

Universidade de Coimbra - Faculdade de Ciências e Tecnologia
Departamento de Matemática



VISUALIZAÇÃO COMPUTACIONAL - 2005/2006
4ºANO

Projecto - O Jogo do Galo

Coimbra, 2 de Abril de 2006

Trabalho realizado por:



Augusto Bruno Rodrigues Pinho



Sara Margarida Gaspar Silva

NO ÂMBITO DA CADEIRA DE VISUALIZAÇÃO COMPUTACIONAL

Índice

Objectivo	4
Capítulo I	5
A OpenGL	5
Capítulo II	6
2.1 Jogador1 X Jogador2	7
2.2 Jogador X Computador	9
2.3 Computador X Computador	12
2.4 Funções utilizadas	14
2.5 Análise dos resultados	20

Objectivo

O presente trabalho tem por objectivo implementar uma versão interactiva do **Jogo do Galo**, utilizando a OpenGL, empregando a linguagem de programação C/C++.

O **Jogo do Galo** possui três modos de funcionamento:

- Computador X Computador (automático, sem intervenção do utilizador)
- Jogador X Computador
- Jogador1 X Jogador2

No Capítulo I será feita uma breve referência sobre o que é a OpenGL analisando-se algumas Bibliotecas e o seu contexto histórico. O Capítulo II destina-se a explicação da abordagem seguida e análise dos resultados.

Capítulo I

A OpenGL

A OpenGL (Open Graphics Library) foi desenvolvida em 1992 pela Silicon Graphics, maior empresa de computação gráfica do mundo. Por ser uma ferramenta portátil e rápida, é amplamente utilizada na construção de jogos electrónicos, tal como o jogo do galo, entre outros. É importante não confundir: OpenGL não é uma linguagem de programação, e sim uma eficiente API¹ (Application Programming Interface). Quando se diz que um programa é escrito em OpenGL significa que são feitas uma ou mais chamadas às suas funções.

A OpenGL é considerada uma ferramenta relativamente simples, já que, para se alcançar a aparência desejada, é preciso apenas determinar os passos que devem ser feitos para tal finalidade, poupando assim tempo na construção de qualquer programa. Para isso são oferecidas várias primitivas de baixo nível, sistemas de iluminação, coloração, textura e diversos outros recursos. O sistema de gerenciamento de janelas é feito pela biblioteca GLUT (OpenGL Utility Toolkit). A GLU (OpenGL Utility library) fornece-nos varias funções auxiliares, principalmente primitivas gráficas.

A OpenGL funciona como uma máquina de estados, ou seja, uma parte da memória é reservada o estado actual das diversas variáveis envolvidas no processo de renderização. Ao desenhar algo, todos esses valores armazenados na memória são usados para a sua representação no ecrã, ou seja, primeiro é necessário fornecer todas as características do objecto para só depois desenhá-lo.

¹ Isto significa que OpenGL trabalha entre o Software e o Hardware, simplificando os cálculos necessários para as transformações vectoriais e matriciais.

Capítulo II

O projecto Jogo_do_Galo possui três versões de jogo Galo, para evitar repetições de código foram criadas varias funções que vão ser chamadas em alturas diferentes por funções distintas. Para uma mais fácil explanação será feita a análise do código para cada uma das versões de uma forma esquemática, em que o leitor terá a oportunidade de analisar a interligação das várias funções. Mais á frente² será feita uma descrição mais detalhada das funções utilizadas.

² No capítulo 2.5

2.1 Jogador1 X Jogador2

Existem 2 situações possíveis, que serão analisadas separadamente.

1º Situação

Se ainda não foi encontrado o vencedor e ainda não foram esgotadas as 9 jogadas faz:



desenha_galo();

- **desenha_cabecalho ();**
 - desenha_quadrado();
 - escrevetexto (... , Jogo do Galo);
 - Consoante o jogador*
 - escreve texto (... , Jogador ...);

- **desenha_rodape();**
 - desenha_quadrado();
 - escrevetexto();

- **desenha_quadrados ();**

Efectua uma das seguintes operações (a) ou (b):

(a) desenho quadrados_bonecos():

Escolhe a cor:

- desenha_quadrado

Para cada posição (k,l) e consoante o jogador

- desenha_Boneco(k, l);

ou

- desenha_Comp(k, l);

(b) desenha quadrados_imagem():

- Desenha textura consoante o jogador;

- **desenha_linhas();**

2º Situação

Se estivermos no início do jogo:

- Inicia a matriz:

Inicia_galo ()

- Começa a decorrer o jogo

Inicio=0;

- Numero de jogadas

Jogadas=0

- Define o 1º jogador

Jogador =A



vitoria ():

■ **Se o número de jogadas for superior ou igual a 5 e ainda não foi encontrado o vencedor faz:**

- linhas();
- pintarvencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador d);
- escrevetexto(..., Ganhou o Jogo !!);

■ **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- colunas();
- pintarvencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador D);
- escrevetexto(..., Ganhou o Jogo !!);

■ **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- diagonal();
- pintarvencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador D);
- escrevetexto(..., Ganhou o Jogo !!);



swap (A, B)

2.2 Jogador X Computador

Existem 2 situações possíveis, que serão analisadas separadamente.

1º Situação

Se ainda não foi encontrado o vencedor e ainda não foram esgotadas as 9 jogadas faz:



desenha_galo();

■ **desenha_cabecalho ();**

- desenha_quadrado();
- escrevetexto (... , Jogo do Galo);

Consoante o jogador

- escreve texto (... , Jogador ...);

■ **desenha_rodape();**

- desenha_quadrado();
- escrevetexto();

■ **desenha_quadrados();**

Efectua uma das seguintes operações (a) ou (b):

(a) desenho quadrados_bonecos();

Escolhe a cor:

- desenha_quadrado

Para cada posição (k,l) e consoante o jogador

- desenha_Boneco(k, l);

ou

- desenha_Comp(k, l);

(b) desenha quadrados_imagem();

- Desenha textura consoante o jogador;

■ **desenha_linhas();**

2º Situação

Se estivermos no início do jogo:

- Inicia a matriz:

Inicia_galo ()

- Começa a decorrer o jogo

Inicio=0;

- Numero de jogadas

Jogadas=0

- Define o 1º jogador



vitoria ():

- **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- linhas();
- pintavencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador d);
- escrevetexto(..., Ganhou o Jogo !!);

- **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- colunas();
- pintavencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador d);
- escrevetexto(..., Ganhou o Jogo !!);

- **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- diagonal();
- pintavencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador d);
- escrevetexto(..., Ganhou o Jogo !!);



swap (A, PC1);



Se ainda não foi encontrado o vencedor e for o computador a jogar e o número de jogadas inferior a nove.

- **jogarpc()**
 - **inic_Random();**

- **jogadainteligente(jogador1, jogador2)**

Se não jogou faz:

Enquanto não está ocupado (ocupado=0) e o número de jogadas for inferior a 10 faz:

- i=aleatorio();

- j=aleatorio();

- Se não está ocupada a posição i, j

(if ocupado(i,j)=0) passa a ocupada (ocupado(i,j)=1) e foi efectuada mais uma jogada.

- Sleep(700); (Pausa de 7 segundos)

2.3 Computador X Computador

Existem 2 situações possíveis, que serão analisadas separadamente.

1º Situação

Se ainda não foi encontrado o vencedor e ainda não foram esgotadas as 9 jogadas faz:



desenha_galo():

- **desenha_cabecalho ();**
 - desenha_quadrado();
 - escrevetexto (... , Jogo do Galo);

Consoante o jogador

 - escreve texto (... , Jogador ...);

- **desenha_rodape();**
 - desenha_quadrado();
 - escrevetexto();

- **desenha_quadrados();**

Efectua uma das seguintes operações (a) ou (b):

(a) desenho quadrados_bonecos():

Escolhe a cor:

- desenha_quadrado

Para cada posição (k,l) e consoante o jogador

- desenha_Boneco(k, l);

ou

- desenha_Comp(k, l);

(b) desenha quadrados_imagem():

- Desenha textura consoante o jogador;

- **desenha_linhas();**

2º Situação

Se estivermos no início do jogo:

- Inicia a matriz:

Inicia_galo ()

- Começa a decorrer o jogo

Inicio=0;

- Numero de jogadas

Jogadas=0

- Define o 1º jogador



vitoria ():

- **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- linhas();
- pintavencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador d);
- escrevetexto(..., Ganhou o Jogo !!);

- **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- colunas();
- pintavencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador d);
- escrevetexto(..., Ganhou o Jogo !!);

- **Se o numero de jogadas for superior ou igual a 5 a ainda não foi encontrado o vencedor:**

- diagonal();
- pintavencedor();

Consoante o jogador vencedor

- escrevetexto(..., Jogador d);
- escrevetexto(..., Ganhou o Jogo !!);



Se ainda não encontrou o vencedor e não efectuou as nove jogadas faz:

➤ **swap (PC1, PC2)**

- **jogarpc()**

- **inic_Random();**
- **jogadainteligente(jogador1, jogador2)**

Se não jogou faz:

Enquanto não está ocupado (ocupado=0) e o número de jogadas for inferior a 10 faz:

- **i=aleatorio();**

- `j=aleatorio();`
- Se não está ocupada a posição `i, j` (`if ocupado(i,j)=0`) passa a ocupada (`ocupado(i,j)=1`) e foi efectuada mais uma jogada.
- `Sleep(700);` (Pausa de 7 segundos)

2.4 Funções utilizadas

1. `void inic_Random();`

Função responsável por inicializar a semente dos números aleatórios.

2. `int aleatorio()`

Vai criar número aleatórios cujos os valores são 0,1 ou 2 recorrendo à função `rand()`.

3. `void inicia_galo()`

Vai inicializar uma matriz 3 por 3 com o valor de 100. Cada posição da matriz corresponde à posição que um determinado jogador ocupa. Através da matriz `galo[MAX][MAX]` vai ser possível saber se uma determinada posição está ocupada ou não.

4. `void swap(int j, int i)`

Esta função vai ser responsável por trocar o jogador

5. `int ocupado(int i, int j)`

Os argumentos *i* e *j* vão estar associados à posição da matriz `galo[i][j]` inicializada em (3) se `galo[i][j] != 100` a posição está ocupada(retornando 1) estando livre(retornando 0) caso contrario.

6. `void desenha_linhas()`

Vai desenhar as linhas delimitadoras de cada quadrado do jogo galo.

7. `void desenha_quadrado(GLint i, GLint j)`

Desenha um rectângulo, em que um dos pontos vai ter coordenadas (*i* , *j*) e lados *xmin* e *ymin*.

8. `void desenha_Comp(GLint j, GLint i)`

Está função vai ser responsável por desenhar um computador na posição *j* e *i* do jogo do galo.

9. `void desenha_Boneco(GLint j, GLint i)`

Está função vai ser responsável por desenhar um Boneco na posição *j* e *i* do jogo do galo.

10. `void desenha_quadrados_bonecos()`

Vai desenhar os quadrados do jogo. Caso a posição não esteja ocupada o quadrado vai ter cor azul celeste. No caso da posição estar ocupada vai possuir a cor correspondente ao jogador e respectiva animação (boneco ou computador).

11. `void desenha_quadrados_imagem()`

Vai desenhar os quadrados do jogo. Caso a posição não esteja ocupada o quadrado vai ter uma textura inicial. No caso da posição não estar ocupada vai possuir a cor correspondente ao jogador e respectiva animação (textura1 ou textura2).

12. void `desenha_quadrados()`

Vai chamar as funções (10) ou (11)

13. void `escrevertexto(float x, float y, void *font, char *string)`

Vai começar a escrever o texto (`char*string`) na posição `x` , `y` com o tipo de letra e tamanho defenido por `void*font`.

14. void `desenha_cabecalho()`

Vai desenhar o rectângulo do cabeçalho e escrever o texto chamando a função `escretexto(...)`.

15. void `desenha_rodape()`

Vai desenhar o rectângulo do rodapé e escrever o texto chamando a função `escretexto(...)`.

16. int `linha(int a, int b, int c)`

Os inteiros `a`, `b` e `c` são dados por³,

```
a=galo[r][0];  
b=galo[r][1];      com r=0,1,2  
c=galo[r][2];
```

Se `a=b=c` são iguais o jogador (variável global que está definida no tipo do jogo em questão) fez linha devolvendo o valor 1 caso contrario devolve o valor 0.

17. int `coluna(int a, int b, int c, int j)`

Os inteiros `a`, `b` e `c` são dados por³,

```
a=galo[0][r];  
b=galo[1][r];      com r=0,1,2  
c=galo[2][r];
```

³ Estes valores serão calculados na função `vitoria()`

Se $a=b=c$ são iguais o jogador (variável global que está definida no tipo do jogo em questão) fez coluna devolvendo o valor 1 caso contrario devolve o valor 0.

18. `int diagonal(int a, int b, int c, int j)`

Os inteiros a, b e c são dados por⁴

```
a=galo[0][0];          a=galo[0][2];
b=galo[1][1];          ou  b=galo[1][1];
c=galo[2][2];          c=galo[2][0];
```

Se $a=b=c$ são iguais o jogador (variável global que está definida no tipo do jogo em questão) fez diagonal devolvendo o valor 1 caso contrario devolve o valor 0.

19. `void pintarvencedor()`

Consoante o jogador⁵ (A=11, B=22, PC1=33 e PC2=44) a função *pintarvencedor* vai escrever o vencedor no ecrã chamando a função *escreve texto(float x,float y, void *font, char *string)*.

20. `void vitoria()`

Vai chamar as funções linhas colunas e diagonal para ver se há vencedor. A função vitoria vai chamar as funções

- linhas (int a, int b, int c)
- colunas (int a, int b, int c)
- diagonal (int a, int b, int c)

para ver se existe vencedor. Caso exista vai chamar a função *pintarvencedor()*.

⁴ Estes valores serão calculados na função vitoria ()

⁵ O jogador é uma variável global que corresponde ao vencedor actual no momento da chamada nesta função. A variável jogador é alterada na função *desenha_jogo_jj()* ou *desenha_jogo_pj()* ou *desenha_jogo_pcpc()*, consoante o jogo em questão. Cada jogador esta associado a um número. O jogador A tem associado o número 11. O jogador B tem associado o número 22. O jogador PC1 tem associado o número 33 e o jogador PC2 o número 44.

21. `int jogadainteligente (int jogador1, int jogador2)`

Vai testar as jogas do adversário duas a duas de forma a tapar uma terceira jogada do outro jogador.

22. `void jogarpc()`

O computador vai tentar efectuar uma jogada inteligente, caso não consiga joga aleatoriamente.

23. `void init(void)`

Vamos inicializar a janela do Windows e ler do ficheiro as texturas.

24. `void desenha_galo()`

A função `desenha_galo()` vai chamar as funções `desenha_cabecalho()`, `desenha_rodape()`, `desenha_quadrados()` e `desenha_linhas()`, desenhando desta forma o jogo do galo.

25. `void desenha_jogo_jj(void)`

Ver Capítulo 2.2

26. `void desenha_jogo_pj(void)`

Ver Capítulo 2.3

27. `void desenha_jogo_pcpc(void)`

Ver Capítulo 2.4

28. `void desenhar_saida()`

Função que vai desenhar o menu de saída, chamando a função `escrevetexto(...)`

29. `void desenha_menu()`

Função que vai desenhar o menu, chamando a função `escrevetexto(...)`

30. void desenha()

Consoante a opção do utilizador vai chamar as funções

- desenha_menu();
- desenha_jogo_jj();
- desenha_jogo_pj();
- desenha_jogo_pcpc();
- desenhar_saida();

desenhando as janelas predefinidas.

31. void mainMenu(GLint mMOption)

Vai definir o menu da tecla direita do rato, tendo como opções:

- Imagens ;
- Desenhos ;
- Novo jogo;

32. void submenu(GLint Option)

Vai definir um novo menu, chamado Jogo que pode ser chamado pela tecla direita do rato tendo como opções:

- Menu Principal
- Jogador X Jogador;
- Jogador X Computador;
- Computador X Computador;
- Sair

33. void accao(int x, int y)

Vai definir uma área de clique para cada posição do jogo galo.

Para cada jogada possível vai guardar a informação da jogada na matriz galo[][].

34. void mouse (int button, int state, int x, int y)

Vai definir a acção do rato na área de clique definida na função 33.

35. void keyboard(unsigned char key, int x, int y)

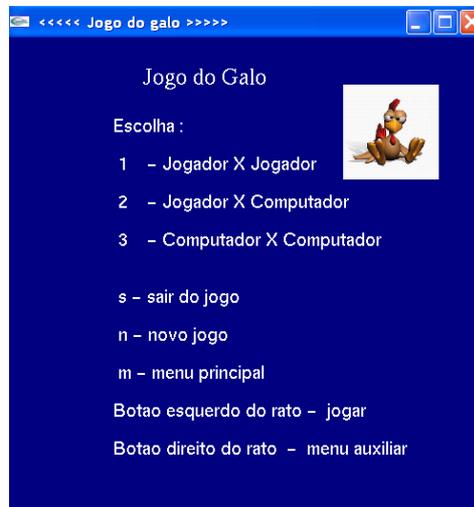
Consoante a opção do jogador (tecla m, n,s,1,2 ou 3) vai desencadear a respectiva acção.

36. void dimensao(GLint newWidth, GLint newHeight)

A função dimensao(GLint newWidth, GLint newHeight) vai ser responsavel por redimensionar a janela.

2.5 Análise dos resultados

O utilizador ao iniciar o jogo vai ter a oportunidade de escolher uma das seguintes opções:



Ao premir a tecla 2 vamos dar inicio ao jogo *Jogador X Computador*



No caso de haver um vencedor aparecera no ecrã uma imagem a congratular o vencedor

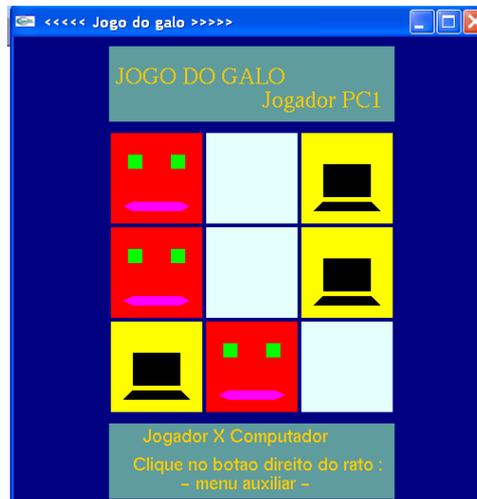


No decorrer do jogo ou no menu inicial o utilizador tem a oportunidade de utilizar o menu auxiliar definido a partir da tecla direita do rato,



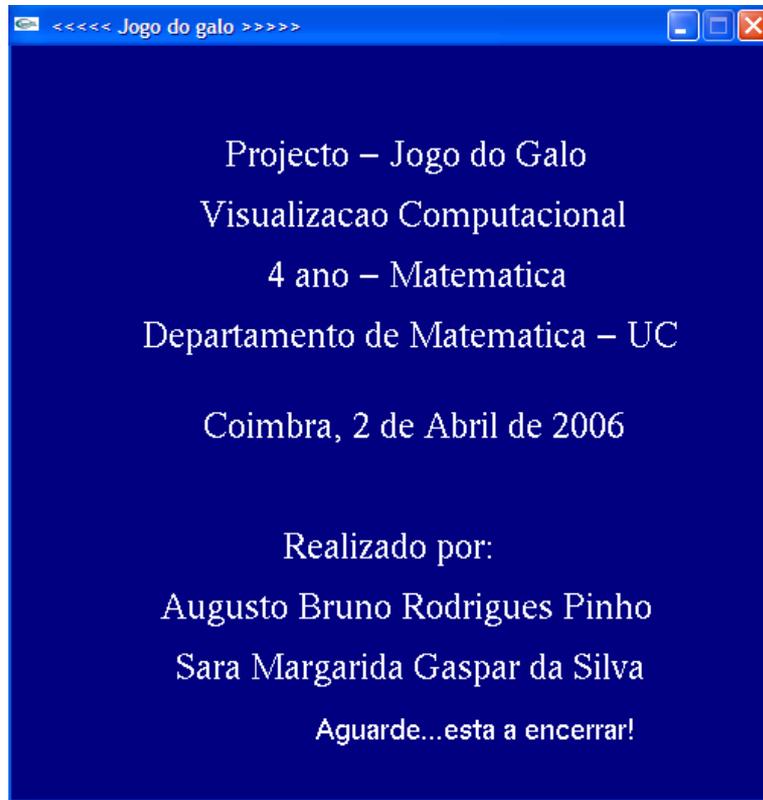
Através deste menu é possível mudar os objectos do jogo, de desenhos para imagens e vice-versa, começar um novo jogo, voltar ao menu principal ou sair do jogo.

Se o utilizador pretender mudar para Desenhos, deve clicar em Desenhos,



para mudar novamente basta clicar em Imagens.

Suponhamos que o utilizador pretende sair do jogo,



As outras duas versões funcionam de modo semelhante.