



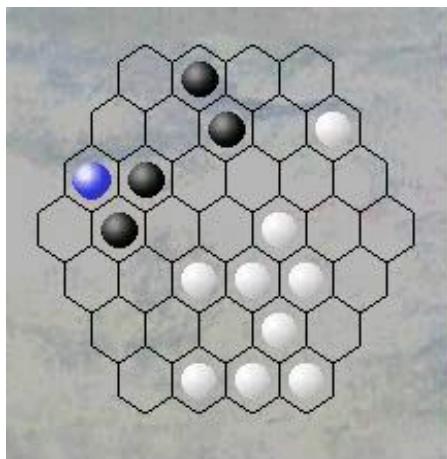
Universidade do Porto
Faculdade de Engenharia
FEUP

Relatório Intercalar – 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

Resumo:

Neste relatório pretende-se expor em detalhe o tema do trabalho bem como a abordagem à sua realização. Pretende-se implementar na linguagem Prolog o jogo de tabuleiro Nosferatu.

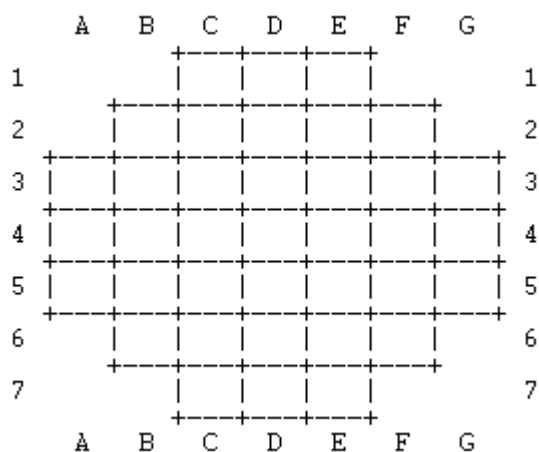
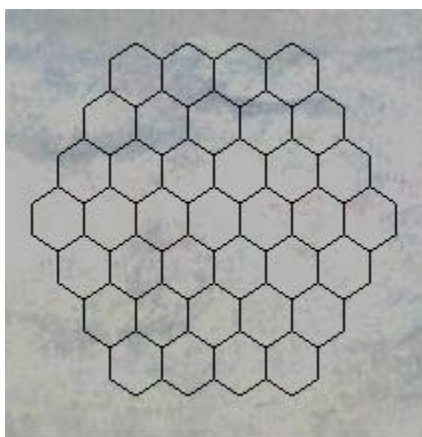
1 – Introdução

O objectivo deste trabalho é aprender a programar em Prolog desenvolvendo o jogo de tabuleiro “Nosferatu”. Este jogo foi escolhido pela sua parecença com o jogo “Damas” e pela simplicidade das suas regras que também permitem alguma complexidade de jogo. O principal desafio deste trabalho será o desenvolvimento de uma inteligência artificial com três níveis de dificuldade que permita jogar contra o computador ou um jogo computador vs computador. O trabalho incluirá uma interface 3D para visualização e manipulação do jogo, utilizando a tecnologia OpenGL e a linguagem C++, que será desenvolvida na disciplina LAIG.

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.

Neste trabalho usamos um tabuleiro com um formato um pouco diferente do original, representado abaixo, sendo que o do lado esquerdo é o original.



O tabuleiro começa vazio. Enquanto estiverem peças fora do tabuleiro 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.

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.

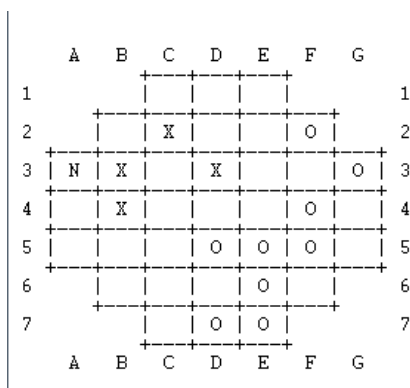
Após todas as peças terem sido colocadas, a cada jogada o jogador tem que fazer uma de duas coisas: se puder comer uma peça, tem que o fazer, senão, pode mover uma peça. O movimento das peças é parecido com o jogo “Damas” sendo que o Nosferatu é análogo

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.

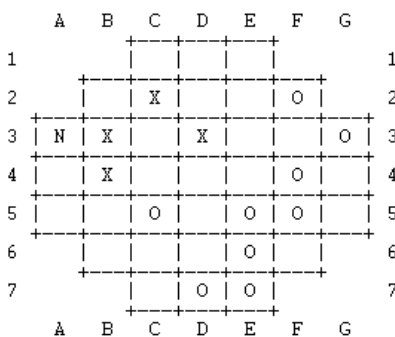
Se um aldeão for capturado por um vampiro, sai do tabuleiro. Se um aldeão for capturado pelo Nosferatu pode-se fazer uma de duas coisas: o aldeão sai do tabuleiro, o aldeão transforma-se em vampiro. Se um vampiro for capturado por um aldeão, sai do tabuleiro.

Abaixo estão representadas uma serie de jogadas exemplo:

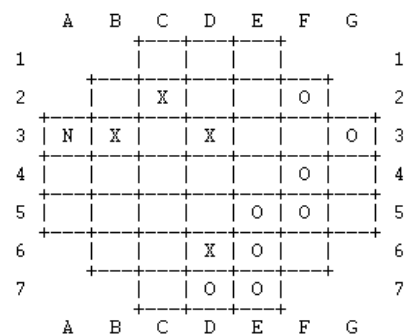
(Aldeão (O), Vampiros (X), Nosferatu (N))



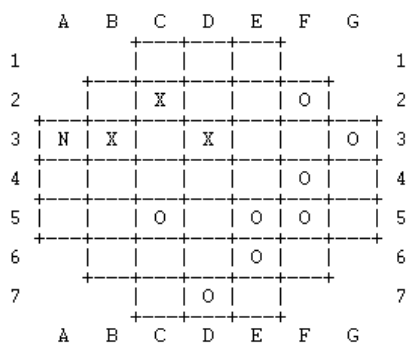
①



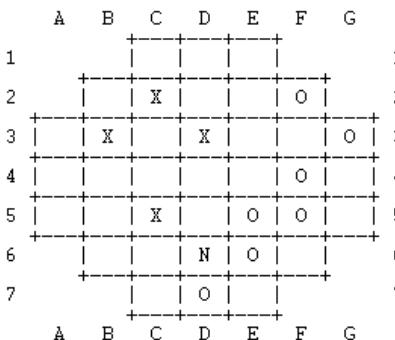
②



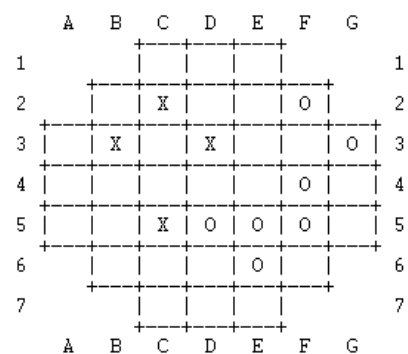
③



④



⑤



⑥

3 - Representação do Estado do Jogo:

Vamos usar uma lista de sete listas para representar o tabuleiro do jogo. Sendo que cada uma destas sete listas corresponde a uma linha do tabuleiro.

Inicialmente o tabuleiro está vazio, neste caso usamos a letra ‘V’ para representar o estado vazio, abaixo encontra-se uma representação do estado inicial:

```
estadoInicial([      [V,V,V],
                     [V,V,V,V,V],
                     [V,V,V,V,V,V,V],
                     [V,V,V,V,V,V,V],
                     [V,V,V,V,V,V,V],
                     [V,V,V,V,V],
                     [V,V,V]
                ]).
```

E o estado após algumas jogadas:

```
possivelEstado([      [V,V,V],
                      [V,X,V,V,O],
                      [V,V,V,X,V,V,V],
                      [N,X,V,V,V,O,V],
                      [V,X,V,O,O,O,V],
                      [V,V,O,O,V],
                      [V,O,O]
                ]).
```

4 - Representação de um Movimento:

Todas as jogadas serão inseridas com base nas posições do tabuleiro. No caso de ser uma inserção de uma nova peça, apenas é necessário uma coordenada (a posição onde se pretende inserir), no caso de ser um movimento de uma peça, é necessário dizer qual é a posição da peça e uma outra posição para onde a quer mover. Vamos usar dois predicados distintos, um para cada situação. Abaixo estão representados os predicados de inserção e movimento, respectivamente:

```
inserir_peca(jogador, posicao, tabuleiro, novo_tabuleiro)
```

Após verificar a validade da inserção da peça, gera a lista de listas representativa do tabuleiro resultante da operação, devolvida em `novo_tabuleiro`. Esta manipulação apenas consiste na troca do conteúdo de duas posições da representação do tabuleiro.

```
movimento(jogador, posicao, nova_posicao, tabuleiro, novo_tabuleiro)
```

Após verificar a validade do movimento, gera a matriz resultante do estado do jogo, devolvida em `novo_tabuleiro`. Nesta situação a manipulação da matriz consiste na troca do conteúdo de duas ou mais posições da mesma, dependendo se alguma peça foi capturada ou não. A verificação de uma peça ser capturada ou não será feita através da análise das posições inicial e final da peça que se movimentou.

No momento de criação do jogador, será também atribuída uma equipa a este (branca ou preta). Em função disto os jogadores terão um *plafond* de peças para inserir. Seguem-se inserções sucessivas até que se esgotem as peças para inserção. Vamos usar um predicado que gira este ponto, chamando ou `insere_peca` ou `movimento`.

5 - Visualização do Tabuleiro:

O tabuleiro do jogo será representado em modo de texto, utiliza o carácter ‘O’ representa as peças brancas (aldeões), o carácter ‘X’ para as peças pretas (Vampiros), e o carácter ‘N’ para a peça especial (o Nosferatu). Também estão apresentados os coordenados no tabuleiro, para que seja fácil a visualização das posições.

Abaixo está representado o tabuleiro em modo texto:

	A	B	C	D	E	F	G	
1			+	+	+	+		1
2		+	+	+	+	+	+	2
3	+	+	+	+	+	+	+	3
4	+	+	+	+	+	+	+	4
5	+	+	+	+	+	+	+	5
6		+	+	+	+	+	+	6
7			+	+	+	+		7
	A	B	C	D	E	F	G	

O tabuleiro após algumas jogadas, em modo texto:

	A	B	C	D	E	F	G	
1			+	+	+	+		1
2		+		X			O	2
3	+	+			X			3
4	+	N	X				O	4
5	+		X		O	O	O	5
6		+			O	O		6
7			+		O	O		7
	A	B	C	D	E	F	G	

6 - Conclusões e Perspectivas de Desenvolvimento:

Apesar do jogo ser pouco conhecido, tem muitas semelhanças com o jogo “Damas” que é um jogo muito conhecido. A simplicidade das regras permite que um jogador que nunca tenha jogado antes se familiarize rapidamente com o funcionamento do jogo, especialmente se já conhecer o jogo “Damas”.

O trabalho será desenvolvido em conjunto e em três partes: programação da visualização em modo de texto e interface para o utilizador, programação da inteligência artificial de vários níveis, e, por último, programação das sockets para comunicação com a interface gráfica 3D desenvolvida na disciplina de LAIG.

Tendo em conta as dificuldades que iremos encontrar visto o desenvolvimento ser feito numa linguagem que não nos é familiar, e a implementação da inteligência artificial ser algo complexa, estimamos que falte fazer 95% do trabalho.

Bibliografia:

<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>

Anexos:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Nosferatu em Prolog %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gaspar Furtado & Miao Sun %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Grupo207 %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOG MIEIC 2010/2011 %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
:-use_module(library(lists)).
```

```
peca(v, ' ').
peca(o, 'O').
peca(x, 'X').
peca(n, 'N').
```

```
print_cell(v):-write(' ').
print_cell(o):-write('O').
print_cell(x):-write('X').
print_cell(n):-write('N').
```

```
mainmenu:-
    write('*****'),nl,
    write('*'),nl,
    write('* Bemvindo ao Nosferatu *'),nl,
    write('*'),nl,
    write('*****'),nl,nl.
```

```
modJogo:-
    write('*****'),nl,
    write('*'),nl,
    write('* Escolhe o mode do jogo: *'),nl,
    write('*'),nl,
    write('* 1.Humano VS Humano *'),nl,
    write('* 2.Humano VS Computador *'),nl,
    write('* 3.Computador VS Computador *'),nl,
    write('*'),nl,
    write('*****'),nl,nl.
```

```
nivelJogo:-
    write('*****'),nl,
    write('*'),nl,
    write('* Nivel do Jogo *'),nl,
    write('*'),nl,
    write('* 1.Easy *'),nl,
    write('* 2.Normal *'),nl,
    write('* 3.Hard *'),nl,
    write('*'),nl,
    write('*****'),nl,nl.
```

% V representa estado vazio.

```
estadoInicial([      [v,v,v],
                     [v,v,v,v,v],
                     [v,v,v,v,v,v,v],
                     [v,v,v,v,v,v,v],
                     [v,v,v,v,v,v,v],
                     [v,v,v,v,v],
                     [v,v,v]
                ]).
```

% X representa peca preta(Vampiro), O representa peca branca(aldeao), e N peca especial(Nosferatu)

```
possivelEstado([    [v,z,v],
                    [v,x,d,v,o],
                    [v,v,v,w,v,v,v],
                    [n,x,v,q,v,o,v],
                    [v,x,v,o,o,o,v],
                    [v,v,o,o,v],
                    [v,o,o]
                ]).
```

```
inicio:-
    % mainmenu, modJogo, nl,nl,nivelJogo,
    nl,nl,tabuleiro.
```

```
mostra_tabuleiro(Tab):- write('  A  B  C  D  E  F  G  '),nl,
                        % linhas([Tab]),
                        write('  A  B  C  D  E  F  G  '),nl.
```

```
%linhas([]).
%linhas([H|T]):- write(' +---+---+---+---+---+---+---+ '), nl, linha(T).
```

```
%linha([]).
%linha([H|T]):- write(H), write(' '), linhas(T).
```