**实验四 复杂的动作决策**

**一、实验目的**

进一步了解 demeer5 并能熟悉的修改 demeer5 的内容以达到对场上的球员的控制：

1.能理解 UVA 程序中原来的 demeer5 中的全部内容；

2.能通过修改 demeer5 中的具体函数内容实现对场上球员的控制；

3.能通过底层动作的简单组合控制场上队员做出一些复杂动作决策；

4.对 WorldModel 有初步的认识，学会在 WorldModel,basicplayer 里添加新函数。

**二、实验设备**

硬件环境：PC机；

软件环境：虚拟机

**三、实验内容**

**1.在WM中增添函数判断守门员位置，朝空隙较大的一方射门**

这里应该使用角度来比较空隙大小。将自身和守门员、球门上界、球门下界分别连线，比较夹角。将球向比较大的夹角方向踢。

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();

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 ) );

}

在worldModel.cpp里面添加:

double WorldModel::getAgentDistFromGoalie()

{

VecPosition posAgent = getAgentGlobalPosition();

VecPosition posGoalie = getGlobalPosition( getOppGoalieType() );

double agentX = posAgent.getX();

double agentY = posAgent.getY();

double goalieX = posGoalie.getX();

double goalieY = posGoalie.getY();

return sqrt( (agentX-goalieX)\*(agentX-goalieX) + (agentY-goalieY)\*(agentY-goalieY) );

}

**2.在Basicplay中添加函数，如果无人阻挡则向前带球，否则闪避带球。**

首先添加函数isOpponentAtAngleEx(),判断在视野中指定夹角和距离构成的扇形中是否存在对手球员。若没有则向前带球。闪避带球则是分别判断当前视野的左45°和右45°范围内有无对方球员。如有则向另一方向带球以实现躲避。

在worldmodel.cpp里添加：

**//这个函数判断在扇区内有没有敌方球员，若有则ture 否则false**

bool WorldModel::isOpponentAtAngleEx( AngDeg angA, AngDeg angB, double dDist )

{

VecPosition posAgent = getAgentGlobalPosition();

VecPosition posOpp;

AngDeg angOpp;

int iIndex;

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

o != OBJECT\_ILLEGAL;

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

{

posOpp = getGlobalPosition(o);

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

if ( angA <= angOpp && angOpp <= angB && posAgent.getDistanceTo( posOpp ) < dDist)

return true;

}

iterateObjectDone( iIndex ) ;

return false;

}

在basicplayer.cpp里添加：

SoccerCommand BasicPlayer::dribble\_Dodge(VecPosition posAgent)//闪避带球

{

SoccerCommand soc;

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

soc = dribble(ang, DRIBBLE\_WITHBALL );

if( WM->isOpponentAtAngleEx( ang - 45, ang , 7.5 ) )

{

ang += 45;

}

else if( WM->isOpponentAtAngleEx( ang, ang + 45 , 7.5 ) )

{

ang -= 45;

}

soc = dribble( ang , DRIBBLE\_FAST);

return soc;

}

1. **实验截图**

****

1. **实验心得**

通过学习，我们能够理解 UVA 程序中原来的 demeer5 中的全部内容，对于球员的基本行为动作更为了解，并且能通过底层动作的简单组合控制场上队员做出一些复杂动作决策，同时对WorldModel有了初步的认识，初步学会了如何在 WorldModel,basicplayer 里添加新函数。