

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

**Методичні рекомендації
до самостійної роботи
з навчальної дисципліни**

"КОМП'ЮТЕРНА АНІМАЦІЯ"

**для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
спеціалізації "Технології електронних мультимедійних видань"
усіх форм навчання**

Харків. Вид. ХНЕУ, 2012

Затверджено на засіданні кафедри комп'ютерних систем і технологій.
Протокол № 2 від 04.10.2011 р.

Укладачі: Євсєєв О. С.
Прибиткова Н. І.

М54 Методичні рекомендації до самостійної роботи з навчальної дисципліни "Комп'ютерна анімація" для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" спеціалізації "Технології електронних мультимедійних видань" усіх форм навчання / укл. Євсєєв О. С., Прибиткова Н. І. – Х. : Вид. ХНЕУ, 2012. – 48 с. (Укр. мов.)

Наведено методи, засоби та завдання для організації самостійного вивчення й поглиблення отриманих у рамках лекційного курсу і аудиторних лабораторних робіт знань, вмінь та навичок. Запропоновано різні види виконання самостійної роботи, серед яких поглиблення навичок, отриманих на лабораторних роботах, підготовка та проведення семінарів, написання есе.

Рекомендовано для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" спеціалізації "Технології електронних мультимедійних видань" усіх форм навчання.

Вступ

Метою самостійної роботи з навчальної дисципліни є поглиблення знань, які було отримано на лекційних заняттях, та підтвердження і реалізація навичок, що були сформовані на практичних заняттях для розробки анімаційних роликів з використанням творчого підходу та додатків з елементами анімації. Важливим завданням є формування компетенцій, що дозволяють студенту реалізовувати на практиці отримання знання, та розробляти такі анімовані документи та додатки, що є актуальними в сучасних умовах, мають попит на ринку веб-послуг на цей час та потенціал щодо подальшого розвитку.

Така робота потребує від студентів інтенсивної самостійної роботи над спеціальними інформаційними джерелами та інструментальними засобами, що дозволяють створювати анімацію з використанням творчого підходу та врахування усіх новітніх прийомів. Вивчення таких самостійних робіт потребує від студентів також поглиблення знань у суміжних дисциплінах, таких як прикладна математика, основи композиції та дизайну, основи об'єктно-орієнтованого програмування. Рекомендується працювати з періодичними мультимедійними виданнями, які відбивають тенденції розвитку сучасних інформаційних технологій, сучасні технології в анімації, дизайні й програмуванні.

Основні види самостійної роботи, які запропоновані студентам:

1. Поглиблене вивчення лекційного матеріалу.
2. Аналіз інформаційних джерел та агрегування даних.
3. Створення власних анімаційних мультимедійних додатків.
4. Інтеграція власних проектів із тими, що були виконані на лабораторних заняттях.

Змістовний модуль I. Створення комп'ютерної анімації за допомогою технології Macromedia Flash

Самостійна робота № 1

Використання інструментів роботи з кольором та текстом

Мета роботи: отримання знань та навичок щодо роботи із палітрою колірних зразків "Kuler".

Об'єктом СР є анімаційний додаток, що потребує гармонійного колірної виконання.

Предметом СР є процес створення колірної гама для анімованого додатка за допомоги палітри колірних зразків "Kuler".

Методи для виконання СР: аналіз та синтез, планування, розроблення контенту.

У результаті виконання СР формуються такі компетенції: створення власних колірних палітр та виконання операцій із ними, використання колірних палітр при роботі з графічними та текстовими об'єктами.

Нормування та вихідний контроль: самостійна робота виконується в рамках вивчення теми № 3 "Основні інструменти Macromedia Flash MX"; час необхідний для виконання – 4 години; результат роботи – звіт щодо самостіно створеного анімаційного проекту.

Контроль якості: диференційований.

Методичні рекомендації

Починаючи з версії CS4 до складу програми Adobe Flash входить палітра роботи із колірними зразками "Kuler" (рис. 1.1) яка потрібна для того, щоб підібрати відповідні кольори для графічних зображень.

Adobe Kuler дозволяє вивчати і створювати колірні теми, а також обмінюватися ними між різними додатками Adobe. Можна переглядати тисячі колірних тем, відсортованих за часом створення, популярністю або рейтингу, або виконувати пошук за ключовим словом, заголовком або автором. Теми можна завантажити і перенести панель "Колірні зразки" одним натисненням кнопки. Можна також використовувати інтерактивне колірне коло, що підтримує стандартні правила гармонії, для створення власних колірних тем, які можна зберігати, переносити на панель "колірні зразки" або вивантажувати для спільного використання.

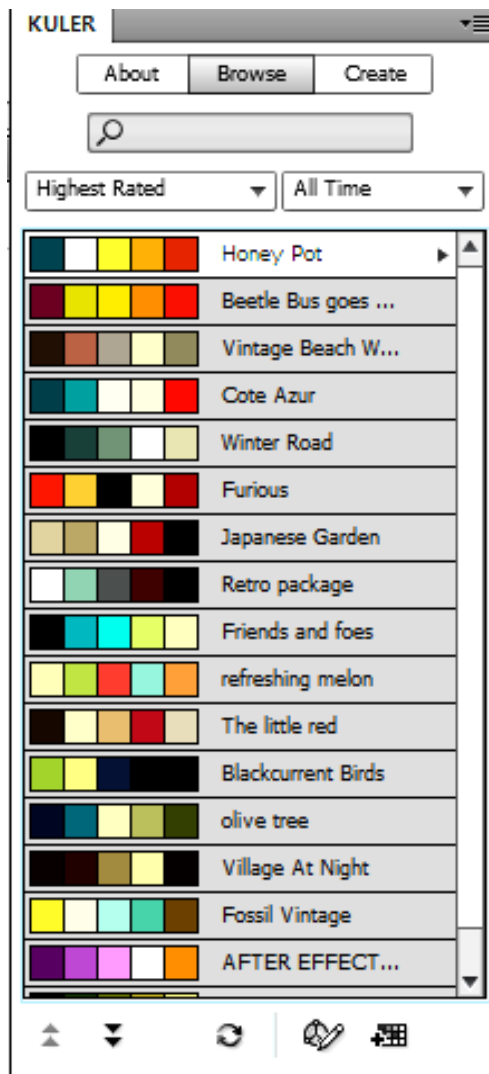


Рис. 1.1. Палітра "Kuler"

Для перегляду колірних тем потрібне підключення до Інтернету.

Пошук тем можна виконати за допомоги таких дій:

1) виберіть "Вікно" > "Розширення" > "Kuler", потім виберіть панель "Огляд";

2) виконайте одну з таких дій:

- у полі пошуку введіть назву теми, тег або ім'я автора (примітка: використовувати можна тільки алфавітні й числові символи (Aa-Яя, 0-9));
- відфільтруйте результати пошуку, вибравши параметр з спливаючих меню над результатами.

Перегляд тем Kuler в Інтернеті:

1) на панелі "Огляд" виберіть тему з результатів пошуку;

2) клацніть трикутник праворуч від теми та виберіть "Переглянути в kuler".

Збереження частих пошуків:

1) виберіть параметр "Настроювані" у першому спливаючому меню на панелі "Огляд";

2) у діалоговому вікні введіть критерії пошуку та збережіть результати.

Щоб почати пошук, виберіть критерії з першого спливаючого меню.

Для видалення збереженого пошуку виберіть "Налаштування" у спливаючому меню. Видаліть інформацію з відповідних полів пошуку і натисніть "Зберегти".

Робота з темами

За допомогою панелі "Kuler" можна створювати або редагувати теми і включати їх у проекти.

Додавання теми до панелі / палітри "Зразки" додатка:

1) на панелі "Огляд" виберіть потрібну тему;

2) клацніть трикутник праворуч від теми і виберіть "Додати до панелі зразків".

Також можна додати тему з панелі "Створити" за допомогою кнопки "Додати до зразків" внизу панелі.

Редагування теми

Для редагування теми слід виконати такі дії:

1) на панелі "Огляд" знайдіть тему, яку потрібно відредагувати, і клацніть двічі на цій темі в результатах пошуку. Тема відкриється на панелі "Створити";

2) на панелі "Створити" відредагуйте тему за допомогою доступних інструментів;

3) виконайте одну з таких дій:

- збережіть тему за допомогою кнопки "Зберегти тему";
- додайте тему на панель "Зразки" за допомогою кнопки "Додати до зразків" внизу панелі;

- завантажте тему до онлайн-послуги "Kuler" за допомогою кнопки "Завантажити", розташованої в нижній частині панелі.

Інструменти панелі "Створити"

Панель "Створити" надає набір інструментів для створення і редагування тем.

Для ознайомлення із інструментами виконайте такі дії:

1) виберіть правило гармонії у спливаючому меню "Правило". У правилі гармонії основний колір використовується як вихідний для створення кольорів в колірній групі. Наприклад, якщо вибрані синій основний колір і правило гармонії "Додаткові", то колірна група створюється за допомогою цього основного кольору (синього) та додаткового по відношенню до нього червоного.

2) виберіть правило "Настроювані", щоб створити тему з допомогою довільних корегувань;

3) змініть колір за допомогою колірного кола. Під час внесення корегувань вибране правило гармонії продовжує керувати створенням кольорів з групи;

4) перемістіть повзунок яскравості поруч з колом, щоб змінити яскравість кольору;

5) задайте основний колір, перетягнувши маркер основного кольору (найбільший маркер кольору з подвійною рамкою) навколо кола. Також можна задати основний колір, відрегулювати повзунки кольорів внизу діалогового вікна.

6) встановіть інші чотири кольори в колірній групі так само, як основний колір. Виберіть зразок кольору і клацніть кнопку з оком під колірною групою;

7) встановіть колір фону і переднього плану у програмі чи колір обведення або заливки в якості основного кольору. Натисніть одну з перших двох кнопок під колірною групою;

8) видаліть колір із колірної групи, вибравши зразок кольору та натиснувши кнопку "Видалити колір" під колірною групою. Додайте новий колір, виділивши порожній зразок кольору та натиснувши кнопку "Додати колір";

9) спробуйте різні колірні ефекти, вибравши нове правило гармонії та переміщуючи маркери в колірному колі;

10) двічі клацніть будь-який зразок у колірній групі для встановлення активного кольору (переднього плану / фону або обведення / заливання) програми. Якщо у програмі немає функції установки активного або вибраного кольору, палітра "Kuler" сама задає відповідний колір переднього плану або заливки.

Завдання

1. Проаналізуйте завдання до розділу, присвяченому дизайну мультимедійної презентації, комплексного курсового проекту. Зробіть такі колірні схеми:

а) для всіх текстових елементів та ефектів (заголовки, надписи в меню, основний текст, гіперпосилання);

б) до вікон презентації, декоративних елементів та елементів навігації.

2. Оцініть створену колірну схему з боку вимог теорії кольору та композиції та дизайну. Помістіть свої спостереження до звіту.

Самостійна робота № 2

Розробка кліпу з використанням анімації розкадрування та шарів.

Створення фрактальної графіки

Мета роботи: поглиблення знань, які було отримано на лекції, та підтвердження і реалізація навичок, що були сформовані на лабораторній роботі за наступними темами. Методи створення анімації. Шари, створення й редагування символів.

Об'єктом СР є інструмент створення, фрактальної графіки та анімації.

Предметом СР є процес створення анімації на основі ключових кадрів, що отримуються шляхом моделювання фракталів.

Методи для виконання СР: моделювання математичних функцій у тривимірному просторі.

У результаті виконання СР формуються такі компетенції: моделювання тривимірних фракталів, створення ефектної анімації розкадрування з використанням фрактальної графіки.

Нормування та вихідний контроль: самостійна робота виконується в рамках вивчення тими № 4 "Методи створення анімації" та № 5 "Шари, створення й редагування символів"; час необхідний для виконання – 4 години; результат роботи – відеоролик та статичні зображення з моделями фракталів.

Контроль якості: диференційований.

Методичні рекомендації

У рамках вивчення курсу комп'ютерна анімація, авторами приділялася увага вивченню основ фрактальної графіки. Багато фахівців вважають

фрактальну графіку дуже перспективним напрямом розвитку, особливо стосовно комп'ютерної анімації. Це пояснюється тим, що потужність комп'ютерних обчислень, що постійно збільшується, дозволяє реалізовувати накопичений раніше теоретичний досвід досліджень в області фрактальної графіки й анімації. Так, наприклад, фрактальною моделлю можуть бути описані об'єкти неживої природи або імітація різних природних об'єктів. При цьому, структура тривимірних моделей, що отримуються (рівні вкладень складових об'єкта, що отримується) можуть бути нескінченні. Це дозволяє створювати наприклад комп'ютерну анімацію ґрунту, піску, трави, яка дозволяє максимально близько наблизитися до об'єкта, проникати усередину його й подорожувати усередині об'єктів, що моделюються.

Розглянемо способи моделювання фракталів і створення на їхній основі комп'ютерної анімації. Для цього скористаємося програмою Mandelbulber, яку можна завантажити за адресою <http://sites.google.com/site/mandelbulber>.

Інтерфейс становить окреме вікно управління з множиною параметрів і різні вікна для відображення результуючої інформації й параметрів, що відкриваються окремо. Внизу вікна управління є блок основних кнопок. Цей блок (рис. 2.1) є на екрані не залежно від того, яка закладка є активною.

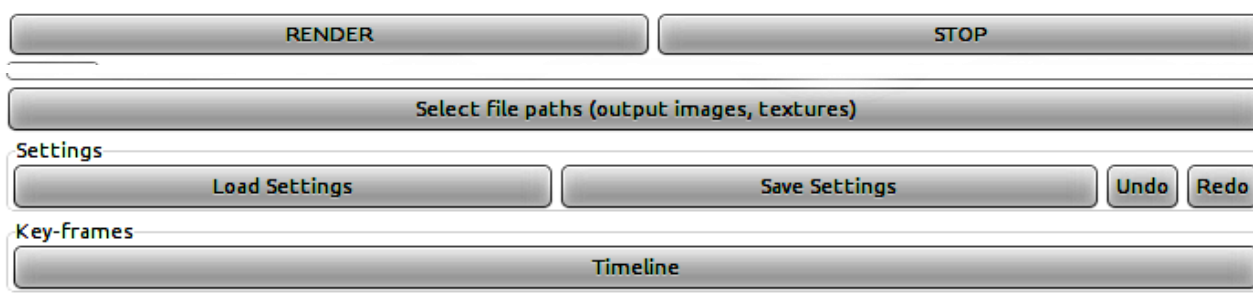


Рис. 2.1. Блок основних кнопок управління Mandelbulber

Розглянемо основні команди:

- **Render** – початок рендерінгу зображень;
- **Stop** – зупинити рендерінг;
- **Select file paths** – вибір шляху до файла – відкрити вікно з встановленням шляхів додаткових файлів;
- **Load setting** – завантаження всіх параметрів фракталу з файла;

- **Save settings** – зберегти всі параметри фракталу у файл;
- **Undo** – скасувати всі зміни, зроблені з моменту останнього рендерінга (100 рівнів скасування);
- **Redo** – повторити зміни.

У вікні вибору шляхів до файлів можна управляти наступними параметрами:

Destination image sequence – шлях та ім'я файлу (префікс) для збережених зображень. До ім'я буде автоматично додано номер зображення й розширення . *JPG* або . *PNG*;

Background – фон – шлях для зображення, що буде використовуватися як фонове зображення;

Environment map – шлях для зображення, що буде використовуватися як підставне відбиття;

Ambient occlusion colour map – шлях для зображення, що буде використовуватися як карта кольорів для ефекту навколишньої оклюзії (методика розрахунку світла);

Animation path – шлях до файлів, у яких будуть зберігатися записані траєкторії польоту й іншої анімації;

Keyframe sequence – шлях, ім'я файла-префікса для ключових кадрів.

Вкладка "Фрактал"

На цій вкладці можна налаштовувати основні параметри фракталу, що моделюється.

Основним параметром є Formula (формула), що використовується для побудови фракталу і міститься в розділі Параметри фракталу.

Роглянемо найбільш цікаві тип формул, що використовуються (рис. 2.2).

Також існує **Trigonometric with calculation of analytic DE** – тригонометрична з розрахунком аналітичної ОВ (оцінки відстані) – така ж, як *тригонометрична формула*, але оцінка відстані розраховується за аналітичними формулами. Ця формула працює швидше, ніж *Тригонометрична*, але не так точно для менш потужних фракталів і фракталів Джулія.

Trigonometric (Mandelbulb) –
тригонометричні

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\alpha = \text{atan2}(y, x)$$

$$\beta = -\text{atan2}(z, \sqrt{x^2 + y^2})$$

$$x = r^p \cos(p\beta) \cos(p\alpha) + a$$

$$y = r^p \cos(p\beta) \sin(p\alpha) + b$$

$$z = r^p \sin(p\beta) + c$$

Polynomic power 2 – поліноміальна
потужність 2

$$\text{newx} = (x^2 - y^2) \left(1.0 - \frac{z^2}{x^2 + y^2} \right)$$

$$\text{newy} = 2xy \left(1.0 - \frac{z^2}{x^2 + y^2} \right)$$

$$\text{newz} = -2z \sqrt{x^2 + y^2}$$

$$x = \text{newx} + a$$

$$y = \text{newy} + b$$

$$z = \text{newz} + c$$



Рис. 2.2. Тригонометричні та поліноміальні типи формул фракталів

Також слід звернути увагу на **Polynomic power 2 minus Z** та **Hypercomplex** (рис. 2.3).

До моделей з досить незвичним виглядом можна віднести **Кватерніон** та **Губка Менгера** (рис. 2.4).

Polynomic power 2 minus Z -
 Поліноміальна потужність 2 - мінус z

$$\begin{aligned} newx &= (x^2 - y^2) \left(1.0 - \frac{z^2}{x^2 + y^2} \right) \\ newy &= 2xy \left(1.0 - \frac{z^2}{x^2 + y^2} \right) \\ newz &= 2z \sqrt{x^2 + y^2} \\ x &= newx + a \\ y &= newy + b \\ z &= newz + c \end{aligned}$$

Hypercomplex - Гіперкомплекс

$$\begin{aligned} newx &= x^2 - y^2 - z^2 - w^2 \\ newy &= 2(xy - wz) \\ newz &= 2(xz - yw) \\ neww &= 2(xw - yz) \\ x &= newx + a \\ y &= newy + b \\ z &= newz + c \\ w &= neww \end{aligned}$$

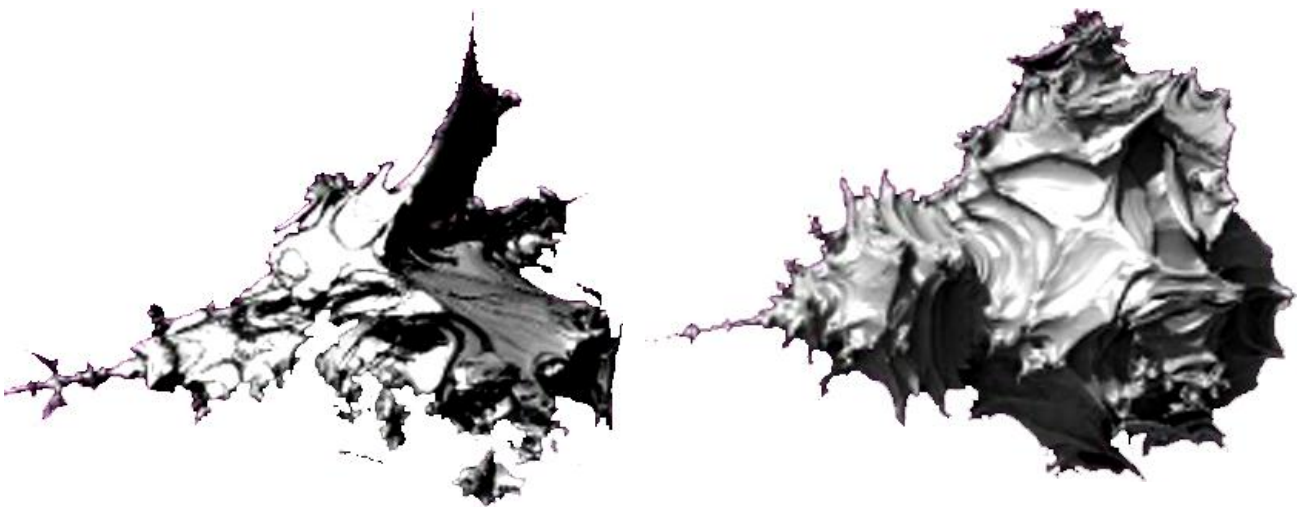


Рис. 2.3. Поліноміальні та гіперкомплексні типи формул фракталів

Вкладка IFS

Параметри на вкладці калейдоскопічної IFS (рис. 2.5) використовуються, якщо обрано формулу калейдоскопічної IFS або у випадку, якщо включено IFS режим вкладки. General IFS parameters – основні параметри IFS.

Scale – general scale of IFS formula **Масштаб** – загальні масштаби IFS формули (IFSScale).

Rotation alfa, beta, gamma – обертання альфа, бета, гама - основний кут повороту IFS формули (IFSMaInRotation).

Offset x, y, z – зсув x, y, z – вектор зсуву для масштабування основного обертання (IFSOffset).

abs(x), abs(y), abs(z) – включає / відключає функції абсолютного значення для кожної осі. Ці функції роблять додаткові дзеркальні відбиття IFS формули.

Quaternion – Кватерніон

$$newx = x^2 - y^2 - z^2 - w^2$$

$$newy = 2xy$$

$$newz = 2xz$$

$$neww = 2xw$$

$$x = newx + a$$

$$y = newy + b$$

$$z = newz + c$$

$$w = neww$$



Menger sponge – Губка Менгера

Формально губка Менгера може бути визначена в таким чином:

$$M := \bigcap_{n \in \mathbb{N}} M_n$$

де M_0 , одиничний куб та

$$M_{n+1} := \left\{ (x, y, z) \in \mathbb{R}^3 : \exists i, j, k \in \{0, 1, 2\} : (3x - i, 3y - j, 3z - k) \in M_n \right. \\ \left. \text{and at most one of } i, j, k \text{ is equal to } 1 \right\}.$$

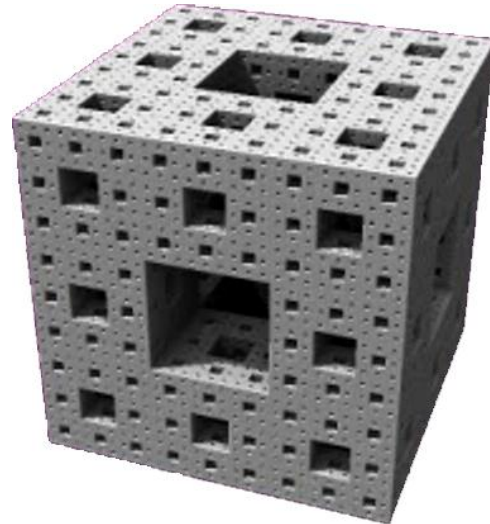


Рис. 2.4. Моделі фракталів Кватерніон та Губка Менгера

Fractal
IFS
Hybrid
Mandelbox
View
Shaders
Lights
Image
Post effects
Animation
Sound
About...

General IFS parameters

scale: rotation alfa: rotation beta: rotation gamma:

offset x: offset y: offset z: abs(x) abs(y) abs(z)

Symetry vectors

symetry x	symetry y	symetry z	alfa	beta	gamma	distance	intensity	enabled
0,80901699	0,30901699	-0,50000000	0	0	0	0	1	<input checked="" type="checkbox"/>
-0,50000000	0,80901699	0,30901699	0	0	0	0	1	<input checked="" type="checkbox"/>
0,30901699	-0,50000000	0,80901699	0	0	0	0	1	<input checked="" type="checkbox"/>
-0,80901699	0,30901699	0,50000000	0	0	0	0	1	<input type="checkbox"/>
0,50000000	-0,80901699	0,30901699	0	0	0	0	1	<input type="checkbox"/>
1	0	0	0	0	0	0	1	<input type="checkbox"/>
1	0	0	0	0	0	0	1	<input type="checkbox"/>
1	0	0	0	0	0	0	1	<input type="checkbox"/>
1	0	0	0	0	0	0	1	<input type="checkbox"/>

Рис. 2.5. Вкладка IFS

У вкладці **Гібрид** (рис. 2.6) можна комбінувати різні формули для створення гібридних фракталів, які успадковують характеристики кожної зі своїх формул.

Використовуючи поля для введення ітерацій, ви можете вказати, скільки разів розраховується кожна формула, перш ніж програма перейде до наступної формули. Якщо обрано прапорець "Cyclic loop", програма вертається до першої формули, коли вона завершує останню зазначену формулу. Якщо прапорець "Cyclic loop" не встановлений, перша формула розраховується певне число ітерацій, а потім наступні.

Для кожної формули визначається параметр потужності/масштабу.

Примітка: для більшості гібридних формул необхідно знизити "фактор кроку OB", щоб уникнути ефекту перевантаження (багато чорних плям або важких шумів на зображенні).

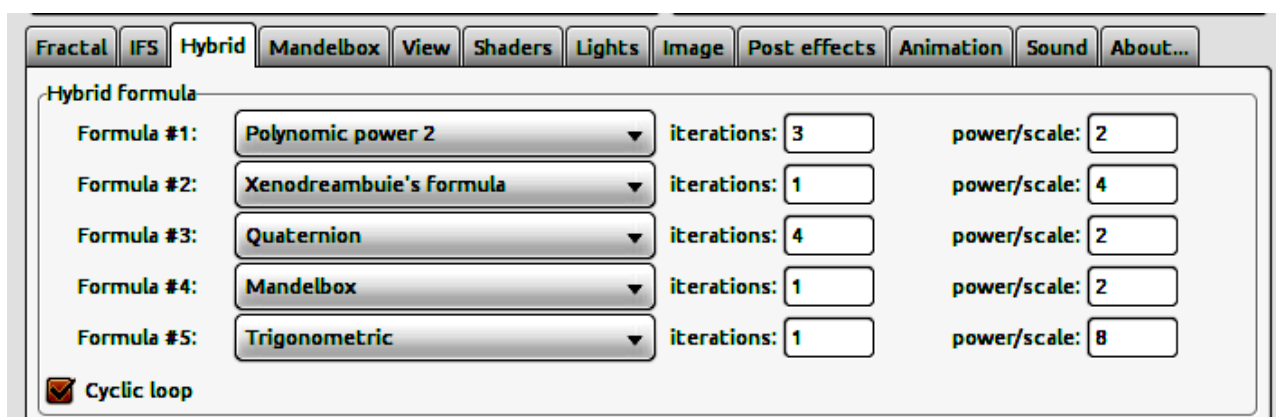


Рис. 2.6. Вкладка Гібрид

Вкладка Mandelbox

Параметри на вкладці *Mandelbox* (рис. 2.7) можна використовувати, коли обрана формула *Tglad (Mandelbox)* або включено режим IFS вкладки.

Main Mandelbox Parameters – Основні параметри Mandelbox:

Scale-масштаб – параметр масштабу для Mandelbox формули (scale);

Folding limit – граничне значення для кубічного складання (foldingLimit);

Folding value – значення зсуву для кубічного складання (foldingValue);

Fixed radius – параметр для сферичного складання – вище граничне значення (fixedRadius);

Min radius – параметр для сферичного складання – нижнє граничне значення (minRadius).

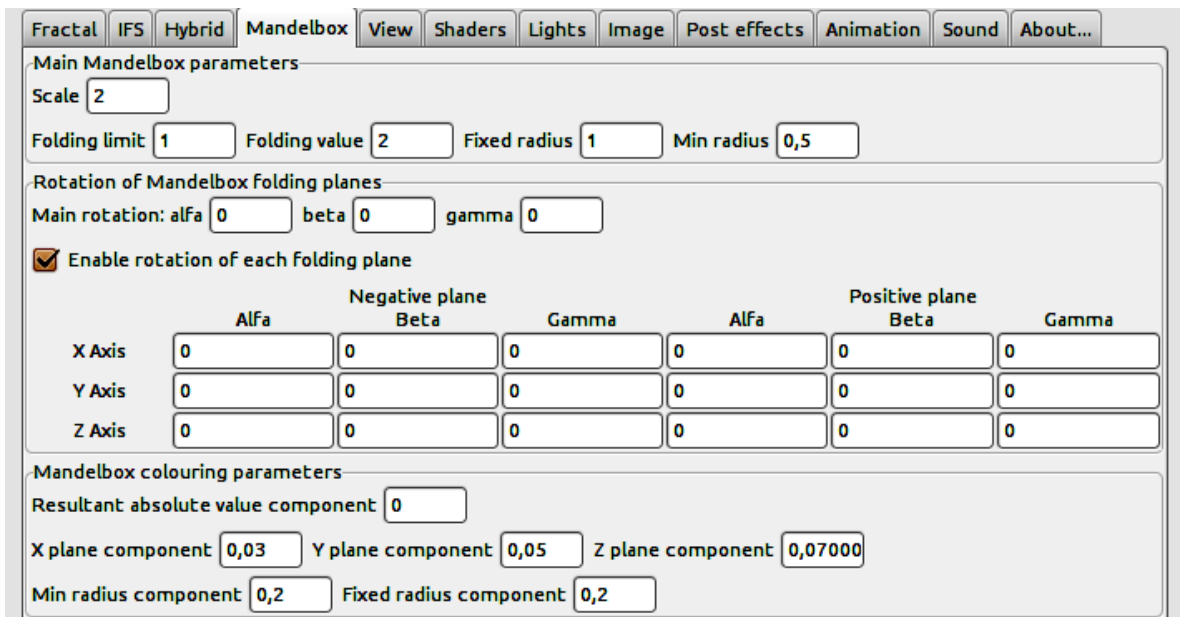


Рис. 2.7. Вкладка Mandelbox

Вкладка Shaders

Ця вкладка (рис. 2.8) призначена для настроювання параметрів остаточного оформлення об'єктів, що моделюються, і дозволяє вибирати параметри кольорів об'єкта а також різні інші характеристики поверхні.

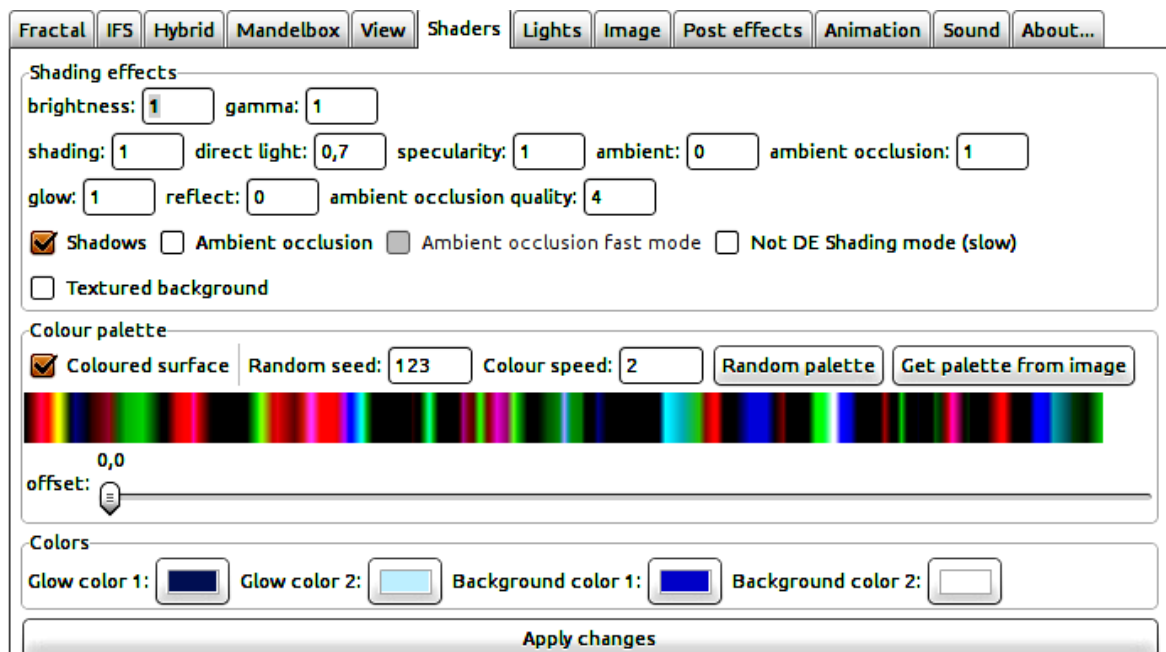


Рис. 2.8. Вкладка Shaders

Вкладка Lights

Призначена для настроювання параметрів освітлення об'єкта (рис. 2.9), що моделюються, за допомогою основного й додаткового джерел світла з можливістю настроювання множини параметрів.

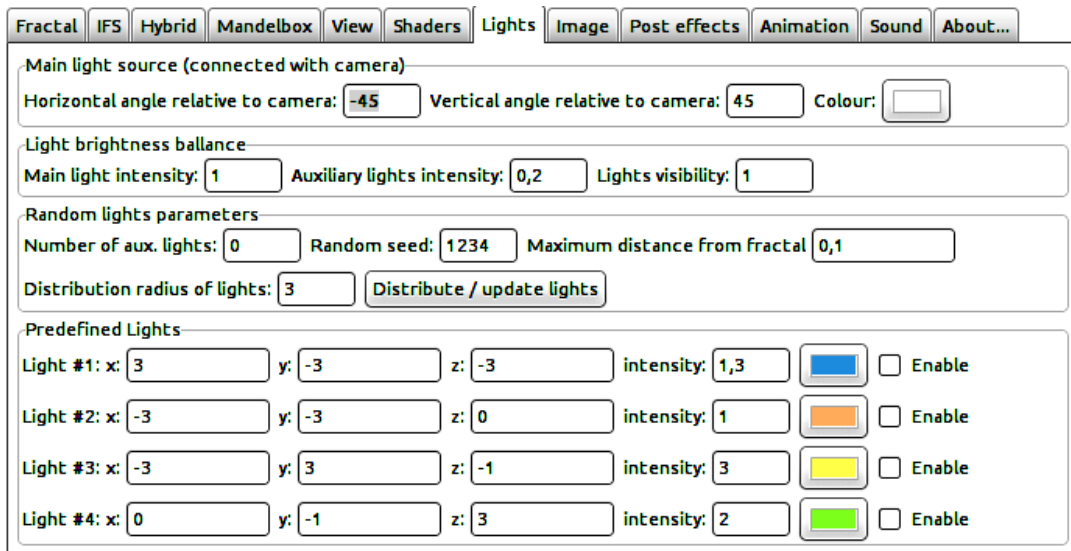


Рис. 2.9. Вкладка Lights

Вкладка Animation

Вкладка (рис. 2.10) призначена для вибору параметрів створення анімації (кількість проміжних кадрів на 1 ключовий (Frames per key), кадри, які підлягають рендерінгу). А також безпосередньо запускати процес генерування кадрів (Render from keyframes).

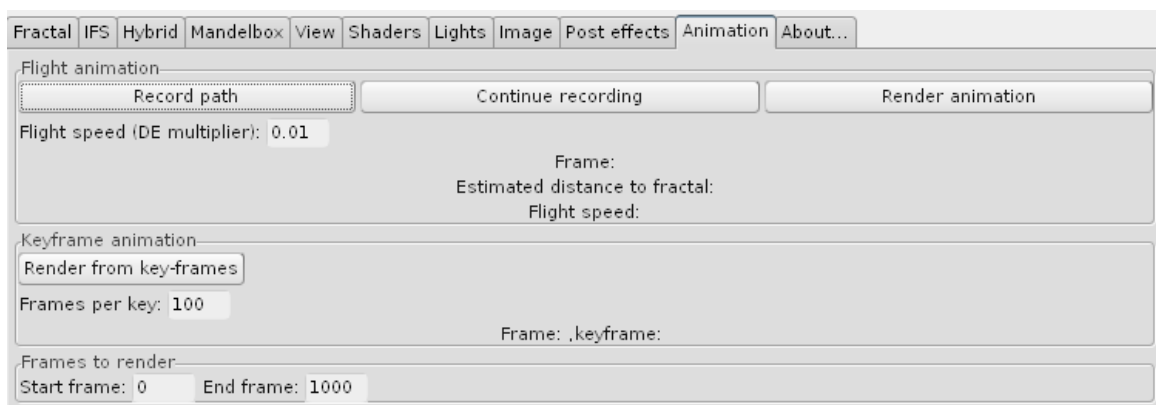


Рис. 2.10. Вкладка Animation

Time line window (вікно шкали часу) наведено на рис. 2.11.

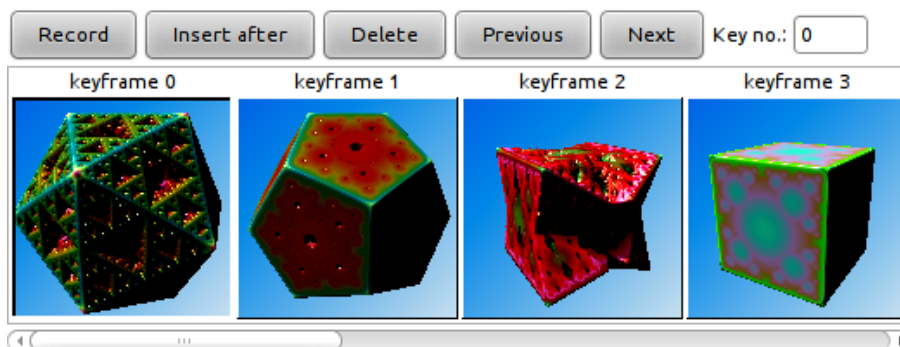


Рис. 2.11. Time line window (вікно шкали часу)

Вікно Timeline дозволяє переглядати, додавати, редагувати й видаляти ключові кадри для створення анімації. Використовуйте шкалу разом із вкладкою "Анімація", щоб створювати анімацію, у якій може бути враховано багато параметрів.

Отримавши модель будь якого фракталу, її можна використовувати як ключовий кадр. Для цього необхідно натиснути кнопку Insert After на шкалі часу.

Для отримання наступних ключових кадрів, можна моделювати інші фрактали (або змінювати параметри для моделі даного фракталу). Після закінчення рендерінгу й отримання зображення – необхідно використовувати команду Insert After для додавання ключового кадру на шкалу часу.

Також можна скористатися вкладкою View, для того щоб змінювати координати точки, з якої виконуються нагляд за об'єктом: виконувати обльоти або наближатися до об'єкта. Або у вікні з рендеренгом виконувати команду "move the camera" – "передвигати камеру". Отримані таким чином ключові кадри також необхідно додавати до шкали часу.

За допомогою кнопки Record можна перенести моделі із ключового кадру у вікно рендеринга.

Після того, як сформований набір ключових кадрів, необхідно перейти на вкладку анімація й виконати команду Render from key-frames. Як параметр використовується кількість проміжних кадрів на 1 ключовий (Frames per key). Також можна вибирати кадри, які підлягають рендерингу.

Програмою буде сгенерована послідовність зображень, які за замовчуванням будуть зберігатися в папку C:\Documents and Settings\Admin\mandelbulber\images. Розрахунок проміжних кадрів виконується на основі методики сплайнів Catmull-Rom.

Для того, щоб зробити анімацію з послідовності кадрів, можна скористатися Adobe Flash. Створивши для цього документ із покадровою анімацією, у який будуть імпортовані всі отримані зображення. Не дивлячись на те, що отриманих зображень может бути декілька сотень або тисяч, процес імпортування й створення анімації не буде трудомістким. При імпортуванні зображення, достатньо указати тільки ім'я першого файлу. Зважаючи на те, що всі файли у своїй назві містять однаковий префікс і відрізняються тільки номерами, під час імпортування першого файлу, Flash сам запропонує імпортувати всю послідовність і розташувати її на окремих ключових кадрах. Отриману таким чином анімацію можна зберігати й надалі демонструвати у форматі SWF, або скористатися функцією імпорту відео в будь-якому іншому форматі, передбаченому Flash.

Завдання

1. Зробіть експерименти з декількома функціями для побудови фракталів.
2. Оберіть оптимальні параметри, що управляють функцією, а також спробуйте налаштувати додаткові параметри, що управляють оточенням отриманої моделі (фонове зображення, освітлення, трасування і т. ін.)
3. Після отримання необхідного фракталу, зробіть рендерінг зображення з високою роздільною здатністю та збережіть зображення.
4. Встановіть параметри, що відповідають за роздільну здатність зображення 400x300 та переходьте до створення анімації.
5. Зробіть декілька ключових кадрів, використовуючи інструмент переміщення камери, тим самим імітуючи підльот до фракталу.
6. Змініть параметри побудови та освітлення фракталу, та зробіть декілька ключових кадрів, які дозволять отримати анімацію трансформації фракталу.
7. Виконайте рендерінг кадрів для анімації з використанням отриманих ключових кадрів (це може зайняти багато часу).
8. Використайте Adobe Flash та створіть на основі отриманих зображень та покадрової анімації відеоролик, який би висвітлював отримані вами фрактали.
9. Зробіть у Adobe Flash додаток, який би дозволяв переглядати слайд-шоу з отриманими зображеннями фракталів та відеоролик.

Самостійна робота № 3

Оптимізація імпортованої фото- і відеоінформації у анімаційному кліпі. Оптимізація подання кліпу та його складників для використання у різних задачах

Мета роботи: практичних навичок щодо оптимізації flash-додатків.

Об'єктом СР є анімаційний додаток, що потребує оптимізації.

Предметом СР є процес оптимізації flash-додатків при створенні анімації для використання онлайн та офлайн.

Методи для виконання СР: аналіз та синтез, планування, розроблення контенту.

У результаті виконання СР формуються такі компетенції: оптимізація імпортованої фото- і відеоінформації, оптимізація звуку та графіки, врахування особливостей оптимізації flash-додатків під час публікації до мережі Інтернет.

Нормування та вихідний контроль: самостійна робота виконується в рамках вивчення тем № 6 "Імпортування й оптимізація об'єктів. Робота з відео й звуком" та № 9 "Тестування й оптимізація кліпу для різних способів подання"; час необхідний для виконання – 4 години; результат роботи – звіт щодо оптимізованого flash-додатку.

Контроль якості: диференційований.

Методичні рекомендації

При розробці складних проектів, пов'язаних з анімацією, динамікою, обробкою даних, виникає необхідність оптимізувати графіку. Уповільнення роботи можуть викликати багато факторів: складна графіка, велика кількість об'єктів, складна анімація, велика частота кадрів та ін., все це треба враховувати при розробці. Головне завдання оптимізації Flash-додатків – максимально зменшувати розмір swf-файла.

Символи

На сцені мають бути символи тільки двох типів Movie Clip і Graphic (замість Button потрібно використовувати Movie Clip). Зображення і векторна графіка повинні бути конвертовані в Graphic; якщо надалі буде передбачено використати властивість Blend або програмно, потрібно буде звертатися до властивостей символів (збережених як Graphic), їх слід конвертувати в Movie Clip (рис. 3.1).

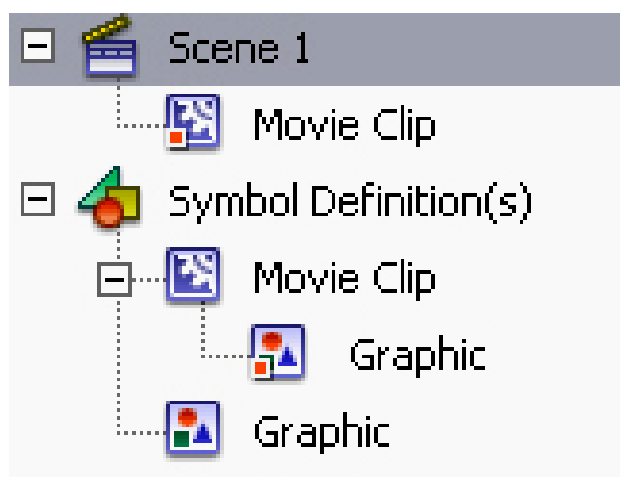


Рис. 3.1. Розташування символів у бібліотеці

Оптимізація векторної графіки

Лінії, намальовані олівцем, займають менше місця, ніж лінії, намальовані пензлем. У векторному поданні будь-яка крива представляється як

набір точок-координат і заданим рівнянням дуги кола між кожною парою точок. Отже, чим більше задано точок, тим точніше можна описати криву. Але практично немає необхідності ускладнювати опис кривої, якщо в цьому немає особливої потреби, оскільки це забирає зайву пам'ять і швидкість відтворення. Flash дозволяє згладжувати, випрямляти і оптимізувати лінії (меню Modify > Shape > Optimize або Ctrl + Alt + Shift + C, (рис. 3.2)). Чим пряміше лінії, тим менше місця вони займають.

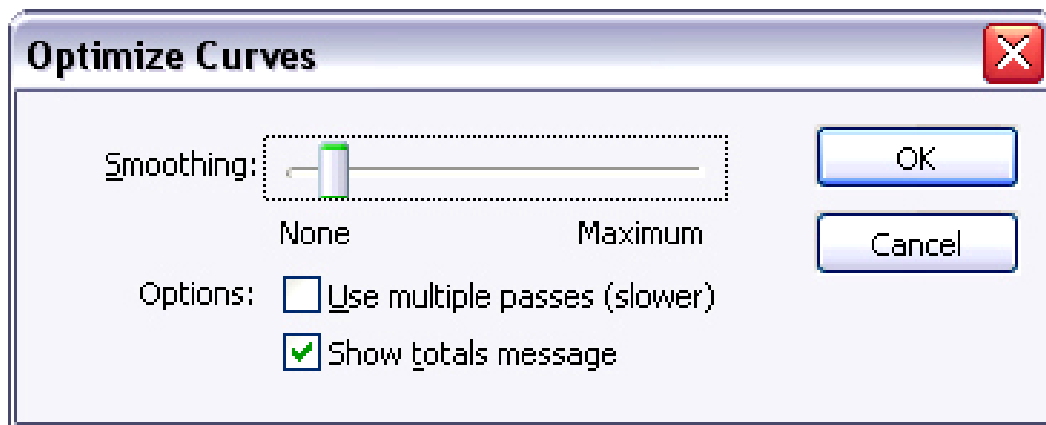


Рис. 3.2. Вікно оптимізації кривих

Movie Clip

Замість властивостей `_alpha` або `_visible` символів типу Movie Clip потрібно робити перехід на порожній кадр всередині цих символів (рис. 3.3) або взагалі видаляти, якщо вони далі не використовуються, таким чином зменшуючи розмір fla-файла.

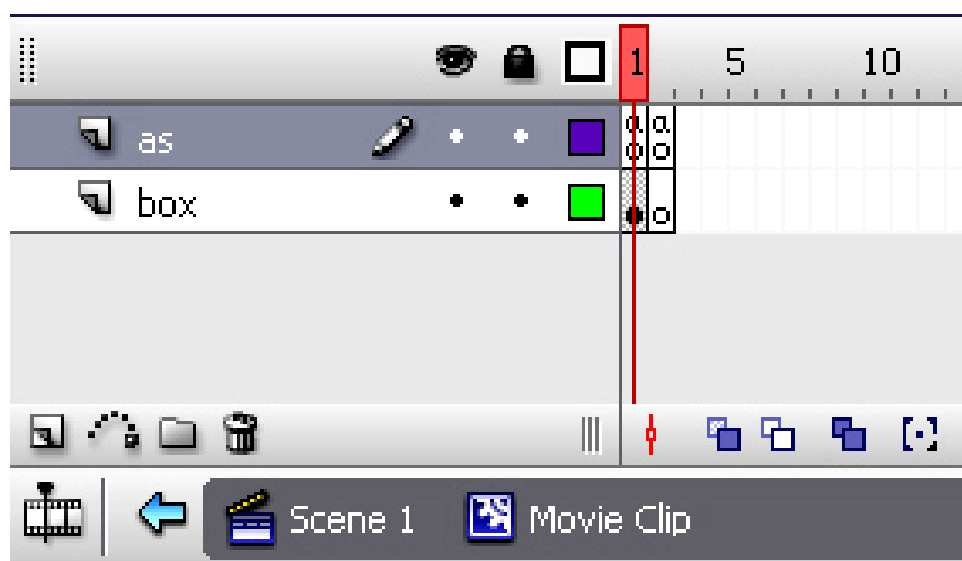


Рис. 3.3. Шкала часу із розташованим на ній символом типу Movie Clip

Текст

Flash перетворює всі букви, що використовуються в проекті, в символи. Кожна буква зберігається у вигляді об'єкта, який потім розмножується потрібну кількість разів. Причому, якщо ми напишемо ту ж фразу іншим шрифтом, Flash буде змушений зберегти зображення букв цього шрифту теж (плюс Flash розрізняє прописні і малі літери). Щоб цього уникнути, потрібно вибрати Use Device Fonts (рис. 3.4) в опціях тексту. При цьому Flash буде зберігати не зображення букв, а тільки характеристики і назву шрифту (на практиці, це всього кілька байт). При відтворенні буде використаний вказаний шрифт, або, якщо такого шрифту не виявиться в системі, Flash Player використовує найближчий за характеристиками шрифт. Тому великі обсяги тексту краще не зберігати в Flash, а використовувати, наприклад, XML.

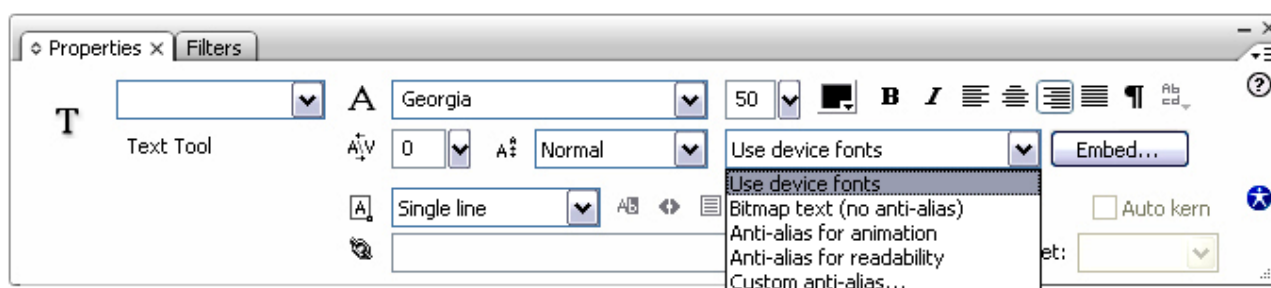


Рис. 3.4. Палітра налаштування шрифту для правильної оптимізації

Анімація

Анімація у Flash базується на математичному підґрунті. Звідси анімаційні послідовності набагато оптимальніше оформлювати у вигляді Tween (якщо це можливо), ніж у вигляді послідовностей ключових кадрів. Tween просто містить формулу з перетворення об'єкта анімації на певному відрізку, що значно економить розмір. Анімаційні ефекти потрібно переносити на окремі шари від всіх інших елементів, це економить ресурси, так як в процесі анімації будуть оброблятися й статичні символи, виконуючи цим зайву роботу. Якщо створювана анімація не прямолінійна (з уповільненням або прискоренням, наприклад ефект падіння чого-небудь під дією сили тяжіння), то використовуємо Edit для Tween (рис. 3.5). Чим більше анімації символів в сцені, тим складніше Flash обробляти їх, і тим повільніше буде відтворення.

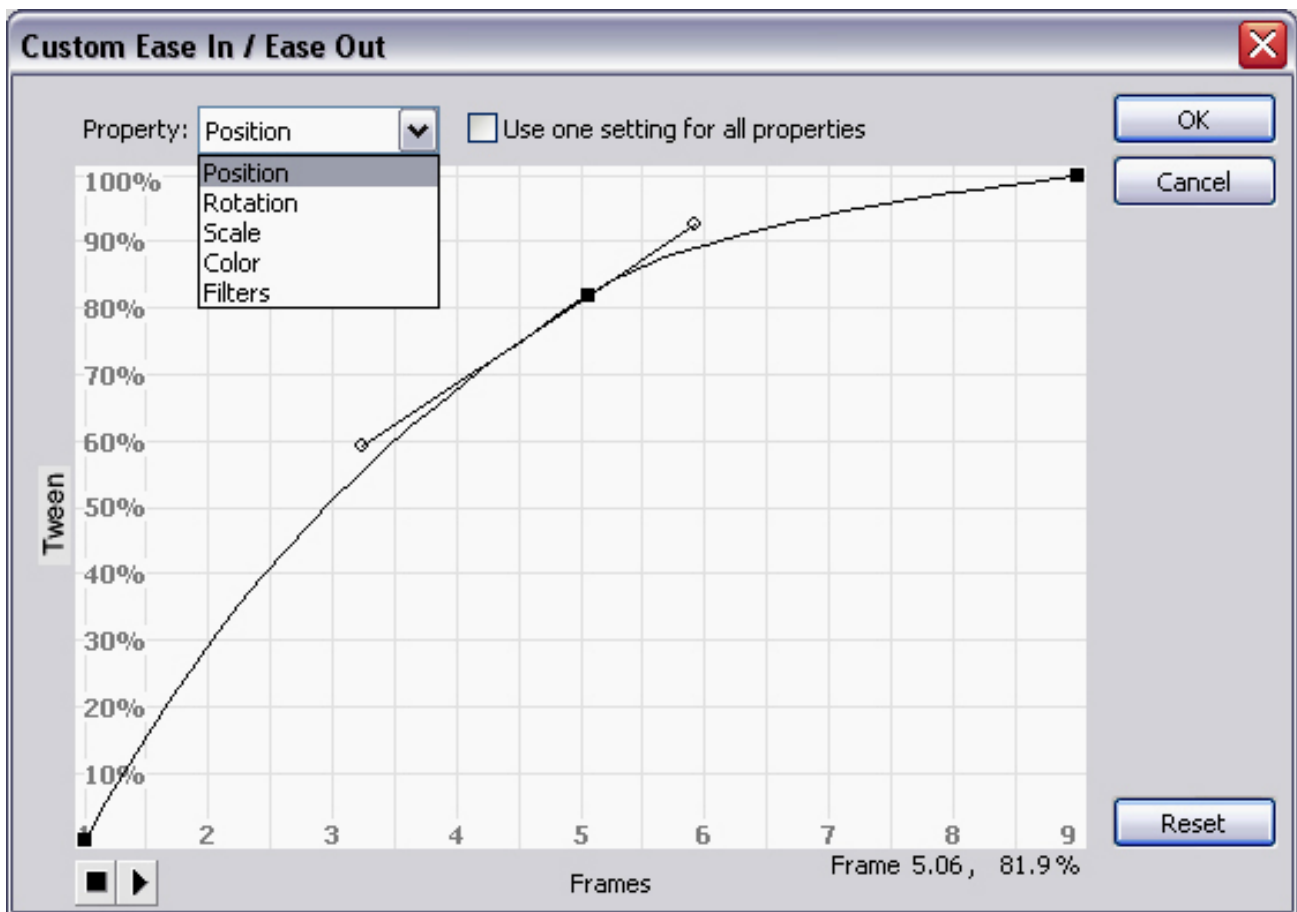


Рис. 3.5. Редагування анімації в разі присутності ефектів прискорення чи уповільнення руху

Швидкість виконання анімації

Загальновідомо, що плавність анімації досягається великою кількістю програвань кадрів в секунду (fps – frames per second, кількість кадрів за секунду). Тому, багато розробників прагнуть вказувати велику швидкість у властивостях Flash-фільму (за замовчуванням, Flash використовує значення 12 fps, для якісної анімації потрібно 25 – 30 fps).

Проте, збільшення кількості кадрів в секунду вимагає більшої продуктивності комп'ютера, на якому виконується анімація, і якщо її не вистачає, Flash скорочує частоту кадрів. Отже, навіть якщо ви встановите fps рівним 100, Flash буде виходити з можливостей, наявних для відтворення.

Є способи підвищити fps. Ключовий, проте не самий вигідний – зменшити розміри відтворюваного кліпу: на слабкій машині кліп з розмірами 300x200 буде відтворюватися значно краще, ніж, наприклад, кліп з розмірами 600x400.

Відтворення у мережі Інтернет

Flash, спеціально адаптований для відтворення у мережі Інтернет, є потоковим форматом. Це означає, що фільми Flash можуть почати відтворюватися, не завантажившись до кінця. З одного боку, це перевага, тому що фільм починає відтворюватися дуже передчасно, і користувачеві немає потреби чекати кінця завантаження. З іншого боку, якщо канал, по якому передаються дані, виявиться вужче, ніж потрібно для передачі Flash-потoku, анімація буде припинена, і користувачеві доведеться чекати наступної частини. Якщо для вас незначні такі затримки, або у вас є впевненість в тому, що канали зв'язку не підведуть – вам немає про що турбуватися. Але якщо вам хочеться, щоб ваш кліп відтворювався без затримок, тоді його треба забезпечити попередніми завантажувачем (preloader) – це прийом, що дозволяє затримати відтворення до повного завантаження фільму.

Має сенс протестувати роботу вашого фільму на декількох швидкостях за допомогою функції Test Movie (Control > Test Movie, Ctrl + Enter). Швидкість можна вибирати в меню Debug, яке з'являється після запуску фільму.

Потоковий звук

Ще один аргумент, що відноситься до продуктивності Flash анімації – це потоковий звук. У Flash є декілько способів синхронізації звуку. Якщо ви використовуєте дуже великий фрагмент музики (або будь-якого іншого звукового супроводу), і події анімації повинні збігатися з подіями звуку, то вам потрібно використовувати синхронізацію Stream (потік). У цьому випадку вся анімація буде синхронізована зі звуком, і два ці потоку йтимуть паралельно, не відстаючи один від одного.

Завдання

1. Визначте розмір flash-презентації до свого комплексного курсового проекту (фізичний об'єм та завантаження процесору у відсотках під час відтворення аудіо- та відеоконтенту, а також анімації растрової графіки).

2. Зробіть оптимізацію різних видів контенту презентації згідно з методичними рекомендаціями.

3. Визначте ті ж параметри презентації, що й в п 1. завдання. Проаналізуйте зміни та опишіть їх у звіті.

Змістовний модуль 2. Використання Macromedia Flash-технології та 3DS MAX для створення сучасних мультимедійних документів

Самостійна робота № 4

Аналіз сучасних тенденцій та інструментів, що використовуються при створенні Flash анімації для маркетингу. Створення додатків для соціальних мереж

Мета роботи: поглиблення знань, які було отримано на лекції, та підтвердження і реалізація навичок, що були сформовані на лабораторній роботі за темою використання технології Flash в електронному маркетингу.

Об'єктом СР є додаток для соціальної мережі ВКонтакте.

Предметом СР є вивчення взаємодії інтерфейсу Flash з соціальною мережею та розробка відповідного додатка.

Методи для виконання СР: аналіз інформаційних потоків, програмування мовою Action Script.

У результаті виконання СР формуються такі компетенції: розробка додатка, який реалізує обмін даними, для соціальної мережі ВКонтакте.

Нормування та вихідний контроль: самостійна робота виконується в рамках вивчення теми № 7 "Використання технології Flash в електронному маркетингу"; час необхідний для виконання – 4 години; результат роботи – додаток, розміщений у середовищі соціальної мережі.

Контроль якості: диференційований.

Методичні рекомендації

На лекціях за цією темою зверталось багато уваги на популярність, що завоювали проекти соціальних мереж в Інтернеті. Теперішнім часом сформувалося конкурентне поле, на якому розроблювачі різних проектів, намагаються залучити у свою мережу усе більше нових користувачів а також утримати активних користувачів, що існують. Ефективна конкурентна боротьба на даному ринку можлива тільки за рахунок постійного додавання нових функцій в інтерфейси системи, що дозволяють користувачам ефективно обмінюватися інформацією. Крім функціонала, що забезпечується розроблювачами також з'явилася можливість створювати додатків для соціальних мереж і звичайними користувачами. Ринок додатків для соціальних мереж перебуває на етапі бурхливого розвитку. Затребуваність додатків для соціальних мереж пояснюється бажанням різних компаній отримати доступ до величезної

аудиторії, що проводить час у соціальних мережах. Серед таких додатків можуть бути інформаційно-довідкові, розважальні, онлайнві ігри, додатки для електронної комерції. Розглянемо одну з російськомовних соціальних мереж і процес створення додатків до неї. Велику популярність завоювала мережа Вконтакті (рис. 4.1).



Рис. 4.1. Логотип соціальної мережі Вконтакті

Розглянемо додаток, що дозволяє відобразити ім'я, прізвище, дату народження, аватар користувача, що зайшов на сторінку в соціальній мережі Вконтакті.

Цей додаток не використовує додаткових класів для взаємодії з `api.vkontakte.ru`.

Нижче наведені дії, необхідні для програмування на Action Script 3, у середовищі Adobe Flash CS4 або вище.

1. Створюємо новий документ (рис. 4.2). `File->New->FlashFile (ActionScript 3.0)`.

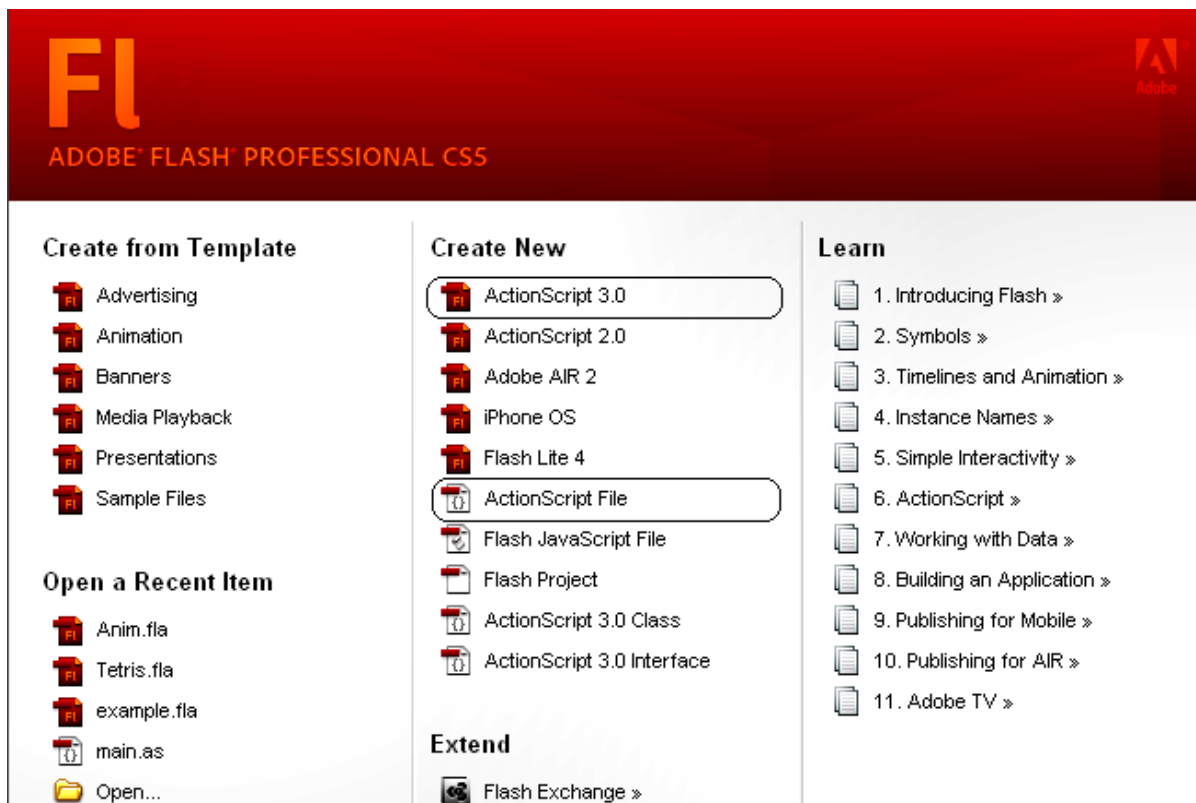


Рис. 4.2. Створення нового документа, створення as-файла

2. Зберігаємо документ. Створюємо нову папку й зберігаємо в неї fla файл. Можна змінити ім'я, наприклад example fla.

3. Створюємо Новий as-файл, у якому буде писаний код. File -> New -> ActionScript File

Нижче наведено код для as-файла. Фрагмент коду в електронному вигляді, можна отримати у викладача.

```
package {
    import flash.net.URLLoader;
    import flash.net.URLRequest;
    import flash.display.*;
    import flash.events.*;
    import flash.text.*;
    public class Main extends Sprite {
        private var api_id:Number=743131; // id вашого додатка
        private var api_secret:String="sE4VlnFfSQ"; // секретний ключ
        private var externalXML:XML;
        private var loader:URLLoader = new URLLoader();
        private var photoLoader:Loader = new Loader();
        private var viewer_id:Number;
        private var sig:String;
        private var uid:String;
        private var bdate_txt:TextField = new TextField();
        private var name_txt:TextField = new TextField();
        public function Main():void {
            // дізнаємось id користувача, який переглядає додаток
            var params:Object=LoaderInfo(root.loaderInfo).parameters;
            viewer_id = (params['viewer_id']) ? parseInt(params['viewer_id']) : 1;
            // отримуємо профіль цього користувача
            getProfile(viewer_id);
        }
        private function getProfile(uid):void { // функція отримання профіля
            sig=MD5.encrypt(viewer_id+'api_id='+api_id+'fields=bdate,photo_bigmethod=getPr
ofilesuids='+uid+'v=2.0'+api_secret);
            var request:URLRequest=new
            URLRequest("http://api.vkontakte.ru/api.php?api_id="+api_id+"&method=getProfile
s&uids="+uid+"&fields=bdate,photo_big&v=2.0&sig="+sig);
            loader.load(request);
            loader.addEventListener(Event.COMPLETE, onComplete);
        }
    }
}
```

```

private function onComplete(event:Event):void { // функція, що виконується
коли профіль було завантажено
    bdate_txt.text="";
    if (loader!=null) {
        externalXML=new XML(loader.data);
        // відображаємо дату народження
        var bdate = externalXML..bdate;
        bdate_txt.text=bdate;
        bdate_txt.x=490;
        bdate_txt.y=380;
        addChild(bdate_txt);
        // відображаємо ім'я та прізвище
        var uname = 'Уважаемый ' + externalXML..first_name + ' ' +
externalXML..last_name;
        name_txt.text=uname;
        name_txt.x=410;
        name_txt.y=360;
        addChild(name_txt);
        // завантажуюмо аватар
        var photo = externalXML..photo_big;
        var request2:URLRequest=new URLRequest(photo);
        photoLoader.load(request2);
        photoLoader.contentLoaderInfo.addEventListener(Event.COMPLETE,
onloaded);
    } else {
        trace("Error!");
    }
}
private function onloaded(e:Event):void {
    // відображаємо завантажений аватар
    photoLoader.x = (550-photoLoader.width)/2;
    photoLoader.y = (400-photoLoader.height)/2;
    addChild(photoLoader);
}
}
}
}

```

5. Зберігаємо as файл. Зберігаємо в ту ж папку, у яку зберігали fla файл. Назвемо його "Main.as".

6. Підключаємо Main.as до проекту. Вибираємо вкладку з першим створеним документом, натискаємо на порожньому місці, щоб зняти виділення з текстового поля, і в PROPERTIES встановлюємо Class "Main" – створений раніше as файл із кодом (рис. 4.3).

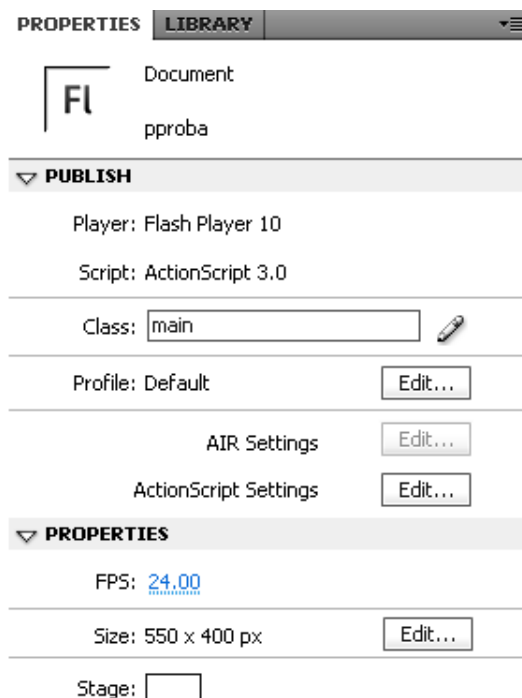


Рис. 4.3. Підключення Main.as до проекту

7. Експортуємо додаток. Натискаємо Ctrl + Enter. Запускається створений додаток. Щоб побачити результат, його потрібно завантажити в середовище Вконтакті. У папці з fla файлом з'являється swf файл. Його й потрібно завантажити в середовище Вконтакті.

8. Завантажуємо додаток у Вконтакті:


- на сайті Вконтакті, реєструємося у вашому обліковому записі.
- у самому нижньому горизонтальному меню, вибираємо пункт "розроблювачам" (рис. 4.4).
- - натискаємо "створити додаток", угорі сторінки (рис. 4.5).
- заповнюємо дані у запропоновані поля, натискаємо "перейти до завантаження додатка" (рис. 4.6).
- у наступному вікні натискаємо "огляд...", вибираємо створений swf файл і натискаємо "почати завантаження" (рис. 4.6).

о сайте техподдержка блог правила реклама **разработчикам** вакансии


ВКонтакте © 2006-2011 Русский
Павел Дуров

Рис. 4.4. Створення додатка Вконтакті


Разработчикам С чего начать	Документация ВКонтакте API	Примеры интеграции Оцените преимущества платформы	Подключить сайт
			Создать приложение



Приложения ВКонтакте (Flash/Iframe)
Интерактивные приложения используют **API ВКонтакте** для глубокой интеграции с сайтом. Вы можете использовать как сервера ВКонтакте, так и любое количество собственных серверов.
[Подробнее »](#)



Standalone/Mobile приложения
Запускаются в виде обычных программ на устройстве пользователя — компьютере или смартфоне. Все методы, доступные приложениям на платформе **API ВКонтакте**, работают и для desktop-приложений.
[Подробнее »](#)



Виджеты и сторонние сайты
Набор **виджетов ВКонтакте** для сторонних сайтов позволит моментально добавить к Вашему проекту социальную составляющую. Также возможна более глубокая интеграция через **Open API** или **OAuth 2.0**.
[Подробнее »](#)

Последние новости

- Изменение в инициализации Flash-приложений
28 июл 2011 в 22:18
- Методы для работы с документами
30 мая 2011 в 23:30
- Авторизация на базе OAuth 2.0 и упрощение API
11 апр 2011 в 14:01
- Изменение правил размещения рекламы в приложениях
10 фев 2011 в 15:10
- Обновление Flash-посредника и контейнера
8 фев 2011 в 13:36

[Предыдущие новости](#)

Полезная информация

- Партнёрская модель
- Локализация
- Правила размещения

Рис. 4.5. Створення додатка Вконтакті (2)

Создание приложения

Название:

Описание:

Категория:

Загружая приложение, Вы соглашаетесь с правилами размещения приложений

[Перейти к загрузке приложения](#)

Рис. 4.6. Створення додатка Вконтакті (3)

Тепер можна перевірити роботу додатка. На рис. 4.7. наведено результат виконання додатка.

Уважаемый



Алексей Евсеев
25.9.1977 г.р.

Рад видеть Вас на своей странице !

Рис. 4.7. Результат виконання додатка

У результаті виконання цього нескладного додатка ми бачимо на екрані інформацію про користувача, що зайшов: прізвище, ім'я, дату народження й аватар. Незважаючи на простоту, найцікавішою частиною додатка, є можливість отримання даних через API ВКонтакті. Отримані таким чином дані можна використати разом з іншими даними а також графікою й анімацією у вашому Flash додатку.

Завдання

1. Якщо у вас немає облікового запису ВКонтакті – зареєструйтесь.
2. Реалізуйте приклад з цієї самостійної роботи.
3. Спробуйте проаналізувати потреби користувачів соціальної мережі.
4. Сформулюйте вимоги до додатка для ВКонтакті та розробіть схему сценарію діалога з користувачем.
5. Реалізуйте власний додаток за допомогою Flash та розмістіть його ВКонтакті.

Самостійна робота № 5

Використання об'єктів офісних програмних продуктів у складі Flash презентації

Мета роботи: отримання знань та навичок, щодо використання даних додатків офісних програмних продуктів у flash-презентації.

Об'єктом СР є flash-презентація, до якої інтегруються дані з додатків офісних програмних продуктів.

Предметом СР є процес інтеграції даних із додатків офісних програмних продуктів (MS Excel) до flash-презентації.

Методи для виконання СР: аналіз та синтез, планування, розроблення контенту.

У результаті виконання СР формуються такі компетенції: перетворення таблиць MS Excel до формату XML, завантаження даних XML до flash-презентації.

Нормування та вихідний контроль: самостійна робота виконується в рамках вивчення теми № 8 "Створення презентацій на основі технології Macromedia Flash"; час необхідний для виконання – 4 години; результат роботи – звіт щодо процесу та результатів інтеграції даних з MS Excel до flash-презентації.

Контроль якості: диференційований.

Методичні рекомендації

Завантаження таблиць з Microsoft Excel

Одним з найпоширеніших завдань з інтеграції об'єктів офісних програмних продуктів до Flash є взаємодія з таблицями Microsoft Excel.

Розглянемо приклад завдання, яке пов'язане з управлінням графічними об'єктами у Flash шляхом використання даних з таблиць Microsoft Excel (координатів графічних об'єктів, їх розмірів та кольору).

Для того, щоб завантажити до Flash-презентації дані з таблиць Microsoft Excel, слід виконати наступну послідовність дій:

1. Перетворіть таблицю, створену в Microsoft Excel до формату XML з іменем shapes_data.xml (за допомоги панелі інструментів "Список", рис. 5.1; 5.2).

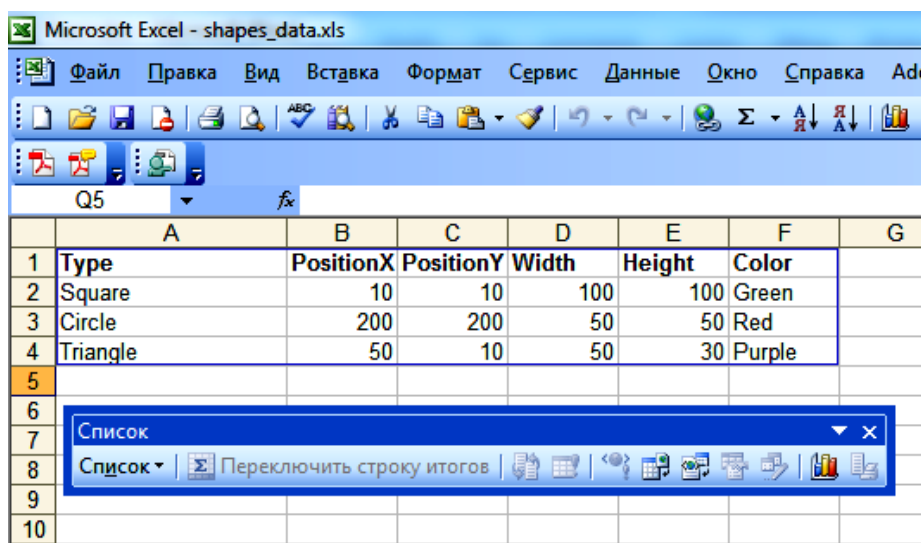


Рис. 5.1. Таблица з даними у Microsoft Excel та панель інструментів "Список"

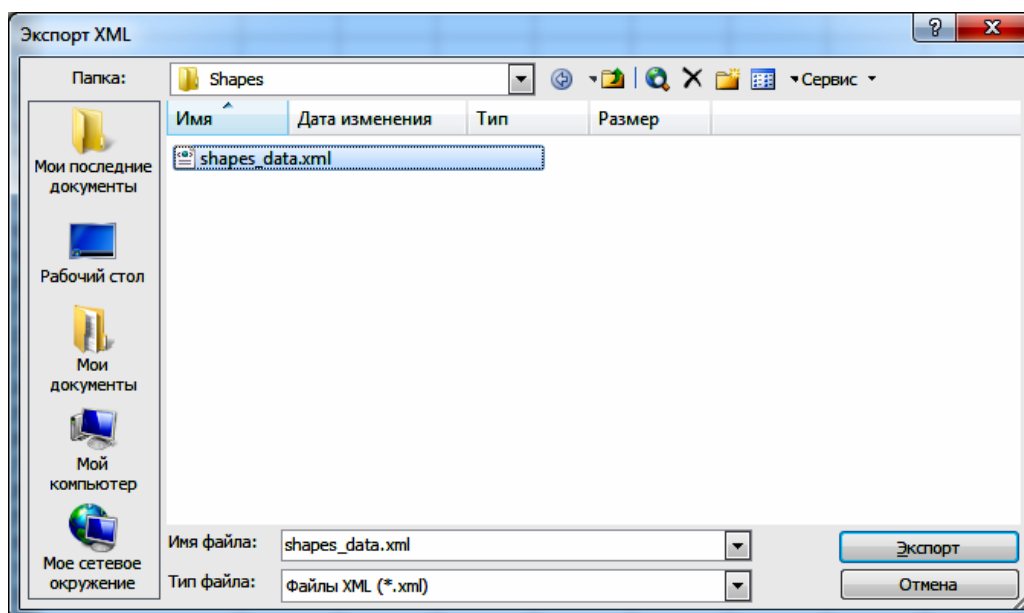


Рис. 5.2. Экспорт данных до XML

XML-файл повинен містити такий текст:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:Root xmlns:ns1="http://www.Knowing.net/devx/flash/shapes.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ns1:root>
    <ns1:Row>
      <ns1:Type>Square</ns1:Type>
      <ns1:PositionX>10</ns1:PositionX>
      <ns1:PositionY>10</ns1:PositionY>
      <ns1:Width>100</ns1:Width>
      <ns1:Height>100</ns1:Height>
      <ns1:Color>Green</ns1:Color>
    </ns1:Row>
    <ns1:Row>
      <ns1:Type>Circle</ns1:Type>
      <ns1:PositionX>200</ns1:PositionX>
      <ns1:PositionY>200</ns1:PositionY>
      <ns1:Width>50</ns1:Width>
      <ns1:Height>50</ns1:Height>
      <ns1:Color>Red</ns1:Color>
    </ns1:Row>
    <ns1:Row>
      <ns1:Type>Triangle</ns1:Type>
      <ns1:PositionX>200</ns1:PositionX>
      <ns1:PositionY>10</ns1:PositionY>
      <ns1:Width>50</ns1:Width>
      <ns1:Height>30</ns1:Height>
      <ns1:Color>0xFF4433</ns1:Color>
    </ns1:Row>
  </ns1:root>
</ns1:Root>
```

2. У програмі Adobe Flash створіть новий документ ActionScript 2.0 та для першого кадру запишіть такий програмний код:

```
var xml:XML = new XML();
xml.ignoreWhite = true;
xml.load("./shapes_data.xml");
xml.onLoad = function(success)
{
    if(success)
    {
        shapeParser = new ShapeParser(xml);
    }
}
```


3. У цьому ж документі створіть три символи типу MovieClip, де намалюйте коло, трикутник та квадрат відповідно (рис. 5.3). Задайте змінні до кожного з цих символів (CircleMovie, SquareMovie, TriangleMovie) для подальшого управління ними за допомоги програмного коду (рис. 5.4).

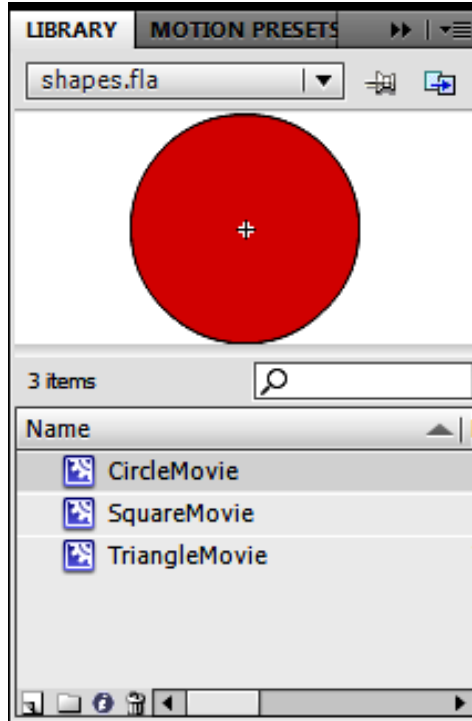


Рис. 5.3. Символи у бібліотеці

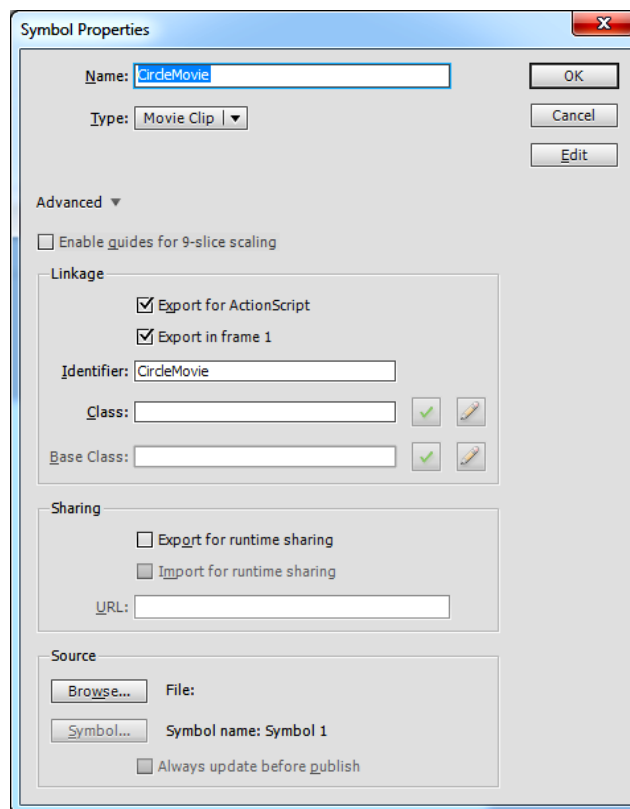


Рис. 5.4. Приклад налаштування символу

4. Далі створіть обробники поведінки кожної фігури (файли CircleMovie.as, SquareMovie.as, TriangleMovie.as), а також файл із кодом, який буде керувати проектом загалом (Shape.as) та файл із парсером, який дозволить зчитати дані з XML-файлу (ShapeParser.as). Усі файли повинні лежати в одній папці із основним файлом проекту.

Код файлу Circle.as:

```
class Circle extends Shape
{
    function Circle(xml:XMLNode)
    {
        super();

        _root.attachMovie("CircleMovie",shapeName,_root.getNextHighestDepth());
        setPosition(xml);
    }
}
```

Код файлу Square.as:

```
class Square extends Shape
{
    function Square(xml:XMLNode)
    {
        super();

        _root.attachMovie("SquareMovie",shapeName,_root.getNextHighestDepth());
        setPosition(xml);
    }
}
```

Код файлу Triangle.as:

```
class Triangle extends Shape
{
    function Triangle(xml:XMLNode)
    {
        super();

        _root.attachMovie("TriangleMovie",shapeName,_root.getNextHighestDepth())
```

;

```

        setPosition(xml);
    }
}

```

Код файла Shape.as:

```

class Shape
{
    static var shapeCount:Number = 0;
    var shapeld:Number;
    var shapeName:String;
    function Shape()
    {
        shapeld = Shape.shapeCount++;
        shapeName = shapeld.toString();
    }

    function setPosition(xml:XMLNode)
    {
        var posXNode:XMLNode = xml.childNodes[1];
        _root[shapeName]._x = parseIntNode(posXNode);

        var posYNode:XMLNode = xml.childNodes[2];
        _root[shapeName]._y = parseIntNode(posYNode);

        var widthNode:XMLNode = xml.childNodes[3];
        _root[shapeName]._width = parseIntNode(widthNode);

        var heightNode:XMLNode = xml.childNodes[4];
        _root[shapeName]._height = parseIntNode(heightNode);

        var colorNode:XMLNode = xml.childNodes[5];
        setColorFromNode(_root[shapeName], colorNode);
    }

    function parseIntNode(node:XMLNode):Number
    {
        var num:Number = parseInt(node.childNodes[0].nodeValue);
        trace(num);
        return num;
    }
}

```

```

    }

function setColorFromNode(target:MovieClip, node:XMLNode)
{
    var color:String = node.childNodes[0].nodeValue;
    var rgb:Number;
    switch(color)
    {
        case "Red" :
            rgb = 0xFF0000;
            break;
        case "Green" :
            rgb = 0x00FF00;
            break;
        default:
            rgb = parseInt(color);
            break;
    }
    var shapeColor:Color = new Color(target);
    shapeColor.setRGB(rgb);
}
}

```

Код файла ShapeParser.as:

```

class ShapeParser
{
    var shapes:XMLNode;
    function ShapeParser(xml:XML)
    {
        //Get to the "Shapes" list
        //Get to "Row" element
        shapes = xml.firstChild.firstChild;
        trace(shapes);
        for(var i:Number = 0; i < shapes.childNodes.length; i++)
        {
            var rowNode:XMLNode = shapes.childNodes[i];
            parseRow(rowNode);
        }
    }
}

```

```

    }

    function parseRow(rowNode:XMLNode)
    {
        var typeNode:XMLNode = rowNode.firstChild;
        var stringNode:XMLNode = typeNode.childNodes[0];
        var type:String = stringNode.nodeValue;
        trace("Type is " + type);
        var shape:Shape;
        switch(type)
        {
            case "Square" :
                shape = new Square(rowNode);
                break;
            case "Circle" :
                shape = new Circle(rowNode);
                break;
            case "Triangle" :
                shape = new Triangle(rowNode);
        }
    }
}

```

5. Запустіть проект та перегляньте відповідність розташування та властивостей кожної з фігур до даних з таблиці. Змініть дані в таблиці та перезапустіть проект.

Завдання

1. Створіть таблицю з даними довільної тематики у MS Excel (повинно бути не менш, ніж 5 стовбців та не менш, ніж 10 записів).
2. Перетворіть створену таблицю до формату XML.
3. Користуючись методичними рекомендаціями, завантажте XML-дані до flash-презентації, розробленої за тією ж тематикою, що й таблиця з даними.
4. Змініть дані у таблиці MS Excel та виконайте п. 2 завдання.
5. Перевірте, чи змінились дані у flash-презентації.
6. Процес виконання самостійної роботи та її результати оформіть у вигляді звіту.

Самостійна робота № 6

Розробка смарткліпу з використанням мови програмування ActionScript. Розробка елементів 3D інтерфейсу

Мета роботи: поглиблення знань, які було отримано на лекції, та підтвердження і реалізація навичок, що були сформовані на лабораторній роботі за темою смарткліпи.

Об'єктом СР є елементи 3D інтерфейсу для смарткліпу.

Предметом СР є вивчення особливостей функціонування та реалізація елементів інтерфейсу, що імітують тривимірний простір.

Методи для виконання СР: проектування інтерфейсів, аналіз інформаційних потоків.

У результаті виконання СР формуються такі компетенції: вміння реалізувати елементи 3D інтерфейса засобами ActionScript.

Нормування та вихідний контроль: самостійна робота виконується в рамках вивчення теми № 11 "Смарткліпи"; час необхідний для виконання – 4 години; елементи 3D інтерфейсу.

Контроль якості: диференційований.

Методичні рекомендації

Велику увагу розроблювачі сучасних додатків приділяють інтерфейсу взаємодії з користувачем. Інтерфейс повинен бути не просто зручним інструментом передавання інформації користувачеві, а становити спеціальне середовище, створене для комфортного й максимально-ефективного сприйняття інформації. В умовах, коли запропонована користувачеві інформація об'ємна, різноманітна й мультимедійна, інструменти інтерфейсу повинні відповідати новим вимогам, що висуваються до роботи з великими обсягами мультимедійних даних. Одним з підходів, що дозволяє оптимізувати роботу з багатомірними масивами мультимедійної інформації, є використання інструментів інтерфейсу, які дозволяють подавати роботу з додатками в тривимірному форматі.

Сучасні вимоги до інтерфейсів веб-додатків є не менш строгими по відношенню до локальних мультимедійних програм. Тому розроблювачам потрібний інструмент, який би дозволив створювати елементи 3D інтерфейсів для веб-додатків. Як такий, інструмент може бути обраний Adobe Flash з вбудованою до нього мовою програмування Action Script.

Розглянемо створення тривимірного елемента меню у вигляді каруселі. Ми будемо використовувати TweenMax для анімації елементів меню. Тому необхідно завантажити TweenMax для AS3 <http://www.greensock.com/tweenmax>. TweenMax збереже нам багато часу на програмування анімації. Необхідно зберегти "GS" папку в тій же папці, де перебуває ваш FLA-файл.

Створення нового документа

Створіть новий Flash ActionScript 3 документ із наступними властивостями (рис. 6.1).

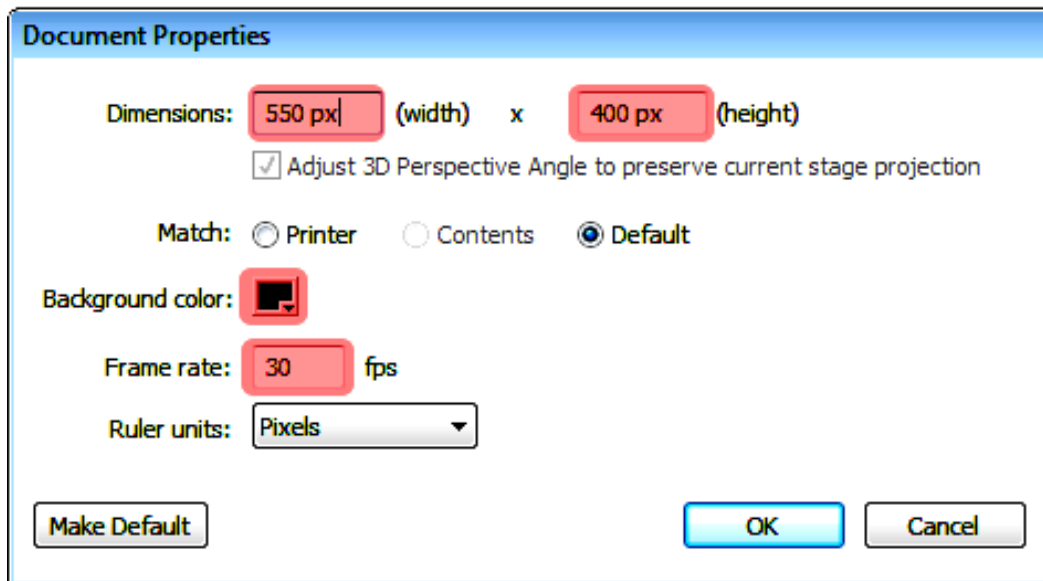


Рис. 6.1. Створення нового документа

Форма елемента меню

Намалюйте прямокутник з округленими кряями з наступними характеристиками (рис. 6.2).

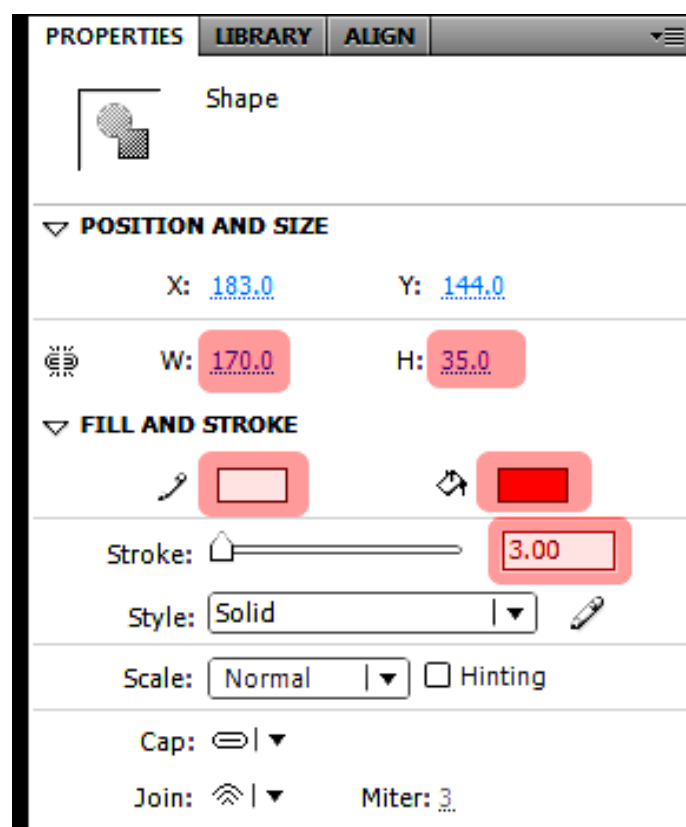


Рис. 6.2. Створення форми елемента меню

Елемент меню *Movie Clip*

Перетворіть прямокутник в movie clip (виберіть прямокутник і натисніть F8). Привласніть ім'я для нього "Menu Item" і встановіть реєстраційну точку на центр. Також зв'яжіть прямокутник із класом названим "MenuRectangle" (рис. 6.3). Ми зв'язали елемент меню із класом, тому що ми будемо створювати ці елементи меню за допомогою ActionScript.

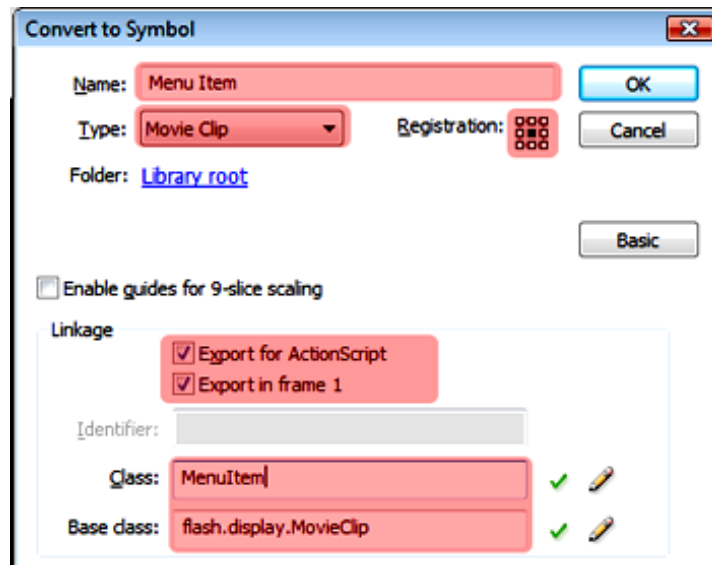


Рис. 6.3. Перетворення у символ

Шар тексту

Зробіть подвійний клік на movie clip елементі меню. З'явившись усередині movie clip, створіть новий шар з ім'ям "menu text" (рис. 6.4).

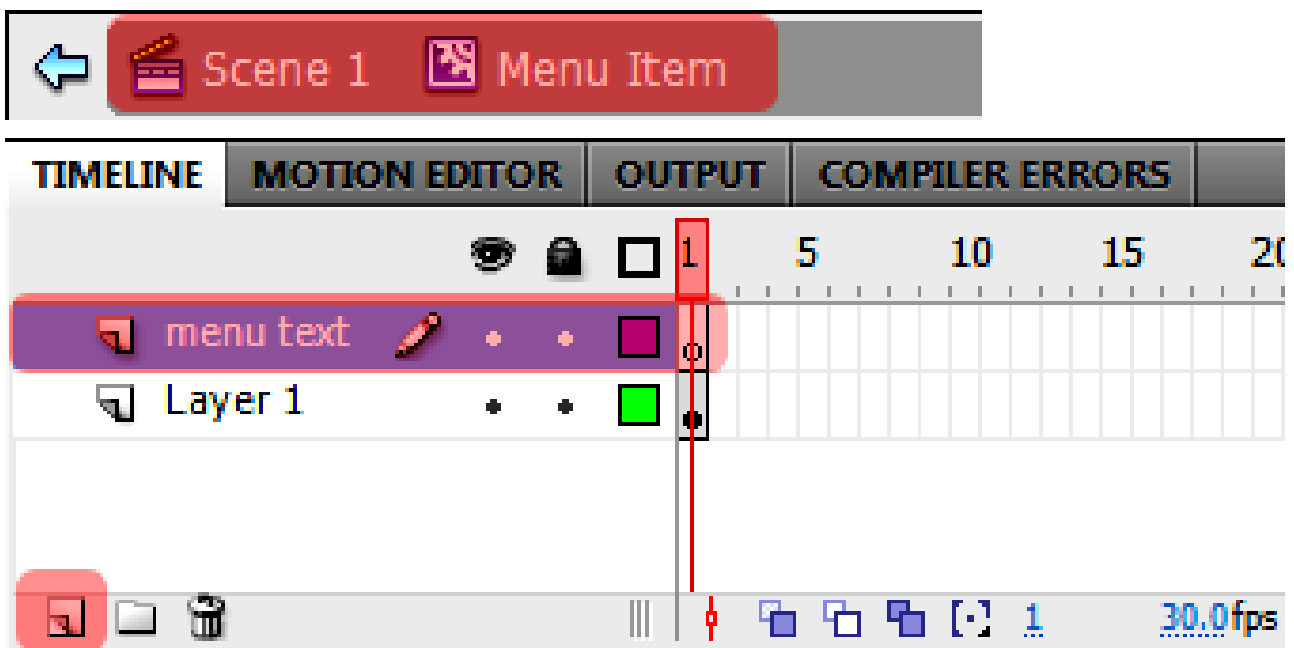


Рис. 6.4. Створення нового шару

Текст елемента меню

На шарі "menu text" створіть текстове поле **dynamic**, щоб воно перебувало нагорі прямокутника. Введіть деякий текст у ньому й встановіть наступні властивості (рис. 6.5). Ми привласнюємо текстовому полю **instance name** "menuText".

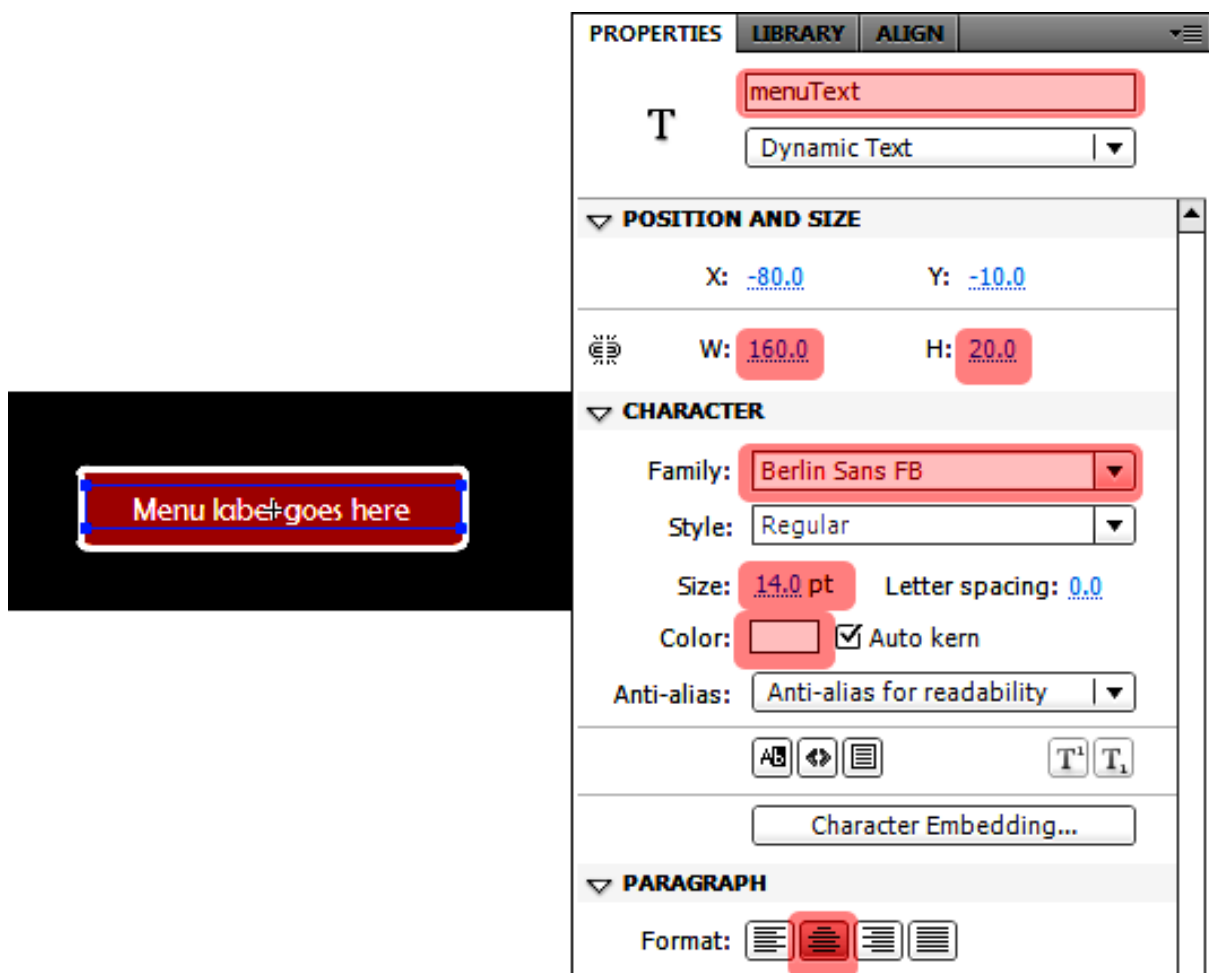


Рис. 6.5. Додавання тексту

Вбудовані символи

Для того, щоб текст виглядав гладким в Flash фільмі, необхідно вмонтувати деякі символи. Для цього – поки виділене текстове поле "menuText", натисніть кнопку "Character Embedding" (рис. 6.6) і додайте прописні й рядкові символи (додайте більше, якщо будете використовувати більше символів).

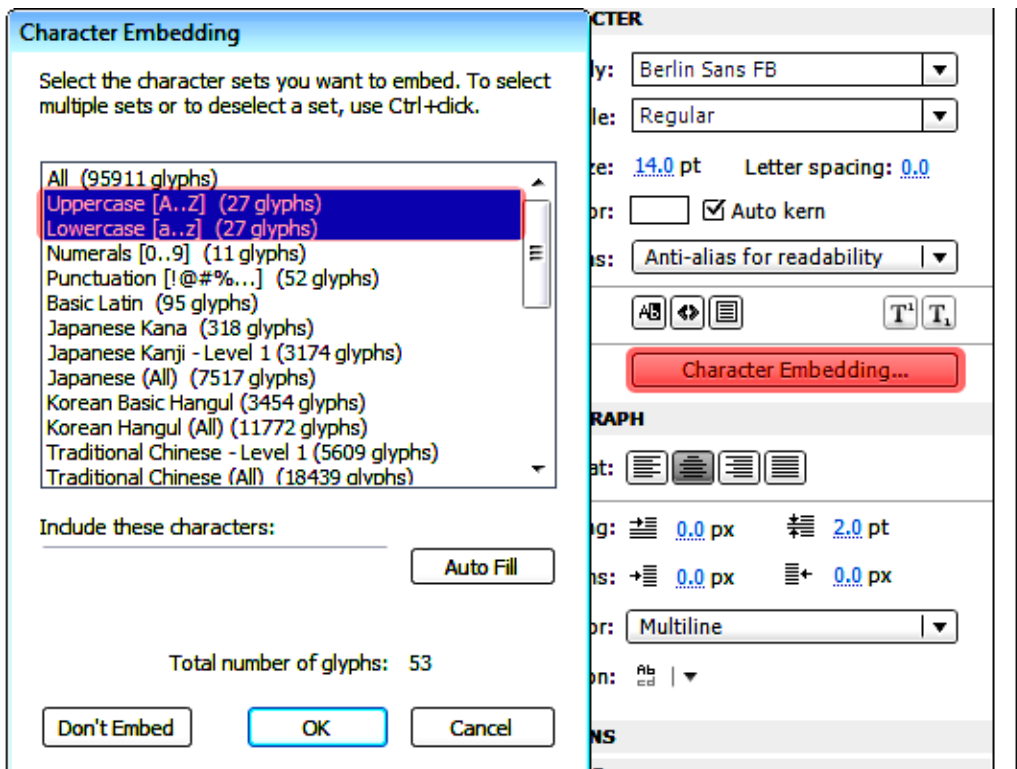


Рис. 6.6. Вбудовані символи

Завершальний крок

Тепер повертайтеся на головну лінійку часу й заберіть елемент меню зі сцени. Ваш сцена повинна стати повністю порожньою.

Створення XML файлу

Тепер ми готові до створення XML файлу, з якого ми будемо завантажувати напис для елементів меню URL адреси, на які вони будуть посилатися. За допомогою текстового редактора наберіть наступне або отримайте необхідний код у викладача.

```

<menu>
  <items>
    <item>
      <label>Home</label>
      <linkTo>http://tutorials.flashmymind.com</linkTo>
    </item>
    <item>
      <label>Tutorials</label>
      <linkTo>http://tutorials.flashmymind.com/complete-list-of-flash-and-actionscript-3-tutorials/</linkTo>
    </item>
  
```

...

Частина коду пропущено. Або додайте самостійно за аналогією, або отримайте повний варіант коду у викладача

```

</items>
</menu>

```

Структура XML файла проста. Для кожного елемента ми вказуємо підпис і URL, на який він повинен посилатися. Збережіть цей файл в обраному місці.

Спочатку необхідно імпортувати TweenMax, тому що він знадобиться нам пізніше під час анімації. Після цього ми вказуємо, де розташований XML файл, змінюємо шлях відповідно до настроювань. Після цього ми завантажуюмо XML файл і викликаємо **xmlLoaded()** функцію. У функції **xmlLoaded()** ми створюємо новий XML об'єкт із завантажених XML даних. У завершенні ми запускаємо функцію **createMenu()**.

Насправді, код для цього проекту досить простий. Спочатку оголошуються деякі додаткові змінні, які нам необхідні. У функції **createMenu()** ми містимо в цикл всі елементи, знайдені в XML файлі. Ми розташовуємо елемент в 3D окружності й після цього перетворюємо їх 3D координати в 2D координати.

Ми додаємо напис до кожного елемента й створюємо "linkTo" атрибут, що буде зберігати URL для кожного елемента меню.

Ми анімуємо елемент, коли миша переміщається над ним або поруч із ним. Ми використовуємо TweenMax, щоб дещо спростити нашу роботу. Ви можете легко додати більше анімації, використовуючи TweenMax. Ви можете змінювати значення, щоб переконатися, як буде відобразитися меню.

Коли по елементу натискають, ми переходимо по URL, який був прив'язаний до елемента меню.

У кожному кадрі ми повертаємо карусель відповідно до положення курсора миші. Ми обчислюємо нові 3D-координати, а потім перетворюємо їх в 2D координати. Ми фактично тільки змінюємо у та z координати, тому що це вертикальна карусель. В sort () ми призначаємо нові індекси підлеглого елемента відповідно до Z-координати елементів. Таким чином, елементи перекривають один одного коректно, що дає нам ефект 3D.

Програмний код пересування каруселі, оброблювачі подій mouse, створення 3D меню, завантаження XML ви можете запитати у викладача.

На основі розглянутого вище засобу можна створити й багато інших елементів інтерфейсу для веб-додатків, що дозволяють організувати введення й перегляд мультимедійної інформації.

Завдання

1. Реалізуйте приклад, що наведено у цій самостійній роботі.
2. Змініть дані XML файла, з якого отримується інформація для елементів меню, та спробуйте ускладнити його структуру.
3. Спробуйте додати до елементів меню графічні зображення.
4. Спробуйте створити вкладені підменю, які б розкривалися після вибору елементів головного рівня.

Самостійна робота № 7

Розробка презентації з використанням кліпу, розробленого у середовищі 3D MAX Studio, що моделює роботу пристрою

Мета роботи: отримання практичних навичок щодо використання плагіну Illustrate для 3DS Max з метою експорту анімації роботи тривимірного пристрою до середовища розробки AdobeFlash.

Об'єктом СР анімації роботи тривимірного пристрою.

Предметом СР є процес експорту анімації роботи тривимірного пристрою до середовища розробки AdobeFlash.

Методи для виконання СР: аналіз та синтез, планування, розроблення контенту.

У результаті виконання СР формуються такі компетенції: налаштування плагіну Illustrate у середовищі 3D MAX Studio для коректного експорту анімації у векторному форматі до AdobeFlash.

Нормування та вихідний контроль: самостійна робота виконується в рамках вивчення теми № 13 "Створення анімованих документів у середовищі 3D MAX Studio"; час, необхідний для виконання – 4 години; результат роботи – звіт щодо перенесення тривимірної анімації до AdobeFlash.

Контроль якості: диференційований.

Методичні рекомендації

Проблема використання тривимірної анімації, створеної у середовищі 3DS Max, у Flash завжди була актуальною. Складність процесу полягає в тому, що продукти, зроблені у 3DS Max не призначені для використання в Інтернет і тому їх потрібно оптимізувати. Найпростішим шляхом є перетворення тривимірної анімації до відеофайлу та імпорт його до Flash, але наслідком такого процесу є великий обсяг файлу, що може призвести до повільної роботи Flash у мережі Інтернет.

Ще одним шляхом вирішення проблеми є використання у 3DS Max плагіна Illustrate, який перетворює анімацію у послідовність векторних об'єктів та дозволяє імпортувати ці об'єкти до ключових кадрів Flash.

Істотним недоліком застосування цього плагіну є неможливість застосування текстур. Це й зрозуміло – адже растрові текстури займають більше місця, ніж векторні файли, а мінімізація розмірів є однією з основних причин, по яким до основи Flash покладена векторна графіка. Інший, найбільш вагомий недолік цього модуля полягає в тому, що при рендерінгу до векторних форматів він не передає півтонів і колірних переходів. До того ж, знову-таки, при рендерінгу саме до таких форматів виявляються деякі додаткові похибки прорахунку. Не завжди коректно відпрацьовується глибина, тобто деякі деталі заслонених об'єктів іноді

вилазять вперед. Особливість цього модуля в тому, що кожен об'єкт тривимірної сцени стає окремою групою в swf файлі і може редагуватися окремо від інших, тому навіть частково закриті об'єкти сцени насправді прораховуються повністю, і у Flash-редакторі можна це побачити. Лінії прораховуються коректно, саме як лінії (при перенесенні зображень з векторних редакторів до Flash, лінії часто імітуються геометричними фігурами, що збільшує розмір файла).

Найзручніший спосіб налаштувати плагін Illustrate – скористуватися майстром. Для цього слід зробити таке: відкрийте вікно налаштувань плагіну (Illustrate > Options) та виберіть пункт меню "Rendering Wizard". Відкриється вікно майстра, у якому слід відповісти на такі питання:

- вихідний формат зображення, що буде рендеритися;
- призначення Illustrate-рендера "активним рендером";
- вибір кольору заднього фону;
- визначення стилю рендеринга всіх об'єктів (можливі варіанти показані на рис. 7.1);
- вихідні параметри (номер кадру, ширину і висоту зображення).

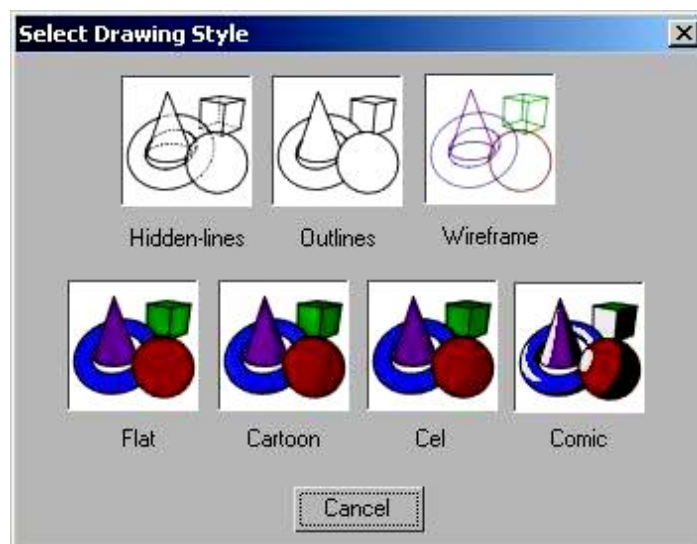


Рис. 7.1. **Можливі варіанти відображення графіки після рендерингу**

Після цього слід закінчити роботу майстра та виконати рендеринг анімації.

Завдання

1. Відкрийте файл з анімацією механічного пристрою, створеного в рамках виконання комплексного курсового проекту, у 3DS Max.
2. Налаштуйте плагін Illustrate для коректного експорту анімації для подальшого її відтворення у AdobeFlash.
3. Імпортуйте отриману анімацію до презентації, створеної у AdobeFlash.
4. Проаналізуйте роботу анімації тривимірного пристрою та порівняйте її із анімацією у 3DS Max.
5. Оформіть результати роботи у вигляді звіту.

Рекомендована література

1. Приложение, показывающее имя, фамилию, дату рождения, аватар пользователя [Электронный ресурс]. – Режим доступа : <http://easyflash.org/flashlearn/flatexttutorials/798-httpflappsruforum-topic3html.html> (10.10.11).
2. Vertical 3D Carousel with ActionScript 3 and XML [Electronic resource]. – Access mode : <http://tutorials.flashmymind.com/2009/05/vertical-3d-carousel-with-actionscript-3-and-xml> (11.10.11).
3. Mandelbulber. User manual [Electronic resource]. – Access mode : <http://sites.google.com/site/mandelbulber/user-manual> (08.10.11).
4. Illustrate 5.1 и его возможности [Электронный ресурс]. – Режим доступа : <http://mania3d.narod.ru/t-Dateien/illustrator.htm> (07.10.11).
5. How To Extract SWF Flash From Excel or Word [Electronic resource]. – Access mode : <http://www.walkernews.net/2008/03/22/how-to-extract-swf-flash-from-excel-or-word/> (11.10.11).

Зміст

Вступ	3
Змістовний модуль I. Створення комп'ютерної анімації за допомогою технології Macromedia Flash	3
Самостійна робота № 1. Використання інструментів роботи з кольором та текстом	3
Самостійна робота № 2. Розробка кліпу з використанням анімації розкадрування та шарів. Створення фрактальної графіки	8
Самостійна робота № 3. Оптимізація імпортованої фото- і відеоінформації у анімаційному кліпі. Оптимізація подання кліпу та його складників для використання у різних задачах	18
Змістовний модуль II. Використання Macromedia Flash-технології та 3DS MAX для створення сучасних мультимедійних документів	24
Самостійна робота № 4. Аналіз сучасних тенденцій та інструментів, що використовуються при створенні Flash анімації для маркетингу. Створення додатків для соціальних мереж	24
Самостійна робота № 5. Використання об'єктів офісних програмних продуктів у складі Flash презентації	30
Самостійна робота № 6. Розробка смарткліпу з використанням мови програмування ActionScript. Розробка елементів 3D інтерфейсу	38
Самостійна робота № 7. Розробка презентації з використанням кліпу, розробленого у середовищі 3D MAX Studio, що моделює роботу пристроя	44
Рекомендована література	46

НАВЧАЛЬНЕ ВИДАННЯ

**Методичні рекомендації
до самостійної роботи
з навчальної дисципліни
"КОМП'ЮТЕРНА АНІМАЦІЯ"**

**для студентів напрямку підготовки
6.051501 "Видавничо-поліграфічна справа"
спеціалізації "Технології електронних мультимедійних видань"
усіх форм навчання**

**Укладачі: Євсєєв Олексій Сергійович
Прибиткова Наталія Іванівна**

Відповідальний за випуск Пушкар О. І.

Редактор Бутенко В. О.

Коректор Бриль В. О.

План 2012 р. Поз. № 363.

Підп. до друку Формат 60×90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. 3,0. Обл.-вид. арк. 3,75. Тираж прим. Зам. №

Видавець і виготівник – видавництво ХНЕУ, 61166, м. Харків, пр. Леніна, 9а

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № 481 від 13.06.2001 р.*

**Методичні рекомендації
до самостійної роботи
з навчальної дисципліни**

"КОМП'ЮТЕРНА АНІМАЦІЯ"

**для студентів напряму підготовки 6.051501
"Видавничо-поліграфічна справа" спеціалізації
"Технології електронних мультимедійних видань"
усіх форм навчання**