

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

Євсєєв О. С.

КОМП'ЮТЕРНА АНІМАЦІЯ

Навчальний посібник

Харків. Вид. ХНЕУ ім. С. Кузнеця, 2014

УДК 004.93(075.8)

ББК 32.973я73

Є25

Рецензенти: канд. техн. наук, доцент кафедри інформаційних систем Харківського інституту банківської справи Університету банківської справи Національного банку України *Кавун С. В.*; канд. техн. наук, доцент кафедри медіасистем та технологій Харківського національного університету радіоелектроніки *Бізюк А. В.*

Затверджено рішенням вченої ради Харківського національного економічного університету імені Семена Кузнеця.

Протокол № 10 від 02.06.2014 р.

Євсєєв О. С.

Є25 Комп'ютерна анімація : навчальний посібник для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" / О. С. Євсєєв. – Х. : Вид. ХНЕУ ім. С. Кузнеця, 2014. – 152 с. (Укр. мов.)

Наведено основні теоретичні положення та практичні рекомендації щодо розробки комп'ютерної анімації для різних сфер застосування. Розглянуто створення інтерактивних анімаційних додатків як однієї з найбільш перспективних технологій подання інформації в Інтернеті. Досліджено інструментальні засоби для створення анімації, а саме Adobe Flash і мову програмування Action Script. Вивчено використання інструментів в анімації, застосування різних типів шарів, видів анімації, створення рекламної продукції на основі анімації.

Рекомендовано для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання.

ISBN 978-966-676-549-2

УДК 004.93(075.8)

ББК 32.973я73

© Харківський національний економічний університет імені Семена Кузнеця, 2014
© Євсєєв О. С., 2014

Вступ

Стрімкі темпи розвитку інформаційних технологій потребують від учасників бізнес-процесів використання сучасних інструментів для обміну інформацією зі своїми потенційними та наявними клієнтами. Сучасні веб-портали та інші види інтерактивних медіа використовують у своєму складі компоненти, що побудовані у вигляді анімаційних кліпів, браузерних ігор, інструментів збирання маркетингової інформації від користувачів, інтерактивних веб-додатків та віджетів, користуються великою популярністю у користувачів мережі Інтернет. Попит на розробку зазначених елементів дуже великий на сучасному ринку інформаційних послуг, і затребуваність у фахівцях, що мають відповідні компетентності, зростає з кожним днем. У цьому навчальному посібнику розглядаються теоретичні основи та практичні рекомендації, що дозволяють опанувати необхідні компетентності щодо розробки сучасних анімованих інтерактивних кліпів та використання їх у якості компонентів у складі веб-порталів інтерактивних медіа. Більшість зазначених навичок можна набути, оволодівши навчальною дисципліною "Комп'ютерна анімація" та засобом розробки анімаційних додатків Adobe Flash. Комп'ютерна анімація 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 подекуди спростити завдання розробки мультимедійного матеріалу, скоротивши при цьому витрати.

Навчальна дисципліна "Комп'ютерна анімація" вивчається студентами напряму підготовки 6.051501 "Видавничо-поліграфічна справа" спеціалізації 6.05150102 "Технології електронних мультимедійних видань" у 4 семестрі.

У процесі навчання студенти отримують необхідні знання під час лекційних занять, виконують лабораторні завдання щодо створення

анімаційних фільмів різного призначення. Для більш якісного оволодіння навчальним матеріалом для студентів передбачені завдання до лабораторних занять, а також завдання для самостійної роботи та підготовки есе. Такий підхід дозволяє отримати необхідні компетенції та вміти їх реалізувати на більш високому якісному рівні. Вивчивши дисципліну "Комп'ютерна анімація", студенти зможуть не тільки розробляти анімаційні документи, але й створювати повнофункціональні анімовані веб-додатки на основі технології Флеш, що будуть дозволяти користувачу в інтерактивному режимі використовувати обмін інформацією в галузі мультимедійних технологій.

Поточний контроль знань реалізовується під час проведення лабораторних занять шляхом виконання індивідуальних та контрольних завдань і тестування; для підсумкового контролю передбачений залік.

Розділ 1. Створення комп'ютерної анімації за допомогою технології Adobe Flash

1.1. Вступ до анімації

***Мета:** отримання теоретичних знань щодо історії анімації, принципів використання растрової й векторної графіки в анімації, історії виникнення та застосування кривих Безьє, видів анімаційних документів і додатків.*

***Ключові слова:** анімація, оптичні іграшки, мультиплікація, растрова графіка, векторна графіка, фрактальна графіка, криві Безьє.*

1.1.1. Історія анімації

У цьому параграфі йдеться про появу першої анімації, її призначення й змістове навантаження; різні типи анімації, способи їх створення й використання; перспективні напрями розвитку технологій анімації.

! ***Анімація** – це відтворення руху шляхом відображення послідовності малюнків-кадрів із частотою, при якій забезпечується цілісне зорове сприйняття образів [24].*

Анімація може бути зроблена лише з використанням розуміння фундаментального принципу роботи людського зору – інертності зорового сприйняття. Наприклад, якщо в темній кімнаті вимкнути телевізор, що працював, чи екран монітора, то ще деякий час екран буде світлим. Якщо запалити лампу, то око це помітить миттєво. А от якщо лампу вимкнути, то око буде бачити її зображення ще 1/16 секунди. Фактично **людське око бачить світлові хвилі більш тривалий час, ніж вони існують**. На цій властивості людських очей побудоване кіно.

Експерименти з анімацією були розпочаті з виготовлення різних оптичних іграшок. Вони працювали на такому принципі: попереднє зображення ще зберігається в очах, а око вже зауважує наступне зображення. Зображення ніби накладаються один на одного, тому можна побачити загальну картинку. Вперше цей принцип був продемонстрований у 1828 році французом, Паулем Роґетом (Paul Roget). Це був диск, на одному боці якого був зображений птах, а на іншому – порожня клітка. Коли диск обертася, птах з'являвся в клітці (рис. 1.1.1).

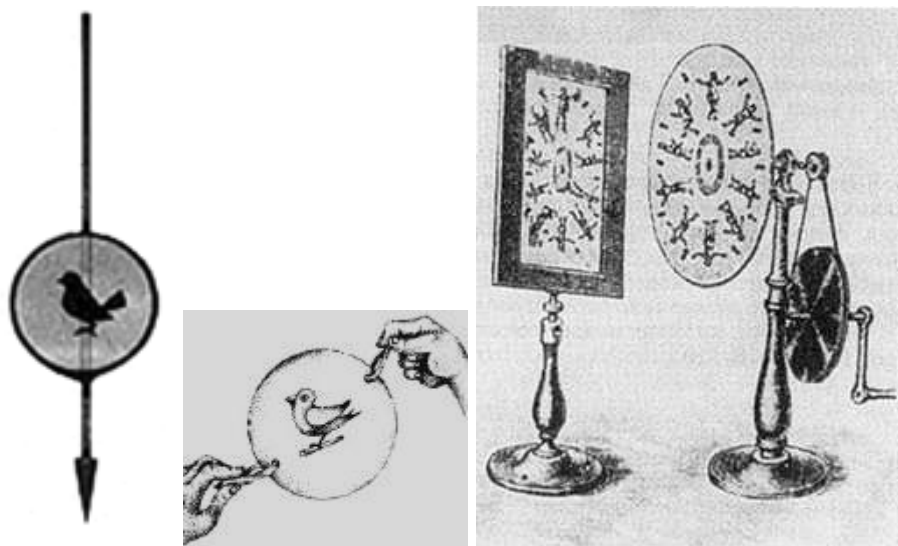


Рис. 1.1.1. Оптичні іграшки й апарати

Наступним кроком було виготовлення апаратів, які створювали враження картинок, що рухаються (рис. 1.1.2).

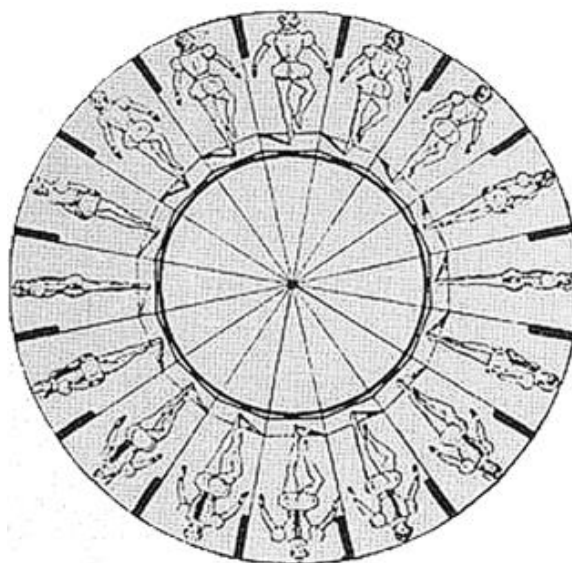


Рис. 1.1.2. Оптичні іграшки й апарати

Вознесенський А. С., згодом відомий сценарист російського кіно, у віці сімнадцяти років напросився на бесіду зі Львом Толстим. Це було в 1897 році, а двадцять років поспіль Вознесенський зізнався, що від цієї зустрічі в його пам'яті утрималася лише одна деталь: "Більше всього мене зацікавила та річ, що Лев Миколайович взяв зі стола й перебирав пальцями під час бесіди... Це була крихітна книжечка (рис. 1.1.3), яку він притримував лівою рукою за корінець, великим пальцем правої руки він змушував швидко перевертатися всі сторінки її, і від цього балерина, зображена на сторінках, повільно піднімала й потім опускала ногу".



Рис. 1.1.3. Анімація з використанням невеличкої книжечки

Така іграшка називалася "фоліоскопом" і належала до великого сімейства оптичних приладів, що носили імена зі схожими закінченнями: **зоотроп, фантаскоп, стробоскоп, тауматроп, праксипоскоп**. Усі вони тим або іншим чином передавали рух, точніше, створювали ілюзію руху [26].

! **Анімаційне кіно** – це особливий вид кіномистецтва, в основі якого лежить оживання на екрані різних неживих об'єктів. Раніше більше розповсюдженим був термін "мультиплікація" (*multiplicatio* – множення).

Ця назва відповідала двом найпоширенішим методам одержання зображення – мальованому й об'ємному. *Мальована анімація* може бути виконана покадровою зйомкою графічної інформації, тіньової (силуетної), заснованої на плоских маріонетках і перекладках, включаючи колажі. *Об'ємна анімація* робиться покадровою зйомкою ляльок або барельєфів [26].

Протягом усього існування людство намагалось відобразити відчуття руху у своєму мистецтві.

Розробка фотокамери й проектора Томасом А. Едісоном (Thomas A. Edison) та іншими забезпечило перший реальний практичний спосіб створення анімації.

Першим мультфільмом, що одержав широку популярність, стала у 1906 році стрічка Дж. Стюарта Блектона "Смішні обличчя", що становила серію простеньких малюнків із зображенням веселих гримас (рис. 1.1.4).



Рис. 1.1.4. Кадри першого відомого мультиплікаційного фільму

Уолт Дісней (1901 – 1966), американський режисер, художник, продюсер (справжнє ім'я – Уолтер Елайос) у 1923 році випускає серію "Аліса в країні мультиплікації". У 1928 році випускає звуковий мультфільм "Пароплавик Віллі", де вперше з'являється герой Міккі Маус.

Історія комп'ютерної анімації тісно пов'язана з появою і розвитком спеціалізованих графічних програмних пакетів. Першим кроком у технології візуальних ефектів була, вигадана в 1961 році Іваном Сазерлендом система Sketchpad, яка започаткувала еру комп'ютерної графіки. В цій системі за допомогою світлового пера користувачі могли створювати малюнки безпосередньо на екрані монітора (рис. 1.1.5).

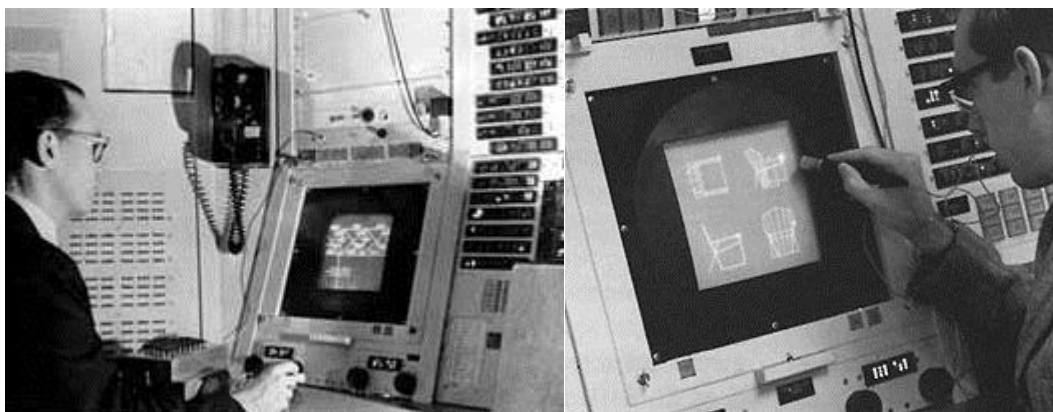


Рис. 1.1.5. Система Sketchpad

У 1967 році Сазерленд разом із Девідом Евансом розпочали роботу зі створення навчального курсу комп'ютерної графіки. Зріс інтерес до цієї галузі. В університеті штату Юта (США), де були започатковані такі дослідження, в цей час працювали: Джим Кларк – засновник компанії Silicon Graphics Inc., Ед Кетмул – один із "піонерів" у галузі створення фільмів за допомогою комп'ютера, Джон Вернок – засновник компанії Adobe Systems.

У 1986 році фірма AT&T випустила перший пакет для роботи з анімацією на персональних комп'ютерах (TOPAS), який коштував 10 000 доларів і працював на комп'ютерах із процесором Intel286 й операційною системою DOS. У 1990 році фірма AutoDesk розпочала продаж продукту 3D Studio. У 1997 році компанія Macromedia придбала у компанії FutureWare невелику графічну програму для веб, з якої була започаткована нині загально відома програма комп'ютерної анімації Adobe Flash. У 1998 році розпочався випуск додатка Maya, що коштував від 15 000 до 30 000 доларів.

Для плавного відтворення анімації необхідна швидкість, що забезпечує зміну частоти кадрів не менше 10 за секунду. Для комп'ютерної анімації частота зміни кадрів за секунду екранного часу складає 10 – 16, для кінематографії – 24, для системи PAL чи SECAM телемовлення – 25, для системи NTSC телемовлення – 30. Більша кількість кадрів дозволяє домогтися плавних рухів персонажів і появи об'єктів у зображенні в різні моменти часу. При недостатній кількості кадрів стають помітні розходження в послідовних зображеннях об'єктів, що призводить до їх різких переміщень. У комп'ютерній анімації на перший план виходить розмір файла, у якому зберігаються зображення. Тому при створенні комп'ютерної анімації намагаються знайти компроміс між якістю анімації і розміром файла, що і визначає загальну кількість кадрів анімації.

Різниця між анімацією і відео полягає в тому, що відео використовує безупинний рух і розбиває його на множину дискретних кадрів, а анімація використовує множину незалежних малюнків або графічних файлів, що виводяться в певній послідовності для створення ілюзії безупинного руху.

Процес створення анімації дуже простий. Фактично він будується на повторенні таких етапів:

- 1) розміщення об'єктів у заданих точках екрана;
- 2) відображення об'єктів протягом визначеного проміжку часу;
- 3) знищення об'єктів.

1.1.2. Растрова й векторна графіка

У цьому параграфі йдеться про принципи використання растрової й векторної графіки в анімації. Розглянуто основні характеристики файлів, які необхідно враховувати для використання в анімаційних документах.

! **Звичайна графіка** – це результат візуального подання реального або уявного об'єкта, отриманий традиційними методами – малюванням або друкуванням.

! **Комп'ютерна графіка** включає також методи і засоби створення і обробки зображень за допомогою програмно-апаратних комплексів і охоплює всі види і форми подання зображень, доступних для сприйняття людиною на екрані монітора або у вигляді копії на певному носії [24].

Залежності від способу опису та формування зображення розрізняють растрову, векторну та фрактальну графіку.

Історично термін "растр" вказував на те, що пристрій у процесі відтворення зображення використовує набори пікселів (точок), організовані в вигляді послідовностей рядків розгортки. Растрові дані є набором числових значень, які визначають кольори окремих пікселів, упорядкованих таким чином, щоб їх легко було відобразити на растрових пристроях.

Растрова графіка використовується в випадках, коли потрібна висока точність у передачі кольорів і напівтонів. Однак при цьому розміри файлів суттєво збільшуються зі зростанням роздільної здатності (одиниці, десятки і сотні Мбайт). До недоліків растрової графіки, окрім великих розмірів файлів, слід віднести пікселізацію зображень при їх збільшенні та деформацію при зменшенні.

У векторній графіці базовим елементом зображення є лінія, яка описується математично як єдиний об'єкт, тому обсяг даних для відображення об'єкта засобами векторної графіки суттєво менший, ніж у растровій графіці. Лінія характеризується формою, товщиною, кольором, типом (суцільна, пунктирна тощо). Замкнуті лінії мають властивість заповнення простору, що ними охоплюються, іншими об'єктами або кольором. Найпростішими лініями є пряма (нескінченна), відрізок прямої, криві другого порядку (не мають точок згину – параболи, гіперболи, еліпси, кола), криві третього порядку (можуть мати точки згину), криві Безьє (засновані на використанні пари дотичних, проведених до відрізка лінії в її кінцях, кути нахилу і довжина яких впливають на форму лінії).

Векторна графіка зручна для зберігання й обробки зображень, що складаються з ліній, або можуть бути розкладені на прості геометричні об'єкти. Векторні дані легко масштабувати та виконувати над ними інші перетворення (наприклад, повертання зображення, додавання, видалення або зміну окремих елементів зображення). Поряд із цим векторні файли важко застосувати для зберігання складних фотореалістичних зображень. Векторні дані краще відображаються на векторних пристроях виводу (плотерах, дисплеях із довільним скануванням). Ефективно векторну графіку можна відобразити тільки на растрових дисплеях із високою роздільною здатністю.

На рис. 1.1.6 подано приклад масштабування растрової та векторної графіки. З цього прикладу можна побачити, як у випадку з растровою графікою спостерігається погіршення якості, нерівність країв об'єктів, неоднорідність заливання й т. д. У ході масштабування векторної графіки якість малюнка не залежить від його розмірів, малюнок можна збільшувати в багато разів, не спостерігаючи при цьому яких-небудь змін якості.

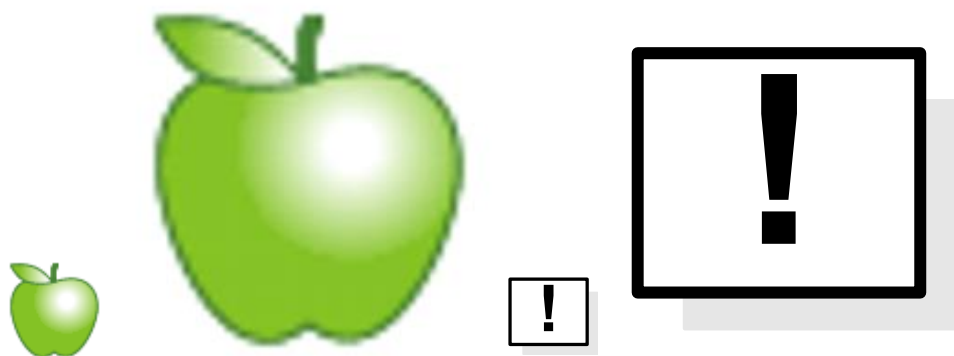


Рис. 1.1.6. Порівняння масштабування векторної та растрової графіки

Фрактальна графіка, як і векторна, заснована на математичних обчисленнях. Її базовим елементом є математична формула, виключно на основі якої будується зображення. Таким способом будують як найпростіші регулярні структури, так і складні ілюстрації, що імітують природні ландшафти і тривимірні об'єкти.

Слід підсумувати викладене й сформулювати види, якими може бути подана анімація: класична (мальована), лялькова, flash-анімація, 3D-анімація.

Розглядаючи види програмної анімації, можна виділити три види:

- анімація на основі зміни окремих графічних елементів;
- анімація на основі векторного морфінгу (Flash-анімація);
- програмна тривимірна анімація.

1.1.3. Криві Безьє

У цьому параграфі наведено дані про історію появи кривих Безьє, їх математичний зміст, різні види кривих Безьє, переваги використання в анімації.

Криві Безьє були розроблені у 60-х роках ХХ століттях незалежно один від одного П'єром Безьє з автобудівної компанії "Рено" і Полем де Кастельжо з компанії "Сітроен", де застосовувалися для проектування кузовів автомобілів. Незважаючи на те, що відкриття де Кастельжо було зроблено трохи раніше, ніж Безьє, його дослідження не публікувалися й ховалися компанією як виробнича таємниця до кінця 1960-х років. Криві Безьє були опубліковані в 1962 році П'єром Безьє. Ім'ям де Кастельжо названий розроблений ним рекурсивний метод обчислення й побудови кривих (алгоритм де Кастельжо).

Згодом це відкриття стало одним із найважливіших інструментів систем автоматизованого проектування і програм комп'ютерної графіки.

Крива Безьє – параметрична крива, що задається виразом:

$$B(t) = \sum_{i=0}^n P_i b_{i,n}(t), \quad 0 < t < 1,$$

де P_i – функція компонентів векторів опорних вершин;

$b_{i,n}(t)$ – базисні функції кривій Безьє, називані також поліномами Бернштейна [29].

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}; \quad \binom{n}{i} = \frac{n!}{i!(n-i)!},$$

де n – ступінь полінома;

i – порядковий номер опорної вершини.

Кубічні криві Безьє (рис. 1.1.7).

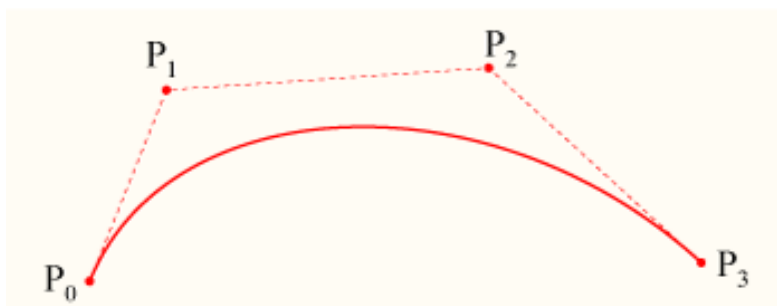


Рис. 1.1.7. Кубічні криві Безьє

Кубічні криві Безьє ($n = 3$), що широко використовуються в графічних редакторах, задаються рівнянням:

$$B(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, \quad t \in [0, 1].$$

Система кривих Безьє, заснована на тригонометричних формулах, виявилася дуже вдалою й простою у використанні, була покладена в основу графічних і багатьох інших програм. У комп'ютерній графіці криві Безьє займають одне з головних місць. Це не тільки основа векторної графіки, але й спосіб опису шрифтів, універсальний спосіб передачі виділення. Використовуючи однаковий принцип побудови кривих Безьє, з ними однаково легко працювати в будь-якій програмі.

Основна функція кривих Безьє – малювання геометричних фігур різної складності. За допомогою кривих Безьє створення складних геометричних фігур зводиться до дуже простих й інтуїтивно зрозумілих маніпуляцій. Фігури, отримані таким чином, можна трансформувати й масштабувати без втрати якості (це одна з головних переваг векторної графіки) [28]. Інше основне призначення кривих Безьє – виділення. Розмір файла від додавання в нього контуру майже не збільшується.

1.1.4. Види анімаційних документів і додатків

Параграф містить відомості про галузі застосування анімації; особливості побудови анімаційних документів.

Можна виділити основні сфери застосування програмної анімації [29]:

- Анімовані елементи оформлення сайтів для залучення уваги відвідувачів, використання програмної анімації в рекламних банерах.
- Анімовані елементи користувальницького інтерфейсу (динамічні курсори, кнопки і т. д). За допомогою програмної анімації можна створювати цікаві ефекти реагування на дії користувача. Такі елементи керування можуть реагувати більш вибірково на дії користувача, наприклад, відслідковуючи напрями руху курсора миші або частоту натискань її кнопки.
- Елементи анімаційних фільмів. Можливо застосування програмної анімації разом зі звичайною, "ручною" анімацією. Програмна анімація гарна для створення циклічних фонових процесів. На відміну від звичайних циклів анімації, створених вручну, можна одержати більш складний і неповторний рух, а також більш цікаві варіанти руху залежно від дій користувача.
- Ілюстрації різних процесів для навчальних програм. Програмна анімація цікава для ілюстрації деяких фізичних процесів і для побудови графіків математичних залежностей, діаграм у формах голосування.
- Ігри. Ускладнення поведження об'єктів і персонажів у грі залежно від подій, що відбуваються, і введення випадкових складових дозволяє створювати більш цікаві ігри.

Одним із потужних програмних продуктів, призначеним для створення комп'ютерної анімації, є Adobe Flash. Цей інструментальний засіб одержав всесвітнє поширення в основному завдяки вдалому сполученню інструментів

як для "ручної" анімації за допомогою комп'ютера, так і для програмної анімації, з використанням спеціально вбудованої мови програмування Action Script.

Основні підходи до створення комп'ютерної анімації

Комп'ютерну анімацію можна розділити на два основних типи:

1) анімація, створена з використанням підходів, які застосовуються у звичайній анімації, але отримана завдяки комп'ютеру;

2) комп'ютер використовується не тільки для створення й відображення кадрів, але й для динамічної зміни властивостей об'єктів, програмним способом можна задавати поведження об'єктів, їхню реакцію на зовнішні впливи, створювати сцени й управляти об'єктами на основі алгоритмічного підходу й т. д.

Другий тип анімації називають програмною анімацією.

Об'єкти у програмній анімації можуть бути створені дизайнером на етапі проектування або згенеровані динамічно. Досить часто застосовується зміна координат об'єкта, його розміри, кут повороту, динамічні з прозорістю, що змінюється. Контроль за властивостями об'єктів і керування ними можуть здійснюватися з окремого кліпу. Але також розповсюджений підхід, коли об'єкт містить у собі всю необхідну функціональність, максимально незалежний у поведженні від інших об'єктів у кліпі й сам контролює своє поведження.

Об'єкти, на яких будується програмна анімація, бувають як простими (окружності, відрізки, відрізки-крапки тощо), так і досить складними – це можуть бути, наприклад, кліпи з анімацією з декількома рівнями вкладеності.

Контрольні запитання

1. Опишіть принципи, на яких заснована анімація. Які оптичні іграшки вам відомі?

2. З чого починається історія комп'ютерної анімації? Опишіть основні етапи.

3. У чому полягає різниця між растровою, векторною й фрактальною графікою?

4. Розкажіть про сутність, призначення й математичний зміст кривих Безьє.

5. Опишіть сфери використання комп'ютерної анімації.

Практична складова "Ознайомлення з практичними аспектами створення найпростішої анімації, аналіз прикладів анімаційних кліпів"

Мета виконання завдання:

- вивчити конструкцію оптичних іграшок;
- створити власну оптичну іграшку, попередньо продумавши її сутність і конструкцію;
- ознайомитися із запропонованими прикладами анімованих роликів, відповідно до наведеного плану провести їхній аналіз.

Допоміжний матеріал

Оптична іграшка

Для ознайомлення з принципами створення найпростіших анімаційних сюжетів можна самостійно створити нескладний оптичний пристрій (іграшку), яка б дозволяла відтворювати в спрощеному вигляді певну динамічну сцену навколишнього світу.

Для створення оптичної іграшки рекомендується скористатися такою процедурою.

1. Розглянути конструкцію тих оптичних іграшок, які розглядалися на лекції. Зібрати додаткову інформацію в Інтернеті або в друкованих виданнях.

2. Продумати сценарій анімації (він повинен бути найпростішим).

3. Вибрати той вид оптичної іграшки, що найкращим способом підходить для завдання.

4. Намалювати основні об'єкти анімації. Для малювання можна використовувати програми комп'ютерної графіки, отримані зображення роздрукувати.

5. Зібрати оптичну іграшку й випробувати її.

Дороблена до наступної лабораторної роботи оптична іграшка демонструється аудиторії й пропонується для обговорювання.

Аналіз анімованих кліпів

Для більш повного сприйняття переглянутих анімаційних роликів, розроблених різними авторами, необхідно проаналізувати поданий у них матеріал і технічний бік їхнього створення. Для проведення аналізу рекомендується скористатися такою процедурою.

1. Придумати назву переглянутого ролика.
2. Описати основну ідею (призначення створення або основну мету).
3. Зробити поділ анімації на окремі сцени, описати їх суть.
4. Виділити для обраних сцен окремі об'єкти, як анімовані, так і статичні.
5. Описати характер поведінки анімованих об'єктів.
6. Описати конструкцію окремих об'єктів (із яких фігур, примітивів, зображень складається кожний об'єкт).
7. Чи наявне програмне малювання у кліпі чи анімовані об'єкти намальовані за допомогою комп'ютера?

Результати проведеного аналізу слід подати у вигляді невеликого ілюстрованого звіту.

1.2. Середовище розробки Adobe Flash

Мета: ознайомлення з теоретичними аспектами Flash-технології та її призначенням, вивчення особливостей користувальницького інтерфейсу та принципів роботи з об'єктами.

Ключові слова: Adobe Flash, векторний морфінг, тимчасова діаграма, інтерфейс користувача, інструменти вибору.

1.2.1. Основи технології Adobe Flash

У цьому параграфі йдеться про історію Flash-технології та призначення Adobe Flash.

! Під терміном "**Flash**" необхідно розуміти технологію, що дозволяє створювати й використовувати інтерактивні анімовані мультимедійні додатки.

Сфера використання Flash: презентації на дисках, веб-сайти, рекламні банери, ілюстрації до мультимедійних видань і навчальних посібників, ігри та мультфільми. У ході створення додатка можуть бути використані звукові й графічні файли (растрові й векторні), відеопотік, можна створювати інтерактивні інтерфейси й повноцінні веб-додатки з використанням PHP і XML.

У середовищі розробки Adobe Flash, створюються вихідні файли з розширенням .fla, потім файли компілюються у формат, що відтворює Flash Player – .swf (Shockwave Flash).

! В основі технології Flash лежить **векторний морфінг**, – плавне "перетікання" одного ключового кадру в іншій. Це дозволяє робити досить складні мультиплікаційні сцени, задаючи лише кілька ключових кадрів для кожного персонажа.

Іншою перевагою технології Flash є повна програмованість. Flash використовує мову програмування Action Script – об'єктно-орієнтовану мову, схожу за синтаксисом з JavaScript. На цей час є версії Action Script 2.0 та 3.0.

Можливість легкого перенесення і створення інтерактивних мультимедійних додатків обумовили швидке зростання популярності формату SWF серед веб-дизайнерів. Тому майже одночасно з появою самого формату, фірмою Macromedia були створені компоненти, що вбудовуються (Plug-In) для двох основних браузерів: Internet Explorer і Netscape Communicator. А це, у свою чергу, сприяло ще більш широкому поширенню SWF на просторах мережі Інтернет. У результаті розроблювачі цих браузерів оголосили про намір включити підтримку SWF безпосередньо в ядро своїх продуктів [1]. Наявність простих і ефективних інструментів для створення Flash-додатків і описана підтримка технології Flash слугували причиною її подальшого розвитку, стрімкого поширення й захоплення значної частки ринку анімованих додатків.

Основні функціональні складові Adobe Flash, які забезпечують високу затребуваність Flash-додатків:

- векторна графіка;
- підтримка декількох видів анімації;
- можливість створення інтерактивних елементів інтерфейсу;
- підтримка взаємодії з імпортованими графічними форматами (у тому числі растровими);
- можливість додавання синхронного звукового супроводу;
- забезпечення експорту Flash-фільмів у формат HTML, а також у кожному із графічних форматів, що використовуються в Інтернеті;
- платформова незалежність;
- можливість перегляду Flash-фільмів як в автономному режимі, так і за допомогою веб-браузера;
- наявність інструментів візуальної розробки, які звільняють розроблювача Flash-фільмів від багатьох рутинних операцій, а також від детального вивчення технічних аспектів реалізації Flash-технології.

Підтримка мови HTML5 у Flash і Toolkit for CreateJS. Кількість різноманітних пристроїв і платформ постійно збільшується, створення уніфікованого контенту стає все більш складним завданням. Adobe® Flash® Professional Toolkit for CreateJS – це безкоштовне розширення, доступне користувачам Flash Professional CS6, яке дозволяє перейти на

створення контенту на базі HTML5, одночасно з використанням багатьох можливостей зі створення анімації та графіки Flash Professional CS6.

Розширення підтримує більшість основних можливостей роботи з анімаціями та ілюстраціями Flash Professional, включаючи вектори, растрові зображення, класичні анімації, звуки, напрямні руху, анімовані маски

і розробку сценаріїв JavaScript на часовій шкалі. Розширення Toolkit for CreateJS дозволяє одним натисканням кнопки експортувати вміст у робочій області та бібліотеці, наприклад JavaScript, для попереднього перегляду в браузері, завдяки чому можна швидко почати створювати привабливий вміст на основі HTML5. За допомогою розширення Toolkit for CreateJS користувачі Flash Pro можуть виконати перехід до HTML5.

Переваги Flash порівняно з HTML5:

- розвинені можливості роботи з мікрофоном, веб-камерою;
- можливість на Flash створювати додатки для Multi-touch екранів, програми для offline використання з безпроблемним перекладом додатків в веб або на мобільний пристрій;
- за допомогою Flash під Adobe Air можна створювати додатки для роботи з Kinect (безконтактний ігровий контролер);
- можливість захистити передачу даних і контенту від злодіяств;
- кросплатформність, що дозволяє зробити додаток для роботи в браузері, завантажити на комп'ютер з AppStore або GooglePlay. Працювати буде аналогічно на всіх платформах. Необхідний 1 розробник під всі платформи, а також працездатність у всіх браузерах із Desktop / Notebook платформ. А це понад 90 % пристроїв, які використовують відвідувачі веб-сайтів.

1.2.2. Користувальницький інтерфейс

Цей параграф містить відомості про: структуру основного вікна Adobe Flash і опис основних об'єктів; робочу область; панелі інструментів; особливості інтерфейсу й подання інформації.

Організація користувальницького інтерфейсу

Організація користувальницького інтерфейсу редактора Adobe Flash є звичною порівняно з іншими Windows-додатками (смуга меню, кнопки основної панелі інструментів, на яку винесені команди, що найбільш часто використовуються й т. д). Найбільш помітною й важливою відмінністю

в інтерфейсі є незвичайна панель часової діаграми. Основні елементи вікна Adobe Flash подані на рис. 1.2.1.

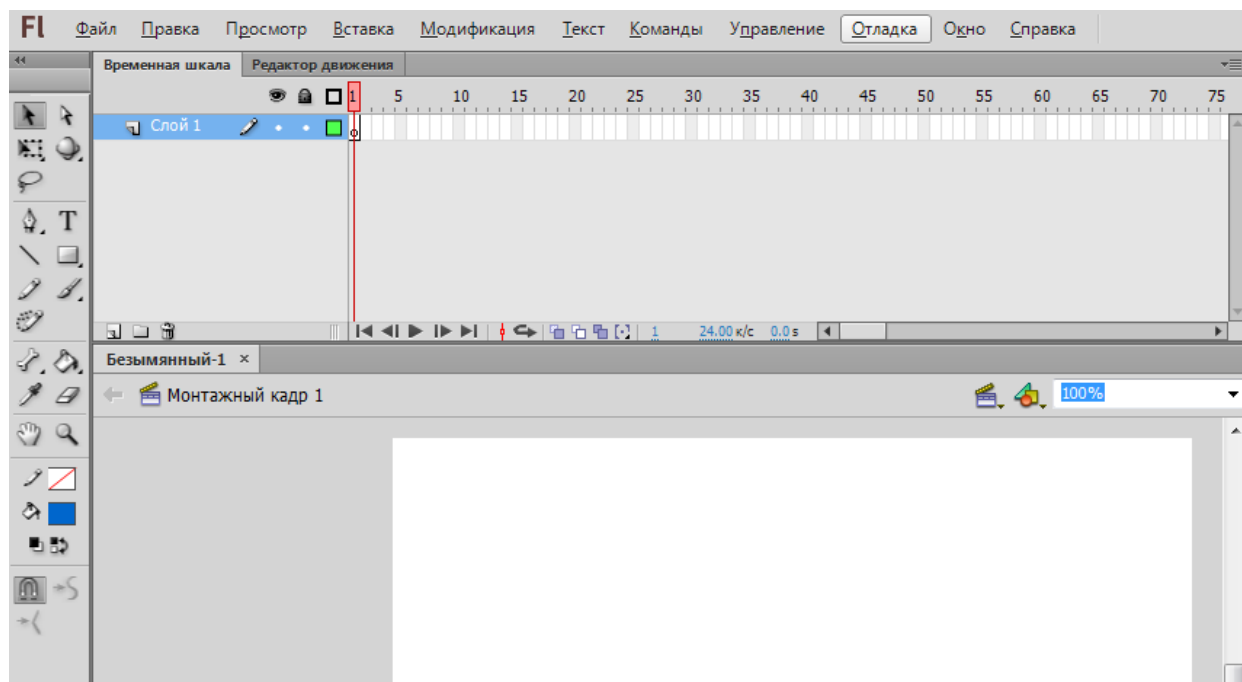


Рис. 1.2.1. Структура основного вікна Flash

Панель інструментів редагування

Панель інструментів редагування, що розташована вертикально в лівій частині, забезпечує доступ до інструментів, які застосовуються для створення й редагування графічних об'єктів. Багато які із цих інструментів можуть бути знайомі по роботі із графічними редакторами. Для зручності роботи панель інструментів розділена на чотири частини (рис. 1.2.2):



Рис. 1.2.2. Панель інструментів редагування

- *Tools* (інструменти), у якій зібрані кнопки вибору конкретного інструмента; ці інструменти розділяються на два види: інструменти вибору й інструменти малювання;

- *View* (вид), що містить засоби керування переглядом зображення, що розташовуються на столі; у цьому полі містяться дві кнопки;
- *Hand Tool* (рука) – клацання на кнопці вмикає режим, при якому робочу область можна переміщати в будь-якому напрямі за допомогою миші, не користуючись смугами прокручування;
- *Zoom Tool* (масштаб) – клацання на кнопці вмикає режим швидкого масштабування зображення в робочій області; при вимкненні цього режиму в полі Options з'являються дві додаткові кнопки, що дозволяють вибрати напрям масштабування (збільшення або зменшення);
- *Colors* (кольори) – кнопки забезпечують роздільний вибір кольору контуру й заливки об'єктів; докладніше їхнє призначення розглянуто в розділі "Робота з кольором";
- *Options* (параметри) – подані елементи установки додаткових параметрів обраного інструмента; для інструментів, що не мають додаткових параметрів, поле Options залишається порожнім.

Робоча область

У центральній частині вікна розташована **робоча область**. У робочій області виконуються будь-які операції редагування об'єктів, але в "кадр" потрапляють тільки ті об'єкти (або їхні фрагменти), які розташовані в межах **монтажного столу**. Частина робочої області, що перебуває за межами монтажного столу призначена для:

- виконання "чорнових" робіт;
- для реалізації ефекту поступового входу об'єкта в кадр (або навпаки, виходу з нього).

Над панеллю тимчасової діаграми розташована панель інструментів, що належить до робочої області. На ній розміщені такі елементи інтерфейсу (рис. 1.2.3):

- кнопка зі стрілкою, що забезпечує повернення в режим редагування всієї сцени;
- текстове поле з ім'ям сцени, що редагується (або сцени, до якої належить символ, що редагується);
- текстове поле з ім'ям символу, що редагується; якщо даний символ містить усередині себе інші символи, то при переході до редагування вкладеного символу на панелі відображається ланцюжок імен символів, що відповідає їхньої ієрархії;

- кнопка вибору сцени; клацання на кнопці відкриває меню, що містить список сцен фільму;
- кнопка вибору символу; клацання на кнопці відкриває меню, що містить список символів фільму;
- список, для масштабування відображення робочої області.

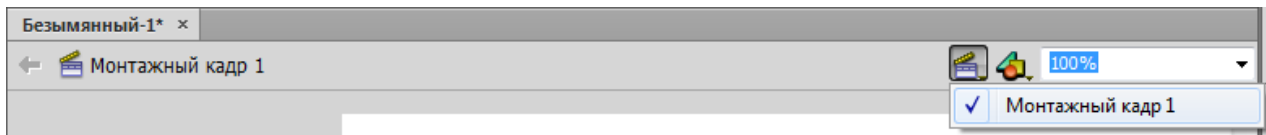


Рис. 1.2.3. Панель інструментів робочої області

1.2.3. Робота з окремими об'єктами

Цей параграф містить інформацію про вибір і виділення об'єктів, а також про інструменти для роботи з об'єктами.

! Термін "**об'єкт**" у Adobe Flash використовується для позначення будь-якого елемента фільму, що розташовується на столі. Використовуючи інструменти Flash, можна переміщувати, копіювати, видаляти, трансформувати об'єкти, змінювати порядок розміщення, вирівнювати їх відносно один одного й групувати.

Вибір і виділення об'єктів

Для виконання маніпуляцій над об'єктом його необхідно попередньо **вибрати** (обрана область – Selection). **Виділення** (highlight – підсвічування) є відповідною реакцією програми на дію користувача. Для візуального виділення об'єкта використовуються спеціальні засоби: інверсний колір, штрихування й т. д. У Flash, на відміну від багатьох інших програм, використовується декілька різних способів виділення обраного об'єкта. Спосіб виділення залежить від способу вибору об'єкта, а також від плану його подальших перетворень.

У Flash можна вибрати об'єкт цілком або тільки його частину, групу об'єктів, екземпляр символу, текстовий блок (одне слово або декілька). Для вибору об'єктів, розташованих у робочій області, зручніше за все використовувати інструменти, розміщені на панелі *Tools*.

Інструменти для вибору об'єктів

Графічні об'єкти у Flash варто розглядати як ті, що складаються із двох компонентів: контуру (Stroke) і заливки (Fill). Властивості цих компонентів (колір, форма, взаємне розташування) можуть змінюватися користувачем незалежно один від одного. Можна також видалити один із цих компонентів.

Слід розглянути інструменти вибору (рис. 1.2.4):



Рис. 1.2.4. Інструменти вибору

- *Arrow Tool* (стрілка), за допомогою якого можна вибрати на столі довільний об'єкт або область прямокутної форми; область може містити групу об'єктів, один об'єкт або його частину; інструмент впливає і на контур об'єкта, і на заливку, тобто якщо в область вибору потрапляє й контур, і заливка, то обраними будуть обидва компоненти; обрана область може бути переміщена або відредагована незалежно від іншої частини зображення;
- *Subselection Tool* (вибір підобласті), що дозволяє вибрати (виділити) об'єкт цілком; якщо обрана область захоплює кілька об'єктів, то обраними будуть усі ці об'єкти; даний інструмент дозволяє змінювати форму об'єкта, але виділений об'єкт не може бути переміщений;
- *Lasso Tool* (петля), що забезпечує вибір області довільної форми; у цю область можуть бути включені декілька об'єктів (або їхні фрагменти), окремий об'єкт, або його частина; інструмент впливає й на контур об'єкта і на заливку; виділена область може бути переміщена або відредагована незалежно від іншої частини зображення;
- *Free Transform Tool* (довільна трансформація), за допомогою якого можна вибрати на столі довільну область прямокутної форми; область може містити групу об'єктів, один об'єкт або його частину; інструмент впливає й на контур об'єкта, й на заливку; відмінність від інструмента *Arrow* полягає в тому, що даний інструмент має більші функціональні можливості з редагування контуру й заливки об'єкта;
- *Pen Tool* (перо), що насправді належить до інструментів малювання, але може застосовуватися для вибору контуру об'єкта.

Контрольні запитання

1. На чому ґрунтується Flash-технологія? Опишіть її основні переваги та недоліки.
2. Яка історія появи Flash-технології? Опишіть основні етапи її розвитку.

3. Опишіть структуру основного вікна Adobe Flash і подайте опис основних об'єктів.

4. Опишіть складові панелі інструментів редагування.

5. Опишіть характеристики інструментів для вибору об'єктів у Adobe Flash. У чому їхнє розходження?

1.3. Основні інструменти Adobe Flash

Мета: ознайомлення з основними інструментами малювання Adobe Flash, вивчення принципів роботи з кольором та основних підходів до додавання текстової інформації.

Ключові слова: інструменти малювання, малювання кривих, робота з кольором, статичний текст.

1.3.1. Інструменти малювання

У цьому параграфі наведено інформацію про панель інструментів малювання, основні елементи та їхні властивості, *Oval Tool* (овал), *Rectangle Tool* (прямокутник), *Pen Tool* (перо), *Pencil Tool* (олівець), *Brush Tool* (пензлик), *Eraser Tool* (ластик); малювання кривих за допомогою пера.

Adobe Flash є досить потужним графічним редактором, що дозволяє повністю забезпечити потреби розробників у створенні статичних зображень і подальшої розробки Flash-фільму.

Для малювання використовуються інструменти, розташовані на панелі *Tools* (рис. 1.3.1) [24].

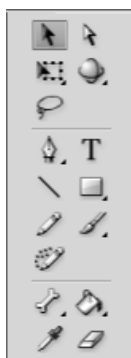


Рис. 1.3.1. Інструменти малювання

До основних інструментів належать:

- *Line Tool* (лінія) – інструмент, що дозволяє малювати прямі й ламані лінії, які при необхідності можуть використовуватися як контур об'єкта; інструмент не має кнопок-модифікаторів; як параметри настроювання можна вибрати товщину й колір лінії; настроювання виконується за допомогою інспектора властивостей (рис. 1.3.2.а);

- *Oval Tool* (овал) – інструмент, призначений для створення стандартних геометричних фігур: еліпсів і окружностей різного радіуса; створена фігура складається із двох компонентів, які можуть редагуватися окремо один від одного: контуру (Outline) і заливки (Fill); параметри – товщина й колір лінії (рис. 1.3.2.б);
- *Rectangle Tool* (прямокутник) – інструмент, призначений для створення відповідних геометричних фігур; отримана фігура складається з контуру й заливки; як додатковий параметр можна вказувати радіус округлення кутів прямокутника; можна вибирати товщину й колір лінії (рис. 1.3.2.в);
- *Pen Tool* (перо) – інструмент, що дозволяє малювати прямі й згладжені криві лінії з високою точністю на основі механізму кривих Безьє; для цього в процесі малювання Flash створює базові точки (Anchor points) і потім будує за ними намальований сегмент; даний інструмент є, мабуть, найбільш складним у використанні, тому його робота буде розглянута окремо (рис. 1.3.2.г);
- *Pencil Tool* (олівець) – інструмент, призначений для малювання довільних ліній, однак при відповідному налаштуванні параметрів він виконує "розпізнавання" намальованих фігур і автоматичне їхнє перетворення до стандартних геометричних фігур (рис. 1.3.3.а);
- *Brush Tool* (пензлик) – інструмент дозволяє малювати лінії, що нагадують мазки пензликком; незважаючи на відносну простоту використання, пензлик має досить велику кількість додаткових параметрів (рис. 1.3.3.б);
- *Eraser Tool* (ластик) – це "гумка, що стирає", за принципом дії й за набором параметрів близька до інструмента *Brush Tool* (рис. 1.3.3.в).

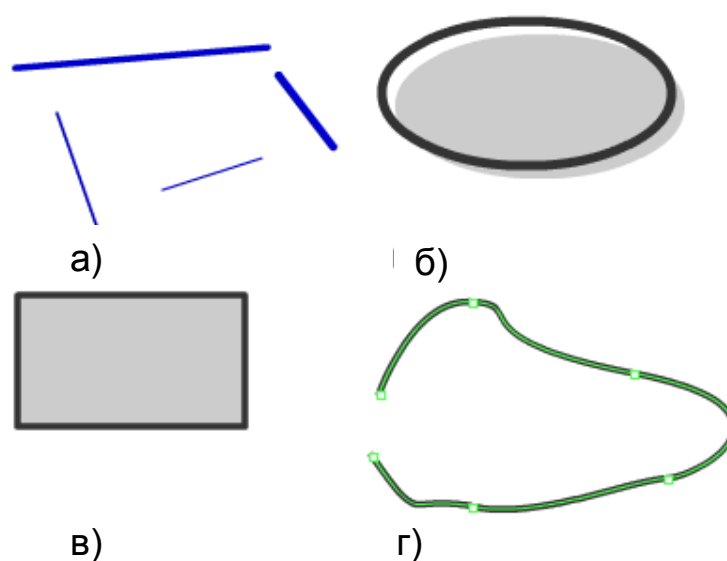


Рис. 1.3.2. Інструменти малювання

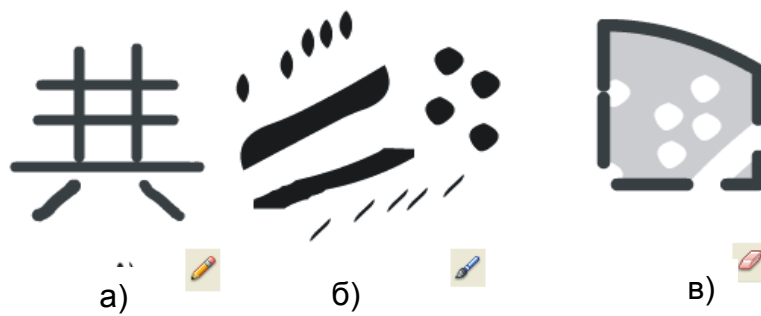


Рис. 1.3.3. Інструменти малювання

Результати використання будь-якого з інструментів можуть бути скасовані або скореговані. Наприклад, для зміни форми ліній і контурів об'єктів можна використовувати інструмент *Arrow* (розглянутий при описі операцій вибору об'єктів).

Тепер слід розглянути докладніше особливості застосування деяких інструментів.

Використання інструмента малювання і його налаштувань на прикладі *Rectangle Tool*. При роботі з інструментом *Rectangle* можна змінювати такі параметри:

- величина округлення кута прямокутника – кнопка-модифікатор *Round Rectangle Radius* (радіус округлення – прямокутника), у діалоговому вікні, що відкрилося, вводиться величина округлення;
- колір контуру й колір заливки – обираються за допомогою інструментів на панелі інспектора властивостей;
- також за допомогою інспектора властивостей установлюються товщина й стиль лінії контуру;
- додатковий параметр *Sharp Corners* (гострі кути), що задає форму контуру; зміна виконується в додатковому вікні *Custom...*(налаштування) (рис. 1.3.4), у цьому вікні також можна змінити товщину контуру.

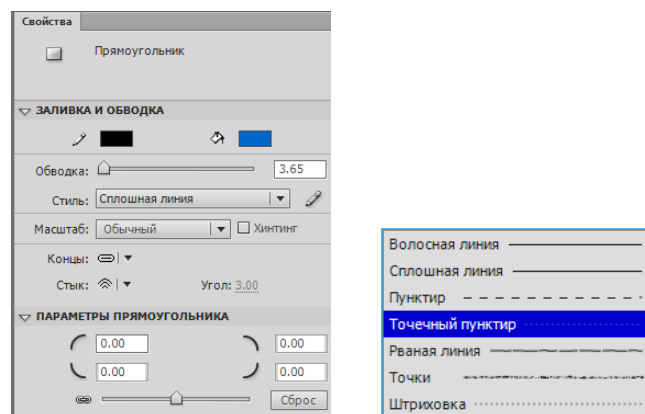


Рис. 1.3.4. Діалогове вікно для установки додаткових параметрів інструмента *Rectangle*

Після того, як за допомогою одного із трьох інструментів фігура буде намальована, для корегування параметрів можна використовувати розширений формат інспектора властивостей (рис. 1.3.5). Він відрізняється наявністю чотирьох полів, що містять таку інформацію:

- для лінії – координати лівого (поля X і Y) і правого (поля W і H) її закінчень; координати вимірюються щодо верхнього лівого кута столу фільму;
- для кола – координати центра (поля X і Y) і радіус (поля W і H);
- для прямокутника – координати верхнього лівого кута (поля X і Y) і довжини сторін (поля W і H).

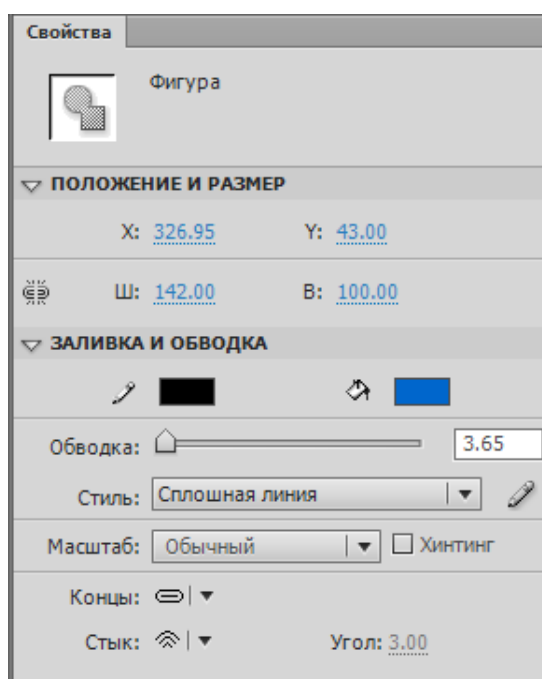


Рис. 1.3.5. Розширений формат інспектора властивостей для стандартних фігур

Інструмент Pen Tool

Прямі або ламані лінії, або сегменти кривих будуються за базовими точками на основі механізму кривих Безьє. Щоб намалювати пряму лінію, необхідно вказати першу й останню точку відрізка. Наприклад, щоб одержати ламану, що складається із трьох відрізків, потрібно чотири точки (рис. 1.3.6а). Намальовану лінію можна залишити незамкнутою (двічі клацнути на останній крапці лінії), або перетворити в замкнуту фігуру (необхідно клацнути на початковій крапці). У першому випадку достатньо щоб створити замкнуту фігуру (рис. 1.3.6б).

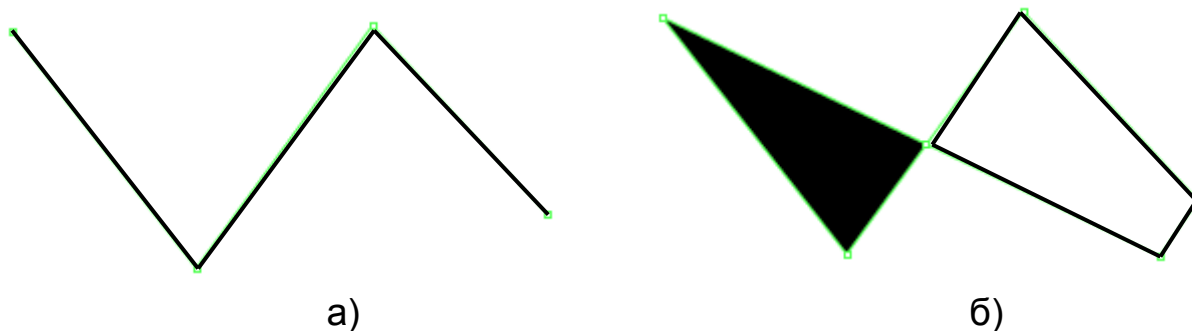


Рис. 1.3.6. Ламана лінія, намальована інструментом Pen Tool

Також можна намалювати криві за допомогою пера, на основі переміщення **дотичних**.

! **Дотична** – це пряма, проведена через базову точку (рис. 1.3.7б). Довжина й напрям дотичної визначають розмір і глибину (величину вигину) сегмента кривої. Дотична є ніби посередником між покажчиком миші й лінією, що буде намальована.

Щоб намалювати сегмент кривої, необхідно в той момент, коли обрана друга базова точка, не відпускаючи кнопку, перемістити покажчик у напрямі, протилежному необхідному напрямку вигину; при переміщенні покажчика на екрані з'явиться дотична, розмір і напрям якої визначають глибину й кут вигину (рис. 1.3.7б).

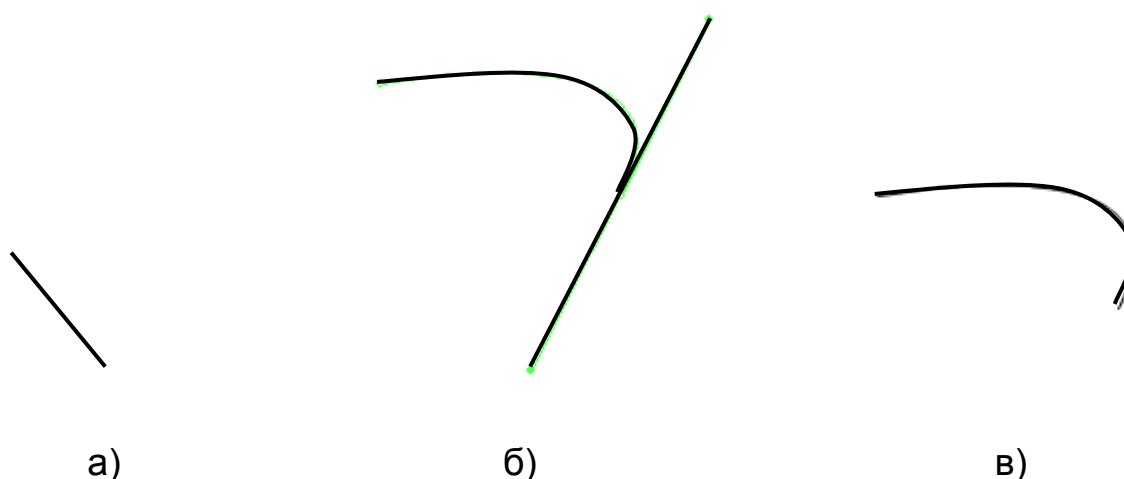


Рис. 1.3.7. Малювання кривих за допомогою пера

Лінія вважається "незавершеною" до натискання клавіші <Esc> або перемикання на інший інструмент. Після завершення малювання лінії вона може бути змінена за допомогою кожного із чотирьох інструментів: *Pen*, *Arrow*, *Free Transform* або *Subselect*.

Для інструмента *Pen* також можна встановити товщину й колір лінії, а також колір заливки. Установка товщини лінії виконується за допомогою інспектора властивостей.

1.3.2. Робота з кольором

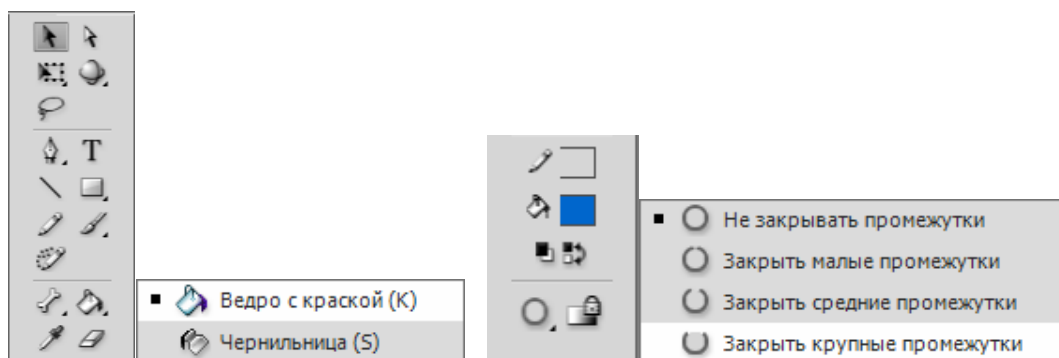
Цей параграф містить відомості про панель інструментів роботи з кольором, інструмент *Paint Bucket Tool*, інструмент *Fill Transform*.

У Adobe Flash передбачена можливість застосування, створення й модифікації колірної палітри фільму й окремих його елементів. Можна використовувати палітру, встановлену за замовчуванням або створену самостійно.

Для зафарбовування ліній і контурів можуть використовуватися тільки базові кольори. Для заливки може бути використаний однотонний колір, градієнт або растрове зображення. Flash підтримує застосування прозорого (того, що не можна побачити) кольору для зафарбовування ліній і заливки. Це дозволяє створювати, наприклад, тільки контур об'єкта без заливки, або, навпаки, зафарбований об'єкт без контуру. Flash також дозволяє ефективно управляти колірною схемою текстових символів.

У Flash для роботи з кольором використовуються:

- інструменти *Ink Bottle Tool* (пляшка чорнил), *Paint Bucket Tool* (банка фарби), *Fill Transform* (Трансформація заливки) і *Dropper Tool* (піпетка), розташовані на панелі *Tools* (рис. 1.3.8а);
- кнопки *Stroke Color* (колір контуру), *Fill Color* (колір заливки), *Default Colors* (колір за замовчуванням), *No Color* (безбарвний) і *Swap Colors* (поміняти кольори), розташовані в полі *Colors* панелі *Tools* (рис. 1.3.8б);
- панелі *Color Mixer* (змішувач кольору) і *Color Swatches* (зразки кольору) (рис. 1.3.8в).



а)

Рис. 1.3.8. Інструменти для роботи з кольором 1

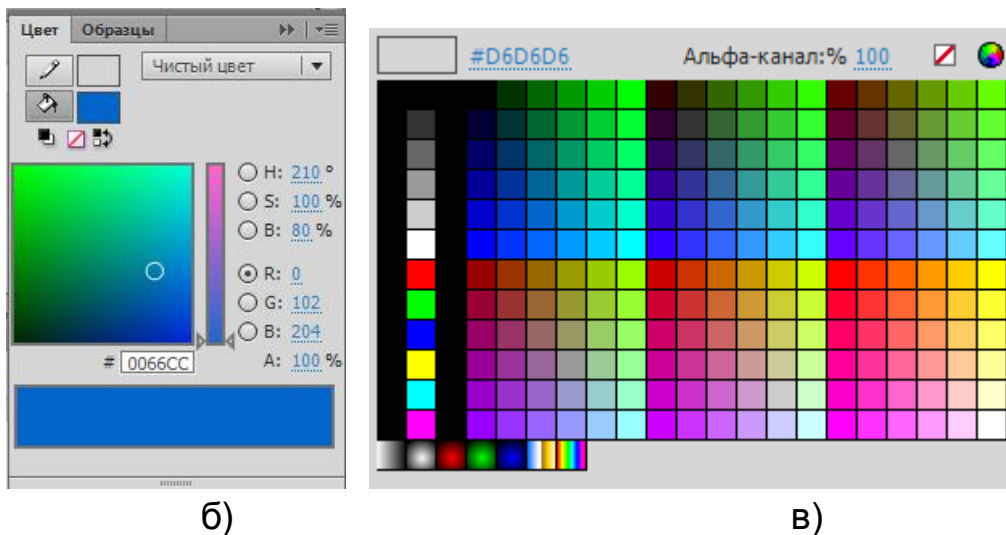


Рис. 1.3.8. Інструменти для роботи з кольором 2

Інструмент Ink Bottle Tool.

Призначений для зміни кольору лінії або контуру об'єкта; можуть використовуватися тільки базові кольори. За допомогою інструмента можна змінювати товщину й стиль лінії (контуру).

Інструмент Paint Bucket Tool.

Призначений для зміни кольору заливки об'єкта або "мазка" пензля, а також для зафарбовування довільної замкнутої області на столі; для зафарбовування можуть використовуватися базові кольори, градієнти й растрові зображення. Інструмент може також застосовуватися для зафарбовування й незамкнутих областей (потрібно дозволити Flash автоматично закривати проміжки між лініями, що обмежують область).

Інструмент Fill Transform.

Дозволяє змінювати параметри градієнтної або растрової заливки певної області. Щоб перейти в режим редагування заливки, необхідно включити інструмент *Fill Transform* і вибрати заливку, що редагується.

1.3.3. Робота з текстом

У цьому параграфі йдеться про створення і редагування тексту; основні його властивості, статичний текст.

У Flash передбачені широкі можливості щодо роботи з текстом. Як і у звичайному текстовому редакторі можна встановлювати розмір, шрифт, стиль, інтервал, колір і спосіб вирівнювання. Як і інші об'єкти, шрифт можна трансформувати – повертати, масштабувати, нахилити. При

цьому зберігається можливість редагування символів. На основі фрагмента тексту може бути створене гіперпосилання.

Flash-фільм може містити статичні, динамічні текстові поля, багаторядкові текстові області, що можна прокручувати, а також поля, що редагуються користувачем (за призначенням аналогічні текстовим полям, що розташовуються у формі на веб-сторінці).

Створення й редагування тексту

У правому куті рамки, що виділяє (верхньому або нижньому), є маркер, який визначає тип текстового поля (рис. 1.3.9):



Рис. 1.3.9. Маркери текстового блоку

- текстове поле фіксованої ширини *Fixed text block* (прямокутний маркер у правому верхньому куті) – у такому полі виконується автоматичний перенос на наступний рядок, якщо наступний символ не вміщається на поточному рядку; після публікації фільму зміст такого поля не може бути змінено за допомогою сценарію або користувачем веб-сторінки;

- текстове поле, що розширюється *Expanded text block* (круглий маркер у правому верхньому куті) – однорядкове текстове поле, ширина якого автоматично збільшується при введенні тексту; таке поле також не може бути змінено після публікації;

- текстове поле з динамічно змінюваним змістом *Dynamic text block* або *Input text block* (білий прямокутний маркер у правому нижньому куті) – зміст поля першого типу може змінюватися за допомогою сценарію, а другого – читачем веб-сторінки (наприклад, при введенні даних у форму);

- текстове поле, з можливістю прокручування змісту у текстовій області за допомогою вертикальної й/або горизонтальної смуги прокручування (чорний прямокутний маркер у правому нижньому куті).

! **Статичний текст** (*Static Text*) – це текстовий зміст фільму, що не може бути змінений після публікації фільму.

Формат інспектора властивостей статичного тексту подано на рис. 1.3.10.

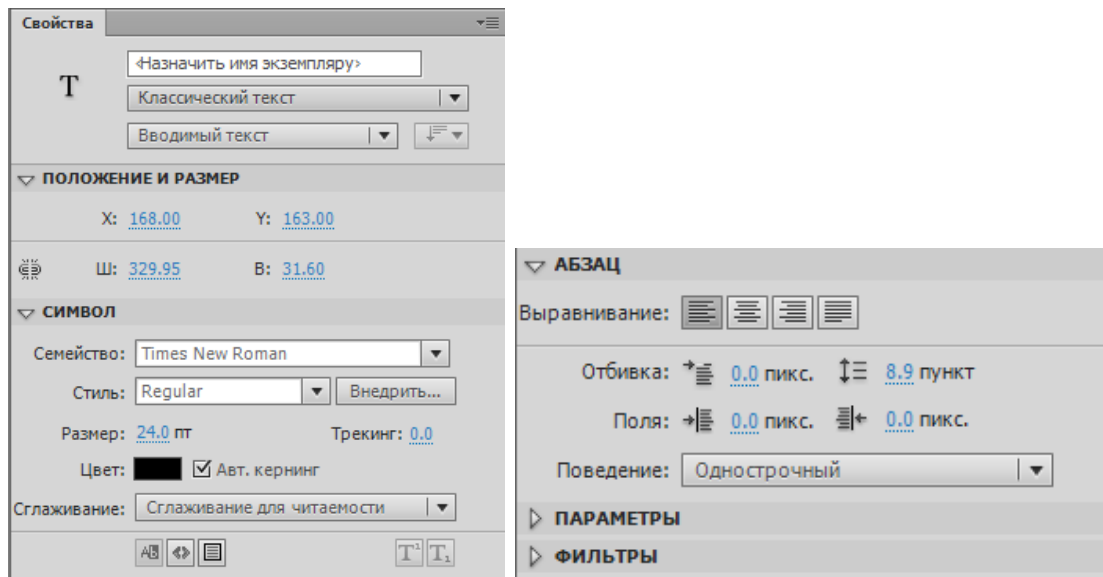


Рис. 1.3.10. Інспектор властивостей статичного тексту

Варто окремо розглянути деякі інструменти панелі елементів управління:

- текстове поле *URL* (Universal Resources Locator) – дозволяє доповнити гіперпосилання;
- кнопка *Selectable* (той, що можна виділити) – дозволяє встановити або скасувати для тексту властивість – можливість виділення фрагмента;
- кнопка *Change Direction of Text* (змінити напрям тексту) дозволяє відкрити меню, за допомогою якого можна вибрати орієнтацію напису: горизонтальну або вертикальну.

Використовуючи кнопку *Format* (формат), можна налаштовувати форматування тексту, включаючи такі параметри, як величину абзацного відступу, міжрядковий інтервал, ліве та праве поля.

Діалогове вікно *Format Option* ідентичне для всіх типів текстових полів.

Контрольні запитання

1. Які інструменти малювання Adobe Flash вам відомі, у чому їхня основна відмінність?
2. Опишіть процедуру побудови кривих за допомогою переміщення дотичних.

3. Опишіть інструменти роботи з кольором.
4. Які типи текстових полів існують у Adobe Flash?
5. Опишіть можливості Adobe Flash щодо настроювання текстових полів.

1.4. Методи створення анімації

Мета: вивчення теоретичних основ покадрової анімації та аналіз розробки анімації у середовищі Adobe ImageReady та Adobe Flash, вивчення поняття ключових кадрів та ознайомлення з методами створення анімації, що засновуються на розкадруванні руху та розкадруванні форми.

Ключові слова: покадрова анімація, Gif-анімація, анімації у Adobe ImageReady, ключові кадри, векторний морфінг, розкадрування руху, розкадрування форми.

1.4.1. Покадрова анімація

У цьому параграфі йдеться про: принцип покадрової анімації, Gif-анімацію, створення анімації за допомогою інструментарію Adobe ImageReady, створення покадрової анімації за допомогою Adobe Flash.

! **Покадрова анімація** – це набір зображень різних фаз руху – **кадрів**, які прокручуються з великою швидкістю.

Переваги покадрової анімації:

- відносна наявність створення;
- широкі можливості для творчості.

Недоліки покадрової анімації:

- велика працеемність створення фільмів;
- проблеми, що виникають при збереженні покадрової анімації у цифровому вигляді.

Gif-анімація. Принцип створення Gif-анімації полягає в послідовній зміні Gif-зображень. При цьому основними параметрами такого зображення є:

- кількість зображень (складових частин);
- час затримки між зміною зображень.

Одним із відомих засобів для створення Gif-анімації є **Adobe ImageReady**. Але цей програмний засіб дозволяє створювати лише вузькоспеціалізовані документи.

Створення покадрової анімації за допомогою Adobe Flash. У покадровій анімації зміст сцени змінюється в кожному кадрі, без неї неможливо обійтися при створенні анімованого персонажу. Для цього необхідно для кожної зміни створювати ключовий кадр. У ході роботи над кліпом можна використовувати інструменти малювання, працювати з буфером обміну, імпортувати графіку. Корисно періодично переглядати результати роботи.

1.4.2. Ключові кадри

У цьому параграфі йдеться про ключові й звичайні кадри.

! ***Анімація** створюється при зміні характеристик об'єкта в послідовності кадрів: його положення, розмірів, прозорості й інших властивостей, ці зміни можуть відбуватися одночасно.*

У Adobe Flash крім покрокової (frame-by-frame) анімації існує анімація шляхом розрахунку проміжних кадрів (tweened). У другому випадку розмір файлу істотно менше, тому що зберігаються тільки значення ключових кадрів, у яких задаються параметри об'єктів [24].

! ***Ключовим кадром (Keyframe)** називається кадр, у якому задані зміни анімаційної сцени. Для покрокової анімації (frame-by-frame) ключовим є кожний кадр. В анімації з розрахунком проміжних кадрів (tweened) ключові кадри визначають стан об'єктів у певні моменти, всі кадри між ними автоматично розраховуються Flash.*

На осі часу ключовий кадр позначений кружком, якщо в ньому присутні об'єкти, якщо він порожній – вертикальною рисою.

Adobe Flash може розраховувати проміжні кадри для анімації двох типів:

анімація руху – задаються положення, розмір і поворот об'єкта, групи або текстового блоку в один момент часу, а потім встановлюються їхні нові значення в інший момент; анімація руху поділяється на Classic Tween та Motion Tween;

анімація форми, при якій первісне зображення об'єкта перетворюється в нове за заданий проміжок часу.

Кожний кадр фільму Flash може містити будь-яку кількість об'єктів, що розміщуються за різними шарами. В анімації за допомогою шарів

можна організувати об'єкти так, щоб вони не заважали один одному. При одночасній анімації декількох груп або символів вони повинні бути розміщені на різних шарах. Зазвичай фоновий шар містить статичну сцену, а кожний наступний шар – по одному анімованому об'єкту. У вікні **Timeline** (вісь часу) вони відображаються у вигляді окремих рядків. Одночасно редагується вміст тільки одного шару.

При створенні в анімації фонового зображення потрібно, щоб воно повторювалося протягом декількох кадрів. При додаванні нових кадрів у послідовність у кожному з них повторюється вміст кадру, після якого вони вставлені.

1.4.3. Створення анімації за допомогою розкадрування

У цьому параграфі містяться відомості про: розкадрування руху – принципи створення й особливості, створення анімації на основі розкадрування форми.

! *Анімація руху (motion tween та classic tween) дозволяє змінювати положення об'єкта рис. 1.4.1 [24], а також розмір, поворот, нахил, колір, застосовуючи градієнтну зміну кольорів або змінюючи їхню прозорість. Анімацію руху можна застосовувати до екземплярів, груп або текстових блоків. Щоб виконати такі зміни для групи або текстового блоку, їх попередньо варто **перетворити в символи**.*

Класична анімація – це старий спосіб створення анімації в Flash Professional. Ці анімації схожі на нові анімації руху, проте іноді їх складніше створювати і вони менш гнучкі. Однак класична анімація надає деякі можливості управління, недоступні для анімації руху.



Рис. 1.4.1. Результат розрахунку проміжних кадрів анімації [24]

Зміни в класичній анімації описані в ключових кадрах. Ключові кадри задаються в основних моментах анімації, а Flash Professional створює вміст кадрів між ключовими кадрами. Проміжні кадри такої анімації виділяються світло-блакитним або світло-зеленим зі стрілкою, намальованою між кадрами. Ключові кадри позначаються на часовій шкалі таким чином: суцільне коло є ключовим кадром зі вмістом, порожнє коло перед кадром становить порожній ключовий кадр. Наступні кадри, додані в один шар, мають той же вміст, що і ключовий кадр. У класичній анімації можна редагувати тільки ключові кадри. Tween-кадри можна переглянути, але їх неможливо редагувати безпосередньо. Щоб відредагувати tween-кадри, слід змінити один із визначальних ключових кадрів або додати новий ключовий кадр між початковим і завершальним.

Створити анімацію руху можна двома способами:

При кожній зміні положення ключових кадрів або зміні їхнього вмісту Flash автоматично розраховує проміжні кадри.

Анімація руху (не класична) є дуже потужним інструментом створення анімації за допомогою розкадрування. Дозволяє дуже швидко створювати навіть складні анімаційні перетворення, використовуючи послідовно декілька різних властивостей для зміни об'єкта, а також відразу задіювати траєкторію руху, яку можна настроювати як звичайну криву Безьє.

Діапазон анімації – це група кадрів на шкалі часу, в якій одна або декілька властивостей об'єкта змінюються з часом. Ці діапазони анімації можна вибрати для одного об'єкта, перетягнути з одного місця на часовій шкалі в інше і навіть в інший шар. У кожному діапазоні анімації змінюватися буде тільки один об'єкт у робочій області. Цей об'єкт називається цільовим об'єктом діапазону анімації.

Ключовий кадр властивості – це кадр у діапазоні анімації руху, де явно визначені одне або кілька значень властивостей для цільового анімаційного об'єкта. Ці властивості можуть включати положення, альфа (прозорість), колір і т. д. Для кожної властивості, визначеної користувачем, створюється окремий ключовий кадр властивості. Якщо в одному кадрі задати зміну більше ніж однієї властивості, то ключові кадри для кожної з цих властивостей будуть розміщені в цьому кадрі. Всі властивості діапазону анімації і його ключові кадри властивостей можна переглянути на панелі редактора руху (рис. 1.4.2). З контекстного меню діапазону анімації можна також вибрати, які типи ключових кадрів властивостей відображати на часовій шкалі.

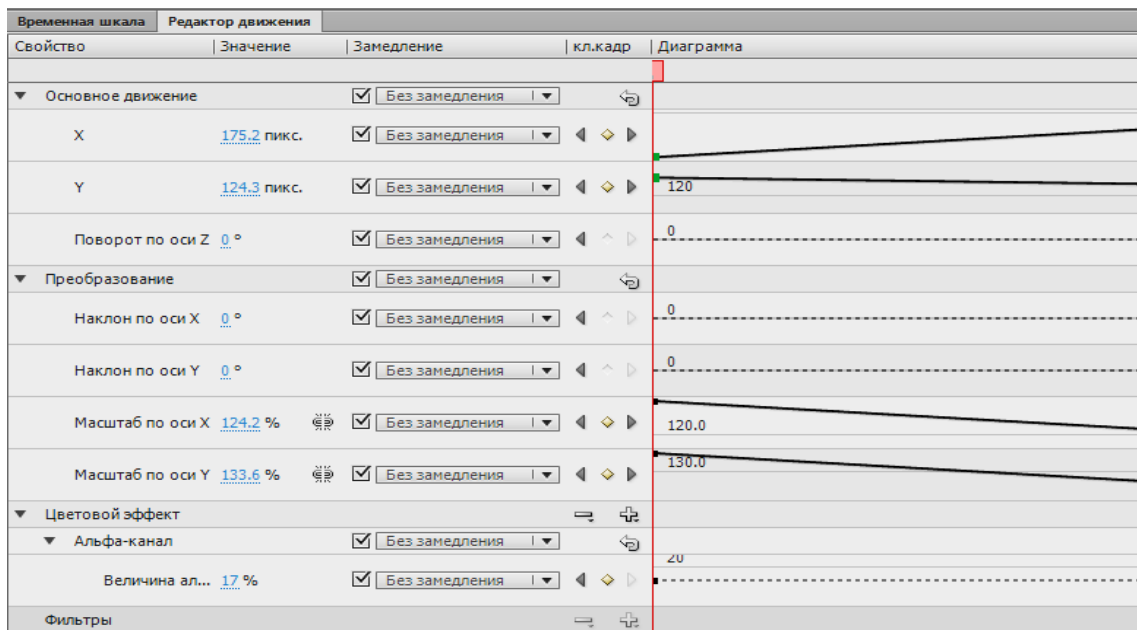


Рис. 1.4.2. Панель "Редактор руху"

Об'єкти і властивості, для яких можлива анімація руху. До об'єктів, що анімуються, входять фрагменти ролика, графічні символи і символи кнопок, а також текстові поля. Ці об'єкти можуть мати такі властивості:

- положення за осями X і Y на площині;
- положення за осями Z у тривимірному просторі (тільки фрагменти роликів);
- обертання на площині (навколо осі Z);
- обертання навколо осей X, Y і Z у тривимірному просторі (тільки фрагменти роликів).
- нахил за осями X і Y;
- масштаб за осями X і Y;
- кольорові ефекти, прозорість.

Для створення тривимірного руху в параметрах публікації потрібно вказати, що FLA-файл призначений для Action Script 3.0 і Flash Player 10 або пізнішої версії. Adobe AIR також підтримує тривимірний рух.

Кольорові ефекти включають в себе альфа-канал (прозорість), яскравість, тон і розширені параметри для кольорів. Кольорові ефекти можна анімувати тільки для символів і тексту TLF. Анімуючи ці властивості, можна домогтися плавної появи або зміни кольору об'єкта. Щоб створити анімацію руху кольорового ефекту для класичного тексту, слід перетворити текст на символ. Властивості фільтрів – фільтри не можна застосовувати до графічних символів.

Основні відмінності між анімацією руху і класичною анімацією.

Анімація руху складається з одного цільового об'єкта, що використовується у всьому діапазоні анімації. Класична анімація дозволяє створювати анімацію між двома ключовими кадрами, які містять екземпляри однакових або різних символів.

Для ефекту переходу між двома кольорами, наприклад тонування або альфа-прозорості, можна використовувати класичну анімацію. При анімації руху до одного переходу можна застосовувати тільки один кольоровий ефект.

Для анімації тривимірних об'єктів можна використовувати тільки анімацію руху. Не можна анімувати тривимірний об'єкт за допомогою класичної анімації руху.


Один шар може містити кілька екземплярів анімації, але тільки одного типу (або класичної анімації, або анімації руху).

Застосування наборів налаштувань руху.

Набори налаштувань руху – це попередньо сконфігуровані анімації руху, які можна застосовувати до об'єкта в робочій області. Необхідно просто вибрати об'єкт і натиснути кнопку **Застосувати** на панелі **Набори налаштувань руху**.

Застосування наборів налаштувань руху – це найбільш швидкий спосіб ознайомлення з основами анімації в Flash. Побачивши, як працюють набори налаштувань, стане легше створювати власну анімацію. Можна створювати та зберігати власні користувальницькі набори налаштувань. Набори налаштувань руху можуть містити тільки анімацію руху. Класичну анімацію не можна зберегти в якості набору налаштувань руху.

Анімація форми

 **Анімація форми** (*shape-tweened*) – це метод розрахунку проміжних кадрів, що дозволяє перетворювати одну форму в іншу, змінюючи при цьому її розмір, положення й колір. При створенні анімації форми варто врахувати, що Flash не може виконати таку анімацію для груп, символів, текстових блоків і растрових зображень.

Анімація форми фігур дозволяє реалізовувати перетворення однієї форми в іншу – створювати ефект метаморфози (рис. 1.4.3). Для керування цими змінами можна додатково застосовувати спеціальні ідентифікатори форми (*shape hints*), які дозволяють створювати складні зміни й перетворення окремих частин вихідної фігури в нові. Ідентифіка-

тори форми позначають окремі крапки фігури до й після зміни її форми (рис. 1.4.4) [24]. Можна задати до 26 ідентифікаторів. У такий спосіб можна створити зміну виразу намальованих облич, перетворення одних тварин в інших тощо. Для фігур складної форми бажано задати проміжні стани у вигляді додаткових ключових кадрів, що дозволить контролювати фази перетворення. Можна виконувати одночасну анімацію декількох фігур в одному шарі, але кращий результат можна одержати, поділяючи фігури за шарами.

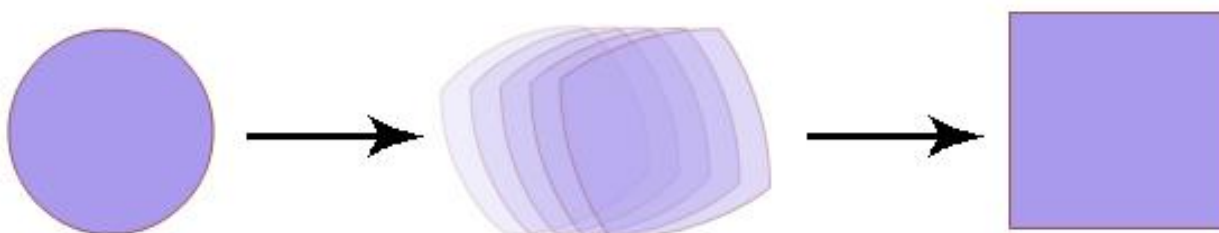


Рис. 1.4.3. Ефект метаморфози щодо перетворення форми

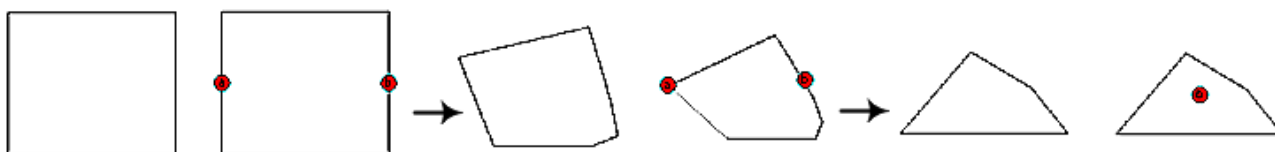


Рис. 1.4.4. Перетворення форми – автоматичне й з використанням ідентифікаторів [24]

Для застосування анімації форми (shape-tween) до груп, символів, текстових блоків і растрових зображень, їх попередньо варто перетворити командою **Modify > Break Apart** (Змінити > Розбити на частині). За цією командою розривається зв'язок між екземпляром і символом, перетворюючи екземпляр у набір непов'язаних ліній і фігур, що дозволяє змінювати його, не роблячи впливу на інші екземпляри. Зміни символу надалі також не будуть впливати на даний екземпляр.

Використання ідентифікаторів форми. Ідентифікатори форми (shape hints), позначені буквою латинського алфавіту, відображаються жовтим кольором у початковому й зеленим кольором у кінцевому ключовому кадрі. Червоний колір ідентифікатора означає, що він розташований не на кривій і не може брати участь у трансформації.

Анімація зворотної кінематики

❗ **Зворотня кінематика (ЗК)** – це спосіб анімації об'єктів за допомогою кісток, об'єднаних у лінійні або гіллясті каркаси з відносинами "батько – нащадок". У ході руху однієї кістки з'єднані з нею кістки переміщуються щодо неї. Зворотня кінематика дозволяє легко створювати природні рухи. Для анімації з використанням зворотної кінематики досить указати початкове і кінцеве положення кісток на часовій шкалі. Flash автоматично інтерполює положення кісток каркаса між початковим і кінцевим кадрами. [1]

Зворотню кінематику можна використовувати двома способами:

1) використовуючи фігуру як контейнер для декількох кісток. Наприклад, можна додати кістки до малюнка змії, щоб вона переміщалася реалістично. Ці фігури можна намалювати в режимі малювання об'єктів;

2) об'єднуючи в ланцюжок екземпляри символу. Наприклад, можна пов'язати фрагменти роликів, що відображають руку, передпліччя і долоню, щоб вони реалістично переміщалися відносно один одного. Кожен екземпляр має тільки одну кістку.

Фігура з каркасом кісток зворотної анімації наведена на рис. 1.4.5а. Головна частина кожної кістки закруглена, а хвостова частина загострена. Перша додана кістка (коренева кістка) має округлість у головній частині. Група з декількох символів із каркасом кісток зображена на рис. 1.4.5б. Плечі і стегна фігури є точками розгалуження каркаса. Точками перетворення за замовчуванням є головна частина кореневої кістки, внутрішні з'єднання і хвостова частина останньої кістки гілки.

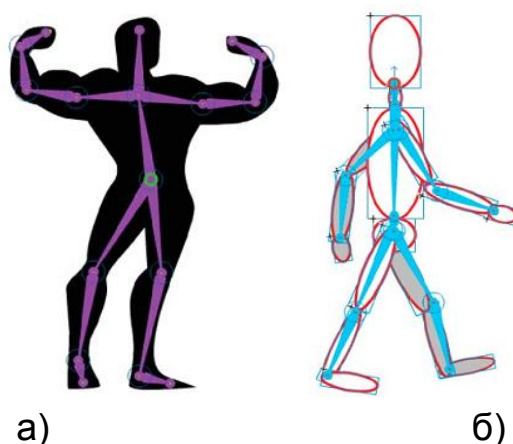


Рис. 1.4.5. Зворотня кінематика у вигляді фігури для декількох кісток (а) і у вигляді об'єднаних у ланцюжок екземплярів символів (б)

Стилі кістки. Існує 4 способи замальовки кісток у робочій області Flash:

- суцільний – це стиль за замовчуванням;
- каркасний, який зручно використовувати, коли суцільний стиль закриває занадто великий відсоток зображення під кісткою;
- лінія – зручно використовувати для невеликих каркасів;
- не задано – коли кістки ховаються, щоб було видно тільки те зображення, що знаходиться під ними.

Шари пози. При додаванні кісток до екземплярів символів або фігур Flash створює для них новий шар на часовій шкалі. Цей новий шар називається шаром пози.

Додавання кісток до символів. Кістки ЗК можна додавати до фрагментів ролика, графічних об'єктів і екземплярів кнопок. Для використання тексту спочатку його потрібно перетворити в символ. Примірники символів перед додаванням кісток можуть розташовуватися в різних шарах. Flash додає їх в шар пози.

Примітка. Також можна розбити текст ("Модифікація" > "Розділити") на окремі фігури і додати до них кістки. При створенні ланцюжка об'єктів слід урахувати їх взаємини, наприклад, можна створити ланцюжок від плеча до ліктя, а потім до зап'ястя. **Додавання кісток до фігур.** Кістки додаються до окремих фігур або до групи фігур, розташованих у тому ж шарі. У будь-якому випадку необхідно вибрати всі фігури перед додаванням першої кістки. Після додавання кісток Flash перетворює всі фігури і кістки в об'єкт фігури зворотної кінематики і переміщує об'єкт у новий шар пози.

Після додавання кісток до фігури, до неї застосовуються такі обмеження:

- не можна об'єднувати фігуру зворотної кінематики з іншими фігурами, що не входять до неї;
- не можна повертати, масштабувати і нахилити фігуру за допомогою інструмента "Вільне перетворення";
- редагувати контрольні точки фігури не рекомендується.

Обмеження руху кісток зворотної кінематики

Щоб створити більш реалістичний рух ЗК-каркасів, можна управляти свободою руху окремих кісток. Наприклад, дві кістки руки можна обмежити таким чином, щоб лікоть не міг згинатися в неправильному напрямі. Можна також обмежити швидкість руху кістки, щоб створити ефект наявності у кістки ваги.

Додавання пружності кісткам

Дві властивості кісток можна використовувати для додавання пружності кісткам ЗК. Властивості "Інтенсивність" і "Демпфірування" кісток забезпечують дійсне фізичне переміщення кісток ЗК шляхом інтеграції динамічної фізики в систему кісток ЗК. Ці властивості полегшують процес створення анімації з використанням елементів фізики. Властивості "Інтенсивність" і "Демпфірування" забезпечують правдоподібне переміщення в анімації, а також легкі в налаштуванні. Ці властивості краще задавати перед додаванням поз у шар пози.

Інтенсивність. Твердість пружини. Чим більше значення, тим більший ефект жорсткості пружини.

Демпфірування – це ступінь "загасання" коливань пружини. Чим більше значення, тим швидше припиняться коливальні рухи пружини. При значенні 0 пружність пружини залишатиметься незмінною протягом кадрів шару пози.

Далі перераховані фактори, що впливають на остаточний вигляд анімації кісток при роботі з властивостями:

- значення властивості "Пружина";
- значення властивості "Інтенсивність";
- значення властивості "Демпфірування";
- число кадрів між позами в шарі пози;
- загальне число кадрів у шарі пози;
- число кадрів між кінцевою позою і останнім кадром наступної пози.

Анімація каркаса.

У Flash анімація каркасів ЗК проводиться не таким способом, як анімація інших об'єктів. У випадку з каркасами до шару пози просто додаються додаткові кадри, а для створення ключового кадру положення каркаса змінюється в робочій області. Ключові кадри в шарі пози називаються позами. Оскільки каркаси ЗК зазвичай використовуються для створення анімації, всі шари пози автоматично виконують функції шарів анімації.

Однак шари пози ЗК відрізняються від шарів анімації, оскільки в шарі пози можна анімувати тільки властивості положення кісток. Для анімації інших властивостей об'єкта ЗК, таких, як розташування, перетворення, колірні ефекти або фільтри, слід перенести каркас і пов'язані з ним об'єкти на фрагмент ролика або графічний символ.

Контрольні запитання

1. У чому полягають основні принципи створення покадрової анімації?
2. У яких програмних продуктах можна створювати покадрову анімацію?
3. Що таке ключові кадри? Чим вони відрізняються від звичайних?
4. У чому полягають принципи створення й особливості анімації руху?
5. Що таке розкадрування форми? У чому його особливості?
6. У чому полягають принципи зворотної кінематики?
7. Дайте визначення кісток та каркасів.

Практична складова "Створення покадрової анімації та анімації за допомогою розкадрування форми"

Компетентності, що формуються завдяки цій складовій частині:

- практичні навички створення покадрової анімації;
- створення анімації за допомогою розкадрування форми;
- навички використання анімації розкадрування форми в анімаційному проекті спільно з іншими видами анімації.

Допоміжний матеріал

Створення покадрової анімації за допомогою Adobe Flash

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. 1.4.6.

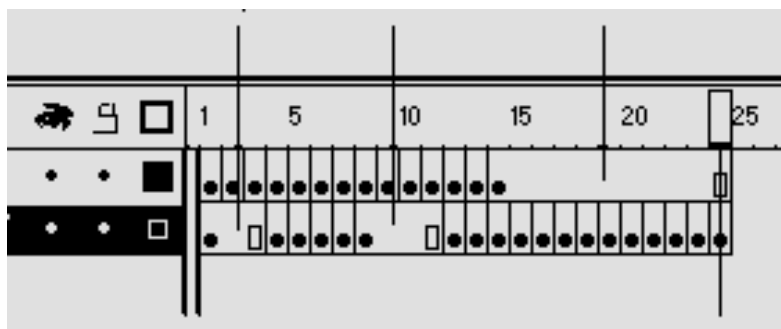


Рис. 1.4.6. Покадрова анімація

Переваги покадрової анімації полягають у найбільшому контролі за анімацією. 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, та для її створення потрібні великі трудозатрати.

Створення покадрової анімації за допомогою інструментарію Adobe Flash та графічного редактора

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

1. Намалювати в будь-якому графічному редакторі (наприклад, Adobe Photoshop) або безпосередньо у редакторі Adobe Flash, використовуючи відповідні інструменти, послідовність зображень, які потім відтворювали якусь динамічну подію. Також можна знайти в Інтернеті рисунок для анімації. Наприклад, це можуть бути рисунки, що відображають фази руху якогось із динамічних процесів. На рис. 1.4.7 наведено фази руху коня, що скаче.

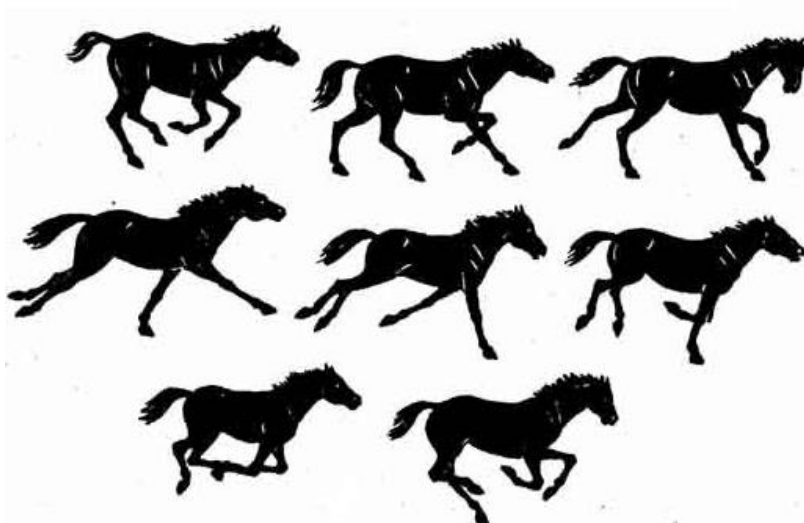


Рис. 1.4.7. Створення рисунка в Adobe Photoshop

2. У графічному редакторі порібно послідовно вирізати кожен окрему фазу руху та копіювати її у буфер обміну.

3. У Adobe Flash потрібно вставити отримане зображення з буфера обміну на поточний кадр. Поточний кадр автоматично стане ключовим. Після чого другому кадру потрібно виконати команду "Вставити пустий ключевой кадр".

4. Після цього необхідно повернутися до графічного редактора та повторити пункт 2, а потім пункт 3, і так доки всі фази руху не будуть розміщені на ключових кадрах. Далі необхідно тестувати розроблену покадрову анімацію, яка покаже у динаміці коня, що скаче.

Створення анімації форми за допомогою інструментарію Adobe Flash

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 такою 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 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 графічний матеріал із папки, вказаної викладачем, або зображення, отримані завдяки мережі Інтернет, створити покадрову анімацію.

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

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

4. Створити анімацію форми для складних об'єктів із використанням ідентифікаторів форми.

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

6. Додати в сценарій отримані результати.

Практична складова "Створення анімації за допомогою розкадровування руху (Classic Tween та Motion Tween). Створення анімації на основі зворотної кінематики"

Компетентності, що отримуються завдяки цій складовій частині:

- вміння створювати анімацію на основі розкадровування руху класичної анімації;
- створення анімації з використанням розкадровування руху motion tween;
- вміння використовувати зворотну кінематику для створення анімації.

Допоміжний матеріал

Розкадровування руху (класична анімація) 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 (класична анімація), достатньо скористатися такою процедурою:

1. Створити рисунок довільної форми. Виділити його, перетворити в символ типу "Графіка" (F8, "Графіка").
2. Задати початкове положення символу в першому кадрі.
3. Виділити останній кадр анімації (наприклад 25), та зробити його ключовим (рис. 1.4.8). Задати кінцеве положення символу в останньому кадрі (перемістити символ у необхідне місце).

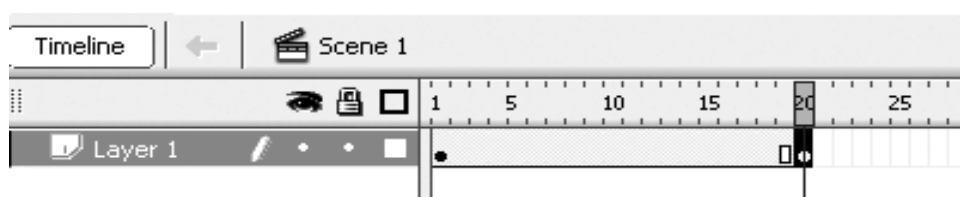


Рис. 1.4.8. Налаштування часової шкали майбутнього ролика

4. 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 світло-синя стрілочка (рис. 1.4.9). Якщо ж анімація руху зроблена неправильно, то з'явиться синій пунктир.

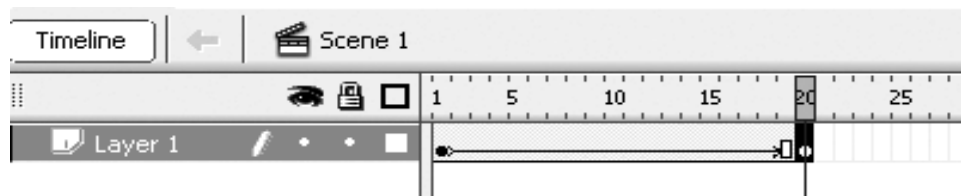


Рис. 1.4.9. Розкадровування руху

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

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

7. Перейти на перший ключовий кадр і переглянути властивості анімації. Спробувати проекспериментувати з обертанням символу, прискоренням/плавним зупиненням анімації, збільшенням розміру символу, прозорістю та іншими властивостями.

Розкадровування руху (Motion tween) дозволяє ефективно створювати анімацію з великим набором функціональних можливостей. Автор отримує потужний інструмент – використання ключових кадрів властивостей, а також можливість використання траєкторії для об'єктів, що анімуються.

Слід розглянути покроково створення анімації:

1. Для цього виду анімації теж необхідно використовувати лише символи. Тому варто створити рисунок довільної форми, виділити його, і перетворити в символ типу "Графіка" (F8, "Графика").

2. Виділити перший кадр та з контекстного меню обрати команду "Создать анимацию движения". Після першого ключового кадру буде створено 25 кадрів зеленого кольору без позначень.

3. Обрати будь-який кадр із зеленого діапазону та змінити положення символу. Слід звернути увагу на шкалу часу – кадр, на якому відбулась зміна, відмічено маленьким замальованим ромбом – це свідчить про те, що це є ключовий кадр властивостей.

4. Варто спробувати протестувати анімаційний кліп. Також слід звернути увагу на траєкторію, яку додано до анімації. Цю траєкторію можна редагувати як звичну криву Без'є.

5. Переключитися на вкладку "Редактор движения". На цій вкладці подані всі параметри анімації, які можна редагувати крім положення символа (обертання, масштабування, прискорення, різноманітні кольорові ефекти та інше).

6. Внести зміни в анімацію на ще одному з кадрів. Цей кадр теж буде позначено маленьким ромбом та він стане ключовим кадром властивостей.

7. Спробувати використати стандартні набори налаштування анімації руху – обрати інструмент "Набори настроек движения" на вертикальній панелі інструментів.

8. Спробувати створити свою, більш складну анімацію руху, та зберегти налаштування анімації руху у бібліотеку (рис. 1.4.10).

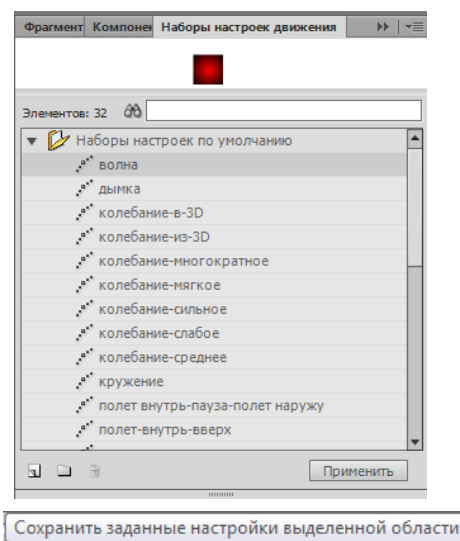


Рис. 1.4.10. Набори налаштувань руху

Зворотня кінематика. Дозволяє за допомогою інструмента "Кость" створювати гіллясті структури та ефективно реалізовувати анімацію з природними рухами.

Для випробування такої анімації слід виконати такі кроки:

1. Необхідно створити або імпортувати векторний малюнок, який би мав окремі частини, які повинні переміщуватися у складі суцільного каркасу. Слід розглянути приклад у вигляді скелета.

2. Окремі групи кісток об'єднати у символи.

3. Використовуючи інструмент "Кость" з'єднати окремі символи в логічні послідовності (рис. 1.4.11).

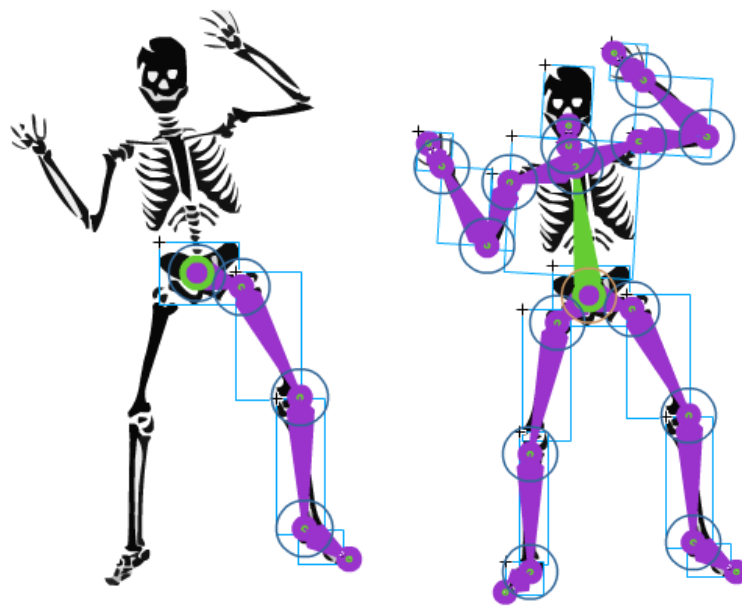


Рис. 1.4.11. Арматура 3К

4. Слід звернути увагу, що всі кістки автоматично перенесено на окремий шар – арматура. Перший кадр цього шару помічено маленьким ромбом, він має властивості 3К-арматури.

5. Виділити, наприклад, десятий кадр, та виконати команду "Добавить позу". Цей кадр також отримає властивості 3К-арматури. Спробувати змінити положення кісток у інтегрованому каркасі-скелеті. Додати ще одну позу та внести деякі зміни до положення кісток.

6. Протестувати отриману анімацію. Спробувати проекспериментувати з параметрами "Сила" та "Пружина".

Завдання

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

2. Використовуючи малюнки, які були отримані на попередньому занятті або знайдені в мережі Інтернет, створити анімації 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. Використовуючи графічний матеріал із мережі Інтернет або намальований у середовищі Flash, створити анімацію з використанням зворотної кінематики.

1.5. Шари. Створення й редагування символів

Мета: ознайомлення зі складною структурою кліпу, релізованою завдяки використанню шарів, використанню символів та створення бібліотек і користування їх ресурсами.

Ключові слова: шар, напрямний шлях, маска, символ, бібліотека.

1.5.1. Використання шарів в анімації

У цьому параграфі наведено інформацію про використання шарів в анімації, типи шарів, властивості шарів, створення анімації за напрямним шляхом, створення анімації маски.

! **Шар** (Layer) – це невелика частина сцени фільму, здатна містити будь-яке число графічних об'єктів.

Переваги використання шарів проявляються, насамперед, при створенні багатопланових сцен, оскільки об'єкт на більш високо розташованому шарі в ролику завжди буде перекривати інші об'єкти (рис. 1.5.1). Для створення нового шару достатньо вибрати пункт меню **Создать слой**.

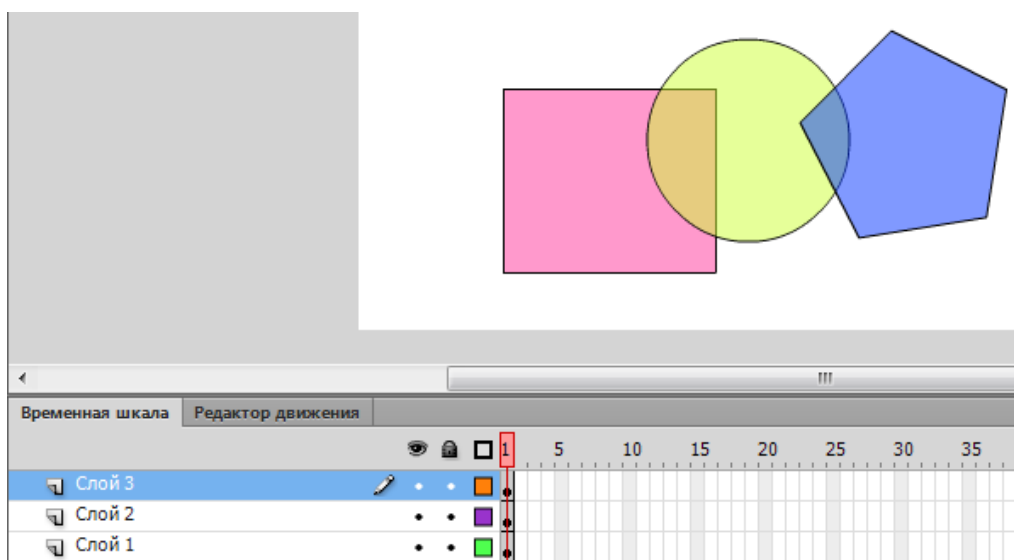


Рис. 1.5.1. Розташування об'єктів на різних шарах

Завжди можна змінити порядок розташування шарів, а також редагувати об'єкти одного шару незалежно від елементів інших шарів. Однак за необхідності можна вибрати одночасно об'єкти з різних шарів і працювати з ними як з єдиним цілим. Наприклад, можна змінити їхній колір або згрупувати.

Типи шарів. Flash дозволяє використовувати п'ять типів шарів:

1. Звичайні шари – міститься велика частина графіки в FLA – файлі.

2. Маскуючі шари, що містять об'єкти, які використовуються в якості масок для приховування окремих областей нижніх шарів; шари, що маскуються – лежать під маскуючим шаром і асоціюються з ним. Видно тільки ту частину шару, яка не вкрита маскою.

3. Направляючі шари, що містять траєкторії, які можна використовувати для правильного розташування об'єктів на інших шарах чи для задання руху при класичній анімації на інших шарах; шари, що направляються, де об'єкти можуть розташовуватися або анімуватися за траєкторією, що міститься на направляючому шарі. Такі шари можуть містити класичну анімацію, але не анімацію руху.

4. Шари анімації руху містять символи, для яких виконується відповідна анімація та ключові кадри властивостей.

5. Шари каркасів містять об'єкти, до яких прикріплені кістки зворотної кінематики.

! *Маска* (шар, що маскує; *Mask layer*) є звичайним шаром, за винятком того, що будь-яка заливка на ньому інтерпретується Flash як отвір, через який видно нижчий шар. При цьому колір заливки (у тому числі растрової), наявність градієнта, контур заливки і його тип повністю ігноруються Flash.

! *Шар направляючих* (*Guide layer*) можна використовувати для створення траєкторій, за яким будуть рухатися об'єкти, що анімуються.

Лінії, створені на такому шарі, не помітні в swf-файлі. Створити шар направляючих можна за допомогою команди **"Добавить направляющую классической анимации"**, а саму траєкторію інструментом **"Карандаш"** на цьому ж шарі.

Створення й видалення шарів. Створити шар можна або за допомогою команди **"Создать слой"**, або клацнувши на відповідну піктограму на часовій шкалі (рис. 1.5.2).

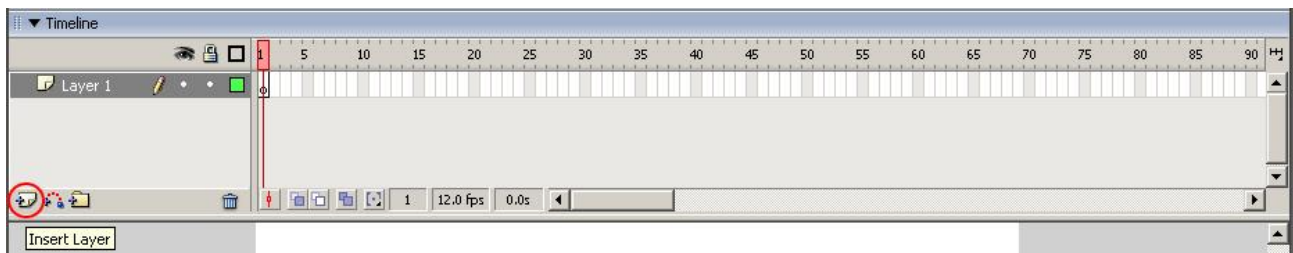


Рис. 1.5.2. Створення нового шару

Видалити шар можна або за допомогою команди **"Удалить слой"** у контекстному меню шару, або клацнувши на відповідну піктограму на часовій шкалі.

Властивості шарів. Для будь-якого шару можна встановити такі властивості: **Name** (ім'я шару); **Show** (зробити шар видимим/невидимим); **Lock** (заблокувати/розблокувати шар для редагування); **Type** (визначити тип шару); **Outline color** (задати колір контурів об'єктів шару); **View layer as outlines** (відображати тільки контури об'єктів шару); **Layer height** (висота шару, %).

Створення анімації за напрямним шляхом. При створенні класичної анімації з рухом об'єкта за заданою траєкторією необхідно спочатку створити класичну анімацію руху та виконати анімацію об'єкта з початкової точки до точки призначення, а потім виконати для цього шару команду **"Добавить направляющую классической анимации"**. На доданому шарі – **"Направляющая"** необхідно намалювати траєкторію для переміщення об'єкта. На останньому кадрі шару **"Направляющая"**, що відповідає останньому кадру класичної анімації, необхідно додати пустий (не ключовий) кадр (рис. 1.5.3).

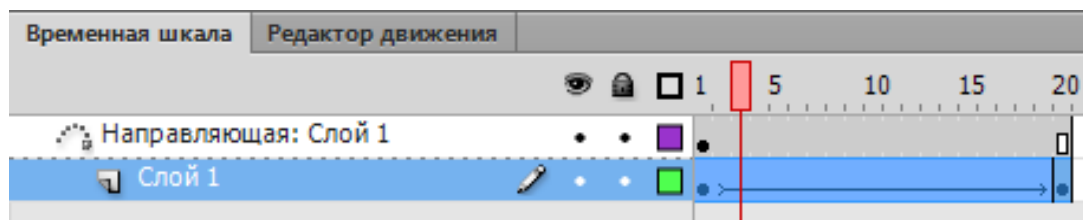


Рис. 1.5.3. Класична анімація за направляючим шляхом

Слід звернути увагу, що центр об'єкта, який анімується, необхідно встановити безпосередньо на траєкторії руху, як на першому, так і на останньому кадрі, а у властивостях анімації потрібно ввімкнути прапо-

рець "Синхр". Також можна проекспериментувати з прапорцем "Ориент по траекторії", це дасть змогу нахилити об'єкт, що анімується, безпосередньо до траекторії.

Створення анімації маски. Щоб створити ефект маски, необхідний шар маски та шар, що маскується. Для маски може бути використаний тільки один об'єкт на шарі маски (рис. 1.5.4).

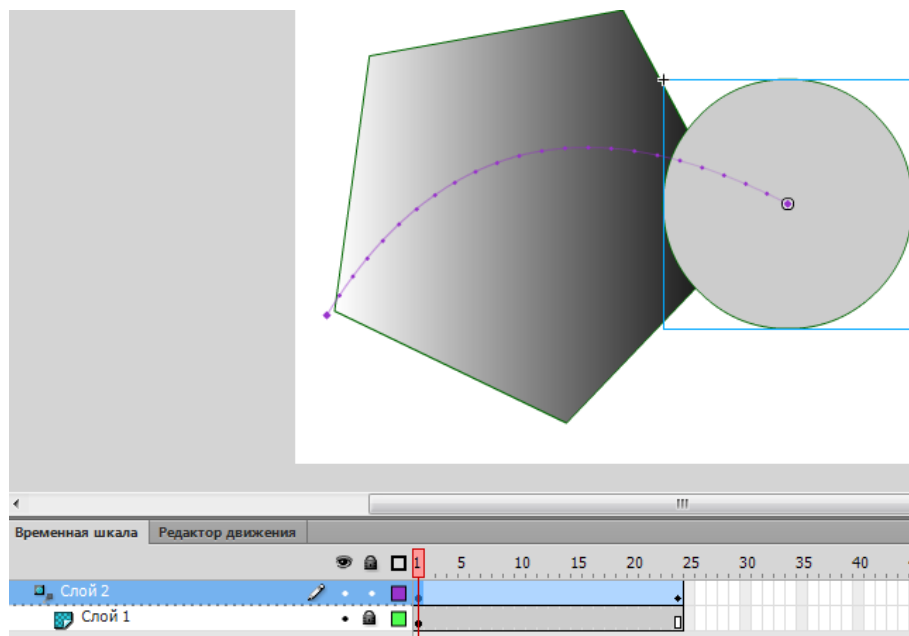


Рис. 1.5.4. Шар, що маскується

Для створення ефекту маски необхідно створити два шари та вставити будь-яке зображення чи об'єкт на нижній шар (шар, що маскується). Виділити верхній шар та виконати для нього команду **Mask** (маска). Таким чином, верхній шар став шаром маски, а нижній – шаром, що маскується.

На верхньому шарі намалювати будь-яку фігуру поверх тієї, що знаходиться на нижньому шарі. Ця фігура буде отвором, через який можна побачити шар, що маскується. Можна виконувати різні види анімацій як з шарами маски, так і з шарами, що маскуються.

1.5.2. Створення й використання символів

У цьому параграфі наявна інформація про створення й використання символів.

! *Символом у Flash називається графічне зображення, кнопка або фрагмент фільму, перетворений у внутрішній формат Flash; при*

цьому перетворений символ автоматично потрапляє до бібліотеки символів і стає її елементом. У фільмах використовуються не самі символи, а їхні копії, винесені з бібліотеки – трансформи (instances).

У Flash існує три основних типи символів: *графічні, кнопки й кліпи*.

Графічний символ (graphic symbol) використовується у фільмі як статичне або анімоване зображення. Його дія описується за допомогою часової діаграми основного фільму. Графічному символу не можуть бути призначені інтерактивні засоби керування й звуковий супровід.

Символ-кнопка (button symbol) призначений для ввімкнення у фільм інтерактивних кнопок, що реагують на дії користувача й керують відтворенням фільму. Із цією метою символу-кнопці можуть бути призначені події (events), на які вона повинна реагувати, і дії (Actions), які ініціалізуються цими подіями. Крім того, із кнопкою може бути пов'язаний звуковий супровід.

Символ-кліп фрагмент ролика (movie clip symbol) слугує для включення в бібліотеку фільму анімацій, що повторно використовуються. Кожний кліп має свою власну часову діаграму, що відтворюється незалежно від часової діаграми основного фільму. Кліп може містити власні інтерактивні елементи керування, а також кліпи більш низького рівня вкладеності. З іншого боку, і кнопка може містити кліп, що дозволяє її "пожвавити".

Оскільки відмінна риса будь-якого символу – це зв'язок з бібліотекою, то до сімейства символів цілком можна віднести ще **чотири різновиди елементів фільму**: звук; імпортовані растрові зображення; шрифт-символ; відеокліп.

Кожен із цих елементів, будучи імпортованим із зовнішнього джерела, обов'язково автоматично опиняється в бібліотеці фільму, а потім може бути використаний у фільмі довільне число разів без необхідності "розмноження" [6].

1.5.3. Призначення бібліотек

У цьому параграфі йдеться про призначення бібліотек, вікно бібліотеки символів, типи бібліотек, відкриті бібліотеки.

У бібліотеці фільму Flash зберігаються символи, які використовуються у фільмі, як створені у Flash, так й імпортовані. Засоби бібліотеки дозволяють переглядати й організовувати їх у ході роботи. Вікно **Library** (бібліотека) відображає список з іменами всіх елементів, що втримуються в бібліотеці. Значок поруч із іменами елементів у вікні бібліотеки показує тип цього файлу.

Вікно бібліотеки символів. При виборі елемента у вікні **Library** (бібліотека) та у верхній частині даного вікна – в області попереднього перегляду – з'являється зменшене зображення цього елемента (рис. 1.5.5). Якщо він містить анімацію або є звуковим файлом, то його можна відтворити, натиснувши кнопку **Play** (відтворити) у вікні попереднього перегляду бібліотеки або в панелі Controller (панель відтворення), щоб попередньо переглянути елемент. Елементи в бібліотеці можна розкласати в папки, а стовпчики вікна **Library** (бібліотека) дозволяють згрупувати елементи за іменами, за типом й за тим, скільки разів елемент використовувався у фільмі (це називається індексом використання), за статусом редагування (якщо елемент пов'язаний з відкритою бібліотекою), за датою останньої зміни. Вікно **Library** (бібліотека) також містить меню **Library Options** (опції бібліотеки), що спливає, за допомогою якого можна змінювати елементи бібліотеки. Відновлення імпортованих файлів після редагування їх у зовнішньому редакторі виконується за допомогою налаштування **Update** (відновлення) у меню **Library Options** (опції бібліотеки). Працюючи у Flash, можна відкрити бібліотеку будь-якого файла Flash *.FIA і зробити елементи бібліотеки цього файла доступними для використання в поточному фільмі.

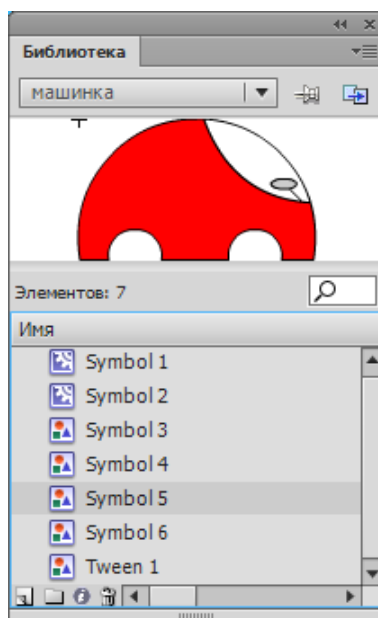


Рис. 1.5.5. Вікно бібліотеки

Вбудовані бібліотеки. У Flash можна створювати постійні бібліотеки, які будуть доступні щораз при запуску програми, Flash також містить кілька вбудованих бібліотек, що містять такі елементи, як кнопки,

графіка, фрагменти фільму й звуки, які можна використовувати у фільмі Flash. Убудовані бібліотеки Flash і створені бібліотеки можна знайти, вибравши команду **Window -> Other Panels/ Common Libraries**.

Відкриті бібліотеки. Маловідомим аспектом роботи у Flash є здатність спільно використовувати матеріали бібліотеки в декількох анімаціях. Функція Shared Libraries використовується при програванні відеоматеріалів у браузері, що робить доступними матеріали бібліотеки.

Функцію Shared Libraries можна також використовувати й під час створення відеоматеріалів. Наприклад, один логотип може використовуватися декількома відеороликами вузла. Ці загальні матеріали дозволяють команді змінити або замінити будь-який символ у відео будь-яким іншим матеріалом із мережі. Наприклад, можна в загальній бібліотеці змінити кольори логотипа. Коли зміна внесена, ім'я й властивості логотипа зберігаються, але зміни в ньому відображаються у всіх фільмах, що використовують його.

Відкриті бібліотеки дозволяють оптимізувати роботу й керування елементами фільму й можуть бути використані, наприклад, для таких цілей:

- щоб зробити звуковий файл загальним для всього сайту;
- щоб створити єдиний шрифт для декількох сайтів;
- щоб забезпечити єдине джерело елементів в анімаціях, які використовуються у різних сценах або фільмах;
- щоб створити єдину бібліотеку для синхронного керування елементами різних фільмів і сайтів.

Відкриту бібліотеку потрібно розмістити в веб так, щоб пов'язані з нею фільми мали доступ до її елементів. Для цього необхідно викласти до веб фільм, що містить відкриту бібліотеку, при цьому бібліотека має ту ж адресу, що й сам фільм [1].

Контрольні запитання

1. Які типи шарів використовує Flash?
2. Які властивості мають шари у Flash?
3. Що таке символ?
4. Які типи символів використовує Flash?
5. Для чого потрібні бібліотеки у Flash?
6. Які типи бібліотек використовує Flash?

Практична 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

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

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

1. Необхідно намалювати фігуру та перетворити її у символ "Графіка". Створити для цього символу анімацію руху.

2. Необхідно додати шар на панелі "**Временная шкала**", та зробити його шаром, що спрямовує – використовуючи команду "**Добавить направляющую классической анимации**". На шарі, що спрямовує, необхідно зобразити шлях (наприклад, інструментом "**Карандаш**", важливо, щоб початок лінії відповідав центру символу на першому та останньому ключових кадрах) (рис. 1.5.6).

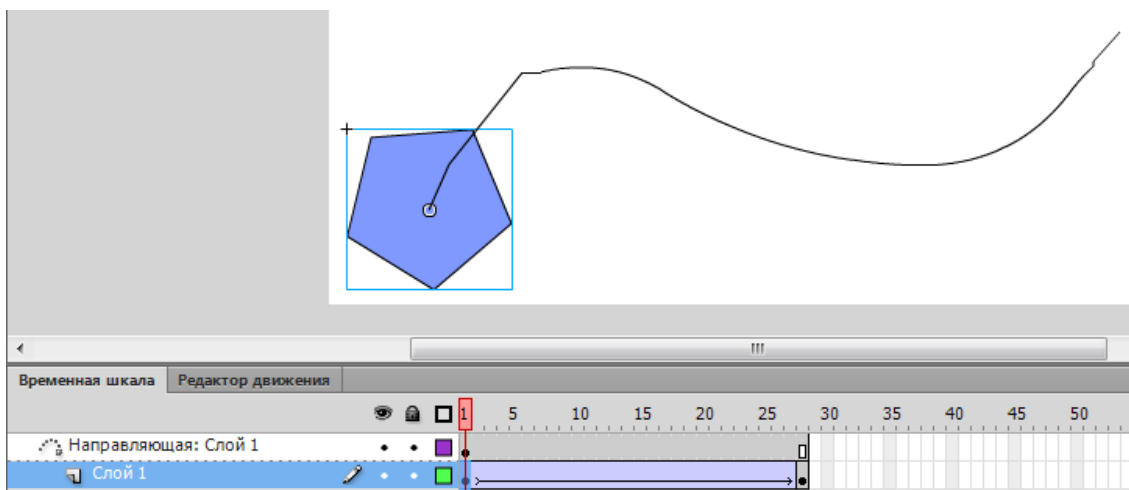


Рис. 1.5.6. Створення шляху на шарі, що направляє

3. Перевірити, щоб лінія життя шару, що спрямовує, продовжувалася так, щоб вона співпадала з лінією життя робочого шару.

4. Запустити фільм на виконання (Ctrl + Enter).

Створення анімації маски

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

Слід звернути увагу, що цікаві ефекти з анімацією маски можна отримати, використовуючи анімацію руху або форми на шарі, що маскує, а також на шарі, що маскується.

Варто розглянути приклад створення використання маски, коли маскуючий шар статичний, а анімація відбувається на шарі, який масковано.

1. Потрібно намалювати широкий прямокутник із використанням градієнтної заливки так, як це показано на рис. 1.5.7.

2. Для отриманого прямокутника необхідно виконати анімацію руху, щоб прямокутник переміщувався з лівого краю до правого (рис. 1.5.7).

3. Необхідно додати ще один шар до проекту, а для цього шару встановити властивість **"Маска"**.

4. На отриманому шарі, що маскує, необхідно намалювати коло, яке б було розташоване на початку руху прямокутника (рис. 1.5.7).

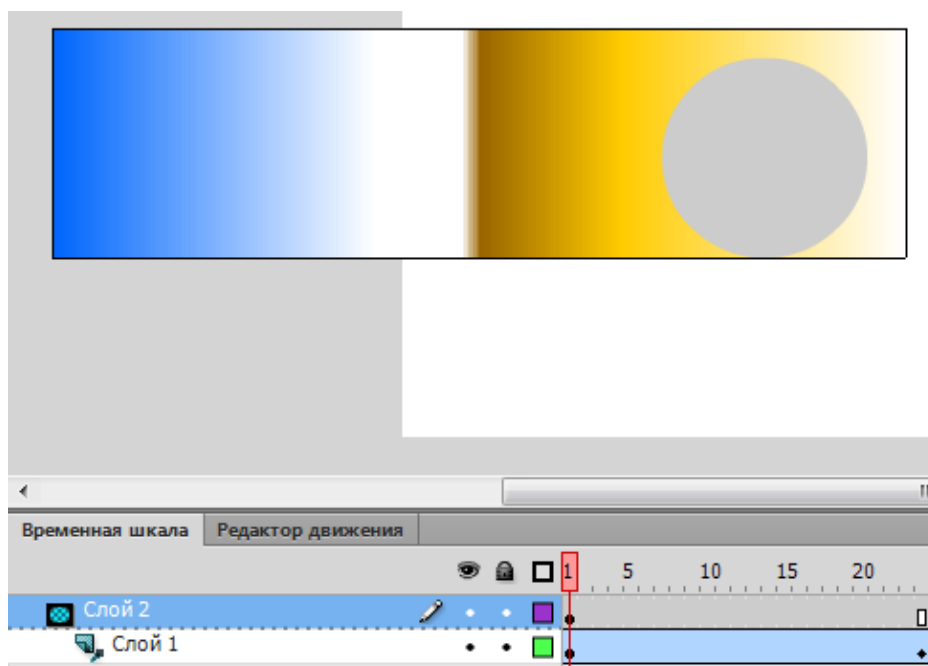


Рис. 1.5.7. Анімація з використанням шару маски

5. Запустити фільм на виконання (Ctrl + Enter). У результаті можна побачити анімацію шара, який обертається та має цікаве наповнення.

Самостійно зробити анімацію, за якою шар, який маскується, був би статичним, а анімація відтворювалася б на шарі, що маскує.

Для цього варто скористатися такими кроками.

1. На шар, що маскується, додати текст "Комп'ютерна анімація".

2. На шарі, що маскує, необхідно зробити анімацію руху так, щоб на першому ключовому кадрі певний об'єкт, наприклад коло, був за розмірами значно менший за напис. На останньому ключовому кадрі це коло повинно перекривати за своїми розмірами напис (рис. 1.5.8).

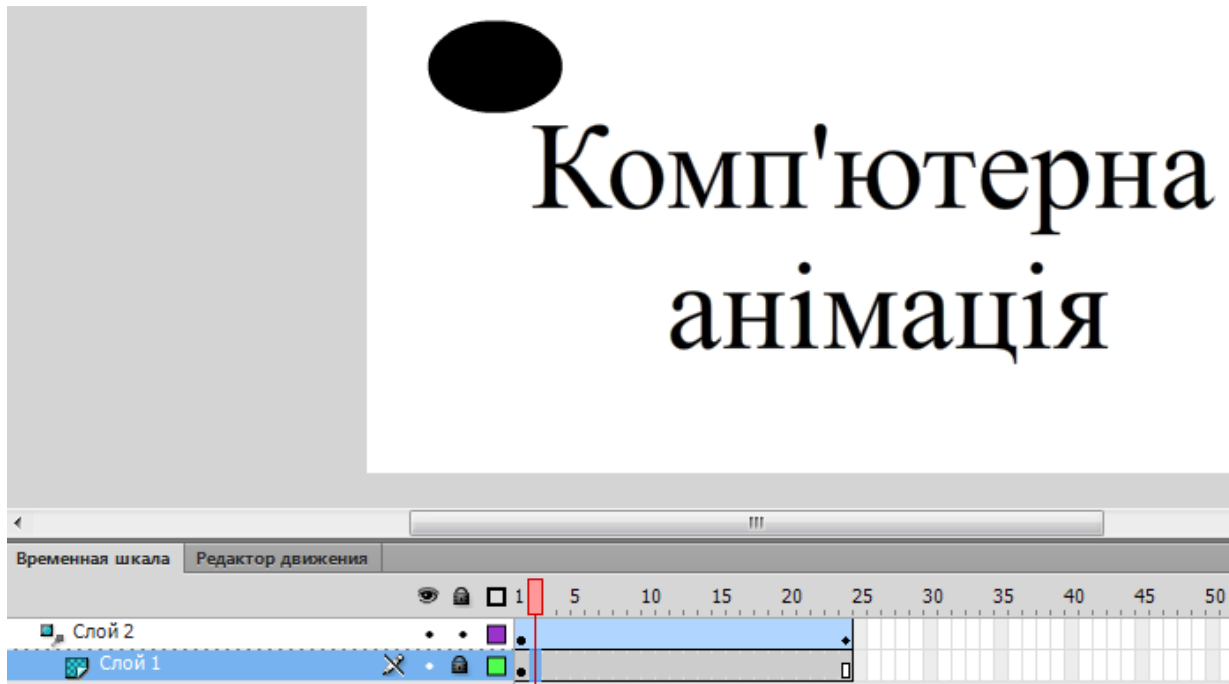


Рис. 1.5.8. Анімація маски

3. Протестувати отриману анімацію, результатом якої буде повільна поява тексту на екрані

Завдання

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

2. Для певних об'єктів організувати в 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.

Практична складова "Символи-контейнери"

Компетентності, що отримуються завдяки цій складовій:

- 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, слід скористатися такою послідовністю дій.

1. У новому документі необхідно створити три шари.
2. У першому шарі створити рисунок кузова автомобіля (рис. 1.5.9).

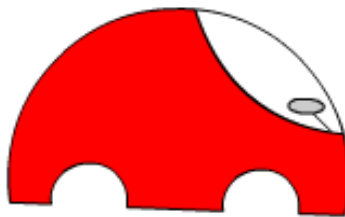


Рис. 1.5.9. Вміст першого шару документа

3. На другому шарі зобразити колесо і зробити з нього символ типу "Фрагмент ролика" (рис. 1.5.10).

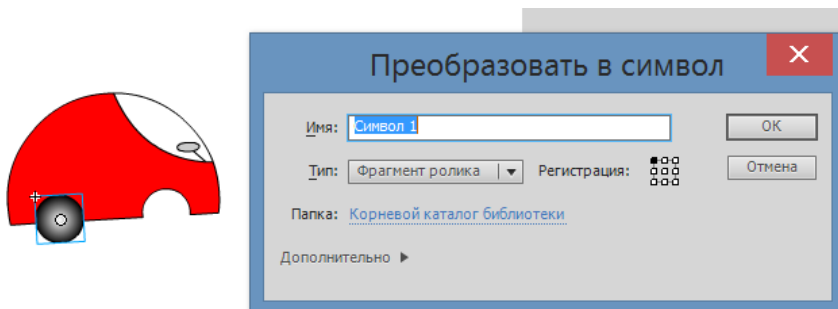


Рис. 1.5.10. Створення символу типу MovieClip (фрагмент ролика) для оформлення коліс

4. Увійти до символу (подвійним клацанням миші на ньому).
5. Виконати анімацію рисунка так, щоб він змінював свій колір у кожному ключовому кадрі (можна взяти 5 ключових кадрів). Для анімації використовувати розкадрування форми.
6. Повернутися на основну сцену (Scene 1) і скопіювати колесо на третій шар (можна використовувати бібліотеку символів Ctrl + L).
7. Виділити автомобіль і задати його 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 автомобіль і задати його кінцеве положення на робочому столі.
11. Створити розкадрування для кожного шару. Для шарів коліс та для шару з кузовом вона буде класичною анімацією руху (рис. 1.5.11). Для додання більш приємного сприйняття анімації доцільно створити кілька послідовних класичних анімацій.

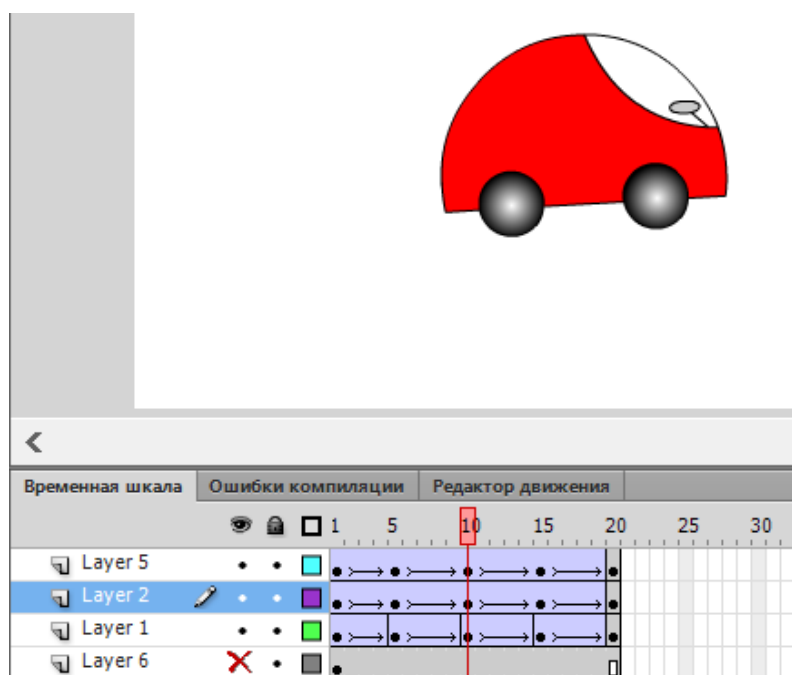


Рис. 1.5.11. Розкадрування елементів основної сцени

12. Запустити фільм (Ctrl+Enter).
13. Можна оформити роботу, створивши шар із фоном. Для цього фоновий шар потрібно перемістити вниз відносно інших шарів (рис. 1.5.12).

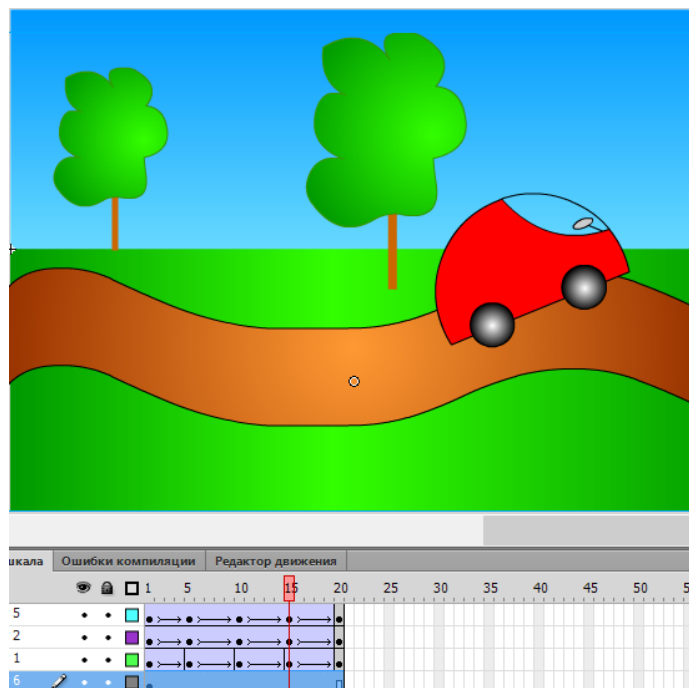


Рис. 1.5.12. Створення фонового шару

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.

1.6. Імпортування й оптимізація об'єктів.

Робота з відео й звуком

Мета: вивчити теоретичні основи імпортування об'єктів графіки, відео та звуку у кліпи.

Ключові слова: графічні формати, імпортування, трасування, імпортування звуку, імпортування відео, частота кадрів.

1.6.1. Імпортування зображень

У цьому параграфі йдеться про графічні формати, які можна імпортувати у Flash та особливості їх імпортування, трасування (векторизацію) растрових зображень.

Основне завдання Flash – передавати через мережу Інтернет компактну векторну графіку та інтерактивну анімацію. Adobe Flash не має

на меті роботу з надскладною графікою і не редагує багато параметрів растрових зображень.

У табл. 1.6.1 наведено список графічних форматів, які можна імпортувати у Flash.

Таблиця 1.6.1

Графічні формати, які можна імпортувати у Flash

Назва формату	Розширення файлів
Adobe Illustrator	ai, eps
Adobe Photoshop	psd
AutoCAD® DXF	dxf
FutureSplash Player	spl
GIF (звичайний та анімований)	Gif
JPEG	jpg, jpe, jpeg
Графічний файл Adobe XML	Fxg
Flash Player 6/7	swf
PNG	png
Метафайли Windows	wmf
Растрові файли Windows	bmp
Розширені метафайли Windows	emf
Зображення QuickTime	qtif (потрібно встановити QuickTime)
TIFF	tif (потрібно встановити QuickTime)

Використання файлів FreeHand. FreeHand є дуже гарним засобом створення графіки для подальшого імпорту у Flash. FreeHand дозволяє експортувати багат шарові зображення, які здатен розпізнавати Flash.

Використання файлів Adobe Illustrator. При виконанні імпорту файла Adobe Illustrator у Flash необхідно розгрупувати всі об'єкти, що з'явилися. Як тільки імпортоване зображення Adobe Illustrator буде розгруповане, з ним можна буде робити ті ж маніпуляції, що й зі всіма іншими Flash об'єктами.

Використання файлів AutoCAD. Flash підтримує файли AutoCAD (*.DXF) починаючи з версії 10. Оскільки DXF-файли не підтримують стандартні системні шрифти, у процесі імпорту Flash спробує підібрати подібні шрифти, однак результат буде непередбачений (включаючи вирівнювання тексту). Оскільки DXF формат не підтримує заливку областей у процесі експорту з Flash, залиті області будуть експортовані тільки як контури.


Саме тому DXF-формат чудово підходить для лінійних типів малюнків, таких, як плани й карти. Flash дозволяє імпортувати тільки двовимірні DXF-файли. Тривимірні файли не підтримуються. Flash не підтримує масштабні параметри DXF-файла. Всі імпортовані DXF-файли з'являються у вигляді фільмів розміром 12x12 дюймів, які можна масштабувати. Flash також підтримує DXF-файли тільки у форматі ASCII. Якщо необхідно помістити у Flash бінарний DXF-файл, треба конвертувати його з бінарного в ASCII-формат перш ніж імпортувати.

Перед імпортом необхідно підготувати зображення: встановити необхідний розмір картинки й кількість кольорів, відретушувати зображення й оформити його остаточний вигляд. Все це допоможе досягти найкращих результатів.

При багаторазовому використанні імпортованого зображення розмір swf-файла не буде пропорційно збільшуватися. Це пояснюється тим, що відбувається вставка не самого зображення як такого, а його екземпляра, тобто Flash оперує не самим зображенням, а математичними даними.

Слід розглянути, як відбувається імпорт зображень. Існує два способи імпортувати у Flash графіку, яка була створена в іншому графічному пакеті: шляхом обрання у меню **File** пункту **Import** або шляхом вставки через **clipboard**.

Імпортоване зображення буде поміщено до бібліотеки та вставлено до сцени у вигляді символу. Якщо імпортуються файли типу png чи gif, то буде збережений режим прозорості, у випадку з анімованим gif, зображення автоматично буде розбито на кадри зі збереженням часових проміжків. Для більш зручної роботи з файлами останнього типу рекомендовано імпортувати анімаційну графіку безпосередньо до символу кліпу (створити новий "Movie clip" (**Insert/New Symbol**), імпортувати анімаційний файл). Кліп, що містить анімацію, можна легко помістити в необхідне місце на сцені з потрібного кадру, тим самим позбавляючись від складнощів із покадровим позиціюванням зображення. Ефект анімаційного кліпу імпортованої графіки можна отримати, якщо файли, що імпортуються, мають однакові імена з цифровим закінченням, наприклад, pics1.gif, pics2.gif та ін. У такому випадку файли будуть імпортовані до послідовних кадрів.

 **Трасування** (векторизація) растрових зображень. Для зменшення розміру файла можна скористатися трасуванням, тобто конвертацією растрової графіки у векторну.

Не слід проводити трасування з фотозображенням, бо у більшості випадків це призведе до збільшення обсягу й зменшенню якості, а ось картинки з мальованою графікою (креслення, схеми, товарні знаки фірм та ін.) трасуються дуже легко. Для трасування треба відкрити вікно трасування Modify/Bitmap/Trace Bitmap. Параметри: Color Threshold – розходження кольору: рівень розходження кольору сусідніх точок (чим менше число, тим більше векторів буде в малюнку); Minimum Area – мінімальний розмір вектора; Curve Fit – значення точності кривих при трасуванні; Corner Threshold – рівень розходження кривих і кутів. Неможливо дати якісь універсальні параметри значень, а тільки застосовуючи різні настроювання можна досягти найкращого результату [6].

1.6.2. Імпортування звуку

Цей параграф містить відомості про процедуру імпорту звуку у Flash.

Можна імпортувати у Flash такі формати звукових файлів:

Назва формату	Розширення файлів
Adobe Soundbooth	asnd
Wave	wav
Формат файлів для обміну аудіоданими	aiff
MP3	mp3
Звук фільмів QuickTime	mov, qt (потрібно встановити QuickTime)

Flash зберігає звуки в бібліотеці також як растри й символи. Аналогічно роботі із графічними символами, необхідна тільки одна копія звукового файла, щоб використовувати цей звук у різних місцях фільму. Звуки можна вносити в розподілені бібліотеки, щоб використовувати той самий звук у різних фільмах Flash.

Щоб використовувати звук у відкритій бібліотеці, треба створити для звукового файла рядок-ідентифікатор у діалоговому вікні Symbol Linkage Properties (властивості зв'язку символу). Цей ідентифікатор може також використовуватися в діях Action Script для доступу до звуку як до об'єкта.

Звукові файли займають значний обсяг дискової й оперативної пам'яті. Однак звук у форматі MP3 займає менший обсяг, ніж звук у форматах WAV або AIFF. Взагалі у форматах WAV або AIFF краще

використовувати 16-бітний звук із частотою 22 кГц. Звук моно займає вдвічі менший обсяг, ніж звук стерео. Flash дозволяє імпортувати 8-бітні або 16-бітні звуки із частотами оцифрування 11 кГц, 22 кГц і 44 кГц. Крім того, Flash може знижувати частоту оцифрування звуку при експорті.

Якщо припускається додавати до звуків у Flash ефекти, то краще імпортувати 16-бітні звуки. При обмеженому обсязі оперативної пам'яті треба робити звукові фрагменти короткими або використовувати 8-бітні звуки замість 16-бітних.

Імпортований звук буде поміщений у бібліотеку поточного фільму. Після цього необхідно створити новий шар для звуку і на панелі **Properties** у списку **Sound** обрати необхідний звуковий трек.

Для визначення точок початку й кінця відтворення звуку або керування його гучністю під час відтворення використовуються засоби редагування звуку в панелі властивостей (Property).

Flash може змінити точки, у яких звук починає або припиняє звучати. Це корисно для створення звукових файлів меншого обсягу й видалення фрагментів, що не використовуються [1].

1.6.3. Імпорт відео

У цьому параграфі йдеться про можливості Adobe Flash для імпорту відео, формати, що підтримуються, основні характеристики відео файлів, попередня обробка відео й оптимізацію в Adobe Flash.

Adobe Flash зможе імпортувати відеоформати тільки у тому випадку, якщо в операційній системі вже є відповідні програвачі, тому необхідно подбати про їхню установку заздалегідь.

Необхідно встановити QuickTime версії 4 або вище (можна одержати з www.apple.com/quicktime) для імпортування таких форматів:

Назва формату	Розширення файлів
Відеофайл для Adobe Flash	flv, f4v
Фільм QuickTime	mov, qt
Відео для Windows	avi
MPEG	mpg,.m1v,.m2p,.m2t,.m2ts, .mts,.tod,.mpe,.mpeg
MPEG-4	mp4,.m4v,.avc
Цифрове відео	dv,.dvi
3GPP и 3GPP2 для мобільних пристроїв	.3gp,.3gpp,.3gp2,.3gpp2, .3p2

Користувачам Windows необхідно встановити DirectX версії 7 або вище.

Для імпорту відео використовується така послідовність дій:

1. Створити новий фільм Flash і, скориставшись інспектором параметрів (Property), установити частоту (frame rate) таку ж, як у вихідному файлі.

2. Якщо створити новий символ відеокліпу у Flash-фільмі й імпортувати відео в нього, буде зручніше маніпулювати відеоматеріалом, ніж якби було імпортовано відеокліп прямо на головний монтажний стіл.

3. Для імпорту відео виконати **File > Import/Import Video**.

Частота кадрів. Перш ніж імпортувати відео, потрібно подумати про частоту кадрів, установлену у Flash. Якщо, як приклад, відеофайл знятий із частотою 30 кадрів на секунду (fps), тоді, щоб відтворити його щонайкраще, потрібно встановити у Flash таку ж частоту – 30 fps. Частота кадрів імпортованого фільму зазначена поруч із його довжиною, праворуч від вікна перегляду. Причина, з якої необхідне настроювання частоти у Flash, полягає в тому, що звукова частина відеофільму завжди відтворюється із частотою кадрів оригінального відеофайла. Імпортоване відео буде відтворюватися з тією частотою, що вказана у Flash, однак звук – тільки із частотою оригіналу.

Якість. Перший параметр – якість (Quality). Цей регулятор визначає ступінь компресії відеокліпу. Установити якість на 100% недостатньо, оскільки кліп, що вийде в результаті, може виявитися навіть більшого розміру, ніж оригінальний файл. Краще трохи знизити значення якості (до 95 %). Для Інтернету рівень стиску чим більше, тим краще, але для презентацій на локальних ресурсах бажано вживати найменшу компресію.

Інтервал ключового кадру. Наступний параметр – інтервал ключового кадру (Keyframe interval). Цим регулятором настроюється частота ключових кадрів відеокліпу. Саме цей параметр практично визначає розмір і "відтворюваність" майбутнього файла. Ключові кадри дуже важливі, адже саме в них утримуються повні кадри відеокліпу. При установці значення 30 ключовим буде кожний тридцятий кадр.

Масштаб. Параметр Масштаб (Scale) визначає розміри зображення (ширину й висоту) відеокліпу після його імпорту. Якщо зменшити зображення порівняно з оригінальним, то знизиться розмір файла, а отже, і відтворюватися він буде краще.

Синхронізація. Наступний параметр – синхронізація відео за частотою (Synchronize video to Adobe Flash document frame rate). Тут

можна компенсувати розбіжність у частоті кадрів, таким чином, щоб у підсумку відео відтворювалося з тією же частотою кадрів, як і фільм Flash.

Передача потокового відео з використанням сервера Adobe Flash Media Server

Сервер Flash Media Server у реальному часі передає потоковий вміст у Flash Player і середовище AIR. Функція визначення пропускної здатності мережі дозволяє серверу Flash Media Server враховувати максимальну швидкість доставки аудіо- та відеовмісту, доступну користувачеві.

Передача потокового відео з використанням сервера Flash Media Server дає порівняно з упровадженим відео та послідовним завантаженням наступні переваги.

Відтворення починається швидше, ніж для впровадженого відео.

При потоковому передаванні даних на клієнтському комп'ютері використовується менше пам'яті і місця на диску, оскільки не потрібне завантаження повного файлу.

Мережеві ресурси використовуються ефективніше, оскільки клієнту передається тільки частина відео, що проглядається.

Доставка мультимедіа надійніша, бо при потоковій передачі дані не зберігаються в кеші клієнтського комп'ютера.

Потокова передача даних дає можливість проводити відео- та аудіопрезентації в режимі реального часу і захоплювати відео з цифрової або веб-камери.

Flash Media Server забезпечує багатосторонню передачу даних для додатків відеоконференцій, відеочатів та обміну відеоповідомленнями.

Контрольні запитання

1. Які графічні формати можна імпортувати у Flash?
2. Які відмінності імпортування векторної та растрової графіки у Flash?
3. Що таке трасування?
4. Як відбувається імпортування звуку у Flash?
5. Як відбувається імпортування відео у Flash?
6. У чому сутність та переваги потокового передавання даних?

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

Компетентності, що отримуються 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, слід скористатися такою послідовністю дій.

Імпортування звуку

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

1. Створити новий шар для звуку.
2. Імпортувати звуковий файл у бібліотеку символів (Файл/Імпорт/Імпортувати в бібліотеку).
3. На панелі "Свойства" для шару зі списку "Звук" вибрати імпортований звук (рис. 1.6.1).

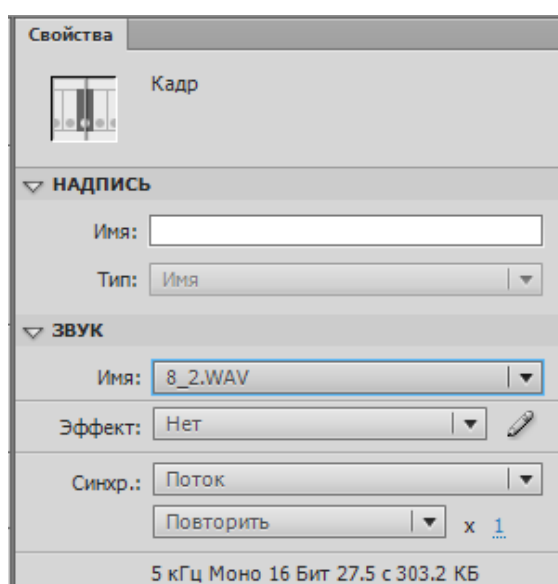


Рис. 1.6.1. Властивості шару

4. Для коректного відтворення потокового звуку необхідно вибрати на панелі "Свойства" в списку "Синхронизация" пункт "Поток" (рис. 1.6.2).



Рис. 1.6.2. Відображення імпортованого звуку

5. За необхідності відредагувати звуковий файл, використовуючи вбудований звуковий редактор (панель Свойства/кнопка "Редактировать").

Для вставки **подієвого** звуку в кнопку необхідно виконати такі 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). Це може бути кадр Over або Down.
5. Для вибраного кадру вибрати звук із бібліотеки.
6. Для коректного відтворення подієвого звуку на панелі "Свойства" необхідно переконаватися, що в списку "Синхронизация" **не вибрано** пункт "Поток".

Імпортування відео

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

1. Створити новий шар для відео, на цьому шарі намалювати прямокутник білого кольору, який за своїми розмірами відповідав би роздільній здатності відео, що імпортується. Прямокутник перетворити у символ "Фрагмент ролика".
2. Зайти у символ "Фрагмент ролика" та імпортувати відеофайл (Файл/Імпорт/Імпортировать відеофайл). Обрати файл із папки з розширенням *.flv.
3. У вікні майстра імпорту відео обрати потрібний параметр: для точнішої синхронізації звуку та інтерактивних елементів Flash-фільму

вибрати "Загрузить внешний видеофайл с компонентом воспроизведения". Потім, у якості параметра "Обложка" обрати – "нет" рис. 1.6.3.

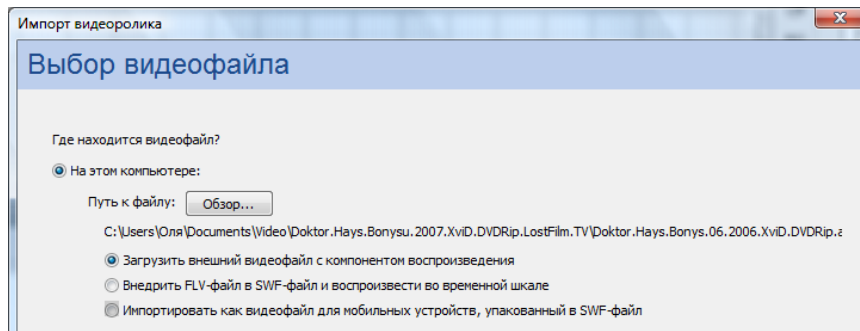


Рис. 1.6.3. Майстер імпорту відеоролика

4. Після імпортування відео у символ "Фрагмент ролика" з ним можна роботи всі можливі перетворення та анімації, які виконуються з іншими символами.

Завдання

1. Імпортувати в Flash растрові графічні файли таких форматів: *.bmp, *.jpg, *.gif. У звіті описати переваги і недоліки імпорту кожного формату.

2. Використовуючи команду "Модификация/Растровое изображение/Векторизация" виконати 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. Для цього для потокового звуку, який необхідно контролювати – змінити ім'я класу, наприклад на snd (рис. 1.6.4)

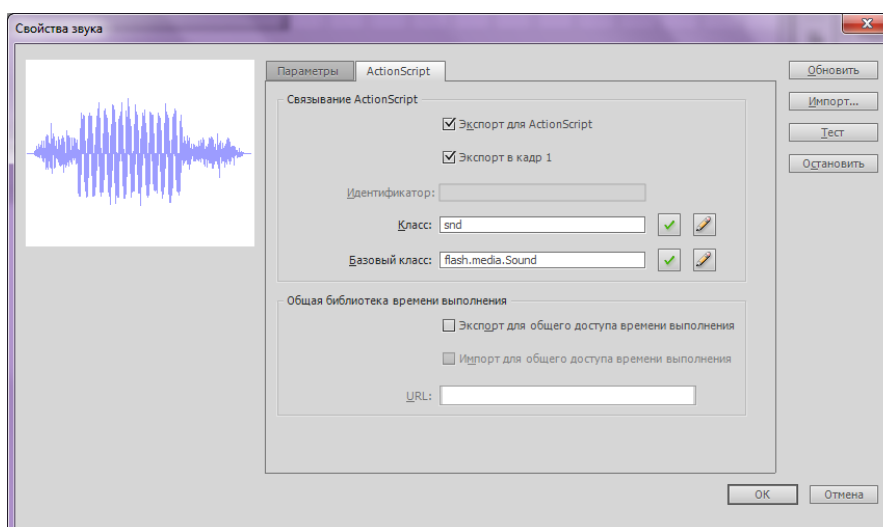


Рис. 1.6.4. Змінення ім'я класу

На першому кадрі додати таку скрипкову конструкцію до редактора "Действия":

```
var sound:Sound=new snd();
var chn:SoundChannel;
var soundisPlaying:Boolean=false;
stop();
btn.addEventListener(MouseEvent.CLICK,playSnd);
function playSnd(e:MouseEvent) {
    soundisPlaying?chn.stop():chn=sound.play();
    soundisPlaying=!SoundisPlaying;
}
```

Намалювати будь-яку фігуру та перетворити її на символ "Кнопка".
Змінити ім'я символу на btn.

Запустити ролик та спробувати управляти звуком за допомогою кнопки.

4. У звіті порівняти імпортування і застосування потокового і подієвого типів звуку.

5. У створений раніше анімований кліп додати відеоролик, застосувати до нього анімацію і візуальні ефекти (скалірування, обертання, зміну колірної відтінку, прозорості, яскравості, накладення анімованого тексту й об'єктів).

Розділ 2. Використання Adobe Flash-технології та 3DSMax для створення сучасних мультимедійних документів

2.7. Використання технології Adobe Flash в електронному маркетингу

Мета: розглянути способи використання анімаційної продукції у сфері Інтернет-реклами, ознайомитися з особливостями створення Flash-банера, розглянути методи, завдяки яким можна аналізувати вплив рекламної продукції на споживача.

Ключові слова: реклама в Інтернет, інструменти реклами, кнопка, Flash-банер, аналіз ефективності реклами, факторний аналіз, багатокритеріальна оцінка.

2.7.1. Рекламна продукція на основі Flash-технології

У цьому параграфі йдеться про тенденції розвитку рекламної продукції в мережі Інтернет, сучасні інструменти, що використовуються для реклами, місце Flash-технології в індустрії реклами.

Зростання ринку реклами в мережі Інтернет досягає рекордних показників.

На вітчизняному ринку Інтернет-реклами зростання відбувається теж стрімкими темпами, що не поступаються світовим тенденціям. У табл. 2.7.1 наведено характеристики витрат на рекламу (як Інтернет, так і медіа-рекламу взагалі) у 2012 – 2013 роках, та прогнозні значення на 2014 рік [28].

Таблиця 2.7.1

Темпи зростання витрат на рекламу

Назва	Всього за 2012 рік, млн грн	Всього за 2013 рік, млн грн	Процент змін з 2013 року до 2012 року	Прогноз на 2014 рік, млн грн	Процент змін з 2014 року до 2013 року
	1	2	3	4	5
Інтернет-реклама	680	1 060	35 %	1 440	36
У тому числі	340	400	18 %	440	10 %
Банерна реклама					

	1	2	3	4	5
Контекст	160	300	25 %	420	40 %
Відеореклама	Немає даних	150	–	300	100 %
Digital	180	210	17 %	280	33 %
Усього Медіа-реклама	9 478	10 553	11 %	11 643	10 %

Відома аналітична компанія Nielsen [30] у своїх звітах зазначає, що світовий ринок Інтернет-реклами у 2013 році зріс на третину (32,4 %) та склав 22 млрд дол. США. При цьому частка Інтернет-реклами у складі загальних витрат на рекламу залишається невеликою та складає 4,5 %. Частка витрат на рекламу по телебаченню складає 58 % ринку, але зростання витрат на такий тип реклами у 2013 році склало лише 4,3 %.

Рекордне зростання у 2013 році продемонструвала реклама для мобільних пристроїв, витрати на яку склали 13 млрд дол. США, у 2014 прогнозується також стрімке підвищення на 37 % до 18 млрд дол. Майже всі інші види реклами, а це реклама на радіостанціях, у газетах, журналах та кінотеатрах скоротили свої бюджети. Вийнятком у 2013 році серед "класичних" видів реклами стала зовнішня реклама, яка зросла на 5,1 %. Багато аналітичних та рекламних агенцій прогнозують зростання у 2014 – 2016 роках сукупних витрат на Інтернет-рекламу та на рекламу для мобільних пристроїв [23].

Крім темпів зростання, ринок реклами в мережі Інтернет характеризується постійним розвитком і вдосконаленням технологій та інструментів, що використовуються. Існує велика кількість успішно функціонуючих технологій та інструментів Інтернет-реклами. Як основні варто назвати такі: посилання на тематичних сайтах, банерні покази, блоги, контекстна реклама, rich media, PR-статті в мережевих виданнях і IT-пресі, прес-релізи, поштові розсилання, календарі, шпалери на робочий стіл, інтерактивні ігри, віртуальні екскурсії, титульні новини, проведення опитувань й анкетування аудиторії, організація спеціальних подій і заходів.

Сучасний інструмент, що активно використовується у Інтернет-рекламі – це поведінкова реклама. Цей тип реклами надається компаніями Google, Yandex, Veon та іншими і заснований на поведінковому таргетингу. Цей механізм дозволяє показувати рекламні оголошення, ґрунтуючись на попередньому поведженні користувача в мережі Інтернет. Інтереси

користувачів визначаються системою на основі статистичної інформації про поведінку користувача в Інтернеті і його споживчих переваг. Статистика збирається з мільйонів користувачів та веб-сайтів, що розміщують контекстну рекламу. Наприклад, якщо у браузері від Google або у пошуковому сервісі від Google шукати характеристики смартфона, то незабаром контекстна реклама від Google, яка буде відобразитися у браузері, буде стосуватися продажу або ремонту відповідних смартфонів безпосередньо у вашому регіоні. Більш того, якщо зареєструватися у якому-небудь із сервісів Google та аутентифікуватись, то ці рекламні повідомлення будуть супроводжувати вас на будь-якому комп'ютері або взагалі – електронному пристрої з доступом до мережі Інтернет.

Одним із найефективніших інструментів реклами є телебачення. Телевізійна реклама дає споживачеві більш повне, візуальне сприйняття продукту, дозволяє передавати споживачеві асоціативні образи товарів і послуг, формує стиль та імідж компанії.

При цьому сучасне телебачення багато в чому зобов'язане одній з найбільш ефективних сучасних технологій – анімації. У свою чергу розвиток інформаційних технологій дозволяє використовувати анімацію як ефективний інструмент реклами не тільки на телебаченні, але в багатьох інших додатках: веб-сайтах, мультимедійних презентаціях, розважальних програмах і т. д.

Анімація дозволяє перебороти й ефективно використовувати в рекламі цілий ряд протиріч [20]:

- багато ідей рекламодавців дуже складно реалізувати, використовуючи відеозйомку, а іноді це неможливо;
- розвиток сучасних інформаційних технологій часто дозволяє зробити більш вигідну пропозицію за співвідношенням ціна/якість ролика з використанням анімації порівняно з традиційною відеозйомкою;
- аудиторія негативно ставиться до реклами, але любить розважальну мультиплікацію.

Причина негативного відношення до реклами криється в тому, що реклама змушує споживача: оцінювати рекламні повідомлення, робити висновки про якість, приймати рішення щодо необхідності покупки. Завдання ефективною реклами – це не тільки звернути на себе увагу, а й викликати позитивні емоції.

Це завдання стає більш важливим, коли розглядається реклама продукції в Інтернеті, де покупця необхідно привернути, залучити до себе, спробувати затримати на сайті як можна довше.

2.7.2. Створення Flash-банера рекламного характеру

У цьому параграфі йдеться про створення кнопок у Adobe Flash, основні інструменти, що використовуються для створення Flash-банерів.

! **Кнопка** – це базовий елемент управління, що надається Flash. Кнопки використовуються дуже часто, і не тільки в додатках, але й у звичайних інтерактивних фільмах Flash, формах, для створення веб-сайтів.

Для створення кнопки необхідно створити новий Symbol типу Button (пункт New Symbol у меню Insert або пункт New Symbol у додатковому меню вікна бібліотеки).

Кнопка має свою тимчасову шкалу (рис. 2.7.1), яка складається з чотирьох кадрів (розділів тимчасової шкали). Ці кадри мають назву: Up, Over, Down і Hit.

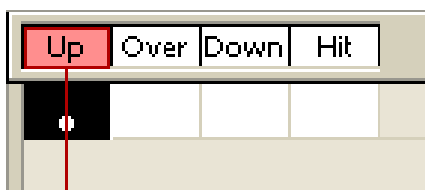


Рис. 2.7.1. Чотири кадри, з яких складається кнопка

Ці кадри містять зображення, що будуть виведені Flash в різних випадках:

- кадр up, – кнопка не натиснута, і курсор миші не розташований над нею;
- кадр over – курсор миші розташований над кнопкою, кнопка не натиснута;
- кадр Down – користувач натискає ліву кнопку мишки;
- кадр Hit – задає область, що буде відгукуватися на натискання миші ("гарячу" область кнопки). Це зображення не відображається Flash.

До кнопок також можна прив'язувати звуки (це можна робити до кожного стану кнопки). Насамперед, потрібні звуки необхідно імпортувати.

Потім додається новий шар для звуків, виділяється конкретний розділ на шкалі анімації залежно від стану кнопки (Up, Over, Down і Hit), створюється новий порожній ключовий кадр. Після цього необхідно перекинути потрібний звук на робочий лист.

Ще однією особливістю кнопки Flash є те, що "гаряча" область кнопки не обов'язково повинна збігатися із самою кнопкою. Кнопка може спрацьовувати, якщо користувач клацне на іншому місці на робочому листі.

! **Банером** (англ. *banner*) – називається прямокутне графічне зображення рекламного характеру. Банери розміщують для залучення потенційних клієнтів або для формування іміджу [26].

Основні види банерів:

Зовнішня реклама – розтяжки-банери.

Інтернет-реклама – графічне зображення або текстовий блок рекламного характеру, що є гіперпосиланням на веб-сторінку з розширеним описом продукту або послуги.

Традиційні Інтернет-банери є графічними зображеннями у форматі GIF або JPEG. Вони можуть бути як статичні, так і анімовані (де ефект руху досягається чергуванням декількох зображень). Сучасні банери виготовляються на основі технологій Flash і Java. На відміну від попередніх, що використовують растрову графіку, ці банери використовують векторну графіку, що дозволяє робити анімаційні ефекти при невеликому розмірі банера. Крім того, Flash-банери надають можливість використання звукових ефектів, що підвищує ефективність банера. Існують інтерактивні банери, що використовують у своєму складі елементи управління сюжетом банера.

Сучасні підходи створення банерів використовують такі технології, як Rich-media чи Pop-up банери, що розміщуються у верхній частині веб-сторінки, можуть мати вільну форму та в обов'язковому порядку мають кнопку для закриття або згортання. Такий вид банерів, як Screen-Glade при наведенні на них курсора відкривають додаткову область, що містить основну рекламну інформацію.

При встановленні розмірів рекламних роликів під Flash Professional необхідно слідувати нормативам Interactive Advertising Bureau (IAB). Далі подані рекомендовані розміри рекламних матеріалів:

Широкий хмарочос 160x600; хмарочос 120x600; напівсторінкова реклама 300x600; повнорозмірний банер 468x60; банер половинного розміру 234x60; мікропанель 88x31; кнопка 1 120x90; кнопка 2 120x60; вертикальний банер 120x240; квадратна кнопка 125x125; передня панель 728x90; середній прямокутник 300x250; квадратне спливаюче вікно 250x250; вертикальний прямокутник 240x400; великий прямокутник 336x280; прямокутник 180x150.

Шаблони, вбудовані у Flash, дозволяють систематизувати створення банерів, використовуючи заготовки у вигляді стандартизованих розмірів файлів.

Важливим завданням при створенні банерів є використання тексту або графічного матеріалу, що може привернути увагу та запам'ятися користувачу (рис. 2.7.2).

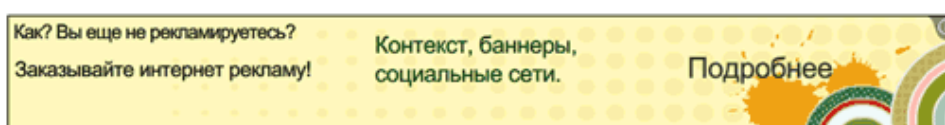


Рис. 2.7.2. Приклади рекламного банеру формату 468x60

На рис. 2.7.3 наведено приклади банерів та їх контенту.



На рекламі заряджувального пристрою Ansmann наведено логотип Brand name, що відрізняє його від інших, наведено важливішу відмінну рису – швидкісна зарядка, в центрі розташовано рисунок, на якому можна побачити сам зарядний пристрій



Контекстний банер, що динамічно генерується



Меню, що випадає, і великий заголовок привертають увагу. Для скорочення шляху користувача до результату телефон зазначено безпосередньо на банері

Рис. 2.7.3. Приклади вдало створених банерів



Банер відмінно передає зміст – прохолоду від кондиціонерів, що рекламуються. Про сніг у літню спеку мріє майже кожний офісний працівник

Яндекс Директ



[Скидки на матрасы и подарки](#)

Матрасы на любой вкус: жесткие, мягкие, средние, пружинные и беспружинные.

Адрес и телефон materasso.com.ua

[Детский интернет магазин Panama](#)

Мегамаркет детских товаров Panama обновил ассортимент! Бесплатная доставка.

panama.ua



[Супер скидки на матрасы до 50%!](#)

Лучшие матрасы Аскона. Бесплатная доставка. Рассрочка без переплаты!

[Кровати от 3920 грн.](#)

[Как выбрать матрас](#) · [Американские матрасы №1](#)

Адрес и телефон askona.ua

Динамічно згенерований банер, створений з повідомлень контекстної поведінової реклами

Закінчення рис. 2.7.3

Дуже важливою характеристикою банера є його розмір у байтах. Чим більше розмір банера, тим довше банер буде завантажуватися на сторінку й тим менша ймовірність, що користувач встигне його переглянути.

Основні завдання банера:

1. Привернути увагу.
2. Зацікавити.
3. Підштовхнути до переходу на сайт. Це завдання досягається за допомогою елемента недосказаності в змісті банера.
4. Спонукає до дії, тобто до покупки товару або послуги на самому сайті.

Параметри ефективності банерної реклами

Кількість показів банера – це основний параметр для рекламної кампанії. Покази зазвичай вимірюються тисячами.

Кількість кліків – це другий, не менш важливий, параметр. Для рекламодавця важливіше кількість кліків, ніж кількість показів.



Ефективність банера оцінюється параметром **CTR** (*click through ratio*) – це відношення кількості кліків до кількості показів, що вимірюється у відсотках.

Вартість банерної реклами в основному визначається тим, скільки коштує тисяча показів банера на даному сервері. Для позначення цієї величини використовується параметр **СРМ** (cost per mille = cost per thousand impressions) – вартість тисячі показів.

Існують **мережі банерного обміну**, за якими кожний учасник надає певне місце на своєму сайті для показу банерів інших учасників мережі й одержує можливість показувати свої банери на інших сайтах пропорційно (за винятком комісійних).

2.7.3. Аналіз впливу інформації, що подається споживачу, й оптимізація Flesh-банера

У цьому параграфі йдеться про використання факторного аналізу для розробки реклами, способи оцінювання ефективності реклами.

Розробка якісної реклами будь-якого продукту є складним багатоетапним процесом. Слід розглянути етапи методу розробки рекламної продукції, що використовує факторний аналіз [32] (розглянуті етапи є підготовчими й передують безпосередній розробці):

- виявити основні фактори, що впливають на оцінку й вибір продукції подібного роду;
- зрозуміти очікування цільової аудиторії від "ідеального" продукту й оцінити, яким із цих очікувань відповідає продукт, що просувається, а яким – ні. Також необхідно порівняти продукт, що просувається, із продукцією конкурентів, оцінити його конкурентні переваги й недоліки;
- вибрати групу, очікуванням якої продукт, що просувається, найбільше відповідає (тобто цільову аудиторію);
- визначити оптимальні способи рекламування. Тут необхідно зрозуміти, які засоби реклами найбільше повно охоплюють цільову аудиторію і які змістовні елементи реклами будуть максимально сприятливо сприйматися цільовою аудиторією.

Після реалізації розглянутих етапів переходять до розробки рекламного повідомлення. Для того, щоб розробити вдале рекламне повідомлення, необхідно знати, чи дійсно воно добре сприймається цільовою аудиторією й чи формує воно образ продукту, що відповідає сприятливим очікуванням цільової аудиторії. Варто розглянути завдання аналізу основних факторів сприйняття цільовою аудиторією продукту, що рекламується.

ся. Завдання аналізу вирішується на основі факторного аналізу й методу семантичного диференціала в декілька етапів.

1. Виявляються основні фактори оцінки продукту споживачами, очікування від цього продукту.

2. Оцінюється, наскільки продукт відповідає очікуванням споживачів, у цільовій групі:

- розробляються 15 – 20 оцінних шкал, для цього необхідно провести 20 неформалізованих інтерв'ю;

- оцінка групою респондентів продукції, що просувається, за цими шкалами, продукції основних конкурентів і "ідеальної" продукції. Вибірка респондентів повинна бути досить великою й різноманітною (за соціально-демографічними і соціально-економічними показниками). Об'єктів, що оцінюються не повинно бути занадто багато, їхня кількість ускладнить оцінку.

3. Аналіз отриманої інформації. Розраховуються середні значення для кожного продукту за всіма шкалами:

- середнє арифметичне – середня оцінка продукту в групі;
- мода – визначає, які оцінки виставлялися найчастіше.

4. Розрахувавши середні значення, можна зрівняти оцінювані об'єкти між собою. Порівняння об'єктів із "ідеалом" дасть інформацію про те, наскільки вони відповідають очікуванням споживачів.

За допомогою попарного порівняння можна виявити латентні фактори оцінки об'єктів споживачами й довідатися, які з них більш впливові.

5. На основі виділених факторів будується семантичний простір сприйняття об'єктів. Аналіз розташування об'єктів у цьому просторі дозволить оцінити, наскільки вони відповідають факторам, що виділені дослідником. При цьому "ідеальний" продукт буде наближений до верхнього правого кута простору, тому що в силу своєї "ідеальності" він буде максимально відповідати виявленим факторам оцінки. Положення "ідеального" продукту в системі координат семантичного простору показує очікування респондентів. Положення інших об'єктів демонструє, наскільки вони відповідають очікуванням споживачів. Аналіз їхнього положення дозволяє зробити висновки про те, як сильно вони відрізняються від "ідеалу" і які їхні розходження між собою. На рис. 2.7.4 зображено приклад семантичного простору [32]. Осі координат – це два обраних фактори: зручність використання продукту й ціна.

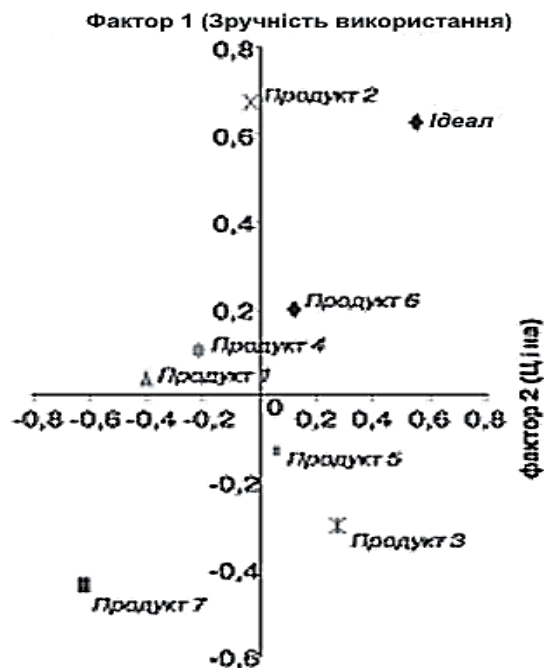


Рис. 2.7.4. Приклад семантичного простору

Способи оцінки ефективності реклами

Оцінка ефективності реклами стає усе більше важливим питанням і, в першу чергу, це пов'язано з підвищенням ролі реклами й збільшенням затрачуваних на неї засобів. Часто рекламодавці можуть тільки підозрювати про те, що частина рекламної компанії не одержала очікуваного ефекту, вкладені кошти себе не виправдали, але не можуть визначити точно, яким чином це відбулося, і як вони зможуть уникнути цього в майбутньому. Така проблема зажадала розробки спеціальних методів оцінки ефективності реклами, які будуть розглянуті далі.

Прийнято виділяти два види ефективності реклами: комунікаційну й кінцеву (за результатами продажів) [20]. Більшою мірою, маркетингові дослідження спрямовані на визначення комунікаційної ефективності реклами, визначити яку істотно легше порівняно з кінцевою ефективністю. Слід розглянути види досліджень, які можна проводити для вирішень завдання оцінки ефективності рекламної кампанії.

1. Когнітивний рівень. Вплив на зміну знань про фірму, її товари і послуги.

2. Афективний рівень. Вплив на формування позитивного ставлення до виробника.

3. Рівень, що спонукує. Вплив на формування намірів вступити в контакт із фірмою, придбати її товар.

Дослідження ефективності рекламних кампаній варто проводити до й після проведення рекламної кампанії. Якщо марка тільки виходить на ринок, показникам привласнюються нульові значення, дослідження проводиться тільки після рекламної кампанії.

Дослідження ефективності рекламної кампанії дозволяє враховувати такі показники й оцінювати на їхній основі вплив реклами на потенційного споживача [32]:

- охоплення цільової аудиторії;
- активне, пасивне знання рекламованої марки (товару);
- розуміння й розпізнання рекламного повідомлення (реклами);
- легкість запам'ятовування елементів реклами, рекламного повідомлення;
- притягальну й агітаційну силу рекламного повідомлення;
- намір купити, користуватися рекламованим товаром;
- загальне ставлення до реклами;
- сформований образ фірми.

Для визначення ефективності рекламної кампанії проводять тільки кількісні дослідження (питання про виміри ефективності реклами до її розміщення належать до питання тестування). Мінімальна кількість опитаних – 400 осіб.

Можливе використання таких варіантів опитування.

1. **Телефонне опитування.** Проводиться швидко, витрати невеликі. Не можна проводити в містах із низьким рівнем телефонізації, немає можливості показати зображення, мала кількість запитань.

2. **Квартирне опитування.** Можливість проводити глибокі дослідження, ставити складні запитання (велику кількість) й показати зображення. Реалізація вимагає більше коштів і багато часу.

3. **Вуличне опитування.** Проводиться швидко, витрати невеликі. Низький рівень контролю за роботою інтерв'юерів, мала кількість запитань.

Основні запитання, які варто ставити:

1. Рівень популярності марки "до" і "після" проведення рекламної кампанії.

- Які торговельні марки Ви знаєте? Перша названа марка (top-of-mind) – це непрямий показник лідерства на ринку (звичайно, якщо розглядати масові ринки). Спонтанне знання (знання марки без підказки) – визначає основних гравців ринку.

- Які торгівельні марки Ви знаєте з даного списку? Знання з підказкою (наведене знання, пасивне знання) – визначає ступінь конкурентоспроможності ринку.

2. Знання реклами даної категорії товарів.

Аналогічно рівню знання марки фіксується як спонтанне, так і пасивне знання.

Існує метод виміру легкості запам'ятання реклами (М. Пикулева), що базується на трьох складових: *"спонтанний спогад"* – реклама пригадується при згадуванні категорії товару; *"спогад при пред'явленні товару певної торгівельної марки"* – реклама пригадується після проголошення назви конкретного товару або його демонстрації; *"спогаду після переказу реклами"*. Сума цих трьох показників становить частку цільової аудиторії, що згадала рекламне повідомлення. Оцінюється співвідношення показників між собою.

3. Відношення до реклами в цілому.

Ефективніша та реклама, що подобається. Тому в дослідженнях з ефективності рекламної кампанії ставляться питання: "Чи сподобалася Вам реклама товару N?", "Наскільки Вам сподобалася реклама товару N?", "Чи цікава Вам реклама товару N?" і т. д.

4. Розуміння основної ідеї реклами, переказ змісту.

Для майбутніх кампаній дуже важливо оцінити, що запам'ятали в рекламі, чи зрозуміли те, що рекламодавці хотіли донести до споживачів, як саме переказують зміст реклами, який характер їй привласнюють.

Респондентам може бути запропонований список, для оцінки/ототожнення реклами з певним типом події, наприклад: весела, молодіжна, енергійна, оригінальна, незрозуміла, фальшива, нудна.

5. Які почуття стосовно товару виникають після перегляду реклами.

Зазвичай питання звучить так: "Що Ви можете сказати про дану торгівельну марку після того, як Ви побачили цю рекламу?". Респондентам пропонують певні характеристики, з яких можна вибрати найбільш відповідну.

6. Вплив реклами на поведження споживача.

Абсолютний показник ефективності реклами визначається зробленими покупками. Зважаючи на те, що думка респондента може не збігатися з поведженням, не зайвим буде питання про переваги, останню покупку й т. д.

Контрольні запитання

1. Які тенденції розвитку реклами в мережі Інтернет? Опишіть основні інструменти реклами, що використовуються в Інтернеті.
2. Опишіть процедуру створення кнопки в Adobe Flash.
3. Що таке банер? Які існують види банерів? Опишіть їхнє призначення.
4. Опишіть принципи використання факторного аналізу для розробки реклами.
5. Опишіть способи оцінювання ефективності реклами.

Практична складова "Створення анімованої кнопки (банера)"

Компетентності, що отримуються 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, слід скористатися такою процедурою:

1. Створити рисунок (будь-яку геометричну фігуру, яка буде кнопкою).
2. Виділити і зробити з рисунка кнопку (команда "Преобразовать в символ" тип "Кнопка") (рис. 2.7.5). Створеній кнопці необхідно присвоїти ім'я (краще латиницею).

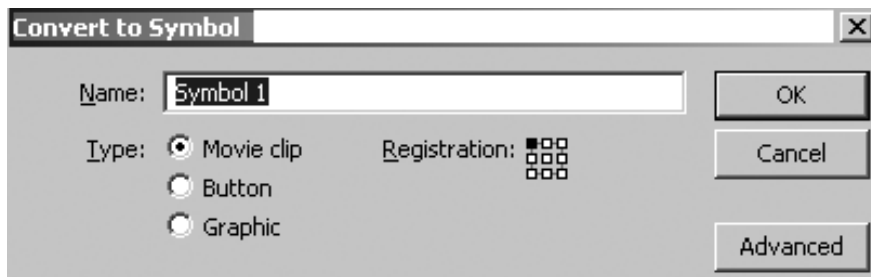


Рис. 2.7.5. Створення кнопки

3. Далі необхідно зайти всередину кнопки подвійним натисненням для продовження її редагування.

4. Зробити ключовим кадр Over (виділити кадр, F6) (рис. 2.7.6).

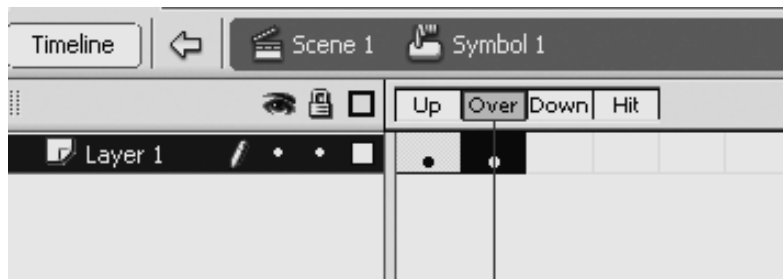


Рис. 2.7.6. Часова шкала кнопки

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

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

7. У середині нього виконати анімацію (довільна анімація, яка відбуватиметься при наведенні на кнопку).

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

Завдання

1. Проаналізувати створений раніше кліп і додати до нього символі-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.

2.8. Створення презентацій на основі технології Adobe Flash

Мета: ознайомитися з основними видами мультимедійних презентацій, методами їх створення та об'єктами, що в них використовуються.

Ключові слова: презентація, види мультимедійних презентацій, майстер створення, PowerPoint, конвертація.

2.8.1. Призначення й основні характеристики презентацій

У цьому параграфі наведено інформацію про типи презентацій; класифікацію за способом реалізації, призначенням, змістовим навантаженням, основні характеристики презентації, аналіз розроблених прототипів.

! **Презентація** (від англ. *presentation*) – спосіб наочного подання інформації з використанням аудіовізуальних засобів.

Презентація зазвичай містить у собі текст, ілюстрації до нього й завжди витримана в єдиному графічному стилі.

Сучасні мультимедійні презентації дозволяють використовувати аудіо-, відеовставки й анімацію. Мультимедійні презентації стають динамічними й інтерактивними, використовують гіпертекстові посилання [30].

Основні види об'єктів, яким можуть бути присвячені презентації:

- підприємства й організації – презентації, які можуть надати потенційному споживачеві або партнерові компанії не тільки інформацію про товари, послуги й сервіс, але й здатні донести до споживачів інформацію про рівень організації, імідж, місію й потенціал компанії непрямым способом, використовуючи дизайн і аудіо-візуальні елементи презентації;
- товари й послуги, враховуючи електронні – презентації повинні підкреслити візуальну привабливість товару, передати покупцю інформацію про унікальність споживчих властивостей продукту, використовуючи мультимедійні можливості презентації;
- заходи – використання мультимедійних презентацій для інформування про заходи дозволяє точно передати призначення й атмосферу

проведення заходу. Використання мультимедійних технологій дозволяє домогтися підвищення ефективності сприйняття в цільовій аудиторії (презентації дозволяють одержати більшу ефективність незалежно від типу події, що презентується: політична акція, молодіжний розважальний захід, проект щодо захисту навколишнього середовища й т. д.);

- особистості – презентації особистостей повинні ототожнювати людину, яку презентують з тим іміджем, на який, на її думку, повинні позитивно реагувати оточуючі. Крім іміджу особистостей у презентаціях передається інформація про кар'єру, основну діяльність, отримані результати, сімейний стан й т. д.

Призначення електронних презентацій:

- донесення рекламної інформації до партнерів і клієнтів;
- забезпечення довідковою інформацією (віртуальний каталог продукції);
- візуальне оформлення конференцій, семінарів і виставок;
- інформаційний супровід проектів і об'єктів, що неможливі або неефективні при використанні звичайних способів;
- інформування про ідеї й досягнення, що містять нетривіальний підхід і вимагають ємного інформаційного супроводження для розуміння;
- інтерактивний навчальний матеріал.

Основні цілі презентацій подані на рис. 2.8.1.

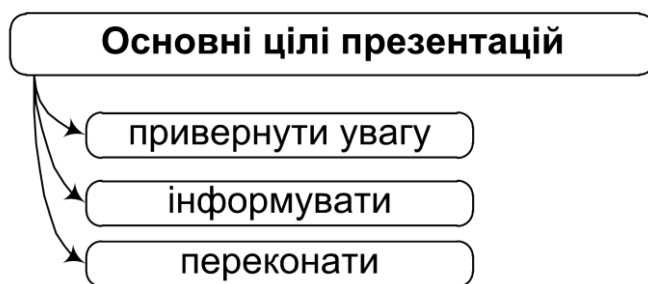


Рис. 2.8.1. Основні цілі презентацій

Класифікація презентацій з точки зору подання інформації:

- презентація для доповіді й демонстрації на мультимедійному проекторі;
- презентація для демонстрації на виставці;
- презентація для реклами в громадських місцях: білборд, плазмова панель;

- презентація проекту на CD/DVD диску;
- презентація для демонстрації на телебаченні;
- презентація для завантаження через Інтернет;
- комплексні презентації, що містять натурні моделі, зразки, роздавальний матеріал.

Класифікація презентацій з точки зору їхнього змісту:

- Представницька презентація (презентація організації), що містить повну інформацію про компанію, її види діяльності, успіхи у розвитку.

- Маркетингова презентація, що містить основні відомості про пріоритетні напрями діяльності компанії й продукції, що вона випускає. Призначення маркетингових презентацій складається в підготуванні ґрунту для майбутніх торгівельних презентацій.

- Торгівельна презентація (презентація товару), що містить досить детальну інформацію про товар.

- Електронний каталог товарів.

- Презентація проекту. Ціль цього виду презентації – інформування людей про який-небудь проект, формування позитивного ставлення до проекту, пошук зацікавлених осіб.

- Презентація обсягу й змісту виконаних робіт, тобто звіт (наприклад, річний звіт банку).

- Мультимедійний бренд-бук.

- Віртуальний тур – це мультимедійний спосіб подання навколишнього простору. Це може бути кругова панорама берега моря, на якому розташований готель, огляд території нового складського комплексу, відеоекскурсія офісом або відео-каталог експозиційних площ виставки. Мультимедійний віртуальний тур створює повний ефект присутності й максимально реалізує принцип "краще один раз побачити".

- Навчальна презентація.

Види презентацій з погляду технологій, що використовуються:

- HTML-презентація. Найпоширеніший вид презентації. Є самостійним веб-сайтом презентаційного змісту. Має невеликий розмір і може публікуватися як в Інтернеті, так і на компакт-диску. Відносно нескладні для створення й відповідно не вимагають великих матеріальних витрат.

- Flash-презентація. На даний момент – це вид презентацій, що найбільш швидко розвивається та має найбільший потенціал. Flash-

презентації мають можливість створення ефектного багатофункціонального анімаційного матеріалу. Мають більші можливості інтерактивності. Можуть публікуватися в мережі Інтернет і легко відправлятися по e-mail. Flash-презентації також мають невеликий розмір.

- Презентація, створена за допомогою спеціалізованих програм (наприклад, PowerPoint). Як правило, використовується при публічних виступах або поданні навчального матеріалу з метою підсилити ефектність виступу й зробити запропонований матеріал більш наочним. Дуже прості в розробці, тому що орієнтуються на функції основних офісних додатків, але при цьому не достатньо ємні за функціональним наповненням.

- Презентації у вигляді повноекранних додатків. Можуть бути засновані на технології Flash або інших спеціалізованих пакетах комп'ютерної анімації й обробки відео. Як правило, створюються з індивідуальним підходом до дизайну презентації й мають максимальний набір мультимедійних елементів. У наслідку чого є найкращими з тих, що розглядались, з точки зору візуального ефекту від перегляду й ефективності впливу, але при цьому досить дорого коштують.

- Презентація змішаного типу. Використовуються одночасно декілька зазначених технологій, об'єднаних єдиним інтерфейсом (програмна оболонка мовою високого рівня для перегляду). Як правило, використовується для публікації й поширення на компакт-дисках.

2.8.2. Об'єкти, що використовуються в презентаціях

У цьому параграфі йдеться про основні типи об'єктів, формати даних для презентацій і їхню характеристику, підготовку даних для використання в презентації.

У презентаціях можна використовувати різні види мультимедійного контенту. На рис. 2.8.2 наведено форми подання інформації електронних презентацій.

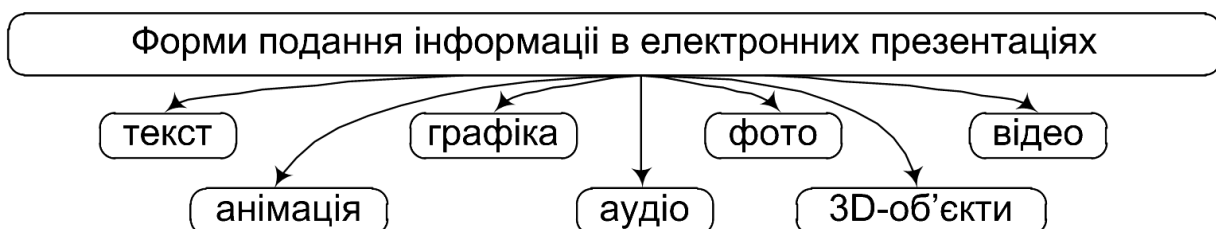


Рис. 2.8.2. Форми подання інформації

Вимоги до даних, що підготовлюються для презентації, більшою мірою залежать від подальшого застосування презентації.

Можна запропонувати такі рекомендації для матеріалу, що підготовлюється залежно від способу демонстрації презентації.

Презентація для доповіді й демонстрації на мультимедійному проекторі. Час обмежений регламентом доповіді; важливий компроміс між кількістю запропонованого матеріалу, рівнем деталізації й можливістю сприйняття інформації з одного слайда; розмір використовуваних файлів принципового значення не має; можна розраховувати на те, що користувачі зможуть витратити додатковий час на перегляд і аналіз матеріалів презентації.

Презентація для демонстрації на виставці. Час презентації обмежений можливістю потенційного користувача приділяти увагу саме цій презентації відносно обсягу всієї виставки; важливо сфокусувати увагу на основних характеристиках; розмір використовуваних файлів принципового значення не має.

Презентація для реклами в громадських місцях: білборд, плазмова панель. Час строго обмежений певним значенням – час очікування обслуговування, поїзда, черги, зупинки тощо, важливо сфокусувати увагу на основних характеристиках; у користувача немає можливості сприймати детальну й докладну інформацію; розмір використовуваних файлів принципового значення не має.

Презентація проекту на CD/DVD диску – розмір використовуваних файлів принципового значення не має; можливе надання докладної інформації й деталізація характеристик; можна розраховувати на те, що користувач зможе витратити додатковий час на перегляд презентації;

Презентація для демонстрації на телебаченні – обмежений час; розмір використовуваних файлів значення не має; докладну й детальну інформацію краще не використовувати.

Презентація для завантаження через мережу Інтернет. Інформація може пропонуватися в деталізованому вигляді; розмір використовуваних файлів має принципове значення й повинен відповідати можливій швидкості завантаження потенційних користувачів; можна розраховувати на те, що користувачі зможуть витратити додатковий час на перегляд і аналіз матеріалів презентації.

2.8.3. Майстер створення презентацій

У цьому параграфі йдеться про типи презентацій, що створюються майстром, способи перетворення презентацій різних форматів у Adobe Flash.

В Adobe Flash передбачені майстри створення документів. Одним із видів документів, які користувач може створити на основі зразків, є презентація.

Структура згенерованої презентації містить у своєму складі чотири шари (рис. 2.8.3). Це, відповідно, шари для розташування: інструкцій, подій, фонового зображення, навігаційних кнопок та основного фрагменту.

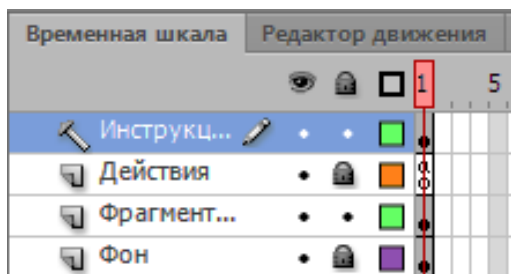


Рис. 2.8.3. Структура шарів презентації

У презентації міститься лише один кадр, на якому знаходиться фрагмент – movieclip. Цей фрагмент містить чотири кадри (рис. 2.8.4) (кількість можна збільшувати чи зменшувати за потреби сценарію презентації). Використовуючи навігаційні кнопки та звичайну клавіатуру фактично виконується управління цим фрагментом та його кадрами (рис. 2.8.5).

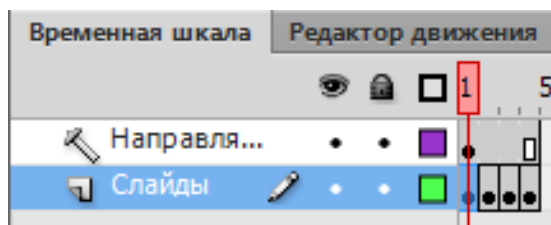


Рис. 2.8.4. Титульний кадр презентації



Рис. 2.8.5. Один із кадрів презентації, що згенеровано майстром

Презентації Flash мають велику кількість переваг порівняно із презентаціями інших форматів.

Доступність. Презентація, конвертована у Flash, може легко переглядатися в будь-якому Інтернет-броузері: Internet Explorer, Mozilla, Netscape, Safari, Opera і багато інших. Оскільки більше 98 % всіх броузерів вже встановили плагін Flash, користувачу не буде потрібно встановлювати ніякого додаткового програмного забезпечення. Також презентацію легко передавати через Інтернет-сайти, форуми, e-Mail і т. д.

Сумісність. Flash-презентація може бути переглянута на будь-якому комп'ютері (Макінтош, Лінукс, Solaris, HP, SGI, комп'ютери PDA (кишенькові ПК)).

Компактний розмір файлу. Використання Flash може зменшити розмір презентації до 10 %.

Інтеграція звуку. Файли Flash можуть також інтегрувати аудіозапис, включаючи музику й голос за кадром.

Роздільна здатність презентації при показі. Файли Flash можуть програватися фактично за будь-якої роздільної здатності й/або розмірі екрана.

Потокова технологія. Файли Flash використовують потокову технологію, що дозволяє онлайн-глядачам починати перегляд презентації не чекаючи, поки вся презентація завантажиться на їхній комп'ютер.

Інтерактивні елементи керування. Файли Flash можуть містити додаткові елементи керування, які дозволяють користувачеві управляти презентацією й вводити додаткову інформацію.

Безпека. Flash дозволяє підтримувати більш високий рівень контролю.

Тому актуальним завданням є перетворення презентацій з інших форматів (наприклад Powerpoint) в Adobe Flash.

Це підтверджує не тільки кількість розроблених і запропонованих на ринку програм для конвертації, але й висока якість їхньої реалізації. Існують як самостійні програми, наприклад – ActiveSWF FlashSpring, – <http://www.activeswf.com/powerpoint-to-flash.html>, що забезпечують високу якість трансформації, так і відмінні плагіни для PowerPoint – PresentationPro PowerConverter. <http://tinyurl.com/2h3ro>.

Контрольні запитання

1. Опишіть суть, призначення й основні характеристики презентацій.
2. Які види презентацій вам відомі?
3. Опишіть процедуру створення презентації в Adobe Flash.
4. Опишіть переваги використання Flash-презентацій. Яким чином можна перетворити презентацію з інших форматів у Adobe Flash?

Практична складова "Динамічні текстові поля. Активна кнопка"

Компетентності, що отримуються завдяки цій складовій частині:

- вміння створювати динамічні текстові поля в 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 із фіксованою шириною і висотою. Flash дозволяє використовувати в динамічних текстових полях елементи оформлення HTML.

Для створення динамічного текстового поля необхідно виконати такі дії:

1. У новому документі створити текстове поле типу "Динамический текст" (рис. 2.8.6).

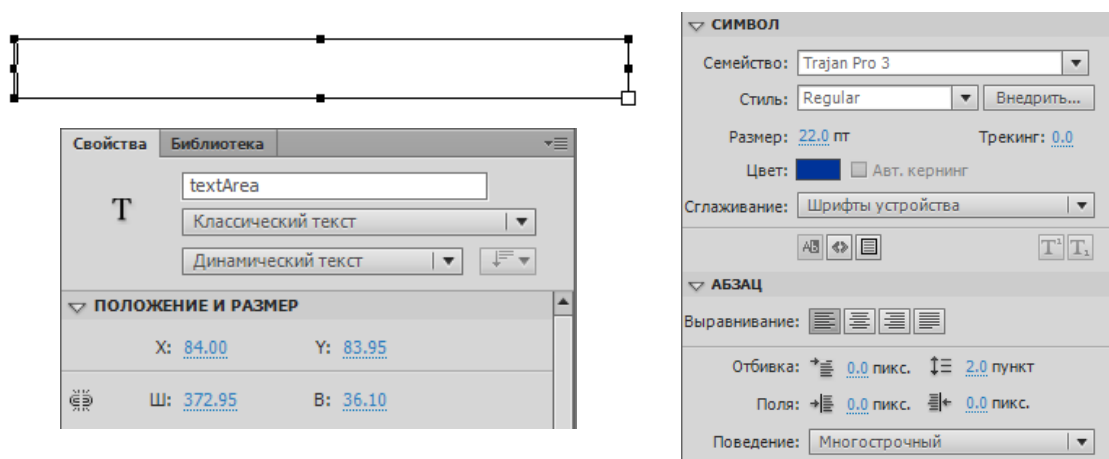


Рис. 2.8.6. Створення динамічного текстового поля

2. Задати властивості для текстового поля (див. рис. 2.8.6):

- Поведение: Многострочный.
- Сглаживание – шрифты устройства.
- Имя экземпляра: textArea.

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

```
text_btn.addEventListener(MouseEvent.CLICK, txt_show);  
function txt_show(Event:MouseEvent):void {  
    textArea.text='Комп'ютерна анімація';  
}
```

4. На тому ж шарі, на якому міститься динамічне текстове поле, намалювати будь-який об'єкт та перетворити його у символ "Кнопка". Задати для кнопки ім'я символу text_btn.

5. Запустити фільм (Ctrl + Enter).

6. Спробувати додати ще одну кнопку, яка б змінювала текст для динамічного поля на "Динамічні тестові поля".

Активна кнопка (кнопка, що з'являється)

Кнопкам у Flash властива велика кількість подій, які можна аналізувати і реагувати на них, використовуючи інструменти класу MouseEvent, зв'язавши з кнопкою прослуховувач подій EventListener.

Використовуються такі константи для аналізу подій миші CLICK, CONTEXT_MENU, DOUBLE_CLICK, MIDDLE_CLICK, MIDDLE_MOUSE_DOWN, MIDDLE_MOUSE_UP, MOUSE_DOWN, MOUSE_MOVE, MOUSE_OUT, MOUSE_OVER, MOUSE_UP, MOUSE_WHEEL, RIGHT_CLICK, RIGHT_MOUSE_DOWN, RIGHT_MOUSE_UP, ROLL_OUT, ROLL_OVER.

Слід розглянути приклад включення та виключення анімації після натискання на відповідні кнопки. Для того щоб забезпечити реакцію кнопки на подію наведення або натискання, необхідно виконати такі дії:

1. Створити 3 шари. Перший – шар для анімації руху, який безпосередньо буде включено і виключено. Другий – шар, на якому буде міститися програмний код. Третій – це шар, на якому будуть міститися кнопки.

2. На першому шарі – створити довільну анімацію руху, наприклад на 25 кадрів.

3. На третьому шарі, намалювати два об'єкти, кожний з яких перетворити на відповідну кнопку – програвання та зупинка. Для цього необхідно виконати команду **Преобразовать в символ – "Кнопка"**.

4. Для кнопки, що відповідатиме за програвання необхідно вказати ім'я екземпляра **play_btn**, а для кнопки, що відповідає за зупинку – **stop_btn**.

5. На перший кадр шару, який призначений для розміщення коду, необхідно додати такі рядки:

```
stop();
play_btn.addEventListener(MouseEvent.CLICK, playF);
stop_btn.addEventListener(MouseEvent.CLICK, stopF);
function playF(Event:MouseEvent):void {
    play();
}
function stopF(Event:MouseEvent):void {
    stop();
}
```

6. Спробувати протестувати кліп, а після тестування проекспериментувати з подіями, на які реагуватиме анімація. Зробіть так, щоб кнопки реагували не на натискання лівою кнопкою миші, а на проведення курсору, прокручування колеса та інше).

Завдання

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

2. Додати до індивідуального мультимедійного проекту динамічне текстове поле.

2.9. Тестування й оптимізація кліпу для різних способів подання

Мета: ознайомлення з існуючими методами тестування й оптимізації анімаційних кліпів для різних призначень.

Ключові слова: тестування, оптимізація, мультимедійні видання, анімація для Інтернет, анімація на локальних носіях.

2.9.1. Тестування й налагодження розроблених анімаційних продуктів

У цьому параграфі йдеться про тестування й оптимізацію фільмів, установку параметрів публікації.

! **Оптимізація** – це процедура задання відповідних параметрів експорту імітованій растровій графіці та звуку, вилучення непотрібних екземплярів із бібліотеки, перевірка усіх сценаріїв, убудованих у фільм, щодо помилок та їх виправлення.

Хоча Flash може програвати кліпи безпосередньо в робочому середовищі, багато анімаційних і інтерактивних функцій не будуть працювати, доки кліп не експортований в остаточний формат. Використовуючи команди меню Control, можна експортувати кліп у форматі Flash і відразу ж відтворити його в новому вікні. Експортований кліп буде використовувати опції, установлені в діалоговому вікні "Параметри публікації" (Publish Settings).

Цим же вікном можна скористатися для тестування швидкості завантаження кліпу з мережі на комп'ютер клієнта. Для тестування всіх інтерактивних функцій і анімації виконується така послідовність дій: **Управление - > Тестировать ролик (в Flash Professional или в Браузере)** або **Управление - > Тестировать монтажный кадр**. Flash створить кліп у SWF-форматі, відкриє його в окремому вікні й відтворить. SWF-файл запишеться в ту ж директорію, що й файл FLA. Це не тимчасовий файл.

Для публікації Flash-фільму можна встановити такі параметри у вікні **Файл - > Параметры публикации**:

- **Опубликовать** – вкладка призначена для вибору форматів файлів, які повинні бути створені при публікації фільму; додатково можна ввести власне ім'я для кожного файла, що генерується;
- **Flash** – вкладка забезпечує установку параметрів експорту файлу FLA у формат SWF;
- **HTML** – елементи цієї вкладки забезпечують вибір деяких додаткових параметрів розміщення фільму на HTML-сторінці.

Можна забезпечити збереження встановлених параметрів без ініціалізації команди **Publish**; уведені значення будуть використовуватися за замовчуванням для всіх наступних публікацій.

Можна вибрати "**Цель**" – версію плеєра або іншого програмного засобу, у якому буде відтворюватися отриманий проект, а також "**Сценарий**" – версія мовою програмування **Action Script**.

Вкладка **Flash** містить наступні елементи:

- Прапорець **Составить отчет о размере** – створити звіт про розміри кадрів фільму, звіт зберігається в окремому текстовому файлі;
- **Пропустить инструкции Trace** – скасувати дію Trace, що використовується при налагодженні сценаріїв на Action Script;
- **Защита от импорта** – захищає від імпорту SWF-файла з вікна плеєра (або броузера) у формат FLA, також можна користуватися паролем;

- **Сжать ролик** – додатковий стиск фільму при експорті у SWF;
- **Качество JPEG** – якість JPEG-зображень, що використовуються у фільмі.

Вкладка HTML містить такі елементи:

- Список **Шаблон** – дозволяє вибрати тип шаблону, відповідно до якого буде згенерований HTML-документ; список містить такі варіанти:
 - **Только Flash** – шаблон забезпечує включення в HTML-документ тільки тих тегів, які необхідні для завантаження фільму (тобто тегів <OBJECT> і <EMBED>);
 - **Гиперкарта** (сенсорна карта) – додається тег з параметрами, що дозволяють створювати клієнтську сенсорну карту; при виборі даного шаблону необхідно на вкладці Formats дозволити генерацію файлу в одному із графічних форматів;
 - Список **Размер** (розміри) разом із полями Ширина і Высота дозволяє вибрати спосіб виміру й розміри вікна;
 - Список **Оконный режим** (режим вікна) призначений для вибору варіанта подання вмісту фільму на веб-сторінці [6].

2.9.2. Оптимізація й доробка анімації для використання в мережі Інтернет

У цьому параграфі йдеться про основні характеристики Flash-документів для Інтернет і способи їхньої оптимізації.

Чим більше обсяг swf-файла, тим довше він завантажується в браузер користувача й тим повільніше відтворюється кліп. Однак є багато способів оптимізувати кліп, а отже й зменшити обсяг swf-файла:

- якщо графічний елемент використовується в кліпі більше одного разу, необхідно зробити із нього символ;
- перш ніж імпортувати растрову графіку, треба визначити чи можливо перетворити її у векторну;
- при імпорті векторної графіки необхідно переконатися, що в ній не існує схованих ліній або об'єктів;
- треба використовувати кадровану анімацію замість покрокової;
- без потреби не треба перетворювати текстовий об'єкт у графічний вигляд;
- при виборі шрифтового оформлення кліпу треба віддавати перевагу шрифтам, що мають більш просте накреслення символів;

- для оптимізації звуку необхідно використовувати MP3 компресію. Так само на розмір звукових даних впливають частота дискретизації (sample rate) і кількість каналів (стерео, моно);

- треба обмежити використання спеціальних типів ліній, таких, як пунктир, нечіткі лінії, крапки й т. д.;

- графічні об'єкти зі складною границею можна оптимізувати за допомогою опції меню: **Модифікація - > Фигура/Оптимізація**;

- слід ощадливо застосовувати градієнти. Заповнення області градієнтом збільшує розмір файлу, порівняно із заливкою основним кольором;

- використання прозорості повинне бути виправдане, тому що цей ефект сповільнює відтворення;

- одночасно з експортом кліпу треба виводити звіт і уважно вивчати його для виявлення "вузьких" місць [1].

Також Flash Professional містить у своєму складі та дозволяє підключати інші програмні продукти компанії Adobe для використання потужних інструментів, які дозволяють експортувати отримані анімаційні кліпи як в формат HTML5, так і в формат Adobe AIR.

Контрольні запитання

1. У чому полягає оптимізація анімаційного фільму у Flash?
2. Які параметри публікації можна встановити у Flash?
3. Як можна оптимізувати анімацію для Інтернет?

2.10. Action Script. Основні поняття мови програмування

Мета: вивчення теоретичних основ мови програмування Action Script та ознайомлення з її об'єктною моделлю.

Ключові слова: програмування, Action Script, змінні, масиви, функції, параметри, методи.

2.10.1. Базові поняття Action Script

У цьому параграфі йдеться про основні типи даних, які використовує Flash, способи визначення масивів.

! **Action Script** – це вбудована у Adobe Flash об'єктно-орієнтована мова програмування, за синтаксисом схожа з JavaScript, яка додає можливість значно збільшити інструментальний набір

розробника анімації, зменшити трудозатрати та розробляти анімаційні проекти більш високого рівня. Action Script дозволяє генерувати об'єкти динамічно, використовувати програмну анімацію та створювати інтерактивні кліпи з можливістю управління та зворотним обміном даними.

Action Script є мовою програмування, що використовується у середовищах виконання Adobe ® Flash ® Player і Adobe ® AIR ™. Вона забезпечує інтерактивність, обробку даних і багато інших можливостей у контенті Adobe Flash, Flex і AIR, а також у додатках.

Action Script 3.0 перевершує можливості створення сценаріїв попередніх версій Action Script. Спеціально розроблена, щоб полегшити створення складних додатків із великим набором даних і об'єктно-орієнтованим, багаторазово використовуваним програмним кодом. Хоча Action Script 3.0 не обов'язкова для вмісту, вона відкриває можливість підвищення продуктивності, доступної тільки з AVM2, нової віртуальної машини. Код Action Script 3.0 може виконуватися до десяти разів швидше, ніж код колишньої версії Action Script.

Старіша версія віртуальної машини Action Script, AVM1, виконує код Action Script 1.0 і Action Script 2.0. AVM1 підтримується в Flash Player 9 і 10 для зворотної сумісності з існуючим і старим вмістом.

Основні типи даних, які використовує Action Script.

Примітивне значення – це значення, яке відноситься до одного з таких типів даних: Boolean, Int, Number, String і Uint. Працювати з примітивними значеннями зазвичай виходить швидше, ніж зі складними, оскільки Action Script зберігає примітивні значення спеціальним способом, роблячи можливою оптимізацію швидкодії і використання пам'яті.

Складне значення – це альтернатива примітивним значенням. До типів даних, що визначають набори складних значень, відносяться Array, Date, Error, Function, RegExp, XML і XMLList.

- **String** (рядок) – будь-який фрагмент даних, який інтерпретується як звичайний текст. Про те, що обраний текст буде використовуватись як рядок, свідчить наявність лапок.

- **Number** (число) – ціле число чи число з "плаваючою" комою.

- **Boolean** (логічний) – логічний тип використовується для результатів порівняння та деяких функцій. Єдиними значеннями логічного типу даних є **true** (істина) та **false** (брехня).

- **Null** – цей тип даних дозволяє вказати на відсутність даних.
- **Undefined** (невизначений) – дозволяє вказати на відсутність типу даних у змінній, тобто на те, що її значення не визначено.
- **Array** (масив) – це список даних (тобто спеціальний тип даних); самі дані всередині масиву можуть мати інший тип, наприклад, String чи Number, чи їх комбінацію.

Найпоширенішим типом масиву Action Script є індексний масив. У індексному масиві кожен елемент зберігається в пронумерованому слоті (так званому індексі). При зверненні до елементів цей номер використовується як адреса. Індексні масиви підходять для більшості потреб програмування. Клас Array – це один загальний клас, що використовується для подання індексного масиву. Екземпляр Array може містити значення даних будь-якого типу.

! ***Macue (Array)** – це тип даних, який містить декілька фрагментів інформації. Розміщений в масиві фрагмент даних називається елементом (element) масиву.*

Способи визначення масивів [7].

Створити об'єкт типу Array можна за допомогою **new**:

Наприклад:

```
var names:Array = new Array();
var names:Array = new Array(3);
var names:Array = new Array("John", "Jane", "David");
```

Для визначення кількості елементів масиву існує спеціальна властивість `length` (довжина). Це єдина властивість, яку має масив. Таким чином, щоб визначити довжину будь-якого масиву, достатньо вказати його ім'я та властивість `length` (через крапку).

```
trace(names.length);
```

Для прикладів, що наведені, результатами буде відповідно 0, 3 та 3.

Для відтворення елементів масиву можна використовувати відповідні індекси, наприклад:

```
trace(names[0]); // output: John
trace(names[1]); // output: Jane
trace(names[2]); // output: David
```

Три методи класів Array і Vector: push (), unshift () і splice () – дозволяють вставляти елементи в індексний масив. Метод push () додає один або декілька елементів у кінець масиву. Іншими словами, у останнього елемента, вставленого в масив за допомогою методу push (), буде найбільший індекс. Метод unshift () вставляє один або більше елементів у початок масиву, при цьому перший індекс завжди дорівнює 0. Метод splice () вставляє будь-яке число елементів за заданим індексом в масиві.

Для вилучення елементів із масивів існує три методи класів Array і Vector: pop (), shift () і splice (). Метод pop () видаляє елемент з кінця масиву. Інакше кажучи, він видаляє елемент із найбільшим індексом. Метод shift () видаляє елемент із початку масиву, тобто кожен раз видаляється елемент з індексом 0. Метод splice (), який також може використовуватися для вставки елементів, видаляє довільне число елементів, починаючи з індексу, заданого першим із переданих методу аргументів.

Клас Vector.

Інший тип індексного масиву, що підтримується в Action Script 3.0, представлений класом Vector. Екземпляр Vector – це типізований масив, тобто всі елементи в екземплярі Vector завжди мають однаковий тип даних.

Коли оголошується змінна Vector або створюється екземпляр Vector, необхідно чітко вказати тип даних для об'єктів, які може містити Vector. Зазначений тип даних називається базовим типом об'єкта Vector. Щоб створити екземпляр Vector, потрібно викликати конструктор Vector.<T> (). Також для створення об'єкта Vector можна викликати глобальну функцію Vector.<T> (). Ця функція перетворює зазначений об'єкт в екземпляр Vector.

```
var v:Vector.<String>;  
v = new Vector.<String>();
```

```
Використання конструктора Vector.<T>()  
var names:Vector.<String> = new Vector.<String>();  
trace(names.length); // output: 0
```

2.10.2. Об'єктна модель Action Script

У цьому параграфі йдеться про об'єктно-орієнтовану модель Action Script, основні поняття, визначення класів.

! **Клас** – це абстрактне уявлення об'єкта. У класі зберігається інформація про типи даних, які може містити об'єкт, і про його можливі види поведінки.

Корисність такої абстракції може бути не такою помітною при створенні невеликих сценаріїв, що містять всього кілька взаємодіючих між собою об'єктів. Однак із розширенням області дії програми і збільшення числа керованих об'єктів стає очевидно, що класи забезпечують більш повний контроль над створенням і взаємодією об'єктів.

Функції.

! **Функція** – це підпрограма, блок коду, який реалізує набір операцій, призначена для виконання конкретного завдання.

Функції можуть використовуватися скрізь, де є ділянки коду, що повторюється.

Для створення функцій використовується слово **function**, яке супроводжується іменем функції. Синтаксис оголошення функції виглядає таким чином:

```
function myFunction ()  
{  
  //код, що виконується в функції  
}
```

Для виконання функції (виклику чи звернення до функції) необхідно ввести ім'я функції, що супроводжується круглими дужками, які містять її параметри, та закінчити строку крапкою з комою.

```
myFunction();
```

Оскільки функція використовує тільки ту інформацію, якою вона володіє, необхідність скоригувати її залежно від ситуації можна задовольнити завдяки параметрам.

! **Параметри** – це дані, які потрібно передати функції для використання.

Параметри в функції діють аналогічно до змінних у Action Script, при необхідності їх вміст може бути змінено динамічно. Параметри розташовують у круглих дужках поряд з іменем функції.

```
function myFunction (parameters)
{
  //код, що виконується в функції
}
```

При оголошенні функції можна одразу задати тип параметрів:

```
function myFunction (parameter: type) :void
{
  //код, що виконується в функції
}
```

Таким чином, якщо передати до функції параметр іншого типу, то при перевірці коду у вікні Output (вивід) виникне повідомлення про помилку.

Для повернення функцією значення використовується оператор **return**. Він виконує два завдання: по-перше, коли інтерпретатор досягає оператора **return**, він завершує виконання функції, по-друге, оператор **return** повертає значення, яке було привласнено йому в результаті виконання виразу (expression), хоча сам вираз не обов'язковий.

```
function myFunction (parameters)
{
  //код, що виконується в функції
  return expression;
}
```

! **Вкладені функції (nested function)** – це досить потужний інструмент багатократного виконання коду при збереженні можливості застосування результату по-різному у кожній функції.

Вкладена функція доступна тільки в рамках батьківської функції доти, доки посилання на цю функцію передаються в зовнішній програмний код. Якщо вкладені функції передаються в зовнішній програмний код, вони віддаються як закриття функцій, і це означає, що функція зберігає всі визначення, що були в області дії під час її визначення.

! **Вбудовані функції (built-in function)** – власні функції Flash.

Функції як методи.

! **Методи (method)** – це функції, які входять до складу визначення класу. Коли створюється екземпляр класу, з ним пов'язується

метод. На відміну від функції, оголошеної за межами класу, метод не можна використовувати окремо від екземпляра, з яким він пов'язаний.

Методи визначаються з використанням ключового слова `function`. Як і властивості класу, будь-який їх атрибут можна застосувати до методів, включаючи особисті, захищені, загальнодоступні, внутрішні, статичні або користувальницькі простори імен. Інструкцію `function` можна використовувати таким чином:

```
public function sampleFunction (): String {}
```

Також можна використовувати змінну, якій присвоюється вираз функції, як показано далі.

```
public var sampleFunction: Function = function () {}
```

Контрольні запитання

1. Які основні типи даних використовує Action Script?
2. Як у Action Script відбувається побудова масивів?
3. Що таке методи у Action Script?
4. Що таке функція, які особливості написання функцій у Action Script?
5. Що таке вбудовані функції, які вони бувають?
6. Як відбувається побудова класів у Action Script?

Практична складова "Програмування на Action Script"

Компетентності, що отримуються завдяки цій складовій частині:

- вміння створювати анімацію символів за допомогою мови програмування Action Script;
- програмне змінення властивостей символів-контейнерів.

Допоміжний матеріал

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

Для дублювання символів-контейнерів використовуватися створений у бібліотеці символ, що пов'язаний з базовим класом `flash.display.MovieClip`, також оператор класу `New` та метод класу `addChild()`.

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

1. Створити символ MovieClip (фрагмент ролика) у вигляді багатогранника або зірки, або іншої геометричної фігури довільного кольору і розміру.

2. Знайти у Movie Clip і зробити розкадровування руху для об'єкта (який спочатку необхідно перетворити у символ "Графика"). Розкадровування руху можна зробити таким чином, щоб у першому ключовому кадрі символ був маленького розміру та непрозорим. У останньому кадрі цієї анімації робити символ значно більших розмірів, перемістити його у нижній лівий кут анімаційної області та показник прозорості зробити таким, щоб символ став практично абсолютно прозорим. Доцільно також налаштувати траєкторію руху та зробити для символу один оберт протягом анімації (рис. 2.10.1).

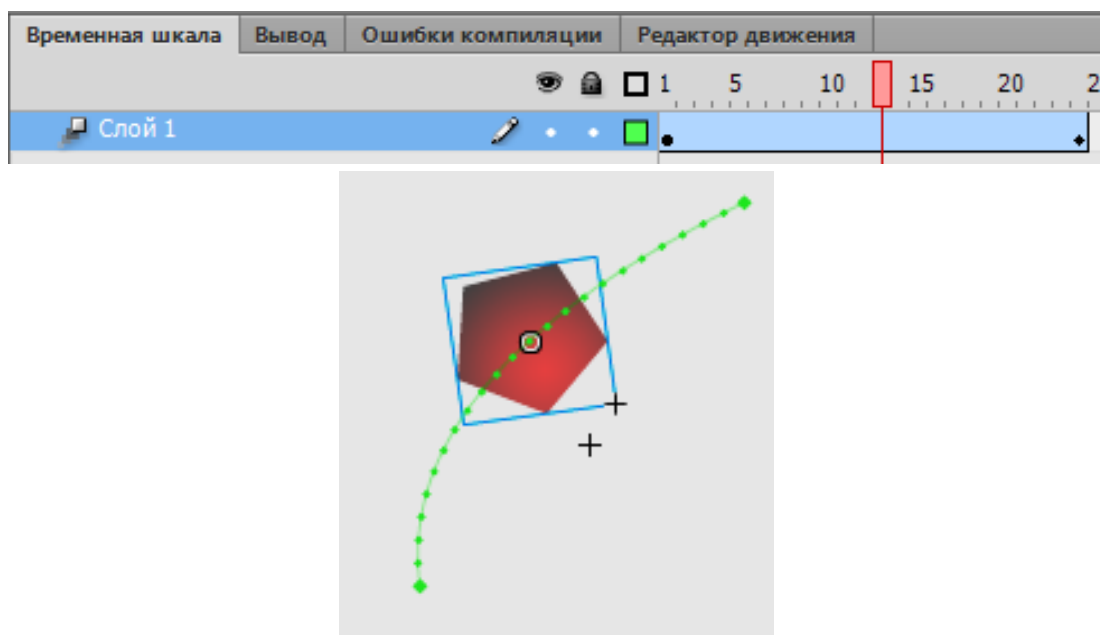


Рис. 2.10.1. Анімація усередені символа – контейнера

3. Вийти на основну сцену.

4. Знайти у бібліотеці символ та зробити для нього зв'язування з Action Script – задати ім'я для символу та класу `star_mc`. Слід звернути увагу, що у цьому випадку обов'язковим є задання імені класу `star_mc`, задавати таке ж ім'я для символу – необов'язково (рис. 2.10.2 та 2.10.3). Більш того, повернувшись на основну сцену, можна видалити Movie Clip зі сцени, залишивши його при цьому тільки у бібліотеці.

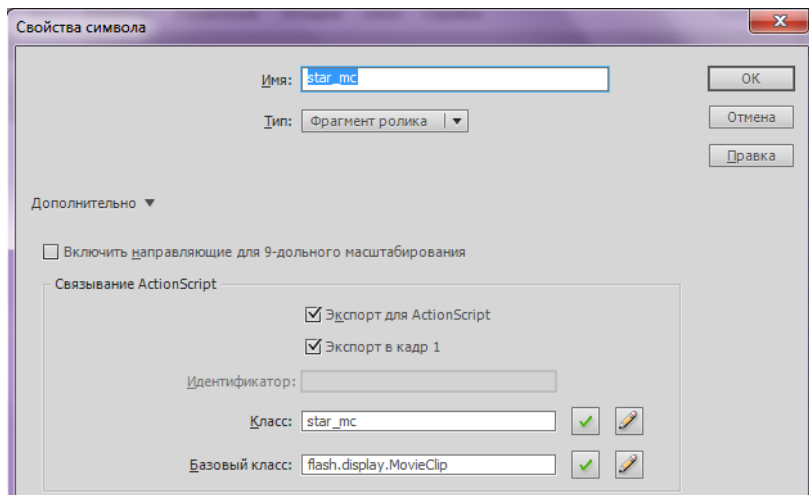


Рис. 2.10.2. Зв'язування символу з класом Action Script

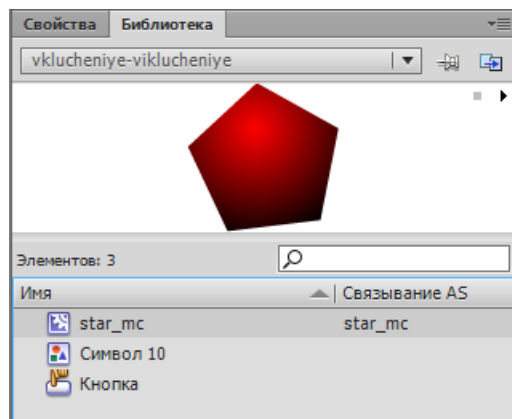


Рис. 2.10.3. Символ у бібліотеці

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

```
var star_mc2:star_mc;
```

```
duplbtn.addEventListener(MouseEvent.CLICK, duplicateStar);
```

```
function duplicateStar(e:MouseEvent):void{
    // створюємо новий об'єкт на основі класу
    star_mc2 = new star_mc();
    // додаємо на сцену
    addChild(star_mc2);
    star_mc2.x=Math.random()*300
    star_mc2.y=Math.random()*300;
}
```

6. Повернутися на основну сцену та з довільно намальованої фігури створити кнопку та задати для неї ім'я duplbtn рис. 2.10.4.

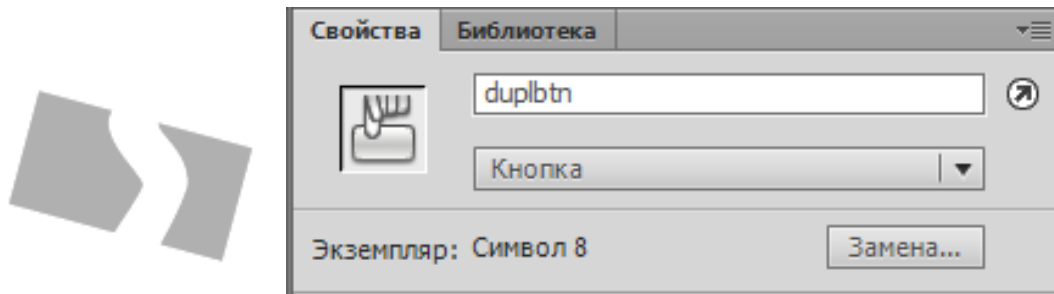


Рис. 2.10.4. Кнопка duplbtn

Крім положення символів, можна змінювати інші властивості, наприклад прозорість, розміри, нахил, відимість. Далі наведено приклади управління такими властивостями:

```
star_mc2.alpha = 0.3;  
star_mc2.visible = false;  
star_mc2.rotation = 20;  
star_mc2.width = 200;  
star_mc2.height = 100.
```

Видалення символів-контейнерів відбувається за допомогою команди:

```
removeChild(star_mc2).
```

Слід розглянути приклад програмного коду, що дозволяє перетягувати об'єкти:

1. Для цього необхідно намалювати два об'єкти – коло та квадрат, та перетворити їх на символи типу "Кнопка". Необхідно задати ім'я до цих символів – відповідно square та circle.

2. До першого кадру редактора "Действия" необхідно додати такий код:

```
import flash.display.DisplayObject;  
import flash.events.MouseEvent;  
var offsetX:Number;  
var offsetY:Number;  
var draggedObject:DisplayObject;  
// Ця функція викликається при натисканні кнопки миші.
```

```

function startDragging(event:MouseEvent):void
{
// запам'ятовує, який об'єкт тягнути
draggedObject = DisplayObject(event.target);
// Записує зміщення між тим, де курсор був, коли
// кнопка миші була натиснута і x, y координати
// об'єкта, що переміщується коли була натиснута кнопка миші.
offsetX = event.stageX – draggedObject.x;
offsetY = event.stageY – draggedObject.y;
// Переміщує вибраний об'єкт в початок списку відображення
stage.addChild(draggedObject);
// Каже Flash Playerу, щоб почав прослуховування події MouseEvent
stage.addEventListener(MouseEvent.CLICK, dragObject);
}
// Ця функція викликається, коли користувач відпускає кнопку миші.
function stopDragging(event:MouseEvent):void
{
// Tell Flash Player to stop listening for the MouseEvent event.
stage.removeEventListener(MouseEvent.CLICK, dragObject);
}
// Ця функція викликається кожного разу, коли миша рухається,
// При цьому, кнопка миші натиснута.
function dragObject(event:MouseEvent):void
{
// Перемістити об'єкт, що перетягується в положення курсору, зберігаючи
// зміщення між розташуванням курсора і розташуванням
// об'єкта, що переміщується.
draggedObject.x = event.stageX – offsetX;
draggedObject.y = event.stageY – offsetY;
// Доручити Flash Player, щоб оновив екран після цієї події.
event.updateAfterEvent();
}
circle.addEventListener(MouseEvent.CLICK, startDragging);
circle.addEventListener(MouseEvent.CLICK, stopDragging);
square.addEventListener(MouseEvent.CLICK, startDragging);
square.addEventListener(MouseEvent.CLICK, stopDragging);

```

3. У разі виконання цього ролику можна буде перетягувати обидва об'єкти, при чому той об'єкт, яким користувалися пізніше, буде знаходитися поверх іншого.

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. Зробити об'єкт, що перетягується анімованим
4. Пересуваючи об'єкт робочою областю, визначити його координати.


2.11. Смарткліпи

Мета: ознайомлення з теоретичними основами створення інтерактивних кліпів та додатків, вивчення елементів керування.

Ключові слова: смарткліп, інтерактивність, панель Actions, управління, додатки, елементи керування.

2.11.1. Створення інтерактивних кліпів і додатків

У цьому параграфі йдеться про поняття інтерактивних фільмів (смарткліпів), панель Actions.

 У **простій анімації** Flash відтворює сцени й кадри фільму послідовно один за іншим. В інтерактивному фільмі глядачі за допомогою клавіатури або миші можуть переходити до різних частин фільму, переміщати об'єкти, вводити інформацію до анкет й виконувати багато інших дій, активно беручи участь у фільмі.

Інтерактивні ефекти у фільмі створюються на основі процедур – наборів інструкцій, написаних у Action Script, які запускаються при настанні певної події й, по суті, є макрокомандами. Події, які можуть активізувати процедуру, – це або досягнення відтворюючою голівкою певного кадру, або якась дія користувача: натискання кнопки або клавіші на клавіатурі. Приклад інтерактивного кліпу подано на рис. 2.11.1.

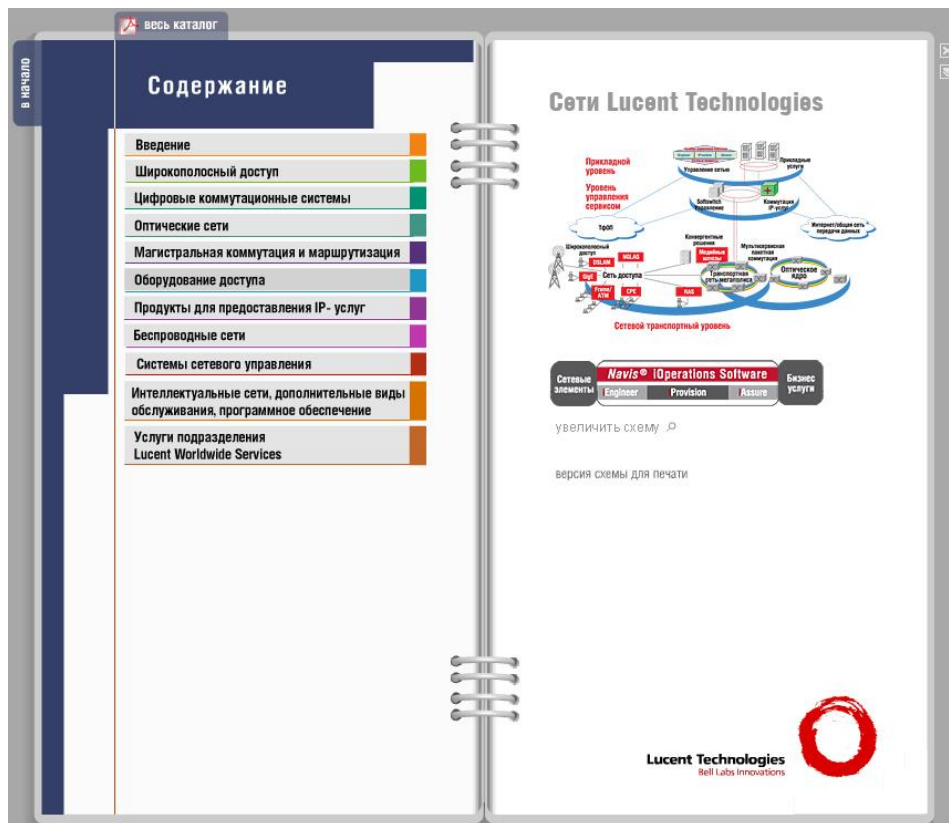


Рис. 2.11.1. Приклад інтерактивного кліпу

Сценарії можуть складатися з окремої інструкції, яка описує таку дію, як, наприклад, зупинка відтворення або з ряду інструкцій тип перевірки виконання деякої умови й запуск деяких процедур.

Панель Actions (Действия або Дії). Одним із центральних елементів середовища є панель Actions. Панель складається з декількох елементів: область уведення коду, область команд (дерево команд, згрупованих у папки за тематиками), нижче дерева команд розташована область, що містить ієрархічне дерево з посиланнями на елементи фільму, які містять код. Остання область призначена для швидкого переходу між ділянками коду, особливістю є те, що сам елемент, що містить код, не буде активним. Для полегшення написання коду є підсвічування коду, що змінює колір слів, що входять у словник Action Script. Завдяки даній особливості можна завжди візуально перевірити правильність написання команд. Через те, що імена команд і ідентифікатори є чутливими до регістра, дана можливість є гарантом правильності написання слів.

Якщо у попередніх версіях Action Script можна було розташовувати код у сценаріях кадрів, кліпів та кнопок, то у Action Script 3.0, враховуючи орієнтацію на об'єктне програмування код може міститися або у першому

кадрі (не рекомендується розміщувати код в інших кадрах) або необхідно створювати та потім підключати відповідні класи. Сценарії кадрів розташовуються на ключових кадрах, при додаванні коду в ключовий кадр на тимчасовій шкалі в полі кадру грецька буква альфа. Код можна використовувати у будь-якому ключовому кадрі, при цьому при переході на інший кадр система не забуває те, що було на попередньому. Прийнято відводити під кадри цілий шар і розташовувати його з самого верху. Це викликано тим, що при завантаженні кліпу спочатку завантажуються нижні шари, і відповідно для запобігання помилки, пов'язаної з відсутністю елемента, використовуваного в коді, код повинен завантажуватися в останню чергу [9].

Для управління інтерактивним додатком код Action Script можна поділити на три типи (операції):

- зупинити показчик відтворення у кадрі:

```
stop();
```

- слідкувати за подією, щоб запустити відтворення кліпу при натисканні глядачем кнопки:

```
function startMovie(event:MouseEvent):void
```

```
{
```

```
  this.play();
```

```
}
```

```
playButton.addEventListener(MouseEvent.CLICK, startMovie);
```

- слідкувати за подією, щоб відправити браузер на відповідну URL-адресу при натисканні глядачем кнопки [9].

```
function gotoAuthorPage(event:MouseEvent):void
```

```
{
```

```
  var targetURL:URLRequest = new URLRequest("http://example.com/");
```

```
  navigateToURL(targetURL);
```

```
}
```

```
homeButton.addEventListener(MouseEvent.CLICK, gotoAuthorPage).
```

2.11.2. Елементи керування інтерактивним фільмом, засоби збирання й подання інформації

У цьому параграфі йдеться про компоненти користувальницького інтерфейсу та компоненти управління фільмом.

У складі Flash містяться три типи компонентів (рис. 2.11.1): компоненти флекс, компоненти користувальницького інтерфейсу та компоненти керування відеооб'єктами.

До компонентів користувальницького інтерфейсу відносять:

- CheckBox – прапорець;
- ComboBox – комбінований список (тобто список, що розкривається, з можливістю редагування пунктів);
- List – список;
- Button – кнопка;
- RadioButton – перемикач;
- UIScrollBar – смуга прокручування;
- ScrollPane – вікно;
- а також TextArea, NumericStepper, TextInput, ColorPicker, TileList, ProgressBar, UILoader, DataGrid, Label, Slider.

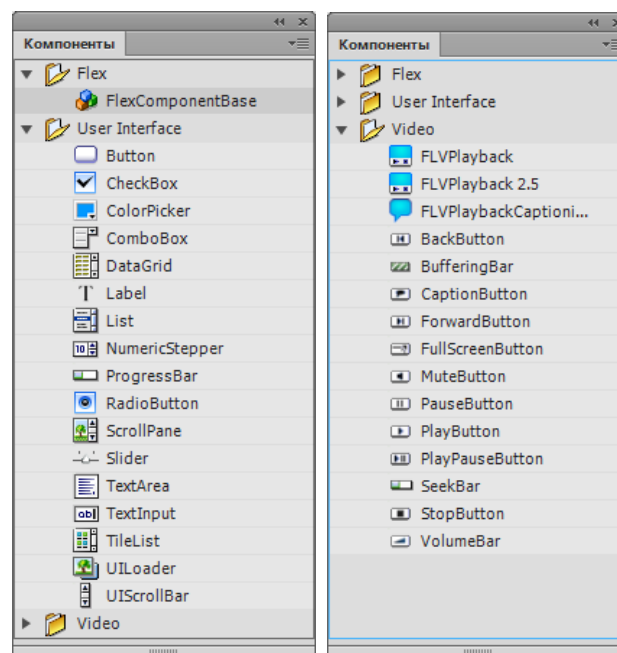


Рис. 2.11.1. Вікно компонентів Flash

Найпоширеніший варіант застосування компонентів – це створення на їхній основі аналогу форми, що міститься на веб-сторінці. Разом із тим, на основі компонентів можна створити для Flash-фільму "традиційний" для Windows-додатків користувальницький інтерфейс. У цьому випадку робота користувача з фільмом буде мало чим відрізнятися від роботи зі звичайним додатком.

При використанні компонентів як елементів форми Flash забезпечує:

- пересилання уведених даних на сервер за заданою адресою;
- обробку даних на клієнтській машині засобами мови Action Script.

Для установки вихідних параметрів екземпляра компонента, включеного у фільм, доцільно скористатися інспектором властивостей. Його формат для компонентів трохи відрізняється від використовуваного для інших елементів фільму. Основна відмінність полягає в тому, що панель інспектора властивостей компонента має дві вкладки.

- **Properties** (властивості), за допомогою елементів якої встановлюються властивості компонента як екземпляра символу-кліпу; ця вкладка однакова для всіх компонентів;

- **Parameters** (параметри), призначена для установки параметрів компонент як елемента користувальницького інтерфейсу; для кожного типу компонента набір параметрів різний.

Для **прапорця (Checkbox)** можуть бути задані такі параметри (рис. 2.11.2):

- **Label** – текстова мітка;
- **Initial Value** – вихідний стан прапорця; параметр приймає значення **true (1)** або **false (0)**;
- **Label Placement** (розташування текстової мітки) – визначає положення текстової мітки щодо зображення прапорця;
- **Change Handler** (змінити оброблювач) – ім'я функції, що викликається при зміні стану прапорця; функція, що викликається повинна бути визначена в сценарії, пов'язаному з тією же тимчасовою діаграмою, до якої ставиться прапорець; даний параметр є необов'язковим і застосовується тільки в тому випадку, якщо необхідно активізувати деяку дію, коли користувач змінює стан прапорця.

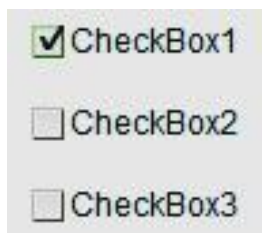


Рис. 2.11.2. Прапорець (CheckBox)

Усі компоненти інтерфейсу у Flash мають свої відмінності у параметрах та властивостях, але буде нескладно більш детально ознайомитися з використанням інших компонентів, використовуючи додатковий довідковий матеріал.

Крім компонентів користувальницького інтерфейсу компоненти Flash Action Script 3.0 також містять у своєму складі такі компоненти і класи:

Компонент FLVPlayback (fl.video.FLVPlayback), який є компонентом на базі SWC.

Компонент FLVPlayback дає можливість додати у додаток Flash відеопрогравач для відтворення потокових файлів, що послідовно завантажуються з використанням протоколу HTTP, з сервісу Adobe® Flash® Video Streaming Service (FVSS) або з сервера Adobe Macromedia® Flash® Media Server (FMS).

Компоненти користувальницького інтерфейсу для відтворення FLV-файлів на базі FLA, що працюють з компонентом FLVPlayback версій Action Script 2.0 та Action Script 3.0.

Компонент FLVPlayback Captioning, що забезпечує введення кодованих субтитрів між кадрами для компонента FLVPlayback.

Також можна завантажувати додаткові компоненти з сервера Adobe Exchange за адресою www.adobe.com/go/flash_exchange_ru [7].

Контрольні запитання

1. Що таке інтерактивність?
2. Як у Flash реалізований принцип інтерактивності?
3. Яке призначення панелі Actions. Її складові?
4. Які основні характеристики елементів керування інтерактивним фільмом?
5. Які основні типи компонентів вам відомі? Перерахуйте найбільш затребувані?

Практична складова "Робота з компонентами Adobe Flash"

Компетентності, що отримуються завдяки цій складовій частині:

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

Допоміжний матеріал

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

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

1. На робочий стіл помістити такі компоненти (рис. 2.11.3):

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

2. Задати властивості для компонентів (рис. 2.11.3):

TextArea: Instance Name – txt; editable – вимкнути;

ComboBox: Instance Name – aCb;

Button: Instance Name – btn; label – ОК;

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

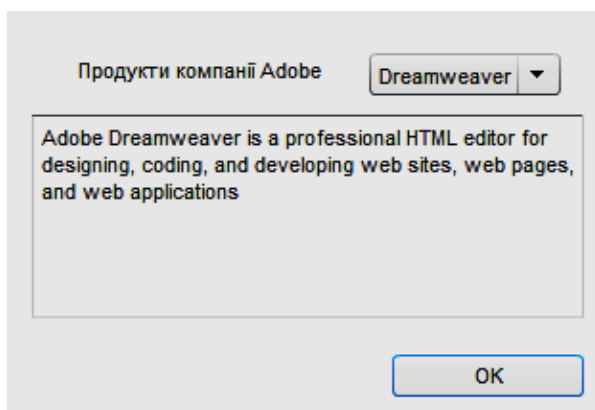


Рис. 2.11.3. Динамічна форма з розміщеними на ній компонентами

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

// Імпорт відповідного класу та заповнення списку ComboBox значеннями:

```
import fl.data.DataProvider;
```

```
aCb.dataProvider = new DataProvider( ['Flash', 'Dreamweaver']);
```

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

```
//Функція відображення тексту згідно з вибраним пунктом списку
```

```
function doSubmit(e:MouseEvent) {
```

```
  switch (aCb.selectedIndex) {
```

```
    case 0 :
```

```
      txt.text='Welcome to Adobe Flash. Flash provides everything you need  
to create and deliver rich web content and powerful applications';
```

```
    case 1 :
```

```
      txt.text='Adobe Dreamweaver is a professional HTML editor for  
designing, coding, and developing web sites, web pages, and web  
applications';
```

```
  }
```

```
}
```

4. Зберегти проект та спробувати його протестувати.

Склад компонентів динамічної форми можна змінити залежно від необхідності. Спробувати додати компонент "Слайдер", за допомогою якого можна управляти іншим елементом. Для цього потрібно виконати такі дії:

- На попередній формі намалювати коло та перетворити його у символ, наприклад "Кнопка". Додати до форми компонент "Слайдер" (рис. 2.11.4).

- Для кнопки задати ім'я символу btnc, а для слайдера sld.

- На першому кадрі у редакторі "Действия" додати такий код:

```
import fl.controls.Slider;
```

```
import fl.events.SliderEvent.
```

- Додати обробку зміни слайдера та функцію, що буде реагувати на його зміни, а саме – збільшувати або зменшувати розміри кнопки btnc.

```
sld.addEventListener(SliderEvent.CHANGE, changeHandler);
```

```
function changeHandler(event:SliderEvent):void {
```

```
  btnc.width=sld.value*50;
```

```
  btnc.height=sld.value*50;
```

```
}
```

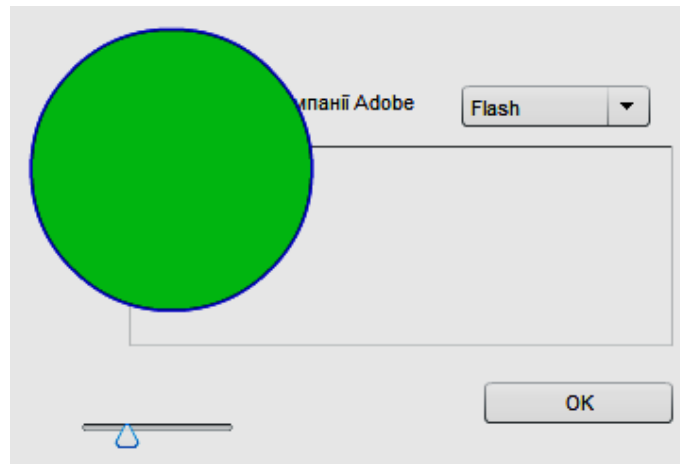


Рис. 2.11.4. Динамічна форма зі слайдером та кнопкою, що змінює розмір

- Протестувати кліп та спробувати змінити положення слайдера в обидві сторони.

Завдання

1. Створити форми переглядання інформації за допомогою компонентів TextArea, ComboBox, Button, Label.
2. Додати компонент Slider.
3. Додати компоненти до форми за власним бажанням, наприклад ColorPicker та NumericStepper.
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.

2.12. Програмне малювання у Adobe Flash

Мета: вивчення методів програмного малювання та створення програмної анімації за допомогою мови програмування Action Script.

Ключові слова: програмне малювання, згенерувати динамічно, програмна анімація, масштабування, градієнтна заливка, алгоритм.

2.12.1. Огляд методів програмного малювання

У цьому параграфі йдеться про основні методи програмного малювання, малювання прямих і кривих Безьє другого порядку (квадратичних сплайнів), суцільні й градієнтні заливки.



Програмне малювання використовує об'єкти, які було намальовано не безпосередньо вручну, а згенеровано динамічно. Для програмної анімації досить часто застосовується зміна координат об'єкта, його розміри, кут повороту, динамічна зміна прозорості.

У Action Script 3.0 основні дії щодо програмного малювання виконуються за допомогою трьох спеціальних об'єктів. Кожен з об'єктів Shape, Sprite і MovieClip наділений властивістю graphics, що є екземпляром класу Graphics. Клас Graphics включає властивості та методи, призначені для створення ліній, заливок і фігур.

Якщо для створення вмісту потрібно використовувати як полотно тільки екранний об'єкт, можна використовувати екземпляр Shape. Екземпляр Shape краще за інших екранних об'єктів підходить для малювання, оскільки він не має службових даних, пов'язаних із додатковими функціональними можливостями в класах Sprite і MovieClip. Якщо потрібно, щоб екранний об'єкт, на якому можна створювати графічний вміст, також містив і інші екранні об'єкти, можна використовувати екземпляр Sprite. Якщо екранний об'єкт повинен бути присутнім на множині кадрів, то краще використовувати MovieClip.

Усі малюнки, що створюються за допомогою екземпляра Graphics, базуються на основному малюнку з лініями і кривими. У результаті цього всі малюнки Action Script повинні бути створені із застосуванням тих же операцій:

- визначення стилів ліній і заливок;
- визначення початкового положення малюнків;
- створення ліній, кривих і фігур (за бажанням переміщення точки малюнка);
- при необхідності завершення створення заливки.

Слід розглянути основні методи класу Graphics.

beginFill(color:uint, alpha:Number = 1.0):void

Задає просту заливку суцільним кольором для використання при наступних викликах інших методів об'єкта Graphics (наприклад, LineTo () або DrawCircle ()) при малюванні.

beginGradientFill(type:String, colors:Array, alphas:Array, ratios:Array, matrix:Matrix = null, spreadMethod:String = "pad", interpolationMethod:String = "rgb", focalPointRatio:Number = 0):void

Задає заливку градієнтом для використання при наступних викликах інших методів Graphics (наприклад, LineTo () або DrawCircle ()) об'єкта.

beginShaderFill(shader:Shader, matrix:Matrix = null):void

Задає затінення для використання при наступних викликах інших методів об'єкта Graphics (наприклад, LineTo () або DrawCircle ()).

clear():void

Видаляє графіку, промальовану в даному об'єкті "Graphics", і скидає параметри заливки і стилю лінії. Задає затінення для використання при наступних викликах інших методів об'єкта Graphics (наприклад, LineTo () або DrawCircle ()).

copyFrom(sourceGraphics:Graphics):void

Копіює всі команди малювання вихідного об'єкта Graphics в об'єкт Graphics, що викликає.

curveTo(controlX:Number, controlY:Number, anchorX:Number, anchorY:Number):void

Малює криву Безьє другого порядку з використанням поточного стилю лінії від поточного положення малювання до точки з координатами (anchorX, anchorY) і з застосуванням опорної точки з координатами (controlX, controlY).

drawCircle(x:Number, y:Number, radius:Number):void

Малює коло.

drawEllipse(x:Number, y:Number, width:Number, height:Number):void

Малює еліпс.

drawGraphicsData(graphicsData:Vector.<IGraphicsData>):void

Відправляє послідовність екземплярів IGraphicsData для малювання.

drawPath(commands:Vector.<int>, data:Vector.<Number>, winding:String = "evenOdd"):void

Відправляє послідовність команд для малювання.

drawRect(x:Number, y:Number, width:Number, height:Number):void

Малює прямокутник.

drawRoundRect(x:Number, y:Number, width:Number, height:Number, ellipseWidth:Number, ellipseHeight:Number = NaN):void

Малює прямокутний примітив з округленими кутами.

drawTriangles(vertices:Vector.<Number>, indices:Vector.<int> = null, uvData:Vector.<Number> = null, culling:String = "none"):void

Візуалізує набір трикутників, щоб надати растровому зображенню тривимірний вигляд.

endFill():void

Застосовує заливку до ліній і кривих, доданих з моменту останнього виклику методу `beginFill()`, `beginGradientFill()` або `beginBitmapFill()`.

lineBitmapStyle(bitmap:BitmapData, matrix:Matrix = null, repeat:Boolean = true, smooth:Boolean = false):void

Вказує растрове зображення штриха лінії для використання при малюванні ліній.

lineGradientStyle(type:String, colors:Array, alphas:Array, ratios:Array, matrix:Matrix = null, spreadMethod:String = "pad", interpolationMethod:String = "rgb", focalPointRatio:Number = 0):void

Вказує градієнт для використання при малюванні ліній.

lineShaderStyle(shader:Shader, matrix:Matrix = null):void

Вказує затінення штриха лінії для використання при малюванні ліній.

lineStyle(thickness:Number = NaN, color:uint = 0, alpha:Number = 1.0, pixelHinting:Boolean = false, scaleMode:String = "normal", caps:String = null, joints:String = null, miterLimit:Number = 3):void

Вказує стиль лінії, який буде використовуватися при наступних викликах таких методів `Graphics`, як `LineTo()` або `DrawCircle()`.

lineTo(x:Number, y:Number):void

Малює лінію з використанням поточного стилю лінії від поточного положення інструменту малювання до точки з координатами (x, y); потім поточним положенням стає точка (x, y).

moveTo(x:Number, y:Number):void

Переміщує поточне положення інструменту малювання в точку з координатами (x, y).

Методи **moveTo()** і **lineTo()** використовуються для малювання ліній. **moveTo()** встановлює позицію "олівця" (курсору) у позиції x, y. **lineTo()** проводить лінію від позиції, установленної **moveTo()**, до позиції зазначеної параметрами x, y у самій функції **lineTo()**.

Але, перш ніж малювати лінії цими методами, потрібно встановити властивості лінії методом **lineStyle()**.

```
myShape.graphics.lineStyle(2, 0x990000, .75);
```

Для зафарбовування намальованих фігур використовуються методи **beginFill()** і **endFill()**.

beginFill() включає режим зафарбовування. Після його використання всі намальовані контури будуть зафарбовуватися.


```

        myShape.graphics.beginFill(0x00FF00);
endFill() відключає режим зафарбовування.
Приклад коду, що малює червоний трикутник із використанням beginFill():
var triangleHeight:uint = 100;
var triangle:Shape = new Shape();

// червоний трикутник стартує з точки 0, 0
triangle.graphics.beginFill(0xFF0000);
triangle.graphics.moveTo(triangleHeight / 2, 0);
triangle.graphics.lineTo(triangleHeight, triangleHeight);
triangle.graphics.lineTo(0, triangleHeight);
triangle.graphics.lineTo(triangleHeight / 2, 0);
triangle.graphics.endFill()
addChild(triangle);

```

2.12.2. Методи програмного малювання кривих Безьє, градієнтних ліній та заливок

У цьому параграфі йдеться про малювання прямих і кривих Безьє другого порядку (квадратичних сплайнів).

Слід розглянути метод, що малює криві: **curveTo**(control, control, anchor, anchor), описання параметрів вже розглянуто. Наступний приклад створює червону скруглену фігуру розміром 100 пікселів та використовує для цього метод **curveTo**.

Приклад:

```

var size:uint = 100;
var roundObject:Shape = new Shape();
// червона округлена форма
roundObject.graphics.beginFill(0xFF0000);
roundObject.graphics.moveTo(size / 2, 0);
roundObject.graphics.curveTo(size, 0, size, size / 2);
roundObject.graphics.curveTo(size, size, size / 2, size);
roundObject.graphics.curveTo(0, size, 0, size / 2);
roundObject.graphics.curveTo(0, 0, size / 2, 0);
addChild(roundObject);

```

Створення фігур за допомогою вбудованих методів

Для зручності в Action Script 3.0 є методи для створення таких звичайних фігур, як кола, овали, прямокутники і трикутники із закругленими кутами. До таких методів належать методи класу graphics: drawCircle(), drawEllipse(), drawRect(), drawRoundRect().

Їх можна використовувати замість методів `LineTo()` і `curveTo()`. Слід зазначити, що перед викликом цих методів потрібно вказувати стилі ліній і заливок. У наступному прикладі показано створення червоних, зелених і синіх квадратів висотою і шириною 100 пікселів. Цей код використовує метод `drawRect()` і додатково вказує, що колір заливки має 50 % альфа-каналу (0.5).

Приклад:

```
var squareSize:uint = 100;
var square:Shape = new Shape();
square.graphics.beginFill(0xFF0000, 0.5);
square.graphics.drawRect(0, 0, squareSize, squareSize);
square.graphics.beginFill(0x00FF00, 0.5);
square.graphics.drawRect(200, 0, squareSize, squareSize);
square.graphics.beginFill(0x0000FF, 0.5);
square.graphics.drawRect(400, 0, squareSize, squareSize);
square.graphics.endFill();
this.addChild(square);
```

Градiєнтні лінії та заливка. Для реалізації градієнтних ліній та заливок використовуються методи **lineGradientStyle()** та **beginGradientFill()** і **endFill()**. Обидва методи мають однакові параметри, що були наведені. Але, зважаючи на тривіальність використаного підходу щодо градієнтного зафарбовування, слід розглянути більш детально кожен з чотирьох параметрів.

Перші чотири параметри є обов'язковими: тип, кольори, альфа-канали та коефіцієнти. Інші чотири параметра є необов'язковими, але можуть бути корисними при додатковому налаштуванні.

Перший параметр визначає тип градієнта, що створюється. Допустимими значеннями є такі: `GradientFill.LINEAR` або `GradientFill.RADIAL`.

Другий параметр визначає масив значень кольорів, призначених для використання. У лінійному градієнті кольори будуть організовані зліва направо. У радіальному градієнті вони будуть організовані з середини на зовні. Порядок кольорів масиву задає послідовність, за якою кольори будуть подані в градієнті.

Третій параметр визначає значення альфапрозорості відповідних кольорів попереднього параметра.

Четвертий параметр визначає коефіцієнти або ступінь насиченості кожного кольору у градієнті. Діапазон допустимих значень – від 0 до 255.

Ці значення не позначають довжину або ширину, швидше розташування всередині градієнта; 0 відповідає початку градієнта, 255 – кінцю градієнта. Масив коефіцієнтів повинен зростати послідовно і мати ту ж кількість елементів, яку було задано в другому і третьому параметрах для масивів альфа каналів і кольорів.

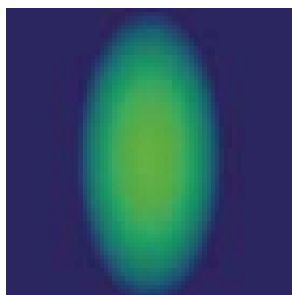
Незважаючи на факультативність п'ятого параметра, – матриці перетворення, він часто використовується, оскільки забезпечує простий і ефективний спосіб управління зовнішнім виглядом градієнта. Цей параметр містить екземпляр Matrix. Найпростішим способом створення об'єкта Matrix для градієнта є використання методу createGradientBox() класу Matrix, який створює матрицю, що використовується для визначення градієнта. За допомогою параметрів, переданих у метод createGradientBox(), задаються масштаб, поворот і положення градієнта.

Метод createGradientBox() приймає такі параметри:

- ширина градієнта: ширина (у пікселях), на яку поширюватиметься градієнт;
- висота градієнта: висота (у пікселях), на яку поширюватиметься градієнт;
- поворот градієнта: поворот (у радіанах), який буде застосований до градієнта;
- переміщення по горизонталі – як далеко (у пікселях) по горизонталі буде переміщений градієнт;
- переміщення по вертикалі – як далеко (у пікселях) по вертикалі буде переміщений градієнт.

Далі наведено приклад, що створює радіальну градієнтну заливку області, використовуючи синій та зелений колір та з визначенням усіх параметрів.

Приклад:



```
import flash.display.Shape;  
import flash.display.GradientType;
```

```

import flash.geom.Matrix;
var type:String = GradientType.RADIAL;
var colors:Array = [0x00FF00, 0x000088];
var alphas:Array = [1, 1];
var ratios:Array = [0, 255];
var spreadMethod:String = SpreadMethod.PAD;
var interp:String = InterpolationMethod.LINEAR_RGB;
var focalPtRatio:Number = 0;
var matrix:Matrix = new Matrix();
var boxWidth:Number = 50;
var boxHeight:Number = 100;
var boxRotation:Number = Math.PI/2; // 90°
var tx:Number = 25;
var ty:Number = 0;
matrix.createGradientBox(boxWidth, boxHeight, boxRotation, tx, ty);
var square:Shape = new Shape;
square.graphics.beginGradientFill(type, colors, alphas, ratios, matrix,
spreadMethod, interp, focalPtRatio);
square.graphics.drawRect(0, 0, 100, 100);
addChild(square).

```

Слід звернути увагу, що ширина і висота заливки градієнта виділяється за допомогою ширини і висоти матриці градієнта, а не ширини або висоти, створених за допомогою об'єкта Graphics. У наступному прикладі показано візуальну відмінність між розміром матриці градієнта і розміром самого малюнка.

Приклад:

```

var myShape:Shape = new Shape();
var gradientBoxMatrix:Matrix = new Matrix();
gradientBoxMatrix.createGradientBox(100, 40, 0, 0, 0);
myShape.graphics.beginGradientFill(GradientType.LINEAR,
[0xFF0000, 0x00FF00, 0x0000FF], [1, 1, 1], [0, 128, 255], gradientBoxMatrix);
myShape.graphics.drawRect(0, 0, 50, 40);
myShape.graphics.drawRect(0, 50, 100, 40);
myShape.graphics.drawRect(0, 100, 150, 40);
myShape.graphics.endFill();
this.addChild(myShape).

```

Метод `lineGradientStyle()` функціонує схоже з методом `beginGradientFill()`, але, крім визначення градієнта, перед початком малювання необхідно вказати товщину обведення за допомогою методу `LineStyle()`.

2.12.3. Алгоритмізація й реалізація проекту, що розробляється

У цьому параграфі йдеться про основні аспекти алгоритмізації й реалізації проекту, що розробляється.

У процесі створення будь-якої програми, чи то невелика навчальна програма або серйозний проект, над яким працюють десятки програмістів, можна виділити декілька етапів. Витрати праці й часу на виконання цих етапів різні, також ці витрати залежать від складності програми. Основними етапами розробки програмного продукту є:

1. Постановка задачі.
2. Розробка математичної моделі задачі.
3. Етап алгоритмізації.
4. Етап програмування.
5. Налаштування й аналіз результатів.

! ***Алгоритм** – це певним чином організована послідовність дій, кінцеве число кроків, що призводить до вирішення задачі.*

Властивості алгоритмів: визначеність, дискретність, цілеспрямованість, кінцевість, масовість.

Вимоги до виконання алгоритму:

1. Дії в алгоритмі виконуються в порядку їхнього запису.
2. Не можна міняти місцями ніякі дві дії алгоритму.
3. Не закінчивши однієї дії, не можна переходити до наступної.

Відмінними рисами реалізації алгоритмізації Flash-проекту є приділення уваги до визначення таких питань:

1. Розробка сценарію майбутнього проекту.
2. Спосіб організації інформації у проекті (з файлів якого типу він буде складатися і як вони будуть розташовані на носії).
3. Вимоги до мети, стилю та дизайну проекту.
4. З яких шарів та сцен буде складатися кожний файл проекту?
5. Які об'єкти, актори та інші анімаційні компоненти будуть використовуватися?
6. З яких компонентів буде складатися інтерфейс проекту?
7. Як буде організована навігація?
8. Яким чином будуть подані графічні, аудіо- та відеоелементи?
9. У якому форматі буде опублікований проект?

Контрольні запитання

1. Які основні методи програмного малювання вам відомі?
2. У чому полягають методи програмної заливки об'єктів у Flash?
3. Які методи малювання геометричних фігур вам відомі?
4. Якими способами можна програмно застосовувати градієнтну заливку у Flash?
5. У чому полягає особливість алгоритмізації проектів у Flash?

Практична складова "Програмне малювання"

Компетентності, що отримуються 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

Для отримання практичних навичок щодо програмного малювання необхідно скористатися наведеними прикладами, за допомогою програмного коду спробувати використати такі методи (рис. 2.12.1):

- задання способів заливки геометричних фігур та задання стилів ліній з використанням методів `beginFill()`; `endFill()`; `lineStyle()`;
- креслення прямих ліній та кривих Безьє з використанням методів `moveTo()`; `lineTo()`; `curveTo()`;
- використання методів `DrawCircle()`, `DrawEllipse()`, `DrawRect()` для малювання геометричних фігур;
- створення градієнтної заливки та використання градієнтів для стилю ліній з використанням методів `BeginGradientFill()`, `LineStyle()`.



а



б

Рис. 2.12.1. Фігури, створені методами програмного малювання



в



г

Закінчення рис. 2.12.1.

Завдання

У завданні будуть наведені вимоги до використання методів програмної анімації, для отримання фігур, які аналогічні рис. 2.12.1.

1. За допомогою методу малювання прямими відрізками зобразити такі геометричні фігури: фіолетовий шестигранник, помаранчеву напівпрозору трапецію з блакитним контуром, червону зірку з синім контуром довільної товщини, а також аналогічну рис. 2.12.1а.

2. За допомогою методу малювання кривими зобразити фігуру, аналогічну рис. 2.12.1б.

3. Використовуючи методи зображення прямокутників та кругів, зобразити фігуру як на рис. 2.12.1в.

4. Використовуючи градієнтну заливку та методи зображення прямокутників та еліпсів, зобразити фігуру, як на рис. 2.12.1г.


2.13. Створення анімованих документів у середовищі Autodesk 3DSMax

Мета: ознайомлення з основами побудови тривимірної графіки та анімації, вивчення теоретичних основ побудови фільмів у середовищі Autodesk 3DSMax.

Ключові слова: тривимірна графіка та анімація, Autodesk 3DSMax, клонування, матеріали, рендерінг.

2.13.1. Користувальницький інтерфейс 3DSMax

У цьому параграфі йдеться про структуру основного вікна 3DSMax, основні об'єкти 3DSMax, основні команди й операції з об'єктами.

 **Autodesk 3DSMax** – є потужним візуальним середовищем для створення тривимірної графіки, анімації та тривимірного моделю-

вання, що дозволяє використовувати текстурування, освітлення, імітацію фізичних реакцій, використання візуальних ефектів щодо візуалізації.

Структура основного вікна. Вікно 3DSMax містить три вікна проєкцій, у кожному з яких показана тривимірна сцена зі своєї точки (рис. 2.13.1).

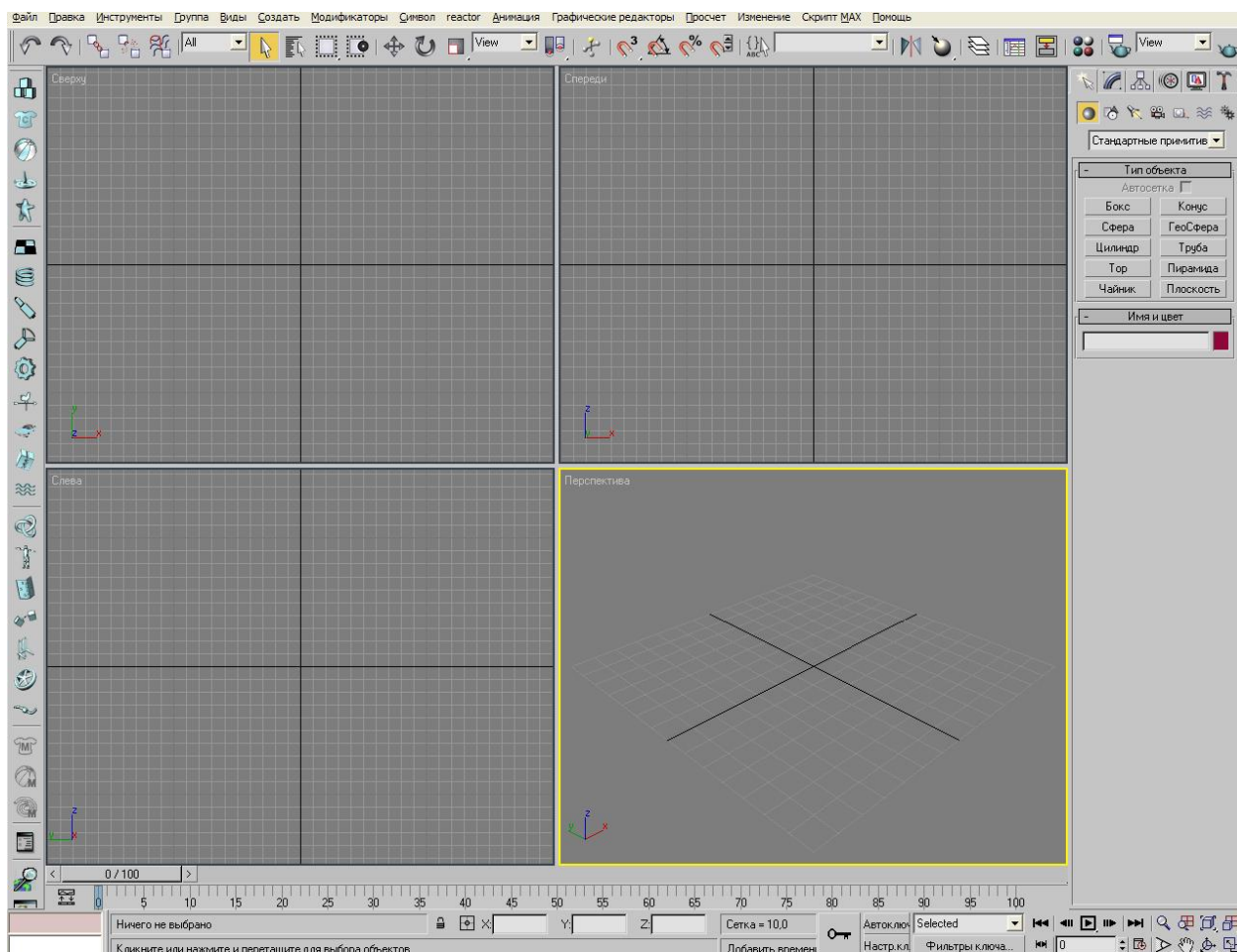


Рис. 2.13.1. Вікно 3DSMax

У верхній частині вікна програми розташоване головне меню, а під ним – головна панель інструментів **Main Toolbar** (основна панель інструментів). Пункти меню частково повторюють інструменти й команди основної панелі інструментів, а також панелі **Command Panel** (командна панель).

Командна панель містить шість вкладок: Create (створення), Modify (зміна), Hierarchy (ієрархія), Motion (рух), Display (відображення), Utilities (утиліти). Найбільш часто використовуються вкладки Create (створення) і Modify (зміна).

Вкладка **Create** (створення) слугує для створення основних (примітиви, криві та ін.) і допоміжних (джерела світла, віртуальні камери, об'ємні деформації й ін.) об'єктів сцени. Вкладка **Modify** (зміна) дозволяє змінювати параметри виділеного об'єкта сцени. Також з її допомогою виділеному об'єкту можна призначити *модифікатор* – певну дію, що деформує об'єкт [2].

У нижній частині вікна 3DS Max розташована шкала анімації, під нею – координати перетворень, рядок стану, а також кнопки керування анімацією й положенням об'єктів у вікнах проекцій.

Основні об'єкти 3DSMax. Об'єкти в 3DSmax можна розділити на декілька категорій: Geometry (геометрія), Shapes (форми), Lights (джерела світла), Cameras (камери), Helpers (допоміжні об'єкти), Space Wraps (об'ємні деформації), Systems (додаткові інструменти), Об'єкти групи Geometry (геометрія):

- **Primitives** (примітиви) – базові об'єкти, що становлять геометричні тіла. Цей вид складається із двох основних наборів – **Standard** (стандартного) і **Extended** (розширеного). Основна особливість цього типу об'єктів – мінімальний набір параметрів, що дозволяє "на льоту" міняти структуру й ступінь деталізації об'єкта.

- **Compound Objects** (складові об'єкти) – структуровані складні об'єкти, що використовують математичні, логічні й поведінкові алгоритми. Для створення таких об'єктів необхідні набори більш простих двовимірних і тривимірних примітивів Max.

- **Patch Grids** (клаптеві сітки) – площинні об'єкти, що складаються з **Patch** (поверхні шматка) і **Vertices** (ґрати керуючих вершин). Вершини впливають на поверхню як вузли деформацій і плавно змінюють форму.

- **NURBS-Surfaces** (тривимірні NURBS-поверхні) – складні тривимірні об'єкти, описувані як тривимірні сплайни. Для них характерна відсутність різких ("кутових") змін форми й профілю, а також дуже гнучка структура геометрії.

- **Particle Systems** (системи часток) – процедурні об'єкти, створювані як область простору, що імітує багато реальних процесів (сніг, дощ, фонтан, дим, іскри тощо). Більшість систем часток у Max універсальні й дозволяють міняти алгоритм своєї поведінки (частота, притягання, взаємні зіткнення та ін.) у широкому діапазоні.

- **Doors** (двері) – дозволяють створити три типи дверей – **Pivot** (закріплені на осі), **Sliding** (роздвижні) і **BiFold** (складні). Перші нагадують

звичайні вхідні двері, другі – двері купе, треті – двері автобуса. Можна створювати одинарні або парні двері за допомогою параметра Double Doors (подвійні дверцята), регулювати розмір дверної коробки – параметри Width Frame (ширина рами) і Dept Frame (глибина рами), самих об'єктів – Height (висота), Width (ширина), Depth (глибина) і навіть товщину скла – Glass Thickness (товщина скла).

- **Windows** (вікна) – дозволяє додати в сцену 6 типів вікон: Sliding (роздвижні), Pivoted (закріплені на осі), Awning (навісні), Casement (стулчасті), Projected (проектовані), Fixed (закріплені).

- **Stairs** (сходи) – в 3DSMax можна створювати чотири типи сходів: L-Type (L-образна), Straight (пряма), Spiral (гвинтова), U-Type (U-образна).

- У групі **AEC Extended** (додаткові елементи для архітектурних, інженерних і конструкторських робіт) входять об'єкти Foliage (рослинність), Railing (огородження, поруччя) і Wall (стіна). Об'єкти Railing (огородження, поруччя) і Wall (стіна) застосовуються в архітектурному моделюванні. Об'єкт Foliage (рослинність) слугує для моделювання тривимірної рослинності.

Основні команди й операції над об'єктами. Основні дії, які можна робити з об'єктами – це переміщення, масштабування, обертання, вирівнювання й клонування.

У центрі виділеного об'єкта з'являються три координатні осі – X, Y, Z, які визначають систему координат, прив'язану до об'єкта. Ці координатні осі становлять так звану **локальну систему координат об'єкта**. Крапка, з якої виходять осі локальної системи координат, називається опорною (Pivot Point).

Щоб виконати будь-яку найпростішу дію з об'єктом, при якій його положення в тривимірному просторі зміниться, необхідно викликати контекстне меню, клацнувши правою кнопкою миші на об'єкті. У меню варто вибрати одну з операцій – **Move** (переміщення), **Scale** (масштабування), **Rotate** (обертання).

Переміщення. Переміщувати об'єкт можна вздовж осі X, Y, Z або в площинах XY, YZ, XZ. Координати переміщення можна вказати вручну у вікні **Move Transform Type-In** (уведення значень переміщення), що відкривається при натисканні клавіші F12 або клацанням на значку прямокутника біля рядка **Move** (переміщення) контекстного меню.

Обертання. При виборі в контекстному меню команди **Rotate** (обертання) на місці осей системи координат об'єкта з'явиться схема-

тичне відображення можливих напрямів повороту. Якщо підвести покажчик миші до кожного з напрямів, схематична лінія підсвілюється жовтим кольором, тобто поворот буде зроблений у даному напрямі. У процесі повороту у вікні проєкцій з'являються цифри, що визначають кут повороту вздовж кожної з осей.

Масштабування. При виборі в контекстному меню команди **Scale** (масштабування) на місці осей системи координат об'єкта з'явиться схематичне відображення можливих напрямів масштабування. Якщо підвести покажчик миші до кожного з напрямів, схематична лінія підсвілюється жовтим кольором. Масштабувати об'єкт можна вздовж осі X, Y, Z, у площинах XY, YZ, XZ або одночасно у всіх напрямках.

Вирівнювання об'єктів. Щоб вирівняти один об'єкт щодо іншого, потрібно виділити перший об'єкт, виконати команду **Tools->Align** (інструменти->вирівнювання) і клацнути на другому об'єкті. На екрані з'явиться вікно, у якому необхідно вказати принцип вирівнювання.

У 3DSMax з'явилась можливість вирівнювання об'єктів, що називається Quick Align (швидке вирівнювання). За допомогою цієї команди можна вирівняти об'єкти, не викликаючи вікно Align Selection (вирівнювання виділених об'єктів). Вирівнювання виробляється по опорних точках об'єктів.

Клонування об'єктів. Щоб створити копію виділеного об'єкта у вікні проєкції, потрібно виконати команду **Edit->Clone** (редагування -> клонування). На екрані з'явиться вікно Clone Objects (клонування об'єктів). У цьому вікні можна вибрати один із трьох варіантів клонування:

- **Copy** (незалежна копія об'єкта) – створена копія не буде пов'язана з оригіналом.
- **Instance** (прив'язка) – копія буде пов'язана з вихідним об'єктом. При зміні параметрів одного з об'єктів автоматично змінюються параметри іншого.
- **Reference** (підпорядкування) – копія буде пов'язана з вихідним об'єктом. При зміні параметрів вихідного об'єкта автоматично будуть змінені параметри клонованого об'єкта, однак при зміні параметрів клонованого об'єкта вихідний об'єкт змінений не буде [2].

2.13.2. Основні характеристики анімації в 3DSMax та її експортування

У цьому параграфі йдеться про основні характеристики анімації в 3DSMax, експортування анімації.

У 3DSMax можна анімувати будь-які характеристики всіх об'єктів: примітивів, джерел світла, камер, допоміжних об'єктів і ін. Задаючи значення об'єктів у ключових кадрах, можна зробити так, щоб об'єкти переміщалися в сцені, змінювали текстуру, збільшувалися або зменшувалися в розмірах і т. д.

Найпростіший тип анімації – переміщення об'єктів у тривимірній сцені.

До створеної геометрії сцени, призначеним матеріалам і встановленим джерелам світла й знімальним камерам у середовищі 3DSMax може бути привласнена послідовність змін, що характеризують кілька базових параметрів:

Active Time Segment (активний сегмент часу) – інтервал, вимірюваний у часових одиницях, що задається Start Time (початком), End Time (кінцем) і Length (довжиною) анімації сцени.

Current Frame (поточний кадр) – стан тривимірної сцени, взятий в конкретний момент у межах активного сегмента часу.

Animation Key (ключ анімації) – значення параметра анімації, що описує стан об'єктів і властивостей сцени.

Keyframe (ключовий кадр) – кадр анімації, що містить Key (ключ) і є опорним.

Animation Controller (контролер анімації) – закон зміни анімації між ключами (або в заданому проміжку часу), що описується математичною, графічною або алгоритмічною функцією.

Animation Track (трек анімації) – тимчасова лінія, що містить ключі анімації і є інструментальною шкалою анімації об'єктів і властивостей сцени.

Animation Range (діапазон анімації) – часовий інтервал на треку анімації, що вимірюється між першим і останнім ключами.

Описані характеристики становлять основу анімації тривимірних сцен і регулюються різними елементами керування. Слід розглянути їх типи й основні параметри.

Time Bar (шкала часу). Розташовувана внизу Головного вікна 3DSMax горизонтальна панель **Time Bar** (шкала часу) (рис. 2.13.2) з рельєфною кнопкою повзунка призначена для навігації по **Active Time Segment** (активному сегменті часу) [2].

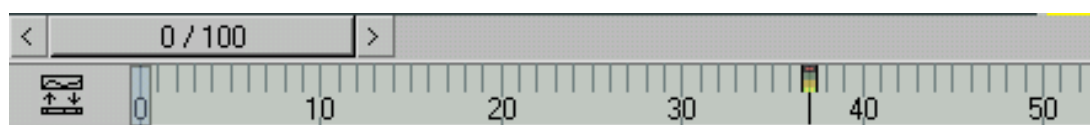


Рис. 2.13.2. Time Bar (шкала часу)

За замовчуванням тривалість, що створюється в анімації 3DSMax дорівнює 101 кадру при форматі створюваного відео NTSC (29,97 кадрів на секунду). При таких налаштуваннях можна створити анімацію тривалістю близько трьох секунд.

Для установки параметрів відображення анімації у вікні проєкції використовується вікно Time Configuration (конфігурація часу) (рис. 2.13.3), що викликається за допомогою однойменної кнопки, розташованої під кнопками керування анімацією.

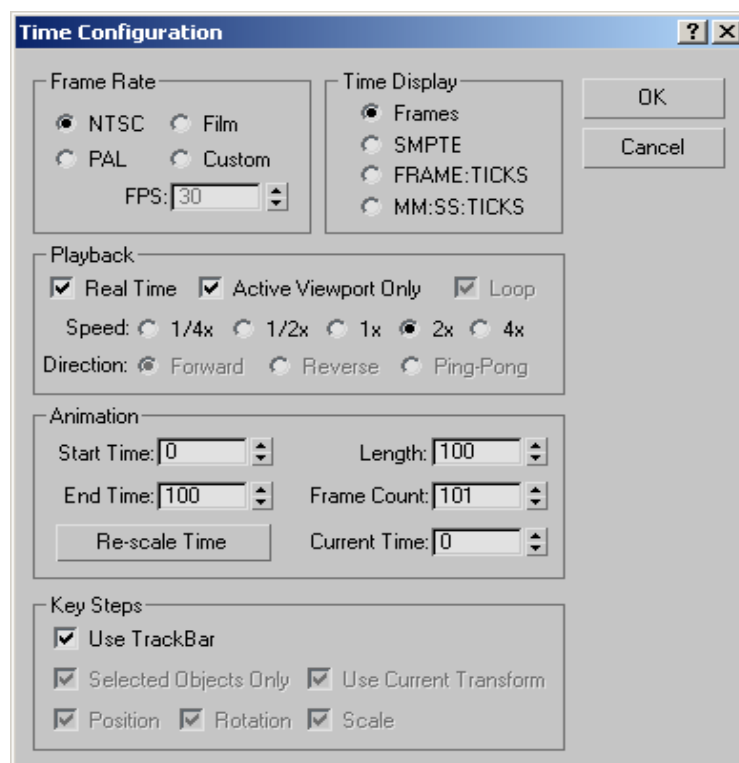


Рис. 2.13.3. Вікно Time Configuration (Конфігурація часу)

У вікні *Time Configuration* (конфігурація часу) можна встановити такі параметри: формат відео (Pal/NTSC), кількість кадрів у секунду (FPS), спосіб відображення інформації про час на повзунку анімації, час початку й кінця анімації, тривалість анімації та ін.

Експортування анімації. Для того, щоб експортувати результат роботи в зовнішній додаток, необхідно зробити рендерінг. Для цього потрібно викликати вікно з параметрами рендерінгу (**Rendering -> Render...**). У діалоговому вікні, що відкрилося (рис. 2.13.4), необхідно встановити параметри експорту.

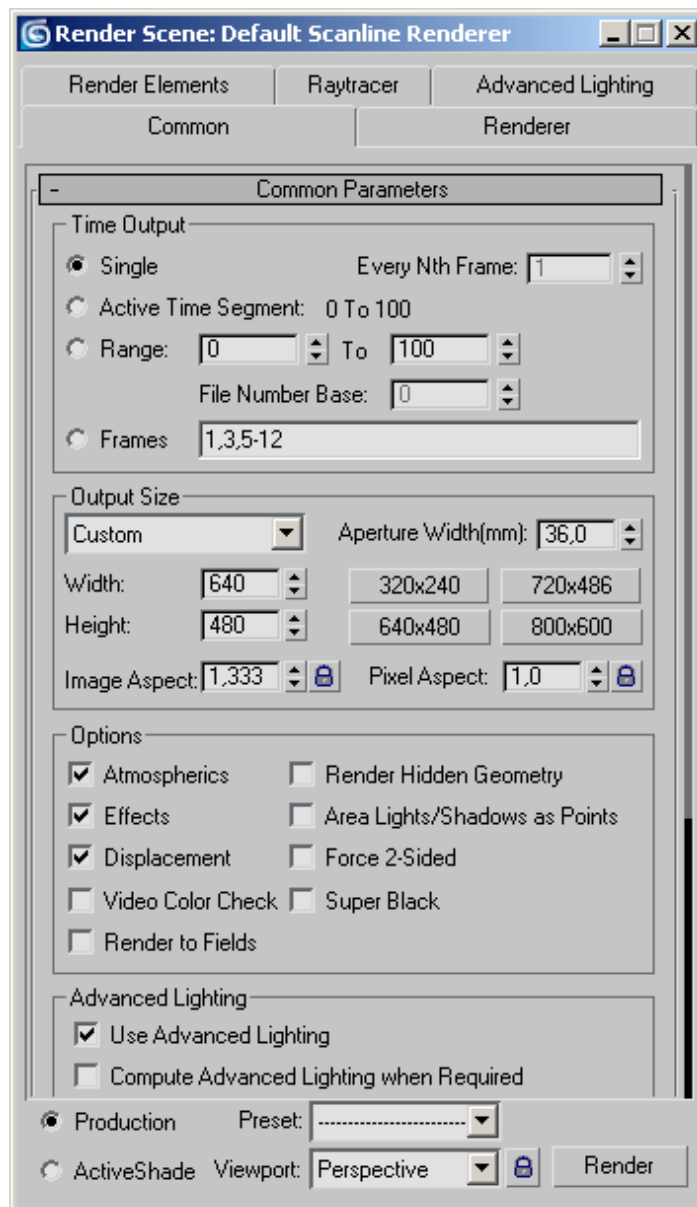


Рис. 2.13.4. **Діалогове вікно Render Scene**

Common Parameters (основні параметри):

- Група **Time Output** (часовий інтервал) містить перемикачі установки тимчасових інтервалів сцени, що візуалізується, кроку кадрів і деякі інші.
- **Single** (одиничний) – вибирає обрахування поточного кадру.
- **Active Time Segment** (активний часовий сегмент) – використовується для візуалізації повної анімації сцени.
- **Range** (діапазон) – задає парою лічильників діапазон кадрів.
- **Frames** (кадри) – список кадрів, що вказується безпосередньо, через кому або тире (наприклад, рядок "1, 3, 5 – 12" буде означати, що будуть обраховані одиничні кадри 1-й і 3-й, а також набір 8-ми кадрів із 5-го по 12-й).

- **Every Nth Frame** (кожний n-ий кадр) – дозволяє задати крок обрахування для варіантів **Active Time Segment** (активний часовий сегмент) і **Range** (діапазон) (за замовчуванням цей лічильник дорівнює 1, що означає обрахування всіх обраних кадрів).
- **File Number Base** (база нумерованих файлів) – за замовчуванням приймається рівною 0.
- Група **Output Size** (розмір зображення) – контролює розмір кадрів, що візуалізуються:
 - список, що розкривається, стандартних наборів відео-, кіно- і фотостандартів дозволяє вибрати необхідні **Width** (ширину), **Height** (висоту) і **Image Aspect** (пропорції зображення), а також **Pixel Aspect** (пропорції пікселя). Пункт **Custom** (користувальницький) зі списку дозволяє вручну встановлювати власні значення цих параметрів.
 - Група **Options** (настроювання) містить такі прапорці керування режимами візуалізації:
 - **Video Color Check** (контроль кольоровості) – відповідність до кольорів телемовлених стандартів.
 - **Force 2-Sided** (зробити двосторонніми) – відображення зворотних граней геометрії об'єктів.
 - **Atmospherics** (атмосферні ефекти) – робити візуалізацію ефектів **Environment** (оточення).
 - **Effects** (спеціальні ефекти) – робити візуалізацію ефектів однойменного типу.
 - **Super Black** (чистий чорний колір) – використовується для відеододатків.
 - **Displacement** (заміщення) – робити візуалізацію матеріалів із текстурами типу **Displacement** (заміщення).
 - **Render Hidden Geometry** (обраховувати невидиму геометрію) – виконує візуалізацію всіх об'єктів сцени поза залежністю від їхньої видимості.
 - **Render to Fields** (обрахування в поля) – виконує візуалізацію для скидання на відео.
 - Група **Render Output** (вивід візуалізації) дозволяє задати параметри виводу результатів обрахування й містить такі кнопки й прапорці:
 - **Files** (файли) – викликає діалогове вікно **Render Output File** (файл виводу візуалізації). У ньому варто вибрати графічний формат, каталог на диску й ім'я файла, що зберігається.

- **Save File** (зберегти файл) – виконує збереження результатів обрахування на диск у файл із обраним ім'ям.
- **Virtual Frame Buffer** (віртуальний кадровий буфер) – виконує вивід результатів обрахування в однойменне службове вікно, що буде описане далі.
- **Net Render** (мережева візуалізація) – потокове обрахування декількома комп'ютерами по локальній мережі.
- **Skip Existing Images** (пропускати існуючі файли) – включає режим пропуску кадрів, файли яких уже створені.
- **Devices** (пристрої) – викликає діалогове вікно налаштування пристроїв скидання (цифрових відеомагнітофонів, стрічкових накопичувачів та ін.).

Після установки необхідних налаштувань потрібно натиснути клавішу **Render** [2].

2.13.3. Базові матеріали в 3DSMax

У цьому параграфі йдеться про використання матеріалів у 3DSMax.

3DSMax містить окремий модуль для роботи з матеріалами, що називається **Material Editor** (редактор матеріалів). З його допомогою можна керувати такими властивостями об'єктів, як колір, фактура, яскравість, прозорість та ін. Вікно **Material Editor** викликається за допомогою команди **Rendering-> Material Editor**.

3DSMax містить кілька типів матеріалу, кожен з яких містить у собі спеціальні налаштування. Призначувані об'єктам матеріали можуть характеризуватися різними параметрами: **Specular Level** (рівень блиску), **Glossiness** (глянець), **Self-Illumination** (самовисвітлення), **Opacity** (непрозорість), **Diffuse Color** (колір дифузійного розсіювання), **Ambient** (колір підсвічування) і т. д. У 3DSMax використовуються такі типи матеріалів:

- **Standard** (стандартний) – найпоширеніший матеріал, що використовується для текстурування більшості об'єктів у 3DSMax.
- **Advanced Lighting Override** (освітлюючий) – керує налаштуваннями, які належать до системи прорахунку світла, що розсіюється.
- **Architectural** (архітектурний) – дозволяє створювати матеріали високої якості, що володіють реалістичними фізичними властивостями. Дозволяє домогтися гарних результатів, тільки якщо в сцені використо-

вуються джерела світла Photometric Lights (фотометрія), а прорахунок висвітлення враховує розсіювання світла Global Illumination (загальне висвітлення).

- **HI Blend** (змішування) – виходить при змішуванні на поверхні об'єкта двох матеріалів. Параметр Mask (маска) його налаштувань визначає малюнок змішування матеріалів. Ступінь змішування задається за допомогою Mix Amount (величина змішування). При нульовому значенні цього параметра відобразатися буде тільки перший матеріал, при значенні 100 – другої.

- **Composite** (складовий) – дозволяє змішувати до 10 різних матеріалів, один із яких є основним, а інші – допоміжними. Допоміжні матеріали можна змішувати з головним, додавати й віднімати з нього.

- **Double Sided** (двосторонній) – підходить для об'єктів, які потрібно текстурувати по-різному з передньої й задньої сторони.

- **Ink 'n Paint** (нефотореалістичний) – слугує для створення мальованого двовимірного зображення й може бути використаний при створенні двовимірної анімації.

- **Matte/Shadow** (матове покриття/тінь) – має властивість зливатися з фоновим зображенням. При цьому об'єкти з матеріалом Matte /Shadow (матове покриття/тінь) можуть відкидати тінь і відображати тіні, що відкидаються іншими об'єктами. Така властивість матеріалу може бути використана при сполученні реальних знятих кадрів і тривимірної графіки.

- **Morpher** (морфінг) – дозволяє керувати розфарбовуванням об'єкта залежно від його форми.

- **Mutti/ Sub-Object** (багатокомпонентний) – складається із двох і більше матеріалів, використовується для текстуровання складних об'єктів.

- **Raytrace** (трасування) – для візуалізації цього матеріалу використовується трасування променів. При цьому відслідковуються шляхи проходження окремих світлових променів від джерела світла до об'єктива камери з урахуванням їх відбиття від об'єктів сцени й переломлення в прозорих середовищах.

- **Shell Material** (оболонка) – використовується, якщо сцена містить велику кількість об'єктів. Щоб було зручніше розрізняти об'єкти у вікні

проекцій, можна вказати в налаштуваннях матеріалу, як об'єкт буде розфарбований у вікні проекції і як – після візуалізації.

- **Shellac** (шелак) – багатошаровий матеріал, що складається з декількох матеріалів: Base Material (основний матеріал) і Shellac Material (шелак). Ступінь прозорості останнього можна регулювати.

- **Top/Bottom** (верх/низ) – складається із двох матеріалів, призначених для верхньої й нижньої частини об'єкта. У налаштуваннях можна встановити різний рівень змішування матеріалів.

Кожний тип матеріалу має свій спосіб затінення (шейдер). Типи затінення можуть надавати характерне для того або іншого матеріалу оформлення.

Наприклад, тип замінення Metal (метал) робить обраний тип матеріалу більше схожим на металевий. За замовчуванням об'єкту задається тип матеріалу Standard (стандартний). Щоб змінити тип, необхідно натиснути кнопку Get Material (установити матеріал) і вибрати необхідний у вікні Material/Map Browser (вікно вибору матеріалів і карт).

Задати об'єкту матеріал можна двома способами:

- перетягнути створений матеріал із вікна Material Editor (редактор матеріалів) на об'єкт у вікні проекції;

- виділити об'єкт (об'єкти) у вікні проекції, вибрати необхідний матеріал у вікні Material Editor (редактор матеріалів) і клацнути на кнопці Assign Material to Selection (призначити матеріал виділеним об'єктам) на панелі інструментів вікна Material Editor (редактор матеріалів).

Матеріали, що використовуються, можна зберігати в бібліотеці матеріалів у файли з розширенням *.MAT. Однак при цьому варто пам'ятати, що використання бібліотек матеріалів із більшою кількістю зразків помітно збільшує час завантаження програми й знижує її продуктивність.

В одній сцені можуть використовуватися різні матеріали, деякі параметри яких збігаються. Тому для групи параметрів у 3DSMax передбачена можливість швидкого копіювання.

Якщо в сцені необхідно вибрати той самий колір для декількох параметрів, можна не використовувати вікно Color Selection (вибір кольору) щораз, а настроїти колір для одного параметра, після чого просто копіювати й вставити необхідний колір.

У такий же спосіб зручно копіювати матеріали. У деяких сценах можуть знадобитися два матеріали, схожі за налаштуваннями. У цьому випадку можна створити перший матеріал, копіювати його й виправити необхідні параметри в клонованому матеріалі. Це набагато простіше, ніж створювати другий матеріал із нуля, порівнюючи його параметри з першим і вводячи значення вручну [2].

Контрольні запитання

1. Яке призначення програмного продукту 3DSMax?
2. Які основні об'єкти є у 3DSMax?
3. Які основні команди 3DSMax вам відомі?
4. Перерахуйте основні характеристики анімації у 3DSMax?
5. У чому полягає принцип використання матеріалів у 3DSMax?

Використана література

Основна

1. Adobe Flash Professional. Справка и учебные материалы. © Adobe Systems Incorporated, 2013. – helpx.adobe.com/ru/flash/flash-cs6-tutorials.html.

2. 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.

3. 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. – 47 с

4. 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.

5. 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.

6. Обручев В. Adobe Flash CS6. Официальный учебный курс / В. Обручев. – М. : Эксмо-Пресс, 2013.

7. Программирование на ADOBE ACTION SCRIPT 3.0. Руководство пользователя. © Adobe Systems Incorporated, 2013.

8. Райтман М. Action Script 3.0 для Adobe Flash Professional CS5 (+ CD-ROM) / М. Райтман. – М. : Эксмо, 2011. – 432 с.

9. Райтман М. Adobe Flash CS5. Официальный учебный курс (+ CD-ROM) / М. Райтман. – М. : Эксмо, 2011. – 448 с.

10. Робоча програма навчальної дисципліни "Комп'ютерна анімація" для студентів спеціальності 6.092700 "Технологія електронних мультимедійних видань"

медійних видань" усіх форм навчання / О. І. Пушкар, О. С. Євсєєв. – Х. : Вид. ХНЕУ, 2006. – 36 с.

11. Финкэнон Д. Flash-реклама. Разработка микросайтов, рекламных игр и фирменных приложений с помощью Adobe Flash / Д. Финкэнон. – М. : Рид Групп, 2012. – 288 с.

Додаткова

12. Аткінс Э. Adobe AIR. Практическое руководство по среде для настольных приложений Flash и Flex / Э. Аткінс, К. Ротондо. – СПб. : Символ-Плюс, 2009.

13. Коул А. Изучаем Flex 3. Руководство по разработке насыщенных интернет-приложений / А. Коул. – М. : СПб. : Символ-Плюс, 2009. – 384 с.

14. Кривуля Н. Г. В зеркале времени. Анимация двух Америк / Н. Г. Кривуля. – М. : Аметист, 2007.

15. Раппапорт А. Жест и пространство в искусстве мультипликации / А. Раппапорт // Проблема синтеза в художественной культуре, отв. ред. Б. В. Раушенбах. – М. : Наука, 1985.

16. Саймон М. Как создать собственный мультфильм. Анимация двухмерных персонажей / М. Саймон. – М. : НТ Пресс, 2006. – 336 с.

17. Технология подготовки и представления презентаций: учебн. пособ. / А. И. Пушкар, В. В. Федько, В. И. Плоткин и др. – Х. : ХГЭУ, 2003. – 122 с.

18. 3DSMax 2013. Лучший самоучитель / под ред. А. В. Харьковский. – 4-е изд., доп. и перераб. – М. : Астрель, 2013. – 480 с.

19. Ясевич П. Психосемантика при разработке рекламных сообщений / П. Ясевич, Ю. Перминова. – М. : Адвертолоджи. 2005. № 2.

20. Barrier M. The animated man: a life of Walt Disney / M. Barrier. – California, 2007.

21. Ken A. Priebe. The Advanced Art of Stop-Motion Animation / A. Ken. – Course Technology, a part of Cengage Learning, 2011.

22. В 2014 году резко возрастут мировые расходы на рекламу. Companion Online [Электронный ресурс]. – Режим доступа : <http://www.companion.ua/articles/content?id=267184>.

23. Вікіпедія – вільна енциклопедія [Електронний ресурс]. – Режим доступу : <http://uk.wikipedia.org>.

24. Добавление интерактивных возможностей с помощью фрагментов кода. Flash Professional [Электронный ресурс]. – Режим доступа : http://help.adobe.com/ru_RU/flash/cs/using/WSb03e830bd6f770ee1ca-b0432124bc51a804-8000.html.

25. История анимации [Электронный ресурс]. – Режим доступа : <http://www.myltik.ru/index.php?topic=interes/history>.

26. Корабельникова Г., Гурский Ю. Flash. Кривые Безье – основа основ. Компьютерная газета [Электронный ресурс] / Г. Корабельникова, Ю. Гурской. – Режим доступа : <http://www.nestor.minsk.by/kg>.

27. Объем рынка Интернет-рекламы Украины 2013 и развитие рынка в 2014 году [Электронный ресурс]. – Режим доступа : <http://www.imena.ua/blog/ukr-rek-2014>.

28. Основы анимации. Справка Flash Professional [Электронный ресурс]. – Режим доступа : <http://helpx.adobe.com/ru/flash/using/animation-basics.html>.

29. Расходы на интернет-рекламу в мире выросли в 2013 году на треть. Nielsen: ibusiness.ru [Электронный ресурс]. – Режим доступа : <http://ibusiness.ru/blogs/29980>.

30. Рисование во Flash. Flash Professional [Электронный ресурс]. – Режим доступа : http://help.adobe.com/ru_RU/flash/cs/using/WSd60f2311-0762d6b883b18f10cb1fe1af6-7e8aa.html.

31. Эффективность рекламы [Электронный ресурс]. – Режим доступа : http://www.preview.ru/effektivnost_reklamy.

Додатки

Додаток А
Таблиця А.1

Структура складових професійних компетентностей з навчальної дисципліни "Комп'ютерна анімація" за Національною рамкою кваліфікацій України

143

Складові компетентності, які формуються в рамках теми	Мінімальний досвід	Знання	Вміння	Комунікації	Автономність і відповідальність
1	2	3	4	5	6
Тема 1. Вступ до анімації					
Проводити аналіз анімованих роликів, відповідно до розробленого плану	Принципи, на яких заснована анімація. Історія комп'ютерної анімації, основні етапи розвитку	Знання видів оптичних іграшок. Растрова, векторна та фрактальна графіка. Суть, призначення й математичний зміст кривих Безьє	Проектувати конструкцію оптичних іграшок та створювати власну оптичну іграшку	Презентувати результати аналізу анімованих роликів	Відповідальність за результати виконаного аналізу і запропонованого звіту

1	2	3	4	5	6
Тема 2. Середовище розробки Adobe Flash					
Реалізовувати різні види анімації, що базуються як на покадровому підході так і на розкадруванні	Основи Flash-технології. Її основні переваги й недоліки. Історія появи Flash-Технології та сновні етапи її розвитку	Знання структури основного вікна Flash та опис основних об'єктів. Складові панелі інструментів редагування	Створювати покадрову анімацію. Створювати анімацію на основі розкадрування руху	Презентувати результати розроблених анімаційних документів з використанням покадрової анімації та анімації руху	Самостійно виконувати завдання щодо створення покадрової анімації та анімації руху
Тема 3. Основні інструменти Adobe Flash					
Використовувати розкадрування форми для створення анімації та використання її у складі анімованих додатків	Інструменти малювання Flash, їхня основна відмінність	Процедура побудови кривих за допомогою переміщення дотичних. Інструменти роботи з кольором. Типи текстових полів у Flash	Створювати анімацію за допомогою розкадрування форми. Використовувати анімацію розкадрування форми в анімаційному проекті спільно з іншими видами анімації	Презентувати результати створених анімованих документів із використанням анімації форми	Самостійно виконувати завдання щодо створення анімованих документів з використанням анімації форми

1	2	3	4	5	6
Тема 4. Методи створення анімації					
Створювати символи button і movie-clip	Основні принципи створення покадрової анімації	Програмні продукти для створення покадрової анімації. Поняття ключових кадрів. Принципи створення й особливості анімації руху. Розкадрування форми та її особливості	Додавати у символ button символ movie-clip для створення анімованої кнопки. Призначати події (events), на які повинна реагувати кнопка	Презентувати результати створення анімованих кнопок у вигляді мультимедійних документів, що розміщуються у мережі Інтернет	Самостійно виконувати завдання щодо розробки мультимедійних документів з використанням анімованих кнопок
Тема 5. Шари. Створення й редагування символів					
Створювати анімацію з використанням шарів маскування	Типи шарів, що використовуються у Flash та їх властивості	Поняття символа та типи символів. Бібліотеки у Flash та типи бібліотек	Створювати анімацію руху за задалегідь обраним шляхом складної форми. Створювати анімацію на основі шару-маски	Презентувати результати розроблених анімованих документів, що містять у своєму складі різні типи шарів, а також анімацію маски	Самостійно виконувати завдання щодо анімованих документів, що містять у своєму складі різні типи шарів, а також анімацію маски

1	2	3	4	5	6
Тема 6. Імпортування й оптимізація об'єктів. Робота з відео й звуком					
Створювати анімацію на основі використання символів-контейнерів	Графічні формати, що імпортуються у Flash	Відмінності імпортування векторної і растрової графіки во Flash. Трасування. Імпортування звука та відео у Flash?	Імпортувати об'єкти графіки, відео і звуку в кліпи. Використовувати символи-MovieClip для створення більш ефектної анімації, що дозволяють відтворювати ефект внутрішнього руху у об'єктах що рухаються	Презентувати результати розроблених анімованих документів, що містять у своєму складі символи-кліпи та імпортовані відео та аудіооб'єкти	Самостійно виконувати завдання щодо розробки анімованих документів, що містять у своєму складі символи-кліпи та імпортовані відео та аудіооб'єкти
Тема 7. Використання технології Adobe Flash в електронному маркетингу					
Розробляти елементи електронного маркетингу з використанням технології Adobe Flash	Тенденції розвитку реклами в мережі Інтернет та основні інструменти реклами, що використовуються в мережі Інтернет	Процедура створення кнопки у Flash. Поняття банера, види банерів, їхнє призначення. Принципи використання факторного аналізу для розробки реклами. Багатокритеріальна оцінка якості рекламної продукції	Створювати цілісний анімаційний проект з урахуванням наданих вимог до його складу	Презентувати результати розроблених мультимедійних документів для електронного маркетингу	Самостійно виконувати завдання щодо розробки мультимедійних документів для електронного маркетингу

1	2	3	4	5	6
Тема 8. Створення презентацій на основі технології Adobe Flash					
Створювати інтерактивні мультимедійні презентації у Adobe Flash	Суть, призначення й основні характеристики презентацій. Види презентацій та процедура створення презентації во Flash	Переваги використання Flash-Презентацій. Способи перетворення презентації з інших форматів в Flash	Розробляти багатодокументні мультимедійні додатки. Сполучати різні види контенту у одній мультимедійній презентації	Подавати результати різних сфер діяльності у вигляді багатоконтентних мультимедійних презентацій	Самостійно виконувати завдання щодо розробки багатоконтентних мультимедійних Flash-презентацій
Тема 9. Тестування й оптимізація кліпу для різних способів подання					
Тестувати й оптимізувати розроблені мультимедійні кліпи	Принципи оптимізації анімаційних кліпів	Існуючі методи тестування й оптимізації анімаційних кліпів для різних призначень	Виявляти переваги і недоліки оптимізації графічної інформації для використання у Flash	Презентувати результати аналізу переваг і недоліків, що були виявлені шляхом оптимізації графічної інформації для використання у Flash	Самостійно виконувати завдання щодо тестування й оптимізація кліпу для різних способів подання

1	2	3	4	5	6
Тема 10. Action Script. Основні поняття мови програмування					
Створювати та реалізовувати анімацію символів за допомоги мови програмування Action Script	Основні поняття Action Script	Теоретичні основи мови програмування Action Script та ознайомлення з її об'єктною моделлю	Використовувати програмну зміну властивостей символів-контейнерів	Презентувати результати розроблених кліпів, що використовують у своєму складі елементи реалізовані мовою програмування Action Script	Самостійно виконувати завдання щодо розробки анімаційних кліпів із використанням програмної зміни властивостей символів-контейнерів
Тема 11. Смарткліпи					
Використовувати компоненти Adobe Flash у складі анімаційних додатків	Основні елементи інтерактивних кліпів	Теоретичні основи створення інтерактивних кліпів та додатків. Вивчення елементів керування	Обирати оптимальний склад компонентів для динамічної форми. Створювати динамічні текстові поля у інтерактивних веб-додатках. Створювати кнопки, що реагують на певні події	Презентувати результати розроблених смарткліпів, що створено з використанням компонентів Flash	Самостійно виконувати завдання щодо розробки смарткліпів, що реалізують динамічні форми

1	2	3	4	5	6
Тема 12. Програмне малювання у Adobe Flash					
Використовувати основні методи класів програмного малювання у Adobe Flash	Основні поняття програмного малювання	Методи програмного малювання та створення програмної анімації за допомогою Action Script	Програмно зафарбовувати створені малюнки суцільним кольором та градієнтом	Презентувати результати розроблених анімаційних додатків	Самостійно виконувати завдання щодо розробки програмної анімації
Тема 13. Створення анімованих документів у середовищі Autodesk 3DSMAX					
Використовувати основні інструменти 3DMAX Studio для розробки тривимірних анімаційних документів	Основні поняття побудови тривимірної графіки	Основи побудови тривимірної графіки та анімації. Вивчення теоретичних основ побудови фільмів у середовищі 3DMAX Studio	Створювати тривимірні анімаційні документи, що відтворюють збирання, розбирання та роботу складних механічних пристроїв	Презентувати результати розроблених тривимірних анімаційних кліпів	Самостійно виконувати завдання щодо розробки тривимірних анімаційних документів

Зміст

Вступ .	3
Розділ 1. Створення комп'ютерної анімації за допомогою технології Adobe Flash	5
1.1. Вступ до анімації	5
1.1.1. Історія анімації	5
1.1.2. Растрова й векторна графіка	9
1.1.3. Криві Безьє	11
1.1.4. Види анімаційних документів і додатків	13
Практична складова "Ознайомлення з практичними аспектами створення найпростішої анімації, аналіз прикладів анімаційних кліпів"	15
1.2. Середовище розробки Adobe Flash	16
1.2.1. Основи технології Adobe Flash	16
1.2.2. Користувальницький інтерфейс	18
1.2.3. Робота з окремими об'єктами	21
1.3. Основні інструменти Adobe Flash	23
1.3.1. Інструменти малювання	23
1.3.2. Робота з кольором	28
1.3.3. Робота з текстом	29
1.4. Методи створення анімації	32
1.4.1. Покадрова анімація	32
1.4.2. Ключові кадри	33
1.4.3. Створення анімації за допомогою розкадрування	34
Практична складова "Створення покадрової анімації та анімації за допомогою розкадрування форми"	42
Практична складова "Створення анімації за допомогою розкадрування руху (Classic Tween та Motion Tween). Створення анімації на основі зворотної кінематики"	45
1.5. Шари. Створення й редагування символів	49
1.5.1. Використання шарів в анімації	49
1.5.2. Створення й використання символів	52
1.5.3. Призначення бібліотек	53
Практична складова "Шар, що спрямовує рух, і шар маски"	56
Практична складова "Символи-контейнери"	59
1.6. Імпортування й оптимізація об'єктів. Робота з відео й звуком	61
1.6.1. Імпортування зображень	61
1.6.2. Імпортування звуку	64
1.6.3. Імпорт відео	65
Практична складова "Імпортування і оптимізація об'єктів. Робота з відео і звуком"	68

Розділ 2. Використання Adobe Flash-технології та 3DSMax	
для створення сучасних мультимедійних документів.....	72
2.7. Використання технології Adobe Flash в електронному маркетингу.....	72
2.7.1. Рекламна продукція на основі Flash-Технології.....	72
2.7.2. Створення Flash-банера рекламного характеру.....	75
2.7.3. Аналіз впливу інформації, що подається споживачу, й оптимізація Flash-банера.....	79
Практична складова "Створення анімованої кнопки (банера)".....	84
2.8. Створення презентацій на основі технології Adobe Flash.....	86
2.8.1. Призначення й основні характеристики презентацій на основі технології Adobe Flash.....	86
2.8.2. Об'єкти, що використовуються в презентаціях.....	89
2.8.3. Майстер створення презентацій.....	90
Практична складова "Динамічні текстові поля. Активна кнопка".....	93
2.9. Тестування й оптимізація кліпу для різних способів подання.....	95
2.9.1. Тестування й налагодження розроблених анімаційних продуктів.....	95
2.9.2. Оптимізація й доробка анімації для використання в мережі Інтернет....	97
2.10. Action Script. Основні поняття мови програмування.....	98
2.10.1. Базові поняття Action Script.....	98
2.10.2. Об'єктна модель Action Script.....	101
Практична складова "Програмування на Action Script".....	104
2.11. Смарткліпи.....	109
2.11.1. Створення інтерактивних кліпів і додатків.....	109
2.11.2. Елементи керування інтерактивним фільмом, засоби збирання й подання інформації.....	111
Практична складова "Робота з компонентами Adobe Flash".....	114
2.12. Програмне малювання у Adobe Flash.....	117
2.12.1. Огляд методів програмного малювання.....	117
2.12.2. Методи програмного малювання кривих Безьє, градієнтних ліній та заливок.....	121
2.12.3. Алгоритмізація й реалізація проекту, що розробляється.....	125
Практична складова "Програмне малювання".....	126
2.13. Створення анімованих документів у середовищі Autodesk 3DSMax.....	127
2.13.1. Користувальницький інтерфейс 3DSMax.....	127
2.13.2. Основні характеристики анімації в 3DSMax та її експортування.....	131
2.13.3. Базові матеріали в 3DSMax.....	136
Використана література.....	140
Основна.....	140
Додаткова.....	141
Додатки.....	143

НАВЧАЛЬНЕ ВИДАННЯ

Євсєєв Олексій Сергійович

КОМП'ЮТЕРНА АНІМАЦІЯ

**Навчальний посібник
для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"**

Відповідальний за випуск **Пушкар О. І.**

Відповідальний редактор **Сєдова Л. М.**

Редактор **Бутенко В. О.**

Коректор **Маркова Т. А.**

План 2014 р. Поз. № 38-П.

Підп. до друку 24.12.2014 р. Формат 60 x 90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. 9,5. Обл.-вид. арк. 11,88. Тираж 400 прим. Зам. № 336.

Видавець і виготівник – видавництво ХНЕУ ім. С. Кузнеця, 61166, м. Харків, пр. Леніна, 9-А

Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи

Дк № 481 від 13.06.2001 р.