/\*\*

\*Copyright:

\*This code is free software; you can redistribute it and/or modify

\*it under the terms of the GPL-3.0

\*

\* @author Zeta

\* @date 2020.10.27

\*

\*/

//参考了百度文库里面有关文档的代码，感谢所有以前的学长。正是他们的无私奉献，才能使得这份资料薪火相传。有些地方进行了调整。

//这个文件里面的代码要放到uva\_base\_v13.2里面在2D平台上才可以运行

//PlayerTeams.cpp130行到138行是PlayOn模式开球后拿球到决策

//把原来到soc注释，在后面添加代码;每次修改后构建之后才能看到效果

//(1) 在 playOn 模式下,拿到球以后朝前方快速带球。

soc=dribble(0.0,DRIBBLE\_FAST);

//(2) 在 PlayOn 模式下,拿到球以后朝球门方向慢速带球。

AngDeg ang=(VecPosition(52.5,0.0)-posAgent).getDirection();

soc=dribble(ang,DRIBBLE\_SLOW);

//(3) 在 playOn 模式下,拿到球以后把球围绕自己身体逆时针转。

soc=kickBallCloseToBody(-120);

//(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)。

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);

}

//(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) {

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

}

else{

AngDeg ang=(VecPosition(52.5,0)-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;否则向球门方向快速带球。

Circle cir(posAgent,7);

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

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

if(num>0)

soc=dribble(0,DRIBBLE\_WITHBALL);

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)。

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

if(WM->isOpponentAtAngle(ang,100)==false){

VecPosition posGoal( PITCH\_LENGTH/2.0, (-1 + 2\*(WM->getCurrentCycle()%2)) \* 0.4 \* SS->getGoalWidth() );

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)

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

else{

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

soc=dribble(ang,DRIBBLE\_FAST);

}

//(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。

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);

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

//(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)。

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);

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

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

soc=leadingPass(WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATES,WM->getAgentObjectType()),1.0);

else{

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

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

else soc=dribble((VecPosition(52.5,0)-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,DRIBBLE\_SLOW);

else soc=dribble((VecPosition(52.5,0)-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。

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);

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

//(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。

//baofu上课说了这一题，判断夹角而不是守门员的位置，改一下。

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);

//(15) 在 playOn 模式下,垂直带球。

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

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。

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);

//跑位的策略要在180行修改，并且把原来的soc = turnBodyToObject( OBJECT\_BALL )

//成soc,这样就会执行修改的策略而不是固定的转身动作。

//(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。

//134行，拿球策略

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

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

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

//180行，跑位策略

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

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

soc=moveToPos(pos,20);

}

//(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，则直接朝着里对方球员远的球门点射门。

if(WM->getPlayerNumber()==10&&posAgent.getX()>30)

soc=kickTo(VecPosition(52.5,0),PS->getPlayerWhenToTurnAngle());

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

//(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

//把球给9号，(134行)

//把球给9号，(134行)

if(WM->getAgentObjectType()!=OBJECT\_TEANMATE\_9)

{

soc=leadingPass(OBJECT\_TEAMMATE\_9,1.0);

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

getDirection();

soc=dribble(ang,DRIBBLE\_FAST);

}

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())；

}

//180行

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

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

//(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\_7,1);

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);

}

//(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->getBallPos().getX()<=0)

{

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

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

}

else

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

// 拿球以后，需要自己添加 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。

soc=leadPass(WM->getClosestInSetTo(OBJECT＿SET＿TEAMMATES,

WM->getPosOpponentGoal()),1.0);

//(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。

//参考了书上209页的内容

//打开WorldModel.h,在里面预定义函数

ObjectT getTeammateAtAngleEx(AngDeg angA, AngDeg angB, double dDist);

//打开WorldModel.cpp,在里面添加一个新函数

ObjectT WorldModel::getTeammateAtAngleEx(AngDeg angA, AngDeg angB, double dDist)

{

VecPosition posAgent = getAgentGlobalPosition();

VecPosition posTeam;

AngDeg angTeam;

int iIndex;

for(ObjectT o = 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)

return o;

}

iterateObjectDone(iIndex);

return OBJECT\_ILLEGAL;

}

//将if(WM -> isBallKickable())内的内容修改为：

else if ( WM->isBallKickable())

{

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

ObjectT o = WM->getTeammateAtAngleEx(ang-30, ang, 20);

if( o != OBJECT\_ILLEGAL)

{

soc = leadingPass(o,1);

}

else

{

soc = dribble(ang, DRIBBLE\_SLOW);

}

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

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。

//在WorldModel.h中：找到public:，在之后加上：

bool isOwnSideOpponentMost();

ObjectT getMaxXTeammateInSide(bool isOwnSize);

//在WorldModel.cpp中：在开头加上：

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;

}

}

iterateObjectDone(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;

}

//再到PlayerTeams.cpp中的deMeer5()函数中：找到以下代码：

else if ( WM->isBallKickable())

//将大括号里面的内容修改为下面大括号里的内容：

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;

}

//（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.h中：

找到public:，在之后加上：

ObjectT getMaxXTeammate();

在WorldModel.cpp中：

在开头加上：

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;

}

//再到PlayerTeams.cpp中的deMeer5()函数中：

//找到以下代码：

else if ( WM->isBallKickable())

//将大括号里面的内容修改为下面大括号里的内容：

else if ( WM->isBallKickable()) // if kickable // 如果球已知，而且当前球在我脚下(可踢)

{

Circle cir(WM->getAgentGlobalPosition(), 7);

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

if (num > 2)

{

ObjectT mate = WM->getMaxXTeammate();

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

}

else

{

soc = dribble((WM->getPosOpponentGoal() - WM->getAgentGlobalPosition()).getDirection(), DRIBBLE\_FAST);

}

ACT->putCommandInQueue(soc);

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

return 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。

//180行跑完策略

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

soc=moveToPos(WM->getOffsideX()-2,posAgent.getY(),20);

//(27)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。

//找到代码：

formations->setFormation( FT\_433\_OFFENSIVE );//设置球队出场阵形！

soc.commandType = CMD\_ILLEGAL;//初始化soc命令对象

//在该代码后加入：

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;

}

//（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。

//和第6题类似

Circle cir(posAgent,7);

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

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

if(num>0)

soc=dribble(0,DRIBBLE\_WITHBALL);

else

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。

//180行跑完策略

if(WM->isBallInOurPossesion() && WM->getPlayerNumber==10 && posAgent.getX()<30)

VecPosition pos=WM->getBallPos()+VecPosition(0,10);

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worldmodel类里面参照isOpponentAtAngle()写一个判断己方队友的函数isTeammateAtAngle()

//在worldmodel.h中：

bool isTeammateAtAngle(AngDeg angA, AngDeg angB);

//在worldmodelHighLevel.cpp中：

bool WorldModel::isOpponentAtAngle( AngDeg ang , double dDist )

{

VecPosition posAgent = getAgentGlobalPosition();

VecPosition posOpp;

AngDeg angOpp;

int iIndex;

for ( ObjectT o = iterateObjectStart( iIndex, OBJECT\_SET\_TEAMMATES );

o != OBJECT\_ILLEGAL;

o = iterateObjectNext ( iIndex, OBJECT\_SET\_TEAMMATES ) )

{

posOpp = getGlobalPosition( o );

angOpp = ( posOpp - posAgent ).getDirection() ;

if ( fabs( angOpp - ang ) < 60 &&

posAgent.getDistanceTo( posOpp ) < dDist )

return true;

else if ( fabs( angOpp - ang ) < 120 &&

posAgent.getDistanceTo( posOpp ) < dDist/2.0 )

return true;

}

iterateObjectDone( iIndex );

return false;

}

在playerTeams.cpp中：

Circle cir(posAgent,5);

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

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

if(WM->isTeammateAtAngle(ang, 100)==false && num>=2)

{

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，则快速向球门放向 带球，距离守门员 5-10 米时向空隙较大的一侧射门。

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

if(WM->isOpponentAtAngle(ang,100)==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，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。

//和(31)重复

// (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。

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();

if (number >= 8 && number <= 11)

{

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

if (WM->isOpponentAtAngle(ang, 100) == 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(8)

//!!其他模式在PlayerTeams.cpp里面改好像是不行的，之前试过角球模式，不行(教材有点问题）

//!!改其他模式在Player.cpp的mainloop()里面改，192行点球和frozen模式后面加

//之前改过间接任意球是可以的

//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);

}

}

//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。

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 );

}

}

//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。

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

}

//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。

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 );

}

//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 );

}

//playOn 模式下，防守行为 （5 ）

//!我觉得这部分都加在180行无球的跑位策略里面

//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(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

//180行

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 );

}

//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)

{

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.h中添加

bool isOpponentCloserToBall()

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;

}

//在PlayerTeams中的180行

if(WM-> isOpponentCloserToBall())

{

ObjectT teammate=WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS,WM->getBallPos());

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

{

soc=moveToPos(WM->getGlobalPosition(teammate),40);

}

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

if(WM->getClosestInSetTo(OBJECT\_SET\_OPPONENTS,WM->getBallPos())==OBJECT\_OPPONENT\_11)

{

VecPosition oppPos=WM->getGlobalPosition(OBJECT\_OPPONENT\_11);

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

{

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

double dis=selfPos.getDistanceTo(oppPos);

if(dis>5)

{

soc=moveToPos(VecPosition(oppPos.getX(),oppPos.getY()+4),40);

}

else

{

soc=intercept(0);

}

}

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

{

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

double dis=selfPos.getDistanceTo(oppPos);

if(dis>5)

{

soc=moveToPos(VecPosition(oppPos.getX(),oppPos.getY()+4),40);

}

else

{

soc=intercept(0);

}

}

}

//48 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

//加到129行后面也可以？

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

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

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

ObjectT agent = WM->getAgentObjectType();

if(agent==closestTeam || agent==secondClosestTeam)

{

VecPosition oppPos=WM->getMarkingPosition(closestOpp,2.0,MARK\_BISECTOR);

soc=moveToPos(oppPos,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);

}

}

// 其他比赛模式下，防守行为 （7 ）

//!!改其他模式在Player.cpp的mainloop()里面改，192行点球和frozen模式后面加

//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。

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。

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);

}

else if(num == 3)

{

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),-34.0+10\*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。

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.

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。

else if ( WM->isOffsideThem() )

{

if ( 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

}