



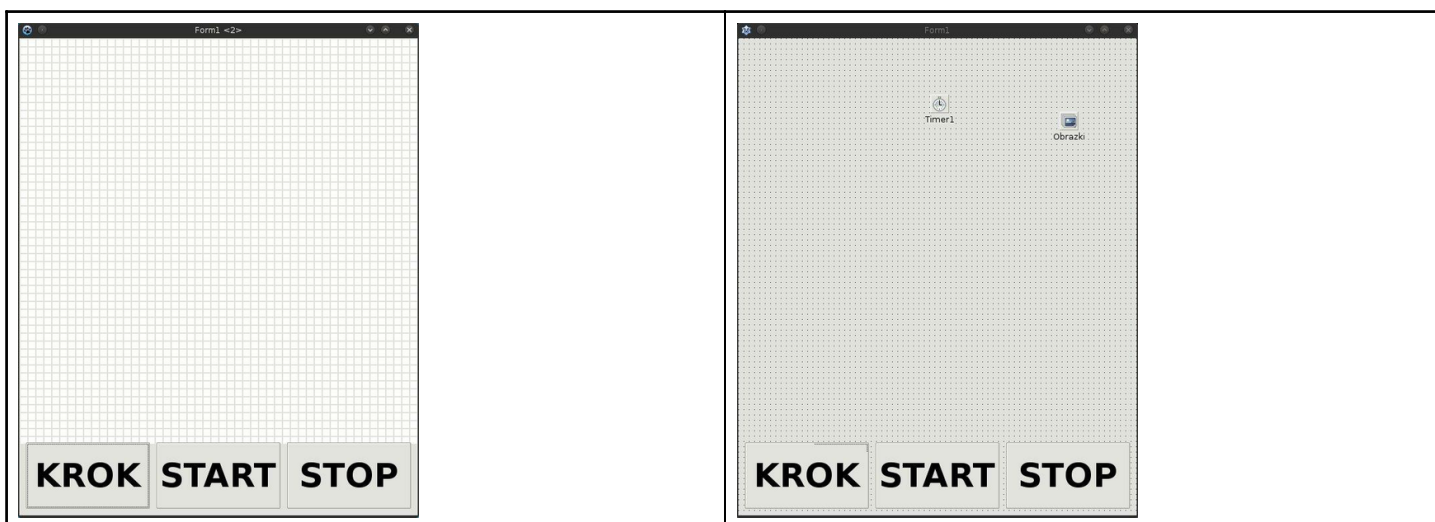
Nazwa implementacji: Gra w życie

Autor: Piotr Fiorek

Opis implementacji:

Implementacja przedstawia „Grę w życie” jako przykład prostej symulacji opartej na automatach.

Zaprojektuj prostą grę, której celem jest odwzorowanie popularnej gry w życie.



Okno Formularza

Należy dodać trzy przyciski, timer oraz kontener na obrazki (TPictureList). W przypadku TPictureList należy określić jego rozmiary (powinien być taki sam jak rozmiar pól obrazkowych, które zamierzamy stworzyć, w przeciwnym wypadku obrazki będą przycięte) oraz po dwukrotnym kliknięciu w obiekt listę obrazków przedstawiających komórkę żywą, oraz martwą. W oknie „Object Inspector” można również ustawić nazwy poszczególnych elementów. W tej implementacji używamy nazw:

- „Obrazki” dla obiektu TpictureList
- „Timer1” dla obiektu Timer
- „KROK” dla pierwszego przycisku
- „START” dla drugiego przycisku



- „STOP” dla trzeciego przycisku

Kod:

```
unit zycie;
```

```
{$mode objfpc}{$H+}
```

```
interface
```

```
uses
```

```
Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs,  
StdCtrls, ExtCtrls;
```

```
type
```

```
{ TForm1 }
```

```
TForm1 = class(TForm)
```

```
    // deklaracje elementów w oknie
```

```
    Obrazki: TImageList;
```

```
    KROK: TButton;
```

```
    START: TButton;
```

```
    STOP: TButton;
```

```
    Timer1: TTimer;
```

```
    // deklaracje procedur
```

```
    procedure FormCreate(Sender: TObject);
```

```
    procedure KROKClick(Sender: TObject);
```

```
    procedure STARTClick(Sender: TObject);
```

```
    procedure STOPClick(Sender: TObject);
```

```
    procedure Timer1Timer(Sender: TObject);
```

```
    procedure KomorkaClick(Sender: TObject);
```

```
    procedure PrzerysujEkran();
```

```
private
```

```
    // zmienne używane w programie
```

```
    gramy: Boolean; // zmienna przechowująca informację czy gra się toczy
```

```
    biezaca, nastepna: array [0..50,0..50] of Boolean; // tablice stanu gry - bierzaca oraz tymczasowa do przeliczenia stanu w
```

```
    komorki: array[0..50, 0..50] of TImage; // tablica z obrazkami komórek
```

```
public
```

```
    { public declarations }
```

```
end;
```

```
var
```

```
    Form1: TForm1;
```

```
implementation
```

```
{ TForm1 }
```

```
// procedura wywoływana podczas tworzenia okna
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
var
```

```
    i, j: Integer; // zmienne używane jako iteratory
```

```
begin
```

```
    gramy:= False; // na początku gra jest przzerwana
```





```
// tworzenie komorek
for i:=0 to 50 do // przechodzenie po rzędach tablic
begin
  for j:=0 to 50 do // przechodzenie po kolumnach tablic
  begin
    komorki[i,j]:= TImage.Create(self); // stworzenie komórki
    komorki[i,j].Parent:= self; // przypisanie jej do ekranu
    komorki[i,j].Width:= 10; // określenie szerokości komórki
    komorki[i,j].Height:= 10; // określenie wysokości komórki
    Obrazki.GetBitmap(0, komorki[i,j].Picture.Bitmap); // załadowanie obrazka do komórki
    komorki[i,j].Left:= (j+1)*2 + j*10; // określenie położenia komórki od lewej krawędzi okna
    komorki[i,j].Top:= (i+1)*2 + i*10; // określenie położenia komórki od górnej krawędzi okna
    komorki[i,j].Tag:= i*50+j; // przypisanie komórce numeru
    komorki[i,j].OnClick:= @KomorkaClick; // przypisanie do komórki funkcji wywoływanej w momencie kliknięcia w komórkę

    biezaca[i,j]:=False; // ustawienie komórki na stan "martwa"
  end;
end;

nastepna:=biezaca; // skopiowanie stanu tablicy
end;

// funkcja obliczająca stan gry w następnym kroku
procedure TForm1.KROKClick(Sender: TObject);
var
  sasiedzi, rzad, kolumna: Integer; // zmienne pomocnicze
begin
  // tutaj sie wszystko dzieje
  for rzad:=1 to 50 do // przejdź po wszystkich rzędach
  begin
    for kolumna:=1 to 50 do // przejdź po wszystkich kolumnach
    begin
      sasiedzi:=0; // zmienna przechowująca informację o ilości żywych sąsiadów komórki
      if rzad>0 then // sprawdzenie czy istnieje rząd powyżej komórki
      begin
        if biezaca[rzad-1, kolumna] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy komórka powyżej jest żywa i zwiększenie
        if kolumna>0 then if biezaca[rzad-1, kolumna-1] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy istnieje lewy sąsiad
        if kolumna<50 then if biezaca[rzad-1, kolumna+1] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy istnieje prawy sąsiad
      end;
      if rzad<50 then // sprawdzenie czy istnieje rząd poniżej komórki
      begin
        if biezaca[rzad+1, kolumna] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy komórka poniżej jest żywa i zwiększenie
        if kolumna>0 then if biezaca[rzad+1, kolumna-1] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy komórka poniżej ma lewego sąsiada
        if kolumna<50 then if biezaca[rzad+1, kolumna+1] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy komórka poniżej ma prawego sąsiada
      end;
      if kolumna>0 then if biezaca[rzad, kolumna-1] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy komórka na lewo istnieje
      if kolumna<50 then if biezaca[rzad, kolumna+1] = True then sasiedzi:=sasiedzi+1; // sprawdzenie czy komórka na prawo istnieje

      if (sasiedzi<2) or (sasiedzi>=4) then nastepna[rzad,kolumna]:=False; // jeśli komórka mniej niż dwóch, lub więcej niż trzech sąsiadów
      if sasiedzi=2 then nastepna[rzad,kolumna]:=biezaca[rzad,kolumna]; // jeśli komórka ma dwóch sąsiadów w następnej rundzie
      if sasiedzi=3 then nastepna[rzad,kolumna]:=True; // jeśli komórka ma trzech sąsiadów w następnej rundzie będzie żywa
    end;
  end;
end;
biezaca:=nastepna; // skopiowanie tablicy reprezentującej następną rundę do aktualnej tablicy
```





```
PrzerysujEkran(); // przerysowanie ekranu nowym stanem gry
end;

// rozpoczęcie lub wznowienie gry
procedure TForm1.STARTClick(Sender: TObject);
begin
  gramy:= True;
end;

// przerwanie gry
procedure TForm1.STOPClick(Sender: TObject);
begin
  gramy:= False;
end;

// procedura wywoływana co sekundę przez timer
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if gramy = True then KROKClick(self); // jeśli gra się toczy wywołuje funkcję generującą następny krok w grze
end;

// procedura wywoływana w momencie kliknięcia w komórkę
procedure TForm1.KomorkaClick(Sender: TObject);
var
  rzad, kolumna: Integer; // zmienne pomocnicze
begin
  kolumna:= (Sender as TImage).Tag mod 50; // wyciąganie numeru kolumny klikniętej komórki z jej numeru
  rzad:= (Sender as TImage).Tag div 50; // wyciąganie numeru rzędu z klikniętej komórki

  if biezaca[rzad, kolumna] = True then // jeśli komórka była "żywa"
  begin
    biezaca[rzad, kolumna]:= False; // zmieniamy jej stan
    nastepna[rzad, kolumna]:= False; // zmieniamy jej stan
    Obrazki.GetBitmap(0, komorki[rzad, kolumna].Picture.Bitmap); // zmieniamy jej obrazek
  end
  else
  begin
    biezaca[rzad, kolumna]:= True; // zmieniamy jej stan
    nastepna[rzad, kolumna]:= True; // zmieniamy jej stan
    Obrazki.GetBitmap(1, komorki[rzad, kolumna].Picture.Bitmap); // zmieniamy jej obrazek
  end;
end;

// funkcja przenosząca stan tablicy na komórki na ekranie
procedure TForm1.PrzerysujEkran();
var
  i, j: Integer; // zmienne używane jako iteratory
begin
  for i:=0 to 50 do // przechodzenie po rzędach tablicy
    for j:=0 to 50 do // przechodzenie po kolumnach tablicy
      if biezaca[i,j] = True then // jeśli komórka jest żywa
        Obrazki.GetBitmap(1, komorki[i,j].Picture.Bitmap) // załaduj obrazek żywej
      else
        Obrazki.GetBitmap(0, komorki[i,j].Picture.Bitmap); // załaduj obrazek martwej
    end;
  end;
end;
```





```
initialization  
{ $I zycie.lrs }
```

```
end.
```

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

