期中考试周要开始了，大家加油哈C:\Users\zkc\AppData\Local\Temp\SGPicFaceTpBq\12016\05980D94.png

1. main.cpp

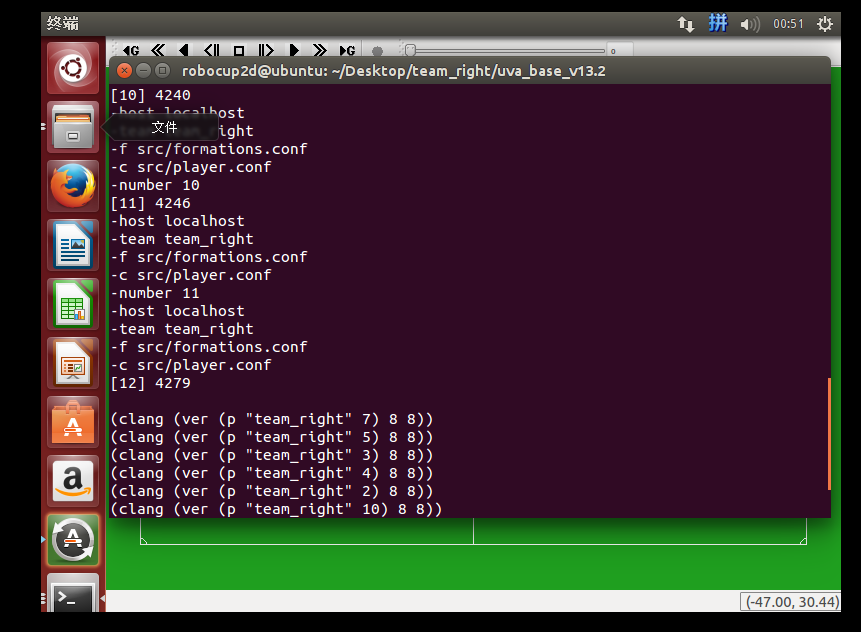
1)传入参数（可看可不看）

start.sh脚本通过以下语句传参数：

${prog} -number ${i} -host ${host} -team ${team} -f ${fconf} -c ${pconf} &

分别传入5个参数，其中f为阵型文件，c为设定的球员的动作的参数值。

输出查看，传入的参数如下：



2)初始化（可看可不看）

// read formations file

Formations fs( strFormations, (FormationT)cs.getInitialFormation(), iNr-1 );

// create worldmodel

WorldModel wm( &ss, &cs, &fs );

// make connection with server

Connection c( strHost, iPort, MAX\_MSG );

// link actHandler and worldmodel

ActHandler a( &c, &wm, &ss );

// link senseHandler with wm

SenseHandler s( &c, &wm, &ss, &cs );

// create player

Player bp( &a, &wm, &ss, &cs, &fs, strTeamName, dVersion, iReconnect );

// start listening，创建线程，用于从server获取信息

pthread\_create( &sense, NULL, sense\_callback, &s);

//为每个球员创建线程，

pthread\_create( &listen, NULL, stdin\_callback, &bp);

pthread\_create是类Unix操作系统（Unix、Linux等）的创建线程的函数。

第一个参数为指向线程标识符的指针。

第二个参数用来设置线程属性。

第三个参数是线程运行函数的起始地址。

最后一个参数是运行函数的参数。

//开始主循环

bp.mainLoop();

2.Player.cpp

1) mainLoop( )，根据球员角色（前锋，中场，后卫，守门员）确定要执行的函数

switch ( formations->getPlayerType( ) )

{

case PT\_GOALKEEPER:

soc = goalieMainLoop( );

break;

case PT\_DEFENDER\_SWEEPER:

case PT\_DEFENDER\_CENTRAL:

case PT\_DEFENDER\_WING:

soc = defenderMainLoop( );

break;

case PT\_MIDFIELDER\_CENTER:

case PT\_MIDFIELDER\_WING:

soc = midfielderMainLoop( );

break;

case PT\_ATTACKER:

case PT\_ATTACKER\_WING:

soc = attackerMainLoop( );

break;

case PT\_ILLEGAL:

default:

break;

}

查看不同角色对应的函数，惊讶的发现都是调用的deMeer5()函数

/\*! This is the main decision loop for the goalkeeper. \*/

SoccerCommand Player::goalieMainLoop( )

{

return deMeer5\_goalie();

}

/\*! This is the main decision loop for a defender. \*/

SoccerCommand Player::defenderMainLoop( )

{

return deMeer5() ;

}

/\*! This is the main decision loop for a midfielder. \*/

SoccerCommand Player::midfielderMainLoop( )

{

return deMeer5() ;

}

/\*! This is the main decision loop for an agent. \*/

SoccerCommand Player::attackerMainLoop( )

{

return deMeer5() ;

}

扩展：阵型文件formations.conf（脑洞一下，效果出众）

1.阵型文件在start.sh脚本里作为参数传入

2.定义了球员类型，每种类型对应一个编号，#为shell脚本的注释符号

# 0 = PT\_UNKNOWN

# 1 = PT\_GOALKEEPER

# 2 = PT\_DEFENDER\_CENTRAL

# 3 = PT\_DEFENDER\_SWEEPER

# 4 = PT\_DEFENDER\_WING

# 5 = PT\_MIDFIELDER\_CENTER

# 6 = PT\_MIDFIELDER\_WING

# 7 = PT\_ATTACKER\_WING

# 8 = PT\_ATTACKER\_CENTRAL

3.定义阵型类型，每种阵型对应一个编号

# 0 = FT\_UNKNOWN = 000

# 1 = FT\_INITIAL = 111

# 2 = FT\_433\_OFFENSIVE

# 3 = FT\_334\_OFFENSIVE

# 4 = FT\_DEFENSIVE = 442

# 5 = FT\_OPEN\_DEFENSIVE = 442

# 6 = FT\_343\_ATTACKING = 244

4.阵型信息（如何读入的，详细见Formations.cpp中的readFormations()函数 ）

# Formation 2 = FT\_433\_offensive

# Formation type number，每个阵型对应的有编号

2

# X-position of eleven players，11个数字，对应1-11这11个球员，X坐标信息

-50.0 -16.5 -21.0 -15.0 -16.5 0.0 0.0 -3.0 15.0 18.0 18.0 # X\_pos

# Y-position of eleven players

0.0 10.0 0.0 0.0 -10.0 -11.0 11.0 0.5 -0.5 19.0 -19.0 # Y\_pos

# Player types for eleven players，球员类型，1号对应类型1，为守门员

1 4 3 2 4 6 6 5 8 7 7 # P\_type

·以下几行，每行9个数字，对应0 – 8这9种球员类型

·通过X\_pos，球X坐标和X-attraction factors，确定当前周期球员打算前往的X坐标，Y坐标计算同理，计算公式如下，在Formations.cpp中的getStrategicPosition()函数中。

x = posHome.getX() + posBall.getX() \* ptInfo->getAttrX();

# X-attraction factors to ball for each player type

0.0 0.1 0.7 0.65 0.7 0.65 0.7 0.5 0.6 # X\_attr

# Y-attraction factors to ball for each player type

y = posHome.getY() + posBall.getY() \* ptInfo->getAttrY();

0.0 0.1 0.2 0.4 0.25 0.3 0.25 0.3 0.25 # Y\_attr

确定哪些球员类型要守在球后面，用于调整球员坐标

如果计算出来要前往的坐标比当前周期球的位置还要靠近敌方半场，还怎么防守嘛C:\Users\zkc\AppData\Local\Temp\SGPicFaceTpBq\12016\0615B29B.png

# Booleans denoting for each player type whether to always remain behind ball

0 1 1 1 0 0 0 0 0 # Behind\_ball

·确定每种类型球员X坐标的范围，不能头铁，一下子冲出界啊

# Minimal X-coordinate for each player type

0.0 -50.5 -42.0 -47.0 -45.0 -36.0 -36.0 -2.0 -2.0 # X\_min

# Maximal X-coordinate for each player type

0.0 -30.0 0.0 2.0 2.0 42.0 42.0 44.0 44.0 # X\_max

3. PlayerTeams.cpp（划重点，划重点了）

·deMeer5()是球员的策略函数，主要修改的就是这个函数

·代码里已经超级详细的注释了，这里就不说了，感谢以前的学长

