->getPlayerWhenToTurnAngle() );

else if( WM->getPlayerNumber() == 7 )

soc = moveToPos( WM->getGlobalPosition( OBJECT\_OPPONENT\_10 ), PS->getPlayerWhenToTurnAngle() );

else if( WM->getPlayerNumber() == 8 )

soc = moveToPos( WM->getGlobalPosition( OBJECT\_OPPONENT\_11 ), PS->getPlayerWhenToTurnAngle() );

ACT->putCommandInQueue( soc );

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，

其他球员其他球员采用人 球都盯的方式盯防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() )

{

ObjectT teammate1 = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL );

ObjectT teammate2 = WM->getSecondClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL );

ObjectT opp = WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

//ObjectT opp2 = WM->getSecondClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

if( WM->getAgentObjectType()==teammate1 || WM->getAgentObjectType()==teammate2 )

soc = mark( opp, 0.0, MARK\_ILLEGAL );

else

{

ObjectT nearestOpp = WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, WM->getAgentObjectType() );

VecPosition posOpp = WM->getGlobalPosition( nearestOpp );

VecPosition oppToBall = posBall - posOpp;

VecPosition target( oppToBall.getX()/2.0, oppToBall.getY()/2.0 );

soc = moveToPos( posAgent+target, PS->getPlayerWhenToTurnAngle() );// 跑到对方球员 与球连线的 中点

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

//其他比赛模式下，防守行为（7）

/\*50.在对方边线球的模式下，如果我是 4 号，那么我就跑向距离球最近的对方球员处。\*/

if( WM->isOffsideThem() )

{

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

soc = moveToPos( WM->getClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL ), PS->getPlayerWhenToTurnAngle() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*51.在对方边线球的模式下，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。\*/

if( WM->isOffsideThem() )

{

ObjectT o = WM->getSecondClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL );

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

{

ObjectT opp = WM->getSecondClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

soc = moveToPos( WM->getGlobalPosition( opp ), PS->getPlayerWhenToTurnAngle() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, 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() )

{

if( posBall.getY() < 0 )

{

VecPosition pos2 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_2 );

VecPosition pos3 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_3 );

VecPosition pos6 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_6 );

VecPosition pos7 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_7 );

if( WM->isInField( pos2 ) && WM->isInField( pos3 ) && WM->isInField( pos6 ) && WM->isInField( pos7 ))

{

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_2 )

soc = moveToPos( posBall+VecPosition( 10, 0 ), PS->getPlayerWhenToTurnAngle() );

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_3 )

soc = moveToPos( posBall+VecPosition( -10, 0 ), PS->getPlayerWhenToTurnAngle() );

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_6 )

soc = moveToPos( posBall+VecPosition( 0, 10 ), PS->getPlayerWhenToTurnAngle() );

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_7 )

soc = moveToPos( posBall+VecPosition( 0, -10 ), PS->getPlayerWhenToTurnAngle() );

}

}

else

{

VecPosition pos4 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_4 );

VecPosition pos5 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_5 );

VecPosition pos7 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_7 );

VecPosition pos8 = WM->getGlobalPosition( OBJECT\_TEAMMATE\_8 );

if( WM->isInField( pos4 ) && WM->isInField( pos5 ) && WM->isInField( pos7 ) && WM->isInField( pos8 ))

{

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_4 )

soc = moveToPos( posBall+VecPosition( 10, 0 ), PS->getPlayerWhenToTurnAngle() );

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_5 )

soc = moveToPos( posBall+VecPosition( -10, 0 ), PS->getPlayerWhenToTurnAngle() );

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_7 )

soc = moveToPos( posBall+VecPosition( 0, 10 ), PS->getPlayerWhenToTurnAngle() );

if( WM->getAgentObjectType() == OBJECT\_TEAMMATE\_8 )

soc = moveToPos( posBall+VecPosition( 0, -10 ), PS->getPlayerWhenToTurnAngle() );

}

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*53.在对方边线球的模式下，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。\*/

if( WM->isOffsideThem() )

{

ObjectT o = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL );

ObjectT agent = WM->getAgentObjectType();

if( agent!=o && WM->getGlobalPosition( agent ).getX()>0 )

soc = moveToPos( posAgent+VecPosition( -10, 0 ), PS->getPlayerWhenToTurnAngle() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*54.在对方边线球的模式下，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。\*/

if( WM->isOffsideThem() )

{

Circle cir( posAgent, 5 );

int num = WM->getNrInSetInCircle( OBJECT\_SET\_OPPONENTS, cir );

if( num > 0 )

soc = moveToPos( WM->getGlobalPosition( WM->getOppByDistAndPos( 5, 2, 0 )), PS->getPlayerWhenToTurnAngle() );

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*55.在对方边线球的模式下，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。\*/

if( WM->isOffsideThem() )

{

if( posBall.getX() < 0 )

{

ObjectT agent = WM->getAgentObjectType();

if( agent==OBJECT\_TEAMMATE\_4 || agent==OBJECT\_TEAMMATE\_5 )

{

ObjectT opp = WM->getSecondClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

soc = moveToPos( WM->getGlobalPosition( opp ), PS->getPlayerWhenToTurnAngle() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, 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 agent = WM->getAgentObjectType();

ObjectT oppSecNearBall = WM->getSecondClosestInSetTo( OBJECT\_SET\_OPPONENTS, OBJECT\_BALL );

if( posBall.getY() < 0)

{

if( agent==OBJECT\_TEAMMATE\_2 || agent==OBJECT\_TEAMMATE\_3 )

soc = moveToPos( WM->getGlobalPosition( oppSecNearBall ), PS->getPlayerWhenToTurnAngle() );

}

else

{

if( agent==OBJECT\_TEAMMATE\_4 || agent==OBJECT\_TEAMMATE\_5 )

soc = moveToPos( WM->getGlobalPosition( oppSecNearBall ), PS->getPlayerWhenToTurnAngle() );

}

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

/\*-----------------------------end---------------------------------\*/