```
"MyAbility_null"
{
    // 基础的定义，如果技能是单位目标还是无目标，单位目标的类型是什么
    //-------------------------------------------------------------------------------------------------------------
    "AbilityBehavior"                   "DOTA_ABILITY_BEHAVIOR_NO_TARGET"
    "AbilityUnitDamageType"                 "DAMAGE_TYPE_PURE"
    "AbilityUnitTargetTeam"             "DOTA_UNIT_TARGET_TEAM_ENEMY"
    "BaseClass"                             "ability_datadriven"
    "AbilityTextureName"                "juggernaut_blade_fury"

    "AbilityCastPoint"                  "0.0"
    "AbilityCooldown"                   "1.0 2.0 3.0 4.0"
    "AbilityManaCost"                   "5"

    "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"  "CASTER"
                "Radius"  "%radius"
            }
            "Damage"      "%damage"
            "Type"          "DAMAGE_TYPE_PURE"
        }
    }
    // Special
    //-------------------------------------------------------------------------------------------------------------
    "AbilitySpecial"
    {
        "01"
        {
            "var_type"          "FIELD_INTEGER"
            "damage"            "100 200 300 400"
        }
        "02"
        {
            "var_type"          "FIELD_INTEGER"
            "radius"        "400"
        }
```

```
    }
}
```

Damage 中 Target 的扩展已经在单体目标教程后篇中讲了

接下来我加入音效和特效
我选用痛苦女王的特效
particles/units/heroes/hero_queenofpain/queen_scream_of_pain_owner.vpcf
音效是巫医
Hero_WitchDoctor.Maledict_Tick
soundevents/game_sounds_heroes/game_sounds_witchdoctor.vsndevts

```
"MyAbility_null"
{
    // 基础的定义，如果技能是单位目标还是无目标，单位目标的类型是什么
    //-------------------------------------------------------------------------------------------------------------
    "AbilityBehavior"               "DOTA_ABILITY_BEHAVIOR_NO_TARGET"
    "AbilityUnitDamageType"             "DAMAGE_TYPE_PURE"
    "AbilityUnitTargetTeam"         "DOTA_UNIT_TARGET_TEAM_ENEMY"
    "BaseClass"                         "ability_datadriven"
    "AbilityTextureName"            "juggernaut_blade_fury"

    "AbilityCastPoint"              "0.0"
    "AbilityCooldown"               "1.0 2.0 3.0 4.0"
    "AbilityManaCost"               "5"

    "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"  "CASTER"
                "Radius"  "%radius"
            }
            "Damage"      "%damage"
            "Type"          "DAMAGE_TYPE_PURE"
        }

        "AttachEffect"
        {
```

```
            "EffectName"
        "particles\units\heroes\hero_queenofpain\queen_scream_of_pain_owner.vpcf"
            "EffectAttachType"  "follow_chest"
            "Target"            "CASTER"
        }

        "FireSound"
        {
            "EffectName"  "Hero_WitchDoctor.Maledict_Tick"
            "Target"      "CASTER"
        }
    }
    // Special
    //-------------------------------------------------------------------------------------------------------------
    "AbilitySpecial"
    {
        "01"
        {
            "var_type"          "FIELD_INTEGER"
            "damage"            "100 200 300 400"
        }
        "02"
        {
            "var_type"          "FIELD_INTEGER"
            "radius"        "400"
        }
    }
}
```

接下来加点效果

1. 对周围的敌人击晕
2. 周围的敌人将减少 50%的攻击力

```
"MyAbility_null"
{

    "AbilityBehavior"               "DOTA_ABILITY_BEHAVIOR_NO_TARGET"
    "AbilityUnitDamageType"             "DAMAGE_TYPE_PURE"
    "AbilityUnitTargetTeam"         "DOTA_UNIT_TARGET_TEAM_ENEMY"
    "BaseClass"                         "ability_datadriven"
    "AbilityTextureName"            "juggernaut_blade_fury"

    "AbilityCastPoint"              "0.0"
    "AbilityCooldown"               "1.0 2.0 3.0 4.0"
```

```
"AbilityManaCost"                   "5"

"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"  "CASTER"
            "Radius"  "%radius"
        }
        "Damage"      "%damage"
        "Type"          "DAMAGE_TYPE_PURE"
    }

    "Stun"
    {
        "Target"
        {
            "Types"         "DOTA_UNIT_TARGET_BASIC|DOTA_UNIT_TARGET_HERO"
            "Teams"         "DOTA_UNIT_TARGET_TEAM_ENEMY"
            "Flags"         "DOTA_UNIT_TARGET_FLAG_NONE"
            "Center"  "CASTER"
            "Radius"  "%radius"
        }
        "Duration"          "2"
    }

    "ApplyModifier"
    {
        "ModifierName"          "modifier_MyAbility_null_buff"
        "Target"
        {
            "Types"         "DOTA_UNIT_TARGET_BASIC|DOTA_UNIT_TARGET_HERO"
            "Teams"         "DOTA_UNIT_TARGET_TEAM_ENEMY"
            "Flags"         "DOTA_UNIT_TARGET_FLAG_NONE"
            "Center"  "CASTER"
            "Radius"  "%radius"
        }
    }
```

```
            "AttachEffect"
            {
                "EffectName"
        "particles\units\heroes\hero_queenofpain\queen_scream_of_pain_owner.vpcf"
                "EffectAttachType"  "follow_chest"
                "Target"            "CASTER"
            }

            "FireSound"
            {
                "EffectName"  "Hero_WitchDoctor.Maledict_Tick"
                "Target"        "CASTER"
            }
        }

        "Modifiers"
        {
            "modifier_MyAbility_null_buff"
            {
                "Duration"          "10"
                "Properties"
                {
                    "MODIFIER_PROPERTY_DAMAGEOUTGOING_PERCENTAGE"      "-50"
                }
            }
        }
        // Special
        //-------------------------------------------------------------------------------------------------------
        "AbilitySpecial"
        {
            "01"
            {
                "var_type"          "FIELD_INTEGER"
                "damage"            "100 200 300 400"
            }
            "02"
            {
                "var_type"          "FIELD_INTEGER"
                "radius"        "400"
            }
        }
}
```

接下来细讲一下 Lua 如何造成伤害

有两种方式
一种是通过 Target 传递单位组
一种是直接通过 API 获取单位组

先从 Target 传递单位组开始
在单体目标教程-中篇使用的是 MyAbility.lua
在这里面添加 MyAbility_null_1 函数

OnSpellStart 里面的内容如下

```
"OnSpellStart"
{
	"RunScript"
	{
		"ScriptFile"		"scripts/vscripts/MyAbility.lua"
		"Function"			"MyAbility_null_1"
		"Target"
		{
			"Types"		"DOTA_UNIT_TARGET_BASIC|DOTA_UNIT_TARGET_HERO"
			"Teams"		"DOTA_UNIT_TARGET_TEAM_ENEMY"
			"Flags"		"DOTA_UNIT_TARGET_FLAG_NONE"
			"Center"	"CASTER"
			"Radius"	"%radius"
		}
	}

	"Stun"
	{
		"Target"
		{
			"Types"		"DOTA_UNIT_TARGET_BASIC|DOTA_UNIT_TARGET_HERO"
			"Teams"		"DOTA_UNIT_TARGET_TEAM_ENEMY"
			"Flags"		"DOTA_UNIT_TARGET_FLAG_NONE"
			"Center"	"CASTER"
			"Radius"	"%radius"
		}
		"Duration"		"2"
	}

	"ApplyModifier"
	{
		"ModifierName"		"modifier_MyAbility_null_buff"
		"Target"
		{
			"Types"		"DOTA_UNIT_TARGET_BASIC|DOTA_UNIT_TARGET_HERO"
```

```
            "Teams"         "DOTA_UNIT_TARGET_TEAM_ENEMY"
            "Flags"         "DOTA_UNIT_TARGET_FLAG_NONE"
            "Center"  "CASTER"
            "Radius"  "%radius"
        }
    }

    "AttachEffect"
    {
        "EffectName"
    "particles\units\heroes\hero_queenofpain\queen_scream_of_pain_owner.vpcf"
        "EffectAttachType"  "follow_chest"
        "Target"            "CASTER"
    }

    "FireSound"
    {
        "EffectName"  "Hero_WitchDoctor.Maledict_Tick"
        "Target"        "CASTER"
    }
}
```

Lua 里面的代码如下

```
function MyAbility_null_1( keys )
    local caster = keys.caster                  --获取施法者
    local targets = keys.target_entities    --获取传递进来的单位组
    local al = keys.ability:GetLevel() - 1    --获取技能等级，并且减 1

    --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=keys.ability:GetLevelSpecialValueFor("damage", al),    --在这里面必须技能等级减 1
            damage_type=keys.ability:GetAbilityDamageType()}      --获取技能伤害类型，就是 AbilityUnitDamageType 的值
        ApplyDamage(damageTable)      --造成伤害
    end
end
```

关于 keys 所能接收到的内容在单体目标教程后篇已讲

GetLevelSpecialValueFor 这个 API 函数可以获取 AbilitySpecial 里面的内容，注意传递技能等级的数值，GetLevelSpecialValueFor 里面等级 1 对应整数 0，等级 2 就对应 1，而 GetLevel 和 SetLevel 这两个 API 却不同等级 1 对应 1，等级 2 就对应 2

for 循环是 Lua 的基础知识，这里不多说了

上面的原理就是通过 Target 的扩展功能传递单位组到所调用函数内，keys 会接收传递进来的内容，通过 keys.target_entities 来获取传递进来的单位组，在通过 for 循环遍历单位组内的单位，依次造成伤害，其中伤害取自 AbilitySpecial 里面的 damage

接下来是利用 API 来获取，代码会多一些
在 MyAbility_null_1 函数下面建立 MyAbility_null_2 函数

OnSpellStart 里面的内容

```
"OnSpellStart"
{
    "RunScript"
    {
        "ScriptFile"        "scripts/vscripts/MyAbility.lua"
        "Function"              "MyAbility_null_2"
        "Target"            "CASTER"
    }

    //省略后面音效等内容
}
```

Lua 里面如下

```
function MyAbility_null_2( keys )
    local caster = keys.caster          --获取施法者
    local al = keys.ability:GetLevel() - 1    --获取技能等级，并且减 1

    local c_team = caster:GetTeam()   --获取施法者所在的队伍
    local vec = caster:GetOrigin()      --获取施法者的位置，就是三维坐标
    local radius = keys.ability:GetLevelSpecialValueFor("radius", al)  --获取范围

    local teams = DOTA_UNIT_TARGET_TEAM_ENEMY
    local types = DOTA_UNIT_TARGET_BASIC+DOTA_UNIT_TARGET_HERO
    local flags = DOTA_UNIT_TARGET_FLAG_NONE

    --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(c_team, vec, 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=keys.ability:GetLevelSpecialValueFor("damage", al),     --                在
```

GetLevelSpecialValueFor 里面必须技能等级减 1

```
            damage_type=keys.ability:GetAbilityDamageType()}      --获取技能伤害类型，就是 AbilityUnitDamageType 的值
        ApplyDamage(damageTable)      --造成伤害
    end
end
```

通过上面的代码可以看出，原理是类是的，只是获取方式不同
其中注意 types 那里，不是文本里面的|号而是+号
FindUnitsInRadius 是获取范围内单位的函数，返回一个 table，里面记录了所有获取到的单位
注意 FIND_CLOSEST 这里，这里的意思是按离 vec 这个点最近的单位进行排序，所以 targets[1] 就是离英雄最近的单位，因为 vec 这个点取自施法者所在的点
以下是排序的常量
Find Types

- FIND_ANY_ORDER "任何顺序"
- FIND_CLOSEST "按最近"
- FIND_FARTHEST "按最远"
- FIND_UNITS_EVERYWHERE "随机"

FindUnitsInRadius 这个 API 注意使用，最好不要每 0.1 一秒以下使用这个 API，会影响游戏体验，就是会变得卡，所以目前效率存在一定问题，坐等优化吧

还没完，接下来就教你怎么使用计时器
同样也是两种，一种在技能文本里，一种 Lua 里面，但是有些区别

我们先来文本里的
为了方便，我就清空 OnSpellStart 里面的内容

```
"OnSpellStart"
{
    "ApplyModifier"
    {
        "ModifierName"    "modifier_MyAbility_null_time"
        "Target"
        {
            "Types"        "DOTA_UNIT_TARGET_BASIC|DOTA_UNIT_TARGET_HERO"
            "Teams"        "DOTA_UNIT_TARGET_TEAM_ENEMY"
            "Flags"        "DOTA_UNIT_TARGET_FLAG_NONE"
            "Center"  "CASTER"
            "Radius"  "%radius"
        }
    }
}

"Modifiers"
```

```
{
    "modifier_MyAbility_null_time"
    {
        "IsHidden"              "1"
        "Duration"              "10"
        "ThinkInterval"         "1"
        "OnIntervalThink"
        {
            "Damage"
            {
                "Type"          "DAMAGE_TYPE_PURE"
                "Damage"        "10"
                "Target"   "TARGET"
            }
        }
    }
}
```

这效果是施放技能之后，获取范围内的敌人传进 modifier_MyAbility_null_time
然后传进去的单位都会有一个 modifier_MyAbility_null_tim，而这个 modifier 里面有个计时器
这个计时器持续 10 秒，每 1 秒对单位造成伤害，Duration 就是计时器持续时间，ThinkInterval 计时间隔，OnIntervalThink 响应计时器到期

这里提醒一下，modifier_MyAbility_null_tim 添加给单位的时候就会创建计时器，而这时计时器不会立马执行，而是先等第一个计时间隔到期之后才执行第一次 OnIntervalThink，可以利用这个特点做延迟多少秒后造成击晕等效果，如果是延迟 1 秒，需要把持续时间(Duration)设置为 1.1 即可，如果不设置 Duration，计时器会一直不停的持续下去，知道单位死亡，但是不建议这样做，这样容易引起游戏崩溃

接下来开始 Lua
Lua 的计时器就很自由了，我就直接在 MyAbility_null_1 里面修改
首先 OnSpellStart 内容

```
"OnSpellStart"
{
    "RunScript"
    {
        "ScriptFile"          "scripts/vscripts/MyAbility.lua"
        "Function"                "MyAbility_null_1"
        "Target"
        {
            "Types"         "DOTA_UNIT_TARGET_BASIC|DOTA_UNIT_TARGET_HERO"
            "Teams"         "DOTA_UNIT_TARGET_TEAM_ENEMY"
            "Flags"         "DOTA_UNIT_TARGET_FLAG_NONE"
            "Center"  "CASTER"
```

```
                "Radius"  "%radius"
            }
        }
}
```

Lua 里的内容

```
function MyAbility_null_1( keys )
    local caster = keys.caster                    --获取施法者
    local targets = keys.target_entities      --获取传递进来的单位组
    local al = keys.ability:GetLevel() - 1      --获取技能等级，并且减 1

    local time = 0
    GameRules:GetGameModeEntity():SetContextThink(DoUniqueString("MyAbility_null_1_tim
e"),
        function( )
            if time < 10 then
                --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
                    if IsValidEntity(unit) and unit:IsAlive() then      --这里判断单位是否为
空，且单位是否存活
                        local damageTable = {victim=unit,       --受到伤害的单位
                            attacker=caster,         --造成伤害的单位
                            damage=keys.ability:GetLevelSpecialValueFor("damage",    al),
    --在 GetLevelSpecialValueFor 里面必须技能等级减 1
                            damage_type=keys.ability:GetAbilityDamageType()}         -- 获
取技能伤害类型，就是 AbilityUnitDamageType 的值
                        ApplyDamage(damageTable)       --造成伤害
                    end
                end
                time=time+1
                return 1
            else
                return nil
            end
        end, 0)
end
```

Lua 里面的计时器就是这样用的

GameRules:GetGameModeEntity():SetContextThink(string a,handle b,float c)

第一个参数是字符串 a，意思就是计时器的名字，利用 DoUniqueString()这个 API 就可以生成一个独立的字符串，具体生成个啥样我也不知道，那是底层的玩意了

第二个参数其实就是传递一个函数进去就行了

第三个参数是小数 c，这个参数意味着延迟多少秒后执行第二个参数所填的函数，我填的是 0 所以会里面就执行一次，如果填 1，效果就跟技能文本里面的一样了

在计时器之前有个局部变量 time，这是用来计算时间的
在 function()…end 里面，先判断时间，因为是持续 10 秒，time 初始化为 0，所以当 time 小于 10 就继续调用函数，return　1(返回 1)的意思就是说 1 秒后再执行一次 function()…end，大于等于 10 就删除计时器，return　nil(返回 nil)的意思就是删除计时器

在 for 循环里面我加了点判断 IsValidEntity(unit) and unit:IsAlive()
意思就是判断单位是否存在实体和单位是否存活，如果不为空就返回 true，如果存活就返回 true
如果对一个不存在的单位，或者已阵亡的英雄继续造成伤害有可能会导致系统崩溃

对于这个 SetContextThink，其实还有 SetThink 的 API，只不过 SetThink 这个计时器不能独立运行，本次技能施放调用的 SetThink 会影响到前一次技能施放调用 SetThink，所以才使用这个 SetContextThink，这个可以独立运行，不会有冲突

好了，基本讲完了
如有不懂的地方欢迎提问

这次技能的附件: