

В.Ю. Коцюбинський
В.В. Кабачій

КОМП'ЮТЕРНА ГРАФІКА

Adobe Photoshop GIF Animator

Web-design

CorelDraw

HTML

FrontPage Express

TurboPascal

Internet

В.Ю. Коцюбинський, В.В. Кабачій

КОМП'ЮТЕРНА ГРАФІКА

Затверджено Ученою радою Вінницького державного технічного університету як навчальний посібник для студентів всіх форм навчання спеціальності 7.091401 – “Системи управління і автоматики”.
Протокол № 13 від 4 липня 2002 р.

Рецензенти:

О.Д. Азаров, доктор технічних наук, професор
В.О. Поджаренко, доктор технічних наук, професор
І.Й. Хаймзон, доктор технічних наук, професор

Рекомендовано до видання Ученою радою Вінницького державного технічного університету Міністерства освіти і науки України

Коцюбинський В.Ю., Кабачій В.В.

К75 Комп'ютерна графіка. Навчальний посібник. – Вінниця: ВДТУ, 2003. – 88 с.

Навчальний посібник присвячений вивченню теоретичних основ комп'ютерної графіки, принципів створення і оброблення різноманітних графічних зображень з використанням комп'ютерної техніки та сучасного програмного забезпечення, розглянуті питання оброблення та зберігання графічної інформації, програмування способів подання об'єктів з використанням апарата перетворень та проекції зображень. Розглянуті основні принципи подання інформації в мережі Інтернет та створення анімаційних зображень.

Посібник призначений для студентів спеціальностей "Комп'ютеризовані системи управління і автоматики", "Системи управління і автоматики, інформаційні технології в менеджменті та бізнесі", "Системи управління і автоматики, інформаційні технології в менеджменті (з поглибленим вивченням іноземної мови на рівні технічного перекладача)", а також буде корисним інженерам та науковим співробітникам у галузі комп'ютеризованих систем управління та студентам суміжних спеціальностей.

Вступ	5
1 Комп'ютерна графіка: основні поняття	6
1.1 Растрова і векторна графіка	6
1.2 Роздільна здатність	8
1.3 Моделі кольорів	9
1.4 Масштабування зображень	11
1.5 Методи стиснення зображень	12
2 Формати файлів і стиснення	15
2.1 Файл в оригінальному або стандартному форматі	15
2.2 Опції стиснення без втрат	16
2.3 Стиснення із втратами	17
2.4 Структура файлів зображень типу BMP і TIFF	18
2.5 Формати файлів для різних застосувань	21
3 Програмування графіки в середовищі Turbo Pascal	24
3.1 Апаратна і програмна підтримка графіки	24
3.2 Ініціалізація графіки	25
3.3 Використання базових процедур і функцій	27
3.4 Робота з кольором	30
3.5 Робота з текстом	31
3.6 Процедури для побудови геометричних фігур	32
3.7 Маніпулювання фрагментами зображень	35
4 Афінні перетворення на площині	36
4.1 Однорідні координати точки	40
4.2 Афінні перетворення у просторі	44
4.3 Проекції	50
4.4 Текстура зображення	54
5 Графіка в Internet	56
5.1 Основні поняття мережі Internet	56
5.2 Можливості програми Internet Explorer	57
5.3 Перегляд Web-сторінок в Internet	58
5.4 Гіпертекстові посилання	59
5.5 Графіка і мультимедіа на Web-сторінках	60
5.6 Застосування допоміжних і вбудованих додатків	60
5.7 Керування відображенням Web-сторінки	61
5.8 Папка обраних адрес URL	62

6 Графіка для Web. Створення Web-сторінок	63
6.1 Редактори HTML	63
6.2 Інтерфейс програми FrontPage Express	64
6.3 Форматування документа HTML	64
6.4 Введення гіперпосилань	65
6.5 Робота з текстом	66
6.6 Створення таблиць і форм	69
6.7 Елементи графічного дизайну	70
6.8 Створення зображень	70
6.9 Майстри і шаблони	73
6.10 Використання готових матеріалів	74
6.11 Зміна властивостей тегів HTML	74
6.12 Правка коду HTML	75
6.13 Публікація Web-сторінки	76
7 Види анімації	78
7.1 Теорія анімації	78
7.2 Анімація об'єкта	78
7.3 Покадрова анімація. Photolmpact GIF Animator. Macromedia Flash	79
7.4 Інтерполяційна анімація	81
Література	85

Вступ

Стрімкі темпи розвитку комп'ютерних технологій забезпечують реалізацію потужних обчислюваних можливостей, які до недавнього часу були недоступні для інженерів та інших спеціалістів, що займаються створенням і обробленням сучасної графічної інформації. Особливості розв'язання таких задач полягають у необхідності використання ЕОМ, що дозволяють оперувати великими масивами графічних даних в критичних до витрат часу умовах. З іншого боку запорукою успішного розв'язання практично усіх задач комп'ютерної графіки є висока кваліфікація спеціаліста як у теоретичних питаннях, так і у практичних навичках при використанні сучасних програмних продуктів. При створенні даного посібника автори намагалися висвітлити основні принципи створення і оброблення різноманітних графічних зображень з використанням комп'ютерної техніки та сучасного програмного забезпечення.

У посібнику розглянуті основні поняття комп'ютерної графіки, способи збереження та стискання графічних файлів, основи апаратного та програмного забезпечення для роботи з графікою, математичний апарат перетворень та проєкцій зображень, сучасні принципи подання графіки в Інтернеті та основні положення принципів анімації.

При підготовці посібника особлива увага приділялася орієнтації читача на останні досягнення в галузі комп'ютерних технологій, пов'язаних з обробленням графічної інформації. Саме тому більшість практичних прикладів наводиться з використанням сучасних версій програмних продуктів відомих світових виробників.

Матеріал, викладений у посібнику, може бути використаний при вивченні курсів "Комп'ютерна графіка", "Інженерна та комп'ютерна графіка", а також буде корисний широкому колу фахівців та науковців у галузі інформаційної техніки.

1 Комп'ютерна графіка: основні поняття

Комп'ютерна графіка являє собою одну із сучасних технологій створення різноманітних зображень за допомогою апаратних і програмних засобів комп'ютера, відображення їх на екрані монітора або друк на принтері і потім зберігання у файлі. У розділі розглядаються поняття подання графічних зображень, роздільної здатності пристроїв відображення, моделей кольорів і масштабування зображень.

Існує два засоби подання графічних зображень: растровий і векторний. Відповідно розрізняють растровий і векторний формати графічних файлів, що містять інформацію графічного зображення. Растрові формати добре підходять для зображень із складними гаммами кольорів, відтінків і форм. Це такі зображення, як фотографії, малюнки, відскановані дані. Векторні формати добре застосовуються для креслень і зображень із простими формами, тінями і фарбуванням.

1.1 Растрова і векторна графіка

Найбільш просто реалізувати растрове подання зображення. Растр, або растровий масив (bitmap), подає сукупність бітів, розташованих на сітчастому полі-канві. Біт може бути включений (одиничний стан) або виключений (нульовий стан). Стани бітів можна використовувати для подання чорного або білого кольорів, так що, з'єднавши на канві декілька бітів, можна створити зображення з чорних і білих точок.

Растрове зображення нагадує лист паперу в клітинку, на якому кожна клітинка зафарбована чорним або білим кольором, у сукупності формуючи малюнок.

Основним елементом растрового зображення є піксел (pixel). Під цим терміном часто розуміють декілька різних понять: окремий елемент растрового зображення, окрема точка на екрані монітора, окрема точка на зображенні надрукованому принтером. Тому на практиці ці поняття часто позначають так:

- піксел – окремий елемент растрового зображення;
- видеопіксел – елемент зображення на екрані монітора;
- точка – окрема точка, створювана принтером або фотонабірним автоматом.

Колір кожного пікселя растрового зображення – чорний, білий, сірий або будь-який із спектра – запам'ятовується за допомогою комбінації бітів. Чим більше бітів використовується для цього, тим більшу кількість відтінків кольорів для кожного пікселя можна одержати. Число бітів, використовуваних комп'ютером для збереження інформації про кожний піксел, називається бітовою глибиною або глибиною кольору.

Найбільш простий тип растрового зображення складається з пікселів, що мають два основних кольори – чорний і білий. Для збереження такого типу пікселів потрібно один біт у пам'яті комп'ютера, тому зображення, що складаються з пікселів такого виду, називаються одно-бітовими зображеннями. Для відображення більшої кількості кольорів використовується більше бітів інформації. Число можливих і доступних кольорів або градацій сірого кольору кожного пікселя дорівнює двом у степені, рівної кількості бітів, що відводяться для кожного пікселя. 24 біта забезпечують більше 16 мільйонів кольорів. Про 24-бітові зображення часто говорять як про зображення з природними кольорами, тому що такої кількості кольорів більш ніж достатньо, щоб відобразити всі можливі кольори, що спроможне розрізнити людське око.

Основний недолік растрової графіки полягає в тому, що кожне зображення для свого збереження потребує велику кількість пам'яті. Прості растрові картинки як копії, екрана комп'ютера або чорно-білі зображення, займають до декількох сотень кілобайтів пам'яті. Деталізовані високоякісні малюнки, наприклад, зроблені за допомогою сканерів із високою роздільною здатністю займають уже десятки мегабайтів. Для вирішення проблеми опрацювання об'ємних (у сенсі витрат пам'яті) зображень використовуються два основних засоби:

- збільшення пам'яті комп'ютера;
- стискання зображень.

Іншим недоліком растрового подання зображень є зниження якості зображень при масштабуванні.

Векторне подання, на відміну від растрової графіки, визначає описання зображення у вигляді ліній і фігур, можливо, із зафарбованими областями, що заповнюються суцільним або градієнтним кольором. Хоча це може здаватися більш складним, ніж використання растрових масивів, але для багатьох видів зображень використання математичних описань є більш простим засобом.

У векторній графіці для описання об'єктів використовуються комбінації комп'ютерних команд і математичних формул. Це дозволяє різним пристроям комп'ютера, таким як монітор і принтер, при рисуванні цих об'єктів обчислювати, де необхідно поміщати реальні точки.

Векторну графіку часто називають об'єктно-орієнтованою або креслярською графікою. Є ряд найпростіших об'єктів, або примітивів, наприклад: еліпс, прямокутник, лінія. Ці примітиви і їхні комбінації використовуються для створення більш складних зображень. Якщо подивитися зміст файла векторної графіки, виявляється подібність із програмою. Він може містити команди, схожі на слова, і дані в коді ASCII, тому векторний файл можна відредагувати за допомогою текстового редактора. Приведемо в умовному спрощеному вигляді команди, що

описують коло: об'єкт – коло; центр – 50, 70; радіус – 40; лінія: колір – чорний, товщина – 0.50; заливка – немає.

Даний приклад показує основну перевагу векторної графіки – описання об'єкта є простим і займає мало пам'яті. Для описання цього ж кола засобами растрової графіки треба було б запам'ятати кожну окрему точку зображення, що зайняло б набагато більше пам'яті.

Крім того, векторна графіка в порівнянні з растровою має такі переваги:

- простота масштабування зображення без погіршення його якості;
- незалежність обсягу пам'яті, необхідної для збереження зображення, від обраної колірної моделі.

Недоліком векторних зображень є їх деяка штучність, що полягає в тому, що будь-яке зображення необхідно розбити на кінцеву множину складових його примітивів.

Растрова і векторна графіка існують не відокремлено одна від одної. Так, векторні рисунки можуть містити в собі і растрові зображення. Крім того, векторні і растрові зображення можуть бути перетворені один в одного – у цьому випадку говорять про конвертацію графічних файлів в інші формати. Достатньо просто виконується перетворення векторних зображень у растрові. Не завжди можливо здійснити перетворення растрової графіки у векторну, тому що для цього растрова картинка повинна містити лінії, що можуть бути ідентифіковані програмою конвертації (типу CorelTrace у складі пакета CorelDraw) як векторні примітиви. Це стосується, наприклад, високоякісних фотографій, коли кожний піксел відрізняється від сусідніх.

1.2 Роздільна здатність

Роздільна здатність – це кількість елементів у заданій області. Цей термін можна застосовувати до багатьох понять, наприклад, таких як:

- роздільна здатність графічного зображення;
- роздільна здатність принтера, як пристрою виведення;
- роздільна здатність миші як пристрою введення.

Наприклад, роздільна здатність лазерного принтера може бути задана 300 dpi (dot per inch – точок на дюйм), це означає, що спроможність принтера дозволяє надрукувати на відрізу в один дюйм 300 окремих точок. У цьому випадку елементами зображення є лазерні точки, а розмір зображення вимірюється в дюймах.

Роздільна здатність графічного зображення вимірюється в пікселях на дюйм. Відзначимо, що піксел у комп'ютерному файлі не має визначеного розміру, тому що зберігає лише інформацію про свій колір. Фізичний розмір піксел одержує при відображенні на конкретному пристрої виведення, наприклад, на моніторі або принтері. Роздільна

здатність технічних пристроїв по-різному впливає на виведення векторної і растрової графіки.

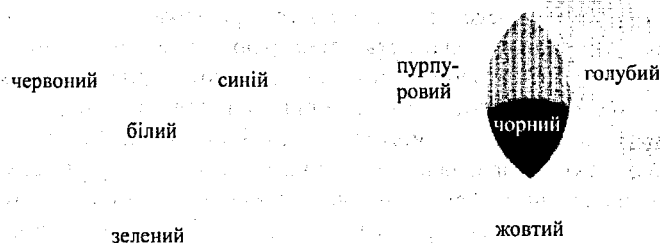
Так, при виведенні векторного рисунка використовується максимальна роздільна здатність пристрою виведення. При цьому команди, що описують зображення, повідомляють пристрою виведення положення і розміри якогось об'єкта, а пристрій для його прорисовування використовує максимально можливу кількість точок. Таким чином, векторний об'єкт, наприклад, коло, роздруковане на принтерах різної якості, має на аркуші паперу однакові положення і розміри. Проте більш гладко коло виглядає якщо друкувати на принтері з більшою роздільною здатністю, тому що складається з більшої кількості точок принтера.

Значно більший вплив роздільна здатність пристрою виведення робить на виведення растрового рисунка. Якщо у файлі растрового зображення не визначено, скільки пікселів на дюйм повинен створювати пристрій виведення, то за замовчуванням для кожного пікселя використовується мінімальний розмір. У випадку лазерного принтера мінімальним елементом слугує лазерна точка, в моніторі – відеопіксел. Оскільки пристрої виведення відрізняються розмірами мінімального елемента, що може бути ними створений, то розмір растрового зображення при виведенні на різних пристроях також буде неоднаковий.

1.3 Моделі кольорів

Деякі предмети видимі тому, що випромінюють світло, а інші – тому, що його відбивають. Коли предмети випромінюють світло, вони створюють у нашому сприйнятті те, що бачить око людини. Коли предмети відбивають світло, то їхній колір визначається кольором світла, що падає на них і кольором, що ці об'єкти відбивають. Випромінюване світло виходить з активного джерела, наприклад, екрана монітора. Відбите світло відбивається від поверхні об'єкта, наприклад, аркуша паперу.

Існують два методи описання кольору: система адитивних і субтрактивних кольорів. Система адитивних кольорів працює з випромінюваним світлом. Адитивний колір одержується при об'єднанні різнокольорових променів світла. У системі використовуються три основних кольори: червоний, зелений і синій (Red, Green, Blue – RGB). При змішуванні їх у різних пропорціях утворюється відповідний колір. Відсутність цих кольорів дає в системі чорний колір. Схематично змішання кольорів показано на рис. 1.1.



а). Адитивний колір

б). Субтрактивний колір

Рисунок 1.1 – Система змішування кольорів

У системі субтрактивних кольорів відбувається зворотний процес: деякий колір утворюється виключенням інших кольорів із загального променя світла. При цьому білий колір утворюється в результаті відсутності усіх кольорів, а присутність усіх кольорів дає чорний колір. Система субтрактивних кольорів працює з відбитим кольором, наприклад, від листа паперу. Білий папір відбиває усі кольори, пофарбований – деякі поглинає, інші відбиває.

У системі субтрактивних кольорів основними є блакитний, пурпурний і жовтий кольори (Cyan, Magenta, Yellow – CMY) – додаткові червоному, зеленому і синьому. Коли ці кольори змішують на папері в рівній пропорції, утворюється чорний колір. Цей процес проілюстрований на рис. 1. У зв'язку з тим, що друкарські фарби не цілком поглинають світло, комбінація трьох основних кольорів виглядає темно-коричневою. Тому для корегування тонів і одержання істинно чорного кольору в принтери додають трохи чорної фарби. Системи кольорів, основані на такому принципі чотирикольорового друку, позначають аббревіатурою CMYK (Cyan, Magenta, Yellow, black).

Існують і інші системи кодування кольорів, наприклад, зображення його у вигляді тону, насиченості і яскравості (Hue, Saturation, Brightness – HSB).

Тон являє собою конкретний відтінок кольору, відмінний від інших: червоний, блакитний, зелений і т.п. Насиченість характеризує відносну інтенсивність кольору.

При зменшенні, наприклад, насиченості червоного кольору, він робиться більш пастельним або блідим. Яскравість (або освітленість) кольору показує величину чорного відтінку, що добавляється до кольору, що робить його більш темним. Система HSB добре узгоджується з моделлю сприйняття кольору людиною. Тон є еквівалентом довжини хвилі

світла, насиченість – інтенсивності хвилі, а яскравість – загальною кількістю світла. недоліком цієї системи є необхідність перетворення її в інші системи: RGB – при виведенні зображення на монітор; CMYK – при виведенні на чотирикольоровий принтер.

Інше позначення системи HSB – HSL (Hue, Saturation, Light – тон, насиченість і освітленість).

Розглянуті системи працюють із усім спектром кольорів – мільйонами можливих відтінків. Проте користувачу часто достатньо не більше декількох сотень кольорів. У цьому випадку зручно використовувати індексовані палітри – набори кольорів, що містять фіксовану кількість кольорів, наприклад, 16 або 256, із яких можна вибрати необхідний колір. Перевагою таких палітр є те, що вони займають набагато менше пам'яті, ніж повні системи RGB і CMYK. При роботі з зображенням комп'ютер створює палітру і присвоює кожному кольору номер, потім, вказуючи колір окремого пікселю або об'єкту, просто запам'ятовує номер, що має даний колір у палітрі. Для запам'ятовування числа від 1 до 16 вводимо 4 біта пам'яті, а від 1 до 256 – 8 бітів, тому зображення, що мають 16 кольорів називають 4-бітовими, а 256 кольорів – 8-бітовими. При порівнянні з 24 бітами, необхідними для збереження повного кольору в системі RGB, або з 32 бітами – у системі CMYK, економія пам'яті очевидна.

При роботі з палітрою можна застосовувати будь-які кольори, наприклад, системи RGB, але обмежена їхня кількість. Так, при використанні 256-колірної палітри в процесі задання і нумерації кожний колір у палітрі описується як звичайний 24-бітовий системи RGB. А при посиланні на якийсь колір уже вказується його номер, а не конкретні дані системи RGB, що описують цей колір.

1.4 Масштабування зображень

Масштабування полягає в зміні вертикального і горизонтального розмірів зображення. Масштабування може бути пропорційним – у цьому випадку співвідношення між висотою і шириною рисунка не змінюється, а змінюється загальний розмір, і непропорційним – у цьому випадку обидва виміри змінюються по-різному. Масштабування векторних рисунків виконується просто і без втрати якості. Оскільки об'єкти векторної графіки створюються за їх описанням, то для зміни масштабу векторного об'єкта, достатньо змінити його описання. Наприклад, щоб збільшити в два рази векторний об'єкт, варто подвоїти значення, що описує його розмір.

Масштабування растрових рисунків є набагато більш складним процесом, ніж для векторної графіки, і часто супроводжується втратою якості. При зміні розмірів растрового зображення виконується одна з таких дій:

- одночасна зміна розмірів усіх пікселів (у більшу або меншу сторону);
- збільшення або зменшення пікселів із рисунка для відтворення вибраних у ньому змін, назване вибіркою пікселів у зображенні.

Найпростіший засіб зміни масштабу растрового рисунка полягає в зміні розміру всіх його пікселів. Оскільки всередині самого рисунка піксели не мають розміру й одержують його вже при виведенні на зовнішній пристрій, то зміна розміру пікселів растра дуже схожа на масштабування векторних об'єктів – необхідно змінити тільки описання піксела, а інше виконає пристрій виведення.

Пристрій виведення для створення пікселу визначеного фізичного розміру використовує стільки своїх мінімальних елементів (лазерних точок – для лазерного принтера, відеопікселів – для монітора), скільки зможе. При масштабуванні зображення кількість вхідних у нього пікселів не змінюється, а змінюється кількість утворюваних пристроєм виведення елементів, що йдуть на побудову окремого піксела зображення.

Вибірка растрового рисунка може бути зроблена двома різними способами. Першим способом просто дублюється або видаляється необхідна кількість пікселів. При цьому в результаті масштабування, як правило, погіршується якість зображення. Наприклад, при збільшенні розміру рисунка зростають його зернистість і дискретність. При зменшенні розміру рисунка втрачає в якості не настільки помітні, проте при наступному відновленні зменшеного рисунка до старого розміру знову зростають зернистість і дискретність. Це пов'язано з тим, що при зменшенні розміру рисунка частина пікселів була видалена з вихідного зображення і загублена безповоротно, а при наступному відновленні розмірів рисунка відсутні піксели дублювались із сусідніх.

Другим способом за допомогою визначених обчислень можна створити піксели іншого кольору, обумовленого кольорами початкового пікселу і його оточення. Цей метод називається інтерполяцією і є більш складним, ніж просте дублювання. При інтерполяції крім дубльованих пікселів, відбираються і сусідні з ними, за допомогою яких знову створювані піксели одержують від існуючих усереднений колір або відтінок сірого. У результаті переходу між пікселами стають більш плавними, що дозволяє прибрати або зменшити ефект «пилкоподібності» зображення.

1.5 Методи стискання зображень

Як і багато іншої інформації, графіка може бути стиснута. Це вигідно з погляду економії пам'яті комп'ютера, тому що, наприклад, високоякісні зображення, як уже зазначалося, мають розміри до декількох

десятьків мегабайтів. Для файлів графічних зображень розроблена множина схем і алгоритмів стиснення, основними з яких є такі:

- групове стиснення;
- кодування методом Хафмана;
- стиснення за схемою LZW;
- арифметичне стиснення;
- стиснення з втратами,
- перетворення кольорів RGB у кольори YUV.

В основі більшості схем стиснення лежить використання одного з таких властивостей графічних даних: надмірність (избыточность), передбачуваність (предсказуемость) і необов'язковість. Зокрема, групове кодування (RLE) ґрунтується на використанні першої властивості. Кодування за методом Хафмана й арифметичне кодування, ґрунтовані на статистичній моделі, використовують передбачуваність, пропонуючи більш короткі коди для пікселів, що часто зустрічаються. Алгоритми стиснення з втратами ґрунтуються на надмірності даних.

Варто врахувати, що алгоритм, що забезпечує великий ступінь стиснення, звичайно більш складний і тому потребує для розпакування даних більше процесорного часу. Розглянемо докладніше декілька алгоритмів стиснення.

Групове стиснення

Групове стиснення являє собою одну з найпростіших схем стиснення файлів. Суть його полягає в тому, що серія повторюваних розмірів замінюється єдиним розміром і її кількістю. На прикладі можна помітити вигоду в довжині між aabbbbbbcddeeeeeeaa і «2a7b1c3d4e3a». Даний алгоритм простий у реалізації і добре стискає графічні файли з великими однотонними областями. Групове кодування використовується в багатьох форматах растрових файлів, таких як TIFF, PCX і т.д.

Кодування методом Хафмана

Зміст методу Хафмана полягає в заміні даних більш ефективними кодами. Коротші коди використовуються для заміни розмірів, що частіше зустрічаються. Наприклад у виразі abbbcccddeeeeeeef є шість унікальних розмірів, із частотами появи: a:1, b:3, c:3, d:2, e:9, f:1. Для утворення мінімального коду використовується двійкове дерево. Алгоритм об'єднує в пари елементи, що з'являються найрідше, потім пара об'єднується в один елемент, а їхні частоти об'єднуються. Ця дія повторюється доти, поки елементи не об'єднуються в пари. У даному прикладі треба об'єднати a і f – це перша пара, а присвоюється нульова гілка, a f – 1-a. Це означає, що 0 і 1 будуть молодшими бітами кодів для a та f відповідно. Більш старші біти будуть отримані з дерева в міру його побудови.

Підсумовування частот дає в результаті 2. Тепер найнижча частота – 2, тому пара a і T об'єднується з d (яка теж має частоту 2). Вихідній парі присвоюється нульова гілка, ad – гілка 1. Таким чином, код для a закінчується на 00; для f на 01, d закінчується на 1 і буде на один біт коротіший в порівнянні з кодами для a та f .

Дерево продовжує будуватися подібним чином так, що найменш поширені розміри описуються більш довгими кодами. Дане кодування має потребу в точній статистиці, що виражається в тому, як часто кожний розмір з'являється у файлі. Отже, для роботи зі схемою Хафмана необхідні два етапи: на першому етапі створюється статистична модель, на другому кодуються дані. Слід зазначити, що компресія і декомпресія, за Хафманом, – достатньо повільний процес.

Контрольні запитання

1. Що таке комп'ютерна графіка?
2. Чим відрізняється векторна графіка від растрової?
3. Що таке роздільна здатність?
4. Що таке графічний примітив?
5. Як створити растрове зображення та здійснити зміни в ньому засобами програми Adobe Photoshop?
6. Як залежить подання графічного зображення від роздільної здатності пристрою виведення?
7. Зробіть перелік і охарактеризуйте існуючі формати графічних даних.
8. Що таке конвертація даних графічних форматів та якими технічними засобами можна її здійснити?
9. Як створити векторне зображення та здійснити зміни в ньому засобами програми CorelDRAW 7?
10. Яким чином здійснюється масштабування растрових і векторних зображень?
11. Як здійснюється змішування кольору в комп'ютері?
12. Охарактеризуйте основні схеми стиснення зображень.

2 Формати файлів і стискання

Формати файлів вигадали не тільки для зручності. Вони визначають тип інформації, що зберігається у файлі, сумісність цього файла з різноманітними програмними додатками і можливості обміну даними з іншими користувачами. Вибір найкращого формату для зберігання файла після сканування залежить від декількох чинників – програми, застосовуваної для наступного опрацювання зображення, вільного місця на диску, необхідності обміну даними з іншими користувачами і засобу кінцевого виведення даних. Часто між першим записом і остаточним виведенням файл проходить через декілька цифрових перетворень: спочатку записується в оригінальному форматі пакета обробки зображення, потім у форматі, зручному для компонування сторінки, подання в мультимедіа або для тривалого збереження.

2.1 Файл в оригінальному або стандартному форматі

Багато програм опрацювання зображень і рисування – наприклад, Adobe Photoshop, Fractal Design Painter і HSC Live Picture – дозволяють зберігати зображення в оригінальному форматі, який розпізнається тільки самими цими програмами. Оригінальні формати файлів зручні, оскільки файли мають менший розмір і є можливість зберігати складні типи даних (наприклад, у Photoshop прошарки, маски вибірки, векторні шляхи і канали). Якщо ви працюєте в подібному додатку, то часто зручніше зберігати зображення в оригінальному форматі, особливо якщо виведення зображення буде здійснюватися безпосередньо з цього додатка. Якщо потім виникне необхідність використовувати зображення для компонування сторінки, перенести його в інше середовище або передати постачальнику або клієнту, то ви завжди зможете перезаписати його у форматі, зручному для такого користувача. Зберігайте файл у стандартному форматі, що читається більшістю програмних додатків – для виведення на друк зручні TIFF або EPS, для мультимедіа або відео – PICT і BMP, а для презентацій – TIFF або PICT.

Якщо файл буде передаватися з одного додатку в інший і є можливість вибору між різними сумісними форматами, то зберігайте файл в форматі, який містить найбільш повну інформацію про зображення.

Якщо файл хоча б один раз буде переданий на іншу платформу, то використовуйте угоду про імена DOS (в імені файла не більше восьми символів плюс три символи розширення), для більш легкого визначення типу файла завжди користуйтеся трисимвольним розширенням.

Стискання

Розмір файла багато в чому визначається глибиною кольору і розрішенням при скануванні. Природно, що користувачі сканованих зображень для їхнього збереження використовують методи стискання файлів або жорсткого диска. Методи стискання файлів розділяються на дві загальні категорії: з втратами і без втрат даних. При стисканні без втрат дані упаковуються у файл без фактичного видалення інформації, так що не відбувається втрати деталей, кольору або якості виведення зображень. При стисканні з втратами користувач визначає об'єм даних, що будуть відкинуті, — чим вищий ступінь стискання, тим більш вірогідні втрати якості зображення.

2.2 Опції стискання без втрат

До широко використовуваних способів стискання без втрат відносяться утиліти стискання і стискання у форматі TIFF. Утиліти стискання упаковують файли всіх типів без зміни вихідного формату. З іншого боку, опції стискання TIFF використовуються при роботі виключно із зображеннями.

Опції стискання TIFF

У різні версії формату TIFF було вбудовано декілька стандартів стискання. Опція стискання LZW (Lempel-Ziv-Welch) підтримується багатьма поширеними графічними пакетами і має середній коефіцієнт стискання біля 2:1. Багато пакетів OCR і ділових комунікацій відрізняються наявністю опції компресії TIFF PackBits, застосовуваної тільки до файлів формату TIFF типу bitmap (чорно-білим).

Абревіатура GIF розшифровується як Graphics Interchange Format (формат для обміну графікою). Цей формат розробила CompuServe — служба комунікацій, що існувала до Internet. Для зменшення розміру файла зображення в цьому форматі використовується схема стискання без втрати якості, відповідно до якого виявляється і видаляється надлишкова інформація (метод групового кодування). Формат GIF є споконвіку 8-бітовим, що дозволяє використовувати в зображенні до 256 різноманітних кольорів. Для деяких зображень цього достатньо, але не для всіх. Формат GIF зручно використовувати для зображень, що містять багато великих областей одного кольору. Формат GIF можна використовувати і для фотографічних зображень, але для них стискання буде незначним.

Для прикладу зображення, розмір якого 640x480, якщо воно зберігається у вигляді файла BMP, то займає 1 036 856 байт. У форматі GIF воно займає 272 847 байт (коефіцієнт стискання — 1:4). Для подібних зображень існує інша схема стискання: JPEG. Це ж зображення у форматі

JPEG займає усього лише 105 885 байт. Проте це усе ще значний розмір, навіть за умови, що формат JPEG здійснює стискування із коефіцієнтом 10:1.

2.3 Стискування із втратами

Якщо ступінь стискування 2:1 недостатній, потрібно розглянути можливість використання JPEG. Абревіатура JPEG означає назву організації, яка розробила стандарт стискування, – Joint Photographic Experts Group (Об'єднана група експертів в області фотографії). В більшості програмних пакетів користувач JPEG може сам визначати ступінь стискування, який змінюється від 2:1 (стискування без втрат) до 40:1. В деяких програмах, таких як Adobe Photoshop можна вибрати конкретний коефіцієнт стискування; замість нього задається рівень якості. Чим вищий коефіцієнт стискування (або, відповідно нижчий рівень якості), тим більша частина даних зображення відкидається при компресії.

На відміну від формату GIF, у файлах JPEG деяка частина даних губиться. Алгоритм стискування зображень, використовуваний у форматі JPEG, набагато складніший, ніж у GIF. У ньому враховуються кольори, до яких найбільш і найменш чутливе людське око, використовується математичний апарат для уявлення частин зображення і т.д. При записі зображення за алгоритмом JPEG стискування у клітинках розміром 8 x 8 пікселів. У кожній клітинці оцінюється близькість значень кольору і зберігаються тільки значення, що істотно відрізняються від інших. (Чим вище вибраний вами рівень компресії, тим більш широкий діапазон значень кольорів вважається близьким, і тим більше значень кольору відкидається.) При відкритті і декомпресії файла всім пікселам клітинки, що раніше мали вихідні значення кольору, присвоюються одні і те ж значення. Але найразючішим є той факт, що зображення у форматі JPEG не є 8-бітовими. Вони 24-бітові, або так звані повнокольорові або фотореалістичні (true color) зображення.

Вражає, що, незважаючи на споконвічне триразове збільшення розміру (24-бітове зображення за розміром втричі перевищує відповідне 8-бітове зображення), у форматі JPEG досягається високий коефіцієнт стискування. Формат JPEG не тільки є гарним засобом стискування фотографічних зображень для використання в Web, але і дозволяє забезпечити краще уявлення графіки, ніж формат GIF. Таким чином, існує і другий доказ на користь формату JPEG в порівнянні з GIF: навіть при меншому розмірі файла кожний піксел у форматі JPEG описується великим числом бітів.

Щоб одержати мінімально можливий розмір файла при стискуванні, дотримуйтеся таких правил.

Для GIF-файлів використовуйте найменшу кількість кольорів, при якому зображення виглядає достатньо добре. Це найкраще зробити за

допомогою програми PhotoImpact GIF SmartSaver або HVS WebFocus для GIF-файлів.

Для JPEG-файлів використовуйте найвищий рівень стиснення, при якому зображення виглядає відносно неякісно. Це найкраще зробити за допомогою програми PhotoImpact JPEG SmartSaver або HVS WebFocus для JPEG-файлів.

Недоліки: відеомонітори багатьох користувачів підтримують тільки 8-бітові палітри кольору. Це означає, що навіть якщо ви помістили на свою Web-сторінку чудове 24-бітове зображення у форматі JPEG, багато відвідувачів побачать його лише в 8-бітовому варіанті. Більш того, якщо на сторінці утримується декілька зображень, усі вони будуть відображатися з використанням тих самих 256 кольорів.

Якщо в зображенні є великі області одного кольору, вибирайте метод стиснення GIF. Для фотографічних або подібних їм зображень використовуйте метод стиснення JPEG. Якщо зображення відповідає обома характеристикам, визначайте тип стиснення експериментальним шляхом.

2.4 Структура файлів зображень типу BMP і TIFF

Стисла інформація про графічні файли приведена в табл. 2.1.

Розглянемо структуру файлів зображень типу BMP і TIFF, що одержали найбільше поширення на практиці.

BMP: Windows Device Independent Bitmap

BMP (апаратно-незалежне побітове зображення Windows) підтримується будь-якими Windows-сумісними програмами. Структура файла BMP використовується Windows для збереження растрових зображень. У цьому форматі зберігаються рисунки фону, піктограми й інші растрові зображення Windows. Цей формат зводить до мінімуму можливість помилок або неправильної інтерпретації растрових даних. Формат BMP правильно відображає дані незалежно від апаратних і програмних пристроїв (монітора комп'ютера, відеокарти і драйвера дисплея). Така незалежність від пристроїв забезпечується застосуванням системних палітр. Проте даний формат має і свої недоліки. Тільки версії формату з 4- і 8-бітовим кольором піддаються стисненню (до них можна застосувати метод стиснення RLE), отже, 24-біто-файли BMP будуть дуже великими. Крім того, застосування файлів BMP обмежене платформами Windows і OS/2.

Таблиця 2.1 – Типи графічних файлів

Назва	Тип	Використання	Фірма	Розширення
BMP (Windows BitMap)	Растровий	Зберігання та відображення в середовищі Windows	Microsoft	bmp
GIF (Graphics Inter-change Format)	Растровий	Передача даних у мережі CompuServe	CompuServe Inc.	gif
Kodak Photo CD	Растровий	Для фотографічної інформації	Eastman Kodak	pcd
PCX (PC Paintbrush File Format)	Растровий	В графічних редакторах	Zsoft Corp.	pcx
JPEG ((Joint Photographic Experts Group)	Растровий	Для фотографічної інформації	Joint Photographic Experts Group	jpg
TIFF (Tagged Image File Format)	Растровий	Обмін даними між настільними і видавничими системами	Aldus Corp.	tif
DXF (Drawing Interchange Format)	Векторний	Обмін кресленнями та даними САПР	Autodesk Inc.	dxf
CDR (Corel Draw)	Векторний	Креслярська, видавнича та інші види графіки	Corel	cdr
WMF (Windows Metafile)	Векторний	Збереження і відображення в середовищі Windows	Microsoft	wmf

Кожний файл BMP містить заголовок файлу, заголовок зображення, растрові дані і карту кольорів (крім зображення 24-бітним кольором).

Заголовок файла BMP містить інформацію про тип і розмір файла, а також про розташування в ньому даних. Після заголовка файла впливає структура, що задає інформацію про розмір, кольори і стиснення зображення.

Стиснення. Поле biCompression у BITMAPINFO визначає, який тип стиснення використовується у файлі. Якщо значення поля дорівнює BI_RGB, то рисунок не стискається. Якщо ж значення поля BI_RLE4 або BI_RLE8, то рисунок використовує метод стиснення груповим кодуванням для зображень, відповідно, з 4 біт/піксел або 8 біт/піксел.

Карта кольорів. Значення поля `biClrUsed` визначає, чи будуть використовуватися растровим рисунком. індексовані кольори. Якщо рисунок складається з 1-, 4-, або 8-бітових пікселів, то повинна використовуватися карта кольорів. Якщо значення поля `biClrUsed` дорівнює нулю, то застосовується або повна карта, або 24-бітовий колір. Якщо поле не дорівнює нулю, то його значення дорівнює дійсній кількості кольорів, що будуть доступні програмі-генератору графіки або драйверам пристроїв. Поле `biClrImportant` визначає, скільки кольорів на індексованій карті вважаються важливими для зображення рисунка. Якщо його значення дорівнює нулю, то важливими будуть усі кольори.

Растрові дані. За картою кольорів ідуть растрові дані. Вони зберігаються у вигляді або 8-бітових індексів карти кольорів, або у вигляді буквених 24-бітових даних системи RGB. Піксели в растрі записуються, починаючи з нижнього лівого кута зображення, і читаються зліва направо і знизу вгору.

TIFF: Tagged Image File Format

Формат TIFF зберігає множину даних зображення в позначених полях, що і дало його назву («Формат файла позначеного зображення»). Кожне позначене поле береже інформацію про растровий рисунок або посилання на інші поля. Програма, що читає файл, може пропускати невідомі або непотрібні їй поля. Ця багатогранність дозволяє формату знаходити застосування в різноманітних комп'ютерних системах. Крім того, формат TIFF може зберігати різні додаткові дані про растровий рисунок, включаючи: криву корекції для зображення з відтінками сірого; поля детальної інформації про зображення (назва програми, автора, дату створення і коментар); розмір зображення і роздільну здатність; детальну інформацію про колір оригіналу.

Більшість програм, що читають файл TIFF, спроможні без проблем прочитати файли цього формату, створені на інших системах. Структура формату TIFF мінлива, що дозволяє йому берегти різні типи зображень.

Файли TIFF починаються з невеличкого заголовка, за яким випливає покажчик на перший набір позначених полів. Позначені поля збираються в каталоги файла зображення (КФЗ), що просто містить лічильник кількості позначених полів і покажчик на перше з них. Довжина позначеного поля дорівнює 12 байтів, тому програма, яка зчитує, може швидко знайти кожне з них.

Заголовок файла. Заголовок файла TIFF складає 8 байтів. Перші 2 байти вказують, який формат у файла, наступні 2 байта містять номер версії, останні 4 байти заголовка вказують на перший КФЗ. Звичайно перший КФЗ повинен йти відразу після заголовка, але у файлі TIFF усе локалізується за допомогою покажчиків.

Каталог файла зображення. Він містить двобайтовий лічильник загальної кількості позначених полів, за якими йдуть саме позначені поля.

Двобайтова довжина обмежує розміри кожного КФЗ 256-а позначеними полями.

Позначені поля. Довжина позначеного поля дорівнює 12 байтів. Перші 2 байти – код мітки (тип позначеного поля), наступні 2 байти містять тип даних, на які вказує поле. Тип даних дозволяє програмі дізнатися, скільки байтів потрібно читати для кожного окремого елемента даних. Усього п'ять різних типів: однобайтовий код символу ASCII; однобайтове ціле число; двобайтове ціле число; чотирибайтове ціле число, восьмибайтове дробове число.

Наступні 4 байти показують, на скільки різних елементів даних вказує позначене поле. Якщо дані займають у загальному 4 байта або менше, то елементи даних утримуються в останніх 4 байтах позначеного поля. Якщо ж дані займають більше 4 байтів, то останні 4 байта позначеного поля використовуються як показник даних.

Стискання. Файли даного формату спроможні берегти піксельні дані, стиснуті за допомогою різноманітних схем, наприклад, кодуванням за методом Хаффмана.

2.5 Формати файлів для різних застосувань (файли для мультимедіа, презентації, відео і опрацювання поточних документів)

Якщо файл буде остаточно використаний в додатку для компонування сторінки, то його треба зберегти в одному з двох стандартних форматів, використовуваних для виведення на пресу: TIFF або EPS. Уникайте формату PICT – він ненадійний при друкуванні з високим розділенням на приладах, що підтримують PostScript, і файлів формату PCX, в яких не зберігається інформація системи CMYK про кольороділення. Два формати – TIFF, і EPS – мають декілька опцій, які дозволяють краще пристосувати дані формати до конкретних задач.

Серед форматів файлів, не призначених для виведення на пресу, найбільшою популярністю користуються BMP і TARGA для IBM. Якщо скановане зображення використовується як для преси, так і для мультимедіа, не забудьте створити принаймні дві копії файла – одну у форматі TIFF або EPS для виведення на пресу, а другу в форматі, який найбільше підходить для середовища, не зв'язаного з пресою.

BMP (*.bmp)

Формат Windows Bitmap (BMP) розроблявся як спільний зі всіма додатками Windows. Для додатків в операційній системі OS/2 є своя версія BMP. Файли формату BMP розпізнаються багатьма мультимедійними програмами, програмами підготовки презентацій і цифрової обробки відео на платформах PC и Macintosh. В форматі BMP можна зберігати

чорно-білі, сірі півтонові, індексні кольорові і кольорові зображення системи RGB (але не двокольорові або кольорові зображення системи CMYK).

CT фірми Scitex (.ct)*

Формат файлів Continuous Tone (CT) фірми Scitex споконвічно призначався для спеціалізованих систем додрукової підготовки кольорових ілюстрацій, виготовлених Scitex. Записуйте файли в цьому форматі, тільки якщо збираєтесь переносити їх в систему Scitex для компоновки або розділення кольорів. Зображення, збережені в форматі CT, не зберігають інформацію про альфа-канали і лінії вирізу.

EPS (.eps, *.epsf)*

Ще один стандартний формат, застосовуваний на комп'ютерах різних платформ, – це Encapsulated PostScript (скорочується як .eps на PC і як .epsf на Macintosh). Споконвічно він розроблявся для записування векторних зображень у вигляді, що дозволяє експортувати їх у графічні додатки чи для верстання публікацій, але згодом призначення цього формату настільки розширилося, що стало включати і растрові зображення. Як і TIFF, формат EPS добре підходить для виведення на пресу, проте, оскільки він розроблявся на основі PostScript, цей формат дозволяє зберігати додаткові типи інформації, що в деяких ситуаціях дає йому перевагу над TIFF. Зберігайте файл у форматі EPS, якщо вам необхідно зберегти таку інформацію:

- Двоколірні, триколірні або чотириколірні криві.
- Альфа-канали (можливі й у TIFF).
- Кольороподіл (можливий і в TIFF).
- Лінії вирізу.
- Про спеціальне напівтонове маскування тонової кривої, що ставиться до конкретного зображення, а не до всього документа, що роздруковується.

Попередження: На відміну від векторних зображень формату EPS, растрові зображення цього формату неможливо збільшувати до нескінченності без втрати якості.

TARGA (.tga)*

Формат TARGA, розроблений фірмою Truevision для високопрофесійних графічних і відео додатків, усе ще широко застосовується на платформі PC. У форматі TARGA записуються сірі напівтонові і кольорові зображення системи RGB із глибиною кольору 16, 24 і 32 біта. 32-бітний файл формату TARGA може містити інформацію про альфа-канал.

TIFF (*.tif)

Формат TIFF (Tagged Image File Format) був розроблений спеціально для використання в додатках, зв'язаних з компонуванням сторінки. Він підтримується всіма основними графічними пакетами і пакетами редагування зображень і читається на багатьох платформах. Існує кілька варіантів TIFF, але більшість програмних додатків підтримують останню версію 6.0. У файлі формату TIFF можна зберігати чорно-білі і сірі напівтонові зображення, кольорові зображення систем RGB, CMYK і CIELAB (але не двоколірні). Формат TIFF має декілька переваг. Ви можете:

- зберегти інформацію про альфа-канал у додатках, що підтримують його;
- стиснути файл, використовуючи метод LZW (Lempel-Ziv-Welch). LZW відноситься до числа методів стиснення без втрат якості, він не руйнує дані цифрового зображення. Використовуючи метод LZW, можна стиснути дані тільки в 2 рази;
- зберегти файл для використання на конкретній платформі (версії формату TIFF для комп'ютерів PC і Macintosh розміщують дані в різному порядку). Більшість основних додатків, що працюють з компонуванням сторінки, читають файли формату TIFF, створені на будь-якій платформі. У сумнівних випадках зберігайте файл у форматі TIFF для тієї платформи, на якій працює ваша програма.

Контрольні запитання

1. Які формати графічних файлів Вам відомі?
2. Яка структура файлів зображень типу BMP?
3. Яка структура файлів зображень типу TIFF?
4. Яке призначення форматів BMP та TIFF?
5. Порівняйте можливості таких графічних форматів – BMP та GIF.
6. Порівняйте можливості таких графічних форматів – JPEG та TIFF.
7. Закодуйте за методом Хафмана послідовність пікселів, що задані такою комбінацією: 2ab3c2def 4g3hi2j.

3 Програмування графіки в середовищі Turbo Pascal

У залежності від використовуваних засобів побудови зображення можна виділити чотири види графічних програм:

- програми базуються тільки на стандартних процедурах і функціях, що містяться в бібліотеці GRAPH;
- графічні зображення створюються засобами BIOS. Такі програми використовують функції переривання 10h;
- пряме звернення до портів відеоадаптера або комірок відеопам'яті (це найшвидший спосіб);
- Pascal-код доповнюється вставками на мові Асемблера або безпосередньо в машинному коді, що виконують різноманітні графічні дії.

Далі мова йтиме про створення графічних програм, в основі яких лежить використання підпрограм бібліотеки GRAPH. Для полегшення процесу програмування графіки, крім даної бібліотеки фірма Borland Int. розробила також набір драйверів, що дозволяють працювати практично з усіма типами моніторів, і набір шрифтів для виведення символів різних форм і розмірів.

3.1 Апаратна і програмна підтримка графіки

Апаратна підтримка комп'ютерної графіки забезпечується відеокартою і монітором. На екрані монітора, як мінімум, 25 разів за секунду формується зображення. Людське око не спроможне вловити таке швидке миготіння кадрів, і створюється ілюзія нерухомості зображення. Зображення на екрані будується за допомогою пікселів, що об'єднуються в телевізійні рядки.

Для керування моніторами використовуються спеціальні відеокарти. Відеокарти (відеоадаптери, контролери монітора) являють собою складні електронні пристрої, які керуються власним процесором і мають свою відеопам'ять. У загальному випадку відеокарта складається з контролера електронно-променевої трубки і відеобуфера (відеопам'яті). Для комп'ютерів IBM PC розроблені декілька стандартів відеокарт: MDA (Monochrome Display Adapter), CGA (Color Graphics Adapter), EGA (Enhanced Graphics Adapter), VGA (Virtual Graphics Array). У даний час актуальним є ознайомлення зі стандартом VGA (його модифікацією є стандарт SVGA – Super VGA), тому що інші вже застаріли.

Програмну підтримку графіки забезпечують спеціалізована бібліотека GRAPH, набори драйверів і шрифтів. Підключення модуля GRAPH до програми користувача здійснюється за допомогою зарезервованого слова USES: Uses GRAPH.

Графічні драйвери знаходяться в BGI-файлах і підключаються при

ініціалізації графічного режиму. Список підтримуваних у середовищі Turbo Pascal відеокарт і відповідних драйверів приведений у табл. 3.1.

Таблиця 3.1 – Відеокарти і драйвери

Відеокарта	Драйвер
IBM CGA, MCGA	CGA. BGI
IBM EGA, VGA	EGAVGA. BGI
IBM 3270 PC	PC3270. BGI
IBM 8514	IBM8514. BGI

Виведення тексту в графічному форматі здійснюється за допомогою використання шрифтів. Стандартний набір у Turbo Pascal містить у собі такі шрифти: TriplexFont, SmallFont, SansSerifFont, GothicFont, що розміщуються у файлах trip. chr, litt. chr, sans. chr і goth. chr відповідно.

3.2 Ініціалізація графіки

Перед початком роботи з графікою необхідно встановити режим роботи монітора. Кожен із драйверів підтримує від 1 до 3 відеорежимів. Драйвер і режим можуть бути задані як число або як символічна константа, деякі з яких приведені в табл. 3.2.

Таблиця 3.2 – Графічні режими монітора

Драйвер	Режим	Роздільна здатність	Файл
CGA(1)	CGACO (CGANI)	320X200(640X200)	Cga. Bgi
EGA (3)	EGALo (EGANI)	640x200(640x350)	EgaVga. Bgi
VGA (9)	VGALo (VGANI)	640X200(640x350)	EgaVga. Bgi

Процедура, що встановлює графічний режим, має формат: InitGraph(DriverVar, ModeVar, "шлях до файла графічного драйвера").

Цілочислові змінні DriverVar і ModeVar задають драйвер і режим. Перший параметр може задаватися як за іменем, так і цифрою. На випадок, якщо тип дисплея комп'ютера невідомий, є стандартна константа Detect. Якщо відповідне типу дисплея значення присвоєне параметру Driver VAR (DriverVar:=Detect;), то InitGRAPH автоматично ініціює потрібний

драйвер, встановлює найбільш придатний графічний режим. Третій параметр InitGRAPH – це шлях до драйвера, якщо він знаходиться в поточному каталозі, то замість маршруту можна зазначити порожній рядок тобто два апострофи. Для різноманітних маніпуляцій із відеорежимами в Turbo Pascal є такі процедури і функції:

GetDriveName: string; – функція повертає ім'я встановленого драйвера.

GetGraphMode: integer; – функція повертає поточний графічний режим.

GetModeRange(graphDriver: integer; Var loMode: integer); – процедура повертає мінімальний і максимальний графічні режими для даного драйвера.

DetectGraph(Var GraphDriver, GraphMode: integer); – процедура повертає номери поточного драйвера і режиму.

GetMaxMode: integer; – функція повертає номер максимального режиму встановленого драйвера.

GetModeName(GraphMode: integer): string; – функція повертає ім'я поточного драйвера.

GraphDefaults; – процедура встановлює поточний показник (CP) у початок і перевстановлює графічну систему.

SetGraphMode(Mode: integer); – процедура переводить систему в графічний режим і очищує екран.

RestoreCRTMode; – процедура відновлює відеорежим, що був до ініціалізації графіки.

Коли всі роботи для конструювання і показу графіки виконані, необхідно вийти з графічного режиму за допомогою процедури, яка не має параметрів, CloseGraph, тобто закрити графічний режим.

Пам'ять відеобуфера підрозділяється на відеосторінки. Їхня кількість залежить від поточного режиму і типу адаптера. Більше однієї сторінки мають відеокарти EGA, VGA, Hercules. Нумерація сторінок починається з нуля. У кожний момент часу на екрані монітора може бути видимою одна сторінка. За замовчуванням це сторінка з номером 0. Сторінка, на якій у даний момент формується зображення, називається активною.

Для роботи зі сторінками передбачені дві процедури: SetActivePage(Page: word) – встановлює активну сторінку для побудови зображення, при цьому створюваний рисунок не відображається на екрані. Після побудови зображення можна візуалізувати цю сторінку за допомогою процедури SetVisualPage(Page: word).

Для запобігання і нейтралізації помилок у графічному режимі передбачені дві функції GraphResult і GraphErrorMsg. GraphResult повертає значення 0 (grOK), якщо остання графічна операція виконувалася без помилок. GraphError – повертає рядок повідомлення про помилку, що відповідає коду помилки.

Приведемо послідовність операторів, що виконують ініціалізацію графічного режиму:

```

Uses Graph;
Var DriverVar, ModeVar : integer;
ErrorCode :integer;
Begin
    DriverVar := Detect;
    InitGraph(DriverVar, Modevar, '');
    ErrorCode := GraphResult;
    IF ErrorCode <> grOK
    Then
    begin
        writeln('Графічна системна помилка: ', ErrorCode);
        Halt(1);
    end End.

```

Іноді в процесі роботи потрібно періодично переходити з текстового режиму в графічний і навпаки. Нижче приводиться приклад подібних переключень між режимами.

```

Uses Graph;
Var DriverVar, ModeVar: integer;
Begin
    ClrScr;
    Writeln('Текстовий режим');
    DriverVar := Detect;
    InitGraph(DriverVar, Modevar, '');
    OutTextXY(200,100, 'Графічний режим');
    RestoreCRTMode;
    Writeln('Текстовий режим');
    .....
    SetGraphMode(ModeVar);
    OutTextXY(200,100, 'Графічний режим');
    CloseGraph
End.

```

Варто мати на увазі, що при переході з одного режиму в інший виконується очищення екрану. Тобто при необхідності програміст повинен сам піклуватися про зберігання і відновлення вмісту екрану або його області.

3.3 Використання базових процедур і функцій

Для побудови зображення на екрані використовується прямокутна система координат. Відлік починається від верхнього лівого кутка екрану з нуля. Значення X (стовпчик) збільшується зліва направо, значення (рядок) збільшується зверху вниз. Щоб будувати зображення, необхідно вказувати точку початку виведення. В графічних режимах існує невидимий

поточний показчик CP (Current Pointer). Процедура MoveTo(X, Y) переміщує поточний показчик у точку з координатами X, Y. Процедура MoveRel(d, d) переміщує CP на dX точок по горизонталі і на d точок по вертикалі щодо CP. Для контролю місця розташування CP використовують функції GetX і GetY, що повертають відповідно значення X- і Y-координати CP. Для того, щоб дізнатися про максимально припустимі значення CP для встановленого режиму, використовуються функції GetMax: integer і GetMaxY: integer.

Графічний екран може розглядатися як одне велике або декілька менших за розміром вікон. Вікно – це прямокутна область екрана, що виконує функції повного екрана. Після установки вікна інша площина екрана ніби не існує, і все введення-виведення здійснюється через це вікно. У кожний момент може бути активним тільки одне вікно. Якщо вікон декілька, то за переведення введення-виведення в потрібне вікно відповідає програміст.

Для створення вікна призначена процедура SetViewport(x1, x2, y1, y2: integer; Clip: boolean), де x1, y1 – координати лівого верхнього кутка, x2, y2 – координати правого нижнього кутка вікна. Параметр Clip визначає, буде рисунок відсікатись при виході за межі вікна (Clip:=True) чи ні.

Для очищення вікна використовується процедура ClearViewport. Після її виконання всі зображення у вікні стираються, і CP встановлюється в ліву верхню точку вікна. Координатну систему повного екрана можна відновити за допомогою ClearDevice або задавши в процедурі встановлення вікна максимально можливі значення:

SetViewport(0,0, GetMaxX, GetMaxY, True).

Атрибути потокового вікна можна одержати за допомогою процедури GetViewportSetting (Var Vp: ViewPortType). Змінна Vp відноситься до стандартного типу ViewPortType:

Type

ViewPortType=RECORD

Clip: x1,x2,y1,y2: integer; boolean

end;

Тіло («зафарбування») графічного вікна встановлюється за допомогою процедур SetFillPattern і Bar, наприклад:

SetViewport(100, 50, 500, 200, True);

SetFillStyle(1,3); (* вибір типу фону*)

Bar(100,50,500,200); (*формування фону*).

У бібліотеці GRAPH виведення точки здійснюється процедурою PutPixel(X, Y: integer; Color: word), де X, Y – екранні координати розташування точки, Color – її колір (від 0 до 15). Для визначення кольору точки в конкретній позиції екрана служить функція GetPixel(X, Y: integer): word. З точок можлива побудова ліній. Це виконує процедура Line(X1, Y1,

X2, Y2), де X1, Y1 – координати початку, X2, Y2 – координати кінця лінії. Наприклад: Line(1, 1, 200, 1).

Для креслення ліній застосовуються також ще дві процедури: LineTo(X, Y) і LineRel(d, d). Перша будує лінію з точки поточкового розташування показника в точку з координатами X, Y; а друга проводить лінію від точки поточкового розташування показника в точку CPx+d, CPy+dy, де CPx і CPy – поточні координати CP. Установка стилю (тонкі, широкі, штрихові пунктирні лінії і т.д.) проводиться процедурою SetLineStyle (LineStyle: word; Pattern: word; Thickness: word). Параметр LineStyle встановлює тип рядка, що може бути заданий поіменованою константою або відповідним їй цифровим значенням із табл. 3.3, Pattern – зразок. Thickness – товщина лінії, обумовлена константами з табл. 3.4.

Таблиця 3.3 – Тип лінії

Константа	Значення	Тип
SolidLn	0	Безперервна лінія
Dotted Ln	1	Лінія з точок
CentedLn	2	Лінія з точок і тире
DashedLn	3	Штрихова лінія
UserLn	4	Тип користувача

Таблиця 3.4 – Товщина лінії

Константа	Значення	Товщина
NormWidth	1	Нормальна товщина
ThickWidth	3	Жирна лінія

Процедура GetLineSettings(Var lineInfo: LineSettingsType) повертає поточний стиль, образ і товщину лінії, установлені SetLineStyle. Тип LineSettingsType описується стандартно так (визначається у модулі GRAPH):

Type

LineSettingType=Record

LineStyle: word;

Pattern: word;

Thickness: word end;

Для встановлення режиму виведення (копіювання або накладання) лінії на екран призначена процедура SetWriteMode(Mode). Значення Mode визначається стандартними константами:

Const

CopyPut=0;

XORPut=1;

Аргумент Mode може приймати одне з двох значень. Якщо Mode=0, то пікселі, що лежать на відрізку прямої лінії, перевизначають пікселі на екрані, і лінія на екрані має поточний колір. Якщо Mode=1, то пікселі, що утворюють лінію, мають код кольору, рівний результату операції, що виключає АБО над кодами поточного кольору і кольорами пікселів на екрані, через які ця лінія проходить.

3.4 Робота з кольором

У відеобуфері (відеопам'яті) для збереження інформації про колір піксела відводиться фіксована кількість бітів пам'яті. Проте практично будь-який сучасний відеоадаптер спроможний відобразити значно більшу кількість кольорів, ніж це визначається кількістю бітів, що відводяться у відеобуфері для одного піксела. Для використання цієї можливості вводиться поняття палітри. Палітра – це масив, у якому кожному можливому значенню піксела зівставляється значення кольору, виведене на екран. Розмір палітри і її організації залежати від типу використовуваного відеоадаптера.

При використанні відеокарти EGA/VGA для установки кольору піксела використовується 6 бітів. Для формування кольору використовується система RrGgBb, де RGB – червоний, зелений і блакитний кольори нормальної яскравості, а rgb – ті ж кольори, але яскравість їх у два рази менша. Для EGA/VGA-карт драйвер EGAVGA.BGI установлює 54 кольори. У табл. 3.5 приведені 16 основних кольорів.

До того моменту, поки колір не визначено, для виведення використовується колір, що має максимальний номер палітри і фон, встановлюваний за мінімальним номером. Для EGA- і VGA-адаптерів у якості фону може бути задане будь-яке ціле число в діапазоні 0-63.

Будь-який колір для створення фігур і виведення тексту може бути встановлений за допомогою процедури SetColor(Color: word). Тіло задається процедурою SetBkColor(Color: word).

Для перевірки максимально припустимої кількості використовуваних програмою кольорів служить процедура GetMaxColor: word, що повертає максимальне значення коду кольору в палітрі мінус 1.

Процедури GetColor: word і GetBKColor: word повертають номери поточних кольорів відповідно для виведених елементів зображення і для фону.

Для керування палітрами можуть бути використані такі процедури і функції:

- GetDefaultPalette(Var palette: paletteType) – повертає апаратну палітру в запис PaletteType;
- GetPalette(Var palette: PaletteType) – повертає поточну палітру і її розмір;

- SetPalette(colorNum: word; color: shortint) – змінює один з кольорів палітри на новий колір через colorNum і color;
- SetAllPalette(Var palette) – змінює порядок розташування кольорів у палітрі;
- GetPaletteSize: integer – повертає кількість кольорів у палітрі.

Таблиця 3.5 – Кольори відеокарти EGA/VGA

Константа	Значення	Код	Колір
EGABlack	0	000000	Чорний
EGABlue	1	000001	Синій
EGAGreen	2	000010	Зелений
EGACyan	3	000011	Синьо-зелений
EGARed	4	000100	Червоний
EGAMagenta	5	000101	Червоно-синій
EGABrown	6	000110	Коричневий
EGALightGray	7	000111	Світло-сірий
EGADarkGray	56	111000	Темно-сірий
EGALightBlue	57	111001	Яскраво-синій
EGALightGreen	58	111010	Яскраво-зелений
EGALightCyan	59	111011	Яскравий синьо-зелений
EGALightRed	60	111100	Яскраво-червоний
EGALightMagenta	61	111101	Яскравий червоно-синій
EGAYellow	62	111110	Жовтий
EGAWhite	63	111111	Білий

3.5. Робота з текстом

У графічному режимі для виведення на екран тексту використовуються процедура OutText(TextString: string), що виводить рядок тексту з потокового положення CP, і OutTextXY(X, Y, Text), де X, Y – координати точки початку виведення тексту, Text – константа або змінна типу string. Визначені проблеми створює виведення числових даних, тому що в модулі GRAPH немає спеціально призначених для цього процедур. Для подібного виведення необхідно перевести числову константу в рядкову, наприклад, за допомогою процедури Str.

Встановити потрібний шрифт виведення можна процедурою SetTextStyle(Font: word; Direction: word; CharSize: word), де Font – обраний шрифт, що задається константами з табл. 3.6; Direction – напрямок виводу (горизонтальний або вертикальний), що задається константами з табл. 3.7, CharSize – розмір виведених символів.

Таблиця 3.6 – Типи шрифтів

Константа	Значення	Тип шрифту
DefaultFont	0	8 x 8-бітовий шрифт
TriplexFont	1	Штриховий шрифт
SmallFont	2	Малий шрифт
SansSerifFont	3	Шрифт SansSerif
GothicFont	4	Готичний шрифт

Таблиця 3.7 – Орієнтація тексту

Константа	Значення	Орієнтація
HorizDir	0	Зліва направо
VertDir	1	Знизу вгору

Таблиця 3.8 – Орієнтація ліній

Константа	Значення	Орієнтація ліній
LeftText	0	Вирівняти вліво
CenterText	1	Центрування по горизонталі
RightText	2	Вирівняти вправо
BottomText	0	Вирівняти вниз
CenterText	1	Центрування по вертикалі
TopText	2	Вирівняти вверх

Для вирівнювання тексту необхідно використовувати процедуру SetTextJastify (Horiz, Vert: word). Вирівнювання щодо СР виконується по вертикалі і по горизонталі за допомогою параметрів Horiz і Vert (табл. 3.7).

3.6 Процедури для побудови геометричних фігур

Бібліотека GRAPH містить ряд процедур, які на основі параметрів, що задаються, формують різноманітні геометричні фігури.

Для побудови прямокутних фігур існують декілька процедур. Процедура креслення одновимірного прямокутника має такий формат:

Rectangle (X1, Y1, X2, Y2: integer),

де X1, Y1 – координати лівого верхнього кута, X2, Y2 – координати правого нижнього кута прямокутника. Більш зручні для сприйняття зафарбовані прямокутники можна будувати з використанням процедури:

Bar(x1, y1, x2, y2: integer),

що малює зафарбований стовпчик. Колір зафарбування встановлюється за допомогою SetFillStyle. Процедура

Bar3D(x1, y1, x2, y2: integer; Depth: word; Top: boolean)
викреслює тривимірний зафарбований прямокутник. При цьому використовуються тип і колір зафарбовування, встановлені за допомогою процедури SetFillStyle. Параметр Depth являє собою число пікселів, що задають глибину тривимірного контуру. Найчастіше його значення приймають рівним чверті ширини прямокутника:

$$\text{Depth} = (\text{X2} - \text{X1}) \text{DIV } 4.$$

Параметр Top визначає, чи потрібно будувати над прямокутником вершину (Top:=True), чи ні (Top:=False).

У Turbo Pascal є процедура DrawPoly, що дозволяє будувати будь-які багатокутники лінією потокового кольору, стилю і товщини. Вона має такий формат:

DrawPoly(NumPoints: word; VAR PolyPoints).

Нетипізований параметр PolyPoints містить координати кожного перетинання в багатокутнику, параметр NumPoints задає число координат у PolyPoints. Варто мати на увазі, що при кресленні замкнутої фігури з N вершинами у виклику процедури Drawpoly потрібно передати N+1 координату, де координата вершини з номером N буде дорівнювати координаті вершини з номером 1. Зафарбування багатокутника можна зробити, використовуючи процедуру

FillPoly(NumPoints: word; VAR PolyPoints).

Алгоритми побудови кривих складні для самостійної реалізації, тому раціонально використовувати готові процедури модуля GRAPH. Для задання кутів застосовується полярна система координат. Процедура креслення кола поточним кольором має такий вигляд:

Circle(X,Y,Radius: word),

де X, Y – координати центру окружності, а Radius – радіус. Для створення псевдоб'ємних фігур використовуються дуги. Їх можна викреслити за допомогою процедури

Arc(X,Y: integer; StAngle, EndAngle, Radius: word), де StAngle і EndAngle – початковий і кінцевий кут вирисовування. Колір для креслення встановлюється процедурою SetColor. Робота з різноманітними типами кривих потребує інформації про координати останнього обертання до Arc. Її можна одержати, виконавши процедуру

GetArcCoords(VAR ArcCoords: ArcCoordsType), де змінна ArcCoords належить до стандартного типу

Type

ArcCoordsType=Record

x, y: integer;

xStart, yStart: integer;

zEnd, yEnd: integer

End;

Процедура повертає змінну, що містить координати центру X і Y, початкову xStart, yStart і кінцеву xEnd, yEnd позиції покажчика при виконанні останньої команди Arc.

Для побудови еліптичних дуг призначена процедура

Ellipse(X, Y: integer; StAngle, EndAngle: word; x, y: word),

де X, Y – центр еліпса, x і y – горизонтальна і вертикальна осі. Дуга еліпса викреслюється від початкового кута StAngle до кінцевого кута EndAngle поточним кольором. Щоб створити зафарбованим еліпс, використовується процедура

FillEllipse(X, Y: integer; x, y: word),

де X, Y – центр еліпса, x і y – горизонтальна і вертикальна осі. Заповнювач установлюється процедурами SetFillStyle або SetFillStyle.

У програмах часто потрібно розділити коло на сектори. Це можна зробити, використовуючи процедуру

PiveSlnce(X, Y: integer; StAngle, EndAngle, Radius: word),

що рисує і заповнює маскою сектор кола. Контур сектора викреслюється поточним кольором, а при зафарбуванні використовується тип і колір зафарбування, задані процедурами SetFHISyle і SetFillPattern. Можна створити і заповнити сектор в еліпсі. Для цього використовується процедура

Sector(X, Y: integer; StAngle, EndAngle, x, y: word),

де, Y, – центр; x, y – горизонтальний і вертикальний радіуси. Креслення проводиться поточним кольором і заповнюється стилем, заданим процедурами SetFillStyle або SetFillPattern.

У такому прикладі приведена програма, що рисує на жовтому фоні синього сніговика.

```

Program SnowMan;
Uses Graph, Crt;
Var DriverVar, ModeVar: integer; c: char;
Begin
  DriverVar:=Detect;
  InitGraph(DriverVar, ModeVar, '');
  SetBkColor(14); SetColor(1);
  Circle(300,300,50);
  Circle(300,215,35);
  Circle(300,160,20);
  Circle(300,180,5); Circle(290,140,5); Circle(310,140,5);
  Line(335,215,370,200); Line(265,215,230,200);
  c:=readkey;
  SetBkColor(0); SetColor(GetMaxColor);
  CloseGraph;
End.

```

3.7 Маніпулювання фрагментами зображень

У модулі GRAPH передбачений ряд додаткових процедур, що дозволяють зберігати і відновлювати окремі фрагменти зображення на екрані. Для маніпулювання фрагментом варто з'ясувати його розмір. Це робиться з використанням функції

ImageSize(x1, y1, x2, y2: integer): word,

де x1, y1 і x2, y2 – координати верхнього лівого і правого нижнього кутів прямокутної області екрана. Функція повертає розмір зазначеного фрагмента в байтах. Образ фрагмента зручно берегти в динамічній області пам'яті, тобто результат роботи ImageSize може бути вхідною інформацією для процедури GetMem, що виділяє зазначений обсяг у верхній області пам'яті. Зберігання образу фрагмента в пам'яті виконується процедурою

GetImage(x1, y1, x2, y2: integer; Var BitMap),

де x1, y1, x2, y2 – координати фрагмента, BitMap – нетипізований параметр, що повинен бути не меншим розміру пам'яті, відведеному для області екрана, плюс 6. Повернути образ фрагмента з пам'яті на екран можна за допомогою процедури

PutImage(X, Y: integer; Var BitMap; Mode: word),

де X, Y – координати точки екрана, починаючи з якої буде виводиться фрагмент, BitMap – уже відомий нетипізований параметр, що задає ширину і висоту фрагмента.

Контрольні запитання

1. Що таке апаратна та програмна підтримка комп'ютерної графіки?
2. Поясніть призначення відеокарти та назвіть її складові.
3. Охарактеризуйте основні графічні режими роботи комп'ютера.
4. За допомогою яких операторів здійснюється ініціалізація графічного режиму при роботі в середовищі PASCAL?
5. Які базові процедури і функції роботи з графікою середовища PASCAL ви знаєте?
6. Яким чином здійснюється перехід між текстовим і графічним режимами?
7. Які особливості має координатна сітка екрана в графічному режимі?
8. Що таке поточний показчик?
9. Яким чином здійснюється керування кольором і стилем геометричних фігур і тексту, що виводяться?

4 Афінні перетворення на площині

Виведення зображення на екран дисплея і різноманітні дії з ним, у тому числі і візуальний аналіз, потребують від користувача достатньої геометричної грамотності. Геометричні поняття, формули і факти, що відносяться, насамперед, до плоского і тривимірного випадків, грають у задачах комп'ютерної графіки особливу роль.

Якщо розглянути графічну реалізацію тривимірних об'єктів, то можна зауважити, що вона тісно зв'язана з властивостями об'єктів. Система координат екрана, як відомо, є двовимірною, тому на екрані можлива емуляція тривимірної системи координат, яка найбільш зручна для наступних розрахунків. Надалі всі об'єкти вважаються тривимірними, а відображення здійснюється за допомогою набору функцій розробленої бібліотеки.

Одним із прикладів реалізації даного підходу може служити такий. Кожен об'єкт, у найпростішому випадку, являє собою паралелепіпед і зберігається в пам'яті розмірами по трьох осях. Також у його структуру входить набір спеціальних точок, що відповідають за з'єднання блоків у просторі. У загальному випадку, це точка прив'язки і вихідна точка. У цілому, виходить гнучка графічна модель, що дозволяє змінювати розміри блоків практично миттєво. Таким чином, з'являється можливість реалізувати найпростіший графічний редактор тривимірних об'єктів. При цьому всі блоки будуть змінюватися, створюючи загальну графічну модель. Маючи справу з графічною моделлю, можна реалізувати обертання сукупності тривимірних об'єктів. Це здійснюється за допомогою набору функцій, що обертають об'єкти. Для обертання кожного об'єкта існує алгоритм, що розбиває об'єкт (у найпростішому випадку паралелепіпед) на набір точок, кожна з яких обертається, використовуючи найпростіші перетворення в просторі шляхом множення матриці радіуса-вектора на матриці перетворень у просторі. Розглянемо більш докладно даний підхід з формальної сторони.

У комп'ютерній графіці усе, що відноситься до двовимірного випадку прийнято позначати символом 2D (2-dimension).

Припустимо, що на площині введена прямолінійна координатна система. Тоді кожній точці M ставиться у відповідність упорядкована пара чисел (x, y) її координат (рис.4.1). Вводячи на площині ще одну прямолінійну систему координат, ми ставимо у відповідність тій же точці M іншу пару чисел $-(x^*, y^*)$.

Перехід від однієї прямолінійної координатної системи на площині до іншої описується співвідношеннями:

$$x^* = \alpha x + \beta y + \lambda \quad (4.1)$$

$$y^* = \gamma x + \beta y + \mu \quad (4.2)$$

де $\alpha, \beta, \gamma, \lambda, \mu$ – довільні числа, зв'язані нерівністю:

$$\begin{vmatrix} \alpha & \beta \\ \gamma & \delta \end{vmatrix} \neq 0 \quad (4.3)$$

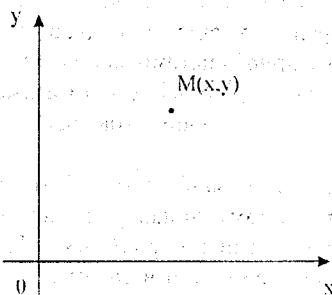


Рисунок 4.1

Формули (4.1) і (4.2) можна розглядати з двох боків: або зберігається точка і змінюється координатна система (рис. 4.2) – у цьому випадку довільна точка M залишається тією ж, змінюються лише її координати $(x, y) \rightarrow (x^*, y^*)$, або змінюється точка і зберігається координатна система (рис. 4.3) – у цьому випадку формули (4.1) і (4.2) задають зображення, що переводить довільну точку $M(x, y)$ у точку $M^*(x^*, y^*)$, координати якої визначені в тій же координатній системі.

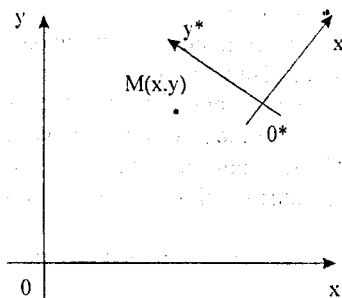


Рисунок 4.2

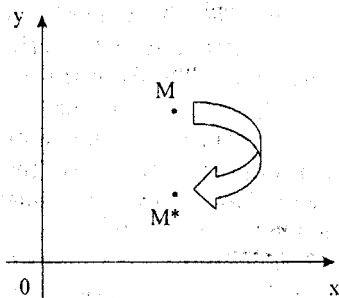


Рисунок 4.3

Надалі, формули (4.1) і (4.2) будуть розглядатися як правило, відповідно до якого в заданій системі прямолінійних координат перетворюються точки площини.

У афінних перетвореннях площини особливу роль грають деякі важливі частинні випадки, що мають добре просліджувані геометричні характеристики. При дослідженні геометричного змісту числових коефіцієнтів у формулах (4.1) і (4.2) для цих випадків зручно вважати, що задана система координат є прямокутною декартовою.

1. Поворот навколо початкової точки на кут (рис.4.4) описується формулами:

$$x^* = x \cos \varphi - y \sin \varphi \quad (4.3)$$

$$y^* = x \sin \varphi + y \cos \varphi \quad (4.4)$$

2. Розтягання (стиск) вздовж координатних осей можна задати так:

$$x^* = \alpha x \quad (4.5)$$

$$y^* = \delta y \quad (4.6)$$

$$\alpha > 0, \delta > 0 \quad (4.7)$$

Розтягання (стиск) вздовж осі абсцис забезпечується за умови, що $\alpha > 1$ ($\alpha < 1$). На рис. 4.5 $\alpha = \beta > 1$.

3. Відображення (щодо осі абсцис) (рис.4.6) задається за допомогою формул:

$$x^* = x \quad (4.8)$$

$$y^* = -y \quad (4.9)$$

На рис. 4.7 вектор переносу MM^* має координати λ, μ . Перенос забезпечує співвідношення:

$$x^* = x + \lambda \quad (4.10)$$

$$y^* = y + \mu \quad (4.11)$$

Вибір цих чотирьох окремих випадків визначається двома обставинами.

1. Кожне з приведених вище перетворень має простий і наочний геометричний зміст (геометричним змістом наділені і постійні числа, що входять у приведені формули).

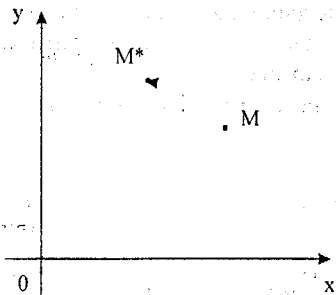


Рисунок 4.4

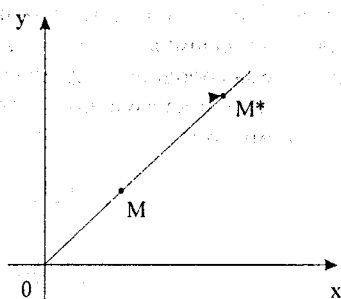


Рисунок 4.5

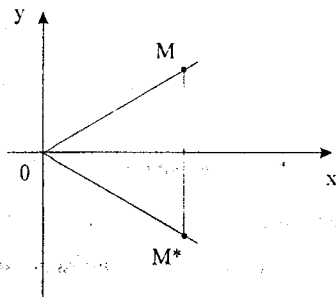


Рисунок 4.6

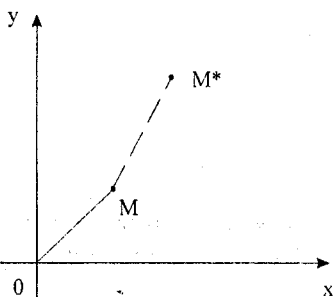


Рисунок 4.7

2. Як відомо з курсу аналітичної геометрії, будь-яке перетворення виду (4.1) завжди можна представити як послідовне виконання (суперпозицію) найпростіших перетворень виду 1–4 (чи частини цих перетворень).

Таким чином, справедлива наступна важлива властивість афінних перетворень площини: будь-яке відображення виду (4.1) можна описати за допомогою відображень, що задаються формулами (4.3) – (4.11).

Для ефективного використання цих відомих формул у задачах комп'ютерної графіки більш зручним є їхній матричний запис. Матриці, що відповідають випадкам 1–3, будуються легко і мають відповідно такий вигляд:

$$\begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

4.1 Однорідні координати точки

Нехай M – довільна точка площини з координатами (x, y) , обчисленими щодо заданої прямолінійної координатної системи. Однорідними координатами цієї точки називається будь-яка трійка одночасно не рівних нулю чисел x_1, x_2, x_3 , зв'язаних із заданими числами x і y такими співвідношеннями:

$$x_1/x_3 = x, \quad x_2/x_3 = y. \quad (4.12)$$

При розв'язанні задач комп'ютерної графіки однорідні координати звичайно вводяться так: довільній точці $M(x, y)$ площини ставиться у відповідність точка $M^*(x, y, 1)$ у просторі.

Необхідно помітити, що довільна точка на прямій, що з'єднує початок координат, точку $O(0, 0, 0)$ із точкою $M^*(x, y, 1)$, може бути задана трійкою чисел виду (hx, hy, h) .

Будемо вважати, що $h = 0$. Вектор з координатами hx, hy, h є напрямним вектором прямої, що з'єднує точки $O(0, 0, 0)$ і $M^*(x, y, 1)$. Ця пряма перетинає площину $z = 1$ у точці $(x, y, 1)$, що однозначно визначає точку (x, y) координатної площини xy .

Так само між довільною точкою з координатами (x, y) і безліччю трійок чисел виду (hx, hy, h) , $h \neq 0$, встановлюється взаємно однозначна відповідність, що дозволяє вважати числа hx, hy, h новими координатами цієї точки.

Широко використовувані в проєкційній геометрії однорідні координати дозволяють ефективно описувати так звані невласні елементи (власне кажучи, ті, котрим проєктна площина відрізняється від звичайної евклідової площини).

У проєктивній геометрії для однорідних координат прийняте позначення:

$$x : y : 1 \quad (4.13)$$

або

$$x_1 : x_2 : x_3 \quad (4.14)$$

(тут неодмінно потрібно, щоб числа x_1, x_2, x_3 одночасно не перетворювалися в нуль).

Застосування однорідних координат є зручним вже при розв'язанні найпростіших задач.

Розглянемо, наприклад, питання, зв'язані зі зміною масштабу. Якщо пристрій відображення працює тільки з цілими числами (чи якщо

необхідно працювати тільки з цілими числами), то для довільного значення h (наприклад, $h = 1$) точку з однорідними координатами (0.5, 0.1, 2.5) представити не можна. Однак при розумному виборі h можна домогтися того, щоб координати цієї точки були цілими числами. Зокрема, при $h = 10$ для розглянутого прикладу маємо (5, 1, 25).

Розглянемо інший випадок. Щоб результати перетворення не приводили до арифметичного переповнення, для точки з координатами (80000, 40000, 1000) можна взяти, наприклад, $h = 0.001$. У результаті одержимо (80, 40, 1).

Приведені приклади показують корисність використання однорідних координат при проведенні розрахунків. Однак основною метою введення однорідних координат у комп'ютерній графіці є їхня безсумнівна зручність у застосуванні до геометричних перетворень.

За допомогою трійок однорідних координат і матриць третього порядку можна описати будь-яке афінне перетворення площини.

Вважаючи, $h = 1$, порівняємо два записи:

$$(x^* y^* 1) = (xy1) \begin{pmatrix} \alpha & \gamma & 0 \\ \beta & \delta & 0 \\ \lambda & \mu & 1 \end{pmatrix} \quad (4.15)$$

Неважко помітити, що після перемноження виразів, що знаходяться у правій частині останнього співвідношення, ми одержимо формули (4.1) і (4.2) і вірна числова рівність $1 = 1$. Тим самим порівнювані записи можна вважати рівносильними.

Елементи довільної матриці афінного перетворення не несуть у собі явно вираженого геометричного змісту. Тому, щоб реалізувати те чи інше відображення, тобто знайти елементи відповідної матриці за заданим геометричним описом, необхідні спеціальні прийоми. Звичайну побудову цієї матриці у відповідності зі складністю поставленої задачі і з описаними вище окремими випадками розбивають на кілька етапів.

На кожному етапі пишеться матриця, що відповідає тому чи іншому з виділених вище випадків 1-4, що мають добре виражені геометричні властивості.

Випишемо відповідні матриці третього порядку.

А. Матриця обертання (rotation)

$$[R] = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.16)$$

Б. Матриця розтягання-стискання (dilatation)

$$[D] = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.17)$$

В. Матриця відображення (reflection)

$$[M] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.18)$$

М. Матриця переносу (translation)

$$[T] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \lambda & \mu & 1 \end{pmatrix} \quad (4.19)$$

Розглянемо приклади афінних перетворень площини.

Приклад 1. Побудувати матрицю повороту навколо точки $A(a,b)$ на кут φ (рис. 4.8).

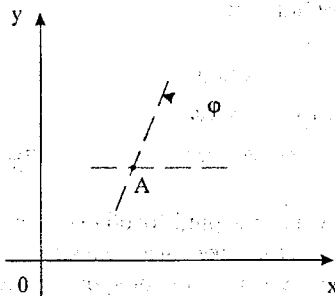


Рисунок 4.8.

1-й крок: Перенос на вектор $-A(-a,-b)$ для зсуву центра повороту з початком координат;

$$[T_{-A}] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a & -b & 1 \end{pmatrix} \quad (4.20)$$

матриця відповідного перетворення.

2-й крок. Поворот на кут φ

$$[R_\varphi] = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.21)$$

матриця відповідного перетворення.

3-й крок. Перенос на вектор $A(a,b)$ для повернення центра повороту в попереднє положення;

$$[T_A] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix} \quad (4.22)$$

матриця відповідного перетворення.

Перемножимо матриці у тому ж порядку, як вони виписані:

$$[T_A][R_\varphi][T_A].$$

У результаті одержимо, що шукане перетворення (у матричному записі) буде мати такий вигляд:

$$(x^* y^* 1) = (x y 1) \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ -a \cos \varphi + b \sin \varphi + a & -a \sin \varphi - b \cos \varphi + b & 1 \end{pmatrix} \quad (4.23)$$

Елементи отриманої матриці (особливо в останньому рядку) не так легко запам'ятати. У той же час кожна з трьох матриць, що перемножуються, за геометричним описом відповідного відображення легко будується.

Приклад 2. Побудувати матрицю розтягання з коефіцієнтами розтягання α вздовж осі абсцис і β вздовж осі ординат із центром в точці $A(a,b)$.

1-й крок. Перенос на вектор $-A(-a,-b)$ для сполучення центра розтягання з початком координат;

$$[T_{-A}] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a & -b & 1 \end{pmatrix} \quad (4.24)$$

матриця відповідного перетворення.

2-й крок. Розтягання вздовж координатних осей з коефіцієнтами α і β відповідно; матриця перетворення має вигляд:

$$[D] = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.25)$$

3-й крок. Перенос на вектор $A(a,b)$ для повернення центра розтягання в попереднє положення; матриця відповідного перетворення:

$$[T_A] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix} \quad (4.26)$$

Перемножимо матриці у тому ж порядку:

$$[T_A][D][T_{-A}],$$

отримаємо остаточно:

$$(x^* y^* 1) = (xy1) \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \delta & 0 \\ (1-\alpha)a & (1-\delta)b & 1 \end{pmatrix} \quad (4.27)$$

Міркуючи подібним чином, тобто розбиваючи запропоноване перетворення на етапи підтримувані матрицями $[R]$, $[D]$, $[M]$, $[T]$, можна побудувати матрицю будь-якого афінного перетворення за його геометричним описом.

4.2 Афінні перетворення у просторі

Розглянемо тривимірний випадок 3D (3-dimension) і відразу введемо однорідні координати.

Діючи аналогічно тому, як це було зроблено в двовимірному випадку, замінимо координатну трійку (x, y, z) , що задає точку у просторі, на четвірку чисел $(x, y, z, 1)$ або, на четвірку (h_x, h_y, h_z) , $h = 0$.

Кожна точка простору (крім початкової точки O) може бути задана четвіркою одночасно не рівних нулю чисел; ця четвірка чисел визначена однозначно з точністю до загального множника.

Запропонований перехід до нового способу завдання точок дає можливість скористатися матричним записом і в більш складних тривимірних задачах.

Будь-яке афінне перетворення в тривимірному просторі може бути представлено у вигляді суперпозиції обертань, розтягань, відображень і переносів. Тому цілком доречно спочатку докладно описати матриці саме цих перетворень (ясно, що в даному випадку порядок матриць повинен бути рівним чотирьом).

А. Матриці обертання у просторі.

Матриця обертання навколо осі абсцис на кут φ :

$$[R_x] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.28)$$

Матриця обертання навколо осі ординат на кут ψ :

$$[R_y] = \begin{pmatrix} \cos \psi & 0 & -\sin \psi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.29)$$

Матриця обертання навколо осі аплікат на кут χ :

$$[R_z] = \begin{pmatrix} \cos \chi & \sin \chi & 0 & 0 \\ -\sin \chi & \cos \chi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.30)$$

Корисно звернути увагу на місце знака «-» у кожній із трьох приведених матриць.

Б. Матриця розтягання-стискання:

$$[D] = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.31)$$

де $\alpha > 0$ – коефіцієнт розтягання (стискання) вздовж осі абсцис;
 $\beta > 0$ – коефіцієнт розтягання (стискання) вздовж осі ординат;
 $\gamma > 0$ – коефіцієнт розтягання (стискання) вздовж осі аплікат.

В. Матриці відображення.

Матриця відображення щодо площини xu :

$$[M_z] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.32)$$

Матриця відображення щодо площини yz :

$$[M_x] = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.33)$$

Матриця відображення щодо площини zx :

$$[M_y] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.34)$$

М. Матриця переносу (тут (λ, μ, ν) – вектор переносу):

$$[T] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \lambda & \mu & \nu & 1 \end{pmatrix} \quad (4.35)$$

Як і в двовимірному випадку, усі виписані матриці невироджені.

Приведемо важливий приклад побудови матриці складного перетворення за його геометричним описом.

Приклад 3. Побудувати матрицю обертання на кут φ навколо прямої L , що проходить через точку $A(a, b, c)$ і яка має напрямний вектор (l, m, n) . Можна вважати, що напрямний вектор прямої є одиничним:

$$l^2 + m^2 + n^2 = 1$$

На рис. 4.9 схематично показано матрицю якого перетворення потрібно знайти.

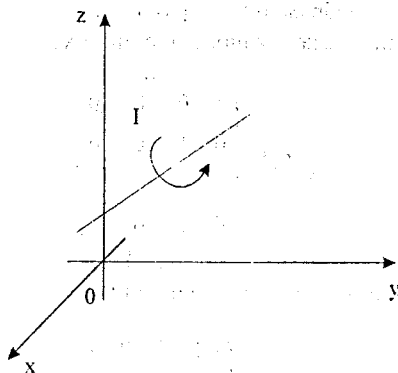


Рисунок 4.9

Розв'язування сформульованої задачі розбивається на кілька кроків. Опишемо послідовно кожний з них.

1-й крок. Перенос на вектор $-A$ $(-a, -b, -c)$ за допомогою матриці:

$$[T] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a & -b & -c & 1 \end{pmatrix} \quad (4.36)$$

У результаті цього переносу ми отримаємо, що пряма L буде проходити через початок координат.

2-й крок. Сполучення осі аплікату з прямою L двома поворотами навколо осі абсцис і осі ординат.

1-й поворот – навколо осі абсцис на кут ψ (який підлягає визначенню). Щоб знайти цей кут, розглянемо ортогональну проекцію L' вихідної прямої L на площину $X = 0$ (рис. 4.10).

Напрямний вектор прямої L' визначається просто – він дорівнює $(0, m, n)$.

Звідси відразу ж випливає, що

$$\cos \psi = n / d, \quad \sin \psi = m / d, \quad (4.37)$$

$$\text{де } d = \sqrt{m^2 + n^2} \quad (4.38)$$

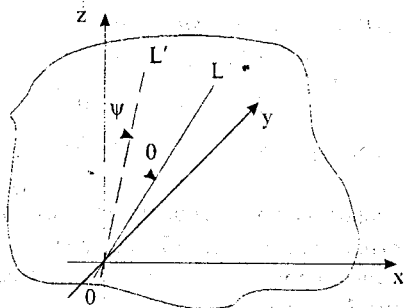


Рисунок 4.10

Відповідна матриця обертання має вигляд:

$$[R_y] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & n/d & m/d & 0 \\ 0 & -m/d & n/d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.39)$$

Під дією перетворення, описуваного цією матрицею, координати вектора (l, m, n) зміняться. Підрахувавши їх, у результаті одержимо

$$(l, m, n, 1) [R_y] = (l, 0, d, 1). \quad (4.40)$$

2-й поворот навколо осі ординат на кут θ обумовлений співвідношеннями:

$$\cos \theta = l, \quad \sin \theta = -d \quad (4.41)$$

Відповідна матриця обертання записується в такому вигляді:

$$[R_x] = \begin{pmatrix} 1 & 0 & d & 0 \\ 0 & 1 & 0 & 0 \\ -d & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.42)$$

3-й крок. Обертання навколо прямої L на заданий кут.

Оскільки тепер пряма L збігається з віссю апікат, то відповідна матриця має вигляд:

$$[R_z] = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 & 0 \\ -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4.43)$$

4-й крок. Поворот навколо осі ординат на кут θ .

5-й крок. Поворот навколо осі абсцис на кут ψ .

Однак обертання в просторі некомутативне. Тому порядок, у якому проводяться обертання, є дуже важливим.

6-й крок. Перенос на вектор A (a, b, c).

Перемноживши знайдені матриці в порядку їхньої побудови, одержимо матрицю:

$$[T][R_x][R_y][R_z][R_y]^{-1}[R_x]^{-1}[T]^{-1}.$$

Випишемо остаточний результат, вважаючи для простоти, що вісь обертання проходить через початкову точку.

$$\begin{pmatrix} l^2 + \cos \varphi(1 - l^2) & l(1 - \cos \varphi)m + n \sin \varphi & l(1 - \cos \varphi)n - m \sin \varphi & 0 \\ l(1 - \cos \varphi)m - n \sin \varphi & m^2 + \cos \varphi(1 - m^2) & m^2(1 - \cos \varphi)n + l \sin \varphi & 0 \\ l(1 - \cos \varphi)n + m \sin \varphi & m(1 - \cos \varphi)n - l \sin \varphi & n^2 + \cos \varphi(1 - n^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Розглядаючи подібні приклади, ми будемо одержувати в результаті невідроджені матриці вигляду:

$$[A] = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & 0 \\ \beta_1 & \beta_2 & \beta_3 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & 0 \\ \lambda & \mu & \nu & 1 \end{pmatrix}. \quad (4.44)$$

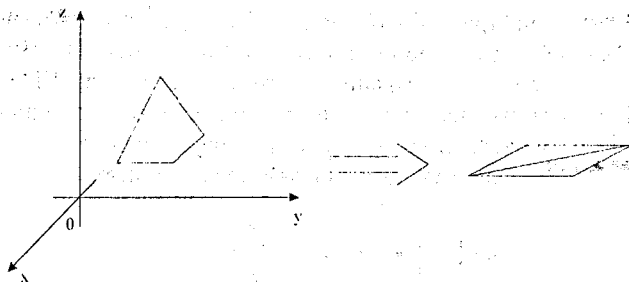
За допомогою таких матриць можна перетворити будь-які плоскі і просторові фігури.

Приклад 4. Потрібно піддати заданому афінному перетворенню опуклий багатогранник.

Для цього спочатку за геометричним описом відображення знаходимо його матрицю $[A]$. Зауважуючи далі, що довільний опуклий багатогранник однозначно задається набором усіх своїх вершин $V_i (x_i, y_i, z_i)$, $i = 1, \dots, n$, будемо матрицю

$$[V] = \begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{pmatrix} \quad (4.45)$$

Піддаючи цей набір перетворенню, описуваному знайденою невідродженою матрицею четвертого порядку, $[V][A]$, ми одержуємо набір вершин нового опуклого многогранника – вихідний образ.



4.3 Проекції

Всі види проекцій, що використовуються в комп'ютерній графіці, являють собою комбінації двох головних типів проекцій: паралельної і центральної (перспективної).

Паралельна проекція. При паралельній проекції точки предмета проєктуються на поверхню пучком променів, паралельних заданому напрямку. Координати зображення визначаються шляхом загального розв'язання рівняння прямої, що проходить через точку A зображення паралельно одиничному вектору, і рівняння поверхні проекції:

$$\begin{aligned}(X_A - X_A')/VX &= (Y_A - Y_A')/VY = (Z_A - Z_A')/VZ; \\ F(XA', YA', ZA') &= 0.\end{aligned}$$

де XA, YA, ZA – координати точки предмета;
 XA', YA', ZA' – координати точки зображення;
 VX, VY, VZ – координати одиничного вектора;
 $F(XA', YA', ZA')$ – рівняння поверхні проекції.

Якщо проєктування здійснюється на площину і промені, що проєктуються, перпендикулярні до неї, то проекція називається **ортогональною** або **перпендикулярною**. Цей вид проекції широко використовується в технічному кресленні. Якщо осі X і Y лежать у площині проекції, а Z перпендикулярна їй, то при представленні предмета в координатному базисі площини проекції, координати точок зображення можна визначити по координатах точок предмета за допомогою співвідношення:

$$[XA', YA', ZA'] = [XA, YA, ZA],$$

де $Z = \text{const}$ – координата площини проекції по осі Z .

Як впливає з рівняння, для отримання ортогональної проекції достатньо визначити координати XA і YA . Проте цим не обмежується процедура побудови тривимірної моделі на екрані дисплея. Предмет звичайно задається у своїй об'єктній системі координат XYZ , осі якої у вихідному стані паралельні екранній системі дисплея xuz . Зображення предмета відображається на екрані з деяким масштабним коефіцієнтом m , а початок об'єктної системи координат розташовується в точці X_0, Y_0, Z_0 екрана. Для цього більш загального випадку можна записати:

$$\begin{bmatrix} X_{A'} \\ Y_{A'} \\ Z_{A'} \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + m \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}.$$

Вибором значення m можна здійснювати **масштабування зображення** об'єкта. Щоб одержати тривимірне зображення, предмет разом із прив'язаною системою координат розвертається таким чином, що проекції координатних осей одержують певну орієнтацію один відносно

одного, а відрізки, взяті по координатних осях, відображаються на проекції з певним співвідношенням масштабних коефіцієнтів.

Для відтворення предмета на екрані в заданій проекції необхідно визначити *матрицю перетворення координат*. Основою для цього є принципи побудови зображення в даній проекції. В ізометричній проекції координатні осі X,Y,Z предмета відображаються на площині екрана дисплея під кутом 120 градусів, а масштабні коефіцієнти по всіх трьох осях однакові. З врахуванням цих властивостей ізометричної проекції можна записати:

$$\begin{bmatrix} X_{A'} \\ Y_{A'} \\ Z_{A'} \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + m \begin{bmatrix} 1 & 3^{0.5}/2 & 0 & 3^{0.5}/2 \\ 1 & -0.5 & 1 & -0.5 \\ 0 & 3^{-0.5} & 3^{-0.5} & 3^{-0.5} \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}$$

Аналогічно можна отримати формулу перетворення для фронтальної проекції, при якій осі X та Y проєктуються паралельно осям x, у дисплея, а проєкція осі Z збігається з бісектрисою кута між осями x і y. У цьому випадку, перетворення координат виконується в такий спосіб:

$$\begin{bmatrix} X_{A'} \\ Y_{A'} \\ Z_{A'} \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + m \begin{bmatrix} 1 & 1 & 0 & -2^{0.5}/4 \\ 1 & 0 & 1 & 2^{0.5}/4 \\ 0 & 2^{0.5}/4 & 2^{0.5}/4 & (2/3)^{0.5} \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}$$

Відзначимо, що фронтальна проєкція не є лінійним перетворенням простору предметів.

Формули перетворення дозволяють виконати креслення зображення предмета на дисплеї в ізометричній або фронтальній проекції, якщо задано його математичний опис. Проте такий вузькопрактичний підхід не дозволяє повною мірою використовувати можливості комп'ютерної графіки. У програмах формування зображення звичайно передбачається операція обертання предмета навколо двох або трьох осей. Це дозволяє одержати такий вигляд об'єкта, що найбільш виразно відображає його геометричну форму.

Центральна проєкція. При центральній проєкції всі промені, що проєктуються, проходять через певну точку простору S – центр проєкції. Фізичним пристроєм, що реалізує центральну проєкцію, є об'єктив. При візуальному спостереженні роль об'єктива виконує око. У об'єктиві промені, що з'єднують сполучені точки в просторі предметів і зображень, проходять через задню головну точку – центр проєкції. Координати

кожної точки зображення можуть бути обчислені шляхом визначення точки перетинання прямої, що проходить через предметну точку A і центр проєкції S , із поверхнею проєкції зображення. Якщо в обраній об'єктній системі координат відомі координати точок A та S , а також рівняння поверхні зображення $\varphi(X,Y,Z)=0$, то координати точки зображення A визначаються в результаті розв'язання системи рівнянь:

$$(XA - X_s)/(XA - X_s) = (YA - Y_s)/(YA - Y_s) = (ZA - Z_s)/(ZA - Z_s) \\ \varphi(XA', YA', ZA') = 0.$$

Для одержання чіткого зображення необхідно, щоб світлочутлива поверхня була перпендикулярна оптичній осі об'єктива і розташовувалася на визначеній відстані від центра проєктування, що, як правило, приймають рівним фокусній відстані f .

З урахуванням відзначених особливостей центральної проєкції в реальних пристроях формування зображення зв'язок координат точок простору предметів і простору зображень може бути виражений в іншій формі, ніж у системі рівнянь. Введемо систему координат xuz для площини зображення, пов'язану з системою координат об'єктива $x' y' z'$ і системою координат простору предметів XYZ . Особливість центральної проєкції можна виразити так: вектори rAS і RAS' , що з'єднують центр проєкції S із сполученими точками A та A' відповідно, є колінеарними. Звідси випливає, що $RAS = \lambda rSA'$, де λ – константа даної пари точок A і A' .

Формулу можна записати у такому вигляді:

$$\begin{bmatrix} X - X_s \\ Y - Y_s \\ Z - Z_s \end{bmatrix} = \lambda [C] \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}$$

де X_s, Y_s, Z_s – координати центру проєктування S у системі XYZ , x_0, y_0, z_0 – координати центру проєктування S і системі xuz .

Якщо врахувати, що поверхня проєкції є плоскою, початок системи координат xuz , як правило, збігається з головною точкою картинної площини, розташованої на відстані f від S ($z_0 = -f$), то формула прийме вигляд:

$$\begin{bmatrix} X - X_s \\ Y - Y_s \\ Z - Z_s \end{bmatrix} = \lambda [C] \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Крім константи шляхом розподілу першого і другого рядків на третій одержуємо рівняння, що зв'язує координати сполучених точок у системах XYZ і xyz:

$$(X-X_s)=(Z-Z_s)(c11x+c12y+c13f)/(c31x+c32y+c33f)$$

$$(Y-Y_s)=(Z-Z_s)(c21x+c22y+c23f)/(c31x+c32y+c33f)$$

З цієї системи по координатах x , y точок зображення можна визначити координати X , Y , Z з'єднаних точок у просторі предметів, якщо задане рівняння поверхні, що спотерігається $F(X,Y,Z)=0$. Потім по координатах X,Y,Z точки A , рівнянню поверхні $F(X,Y,Z)=0$ і відомим умовам освітлення можуть бути визначені атрибути точки A (яскравість, колір) і розраховані відповідні атрибути точки A' зображення. Описана процедура **синтезу зображень** оснований на відслідковуванні променя, що виходить із простору зображень у простір предметів, тобто в напрямку, протилежному ходу променів у реальній системі. Цей підхід у комп'ютерній графіці називається **методом оберненого трасування променів**.

Характерною рисою центральної проекції є істотна різниця в масштабах зображення предметів, що знаходяться на різноманітних відстанях від центру проектування. Це пов'язано зі зменшенням кутових розмірів предмета і відповідно зі зменшенням лінійних розмірів площини зображення при віддаленні від нього.

4.4 Текстура зображення

Під текстурою розуміють специфічний характер тону (кольору) у зображенні об'єкта або на його фрагменті. До текстури також відносять наявність на зображенні характерних ліній.

Текстура може мати впорядкований характер тону і бути у вигляді геометрично правильних або майже правильних рисунків. Подібну текстуру мають, наприклад, цегельна кладка, кахельне облицювання, шашівниця і так далі. Інший тип – стохастична структура. Вона властива природним об'єктам і, як правило, є наслідком шорсткості об'єктів, що спостерігаються.

Одним із принципів формування текстури зображення є перенос регулярного або стохастичного рисунка на поверхню об'єкта. Для нанесення рисунка необхідно вирішити задачу перетворення систем координат. Якщо рисунок заданий у просторі текстури в системі координат (u,w) , а поверхня – у системі координат (θ, φ) , то для переносу рисунка потрібно знайти або задати функцію відображення одного простору на інший, тобто визначити співвідношення $\theta = f(u, w)$, $\varphi = g(u, w)$.

У багатьох практичних випадках функції відображення лінійні: $\theta = Au + Bv + C$, $\varphi = Du + Ev + F$. Коефіцієнти A, U, C, D, E, F обчислюються зі співвідношень між відомими точками в системі координат.

Характерні точки візерунка з простору текстури переносяться в об'єктний простір, потім у простір зображення та певним чином з'єднуються відрізками прямих. У процедурах нанесення візерунків часто використовується розбивка поверхонь і візерунків на фрагменти.

При переносі на поверхню предмета текстури, що являє собою безупинне регулярне і випадкове поле яскравості, найбільш підходящим є метод оберненого трасування променів. Центр кожного пікселя зображення проєктується на поверхню предмета і по координатах точки на поверхні визначається відповідна їй точка в просторі текстури. Щоб усунути ефекти, пов'язані з просторовою дискретизацією, використовують процедури згладжування. Достатньо простою процедурою згладжування є трасування чотирьох точок, що відповідають кутам пікселя, і використання середнього значення яскравості текстури для цих чотирьох точок. Використання чотирьох точок пікселя дозволяє оцінювати достатність процедури згладжування. Якщо різниця яскравостей для зазначених чотирьох точок більша деякого граничного розміру, можна зробити розбивку пікселя на чотири частини і більш точно обчислити яскравість для пікселя.

Формування стохастичних текстурних полів, близьких до природних, являє собою складну у обчислювальному відношенні процедуру. Для синтезу текстур кращою вважається *авторегресивна модель*, і моделі з використанням методів *лінійного програмування*. Синтез здійснюється з використанням статистичних характеристик текстур-прототипів: щільності ймовірності й автокореляційної функції. Для визначення щільності ймовірності в межах деякого вікна вимірюють гістограму і для її відображення використовують перших чотири моменти – середнє значення, стандартне відхилення, асиметрію й ексцес. Форму автокореляційної функції виражають через її чотиривимірний центральний просторовий момент. У результаті формується восьмивимірний вектор ознак текстури. Поряд з описаним основним варіантом визначення параметрів, що характеризують стохастичною текстурю, використовують ряд спрощених процедур, що забезпечують зниження обчислювальних витрат. Синтезовані на основі прототипів стохастичні текстури добре моделюють природні шорстксті поверхні, якщо вони мають порівняно невеличку кривизну.

Для поверхонь із значною кривизною (куля, еліпсоїд і близькі до них) перенесення на них навіть стохастичної текстури не забезпечує їхнього реалістичного вигляду. Виникає відчуття, що текстура намальована на гладкій поверхні. Причина цього полягає в тому, що текстура, сформована без урахування форми об'єкта, не передає змін освітленості, обумовлених рельєфом поверхні. Для реальних шорстких

поверхонь вектор нормалі містить невеличку випадкову складову, що визначає характер зміни освітленості в зображенні. Щоб одержати зображення, близьке до реалістичного, необхідно змоделювати саме цей фактор. У методі моделювання рельєфу поверхні $Q(u, w)$ за допомогою допоміжної функції $F(u, w)$ у напрямку нормалі до поверхні вноситься збурення, що призводить до зміни відбитка від елементів знову утвореної поверхні.

Контрольні запитання

1. Пояснення однорідної системи координат.
2. Охарактеризуйте основні види перетворень на площині та у просторі.
3. Наведіть приклади застосування однорідної системи координат.

5 Графіка в Internet

5.1 Основні поняття мережі Internet

Internet – глобальна інформаційна мережа, що об'єднує комп'ютери з усього світу з метою обміну даними та доступу до інформаційних та технологічних ресурсів.

Internet об'єднує декілька інформаційних технологій, серед яких найважливішими є:

- Електронна пошта e-mail.
- Універсальна система доступу до гіпертекстової інформації на *Web-серверах* у системі *World Wide Web*.
- Система підтримки груп новин UseNet newsgroups та організації телеконференцій.
- Система передавання текстових файлів Gopher.

WWW (World Wide Web) – технологія розміщення та доступу до інформації на серверах, яка забезпечує:

- роботу з гіпертекстовою інформацією і мультимедійною формою її подання: текст, звук, графіка, відеозображення;
- універсальний доступ – доступ до документів на серверах різних типів (FTP, Gopher, тощо).

Гіпертекст (гіпертекстова технологія) – це спосіб ієрархічної організації інформації, яка може бути на одному чи багатьох серверах, зв'язок між якими забезпечують гіпертекстові посилання в системі *HTTP (Hyper Text Transfer Protocol)* протоколу.

Гіпертекстове посилання – це короткий підкреслений і відокремлений іншим кольором текст у документі, картинка чи інший елемент, клацнувши на якому мишею отримують доступ до файла іншого документа, звукового файла, відеофайла.

Документи в *WWW* відшукують за уніфікованою адресою документа (*Uniform Resource Locator*), яка має такий загальний вигляд:

<тип інформаційного простору>: //<назва серверу>/<назва каталогу>/<назва підкаталогу>/<назва файла>.

Адреса дозволяє однозначно вказати потрібний документ, на якому б сервері мережі він би не знаходився.

Типи основних інформаційних просторів такі:

- http – *WWW* простір;
- ftp – *FTP* простір;
- gopher – *Gopher* простір;
- news – *UseNet* простір;
- mailt – простір адрес *E-mail*;
- file – простір файлів.

Наприклад, набравши команду <http://www.yahoo.com> можна отримати доступ до пошукової системи Yahoo.

HTTP (Hyper Text Transfer Protocol) – протокол передачі даних у мережі Internet. Іншими словами – система домовленостей (правил), за якими передаються гіпертекстові документи (Web-сторінки).

Web-сторінка – окремий документ у системі *World Wide Web*. *Web-сторінка* може включати текст, графіку, звукове супроводження, анімацію, а також гіпертекстові посилання. Як було зазначено вище адреса визначається *URL* адресою. Для написання *Web-сторінки* використовують мову *HTML*.

Web-вузол – група тематично зв'язаних *Web-сторінок* та супроводжувальних файлів, які розташовуються на одному комп'ютері. Окремі сторінки *Web-вузла*, як правило, зв'язані між собою за допомогою гіперпосилань.

Найчастіше під словом Internet розуміють його найбільш популярну службу – *World Wide Web (WWW)* або просто *Web*. Інформація, яка опублікована в Мережі у форматі цієї служби, представляється у вигляді *Web-сторінок*. Прийом *Web-сторінок* з Інтернету і їхнє відображення – достатньо складна задача. Для цієї мети використовують спеціальні програми перегляду *Web-сторінок* – *броузери*. Броузери з'явилися практично одночасно з концепцією *World Wide Web* і пройшли великий шлях розвитку. У даний час найбільш популярні броузери *Netscape Navigator* і *Microsoft Internet Explorer*, хоча існують і інші програми, призначені для цієї мети. Броузер *Internet Explorer* з'явився пізніше, ніж *Netscape Navigator*, але поступово завоює усе більшу популярність тому, що компанія *Microsoft* включила *Internet Explorer 4.0* в операційну систему *Windows 98* у якості стандартного засобу перегляду *Web*. Починаючи з цієї версії програма називається вже не броузером, а *оглядачем Web*.

Основне призначення оглядача – завантаження *Web-сторінки* з Інтернету і відображення її на екрані. Завантажити *Web-сторінку* можна вказавши її *URL адресу* в Інтернеті.

5.2 Можливості програми Internet Explorer

Програма запускається подвійним клацанням на значку *Internet Explorer* на Робочому столі або за допомогою Головного меню.

Операційна система *Windows 98* розглядає комп'ютер як невід'ємну частину Інтернету. Першочергове призначення *Internet Explorer* – це перегляд *Web-сторінок*, що відображаються в робочій області її вікна. Крім цього він дозволяє завантажувати з Інтернету довільні файли: програми, архіви, довідкову інформацію, не використовуючи додаткового програмного забезпечення. Наприклад, передачею файлів через Інтернет

займається – служба FTP, а оглядач здатний працювати з цією службою за її протоколом.

Пункт Пошта використовують для доступу до електронної пошти, а пункт Новини – для доступу до телеконференцій (груп новин). В обох цих випадках Internet Explorer 4 запускає інший стандартний додаток Windows 98 – програму Outlook Express. Оглядач може використовувати її для навігації по файловій структурі автономний (не залучений до Мережі) комп'ютер. У цьому випадку він діє аналогічно програмі Провідник.

Інтерфейс оглядача Internet Explorer дуже схожий на інтерфейс програми Провідник. Наприклад, рядок меню оглядача практично збігається з рядком меню Провідника.

У той же час, панель інструментів програми Internet Explorer орієнтована на роботу з мережевими документами і тому істотно відрізняється від панелі інструментів Провідника. Проте якщо в оглядачі відкрити локальну папку, то панель інструментів замінить ту, що використовується в програмі Провідник.

1. Перша група кнопок панелі інструментів служити для керування навігацією. Ці кнопки використовують для переходу до попередньої або наступної Web-сторінки, для припинення завантаження сторінки або для відновлення поданої на ній інформації. Їм відповідають команда з меню Перехід, а також деякі команди (Зупинити й Оновити) з меню Вид.

Остання кнопка в цій групі використовується для переходу до основної сторінки – Web-сторінки, що за замовчуванням з'являється на екрані при запуску Internet Explorer.

2. Другу групу кнопок використовують для відкриття спеціальних панелей оглядача. Ці панелі дозволяють проводити пошук потрібної інформації в Інтернеті, забезпечують доступ до набору обраних Web-сторінок, дають можливість повернутися до однієї зі сторінок, що нещодавно проглядалися, і дозволяють використовувати канали – новий тип Web-вузлів, концепція яких з'явилася разом із версією Internet Explorer 4. Ці панелі можна також відкрити за допомогою команди Вид > Панелі оглядача.

3. Останню групу кнопок використовують для виконання службових функцій. Ці кнопки дозволяють перейти в режим на весь екран (команда Вид > У весь екран), відправити повідомлення електронної пошти (Перехід > Пошта), змінити шрифт або мову (Вид > Шрифти), роздрукувати Web-сторінку на папері (Файл > Преса) і відредувати її вихідний текст за допомогою заданого редактора Web-сторінок.

5.3 Перегляд Web-сторінок в Інтернеті

Реальна робота в Інтернеті може бути почата тільки після встановлення зв'язку з Інтернетом. У цьому випадку після запуску

програми Internet Explorer завантажується і відображається *основна сторінка*, на адресу якої настроєний оглядач.

У вікні оглядача відображається один документ. Звичайно Web-сторінки Інтернету переглядають послідовно, з метою знайти ті, що містять потрібну інформацію. Як правило, Web-сторінки являють собою складні комбіновані документи, що містять як текстову інформацію, так і вбудовані об'єкти. У даний час більшість Web-сторінок багато ілюстровані. У якості ілюстрацій використовують як зображення, так і звуки, відеоролики й інші об'єкти, що відносяться до мультимедіа. Розмаїття вбудованих об'єктів постійно збільшується.

Web-сторінки – це *форматовані* електронні документи, що багато в чому нагадують документи, створювані текстовим процесором. Проте Web-документи більш універсальні: вони повинні читатися легко незалежно від того, які розмір і роздільну здатність має екран читача, і від того, скільки кольорів відтворює його устаткування. Найбільш якісні Web-сторінки настільки універсальні, що з ними можна працювати навіть із найпростішим *неграфічним браузером*.

За замовчуванням оглядач Internet Explorer відтворює всю інформацію, наявну на Web-сторінці. Якщо ж з якоїсь причини відобразити включені в Web-документ об'єкти не вдається, програма помічає відповідні місця порожніми рамками або спеціальними значками. Багато Web-сторінок мають *альтернативний текст*, пов'язаний з об'єктами. Якщо засіб перегляду з якихось причин не може відобразити вбудований об'єкт, замість об'єкта відображається пов'язаний з ним альтернативний текст.

5.4 Гіпертекстові посилання

Для переходу до заданої Web-сторінки достатньо ввести її *адресу URL* у поле панелі Адреси і натиснути клавішу ENTER. Але введення довгих адрес URL за допомогою клавіатури не тільки незручне, але і здатне викликати помилки. На щастя, при роботі з Web-сторінками є зручна можливість переходу без явного введення URL-адреси сторінки.

Більшість Web-документів містять *Гіпертекстові посилання* або просто *гіперпосилання*. Гіперпосилання – це об'єкт Web-сторінки, що містить інформацію про адресу іншої Web-сторінки. В якості такого об'єкта звичайно виступає фрагмент тексту, виділений кольором і підкресленням, або графічна ілюстрація, виділена кольоровою рамкою.

Гіпертекстові посилання звичайно «вказують» на Web-сторінку, тематично пов'язану з тією, що переглядається в даний момент. Текст, що є посиланням, може описувати зміст нового документа. При *наведенні* на гіперпосилання покажчик миші приймає форму кисті руки з витягнутим вказівним пальцем. Щоб *перейти по гіперпосиланню*, достатньо на ньому

клацнути. При цьому оглядач завантажить нову Web-сторінку, що знаходиться за тією адресою, що зазначена в посиланні.

Нова Web-сторінка може містити нові гіперпосилання, що дозволяє продовжувати перегляд документів, що відносяться до тієї ж теми. Механізм гіперпосилань є однією з тих особливостей World Wide Web, що найбільшою мірою сприяли росту популярності цієї служби.

Якщо тематична область розшукуваної інформації відома, то почавши з однієї-двох «стандартних» адрес, теоретично можна знайти будь-яку інформацію з даної теми, користуючись тільки гіпертекстовими посиланнями. В Інтернеті чимало спеціальних сторінок-покажчиків, що містять тільки великі добірки гіперпосилань, що дають доступ до Web-сторінок, присвячених певним питанням.

Якщо після серії переходів по гіперпосиланнях необхідно повернутися на одну з раніше переглянутих сторінок, варто скористатися кнопкою Назад на панелі інструментів. Поруч із цією кнопкою є стрілка, що розкривається, яка дозволяє вибрати одну зі сторінок, раніше відвіданих у потоковому сеансі роботи.

5.5 Графіка і мультимедіа на Web-сторінках

Сучасна розмаїтість Web-сторінок і постійна поява нових типів об'єктів, що включаються в їхній склад, призводить до того, що засоби перегляду Web-сторінок не завжди можуть правильно відтворити об'єкти, які зустрічаються в Web-документах. Перші броузери були спроможні відтворювати тільки текст (із гіперпосиланнями). Сучасні засоби перегляду можуть працювати з багатьма типами об'єктів, але все-таки не з усіма.

Проблема відтворення на Web-сторінках різноманітних видів об'єктів вирішується або шляхом підключення додаткових додатків, або шляхом розширення можливостей оглядача. Відповідно, програми, призначені для відображення вбудованих об'єктів, використовуються як *допоміжні* або як *вбудовані додатки*.

5.6 Застосування допоміжних і вбудованих додатків

Допоміжний додаток — це звичайний додаток із числа встановлених на комп'ютері, що можна використовувати для відображення елемента Web-сторінки. У разі споживання такий додаток запускається оглядачем автоматично. При цьому відкривається окреме вікно. Цілісність Web-сторінки трохи порушується, але уся наявна на ній інформація відображається.

Вбудований додаток – це особливий вид додатка, що працює тільки під керуванням оглядача. Вбудовані додатки використовують для відображення об'єктів у рамках Web-сторінок без відкриття окремого вікна. Ці додатки запускають автоматично, коли оглядач зіштовхується з відповідним об'єктом. У даний час існують сотні вбудованих додатків. З їхньою допомогою можна відтворювати будь-які об'єкти, що зустрічаються на Web-сторінках.

Якщо на комп'ютері немає ні допоміжного, ні вбудованого додатка, що підходить для відкриття файла або об'єкта Web-сторінки, цей об'єкт відображається на сторінці порожньою рамкою або спеціальним значком. Як правило, у цьому випадку залишається можливість зберегти об'єкт у вигляді окремого файлу, щоб спробувати відтворити його згодом, коли будуть підібрані необхідні для цього програмні засоби.

5.7 Керування відображенням Web-сторінки

Найпростіші налаштування оглядача виконують за допомогою меню *Вид*. Перша група команд цього меню дозволяє змінити нинішній вигляд вікна програми Internet Explorer. Ці команди керують відображенням панелей інструментів, рядка стану, а також додаткових панелей.

Команду *Шрифти* використовують для керування шрифтами Web-сторінки. З її допомогою задають розмір шрифту (від найменшого до найбільшого) і кодування символів, що використовуються. Це дозволяє правильно відображати документи, написані на різних мовах.

Команди *Зупинити* й *Обновити* ставляться до поточної Web-сторінки. Перша дозволяє перервати її завантаження, а друга – завантажити дану сторінку повторно. Це необхідно, якщо є підозра, що з моменту останнього завантаження даної сторінки її зміст змінився.

Пункт *У вигляді HTML* дозволяє переглянути код HTML. Справа у тому, що зміст будь-якої Web-сторінки записаний на спеціальній мові, що називається HTML. Броузери відображають Web-сторінки на екрані відповідно до *вихідного тексту*, записаним кодом даної мови. Оглядач Internet Explorer дозволяє не тільки переглядати Web-сторінки, але і вивчати їхній код.

Пункт *На весь екран* переводить оглядача у режим повного екрану. При цьому для відображення Web-сторінки виділяються всі можливі місця. Це важливо для сучасних Web-сторінок, що розраховані на перегляд при високій роздільній здатності екрану.

Збереження інформації про Web-сторінки.

У кожного, хто має досвід роботи в Інтернеті, з'являються «улюблені» Web-сторінки, до яких доводиться звертатися регулярно. Це означає, що виникає список адрес URL обраних Web-сторінок. Такі обрані

адреси називають *закладками*. Вони дозволяють легко і швидко починати нові сеанси роботи в Інтернеті з обраної сторінки. Всі сучасні броузери мають засоби створення закладок, і оглядач Internet Explorer у цьому питанні не виняток.

5.8 Папка обраних адрес URL

Основним інструментом для створення закладок у програмі Internet Explorer служить папка Обране. Вона призначена для збереження ярликів Web-сторінок, що представляють особливий інтерес. Повернутися до сторінки, посилання на яку внесене в цю папку, можна одним клацанням миші.

Щоб додати ярлик Web-сторінки в папку Обране, треба клацнути в межах Web-сторінки правою кнопкою миші й у контекстному меню вибрати пункт Додати в обране. Можна також дати команду: Обране > Додати в обране.

Для доступу до ярликів, що знаходяться в папці Обране, служити панель Обране, що відкривають за допомогою відповідної кнопки на панелі інструментів або через меню Вид. Крім того, уся структура папки Обране доступна у вигляді послідовності меню, вкладених в меню Обране.

Коли кількість обраних сторінок досягає сотень, використовувати меню Обране стає незручно. Стандартний засіб наведення порядку в структурі меню складається в створенні вкладених тематичних папок із розміщенням ярликів безпосередньо в них. Провести цю роботу дозволить команда Обране > Упорядкувати обране. При цьому відкривається діалогове вікно Упорядкувати обране. За допомогою елементів керування цього вікна можна створити додаткові папки, вкладені в папку Обране, і розмістити в них ярлики Web-сторінок. Так створюють зручну для використання систему збереження закладок. Папці Обране відповідає папка \Windows\Favorites на жорсткому диску.

Контрольні запитання

1. Охарактеризуйте основні служби, що входять до мережі Інтернет.
2. Основні принципи організації служби WWW.
3. Що таке уніфікований вказівник ресурсу в мережі Інтернет та яке його функціональне призначення?
4. Що таке графіка для Web?
5. Що представляє собою Web-сторінка?
6. Як здійснити правку коду HTML?
7. Яким чином об'єкти різних типів вставляються на Web-сторінку?
8. Які види списків, що дозволяють створити мовою HTML, Ви знаєте?
9. Охарактеризуйте інтерфейс редактора FrontPage Express.

6 Графіка для Web. Створення Web-сторінок

6.1 Редактори HTML

Для створення Web-документів використовується мова HTML, синтаксис якої дозволяє створювати документи за допомогою звичайного текстового процесора, проте на практиці це робити не зовсім зручно. Форматування документів у сучасних текстових процесорах відбувається за принципом відповідності екранного виду документа його реальному виду, одержуваному друкуванням. Це відомий принцип WYSIWYG. Але в Інтернеті ніхто не може сказати заздалегідь, на якому комп'ютері і засобами якої операційної системи буде відтворюватися документ. Автор бачить на екрані одне зображення, а читач може побачити щось зовсім інше. Тобто, принцип відповідності порушується. Тому використання звичайних текстових процесорів для створення Web-документів вважається недоцільним.

Водночас, при створенні Web-документів на мові HTML таки багато чого можна автоматизувати. Найпотужнішим засобом автоматичного створення і публікації Web-документів нині вважається пакет Microsoft Frontpage 97. Він містить у собі все необхідне для роботи з Web-вузлом: програму Frontpage Explorer для навігації по Web-вузлу; потужний редактор Web-сторінок Frontpage Editor; засоби для роботи з графікою; засоби публікації документів та інше.

Але щоб приступити до самостійної розробки найпростіших Web-сторінок, немає необхідності мати у своєму розпорядженні і вивчати весь пакет Front Page 97. Спрощена версія редактора FrontPage Editor входить до складу системи Windows 98 в якості стандартного додатка за назвою FrontPage Express.

Редактор FrontPage Express являє собою простий засіб для створення Web-сторінок без використання мови HTML. Можливості мови HTML закладені в редактор FrontPage Express практично цілком. Освоєння цього редактора займає набагато менше часу, ніж вивчення мови HTML, хоча за це прийдеться розплачуватися деяким обмеженням можливостей.

Основні відмінності FrontPage Express від повної версії FrontPage Editor полягають тільки в тому, що губиться можливість роботи з Web-вузлом як з єдиним цілим, тобто FrontPage Express слугує не для автоматизації розробки Web-вузлів, а тільки для створення Web-сторінок. Даються також майстри і шаблони з числа наявних у повній версії. В цілому, редактор FrontPage Express надає все необхідне для створення окремої Web-сторінки будь-якої складності.

6.2 Інтерфейс програми FrontPage Express

Програма FrontPage Express випущена компанією Microsoft і виконана в єдиному стилі продуктів, що входять у пакет Microsoft Office. Якщо на комп'ютері встановлений пакет Internet Explorer 4.0 або операційної системи Windows 98, запустити цю програму можна через головне меню – *Пуск > Програми > Internet Explorer > FrontPage Express*.

1. У верхній частині вікна розташовуються рядок меню і дві панелі інструментів: Стандартна і Форматування. Самі ці панелі виглядають трохи інакше, ніж у програмі Word, що пов'язано з особливостями мови HTML.

2. Додаткова панель інструментів у програмі FrontPage Express містить поля форми. Лінійка в цій програмі не використовується, тому що точне позиціонування елементів сторінки засобами мови HTML неможливе.

3. Основну частину вікна займає робоча область, у якій можуть бути відкриті одне або декілька вікон, що містять окремі документи. Робота з кількома документами здійснюється точно так саме, як і в програмі Word.

4. Особливістю редактора FrontPage Express є відсутність горизонтальної смуги прокручування, незалежно від ширини вікна документа. Оскільки розмір вікна, у якому документ буде проглядатися кінцевим користувачем, заздалегідь невідомий, браузер сам переформатує документ таким чином, щоб він розташовувався по ширині екрана.

5. У нижній частині вікна розташовується рядок стану, що містить довідкову інформацію. Включити або відключити відображення панелей інструментів і рядки стану дозволяє меню Вид.

6.3 Форматування документа HTML

Форматування документа HTML у редакторі FrontPage Express здійснюється в основному за допомогою засобів панелі інструментів *Форматування*. Ця панель схожа на аналогічну панель програми Word, але варто пам'ятати, що реальне редагування здійснюється шляхом вставки в текст тегів HTML, кількість яких дуже обмежена. Наприклад, список доступних стилів на цій панелі містить тільки стандартні «стилі», використовувані в мові HTML, і не може бути змінений.

При зміні виду, розміру і кольору шрифту в документ вставляється відповідний тег . Курсив і напівжирний шрифт задається (як і рекомендується) тегамі виділення і сильного виділення (і).

Вирівнювання тексту задається за допомогою атрибутів у тегу абзацу <P>. Програма FrontPage Express дозволяє вирівнювати текст по лівому або правому краю, а також по центру.

Не рекомендується використовувати кнопки, що керують розміром відступу зліва. Мова HTML не має тегів, призначених для цієї цілі і редактор моделює такий ефект за допомогою тегів, що мають інше призначення.

Деякі засоби керування шрифтами не винесені на панель інструментів. Для їхнього застосування треба дати команду *Формат > Шрифт*. У діалоговому вікні Шрифт, що відкрилося, можна змінити параметри шрифту. Вкладка *Спеціальні стилі* дозволяє оформляти елементи фрази, а також надає деякі інші можливості, наприклад оформлення верхніх і нижніх індексів.

6.4 Введення гіперпосилань

Одним із найбільш істотних елементів World Wide Web є *гіпертекстові посилання* на інші сторінки. При цьому можна створювати як внутрішні, так і зовнішні гіперпосилання, у тому числі і на документи, що ще не існують у момент створення посилання.

1. Для створення гіперпосилання треба виділити фрагмент тексту, що буде використовуватися як посилання, і клацнути на кнопці *Створити* або *Змінити посилання*, що знаходиться на стандартній панелі інструментів, або дати команду *Вставка > Гіперпосилання*. При цьому відкривається діалогове вікно *Створити гіперпосилання* (або *Змінити гіперпосилання*). Це вікно містить три вкладки, призначені для створення різних видів посилань.

2. Якщо необхідно створити зовнішнє посилання на Web-сторінку, розташовану десь в Інтернеті, вибирають вкладку WWW. У полі *Тип гіперпосилання* вказують *протокол* використовуваний для звертання до посилання. У поле *Адреса (URL)* вводять повну *адресу URL* для документа, на який здійснюється посилання.

3. Якщо необхідно посилатися на документ HTML, що одночасно редагується за допомогою програми FrontPage Express, варто вибрати вкладку *Відкрити сторінки*. Документ, на який треба дати посилання, у цьому випадку вибирають у списку *Відкрити сторінки*. Оскільки документ є локальним, можна посилатися на місце всередині документа, позначене якорем посилання, обравши значення зі списку *Закладка*.

4. Якщо необхідно створити посилання на документ HTML, що ще не існує, використовується вкладка *Нова сторінка*. Для цього треба задати назву нової сторінки (вона буде поміщена в тег <TITLE>) і ім'я файла, у якому буде зберігатися новий документ. Створений документ буде негайно відкритий для редагування в новому вікні.

Гіпертекстові посилання, створені в редакторі, зображуються в документі точно так само, як у браузері: синім кольором і підкресленням. Для того, щоб перевірити чи працює створене посилання, можна клацнути на ньому правою кнопкою миші і вибрати в контекстному меню команду *Перейти по посиланню*.

6.5 Робота з текстом

Текст і шрифти мають величезне значення для візуального подання Web-сторінки.

Базові текстові атрибути мови HTML, параметри шрифту, кольору, розміру, товщини:

Шрифт (Font)	Загальний вигляд тексту. Існує три типи шрифтів: із засічками (serif), без засічок (sans serif) і моноширинний (monospace). У межах кожного типу є багато різновидів шрифтів. Усі вони будуть коротко розглянуті нижче.
Колір (Color)	Символи, слова, фрази, речення, абзаци і цілі Web-сторінки можна оформлювати в різні кольори.
Розмір (Size)	На відміну від друкарських видань, розмір шрифту на HTML-сторінці вибирається з ряду фіксованих значень.
Товщина (Weight)	Існує два варіанти товщини шрифту для Web-сторінки: звичайний і напівжирний.
Накреслення (Style)	Існує декілька варіантів накреслення тексту для Web-сторінки: звичайний, <i>курсив</i> , <u>підкреслений</u> , ~~перекреслений~~ і з фіксованим форматуванням (моноширинний шрифт).

Шрифт. Кількість шрифтів на Web-сторінках найчастіше дуже обмежена. Перші браузери використовували набір стандартних шрифтів, і в розробника просто не було вибору. Сучасні браузери можуть змінювати шрифти, але деякі проблеми все ще залишаються актуальними.

Особливості застосування шрифтів при виведенні тексту. Текст – це букви, слова, фрази, пропозиції й абзаци, розташовані на сторінці. Шрифт – це те, чим текст відображається на сторінці, екрані телевізора, рекламному щиті, доларовій банкноті або будь-якому іншому носії інформації. Дуже важливо правильно вибрати шрифт.

1. Отут головне – відбити призначення виведеної інформації, тобто він повинен відповідати призначенню Web-сторінки і потребам її читачів.

Разом із можливістю підтримки різноманітних шрифтів Web-оглядачі

успадкували від настільних видавничих систем їхні шрифтові проблеми. Головна з них полягає в такому: якщо якогось шрифту немає, його не можна побачити на екрані. Сторінка, створена в Windows 95, може прекрасно виглядати на іншому комп'ютері, що також працює під керуванням Windows 95, тільки в тому випадку, якщо на цьому комп'ютері встановлені потрібні шрифти. Для користувачів системи UNIX шанси побачити потрібний шрифт різко падають.

2. Тому, ніколи не можна розраховувати на те, що на конкретній машині знайдеться визначений шрифт. Якщо ж користувачі використовують операційну систему, відмінну від вашої, проблема шрифтів може виникнути з великим ступенем ймовірності.

Звичайно, якщо текст використовується як частина графічного зображення, він перетворюється в набір пікселів і з рівним успіхом відтвориться на будь-якому комп'ютері, незалежно від використовуваного шрифту. Але якщо шрифт застосовується до звичайного тексту на Web-сторінці, усе стає набагато складнішим. Для відображення цього тексту потрібним шрифтом необхідно, щоб цей шрифт був встановлений на комп'ютері читача Web-сторінки. У іншому випадку текст буде відформатований шрифтом, використовуваним за замовчуванням.

3. Тому найкраще використовувати ті шрифти, що явно є на комп'ютерах вашої аудиторії. Найчастіше це шрифт, використовуваний у визначеному броузері за замовчуванням. Для виділення особливо значущих текстових фрагментів перетворіть усі нестандартні шрифти тексту у графічне зображення.

4. Стандартні шрифти Windows – це Arial, Comic Sans, Courier New, MS Sans Serif і Times New Roman. Всі ці шрифти є представниками таких трьох основних сімейств шрифтів.

- *З засічками (Serif)* – шрифти з маленькими поперечними рисочками на кінцях елементів кожної букви, як у шрифті Times New Roman.
- *Без засічок (Sans Serif)* – шрифти без засічок, такі як MS Sans Serif і Arial
- *Моноширинний (Fixed-width)* – шрифти, усі букви яких мають однакову ширину, таку як Courier New. (Шрифти Times New Roman і Arial називаються пропорційними, оскільки букви мають різну ширину.)

Карти стилів. Вигляд всіх карт стилів дозволяє визначити зовнішній вигляд Web-сторінки і прискорити імпортування на неї визначених стилів. Іншими словами, зовнішній елементів Web-сторінки можна настроїти за своїм смаком, і цей загальний тон буде зберігатися при переході від однієї сторінки до іншої.

Офіційно такий спосіб називається *каскадними картами стилів* (*cascading style sheets*). Стили з майстра-карти стилів успадковуються окремими Web-сторінками, що на них посилаються. Існує три основних способи використання ієрархії стилів: карти стилів, блоки стилів (*style blocks*) і вмонтовані стилі (*inline styles*).

Карта стилів – це не що інше як Web-сторінка, що містить усі визначення стилю, легко і швидко імпортовані на інші Web-сторінки.

Вищий рівень ієрархії – це Web-сторінка, що містить стилі. Насправді це не зовсім Web-сторінка. Текст, що міститься на ній, призначений лише для ілюстрації різноманітних стилів. Файл такої сторінки має розширення .CSS, а не .HTML або .HTM. Приклад змісту подібного файла показаний у такому листингу:

Лістинг. Типова карта стилів

```
<STYLE TYPE="text/css">
BODY {font: 10pt "Arial"}
H1    {font: 19pt/21pt "Comic Sans MS";
font-weight: bold;
Color: red}
H2    {font: 15pt/17pt "ComicSansMS";
font-weight: bold;
color: blue}
P      {font: 10pt/12pt
"Arial";
color: black}
</STYLE>
```

Для того щоб імпортувати карту стилів на Web-сторінку, потрібно додати в HTML-код у заголовку сторінки дескриптор LINK:

```
<LINK REL=STYLESHEET
HREF="http://www._____ .htm"
TYPE="text/css">
```

У файлі карти стилів можуть міститися такі параметри стилю як розмір, гарнітура, товщина, міжрядковий інтервал, кольори, розміри нулів і т.п.

Розмір шрифту

Атрибут *font-size* визначає розмір шрифту в пунктах, дюймах, сантиметрах або пікселях, наприклад:

```
{font-size: 12pt}
```

Назва шрифту

Атрибут *font-family* визначає назву гарнітури, використовуваної в тексті. Назва однієї гарнітури визначається так:

{font-family: Arial}

Можна визначити назви декількох гарнітур, розділивши їх комами:

{font-family: Arial, Helvetica)

Вказавши наприкінці списку назву сімейства шрифтів, можна більш точно визначити необхідний вид шрифту

{font-family: Arial, Helvetica, sans-serif)

Блоки стилів. Блоки стилів є проміжною ланкою в каскадній ієрархії. Вони з'являються на окремих Web-сторінках і відміняють для даної сторінки будь-які визначення стилів з імпортованої карти. Блоки стилів об'являються за допомогою дескриптора STYLE, наприклад

```
<STYLE TYPE ="text/css">
```

```
H1 {font: 18pt "Times";Color: "#FF00FF"}
```

```
</STYLE>
```

Параметр TYPE використовується для визначення каскадної карти стилів на Web-сторінці. Це визначення відміняє визначення в карті стилю, якщо таке існує.

Вмонтовані стилі – це найнижчий рівень ієрархії. Вони можуть розміщатися в одному або декількох місцях Web-сторінки і призначені для зміни стилю окремого текстового фрагмента. Ключове СЛОВО STYLE у цьому випадку є не дескриптором, а параметром іншого дескриптора.

```
<H1 STYLE="font-weight: bold ;color: blue ">
```

6.6 Створення таблиць і форм

Більш складні елементи Web-сторінок не вдається створити методом простого форматування текстового документа. Зокрема, у програмі FrontPage Express є спеціальні засоби для створення таблиць і форм.

Для створення таблиць використовують кнопку *Вставити таблицю* або команду *Таблиця > Вставити таблицю*. При використанні кнопки *Вставити таблицю* число стовпців і рядків вибирають у палітрі, що відкрилась, її можна «розтягнути» за межі початкового розміру (5x4). При використанні команди число рядків і стовпців задають у діалоговому вікні *Додати таблицю*. Подальша робота з таблицею аналогічна роботі з таблицями в програмі Word. Коли курсор знаходиться в одній з комірок, доступні пункти меню *Таблиця*, використовувані для зміни зовнішнього вигляду таблиці в цілому. Вводити, редагувати і форматовувати дані в комірках можна незалежно один від одного.

Редактор FrontPage Express дозволяє створювати форми. Форма – це особливий вид документа HTML, що дозволяє одержувати інформацію від читача. Ця інформація може доставлятися автору електронною

поштою або за допомогою спеціальних програм, що працюють на Web-серверах.

Основний зміст форми складають поля чи форми. Мова HTML підтримує шість видів полів форми: прапорці, перемикачі, текстові поля, текстові області, що розкриваються, списки і кнопки. Одна з кнопок форми повинна бути присутня обов'язково – це кнопка відправлення форми.

Створення форми виконують заданням її поля. Для цього використовують панель інструментів *Поля форми* або меню *Вставка>Поле форми*. Після вставки першого поля, частина документа, що включає це поле, вміщується в пунктирну рамку. Зміст цієї рамки і складає власне форму. При додаванні додаткових полів необхідно стежити за тим, щоб вони також були всередині цієї рамки, інакше документ буде містити кілька форм. Це не заборонено, але при відправленні такої «форми» може бути передана не вся потрібна інформація.

6.7 Елементи графічного дизайну

У Web є два рівні графічного дизайну: розробка окремих зображень і розробка цілих сторінок. Гарна Web-сторінка – це не просто набір гарних зображень. Зображення повинні гармонійно сполучатися один з одним.

На рівні окремих зображень необхідний організований, функціональний дизайн і, звичайно, всякий рисунок повинен відповідати своєму призначенню. Наприклад, зображення, що буде відігравати роль кнопки або посилання на іншу Web-сторінку, повинно не тільки добре виглядати, але і виконувати свої функції. На рівні проектування сторінки зображення повинні гармонійно взаємодіяти і підпорядковуватися загальному задуму сторінки.

6.8 Створення зображення

Існує декілька основних правил розробки зображень.

- зображення розбивається на три частини;
- складові частини зображення розташовуються по діагоналі;
- основне зображення повинно знаходитися в центрі;
- фон – це фон, передній план – це передній план, і в кожного з них своя роль;
- форма повинна відповідати змісту.

Діагоналі. Діагоналі зображення створюються елементами, розташованими вздовж діагональної лінії. Такі діагоналі взаємно посилюють одна одну і допомагають надати зображенню завершеності

форми. Схрещені діагоналі підкреслюють конфлікт або підвищують енергійність зображення. Незалежно від того, якими є діагоналі (паралельними або пересічними), вони визначають зорову структуру образу. Щоб побачити діагоналі на зображенні потрібно зосередитись на засобах розташування об'єктів і ліній.

Фокус у центрі. Не у всякого зображення фокус повинен знаходитися в центрі, але велика частина графіки в Web підпорядковується цьому правилу.

В основному, зображення в Web несуть функціональне навантаження. Тому фокусом може бути список слів, що є посиланнями на інші сторінки, або зображення кукурудзяного качану, що грає роль кнопки. Те, що розташовано в центрі зображення, завжди сприймається як найголовніша його частина.

Передній і задній план. Майже в кожного зображення є передній план і фон. Завданням фону є привертання уваги до переднього плану. Фон, що відволікає увагу, – найгірша з можливих помилок. Перевантажений фон є однією з найістотніших причин дезорієнтації користувача в сучасних Web-сторінках.

Існує безліч засобів змусити фон виконувати свою задачу. Його можна зробити темнішим, світлішим, розмити і т.п. Підійде будь-який спосіб, що запобігає конкуренції між переднім планом і фоном.

Головне тут – встановити правильні співвідношення між переднім планом і фоном. Потрібно зробити фон менш помітним і сильніше виділити передній план. Для створення гарного фону можете скористатися такими прийомами:

- використання дуже світлих або дуже темних відтінків;
- розмивання фону;
- м'які кольори;
- плавні переходи;
- низька контрастність (у деяких випадках, коли розходження у відтінках темного і світлого невелике);
- приємні кольори, наприклад зелений або блакитний;
- великі площі, заповнені кольором.

Для вдалого створення об'єктів переднього плану використовують протилежні прийоми:

- відтінки, середні між темним і світлим;
- яскраві кольори;
- різкі межі;
- сильні контрасти;
- різкі кольори: червоний або жовтий;

- відтінки переднього плану повинні протиставлятися відтінкам фону: темні – на світлому фоні, світлі – на темному.

Комбінована графіка

Всі існуючі сторінки в Web можна віднести до сторінок трьох типів:

- * початкову сторінку (Home page);
- * описову сторінку (Detail page);
- * сторінку відгуків (Feedback page).

Кожний із перерахованих типів Web-сторінок виконує свої функції і висуває до розробників свої вимоги.

Доцільно починати розробку сторінки з нарису на папері. При цьому необхідно звернути увагу на такі моменти:

- * функціональне призначення сторінки (продати виріб, висловити ідею, розважити читача і т.п.);
- * відповідність сторінки загальному духу Web-вузла;
- * звичайні елементи цієї сторінки повинні використовуватися й іншими сторінками, наприклад навігаційні панелі;
- * гіперпосилання на сторінку;
- * необхідна графіка;
- * зручне розміщення всього цього на сторінці.

Початкова сторінка

Сформулюємо вимоги до сторінки.

Ця сторінка є лицем Web-вузла компанії, тому їй необхідна першокласна графіка і чіткий дизайн, що відображає просту рекламу компанії.

Задача сторінки.

Піднести імідж компанії, що виражається гаслом. Забезпечити посилання на інші сторінки компанії.

Призначення сторінки.

Обличчя Web-вузла.

Звичайні елементи.

На лівій стороні – навігаційна панель у вигляді великого логотипа компанії.

Посилання.

Гіперпосилання на основні сторінки вузла: виробництво вермішелі, засоби визначення її якості, замовлення на вермішель, технічна підтримка.

Графіка.

Логотип, навігаційна панель і по одній кнопці на кожне з основних посилань.

Описова сторінка

Сторінка ілюструє деталі кімнати в готелі Howard House – прекрасному і дорогому готелі на березі Верхнього озера.

Сформулюємо вимоги до сторінки.

<i>Задача сторінки.</i>	Показати, що номер Бергмана в готелі Howard House — дійсно першокласні апартаменти, що виправдовують свою непомірно високу ціну.
<i>Призначення сторінки.</i>	Це одна із семи сторінок вузла, присвячених кімнатам.
<i>Звичайні елементи.</i>	Емблема Howard House.
<i>Посилання.</i>	На початкову сторінку.
<i>Графіка.</i>	Емблема і чотири фотографії деталей інтер'єра.

Сторінка відгуків

На цій сторінці компанія дає можливість споживачам висловити свої побажання і пропозиції.

Вимоги до сторінки

<i>Задача сторінки.</i>	Анкета для побажань і пропозицій споживачів по ключових питаннях.
<i>Призначення сторінки.</i>	Єдина сторінка відгуків на Web-вузлі охоплює три теми.
<i>Звичайні елементи.</i>	Логотип компанії, фоновий рисунок.
<i>Посилання.</i>	Перехід на початкову сторінку і кнопки для передачі і поновлення даних анкети.
<i>Графіка.</i>	Логотип, тіло, дотепна піктограма вгорі сторінки. Піктограма являє собою збільшений варіант кнопки для переходу на цю сторінку.

6.9 Майстри і шаблони

Для початківців, що створюють Web-сторінки, редактор FrontPage Express надає в розпорядження користувача набір майстрів і шаблонів. За командою *Файл>Створити* на екрані з'являється діалогове вікно Нова сторінка, що дозволяє вибрати майстра або шаблон для автоматичного створення документа HTML.

Майстри (Майстер персональних основних сторінок і Майстер форм) автоматизують створення документа, задаючи питання і використовуючи відповіді користувача для формування документа. Кожний етап роботи майстра складається з відповіді на питання, що визначають зміст Web-сторінки. Перехід до такого етапу здійснюється клацанням на кнопки *Далі*. Після введення всієї необхідної інформації варто клацнути на кнопку *Готово*. Документ формується автоматично відповідно до отриманих відповідей.

Зрозуміло, майстри створюють тільки структуру документа відповідно до вимог користувача. Як правило, такий документ містить лише заголовки розділів, а самі розділи доводиться все одно заповнювати вручну. У текст підготовленого документа майстер має включити коментарі для подальшої самостійної роботи над Web-сторінкою.

Шаблони використовують інакше. Кожний шаблон являє собою «готовий», заздалегідь сформований документ, який, проте, не можна використовувати негайно. Фактично, шаблон являє собою детально продуману структуру документа, що не містить інформації, специфічної для конкретного користувача. Цей документ необхідно відредагувати, додаючи і редагуючи розділи й окремі пункти і змінюючи формулювання так, як потрібно.

Всі шаблони починаються з коментаря, що докладно описує, що треба зробити для успішного настроювання сторінки. При редагуванні шаблону створюється новий документ, а сам вихідний шаблон не змінюється, так що його можна використовувати і падалі.

6.10 Використання готових матеріалів

Створення Web-сторінки рідко починається з чистого листа. Найчастіше заздалегідь готуються тексти, документи, малюнки, звукові файли, відеозаписи та інші об'єкти, що потрібно розмістити на Web-сторінці. Але, наприклад, щоб використовувати документ Word, його потрібно перетворити у формат HTML. Редактор FrontPage Express вирішує цю задачу автоматично.

1. Щоб помістити на Web-сторінку готовий документ, треба дати команду *Вставка > Файл*. У діалоговому вікні *Вибрати файл* можна зазначити файл, що вставляється на Web-сторінку.

2. Список *Тип файлів* показує, що програма FrontPage Express дозволяє імпортувати файли HTML, текстові файли, файли, створені в різних текстових процесорах (наприклад, Word), і електронні таблиці Excel. Всі ці файли перетворюються у формат HTML автоматично.

3. Аналогічно вбудовують зображення. З декількох десятків видів існуючих графічних форматів в Інтернеті практично застосовують лише два. Формат GIF зручний для комп'ютерної графіки і рисунків; що містять області суцільного кольору. Формат JPEG використовують для збереження фотографій і рисунків, що містять плавні переходи кольорів.

4. Команда *Вставка > Зображення* або кнопка *Вставити зображення* відкривають діалогове вікно *Зображення*.

5. Клацнувши на кнопці *Огляд* і відкривши список *Тип файлів*, можна одержати список форматів файлів, які редактор FrontPage Express може автоматично перетворити у формати GIF або JPEG. При зберіганні документа програма запропонує також зберегти і перетворений файл.

6. Інші об'єкти, у тому числі мультимедійні файли (відеозапису і звукозапису) вставляють за допомогою меню *Вставка*. Для всіх об'єктів вибір потрібного файла на диску виконують після щиголя на кнопці *Огляд*. У таких випадках редактор не здійснює перетворення файла, а тільки записує в документ необхідні теги.

6.11 Зміна властивостей тегів HTML

Як ми вже знаємо, для зміни властивостей тегів і настроювання їхньої дії служать атрибути, що входять у теги. Проте при використанні редактора FrontPage Express ні до тегів, ні до атрибутів немає прямого доступу. Втім, зробити необхідне настроювання можна й іншим способом.

1. Припустимо, що в документ була введена горизонтальна розділювальна лінійка. Для цього використовують команду *Вставка » Горизонтальна лінія*. У мові HTML горизонтальну лінійку створює одиночний тег `<HR>`, що може містити атрибути, що задають довжину, товщину і зовнішній вигляд лінійки.

2. Щоб змінити властивості об'єкта HTML (у даному випадку лінійки), треба встановити на нього курсор (або виділити об'єкт) і дати команду *Виправлення > Властивості*. У даному випадку команда має вигляд *Властивості* горизонтальної лінії. Цю ж команду можна подати за допомогою контекстного меню або за допомогою клавіатурної комбінації ALT+ENTER.

3. Такою командою відкривається діалогове вікно властивостей, що містить параметри, які відносяться до даного елемента сторінки. Вигляд і склад цього діалогового вікна визначається тим, який саме елемент був обраний. Регульовані параметри відповідають доступним атрибутам тега, що задає даний елемент.

4. Після настроювання параметрів за допомогою цього діалогового вікна зовнішній вигляд елемента на екрані змінюється відповідно до зробленого настроювання.

5. Оскільки будь-який фрагмент документа так чи інакше знаходиться під керуванням якогось тега HTML, команду *Властивості* можна використовувати і незалежно від місцезнаходження курсору. У кожному випадку настроюються властивості потокового елемента нижнього рівня, яким би він не був. При роботі з текстом дане настроювання дозволяє внести зміни в шрифтове оформлення.

6. Якщо потрібно змінити властивості елемента більш високого рівня, варто скористатися контекстним меню. У ньому перераховуються всі «відкриті» на даний момент теги HTML. Обравши відповідний пункт, можна змінити властивості потрібного тега.

6.12 Правка коду HTML

Хоча для роботи з редактором FrontPage Express і не потрібно знання мови HTML, можливість внесення змін безпосередньо в код Web-сторінки корисна для ручного виправлення документа або для реалізації ідеї, що програмі не під силу.

1. Якщо безпосередньо в текст документа потрібно вручну ввести теги HTML, можна скористатися командою *Вставка > Розмітка HTML*. У діалоговому вікні *Розмітка HTML* можна ввести довільний фрагмент коду, що може містити як теги, так і текст. Текст, введений таким чином, не перевіряється на правильність програмою FrontPage Express і залишає документ у незмінному вигляді.

2. Наявність у документі вмонтованого користувачем коду HTML відображається на екрані у вигляді невеличкого жовтого значка. Для перевірки правильності цього коду (і всієї сторінки в цілому) документ необхідно зберегти і відкрити для перегляду в браузері.

3. Код HTML, що автоматично генерується програмою FrontPage Express, не завжди оптимальний і може потребувати ручного виправлення. Щоб зробити таке виправлення, не виходячи з програми, варто дати команду *Вид > HTML*. За цією командою код документа відображається в діалоговому вікні *Перегляд або Виправлення HTML*. Тут він доступний для редагування.

Для наочності різні елементи коду виділяються різним кольором. Теги зображуються фіолетовим кольором, ключове слово атрибутів — червоним, значення атрибутів — синім. Це діалогове вікно можна розглядати як текстовий редактор, вмонтований у програму FrontPage Express, що дозволяє формувати документи HTML вручну.

4. Зміни вносяться в документ кладанням на кнопки *OK*, при чому редактор FrontPage Express залишає за собою право на остаточне редагування.

6.13 Публікація Web-сторінки

За допомогою редактора FrontPage Express можна за лічені хвилини створити найпростішу Web-сторінку, особливо якщо скористатися шаблоном. Коли така сторінка створена, постає питання про її публікацію.

Публікацію виконують розміщенням документа (разом з усіма супровідними файлами, наприклад рисунками) в одному із каталогів на жорсткому диску Web-серверу, що входять в структуру Інтернету. Залишається тільки вирішити три питання:

- де знайти Web-сервер, постійно залучений до Інтернету?
- як розмістити на ньому створені файли?
- як оплачувати дані послуги?

1. Web-сервер довго шукати не треба, оскільки він і так уже є в кожного, хто працює в Інтернеті. Більшість користувачів виходять в Інтернет шляхом підключення до серверу свого *постачальника послуг Інтернету (сервіс-провайдера)*. До нього і треба звернутися з проханням про надання місця на жорсткому диску для публікації Web-сторінки.

2. Постачальник послуг Інтернету знає, як правильно розмістити ваші файли на сервері. Питання зводиться тільки до того, як ці дані передати. Можна явитися особисто і передати гнучкий диск із комплектом файлів. Можна відправити дані через мережу, але для цього треба вміти працювати з протоколом *FTP*, за допомогою якого в мережі пересилаються файли. Необхідну консультацію і програмне забезпечення повинен надати постачальник послуг. Можна подумати про придбання й освоєння повного пакета FrontPage 97, у складі якого є зручні засоби для публікації Web-документів.

3. Оплата послуг з розміщення Web-вузлів і Web-сторінок відбувається на договірних умовах. Вартість залежить від обсягу матеріалу і додаткових послуг. У даний час прийнято розміщувати окремі Web-сторінки клієнтів безоплатно, але обмежувати їхній розмір (наприклад, не більш 100 Кбайт). З цього і прийнято починати видавничу діяльність у Всесвітній Мережі.

Контрольні запитання

1. Яке призначення мови HTML?
2. Що таке теги HTML? Назвіть їх основні види.
3. Яку структуру має документ HTML?
4. Що таке гіпертекстове посилання та яке його призначення?
5. Які елементи графічного дизайну Вам відомі?
6. Як створюються таблиці за допомогою мови HTML?
7. Що таке фрейми? Як вони описуються засобами мови HTML?
8. Як відбувається публікація Web-сторінки?

7 Анімація. Види анімації

7.1 Теорія анімації

Анімація — це серія послідовно відображуваних графічних зображень. В анімації кожне зображення називається кадром. Анімаційне зображення може складатися з довільної кількості елементарних кадрів.

Існує безліч засобів подання анімації і ще більше засобів її створення. Наповнити Web-сторінку анімаційними ефектами можна за допомогою файлів формату GIF або шляхом перетворення послідовності зображень у відеокліп і його наступного розміщення на сторінці. Найпростіший метод — створити послідовність файлів GIF-формату. На ринку існують кілька програмних пакетів для створення анімації. Існує два основні анімаційні методи: покадрова анімація і інтерполяція.

7.2 Анімація об'єкта

Додаток PhotoImpact GIF Animator дозволяє не тільки створити покадрову анімацію, але й оживити окреме зображення. Нижче ви дізнаєтеся, як можна перемістити зображення м'яча від однієї межі кадру до іншої.

Як приклад створимо в редакторі Adobe Photoshop зображення м'яча. Почавши з нового зображення розміром 16x16 пікселів (команда File(New)), потім виділіть коло розміром 15x15 пікселів і заповніть його радіальною градієнтою кольоровою заливкою від білого до червоного. Спеціально створюємо об'єкт, що на один піксел менший розміру всього зображення, щоб залишити ореол навколо м'яча.

Щоб оживити м'яч за допомогою додатка PhotoImpact GIF Animator, потрібно виконати такі дії:

1. Найпростіший спосіб завантажити зображення кілька разів — клацніть по кнопці Add an Image.
2. Встановіть значення параметра X-offset для іншого зображення рівного 5 пікселям.
3. Встановіть значення параметра X-offset для зображень, що залишилися, із збільшенням у 5 пікселів: 10, 15 і 20 пікселів.
4. Встановіть час затримки для шкіряного зображення рівним 0,12 і збережіть файл під ім'ям ball.gif.

Встановлюючи негативні значення параметра X-offset, можна змішувати зображення вліво.

Параметр Y-offset надає можливість переміщати зображення вгору й вниз.

Старанно підбираючи значення параметрів X-offset і Y-offset, зображення можна перемішувати по колу під кутом і т.д.

Задаючи різні інтервали затримки для зображень, можна створювати ілюзію їх прискореного або уповільненого руху.

Використовуйте оптимізуючі алгоритми формату GIF (команда File⇒GIF Optimization) для зменшення розміру файла шляхом виключення повторюваних пікселів із наступних анімаційних кадрів.

7.3 Покадрова анімація. PhotoImpact GIF Animator. Macromedia Flash

Щоб створити покадрову анімацію, варто окремо нарисувати кожен кадр послідовності. Більшість анімаційних додатків дозволяє рисувати поверх затіненої версії попереднього кадру. У результаті дизайнер одержує можливість точно нарисувати новий кадр. Метод просвічування одного і більше попередніх кадрів під час створення потокового кадру називають *цибулинням* (*onionskin*). Ця назва пішла від різновиду кальки, яку використовували художники-мультиплікатори, які працювали із папером і олівцем. Викресливши зображення на "цибулинні", мультиплікатор міг точно нарисувати кожен кадр анімаційної послідовності. Нижче буде розглянуто спосіб створення покадрової анімації методом "цибулиння".

Кращий спосіб навчитися створювати анімаційні ефекти – виконувати все на практиці. Наведемо нижче приклад, який ілюструє спосіб створення анімаційних послідовностей у додатку Fractal Design Painter. Сформульовані принципи будуть справедливими і для інших анімаційних програм. Існує кілька додатків, призначених для перетворення послідовностей пронумерованих файлів (або відеокліпів, або інших форм анімації) в анімаційні файли формату GIF. Кращим із них є PhotoImpact GIF Animator, створений компанією Ulead.

1. Відкрийте додаток PhotoImpact GIF Animator.
2. Завантажте пронумерований файл із зображенням із розширенням bmp.
3. Виберіть файли, що варто відкрити.
4. Клацніть на кнопці Insert. Для зображень із першого по шостий параметри завантаження злегка відрізняються. З'явилося два додаткових прапорці. Якщо встановити верхній (Apply above options to other images), задані установки будуть справедливими для всіх додаткових зображень. Якщо встановити нижній прапорець, всі установки будуть застосовані автоматично і всі зображення завантажатимуться без запитань.
5. Встановіть обидва прапорці, а потім клацніть на кнопці Insert. У результаті відкриється вікно додатка GIF Animator, у якому будуть перераховані шість зображень.

6. Щоб переглянути зображення, клацніть на відповідному файлі в списку. Назву файла буде виділено. Час затримки вікна(Delay) (інтервалу, протягом якого кадр буде відображений на екрані) за замовчуванням дорівнює 0,5 с. Із точки зору анімації це достатньо великий проміжок часу, щоб змінити час затримки для всіх кадрів, по черзі виділіть кожен кадр і вручну задайте затримку рівну 12 (0,12 с).
7. Встановіть прапорець Delay і виберіть перемикач Apply to all images, після чого клацніть на кнопці OK. Нове значення затримки буде використовуватися для всіх зображень.
8. За допомогою команди меню File (Save As) збережіть послідовність у форматі GIF і присвойте їй ім'я.

PhotoImpact GIF Animator дозволяє створювати перехідні ефекти, що можна використовувати під час заміни одного зображення іншим. Дизайнер одержує можливість створювати за допомогою анімації формат GIF — подобу презентації слайдів. Для доступу до цієї функціональної можливості використовуйте команду Layer(Add Transition).

PhotoImpact GIF Animator можна використовувати для зберігання анімації формату GIF у вигляді пронумерованих файлів. Засіб зберігання кадрів анімації визначається перемикачем Export Images панелі інструментів.

Група параметрів How to remove визначає спосіб переходу до такого зображення. За замовчуванням це рішення приймає оглядач, але для деяких типів анімації може виникнути необхідність використовувати інші параметри. Наприклад, якщо пікселі попередніх кадрів не віддаляться коректним способом, замість визначеного за замовчуванням режиму скористайтеся To Background Color.

Щоб видалити фоновий колір із послідовності кадрів, можна використовувати прозорість. При необхідності можна задати різноманіття фонового кольору для кожного кадру.

У процесі створення покадрової анімації нове зображення рисується для кожного кадру. У більшості випадків зображення останнього кадру лише злегка змінюється. За замовчуванням додаток Flash відображає вміст останнього кадру в потоковому — задача створення нового кадру зводиться до редагування існуючого зображення.

Варто відразу зауважити, що для створення зображень у додатку Flash прийдеться прикладати мінімум зусиль. Лінія, проведена за допомогою інструмента Pin або Brush, жорстко не зафіксована. Вам надана можливість повертати, скручувати, розтягувати і стискувати форму лінії довільним способом. Значимість цієї можливості неможливо переоцінити, у чому ви зможете переконатися на власному досвіді.

Після запуску у вікні Flash повинен бути завантажений порожній документ. Якщо цього не відбулося, створіть його за допомогою команди File(New). Потім змініть розмір документа.

7.4 Інтерполяційна анімація

Для створення інтерполяційної анімації необхідно розташовувати початкові і кінцеві кадри, з яких за допомогою відповідного програмного забезпечення можна буде генерувати проміжні кадри. Наприклад, можна створити початковий кадр із м'ячем у нижньому лівому кутку і кінцевий – із м'ячем у верхньому правому кутку. Багато анімаційних програмних пакетів спроможні генерувати проміжні кадри, що зображують м'яч, що пересувається з початкового положення в кінцеве. Нижче наведено приклад створення послідовності зображень методом інтерполяції.

Існує пакет створення анімації, функціональні можливості якого значно перевищують можливості програм типу Painter і GIF Animator. Він називається Macromedia Flash і містить у собі інструменти, розроблені спеціально для створення приголомшливих анімаційних ефектів.

Вікно програми складається із шести головних областей, що варто розглянути більш докладно перед тим, як намагатися щось зробити.

- *Головна панель інструментів.* Ці інструменти використовуються в різноманітних ситуаціях. Панелі належать кнопки зберігання і відкриття файла, скасування, повтору команди і т.д.
- *Інструменти рисунка.* Ці інструменти використовуються для рисунка в основному вікні програми.
- *Додаткові інструменти.* Набір інструментів цієї панелі змінюється в залежності від того, який інструмент рисунка обраний в даний момент.
- *Прошарки.* У цій області утримується список прошарків, а також кнопки для керування функціональними можливостями "цибуління".
- *Тимчасова лінія.* Визначає покадровий розподіл анімації, що знаходиться в кожному прошарку.
- *Вікно рисунка.* Відображає вміст потокового кадру.

Розмір анімаційного документа, визначений за замовчуванням, дорівнює 7,5х10 дюймів. Природно, що документ такого розміру є занадто великим для графічної частини Web-сторінки, тому його варто змінити.

1. Виконайте команду Format Document для того, щоб відкрити діалогове вікно.

2. Установіть ширину документа рівною чотирьом, а висоту – одному (дойму).
3. Для початку клацніть мишею на інструменті Pencil. Зверніть увагу на те, що після вибору цього інструмента змінилася область додаткових інструментів.
4. Клацніть на самому верхньому додатковому інструменті і виберіть необхідний режим рисунка.
5. Клацніть на інструменті Oval, після чого попередній інструмент буде замінений.
6. У вікні рисунка нарисуйте широкий овал. Під верхньою тимчасовою лінією (Uppercase Timeline) розташована панель, у якій містяться числа. Нижня тимчасова лінія – це білий рядок, розташований праворуч від шкірного прошарку. У початковий момент існує єдиний прошарок – Layer 1.
7. Створіть групу нових кадрів.
8. Відкрийте контекстне меню.
9. Клацніть на пункті меню Insert Frame. Програма Macromedia Flash установлює кадр не тільки в двадцяту позицію, але і кадри між першим кадром у першій позиції і новому кадру.
10. Виберіть овал.
11. Виконайте команду Insert(Create Symbol) і у відкритому діалоговому вікні наберіть ім'я oval у поле Name і клацніть на кнопці OK.
12. Клацніть правою кнопкою миші на тривимірному прямокутнику в двадцятому кадру.
13. Виберіть із меню пункт Make Key Frame.
14. Щоб активізувати двадцятий кадр, клацніть на числі 20 на тимчасовій лінії. Показчик потокового кадру переміститься до двадцятого кадру.
15. Виберіть із контекстного меню пункт Scale.
16. Перетягніть лівий або правий маркер виділення до середини кадру, а верхній або нижній маркер виділення – убік межі кадру.
17. Клацніть на першому кадру тимчасової лінії Layer 1.
18. Виберіть із меню пункт Twinning.
19. Виберіть пункт Motion.

Діалогове вікно змінює набір своїх параметрів. У результаті чого можна буде вибрати режим інтерполяції для обертання, масштабування (цей прапорець встановлений у даному прикладі) або обертання об'єкта по заданій траєкторії (у даному випадку цей прапорець скинутий).

Бігунок у нижній частині діалогового вікна управляє швидкістю інтерполяції на початку і в кінці процесу анімації. Якщо бігунок пересунути убік (Ease in), то інтерполяція почнеться повільно, а потім буде прискорюватись. Якщо ж бігунок пересунути вбік Ease out, то інтерполяція почнеться швидко, а потім буде сповільнюватися.

20. Виконайте процедуру інтерполяції. Стрілка, що з'явилась на тимчасовій лінії Layer 1 червоного кольору послужить індикатором ходу виконання інтерполяції.
21. Щоб перевірити точність інтерполяції, клацніть мишею на фрагменті тимчасової лінії, що відповідає кадру 10. Це половина відстані між двома ключовими кадрами. Овал повинен бути розташований посередині між початковим і кінцевим положеннями. Якщо овал не перетворюється у символ, то інтерполяція здійснюється неточно. У цьому випадку овал буде мати той самий розмір протягом 19 кадрів, а потім зненацька змінить розмір у 20-у кадрі. Якщо це відбудеться, почніть все спочатку і не забудьте перетворити овал у символ.
22. Щоб опротестувати анімаційний фрагмент, натисніть <Enter>. Ви повинні побачити овал, що зменшується.
23. Щоб зберегти результати роботи для подальшого редагування, виконаєте команду File(Save As).
24. Щоб зберегти анімацію у форматі Shockwave Flash (.SPL), що дозволяє розміщати анімацію на Web-сторінках, виконайте команду File(Export Movie) і погодьтеся з запропонованим за замовчуванням форматом для збереження (FutureSplash Player).
25. Привласніть файлу ім'я (наприклад, oval) і клацніть на кнопці Save.
26. Цей анімаційний фрагмент має можливість відображати тільки ті оглядачі, що використовують доповнення Shockwave Flash.
27. Щоб зберегти фрагмент у форматі GIF, виконайте команду File(Export Movie) і виберіть у списку Save as Type фільтр Animated GIF.

Інтерполяційний метод анімації може бути застосований до будь-яких об'єктів, що можна перетворити в символи.

- * Щоб застосувати метод інтерполяції до тексту, його необовязково перетворювати в символ.
- * Можна задати траєкторію здійснення інтерполяції, параметри інтерполяції для прошарку. Клацніть правою кнопкою миші на цьому прошарку і виберіть в контекстному меню пункт Add Motion Guide. Під поточним прошарком з'явиться новий. Щоб задати траєкторію руху, скористайтесь інструментом Pen. На вихідному прошарку перемістіть крапку закріплення символного об'єкта таким чином, щоб вона *точно* збігалася з кінцевою точкою траєкторії. Цього легко домогтися, включивши режим прив'язки до сітки (цей режим установлено за замовчуванням, якщо ви його не відключили). Об'єкт повинен знаходитися точно в кінцевій точці і на початковому і в кінцевих кадрах вихідної інтерполяції. Натисніть <Enter>, щоб перевірити коректність здійснення інтерполяції вздовж траєкторії.

Контрольні запитання

1. Як створити та розмістити на графічних зображеннях елементи керування Windows 98 за допомогою засобів графічних редакторів Paint Shop Pro та Adobe Photoshop?
2. Які візуальні ефекти створюються засобами Adobe PhotoShop для надання графічним зображенням певного вигляду ви знаєте?
3. Що таке оптимізація графічного файлу?
4. Яке призначення програми GIF PhotoImpact SmartSaver?
5. Яке призначення програми JPEG PhotoImpact SmartSaver?
6. Що таке анімація? Які види анімації ви знаєте?
7. Які особливості має анімаційна програма?
8. Як створити елементарні анімаційні ефекти за допомогою програми PhotoImpact GIF Animator

ЛІТЕРАТУРА

1. Водески Рон. Графика для Web. Библия дизайнера. – К.: Диалектика, 1998. – 608 с.
2. Бурлак М. Самоучитель по компьютерной графике. – К.: Издательская группа ВН, 1999. – 640 с.
3. Аммерал Л. Интерактивная трехмерная машинная графика. – М.: 1992. – 317 с. (Серия «Машинная графика языком Си»)
4. Аммерал Л. Машинная графика на персональных компьютерах. – М.: 1992. – 232 с. (Серия «Машинная графика языком Си»)
5. Аммерал Л. Принципы программирования в машинной графике. – М.: 1992. – 224 с. (Серия «Машинная графика языком Си»)
6. Аджиев В.Д. и др. Машинная геометрия и графика // Математика и кибернетика. – 1990. – №10.
7. Прокофьев Б.П., Сухарев Н.Н., Храмов Ю.Е. Графические средства Turbo C и Turbo C++. /Под ред. Г.В. Генса, Ю.Е. Храмова. – М.: Финансы и статистика, 1992. – 158 с.
8. Загляднов И.Ю., Касаткин В.Н. Построение изображений на экране персональной ЭВМ. – К.: Техника, 1990. – 120 с.
9. Алиев В.Э. Обработка графической информации на ПЭВМ. – М.: МФТИ, 1997. – 506 с.
10. Вальвачев А.Н. Графическое программирование на языке Паскаль: Справ. пособие. – Минск: Выш. шк., 1992. – 143 с.
11. Григорьев В.Л. ВIDEOSИСТЕМЫ ПК фирмы IBM. – М.: Радио и связь, 1993. – 192 с.
12. Корриган Дж. Компьютерная графика: Секреты и решения: Пер с англ. – М.: Энтрон, 1995. – 352 с.
13. Шикин Е.В., Боресков А.В. Компьютерная графика. Динамика, реалистические изображения. – М.: «Диалог-МИФИ», 1995. – 288 с.
14. Дубовий В.М., Кветний Р.Н. Основи застосування ЕОМ в інженерній діяльності. – К.: ІСД МО України, 1994. – 285 с.
15. Богумирский Б. Эффективная работа на IBM PC в среде Windows_95. – СПб.: Питер, 1997. – 1120 с.
16. Богумирский Б. Энциклопедия Windows_98. – СПб.: Питер, 1998. – 816 с.
17. Г. Корн, Т. Корн. Справочник по математике для научных работников и инженеров. – М.: Наука, 1973. – 832 с.

Навчальне видання

Володимир Юрійович Коцюбинський
Владислав Володимирович Кабачій

КОМП'ЮТЕРНА ГРАФІКА

Оригінал-макет підготовлено авторами

Редактор С. А. Малішевська

Навчально-методичний відділ ВДТУ

Свідоцтво Деркомінформу України

серія ДК № 746 від 25.12.2001

21021, м. Вінниця, Хмельницьке шосе, 95, ВДТУ

Підписано до друку 15.04.03 Гарнітура Times New Roman

Формат 29,7x42 1/4

Папір офсетний

Друк різнографічний

Ум. друк. арк. 3.61

Тираж 85 прим.

Зам. № 2003 - 060

Віддруковано в комп'ютерному інформаційно-видавничому центрі
Вінницького державного технічного університету

Свідоцтво Деркомінформу України

серія ДК № 746 від 25.12.2001

21021, м. Вінниця, Хмельницьке шосе, 95