Pagina Inicial

=============================================================================

.---. . .

\\_\_\_ ,-. . ,-. ,-. ,-. ,-. | , . |-. ,-.

\ ,-| | | | | | | | | | |< | | | | |

`---' `-^ ' `-| `-' ' ' `-' ' ` ' ^-' `-'

,|

`'

Saigo no kibō

1) /Anotacoes => exemplos e ideias?

1) /links => Links úteis

2) /codigo => O código

3) /wepons => As armas em ASCII

stream == fluxo de dados

Anotações

NULL = Nulo

EOF = End Of File

!feof(pont\_do\_arquivo);

FOPEN\_MAX = define quantidade maxima de arquivos que rodam ao msm tempo

stdin = entrada padrão = teclado

stdout = saida padrão = tela

stderr = saida de erro padrão = tela

fpos\_T falar sobre

Criar arquivo:

FILE \*file

file = fopen("nome\_do\_arquivo.txt","read(r),write(w),(a)modificar ");//Caminho do arquivo(opcional), nome\_arquivo.txt(pode mudar o txt), o que fazer com ele.

fprintf(file,"Pão de batata");//Primeiro argumento o ponteiro, depois escrever.

fclose(file);//Fechar o arquivo

Exemplo de arquivo 2:

FILE \*file

file = fopen("numeros.txt","r");//Arquivo que contem os inteiros 10 20 30.

if(file == NULL){

printf("Arquivo nao aberto");

getchar();

return 0;//Sair do programa.

}

int x,y,z;

fscanf(file,"%d %d %d",&x, &y, &z);

fprintf("%d %d %d",x,y,z);

fclose;

return 0;

======================================================================================================================================

ESCREVER:

int main (){

FILE \*pont\_Arqv;

// pont\_Arqv = fopen("C:\\Users\\Philippe de Paula\\Desktop\\teste.txt","w"); r - ler, w - escrever, a - alterar

pont\_Arqv = fopen("textoArquivo.txt","w"); // r - ler, w - escrever, a - alterar

fprintf(pont\_Arqv," Mensagem para o arquivo. \n");

fclose(pont\_Arqv);

return 0;

}

======================================================================================================================================

LER:

int main (){

char frase[100];

int x,y,z;

FILE \*pont\_Arqv;

pont\_Arqv = fopen("numero.txt","r"); // r - ler, w - escrever, a - alterar

// pont\_Arqv = fopen("numeros.txt","r");

if(pont\_Arqv == NULL){

printf(" ERRO: Arquivo nao pode ser aberto, encerrando progama . . . \n");

exit(1);

}

fscanf(pont\_Arqv, "%d %d %d",&x, &y, &z);

printf("%d %d %d \n", x, y, z);

fclose(pont\_Arqv);

pont\_Arqv = fopen("textoArquivo.txt","r");

if(pont\_Arqv == NULL){

printf(" ERRO: Arquivo nao pode ser aberto, encerrando progama . . . \n");

exit(1);

}

while(fgets(frase,100,pont\_Arqv) != NULL){

printf(" %s ",frase);

}

fclose(pont\_Arqv);

return 0;

}

======================================================================================================================================

ALTERAR/ACRESCENTAR :

#include <stdio.h>

#include <stdlib.h>

int main (){

char frase[] = "Segunda coisa adicionada \n";

char caracter = '6';

FILE \*pont\_Arqv;

pont\_Arqv = fopen("textoArquivo.txt","a");

if(pont\_Arqv == NULL){

printf(" ERRO: Arquivo nao pode ser aberto, encerrando progama . . . \n");

exit(1);

}

fprintf(pont\_Arqv, " Acrescentei com alteracao \n");

fputs(frase, pont\_Arqv);

fputc(caracter,pont\_Arqv);

fclose(pont\_Arqv);

system("pause");

return 0;

}

======================================================================================================================================

\_ . \_

(\_\_\_\_\_\_\_.-´ `-.\_\_\_\_\_\_\_)

nome XYZ

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

(\_ `-. .-´ \_)

'

ENDGAME

No fim do jogo "X" , "A" e "B" em meio ao enfrentamento contra "W" , "B" resolvi trai-los em meio a batalha "A" e gravemente ferido por "B" após isso "X" e "B" entram em uma batalha enquanto isso "C" e "D" lutam contra "W" , mas por serem fracos demais, tomam um sacode do "W" após isso "X" se vê em desvantagem , já que de um 3v2 passa para um 1v2 e ainda possui o fator de que "C" e "D" correm risco de morte. Nesse momento "X" senti uma energia vinda dos céus , quando se da conta já havia derrotado "B" ao olhar para si mesmo se da conta que possui 6 asas e senti algo familiar sua verdadeira forma Serafim que havia despertado , por fim a luta contra "W" se inicia ao fim da árdua batalha "X" se aproxima de "A" já sem esperanças

2 saidas:

-> "X" ganha de "W" causando dano e "A" morre 'em seus braços'.

-> "X" ganha de "W" sem causar dano( nesse caso "W" lança um poder que eh repelido contra ele), e com a ajuda de "C" e "D" eles conseguem abrir mão de parte de seu poder para fazer com que "A" fique vivo.

-> Final Secreto if("X" == MATA\_TUDO"){

printf("X toma o lugar de "W" ");

}

Apos o fim dos créditos uma curte scene para a PART II.

Enredo

=============================================================================================================================================

Nome do game: PROJETO XYZ.

Gênero: Rpg, Aventura

Estilo:. Jogo de escolhas

Enredo Muito\_LOUCO

X = Protagonista

Raça: Humano

Classe: (Usuário se vira).

A = Vira Best Friend

Raça: Elfo

Classe: Arqueiro

B = Best Friend (Falsiane) na real não e friend.

Raça: Revenant (Senhor dos corvos conhecido por sua inteligencia geralmente são "fantasmas"). Fim do jogo ((Anti Serafim))

Classe: Assasino

C = Emburrado que não quer ajudar.

Raça: Valquíria

Classe: Guerreiros

D = Panelinha com o C(Entretanto mais sensato que o C).

Raça: Vanires ( ama magia acima de tudo)

Classe: Mago.

W = Vilão das trevas Trevosas.

Raça: Demônio

Classe: Arcanista

Batalha de 5 reinos contra "W". Os reinos decidem mandar seus cinco filhos (X,A,B,C,D) para confrontar "W"

Luta dos 5 reinos de várias raças e seus filhos contra um MAL MAIOR “W”. (5 raças o protagonista “X” e uma raça mais fraca, humano). Na verdade, no fim do game ele descobre que é um serafim.

“X” e seu grupo que continha uma “A”,” B”,”C”,” D” toma uma solva do grupo, manda a frase que tá boladão com geral. Perdi a memória, por algum motivo que ainda não sei qual, "X" decidiu refazer a jornada, entretanto solo, ao longo da jornada se encontra com um antigo integrante do seu grupo “A”, que se questiona como “X” poderia estar vivo , vendo que “X” não se lembra decide ficar de prontidão a espera de trair ele novamente ao longo da jornada “X” e vai se recordar pouco a pouco até quando se encontra com os outros figurantes do grupo que

CODIGO

=============================================================================

//#include <ctype.h>

//#include <locale.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//#include <time.h>

#include <windows.h>

typedef enum {

BLACK, BLUE, GREEN, CYAN, RED, MAGENTA, BROWN, LIGHTGRAY, DARKGRAY,

LIGHTBLUE, LIGHTGREEN, LIGHTCYAN, LIGHTRED, LIGHTMAGENTA, YELLOW, WHITE

} COLORS;

static int \_\_BACKGROUND = BLACK;

static int \_\_FOREGROUND = LIGHTGRAY;

textcolor (int color) {

\_\_FOREGROUND = color;

SetConsoleTextAttribute (GetStdHandle (STD\_OUTPUT\_HANDLE), color + (\_\_BACKGROUND << 4));

}

struct atributo {

int agi\_P,for\_P,int\_P;

};

struct personagem\_Principal {

char classe[30];

char genero[10];

char nome[30];

char raca[30]; // Humano

struct atributo atributos;

//struct inventario invent;

};

int selecao\_Seta(int num\_Finalizacoes);

void cria\_Personagem(struct personagem\_Principal\* perso\_Prin, int num\_Finalizacoes);

void gotoxy(int x, int y);

void informacoes();

void le\_Valida\_Classe(struct personagem\_Principal \*perso\_Prin, int num\_Finalizacoes);

void le\_Valida\_Genero(struct personagem\_Principal \*perso\_Prin);

void le\_Valida\_Nome(struct personagem\_Principal \*perso\_Prin);

void limpa\_Tela();

void mostra\_ASCII\_Classe();

void mostra\_Janela();

void mostra\_Menu();

void mostrar\_Menu\_Classes(int num\_Finalizacoes);

void mostrar\_Menu\_Classes\_Extra(int num\_Finalizacoes);

void preenche\_Classe( struct personagem\_Principal \*perso\_Prin, int num\_Classe);

void mostra\_Erro(int num\_Erro);

void set\_Dimensions();

int main () {

int tecla, num\_Finalizacoes = 0;

struct personagem\_Principal perso\_Prin;

do {

set\_Dimensions();

mostra\_Menu();

tecla = selecao\_Seta(num\_Finalizacoes);

switch(tecla) {

case 1: // New Game - Começar

cria\_Personagem(&perso\_Prin, num\_Finalizacoes);

// system("cls");

// printf(" %d ",tecla);

getchar();

getchar();

num\_Finalizacoes++; // APAGA SAPORRA

break;

case 2: // Continue - Continuar?

system("cls");

printf(" %d ",tecla);

getchar();

break;

case 3: // Informations - Informações

system("cls");

//printf(" %d ",tecla);

set\_Dimensions();

informacoes();

getchar();

break;

case 4: // How To Play - Como Jogar

system("cls");

printf(" %d ",tecla);

getchar();

break;

case 5: // Exit - Sair

system("cls");

printf(" %d ",tecla);

getchar();

break;

}

} while(1);

return 0;

}

void cria\_Personagem(struct personagem\_Principal\* perso\_Prin, int num\_Finalizacoes) {

char mostra\_Classe[100];

FILE \*pont\_Classe;

strcpy(perso\_Prin->raca,"Humano");

limpa\_Tela();

printf(" Create your character \n");

printf(" Name: ");

le\_Valida\_Nome(perso\_Prin);

limpa\_Tela();

printf(" Name: %s \n",perso\_Prin->nome);

printf(" Gender ('M'/'F'): ");

le\_Valida\_Genero(perso\_Prin);

limpa\_Tela();

printf(" Name: %s \n",perso\_Prin->nome);

printf(" Gender: %s \n",perso\_Prin->genero);

printf(" Race: %s \n",perso\_Prin->raca);

mostrar\_Menu\_Classes(num\_Finalizacoes);

printf(" Class: ");

le\_Valida\_Classe(perso\_Prin, num\_Finalizacoes);

getchar();

getchar();

limpa\_Tela();

printf(" Name: %s \n",perso\_Prin->nome);

printf(" Gender: %s \n",perso\_Prin->genero);

printf(" Race: %s \n",perso\_Prin->raca);

printf(" Class: %s",perso\_Prin->classe);

/\*

//mostra\_ASCII\_Classe(); ainda nao pronto

// Lembrar de colocar um time antes do jogo começar

\*/

}

void gotoxy(int x, int y) {

COORD p= {x,y};

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), p);

}

void mostra\_Janela() {

int pos\_Pixel;

textcolor(LIGHTCYAN); // LEMBRAR: MUDAR COR

gotoxy(0,0);

printf("%c",201); // Canto superior esquerdo

gotoxy(0,34);

printf("%c",200); // Canto inferior esquerdo

gotoxy(119,0);

printf("%c",187); // Canto superior direito

gotoxy(119,34);

printf("%c",188); // Canto inferior direito

for(pos\_Pixel = 1; pos\_Pixel < 34; pos\_Pixel++) {

gotoxy(0,pos\_Pixel);

printf("%c",186); // Linha Vertical Superior

gotoxy(119,pos\_Pixel);

printf("%c",186); // Linha Vertical Inferior

}

for(pos\_Pixel = 1; pos\_Pixel < 119; pos\_Pixel++) {

gotoxy(pos\_Pixel,0);

printf("%c",205); // Linha Horizontal Superior

gotoxy(pos\_Pixel,34);

printf("%c",205); // Linha Horizontal Inferior

}

textcolor(WHITE);

}

void mostra\_Menu() {

char frase\_Menu[100];

FILE \*pont\_Menu;

system("cls");

pont\_Menu = fopen("menu.txt","r");

if(pont\_Menu == NULL) {

mostra\_Erro(1);

exit(1);

} else {

while(fgets(frase\_Menu,100,pont\_Menu) != NULL)

printf("%s",frase\_Menu);

}

fclose(pont\_Menu);

mostra\_Janela();

}

int selecao\_Seta(int num\_Finalizacoes) {

char tecla = 'H';

int pos\_Y = 20, tecla\_Saida = 1;

int sair;

// Seta Cima -> H = 72

// Seta Baixo-> P = 80

textcolor(LIGHTGRAY );

gotoxy(43,20);

printf("%c%c%c%c%c",185,175,175,175,175); // Espada da Seleção

gotoxy(119,34);

do {

sair = 1;

fflush(stdin);

tecla = toupper(getch());

gotoxy(43,20);

printf(" ");

gotoxy(43,22);

printf(" ");

gotoxy(43,24);

printf(" ");

gotoxy(43,26);

printf(" ");

gotoxy(43,28);

printf(" ");

switch(tecla) {

case 'H':

pos\_Y -= 2;

if(pos\_Y < 20)

pos\_Y = 28;

tecla\_Saida--;

if(tecla\_Saida == 0)

tecla\_Saida = 5;

gotoxy(43,pos\_Y);

printf("%c%c%c%c%c",185,175,175,175,175); // Espada da Seleção

break;

case 'P':

pos\_Y += 2;

if(pos\_Y > 28)

pos\_Y = 20;

tecla\_Saida++;

if(tecla\_Saida == 6)

tecla\_Saida = 1;

gotoxy(43,pos\_Y);

printf("%c%c%c%c%c",185,175,175,175,175); // Espada da Seleção

break;

default:

gotoxy(43,pos\_Y);

printf("%c%c%c%c%c",185,175,175,175,175); // Espada da Seleção

}

gotoxy(119,34);

if(tecla\_Saida == 2 && num\_Finalizacoes == 0)

sair = 0;

// 13 e /n significa 'Enter'

} while(tecla != 13 || !sair);

textcolor(WHITE);

return tecla\_Saida;

}

void mostra\_Erro(int num\_Erro) {

fflush(stdin);

switch(num\_Erro) {

case 1:

printf(" ERROR 001: A file referring to the information to be executed was not opened, closing program . . . \n");

break;

case 2:

printf(" ERROR 002: Invalid Name! \n");

printf(" Enter name again: ");

break;

case 3:

printf(" ERROR 003: Invalid Gender! \n");

printf(" Enter gender again: ");

break;

case 4:

printf(" ERROR 004: Invalid Class! \n");

printf(" Enter class again: ");

break;

}

}

void limpa\_Tela() {

system("cls");

printf("\n");

}

void mostra\_ASCII\_Classe() {

// pont\_Classe = fopen("classe.txt","r");

// fclose(pont\_Classe);

}

void le\_Valida\_Nome(struct personagem\_Principal \*perso\_Prin) {

int pos\_Nome, tamanho\_Nome, tamanho\_Max = 29, valid;

do {

if(!valid)

mostra\_Erro(2);

valid = 1;

fflush(stdin);

gets(perso\_Prin->nome);

fflush(stdin);

tamanho\_Nome = strlen(perso\_Prin->nome);

if(tamanho\_Max > tamanho\_Nome) {

for(pos\_Nome = 0; pos\_Nome < tamanho\_Nome; pos\_Nome++)

if(!isalpha(perso\_Prin->nome[pos\_Nome]) && perso\_Prin->nome[pos\_Nome] != ' ')

valid = 0;

} else

valid = 0;

} while(!valid);

}

void le\_Valida\_Genero(struct personagem\_Principal \*perso\_Prin) {

char genero;

do {

scanf("%c",&genero);

genero = toupper(genero);

if(genero != 'M' && genero != 'F')

mostra\_Erro(3);

} while(genero != 'M' && genero != 'F');

if(genero == 'M')

strcpy(perso\_Prin->genero,"Masculino");

else

strcpy(perso\_Prin->genero,"Feminino");

}

void le\_Valida\_Classe(struct personagem\_Principal \*perso\_Prin, int num\_Finalizacoes) {

int num\_Classe;

do {

scanf(" %d",&num\_Classe);

if(num\_Classe < 1 || num\_Classe > (num\_Finalizacoes + 1) \* 5)

mostra\_Erro(4);

} while(num\_Classe < 1 || num\_Classe > (num\_Finalizacoes + 1) \* 5);

preenche\_Classe(perso\_Prin, num\_Classe);

}

void preenche\_Classe( struct personagem\_Principal \*perso\_Prin, int num\_Classe) {

switch(num\_Classe) {

case 1:

strcpy(perso\_Prin->classe,"Warrior");

break;

case 2:

strcpy(perso\_Prin->classe,"Archer");

break;

case 3:

strcpy(perso\_Prin->classe,"Mage");

break;

case 4:

strcpy(perso\_Prin->classe,"Thief");

break;

case 5:

strcpy(perso\_Prin->classe,"Farmer");

break;

case 6:

strcpy(perso\_Prin->classe,"Paladin");

break;

case 7:

strcpy(perso\_Prin->classe,"Berserk");

break;

case 8:

strcpy(perso\_Prin->classe,"Priest");

break;

case 9:

strcpy(perso\_Prin->classe,"Druid");

break;

case 10:

strcpy(perso\_Prin->classe,"Merchant");

break;

case 11:

strcpy(perso\_Prin->classe,"Necromancer");

break;

case 12:

strcpy(perso\_Prin->classe,"Assasin");

break;

case 13:

strcpy(perso\_Prin->classe,"Guardian");

break;

case 14:

strcpy(perso\_Prin->classe,"Dark Knight");

break;

case 15:

strcpy(perso\_Prin->classe,"Dragon");

break;

}

}

void set\_Dimensions() {

system("MODE con cols=120 lines=35");

}

void informacoes () {

FILE \*arq;

char palavra[100];

char id\_Classe[15][30] = {"Warrior.txt","Archer.txt","Mage.txt","Thief.txt","Farmer.txt",

"Paladin.txt","Berserk.txt","Priest.txt","Druid.txt","Merchant.txt",

"Necromancer.txt","Assasin.txt","Guardian.txt","Dark Knight.txt","Dragon.txt"

};

int pos\_Classe = 0;

do {

system("cls");

arq = fopen(id\_Classe[pos\_Classe],"r");

if(arq == NULL) {

mostra\_Erro(1);

exit(1);

} else {

while(fgets(palavra,100,arq) != NULL)

printf(" %s",palavra);

}

getchar();

fclose(arq);

pos\_Classe++;

} while(pos\_Classe < 15);

}

void mostrar\_Menu\_Classes(int num\_Finalizacoes) {

printf("-------------- Classes --------------\n");

if(num\_Finalizacoes == 0)

mostrar\_Menu\_Classes\_Extra(num\_Finalizacoes);

else if(num\_Finalizacoes == 1)

mostrar\_Menu\_Classes\_Extra(num\_Finalizacoes);

else

mostrar\_Menu\_Classes\_Extra(num\_Finalizacoes);

}

void mostrar\_Menu\_Classes\_Extra(int num\_Finalizacoes) {

switch(num\_Finalizacoes) {

case 0:

printf(" 01) Warrior 02) Archer 03) Mage \n");

printf(" +2 FOR. +2 AGI. +2 INT. \n");

printf(" -1 INT. -2 FOR. -1 FOR. \n");

printf(" -1 AGI. -1 AGI. \n");

printf("\n");

printf(" 04) Thief 05) Farmer \n");

printf(" +2 AGI. +2 AGI. \n");

printf(" -2 FOR. +2 FOR. \n");

printf(" +2 INT. \n");

printf(" -4 CAR. \n");

break;

case 1:

printf(" 01) Warrior 02) Archer 03) Mage \n");

printf(" +2 FOR. +2 AGI. +2 INT. \n");

printf(" -1 INT. -2 FOR. -1 FOR. \n");

printf(" -1 AGI. -1 AGI. \n");

printf("\n");

printf(" 04) Thief 05) Farmer 06) Paladin \n");

printf(" +2 AGI. +2 AGI. +1 INT. \n");

printf(" -2 FOR. +2 FOR. +1 FOR. \n");

printf(" +2 INT. +1 CAR. \n");

printf(" -4 CAR. -3 AGI. \n");

printf("\n");

printf(" 07) Berserk 08) Priest 09) Druid \n");

printf(" +3 FOR. +1 INT. +1 INT. \n");

printf(" -3 INT. +2 CAR. +2 AGI. \n");

printf(" +3 FOR. -2 FOR. \n");

printf(" -1 CAR. \n");

printf("\n");

printf(" 10) Merchant \n");

printf(" +3 CAR. \n");

printf(" +1 INT. \n");

printf(" -2 FOR. \n");

printf(" -2 AGI. \n");

break;

default:

printf(" 01) Warrior 02) Archer 03) Mage \n");

printf(" +2 FOR. +2 AGI. +2 INT. \n");

printf(" -1 INT. -2 FOR. -1 FOR. \n");

printf(" -1 AGI. -1 AGI. \n");

printf("\n");

printf(" 04) Thief 05) Farmer 06) Paladin \n");

printf(" +2 AGI. +2 AGI. +1 INT. \n");

printf(" -2 FOR. +2 FOR. +1 FOR. \n");

printf(" +2 INT. +1 CAR. \n");

printf(" -4 CAR. -3 AGI. \n");

printf("\n");

printf(" 07) Berserk 08) Priest 09) Druid \n");

printf(" +3 FOR. +1 INT. +1 INT. \n");

printf(" -3 INT. +2 CAR. +2 AGI. \n");

printf(" +3 FOR. -2 FOR. \n");

printf(" -1 CAR. \n");

printf("\n");

printf(" 10) Merchant 11) Necromancer 12) Assasin \n");

printf(" +3 CAR. +5 INT. +5 AGI. \n");

printf(" +1 INT. -3 FOR. -3 FOR. \n");

printf(" -2 FOR. -1 AGI. -2 CAR. \n");

printf(" -2 AGI. -1 CAR. \n");

printf("\n");

printf(" 13) Guardian 14) Dark Knigh 15) Dragon \n");

printf(" +5 FOR. +3 FOR. +4 FOR. \n");

printf(" -4 AGI. +2 AGI. +1 AGI. \n");

printf(" -1 INT. -2 INT. -1 INT. \n");

printf(" -3 CAR. -4 CAR. \n");

}

}

Estruturas

int selecao\_Seta(int num\_Finalizacoes);

void cria\_Personagem(struct personagem\_Principal\* perso\_Prin);

void gotoxy(int x, int y);

void informacoes();

void le\_Valida\_Classe(struct personagem\_Principal \*perso\_Prin);

void le\_Valida\_Genero(struct personagem\_Principal \*perso\_Prin);

void le\_Valida\_Nome(struct personagem\_Principal \*perso\_Prin);

void limpa\_Tela();

void mostra\_ASCII\_Classe();

void mostra\_Janela();

void mostra\_Menu();

void mostraErro(int num\_Erro);

void set\_Dimensions();

=============================================================================================================================================

#include <stdio.h>

#include <stdlib.h>

//Esta em ordem alfabetica.

//Depois preencher no codigo.

struct atributo{

int Agi,For,Int;

};

struct personagem\_Principal{

char classe[30];

char genero[30];

//char genero; Somente um caracter

char nome[30];

char raca[30]; // Humano

struct atributo atributos;

//struct inventario invent;

//struct spell spells ;

};

int main (){

struct personagem\_Principal perso\_Prin;

perso\_Prin.atributos.For = 5;//Preenche força.

return 0;

}

=============================================================================================================================================

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TOTAL\_PERSONAGENS 5

//Quando for pedir o nome lembrar de limitar ate 20.

//Converter tudo em uma unica struct.

struct atributo{

char nome[18];

int nivel = 0;

//Arquivo

//ATRIBUTOS FISICOS

//Forca (FR) e Constituicao (CON)

// Dano extra que ele é capaz de causar com armas de combate corporal e o Peso Máximo que o Herói pode carregar ou sustentar

// Pontos de vida e teste de resistencia (por exemplo, prender a respiração, agüentar a dor, resistir a venenos, corrosivos, alucinógenos)

//Destreza (DEX) Agilidade (AGI)

// Lida melhor com armas, usar ferramentas, operar instrumentos delicados, atirar com arco-e-flecha, agarrar objetos em pleno ar

// desviar melhor dos ataques de oponentes, equilibrar-se melhor sobre um muro, dançar com mais graça, agarrar-se em parapeitos ou escapar de armadilhas

//ATRIBUTOS MENTAIS

//Inteligencia (INT) e Percepcao (PER)

// Capacidade de resolver problemas e charadas, afeta a Dificuldade em resistir a Poderes Psiônicos e também concede Pontos de Magia extras

//Carisma (CAR) e Forca de Vontade (WILL)

// Sorte, negociacoes, concentração e determinação

};

struct personagem{

char classe [30];

char genero [30];//Masculino, Feminino.

char nome[30];

char raca[30];

//Struct Inventario. // Cada arma possui um peso, mochila com um maximo de peso //Armas,armaduras,

//Struct Atributos. //Força,Destreza,Inteligencia,Constituição,Carisma,Agilidade.

int moeda;//Moeda do game (P&M's).

char magias[50];

};

/\*

Detalhes de cada personagem.

struct personagem\_A{

char classe [] = "Arqueira";

char genero [] = "Feminino";

char nome[] = "A";

char raca [] = "Elfa";

};

struct personagem\_B{

char classe [] = "Revenant"; --> Anti-Serafim;

char genero [] = "Masculino";

char nome[] = "B";

char raca [] = "Humano";

};

struct personagem\_C{

char classe [] = "Guerreira";

char genero [] = "Feminino";

char nome[] = "C";

char raca [] = "Valkiria";

};

struct personagem\_D{

char classe [] = "Mago";

char genero [] = "Masculino";

char nome[] = "D";

char raca [] = "Vaniries";

};

\*/

int main (){

struct personagem \*personagens;

personagens = malloc(TOTAL\_PERSONAGENS \* sizeof(struct personagem)); //Total de personagens no define \* tamanho da estrutura personagem

if(personagens == NULL){

printf("Falha no arquivamento");

getchar();

exit(1);

}

//prencPersonagens(personagens);

return 0;

}

Links

Lista de links especificados ao lado...

Pesquisar TurboC

http://www.jave.de/download/download.html

https://www.tutorialspoint.com/c\_standard\_library/stdio\_h.htm <== Tutorial de umas bibliotecas

http://www.cmaismais.com.br/referencia/cstdio/fopen/ <== Site lindo sobre as funções fopen,fseek,f\_\_qualquerCoisa

Armas em ASCII => https://www.asciiart.eu/weapons

Símbolos => https://sites.google.com/site/tabeladesimbolos/simbolos-que-nao-sao-do-teclado

Arquivos => http://linguagemc.com.br/arquivos-em-c-categoria-usando-arquivos/

Alocação Dinâmica => http://linguagemc.com.br/alocacao-dinamica-de-memoria-em-c/

Allegro => http://edcomjogos.dc.ufscar.br/tutoriais/tutorial\_allegro.pdf

Dowload Allegro => http://www.mediafire.com/file/471dzl0gfdj103w/allegro-5.0.4-1mpx.DevPak/file

Site Top para fazer rpg => http://www.facom.ufu.br/~albertini/prossiga/index.php/Simulador\_de\_mestre\_de\_jogo\_de\_RPG

https://producaodejogos.com/content-main/programas-para-criar-jogos-a-lista-definitiva/

<https://www.tutorialspoint.com/cprogramming/c_file_io.htm>

https://rpgromaduke.weebly.com/classes-e-raccedilas.html

http://www.geocities.ws/SoHo/7373/mythical2.html

http://loveascii.com/princess.html

<https://rpgromaduke.weebly.com/classes-e-raccedilas.html>

test

=============================================================================

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct status{

int vida;

int lvl;

};

struct magia{

char fire[30];

int qnt\_fire;

};

struct personagem {

char nome[30];

struct status status;

struct magia magias;

};

int main (){

struct personagem p;

//Para preencher a struct de string fica mais facil.

strcpy(p.nome,"Clack");

p.status.vida = 100;

p.status.lvl = 1;

strcpy(p.magias.fire,"Pheonexis Chaos");

p.magias.qnt\_fire = 3;

printf("Nome = %s\n",p.nome);

printf("Ponto de PV = %d\n",p.status.vida);

printf("Lvl do personagem = %d\n",p.status.lvl);

printf("Nome da magia = %s\n",p.magias.fire);

printf("Usos %d",p.magias.qnt\_fire);

return 0;

}

#ifndef COLORS\_H

#define COLORS\_H

#define foreground(color) FORE##color

#define background(color) BACK##color

#define style(style\_) style\_

/\*\* Foreground Colors \*\*/

#define FOREBLACK printf("\033[30m")

#define FORERED printf("\033[31m")

#define FOREGREEN printf("\033[32m")

#define FOREYELLOW printf("\033[33m")

#define FOREBLUE printf("\033[34m")

#define FOREMARGENTA printf("\033[35m")

#define FORECYAN printf("\033[36m")

#define FOREWHITE printf("\033[37m")

#define FORENORMAL\_COLOR printf("\033[39m")

/\*\* Background Colors \*\*/

#define BACKBLACK printf("\033[40m")

#define BACKRED printf("\033[41m")

#define BACKGREEN printf("\033[42m")

#define BACKYELLOW printf("\033[43m")

#define BACKBLUE printf("\033[44m")

#define BACKMAGENTA printf("\033[45m")

#define BACKCYAN printf("\033[46m")

#define BACKWHITE printf("\033[47m")

#define BACKNORMAL printf("\033[49m")

/\*\* Style \*\*/

#define BRIGHT printf("\033[1m")

#define DIM printf("\033[2m")

#define NORMAL printf("\033[22m")

#define RESETALL printf("\033[0m")

#define UNDERLINE printf("\033[4m")

#define BLINKSLOW printf("\033[5m")

#define BLINKRAPID printf("\033[6m")

#define ITALIC printf("\033[3m")

#define NEGATIVE printf("\033[7m")

#endif /\* COLORS\_H \*/

Weapons

OBS: copiar e colar no bloco de notas para ver como fica:

pedra =>

o

machado =>

<)

│

escudos =>

\_\_\_

) O (

`---´

\_\_\_

)\_\_)

\_

(:)

¯

()

\_

| |

)o(

|\_|

.-.

|o|

'-'

.╦.

╠╬╣

'╩'

\_

(>>

¯­­­

/-\

).(

\-/

pá =>

┼

│

Ü

espada =>

│

┼