

GRY KOMPUTEROWA

Potrafimy już przemieszczać obiekty po ekranie monitora. Co prawda są to kwadraty, ale nie stoi na przeszkodzie, aby napisać odpowiednie procedury i zamienić kwadraty na statki kosmiczne. Potrafimy sprawdzić, kiedy znajdują się w odpowiednim punkcie ekranu (zderzą się z innym obiektem). Jeśli dołożymy jeszcze umiejętność reagowania na wciskane klawisze, będziemy mogli pisać prawdziwe gry komputerowe!

Informacja o wciskanych klawiszach przechowywana jest w tzw. "buforze klawiatury". System pobiera je kolejno i wykonuje polecenia. Jeśli zbyt "znęcałeś" się nad klawiaturą, być może zdarzyło się, że usłyszałeś charakterystyczne piski. To znak, że bufor klawiatury się przepełnił - program nie jest w stanie tak szybko wykonać operacji

Poniższy prosty program służy do sprawdzania kodów wciskanych klawiszy. Należy pamiętać o tym, że część klawiszy (np. funkcyjne i kursory) "zostawia" po sobie dwa kody, z których pierwszy ma wartość 0 (zero). Funkcja **READKEY** zapamiętuje kod naciśniętego klawisza. Koniec programu nastąpi, gdy wciśniemy klawisz ESC (kod 27).

```
program kody;
uses crt;
var
  znak: char;
begin
  repeat
    znak:=readkey;
    writeln(znak:3,ord(znak):5);
  until znak=#27;
end.
```

Gra komputerowa LABIRYNT

Poniższy program będzie przesuwiał kwadrat zgodnie z kierunkiem naciskanych kursorów. Koniec programu nastąpi po naciśnięciu klawisza ESC. Cały program w pliku ANIM04.PAS
Zmienna KONIEC zawiera początkowo wartość FAŁSZ, Gdy w trakcie działania programu (po naciśnięciu klawisza ESC) przyjmie wartość PRAWDA pętla zakończy swoje działanie

```
koniec:=false;
repeat
  pilka(x,y,dx,dy,kp);      {rysowanie}
  delay(10);                {wstrzymanie}
  pilka(x,y,dx,dy,0);      {wymazanie}
  {...}
  {tutaj testujemy klawiaturę i przesuwamy piłkę}
  {...}
until koniec;              {koniec pętli, gdy naciśniemy klawisz ESC}
```

Testowanie i obsługę klawiatury

KEYPRESSED - jeszcze jedna przydatna funkcja dotycząca klawiatury dotyczy sprawdzenia, czy cokolwiek znajduje się w buforze (czy naciśnięto jakiś znak). Jeśli jest coś w buforze to funkcja jest prawdziwa.

```
if keypressed then
begin
  znak:=readkey;
  if znak=#0 then znak:=readkey;
  case znak of
    'H': y:=yp-5;          {kod kursora gora}
    'P': y:=yp+5;          {kod kursora dol}
    'M': x:=xp+5;          {kod kursora w lewo}
    'K': x:=xp-5;          {kod kursora w prawo}
    #27: koniec:=true;     {kod klawisza ESC}
  end;
  while keypressed do znak:=readkey; {czyść bufor klawiatury zeby nie puszczalo}
end;
```

Jeżeli w naciśnięto jakiś klawisz (KEYPRESSED), to odczytujemy jego znak (ZNAK:=READKEY)

Jeśli to był klawisz specjalny o kodzie zerowym (ZNAK = #0), np. kursor, to czytamy kolejny znak.

W zależności od wciśniętego klawisza wykonujemy instrukcje (CASE ... OF ... END;)

Kursor w górę wstawia do bufora klawiatury znak H - należy przesunąć piłkę w górę o 5 pikseli. W podobny sposób obsługujemy inne klawisze.

Klawisze o kodach specjalnych, jak na przykład ESC nie mają swoich literowych odpowiedników, dlatego oznaczenie #27. Gdy naciśniemy klawisz ESC, to zmienna KONIEC przyjmuje wartość PRAWDA i pętla główna programu się zakończy.

Bezpiecznie jest umieścić w programie instrukcję pętli WHILE KEYPRESSED DO ZNAK:=READKEY; która powoduje wyczyszczenie bufora klawiatury i przygotowanie na kolejne operacje.

Zmiana wymiarów piłki

Kwadratową piłkę możemy przesuwać po ekranie. Dołożymy zmianę wymiarów piłki. Np. po naciśnięciu klawisza S (mała litera s) piłka będzie się wydłużać w prawo i jednocześnie zmniejszać swoją wysokość. Podobne działanie będzie miał klawisz A - skraca na szerokości a wydłuża na wysokości. Cały program ANIM05.PAS

Zwiększenie szerokości piłki:

```
's': dx := dx+1;
```

Ale skoro jednocześnie chcemy również zmniejszać wysokość to:

```
's': begin dx := dx+1;      dy := dy-1 end;
```

Nie możemy jednak zmniejszać wysokości (DY), bo wartości staną się mniejsze od 0 - należy to sprawdzać i nie pozwolić na dalszą zmianę

```
's': if dy > 2 then begin dx := dx+1; dy := dy-1 end;
```

W podobny sposób opisujemy działanie klawisza A

Po niewielkich modyfikacjach otrzymamy grę komputerową LAPIRYNT.PAS

Gra komputerowa PINKPONG - ANIM06.PAS

Wykorzystujemy program GRAF14.PAS - kwadratowa piłka odbija się od brzegów ekranu

Procedura RAKIETKA - prostokąt symulujący raketkę tenisową - tutaj tylko początek.

```
procedure raketka(x,y,dx,dy,kolor:integer);
```

Raketką będziemy poruszać kursorami GÓRA-DÓŁ, dlatego potrzebna nam będzie dodatkowa zmienna RY - położenie raketki na ekranie.

Rakieta będzie początkowo wyświetlana z prawej strony ekranu - 600, w pozycji Ry - ustalamy na 200. Będzie miała szerokość 5, wysokość 120 i kolor żółty.

```
raketka(600,Ry,5,120,14);
```

W odpowiednich miejscach pętli wyświetlamy i chowamy raketkę - podobnie jak prostokątną piłkę

Podobnie, jak w programie ANIM05.PAS wykonujemy obsługę kursorów GÓRA-DÓŁ

```
'H': Ry:=Ry-5;
```

Jeżeli zmienimy dołożymy warunek, to raketka nie będzie mogła wyjechać poza ekran w górę. W podobny sposób można opisać ruch w dół raketki

```
'H': if Ry>5 then Ry:=Ry-5;
```

```
'P': if Ry>480-5-120 then Ry:=Ry+5;
```

Jak sprawdzić czy piłka opuści planszę z prawej strony? Wystarczy, że współrzędna x piłki będzie większa niż 640. Pisząc warunek musimy jednocześnie zlikwidować odbicie od prawego przegu.

```
if x>640 then koniec:=true;
```

```
if x<0 then dx:=-dx;
```

A teraz sprawdzamy, czy piłka odbije się od raketki. Najbardziej skrajne położenia piłki i raketki opisuje rysunek

Położenie rakiетки - najbardziej skrajne punkty:

lewy górny róg R1 - (600,Ry)

lewy dolny róg R2 - (600,Ry+120) - długość rakiетки 120

Położenie piłki (x,y) - najbardziej skrajne punkty, przy których nastąpi obicie

prawy dolny róg P1 - (x+bok,y+bok)

prawy górny róg P2 - (x+bok,y)

Będziemy sprawdzać tylko te skrajne, a wszystkie pośrednie będą oznaczały, że piłka się odbije

Żeby piłka się odbiła muszą być spełnione następujące warunki:

I - prawy brzeg piłki musi być równy 600, bezpieczniej większe niż lub równe

$x+bok \geq 600$

II - dół piłki musi być większy niż góra rakiетки (róg nas nie interesuje, dlatego

$y+bok > Ry$

III - góra piłki musi być mniejsza niż dół rakiетки (róg nas nie interesuje, dlatego

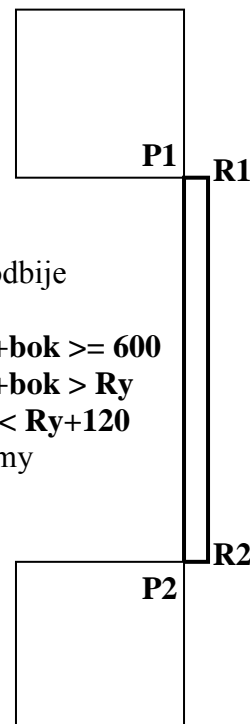
$y < Ry+120$

Jednocześnie muszą zachodzić warunki I i III oraz I i II żeby nastąpiło odbicie, co zapiszemy

```
if (x+bok >= 600) and  
    ((y+bok > Ry) and (y < Ry+120)) then dx:=-dx;
```

Po niewielkich modyfikacjach programu otrzymamy najprostszą grę.

Poruszająca się piłka, którą odbijamy od rakiетки sterowanej kursorami.



Ćwiczenia

LABIRYNT

- po wciśnięciu 'k' obiekt zmienia swój kolor, cyklicznie od 1 do 15,... z 15 na 1 (w koło)
- po wciśnięciu 'l' obiekt zmienia swój kolor, cyklicznie, ale wstecz, z 15 na 14, ... z 1 na 15, w koło
- po wciśnięciu 'A' obiekt przesunie się (płynna animacja) do lewego brzegu ekranu
- 'S' - prawego brzegu
- 'Z' - dolnego brzegu
- 'W' - górnego brzegu

PINKPONG

- po każdym odbiciu prędkość piłki (zmiennne DX i DY) zwiększa się o 1
- po odbiciu od górnego i dolnego brzegu zwiększa się DY
- po odbiciu od lewego i prawego brzegu zwiększa się DX
- po każdym odbiciu zmienia się losowo kolor piłki
- po 10 odbiciach od rakiетки program sam kończy swoje działanie