(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())

{

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

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(-120);

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);

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

AngDeg ang=0;

if(num>0)

{

ObjectT o=WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS,posAgent);

VecPosition p=WM->getGlobalPosition(o);

if((p-posAgent).getDirection()>=0)

{

ang+=45;

soc=dribble(ang,DRIBBLE\_WITHBALL);

}

else{

ang-=45;

soc=dribble(ang,DRIBBLE\_WITHBALL);

}

}

/\*else{

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

soc=dribble(ang,DRIBBLE\_WITHBALL);

}\*/

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);

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

if(num>0)

{

soc=leadingPass(WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,posAgent) ,1);

}

else

{

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

soc=dribble(ang, 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);

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

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

if(num>0)

soc=dribble(0,DRIBBLE\_WITHBALL);

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员方射门（周期数为偶数，球门的右侧射门，周期为奇数，向球门方向左侧射门）。

if ( WM->isBallKickable())

{

if(WM->isOpponentAtAngle(30,30)==false)

{ VecPosition posGoal(PITCH\_LENGTH/2.0,

(-1+2\*(WM->getCurrentCircle()%2))\*0.4\*SS->getGoalWidth());

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(WM->getBallPos().getX()<0)

soc=kickTo(VecPosition(0,0),1.0);

else

{

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

soc=dribble(ang,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())

{

Circle cir(WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,posAgent),5);

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

if(num=0)

soc=leadingPass(WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES, posAgent) ,1.0);

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())

{

{

Circle cir(WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,posAgent),5);

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

VecPosition diff=WM->getGlobalPosition(WM->getClosestInSetTo(

OBJECT\_SET\_TEAMMATES,posAgent));

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

if(num=0&&ang>=-90&&ang<=90)

soc=leadingPass(WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,posAgent) ,1.0);

}

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(WM->getBallPos().getX()<0)

{

soc=leadingPass(WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATES,posAgent)

,1.0);

}

if(WM->getBallPos().getX()>=0)

{

if(WM->getAgentObjectType（）！=OBJECT\_TEANMATE\_10)

{

soc=leadingPass(OBJECT\_TEANMATE\_10 ,1.0)

}

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

{

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

soc=dribble(ang,DRIBBLE\_FAST);

}

}

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(WM->getBallPos().getX()<0)

soc=dribble(0,DRIBBLE\_SLOW);

if(WM->getBallPos().getX()>=0)

{

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

soc=dribble(ang,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(WM->getBallPos().getX(),-34),1.0);

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

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

{

soc=kickTo(VecPosition(WM->getBallPos().getX(),34),1.0);

}

else

soc=dribble(0,DRIBBLE\_WITHBALL);

ACT->putCommandInQueue(soc);

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

}

//(14) 在 playOn 模式下，拿到球后，如果我是 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->isInTheirPenaltyArea(WM->getBallPos())) {

posGoalie=WM->getGlobalPosition(WM->getOppGoalieType());

ang\_goalie=(posGoalie-posAgent).getDirection();

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

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());

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);

ACT->putCommandInQueue(soc);

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

}

(15) 在 playOn 模式下,垂直带球。

if ( WM->isBallKickable())

{

if(WM->getBallPos().getY()<=0) soc=dribble(90,DRIBBLE\_WITHBALL);

else soc=dribble(-90,DRIBBLE\_WITHBALL);

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,然后传给最近的球员。

if ( WM->isBallKickable())

{

if(posAgent.getDistanceTo(posBall)>2.0) {

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

soc=dribble(ang,DRIBBLE\_WITHBALL);

}

else

soc=leadingPass(WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,posAgent),1.0);

ACT->putCommandInQueue(soc);

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

}

//(17) 在 playOn 模式下,10 号带球前进,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=leadingPass(OBJECT\_TEAMMATE\_10,1.0);

else soc=dribble(0,DRIBBLE\_SLOW);

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

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

soc=moveToPos(pos,20);

}

//(18) 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);

}

//(19) 在 playOn 模式下,2 号和 4 号一起去盯防对方拿球队员

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);

}

//(20) 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)

{

posGoalie=WM->getGlobalPosition(WM->getOppGoalieType());

ang\_goalie=(posGoalie-posAgent).getDirection();

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

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());

}

ACT->putCommandInQueue(soc);

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

}

//(21) 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);

ACT->putCommandInQueue(soc);

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

}

//(22) 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())

{

VecPosition posTeam;

AngDeg angTeam;

int iIndex;

for(ObjectTo=iterateObjectStart(iIndex,OBJECT\_SET\_TEAMMATES);o!=OBJECT\_ILLEGAL;o=iterateObjectNext(iIndex,OBJECT\_SET\_TEAMMATES))

{

posTeam=getGlobalPosition(o);

angTeam=(posTeam- posAgent).getDirection();

if(angA<=angTeam&&angTeam<=angB&&posAgent.getDistanceTo(posTeam)< dDist)

soc=leadPass(o,1.0);

ACT->putCommandInQueue(soc);

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

}

//(23).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())

{

if(WM->getBallPos().getX()<=0)

{

VecPosition pos(0,WM->grtBallPos().getY());

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

}

else

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

ACT->putCommandInQueue(soc);

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

}

//(24).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())

{

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

soc=dribble(ang,DRIBBLE\_FAST);

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

soc=moveToPos(VecPosition(52.5,0),20)；

if(WM->isInTheirPenaltyArea(WM->getBallPos()))

{

soc=leadingPass(OBJECT\_TEAMMATE\_10,1.0);

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

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

}

}

//(25).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\_7,1);

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

soc=leadingPass(OBJECT\_TEAMMATE\_9,1);

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

{

if(WM->getBallPos().getX()<40)

soc=dribble((VecPosition(40,0)-posAgent).getDirection(),DRIBBLE\_FAST);

else soc=ShootToGoalex(OBJECT\_GOAL\_L);

}

//(26)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

ObjectT WorldModel::getMaxXTeammateInSide(bool isOwnSize)

{

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((isOwnSize && oPos.getY() >= 0) ||

(!isOwnSize && oPos.getY() <= 0))

{

if (maxMate == OBJECT\_ILLEGAL || oPos.getX() > getGlobalPosition(maxMate).getX())

maxMate = o;

}

}

iterate ObjectDone(iIndex);

return maxMate;

}

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++;

count++;

}

return ownSideCount > count - ownSideCount;

}

//PlayerTeams.cpp

else if( WM->isBallKickable())

{

ObjectT mate = WM->getMaxXTeammateInSide(!WM->isOwnSideOpponentMost());

soc = leadingPass(mate, 1, DIR\_CENTER);

ACT->putCommandInQueue(soc);

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

return soc;

}

//（27）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);

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

if(num>=2){

ObjectT ClosestG;

ClosestG=WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,VecPosition(52.5,0));

soc=leadingPass(ClosestG,1);

ACT->putCommandInQueue(soc);

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

}

}

//（28）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);

//(29)在 playon 的模式下，求出球运动方向的直线方程，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 ballRun = Line::makeLineFromPositionAndAngle(WM->getBallPos(),WM>getBallDirection());

if (ballRun.getDistanceWithPoint(WM->getAgentGlobalPosition()) < 4)

{

soc = moveToPos(ballRun.getPointOnLineClosestTo(WM->getAgentGlobalPosition()), 20);

ACT->putCommandInQueue(soc);

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

return soc;

}

//（30）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);

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

if(num==0){

AngDeg angDribble=(VecPosition(53.0,0)-posAgent).getDirection();

soc=dribble(angDribble,DRIBBLE\_FAST);

ACT->putCommandInQueue(soc);

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

}

}

//（31）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);

moveToPos(pos,20);

//（32）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。

Circle cir(posAgent,7);

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

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

{

leadingPass(WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,posAgent),1.0);

}

//(33 )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 ) );

}

//34 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）。

else if(WM->isOffsideUs()) {

if(WM->getFastestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL, &iTmp )

== WM->getAgentObjectType()) {

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);

}

}

//35 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。

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);

}

}

//36.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

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 );

}

}

//37 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。

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 ) );

}

//38 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。

else if ( WM->isOffsideUs() )

{

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

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

{

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

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

double t ,m ,x2,y1,d,x1,y2;

t=b/( a-53);

m=53\*t+b;

d=pow( 2\*a+2\*t\*m,2) -4\*( pow( t,2)+1) \*(pow(a,2)+pow(m,2)-144) ;

x1 = ( ( 2\*a+2\*t\*m) +sqrt( d) ) /( 2\*( pow( t,2) +1)) ;

x2 = ( ( 2\*a+2\*t\*m) -sqrt( d) ) /( 2\*( pow( t,2) +1)) ;

y1 = t\*x1-53\*t ;

y2 =t\*x2-53\*t ;

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

}

//39 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。

else if ( WM->isOffsideUs( ) )

{

ObjectT o1 = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES , OBJECT\_BALL) ;

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

{

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

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

else

{

ObjectT o3 = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES ,posAgent) ;

soc = leadingPass( o3 , 1) ;

}

}

ObjectT o2 = WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATES , OBJECT\_BALL) ;

if( WM->getAgentObjectType() != o2)

{

double y = posAgent.getY() ;

VecPosition pos( 52.5, y);

soc = moveToPos( pos,20) ;

}

ACT->putCommandInQueue( 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，那么求出距离球第二近的队友的坐标加上（10，5）的点跑。如果该点在球场内，则跑向该点。

else if(WM->isOffsideUs())

{

ObjectT o1=WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,OBJECT\_BALL) ;

ObjectT o2=WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATES,OBJECT\_BALL) ;

double o2Posx;

double o2posy;

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);

}

else

{

ObjectT o3 = WM->getClosestInSetTo( OBJECT\_SET\_TEAMMATES , o1) ;

soc = leadingPass(o3,1);

}

}

if(WM->getAgentObjectType()!=o1&&WM->getAgentObjectType()!=o2&&o2Posx<=42.5&&o2posy<=29)

{

soc=moveToPos(VecPosition(o2Posx+10,o2posy+5),20);

}

ACT->putCommandInQueue( soc );

}

//41 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

int num=WM->getPlayerNumber();

if(WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS,WM->getBallPos())==OBJECT\_OPPONENT\_10&&

(num==2||num==3||num==4))

{

VecPosition pos=WM->getMarkingPosition(OBJECT\_OPPONENT\_10,2.0,MARK\_BALL);

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

ACT->putCommandInQueue( soc );

}

//42 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)

{

VecPosition pos=WM->getMarkingPosition(OBJECT\_OPPONENT\_9,2.0,MARK\_BALL);

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

}

}

//43 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 \*dDist1,\*dDist2;

WM->getClosestInSetTo(OBJECT\_SET\_TEAMMATES,OBJECT\_BALL,dDist1);

WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS,OBJECT\_BALL,dDist2);

if((\* dDist1)>(\* dDist2))

{

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

soc=mark(OBJECT\_BALL,5,MARK\_GOAL);

else

soc=mark(WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS,WM->getPlayerNumber()),5,MARK\_BISECTOR);

ACT->putCommandInQueue( soc = searchBall() ); // if ball pos unknown //执行找球动作！并放入命令队列

ACT->putCommandInQueue( alignNeckWithBody( ) ); // search for it //同时把脖子随身体一起转

}

//44 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

if(!WM->isBallKickable()){

if(WM->getPlayerType()==OBJECT\_OPPONENT\_11)

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

soc=moveToPos(WM->getGlobalPosition(OBJECT\_OPPONENT\_11)+VecPosition(0,-3),1);

soc=intercept(true);

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

soc=moveToPos(WM->getGlobalPosition(OBJECT\_OPPONENT\_11)+VecPosition(0,3),1);

soc=intercept(true);

ACT->putCommandInQueue(soc);

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

}

//45 在 playOn 模式下，防守模式下，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->getPlayerNumber()==6)

{

VecPostion vOpp=WM->getGlobalPosition(OBJECT\_OPPONENT\_9);

soc=moveToPos(vOpp);

}

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

{

VecPostion vOpp=WM->getGlobalPosition(OBJECT\_OPPONENT\_10);

soc=moveToPos(vOpp);

}

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

{

VecPostion vOpp=WM->getGlobalPosition(OBJECT\_OPPONENT\_11);

soc=moveToPos(vOpp);

}

ACT->putCommandInQueue(soc);

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

//46.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->offsidethem)

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

{

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

VecPosition vOPP=WM->getGlobalPosition(oOpp);

soc=moveToPos(vOpp);

ACT->putCommandInQueue(soc);

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

}

//47.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->offsidethem)

ObjectT objTea=WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATE,OBJECT\_BALL);

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

{

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

VecPosition vOpp=WM->getGlobalPosition(objOpp);

soc=moveToPos(vOpp,PS->getPlayWhenToTurnaAngle());

ACT->putCommandInQueue(soc);

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

}

//48.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->offsidethem)

{

ObjectT closeball,Myanget;

closeball=WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATES,OBJECT\_BALL);//获得当前距离求最近的球员

// Myanget=getAngetGlobalPosition();//当前球员获

if(WM->getAgentObjectType() !=closeball&&WM->getGlobalPosition(WM->getAgentObjectType()).getX()>0)//判断

{

soc=moveToPos(WM->getGlobalPosition(WM->getAgentObjectType())+(-10,0),20);

ACT->putCommandInQueue( soc );

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) ); // 把脖子转向球，也就是一直看着球

}

}

//49.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->isKickInThem())

{

Circle cir(posAgent,5);

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

if(num>0)

{

VecPosition pos=WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS,WM->getAgentObjectType());

pos.setX(pos.getX()+2);

soc=moveToPos(pos,0);

ACT->putCommandInQueue(soc);

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

}

}

//50.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() )

{

VecPosition pos;

double x;

pos = WM -> getBallPos();

x = pos.getX();

if ( x<0)

{

if( WM ->getPlayerNumber() == 4 || WM ->getPlayerNumber() == 5 )

{

ObjectT ClosestBall;

double DisBall;

AngDeg ang;

ClosestBall = WM -> getSecondClosestInSetTo ( OBJECT\_SET\_OPPONENTS , &DisBall );

ang = WM -> getRelativeAngle ( ClosestBall, true );

soc = SoccerCommand ( CMD\_DASH , 80, ang );

}

}

}