

Нагорний Дмитро, ст. магістратури факультету кібернетики; науковий керівник – д.ф-м.н., професор Джунь Й. В. (Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука, м. Рівне)

НАПИСАННЯ КРОС-ПЛАТФОРМЕННОЇ ГРИ МОВОЮ ACTION SCRIPT 3.0 У СЕРЕДОВИЩІ ADOBE INTEGRATED RUNTIME

***Анотація.** У статті досліджено необхідність крос-платформенної розробки інтерактивних додатків, що дозволяє розширити обсяг платформ та кінцевих користувачів, а також спрощує процес розробки програм. Розкрито особливості рендерінгу на графічному пристрої та на головному процесорі машини, визначено чим вони відрізняються та яким чином портувати з однієї технології в іншу. Обґрунтовано необхідність особливої уваги принципам швидкодії та якості додатків на різних платформах. Запропоновано розроблений фреймворк, який дозволяє ефективно і швидко вирішувати зазначені вище задачі.*

***Ключові слова:** додаток, фреймворк, рендерінг, оптимізація, крос-платформенні розробки.*

***Аннотация.** В статье исследована необходимость крос-платформенной разработки интерактивных приложений, что позволяет расширить количество платформ и конечных пользователей, а также упрощает процесс разработки программ. Раскрыты особенности рендеринга на графическом устройстве и на главном процессоре машины, чем они отличаются и каким образом портировать с одной технологии в другую. Обосновано необходимость особого внимания принципам быстродействия и качества приложений на разных платформах. Предложен разработанный фреймворк, позволяющий эффективно и быстро решать указанные выше задачи.*

***Ключевые слова:** приложение, фреймворк, рендеринг, оптимизация, кросс-платформенные разработки*

***Annotation.** This article explores the need to cross-platform development of interactive applications, that extend the scope of platforms and end-users, and simplifies the programmes development process. The features of the graphical rendering device and the main processor machines defined how they differ and how to port from one technology to another. Special attention is paid to the principles of performance and quality of applications on different platforms. The result is a developed framework that allows solving the above problem effectively and quickly.*

Keywords: *application, flash, framework, rendering, optimisation, cross-platform development.*

Платформа Adobe Flash є мультимедійною платформою компанії Adobe Systems призначеною для створення веб-додатків та мультимедійних презентацій. Вона широко використовується для створення рекламних банерів, анімацій, ігор, відтворення на веб-сторінках відео- та аудіо записів, а також для розробки інтерактивних сайтів. Ця платформа дозволяє створювати додатки з мінімальними затратами ресурсів, завдяки технології векторного морфінгу. Тому Adobe Flash здобула значну популярність у розробці комп'ютерних ігор та створила цілий різновид ігор у цій галузі.

У соціальних мережах з'явилась низка додатків які швидко завантажуються, а ігровий функціонал дозволяє ефективно спілкуватись гравцям через спеціальні ігрові події [1; 2].

З розвитком мобільних технологій кишенькові пристрої досягли достатньої потужності для відтворення якісних ігор, а швидкість Інтернету дозволяє користуватися соціальними додатками у довільному місці земної кулі. Велика аудиторія ігор у соціальних мережах тепер підтримується мобільними пристроями. На жаль, технологія Adobe Flash не підходить для розробки додатків на мобільні пристрої з причини великого навантаження на центральний процесор. Тим часом, на мобільному ринку з'явилися додатки що працюють лише з растровою графікою, а такий підхід вимагає порівняно великих обсягів фізичної пам'яті.

Векторну графіку можна конвертувати у растрові аналоги, що швидко відображаються на екрані з використанням графічного пристрою. Додаток зберігається у стислому вигляді з файлами, що містять об'єкти у вигляді масиву векторних точок. Такий спосіб збереження графіки у 40 разів кращий за растрові атласи, навіть при 8-бітовому градієнті кольорів.

Історія розробки фреймворку починається з робіт Гея і Роберта Тацумі [3], творців Flash, які об'єднали зусилля з Гері Гроссманом – розробником ActionScript [4] і Петером Сантанджелі – колишнім віце-президентом з інжинірингу компанії Macromedia. Метою створення компанії була розробка ігрового програмного забезпечення.

Метою нашої статті є дослідження способів розробки настільних додатків з використанням векторної графіки, які можна буде з легкістю портувати для використання у мобільних пристроях.

Сьогодні потужні корпорації (Microsoft, Google та Apple) поділили ринок, а разом з ним і розвиток платформ (Windows, Android, iOS відповідно). Розробка контенту не є достатньо рентабельною, якщо вона розрахована лише на одну платформу, тому великої популярності набули технології, що дозволяють конвертувати додаток таким чином, щоб мати можливість встановлювати його на усі платформи з мінімальними змінами

у програмному кодї. Існує низка фреймворків що зарекомендували себе у цьому напрямку: Unity3d, Marmelad, CryEngine, UnrealEngine, AIR – та багато інших.

Основним об'єктом дослідження визначено середовище *AIR (Adobe Intergrated Runtime)* – незалежне від платформи операційне середовище для веб-додатків. Програма, написана з використанням AIR, може виконуватися не лише у браузері, а і як звичайна настільна програма. AIR дає можливість перетворити існуючі веб-сервіси, написані із використанням Flash, Actionscript, HTML або JavaScript, на традиційні програми для ПК або Mobile. Зазвичай веб-сервіси зберігають дані користувача на власних серверах. З іншої сторони, можливість зберігати свою інформацію на власному ПК часто є дуже важливою опцією для користувача. Також AIR-додатки мають змогу працювати без підключення до мережі Інтернет.

Платформа AIR має додаткові переваги, зокрема, дає можливість збереження зображень та анімацій у векторному вигляді. Вона забезпечує роботу з 3D та 2D, потужну спільноту, дешевий софт без обмежень та умови (наприклад, Unity3d, які вимагають ділитись прибутком за розповсюдження додатку, після того, як розмір прибутку становить більше певної суми). Також, AIR легко портується у Flash – один з небагатьох споріднених додатків в усіх відомих браузерах (крім Safari на IOS).

Разом з великими можливостями AIR, він має недолік в плані важкого процесу розробки, спричиненого широким спектром інструментарію, та відсутність функціоналу для виконання задач високого рівня. Також, бібліотека Flash має велику кількість готових рішень, що ускладнюють процес нагляду за роботою Garbage Collector, що вимагає розробляти власні рішення роботи з відмальовкою, растеризацією, кешуванням, трансформаціями та багатьма іншими методиками роботи у розробці ігрового контенту.

Результатом досліджень є розробка фреймворку та гри, написаної на ньому, яка підготовлена для роботи на різних платформах, зокрема: Web, Desktop, IOS, Android. Фреймворк повинен підтримувати різні способи рендерінгу графіки в залежності від платформи, при умові однотипних вхідних даних.

Вхідними даними у межах розробки фреймворка будуть: векторна графіка, звукові файли формату WAV та XML документи [5].

Векторна графіка – це найзручніший метод малювання для художників. Для створення векторних малюнків та анімацій існує програма Adobe Flash Profesional, найкращий редактор векторної графіки на сьогоднішній день, що дозволяє працювати з векторними малюнками у максимальному обсязі: криві Базьє, по кадрова анімація, градієнти кольорів, растеризація, фільтри

та інше. Векторна графіка виглядає якісно при будь яких розмірах, тому її можна конвертувати у растрові аналоги будь якого обсягу.

У Adobe Flash Profesional [6], також, можна писати код, що задіюється у керуванні елементів візуалізації, та зберігати звукові файли. Покадрові анімації, вкладеність контейнерів, керування звуком та компоненти надають достатньо можливостей для використання програми у якості конструктора.

Конструктор – це програма, що дозволяє налаштовувати елементи додатку. Збіркою елементів у ігровій індустрії зазвичай займається левел-дизайнер. Він налаштовує елементи гри в залежності від внутрішніх ігрових подій та від можливостей, наданих програмістом в рамках розроблюваного додатку. Передбачення широкого та зручного функціоналу програмістом для левел-дизайнера є важливою нормою написання ігрових движків, тому у фреймвоку були передбачені та перевірені чисельні способи впливу ігрового режисера на елементи. Також, передбачена структура, яка задіює у роботу над фреймворком сторонніх програмістів.

Основними принципами такої структури є:

- модульність (розподіл функціоналу фреймворка на частини, що мінімально впливають одна на одну);
- документованість (коментовані поля та методи і онлайн документація за елементами);
- рефлексія.

У фреймворку передбачена автоматична оптимізація використання ресурсів, яка включає:

- кешування об'єктів (пулінг). Архітектура влаштована таким чином, що при створенні додаткового об'єкта вимагається наявність типізованого кешу, у який об'єкт додається після використання, та дістається при необхідності;

– кожен тип об'єктів повинен мати своє визначення (definition). Визначення допомагають зберігати інформацію у єдиному форматі без необхідності дублювання, у випадку, коли об'єкт компіляції повторюється кілька разів. Визначення у зв'язку з кешуванням дозволяє швидко створити низку об'єктів з мінімальною кількістю розрахунків;

- інструкції – описують спосіб парсингу та містять зв'язки з конструкторами об'єктів;

– список та вектор – швидкі методи перебору об'єктів. Список перебирає об'єкти через внутрішні зв'язки, без необхідності пошуку по масиву. Вектори – аналоги традиційних масивів, що мають доступ до конкретного об'єкту за індексом, без перебору усіх попередніх елементів через конкретний крок у бітах пам'яті;

- шаблон моделювання – розбиття моделі ігрових рівнів на частини, вони містять масив точок з посиланнями на необхідні визначення;

– дерево нащадків візуалізації. Кожен об’єкт має особливу матрицю перетворення, і через швидку процедуру зчеплення матриць нащадки успадковують перетворення батьківських контейнерів;

– особисте полотно піксельної відрисовки. На відміну від стандартного рендера Flash, дозволяє розділити типи відрисовки в залежності від необхідності перетворень. Якщо об’єкт візуалізації не перетворюється він підрисовується швидше. Збережена векторна графіка займає мінімум фізичної пам’яті. При парсингу, який передбачає конвертацію вхідних даних у програмні, та їх розміщення у базі даних, векторна графіка конвертується у растрову.

Анімації зберігаються в двох способах в залежності від параметрів анімації та використання ресурсів. Дрібні анімації зберігаються у вигляді масиву картинок, що зберігають кожен кадр як окрему картинку, великі анімації зберігаються у вигляді оригіналів картинок та масиву матриць для афінних перетворень. Визначення типу парсингу анімацій залежить від використання оперативної пам’яті пристрою та використання ресурсів центрального процесора. Задає тип парсингу автор гри або художник, режисер, програміст вручну, після оцінки навантаження анімації у різних умовах. Центральний процесор та оперативна пам’ять навантажуються розміром анімацій. Для уникнення великих растрових аналогів кожного кадру, застосовується спосіб збереження матриць елементів та растрових аналогів їх оригіналів. Таким чином, зберігається лише одна картинка у пам’яті і масив матриць третього порядку, що через афінні перетворення відображають елемент у комбінованому вигляді. Якщо анімації дрібні, то існує небезпека перенасичення сцени відображення елементами, від чого страждає швидкодія ЦП. Для такого випадку вибирається спосіб збереження кожного кадру, як окремої картинки (рис. 1).

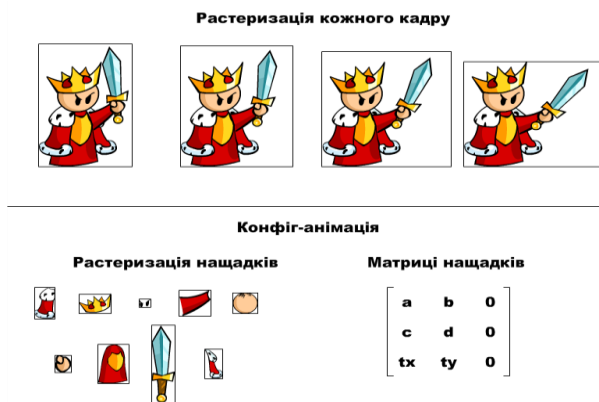


Рис. 1. Растрезація кадрів та конфіг-анімація

Відображення або рендеринг графіки також поділяється на два типи: ручний рендер та рендер графічного процесора. У технології Flash з версії програвача 10.4 існує можливість використання графічного процесора. Але на сьогоднішній день додатки, які розроблені з використанням графічних ресурсів нерентабельні. Відтак, необхідно використовувати рендер, що навантажить центральний процесор. На жаль рендер, розроблений компанією Adobe Systems, має недоліки, тому єдиним рішенням є написати свій власний, а саме – розробити технологію, що здатна оптимально відображати графічні елементи та керувати ними за вимогами програміста. Для мобільних пристроїв використовується рендер графічного процесора, це дозволяє зняти навантаження з недостатньо потужних процесорів.

Узагальнюючи результати проведеного дослідження слід зазначити, що результатом нашої роботи є створення фреймворку, що полегшує розробку двовимірних ігор, оптимізуючи використання усіх ресурсів. У рамках проекту створені ігри з використанням фреймворку, які працюють на різних платформах.

1. Фаронов В. Создание приложений с помощью C#. / В. Фаронов. – Москва : ЭКСМО, 2008. – С. 12.
2. Ахтырченко К. В. Методы и технологии реинжиниринга ИС / К. В. Ахтырченко, Т. П. Сорокваша. – Институт системного программирования РАН.
3. Gay J. «The History of Flash». Adobe. Retrieved 2007-11-10.
4. Coale K. «Macromedia Rides The FutureWave». Wired. Retrieved 2007-11-10.
5. Горбунов-Посадов М. М. Расширяемые программы / М. М. Горбунов-Посадов. – М. : Политих, 1999. – 336 с.
6. Fowler M. Inversion Of Contro