PROJETO DE P.O.O 2022 Classe Balão de texto puchado Terminal(ing/por); Variavel: Linguagem = bool. (0/1) Variavel: Valor do Save = (1 a 4) UML DO TRABALHO +Deletar: Remover acessando Local Salvo. +Criar Conta: Variavel Local acessa 5 Check_Point(0) Nome(string) Genero(string(len(10)) PROJETO DE JOGO.PY Status(strings) Set de informações(strings) Mochila(strings) Salve Informações (.TXT): Salvar ((Local)+Valor_do_Save+'.txt' Trabalha Junto com a classe Menu-Terminal Hierarquia de Herança em cima de Arquivo: LinkStart.py Ballon: Classe Balão de texto puchado Terminal(ing/por); Importa Config.py Importa Menu.py Importa Terminal.py Variavel: Language,NewGame Frabalha Junto com a classe Menu-Terminal Hierarquia de Heranca Importa Ballon.py Classe Local : Person(Herda de Terminal)>Ballon Importa Info_Person.py __STR__()DuckTypeQuack. Func Open_and_read: : Abre o arquivo e salva numa list[information] llon: Classe Balão de texto puchado Terminal(ing/por); amage:Classe Engloba Ação de Dano, Recebimento de Dano e Se morre. IF __NAME__ == __MAIN__; Func Open_and_Write: : Abre o arquivo apagando ele e incrementando Novos valores. JP: Classe Responsável por Upar Personagem, Status, status+itens et: Classe Responsável por Verificar Itens do Set e Mochila do personage Dict_all: Variavel Local List Zipa com valor Local List e junta tudo em um Dictionary de Language: Linguagem (0/1) lenuTerminal>(Ballon(ING/PT) > Terminal.py Variavel-Local: Dictionary de informações do amage,up,Set) >indor_Person.py personagem Func Alocale_Info : Função pega todos valores do acess_Info que se limita ao Personagem Varre verificação da integridade dos até antes das informações de sua mochila, pois no jogo pode haver 3 tipos de mochila. Então lasse Local: PointHistory(Herda Damage,Up,Set,MenuTerminal) arquivos: Config.py. Manipular listas diferentes conforme o nome da mochila e definir tamanho a ser lido. _STR__()DuckTypeQuack. Acessa Menu de informações:Menu.py Save_Info > Saved_Info> Open_and_read: Cria uma Cópia rapida do Dicionario de Carrega o Ponto da História: Point.py verifica Lista Dict_Info_Usuario: Pucha Valor no ponto da História e Narra... Informações geral do personagem averigua tamanho da mochila e envia para ser firltrado e Salvando no final e dando a opção Continuar Interagir Com Monstros Gerados No Local do lido as casas correspondentes com suas Keys para não haver lixo de informação ou voltar para **Menu.** Arquivo,Itens,Missões,Equipamentos,Vida,Danos... Salvar e Retornar Para o ncrementada no salve, usando Backpack_search. Por Ultimo Saved_infor faz a limpesa nas Menu ou continuar guando Finalizar. inhas 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. Arquivo: Terminal.py Classe Local: Set(Herda de Person, Terminal) Trabalha SOBRE Hierarquia de Herança em _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 Foce power e suas respectivas keys **A MAIN** logal_itens_status_def: dict local contendo defesa e suas respectivas kevs Classe Balão de texto(ing/por): glogal_itens: itens de jogo com suas respectivas keys Classe Lista de informação do personagem É AQUI Classe Dicionario de informações do LVL. 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 Variavel: Linguagem = bool. (0/1) __STR__: Balão de informação de acess_set>acess_set1>comparation:Efeito Dominó: acess_set gera uma lista Texto(Tipo_da_linguagem) de keys e passa pra proxima acess_set1 apenas Set itens equipaveis Um TypeDUCK(Exibir um texto recebido) valor local correspondente a casa do item selecionado. acess_set1>recursive>comparation: Responsável por mostrar lista de itens da + Lista do Level: Balão sobre o xp do Level backpack e seleciona usando recursive. e para sair somente com comando de TypeDUCK(Recebe a experiencia total que saida ou quando selecionar item voltando para comparation. deve ser alcançada do respectivo level). 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. **+Limpesa de tela:** Caso seja Linux ou Finaliza dando **save_info** e retorna **Dictionary de informações**, caso contrário Windows... Tratado aqui. mostra que **não pode.** +Configuração_de_linguagen: broken_item: Selecionar o local do set e destroi o item trocando por Ver se o arquivo está str("None") salvo(Tipo_da_linguagem) se n: Criar um Verificar o valo interno é 0/1 para inglês e SET: Ele joga automaticamente o item para o Set pegado ex: Capacete na cabeça, shorte nas português. 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 Arquivo: Menu.py Trabalha Junto com a classe Menu-Terminal Hierarquia de Herança acess_backpack>len_backpack>select_item: EFEITO DOMINÓ. Verifica se está acessando em cima de 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 acess_set,BP_Dell,BP_Use,Loop_search Ballon: Classe Balão de texto puchado Terminal(ing/por); list_status>List_status: Lista os Status do personagem. Variavel: Language,NewGame **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 Classe Local : New Game(Herda Create_Char,Terminal)>Ballon seguida da **save_info.** _STR__()DuckTypeQuack. Select: Pucha Verificação da pasta em uma função: BP_Dell/BP_Use: Verifica o tipo de item e utilisa ou deleta Há = (VERMELHO) Loop_search: Loop de busca verifica se é none e retorna para dizer que o item é none Pucha New Game: Variavel Local e Retorna. UP : (Herda Set Terminal): Se **existir arquvo: Opção Deletar.** _STR__()DuckTypeQuack Se não> Herda Classe Create_account de Create_char com Self. **Up:** vê o level, Herda **Lista do Level** Compara XP atual e verifica se pode upar retorna posição local (1-4) STS: responsavel por filtrar informação para Up e selecionar apena verificação de mana e HP /ariavel Local para selecionar espaço Variavel Local Para armazenar local desejado Next_Point: Avança informação do dicionario do personagem para proximo Arquivo: Config.py **Returnpoint:** retorna pro ponto inicial da história quando morto Trabalha de forma Independente. Damage(UP,Set,MenuTerminal): Classe Única, Config_Menu; lasse Local : Load Game(Herda Terminal)>Ballon STR__()DuckTypeQuack Opções_Action menu: Define se ataca, acessa Backpack, returna 2 para _oad: Verifica >Pasta e Retorna somente (1-a-4) existente. interações, Status e Run para tentar fugir como valor 5. Arquivo: Info_List.py Verigicar Pasta : Procura arquivos salvos como > (Locais + de 1-a-4 + '.txt') Arquivo: ballon.py Trabalha Junto com a classe Menu-Terminal se existir: Mostrar em Ciano. Se Não: Mostrar em vermelho. +Func: Conf_Language. Damage: Dano fisico. Hierarquia de Herança em cima de Trabalha Junto com a classe Menu-Terminal se Existir: incrementa em uma variável lista para ser acessada Somente. +Criar Conta: Se não houver pasta do local Hierarquia de Herança em cima de onde fica o ambiente salvo **TXT.** a **Função**: Damage_int: ve se tem mana e se tiver mais que possa usar, ataque e Cria Pasta: save Classe Local : Configurações(Herda Terminal,Configurações)>Ballon. decremente da mana. Classe InfoList; Retorna Valor Dict acessado pela Cria Arquivo:confing Com string com _STR__()DuckTypeQuack Key Enviada Classe Balão de texto puchado Terminal(ing/por); informação para **0**. Condizente a **lingua** Reconfigurar: Reconfigure Language.-> Salvar e ser lido TXT sense_damage inglêsa. life_mana: se tiver sem mana, avise, se tiver sem vida vá para death_return Variavel-Local: Linguagem = int.0-a-30. Variavel-Local: Dictionary de informações. Classe Local: Menu Inicial (Herda Terminal e as classes a cima)>Ballon. +Func: Reconfig_Language(0/1): Variavel-Retorno: Int(contido no Dictionary) __STR__()DuckTypeQuack Responsável por **TROCAR** valor da **STRING** death_return: leva para returnpoint() Opções_Menu: Do_it Carregar classes a cima ou Retornar!: Voltar para **0/1** e retornar ela mesma guando alterada no arquivo TXT. Verificando caminho com **+Func:__STR__:Ducktype** que recolhe NewGame(1a4) OS.MAKEDIR +Func: LVL_UP > Acessa Dictionary: Pega informações gerais em **STR** de todos os LoadGame(1a4) valor do XP MAXIMO CONDIZENTE AO LVL LoadGame(1a4) arquivo e acessa Func Ballon, Recolhendo +Entrar No Config> N ter retorno None: +Entrar No Config> N ter retorno None: +Func: Loading_Lang. ATUAL **+Salve Informações: 'Config .TXT'** 0/1 (Linguagem: Ing/Pt) Texto contido Na sua **Dictionary** e Responsável por **corrigir** caso haja **alteração +Salve Informações: 'Config .TXT'** 0/1 (Linguagem: Ing/Pt) 0-A-30 selecionadno 0 / 1 definido pela language. imprópria manual no arquivo, caso venha dar +RETORNA VALOR DO XP: Return Variavel um **bug ao inicializar o programa.** Corrige a Local +Func: Ballon> Mostra um dictionary de String e Compila na forma correta. textos englobado em um só local

Arquivo: Create.py

Trabalha Junto com a classe Menu-Terminal

Hierarquia de Herança em cima de