Documentação - Trabalho Prático
"The Legend of Link"

Universidade Federal de Minas Gerais
Algorítmos e Estruturas de Dados 1
02/2015

Luísa Ribeiro Bezerra

1- Introdução

O Trabalho Prático sugerido para a disciplina de Algorítmos e Estrutura de Dados I foi baseado na criação de um jogo utilizando a biblioteca Allegro (www.allegro.cc), que foi apresentada para os alunos durante a execução do curso. Por meio do Allegro, fomos possibilitados a desenvolver um jogo com maior acesso a recursos e ferramentas para a estruturação de elementos que não era possível anteriormente.

O objetivo principal foi o desenvolvimento de um jogo no estilo *action/adventure*. No jogo, o usuário controla um personagem que tem como objetivo escapar de um calabouço e desviar de inimigos - como também pode matá-los. Há um sistema de lógica de chaves e portas, sendo que estas últimas só se abrem após o personagem encontrar a chave e pegá-la. Nas salas, foram implementados um "*mini boss*", que é um morcego, e o "*boss*", que é um mago. Ao derrotá-los, o jogador consegue atingir o objetivo de conclusão do calabouço, assim, completando o desafio imposto ao mesmo.

O enredo e o tema de meu jogo foram baseados no jogo "*The Legend of Zelda - The Minish Cap*", que foi de onde retirei os *resources,* elementos e a resolução de 240 x 160 pixels (Game Boy Advance). A ideia central que foquei foi a Zelda como personagem principal, que tem o objetivo de salvar o Link do chefão final. Destaquei a Zelda para enfatizar o papel da mulher nos games, assim como as desenvolvedoras de games e programadoras são importantes no meio da computação e programação - todas, em geral, somos personagens importantíssimas dentro desta produção maravilhosa que é a criação de jogos digitais!

2- Implementação

2.1- Estrutura de Dados

As entidades do jogo foram declaradas como *structs* para o melhor arranjo do sistema. Dentre elas, destaco:

- *allegro_stuff:* ela funciona como se fosse um "main", onde separei as prefedinições os procedimentos necessários para fazer com que o Allegro funcione. Para tal, separei algumas funções dentro do allegro_stuff que serão analisadas no próximo subtópico;
- *animation:* a struct animation foi feita pra que facilitasse a criação e implementação de animações, já que é uma abstração do procedimento de implementação e é utilizada em várias outras structs;
- *boss, personagem, morcego, flecha, fogo, chave:* são structs que representam entidades do jogo. Para cada um deles, existem três funções: uma para inicializá-los, outra para atualizá-los e outra para desenhá-los, que serão chamados no allegro_stuff.

2.2- Funções e Procedimentos

- *void animation_draw(animation * a, int x, int y):* desenha a parte do *sprite* relativo ao quadro da animação (frame). Ele reconhece um bitmap e uma posição x e y em que ele será desenhado na tela;
- *void animation_init(animation * a, ALLEGRO_BITMAP *s, int x, int y, int width, int height, int length, float speed):* esta função recebe um bitmap e localiza, neste bitmap, a posição x e y iniciais do sprite, o widht (largura) e o height (altura) de cada sprite, o lenght (duração) da animação e o speed (velocidade) da troca de quadros da animação;
- *void animation_update(animation * a):* esta função desenvolve a lógica da animação;
- *void boss_draw(boss *b), void chave_draw(chave *c), void flecha_draw(flecha *f), void fogo_draw(fogo *f), void morcego_draw(morcego *m), void personagem_draw(personagem * p):* estas funções desenham a entidade recebida como parâmetro;
- *void boss_init(boss *b), void chave_init(chave *c, int x, int y, bool viva), void flecha_init(flecha *f, int x, int y, char direcao), void flecha_init_morta(flecha * f), void fogo_init_morto(fogo *f), void fogo_init_vivo(fogo *f, int x, int y), void Init(allegro_stuff* a), void morcego_init(morcego *m, int x, int y), void morcego_init_morto(morcego *m), void personagem_init(personagem * p):* estas funções basicamente recebem uma entidade como parâmetro de entrada. Algumas entidades inicializam-se "vivas" para demonstrar que elas aparecem na tela, e em dado momento elas desaparecem e tornam-se "mortas" (para tal, usamos a variável booleana para confirmar estes dados). Alguns elementos precisam de uma posição x e y e/ou direção para serem inicializados no mapa, portanto, recebem também a posição como parâmetro de entrada. Em especial, a inicialização do allegro_stuff carrega e trata da parte de som, imagens e das inicializações do Allegro.
- *void boss_update(boss *b), void flecha_update(flecha *f), void fogo_update(fogo *f), void morcego_update(morcego *m, int x, int y), void personagem_update(personagem * p), void Update(allegro_stuff * a):* estas funções atualizam a entidade que é enviada como parâmetro. Algumas, como o morcego, recebe também uma posição x e y do personagem, posto que eles se desloca no mapa em direção ao personagem. Há, também o update o allegro_stuff, que organiza a lógica de atualização do próprio Allegro, chamando todas as updates e draws das entidades que ele controla.
- *void carregar (allegro_stuff *a):* esta função carrega o jogo do ponto em que ele foi salvo baseando-se no arquivo "jogo.txt".
- *void salvar (allegro_stuff *a):* esta função salva o jogo em um arquivo "jogo.txt".
- *bool colidiu_flecha_boss(flecha *f, boss *b), bool colidiu_flecha_morcego(flecha *f, morcego *m), bool colidiu_personagem_chave(personagem *p, chave *ch), bool colidiu_personagem_fogo(personagem *p, fogo *f), bool colidiu_flecha_boss(flecha *f, boss *b), bool colidiu_personagem_morcego(personagem *p, morcego *m):* esta função confere se há uma colisão entre as entidades "a" e "b" recebidos como parâmetros, checando se horizontalmente e verticalmente um está parcialmente ou totalmente dentro do outro para decidir se houve colisão ou não.

3- Programa Principal

O programa principal é o int main(int argc, char **argv), e sua estrutura é muito simples como podemos conferir na imagem abaixo:

```
int main(int argc, char **argv){

    allegro_stuff a;

    Init(&a);

    while(a.closed == false){
        Update(&a);
    }

    return 0;
}
```

É visível a sua simplicidade, uma vez que todo o jogo foi desenvolvido em módulos para a melhor ordenação do todo. O programa principal, em suma, cria um objeto do tipo allegro_stuff e o manda como parâmetro para função *void Init(allegro_stuff* a),* que o inicializa. Enquanto o jogo ainda está aberto (a.closed==false), haverá a chamada do *Update(allegro_stuff * a)* para o objeto criado. Esta igualdade torna-se verdadeira (a.closed==true) quando a janela do jogo é fechada ou quando o jogo é salvo.