

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

**Методичні рекомендації
до виконання контрольної роботи
з навчальної дисципліни
"КОМП'ЮТЕРНА АНІМАЦІЯ"
для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
спеціалізації "Технології електронних
мультимедійних видань"
заочної форми навчання**

Харків. Вид. ХНЕУ, 2013

Затверджено на засіданні кафедри комп'ютерних систем і технологій.
Протокол № 4 від 06.12.2011 р.

Укладачі: Євсєєв О. С.
Завгородня О. С.

M54 Методичні рекомендації до виконання контрольної роботи з навчальної дисципліни "Комп'ютерна анімація" для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" спеціалізації "Технології електронних мультимедійних видань" заочної форми навчання / укл. О. С. Євсєєв, О. С. Завгородня. – Х. : Вид. ХНЕУ, 2013. – 48 с. (Укр. мов.)

Наведено комплексне завдання та допоміжний матеріал для засвоєння на практиці й отримання відповідних компетентностей щодо вивчення даної навчальної дисципліни. Запропоновано матеріал, який дасть змогу в рамках практичних занять у стислий час отримати всі практичні навички, що є складовими частинами компетентностей дисципліни, та виконати контрольну роботу, яка є показником якості отриманих знань.

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

Вступ

Сучасні інформаційні технології дозволяють не тільки інтенсифікувати більшість процесів, що взаємодіють у видавничо-поліграфічній діяльності, але й якісно змінити методи розробки продукції та її доведення до кінцевого споживача. Комп'ютерна анімація є одним із напрямків розвитку видавничої діяльності й широко використовується в процесі створення мультимедійних видань, web-орієнтованих проектів і рекламної продукції. Комп'ютерна анімація дозволяє не тільки поліпшити мультимедійне видання шляхом підвищення його наочності й візуального ефекту, але й найчастіше спростити завдання розробки мультимедійного матеріалу, скоротивши при цьому витрати.

У процесі навчання студенти заочної форми отримують необхідні теоретичні знання під час лекційних занять, у рамках практичних занять набувають практичних навичок, що є складовими частинами компетентностей цього курсу, та виконують контрольну роботу, яка є показником якості отриманих знань.

Метою контрольної роботи є поглиблення теоретичних знань та отримання навичок завдяки реалізації на практиці методів розробки анімаційних роликів з використанням творчого підходу та додатків з елементами анімації. Важливим завданням є формування компетентностей, що дозволяють студенту реалізовувати на практиці отримання знання й розробляти такі анімовані документи та додатки, що є актуальними в сучасних умовах, мають попит на ринку web-послуг на сьогодні та потенціал щодо подальшого розвитку.

Така робота потребує від студентів також самостійного опрацювання спеціальних інформаційних джерел та інструментальних засобів, що дозволяють застосувати творчий підхід до анімації та врахувати всі новітні прийоми. Рекомендується працювати з періодичними мультимедійними виданнями, які відбивають тенденції розвитку сучасних інформаційних технологій, сучасні технології в анімації, дизайні й програмуванні. Підготовка до такої контрольної роботи потребує від студентів також поглиблення знань у суміжних дисциплінах, таких як прикладна математика, основи композиції та дизайну, основи об'єктно-орієнтованого програмування.

Запропоновані методичні рекомендації дозволяють у повному обсязі підготуватися до контролю знань з навчальної дисципліни "Комп'ютерна анімація" для студентів спеціалізації "Технологія електронних мультимедійних видань" заочної форми навчання. Методичні рекомендації містять розгорнуту структуровану додаткову інформацію, що дозволяє у стислий час отримати всі необхідні практичні навички для виконання контрольної роботи.

Для виконання головного завдання контрольної роботи – створення комплексного анімаційного проекту – студентові необхідно отримати всі необхідні складові компетентності. Для підвищення якості знань, що отримуються, кожна зі складових компетентностей, супроводжується окремим допоміжним матеріалом. Тому для виконання контрольної роботи рекомендується спочатку ознайомитися з головним завданням, потім – з усіма компонентами допоміжного матеріалу та виконати необхідні складові завдання, а після цього почати створювати комплексний анімаційний проект.

Контрольна робота "Розробка комплексного анімаційного мультимедійного проекту"

Компетентності, що формуються у процесі виконання контрольної роботи:

вміння створювати цілісний анімаційний проект (програмний додаток), який містить у своєму складі різні види анімації та засоби інтерактивної взаємодії з користувачем.

Запропонована контрольна робота дозволяє перевірити всю сукупність знань та навичок, які було отримано в процесі засвоєння дисципліни "Комп'ютерна анімація". Для виконання контрольної роботи студентові необхідно опанувати та засвоїти всі практичні навички, що відповідають темам, які викладалися протягом вивчення дисципліни.

Засвоївши необхідні практичні навички та здобувши відповідні компетентності, необхідно навести результати виконання контрольної роботи у вигляді цілісного анімаційного проекту з інтерактивними елементами. Для якісної реалізації складного анімаційного проекту треба чітко уявляти собі його склад та структурні елементи, сформулювати основну ідею та перерахувати набір елементів, що будуть анімовані.

Для системного подання вимог до реалізації рекомендується скористатися наступною послідовністю дій.

1. Придумайте назву анімаційного проекту, що буде розроблятися.
2. Опишіть основну ідею (призначення створення або основну мету).
3. Зробіть поділ анімації на окремі сцени, опишіть їхню суть.
4. Виконайте для обраних сцен окремі об'єкти, як анімовані, так і статичні.
5. Опишіть характер поведження анімованих об'єктів.
6. Опишіть конструкцію окремих об'єктів (з яких фігур, примітивів, зображень складається кожний об'єкт).
7. Опишіть цілі додавання програмного малювання та анімовані об'єкти поведження яких керується за допомогою комп'ютера.
8. Опишіть зміст форм та компонентів, які додаються до анімаційного проекту, яка інформація буде збиратися та відображатися за їх допомогою.

Наведені компоненти запропонованої послідовності необхідно письмово сформулювати та включити до складу майбутнього звіту. Також необхідно навести сценарій анімаційного проекту, який можна супроводжувати чернетками креслень окремих сцен, алгоритмами поводження окремих об'єктів та ін.

Для реалізації анімаційного проекту треба оволодіти цілим рядом компетентностей. До уваги студентів пропонується дуже детальна підтримка кожного кроку процедури реалізації, поділена на окремі складові. Складові частини, що необхідно засвоїти для отримання відповідних компетентностей:

- створення покадрової анімації і анімації за допомогою розкадровування руху (Motion Tween та Classic Tween);

- створення анімації за допомогою розкадровування форми (Shape Tween);

- шар, що спрямовує рух і шар маски;
- символи-контейнери;
- створення анімованої кнопки (банера);
- імпортування і оптимізація об'єктів; робота з відео і звуком;
- динамічні текстові поля; активна кнопка;
- програмування на ActionScript;
- робота з компонентами Flash;
- програмне малювання.

У цих методичних рекомендаціях до кожної складової наведено допоміжний матеріал, що дозволить оволодіти необхідними компетентностями на більш високому рівні та у стислий час.

Завдання

Результатом виконання цієї контрольної роботи повинен стати анімаційний кліп, який буде містити всі основні компоненти, які були подані на лекційних заняттях. Анімаційний проект може бути побудований у вигляді презентаційного матеріалу, рекламного ролика, історичної довідки, загадки з відповіддю, яку можна буде побачити в кінці кліпу. Проект повинен мати власний сюжет, і якщо обрано реалізацію

загадки, то кінцевою точкою буде отримання відповіді на запитання у непрямому вигляді або як підказки до загадки у фінальній сцені кліпу або за замовленням користувача. Для рекламного ролика, історичної довідки та інших видів реалізації кліпу також було б доцільно додати інтерактивну компоненту у фінальній сцені (голосування, коментування, подання відповіді або власної думки та ін.). Далі наведено складові мультимедійної листівки:

1. Наявність растрової графіки.

2. Символи (Graphic, MovieClip, Button). Для символу Button створювати обробник коду не обов'язково. Важливо, щоб після наведення/натискання на кнопку вона змінювалася.

3. Звуковий супровід.

4. Анімація маски.

5. Анімація спрямовальним шляхом.

6. Морфінг.

7. Розкадровування.

8. Анімація тексту.

Результат роботи повинен складатися з двох файлів:

- результату (*.fla)
- flash-фільму (*.swf)

Зміст мультимедійної листівки повинен відповідати основним композиційним і естетичним принципам.

Методичні рекомендації

Складова "Створення покадрової анімації та анімації за допомогою розкадровування руху (Classic Tween та Motion Tween)"

Компетентності, що формуються завдяки цій складовій частині:

- практичні навички створення покадрової анімації;
- вміння створювати анімацію на основі розкадровування руху.

Допоміжний матеріал

Створення gif-анімації за допомогою інструментарію Adobe Flash та графічного редактора.

Для створення gif-анімації за допомогою інструментарію Adobe Flash та графічного редактора необхідно виконати такі дії:

1. Намалювати в будь-якому графічному редакторі (наприклад, Adobe Photoshop) рисунок для анімації. Усі статичні елементи повинні знаходитися на одному шарі, всі елементи, до яких застосовуватиметься анімація, на різних шарах (рис. 1).

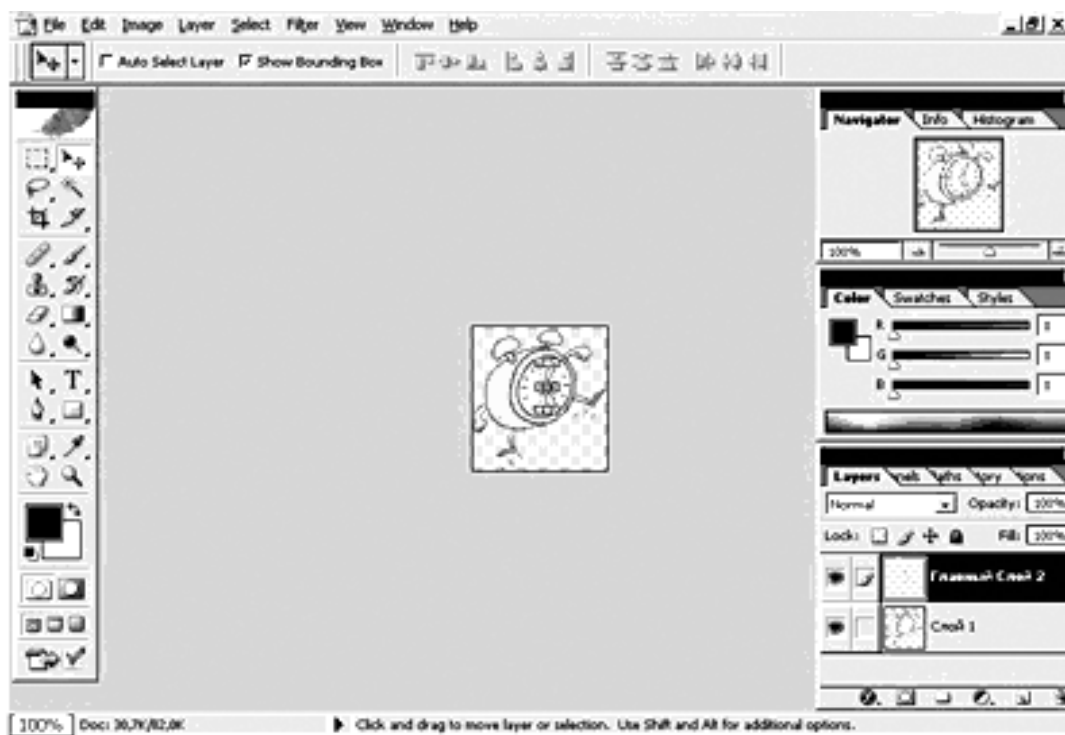


Рис. 1. Створення рисунка в Adobe Photoshop

2. Зберегти файл.

3. Завантажити створений рисунок в Adobe Flash. Якщо рисунок складається з об'єктів на різних шарах, то всі дії виконуються лише до об'єктів, що мають рухатися під час анімації. У першому кадрі до об'єктів, які необхідно анімувати застосовується команда F8 "Графика" (або викликається контекстне меню та обирається "Преобразовать в символ"/Тип "Графика").

4. Потім необхідно обрати, наприклад, 20 кадр та зробити його ключовим. На цьому кадрі необхідно внести зміни до об'єктів порівняно з першим кадром: зрушити їх, змінити їх розмір.

5. Для створення анімації руху необхідно перейти на перший кадр та обрати в контекстному меню "Создать классическую анимацию движения". У результаті між першим та двадцятим кадрами з'явиться фіолетова стрілочка. Якщо ж анімація руху зроблена неправильно, то з'явиться синій пунктир.

6. Для перегляду створеної анімації руху необхідно натиснути Ctrl+Enter. Після перегляду можна скоректувати сформовану анімацію в разі потреби.

7. Якщо анімація відтворюється дуже швидко, можна збільшити кількість неключових кадрів між ключовими.

8. Для того щоб зберегти анімацію у формат GIF, необхідно вибрати "Файл"/ "Экспорт"/ "Экспортировать ролик". Для файла, що створюється, необхідно обрати розширення *.gif і зберегти його з потрібним ім'ям.

9. Для того щоб зберегти анімацію у форматі SWF (Flash Animation), потрібно вибирати "Файл"/"Экспорт"/"Экспортировать ролик". Для файла, що створюється, необхідно обрати розширення *. SWF і зберегти його з потрібним ім'ям.

Створення покадрової анімації за допомогою Adobe Flash CS5

Покадрова анімація – це анімація, що створена на основі ключових кадрів. Користувач сам визначає як вміст кадру, так і його "тривалість" (скільки статичних кадрів займатиме зображення).

На часовій шкалі покадрова анімація виглядає так, як показано на рис. 2.



Рис. 2. Покадрова анімація

Переваги покадрової анімації полягають у наявності, в певному розумінні, більшого контролю над анімацією. Це єдиний спосіб організувати зміну абсолютно незалежних зображень – слайд-шоу (наприклад, створюючи звичайний банер засобами Flash).

До недоліків слід віднести складність модифікації (особливо, якщо це не дискретний набір зображень, а зв'язана анімація); покадрова анімація має чималий об'єм, оскільки доводиться зберігати інформацію про кожен кадр.

Розкадровування руху дозволяє швидко створювати цікаві анімаційні рішення (переміщення об'єктів, обертання, зміну кольору, прозорості тощо), не вимагаючи при цьому від автора монотонних перетворень об'єктів, що анімуються.

Для того щоб створити простий варіант анімації на основі розкадровування руху, достатньо скористатися такою нескладною процедурою:

1. Створити рисунок довільної форми. Виділити його, перетворити в символ типу "Графіка" (F8, "Графіка")
2. Задати початкове положення символу в першому кадрі.
3. Задати кінцеве положення символу в останньому кадрі (виділити останній кадр, зробити його ключовим (F6), перемістити символ в необхідне місце). Останній кадр вибирається довільно (рис. 3).

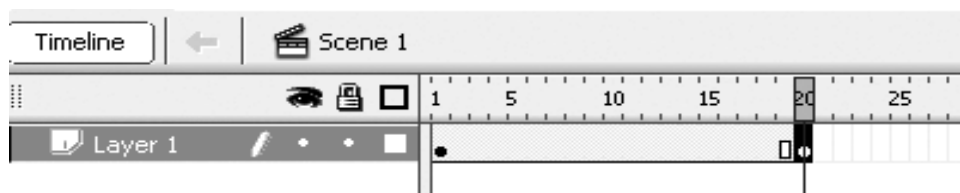


Рис. 3. Налаштування часової шкали майбутнього ролика

4. Виділити перший кадр, відкрити контекстне меню, обрати позицію "Создать анимацию движения" (рис. 4).

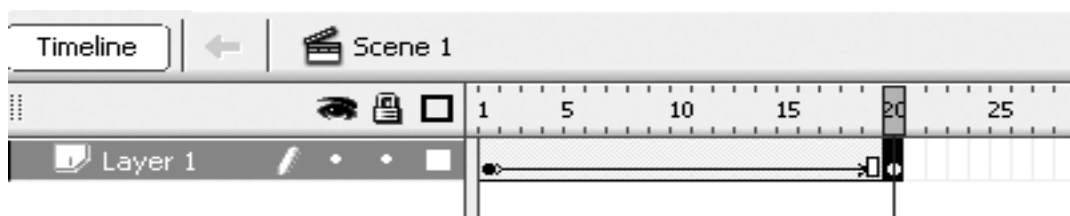


Рис. 4. Розкадровування руху

5. Запустити ролик для перегляду (Ctrl+Enter).

Завдання

1. Використовуючи графічний матеріал із папки, вказаної викладачем, створити покадрову анімацію.

2. Використовуючи малюнки, які були отримані на попередньому занятті, створити просту анімацію руху (Classic Tween та Motion tween).

3. Удосконалити отриману анімацію, додавши параметри анімації: обертання, плавність, повтори й циклічність, зміну кольору, прозорості і т. д.

4. Продумати, які елементи індивідуального мультимедійного кліпу доцільно реалізувати з використанням анімації руху, описати попередні результати в сценарії.

Складова "Створення анімації за допомогою розкадровування форми"

Компетентності, що формуються завдяки цій складовій частині:

- створення *анімації за допомогою розкадровування форми*;
- навички використання анімації розкадровування форми в анімаційному проекті спільно з іншими видами анімації.

Допоміжний матеріал

Анімація форми фігур дозволяє реалізовувати перетворення однієї форми в іншу – створювати ефект метаморфози. Для управління цими змінами можна додатково застосовувати спеціальні ідентифікатори форми (shape hints), які дозволяють створювати складні зміни і перетворення окремих частин початкової фігури в нові.

Для того щоб створити простий варіант анімації на основі розкадровування форми, достатньо скористатися наступною нескладною процедурою:

1. Створити рисунок довільної форми.
2. Задати початкове положення рисунка в першому кадрі.
3. Задати для останнього кадру властивість "ключовий кадр" (виділити кадр, F6). Останній кадр вибирається довільно.

4. В останньому кадрі видалити рисунок і на його місці зобразити інший, відмінний за формою (можна задати кінцеве положення рисунка, пересунувши його на нове місце, тоді перетворення форми відбуватиметься в русі).

5. Виділити перший кадр, у контекстному меню обрати "Создать анимацию формы". Між першим та останнім кадром має з'явитися зелена стрілочка.

6. Запустити фільм для перегляду (Ctrl+Enter).

Завдання

1. Використовуючи рисунки, які були отримані на попередньому занятті і графічний матеріал із папки, вказаної викладачем, створити просту анімацію форми.

2. Удосконалити одержану анімацію, додавши параметри анімації: переміщення, плавність, повтори, зміну кольору, прозорості і т. д.

3. Створити анімацію форми для складних об'єктів з використанням ідентифікаторів форми.

4. Продумати, які елементи індивідуального мультимедійного кліпу доцільно реалізувати з використанням анімації руху.

5. Додати в індивідуальний анімаційний кліп елементи, засновані на анімації форми.

6. Створити сцени в анімаційному кліпі елементи, що використовують покадрову анімацію, анімацію руху і анімацію форми.

7. Додати в сценарій отримані результати.

Складова "Шар, що спрямовує рух, і шар маски"

Компетентності, що формуються завдяки цій складовій частині:

- створення анімації руху за заздалегідь обраним шляхом складної форми;
- створення анімації на основі шару-маски.

Допоміжний матеріал

Створення анімації руху спрямувальним

Шар (Layer) – це невелика частина сцени фільму, здатна містити будь-яку кількість графічних об'єктів.

Шар, що спрямовує рух (Guide layer) використовується для створення траєкторій, за якою рухатимуться об'єкти, що анімуються.

Для створення анімації, що відображає рух об'єкта за задалегідь обраним шляхом (траєкторією) складної форми, скористаємося такою послідовністю дій.

1. У новому документі до існуючого шару Layer 1 необхідно додати шар, що спрямовує рух "Вставка"/"Временная шкала"/"Слой", потім необхідно увійти до контекстного меню створеного шару та обрати "Направляющая" (Motion Guide) (рис. 5).

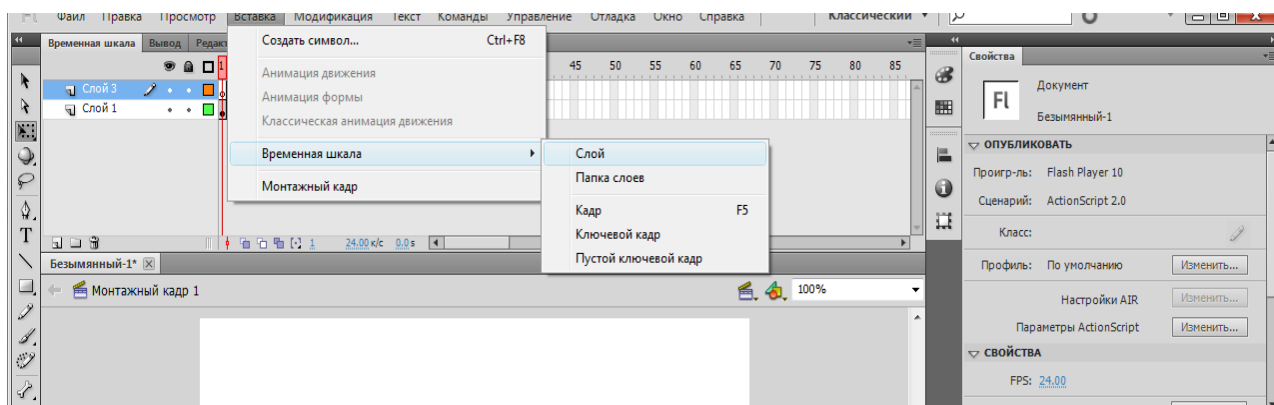


Рис. 5. Створення нового шару

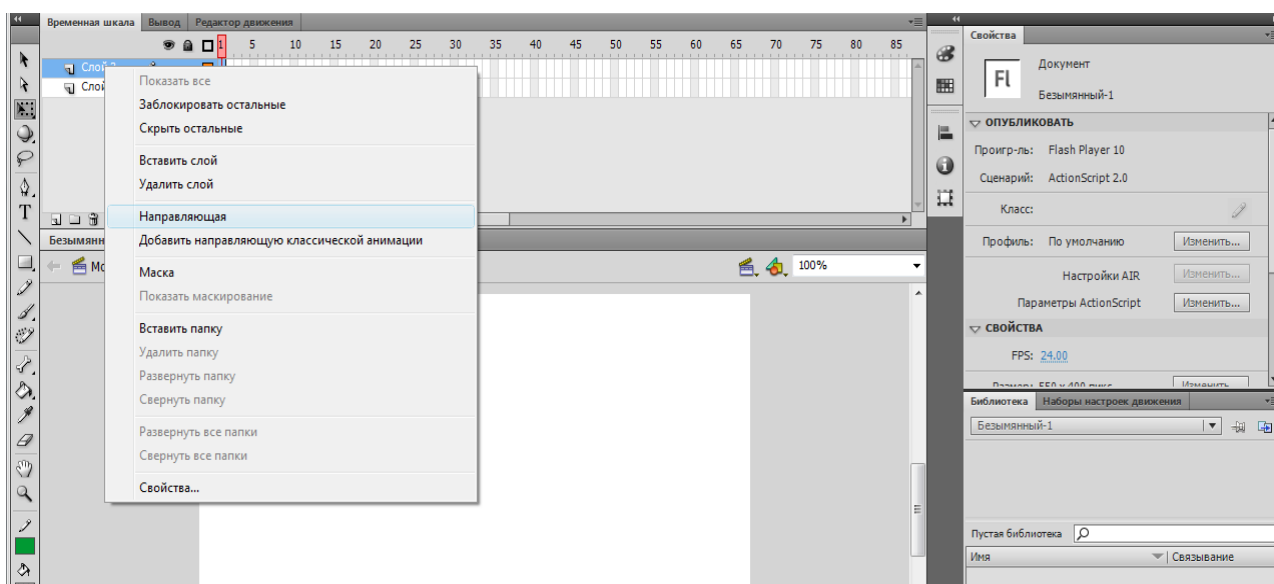


Рис. 6. Шар, що направляє

2. На шарі, що спрямовує, необхідно зобразити шлях (будь-яким інструментом для малювання ліній, головне, щоб лінія шляху не перетиналася сама з собою) (рис. 10).

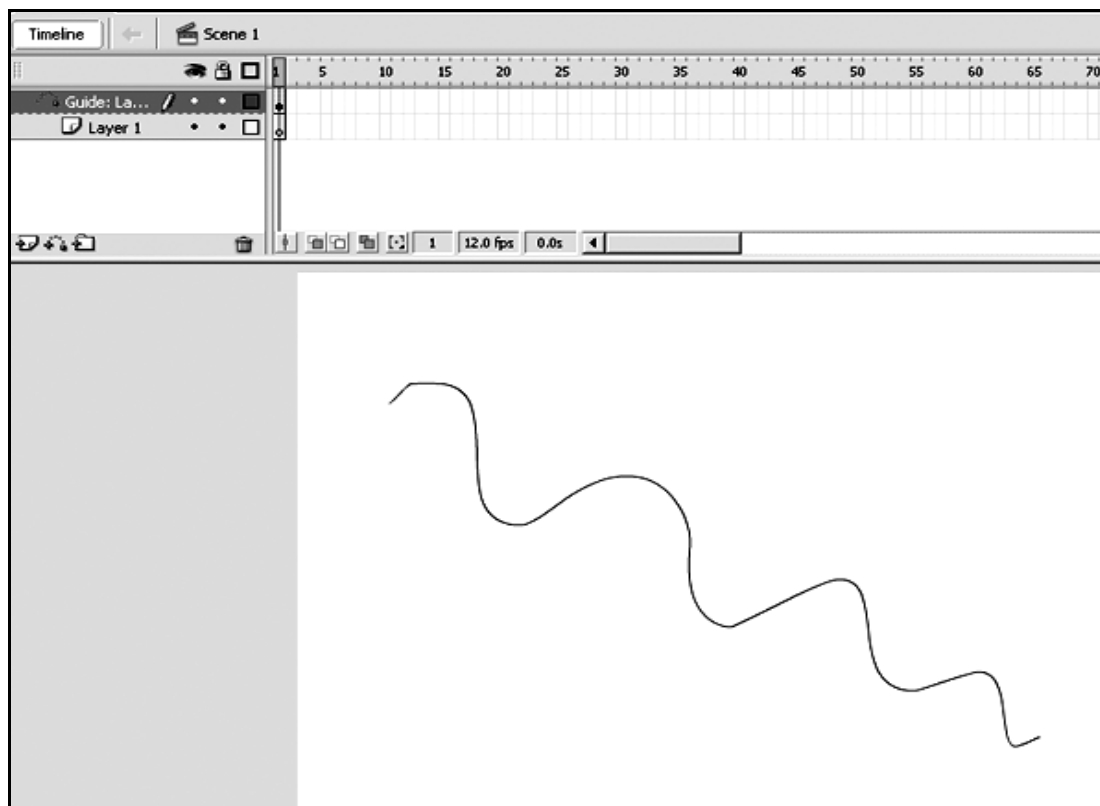


Рис. 7. Створення шляху на шарі, що спрямовує

3. На шарі Layer1 в першому кадрі необхідно створити рисунок.

4. Виділивши рисунок, необхідно перетворити його в символ довільного типу (F8).

5. Для символу задається початкове положення так, щоб його центр співпадав з початком шляху.

6. Останній кадр шару Layer1 призначається ключовим (виділяється кадр, F6). Останній кадр вибирається довільно.

7. Лінія життя шару, що спрямовує продовжується так, щоб вона співпадала з лінією життя робочого шару (виділяємо останній кадр, F5).

8. Кінцеве положення символу задається так, щоб його центр співпадав з кінцем шляху.

9. Виділяється перший кадр робочого шару, в його контекстному меню обирається тип "розкадровування руху".

10. Фільм запускається на виконання (Ctrl+Enter).

Створення анімації маски

Маска (Mask layer) є звичайним шаром, за винятком того, що будь-яке заливання на ньому інтерпретується Flash як отвір, через який видно попередній шар. При цьому колір заливання (зокрема растрової), наявність градієнта, контур заливки і її тип повністю ігноруються Flash.

Для створення анімації, яка ілюструє використання шару-маски, скористаємося наступною послідовністю дій.

1. У новому документі до існуючого шару Слой1 додається шар Слой2 ("Вставка"/"Временная шкала"/"Слой") і перетворюється на той, що маскує (права кнопка миші, Маска) (рис. 8 – 9).

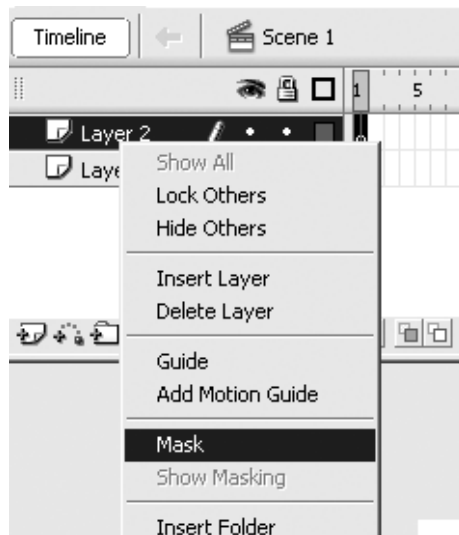


Рис. 8. Створення шару-маски

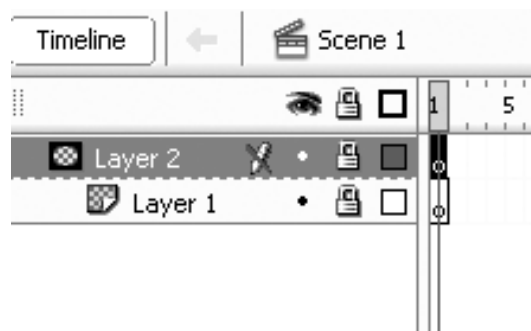


Рис. 9. Шар маски

2. На маскованому шарі (Layer1) створюється напис (для наочності краще маскованим об'єктом вибрати напис).

3. На маскуючому шарі необхідно створити маску (будь-який рисунок або символ довільної форми) (рис. 10).

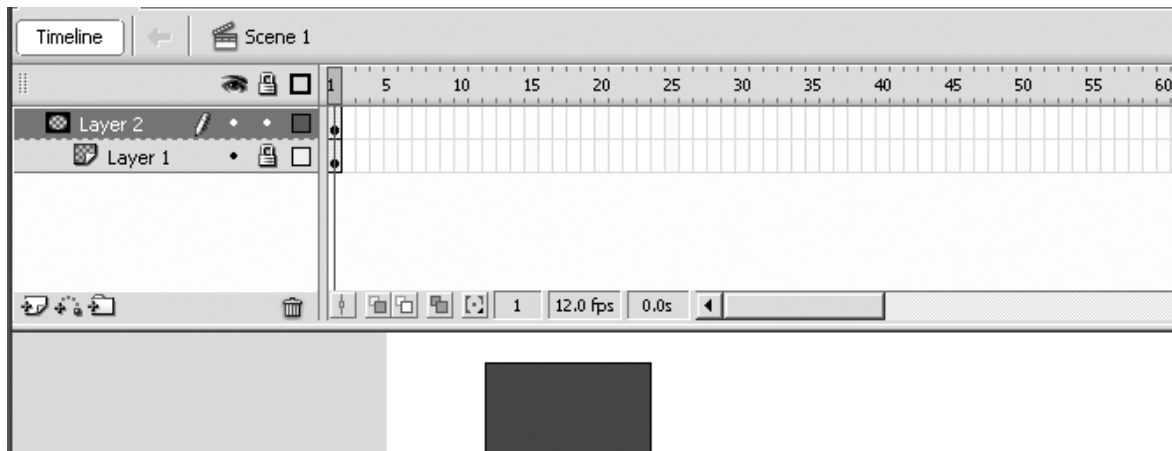


Рис. 10. Маскувальний шар

4. Початкове положення маски в першому кадрі задається так, щоб вона співпадала із початком напису.

5. Лінія життя маскованого і маскувального шарів продовжується (необхідно виділити останній кадр на обох шарах, F5).

6. В останньому кадрі необхідно задати кінцеве положення маски (виділяється останній кадр, він призначається ключовим (F6), маска розтягується до кінця напису). Останній кадр вибирається довільно.

7. Необхідно виділити перший кадр маскувального шару і створити розкадровування для маски.

8. Фільм запускається на виконання (Ctrl+Enter).

Завдання

1. У створеному раніше анімованому кліпі необхідно вибрати об'єкт, пересування якого доцільно організувати за заздалегідь визначеною траєкторією складної форми. Можна також ввести додатковий об'єкт.

2. Для певних об'єктів організувати в індивідуальному кліпі рух за заданим шляхом.

3. Продумати доповнення до створеного кліпу із невеликих фонових об'єктів, які також повинні рухатися за заданим шляхом (це можуть бути сніжинки, листя, метелики, зірки і т. д).

4. В індивідуальний кліп додати анімацію маски.
5. Отримані результати описати в сценарії індивідуального кліпу.

Складова "Символи-контейнери"

Компетентності, що формуються завдяки цій складовій частині:

- створення анімації на основі використання символів-контейнерів;
- створення більш ефектної анімації, що дозволяє створювати ефект внутрішнього руху або інших змін у складі об'єктів, що рухаються, з використанням символів "Фрагмент ролика".

Допоміжний матеріал

Створення символів-контейнерів

Символи-контейнери дозволяють створювати більш реалістичну та насичену анімацію.

Для створення анімації, що відображає рух складного об'єкта, у складі якого є інші об'єкти, які під час руху також можуть синхронно змінювати свій колір, форму або рухатися, скористаємося наступною послідовністю дій.

1. У новому документі необхідно створити три шари.
2. У першому шарі створити рисунок кузова машини (рис. 11).



Рис. 11. Вміст першого шару документа

3. На другому шарі зобразити колесо і зробити з нього символ типу "Фрагмент ролика" (рис. 12).

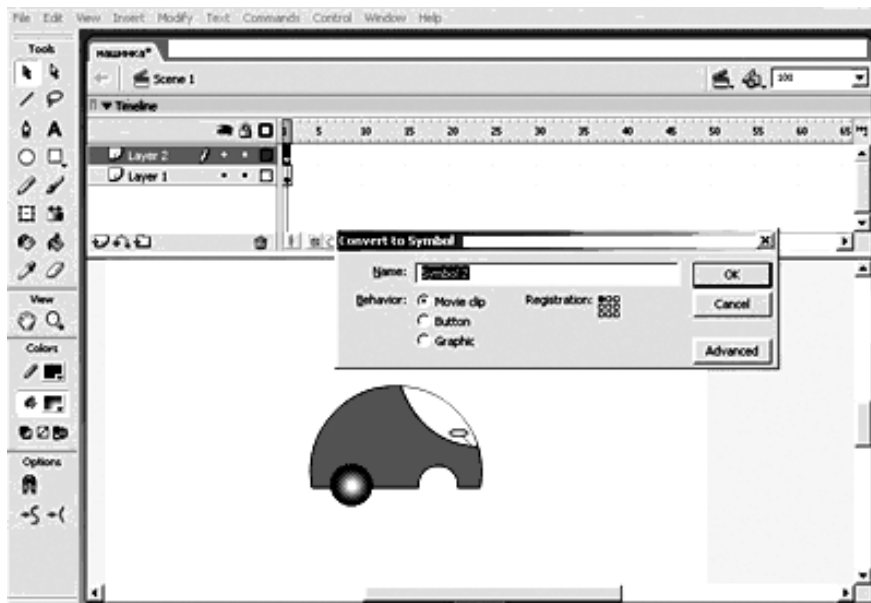


Рис. 12. Створення символу типу MovieClip для оформлення коліс

4. Увійти до символу (подвійним клацанням миші по ньому).

5. Виконати анімацію рисунка так, щоб він змінював свій колір в кожному ключовому кадрі (можна взяти 5 ключових кадрів). Для анімації використовувати розкадровування форми.

6. Повернутися на основну сцену (Scene 1) і скопіювати колесо на третій шар (можна використовувати бібліотеку символів Ctrl+L).

7. Виділити машинку і задати її початкове положення в першому кадрі.

8. Продовжити лінію життя фільму на 20 кадрів. Для цього необхідно виділити 20-й кадр на всіх трьох шарах і натиснути F5.

9. Зробити 20-й кадр кожного шару ключовим (F6).

10. Виділити машинку і задати її кінцеве положення на робочому столі.

11. Створити розкадровування для кожного шару. Для шарів коліс вона буде анімацією руху, для шару з кузовом – анімацією форми (рис. 13).

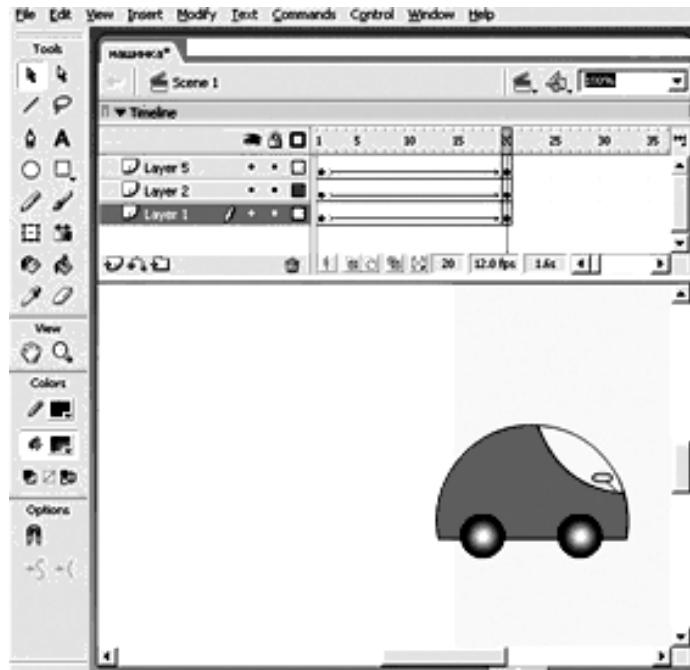


Рис. 13. Розкадровування елементів основної сцени

12. Запустити фільм (Ctrl+Enter).

13. Можна оформити роботу, створивши шар з фоном (для цього фоновий шар потрібно перемістити вниз відносно інших шарів) (рис. 14).

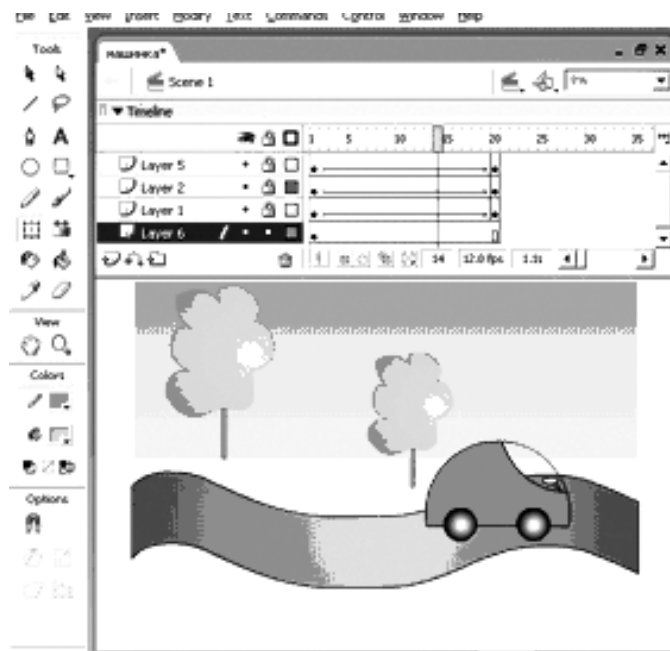


Рис. 14. Створення фонового шару

Завдання

1. У створеному раніше анімованому кліпі необхідно вибрати об'єкт або групу об'єктів, роботу яких доцільно організувати у вигляді символу-контейнера. Можна також ввести додаткові об'єкти.

2. Для певних об'єктів зробити сценарій виконання символу-контейнера.

3. В індивідуальний кліп додати символ-контейнер.

4. Отримані результати описати в сценарії індивідуального кліпу.

Складова "Створення анімованої кнопки (банера)"

Компетентності, що формуються завдяки цій складовій частині:

- створення символів "Кнопка" та "Фрагмент ролика";
- додавання у символ "Кнопка" символу "Фрагмент ролика" для створення анімованої кнопки;
- призначення подій ("Действия – кнопка"), на які повинна реагувати кнопка.

Допоміжний матеріал

Символ-кнопка (button symbol) призначений для додавання у фільм інтерактивних кнопок, які реагують на дії користувача і керують відтворенням фільму. З цією метою символу-кнопці можуть бути присвоєні події ("Действия – кнопка"), на які вона повинна реагувати, і дії (Actions), які ініціалізуються цими подіями. Крім того, кнопка може бути анімованою і з нею може бути пов'язаний звуковий супровід. Кнопка може реагувати на наведення, натиснення, перетягання, натиснення клавішею клавіатури.

Для створення кнопки, що містить анімацію, скористаємося такою процедурою:

1. Створити рисунок (будь-яку геометричну фігуру, яка буде кнопкою).

2. Виділити і зробити з рисунка кнопку (команда "Преобразовать в символ" типу "Кнопка") (рис. 15). Створеній кнопці необхідно присвоїти ім'я (краще латиницею).

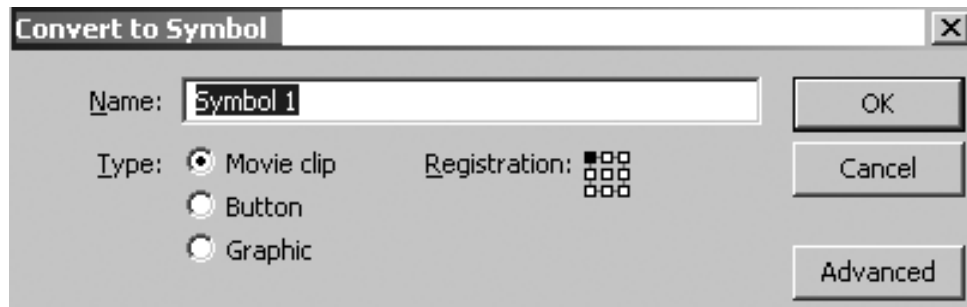


Рис. 15. Створення кнопки

3. Далі необхідно зайти всередину кнопки (подвійне натиснення) для продовження її редагування.

4. Зробити ключовим кадр Over (виділити кадр, F6) (рис. 16).

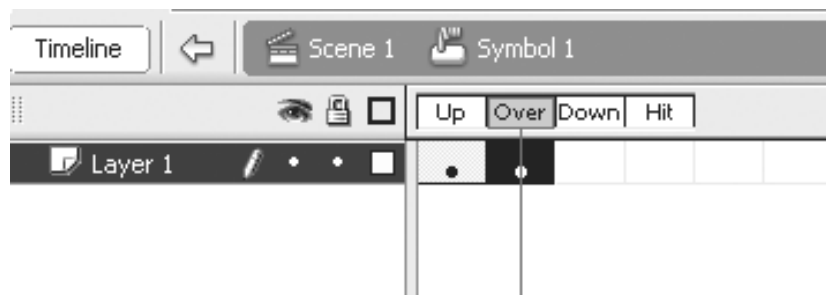


Рис. 16. Часова шкала кнопки

5. У цьому кадрі необхідно з рисунка зробити символ типу "Фрагмент ролика" (Movie Clip, F8).

6. Зайти всередину символу (подвійне клацання на ньому).

7. У середині нього виконати анімацію, яка відбуватиметься в разі наведення на кнопку.

8. Запустити фільм на перегляд (Ctrl+Enter).

Завдання

1. Проаналізувати створений раніше кліп і додати до нього символи-кнопки а також вибрати існуючі об'єкти, які доцільно перетворити на кнопки.

2. Визначити, на які події повинні реагувати кнопки і які дії повинні при цьому виконуватися.

3. Створити анімацію для отриманих кнопок.

4. Отримані результати описати в сценарії індивідуального кліпу.

Складова "Імпортування і оптимізація об'єктів. Робота з відео і звуком"

Компетентності, що формуються завдяки цій складовій частині:

- вміння імпортувати об'єкти графіки, відео і звуку в кліпи;
- виявлення переваг і недоліків оптимізації графічної інформації для використання у Flash.

Допоміжний матеріал

Використання у складі анімованого кліпу растрової графіки, звуку та відео дозволяє отримати цілісні, реалістичні проекти професійного рівня. Такі кліпи дозволяють значно розширити сфери використання Flash-продуктів, дозволяючи використовувати отримані результати на телебаченні, у різних галузях рекламного інформаційного середовища. Для створення анімації, що використовує звуковий супровід, скористаємося наступною послідовністю дій.

Імпортування звуку

Для імпортування звуку у Flash-фільм необхідно виконати наступні дії:

1. Створити новий шар для звуку.
2. Імпортувати звуковий файл в бібліотеку символів (Файл/Імпорт /Імпортувати в бібліотеку).
3. На панелі "Свойства" зі списку "Звук" вибрати імпортований звук.
4. Для коректного відтворення потокового звуку необхідно вибрати на панелі "Свойства" в списку "Синхронізація" пункт "Поток".
5. За необхідності відредагувати звуковий файл, використовуючи вбудований звуковий редактор (панель Свойства/кнопка "Редактировать").

Для вставки **подієвого** звуку в кнопку необхідно виконати наступні дії:

1. Імпортувати звуковий файл.
2. Створити кнопку (F8, "Создание символа", тип "Кнопка").
3. Увійти до режиму редагування кнопки і створити новий шар для звуку.

4. Зробити кадр, у разі спрацьовування якого програватиметься звук, ключовим (на створеному для звуку шарі).
5. Для вибраного кадру вибрати звук з бібліотеки.
6. Для коректного відтворення подієвого звуку на панелі "Свойства" необхідно переконатися, що в списку "Синхронизация" **не вибрано** пункт "Поток".

Імпортування відео

Для імпортування відео в Flash-фільм необхідно виконати наступні дії:

1. Створити новий шар для відео.
2. Імпортувати відео файл (Файл/Імпорт/Імпортировать видео-файл). Обрати файл з вашої папки з розширенням *.flv.
3. У вікні майстра імпорту відео вибрати потрібний параметр: для точнішої синхронізації звуку та інтерактивних елементів Flash-фільму вибрати "Внедрить FLV файл в SWF файл и воспроизвести во временной шкале".

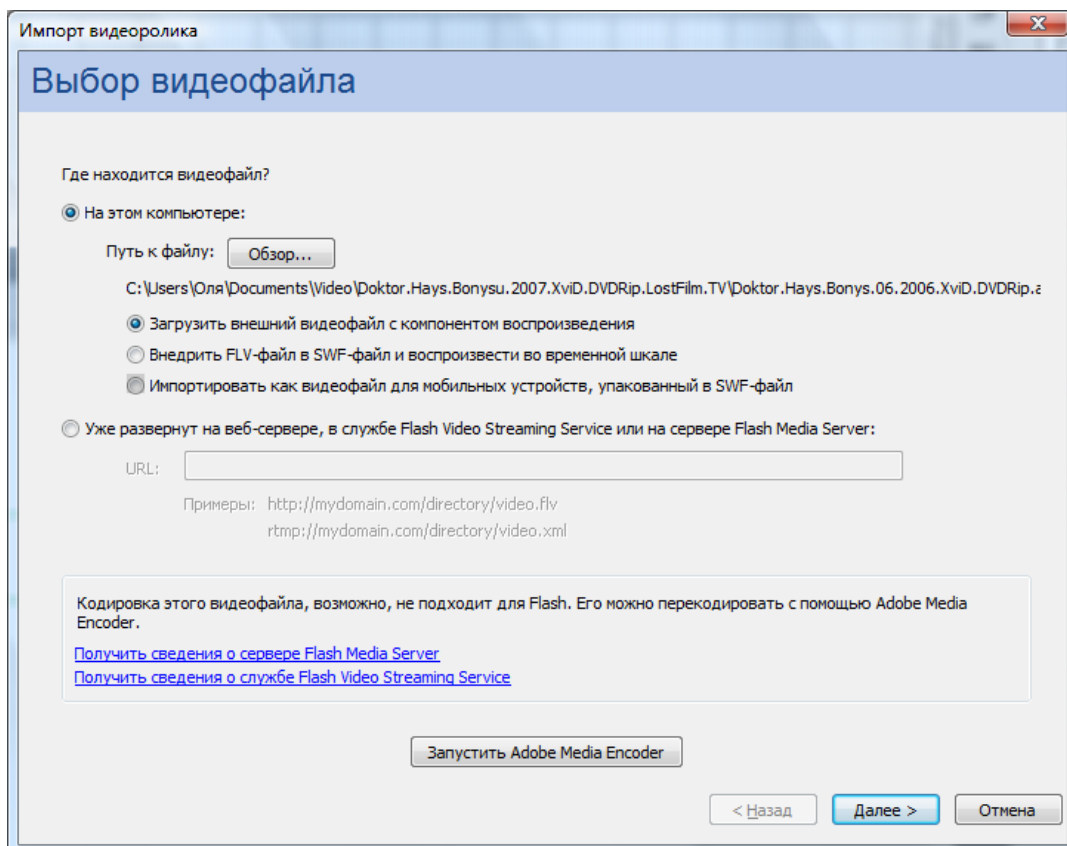


Рис. 17. Майстер імпорту відеоролика

4. За необхідності після імпортування відео його можна додатково налаштувати.

Завдання

1. Імпортуйте в Flash растрові графічні файли наступних форматів: *.bmp, *.jpg, *.gif. У звіті опишіть переваги і недоліки імпорту кожного формату.

2. Використовуючи команду Modify/Bitmap/Trace Bitmap виконаєте трасування двох растрових файлів: із геометричним і реалістичним малюнками. Результат порівняйте у звіті.

3. У створений раніше анімований кліп додайте потоковий і подієвий звуки, а також кнопку, яка вимикає всі потокові звуки при натисненні.

4. Скриптова конструкція для кнопки вводиться в редакторові Actions при виділеній кнопці:

```
on (press) {  
stopAllSounds();  
}
```

5. У звіті порівняйте імпортування і застосування потокового та подієвого типів звуку.

6. У створений раніше анімований кліп додайте відеоролик, застосуйте до нього анімацію і візуальні ефекти (скалювання, обертання, зміну колірної відтінку, прозорості, яскравості, накладення анімованого тексту і об'єктів).

7. У звіті порівняйте імпорт відео для використання в наступних випадках: "Progressive download from a web server" і "Embed video in SWF and play in timeline".

Складова "Динамічні текстові поля. Активна кнопка"

Компетентності, що формуються завдяки цій складовій частині:

- вміння створювати динамічні текстові поля у інтерактивних веб-додатках;
- створення кнопок, що реагують на певні події.

Допоміжний матеріал

Створення динамічних текстових полів

Динамічні поля призначені для відображення тексту, що змінюється під час програвання ролика. У процесі створення динамічного тексту можна розмістити текст в одному рядку або створити текстовий блок з фіксованою шириною і висотою. Flash дозволяє використовувати в динамічних текстових полях елементи оформлення HTML.

Для створення динамічного текстового поля необхідно виконати наступні дії:

1. У новому документі створити текстове поле типу Dynamic Text (рис. 18).

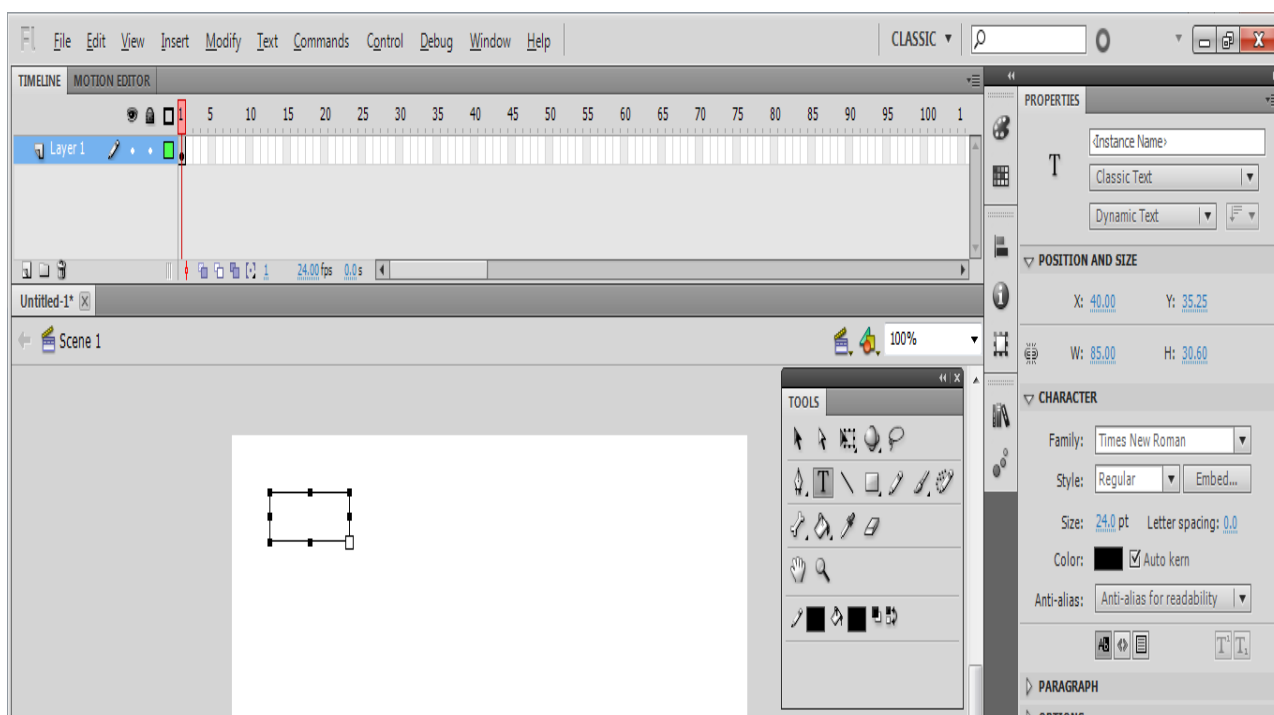


Рис. 18. Створення динамічного текстового поля

2. Задати властивості для текстового поля (рис. 19):

- Line type: Multiline.
- Render text as HTML.
- Var: text (ім'я змінної довільне).

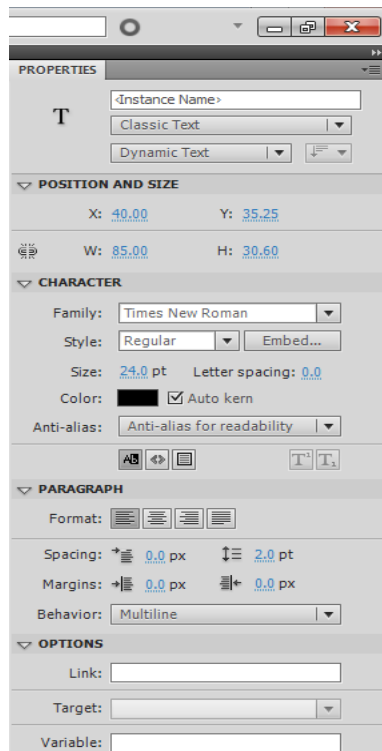


Рис. 19. Властивості динамічного текстового поля

3. Виділити перший кадр (ключовий) і написати в редакторі Actions наступний код:

```
loadVariablesNum("file.txt",0);
```

4. Зберегти проект.

5. У тому ж каталозі, в якому знаходиться і проект, створити текстовий файл "file.txt". Вибрати для файла кодування Юнікод.

6. У текстовому файлі присвоїти змінній text який-небудь текст
text=любой текст

7. Зберегти текстовий файл.

8. Запустити фільм (Ctrl+Enter).

Активна кнопка (кнопка, що з'являється)

Кнопкам у Flash властива велика кількість подій, на які можна реагувати:

`press` – клавіша миші відтиснута, коли курсор знаходиться в межах кнопки;

`release` – клавіша миші відтиснута, коли курсор знаходиться в межах кнопки;

`releaseOutside` – клавіша миші відтиснута, коли курсор знаходиться поза межами кнопки;

`rollOver` – курсор миші входить в межі кнопки;

`rollOut` – курсор виходить за межі кнопки;

`dragOver` – курсор входить в межі кнопки, при цьому була натиснута кнопка і натиснута клавіша миші;

`dragOut` – курсор виходить за межі кнопки, при цьому була натиснута кнопка і натиснута клавіша миші;

`keyPress` ("клавіша") – була натиснута "клавіша". Можна подивитися список клавіш у довідці із Flash (об'єкт `Key`) або використовувати панель параметрів для введення потрібної клавіші.

Перехоплюються ці події за допомогою директиви `on()`. Синтаксис її такий:

```
on (подія)
{
... // Дії користувача
}
```

Для того щоб забезпечити реакцію кнопки на подію наведення, необхідно виконати наступні дії:

1. Вибрати з бібліотеки `Common Libraries` будь-яку кнопку (`Window /Common Libraries/Buttons`).

2. У редакторі `Actions` для кнопки написати код:

```
on (rollOver)
{
this._x=random(200);
this._y=random(200);
}
```

Завдання

1. Створити та налаштувати динамічне текстове поле. Присвоїти йому змінну для зв'язання з текстовим файлом.

2. Створити текстовий файл з інформацією, яку необхідно завантажити до динамічного текстового поля. Здійснити форматування цього файла.

3. Додати до індивідуального мультимедійного проекту динамічне текстове поле.

Складова "Програмування на ActionScript"

Компетентності, що формуються завдяки цій складовій частині:

- вміння створювати анімацію символів за допомогою мови програмування ActionScript;
- програмна зміна властивостей символів-контейнерів.

Допоміжний матеріал

Буде розглянуто приклад створення кліпу, основою динаміки якого далі є використання мови програмування ActionScript, а саме скрипти, що дозволяють дублювати, видаляти символи, та змінювати їх властивості.

Для дублювання символів-контейнерів використовуються такі скриптові компоненти.

duplicateMovieClip(target, newname, depth)

Параметри:

- *target* – шлях до дубльованого символу;
- *newname* – унікальний ідентифікатор для нового екземпляра символу;
- *depth* – унікальне значення глибини для дубльованого символу. Глибина – порядок розташування дубльованих мувікліпів. Він схожий на порядок шарів кінострічки – розташовані нижче символи перекриваються такими, що знаходяться вище і мають більше значення глибини. Необхідно присвоїти кожному новому символу унікальну глибину, якщо привласнити ту, що вже є, то новий символ замінить старий.

Опис: Команда створює екземпляр мувікліпу під час відтворення основного кліпу. Дублювання завжди починається з кадру номер 1, незалежно від того, на якому кадрі знаходився оригінальний (батьківський) символ. Змінні, що містяться в батьківському символі, в дубльований не копіюються. Якщо видаляється батьківський символ, то дубльований теж видаляється. Можна використовувати команду або метод `removeMovieClip`, щоб видалити екземпляр мувікліпу, створеного командою `duplicateMovieClip`.

Для створення анімації символів-контейнерів шляхом дублювання необхідно виконати таку послідовність дій:

1. Створити символ MovieClip у вигляді геометричної фігури довільного кольору і розміру, присвоїти йому імя box (Instance name "box").

2. Зайти в Movie Clip і зробити розкадровування руху для об'єкта (в результаті об'єкт повинен переміщуватися з однієї точки екрана в іншу по прямій).

3. Вийти на основну сцену.

4. Створити окремий шар для кодів. Для першого кадру цього шару в редакторі Actions ввести код:

```
count = 1;
while (count<20)
{
    _root.box.duplicateMovieClip("box"+count, count);
    _root["box"+count]._x = random(550);
    _root["box"+count]._y = random(150);
    _root["box"+count]._xscale = random(150);
    _root["box"+count]._yscale = random(150);
    _root["box"+count]._alpha = random(100);
    count += 1;
}
```

5. Продовжити лінію життя всього фільму на 15 кадрів.

6. На 15-му кадрі шару в редакторі коду Actions ввести код (попередньо цей кадр необхідно зробити ключовим):

```
gotoAndPlay(1);
```

Видалення символів-контейнерів відбувається за допомогою команди:

removeMovieClip(target)

Параметри:

- *target* – шлях (target path) до екземпляра мувікліпу, створеного за допомогою duplicateMovieClip або імені мувікліпу, створеного методами attachMovie або duplicateMovie, об'єкта MovieClip.

Надання нових властивостей символу-контейнера здійснюється за допомогою наступної команди:

setProperty("target", property, value/expression)

Параметри:

- *target* – шлях до кліпу, властивості якого потрібно обрати;
- *property* – властивість, яку потрібно встановити;
- *value* – нове значення властивості;
- *expression* – вираз, обчислення якого приводить до нового значення властивості.

Для того щоб встановити нові властивості для символу-контейнера і видалити символ контейнер після закінчення його роботи, необхідно виконати наступні дії:

1. Створити символ Movie Clip у вигляді пунктирної лінії довільного кольору, присвоїти йому ім'я line1 (Instance name "line1"). Лінія повинна бути зображена вертикально; нижній край повинен знаходитися точно в центрі робочого столу (рис. 20).

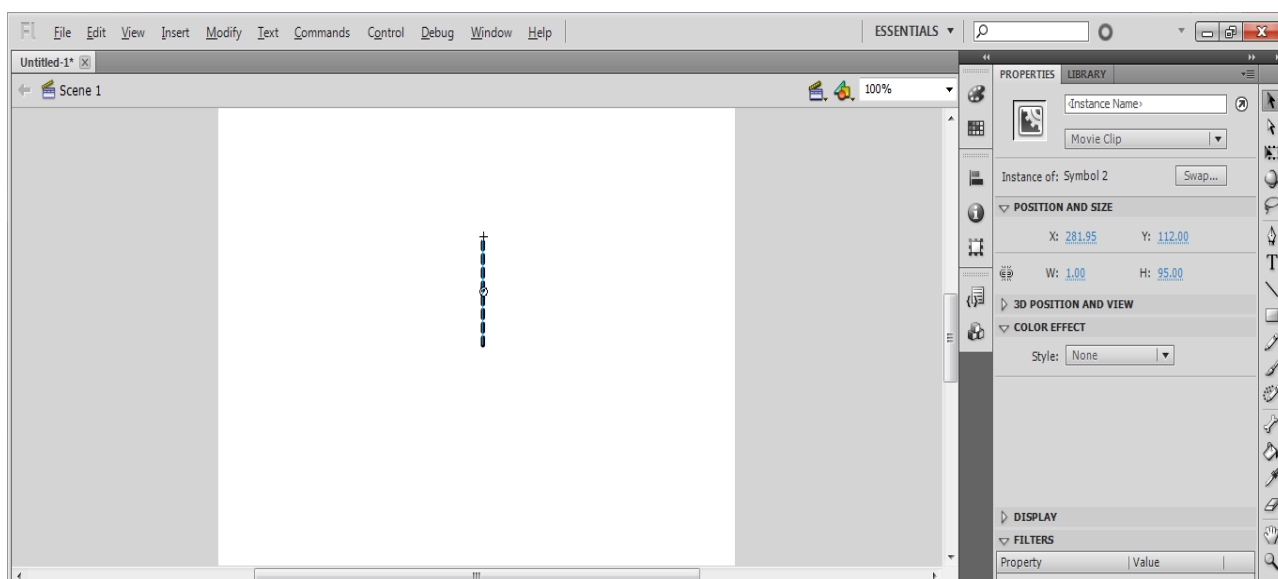


Рис. 20. Початкове положення символу

2. Зайти в Movie Clip і зробити розкадровування руху для об'єкта (в результаті кінцеве положення об'єкта повинне бути, як на рис. 21).

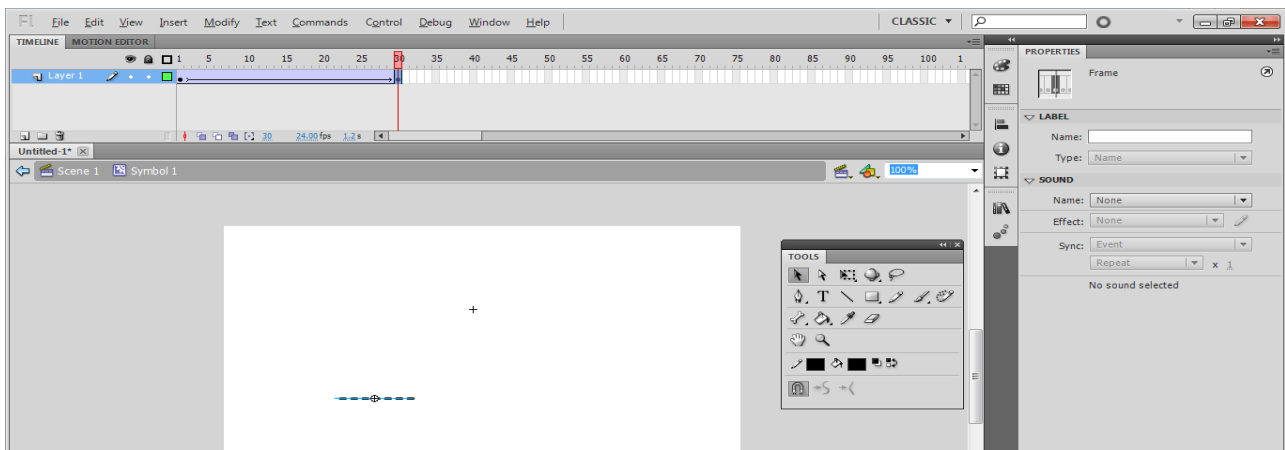


Рис. 21. Розкадровування руху всередині Movie Clip

3. Після розкадровування створити ключовий кадр і для нього в редакторі Actions ввести код:

```
removeMovieClip(_target);
```

4. Вийти на основну сцену.

5. Створити окремий шар для кодів. Для першого кадру цього шару в редакторі Actions ввести код:

```
i = 2;
```

```
setProperty("/line1", _visible, False);
```

6. Для другого кадру шару коду в редакторі Actions ввести код:

```
duplicateMovieClip("/line1", "line"+i, i);
```

```
setProperty("line"+i, _rotation, i*10);
```

```
i = Number(i)+1;
```

7. Для третього кадру шару коду в редакторі Actions ввести код:

```
gotoAndPlay(2);
```

Для того щоб відстежити зміни, що відбуваються з символом-контейнером, необхідно скористатися наступними командами.

```
1. onClipEvent(movieEvent){  
statement(s);  
}
```

Параметри:

- *movieEvent*, є тригером (перемикачем), що називається "подія" (event). Коли подія відбувається, виконується код, що міститься всередині фігурних дужок. Будь-яке з наступних значень може бути визначене для аргументу *movieEvent*:

- *load* – ініціалізація команд у момент появи мувікліпу на часовій лінійці (Timeline);

- *unload* – ініціалізація команд в першому кадрі в момент видалення мувікліпу з лінійки. Команди, пов'язані з подією Unload, обробляються раніше від будь-яких команд, прописаних у кадрі;

- *enterFrame* – ініціалізація команд в кожному кадрі, що відтворюється, так, ніби вони були прописані в кожному кадрі мувікліпу. Команди, пов'язані з подією OnEnterFrame, обробляються після будь-яких команд, прописаних у кадрах;

- *mouseMove* – ініціалізація команд при кожному переміщенні миші. Для визначення поточної позиції миші використовуються властивості `_xmouse` і `_ymouse`;

- *mouseDown* – ініціалізація команд при натисканні лівої кнопки миші;

- *mouseUp* – ініціалізація команд при відпусканні лівої кнопки миші;

- *keyDown* – ініціалізація команд при натисканні клавіші клавіатури.

Метод `Key.getCode` використовується для отримання інформації про натиснуту клавішу;

- *keyUp* – ініціалізація команд при відпусканні клавіші клавіатури;

- *data* – ініціалізація команд при отриманні даних у командах `loadVariables` або `loadMovie`. Коли це пов'язано з командою `loadVariables`, подія `data` відбувається тільки один раз у момент закінчення завантаження останньою змінною. У разі використанні команди `loadMovie` подія `data` може бути багаторазовою у міру надходження нових даних;

- *statement(s)* – команди, що виконуються в момент настання події `mouseEvent`;

2. startDrag(target[lock,left, top, right, bottom])

Параметри:

- *target* – шлях до кліпу, який треба перетягувати;

- *lock* – булеве значення, вказує до чого буде приєднаний кліп в разі перетягання, до центру кліпу щодо позиції миші (`true`), або до точки де користувач натиснув кнопку на кліпі (`false`). Цей параметр необов'язковий;

• *left, top, right, bottom* – значення щодо координат батьківського символ-ролика, які визначають прямокутник обмеження для символ-ролика. Ці параметри необов'язкові.

Опис: команда; примушує мувікліп *target* переміщуватися за покажчиком миші під час відтворення фільму. Тільки один кліп може переміщуватися за одиницю часу. Відразу після виконання операції *startDrag* мувікліп починає перетягуватися до тих пір, поки його не зупинять командою *stopDrag*, або поки команда *startDrag* буде застосована до іншого кліпу.

3. *stopDrag()*

Опис: Команда, що зупиняє поточну операцію перетягання.

Для того щоб відстежити координати переміщення символу-контейнера, необхідно виконати наступні дії:

1. Створити два текстові поля типу *Dynamic Text*, одному присвоїти *var=_root.pos1*, іншому – *var=_root.pos2*. Перетворити кожне текстове поле в *Movie Clip*, одному присвоїти *Instance name "line1"*, іншому – *"line2"*.

2. Створити на тому ж шарі *Movie Clip*, присвоїти йому *Instance name "mouse"*.

3. Для мувікліпу в редакторі *Actions* ввести код:

```
onClipEvent (mouseDown) {
    startDrag(this);
}
onClipEvent (mouseUp) {
    stopDrag();
}
onClipEvent (enterFrame) {
    Xmouse = getProperty("_root.mouse", _x);
    Ymouse = getProperty("_root.mouse", _y);
    setProperty("_root.line1", _y, Ymouse);
    setProperty("_root.line2", _x, Xmouse);
}
onClipEvent (mousemove) {
    _root.pos2 = "y =" add Math.round(Ymouse);
    _root.pos1 = "x =" add Math.round(Xmouse);
}
```

Завдання

1. Створити копії екземпляра символу, кожна копія повинна мати свої координати, прозорість і розмір.
2. Змінити форму, розмір і кількість об'єктів, що відображаються, довільним чином.
3. Створити віртуальну спіраль з пунктирної лінії.
4. Замінити пунктирну лінію на геометричну фігуру довільної форми, видалити заливання.
5. Пересуваючи об'єкт робочою областю, визначити його координати.

Складова "Робота з компонентами Adobe Flash"

Компетентності, що формуються завдяки цій складовій частині:

- вміння та навички створення та роботи з компонентами Flash;
- вміння обирати оптимальний склад компонентів для динамічної форми.

Допоміжний матеріал

Розглянемо приклад створення в Adobe Flash динамічної форми, яка дозволить відображати розгорнуті коментарі щодо обраної опції. Інформація для коментарів буде зберігатися в окремому текстовому файлі.

Для створення динамічної форми, яка містила б компоненти, необхідно виконати наступні дії:

1. На робочий стіл помістити наступні компоненти (рис. 22):

- *TextArea*;
- *ComboBox*;
- *Button*;
- *Label*.

2. Задати властивості для компонентів:

TextArea: Instance Name – txt; editable – false.

ComboBox: Instance Name – cb.

Button: Instance Name – btn; label – ОК.

Label: autoSize – left; text – Продукти компанії Adobe.

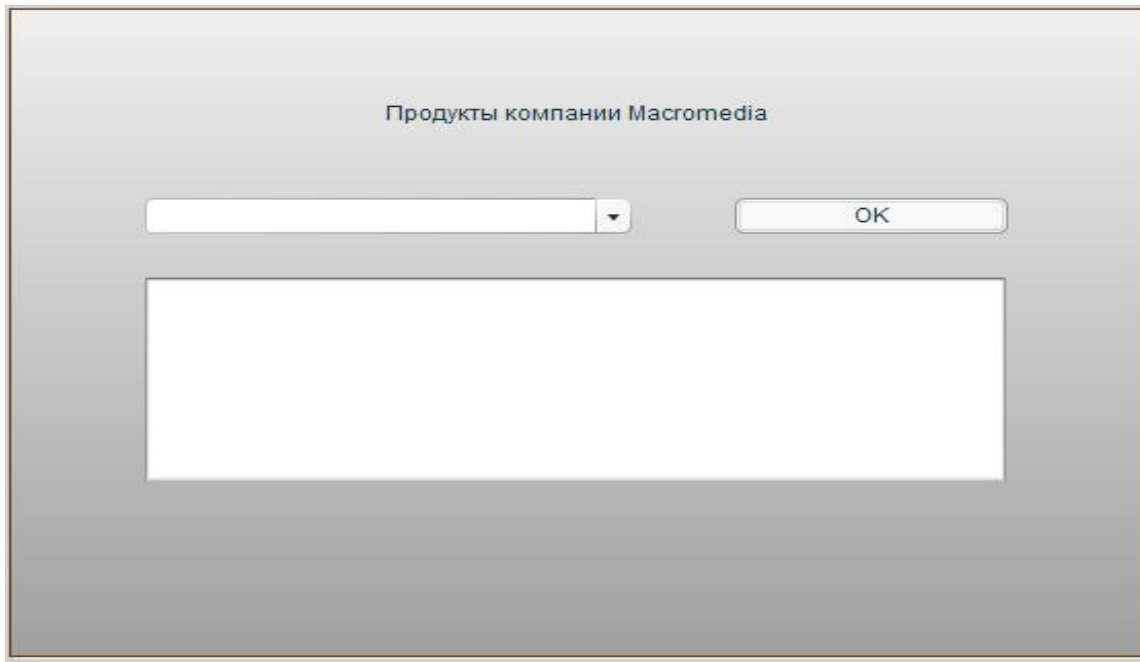


Рис. 22. Динамічна форма з розміщеними на ній компонентами

3. Створити новий шар, для першого кадру якого ввести в редактор Actions наступний код:

```
//Завантаження файлу з текстом  
loadVariablesNum("file.txt",0);
```

```
//Заповнення списку ComboBox значеннями  
cb.dataProvider = ["Flash", "Dreamweaver"];
```

```
//Обробка натиснення на кнопку; викликається функція doSubmit  
btn.addEventListener("click", doSubmit);
```

```
//Функція відображення тексту, згідно з вибраним пунктом списку  
function doSubmit():Void {  
    switch (_root.cb.selectedItem) {  
        case "Flash" :  
            _root.txt.text=fl;  
            break;  
        case "Dreamweaver" :  
            _root.txt.text=dr;  
            break;  
    }  
}
```

4. Зберегти проект.

5. У папці зі збереженим проектом створити файл "file.txt", що містив би змінні, яким був би присвоєний довільний текст, наприклад:

`fl=Welcome to Adobe Flash. Flash provides everything you need to create and deliver rich web content and powerful applications.`

`&dr=Adobe Dreamweaver is a professional HTML editor for designing, coding, and developing web sites, web pages, and web applications.`

Склад компонентів динамічної форми можна змінити, залежно від необхідності. Для цього потрібно виконати такі дії:

1. З попередньої форми видалити компонент `ComboBox`, додати два компоненти `RadioButton` – компонент `DateField` і компонент `Loader` (його потрібно помістити на окремий шар, який повинен бути найнижчим, і розтягнути згідно з розмірами робочого столу) (рис. 23).

2. Зберегти проект і помістити в ту ж папку зображення з ім'ям `pict.jpg` (зображення вибрати довільно).

3. Задати властивості для нових компонентів:

`RadioButton1`: `data – 1`; `label – Flash`; `selected – true`.

`RadioButton2`: `data – 2`; `label – Dreamweaver`.

`DateField`: `_dayNames` – змінити на російськомовні назви днів тижня; `MonthNames` – змінити на російськомовні назви місяців.

`Loader`: `_contentPath` – `pict.jpg`.

4. У функції `doSubmit` виконати наступні зміни:

```
function doSubmit():Void {
    if(_root.radioGroup.selectedData==1){
        _root.txt.text=fl;
    }
    if(_root.radioGroup.selectedData==2){
        _root.txt.text=dr;
    }
}
```

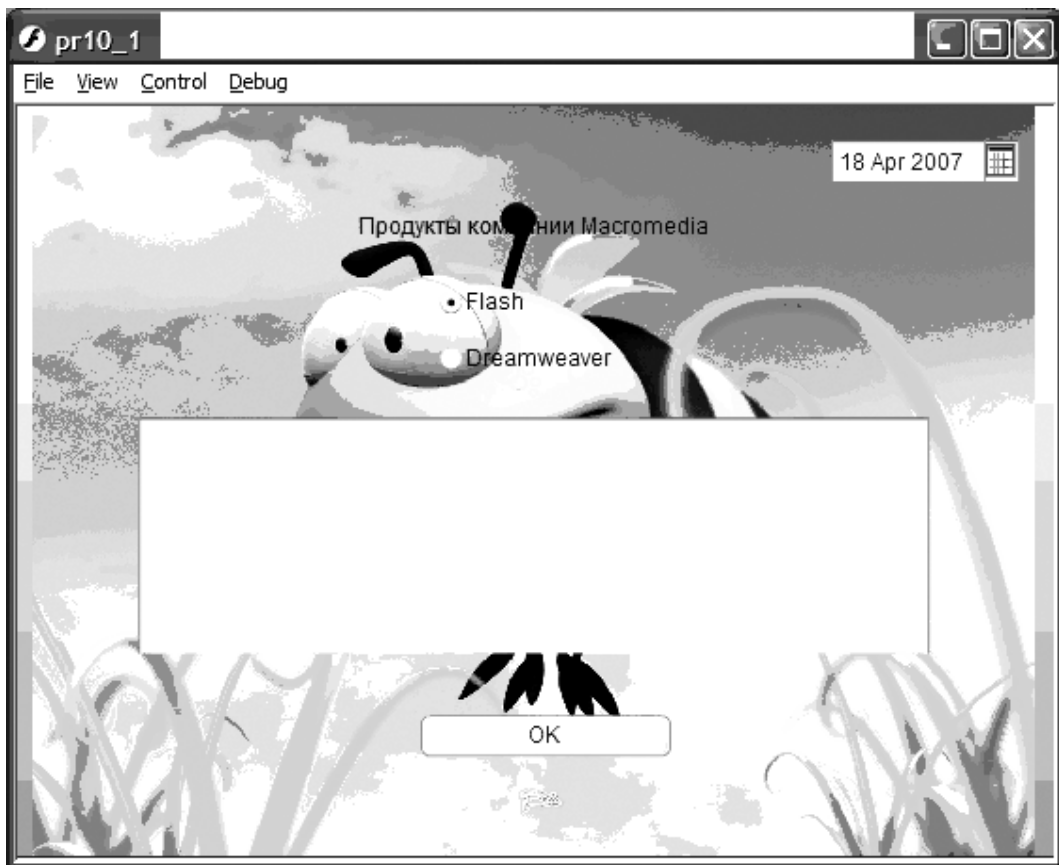


Рис. 23. Змінена динамічна форма

Завдання

1. Створити форми переглядання інформації за допомогою компонентів TextArea, ComboBox, Button, Label.
2. Замінити компонент ComboBox на RadioButton.
3. Додати компоненти DateField і Loader.
4. Продумати призначення динамічної форми у складі індивідуального мультимедійного проекту, описати її в сценарії та додати компоненти, що дозволять керувати проектом і виводити додаткову інформацію.

Складова "Програмне малювання"

Компетентності, що формуються завдяки цій складовій частині:

- вміння та навички програмного малювання у Flash на основі основних функцій;
- вміння програмно зафарбовувати створені рисунки суцільним кольором та градієнтом.

Допоміжний матеріал

Розглянемо приклади малювання прямих ліній, ліній Безьє другого порядку та їх заливання суцільним кольором на прикладах, що наведені нижче.

Для того щоб малювати прямими відрізками, необхідно виконати такі дії.

1. Створити новий порожній мувікліп (ввести код для першого кадру):
`_root.createEmptyMovieClip("myClip", 1).`

2. Створити стиль, лініями якого будуть зображені геометричні фігури:
`myClip.lineStyle(2,0x234567).`

3. Створити стиль заливання
`myClip.beginFill(0x7878FF).`

4. Задати початкову точку малювання:
`myClip.moveTo(70,20).`

5. Накреслити відрізок від початкової точки до кінцевої:
`myClip.lineTo(20,100).`

6. Зобразити геометричну фігуру. Наприклад, червоний трикутник, зелений напівпрозорий квадрат без контура, жовту зірку із синім контуром будь-якої товщини (все це можна побудувати завдяки правильній побудові ліній за координатами).

7. Після зображення кожної фігури необхідно закінчити редагування заливки

`myClip.endFill().`

Для того щоб малювати кривими, необхідно виконати послідовність дій, аналогічну до малювання прямими відрізками, з використанням функції

`curveTo(controlX, controlY, anchorX, anchorY).`

Функція має чотири обов'язкові параметри. За відсутності хоча б одного з них вона не спрацює. Початком кривої вважається поточна позиція курсора ("олівця"), яка встановлюється за допомогою `moveTo()`, або ж позиція, в якій закінчили креслення функції `lineTo()` або `curveTo()`. Кінець кривої вказується параметрами `anchorX` і `anchorY`. Параметри `controlX` і `controlY` указують точку, до якої спрямовані початок і кінець лінії.

Наприклад:

`lineStyle(1);`

`beginFill(0xFF9921, 20);`

`moveTo(70,20);`

```
curveTo(120,20,120,70);
curveTo(120,120,70,120);
curveTo(20,120,20,70);
curveTo(20,20,70,20);
endFill();
```

Для здійснення заливання геометричної фігури градієнтом необхідно виконати такі дії:

```
onClipEvent(load){
import flash.geom.Matrix;

var myMatrix:Matrix = new Matrix();
trace(myMatrix.toString()); // (a=1, b=0, c=0, d=1, tx=0, ty=0)

myMatrix.createGradientBox(200, 200, 0, 50, 50);

var depth:Number = this.getNextHighestDepth();
var mc:MovieClip = this.createEmptyMovieClip("mc_" + depth, depth);
var colors:Array = [0xFF0000, 0x0000FF];
var alphas:Array = [100, 100];
var ratios:Array = [0, 0xFF];
mc.beginGradientFill("linear", colors, alphas, ratios, myMatrix);
mc.lineTo(0, 300);
mc.lineTo(300, 300);
mc.lineTo(300, 0);
mc.lineTo(0, 0);
}
```

Завдання

1. За допомогою методу малювання прямими відрізками зобразити наступні геометричні фігури: фіолетовий шестигранник, помаранчеву напівпрозору трапецію з блакитним контуром, червону зірку із синім контуром довільної товщини.

2. За допомогою методу малювання кривими зобразити жовтий напівпрозорий еліпс з контуром довільного стилю, синусоїду.

3. Накреслити трикутник, зафарбувавши його градієнтним заливанням.

Контрольні запитання

1. У чому полягає проектування мультимедійних проектів?
2. Які основні компоненти складають мультимедійну презентацію?
3. Яким чином можна оптимізувати мультимедійний проект для використання в різних середовищах?
4. У символ якого типу потрібно перетворювати малюнок, щоб створити анімацію руху?
5. Навіщо потрібно задавати ключовий кадр?
6. У чому полягає відмінність між покадровою анімацією і анімацією за допомогою розкадровування?
7. Які додаткові параметри можна задавати для анімації руху?
8. Що таке ідентифікатори форми, для чого вони необхідні?
9. Що таке анімація форми? У чому полягають її особливості?
10. Чи можна у Flash виконувати анімацію форми для груп, символів, текстових блоків і растрових зображень?
11. Які додаткові параметри можна задавати для анімації форми?
12. Що таке символ? Дайте визначення.
13. Для чого призначені символ-кнопка і Movie – Clip?
14. Опишіть послідовність дій для створення анімованої кнопки?
15. На які події може реагувати кнопка?
16. Що таке шар? Дайте визначення.
17. Для чого призначений шар, що спрямовує рух?
18. Опишіть призначення шару-маски.
19. Опишіть послідовність дій для створення анімації за заданим шляхом.
20. Наведіть приклади використання анімації маски.
21. У чому полягає відмінність між символом-контейнером і графічним символом?
22. Яким чином можна вмістити символ-контейнер у кнопку?
23. Які об'єкти може містити символ-контейнер?
24. Яке місце символу-контейнера в інтерактивному застосуванні?
25. Яке призначення має операція трасування, в чому полягає її роль у процесі оптимізації анімаційних додатків?
26. Відеофайли якого типу можна імпортувати до Flash?
27. Які існують способи організації відео у Flash?

28. Чим потоковий звук відрізняється від звуку події?
29. Звук яких форматів можна імпортувати до Flash?
30. Які типи тексту існують у Flash?
31. Яке призначення динамічних текстових полів у Flash?
32. Яким чином можна зв'язати текстовий файл з динамічним текстовим полем?
33. За допомогою яких команд можна створювати та дублювати символи-контейнери?
34. Що таке глибина розташування символу?
35. У яких випадках потрібно видаляти символ-контейнер після закінчення операцій із ним?
36. Яким чином можна програмно змінити властивості символу-контейнера та його координати?
37. Для яких подій символу-контейнера можна створювати програмний код?
38. Для яких елементів Flash можна створювати програмний код?
39. Яке призначення компонентів у Flash?
40. У яких випадках слід використовувати компонент ComboBox, а у яких RadioButton?
41. За допомогою яких компонентів можна працювати з текстовою інформацією?
42. Як можна зв'язати компоненти з функцією обробки дій із ними?
43. Які функції виконує компонент DateField?
44. У яких випадках слід використовувати програмне малювання?
45. Які способи програмного малювання ви можете назвати?
46. Які існують способи градієнтного заливання і чим вони відрізняються?
47. Опишіть послідовність дій для програмного малювання?

Рекомендована література

Комп'ютерна анімація : конспект лекцій для студентів спеціальності 6.092700 "Технологія електронних мультимедійних видань" усіх форм навчання / О. С. Євсєєв, Н. І. Прибиткова. – Х. : Вид. ХНЕУ, 2007. – 104 с.

Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Комп'ютерна анімація" для студентів спеціалізації "Технологія електронних мультимедійних видань" / Укл. О. С. Євсєєв, Н. І. Прибиткова. – Х. : Вид. ХНЕУ, 2008. – 61 с. (Укр. мов.)

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. О. С. Євсєєв, Н. І. Прибиткова. – Х. : Вид. ХНЕУ, 2011. – 51 с. (Укр. мов.)

Зміст

Вступ	3
Контрольна робота "Розробка комплексного анімаційного мультимедійного проекту"	5
Методичні рекомендації	7
Контрольні запитання	40
Рекомендована література	42

