



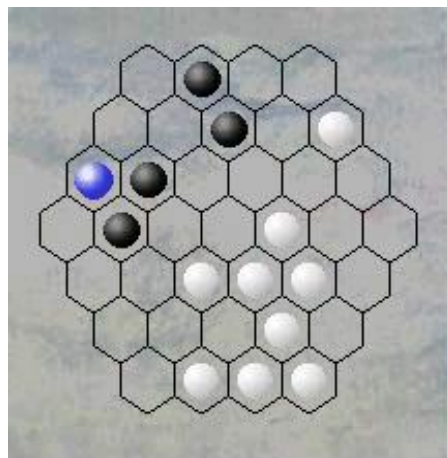
Universidade do Porto
Faculdade de Engenharia
FEUP

Nosferatu

Gaspar de Lacerda Aroso Furtado 080509072

Miao Sun 080509162

Grupo 207



Programação em Lógica

Mestrado Integrado em Engenharia

Informática e Computação

2010-09-28

Nosferatu

Gaspar de Lacerda Aroso Furtado 080509072

Miao Sun 080509162

Grupo 207

Programação em Lógica

Mestrado Integrado em Engenharia

Informática e Computação

2010-11-08

Trabalho realizado no âmbito da disciplina de xxx, do 1º semestre, do 2º ano, do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto, leccionada por Eugénio da Costa Oliveira, Daniel Moura e Rui Gomes.

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Informática e Computação
Rua Roberto Frias, s/n, 4200-465 Porto, Portugal

Resumo

Neste relatório pretende-se expor em detalhe o tema do trabalho e explicar como este foi realizado. Implementou-se na linguagem Prolog o jogo de tabuleiro Nosferatu.

Índice

| | |
|--|-----------|
| 1. Introdução | 9 |
| 1.1 Enquadramento | 9 |
| 1.2 Objectivos | 9 |
| 1.3 Estrutura do Relatório | 9 |
| 2. Descrição do Problema | 11 |
| 2.1 Primeira fase do jogo: inserção de peças | 11 |
| 2.2 Fases seguintes: jogar..... | 11 |
| 3. Trabalho do Autor | 13 |
| 4. Conclusões e Perspectivas de Desenvolvimento | 15 |

Capítulo 1

1. Introdução

1.1 Enquadramento

Este trabalho foi desenvolvido para a disciplina de Programação em Lógica e teve como principal objectivo a familiarização dos alunos com a linguagem Prolog. Implementou-se o jogo de tabuleiro Nosferatu.

1.2 Objectivos

O objectivo deste trabalho consiste em implementar em Prolog o jogo de tabuleiro Nosferatu.

Os objectivos específicos deste trabalho são os seguintes:

- Desenvolver um programa que permita que duas pessoas possam jogar;
- Implementar uma inteligência artificial de vários níveis, para que uma pessoa possa jogar contra o computador e computador vs computador;
- Implementar um protocolo com sockets para comunicar com a visualização 3D feita em c++ OpenGL que será desenvolvida no âmbito da disciplina de LAIG.

1.3 Estrutura do Relatório

Este trabalho encontra-se estruturado em 4 capítulos dos quais, o primeiro é composto por esta introdução ao trabalho.

No segundo capítulo é apresentado o problema.

No terceiro capítulo a implementação é explicada.

O último capítulo contém as conclusões gerais deste trabalho, analisa os seus principais resultados, e apresenta algumas perspectivas de desenvolvimentos futuros.

Capítulo 2

2. Descrição do Problema

O jogo a desenvolver, Nosferatu, é um jogo não muito conhecido, apenas foi inventado em 2001 e não é conhecida nenhuma edição deste em suporte físico. É um jogo de tabuleiro, com 12 peças brancas (aldeões), 5 peças pretas (vampiros), mais 1 peça preta especial, que é o Nosferatu. O jogo é jogado por 2 jogadores, um joga as peças brancas (aldeões), outro joga as peças pretas (vampiros e o Nosferatu). Não é só um jogo entre preto e branco, também é um jogo entre o bem e o mal. O objectivo dos pretos é comer todos os brancos, o objecto dos brancos é só capturar o Nosferatu.

2.1 Primeira fase do jogo: inserção de peças

O tabuleiro começa vazio. Enquanto estiverem peças fora do tabuleiro, a cada jogada o jogador tem que colocar uma peça no tabuleiro, desde que não seja numa posição em que esteja em risco imediato de ser capturada.

2.2 Fases seguintes: jogar

Após todas as peças terem sido colocadas, a cada jogada o jogador tem que fazer uma de duas coisas: se puder comer uma peça, tem que o fazer, senão, pode mover uma peça. O movimento das peças é parecido com o jogo “Damas” sendo que o Nosferatu é análogo a uma dama, e o movimento das peças normais difere apenas no facto de se poderem movimentar em qualquer direcção. A peça Nosferatu pode-se movimentar quantas casas se quiser ao longo de uma linha livre, enquanto que as outras peças apenas se

podem movimentar uma casa por jogada. O Nosferatu é ao mesmo tempo uma peça muito poderosa e uma peça vulnerável pois basta que esta seja apanhada para que se perca o jogo.

A captura de peças é análoga ao jogo “Damas”. Uma peça é capturada quando uma peça do adversário se encontra adjacente a esta e a casa “atrás” desta se encontra vazia. Não existe limite de captura de peças numa jogada. Após comer uma peça, o Nosferatu pode continuar a movimentar-se nessa linha quantas casas quiser.

Se um aldeão for capturado por um vampiro, sai do tabuleiro. Se um aldeão for capturado pelo Nosferatu pode-se fazer uma de duas coisas: o aldeão sai do tabuleiro, o aldeão transforma-se em vampiro. Se um vampiro for capturado por um aldeão, sai do tabuleiro.

Capítulo 3

3. Trabalho do Autor

Desenvolveu-se o programa que permite que dois jogadores joguem o jogo de tabuleiro Nosferatu, acima descrito. Foram implementados uma série de predicados para que todas as condições de jogo fossem validadas. Destes predicados destaca-se:

- `casa_valida(X,Y,P,Tab)`, que verifica se uma casa é válida para inserir a peça P, tendo em conta se está em risco de ser comida;
- `movimento_valido(X,Y,Xf,Yf,Tab)`, que apenas sucede se o movimento for válido, no contexto do jogo;
- `exec_move(X,Y,Xf,Yf,Tab,TabN)`, que modifica o tabuleiro “apagando” a posição (x,y) e inserindo a peça na posição (xf,yf), considerando se existe alguma peça que é comida ou não. Se alguma peça for comida, “apaga-a” do tabuleiro;
- `limpa_linha(O,X,Y,Xf,Yf,D,Tab,Ntab)`, que é usado para movimentos do Nosferatu, considerando se come peças, ou as transforma em vampiros, ou apenas se move.
- `ganhou(J,Tab)`, que apenas sucede quando algum jogador ganhou, devolvendo o mesmo em J.

| | | | | | | | | |
|---|---|---|---|---|---|----|----|---|
| | A | B | C | D | E | F | G | |
| 1 | | | □ | □ | □ | | | 1 |
| 2 | | □ | □ | | | \$ | | 2 |
| 3 | □ | □ | | | □ | \$ | \$ | 3 |
| 4 | □ | | | □ | | | \$ | 4 |
| 5 | □ | | | | | | \$ | 5 |
| 6 | | □ | | | | N | | 6 |
| 7 | | | | | | | | 7 |
| | A | B | C | D | E | F | G | |

Figura 1. Vista do tabuleiro após algumas jogadas

Capítulo 4

4. Conclusões e Perspectivas de Desenvolvimento

Neste trabalho apenas foi cumprido um dos três objectivos a que se propunha: a função pessoa vs pessoa. De notar que pouco falta para que o programa jogue sozinho, todos os predicados necessários a uma inteligência artificial cega (pegar na primeira possibilidade) estão implementados. Quanto à interface com a visualização 3D através de sockets, nada foi implementado.

Os melhoramentos necessários a este trabalho são óbvios: implementar e cumprir todos os objectivos a que se propunha.

Referências Bibliográficas

<http://paginas.fe.up.pt/~eol/LP/1011/>

<http://paginas.fe.up.pt/~eol/LP/1011/documents/Trabalhos/Exemplo1.pdf>

<http://paginas.fe.up.pt/~eol/LP/1011/documents/Trabalhos/Exemplo2.pdf>

<http://paginas.fe.up.pt/~eol/LP/1011/documents/Trabalhos/Exemplo3.pdf>

<http://paginas.fe.up.pt/~eol/LP/1011/documents/Trabalhos/Exemplo4.pdf>

<http://paginas.fe.up.pt/~eol/LP/1011/documents/Trabalhos/Exemplo5.pdf>

<http://paginas.fe.up.pt/~eol/LP/1011/documents/Trabalhos/Exemplo6.pdf>

<http://homepages.di.fc.ul.pt/~jpn/gv/nosferatu.htm>

<http://paginas.fe.up.pt/~eol/LP/1011/Materiais.htm>

<http://www.lix.polytechnique.fr/~liberti/public/computing/prog/prolog/prolog-tutorial.html>

