

Arquivo: Create.py

Trabalha Junto com a classe Menu-Terminal

Hierarquia de Herança em cima de

Classe Balão de texto puchado Terminal(ing/por);

Variavel: Linguagem = bool. (0/1)

Variavel: Valor_do_Save = (1 a 4)

+Deletar: Remover acessando Local Salvo.

+Criar Conta: Variavel Local acessa 5

Funções:

Check_Point(0)

Nome(string)

Genero(string(len(10))

Status(strings)

Set de informações(strings)

Mochila(strings)

Salve Informações (.TXT):

Salvar ((Local)+Valor_do_Save+'.txt'

Arquivo: LinkStart.py

Importa Config.py

Importa Menu.py

Importa Terminal.py

Importa Ballon.py

Importa Info_Person.py

IF __NAME__ == __MAIN__;

Language: Linguagem (0/1)

Variavel-Local: Dictionary de informações do personagem.

Varre verificação da integridade dos arquivos: Config.py.

Acessa Menu de informações:Menu.py

Carrega o Ponto da História: Point.py

Salvando no final e dando a opção Continuar ou voltar para Menu.

Arquivo: Point.py

Trabalha Junto com a classe Menu-Terminal Hierarquia de Herança em cima de

Ballon: Classe Balão de texto puchado Terminal(ing/por);

Damage:Classe Engloba Ação de Dano, Recebimento de Dano e Se morre.

UP: Classe Responsável por Upar Personagem, Status, status+itens

Set: Classe Responsável por Verificar Itens do Set e Mochila do personagem.

MenuTerminal>(Ballon(ING/PT) > Terminal.py

(damage,up,Set) >indor_Person.py

Classe Local : PointHistory(Herda Damage,Up,Set,MenuTerminal)

__STR__()DuckTypeQuack.

verifica Lista Dict_Info_Usuario: Pucha Valor no ponto da História e Narra...

Interagir Com Monstros Gerados No Local do Arquivo,Itens,Missões,Equipamentos,Vida,Danos... Salvar e Retornar Para o Menu ou continuar quando Finalizar.

Arquivo: Menu.py

Trabalha Junto com a classe Menu-Terminal Hierarquia de Herança em cima de

Ballon: Classe Balão de texto puchado Terminal(ing/por);

Variavel: Language,NewGame

Classe Local : Person(Herda de Terminal)>Ballon

__STR__()DuckTypeQuack.

Func Open_and_read: : Abre o arquivo e salva numa list[information]

Func Open_and_Write: : Abre o arquivo apagando ele e incrementando Novos valores.

Dict_all: Variavel Local List Zipa com valor Local List e junta tudo em um Dictionary de informações.

Func Alocale_Info : Função pega todos valores do acess_Info que se limita ao Personagem até antes das informações de sua mochila, pois no jogo pode haver 3 tipos de mochila. Então Manipular listas diferentes conforme o nome da mochila e definir tamanho a ser lido.

Save_Info > Saved_Info> Open_and_read: Cria uma Cópia rapida do Dicionario de Informações geral do personagem averigua tamanho da mochila e envia para ser filtrado e lido as casas correspondentes com suas Keys para não haver lixo de informação incrementada no save, usando Backpack_search. Por Ultimo Saved_infor faz a limpeza nas linhas correspondentes onde a infor puchada Bate com sua lista e retorna no final para um save igual conforme antes salvo. Keys iguais e diferentes para manter a complexibilidade

Acess_info: Usa Open_and_read para verificar tamanho da mochila e salva informação para ser incrementada sem gerar lixo e nem cair em local inapropriado ou None.

Classe Local : Set(Herda de Person, Terminal)

__STR__()DuckTypeQuack.

glogal_itens_name : dict local contendo nome do item e suas respectivas keys

glogal_itens_status_int :dict local contendo Power magic e suas respectivas keys

glogal_itens_status_for :dict local contendo Foece power e suas respectivas keys

glogal_itens_status_def: dict local contendo defesa e suas respectivas keys

glogal_itens: itens de jogo com suas respectivas keys

take: Pega um item e joga direto primeiro no Set do personagem, se n... enviar para BP_direct que fica responsável de saber se vai incrementar o item na backpack ou não, Apenas se o SET não for atendido, ambos retornam Dictionary de informações do usuario.

access_set>access_set1>comparation:Efeito Dominó: access_set gera uma lista de keys e passa pra proxima access_set1 apenas Set itens equipaveis Um valor local correspondente a casa do item selecionado.

access_set1>recursive>comparation: Responsável por mostrar lista de itens da backpack e seleciona usando recursive. e para sair somente com comando de saída ou quando selecionar item voltando para comparison.

Comparation>update_status_item>save_info: Verifica se os itens é trocável, se sim altera for/int/def do personagem conforme item no update_status_item. Finaliza dando save_info e retorna Dictionary de informações, caso contrário mostra que não pode.

broken_item: Selecionar o local do set e destroi o item trocando por str("None")

SET: Ele joga automaticamente o item para o Set pegado ex: Capacete na cabeça, shorte nas pernas... Portanto que não tenha nada lá altera for/int/def do personagem conforme item no update_status_item. Finaliza dando save_info e retorna Dictionary de informações

access_backpack>len_backpack>select_item: EFEITO DOMINÓ. Verifica se está acessando em uma missão ou não, Mostra os itens na backpack no len_backpack, e interage com os itens, definidos separadamentes pela sua primeira letra da key do valor do glogal_itens.../name/int/for/status/def/ Podendo Excluir, Usar,Acessar o Set. Pelo access_set,BP_Dell,BP_Use,Loop_search

list_status>List_status: Lista os Status do personagem.

update_status_item(pega dois itens que está sendo trocado decrementa do item atual) Mesmo sendo 'None' e incrementa Def/For/int de cada um deles respectivos no personagem e em seguida da save_info.

BP_Dell/BP_Use: Verifica o tipo de item e utiliza ou deleta

Loop_search: Loop de busca verifica se é none e retorna para dizer que o item é none

UP : (Herda Set Terminal):

__STR__()DuckTypeQuack

Up: vê o level, Herda Lista do Level Compara XP atual e verifica se pode upar

STS: responsavel por filtrar informação para Up e selecionar apenas verificação de mana e HP

Next_Point: Avança informação do dicionario do personagem para proximo

Returnpoint: retorna pro ponto inicial da história quando morto

Damage(UP,Set,MenuTerminal):

Opções_Action menu: Define se ataca, acessa Backpack, retorna 2 para interações.Status e Run para tentar fugir como valor 5.

Damage: Dano fisico.

Damage_int: ve se tem mana e se tiver mais que possa usar, ataque e decremente da mana.

sense_damage

life_mana: se tiver sem mana, avise, se tiver sem vida vá para death_return

death_return: leva para returnpoint()

LoadGame(1a4)

+Entrar No Config> N ter retorno None:

+Salve Informações: 'Config .TXT' 0/1 (Linguagem: Ing/Pt)

Arquivo: Menu.py

Trabalha Junto com a classe Menu-Terminal Hierarquia de Herança em cima de

Ballon: Classe Balão de texto puchado Terminal(ing/por);

Variavel: Language,NewGame

Classe Local : New Game(Herda Create_Char,Terminal)>Ballon

__STR__()DuckTypeQuack.

Select: Pucha Verificação da pasta em uma função:

Vasio =(GREEN)

Há =(VERMELHO)

Pucha New Game: Variavel Local e Retorna.

SE 0 = Voltar.

Se existir arquivo: Opção Deletar.

Se não> Herda Classe Create_account de Create_char com Self.

retorna posição local (1-4)

Variavel Local para selecionar espaço

Variavel Local Para armazenar local desejado

Classe Local : Load Game(Herda Terminal)>Ballon

__STR__()DuckTypeQuack

Load: Verifica >Pasta e Retorna somente (1-a-4) existente.

Verigicar Pasta : Procura arquivos salvos como > (Locais + de 1-a-4 + '.txt')

se existir: Mostrar em Ciano. Se Não: Mostrar em vermelho.

se Existir: incrementa em uma variável lista para ser acessada Somente.

Classe Local : Configurações(Herda Terminal,Configurações)>Ballon.

__STR__()DuckTypeQuack

Reconfigurar: Reconfigure Language-> Salvar e ser lido TXT

Classe Local: Menu Inicial (Herda Terminal e as classes a cima)>Ballon.

__STR__()DuckTypeQuack

Opções_Menu: Do_it Carregar classes a cima ou Retornar!: Voltar valor:

NewGame(1a4)

LoadGame(1a4)

+Entrar No Config> N ter retorno None:

+Salve Informações: 'Config .TXT' 0/1 (Linguagem: Ing/Pt)

Arquivo: ballon.py

Trabalha Junto com a classe Menu-Terminal Hierarquia de Herança em cima de

Classe Balão de texto puchado Terminal(ing/por);

Variavel-Local: Dictionary de informações.

+Func: __STR__:Ducktype que recolhe informações gerais em STR de todos os arquivo e acessa Func Ballon, Recolhendo Texto contido Na sua Dictionary e selecionadno 0 / 1 definido pela language.

+Func: Ballon> Mostra um dictionary de textos englobado em um só local

Arquivo: Info_List.py

Trabalha Junto com a classe Menu-Terminal Hierarquia de Herança em cima de

Classe InfoList; Retorna Valor Dict acessado pela Key Enviada

Variavel-Local: Linguagem = int.0-a-30.

Variavel-Retorno: Int(contido no Dictionary)

+Func: LVL_UP > Acessa Dictionary: Pega valor do XP MAXIMO CONDIZENTE AO LVL ATUAL

0-A-30

+RETORNA VALOR DO XP: Return Variavel Local

Arquivo: Config.py

Trabalha de forma Independente.

Classe Única, Config_Menu;

+Func: Conf_Language.

+Criar Conta: Se não houver pasta do local onde fica o ambiente salvo TXT. a Função: Cria Pasta: save

Cria Arquivo:confing Com string com informação para 0. Condizente a lingua inglesa.

+Func: Reconfig_Language(0/1):

Responsável por TROCAR valor da STRING para 0/1 e retornar ela mesma quando alterada no arquivo TXT. Verificando caminho com OS.MAKEDIR

+Func: Loading_Lang.

Responsável por corrigir caso haja alteração imprópria manual no arquivo, caso venha dar um bug ao inicializar o programa. Corrige a String e Compila na forma correta.