

GRA EDUKACYJNA „LIVE” - część II

Potrafiśmy rysować pole gry, zmieniać wymiary tego pola, zapalać i gasić pola. Aby zagrać, należy zapamiętać początkowe położenia poszczególnych robaczek, wyliczać położenia robaczek w kolejnym pokoleniu i przerysować tablicę.

TABLICA

Chcemy zapamiętać, gdzie znajduje się zaznaczone pole - najlepiej zastosować zmienną tablicową. Tablica dwuwymiarowa, o wymiarach 1..100 w obie strony, komórki typu całkowitego. Jeśli pole na ekranie jest szare, komórka tablicy ma wartość 0, gdy pole na ekranie ma kolor biały - ustawiamy wartość 1 w komórce tablicy.

- Zadeklaruj tablicę o nazwie **LIVE** jako zmienną globalną (po instrukcji `var Form1:TForm;` wpisz deklarację tablicy z ramki)

```
LIVE: array[1..100,1..100] of integer;
```

(1) Pokaż deklarację nauczycielowi

Początkowo pola tablicy są szare więc nadamy im wszystkim wartość 0.

- Utwórz nową procedurę o nagłówku **PROCEDURE TForm1.ZerowanieLIVE;**
- W procedurze
 - Wyzeruj wszystkie komórki tablicy LIVE (patrz ramka)
 - Zadeklaruj w procedurze niezbędne zmienne
 - Umieść nagłówek nowej procedury w górnej części programu

```
for i:=1 to ile do  
for j:=1 to ile do  
LIVE[i,j]:=0;
```

- Wywołaj tę procedurę **ZerowanieLIVE** w procedurze **FormActivate**

Aby sprawdzić działanie programu wypiszemy zawartość tablicy na ekranie

- Przejdź do procedury **Button1Click** (kliknij podwójnie w przycisk **RYSUJ**)

- Wpisz instrukcje z ramki („rysuje” na ekranie liczby z tablicy - zera)

(X i Y - obliczamy miejsce wypisania liczby na ekranie, funkcja **IntToStr** zamienia liczbę na napis, zadeklaruj odpowiednie zmienne - UWAGA - zmienna „t” jest typu **STRING**)

```
ile:=Scrollbar1.Position;  
bok:=600 div ile;  
for i:=1 to ile do  
for j:=1 to ile do  
begin  
x:=10+bok*(i-1);  
y:=10+bok*(j-1);  
t:=IntToStr(LIVE[i,j]);  
canvas.textout(x,y,t);  
end;
```

(2) Pokaż wynik swojej pracy nauczycielowi - zera w każdej kratce

WSPÓŁRZĘDNE

Bardzo przydatna będzie umiejętność wyliczania położenia pola na formacie (lewy, górny róg) i numeru komórki w tablicy, która odpowiada temu polu. Posłużą do tego dwie funkcje: **WSPnaTAB** która wylicza numer komórki w tablicy na podstawie współrzędnych myszki i **TABnaWSP**, która wylicza lewy górny róg rysowanego pola na podstawie numeru komórki. Ponieważ pola są kwadratowe, wystarczy, że wyliczymy jedną współrzędną - tutaj X.

- Wklej instrukcje z ramek do programu
- Umieść nagłówki funkcji w odpowiednim miejscu (parametry wejściowe „I” i „X” są typu całkowitego i funkcje dają wyniki typu całkowitego)

Aby sprawdzić działanie programu

- Pod suwakiem umieść etykietę: „**TABLICA:**”
- Obok niej umieść jeszcze etykietę: „**(I,J)**”
- W procedurze **FormMouseDown** wpisz instrukcje, które będą wypisywały tekst w postaci „(I,J)”, np. (3,2), gdy klikniemy w odpowiednie pole - patrz ramka:

(X i Y punkt kliknięcia myszką, a „I” i „J” to przeliczone numery komórek w tablicy, sprawdź, czy etykieta, do której wpisujesz ilość pól nosi nazwę **Label6**)

(3) Pokaż wynik swojej pracy nauczycielowi - po kliknięciu w kratkę współrzędne kratki w etykiecie

Potrafiemy „zapalać” (na biało) lub „gasić” (na szaro) pola tablicy. Po każdej zmianie wymiaru białe pola jednak znikają, bo nie zapamiętaliśmy nigdzie ich położenia. Punkt kliknięcia myszką (xM,yM) zamieniamy za pomocą nowych funkcji na współrzędne pola w tablicy **LIVE** i zapisujemy w tablicy.

- W procedurze **POLE**, na samym końcu, dopisz instrukcję warunkową z ramki
- Zadeklaruj brakujące zmienne

(jeżeli kolorujemy pole na biało, do tablicy wpisujemy 1, gdy na szaro - 0)

(+3) Pokaż wynik swojej pracy nauczycielowi - przycisk **RYSUJ** pokazuje jedynki gdy pole białe

```
i:=WSPnaTAB(xM);  
j:=WSPnaTAB(yM);  
if kolor=clWhite  
then LIVE[i,j]:=1  
else LIVE[i,j]:=0;
```

```
function TForm1.TABnaWSP(i:integer):integer;  
var ile,bok:integer;  
begin  
ile:=Scrollbar1.Position;  
bok:=600 div ile;  
TABnaWSP:=(i-1)*bok+2;  
end;
```



```
function TForm1.WSPnaTAB(x:integer):integer;  
var ile,bok:integer;  
begin  
ile:=Scrollbar1.Position;  
bok:=600 div ile;  
WSPnaTAB:=((2+bok*(x div bok)) div bok)+1;  
end;
```

```
i:=WSPnaTAB(x);  
j:=WSPnaTAB(y);  
Label6.Caption:='('+IntToStr(i)+','+IntToStr(j)+')';
```

RYSOWANIE

Po każdej zmianie wymiarów suwakiem (nowa liczba wierszy i kolumn) pola należy przerysować zgodnie z ustawieniami w tablicy LIVE. Tym razem numer komórki w tablicy zamieniamy za pomocą funkcji TABnaWSP na współrzędną lewego górnego rogu pola na formatce, aby wyliczyć, jak ma być rysowane pole

- Utwórz nową procedurę o nazwie **RysowanieLIVE** bez parametrów;
- W procedurze wykonaj następujące instrukcje:
 - wstaw do zmiennej „ILE” pozycję paska ScrollBar1
 - podwójna pętla FOR ze zmiennymi „I” i „J” wykonuje poniższe instrukcje od 1 do „ILE”
 - wyliczamy „X” i „Y” z „I” i „J” za pomocą funkcji TABnaWSP
 - jeżeli komórka tablicy LIVE o współrzędnych [i,j] jest równa zero
 - za pomocą instrukcji POLE narysuj szare pole
 - w przeciwnym razie narysuj białe pole
- Umieść nagłówek nowej procedury w górnej części programu
- Zadeklaruj niezbędne zmienne
- Umieść wywołanie procedury w procedurze **ScrollBar1Change**

[i-1, j-1]	[i, j-1]	[i+1, j-1]
[i-1, j]	[i, j]	[i+1, j]
[i-1, j+1]	[i, j+1]	[i+1, j+1]

(4) Sprawdź działanie programu po zmianie wymiaru tablicy

SPRAWDZANIE

Kolejna funkcja policzy ile jest robaczków wokół wybranego pola. Sprawdzamy kolejno wszystkie osiem pól. Gdy zliczamy robaczki wokół pola na brzegu nie możemy sprawdzać komórek tablicy, których nie ma (np. [0,-1]), bo program się zawiesi.

- Utwórz nową funkcję o nazwie **ILEROBACZKOW**. Parametry wejściowe „i” i „j” typu całkowitego (numer komórki tablicy LIVE), wynik typu całkowitego
- Wpisz do tej funkcji następujące instrukcje:
 - wstaw do zmiennej „ILE” pozycję paska ScrollBar1
 - zmienna „S” przechowuje sumę robaczków wokół badanego pola - wyzeruj ją
 - jeżeli badane pole jest w tablicy LIVE, to zwiększ wartość „S” o zawartość komórki tablicy LIVE

Wszystkie warunki logiczne będą zbudowane w podobny sposób: jeżeli $i > 0$ (gdy z lewej strony) albo $i < \text{ile}$ (gdy z prawej strony), to do zmiennej „s” wstaw sumę „s” i wartości komórki LIVE z odpowiedniej strony - np. dla komórki z prawej strony: **if i<ile then s:=s+LIVE[i+1,j];** a dla komórki z lewej strony u góry: **if (i>0) and (j>0) then s:=s+LIVE[i-1,j-1];** itp.

- do nazwy funkcji przypisz sumę robaczków ze zmiennej „s”
- Umieść nagłówek nowej procedury w górnej części programu
- Zadeklaruj niezbędne zmienne
- Aby sprawdzić działanie funkcji
- umieść na formatce etykietę i napisz na niej **ROBACZKI**
- umieść obok niej kolejną etykietę i napisz w niej 0
- do procedury **FormMouseDown** wpisz instrukcję: **Label8.Caption:=IntToStr(IleRobaczkow(i,j));** (sprawdź, czy pole, do którego wpisujesz nosi nazwę Label8, zamiana współrzędnych „X” i „Y” na „i” i „j” już tam jest)

(5) Sprawdź działanie programu klikając w pola

POKOLENIA

Gra toczy się w turach. Kolejne pokolenie pojawi się po wyliczeniu położenia nowego pokolenia robaczków i narysowaniu na nowo całej tablicy. Jak tworzy się nowe pokolenie? Badany jest teren dokoła każdego robaczka. Jeśli wokół pustego pola znajduje się dokładnie 3 robaczki, to rodzi się na tym polu nowy robaczek. Jeśli wokół pola z robaczkiem jest dokładnie 2 lub 3 robaczki – to ten robaczek przeżywa. W każdym innym przypadku, gdy wokół pola z robaczkiem jest 0,1,4,5,6,7,8 innych robaczków – robaczek ginie (samotność lub tłok).

Wykonujemy kolejny ruch i pojawia się kolejne pokolenie robaczków - część z nich umiera lub rodzą się nowe. Wystarczy sprawdzić wszystkie komórki tablicy (ile wokół każdej jest robaczków) i na tej podstawie wygenerować nowe pokolenie.

- Dopisz zmienną LV do deklaracji tablicy LIVE (chcemy mieć dwie identyczne tablice)
- Utwórz nową procedurę o nagłówku: **TForm1.PokolenieLIVE(Sender: TObject);**
- Umieść nagłówek nowej procedury w górnej części programu
- UWAGA nagłówek umieść koniecznie przed słowem **private**
- Wpisz do procedury następujące instrukcje:
 - wstaw do zmiennej „ile” pozycję paska ScrollBar1
 - do zmiennej „LV” przypisz tablicę „LIVE”
 - w podwójnej pętli o zmiennych sterujących „I” i „J” od 1 do „ILE”
 - wylicz ilość robaczków wokół komórki (I,J) za pomocą funkcji ILE ROBACZKOW i wynik funkcji przypisz do zmiennej „R”

- jeżeli LIVE[i,j] jest równe 0 i wokół jest 3 robaczki (zmienna „R”), to do LV[i,j] wpisz 1
- jeżeli LIVE[i,j] jest równe 1 i wokół jest 2 lub 3 robaczki, to do LV[i,j] wpisz 1
(2 lub 3 robaczki - wygodniej użyć konstrukcji **R in [2,3]**)
- jeżeli LIVE[i,j] jest równa 1 i wokół jest 0,1,4,5,6,7 lub 8 robaczków, to do LV[i,j] wpisz 0
 - do zmiennej „LIVE” przypisz tablicę „LV”
 - Wywołaj przerysowanie szachownicy - instrukcja **RysowanieLIVE**

• Zadeklaruj niezbędne zmienne

• Przypisz tę procedurę **PokolenieLIVE(sender)**; do zdarzenia OnClick dla przycisku RYSUJ

(możesz usunąć niepotrzebne instrukcje w przycisku RYSUJ)

(+5) Sprawdź działanie programu - narysuj układ początkowy i klikaj w przycisk

- Pod suwakiem umieść etykietę i wstaw tekst: „**POKOLENIE:**”
- Obok niej umieść jeszcze jedną etykietę i wstaw tekst: „0”
- zadeklaruj zmienną globalną o nazwie **POK** typu całkowitego
- w procedurze **FormActivate** wyzeruj zmienną POK
- w procedurze **PokolenieLIVE** na końcu zwiększ zmienną POK o 1
- zawartość zmiennej POK wpisz do etykiety Label10

(-6) Sprawdź działanie programu - numer pokolenia na formatce

AUTOMATYCZNIE

• Umieść na formatce (w dowolnym miejscu z prawej strony) obiekt **TIMER** (zakładka SYSTEM)

• We właściwościach (PROPERTIES) tego obiektu ustaw: **Enabled: False, Interval: 10**

(zegar wyłączony, kolejne zdarzenia będą wywoływane co 10 milisekund)

• W zakładce zdarzeń (EVENTS), w polu **OnTimer: PokolenieLIVE**

(gdy włączymy zegar wywoływana będzie procedura PokolenieLIVE)

• Umieść na formatce nowy przycisk o nazwie START

• Kliknij w niego podwójnie

• Wpisz instrukcję warunkową: Jeżeli zegar jest włączony (**Timer1.Enabled=true**) to wyłącz go (**Timer1.Enabled:=False**) i wpisz do własności Caption przycisku napis „START”

(**Button2.Caption:=’START’**) w przeciwnym razie włącz zegar i wpisz do przycisku napis „STOP”

(6) Sprawdź działanie programu - narysuj początkowy układ i wciśnij przycisk START