**（黑色细体是复制粘贴，应该不用看。**

**只考虑对方拿球队员位置。**

**脚本已通过，不知道实际咋样，坐标点待定。**

--desc:hello

**--Tier、Kicker、Receiver防守转换**

**-----使角色朝向球**

Kicker2BallDir = function()

return CRole2BallDir("Kicker")

end

Receiver2BallDir = function()

return CRole2BallDir("Receiver")

end

Tier2BallDir = function()

return CRole2BallDir("Tier")

end

**----获得敌方拿球队员序号num**

function getOppNum()

local oppTable = CGetOppNums()

**-- pairs 迭代 table元素的迭代器**

for i,val in pairs(oppTable) do -- 遍历 表 oppTable里的所有 key 和 value

num = tonumber(val) -- 把 value 字符串转为数字

if COppIsGetBall(num-1) then

return true

end

end

end

**---拿球球员是否在前场**

**function IsOppNum\_x1()**

**local OppGetBall\_x = COppNum\_x(num) -- num-1**

**if OppGetBall\_x > 180 then**

**return true**

**end**

**end**

**---拿球球员是否在中场**

**function IsOppNum\_x2()**

**local OppGetBall\_x = COppNum\_x(num-1)**

**if OppGetBall\_x < 180 and OppGetBall\_x >-180 then**

**return true**

**end**

**end**

**---拿球球员是否在后场**

**function IsOppNum\_x3()**

**local OppGetBall\_x = COppNum\_x(num-1)**

**if OppGetBall\_x < -180 then**

**return true**

**end**

**end**

**------------------------------------------**

**gPlayTable.CreatePlay{**

**firstState = "Start",**

**["Start"] = {**

**switch = function()**

**if COppIsGetBall(num) then**

**return "Defence"**

**-- elseif CIsGetBall(“Kicker”) or CIsGetBall(“Receiver”) or CIsGetBall(“Tier”) then**

**-- return “Attack”**

**elseif CIsGetBall("Kicker") then**

**return "Attack"**

**elseif CIsGetBall("Receiver") then**

**return "Attack"**

**elseif CIsGetBall("Tier") then**

**return "Attack"**

**end**

**end,**

**},**

**["Defence"] = {**

**switch = function()**

**if IsOppNum\_x1() then**

**return "Guard1"**

**elseif IsOppNum\_x2() then**

**return "Guard2"**

**elseif IsOppNum\_x3() then**

**return "Guard3"**

**end**

**end,**

**},**

**---Tier Kicker、Receiver都在后场防守。**

**["Guard1"] = {**

**Kicker = task.GotoPos("Kicker",-220,0,Kicker2BallDir),**

**Receiver = task.GotoPos("Receiver",-220,-50,Receiver2BallDir),**

**Tier = task.GotoPos("Tier",-220,50,Tier2BallDir),**

**Goalie = task.Goalie()**

**{switch = function() --？当先执行⬆，switch不加{}会出现错误警告：'}' expected (to close '{' at line 119) near 'switch'**

**if IsOppNum\_x2() then**

**return "Defence"**

**elseif IsOppNum\_x3() then**

**return "Defence"**

**elseif IsOppNum\_x1() then**

**return "Defence"**

**elseif CIsGetBall("Kicker") then**

**return "Attack"**

**elseif CIsGetBall("Receiver") then**

**return "Attack"**

**elseif CIsGetBall("Tier") then**

**return "Attack"**

**end**

**end,} --↑这段或许过于繁琐(つд⊂) –是有一丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢丢繁琐，可以考虑用or 或语句来把相同的return 放到一个语句里**

**},**

**--上面这一段代码有问题，你看你在“Guard1”这个防守状态机里面return “Defence” 或者 “Attack” 那这段代码就重新从”Defence”状态开始了，如果当前的状态又满足了转向”Guard1”，那状态机就又回到”Defence”了，这样就会在这个状态死循环了。后面也是一样**

**--Receiver、Tier在守门员边界前，Kicker在前方防守。**

**["Guard2"] = {**

**Kicker = task.RefDef("Kicker"),**

**Receiver = task.GotoPos("Receiver",-220,-20,Receiver2BallDir),**

**Tier = task.GotoPos("Tier",-220,20,Tier2BallDir),**

**Goalie = task.Goalie()**

**{switch = function()**

**if IsOppNum\_x2() then**

**return "defence"**

**elseif IsOppNum\_x3() then**

**return "defence"**

**elseif IsOppNum\_x1() then**

**return "defence"**

**elseif CIsGetBall("Kicker") then**

**return "Attack"**

**elseif CIsGetBall("Receiver") then**

**return "Attack"**

**elseif CIsGetBall("Tier") then**

**return "Attack"**

**end**

**end,}**

**},**

**---Tier在守门员边界前，Kicker、Receiver在前方防守。**

**["Guard3"] = {**

**Kicker = task.RefDef("Kicker"),**

**Receiver = task.RefDef("Receiver"),**

**Tier = task.GotoPos("Tier",-220,0,Tier2BallDir),**

**Goalie = task.Goalie()**

**{switch = function()**

**if IsOppNum\_x2() then**

**return "defence"**

**elseif IsOppNum\_x3() then**

**return "defence"**

**elseif IsOppNum\_x1() then**

**return "defence"**

**elseif CIsGetBall("Kicker") then**

**return "Attack"**

**elseif CIsGetBall("Receiver") then**

**return "Attack"**

**elseif CIsGetBall("Tier") then**

**return "Attack"**

**end**

**end, }**

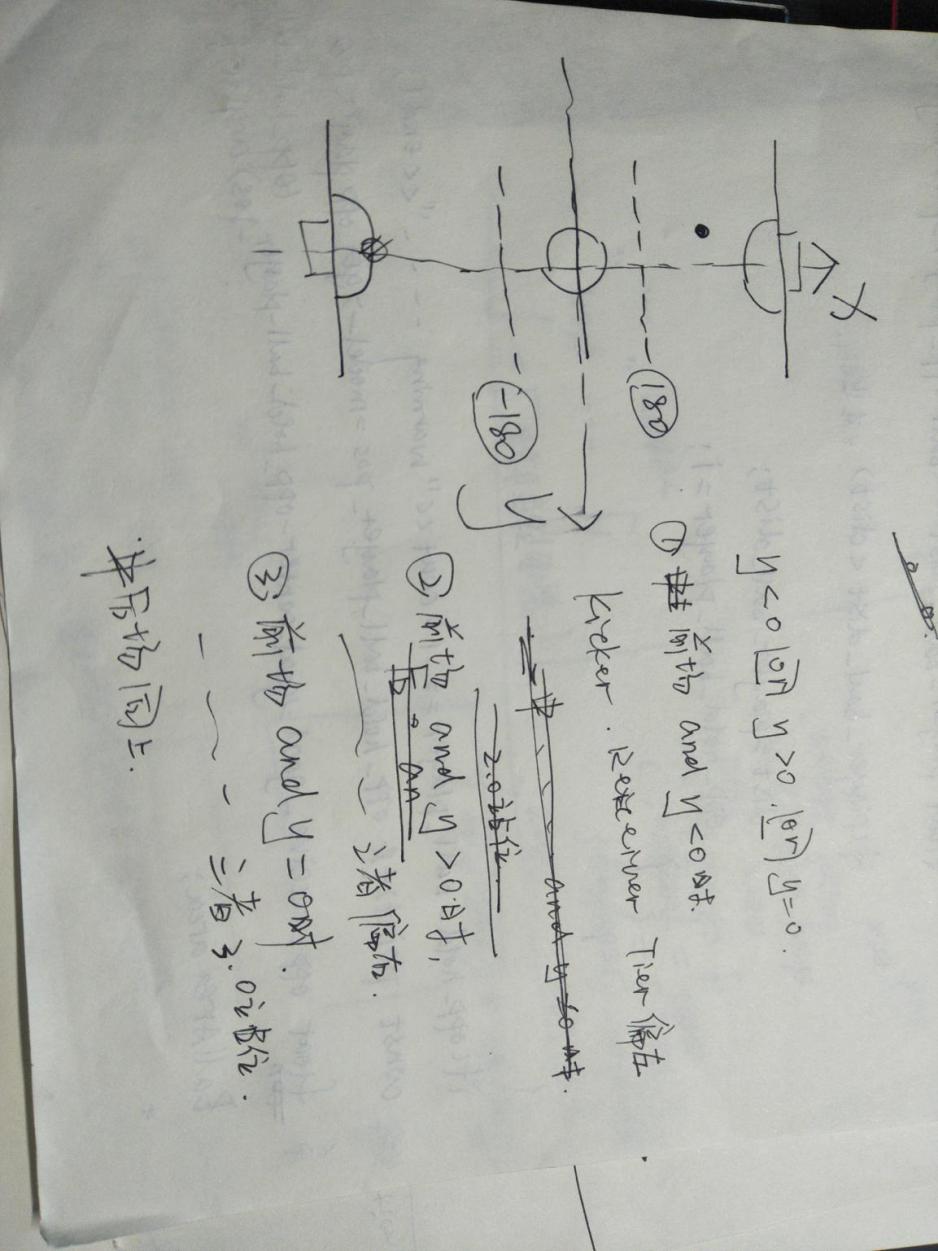
**},**

**name = "Normal\_defence"**

**}**

**唔西迪西4.0**

**考虑敌方拿球y坐标情况将比赛场地划分成6块并作出相应防守方案（？y=0也要考虑）。**

****

**---拿球球员是否在前场**

function IsOppNum\_x1()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x > 180 and OppgetBall\_y=0 then

return true

end

end

**--第二象限**

function IsOppNum\_x1\_0()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x > 180 and OppgetBall\_y<0 then

return true

end

End

**--第一象限**

function IsOppNum\_x1\_1()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x > 180 and OppgetBall\_y>0 then

return true

end

End

**---拿球球员是否在中场**

function IsOppNum\_x2()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x < 180 and OppgetBall\_x >-180 and OppgetBall\_y=0 then **--?能连等吗**

return true

end

End

**--第二、三象限**

function IsOppNum\_x2\_0()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x < 180 and OppgetBall\_x >-180 and OppgetBall\_y<0 then

return true

end

End

**--第一、四象限**

function IsOppNum\_x2\_1()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x < 180 and OppgetBall\_x >-180 and OppgetBall\_y>0 then

return true

end

End

**---拿球球员是否在后场**

function IsOppNum\_x3()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x < -180 and OppgetBall\_y=0 then

return true

end

End

**--第三象限**

function IsOppNum\_x3\_0()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x < -180 and OppgetBall\_y<0 then

return true

end

End

**--第四象限**

function IsOppNum\_x3\_1()

local OppgetBall\_x = COppNum\_x(num)

local OppgetBall\_y= COppNum\_y(num)

if OppgetBall\_x < -180 and OppgetBall\_y>0 then

return true

end

end