//#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, car\_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, int \*num\_Classe);

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

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];

int num\_Classe;

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, &num\_Classe);

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 \n",perso\_Prin->classe);

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

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 \n\n",perso\_Prin->classe);

printf(" Agility = %d \t Carism = %d \n",perso\_Prin->atributos.agi\_P, perso\_Prin->atributos.agi\_P);

printf(" Force = %d \t Intelligence = %d \n\n",perso\_Prin->atributos.for\_P, perso\_Prin->atributos.int\_P);

/\*

//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");

}

}

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

int pontos\_Iniciais = 5;

int pontos\_Aux = 5;

char escolha\_Atributo;

switch(num\_Classe) {

case 1:

perso\_Prin->atributos.for\_P = pontos\_Aux + 2;

perso\_Prin->atributos.agi\_P = pontos\_Aux - 1;

perso\_Prin->atributos.int\_P = pontos\_Aux - 1;

perso\_Prin->atributos.car\_P = pontos\_Aux;

break;

case 2:

perso\_Prin->atributos.for\_P = pontos\_Aux - 2;

perso\_Prin->atributos.agi\_P = pontos\_Aux + 2;

perso\_Prin->atributos.int\_P = pontos\_Aux;

perso\_Prin->atributos.car\_P = pontos\_Aux;

break;

case 3:

perso\_Prin->atributos.for\_P = pontos\_Aux - 1;

perso\_Prin->atributos.agi\_P = pontos\_Aux - 1;

perso\_Prin->atributos.int\_P = pontos\_Aux + 2;

perso\_Prin->atributos.car\_P = pontos\_Aux;

break;

case 4:

perso\_Prin->atributos.for\_P = pontos\_Aux - 2;

perso\_Prin->atributos.agi\_P = pontos\_Aux + 2;

perso\_Prin->atributos.int\_P = pontos\_Aux;

perso\_Prin->atributos.car\_P = pontos\_Aux;

break;

case 5:

perso\_Prin->atributos.for\_P = pontos\_Aux + 2;

perso\_Prin->atributos.agi\_P = pontos\_Aux + 2;

perso\_Prin->atributos.int\_P = pontos\_Aux + 2;

perso\_Prin->atributos.car\_P = pontos\_Aux - 4;

break;

case 6:

perso\_Prin->atributos.for\_P = pontos\_Aux + 1;

perso\_Prin->atributos.agi\_P = pontos\_Aux - 3;

perso\_Prin->atributos.int\_P = pontos\_Aux + 1;

perso\_Prin->atributos.car\_P = pontos\_Aux + 1;

break;

case 7:

perso\_Prin->atributos.for\_P = pontos\_Aux + 3;

perso\_Prin->atributos.agi\_P = pontos\_Aux ;

perso\_Prin->atributos.int\_P = pontos\_Aux - 3;

perso\_Prin->atributos.car\_P = pontos\_Aux;

break;

case 8:

perso\_Prin->atributos.for\_P = pontos\_Aux - 3;

perso\_Prin->atributos.agi\_P = pontos\_Aux ;

perso\_Prin->atributos.int\_P = pontos\_Aux + 1;

perso\_Prin->atributos.car\_P = pontos\_Aux + 2;

break;

case 9:

perso\_Prin->atributos.for\_P = pontos\_Aux - 2;

perso\_Prin->atributos.agi\_P = pontos\_Aux + 2;

perso\_Prin->atributos.int\_P = pontos\_Aux + 1;

perso\_Prin->atributos.car\_P = pontos\_Aux - 1;

break;

case 10:

perso\_Prin->atributos.for\_P = pontos\_Aux - 2;

perso\_Prin->atributos.agi\_P = pontos\_Aux - 2;

perso\_Prin->atributos.int\_P = pontos\_Aux + 1;

perso\_Prin->atributos.car\_P = pontos\_Aux + 3;

break;

case 11:

perso\_Prin->atributos.for\_P = pontos\_Aux - 3;

perso\_Prin->atributos.agi\_P = pontos\_Aux - 1;

perso\_Prin->atributos.int\_P = pontos\_Aux + 5;

perso\_Prin->atributos.car\_P = pontos\_Aux - 1;

break;

case 12:

perso\_Prin->atributos.for\_P = pontos\_Aux - 3;

perso\_Prin->atributos.agi\_P = pontos\_Aux + 5;

perso\_Prin->atributos.int\_P = pontos\_Aux;

perso\_Prin->atributos.car\_P = pontos\_Aux - 2;

break;

case 13:

perso\_Prin->atributos.for\_P = pontos\_Aux + 5;

perso\_Prin->atributos.agi\_P = pontos\_Aux - 4;

perso\_Prin->atributos.int\_P = pontos\_Aux - 1;

perso\_Prin->atributos.car\_P = pontos\_Aux;

break;

case 14:

perso\_Prin->atributos.for\_P = pontos\_Aux + 3;

perso\_Prin->atributos.agi\_P = pontos\_Aux + 2;

perso\_Prin->atributos.int\_P = pontos\_Aux - 2;

perso\_Prin->atributos.car\_P = pontos\_Aux - 3;

break;

case 15:

perso\_Prin->atributos.for\_P = pontos\_Aux + 4;

perso\_Prin->atributos.agi\_P = pontos\_Aux + 1;

perso\_Prin->atributos.int\_P = pontos\_Aux - 1;

perso\_Prin->atributos.car\_P = pontos\_Aux - 4;

break;

}

printf("------- Skill Points -------\n");

while(pontos\_Iniciais != 0) {

limpa\_Tela();

printf(" Total Points %d\n",pontos\_Iniciais);

printf(" A - Agility (Agi.) = %d\n",perso\_Prin->atributos.agi\_P);

printf(" F - Force (For.) = %d\n",perso\_Prin->atributos.for\_P);

printf(" C - Carism (Car.) = %d\n",perso\_Prin->atributos.car\_P);

printf(" I - Intelligence (Int.) = %d\n",perso\_Prin->atributos.int\_P);

printf(" Enter a character to represent the skill you want to add +1.\n");

printf(" %c ",175);

fflush(stdin);

scanf("%c", &escolha\_Atributo);

escolha\_Atributo = toupper(escolha\_Atributo);

switch(escolha\_Atributo) {

case 'A':

perso\_Prin->atributos.agi\_P++;

break;

case 'C':

perso\_Prin->atributos.car\_P++;

break;

case 'F':

perso\_Prin->atributos.for\_P++;

break;

case 'I':

perso\_Prin->atributos.int\_P++;

break;

}

pontos\_Iniciais--;

}

}