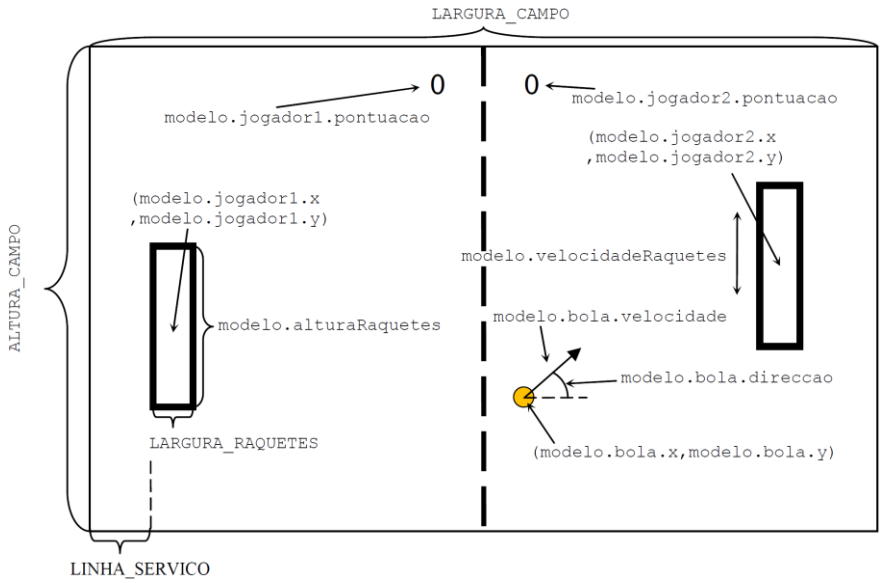
Aula Prática Laboratorial n.º 3

Sumário

Projecto “pingue-pongue”.

Enunciado

1. Crie o projecto de um jogo de pingue-pongue com base no *template* de programa que foi disponibilizado.
2. Analise o referido *template* e observe atentamente o seu funcionamento.
3. Examine as estruturas de dados e as variáveis globais criadas, bem como as funções `inicia_jogo()`, `Init()`, `Draw()` (e as funções nela referenciadas), `Key()`, `KeyUp()`, `cria_menu()`, `menu()` e `MenuStatus()`.
4. Na função `TimerTeclas()` insira o código responsável por animar as raquetes dos jogadores e detectar a colisão das mesmas com os limites inferior e superior do campo de jogo.
5. Na função `Timer()` insira o código responsável por animar a bola, detectar a colisão da mesma com as tabelas e com as raquetes e, no caso de ser marcado um ponto, chamar a função `inicia_jogo()`.



Estruturas de dados e variáveis globais

```
typedef struct {
  GLboolean   q,a,p,l;
}Teclas;

typedef struct {
  GLboolean doubleBuffer;
  GLint     delayMovimento;
  GLint     delayTeclas;
  Teclas    teclas;
  GLuint    menu_vel_bola_id;
  GLuint    menu_tam_bola_id;
  GLuint    menu_vel_raq_id;
  GLuint    menu_tam_raq_id;
  GLuint    menu_id;
  GLboolean menuActivo;
  GLboolean debug;
}Estado;

typedef struct {
  GLfloat   x,y;
  GLint     pontuacao;
}Raquete;

typedef struct {
  GLfloat   x,y;
  GLint     velocidade;
  GLfloat   direccao;
  GLint     tamanho;
}Bola;

typedef struct {
  Raquete   jogador1,jogador2;
  Bola      bola;
  GLint     alturaRaquetes;
  GLint     velocidadeRaquetes;
  GLint     servico;
  GLboolean parado;
}Modelo;

Estado estado;
Modelo modelo;
```

Funções

- `void glutIgnoreKeyRepeat(int ignore);`

  Se `ignore == GL_TRUE`, a repetição automática das teclas não será reportada aos *callbacks* do teclado.

- `void bitmapString(const char *str, double x, double y)`

  Desenha a *string* `str`, nas coordenadas `x`, `y`, usando as funções `glRasterPos2d()` e `glutBitmapCharacter()`.

- `void glLineStipple(2,0xFF00);`

  Define uma escala e um padrão de desenho de linhas descontínuas; no exemplo, a escala `== 2` e o padrão `== 0xFF00` (tracejado grande).

- `void glEnable(GL_LINE_STIPPLE);`
- `void glDisable(GL_LINE_STIPPLE);`

  Liga / desliga o modo de desenho de linhas descontínuas.

Funções para utilização de menus em GLUT

- `int glutCreateMenu(void (*func)(int value));`

  Cria um menu e associa-lhe a função que irá servir de *callback*. Ao regressar, devolve a identificação (um inteiro ID) do menu.

- `void glutMenuStatusFunc(void (*func)(int status, int x, int y));`

  Regista uma função como *callback* da utilização dos menus; a variável `status` assume os valores `GLUT_MENU_IN_USE` ou `GLUT_MENU_NOT_IN_USE`. O *callback* será invocado sempre que se entrar ou sair de um menu.

- `void glutSetMenu(int menu);`

  Activa o menu com a ID `menu`.

- `int glutGetMenu(void);`

  Devolve a ID do menu activo.

- `void glutDestroyMenu(int menu);`

  Destrói o menu com a ID `menu`.

- `void glutAttachMenu(int button);`
- `void glutDetachMenu(int button);`

  Associa / desassocia um botão do rato a / de um menu; o parâmetro *button* pode assumir um dos seguintes valores: `GLUT_LEFT_BUTTON`, `GLUT_MIDDLE_BUTTON` ou `GLUT_RIGHT_BUTTON`.

- `void glutAddMenuEntry(char *name, int value);`

  Adiciona uma opção ao menu activo e define o valor a passar à função de *callback*.

- `void glutAddSubMenu(char *name, int menu);`

  Adiciona o submenu com a ID `menu` ao menu activo.

- `void glutRemoveMenuItem(int entry);`

  Elimina uma opção do menu actual (para a primeira opção, `entry == 1`)

- `void glutChangeToMenuEntry(int entry, char *name, int value);`

  Altera uma opção do menu.

- `void glutChangeToSubMenu(int entry, char *name, int menu);`

  Altera uma opção do menu para um submenu.