

isep

instituto
superior de
engenharia do
porto



15/01/2014

Think & Code

Laboratório/Projecto 5



Think & Code

Conteúdo

Quem Somos	2
Base de dados	3
WebSite	4
Web Service.....	5
Prolog	6
PlEngine	6
Jogo	7
Infraestrutura computacional	9
Mini Jogos.....	10
Hang Man:	10
Tic-Tac-Toe:	11
Labirinto:	11

Quem Somos

A empresa Graphs4Social,S.A. é uma empresa que lançou o desafio, a jovens equipas de desenvolvimento de software, que consiste na construção de uma aplicação que manipula e visualiza grafos de redes sociais.

De uma forma geral, a empresa pretende que lhe seja apresentado um protótipo inicial que deverá ser constituído pelos seguintes módulos:

- Aplicação desktop para Visualização 3D;
- Website para manipulação da rede social;
- Componente de inteligência artificial para representação e inferência sobre o grafo e minijogos.

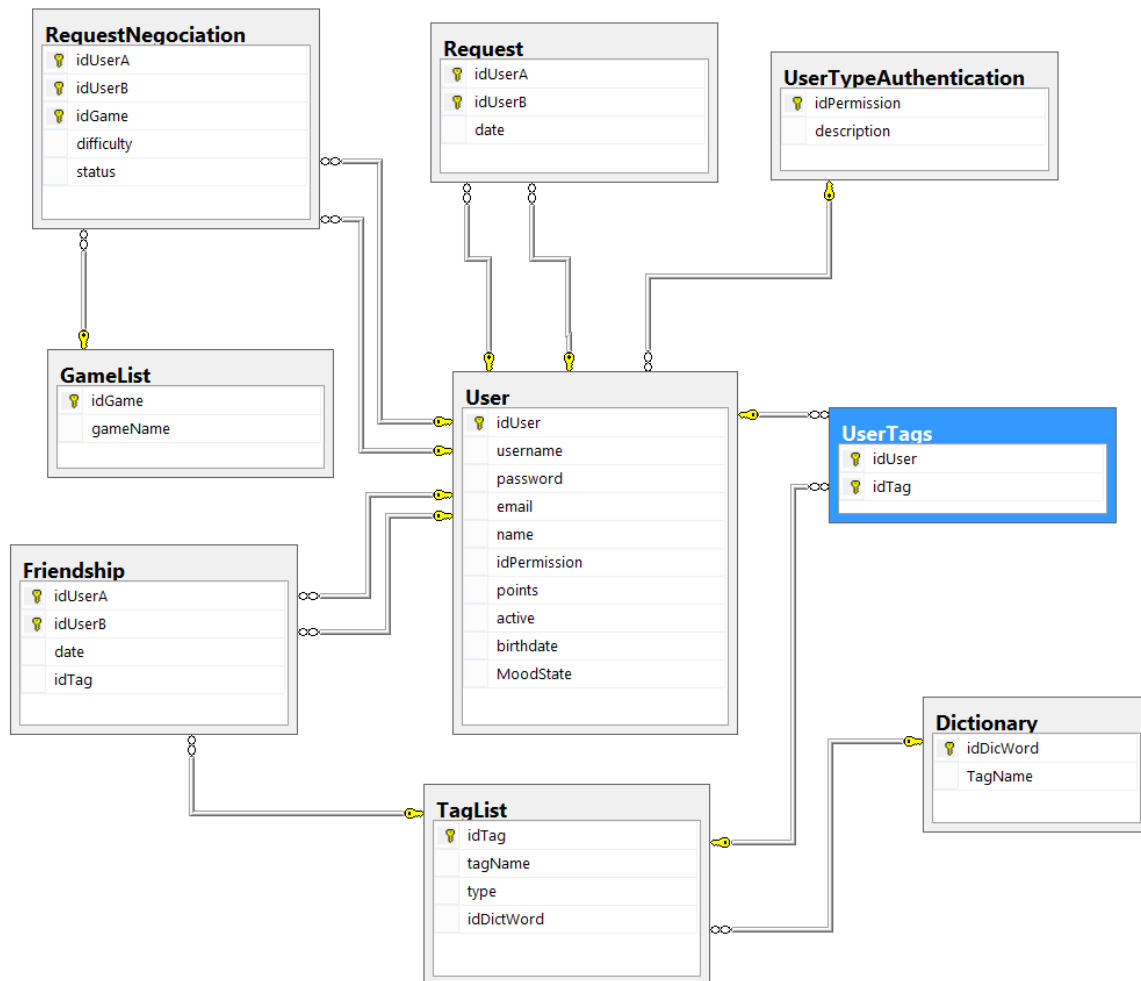
Este relatório visa relatar o trabalho desenvolvido pelo grupo Think&Code, no contexto do desafio proposto na disciplina de LAPRV.

Ao longo do relatório vamos dar a conhecer com mais exatidão o trabalho desenvolvido, ao longo das três semanas.

Começamos por apresentar e explicar a arquitetura do nosso sistema, recorrendo a uma *deployment view*, a uma *logical view* e ainda a um modelo de dados.

De seguida, cada módulo do projeto (navegação e visualização 3D, inteligência artificial, site e Infraestrutura computacional) tem um capítulo destinado a mostrar aquilo que foi implementado, de que forma foi e o que ficou aquém das expectativas de cada uma das componentes.

Base de dados



A base de dados utilizada é bastante simples de compreensão, onde tem por base a tabela User, como facilmente demonstramos na imagem acima. O SGBD utilizado para a sua implementação foi SQL Server, ficando armazenada no servidor.

O seu acesso pode ser feito através do servidor: wvm023.dei.isep.ipp.pt\SQLEXPRESS, com os seguintes dados de acesso: User e a palavra-passe: user.

WebSite

Este módulo consiste numa aplicação Web para suportar o jogo baseado na visualização e manipulação de grafos sociais.

Esta aplicação está dividida em vários componentes, sendo a interface com o utilizador e o seu BackOffice (páginas Web em ASP.NET), a camada de negócio (BLL – Business Logic Layer) e a camada de acesso a dados (DAL – Data Access Layer).

Durante o desenvolvimento deste site, foi criado um layout a pensar no utilizador, de modo a facilitar a sua navegação e não deixar que este se sinta “perdido” durante a sua visita ao site. As páginas (interface com o utilizador) foram desenvolvidas de forma coerente, e validam corretamente toda a informação que é introduzida pelo utilizador.

As páginas estão devidamente protegidas de modo a não estarem acessíveis a todos os utilizadores. Assim sendo, para qualquer utilizador não autenticado, está disponível uma página principal de apresentação, uma Leader Board, a dimensão da rede de utilizadores, a Tag Cloud dos utilizadores e a Tag Cloud das relações de utilizadores.

Em relação aos utilizadores autenticados, têm ao seu dispor uma página My Account que permite configurar a própria conta. Sendo assim, permite editar a informação pessoal (mudar password e mudar as definições pessoais), gerir as tags pessoais (criar, editar e remover tags) e mudar o estado de espírito do utilizador. Permite ainda mostrar a lista de amigos bem como removê-los. Mostra também a Tag Cloud dos amigos e das relações entre amigos.

No que diz respeito aos pedidos de amizade, quando o utilizador está autenticado, na visualização geral dos utilizadores do sistema, permite enviar pedido de amizade, ao qual o utilizador que recebeu o pedido tem a opção de aceitar, recusar ou desafiar, propondo ao utilizador interessado um desafio de ultrapassar determinado nível de um jogo.

Por fim, e ainda na secção da área de utilizador, o mesmo poderá fazer download do jogo que será instalado localmente na sua máquina e posteriormente poderá visualizar o seu grafo de rede ainda não implementado.

As camadas abaixo da interface (a BLL e a DAL) foram desenvolvidas e testadas em separado, de modo a que quando fossem usadas pela interface, se tornasse fácil e rápido desenvolver e testar a aplicação.

Tanto o site como a base de dados estão instalados nos servidores do DEI.

A conexão feita entre o Website para a base de dados processa-se através da biblioteca DataModel, organizada por tipo de classes: BLL para tratamento de dados recebidos da BD, Gateway para retornar dados da BD, Model que é o local onde estão colocados os Objectos de dados(Utilizadores, Tags, etc) e por ultimo uma zona com Ferramentas, que server unicamente para algoritmos de encriptação.

A conexão feita entre o Website para a base de dados processa-se através da biblioteca DataModel, organizada por tipo de classes: BLL para tratamento de dados recebidos da BD, Gateway para retornar dados da BD, Model que é o local onde estão colocados os Objectos de dados(Utilizadores, Tags, etc) e por ultimo uma zona com Ferramentas, que server unicamente para algoritmos de encriptação.

Web Service

O web Service(WCF) tem como objetivo passar dados entre a Base de Dados para a aplicação externa, nomeadamente o jogo. Como apenas é desejado que o jogo aceda ao Webservice foi criada uma palavra-passe extra, para que apenas um pedido que envie essa palavra-passe possa ser respondido com sucesso, evitando deste modo acessos indesejados, garantindo uma maior segurança de dados.

O funcionamento deste web service também é relativamente simples, dispomos a facilidade de aproveitamento da biblioteca DataModel. Dispomos de vários tipos de métodos, facilitando desta forma o desempenho do jogo.

Prolog

PlEngine

PlEngine é uma API de prolog para C++ e outras linguagens orientadas a objectos, que nos permite consultar e operar numa folha prolog fora do ambiente SWI.

Esta API oferece uma gama variada de opções para manipulação de predicados prolog:

- É possível obter resultados de predicados simples até predicados complexos de, por exemplo, pesquisa;

- As funções mais importantes desta API são:

 - PlEngine: recebe um ficheiro e consulta-o externamente ao SWI Prolog;

 - PlTermv: associa um predicado do ficheiro consultado;

 - PlQuery: devolve várias soluções para um determinado predicado;

 - PlCall: idêntico ao anterior, mas funciona melhor com predicados que não tenham mais do que uma solução.

Os projectos que usam esta API tiveram de ser devidamente configurados em termos de library e include directories. Foi também necessário acrescentar uma variável de ambiente com o caminho para a pasta do SWI Prolog.

Jogo

Para esta componente foi necessário criar uma aplicação em C++ tendo como objectivo a visualização de um grafo em 3D. Grafo este que permite ver as relações entre utilizadores bem como outras informações.

Para o login implementamos uma solução em C# que faz a conexão entre o jogo com um webservice de modo a verificar se o utilizador existe na base de dados.

O grafo é construído através da recepção de um vector constituído por tuples com a seguinte estrutura:

```
vector<tuple<int,vector<tuple<int,string>>,User_C>>
```

sendo o primeiro int o id do utilizador e o segundo vector com o tuple <int,string> a lista de amigos do utilizador.

Inicialmente começamos por estudar o projecto demo e as principais melhorias a realizar tais como passar as estruturas de dados para classes de modo simplificar a compreensão de código e implementar as funcionalidades pedidas.

Para se iniciar a construção do grafo, foram construídas esferas que representam um utilizador, sendo as coordenadas geradas automaticamente através da divisão pelo número de amigos do utilizador calculado por um algoritmo de distribuição radial. Cada cilindro representa uma ligação entre dois utilizadores. O z de cada utilizador é calculado tendo em conta o número de 0 a 20 e o raio da esfera segundo o número de tags associadas a esse utilizador.

As ligações entre utilizadores poderão existir no sentido norte-sul, este-oeste, na vertical e na diagonal e a espessura de cada ligação depende das forças correspondentes a uma dada relação.

A classe `lapr5.cpp` tem um método chamado `timer` que é responsável por verificar a detecção de colisões, a navegação do utilizador, a chamada da execução dos mini-jogos. É possível também seleccionar o tipo de navegação. O `timer` basicamente chama um evento sempre que o estado de uma tecla passa a `true`. Evento esse que está associado a cada tecla.

O `picking` tem como funcionalidade principal detectar objectos no modelo do jogo, sendo eles eixos, arcos ou nós. Com este método conseguimos fazer a detecção de colisões no jogo.

Para a câmara livre podemos utilizar as teclas “q,a,up,down,right,left”, e esta só percorre o grafo se não houver colisões entre a câmara e nós ou arcos.

Para mudar para a câmara rasante basta clicar na tecla `R` que é reconhecida no `timer` e coloca a câmara no utilizador que fez login. O nosso voo rasante funciona nos nós (subindo/descendo o nó) mas para o caso dos arcos por vezes para devido a estar mal direccionado.

A funcionalidade mini-mapa, tem como objectivo a projecção visua de todo o grafo da rede. Na sua implementação criamos um `viewport` no canto inferior esquerdo e configuramos uma vista de lado na câmara.

Após desenhar a projecção principal é redesenhada no `viewport` criado numa vista de lado e sendo actualizado consoante a posição da câmara.

Os efeitos de áudio foram apenas usados no mini-jogo (enforcado) no caso de acertar/errar uma letra ou então acertar/errar a palavra.

Infraestrutura computacional

De acordo com os requisitos, foi instalado os serviço de NAGIOS, NSclient++ e SNMP e configurados de modo a dar respostas às especificações dos requisitos.

No nosso servidor Windows está a ser monitorizado, o disco C:, Taxa de utilização do CPU, trafego da placa de rede, monitorização do Webservice e por fim do nosso Website.

No nosso servidor Linux está a ser monitorizado, o disco C:, total de procesos, ultimo login, numero de utilizadores logados e o check load do servidor Linux .

Mini Jogos

Hang Man:

O mini-jogo do enforcado é uma das componentes gráficas do jogo, neste caso foi implementado em duas dimensões.

O jogo é implementado na linguagem C++, e usa as bibliotecas gráficas Glut e Glui(para mapeamento de texturas).

Funcionamento:

O jogo funciona à base de texturas que são “binded” a quadrados desenhados pelo OpenGL: por exemplo, cada letra que o utilizador pode tentar adivinhar está ao seu dispor num teclado constituído por texturas que representam as letras do alfabeto.

A cada letra está associado um índice que representa a letra correspondente à imagem.

Supondo que o utilizador clica numa letra que pensa estar correcta, o programa age da seguinte forma:

- São verificadas as coordenadas do clique do rato(botão esquerdo), e de seguida valida-se se estas coordenadas “colidem” com as coordenadas de alguma letra do teclado do jogo;

- Caso esta validação passe, o programa recebe informação através de um array associativo, da letra (caracter) que corresponde à textura;

- Caso a letra corresponda à palavra que se vai adivinhar, o programa vê as coordenadas da letra nessa palavra, e substitui pela dita letra;

- A palavra a adivinhar está inicialmente exposta com os caracteres ‘_’(underscore);

- Caso o utilizador erre, o programa começa a desenhar o enforcado: começa por desenhar a cabeça, depois corpo e de seguida os quatro membros, fazendo com que o utilizador perca após 7 tentativas erradas;

- Para ganhar o utilizador tem de adivinhar tantas letras quanto o tamanho da palavra-chave.

No caso de haver mais de uma letra por palavra, por exemplo, se a palavra tem dois ‘a’, conta como duas ocorrências certas.

Há efeitos audios neste mini-jogo:

- Caso o utilizador acerte numa letra;

- Caso o utilizador erre numa letra;

- Caso o utilizador perca;

- Caso o utilizador ganhe.

Há também interacção com Prolog via PLEngine: o dicionário de palavras disponíveis está escrito em prolog, assim como as dicas para as palavras, dispostos em dois predicados: word/1 e hint/2, este último associa a palavra a uma linha de texto que dá uma pista ao utilizador;

O utilizador consegue saber quantas letras já acertou, e quantas faltam para ganhar.

Tic-Tac-Toe:

É outra componente gráfica do programa, também em duas dimensões.

Funcionamento:

O utilizador dispõe de um simples tabuleiro do jogo do galo, onde joga sempre primeiro, e joga com o círculo(O).

Cada divisão do tabuleiro está associada a uma inteiro, e a coordenadas X e Y.

O programa detecta e associa cliques do rato com as coordenadas do mesmo e as coordenadas das divisões, devolvendo um inteiro: de 1 a 9.

De seguida, joga a inteligência artificial:

- Programa Prolog que vê as melhores jogadas para impedir o utilizador de ganhar o jogo;
- A comunicação Prolog C++ é novamente feita via PlEngine;
- Cada vez que o utilizador ou o computador jogam, é acrescentado à base de conhecimento um facto(x/1 no caso da IA ou o/1 no caso do jogador);
- Estes factos têm como argumentos as posições que já foram jogadas.

A validação de vitória ou derrota é feita em C++:

- O programa verifica as hipóteses todas cada vez que é feita uma jogada;
- Se chegarmos ao fim do jogo sem vitória/derrota de nenhum lado, o programa devolve empate.

Labirinto:

O jogo funciona num jogo em 3D onde o utilizador terá que procurar pela única saída.

Funcionamento:

O labirinto é construído através de uma `matrix[21][41]` que só tem 0 e 1, onde 0 é o caminho que o utilizador pode caminhar e 1 são as paredes.

Cada movimento do utilizador está associado a duas coordenadas.