Universidade Federal de Minas Gerais

EYS

Documentação do Trabalho prático

Nome: Chrystian Martins Soares Costa

Disciplina: Programação e desenvolvimento de software I

Professor: Pedro Olmo

1 – Introdução

Esta documentação é referente ao trabalho prático final da disciplina de PDS I onde deveríamos desenvolver um jogo eletrônico do tipo JRPG usando elementos semelhantes aos usados nos clássicos jogos da saga “Final Fantasy” Basicamente, o jogador deve se movimentar pelo cenário, com a possibilidade de encontrar algum inimigo, até chegar ao objetivo.

O jogo foi desenvolvido em C utilizando a biblioteca Allegro e os conhecimentos adquiridos em sala, aplicando na prática os conteúdos estudados.

2 - Instruções

O jogador começa no canto inferior esquerdo de um cenário branco e tem como objetivo chegar no canto superior direito do mesmo existindo a possibilidade de se encontrar um monstro pelo caminho.
O jogador se movimenta com as setas do teclado e no combate, utilizando as setas cima e baixo, pode escolher entre 3 opções:

Atacar: o jogador causa 20 de dano ao monstro e recebe o dano respectivo à força do monstro.
Se o monstro morrer, ele desaparece do cenário, se o herói morrer o jogo se encerra.

Especial: o jogador causa um dano aleatório entre 10 e 50 ao monstro e recebe o dano respectivo a força do monstro.
Se o monstro morrer, ele desaparece do cenário, se o herói morrer o jogo se encerra.

Fugir: O jogador tem uma porcentagem de chance de fugir da batalha deixando o monstro no mesmo lugar.

3 - História

O jogo se passa no limbo, o lugar onde as almas lutam para sobreviver, ironicamente.
Você é uma simples alma de uma pessoa comum, chamada Cleiton, sem entender o que está acontecendo você vai em direção ao único lugar que de diferencia da imensidão branca do limbo, para o portal da subterra, mas você não percebe, **QUE ELES TE OBSERVAM.**

4 – Funções e Procedimentos

As seguintes funções estão presentes no jogo:

***void* initHeroi(Heroi *h){**

Inicia as propriedades do herói.

***void* initMonstro(Monstro *m, Heroi *h){**

Inicia as propriedades dos monstros do cenário.

***void* initGlobais(){**

Inicia as propriedades das variáveis globais.

***void* initCursor(Cursor *c){**

Inicia as propriedades do cursor que será usado para selecionar as opções.

***void* desenhaHeroiNaveg(Heroi h){**

Desenha o Herói no cenário de navegação.

***void* desenhaHeroiNavegD(Heroi h){**
***void* desenhaHeroiNavegL(Heroi h){**
***void* desenhaHeroiNavegR(Heroi h){**

Desenha o herói na direção em que o jogador apertou a tecla.

***void* desenhaMonstroNaveg(Monstro m){**

Função que, originalmente, fica desabilitada mas pode ser ativada para mostrar a posição dos monstros no cenário de navegação.

***void* desenhaCenarioNaveg(Heroi *h){**

Desenha o cenário de navegação.

***int* processaTeclaNaveg(Heroi *h, *int* tecla){**

Processa as teclas que serão pressionadas no modo de navegação.

***float* dist(Heroi *h, Monstro *m){**

Verifica a distancia euclidiana entre o herói e o monstro.

***float* distMonstro(Monstro *m1, Monstro *m2){**

Verifica a distancia entre um monstro e outro no cenário.

***int* detectouMonstro(Heroi h, Monstro m){**

Se a distância euclidiana entre o Herói e o monstro for menor que 30 pixels troca o jogo para o modo de batalha.

***int* chegouObjetivoHerois(Heroi h){**

Verifica se o Herói chegou ao objetivo.

***void* desenhaCenarioBatalha(){**

Desenha o modo de Batalha.

***void* desenhaBatalha(Heroi h, Monstro m){**

Desenha o herói na direita e o monstro na esquerda do cenário.

***void* desenhaCursor(Cursor c){**

Desenha o cursor que será usado para selecionar as opções da batalha.

***void* processaTeclaBatalha(Heroi *h, *int* tecla, Cursor *c){**

Processa as teclas da batalha, seta para cima, seta para baixo e o enter.

***int* processaAcaoHeroi(Heroi *h){**

Processa qual das opções o jogador escolheu no menu da batalha.

***int* processaAtaque(ALLEGRO_EVENT *ev, Heroi *h, Monstro *m){**

Se o jogador escolheu “atacar”, a função processa o dano causado e recebido e verifica se o herói ou o monstro morreu.

***int* processaAtaqueE(ALLEGRO_EVENT *ev, Heroi *h, Monstro *m){**

Se o jogador escolheu “especial”, a função processa o dano aleatório causado e o dano recebido e verifica se o herói ou o monstro morreu.

***void* desenhaVitoria(Heroi *h){**

Se o jogador chegou ao objetivo, a função muda a tela para a tela de vitória.

int **processaTeclaVitoria(*int* teclaV, Heroi *h){**

Se o jogador apertar enter o jogo se encerra.

void **desenhaDerrota(){**

Se o herói tiver uma vida <= 0 é carregada a tela de derrota.

int **processaTeclaDerrota(*int* teclaV){**

Se o jogador apertar enter o jogo se encerra.

5 – Descrição Dos structs criados

Foram criados 3 structs:

typedef struct **Heroi:**

int **x, y, centrox, centroy, batalhaX, batalhaY;**

Coordenadas que serão usadas para para desenhar o herói na tela.

int **dano;**

Variável que armazena o dano causado pelo herói.

int **hp;**

Variável que armazena a vida do herói.

char **score[20];**
int **pontos;**

Vetor e variável que armazenam os pontos obtidos por cada monstro derrotado.

int **acao;**

armazena qual a opção que o cursor está.

int **executar;**

armazena o estado da opção selecionada pelo jogador.

int **x_old, y_old, centrox_old, centroy_old;**

retorna o jogador para a posição anterior se a fuga for bem sucedida.

typedef struct Monstro:

int **x, y;**
armazena as coordenadas do monstro.

int **batalhaXm;**
int **batalhaYm;**

Armazenam as coordenadas do monstro no modo batalha.

ALLEGRO_COLOR cor;

Se a função de desenhar os monstros tiver habilitada, desenha os monstros na tela com a cor definida.

int **hp;**

Armazena a vida do monstro.

int **dano;**

Armazena o dano causado pelo monstro

int **pontos;**

Armazena a quantidade de pontos dada pelo monstro de acordo com seu nível de dano.

typedef struct Cursor:

int **x;**
int **y;**

Armazenam as coordenadas iniciais do cursor.

6 – Visão Geral do Programa

Foi desenvolvido o jogo base solicitado pelo professor sendo assim o jogo possui as funções essenciais pedidas no enunciado do TP.