终于到主动技能的最后一块了——点目标技能

之前的4个主动技能教程的内容依然可以用在点目标技能

先附上之前4个教程的链接

先来点目标的基础设置

"MyAbility\_point"

{

"AbilityBehavior" "DOTA\_ABILITY\_BEHAVIOR\_POINT|DOTA\_ABILITY\_BEHAVIOR\_AOE"

"BaseClass" "ability\_datadriven"

"AbilityTextureName" "juggernaut\_omni\_slash"

"AbilityCastAnimation" "ACT\_DOTA\_ATTACK\_EVENT"

"AOERadius" "300"

//-------------------------------------------------------------------------------------------------------------

"AbilityCastPoint" "0.3"

//-------------------------------------------------------------------------------------------------------------

"AbilityCooldown" "1"

//-------------------------------------------------------------------------------------------------------------

"AbilityManaCost" "0"

"OnSpellStart"

{

"Damage"

{

"Target"

{

"Types" "DOTA\_UNIT\_TARGET\_BASIC|DOTA\_UNIT\_TARGET\_HERO"

"Teams" "DOTA\_UNIT\_TARGET\_TEAM\_ENEMY"

"Flags" "DOTA\_UNIT\_TARGET\_FLAG\_NONE"

"Center" "POINT"

"Radius" "%radius"

}

"Type" "DAMAGE\_TYPE\_PURE"

"Damage" "100"

}

}

"AbilitySpecial"

{

"01"

{

"var\_type" "FIELD\_INTEGER"

"radius" "300"

}

}

}

英雄的设置

"npc\_dota\_hero\_juggernaut"

{

"override\_hero" "npc\_dota\_hero\_juggernaut"

"AttackRate" "0.5"

"Ability1" "MyAbility\_unit"

"Ability2" "MyAbility\_null"

"Ability3" "MyAbility\_point"

"VisionNighttimeRange" "1800"

}

注意一下哦，这里有些地方不同

AbilityBehavior里面多了DOTA\_ABILITY\_BEHAVIOR\_AOE

这代表这个技能是点目标的AOE技能

"AOERadius" "300"这句话设置了鼠标指示器的大小

有DOTA\_ABILITY\_BEHAVIOR\_AOE加上AOERadius这个才能把鼠标变成一个蓝色的圈圈

两者缺一不可，可以删除其中一个看看效果

接下来加点效果

声音就不加了，就加个特效和击退效果

特效用particles/base\_destruction\_fx/gensmoke.vpcf

"OnSpellStart"

{

"Damage"

{

"Target"

{

"Types" "DOTA\_UNIT\_TARGET\_BASIC|DOTA\_UNIT\_TARGET\_HERO"

"Teams" "DOTA\_UNIT\_TARGET\_TEAM\_ENEMY"

"Flags" "DOTA\_UNIT\_TARGET\_FLAG\_NONE"

"Center" "POINT"

"Radius" "%radius"

}

"Type" "DAMAGE\_TYPE\_PURE"

"Damage" "100"

}

"FireEffect"

{

"EffectName" "particles/base\_destruction\_fx/gensmoke.vpcf"

"EffectAttachType" "world\_origin"

"TargetPoint" "POINT"

}

"Knockback"

{

"Target"

{

"Types" "DOTA\_UNIT\_TARGET\_BASIC|DOTA\_UNIT\_TARGET\_HERO"

"Teams" "DOTA\_UNIT\_TARGET\_TEAM\_ENEMY"

"Flags" "DOTA\_UNIT\_TARGET\_FLAG\_NONE"

"Center" "POINT"

"Radius" "%radius"

}

"Center" "POINT"

"Duration" "0.5"

"Distance" "300"

"Height" "300"

"IsFixedDistance" "0"

"ShouldStun" "1"

}

}

这里里面的部分知识都在前面的教程提到了，这里不再赘述

接下来是Lua，还是弄伤害就太单调了

技能效果：快速移动到指定的地方，到达之后对那个地方的单位造成伤害

需要修改一下技能，先去掉上面OnSpellStart里面的内容

OnSpellStart里面加入RunScript，所以我们需要在之前MyAbility.lua里面

加入MyAbility\_point函数

"MyAbility\_point"

{

//-------------------------------------------------------------------------------------------------------------

"AbilityBehavior" "DOTA\_ABILITY\_BEHAVIOR\_POINT"

"BaseClass" "ability\_datadriven"

"AbilityTextureName" "juggernaut\_omni\_slash"

//-------------------------------------------------------------------------------------------------------------

"AbilityCastPoint" "0.0"

//-------------------------------------------------------------------------------------------------------------

"AbilityCooldown" "0.3"

//-------------------------------------------------------------------------------------------------------------

"AbilityManaCost" "0"

"OnSpellStart"

{

"RunScript"

{

"ScriptFile" "scripts/vscripts/MyAbility.lua"

"Function" "MyAbility\_point"

"Target" "POINT"

}

"ApplyModifier"

{

"ModifierName" "modifier\_MyAbility\_point\_act"

"Target" "CASTER"

}

"ApplyModifier"

{

"ModifierName" "modifier\_phased"

"Target" "CASTER"

}

}

"Modifiers"

{

"modifier\_MyAbility\_point\_act"

{

"Duration" "0.2"

"OverrideAnimation" "ACT\_DOTA\_ATTACK\_EVENT"

}

}

"AbilitySpecial"

{

"01"

{

"var\_type" "FIELD\_INTEGER"

"radius" "300"

}

}

}

这里可以看出我把AbilityCastAnimation ，DOTA\_ABILITY\_BEHAVIOR\_AOE和AOERadius去除掉了

技能CD改为0.3秒

OnSpellStart里面有两个ApplyModifier，第一个运行了modifier\_MyAbility\_point\_act

这个modifier里面有个OverrideAnimation，就是修改动作的意思，并且这个动作持续时间0.2秒

主宰的剑刃风暴就可以用这个做，这个是修改技能施法前摇结束之后的动作

我删掉的AbilityCastAnimation这是用来修改施法前摇的动作

本质上有些区别的

第二个运行modifier\_phased，但是可以看到Modifiers里面却没有这个modifier

其实这个modifier\_phased是自带的，就是相位移动的BUFF，给单位之后就不会有碰撞

因为我们做的是移动，移动过程中肯定不要有碰撞，没有这个modifier，如果移动到那个地方，刚好有个单位在那里，两个有碰撞的单位靠得太近会卡住，两个都无法移动，这时候添加modifier\_phased再删除就可以解决这个问题

删除这个modifier我是在Lua里面删除的

Lua代码

function MyAbility\_point( keys )

local caster = keys.caster

local point = keys.target\_points[1]

local radius = 400 --设置范围

local teams = DOTA\_UNIT\_TARGET\_TEAM\_ENEMY

local types = DOTA\_UNIT\_TARGET\_BASIC+DOTA\_UNIT\_TARGET\_HERO

local flags = DOTA\_UNIT\_TARGET\_FLAG\_NONE

--设置施法者的面向角度

caster:SetForwardVector( ((point - caster:GetOrigin()):Normalized()) )

GameRules:GetGameModeEntity():SetContextThink(DoUniqueString("MyAbility\_point\_time"),

function( )

--判断单位是否死亡，是否存在，是否被击晕

if caster:IsAlive() and IsValidEntity(caster) and not(caster:IsStunned()) then

--不是死亡，存在这个单位，没被击晕，就运行这里面的内容

local caster\_vec = caster:GetOrigin() --获取施法者的位置

if (point - caster\_vec):Length()>50 then --当单位移动到距离指定地点小于50时不在进行移动

local caster\_abs = caster:GetAbsOrigin()

local face = (point - caster\_abs):Normalized()

local vec = face \* 75.0

caster:SetAbsOrigin(caster\_abs + vec)

return 0.01

else

--获取范围内的单位，效率不是很高，在计时器里面注意使用

local targets = FindUnitsInRadius(caster:GetTeam(), caster:GetOrigin(), nil, radius, teams, types, flags, FIND\_CLOSEST, true)

--利用Lua的循环迭代，循环遍历每一个单位组内的单位

for i,unit in pairs(targets) do

local damageTable = {victim=unit, --受到伤害的单位

attacker=caster, --造成伤害的单位

damage=100,

damage\_type=DAMAGE\_TYPE\_PURE}

ApplyDamage(damageTable) --造成伤害

end

caster:RemoveModifierByName("modifier\_phased") --删除modifier

print("MyAbility\_point Over")

return nil

end

else

caster:RemoveModifierByName("modifier\_phased") --删除modifier

print("Caster is death or stunned")

return nil

end

end, 0)

print("Run MyAbility\_point Succeed")

end

里面有些内容在之前的教程讲过了

新内容是移动的那部分

先看到caster:SetForwardVector( ((point - caster:GetOrigin()):Normalized()) )

(point - caster:GetOrigin()):Normalized())是计算caster的位置面向施法点的向量

可以用面向角度来理解，但是实际上是一个向量，是个Vector

这是高等数学里面知识了

接下来通过(point - caster\_vec):Length()可以获取两点间的距离，这里面的point和caster\_vec谁减谁都没关系，都可以求出两点间的距离

距离大于50就进行移动，这里说一下这个移动方法

我这个移动方法是参考下面网址里面的实例，参考之后加以利用完成

<https://developer.valvesoftware.com/wiki/Dota_2_Workshop_Tools/Scripting/Data_Driven_Motion_Controller_Example>

这种移动有点像黑洞吸引的感觉，把速率调高之后就可以做成很好的移动效果

如果编过魔兽地图，应该晓得魔兽的位移方法，但是在dota2里面并不是很好

不过，还是简单提一下，因为dota2的计算很方便

比如移动到施法者面前500距离处，这样我们就需要先获取面前500距离处的点

local vec=caster:GetOrigin() + 500\*caster:GetForwardVector()

caster:SetOrigin(GetGroundPosition(vec, caster))

就这么简单两句话，魔兽的话还要计算Cos和Sin

GetGroundPosition这句话是必须的，否者就会出现移动之后看不到单位的模型

上面的代码较多，不懂代码或者初学的可能看起来不太懂

不懂就问吧

还需要注意一点，这代码有时候很奇怪的，直接复制进去运行却得不到我说的这种效果

太奇葩了，群里有出现过这种情况

但是呢，有时候自己照着打也会运行错，不用担心，代码就是这样，不会出错就不叫代码了

就像我老师说的：计算机没有错，错的只有你自己

也就是说要么V社错了，要么你错了

所以写代码的时候要仔细看清楚，按照运行流程自己好好查一查

就如上面的Lua代码，其实我已经敲过几次那个代码了，但是在我做这个教程的时候自己再敲了一遍，结果就是错了，经过十多分钟的修改才正确运行，毕竟之前做过了，所以可以很快解决了，我第一次敲这个移动代码，弄了几个小时才成功，所以，不要起来，来路漫漫，还有更过错误等着你呢

最后在模拟一下米拉娜的月神箭

不过这里就不做那么复杂了，命中之后直接击晕5秒吧

为了方便测试，我先注释掉运行函数那一块

修改之后如下

"OnSpellStart"

{

//"RunScript"

//{

// "ScriptFile" "scripts/vscripts/MyAbility.lua"

// "Function" "MyAbility\_point"

// "Target" "POINT"

//}

"LinearProjectile"

{

"EffectName" "particles\econ\items\mirana\mirana\_crescent\_arrow\mirana\_spell\_crescent\_arrow.vpcf"

"MoveSpeed" "1000" //发射速率

"StartRadius" "0" //获取的开始范围

"EndRadius" "50" //获取的终止范围

"FixedDistance" "1000" //距离

"StartPosition" "CASTER"

"HasFrontalCone" "0" //不晓得是什么玩意

"ProvidesVision" "1" //是否给予视野

"VisionRadius" "500" //视野大小

"TargetTeams" "DOTA\_UNIT\_TARGET\_TEAM\_ENEMY"

"TargetTypes" "DOTA\_UNIT\_TARGET\_HERO"

"TargetFlags" "DOTA\_UNIT\_TARGET\_FLAG\_NONE"

}

"ApplyModifier"

{

"ModifierName" "modifier\_MyAbility\_point\_act"

"Target" "CASTER"

}

}

"OnProjectileHitUnit"

{

"Damage"

{

"Type" "DAMAGE\_TYPE\_PURE"

"Damage" "110"

"Target" "TARGET"

}

"Stun"

{

"Duration" "5"

"Target" "TARGET"

}

}

LinearProjectile发射一个线性投射物，米拉娜的月神箭或者帕克的波

记得特效要预载入

OK，这次点目标的教程结束了，这也就代表主动技能的教程到此结束了

接下来的技能教程当然就是被动技能，被动技能就可以不用这样分那么细了