

## **ПРОЕКТУВАННЯ ПЛАТФОРМО-НЕЗАЛЕЖНИХ, ЕФЕКТОМІСТКИХ, ВЕБ-ОРІЄНТОВАНИХ ЩІЛЬНИКОВИХ ДОДАТКІВ**

Розглянуто питання проектування ефектомістких, веб-орієнтованих щільникових додатків мовою програмування JavaFX Script, яка надає змогу створювати ефекти, які є невід'ємною частиною дизайну сучасної електронної техніки. Написані програми незалежні від типу операційної системи та займають малий об'єм ресурсів процесора. Технологія програмування дає змогу розробляти програми однією мовою для багатьох задач та пристроїв: мобільних телефонів, веб-браузерів персональних комп'ютерів тощо.

**Ключові слова:** проектування веб-орієнтованих додатків, мова програмування JavaFX Script, веб-ефекти, операційна система.

**Вступ.** При розробленні сучасних програмних продуктів перед програмістами часто постає проблема реалізації інтерфейсу користувача, який має бути не тільки практичним у використанні, але й мати вишуканий дизайн. Зазвичай для цього задіюються окремі фахівці, які володіють знаннями програм для роботи з зображеннями, внаслідок роботи яких увесь інтерфейс оздоблюється різними картинками. Проте такий підхід не є ефективним, адже картинка не може відобразити всіх ефектів для компонентної моделі. На сьогодні таку проблему можна вирішити за допомогою JavaFX додатків.

### **1. Особливості використання мови JavaFX Script від фірми Sun Microsystems для програмування веб-орієнтованих щільникових додатків**

JavaFX – це платформа для створення Rich Internet Applications (RIAs), які можуть виконуватися на персональних комп'ютерах і мобільних пристроях. Технологія JavaFX дає змогу створювати програми для роботи з мультимедійним контентом, графічними інтерфейсами користувача для бізнес-додатків, ігор для персональних комп'ютерів і мобільних пристроїв, насичених графікою, мультимедіа, веб-сайти та інші задачі [2].

Технологія JavaFX була вперше продемонстрована корпорацією Sun Microsystems на міжнародній конференції розробників – JavaOne у травні 2007 р. Платформа JavaFX 1.0 була анонсована 4 грудня 2008 р. [2].

Кожна програма JavaFX створюється за допомогою декларативної мови програмування JavaFX Script. Для розроблення додатків цією мовою необхідно завантажити і встановити JavaFX 1.0 SDK. З коду програми, написаного мовою JavaFX Script, можна звертатися до будь-яких бібліотек мови Java. Тому спільне використання мов Java і JavaFX Script дає змогу вирішувати різноманітні завдання. Наприклад, логіка бізнес-програми може бути написана мовою Java, а графічний інтерфейс користувача – мовою JavaFX Script.

Програми, написані мовою JavaFX Script, можуть виконуватися на комп'ютерах з встановленим середовищем виконання Java 1.5 і вище. В даний час підтримуються такі операційні системи як Windows, MacOS, GNU / Linux і Solaris.

При створенні додатків для мобільних пристроїв розробникам пропонується мобільний емулятор, що входить до складу JavaFX 1.0 SDK. Середовище виконання JavaFX для мобільних пристроїв випущено навесні 2009 р. Додатки, написані мовою JavaFX Script, зберігаються під розширення \*.java класичним для технології Java2. Додатки JavaFX Script мають ряд класів, які дають змогу програмувати ефекти, анімації, кольори заливки тощо.

Середовищем розроблення додатків мовою JavaFX Script є NetBeans™ IDE 6.5 і версії вище, призначене для її створення, перегляду й відлагодження. У редакторі JavaFX Script є

також Reuters об'єктів JavaFX з вже готовими геометричними фігурами, компонентами інтерфейсу користувача, засобами перетворення та анімацією.

## **2. Ефекти та їх забезпечення мовою програмування JavaFX Script**

Графічний інтерфейс користувача має у своєму розпорядженні елементи інтерфейсу (меню, кнопки, значки, списки і т. п.), представлені користувачеві на дисплеї, виконані у вигляді графічних зображень [7].

Відправною точкою для майже всіх програм JavaFX є визначення вузлів і розміщення їх на графічній площині. Після цього будь-які завдання вирішується шляхом перетворення об'єктів, а також залучення візуальних ефектів, обрізання або анімації.

**Трансформація візуальних компонентів.** Основні перетворення, наявні у JavaFX SDK, забезпечують зорові образи і пояснюють, як використовувати змінні вузли інстанції та класи з пакету `javafx.scene.transform`, застосовуючи перетворення до вузлів.

Перетворення графічного об'єкта – це зміна його координат залежно від конкретних правил. У JavaFX SDK підтримуються такі види перетворень:

- переклади – перенесення графічного об'єкту на інші координати;
- обертання – поворот графічного об'єкту на певну величину кута;
- масштабування – встановлення розмірів графічного об'єкту, відносно іншого;
- шірінг – системи розширення елемента;

Ці перетворення можуть застосовуватися до окремого вузла чи групи вузлів, або для розташування контейнерів [1, 3].

**Зафарбовування візуальних об'єктів.** Кольори та їх різновиди є основною вимогою при побудові досконалого інтерфейсу користувача. Клас `Color` використовує для інкапсуляції кольору за замовчуванням SRGB колірний простір. Кожен колір має неявні значення альфа-версії 1.0 або явні значення, передбачені у конструкторі. Значення альфа визначають прозорість кольору і можуть бути представлені значеннями з плинною крапкою в діапазоні 0.0-1.0 або 0-255. Альфа значення 1,0 або 255 означає, що колір абсолютно непрозорий і альфа значення 0 або 0.0 означає, що колір повністю прозорий. При побудові візуальних об'єктів колір з явним альфа значенням або одержання кольору з альфа компоненту, колірні компоненти ніколи не перемішуються з альфа-компонентом.

Часто потрібно використовувати градієнт, для того щоб плавно зв'язати два і більше різноманітних кольорів між собою. Для цього існують два типи градієнта [2]:

- **Linear Gradient** – заповнення області лінійним градієнтом.
- **Radial Gradient** – заповнює область радіальним градієнтом, фокальна точка якого вказує на початок градієнта, а окіл на його кінець.



*Рис. 1. Використання градієнта JavaFX при розробленні панелі калькулятора*

Користувач може вказати два або більше градієнтів кольору, після чого фарба буде набувати інтерполяційної залежності між кожним кольором (рис. 1).

Додаток забезпечує масив зупинок з зазначенням того, як розподіляти кольори уздовж градієнта. Мінлива зупинка знаходиться в діапазоні від 0.0 до 1.0 і діє як ключ уздовж градієнта. Вона визначається там, де градієнт повинен бути точно певного кольору. Якщо пропорційна мінливість встановлена в позицію "істина" (за замовчуванням), то на початку і в кінці градієнт вказується за відношенням до одиниці площі (0.0-> 1.0) і буде розтягнуто по всій формою вказаного об'єкту. Якщо пропорційна мінливість встановлена у позицію "фальш", то початкова і кінцева точки мають зазначатися як абсолютне значення пікселів і ухилу.

**Надання ефектів візуальним об'єктам.** Ефект – це графічний алгоритм, який створює зображення, як правило, модифікуючи при цьому початкове зображення, взяте, на-

приклад, з серверу Yahoo (рис. 2). В даному випадку це означає фактичну зміну зображення наявного об'єкта [6].



*Рис. 2. Браузер для перегляду зображень з серверу Yahoo*

Існують такі види ефектів (рис. 3):

1. Blend – ефект, у якому поєднуються два входи разом, використовуючи один з визначених Blend Modes (моделі змішування).
2. Bloom – ефект, який робить яскравіше частини вхідного зображення, може світитися, базується на налаштуванні порогу світла.
3. Color Adjust – ефект, який дає змогу на пікселях корегувати тони, насиченості, яскравості й контраст.
4. Drop Shadow – ефект, що виконує роботу з тінями об'єкту із зазначеним кольором, радіусом і зсувом.



*Рис. 3. Фото редактор*

5. Float Map – буфер, який містить дані з плинною точкою, призначений для використання в якості параметра нащадків.
6. Flood – ефект, який надає прямокутну область, яка заповнюється даною фарбою. Це еквівалентно заповненню прямокутника зображенням і використанням Identity ефекту, тобто є більш зручним і потенційно ефективнішим.
7. Gaussian Blur – ефект розмиття гаусівського ядра згортки з встановленим радіусом.
8. Glow – ефект, який робить початкове зображення при появі таким, що світиться, базується на налаштуванні порогів світіння.
9. Identity – буфер, який просто проходить через дане зображення, не змінюючи його і вносить в інший ефект.
10. Inner Shadow – ефект, що робить тіні всередину краями даного контенту із зазначеним кольором, радіусом і зсувом.
11. Invert Mask – ефект, який повертає маску, яка є зворотною вхідній.
12. Lighting – ефект, що імітує джерело світла для висвітлювання даного об'єкту, який можна використовувати для плоских об'єктів, але має більш реалістичний тривимірний вигляд.
13. Motion Blur – ефект згладжування руху, використовує гаусівські ядра згортки з заданим радіусом і кутом.
14. Perspective Transform – ефект, який забезпечує не Афіне перетворення вхідного вмісту. Більшість Perspective Transform зазвичай використовуються для того, щоб забезпечити "Fa-ux" тривимірного ефекту, в іншому випадку має двовимірний вміст. Перспективи перетворення здатні відображати довільні чотирибічні об'єкти в інші довільні чотирикутні, але в перспективі, зберігаючи прямолінійність ліній.
15. Reflection – ефект, що робить відображену версію, введеного нижче фактичного вмісту.
16. SepiaTone – фільтр, який виробляє ефект, схожий на антикварні фотографії.
17. Shadow – ефект, який створює дублікат монохромного входу з розмитими краями. Цей ефект використовується в основному разом зі своїм чорним кольором за замовчуванням для поєднання його з оригіналом, щоб створити тінь. Він також може використовуватися зі світлим кольором і в поєднанні з оригінальним для створення ефекту світіння [2, 5].

### **Висновки**

1. Розглянуто основні принципи побудови складних інтерфейсів користувача та забезпечення їх візуальними ефектами, яких потребує сучасний технічний дизайн.
2. Вказано перелік візуальних ефектів, які можуть використовуватися при побудові складних інтерфейсів.

3. Охарактеризовано мову програмування, яка без особливих зусиль дає змогу реалізувати будь-яку поставлену задачу.

### Список літератури:

1. *Джордан Л.Л.* JavaFX special effects / Л.Л. Джордан. – Appres, 2008. – 274 с.
2. *Фленаган Д.* JavaFX Script : подробное руководство / Д. Фленаган. – O'REILLY. – 972 с.
3. *Вивер Д.Л.* JavaFX Script / Д.Л. Вивер. – Appres, 2007. – 187 с.
4. *Морріс С.* JavaFX in Action / С. Морріс. – Ant., 2008. – 354 с.
5. *Глушаков С.В.* Технологія програмування GUI / С.В. Глушаков, С.Н. Лукошкина. – FOLIO, ХИИТ. – 611 с.

*А.М. Дель Рио (НЛТУ України, г. Львов), Ю.И. Грыцюк, д-р техн. наук, профессор (Львовский государственный университет безопасности жизнедеятельности)*

### ПРОЕКТИРОВАНИЕ ПЛАТФОРМО-НЕЗАВИСИМЫХ, ЭФФЕКТОЕМКИХ, ВЕБ-ОРИЕНТИРОВАННЫХ СОТОВЫХ ПРИЛОЖЕНИЙ

Рассмотрен вопрос проектирования эффектоемких, веб-ориентированных сотовых приложений языком программирования JavaFX Script, предоставляющий возможность создавать эффекты, которые являются неотъемлемой частью дизайна современной электронной техники. Написаны программы независимы от типа операционной системы и занимают малый объем ресурсов процессора. Технология программирования дает возможность разрабатывать программы одним языком для многих задач и устройств: мобильных телефонов, веб-браузеров персональных компьютеров и др.

**Ключевые слова:** проектирование веб-ориентированных приложений, язык программирования JavaFX Script, веб-эффекты, операционная система.

*А.М. Del' Rio (NUFWT of Ukraine, Lviv), Yu.I. Grycyuk, Prof. (Lviv State University of Vital Activity Safety)*

### DESIGNING PLATFORM-INDEPENDENT, EFFECTCAPACITY, WEB-ENABLED HONEYCOMB ANNEX

Considers the design effectcapacity, web-oriented applications programming language honeycomb JavaFX Script, which allows to create effects that are an integral part of the design of modern electronic technology. Written programs are independent of the type of operating system and small amount of processor resources. This technology allows you to program in one language for many tasks and devices: mobile phones, web browsers and personal computers.

**Keywords:** planning of web-enabled additions, programming of JavaFX Script language, web-effect, operating system.