

MODELO RELACIONAL

Map = { mapID }

mapID: número de identificação de um mapa que pode ser utilizado por um jogador.

HaveMapRegion = { mapID, coordinates }

coordinates: código de identificação das coordenadas do jogador;

Region = { coordinates^{*1}, biome, dangerLevel, monument^{*1o} }

dangerLevel: nível de perigo que a região apresenta;

Biomes = { biomesID^{*2}, coordinates^{*1}, resourceAbundance, resourceAvailability, type, climate^{*3} }

resourceAbundance: quantidade de recursos que um bioma pode apresentar — em questão de estar abundante ou não — a um ou mais jogadores;

resourceAvailability: apresenta se há ou não recursos no bioma;

Flora = { flora, biomes^{*2} }

Fauna = { fauna, biomes^{*2} }

Climate = { climateID^{*3}, temperature, event, statusEffect, visibility }

Characters = { charactersID^{*4}, name, position, characterModel, climate^{*3}, type, item }

GatherYield = { character^{*4}, gatherYield }

EnterCombatCharacters = { enterCombatCharactersID^{*5}, firstCharacter^{*4}, secondCharacter^{*4} }

firstCharacter: código do primeiro jogador no combate;

secondCharacter: código do segundo jogador no combate;

CombatLog = { enterCombatCharacters^{*5}, indexLog, log }

PlayerCharacters = { charactersID^{*4}, hydration, poisoned, hunger, type, equippedItems1, equippedItems2, equippedItems3, equippedItems4, equippedItems5, backpack^{*13} }

RecruitableCharacters = { charactersID^{*4}, specialization, recruited }

MainCharacter = { charactersID^{*4}, owner^{*6} }

RespawnLocation = { ownerID^{*6}, description, timer, X, Y }

NPCs = { charactersID^{*4}, isAgressive, aggroRange, enemyGrade, type }

isAgressive: código que apresenta se um NPC é agressivo ou não.

Animals = { charactersID^{*4}, sound, modelType }

Scientists = { charactersID^{*4}, hasDialogue }

hasDialogue: código que apresenta se existe a possibilidade de diálogo com o 'scientist'.

DialogueText = { dialogue, character^{*4} }

Items = { itemsID^{*7}, stackSize, lootGrade^{*8}, name, type, quantity, durability, craftable, type, backpack^{*13}, character^{*4} }

craftable: código que apresenta se um item pode ser criado a partir da junção de outros itens por parte dos jogadores.

Ingredients = { ingredient, items^{*7}, lootGrade^{*8} }

Weapons = { itemsID^{*7}, type }

Melee = { itemsID^{*7}, canBeThrown, fleshGatherRate, oreGatherRate, treeGatherRate }

Ranged = { itemsID^{*7}, recoil, attackRange, ammoCapacity, modSlots, fireMode, fireRate, accuracyModifier }

Consumables = { itemsID^{*7}, instantHeal, healOverTime, hidratationYield, bleedingYield, radiationYield, poisonYield, hungerYield, vomitChance, type }

hidratationYield: código que apresenta a quantidade de hidratação que um consumível oferece ao personagem.

bleedingYield: código que apresenta a quantidade de cura de sangramento que um consumível oferece ao personagem.

radiationYield: código que apresenta a quantidade de redução ou aumento de radiação que um consumível oferece ao personagem.

poisonYield: código que apresenta a quantidade de dano ou cura de envenenamento que um consumível oferece ao personagem.

hungerYield: código que apresenta a quantidade de redução de fome que um consumível oferece ao personagem.

vomitChance: código que apresenta a chance que um consumível oferece ao personagem de sofrer a êmese.

Teas = { itemsID^{*7}, statusUpgradeType, upgradePercentage }

Clothing = { itemsID^{*7}, coldResistance, radResistance, explosionResistance, meleeResistance, rangedResistance, biteResistance, equipmentSlot, wetResistance }

coldResistance: valor que representa a resistência ao frio concedida pelo equipamento

radResistance: valor que representa a resistência à radiação concedida pelo equipamento.

explosionResistance: valor que representa a resistência à dano de explosão concedida pelo equipamento.

meleeResistance: valor que representa a resistência à dano corpo-a-corpo concedida pelo equipamento.

rangedResistance: valor que representa a resistência à dano a distância concedida pelo equipamento.

biteResistance: valor que representa a resistência à dano de mordidas concedida pelo equipamento.

wetResistance: o item concede resistência ao status de molhado. IE: Roupas de radiação, roupa de mergulho.

Components = { itemsID^{*7} }

Resources = { itemsID^{*7}, isPrimary }

DropCharactersItems = { dropCharactersItemsID, item^{*7}, lootGrade^{*8}, character^{*4} }

PlayerCharactersGeneratesItem = { character^{*4}, items^{*7} }

WeaponsAreComposedOfComponentsResources = { components^{*7}, resources^{*7}, weapons^{*7} }

ConsumablesAreComposedOfComponentsResources = { components^{*7}, resources^{*7}, consumables^{*7} }

ClothingAreComposedOfComponentsResources = { components^{*7}, resources^{*7}, clothing^{*7} }

ResourceNodes = { resourceNodesID^{*9}, nodeType, maxYield, durabilityDamage, biomes^{*2} }

ResourceNodesGenerateItems = { resourceNodes^{*9}, item^{*7} }

Monuments = { name*¹⁰, monumentSize, lootGrade, enemyGrade }

regionsMonuments = { name*¹⁰, regionMonument }

Structures = { structureID*¹¹, monument*¹⁰ }

LootCrates = { lootCratesID*¹², grade }

StructuresContainsLootCrates = { structure*¹¹, lootCrates*¹² }

Party = { partyID, character*⁴, capacity }

partyID: código do grupo de jogadores que estão jogando junto.

Backpack = { ownerID*¹³, slot1, slot2, slot3, slot4 }

ownerID: código do dono da 'backpack'.

slotX: A mochila terá uma quantidade de slots determinada pelo valor de totalSlots, cada slot será um atributo diferente, possuindo um valor específico para o ID item que o ocupa ou 0 caso não haja item.