



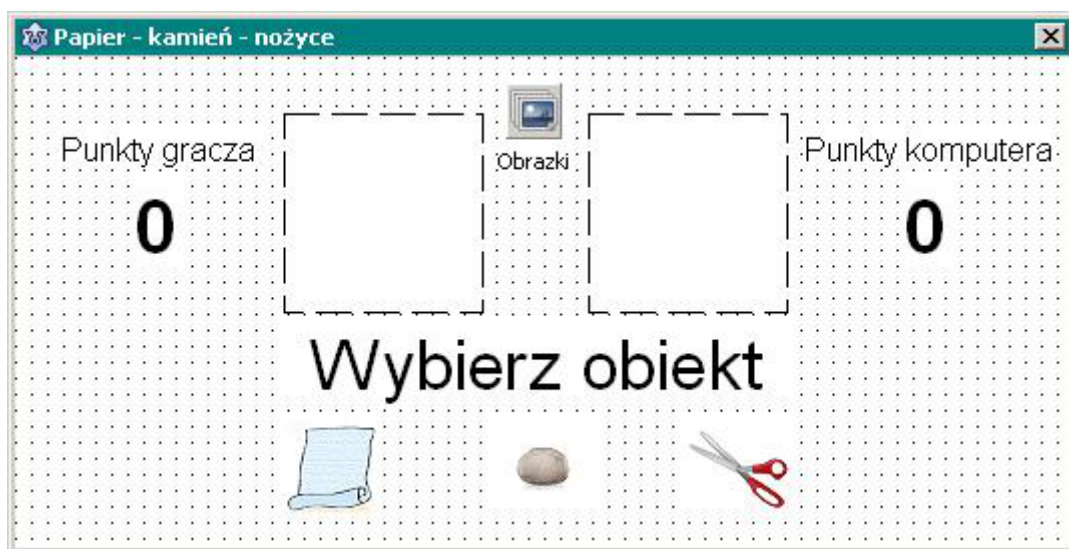
Nazwa implementacji: Gra "papier-kamień-nożyce"

Autor: Stanisław Ubermanowicz Piotr Fiorek

Opis implementacji: Realizacja gry losowej, polegającej na równoczesnym wyborze dwóch przedmiotów, z których jeden wygrywa według zasady: papier owija kamień, kamień tępi nożyce, a nożyce tną papier. Gra powinna mieć 3 przyciski z miniaturami przedmiotów. Kliknięcie oznacza wybór obiektu przez gracza i powoduje wylosowanie obiektu dla komputera jako przeciwnika. Pojawiają się dwa duże obrazki przedmiotów i zależnie od układu liczone są punkty za wygraną lub remis.

Zaprojektuj i wykonaj kompletną grę, która polega na wyborze i porównywaniu obiektów. Jest to rodzaj pojedynku Gracza z komputerem. W każdym ruchu Gracz poprzez kliknięcie na miniaturze obrazka wybiera jeden z trzech przedmiotów: papier, kamień lub nożyce. Uruchamia to procedurę losowania przedmiotu dla komputera, po czym następuje porównanie. W przypadku wyboru takich samych obiektów jest remis, a gdy są one różne, to wygrywa strona silniejsza według zasady: papier owija kamień, kamień tępi nożyce, nożyce tną papier.

Wygląd przykładowego okna projektu



Obiekty i ich właściwości
Obiekty ekranowe utwórz za pomocą myszy z widżetów TImageList, TImage oraz TLabel i nadaj im odpowiednie Właściwości w odpowiednich polach Inspektora obiektów Lazarusa: **Wstaw specjalny kontener TImageList do przechowywania obrazków numerowanych 0, 1, 2; pojawi się ikonka symbolizująca plik obrazków - nadaj nazwę temu obiektowi Name: Obrazki; podaj wymiary przygotowanych uprzednio obrazków (np. Height: 100 i Width: 100); kliknij ikonkę dwukrotnie i w oknie Edytor ImageList wczytaj kolejno 3 obrazki z plików. Wstaw trzy mniejsze kontenerki TImage z obrazkami, jako przyciski do wyboru przez gracza, nazwij je odpowiednio w polu Caption: Papier, Kamień, Nożyce (nie używaj polskich liter); ustal wymiary tych kontenerków mniejsze niż obrazki źródłowe (np. Height: 50 i Width: 50); ustaw dopasowywanie obrazka do wielkości kontenerka, wpisując w polu Stretch: True; wczytaj kolejno obrazki w polu Picture, wskazując w oknie dialogowym ścieżki do plików. Wstaw dwa większe kontenerki TImage dla porównań obiektu wybranego z wylosowanym; nazwij je odpowiednio w polu Name: Gracz, Komputer (nazwy te są użyte w tworzeniu kodu). Wstaw dwa pola tekstowe TLabel, które będą nagłówkami z informacją, czyja to punktacja; nazwij je w polu Name: EtykietaGracza, EtykietaKomputera (nazwy te nie będą używane); wpisz odpowiednie teksty ze spacjami w polach Caption: Punkty gracza, Punkty komputera. Wstaw dwa pola tekstowe TLabel, które będą wyświetlały zsumowaną punktację w grze; nazwij je w polu Name: PunktyGracza, PunktyKomputera (nazwy zawsze wpisuj bez spacji); wpisz stan zerowy punktacji w polach Caption: 0 oraz 0; ustaw czcionkę większą niż domyślna w oknie dialogowym wywołanym z pola Font. Wstaw jedno pole tekstowe TLabel, które będzie wyświetlało wynik porównania obiektów; nazwij ten obiekt w polu Name: Wynik; wpisz tekst z instrukcją początkową dla gracza w polu Caption: Wybierz obiekt; wyłącz automatyczne dopasowywanie wielkości pola do tekstu w opcji AutoSize: False; wybierz z listy tryb wyrównywania tekstu do środka w polu Alignment: taCenter; ustaw czcionkę większą niż domyślna w oknie dialogowym wywołanym z pola Font. Kliknij puste pole okna i zmień kolor tła na dopasowany do tła obrazków w polu Color.**

Tworzenie kodu źródłowego i przypisywanie obiektom obsługi zdarzeń
Dalszą część zadania wykonuje się na zajęciach dokładnie według instrukcji Trenera. W przypadku samodzielnych prób napisania kodu źródłowego należy pamiętać o połączeniu obiektów z procedurami obsługi zdarzeń, w zakładce Zdarzenia okna Inspektora obiektów. Potrzebne są procedury obsługi kliknięć Gracza na trzech miniaturkach,



procedura losowania wyboru obiektu dla komputera oraz procedura porównywania z ustalaniem wyniku i punktacji.

Przykładowe okno podczas gry



Kod źródłowy

```
unit gra;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs, ExtCtrls, StdCtrls;
type
  { TOkno }
  TOkno = class(TForm)
    Papier: TImage;
    Kamien: TImage;
    Nozyce: TImage;
    Gracz: TImage;
    Komputer: TImage;
    Obrazki: TImageList;
    EtykietaGracza: TLabel;
    EtykietaKomputera: TLabel;
    PunktyGracza: TLabel;
    PunktyKomputera: TLabel;
    Wynik: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure PapierClick(Sender: TObject);
    procedure KamienClick(Sender: TObject);
    procedure NozyceClick(Sender: TObject);
    procedure RuchKomputera();
    procedure PoliczWynik();
  private
    { private declarations }
    gracz_wybral, komputer_wybral: Integer;
    punkty_gracza, punkty_komputera: Integer;
  public
    { public declarations }
  end;
```





```
var
  Okno: TOkno;
implementation
{ TOkno }
procedure TOkno.FormCreate(Sender: TObject);
begin
  Randomize();
end;
procedure TOkno.PapierClick(Sender: TObject);
begin
  gracz_wybral:=0;
  Obrazki.GetBitmap(0, Gracz.Picture.Bitmap);
  RuchKomputera();
end;

procedure TOkno.KamienClick(Sender: TObject);
begin
  gracz_wybral:=1;
  Obrazki.GetBitmap(1, Gracz.Picture.Bitmap);
  RuchKomputera();
end;
procedure TOkno.NozyceClick(Sender: TObject);
begin
  gracz_wybral:=2;
  Obrazki.GetBitmap(2, Gracz.Picture.Bitmap);
  RuchKomputera();
end;
procedure TOkno.RuchKomputera();
begin
  komputer_wybral:=Random(3) div 1;
  Obrazki.GetBitmap(komputer_wybral, Komputer.Picture.Bitmap);
  PoliczWynik();
end;
procedure TOkno.PoliczWynik();
begin
  if gracz_wybral=komputer_wybral then
  begin
    Wynik.Caption:='Remis';
  end
  else if (gracz_wybral+1 = komputer_wybral) or (gracz_wybral = komputer_wybral+2) then
  begin
    Wynik.Caption:='Wygrana';
    punkty_gracza:=punkty_gracza+1;
  end
  else
  begin
    Wynik.Caption:='Przegrana';
    punkty_komputera:=punkty_komputera+1;
  end;
  PunktyGracza.Caption:=IntToStr(punkty_gracza);
  PunktyKomputera.Caption:=IntToStr(punkty_komputera);
end;
initialization
  {$I gra.lrs}
end.
```

