



“Lost in Past...”

Jogo de Aventura



Índice

Introdução	2
Modo super user.....	3
Criação da estrutura do jogo	4
Estrutura do Jogador	4
Estrutura do Adversário	4
Estrutura do Mapa	4
Threads: outro tipo de “estrutura” do jogo	4
Funções principais: Jogabilidade	6
Combate entre personagens.....	6
Verificar o vencedor	6
Verificar obtenção do tesouro	6
Funções principais: Estrutura do jogo.....	7
Abrir Mapa (.txt e/ou .bin)	7
Guardar jogo	7
Carregar jogo.....	7
Funções secundárias: Gráficos	8
Gráfico(s) Menu Principal.....	8
Gráfico(s) Banner Jogo	8
Gráfico(s) Créditos Finais	8
Testes de Desenvolvimento	9
Estrutura do jogador	9
Estrutura do adversário.....	9
Estrutura do Mapa	9
Conclusão	10
Webgrafia	11
Anexos.....	12
Código completo.....	13

Introdução

No âmbito da Disciplina de Sistemas Operativos, da Escola Superior de Tecnologia e Gestão (Instituto Politécnico de Beja), no Curso de Engenharia Informática foi proposto a realização de um segundo trabalho prático (usando a linguagem de programação C¹) onde o principal objetivo era programar um jogo de aventura do estilo RPG, usando como referência o primeiro jogo do género, o *Dungeons & Dragons*², criando uma interface gráfica para o mesmo.

Como título, escolhi “*Lost in Past...*”, pois o jogador é transportado para uma viagem no tempo longínquo onde a Terra era dividida por Reinos e as batalhas e conquistas do desconhecido eram constantes. A história do jogo remete o jogador para uma dessas cruzadas, onde, no meio do caminho, encontra um castelo abandonado e decide explorá-lo. Reza a história que naquele castelo havia um tesouro por descobrir guardado por um monstro. O jogador para descobrir o tesouro guardado é auxiliado pelo computador (dando coordenadas) para se movimentar.

Irei abordar a utilização de threads (criação e utilização das mesmas) e aspetos relevantes na conceção da interface gráfica (reaproveitando o restante código do trabalho de grupo anterior). Pretendo também, com este relatório, demonstrar e detalhar a maneira como foi desenvolvido o trabalho ao longo das semanas, explicando (e exemplificando, quando possível) com excertos do código.

¹ [http://en.wikipedia.org/wiki/C_\(programming_language\)](http://en.wikipedia.org/wiki/C_(programming_language))

² <http://www.ddo.com>

Modo super user

Este modo está “oculto”, isto é, para ser utilizado é preciso de chama-lo como parâmetro na prompt³ ou no terminal⁴. Esta funcionalidade permite ao jogador obter uma energia muito superior ao adversário, facilitando assim a vitória no jogo.

O código desta funcionalidade está assim definido:

```
if( (argc == 2) && (strcmp(argv[1], "su") == 0) )
{
    jogador.jogadorEnergia = JOGADOR_ENERGIA_SU_MODE;
}
```

O *argc* informa ao programa a quantidade de parâmetros inseridos pelo utilizador, enquanto que o *argv[1]* serve para comparar se o argumento inserido é igual ao “su”. Se for o jogador fica com a energia superior. Caso contrário, fica com a energia estabelecida previamente⁵.

³ Linha de comandos DOS

⁴ Linha de comandos UNIX, caso o jogador tenha executado o jogo em Mac ou Linux

⁵ Ver estrutura do jogador

Criação da estrutura do jogo

Como no trabalho anterior, todas as estruturas são simples estruturas de dados, sendo que armazena dados como o nome, energia e localização de ambas as personagens (transitaram do trabalho de grupo anterior).

Estrutura do Jogador

```
struct Jogador{
    char nome[JOGADOR_MAX_NOME];
    int jogadorEnergia;
    int jogadorPosicao;
};
```

Estrutura do Adversário

```
struct Adversario{
    int adversarioEnergia;
    int adversarioPosicao;
};
```

Estrutura do Mapa

```
struct Celula{
    int norte;
    int sul;
    int este;
    int oeste;
    int tesouro;
    char descricao[SISTEMA_MAPA_MAX_DESCRICAO];
};
```

Threads: outro tipo de “estrutura” do jogo

Para além das estruturas de dados, o jogo tem um outro tipo de estrutura. Este tipo de estrutura dá pelo nome de “thread” e permite ter os movimentos das personagens quase em simultâneo (dando a perceção que o adversário está em constante movimento à espera do jogador para desencadear uma luta). Abaixo segue um exemplo de criação da thread (o restante código encontra-se nos anexos).

Neste exemplo está criado uma thread:

```
HANDLE hThread1Adversario = CreateThread(  
    NULL,           // default security attributes  
    0,             // use default stack size  
    ThreadProc1Adversario, // thread function  
    NULL,          // argument to thread function  
    0,             // use default creation flags  
    NULL);         // returns the thread identifier
```

Funções principais: Jogabilidade

Estas funções têm um papel fundamental no processamento do jogo. Com elas é definido o rumo que o jogo toma, tomando decisões e guiando o jogador.

Combate entre personagens

Esta função é a responsável pelo confronto entre as personagens do jogo. Quando o jogador se encontra na mesma sala onde o adversário está é desencadeado uma luta. Durante essa luta, o jogador pode usar 3 armas diferentes, cada uma tem um dano diferente e o resultado do final do combate é variável.

Como a função é extensa, a mesma pode ser observada nos anexos.

Verificar o vencedor

Esta função determina o vencedor do jogo dependendo do resultado que o jogador obtém durante o jogo. O jogo termina ou quando o jogador obtém o tesouro, ou quando derrota o adversário.

Verificar obtenção do tesouro

Esta função é semelhante à anterior, sendo que a diferença é que esta função é responsável apenas pela obtenção do tesouro. Caso o jogador tenha obtido o tesouro, o jogo termina e o jogador sai vencedor.

Funções principais: Estrutura do jogo

As funções abaixo explicadas são essenciais para o funcionamento do jogo, pois é com elas que o sistema inicia o mapa, pode (ou não) carregar um jogo previamente guardado. São usados ficheiros (formato .txt e/ou .bin – binário) para armazenar a informação.

Abrir Mapa (.txt e/ou .bin)

Com esta opção, o sistema pode iniciar o jogo baseado num ficheiro, ficheiro esse que pode ser de um de dois tipos: em formato .txt cujo o conteúdo pode ser lido em qualquer editor de texto, ou em formato .bin que trás alguma segurança adicional, sendo que o mesmo só pode ser lido a partir do próprio jogo.

Guardar jogo

Durante o jogo, o utilizador pode guardar o jogo e retomar quando pretender. Ao selecionar a opção de “Guardar Jogo”, o sistema guarda as informações do jogador e adversário (posição, energia, etc) em ficheiro.

Carregar jogo

Esta função faz o inverso da função anterior, ou seja, permite ao jogador e adversário retomar o jogo no preciso momento que o deixou (voltando para a última sala visitada, energia que tinha, etc).

Funções secundárias: Gráficos

Embora o jogo seja todo feito em modo consola, decidi criar um aspeto gráfico para embelezar o jogo, tornando-o um pouco mais atrativo.

Gráfico(s) Menu Principal

Para este menu, a interface é bastante simples, tendo um castelo com as opções para o jogador escolher.

Gráfico(s) Banner Jogo

Esta é uma função que permite ao jogador visualizar, durante o jogo, a sua energia.

Gráfico(s) Créditos Finais

Quando um jogador termina o jogo (independentemente do resultado que obteve), surge um ecrã com o castelo como fundo, com as referências ao autor do trabalho.

Testes de Desenvolvimento

Durante o tempo de desenvolvimento do trabalho fui realizando algumas tarefas de testes de modo a verificar que todo o trabalho está a decorrer dentro da normalidade cumprindo o prazo. Esta funcionalidade servia para poder seguir para o próximo objetivo sabendo que o anterior está correto e a funcionar em pleno.

Estrutura do jogador

Esta função servia para verificar a integridade da estrutura do jogador. Para verificar bastou fazer um output para comparar os dados inseridos estavam corretamente inseridos.

Estrutura do adversário

Os testes para esta estrutura foram idênticos aos testes da estrutura do jogador.

Estrutura do Mapa

Aqui foram listadas os atributos das salas (coordenadas e descrição da sala).

Conclusão

Após estas semanas em que desenvolvi este segundo trabalho prático coloquei em prática os novos conteúdos adquiridos nas aulas (criação de threads e respetiva sincronização).

Como é comum neste tipo de desenvolvimento, foram surgindo ao longo do tempo as perguntas que todos fazemos, “Como se faz?”, “Como funciona?”, mas com a persistência em querer fazer e o sentido de querer perceber os objetivos foram atingidos. Gostaria de ter desenvolvido uma área que aprendemos nas aulas, a interface gráfica, mas com os problemas encontrados no desenvolvimento no jogo, o tempo “gasto” (no bom sentido) na pesquisa de como ultrapassar os obstáculos, acabei por não ter tempo de a fazer. Contudo tive a noção que não é impossível de se fazer uma interface gráfica, apenas precisamos de um pouco mais de tempo e concentração.

Posso acrescentar que o tema em si despertou a curiosidade, e ao mesmo tempo era desafiante, ou seja, a criação de jogos é sempre uma área apetecível e ao mesmo tempo, utilizar uma ferramenta por nós desenvolvida dá outro ânimo.

Webgrafia

<http://www.cplusplus.com/forum/beginner/5404/>
<http://www.mtm.ufsc.br/~azeredo/cursoC/aulas/c790.html>
<http://crasseux.com/books/tutorial/argc-and-argv.html>
<http://stackoverflow.com/questions/12320969/read-char-from-txt-file-in-c>
<http://stackoverflow.com/questions/11546177/how-to-read-lines-of-text-from-fileand-putthem-into-an-array>
<http://www.portugal-a-programar.pt/topic/8864-resolvido-texto-com-cor-em-c/>
<http://msdn.microsoft.com/en-us/library/ms682088.aspx>
<http://www.cplusplus.com/forum/general/81753/>
[http://www.vivaolinux.com.br/topico/C-C++/System\(pause\);/](http://www.vivaolinux.com.br/topico/C-C++/System(pause);/)
<http://stackoverflow.com/questions/15102976/how-to-clear-screen-from-simple-c-program>
<http://faq.cprogramming.com/cgi-bin/faqfaq.cgi#clear>
<http://www.inf.pucrs.br/~pinho/Laprol/Arquivos/ArquivosBinarios.htm>
<http://forum.zwame.pt/showthread.php?t=551594>
<http://stackoverflow.com/questions/4384309/reading-writing-a-structure-into-a-binary-file>
http://cms.ipbeja.pt/pluginfile.php/36121/mod_resource/content/2/html/so/prog_thr.htm

Anexos

Código completo

Devido à extensão do código, o mesmo foi armazenado no Google Code, podendo ser acessado através do endereço: <https://code.google.com/p/jogoaventurawin/Trabalho 2>.

Para o download do projeto completo (incluindo o presente relatório e apresentação), siga este endereço: https://www.dropbox.com/TG2_6219.zip.