Tehnici de animaţie-Pascal

Ne propunem să răspundem la următoarea întrebare: cum facem o figură să se mişte pe ecran? Pentru rezolvarea acestei probleme există mai multe tehnici care vor fi prezentate în continuare, dar toate au acelaşi principiu de funcţionare şi anume:

- se desenează figura:
- se aşteaptă o perioadă de timp;
- se şterge figura şi se deplasează într-o altă poziţie.

Pentru ca figura să pară că se mişcă în mod real, un rol important îl are timpul de aşteptare (timpul în care figura rămâne pe ecran până a se şterge). Acesta se reglează de la caz la caz cu ajutorul procedurii Delay.

În principal, există trei tehnici de realizare a animaţiei. Ele diferă prin procedurile pe care le utilizează şi prin viteza de execuţie. Pentru exemplificarea lor vom folosi o singură problemă şi anume de a mişca un pătrat pe orizontală.

Tehnica 1

Operaţiile de desenare şi ştergere a figurii se fac cu ajutorul procedurii **SetWriteMode**, pe care o vom prezenta în continuare.

Forma generală a acestei proceduri este SetWriteMode(valoare întreagă). Vom folosi această procedură cu un singur parametru şi anume XorPut (valoarea 1 predefinită în Unit-ul Graph). În concluzie, vom pune SetWriteMode(XorPut).

După apelul acestei proceduri procedăm astfel:

- apelăm o procedură care desenează o figură (aceasta va fi vizibilă pe ecran);
- aşteptăm o perioadă de timp;
- apelăm din nou procedura care realizează desenul, exact in aceeaşi poziţie (la acest apel desenul va dispărea, pentru că, de fapt, desenul se realizează acum utilizând culoarea fondului);
- apelăm procedura care realizează desenul într-o altă poziţie (acesta va deveni vizibil, pentru că se realizează utilizând culoarea curentă);
- aşteptăm o perioadă de timp;
- procedeul se repetă până când figura a ajuns în poziţia dorită.

Tehnica 2

Această tehnică de animaţie este superioară primei tehnici, datorită faptului că este mai rapidă operaţia de aducere din memorie pe ecran a unei imagini, decât desenarea ei. Acest fapt constituie un mare avantaj în cazul imaginilor complexe.

Pentru a înţelege această tehnică, trebuie să prezentăm în prealabil câteva proceduri.

Orice imagine care se află pe ecran poate fi salvată în memoria internă. Există posibilitatea ca anumite informaţii (cum ar fi cele care permit vizualizarea unei imagini) să fie salvate în memoria internă, alocând spaţiul necesar pentru aceasta în timpul execuţiei programului. O astfel de

alocare a memoriei poartă numele de alocare dinamică şi se va studia în detaliu în clasa a zecea, dar pentru această tehnică de animaţie ne sunt necesare câteva cunoştinţe minimale.

Variabila de tip **Pointer** are posibilitatea de a reţine o adresă în memorie (a nu se face confuzie între adresa unei zone de memorie şi conţinutul ei). Alocarea spaţiului în memorie (un număr de octeţi consecutivi la o anumită adresă) se face cu ajutorul procedurii **GetMem** care creează în heap o variabilă dinamică de dimensiune specificată.

Sintaxă: Procedure GetMem (Var p : Pointer, dim : Word);

P – variabilă de tip pointer care va conţine adresa de început a zonei alocate variabilei dinamice

Dim – este o expresie care stabileşte lungimea în octeţi a zonei care va fi ocupată în heap. Valorea sa maximă este 65521.

Cum salvăm imaginea? De fapt, se salvează imaginea conţinută într-un dreptunghi pentru care se cunosc coordonatele colţurilor din stânga sus şi dreapta jos: De unde ştim câţi octeţi sunt necesari pentru a salva o imagine? Aici, ne este de mare folos funcţia **ImageSize**. Această funcţie este de tip Word şi returnează numărul de octeţi necesari pentru salvarea unei imagini în cazul când dispunem de acest spaţiu, în caz contrar se returnează valoarea 0.

Până acum ştim cum putem afla numărul de octeţi necesari memorării imaginii, cum să rezervăm spaţiul necesar memorării imaginii, dar nu ştim cum salvăm efectiv imaginea (octeţii corespunzători ei aflaţi în memoria video). Pentru a putea realiza aceasta, folosim procedura **GetImage** are forma următoare: GetImage (x_1,y_1,x_2,y_2,a^).

Parametrii au următoarea semnificaţie:

- x_1,y_1,x_2,y_2 - coordonatele colţului din stânga sus şi dreapta jos ale dreptunghiului care conţine imaginea;
- a^ - spaţiul în care se face salvarea. Referitor la ultimul parametru, facem următoarele precizări:
  - a este variabilă de tip Pointer
  - când scriem a, ne referim la adresa unde s-a făcut rezervarea);
  - când scriem a^ ne referim la conţinutul de la adresa care se găseşte în variabila a.

În concluzie, procedura GetImage are rolul de a salva o imagine în memorie. Operaţia inversă, da a reface imaginea prin aducerea octeţilor de informaţie în memoria video, se face prin utilizarea procedurii **PutImage**. Aceasta are forma generală: PutImage(x_1,y_1,a^,XorPut).

Parametrii au următoarea semnificaţie:

- x_1,y_1 - coordonatele colţului din stânga sus al dreptunghiului care conţine imaginea (nu este obligatoriu să refacem imaginea în poziţia pe care a avut-o înainte de salvare, acest fapt permite şi animaţia);
- a^ - conţinutul zonei de memorie a cărei adresă se găseşte în a;
- parametrul XorPut, prezentat şi la SetWriteMode şi care are rolul ca odată figura să fie vizibilă, odată nu (se desenează utilizând culoarea de fond).

Utilizând tehnica salvării şi refacerii imaginii, pentru a realiza animaţia se procedează astfel:

- se desenează figura într-o primă poziţie;
- se calculează numărul de octeţi necesari salvării ei în memoria internă;
- se salvează imaginea;
- se aşteaptă un interval de timp;
- se reface imaginea dar cu culoarea fondului, ceea ce duce de fapt la ştergerea ei;

- repetitiv se procedează astfel:
  - se reface imaginea într-o poziţie nouă;
  - se aşteaptă un interval de timp;
  - se reface imaginea în aceeaşi poziţie dar, datorită parametrului XorPut, de fapt se şterge;
  - procedeul continuă până când s-a ajuns în poziţia finală.

Tehnica 3

Această tehnică nu este aplicabilă în general, ci numai în cazul în care se lucrează într-un mod grafic, cu un driver grafic ce permite utilizarea mai multor pagini video. O pagină video este memoria necesară (existentă fizic pe placa grafică) ce permite reţinerea unei imagini.

Ideea de bază este următoarea: atât timp cât este vizualizată o pagină, într-o alta se realizează desenul în noua poziţie. Prin aceasta se câştigă timp.

Un exemplu în care putem folosi această tehnică este atunci când folosim driverul VGA si lucrăm în modul VGALo, caz in care dispunem de 4 pagini video. Apar două noţiuni noi şi anume pagina vizualizată şi pagina activă.

Pagina vizualizată este aceea care se vede pe ecran, iar cea activă este cea în care acţionează procedurile grafice. Dacă avem o singură pagină video, cele două naţiuni coincid. În cazul în care se lucrează cu mai multe pagini video, una poate fi vizualizată şi alta activă. Stabilirea paginii care se vizualizează se face cu ajutorul procedurii **SetVisualPage**(nr), unde nr reprezintă numărul paginii. Facem precizarea că paginile se numerotează începând cu 0 (astfel, dacă dispunem de 4 pagini video, acestea sunt numerotate între 0 şi 3). Stabilirea paginii active se face cu procedura **SetActivePage**(nr), unde parametrul nr este de tip Integer şi are semnificaţia de număr al paginii.