编写代码注意事项

## 1、要注意变量的适用范围

local temp = math.random(1,10000)%100 --表A和表B，装备和消耗品的概率都是百分之50

if temp <=50 then

local FBIRCfg = {} --这句应该放在if语句前面

FBIRCfg = CONFIG.FbRewardTableA

base.print("从表A读取了一个装备数据!#####")

else

local FBIRCfg = {}

FBIRCfg = CONFIG.FbRewardTableB --掉落的消耗品，这个不分等级，从21到999都能掉落

base.print("从表B读取了一个消耗品数据!#####")

end

base.print("读取的表是:"..cjson.encode(FBIRCfg))

local FBIRCfg放在了if语句里面，其作用域就在那条语句内，在语句外的base.print会读到一个null。所以应该在if语句外声明local FBICfg!

## 2、每天都要注意以下design的需求文档的变化！

## 3、在lua中数组的使用，可以写成如下形式

testArray = {{betTimes = "lily"},{betTimes = "lucy"},{betTimes = "pupa"},{betTimes = "daibi"}}

--betTimes属性名设置成一样，方便访问，其根据下标序号来访问

for i = 1,#testArray do

print(testArray[i].betTimes)

end

## 4、神奇的正则表达式

{"req\_level"=0,"count"=1,"group"=1,"base\_pr"=50,"delta\_pr"=1,"basic\_times"=2000,"guarantee\_times"=3000,"repeatable"=1,"articleId":26003,"quality":4,"weaponType":2},

要将"eq\_level"=0替换成对应quaility数据的level，如quality为4表示要61级以上，可以用替换零宽断言(零宽度正预测先行断言)方式将前面"eq\_level"=0替换成"eq\_level"=61，正则表达式如下:

"req\_level\"=\d+(?=.+"quality":4)

## 5、--三维数组的实现方法

articles = {{weaponType = 1,group = 8,index = 1,code="secret is age = 17"},{weaponType = 1,group = 2,index = 2,code="secret is age = 19"},

{weaponType = 2,group = 5,index = 1,code="secret is age = 322"},{weaponType = 2,group = 3,index = 2,code="secret is age = 16"},

{weaponType = 5,group = 2,index = 1,code="secret is age = 9"},{weaponType = 3,group = 6,index = 1,code="secret is age = 21 "}}

Arc = {}

for i = 1,#articles do

acs = articles[i]

if Arc[acs.weaponType]==nil then

Arc[acs.weaponType] = {}

end

if Arc[acs.weaponType][acs.group]==nil then

Arc[acs.weaponType][acs.group] = {}

end

Arc[acs.weaponType][acs.group][acs.index] = acs

end

--print(Arc[46]["lucy"][2].name)

for v1,k1 in pairs (Arc) do

for v2,k2 in pairs(k1) do

for v3,k3 in pairs(k2) do

for v4,k4 in pairs(k3) do

print(v4..":"..k4)

end

end

end

end

### 6、Lua中要注意没有countinue。break语句会直接跳出循环

for i = 1,amount do

if weight[i] ==nil then

weight[i] = 0

end

end

local total = 0

base.print("the weight is:>>>"..cjson.encode(weight))

for i = 1,#weight do

total = total + weight[i]

end

local randNum = math.random(1,total)

local slectedArticleTable

local flag = true

for i = 1,#weight do

if weight[i] == 0 then --数字是跳着来的，如1,3,4,5，前面已经将空缺的补0，可以跳过 --break is wrong

--什么都不做 lua中break跳出循环，而且没有countinue

else

local tmpTotal = 0

for j = 1,i do

tmpTotal = weight[j] + tmpTotal

end

base.print("randNum is:"..randNum.."now tmpTotal is:"..tmpTotal.."flag is::"..cjson.encode(flag))

if (randNum <= tmpTotal) and (flag == true) then --不跳出循环，对于后面没选到的也要将他们的购买次数+1

slectedArticleTable = i

flag = nil

base.print("the selected i is:>>>>"..i)

end

betTimes[i] = betTimes[i] + 1

end

end

7、Linux去屏保...xset -dpms 关闭屏保，电源管理xset +dpms恢复

setterm -black 0关屏保

8、ss = {}

if ss[1] == nil then

ss[1] = {}

ss[1][2] = 2

ss[1][1] = 1

end

print(ss[1][1],ss[1][2])

创建二维数组的方法

9、break只能跳出一层循环！！！注意不经意的问题

for i = 1,#weight do

if GROUP ~= 0 then --当已经选出了物品后就不再继续判断了，break只能跳出一层循环

for j = 1,#weight[i] do

if weight[i][j] == 0 then

tmpNum = tmpNum + weight[i][j]

base.print("tempnumber is:"..tmpNum.."random is"..randNum)

if randNum <= tmpNum then

base.print("选择物品是随机到的数是:"..randNum)

GROUP = i

TYPE = j

break --双重循环只跳出了一层...

end

end

end

end

end