



Nazwa implementacji:

Wieże Hanoi

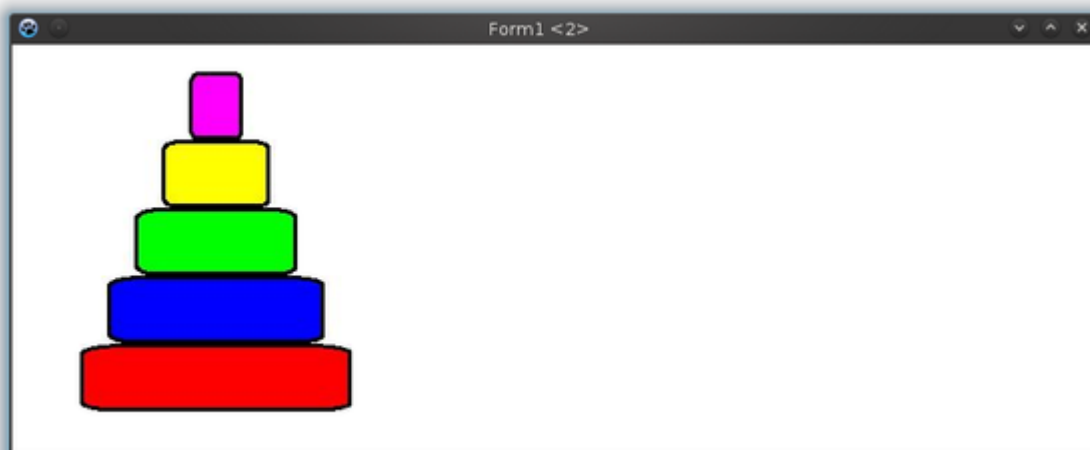
Autor:

Stanisław Ubermanowicz Piotr Fiorek

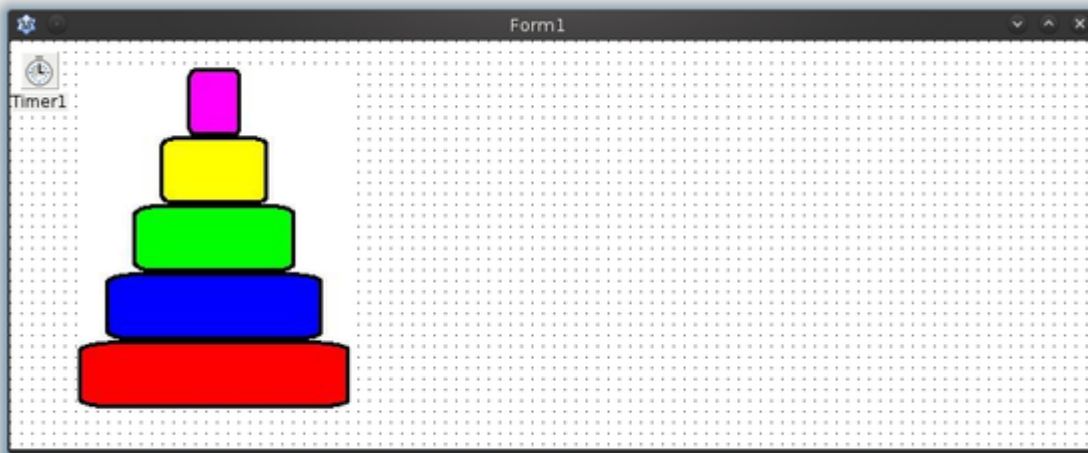
Opis implementacji:

Wizualizacja strategii wygranej w grze decyzyjnej, polegającej na przenoszeniu obiektów o różnej wielkości między 3 cokółami tak, aby obiektu większego nie stawiać na mniejszym. Początkowo na pierwszym cokole znajduje się 5 obiektów, ustawionych jeden na drugim. Najmniejszy obiekt przemieszcza się w co drugim, nieparzystym ruchu, a podczas ruchów parzystych pozostają tylko jedyne możliwości ułożenia obiektu mniejszego na większym.

Zaprojektuj program służący do prezentacji algorytmu z łamigłówki Wieże Hanoi. Program ma animować ruch obiektów (imitujących pierścienie) co 1 sekundę. Łamigłówka to zadanie polegające na przestawieniu pojedynczo pięciu obiektów z lewego cokołu docelowo na cokol prawy, z pomocą cokołu środkowego. WARUNEK: Można kłaść tylko obiekty mniejsze na większe.



W oknie formularza należy umieścić kolejno pięć pól na obrazki (TImage), które będą przedstawiać kolejne krążki oraz obiekt Timera (TTimer).



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

- „Krazek1”, „Krazek2”, „Krazek3”, „Krazek4”, „Krazek5” dla obiektów typu TImage przedstawiających kolejne krążki;
- „Timer1” dla obiektu typu TTimer;

Kod:

```
{W celu uproszczenia obliczeń oznaczono POZYCJĘ z lewej jako 0, środkową 1, a prawą jako 2.}
unit wieze_hanoi;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs,
  ExtCtrls;
type
  {T0kno}
  T0kno = class(TForm)
    Krazek1: TImage;           {Zmienne reprezentujące obiekty}
    Krazek2: TImage;
    Krazek3: TImage;
    Krazek4: TImage;
    Krazek5: TImage;
    Timer1: TTimer;
    procedure FormCreate(Sender: TObject);      {deklaracja procedury startowej}
    procedure Timer1Timer(Sender: TObject);    {procedura wykonywana co sekundę}
  private
    {private declarations}
    ileP: array[0..2] of Integer;              {Tablica 3-elementowa przechowuje liczby określające
                                              ile pierścieni (od 0 do 5) jest w danej chwili na pozycjach 0, 1 i 2}
    pozP: array[1..5] of Integer;             {Tablica 5-elementowa przechowuje liczby określające
                                              aktualne pozycje (od 0 do 2) pięciu pierścieni}
  end;
end;
```

2



```
i: Integer;           {Zmienna iteracyjna, wskazuje numer ruchu (liczbę ruchów)}
public
  {public declarations}
end;
var
  Okno: T0kno;

implementation
  {T0kno}
  procedure T0kno.FormCreate(Sender: TObject); {Procedura wykonywana przy tworzeniu okna,
                                                jej deklaracja tworzona jest po dwukrotnym kliknięciu na pustą przestrzeń formularza}
  begin
    {przypisanie wartości początkowych wszystkim zmiennym}
    ileP[0]:= 5; ileP[1]:= 0; ileP[2]:= 0;
    pozP[1]:= 0; pozP[2]:= 0; pozP[3]:= 0; pozP[4]:= 0; pozP[5]:= 0;
  end;

  procedure T0kno.Timer1Timer(Sender: TObject); {Procedura wywoływana cyklicznie przez Timer
                                                co sekundę; jej deklaracja tworzona jest po dwukrotnym kliknięciu w obiekt Timer}
  begin
    i:= i+1;           {Liczenie ruchów}
    if((i-1) mod 2)=0 then {Przy ruchach nieparzystych obsłuż obiekt najmniejszy – pierwszy}
    begin
      ileP[pozP[1]]:= ileP[pozP[1]]-1; {Na dotychczasowej pozycji zmniejsz liczbę, gdyż stamtąd... }
                                       { ma ubyc jeden obiekt}
      pozP[1]:= pozP[1]-1;             {Cofnij w lewo pozycję obiektu}
      if pozP[1]<0 then pozP[1]:= 2;    {Jeśli był on na pozycji 0, to zmień na pozycję 2}
      ileP[pozP[1]]:= ileP[pozP[1]]+1; {Na nowej pozycji zwiększ liczbę obiektów}
      Krazek1.Left:= 50 + pozP[1] * 250; {Przesuń obiekt poziomo, w osi x}
      Krazek1.Top:= 270 - ileP[pozP[1]] * 50; {Przesuń obiekt pionowo, w osi y}
    end

    else if((i-16) mod 32)=0 then      {Co 32 ruchy obsłuż obiekt największy – piąty}
    begin
      ileP[pozP[5]]:= ileP[pozP[5]]-1; {Na dotychczasowej pozycji zmniejsz liczbę}
      pozP[5]:= pozP[5]-1;             {Cofnij w lewo pozycję obiektu}
      if pozP[5]<0 then pozP[5]:= 2;    {Jeśli był on na pozycji 0, to zmień na pozycję 2}
      ileP[pozP[5]]:= ileP[pozP[5]]+1; {Na nowej pozycji zwiększ liczbę obiektów}
      Krazek5.Left:= 50 + pozP[5] * 250; {Przesuń obiekt poziomo, w osi x}
      Krazek5.Top:= 270 - ileP[pozP[5]] * 50; {Przesuń obiekt pionowo, w osi y}
    end

    else if((i-8) mod 16)=0 then      {Co 16 ruchów obsłuż krążek czwarty}
    begin
      ileP[pozP[4]]:= ileP[pozP[4]]-1; {Na dotychczasowej pozycji zmniejsz liczbę}
      pozP[4]:= pozP[4]+1;             {Przesuń w prawo pozycję obiektu}
      if pozP[4]>2 then pozP[4]:= 0;    {Jeśli był na pozycji 2, to zmień na pozycję 0}
      ileP[pozP[4]]:= ileP[pozP[4]]+1; {Na nowej pozycji zwiększ liczbę obiektów}
      Krazek4.Left:= 50 + pozP[4] * 250; {Przesuń obiekt poziomo, w osi x}
      Krazek4.Top:= 270 - ileP[pozP[4]] * 50; {Przesuń obiekt pionowo, w osi y}
    end

    else if((i-4) mod 8)=0 then      {Co 8 ruchów obsłuż krążek trzeci}
    begin
      ileP[pozP[3]]:= ileP[pozP[3]]-1; {Na dotychczasowej pozycji zmniejsz liczbę}
      pozP[3]:=pozP[3]-1;             {Cofnij w lewo pozycję obiektu}
      if pozP[3]<0 then pozP[3]:= 2;    {Jeśli był na pozycji 0, to zmień na pozycję 2}
      ileP[pozP[3]]:= ileP[pozP[3]]+1; {Na nowej pozycji zwiększ liczbę obiektów}
    end
  end
end
```



```

Krazek3.Left:= 50 + pozP[3] * 250;      {Przesuń obiekt w osi x}
Krazek3.Top:= 270 - ileP[pozP[3]] * 50; {Przesuń obiekt w osi y}
end
else if((i-2) mod 4)=0 then             {Co 4 ruchy obsłuż krążek drugi}
begin
  ileP[pozP[2]]:= ileP[pozP[2]]-1;     {Na dotychczasowej pozycji zmniejsz liczbę}
  pozP[2]:= pozP[2]+1;                 {Przesuń w prawo pozycję obiektu}
  if pozP[2]>2 then pozP[2]:= 0;        {Jeśli był na pozycji 2, to zmień na pozycję 0}
  ileP[pozP[2]]:= ileP[pozP[2]]+1;     {Na nowej pozycji zwiększ liczbę obiektów}
  Krazek2.Left:= 50 + pozP[2] * 250;   {Przesuń obiekt w osi x}
  Krazek2.Top:= 270 - ileP[pozP[2]] * 50; {Przesuń obiekt w osi y}
end;
if ileP[2]=5 then Timer1.Enabled:=False; {Jeśli już wszystkie objekty przeniesione, to zakończ
                                         generowanie przerw przez Timer}
end;
initialization
  {$I wieze_hanoi.lrs}
end.

```

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

Optymalna strategia przy ręcznym przemieszczaniu obiektów:

- Na przemian, przy każdym kroku o numerze nieparzystym co drugie posunięcie wykonuje obiekt najmniejszy, a dla pozostałych obiektów pozostaje tylko jedyny możliwy ruch podczas kroków parzystych.

Dla parzystej liczby obiektów:

- objekty o indeksach nieparzystych krążą w prawo, wracając skokiem z cokołu ostatniego {1,2,0,1,2,...}, a o indeksach parzystych odwrotnie, skaczą najpierw na cokół ostatni i wracając w lewo {2,1,0,2,1...};
- oznacza to, że obiekt najmniejszy z indeksem 1 rozpoczyna, przechodząc z cokołu 0 na cokół środkowy 1.

Dla nieparzystej liczby obiektów:

- **objekty o indeksach nieparzystych skaczą najpierw na cokół ostatni i wracając w lewo {2,1,0,2,1...}, a o indeksach parzystych odwrotnie, krążą w prawo i wracają skokiem z cokołu ostatniego {1,2,0,1,2,...}.**