Table des matières

[Procédures stockées 3](#_Toc26454124)

[spAddCharacterLoot 3](#_Toc26454125)

[spCharacterHasEquip 3](#_Toc26454126)

[spCharLuckRoll 3](#_Toc26454127)

[spCharWisdomRoll 3](#_Toc26454128)

[spCompleteDungeon 4](#_Toc26454129)

[spDeleteAllCharacterLoot 4](#_Toc26454130)

[spDeleteRandomEquip 4](#_Toc26454131)

[spEquipNew 4](#_Toc26454132)

[spGenerateFight 4](#_Toc26454133)

[spGenerateMonsterHit 5](#_Toc26454134)

[spGeneratePlayerHit 5](#_Toc26454135)

[spGetCharacterAttack 5](#_Toc26454136)

[spGetCurrentWeight 5](#_Toc26454137)

[spGetDiscountedValueEquip 6](#_Toc26454138)

[spGetDungeonCompletion 6](#_Toc26454139)

[spGetEquipName 6](#_Toc26454140)

[spGetMonsterFromDungeon 6](#_Toc26454141)

[spGetMonsterName 7](#_Toc26454142)

[spGetPlayerDefense 7](#_Toc26454143)

[spGo 7](#_Toc26454144)

[spHasActiveQuest 7](#_Toc26454145)

[spIsCharacterNude 7](#_Toc26454146)

[spIsOverweight 8](#_Toc26454147)

[spPlayerKilledMonster 8](#_Toc26454148)

[spRandomEquipStolen 8](#_Toc26454149)

[spRandomEvents 8](#_Toc26454150)

[spRandomItemByLevel 9](#_Toc26454151)

[spRandomLoot 9](#_Toc26454152)

[spRandomQuestByLevel 9](#_Toc26454153)

[spSetStatusCharacter 9](#_Toc26454154)

[spUpdateCharacterHealth 10](#_Toc26454155)

[spUpdateCharacterLoot 10](#_Toc26454156)

[spUpdateCharExp 10](#_Toc26454157)

[spUpdateCharLevel 10](#_Toc26454158)

[spUpdateGPCharacter 11](#_Toc26454159)

[spUpdateMonsterHealth 11](#_Toc26454160)

[spUpdateStealEquip 11](#_Toc26454161)

[Fonctions tables 12](#_Toc26454162)

[Fonctions scalaires 13](#_Toc26454163)

# Procédures stockées

### spAddCharacterLoot

Insère un loot donné dans la table de loot du personnage.

CharID : ID du personnage à qui ajouté le loot

lootID : ID du loot à ajouté

qty : quantité à ajouté

### spCharacterHasEquip

Vérifie si le personnage à de l’équipement.

CharID : ID du personnage à vérifié

Returns : 1 si à de l’equip, 0 si non.

### spCharLuckRoll

Utilise l’attribut “Luck” du personnage pour savoir s’il tombe sur un évènement spécial.

CharID : ID du personnage qu’on dont on veut évaluer la chance.

Returns : 1 si chanceux, 0 si non.

### spCharWisdomRoll

Utilise l’attribut “Wisdom” du personnage pour savoir s’il réussit à charmer le tavernier.

CharID : ID du personnage qu’on dont on veut évaluer la sagesse.

Returns : 1 si asser sage pour charmer, 0 si non.

### spCompleteDungeon

Sert à compléter un donjon (Supprimer la bataille en cours et mettre le stade de completion du donjon à 1)

CharID : ID du personnage dont on veut compléter le donjon actif.

### spDeleteAllCharacterLoot

Supprime tout les loot du personage. Sert lorsqu’on vend au Tavernier.

@id : ID du personnage dont on veut supprimer les loots.

### spDeleteRandomEquip

Supprime un équipement aléatoire du personnage s’il n’est pas nu.

CharID : ID du personnage dont on veut supprimer une pièce d’équipement.

DeletedEquipName OUTPUT : Nom de l’équipement qu’on a supprimé (pour affichage)

### spEquipNew

Permet d’équiper un item dont on reçoit l’ID en paramètre.

CharID : ID du personnage dont on veut équiper un nouvel item.

EquipID : ID de l’équipement que l’on veut mettre à notre personnage.

### spGenerateFight

Insère un monstre aléatoire dans la table «Fight » .

CharID : ID du personnage dont on veut associer à un combat.

MonsterID : ID du monstre que l’on veut combattre

MonsterCurrentHP : HP actuel du monstre.

### spGenerateMonsterHit

Génère un coup du monstre en prenant en compte l’attaque du monstre et la défense totale du joueur.

CharID : ID du joueur pour avoir sa défense.

MonsterID : ID Du monstre pour avoir son attaque.

Returns : Le nombre de dégat du monstre

### spGeneratePlayerHit

Génère le coup du joueur en prenant en compte son attaque.

CharID : ID du joueur pour avoir sa défense.

MonsterID : ID Du monstre pour avoir son attaque.

Returns : le nombre de dégat du joueur.

### spGetCharacterAttack

Retourne l’attaque totale du joueur.

CharID : ID du joueur dont on veut obtenir l’attaque totale.

Returns : L’attaque totale du joueur.

### spGetCurrentWeight

Obtient la quantité de loot que possède le joueur à un moment donné.

CharID : ID du joueur dont on veut obtenir le poids de l’inventaire.

Returns : Le nombre de loot que le joueur a.

### spGetDiscountedValueEquip

Obtient le prix a rabais d’un équipement donné.

EquipID : ID de l’equipement dont on veut obtenir le prix réduit.

Returns : Prix réduit de l’équipement.

### spGetDungeonCompletion

Obtient le stade de complétion d’un donjon.

CharID : ID du personnage dont on veut vérifé le donjon.

DunID : ID du donjon à vérifié

Returns : 1 si le donjon est terminé, 0 si non.

### spGetEquipName

Obitent le nom de l’équipement donné.

EquipID : ID de l’equipement dont on veut obtenir le nom.

Name  OUTPUT : Nom de l’équipement donné.

### spGetMonsterFromDungeon

Obtient un monstre aléatoire dans une « rencontre » associé au donjon présentement actif.

DunID : ID du donjon dont n veut obtenir un monstre aléatoirement.

Returns : ID du monstre à combattre.

### spGetMonsterName

Obtient le nom du monstre donné en paramètre.

MonsterID : ID du monstre dont on veut obtenir le nom.

MonsterName OUPUT : Nom du monstre

### spGetPlayerDefense

Obtient la défense totale du joueur.

CharID : ID du joueur dont on veut obtenir la défense totale.

Returns : La défense totale du joueur.

### spGo

### spHasActiveQuest

Sert à savoir si le joueur à un donjon actif.

CharID : ID du joueur dont on veut savoir s’il a un donjon actif.

Returns : 1 si joueur à un donjon actif, 0 si non.

### spIsCharacterNude

Sert à savoir si le personnage est nu (a de l’equipement)

CharID : ID du personnage dont on veut avoir l’état (nu ou non).

Returns : 1 si nu, 0 si non.

### spIsOverweight

Sert à savoir si le personnage à atteint la limite de son inventaire.

CharID : ID du personnage dont on veut évaluer l’état de l’inventaire.

Returns : 1 si inventaire plein, 0 si non.

### spPlayerKilledMonster

Lorsque le joueur tue le monstre, vide la table de combat et update le nombre de kill du donjon actif.

CharID : ID du personnage qui a tué le monstre.

### spRandomEquipStolen

Sert à savoir si le goblin a finalement vole un equipement ou non.

CharID: ID du joueur qui s’est fait volé un equipement.

Returns : 1 si volé, 0 si non.

### spRandomEvents

Sert à choisir aléatoirement l’évent sécial.

Returns : 1-4 aléatoire, numéro de l’event.

### spRandomItemByLevel

Permet de choisir un item aléatoire tout dépendement du niveau du joueur.

level : Permet de retourner un item de se level.

Returns : ID de l’équipement choisit aléatoirement avec le level donné.

### spRandomLoot

Choisit aléatoirement un loot (Et oui, un poulet drop une écaille de dragon. C’est un poulet tueur de dragon, que voulez-vous.)

Returns : ID du loot obtenu de façon aléatoire

### spRandomQuestByLevel

Retourne un donjon selon le niveau du joueur.

CharID : ID du joueur à qui on veut insérer le donjon.

Level : niveau du donjon à sélectionné

### spSetStatusCharacter

Permet d’affecter un statut au joueur (Ou il est rendu dans l’aventure)

CharID : ID du joueur dont on veut changer le statut.

Status : Status qu’on veut affecter au joueur.

### spUpdateCharacterHealth

Retire le hit damage de la vie du joueur.

CharID : ID du joueur dont on veut changer la vie.

HitDamage : Dégat qu’on veut retirer de la vie du joueur.

### spUpdateCharacterLoot

Update le loot du joueur. Augmente la quantité si existant, sinon l’ajoute au loot.

CharID : ID du personnage dont on veut vérifier le loot

LootID : ID du loot qu’on veut ajouté au personnage.

Qty : Quantité obtenu du loot.

### spUpdateCharExp

Update l’expérience du personnage avec une valeur donnée.

CharID : ID du personnage dont on veut changer l’expérience.

monsterExp : Expérience obtenue en tuant le monstre

currentExp : Expérience actuelle du personnage.

### spUpdateCharLevel

Update le niveau du joueur avec un niveau donné.

CharID : ID du personnage

Level : Level actuel du joueur

### spUpdateGPCharacter

Update le gold du personnage a un montant donné.

Id : ID du personnage dont on veut updater l’or.

Gold : montant d’or qu’on veut mettre au joueur.

### spUpdateMonsterHealth

Actualise la vie du monstre après que le joueur l’ai frapper.

MonsterID : ID du monstre a updater.

HitDamage : Valeur du coup du joueur.

### spUpdateStealEquip

Met l’équipement volé du joueur a « null », si équipement a été volé.

CharID : ID du personnage dont on veut actualiser l’inventaire.

eventID : ID de quel équipement on doit volé.

# Fonctions tables

### fnGetActiveDungeonName

Obtient le nom du donjon actif pour un joueur donné.

CharID : ID du joueur dont on veut vérifier le nom du donjon actif.

Returns : Le nom du donjon présentement actif selon le ID du joueur.

### fnGetCharacterEquiped

Obtient le nom des équipements du joueur donnée.

CharID : ID du joueur dont on veut obtenir le nom des équipements.

Returns : Table contenant le nom de tout les équipements, nul si vide.

### fnGetCharacterGold

Obtient le gold du joueur donné.

CharID : ID du joueur dont on veut le gold

Returns :Nombre d’or du joueur

### fnGetCharacterHP

Obtient le HP actuel du joueur donné.

CharID : ID du joueur dont on veut obtenir le HP

Returns : HP actuel du joueur

### fnGetCharacterInfos

Obtient tous les informations du joueur

CharID : ID du joueur dont on veut obtenir les infos.

Returns : Table contenant tous les infos du personnage

### fnGetCharacterLoot

Obtient tout les loots du personnage

### fnGetCompletedDungeons

Obtient la liste des donjons complétés.

CharID : ID du joueur dont on veut la liste des donjons complétés

Returns : Table contenant le nom des donjons complétés, puis leur nombre de complétion.

### fnGetDungeonKC

Obtient le nombre de kill actuel du donjon

CharID : ID du joueur dont on veut le nombre du kill du donjon actif.

Returns : Table contenant le nombre de kill du donon actif

### fnGetNameOneLoot

Obtient le nom d’un loot

LootID : ID du loot dont on veut obtenir le nom.

# Fonctions scalaires

### fnCanBuyEquip

Sert à savoir si le joueur peut acheter l’equipement proposé ou non.

CharID : ID du joueur pour évaluer s’il peut acheter l’équipement

EquipDI : Equipement a evaluer

Returns : 1 si peut acheter, 0 sinon.

### fnCheckBetterEquip

Sert à savoir si l’équipement proposé est meilleur que ce que le joueur a déjà.

CharID : ID du joueur

newEquipID : ID de l’equipement à évaluer

Returns 1 si meilleur, 0 sinon.

### fnCheckIfHasLoot

Sert à savoir si le joueur à du loot ou non

CharID : ID du joueur à évaluer

Returns : 1 si oui, 0 si non.

### fnGetEquipName

Obitent le nom de l’équipement donné.

EquipID : ID de l’equipement à évaluer

Returns : Nom de l’équipement

### fnGetEquipType

Obtient le type de l’équipement donné

EquipID : ID de l’équipement à évaluer

Returns : ID du type de l’équipement

### fnGetGPCharacter

Obtient l’or du personnage

CharID : ID du joueur qu’on veut avoir l’or

Returns : le nombre d’or du joueur actuel.