



Nome: Amanda Souza

Disciplina: Algoritmos e estrutura de Dados I

Professores: Pedro O. S. Vaz de Melo

Ítalo Cunha

I. Introdução da Documentação

Esta documentação tem por objetivo explicar e detalhar a implementação do Trabalho Prático da disciplina Algoritmos e Estruturas de Dados I lecionada pelos professores Pedro O. S. Vaz de Melo e Ítalo Cunha. Tal trabalho tinha como objetivo fazer com que o aluno aplicasse conhecimentos teóricos através do desenvolvimento de um jogo eletrônico do tipo action/adventure, onde o jogador deveria controlar um personagem capaz de se movimentar por calabouços, sem morrer, e chegar ao fim do jogo passando por portas que conectam as fases.

a) Sobre o Jogo / Informações Técnicas

A dinâmica do jogo é totalmente inversa aos RPGs comuns: em “Kill The Hero” você é um monstro da raça orc que tenta dominar de volta um território que foi dominado pelos heróis após ter sido dominado pelos monstros. Confuso, não? O que importa é que você tem que tomar de volta o que é seu por direito (ou quase isso)! Lute com os heróis no seu caminho, conquiste território e ajude o seu povo a ser novamente a raça suprema dominante!

“Kill The Hero” é um jogo 2D do tipo RPG que foi desenvolvido na linguagem C utilizando a biblioteca allegro como auxílio gráfico/visual. Também foram utilizados os conhecimentos adquiridos e/ou aperfeiçoados durante as aulas de Algoritmos e Estruturas de Dados I.

b) Funcionalidades do Jogo

O jogador utiliza-se das teclas de setas do teclado e/ou das teclas A, S, D, W para se movimentar pelas fases da dungeon. Uma batalha começa toda vez que o jogador entra em contato com um inimigo, e pode utilizar as teclas 1 (soco) e 2 (usar arma equipada) para desferir um golpe contra o inimigo e tentar vencer a batalha.

O jogador pode também interagir com o cenário, ao pressionar a tecla E quando estiver de encontro a um baú de tesouros, para pegar possíveis prêmios, ou pressionar a tecla P quando estiver de encontro ao feixe de luz, para que o jogo seja salvo.

A vida do jogador, sua quantidade de ouro e seus pontos de experiência podem ser visualizados na parte superior da tela, ou na tela de status (ver abaixo).

Em qualquer momento do jogo, pode-se acessar o mapa (tecla M), para ter uma ampla visão da cidade onde se situa o jogador; o inventário (tecla I), para se ter acesso a todos os itens adquiridos até o momento, sejam eles equipamentos ou poções; a tela de status do jogador (tecla C), onde pode ser visualizada a vida e os pontos de experiência, como também seu nível, ataque e defesa; o menu (tecla Q), para sair do jogo.

II. Funções e Procedimentos

a) Procedimentos

A seguir estão listados e explicados todos os procedimentos que foram utilizados na implementação do jogo:

- *void mvChase (float *hx, float *hy, float cx, float cy);*

Movimento automático de inimigo que consiste em perseguir o jogador a fim de entrar em batalha.

- *void mvRunTeleport (float *hx, float *hy, float cx, float cy);*

Movimento automático de inimigo que consiste em fugir do jogador e teleportar aleatoriamente pela dungeon em momentos aleatórios e quando entra em contato com qualquer parede da dungeon.

- *void mvProtector (float *hx, float *hy, float cx, float cy, float px, float py, int kh[2]);*

Movimento automático de inimigo que consiste em proteger o *boss* da fase, perseguindo-o. Quando o jogador se aproxima do boss e/ou do “protetor”, este persegue o jogador até que se afaste novamente.

- *void Wall (float *px, float *py, float w, float h, float dh);*

Procedimento que identifica todas as paredes da dungeon e impede qualquer personagem ou inimigo de transpassá-las.

- *void lvlUP (float *chp, int lvl, float *catk);*

Identificação do aumento de nível do personagem, e mudança de seus pontos de vida e de ataque referentes ao novo nível.

- *void resetHero (float *h1hp, float *h1bhp);*

Dá novamente todos os pontos de vida de um inimigo. Utilizada em mudança de fase.

- *void genDungeon(float *hero1x, float *hero2x, float *hero3x, float *hero1y, float *hero2y, float *hero3y, float *chestx, float *chesty);*

Gera uma nova dungeon em uma mudança de fase, com posições x e y aleatórias para todos os inimigos e para o baú de tesouros.

- *void gLoot(int n, float *cxp, float *cgld, int *k);*

Dá uma quantidade de ouro para o personagem dependendo da forma como conseguiu (matando inimigos de diferentes tipos, abrindo o baú de tesouros...).

- *void heal(float *chp, float basehp, int n);*

Cura o personagem dependendo de qual poção foi utilizada (mais 20 por cento dos pontos de vida base para a poção pequena, 40 por cento para a poção média e 85 por cento para a poção grande).

- *void addInv(int item, int inv[12][3]);*

Adiciona um novo item no inventário, levando em consideração ordem crescente, espaços já ocupados e setores separados, sendo a matriz separada em equipamentos nas posições de 0 a 8 e poções nas posições de 9 a 11. Itens repetidos que forem adicionados só aumentarão a quantidade do item adicionado anteriormente.

- *void useItem(int inv[12][3], int pos);*

Diminui em 1 (um) a quantidade de um item sendo utilizado. Se a quantidade deste item for zero, ele é eliminado do inventário.

b) Funções

A seguir estão listadas e explicadas todas as funções que foram utilizadas na implementação do jogo:

- *float xpToUp (float lvl, float *pts);*

Calcula e retorna quantos pontos de experiência são necessários para atingir um nível x+1 assim que o jogador atinge o nível x.

- *int Impact (float cx, float cy, float ch, float cw, float px, float py, float ph, float pw);*

Identifica se houve impacto entre o personagem e algum outro item no ambiente da dungeon, podendo ele ser um inimigo, um baú de tesouros ou feixe de luz. Retorna 1 se

houve impacto e 0 se não.

- *float gRnd(int n);*

Gera e retorna um número aleatório que esteja entre certos valores para atribuir às posições x (de 40 a 840) e y (de 70 a 470) dos inimigos e x (de 40 a 140) do baú de tesouros no procedimento genDungeon.

- *int dmgCalc(int lvl, int base);*

Cálculo do dano causado por certa arma, levando em conta o nível da mesma e o dano de base.

III. Visão Geral e Código

a) Principais Telas

- Tela de Jogo

O jogo se inicia na tela principal de jogo, no ambiente de uma dungeon. O jogador pode interagir com quaisquer elementos da tela, movimentar-se por ela e interagir com as outras telas citadas abaixo através de seu teclado;

- Mapa da Cidade

Ao pressionar a tecla M, é possível ter uma visão geral da cidade, com todos os locais passíveis de visitaç o;

- Invent rio (Mochila)

Ao pressionar a tecla I, o jogador tem acesso a todos os itens que foram adquiridos at  o momento e que foram guardados na mochila, sendo eles equipamentos ou po  es. Para utilizar ou equipar qualquer item basta pressionar a tecla referente a ele (de 1 a 9 para equipamentos e A, B ou C para po  es) e logo em seguida a tecla ENTER. Itens equip veis ter o em sua descri  o “*equipado*” e itens us veis ir o ser decrescidos da quantidade. Se esta chegar a zero, o item ser  retirado do invent rio;

- Status do Jogador

Ao pressionar a tecla C,   poss vel visualizar os pontos de vida e de experi ncia do jogador, tanto de forma num rica quanto gr fica. Tamb m   poss vel visualizar o n vel em que o jogador se encontra e o seu ataque;

- Menu

Ao pressionar a tecla Q, o menu é exibido com as opções “Voltar para o Jogo”, “Sair para o Mapa” e “Sair do Jogo”, sendo que o primeiro volta ao jogo e o último fecha o jogo;

- Tela de Confirmação de Saída

Ao pressionar as teclas ALT + F4, o “x” no canto superior direito da tela ou dar ENTER na opção “Sair do Jogo” do Menu, é exibida uma tela de confirmação com as opções “Sair” (tecla ENTER) e “Voltar ao Jogo” (tecla ESC);

- Tela de Batalha

Quando o jogador entra em contato com qualquer inimigo, uma batalha é iniciada e a Tela de Batalha é exibida. Neste momento do jogo, o jogador pode acessar normalmente qualquer uma das telas citadas anteriormente, salvo a tela principal que permanece bloqueada até o fim da batalha, e interagir com elas. O jogador não tem permissão para se movimentar até que a batalha chegue ao fim. As únicas duas interações exclusivas desta tela são pressionar a tecla 1 para desferir um soco, decrementando os pontos de vida do inimigo com os pontos de ataque base do jogador, ou pressionar a tecla 2 para que seja utilizada a arma equipada (se houver uma), decrescendo os pontos de vida do inimigo com os pontos de ataque da arma;

b) Pontos Principais do Código

O código é dividido em três pontos principais:

- Funções: reúne todas as funções e procedimentos que são utilizados no jogo;
- Inicializações: inicializa tudo que é necessário para o bom funcionamento do jogo (tela, bitmaps, timer, eventos, fonte, teclado, sons, etc);
- Jogo: onde acontece toda a dinâmica do jogo;