MANUAL DE PRÁCTICAS DE LABORATORIO DE LA MATERIA DE SISTEMAS DE INFORMACIÓN INTELIGENTES (STRAWBERRY PROLOG)

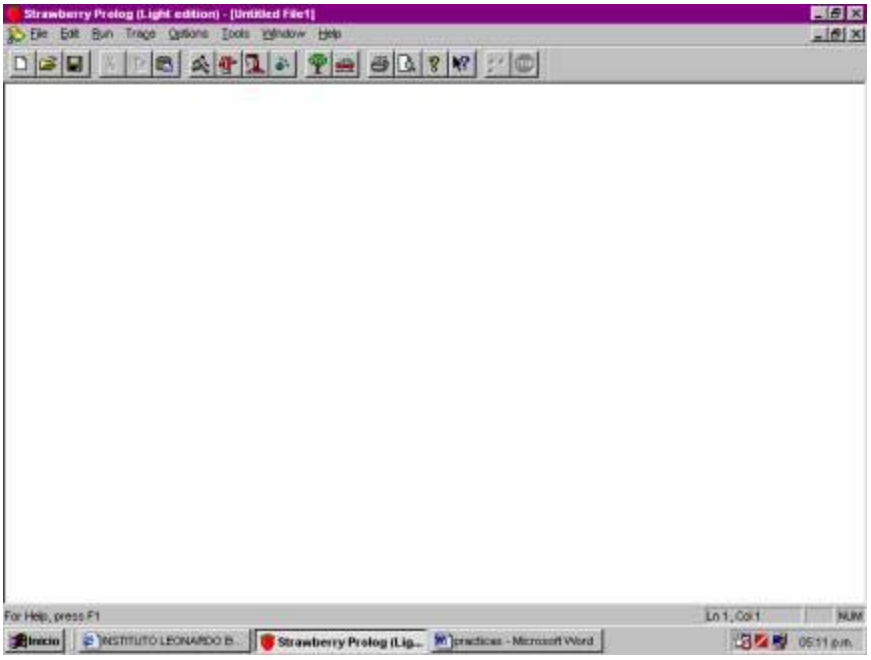
PARA LA LICENCIATURA EN CIENCIAS DE LA INFORMÁTICA.

LIC. EDUARDO BUSTOS FARÍAS
FEBRERO 2002.

INDICE

PRACTICA NO. 0 DE PROLOG

1. Identifique las partes de la pantalla del compilador de Prolog.



a) *Barra de título.*
b) *Barra de menús.*
c) *Barra de herramientas.*
d) *Barra de estado.*
e) *Botón de minimizar.*
f) *Botón de restaurar.*
g) *Botón de cerrar.*
h) *Botón de control.*
i) *Area de trabajo.*

2. Identifique los nombres de los iconos de la barra de herramientas.

PRACTICA NO. 1 DE PROLOG

```
?-
  window(_, _, win_func(_), "Animations", 200, 100, 500, 300).
win_func(init) :-
  G_Br=0,      % G_Br is a global variable (it begin with G_)
  menu( normal, _, _, menu_brush(_), "&Brush"),
  animate( _, _, fail(_), "res/Dillo.avi", 10, 10),
  animate( _, _, fail(_), "res/Search.avi", 10, 150),
  animate( _, _, fail(_), "res/filecopy.avi", 150, 150).

menu_brush(press) :-
  new_brush(G_Br),
  update_window(_),
  G_Br:= (G_Br+1) mod 5.

new_brush(0) :- window_brush(_, rgb(0, 0, 0), x).
new_brush(1) :- window_brush(_, rgb(255, 0, 255), (/)).
new_brush(2) :- window_brush(_, "res/Wall.bmp").
new_brush(3) :- window_brush(_, rgb(127, 0, 127)).
new_brush(4) :- window_brush(_, _).
```

¿Qué hace el programa?

PRACTICA NO. 2 DE PROLOG

```
?-
  window( _, _, win_func(_), "Arc", 100, 0, 600, 600).

win_func(paint) :-
  for(I,3,300,3),
  wait(0.2),
  pen(3, rgb(random(256),random(256),random(256))),
  draw_arc(200-I, 200-I, 200+I, 200+I,0,0,0,0),
  draw_arc(200-I, 400-I, 200+I, 400+I,0,0,0,0),
  draw_arc(400-I, 200-I, 400+I, 200+I,0,0,0,0),
  draw_arc(400-I, 400-I, 400+I, 400+I,0,0,0,0),
  fail.
```

¿Qué hace el programa?

PRACTICA NO. 3 DE PROLOG

```
?-
  window( _, _, win_func(_), "Check Box", 200, 100, 400, 200).

win_func(init) :-
    check_box( _, _, check_func, "Check Box 1", 20, 20, 180, 20),
    check_box( _, _, check_func, "Check Box 2", 20, 40, 180, 20).

check_func(press) :-
  set_check_box_value(_,1-get_check_box_value(_)).
```

¿Qué hace el programa?

PRACTICA NO. 4 DE PROLOG

```
?-
  window( G_W, _, win_func(_), "Child Windows", 200, 100, 400, 200).

win_func(init) :-
  menu( normal, _, _, menu_func(_), "&Close the Active"),
  button( _, _, fail(_), "Will be closed", 20, 20, 180, 20),
  button( _, _, but_func(_), "Close", 20, 80, 80, 20).

but_func(press) :-
  X is first_child(G_W), % try with previous_child(_)
  write(X), nl,
  close_window(X).

menu_func(press) :-
  X is active_child(G_W),
  write(X), nl,
  close_window(X).
```

¿Qué hace el programa?

PRACTICA NO. 5 DE PROLOG

```
?-
  window( G_W, _, dll_func(_), "DLL", 200, 100, 400, 200).

dll_func(init) :-
  button( _, _, but_func(_), "Close", 20, 20, 80, 20).

but_func(press) :-
  User32 is open_DLL("user32"),
  X is get_handle(_),
  User32."SendMessageA"(X,0x10,0,0), % WM_CLOSE=0x10
  write(X),
  close(User32).
```

¿Qué hace el programa?

PRACTICA NO. 6 DE PROLOG

```
?-
  window(G_W, _, win_func(_), "Edit.spj - You can change this string!", 200, 100,
500, 400).
win_func(init) :-
  G_Br=0,      % G_Br, G_W, G_E1 and G_E2 are global variables (they begin with
G_)
  menu( normal, _, _, menu_change(_), "&Change"),
  menu( normal, _, _, menu_brush(_), "&Brush"),
  window_brush(_, rgb(0, 0, 255)),      % blue
  edit( G_E1, _, edit_func(_), "First", 10, 10, 100, 50),
  edit( G_E2, _, edit_func(_), "Second", 150, 150, 300, 40).
edit_func(init) :-
  color_text(_, rgb(255, 0, 0)),         % red
  color_text_back(_, rgb(255, 255, 0)). % yellow
```

```
menu_change(press) :-
  X is get_text(G_W),
  Y is get_text(G_E1),
  Z is get_text(G_E2),
  set_text(Y, G_W),
  set_text(Z, G_E1),
  set_text(X, G_E2).

menu_brush(press) :-
  new_brush(G_Br),
  update_window(_),
  G_Br:= (G_Br+1) mod 10.

new_brush(0) :- window_brush(_, rgb(0, 0, 0), x).
new_brush(1) :- window_brush(_, rgb(0, 0, 255), (+)).
new_brush(2) :- window_brush(_, rgb(0, 255, 0), h).
new_brush(3) :- window_brush(_, rgb(255, 0, 0), v).
new_brush(4) :- window_brush(_, rgb(255, 0, 255), (/)).
new_brush(5) :- window_brush(_, rgb(127, 0, 127), (\)).
new_brush(6) :- window_brush(_, "res\\Wall.bmp").
new_brush(7) :- window_brush(_, "res\\Strawberry.bmp").
new_brush(8) :- window_brush(_, "res\\SP.bmp").
new_brush(9) :- window_brush(_, rgb(127, 0, 127)).
```

¿Qué hace el programa?

PRACTICA NO. 7 DE PROLOG

```
?-
  pen(3, rgb(0,0,0)),
  brush(rgb(0,0,255)),
  window( _, _, win_func(_), "Fill Polygon", 100, 100, 308, 328).

win_func(paint) :-
  fill_polygon(150,30, 200,250, 70,110, 230,110, 100,250).
```

¿Qué hace el programa?

PRACTICA NO. 8 DE PROLOG

```
-
  window(_, _, win_func(_), "Fonts", 200, 100, 400, 300).

win_func(paint) :-
  color_text(_, rgb(255, 0, 0)),   % red
  font(15,50,"Times New Roman"),
  text_out(10, 10, "Strawberry Prolog"),
  color_text(_, rgb(0, 0, 255)),   % blue
  font(20,50,"Courier"),
  text_out(50, 100, "is the most"),
  font(35,50,"Times New Roman"),
  color_text(_, rgb(0, 255, 255)),
  text_out(90, 200, "c"),
  color_text(_,rgb(255, 0, 255)),
  text_out(_,_,"o"),
  color_text(_,rgb(0, 0, 0)),
  text_out(_,_,"l"),
  color_text(_,rgb(255, 255, 0)),
  text_out(_,_,"o"),
  color_text(_,rgb(255, 127, 0)),
  text_out(_,_,"r"),
  color_text(_,rgb(0, 127, 0)),
  text_out(_,_,"f"),
  color_text(_,rgb(255, 0, 0)),
  text_out(_,_,"u"),
  color_text(_,rgb(0, 0, 255)),
  text_out(_,_,"l").
```

¿Qué hace el programa?

PRACTICA NO. 9 DE PROLOG

```
?-
  window( _, _, win_func(_), "Group Box", 200, 100, 400, 200).
```

```
win_func(init) :-
   group_box( _, _, fail, "This is a Group Box", 90, 10, 200, 80),
% You can replace this box
% It is placed only for beauty
   radio_button( G_R1, _, radio_func, "Radio Button 1", 100, 40, 180, 20),
   radio_button( G_R2, _, radio_func, "Radio Button 2", 100, 60, 180, 20).

check_func(press) :-
  set_check_box_value(_,1-get_check_box_value(_)).

radio_func(press) :-
  get_check_box_value(_)=:=0,
  set_check_box_value(G_R1,0),
  set_check_box_value(G_R2,0),
  set_check_box_value(_,1).
```

¿Qué hace el programa?

PRACTICA NO. 10 DE PROLOG

```
?-
  G_F1=0,
  G_F2=0,
  (yes_no("List Demo","Do you want colors",?)->
    G_F1:=1
  ),
  (yes_no("List Demo","Do you want incons",?)->
    G_F2:=1
  ),
  window( _, _, win_func(_), "List Demo", 100, 50, 370, 420).

win_func(init) :-
  button( _, _, but_clean(_), "Clean", 180, 20, 80, 20),
  button( _, _, but_selected(_), "Selected", 180, 50, 80, 20),
  static( G_S1, _, fail(_), "no", 180, 75, 80, 20),
  static( G_S2, _, fail(_), "", 10, 360, 80, 20),
  static( _, _, fail(_), "At:", 180, 110, 40, 25),
  edit( G_At, _, fail(_), "1", 220, 110, 40, 25), % number or end
  static( _, _, fail(_), "Col:", 260, 110, 40, 25),
```

```
  edit( G_Col, _, fail(_), "1", 300, 110, 40, 25),
  static( _, _, fail(_), "Icon:", 180, 140, 40, 25),
  edit( G_Icon, _, fail(_), "0", 220, 140, 40, 25),
  static( _, _, fail(_), "Text:", 180, 170, 40, 25),
  edit( G_Text, _, fail(_), "new", 220, 170, 120, 25),
  button( _, _, but_insert(_), "Insert", 180, 200, 80, 20),
  button( _, _, but_delete(_), "Delete", 270, 200, 80, 20),
  button( _, _, but_column(_), "Column", 180, 225, 80, 20),
  button( _, _, but_label(_), "Set Label", 270, 225, 80, 20),
  check_box( CB1, _, check_func1, "single selection", 170, 255, 160, 20),
  set_check_box_value( CB1,1),
  check_box( _, _, check_func2, "in place edit", 170, 275, 160, 20),
  check_box( _, _, check_func3, "column header", 170, 295, 160, 20),
  group_box( _, _, fail, "type", 165, 315, 190, 65),
  radio_button( G_R1, _, radio_func, "Icon", 170, 335, 85, 20),
  radio_button( G_R2, _, radio_func, "Report", 170, 355, 85, 20),
  radio_button( G_R3, _, radio_func, "Small", 255, 335, 85, 20),
  radio_button( G_R4, _, radio_func, "List", 255, 355, 85, 20),
  set_check_box_value(G_R2,1),
  list_box( G_List, _, list_func, "Name", 10, 10, 150, 350),
  (G_F1=1->
    color_text(G_List,rgb(255,0,0)),
    color_text_back(G_List,rgb(225,255,225))
  ),
  (G_F2=1->
    set_list_icons(G_List, s, "res/method16.ico", "res/property16.ico",
"res/object16.ico"),
    set_list_icons(G_List, b, "res/method32.ico", "res/property32.ico",
"res/object32.ico")
  ),
  insert_list_item(G_List, end, "second", 1),
  insert_list_item(G_List, end, "third", 2),
  insert_list_item(G_List, 0, "first", 0),
  insert_list_item(G_List, end, "last (no icon)", 4),
  how_much.

list_func(edit(Item,Text)):-
  yes_no("List Demo","Are you sure that you want to change this text.",?),
  set_list_label(G_List, Item, 0, Text).

how_much:-
  set_text(get_list_length(G_List)+" elements",G_S2).

but_clean(press):-
  clean_the_list(G_List),
```

```
  how_much.

but_selected(press):-
  set_text("no",G_S1),
  Sel is get_list_selected(G_List),
  set_text("first is "+Sel,G_S1),
  set_text(print(Sel),G_At),
  set_text(get_list_text(G_List, Sel),G_Text).

but_delete(press):-
  delete_list_item(G_List, scan(get_text(G_At))),
  how_much.

but_insert(press):-
  insert_list_item(G_List, scan(get_text(G_At)), get_text(G_Text),
scan(get_text(G_Icon))),
  how_much.

but_column(press):-
  add_list_column(G_List, scan(get_text(G_Col)), get_text(G_Text), 30).

but_label(press):-
  set_list_label(G_List, scan(get_text(G_At)), scan(get_text(G_Col)),
get_text(G_Text)).

check_func1(press) :-
  change_style(G_List, 4, 0, 0),
  set_check_box_value(_,1-get_check_box_value(_)).
check_func2(press) :-
  change_style(G_List, 0x200, 0, 0),
  set_check_box_value(_,1-get_check_box_value(_)).
check_func3(press) :-
  change_style(G_List, 0x4000, 0, 0),
  set_check_box_value(_,1-get_check_box_value(_)).

radio_func(press):-
  set_check_box_value(G_R1,0),
  set_check_box_value(G_R2,0),
  set_check_box_value(G_R3,0),
  set_check_box_value(G_R4,0),
  set_check_box_value(_,1),
  Label is get_text(_),
  label(Label,N),
  change_style(G_List, 0, 3, N).
```

```
label("Icon",0).
label("Report",1).
label("Small",2).
label("List",3).
```

¿Qué hace el programa?

PRACTICA NO. 11 DE PROLOG

```
?-
  window(_, _, win_func(_), "Window", 200, 100, 400, 200).

win_func(init) :-
  menu(normal, _, _, menu_window(_), "&New window"),
  menu(normal, _, _, menu_go(_), "&Go-go"),
  menu(normal, _, _, menu_name(_), "&Change the name"),
  menu(normal, _, _, menu_menu(_), "&Extend the menu").
menu_window(press) :-
  position(_,X, Y),
  window( _, 0, win_func(_), "Window", X+20, Y+20, 400, 200).
menu_go(press) :-
  size(_,X, Y),
  size(_,X+1, Y),
  X<600,
  menu_go(press).
menu_name(press) :-
  read(X, "Give a new name to the window", s),
  set_text(X, _).
menu_menu(press) :-
  read(X, "Give a name for a new command", s),
  menu( normal, _, _, menu_go(_), X).
```

¿Qué hace el programa?

PRACTICA NO. 12 DE PROLOG

```
?-
  window( _, _, win_func(_), "Radio Buttons", 200, 100, 400, 200).

win_func(init) :-
    radio_button( _, _, check_func, "Check Button 1", 20, 20, 180, 20),
    radio_button( _, _, check_func, "Check Button 2", 20, 40, 180, 20),
    radio_button( G_R1, _, radio_func, "Radio Button 1", 200, 20, 180, 20),
    radio_button( G_R2, _, radio_func, "Radio Button 2", 200, 40, 180, 20).

check_func(press) :-
  set_check_box_value(_,1-get_check_box_value(_)).

radio_func(press) :-
  get_check_box_value(_)=:=0,
  set_check_box_value(G_R1,0),
  set_check_box_value(G_R2,0),
  set_check_box_value(_,1).
```

¿Qué hace el programa?

PRACTICA NO. 13 DE PROLOG

```
?-
  G_Color:=rgb(255,0,0), % red
  G_FaceName="Font",
  G_Height=10,
  window(_, _, win_func(_), "Select Font", 200, 100, 400, 200).

win_func(init) :-
  menu( normal, _, _, menu_ChangeFont(_), "Change &Font").

win_func(paint) :-
  color_text(_, G_Color),
  text_out(10, 10, "Strawberry Prolog"),
  text_out(10, 40, G_FaceName+" "+G_Height).

menu_ChangeFont(press):-
```

```
  select_font(_, G_Font, G_Color, G_FaceName, G_Height),
  update_window(_).
```

¿Qué hace el programa?

PRACTICA NO. 14 DE PROLOG

```
?-
  G_X1:=0,
  G_X2:=0,
  window( G_W1, _, speed_func1(_), "Window 1", 200, 100, 400, 200),
  window( G_W2, _, speed_func2(_), "Window 2", 200, 300, 400, 200).

speed_func1(init) :-
  button( _, _, but_func1(_), "Draw", 20, 20, 80, 20).
speed_func2(init) :-
  button( _, _, but_func2(_), "Draw", 20, 20, 80, 20).

but_func1(press) :-
  set_active_window(G_W2),
  brush(rgb(random(256),random(256),random(256))),
  ellipse(G_X1+100,100,G_X1+200,200),
  G_X1:=G_X1+20.

but_func2(press) :-
  set_active_window(G_W1),
  brush(rgb(random(256),random(256),random(256))),
  ellipse(G_X2+100,100,G_X2+200,200),
  G_X2:=G_X2+20.
```

¿Qué hace el programa?

PRACTICA NO. 15 DE PROLOG

```
?-
  window( _, _, speed_func(_), "Speed Test", 200, 100, 400, 200).

speed_func(init) :-
    button( _, _, but_func(_), "Start", 20, 20, 80, 20).

but_func(press) :-
  cursor(w),
  chronometer(_),
  wait,
  chronometer(T),
  beep,
  Str is "1,400,000 steps was done for " + (T/1000)
   + " seconds, i.e. " + (1398102000//T)
   + " steps per second.",
  message("End of Test", Str, i).

wait :-
  p(_),p(_),p(_),p(_),p(_),
  p(_),p(_),p(_),p(_),p(_),
  fail.
wait.

p(a).
p(b).
p(c).
p(d).
```

¿Qué hace el programa?

PRACTICA NO. 16 DE PROLOG

```
% The goal creates a window and waits to get and process messages.
?-
   pen(3, rgb(255, 0, 0)),   % red
   brush(rgb(255, 255, 0)),  % yellow
   window( _, _, sun_func(_), "Sun shine", 200, 100, 400, 440).
```

```
% This predicate starts working at messages
sun_func(close) :-
   not( yes_no("","Did you finish your sun bath?", !)).

% Message 'paint' is sent when the window has to be paint.
sun_func(paint) :-
   ellipse(30, 30, 370, 370),
   line(200,150,200,250),
   line(120,100,160,120),
   line(240,120,280,100),
   ellipse(150, 300, 250, 320),
   beep,
   fail.
```

¿Qué hace el programa?

PRACTICA NO. 17 DE PROLOG

```
?-
  window( _, _, win_func(_), "Timer", 200, 100, 400, 200).

win_func(init) :-
    G_Timer is set_timer(_, 0.5, time_func),
    button( _, _, but_func(_), "Stop", 20, 20, 80, 20).

win_func(close) :-
  kill_timer(_, G_Timer).
% fail.

but_func(press) :-
  kill_timer(parent(_), G_Timer).

time_func(end) :-
  write( (*) ), % * is in brackets because it is an operator
  beep.
```

¿Qué hace el programa?

PRACTICA NO. 18 DE PROLOG

```
?-
  G_F=0,
  (yes_no("Tree Demo","Do you want incons",?)->
    G_F:=1
  ),
  window( _, _, win_func(_), "Tree Demo", 100, 50, 370, 420).

win_func(init) :-
  button( _, _, but_selected(_), "Selected", 180, 15, 80, 20),
  button( _, _, but_clean(_), "Clean", 270, 15, 80, 20),
  static( G_S1, _, fail(_), "no", 160, 40, 120, 20),
  static( G_S2, _, fail(_), "", 10, 360, 80, 20),
  static( _, _, fail(_), "Parent:", 160, 65, 60, 25),
  edit( G_Parent, _, fail(_), "", 220, 65, 120, 25),
  static( _, _, fail(_), "Child:", 160, 95, 60, 25),
  edit( G_Child, _, fail(_), "", 220, 95, 120, 25),
  static( _, _, fail(_), "Brother:", 160, 125, 60, 25),
  edit( G_Brother, _, fail(_), "", 220, 125, 120, 25),
  static( _, _, fail(_), "Icon:", 180, 155, 40, 25),
  edit( G_Icon, _, fail(_), "0", 220, 155, 30, 25),
  static( _, _, fail(_), "Icon Sel:", 250, 155, 60, 25),
  edit( G_Icon2, _, fail(_), "1", 310, 155, 30, 25),
  static( _, _, fail(_), "Text:", 180, 185, 40, 25),
  edit( G_Text, _, fail(_), "new", 220, 185, 120, 25),
  button( _, _, but_insert(_), "Insert", 180, 215, 80, 20),
  button( _, _, but_delete(_), "Delete", 270, 215, 80, 20),
  group_box( _, _, fail, "where", 165, 235, 190, 40),
  radio_button( G_R1, _, radio_func, "root", 170, 255, 85, 17),
  radio_button( G_R2, _, radio_func, "first", 255, 255, 85, 17),
  set_check_box_value(G_R1,1),
  set_check_box_value(G_R2,1),
  check_box( Ch1, _, check_func1, "buttons", 170, 280, 120, 20),
  check_box( Ch2, _, check_func2, "lines", 170, 300, 145, 20),
  check_box( Ch3, _, check_func3, "root lines", 170, 320, 120, 20),
  check_box( _, _, check_func4, "in place edit", 170, 340, 120, 20),
  set_check_box_value(Ch1,1),
  set_check_box_value(Ch2,1),
  set_check_box_value(Ch3,1),
  tree_box( G_Tree, _, tree_func, _, 10, 10, 150, 350),
```

```
  (G_F=1->
    set_tree_icons(G_Tree, "res/method16.ico", "res/property16.ico",
"res/object16.ico")
  ),
  insert_tree_item(G_Tree, Billy, root, first, "Billy", 0, 1),
  insert_tree_item(G_Tree, Daughter, Billy, first, "daughter", 0, 1),
  insert_tree_item(G_Tree, Son, Billy, first, "son", 0, 1),
  insert_tree_item(G_Tree, Tim, root, first, "Tim", 2, 2),
  insert_tree_item(G_Tree, Merry, root, Tim, "Merry", 0, 1),
  insert_tree_item(G_Tree, _, Merry, first, "son", 2, 2),
  insert_tree_item(G_Tree, _, root, sorted, "Ann", 2, 2),
  insert_tree_item(G_Tree, _, Son, last, "grandson", 2, 2),
  insert_tree_item(G_Tree, _, Son, sorted, "granddaughter", 2, 2),
  insert_tree_item(G_Tree, _, Daughter, first, "granddaughter", 2, 2),
  how_much.

tree_func(edit(Item,Text)):-
  yes_no("Tree Demo","Are you sure that you want to change this text.",?),
  set_tree_item(G_Tree, Item, Text, _, _).

how_much:-
  set_text(get_tree_length(G_Tree)+" elements",G_S2).

but_clean(press):-
  clean_the_tree(G_Tree),
  how_much.

but_selected(press):-
  set_text("no",G_S1),
  Sel is get_tree_selected(G_Tree),
  set_text("h: "+Sel,G_S1),
  Parent is tree_parent(G_Tree, Sel),
  (Parent=root->
    set_text("root",G_Parent)
  else
    get_tree_item(G_Tree, Parent, TextParent,_,_),
    set_text(print(TextParent),G_Parent)
  ),
  (Child is tree_child(G_Tree, Sel)->
    get_tree_item(G_Tree, Child, TextChild,_,_),
    set_text(print(TextChild),G_Child)
  else
    set_text("no",G_Child)
  ),
  (Brother is tree_next_brother(G_Tree, Sel)->
```

```
    get_tree_item(G_Tree, Brother, TextBrother,_,_),
    set_text(print(TextBrother),G_Brother)
  else
    set_text("no",G_Brother)
  ),
  get_tree_item(G_Tree, Sel, Text,Icon1,Icon2),
  set_text(print(Icon1),G_Icon),
  set_text(print(Icon2),G_Icon2),
  set_text(Text,G_Text).

but_delete(press):-
  delete_tree_item(G_Tree, get_tree_selected(G_Tree)),
  how_much.

but_insert(press):-
  (get_check_box_value(G_R1)=:=1->
    Parent=root,
    (get_check_box_value(G_R2)=:=1->
      AfterChild=first
    else
      AfterChild is get_tree_selected(G_Tree)
    )
  else
    (get_check_box_value(G_R2)=:=1->
      Parent is get_tree_selected(G_Tree),
      AfterChild=first
    else
      AfterChild is get_tree_selected(G_Tree),
      Parent is tree_parent(G_Tree, AfterChild)
    )
  ),
  insert_tree_item(G_Tree, _, Parent, AfterChild, get_text(G_Text),
scan(get_text(G_Icon)), scan(get_text(G_Icon2))),
  how_much.

radio_func(press) :-
  set_check_box_value(_,1-get_check_box_value(_)).
check_func1(press) :-
  change_style(G_Tree, 1, 0, 0),
  set_check_box_value(_,1-get_check_box_value(_)).
check_func2(press) :-
  change_style(G_Tree, 2, 0, 0),
  set_check_box_value(_,1-get_check_box_value(_)).
check_func3(press) :-
  change_style(G_Tree, 4, 0, 0),
```

```
  set_check_box_value(_,1-get_check_box_value(_)).
check_func4(press) :-
  change_style(G_Tree, 8, 0, 0),
  set_check_box_value(_,1-get_check_box_value(_)).
```

¿Qué hace el programa?

PRACTICA NO. 19 DE PROLOG

```
?-
   window( _, _, win_func(_), "Bitmaps and Icons", 100, 100, 375, 400).

win_func(init):-
         bitmap(_, _, fail(_), default, 50, 50),
     bitmap(_, _, fail(_), default, 250, 50),
     bitmap(_, _, fail(_), default, 100, 100),
     bitmap(_, _, fail(_), default, 200, 100),
     bitmap(_, _, fail(_), default, 50, 150),
     bitmap(_, _, fail(_), default, 150, 150),
     bitmap(_, _, fail(_), default, 250, 150),
     bitmap(_, _, fail(_), default, 100, 200),
     bitmap(_, _, fail(_), default, 200, 200),
     bitmap(_, _, fail(_), default, 150, 250),
     icon(_, _, fail(_), default, 165, 215),
     icon(_, _, fail(_), default, 165, 115),
     icon(_, _, fail(_), default, 110,70),
     icon(_, _, fail(_), default, 220, 70),
     icon(_, _, fail(_), default, 30,112),
     icon(_, _, fail(_), default, 300,110),
     icon(_, _, fail(_), default, 112,170),
     icon(_, _, fail(_), default, 70,115),
     icon(_, _, fail(_), default, 265,115),
     icon(_, _, fail(_), default, 210,170),
     icon(_, _, fail(_), default, 23,76),
     icon(_, _, fail(_), default, 305,72),
     G_Br=0,      % G_Br is a global variable (it begin with G_)
     menu( normal, _, _, menu_brush(_), "&Brush").

menu_brush(press) :-
     new_brush(G_Br),
```

```
    update_window(_),
    G_Br:= (G_Br+1) mod 5.

new_brush(0) :- window_brush(_, rgb(0, 0, 0), x).
new_brush(1) :- window_brush(_, rgb(255, 0, 255), (/)).
new_brush(2) :- window_brush(_, "res/Wall.bmp").
new_brush(3) :- window_brush(_, rgb(127, 0, 127)).
new_brush(4) :- window_brush(_, _).
```

¿Qué hace el programa?

PRACTICA NO. 20 DE PROLOG

```
?-
  window(_, _, win_func(_), "Cursor", 200, 100, 400, 300),
  cursor(?), wait(0.5),
  cursor(s), wait(0.5),
  cursor(c), wait(0.5),
  cursor(t), wait(0.5),
  cursor(o), wait(0.5),
  cursor(x), wait(0.5),
  cursor(n), wait(0.5),
  cursor(d), wait(0.5),
  cursor(u), wait(0.5),
  cursor(!), wait(0.5),
  cursor(v), wait(0.5),
  cursor(w), wait(0.5),
  text_out(10, 50, " End of the Test! ").

win_func(init) :-
  window_brush(_, rgb(255, 255, 0)),   % yellow
  color_text(_, rgb(255, 255, 255)),   % white
  color_text_back(_, rgb(0, 0, 255)).  % blue

win_func(paint) :-
  text_out(10, 10, " Look at the cursor! ").
```

¿Qué hace el programa?

PRACTICA NO. 21 DE PROLOG

```
?-
  window( _, _, win_func(_), "Line", 100, 0, 600, 600).

win_func(paint) :-
  for(I,3,300,3),
  wait(0.2),
  pen(3, rgb(random(256),random(256),random(256))),
  line(200-I, 200-I, 200-I, 200+I, 200+I, 200+I, 200+I, 200-I, 200-I, 200-I),
  line(200-I, 400-I, 200-I, 400+I, 200+I, 400+I, 200+I, 400-I, 200-I, 400-I),
  line(400-I, 200-I, 400-I, 200+I, 400+I, 200+I, 400+I, 200-I, 400-I, 200-I),
  line(400-I, 400-I, 400-I, 400+I, 400+I, 400+I, 400+I, 400-I, 400-I, 400-I),
  fail.
```

¿Qué hace el programa?

PRACTICA NO. 22 DE PROLOG

```
?-
  pen(3,rgb(255,0,0)),
  window_n( _, _, win_func, "Window", 200, 100, 400, 440,
  !, "C:\\WINDOWS\\CURSORS\\Globe.ani", rgb(0,0,0)),
  0 =:= random(2),
  X = [20,120,20+100, 20],
  line(|X).

win_func(init) :- beep.
win_func(paint) :-
  X is 20, Y is 20, W is 100, H is 100,
  line(X, Y, X+W, Y, X+W, Y+H, X, Y+H, X, Y),
  line(20,20, 120, 120).
```

¿Qué hace el programa?