[个人数值设计规范](http://www.gamelook.com.cn/2018/01/317869)

[战斗模拟](https://blog.csdn.net/LannisterPaoMo)

[战斗力计算](https://www.gameres.com/252735.html)

[万国觉醒伤害公式拆解](https://zhuanlan.zhihu.com/p/464730098)

[游戏资源价值的设计](https://mp.weixin.qq.com/s/yw21knewnAsWYJrCJ5HNnQ)

[回合制战斗](https://zhuanlan.zhihu.com/p/80180202)

[dota2维基百科](https://dota2.fandom.com/zh/wiki/Dota_2_Wiki)

[游戏中的经济系统](https://www.gameres.com/902283.html)

[MOBA英雄设计](https://mp.weixin.qq.com/s/w0YTflZba1TheMjCNviOCA)

[微分方程下的游戏技能设计](https://www.163.com/dy/article/I6PHDHH20526DPBA.html)

[抽奖模型](https://zhuanlan.zhihu.com/p/534645684)

**首先确定好最终伤害公式**

设计层和实现层

具备的素质

1.数学能力 2.逻辑能力 3.对数字敏锐的嗅觉 4.耐力 5.经验

DND核心规则之一：数值判定体系 roll点

控制数值的关键是减少数值之间的关联性

excel保存类型尽量使用 XLSB文件格式 优点：1.数据量大时效率高 2.支持VBA编写

excel中采用“ctrl+F3”打开名称管理器 多利用名称管理器保存各种常量，然后把常量放在单独的一个常量页签，以提高数值文档的可读性，不然别人在某个公式看见一个常量根本不知道是什么意思

使用数据条 让单元格中的百分比大小将决定填充的长度，从而使得填充的长度动态变化

**3.重要函数**

SUBSTITUTE**函数**

=SUBSTITUTE(M34,"a","b")

把单元格内文本中的a 换成 b

**TEXTJOIN函数（在microsoft 365版本生效）**

TEXTJOIN 函数将多个区域和/或字符串的文本组合起来，并包括你在要组合的各文本值之间指定的分隔符。 如果分隔符是空的文本字符串，则此函数将有效连接这些区域。

**XLOOKUP函数 （在office365版本 xlookup函数已经是大部分查询函数的上位替代了）**

=XLOOKUP(lookup\_value, lookup\_array, return\_array, [if\_not\_found], [match\_mode], [search\_mode])

=XLOOKUP(查谁, 在哪里找, 从哪里返回这个值对应的数据, 如果没找到返回什么, 匹配模式（近似查找等）, 查找模式（顺序 倒叙等）)

**VLOOKUP函数**

VLOOKUP(查找值,数据表,序列表,匹配条件) vlookup函数的使用范围是纵向查找引用的函数。lookup函数横向纵向都可以查询。从函数的解释可以知道，也就是只能列查询。

VLOOKUP(要找谁，在哪找，返回第几列的内容，精确还是近似)

⭐ **VLOOKUP函数的查找值必须位于查询区域中的第一列！** ⭐

**LOOKUP函数 多条件查找**

LOOKUP(1,0/((条件区域1=条件1)\*(条件区域2=条件2)),查询区域)

**INDIRECT函数 返回由文本字符串指定的引用函数**

=INDIRECT("人物基础DM!"&"R"&ROW()-3&"C"&8+$C$1,FALSE)

引用“人物属性DM”页签下，第当前-3行，第8+$C$1单元格内数字 列 所对应单元格内的内容。

此函数立即对引用进行计算，并显示其内容。 让固定的单元格引用能够变成使用变量的引用。

⭐ **如果需要更改公式中对单元格的引用，而不更改公式本身，则使用INDIRECT函数。⭐** INDIRECT函数在引用时，需要打开对应的表，否则会引用失败。

**INDEX函数 查找 INDEX(区域，行，列)**

INDEX(B2:K8,2,4) 意思是查找B2 K8区域里，第二行第四列 的值

**MATCH函数 MATCH(值，区域，查找模式) 精确查找为0 模糊查找为1或-1 1是升序排列 -1是降序排列**

EXCEL中 一个双引号 " 要用两个连续的双引号来表示 ""

=INDEX(**草稿纸3!$B$1:$B$40**,MATCH(**城池分布系数!C3**,草稿纸3!$A$1:$A$40,0)) 匹配的东西在哪列，后边的范围就是哪列 INDEX 最开始那个范围 是你想返回的值所在的数组

多条件查询时 需要按ctrl+shift+enter 重组数组

=INDEX($C$1:$C$9,MATCH(E2&F2,$A$1:$A$9&$B$1:$B$9,0))

=INDEX(B46:Z73,MATCH(B76,B46:B73,0),MATCH(C76,B46:Z46,0))

可以用2个match函数定位行和列，然后在数据源中找到需要的值

**XLOOKUP函数**

=XLOOKUP(查找的值，查找区域，返回区域，[匹配模式]，[搜索模式]

**CLOUMN函数**用于得到单元格的列号

**Indirect函数**用于用于返回由文本字符串指定的单元格引用

INDIRECT("'day"&$I8&"'!I:I") 假设I8 = 1 此处就代表day1工作表 I列内容

**HOUR函数 提取单元格里的小时以数字形式显示出来**

EXCEL中隐藏了一个将小写数字转换成中文大写数字的函数：**NUMBERSTRING**

=NUMBERSTRING(VALUE,TYPE) =NumberString(1234567890,1)

filter函数，筛选序列，参数一为需要筛选的组，参数二为条件，\*为并且

=**FILTER**(A2:A16,(B2:B16=F2)\*(C2:C16=G2))

33

array

: 这是您要筛选的数据范围，也就是要从中筛选出符合条件的数据的原始数据区域。在这个例子中，原始数据范围是 A2:A16。

include

: 这是筛选条件，用于指定筛选数据的条件。在这个例子中，筛选条件是 (B2:B16 = F2) \* (C2:C16 = G2)。这个条件使用了 \* (乘号) 来表示 AND 条件。意味着只有当 B 列的单元格等于 F2，并且 C 列的单元格等于 G2 时，才会被筛选出来。

[if\_empty]

: 这是可选参数，用于指定如果筛选结果为空时的返回值。如果不提供此参数，当筛选结果为空时，公式会返回 #CALC! 错误。

因此，该公式的作用是从 A 列的范围 A2:A16 中筛选出满足条件 (B2:B16 = F2) \* (C2:C16 = G2) 的数据，并将这些数据作为结果返回。

**把单元格内的数组向下分割在不同的行**

|  |
| --- |
| 700000011|700000012|700000013|700000014 |

|  |
| --- |
| 700000011|700000012|700000013|700000014 |
| 700000021|700000022|700000023|700000024 |
| 700000031|700000032|700000033|700000034 |
| 700000041|700000042|700000043|700000044 |

**=LARGE(K9:K13,2)**

**可能会有人把 LARGE 函数和 MAX 函数搞混，其实 LARGE 函数要比 MAX 函数功能更强大一些，因为它可以选择第 k 大的数值，而 MAX 只能选择最大值。**

**largeif复合函数**

找出满足条件1 和 条件2 中，对应列数（H列）第B6位置的值。一般用于找出一组值中符合条件的第x位的值。若B6 = 1 其实就是找出了最大值！

**=ROMAN(11)**

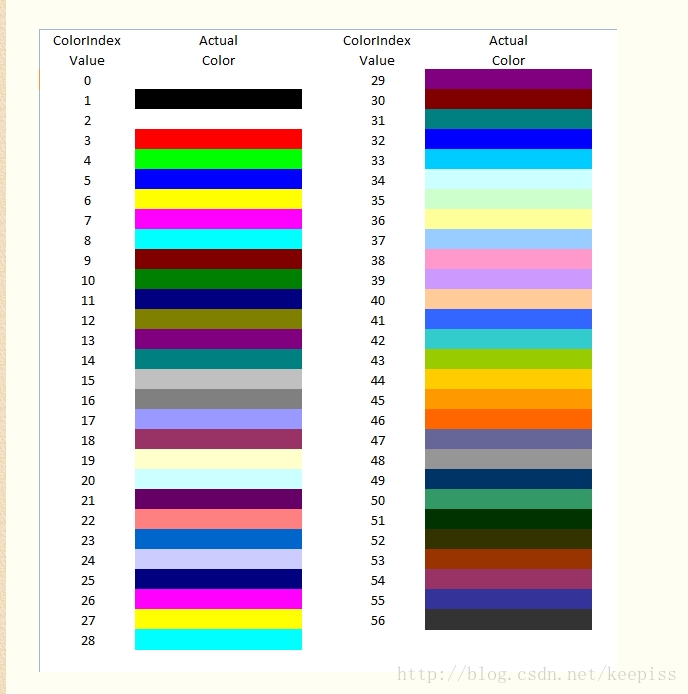
将阿拉伯数字11转化成罗马字

**时间戳转换为日期公式**

=(A1+8\*3600)/86400+70\*365+19

**将时间戳转换为日期格式**

**单元格颜色代码编号**



**公式=INDEX(A:A,RANDBETWEEN(1,10))，**

**公式核心：随机整数函数RANDBETWEEN(1,10)，随机1-10之间的整数**。

**RAND函数 从0-1随机数 配合RANK函数可以创造随机数组地图**

**ROUND函数 取整**

ROUNDUP(M24/10,0)\*10

**顶级取整函数**

=IF(O5<>0, ROUND(O5, -LEN(INT(O5))+3), 0)

|  |  |
| --- | --- |
| 14,963,655 | 15,000,000 |
| 22,445,483 | 22,400,000 |
| 33,668,224 | 33,700,000 |

**向上取整**

=CEILING(C3,3)

**向下取整**

=FLOOR(C3,3)

**把多少秒 转换成 小时分秒的时间格式**

=TEXT(INT(AI4/86400),"0天;;")&TEXT(AI4/86400,"h小时mm分ss秒")

**把年月日日期转换成星期几**

P9 =DATE(2022,6,13)

=TEXT(P9,"AAAA")

**从一组选手中随机选出3名出来**

=INDEX(SORTBY(A2:A12,RANDARRAY(COUNTA(A:A))),SEQUENCE(3))

**根据最大最小值生成数组**

=TEXTJOIN("|", TRUE, SEQUENCE(AE5 - AD5 + 1, 1, AD5))

截图.png

**对不清楚的函数值，选中它按F9查看解析。**

=VLOOKUP(A2,HP装备分成!$A:$G,HP模拟!$K$2+2,0)\*LOOKUP($K$3,HP各系统比重!$R$2:$R$12,HP各系统比重!$S$2:$S$12)\*LOOKUP($K$1,HP模拟!$M$2:$M$6,HP模拟!$N$2:$N$6)

=VLOOKUP(A2,HP装备分成!$A:$G,HP模拟!$K$2+2,0)\*VLOOKUP($K$3,HP各系统比重!$R$1:$S$12,2)\*VLOOKUP($K$1,$M$2:$N$5,2,0)

=INDIRECT("人物基础HP!"&"R"&ROW()&"C"&8+$K$1,FALSE) 人物基础HP页签当前行、第8+k行的值

IF(L4="","",INDEX([ItemConfig\_.xlsm]item!$A:$A,MATCH(L4,[ItemConfig\_.xlsm]item!$X:$X,0))) 如果单元格L4为空，则返回空，否则返回查找内容。

提取单元格中的纯数字 ctrl+e 快速智能填充

=MIDB(A2,SEARCHB("?",A2),2\*LEN(A2)-LENB(A2)) ←这个公式貌似有问题

upper（）

lower（）

单元给内容大小写转换

=RAND()\*(B-A)+A 产生随机数 来自 A or B

=LOOKUP(9E+307,--MID(A1,MIN(FIND({0;1;2;3;4;5;6;7;8;9},A1&1234567890)),ROW(INDIRECT("1:"&LEN(A1)))))

输出文本中自左起第一次开始出现的连续数字。

**FillSeriesWithArray** **填充序列数函数**

Sub FillSeriesWithArray()

Dim startNumber As Long

Dim endNumber As Long

Dim increment As Long

Dim totalNumbers As Long

Dim numbersArray() As Variant

Dim i As Long

Dim targetRange As Range

startNumber = InputBox("请输入起始数字：", "输入起始数字")

endNumber = InputBox("请输入结束数字：", "输入结束数字")

increment = InputBox("请输入递增步长：", "输入递增步长")

' 计算需要填充的数字总数

totalNumbers = ((endNumber - startNumber) \ increment) + 1

' 初始化数组

ReDim numbersArray(1 To totalNumbers, 1 To 1)

' 填充数组

For i = 1 To totalNumbers

numbersArray(i, 1) = startNumber + (i - 1) \* increment

Next i

' 确定目标范围

Set targetRange = Selection.Resize(totalNumbers, 1)

' 一次性将数组填充到目标范围

targetRange.Value = numbersArray

End Sub

**FillSeriesOptimized** **填充序列数函数**

'快速填充序列数

Sub FillSeriesOptimized()

Dim numToFill As Long

Dim startValue As Long

Dim targetRange As Range

Dim i As Long

Dim numbersArray() As Variant

' 判断当前选中单元格上一格的值是否为数字，如果是，则起始值为该值加1；否则起始值为1

If IsNumeric(Selection.Offset(-1, 0).Value) And Not IsEmpty(Selection.Offset(-1, 0).Value) Then

startValue = Selection.Offset(-1, 0).Value + 1

Else

startValue = 1

End If

' 用户输入需要填充的数量

numToFill = InputBox("请输入需要填充的数量：", "填充数量")

' 如果输入非法或者用户取消，则退出子程序

If numToFill <= 0 Then

MsgBox "请输入有效的填充数量！", vbExclamation

Exit Sub

End If

' 初始化数组

ReDim numbersArray(1 To numToFill, 1 To 1)

' 填充数组

For i = 1 To numToFill

numbersArray(i, 1) = startValue + i - 1

Next i

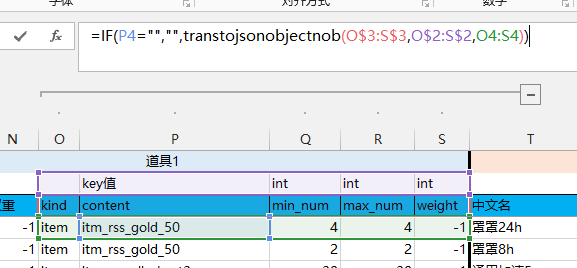
' 设置目标范围并填充数组到工作表

Set targetRange = Selection.Resize(numToFill, 1)

targetRange.Value = numbersArray

End Sub

**transtojsonobjectnob 函数（先定好json格式，然后格式的数据类型，然后转换数据）**



'transToJsonObject跳过空白值的版本，如果值没有填写就会跳过，没有这个属性，也没有对应的值

Public Function transToJsonObjectNoB(ByRef gen As Range, ByRef gentype As Range, ByRef kitsu As Range)

'Dim tempRng As Range

Dim saigo As String

'Set tempRng = ActiveSheet.Range("C4:P4")

Dim temp As Range

Dim temp2 As String

Dim i As Integer

i = 1

'Debug.Print gen(i).Cells(0, 1).Value

For Each temp In gen

If kitsu(i).Value = "" Then

GoTo x

End If

saigo = saigo & """" & temp.Value & """" & ":"

'Debug.Print temp.Value

If gentype(i).Value = "int" Then

If kitsu(i).Value = "" Then

saigo = saigo & "0,"

Else

saigo = saigo & kitsu(i).Value & ","

End If

ElseIf gentype(i).Value = "bool" Then

If kitsu(i).Value = "" Then

saigo = saigo & "false,"

Else

saigo = saigo & kitsu(i).Value & ","

End If

Else

saigo = saigo & """" & kitsu(i).Value & """" & ","

End If

'Debug.Print saigo

x:

i = i + 1

Next

Dim changdu As Integer

changdu = Len(saigo)

If changdu > 0 Then

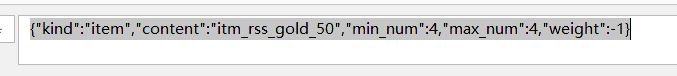
saigo = Left(saigo, changdu - 1)

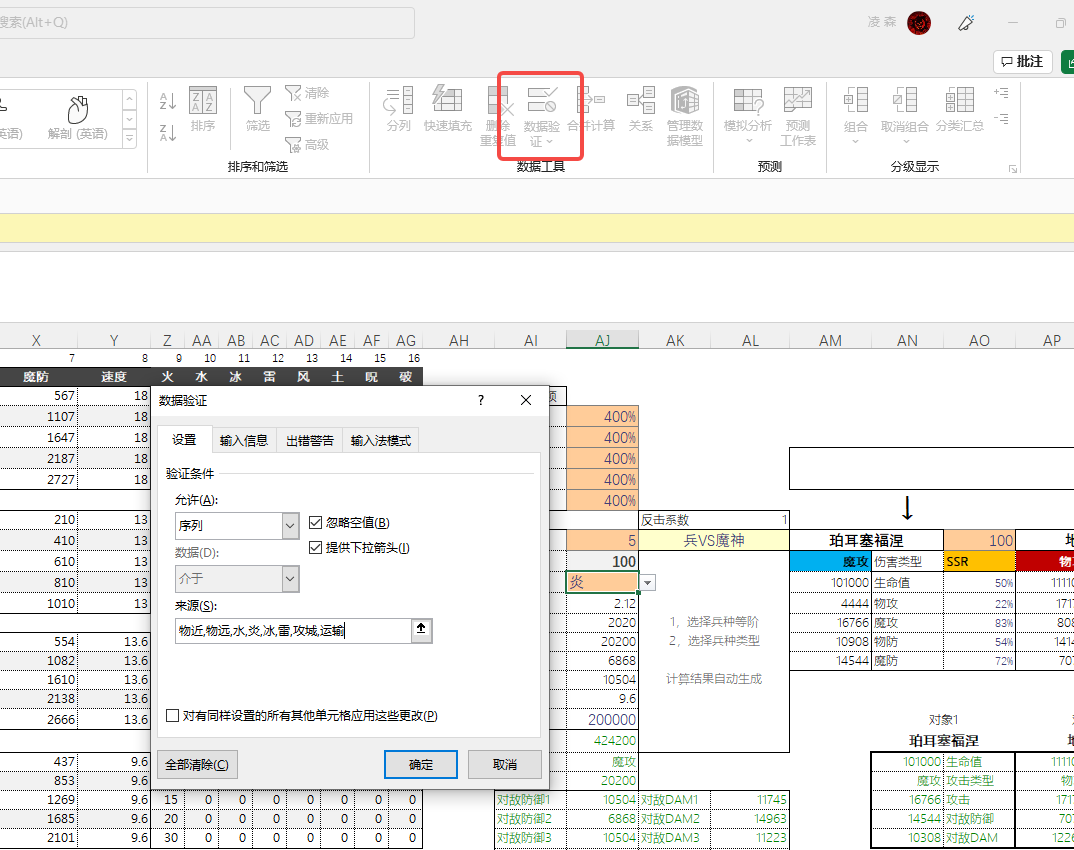
End If

saigo = "{" & saigo & "}"

transToJsonObjectNoB = saigo

End Function





**单元格给下拉菜单操作**

**获取当前excel文件的路径**

Public Function 获取工作簿名称()

Dim a As String

a = ThisWorkbook.Name

'获取工作簿名称 = Split(ThisWorkbook.Name, ".")(0)

'获取工作簿名称 = a

获取工作簿名称 = ActiveWorkbook.FullName

End Function

导表VBA

Sub ExportSheetsAsValues()

'定义变量

Dim wb As Workbook

Dim filePath As String

Dim fileName As String

Dim outputWB As Workbook

Dim outputWS As Worksheet

Dim sheetList As Variant

Dim sheetName As String

Dim i As Long

'设置文件路径和名称

filePath = "C:\Users\Username\Desktop\"

fileName = "ExportedWorkbook.xlsx"

'设置源工作簿和目标工作簿

Set wb = ThisWorkbook

Set outputWB = Workbooks.Add '新建一个工作簿，用于输出

'读取工作表列表

sheetList = wb.Sheets("Sheet1").Range("A2:A100").Value '修改为您实际使用的工作表名称所在位置

'复制工作表到目标工作簿

For i = 1 To UBound(sheetList, 1)

sheetName = sheetList(i, 1)

If Len(sheetName) > 0 Then '如果工作表名称非空

wb.Sheets(sheetName).Copy After:=outputWB.Sheets(outputWB.Sheets.Count)

Set outputWS = outputWB.Sheets(outputWB.Sheets.Count)

outputWS.Name = sheetName

outputWS.Cells.Copy

outputWS.Cells.PasteSpecial xlPasteValuesAndNumberFormats

Application.CutCopyMode = False

End If

Next i

'删除默认工作簿

If outputWB.Sheets.Count > 1 Then

outputWB.Sheets(1).Delete

End If

'保存目标工作簿

outputWB.SaveAs filePath & fileName

outputWB.Close

'释放内存

Set wb = Nothing

Set outputWB = Nothing

MsgBox "导出完成！"

End Sub

这段代码与前面的代码相似，唯一的不同之处在于，在将工作表复制到目标工作簿中后，我们现在使用以下代码将新工作表中的公式转换为值：

outputWS.Cells.Copy

outputWS.Cells.PasteSpecial xlPasteValuesAndNumberFormats

Application.CutCopyMode = False

这里我们首先将新工作表中的所有单元格复制一遍，然后使用 xlPasteValuesAndNumberFormats 参数将值和数字格式粘贴到同一区域内。最后，我们使用 CutCopyMode = False 来清除剪贴板中的内容。

**JsonArr '用于生成json用的数组,两边是中括号，不带引号的,跳过空白格子**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| JsonArr公式 | |  |  |  |
| 数据1 | 数据2 | 数据3 | 数据4 | 公式生成 |
| 1 | 2 | 3 | 4 | [1,2,3,4] |
| 1 |  | 3 | 4 | [1,3,4] |

Public Function JsonArr(ByVal kitsu As Range)

Dim temp As String

temp = ""

Dim gen As String

gen = ","

Dim rng As Variant

For Each rng In kitsu

If rng.Value <> "" Then

temp = temp & rng.Value & gen

End If

Next

Dim changdu As Integer

Dim changdu2 As Integer

changdu = Len(temp)

changdu2 = Len(gen)

If changdu > changdu2 Then

temp = Left(temp, changdu - changdu2)

End If

If temp = "" Then

JsonArr = ""

Else

JsonArr = "[" & temp & "]"

End If

End Function

**JsonArrSTR '用于生成json用的数组,两边是中括号，带引号的,跳过空白格子**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| JsonArrSTR公式 | |  |  |  |
| 数据1 | 数据2 | 数据3 | 数据4 | 公式生成 |
| 1 | 2 | 3 | 4 | [1,2,3,4] |
| 1 |  | 3 | 4 | [1,3,4] |

Public Function JsonArrSTR(ByVal rangeToConvert As Range) As String

Dim result As String

result = ""

Dim delimiter As String

delimiter = ","

Dim cell As Range

For Each cell In rangeToConvert

If cell.Value <> "" Then

result = result & """" & cell.Value & """" & delimiter

'Debug.Print "VB中用两个" & """" & """" & "转义表示字符串内容中的一个" & """" & "。"

End If

Next cell

Dim tempLength As Integer

Dim delimiterLength As Integer

tempLength = Len(result)

delimiterLength = Len(delimiter)

If tempLength > delimiterLength Then

result = Left(result, tempLength - delimiterLength)

End If

JsonArrSTR = "[" & result & "]"

End Function

**adddot '用来连接多个单元格的函数，两个参数分别是范围和连接符号**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| adddot公式 | |  |  |  |  |
| 数据1 | 数据2 | 数据3 | 数据4 | 连接符号 | 公式生成 |
| 1 | 2 | 3 | 4 | - | 1-2-3-4 |
| 1 |  | 3 | 4 | ! | 1!3!4 |

Public Function adddot(ByVal addRange As Range, ByVal note As String)

Dim temp As String

temp = ""

Dim rng As Variant

For Each rng In addRange

If rng.Value <> "" Then

temp = temp & rng.Value & note

End If

Next

Dim length1 As Integer

Dim length2 As Integer

length1 = Len(temp)

length2 = Len(note)

If length1 > length2 Then

temp = Left(temp, length1 - length2)

End If

adddot = temp

End Function

**JsonObject '生成json对象，只能生成单个的json对象，两头是大括号，要生成数组，用再使用本模块中的JsonArr函数，此自定义函数局限性较大，已经没有使用**

|  |  |  |  |
| --- | --- | --- | --- |
| transToJsonObject公式 | | |  |
| 字段类型 | 值 | 数据类型 | 公式生成 |
| 攻击力 | 1 | int | {攻击力:1,防御力:2,生命值:3,攻击力:0,防御力:false,生命值:} |
| 防御力 | 2 | bool |  |
| 生命值 | 3 | string |  |
| 攻击力 |  | int |  |
| 防御力 |  | bool |  |
| 生命值 |  | string |  |
|  |  |  |  |
| transToJsonObject公式 | | |  |
| 字段类型 | 值 | 数据类型 | 公式生成 |
| 攻击力 | 1 | int | {攻击力:1,生命值:3,攻击力:1,防御力:2} |
| 防御力 |  | bool |  |
| 生命值 | 3 | string |  |
| 攻击力 | 1 | int |  |
| 防御力 | 2 | bool |  |
| 生命值 |  | string |  |

Public Function JsonObject(ByRef kitsu As Range) As String

Dim temp As Integer

temp = 1

Dim temp2 As Variant

Dim temp3 As String

Dim json As String

json = ""

Dim mark As Integer

mark = 0

For Each temp2 In kitsu

If temp2.Value <> "" Then

' Debug.Print temp2.Value

If mark = 0 Then

temp3 = """" & temp2 & """"

Else

temp3 = temp2

mark = 0

End If

If temp2.Value = "count" Or temp2.Value = "level" Or temp2.Value = "value" Then

mark = 1

End If

If temp Mod 2 = 1 Then

json = json & temp3 & ":"

Else

json = json & temp3 & ","

End If

End If

temp = temp + 1

Next

Dim changdu As Integer

changdu = Len(json)

If changdu > 0 Then

json = Left(json, changdu - 1)

End If

JsonObject = "{" & json & "}"

End Function

**JsonObjectItem '生成json道具对象，只能生成道具名+数量这种格式例如框选了itm\_rss\_food\_2,100两个格子，则生成{"key": "itm\_rss\_food\_2","count": 100}**

Public Function JsonObjectItem(ByRef kitsu As Range)

Dim temp(2) As String

Dim temp2 As Range

Dim countP As Integer

countP = 0

For Each temp2 In kitsu

temp(countP) = temp2.Value

countP = countP + 1

If countP > 1 Then

Exit For

End If

Next

If temp(0) = "" Then

JsonObjectItem = ""

Else

End If

End Function

生成升序自然数 并且选中最后一个值

Sub FillNaturalNumbers() ' 生成自然数序列

Dim startValue As Integer

Dim numToFill As Integer

Dim iCount As Integer

' 获取起始值，如果当前单元格为空，则从 1 开始

If ActiveCell.Value = "" Then

startValue = 1

Else

startValue = ActiveCell.Value

End If

' 输入要填充到多少

On Error Resume Next ' 防止用户取消输入框导致的错误

numToFill = InputBox("请输入要填充到多少")

On Error GoTo 0 ' 恢复错误处理

' 填充自然数

If numToFill > 0 Then

For iCount = 0 To numToFill - 1

ActiveCell.Offset(iCount, 0).Value = startValue + iCount

Next iCount

ActiveCell.Offset(iCount - 1, 0).Select

Else

MsgBox "请输入有效的填充数量！", vbExclamation

End If

End Sub

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| transToJsonHorizon公式 | | |  |  |  |  |  |
| 道具ID1 | 数量1 | 道具ID2 | 数量2 | 道具ID3 | 数量3 | 道具ID4 | 数量4 |
| 木材 | 100 | 石材 | 500 | 钢铁 | 20 | 魔晶 | 500 |
|  |  |  |  |  |  |  |  |
| 公式生成 | [{key:木材,value:100},{key:石材,value:500},{key:钢铁,value:20},{key:魔晶,value:500}] | | | | | | |

**清楚区域内的内容（格式）**

sub test()

dim rng as range

set rng=range(

"A1:C100"

)

rng.clear

'清除内容及格式

'rng.clearcontents'这个是只清除内容

end sub

**4.设计层**

**等级设计需要注意的点：**

1.最快升级时间

2.平均升级时间

3.下一等级与当前等级升级时间的差异

4.单级升级的最长时间

5.付费档次所产生的升级差异

**职业基础属性设计**

一级属性和二级属性

战斗由 战斗公式生效前的战斗攻击流程 和 战斗公式与自身流程 决定。

攻击会产生哪种结果： “逐步判断”适合结果类型不多的情况 “圆桌理论” 适合更复杂的情况

**公式 很多情况下 公式越复杂，意味着这个公式并不那么好**

[游戏数值策划入门（二）：公式选择-技术圈 (proginn.com)](https://jishuin.proginn.com/p/763bfbd6eed1)

**减法公式：** 攻击-防御=损失 但是适用性太差 问题太大了

减法公式优点在于数值敏感性高，反馈明显。

**等式：** 攻击=损失 SLG游戏可以采用等式设定 等式不适合有防御力设计需求的游戏 通过增加攻击、防御类型解决丰富性问题

**等式变种（比防公式）：**也叫 比例防御公式 比防公式较减防公式更为科学，多应用于MOBA类游戏（数值封闭环境） 1.没有升级、有限升级的游戏 2.沙盒类游戏 3.横向升级游戏

攻击\*X/(X+防御) = 损失 （X是自然数） 【X/(X+防御) 可以理解为 攻击效率 1-X/(X+防御) 是伤害吸收】

攻击\*攻击效率 = 损失

比防公式均衡成长性较差

MOBA类大多采用比防公式

**天龙公式：**攻击\*攻击/(防御+攻击)=损失 攻击乘以攻击在攻防中所占的比重等于损失

**dota2公式：**

伤害 = 攻击方攻击力 \* (1 - 防御方减伤率)

防御方减伤率 = 防御方防御力 / (减伤常数 + 防御方防御力)

每秒进行普攻的次数＝(攻击速度+100) / 100 / 攻击间隔

**完美公式：**优点：推导速度快，调试速度快，100%精度

**攻击\*（第二攻击方数据）/防御 = 损失 --> 攻击\*第二攻击方数据 = 防御\*损失** 2对2

第二攻方数据的认知障碍和属性来源？ 第二攻击方数据=战斗力指数

方法一：直接命名法 方法二：隐藏法-》1.替代减法公式 2.保留传统攻防血体系 3.不提供额外属性

直接命名法：最简单 隐藏法：最高效 战斗力指数：最复杂（需要更复杂的规划和手动操作）

**COK的战斗公式：**

伤害 = 兵数^0.6\*ATK^2/(ATK+DEF(敌))

损兵 = 伤害/（血量\*bonus）

**bouns 为对应属性的加成**

**ROK的战斗公式：**

伤害 = NUM^0.54x攻击x系数xBonus

损兵 = 伤害/(防御\*血量xBonus)

**bouns 为对应属性的加成**

**伤害吸收公式：**有可能是魔兽世界用的公式 吸收X%伤害，但是不超过Y点

**成功率公式：**包含命中率，状态成功率，暴击率 天然一一对应 PS:成功率和攻防不是等价的

发动/抵抗=命中率 简洁方便

（命中/（命中+闪避））\*2=命中率 我的能力在你我能力之和中的占比多少

（（命中-闪避）/（命中+闪避））\*2=命中率 你我能力之差在你我能力之和的占比

**总结：有防御数值，用完美公式 没有防御数值，用等式。**

**三消 消除类游戏公式：** COMBO得分=100\*本次宝石个数\*2^combo次数

**性价比公式：**性价比=属性成长/消耗

**JJC积分变更公式：** P(D) = 1/(1+10^(-D/400)) R(n) = R(0) + K\*(W - P(D))

解释：当玩家A和玩家B的分差为D的时候，计算出玩家A对B的期望胜率P(D)。 玩家的新分数（RANK）由第二个公式取得。 K值是一个系数，玩家当前分数越高，K越低。W是玩家是否获胜（0 or 1）。

**一些伤害公式：**

伤害=攻击X（1-伤害减免百分比）

伤害减免百分比=护甲/（护甲+人物等级XK1+K2）

**物理伤害减免=护甲值/(护甲值+K) 魔兽世界中K值在不同版本是会变化的**

伤害=（攻击XK1+K2）/（攻击+防御XK3+K4）

伤害公式对比

减法公式： 伤害=攻击-防御

乘除法公式1： 伤害=攻击X （1-伤害减免百分比） 【该公式有较大问题】 伤害减免百分比=护甲/（护甲+人物等级X80+400）

乘除法公式2： 伤害=攻击X攻击/（攻击+防御）

**AOE技能伤害的平方根递减公式：**

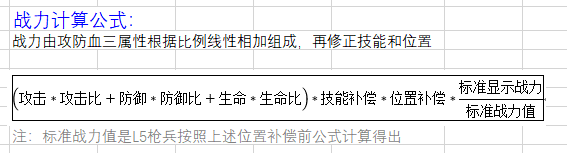
平方根递减公式是为了限制AOE技能对在面对极多目标时的伤害总量

例如圣骑士的神圣风暴，5目标后平方根递减

当目标数量为10时，伤害总量=技能伤害\**10*\*[SQRT(*5*/*10*)]=658.507

|  |  |  |
| --- | --- | --- |
| 目标数量 | 递减前 | 平房根递减后 |
| 1 | 93.127 | 不递减 |
| 2 | 186.254 | 不递减 |
| 3 | 279.381 | 不递减 |
| 4 | 372.508 | 不递减 |
| 5 | 465.635 | 不递减 |
| 6 | 558.762 | 510.0776 |
| 7 | 651.889 | 550.9468 |
| 8 | 745.016 | 588.9869 |
| 9 | 838.143 | 624.7149 |
| 10 | 931.27 | 658.5073 |

S1的战力计算公式



S2的伤害公式

1.魔神VS魔神

伤害=A攻击力\*A攻击力/（A攻击力+D防御力）\*（1-元素抗性比例） \*浮动值

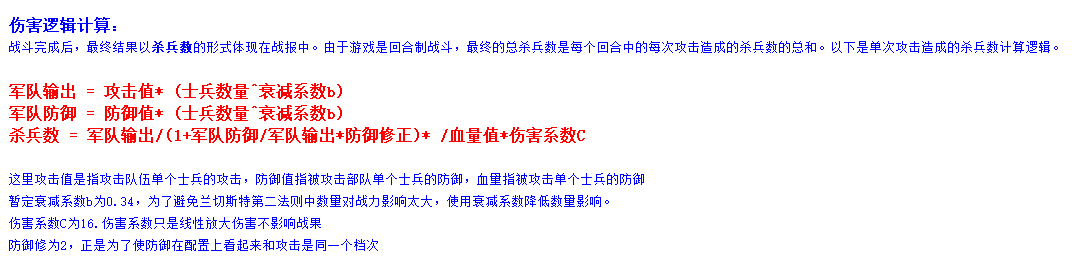
2.人造魔神VS人造魔神

伤害=A攻击力\*A攻击力/（A攻击力+D防御力）\*兵力^N\* 浮动值 \* 元素抗性比例

3.魔神VS人造魔神

先将魔神转化成人造魔神，在用2公式计算

|  |
| --- |
| 魔神各属性转化： |
| 生命值 ＝ 当前生命值　\*　当前等级对应转换参数 |
| 物理攻击 = 当前物理攻击　\*　当前等级对应转换参数　\*　当前等级对应兵力数＾Ｎ |
| 魔法攻击 = 当前魔法攻击　\*　当前等级对应转换参数　\*　当前等级对应兵力数＾Ｎ |
| 物理防御 = 当前物理防御 \* 1/当前等级对应转换参数 |
| 魔法防御 = 当前魔法防御 \* 1/当前等级对应转换参数 |



因为衰减系数b的存在，b值越小，士兵数量给军队伤害的影响越小。

反之，b值越大，士兵数量给军队伤害的影响就越大。

IFUN ROM项目战斗公式

伤害=攻击方攻击^2/(攻击方攻击+防御方防御）\*调控系数\*数量\*if（暴击，min（2.5，1.5+暴击伤害），1）\*if（格挡，0.5，1）

普攻伤害=伤害\*（1+-兵种克制系数）

技能伤害=伤害\*（1+-兵种克制系数）\*技能伤害系数

**怪物提供的经验值公式：**

**BaseXp=95+5\*LV PS:**如果改成0+5\*LV的话，怪物的经验增长率在前期会非常高，会出现越级杀怪的情况（大号带小号）

**经验值公式2：**

**BaseXp=949+LV^k1+LV\*k2**

**k1=3.5**

**k2=50**

**优化经验值公式：**

较复杂的概念源自于设计师对游戏整体的把握，例如我们在设计游戏中主角的成长经验值时，并不能随意给出个公式就了事，那是毫不负责任的。 经验计算的基础模型

我们通常说的成长所需经验公式大多是： 每级升级所需经验=Lv^3\*修正值+修正值

直接拿这个用，可以吗？当然不行！ 首先我们要了解成长所需经验的概念，我们为什么需要这个值，这个值控制了什么？稍加想象就可以明白，成长所需要经验公式的意义其实就是为了控制玩家的升级时间。 因此刚才那个公式应改为玩家练级**时间公式：**

每级升级所需时长=等级^3\*修正值+修正值，单位为秒。 我们再稍微修改一下：

每级升级所需时长=(等级-1)^3+60

经验公式的推导

当我们得到了练级所需的时间后，我们如何算升级所需的经验呢？其实有很多种做法，我暂时只说其中一种。

我们可以先设计怪物的经验计算公式，然后再设计怪物被玩家杀死的平均时长。例如我给怪物的经验计算公式设置为：

杀死同等级怪物所得经验=(怪物等级-1)\*2+60

然后再给杀死同等级怪物所需要的平均时长设置为：

杀死同等级怪物所需时长=5秒

那么升级所需要经验则为（假如我们的游戏只能通过杀怪升级）：

每级升级所需经验=每级升级所需时长/杀死同等级怪物所需时长\*杀死同等级怪物所得经验 更完整一点应该为：

每级升级所需经验=((等级-1)^3+60)/5\*((等级-1)\*2+60)

**每级升级所需经验**=floor(((等级-1)^3+60)/5\*((等级-1)\*2+60)+60,50)

**玩家练级时间公式：**每级升级所需时长=等级^3\*修正值+修正值

**跨级击杀怪物的经验奖励公式：**

当 CharacterLv<MonsterLv≤CharacterLv+4 时

XP=MonsterXP\*(1+0.05\*(MonsterLv-CharacterLv)) 【所能获得的经验上限是怪物经验的1.2倍】

**数据结构**

动态数据在某些情况下会调用静态数据。但是静态数据不会受到动态数据的影响。

CSV数据文件用系统自带的笔记本打开更好

战斗公式的设计，首先考虑战斗流程。

1.命中和闪避 是2个事件单独计算 还是 统一到一个事件计算。

2.采用“逐步判断”还是“圆桌理论”

3.是否存在特殊事件，例如神圣伤害等。

**命中公式**

Formula1=X1+ArmsHit1+Y1+Level1 X1：角色基础物理击中率 ArmsHit1：攻击方装备的命中率

Y1=Hit1/(Hit1+Miss1) Y1：命中率

Hit1=PlayerHit1+Item1+Skill1+State1 PlayerHit1：攻击方角色物理命中总值

Level1=(LV2-LV1)\*K1 ArmsHit1：攻击方装备物理攻击命中率

State1:攻击方buff对应物理命中总值

LV1：防御方等级 LV2：攻击方等级 K1=0.02

**暴击公式**

Formula2=K2\*(Cri1/(Cri1+Lv2\*K3+K4)) State2：攻击方buff对应物理暴击总值

Cri1=Item2+Skill2+State2 K2=0.75 K3=10 K4=80

**PRD机制 伪随机（剑圣暴击被动）**

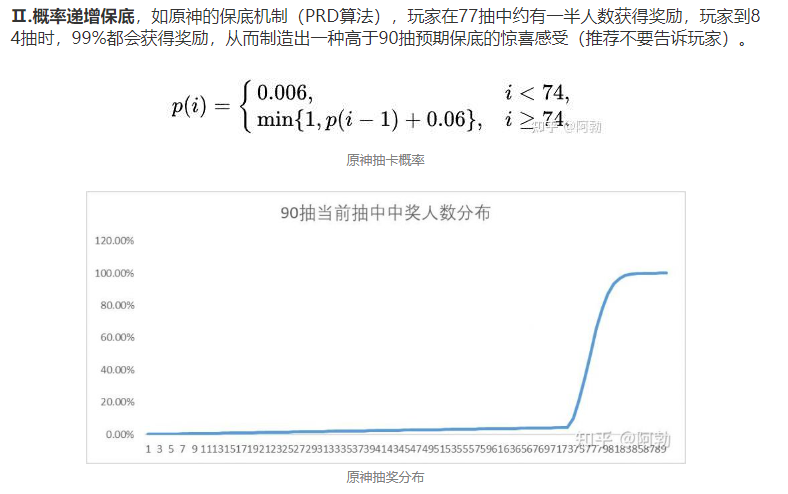
P（N）=C×N (N = 1 2 3 4 5 ``````)

通常几率即为该技能描述的几率

例如剑圣被动 25%概率暴击 那么对应的C值即为0.084744091852316990275274806

|  |
| --- |
|  |

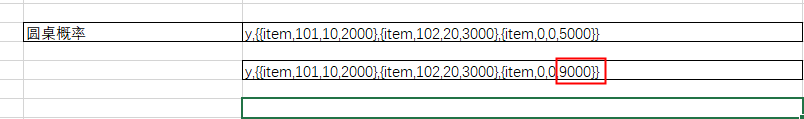
PRD算法在抽卡中的应用 例：原神抽卡概率



圆桌概率随机和独立概率随机

|  |  |  |  |
| --- | --- | --- | --- |
| 配置说明 |  |  |  |
| 随机方式 | y、d | y为圆桌概率，d为独立概率，配置在一个组(用{}包起来)的前面，定义这个组的随机方式 |  |
| 基础单位 | {类型,ID,数量,几率} | 类型包含：item(道具)、hero(英雄)、res(资源)、equip(装备),……(后续系统添加) |  |
| ID为该奖励在该类型中的id |  |
| 数量：奖励数量 |  |
| 几率：随机概率，除以10000 |  |
|  |  |  |  |
| 示例 |  |  |  |
| id | name | 配置格式 | 配置描述 |
| 1 | 圆桌概率 | y,{{item,101,10,2000},{item,102,20,3000},{item,0,0,5000}} | 从N个奖励中随机一种：40%几率获得10个ID为101的item，60%几率获得20个ID为102的item，有且仅有一种奖励出现。如果需要30%概率中奖，70%概率不中奖，则用item,0,0,概率填充不中奖的部分 |
| 2 | 独立概率 | d,{{item,101,10,4000},{item,102,20,6000}} | N个奖励分别独立进行随机：40%几率获得10个ID为101的item，60%几率获得20个ID为102的item，可能出现0种奖励、1种奖励、2种奖励的情况。 |
| 3 | 组合概率 | d,{y,{{item,101,10,4000},{item,102,20,6000}},y,{{item,103,30,3000},{item,104,40,7000}}} | N个圆桌概率组分别独立随机 |

龙哥我再问个问题哈，



这里空掉落如果变成9000，那么中奖概率应该为10%。那前边的2000 和 3000 需要对应的改为 400 和 600 ！ 因为圆桌概率的总权重就是固定的10000，桌子就那么大。且没有优先级判断

不放回抽样掉落的概率问题

期望掉落数量 X

该掉落组的掉落次数 N

空掉落权重 Y

**普通攻击伤害计算公式**

公式1：

最大物理攻击=(PlayerAttMax+Item3+State3)\*SkillCri+Skill3 SkillCri：攻击方技能攻击百分比

最小物理攻击=(PlayerAttMin+Item4+State4)\*SkillCri+Skill4

物理攻击=[最小物理攻击，最大物理攻击]

公式2：

物理伤害=物理攻击\*(1-物理攻击减免百分比)

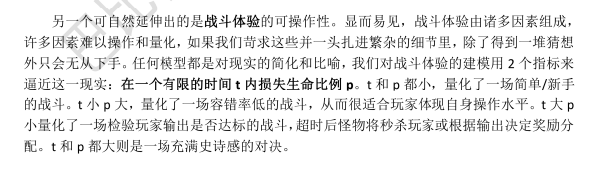
物理攻击减免百分比=K7\*(Def1/(Def1+Lv2\*K5+k6)) K5=25 K6=300 K7=0.75

Def1=PlayerDef1+Item5+State5

**暴击攻击的伤害计算公式**

暴击物理伤害=普通物理伤害\*(1+K8) K8=150% 为暴击增幅系数

**战斗体验**



短时间 大幅度生命值变动 = 低容错

长时间 小幅度生命值变动 = 高容错

**关于装备等级**

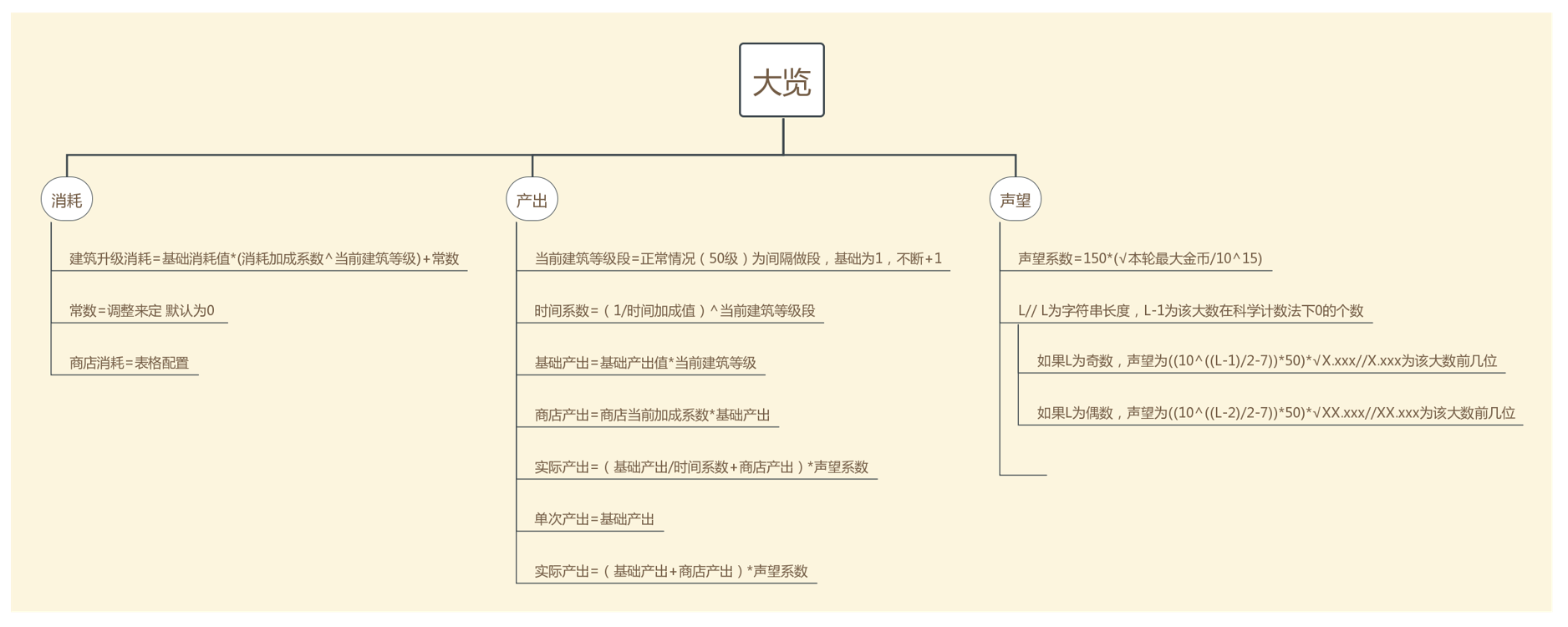
MMORPG游戏会按间隔固定等级来替换装备。

装备数值强度大致等于高一个等级低一个品质的装备数值强度。

同时还要考虑玩家在某等级的停留时间，综合决定装备强度。如果一个玩家在某一个等级停留时间很长，那么他对这个等级的装备追求兴趣就很大。

可以逐级配置装备属性，也可以按阶段比如 1~10 11~20 配置统一属性。

建筑升级消耗案例



excel操作：善用数据透视表

步长含义：按照多少值区分组。

例如步长为20

则 将统计数据每20为一个档位分组

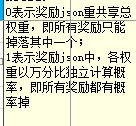
**模拟表**

模拟表应该是动态的，策划通过修改参数和基础数据模拟不同的情况。

比如装备模拟表应该根据装备品质、装备强化等级、装备等级、职业等不同来模拟装备提供的属性占比是多少。

而不应该是一个静态表，这样会导致平衡性调整时很麻烦！

关于权重的一些说明



1是独立权重 0是共享权重

**参数**

**参数影响方式： 1.乘法影响 2.加法影响**

eg：1. 攻击力上升30% 2. 攻击力上升300

**原子模型**

源：游戏中被修正的参数 源是按层级叠加下去的

源\*乘法修正+加法修正=下一级源

**(源1\*乘法修正1+加法修正1)\*乘法修正2+加法修正2··· = 最终的源**

在游戏设计中，原子模型的层级一般是多层的，起原因是：1.简化需求有限 2.方便归类 3.游戏设计需求（增加游戏内的因素） ——》**让玩家感觉到丰富，而不是麻烦**

**兰彻斯特方程：** 基础模型： 500个A和400个A 在完全接触的情况下对打， 我方剩余兵力为： 根号下(500^2-400^2) = 300

累乘： 比如 攻击力上升 20%，上升30%，上升50% 计算方式为： 攻击力2 = 攻击力1\*（20%+30%+50%）+ 攻击力1 =》 攻击力上升100% **先累加再相乘**

连乘：攻击力2 = 攻击力1\*（1+20%）\*（1+30%）\*（1+50%） = 2.34\*攻击力1 攻击力上升 134% **连续相乘**

累乘适用于更开放的数值环境。例如 攻击力可以无上限

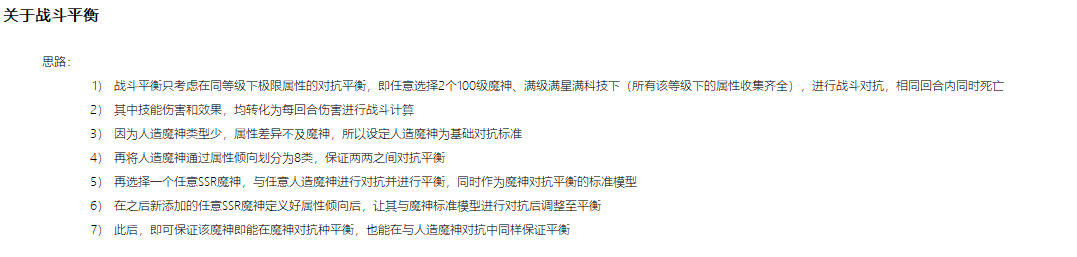
连乘一般用于减伤计算。例如：减伤A(减少20%伤害)和减伤B(减少30%伤害)一起开， 最终的减伤效果为 1-(1-0.2)\*(1-0.3) = 44% 并非 20%+30% 说明减伤收益递减了

**交易系统**

如果玩家能保证DAU（日活跃玩家数量），就不要做AI控制的物品投放机制，因为玩家可以形成完整的生态系统。

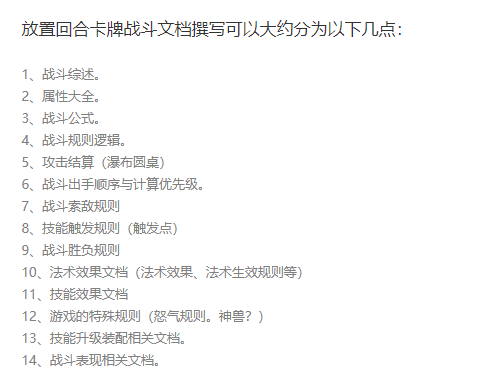
玩家间直接交易系统本质上是让大R用金钱换取非R玩家的游戏时间产出。

S2的战斗平衡思路



**公式对应表**

影响数值变化的部分基本都是影响参数，而不是改变公式！



<https://www.gameres.com/892595.html>

根据满级属性确定成长公式

假设属性公式为y = f(x)，其中x表示等级，y表示属性值。

根据要求，当满级400级时，属性值为24000，可以列出一个方程：

f(400) = 24000

要求每一级提升比例提高，可以考虑使用指数函数来描述属性值与等级之间的关系。假设每一级提升比例为k，则属性公式可以表示为：

y = a \* exp(kx)

其中a为初始属性值，由于要求满级时属性值为24000，因此可以得到：

24000 = a \* exp(400k)

可以解出a的值：

a = 24000 / exp(400k)

将a代入属性公式，可以得到最终的属性公式：

y = 24000 \* exp(kx) / exp(400k)

其中k为每一级提升比例。可以根据实际需求调整k的值，以控制属性值的增长速度。

根据上面的属性公式，当等级x=1时，属性值为：

y = 24000 \* exp(k) / exp(400k)

因为k可以任意设定，所以在没有具体的k值的情况下，无法精确计算出属性值。如果假设每一级提升比例k为0.01，则属性公式可以简化为：

y = 24000 \* exp(0.01x) / exp(4)

当x=1时，可以计算出属性值：

y = 24000 \* exp(0.01\*1) / exp(4) ≈ 60.79

所以，在这个假设的情况下，等级为1时，属性值约为60.79。

设计一个属性公式。满足 1.满级400级的时候 属性值为240000 2.每一级提升比例提高，但幅度不能太夸张 3.不要使用指数函数

可以使用一个简单的多项式函数来设计这个属性公式，例如：

**属性值 = 0.6 \* 等级 ^ 2 + 600 \* 等级**

其中，等级是指当前的等级，属性值是在该等级下计算出来的属性值。这个公式满足以下要求：

当等级为400时，属性值为240000，符合要求1。

该公式的一次项和二次项分别对应等级的平方和线性项，随着等级的提升，属性值的增长率也会提高，但幅度不会太夸张，符合要求2。

该公式不使用指数函数，符合要求3。

升级经验 = 100 \* 等级 ^ 3 + 5000 \* 等级 ^ 2 + 30000 \* 等级

其中，等级是指当前的等级，升级经验是在该等级下所需的升级经验。这个公式满足以下要求：

当等级为1时，升级经验为100，符合要求。

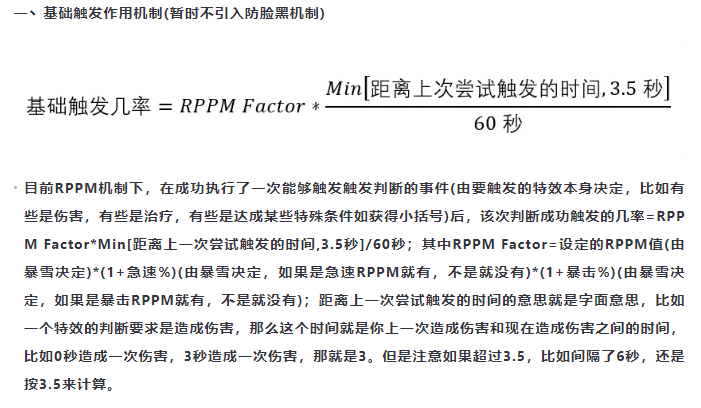
该公式的一次项、二次项和三次项分别对应等级的立方、平方和线性项，随着等级的提升，所需的升级经验增长率也会提高，但幅度不会太夸张，符合要求。

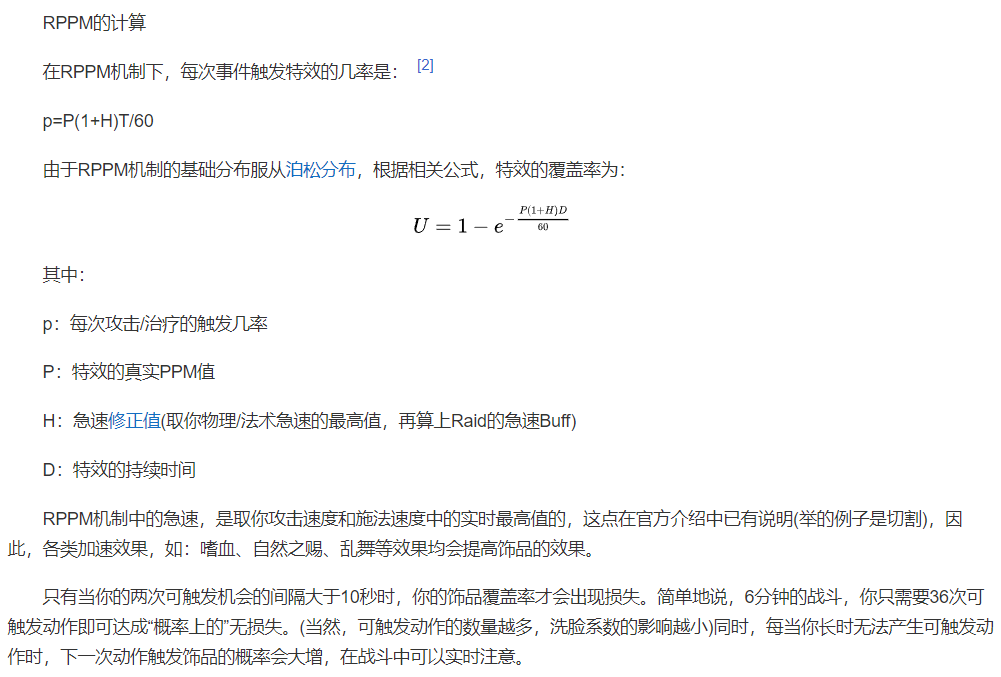
该公式不使用指数函数，符合要求。

**一些魔兽公式** [**躲闪招架格挡计算公式**](https://nga.178.com/read.php?tid=34352654)



[**特效触发RPPM机制（魔兽被动饰品触发等）**](https://ds.163.com/article/5fad851a517f0f5d064442fa/)





[**生产消费的市场监管**](https://www.gameres.com/903930.html)

我们作为设计者，我们是希望扮演优质“商品供应商”的角色，这导致游戏核心设计并不是一个自由市场，而是一个非自由且受到监管的市场。在游戏中我们可以通过扮演道具的生产者或者消费者控制市场变化。在这里我们引入3个带有监管属性的市场，最高限价型市场、最低限价型市场和窗口型市场

最低限价型市场：在游戏中我们和玩家均扮演消费者，而生产者只有玩家。这时我们针对某个商品给出一个固定的回收价格，由于我们拥有无限回购的能力，跟我们竞争的消费者玩家，必须要出比我们更高或者相同回购价格才有可能买到这个道具，那么这个道具就不受供求关系影响，价格将不低于我们的回购价格。在这个市场规则下，由于我们给予了最低回收价格，所以游戏中的生产者，不会因游戏中没有消费者而放弃生产，整体更像游戏中给予生产者的“补贴”，让他们可以继续完成“生产游戏”。

最高限价型市场：在游戏中我们和玩家一起扮演生产者，而消费者只有玩家。这是我们针对某个商品给出一个固定的出售价格，由于我们拥有无限生产的能力，跟我们竞争的生产者玩家，就必须出比我们更低或者相同的价格进行出售才有可能卖出这个道具，那么这个道具就不受供求关系影响，价格将不高于我们的出售价格。在这个市场规则下，我们可以通过限制最高出售价格，从而控制部分恶意囤货哄抬物价，也可以规避玩家逐利于部分性价比过高的产出行为。

窗口型市场：在游戏中我们跟玩家一样，均扮演生产者和消费者，商品的出售价格上限和回收下限均在我们控制范围。值得注意的当我们采用窗口型市场时，我们对于商品物价的上下限范围控制的越小，市场可预期性越高，但是市场自由度越差，交易带来的乐趣也越低。

一般来说游戏中如能做好防线下交易的的情况下（如果游戏中如不采用1级货币作为交易货币且参与交易的货币不能有效的转化为1级货币等），构建最低限价市场有利于实现维护游戏中依靠种地搬砖的小非R持续游戏，从而在一定程度上增强游戏留存；在游戏中构建最高限价市场的意义整体更偏于避免游戏的恶性体验，如避免用户购买极其珍贵的低性价比道具的服体验或者游戏中部分商人型玩家控制市场，提高价格，影响中下层购买付费体验。