

УДК 004.4'274

Трофименко О.Г.,
к.т.н., ст.викладач,
Дубовой В.В.,
інженер,
Лалим В.А.,
студент

Черкаський державний
технологічний університет

ОРГАНІЗАЦІЯ «ШВИДКОГО» ВИВЕДЕННЯ ГРАФІКИ ЗАСОБАМИ МОВИ TURBO PASCAL 7.0

В статье рассмотрены особенности формирования и вывода графических изображений для различных видеорежимов. Предложены алгоритмы, позволяющие ускорить вывод графики средствами Turbo Pascal 7.0.

Investigation is given to the peculiarities of the formation and graphic images output for different videomodes. Algorithms are suggested allowing to speed up an output of the graphics by means of Turbo Pascal 7.0.

Робота по формуванню та виведенню графічних зображень на екран, використовуючи різні відеорежими, є майже невід'ємною частиною процесу написання сучасних програмних продуктів. Розробки, направлені на організацію „швидкого” виведення графіки з метою отримання якісної графічної „картинки”, об'єкти якої динамічно змінюються, представляють великий практичний інтерес як для розробників програмних графічних модулів, так і для користувачів цих програмних продуктів.

При створенні авторами статі графічного редактора 3Dedit в середовищі Turbo Pascal 7.0, робота якого буде описана далі, виникла потреба в алгоритмах швидкого виведення графіки на екран, але, на жаль, робота з динамічною графікою не є сильною стороною Pascal.

Слід сказати, що для виведення графіки для відображення відеопам'яті на основний адресний простір відеорежими VGA використовують область, розташовану в пам'яті, починаючи з адреси 0A000h:0000h. Збільшення відеорозрізнення або числа кольорів приводить до того, що об'єм відеопам'яті перевищує максимальні границі сегменту, а потім і розмір ділянки адресного простору, яка відводиться для відеопам'яті (від 0A0000h:0000h до 0B800h:0FFFFh. З адреси 0C800h:0000h вже починається область ROM BIOS). Щоб вивести зображення, використовуються два режими – переключення банків відеопам'яті для реального режиму і LFB (лінійний кадровий буфер) для захищеного [1].

В реальному режимі виведення на екран здійснюється копіюванням даних в 64-

кілобайтний сегмент, який починається с адреси 0A0000h:0000h, але ця область пам'яті відповідає тільки частині екрана. Щоб вивести зображення в наступну частину екрана, потрібно викликати функцію переміщення вікна (або, що є тим самим, здійснити переключення банку відеопам'яті), яка змінює область відеопам'яті. Наприклад, в режимі 640x480x256 потрібно 307 200 байт для збереження всього відеозображення. Заповнення сегмента області відеопам'яті приводить до зафарбування приблизно 1/5 частину екрана, переміщення вікна на позицію 1 (переключення на банк 1) та повторне заповнення цієї ж області – до зафарбовування наступної 1/5 екрана, тощо.

Стандарт VESA хоча і легалізував список відеорежимів, але не позбавив від розмаїття способів доступу до відеопам'яті і нічого не говорить про регістри, які потрібно програмувати для цього доступу. В більшості випадків пам'ять виявляється організована таким чином, коли кожній точці відповідає своя адреса відеопам'яті. Переключають банки пам'яті за допомогою регістрів, номери і способи програмування яких унікальні для кожного типу SVGA-плат. Пропонований же стандартом VESA спосіб переключення банків через переривання дуже повільний і, на жаль, це є одним з недоліків виведення графічних модулів засобами мови Turbo Pascal 7.0.

Розглянемо можливість швидкого (без використання BIOS) и апаратно-незалежного програмування в режимі 640x400x256 для любих SVGA (и багатьох VGA)-плат.

На щастя, можна так перепрограмувати

відеоадаптер, що доступ до всіх 256 Кбайт відеопам'яті буде здійснюватися через адресний простір розміром 64 Кбайта. При цьому як у режимі перепрограмування, так і в режимі доступу можна обійтися лише стандартними VGA-регістрами. Таким чином, проблема несумісності відеоадаптерів знімається. Потрібно розділити відеопам'ять на чотири площини аналогічно тому, як це робиться в 16-кольорових режимах. Цей спосіб описаний у літературі [2, 3] і використовується розроблявачами комп'ютерних ігор, як правило, для організації декількох сторінок відеопам'яті [4]. При цьому по одній адресі знаходяться чотири точки, а потрібна з них вибирається за допомогою регістра маски. Якщо ж потрібно швидко зафарбувати частину екрана (або екран цілком), дозволяється запис в усі чотири площини, і за одне звертання до відеопам'яті "фарбуються" чотири точки.

Як правило, обсяг відеопам'яті потрібен для того, щоб збільшити розрядність числа, яке зберігає інформацію про колір (глибину кольору), до 16 або навіть 24 розрядів (близько 65 тис. і 16 млн. кольорових відтінків відповідно), а не для збільшення просторового розрізнення. Проблема в тому, що вихід за межі перших 256 Кбайт відеопам'яті не зовсім безболісний, - для її адресації в цьому випадку приходится використовувати сегмент V800h. А це призводить до включення додаткового блоку перевірок заповнення першої області та переходу до наступної при організації виведення кожного пікселу, що також впливає на сповільнення формування всього зображення, особливо, якщо потрібна його динаміка. Тим більше Turbo Pascal 7.0 ці проблеми хоч і розглядає, та результативність виведення графіки замала.

При виникненні задач потребуючих швидкого виведення графіки на екран широко поширеним є використання відеорежимів 640x400x256, 640x480x256 [5] та режиму 800x600x256 [6].

Перед тим як зробити свій вибір та надати перевагу якомусь з трьох запропонованих до розгляду відеорежимів, згадаємо величину, „aspect ratio”, яка представляє собою відношення розміру точки растра по горизонталі до її розміру по вертикалі. Для різних відеорежимів це відношення може змінюватися в досить широких межах (табл. 1).

Таблиця 1

„Aspect ratio” для різних відеорежимів

Відеорежим	Розрізнення	„Aspect ratio”
а) стандартні режими:		
Текстовий	40x25	5/6 = 0,83
Текстовий	80x25	5/12 = 0,42
Текстовий	80x25	40/54 = 0,74
Графічний	320x200	5/6 = 0,83
Графічний	640x400	5/6 = 0,83
Графічний	640x480	1/1 = 1
Графічний	800x600	1/1 = 1
Графічний	1024x768	1/1 = 1
б) нестандартні режими:		
Графічний	320x240	1/1 = 1
Графічний	320x400	10/6 = 1,67
Графічний	360x480	16/9 = 1,78
Графічний	720x350	35/54 = 0,65

При аналізі співвідношення „aspect ratio” варто враховувати дві обставини: по-перше, особливістю людського сприйняття є більш висока вимогливість до розрізнення по горизонталі, чим по вертикалі, і, по-друге, для деяких додатків значно більш зручним або просто необхідним є рівність розрізнення по вертикалі і по горизонталі, особливо там, де потрібна більш точна відповідність зображення на екрані твердої копії, або тоді, коли необхідно обертати зображення в площині екрана, наприклад, в інженерній графіці, у видавничій справі, для імітаторів літаків, тощо.

На ранніх етапах розвитку комп'ютеризації, коли оптимізувалось використання кожного кілобайту відеопам'яті (як з точки зору вартості відеоадаптера, так і з точки зору швидкості формування зображення), це співвідношення було менше одиниці, а інколи навіть менше 1/2.

Є обгрунтоване припущення [7], що оптимальним є співвідношення, величина якого дорівнює золотому перетину (приблизно 0,62), що близько нестандартному режиму 720x350. Але при цьому довелося би проводити глибоке перепрограмування відеоадаптера, тому краще зупинитися на режимі 640x400.

Всі сучасні режими високого розрізнення мають „aspect ratio” рівне 1, що пов'язано, головним чином, з особливостями Windows, а саме з відсутністю можливості перевизначати відеорежим, підбираючи оптимальний для кожного додатку. При цьому оптимальність приноситься в жертву універсальності, та й сама зада-

ча підбору в значній мірі втратила актуальність через стрімке зростання потужностей ПК.

Однак, не звертаючи на панування Windows практично у всіх областях застосування персональних комп'ютерів, розробники ігор як колись віддають перевагу DOS-подібній організації відеопам'яті, яка дає можливість керувати вибором розрізнення екрана.

Нестандартні режими 320x240, 320x400 і 360x480 (див. табл. 1) можуть викликати, скоріш за все, лише академічний інтерес, хоча, якщо як слід замислитися, то можна, мабуть, підібрати такі додатки, для яких ці режими виявляться оптимальними.

Режим 640x400x256 порівняно до 640x480x256 має наступні переваги:

- потрібно менше відеопам'яті (256 Кбайт, а не 512 Кбайт);
- здійснюється на 20% менше розрахунків при практично тій же кількості;
- відсутня необхідність в відстеженні міжсегментного переходу при 16-разрядному коді, що ще більше підвищує продуктивність;
- можна не використовувати відеосегмент V800h, що знижує вимоги до умов успішної роботи програми [2];
- дозволяє прискорити виведення відео-

графіки і позбавитись мерехтіння за рахунок організації двох сторінок відеопам'яті (при використанні відеосегмента V800h), а не одну, як при розрізненні 640x480;

- забезпечує частоту регенерації екрана до 90 Гц, тоді як при 640x480 частота може складати тільки 60 Гц.

Переваги цього відеорежиму були використані при створенні авторами статті графічного редактора 3Dedit в середовищі Turbo Pascal 7.0. Цей редактор дозволяє користувачу створювати кольорові тривимірні фігури за допомогою проєкційних площин, змінювати текстуру їх поверхонь, а, головне, дає змогу проглядати в динаміці заново створені, або існуючі в базі, геометричні об'єкти з різних позицій спостереження, змінюючи відстань до них і кут зору в тривимірній площині. На рисунку 1 представлений графічний інтерфейс цього редактора.

Значну складність при написанні програми викликало те, що процес формування Z-буферу, який відповідає за нашарування площин трьохмірних об'єктів, помітно сповільнює промальовування екрана при кожній зміні позиції спостерігача.

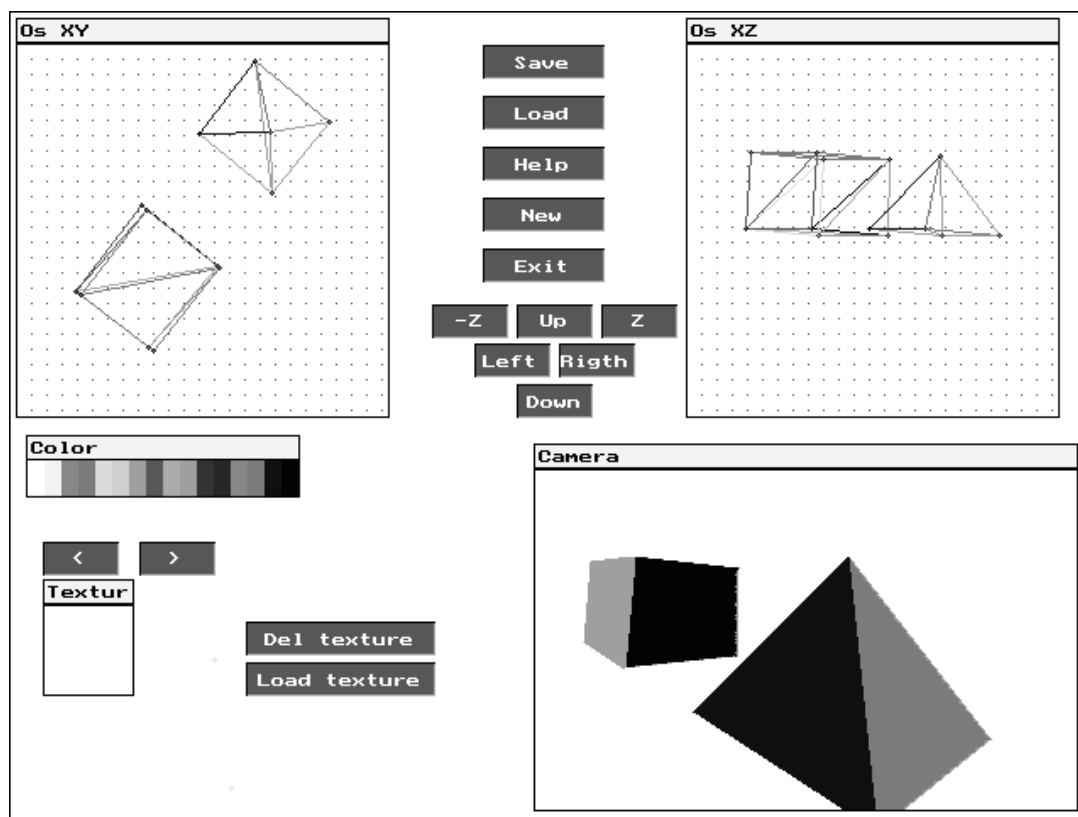


Рис. 1. Приклад графічного інтерфейсу редактору 3Dedit

Ще одною особливістю при написанні програми була проблема перевірок виходу при необхідності за межі області 0A0000h:0000h - 0A0000h:0FFFFh, вже описана вище.

Шляхами розв'язання труднощів, які виникали, було використання в Turbo Pascal 7.0 асемблерних вставок і програмування на рівні портів контролера.

Для наглядної ілюстрації була розроблена тест-програма, яка демонструє переваги роботи запропонованих алгоритмів. Час десятикратного заповнення екрану за допомогою різних способів, виражений в одиницях відліку системного таймера, приведений в таблиці 2 для двох апаратних конфігурацій комп'ютера в реальному і захищеному режимах роботи процесора.

Таблиця 2

Час десятикратного заповнення екрана

Конфігурація / Режим	Стандартний модуль Graph (Borland)	Стандартні процедури Pascal	Процедурні асемблерні вставки	Вбудовані асемблерні вставки
486-100 / Реальний	257	154	112	46
486-100 / Захищений	426	294	264	51
Pentium-100 / Реальний	163	108	54	31
Pentium-100 / Захищений	255	183	136	31

На початку були отримані результати при використанні стандартного модуля Graph, який пропонується фірмою Borland. В порівнянні з ними в таблиці представлені показники при застосуванні стандартних Pascal-процедур роботи з пам'яттю на рівні портів. Добрі результати дало використання вбудованих в Pascal асемблерних вставок у якості процедур. Але найкращі результати, тобто найбільшого зростання швидкості виведення графіки, вдається досягти тільки при відмовленні від виклику процедур і розташування операторів виведення безпосередньо в тілі програми.

Запропоновані алгоритми в даний час можуть бути застосовані і для роботи з одноплатними контролерами, що вбудовуються. Такі контролери знайшли широке використання в різноманітній діагностичній апаратурі (в медицині, в автосервісі, тощо), для вбудовування в станки з ЧПУ, а також в різних виробництвах та галузях промисловості, де повноцінний ПК використовувати не раціонально і не зручно. Як правило, ці ПК працюють під управлінням DOS-подібних операційних систем, і для них потреба в алгоритмах швидкого виведення графіки є вельми актуальною.

Література:

1. Зубков С.В. Assembler для DOS, Windows и UNIX. – М.: ДМК Пресс, 2000. – 608 с.
2. Хошенко А. Видеорежим будущей игры. "Мир ПК", #7-8/96.
3. Фролов А.В., Фролов Г.В. Программирование видеоадаптеров CGA, EGA и VGA. М.: Диалог-МИФИ, 1992.4.
4. Телесин Б. Адаптер VGA. Режим 256 цветов. "Монитор", #1/93.

5. Андрианов С.А. Как программировать SVGA без головной боли // Мир ПК. #5/ 19976.
6. Андрианов С.А. Видеоадаптер: как выйти за предел 256 Кбайт // Мир ПК. # 9/19977.
7. Андрианов С.А. SVGA: быстрый вывод на экран // Мир ПК, #11/1997

Рецензент д.т.н., проф. Первунінський С.М.