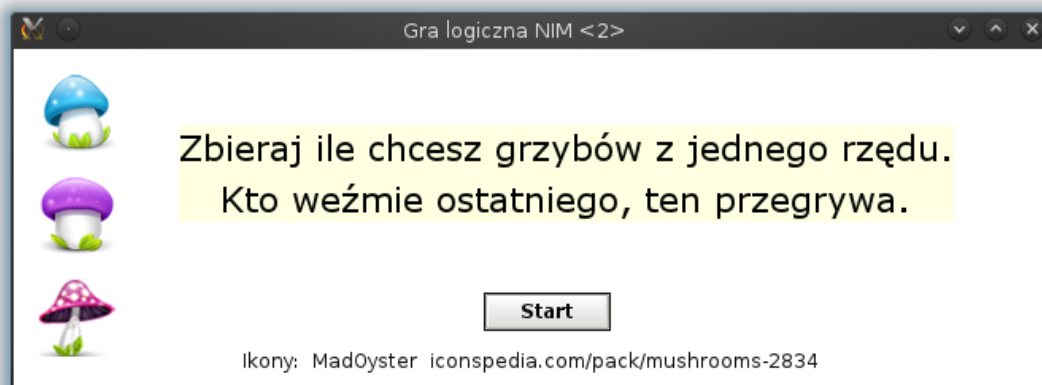


mgr inż. Stanisław Ubermanowicz

Piotr Fiorek

Gra logiczna NIM a001

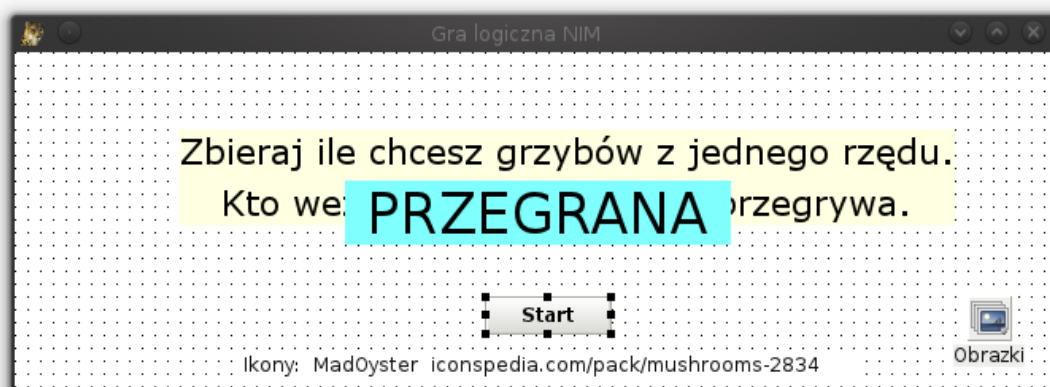
Zaprojektuj grę logiczną NIM. Program losuje w każdym rzędzie od 1 do 10 kulek. Gracz rywalizuje z komputerem. Podczas ruchu można brać dowolną liczbę kulek, ale tylko z jednego rzędu. Przegrywa ten, kto musi zabrać ostatnią kulkę. Strategia wygranej polega na tym, aby utrzymywać parzystość grup binarnych.



Początek gry



Gra po wystartowaniu

Okno Formularza:

Na formularzu należy umieścić tekst (TLabel) który będzie pojawiał po uprzednim odpowiednim ustawieniu w przypadku wygranej lub przegranej gracza. Należy również dodać obiekt kontenera obrazków (TImageList) oraz obiekt guzika (Tbutton).

Ponadto można dodać elementy tekstowe z instrukcją dla gracza.

W oknie „Object Inspector” można również ustawić nazwy poszczególnych elementów. W tej implementacji używamy nazw:

- ☐ „Link” dla obiektu typu TLabel prezentującego adres internetowy
- ☐ „Opis” dla obiektu typu TLabel przedstawiającego opis gry
- ☐ „Info” dla obiektu typu TLabel prezentującego informację o wygranej lub przegranej
- ☐ „Obrazki” dla obiektu typu TImageList przechowującego obrazki kolejnych grzybków
- ☐ „Start” dla obiektu typu TButton będącego guzikiem rozpoczynającym grę

Kod:

```

1 unit gra;
2 {$mode objfpc}{$H+}
3 interface
4 uses
5   Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs,
   StdCtrls,
6   ExtCtrls;
7 type
8   { TOkno}
9   TOkno = class (TForm)
10     Link: TLabel;                                {Zmienne reprezentujące obiekty na
formularzu}
11     Start: TButton;
12     Opis: TLabel;
```

```

Obrazki: TImageList;
14   Info: TLabel;
15   procedure StartClick(Sender: TObject);           {Procedury obsługujące grę}
16   procedure FormCreate(Sender: TObject);
17   procedure Rzad1Click(Sender: TObject);
18   procedure Rzad2Click(Sender: TObject);
19   procedure Rzad3Click(Sender: TObject);
20   procedure Rzad1Action(kulka: Integer);
21   procedure Rzad2Action(kulka: Integer);
22   procedure Rzad3Action(kulka: Integer);
23   procedure SprawdzKoniec();
24   procedure PCMove();
25   private
26     {private declarations}
27     ile1, ile2, ile3: Integer; {Zmienne wskazują, ile grzybów jest w danej
chwili w rzędzie}
28     tura: Integer;             {Zmienna, która określa liczbę tur i to, czyj był
ruch}
29     koniec: Boolean;           {Zmienna określająca czy nastąpił koniec gry}
30     Grzyb1: array[1..10] of TImage; {Lista obiektów pierwszego rzędu grzybów}
31     Grzyb2: array[1..10] of TImage; {Lista obiektów drugiego rzędu grzybów}
32     Grzyb3: array[1..10] of TImage; {lista obiektów trzeciego rzędu grzybów}
33   public
34     {public declarations}
35   end;
36 var
37   Okno: TOkno;
38   i: Integer;           {Zmienna pomocnicza, iteracyjna}
39 implementation
40 {TOkno}                 {Procedura wywoływana tylko raz podczas tworzenia okna}
41 procedure TOkno.FormCreate(Sender: TObject);
42 begin
43   for i:= 1 to 10 do {Twórz wszystkie elementy tablicy}
44   begin
45     Grzyb1[i]:= TImage.Create(self);           {stwórz element}
46     Grzyb1[i].Parent:= self;                   {przypisz go do okna}
47     Grzyb1[i].Visible:= False;                 {ukryj go}
48     Obrazki.GetBitmap(0, Grzyb1[i].Picture.Bitmap); {przypisz mu obrazek}
49     Grzyb1[i].Top:= 16;                         {określ jego położenie od górnej krawędzi
ekranu}
50     Grzyb1[i].Left:= (64*i)-48;                 {określ jego położenie od lewej krawędzi
ekranu}
51     Grzyb1[i].Tag:= i;                         {przypisz mu numer porządkowy}
52     Grzyb1[i].OnClick:= @Rzad1Click; {określ procedurę obsługującą kliknięcie
na obiekt}
53   end;
54   for i:= 1 to 10 do                           {drugi rząd - analogicznie jak powyżej}
55   begin
56     Grzyb2[i]:= TImage.Create(self);
57     Grzyb2[i].Parent:= self;

```

```

Grzyb2[i].Visible:= False;
59   Obrazki.GetBitmap(1, Grzyb2[i].Picture.Bitmap);
60   Grzyb2[i].Top:= 80;
61   Grzyb2[i].Left:= (64*i)-48;
62   Grzyb2[i].Tag:= i;
63   Grzyb2[i].OnClick:= @Rzad2Click;
64   end;
65   for i:= 1 to 10 do                                {trzeci rząd - analogicznie tak jak
powyżej}
66   begin
67     Grzyb3[i]:= TImage.Create(self);
68     Grzyb3[i].Parent:= self;
69     Grzyb3[i].Visible:= False;
70     Obrazki.GetBitmap(2, Grzyb3[i].Picture.Bitmap);
71     Grzyb3[i].Top:= 144;
72     Grzyb3[i].Left:= (64*i)-48;
73     Grzyb3[i].Tag:= i;
74     Grzyb3[i].OnClick:= @Rzad3Click;
75   end;
76   Grzyb1[1].Visible:=True;                            {Pokazanie pierwszych obiektów z każdego
rzędu}
77   Grzyb2[1].Visible:=True;
78   Grzyb3[1].Visible:=True;
79 end;
80 procedure TOkno.StartClick(Sender: TObject); {funkcja obsługująca kliknięcie
guzika START}
81 begin
82   Start.Visible:= False; {ukryj przycisk}
83   Link.Visible:= False; {ukryj adres autora ikon}
84   Opis.Visible:= False; {ukryj opis gry}
85   Info.Visible:= False; {ukryj Info o wygranej lub przegranej}
86   koniec:= False; {ustaw zmienną oznaczającą koniec gry na Fałsz}
87   tura:= 1; {ustaw licznik tur na zero}
88   Randomize; {ustaw pozycję startową generatora liczb losowych}
89   ile1:= Random(10)+1; {wylosuj liczbę kulek w pierwszym rzędzie}
90   for i:= 1 to ile1 do {pętla przebiega przez elementy od pierwszego do
wylosowanego}
91     Grzyb1[i].Visible:= True; {i ustawia, aby elementy były widoczne}
92     ile2:= Random(10)+1; {to samo dla drugiego rzędu}
93     for i:= 1 to ile2 do
94       Grzyb2[i].Visible:= True;
95     ile3:= Random(10)+1; {to samo dla drugiego rzędu}
96     for i:= 1 to ile3 do
97       Grzyb3[i].Visible:= True;
98   end;
99   procedure TOkno.Rzad1Click(Sender: TObject);
100  begin                                     {procedura obsługująca kliknięcie w element z pierwszego
rzędu}
101    {Sender jest ogólną reprezentacją obiektu, który został kliknięty.
Przekształcamy go na konkretny obiekt w tym przypadku TImage i pobieramy z niego
wartość atrybutu Tag, którą od razu przekazujemy do procedury Rzad1Action}

```

```

zad1Action((Sender as TImage).Tag);
103 {sprawdź czy zmienna 'koniec' została ustawiona na Fałsz, jeśli tak, to
zrealizuj ruch komputera}
104 if koniec = False then PCMove();
105 end;
106 procedure TOkno.Rzad2Click(Sender: TObject); {jak wyżej dla drugiego rzędu}
107 begin
108   Rzad2Action((Sender as TImage).Tag);
109   if koniec = False then PCMove();
110 end;
111 procedure TOkno.Rzad3Click(Sender: TObject); {jak wyżej dla trzeciego rzędu}
112 begin
113   Rzad3Action((Sender as TImage).Tag);
114   if koniec = False then PCMove();
115 end;
116 procedure TOkno.Rzad1Action(kulka: Integer); {procedura obsługująca zmiany
stanów rzędu 1}
117 begin
118   tura:= tura+1; {zwiększ licznik tur}
119   for i:= kulka to ile1 do {wykonaj ruch od grzyba klikniętego do
ostatniego}
120     Grzyb1[i].Visible:= False; {ukryj grzyba}
121     ile1:= kulka-1; {ustaw nową wartość pozostałych grzybów}
122     SprawdźKoniec; {sprawdź czy nastąpił koniec gry}
123 end;
124 procedure TOkno.Rzad2Action(kulka: Integer); {to samo dla rzędu drugiego}
125 begin
126   tura:= tura+1;
127   for i:= kulka to ile2 do
128     Grzyb2[i].Visible:= False;
129     ile2:= kulka-1;
130     SprawdźKoniec;
131 end;
132 procedure TOkno.Rzad3Action(kulka: Integer); {to samo dla rzędu trzeciego}
133 begin
134   tura:= tura+1;
135   for i:= kulka to ile3 do
136     Grzyb3[i].Visible:= False;
137     ile3:= kulka-1;
138     SprawdźKoniec;
139 end;
140 procedure TOkno.SprawdźKoniec(); {sprawdzaj, czy wszystkie grzyby zostały
"zabrane"}
141 begin
142   if ile1+ile2+ile3 < 2 then {Sprawdź, czy grzybów jest mniej niż dwa}
143   begin
144     if (tura mod 2) = 1 then {Jeśli była tura gracza}
145       Info.Caption:= 'Przegrana' {wpisz informację o przegranej}
146     else if ile1+ile2+ile3 = 1 then {Jeśli pozostał ostatni grzyb}
147       Info.Caption:= 'Wygrana' {wpisz informację o wygranej}

```

```

else                                     {a jeśli był ruch komputera}
149   Info.Caption:= 'Przegrana';         {wpisz informację o przegranej}
150   koniec:= True;                     {Ustaw zmienną oznaczającą koniec gry}
151   Info.Visible:= True;               {Pokaż napis z informacją o wyniku gry}
152   Start.Visible:= True;              {Pokaż przycisk START, dla uruchomienia
nowej gry}
153   end;                               {po wykonanym ruchu wyjdź z funkcji}
154 end;
155 procedure TOkno.PCMove(); {Procedura realizująca strategię komputera - sztuczny
intelekt}
156 begin
157   if (ile1=ile2) and (ile1<2) and (ile3>1) then {gdy w rzędach 1 i 2 jest 0 lub
1 grzyb,}
158   begin                                     {to pozostaw jednego grzyba w
rzędzie 3.}
159     Rzad3Action(2);                         {kliknij grzyba nr 2 w rzędzie
3.}
160     Exit;
161   end
162   else if (ile1=ile3) and (ile1<2) and (ile2>1) then {gdy w rzędach 1 i 3 są <2
grzyby,}
163   begin                                     {to pozostaw 1 grzyba w
rzędzie 2.}
164     Rzad2Action(2);                         {kliknij grzyba nr 2 w
rzędzie 2.}
165     Exit;
166   end
167   else if (ile2=ile3) and (ile2<2) and (ile1>1) then {gdy w rzędach 2 i 3 są <2
grzyby,}
168   begin                                     {to pozostaw 1 grzyba w
rzędzie 1.}
169     Rzad1Action(2);                         {kliknij grzyba nr 2 w
rzędzie 1.}
170     Exit();
171   end;
172   If (ile1+ile2)=1 then                     {gdy w rzędach 1 i 2 pozostał łącznie jeden
grzyb}
173   begin
174     Rzad3Action(1);                         {usuń wszystkie z rzędu 3}
175     Exit();
176   end
177   else if (ile1+ile3)=1 then                 {analogicznie jak powyżej}
178   begin
179     Rzad2Action(1);
180     Exit();
181   end
182   else if (ile2+ile3)=1 then                 {analogicznie jak powyżej}
183   begin
184     Rzad1Action(1);
185     Exit();
186   end;

```

{Sprawdzenie parzystości binarnej w środkowej fazie gry. Szukanie układu dającego parzystość grup. Jeśli dla danej wartości 'i' poniższe wyrażenia logiczne z operatorem XOR się zerują, to jest to poszukiwany układ parzystości, dający szansę na wygraną.}

```

187 for i:= 1 to 10 do
188   begin
189     if (ile1>=i) and (((ile1-i) xor ile2 xor ile3) = 0) then
190       begin
191         Rza1Action(ile1 - i + 1);           {doprowadź układ do parzystości}
192         Exit();                             {po wykonanym ruchu wyjdź z funkcji}
193       end
194     else if (ile2>=i) and ((ile1 xor (ile2-i) xor ile3) = 0) then
195       begin
196         Rza2Action(ile2 - i + 1);           {doprowadź układ do parzystości}
197         Exit();
198       end
199     else if (ile3>=i) and ((ile1 xor ile2 xor (ile3-i)) = 0) then
200       begin
201         Rza3Action(ile3 - i + 1);           {doprowadź układ do parzystości}
202         Exit();
203       end;
204     end;
    {Losowy wybór posunięcia, jeśli strategię wykorzystał gracz i nie ma ruchu
    wygrywającego}
205     if (ile1>=ile2) and (ile1>=ile3) and ((Random(10)/10)>0.5) then
206       Rza1Action(Random(ile1) + 1)
207     else if (ile2>=ile1) and (ile2>=ile3) then
208       Rza2Action(Random(ile2) + 1)
209     else
210       Rza3Action(ile3);
211   end;
212 initialization
213   {$I gra.lrs}
214 end.

```

UWAGA: Podczas zapisywania projektu nie wolno zapisywać plików ".pas" (plik z kodem) oraz ".lpi" (plik projektu) pod tymi samymi nazwami.