

УДК 519.67

**Сасинюк Марія, ст. магістратури факультету кібернетики**  
(Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука, м. Рівне); науковий керівник – д.ф.-м.н., професор Ляшенко І. М. (Київський національний університет імені Тараса Шевченка)

## **КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ДИНАМІКИ ФІЗИЧНИХ ОБ'ЄКТІВ У ДВОВИМІРНОМУ ПРОСТОРИ В СЕРЕДОВИЩІ ACTION SCRIPT 3.0 І ЙОГО ВПРОВАДЖЕННЯ ПРИ СТВОРЕННІ ДОДАТКІВ ДЛЯ ГРИ**

***Анотація.** У статті розкрито актуальність створення фізичних елементів, а також їх відображення при розробці flash-додатків у середовищі Action Script. Визначено фізичні властивості, які можна задати 2D об'єктам. Досліджено використання фізичного моделювання у роботах відомих компаній.*

***Ключові слова:** flash, додаток, фізика, 2D графіка*

***Аннотация.** В статье раскрыта актуальность создания физических элементов, а также их отражения при разработке flash-приложений в среде Action Script. Определены физические свойства, которые можно задать 2D объектам. Исследовано использование физического моделирования в работах известных компаний.*

***Ключевые слова:** flash, приложение, физика, 2D графика*

***Annotation.** This article explores the relevance of physical elements and their reproduction in the development of flash-applications in Action Script. The physical properties that could be set for 2D objects are studied. The using of physical modeling in the works of famous companies is analyzed.*

***Keywords:** flash, application, physics, 2D graph.*

**ActionScript** є мовою сценаріїв та програмування [1], яка дозволяє програмувати Adobe Flash-кліпи та додатки. Flash-технології, або технології інтерактивної Web-анімації, були започатковані компанією Macromedia, яка розвинула графічну програму для Web (FutureSplash), придбану в 1997 році у компанії Future Wave [2; 3]. Орієнтація на векторну графіку, як основний інструмент розробки, дозволила компанії Macromedia реалізувати всі базові елементи мультимедія – рух, звук та інтерактивність. Відомо, що візуальні та рухомі образи в багато разів швидше сприймаються людиною. Тому відтворення образів комп'ютерної анімації

у свідомості через співвідношення форм, кольорів, текстур, рухів, звуків, а також складностей їх зміни створює передумови для динамічного розвитку фізичного мислення та ефективного засвоєння нової інформації.

**Історія цієї програми** починається в 1995 році – компанія Macromedia після купівлі для анімації програми FutureSplash Animator, випустила власний продукт під назвою «Macromedia Flash 1.0». Спочатку ця програма призначалася для створення нескладної анімації, але після виходу четвертої версії та з появою вбудованої скриптової мови ActionScript, цей додаток перетворився на потужний продукт. Після цього з'явилися перші сайти зроблені цілком у Flash [1].

Математичні засоби технології Flash [3, 4] точно відтворюють динаміку руху, що дає можливість моделювати та експериментувати. Тому, опанувавши методи сучасної комп'ютерної анімації, можна глибше пізнати закономірності матеріального світу й швидше відшукати ефективні алгоритми розв'язання різних задач – технічних та природних [2; 4].

Першою повноцінною технологією моделювання фізики у Flash середовищі був Vox2d. Рушій був імпортований з мови C++, де уже активно використовувався для написання ігор з двохвимірною фізикою. Пізніше, рушій більше не оновлювали і залишили на використання не оптимізовану альфа-версію.

На перевагу Vox2d стрімко розвивався другий, сьогодні успішний, рушій Nape, який написаний на мові Haxe, що схожа на Action Script 3.0, але працює значно швидше. Завдяки цьому максимальна оптимізація Nape є доволі потужною, для того щоб працювати навіть на слабких мобільних пристроях.

Головними ознаками якісного фізичного рушія у ігровій індустрії є такі:

- можливість моделювання багатокутників та кругів;
- колізії між тілами;
- фізичні параметри (тертя, пружність, маса);
- можливість з'єднувати тіла точковими та дистанційними кріпленнями;
- гравітаційні властивості;

**Метою нашого дослідження** є створення програмних продуктів для ефективного комп'ютерного моделювання динаміки фізичних об'єктів у двовимірному просторі в середовищі Action Script 3.0.

**У статті досліджується** актуальна проблема усіх рушіїв, а саме неможливість відтворення однорідних тіл із опуклими формами, із за великого навантаження на роботу центрального процесора та складностей розрахунків при колізіях.

Для візуалізації результатів фізичного моделювання використовуються інструменти рендеру та двовимірної графіки.

Двовимірною комп'ютерною графікою класифікується за типом представлення графічної інформації, і наступними з нього алгоритмами обробки зображень.

Зазвичай комп'ютерну графіку поділяють на векторну і растрову. Растрову графіку застосовують при розробці електронних і поліграфічних видань. Зображення є мозаїкою з великого числа окремих точок (пікселів), які не розрізняються людським оком. У растровому вигляді можна уявити будь-яке зображення. Проте при цьому потрібний великий об'єм пам'яті, необхідний для обробки і зберігання зображень. Розмір файлів для зберігання зображень може досягати декількох десятків мегабайт. Для растрової графіки неминучі спотворення при редагуванні та масштабуванні. Зокрема, збільшення геометричних розмірів зображень супроводжується збільшенням геометричних розмірів пікселів. Вони стають видимими, що призводить до появи «зубчиків» на зображенні. Для растрової графіки ускладнюються операції редагування, бо об'єкти для редагування доводиться виділяти вручну [5].

Програмні засоби для роботи з векторною графікою, навпаки, призначені, в першу чергу, для створення ілюстрацій і меншою мірою для їх обробки. Зображення векторної графіки складаються з набору геометричних примітивів, або об'єктів – точок, прямих, кривих, кіл, прямокутників тощо. Кожний з примітивів описується своїм набором координат, векторів і атрибутів (товщина ліній, колір та ін.). Завдання дизайну, засновані на застосуванні шрифтів і найпростіших геометричних об'єктів, форма яких описана математично, вирішуються засобами векторної графіки набагато простіше. Необхідні елементи (примітиви) легко виділяти і тому зображення векторної графіки легко редагувати. Векторні об'єкти легко трансформуються без погіршення якості, тому зображення можна редагувати без втрат – масштабувати, повертати, деформувати тощо. Якщо для лінії задана певна товщина, то вона буде залишатися незмінною, незважаючи на збільшення чи зменшення зображення [5].

Написання програм відбувалося з використанням фізичного рушія Nape. Фізичний рушій – бібліотека, яка розраховує фізичні взаємодії між об'єктами ігрового світу (симулюється фізика, описувана законами Ньютона). Фізичні рушії, використовувані при розробці ігор, як правило, не симулюють фізичні процеси ігрового світу зі 100 % точністю, а лише виробляють достатньо точну апроксимацію фізичних законів. Сучасні ігрові фізичні рушії складаються з двох частин: підсистеми визначення зіткнень і підсистеми розрахунку фізичних взаємодій. Крім фізики твердого тіла, різні фізичні рушії можуть реалізовувати додаткові можливості: спеціальну підтримку симуляції руху автомобілів, симуляцію води та інших рідин, симуляцію тканин та одягу, симуляцію частинок, додаткову підтримку для симуляції персонажів – високорівневі контролери персонажів, підтримку анімації і т.д.

При написанні програми, необхідно вирішити як робити фізику. З одного боку, можна писати фізику самостійно, з іншого – купити готове рішення, яке повністю покриє спектр поставлених задач. Але найбільш оптимальним рішенням буде модифікування існуючого, але далеко не на 100 % відповідного

рушія. Як і в інших областях розробки ПЗ, це рішення залежить від безлічі факторів: наявності коштів, персоналу, відповідних рішень і т.д.

В залежності від вибору і вимог до фізики складність розробки може значно коливатися. Якщо ж розглядати одну і ту ж гру з фізикою і без неї, за інших рівних даних, то варіант з фізикою (досить об'ємної) торкнеться всіх учасників виробничого процесу. Наявність фізики насамперед впливає на архітектуру коду гри: введення такої складної підсистеми впливає і на графіку, і на штучний інтелект, і навіть на введення користувача.

Додавання фізики в гру ускладнює процес розробки продукту і вимагає додаткових ресурсів. Наші додатки були написані без використання професійних графічних елементів для повноцінної демонстрації можливостей фізичного моделювання. Ці приклади демонструють, як можна відтворити фізику примітивної машинки, керування її елементами та наскільки це розбавляє геймплей, та катера, що імітує плавання, достатнім чином щоб задовольнити потреби гравця.

Після запуску Adobe Flash було створено новий файл з розширенням \*.fla (рис.1) та завантажено графіку зображення машинки: корпус і колеса.

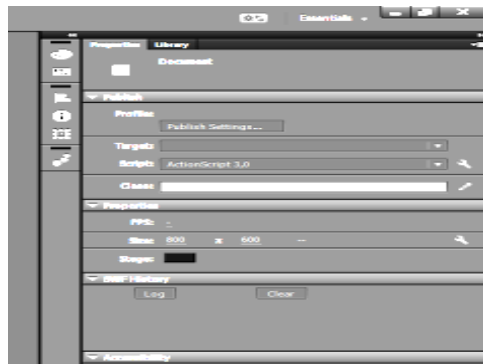


Рис.1. Формування параметрів нового файлу у Adobe Flash

Потім було намальовано прямокутник, який являє собою землю майбутнього додатку. Кожен з цих елементів був поміщений в окремі контейнери типу Sprite, а не Movie Clip, оскільки нами не передбачалося використання часової шкали та кадрів, хоча Adobe Flash Professional дозволяє створювати та анімувати об'єкти покадрово. Після цього було підписано експорт прототипів і опубліковано swc - бібліотеку, яка містить прототипи цих елементів. Це перша частина роботи, яку було зроблено у сердовищі Adobe Flash Professional.

Наступним кроком було запуском Flash Developer та відкриття нового проєкту, до якого було підключено створену раніше бібліотеку та бібліотеку фізичного рушія Nape. В класі Main було зроблено фізичний

світ, після чого створено фізичне тіло землі, фізичне тіло корпусу та фізичне тіло колеса, які було розміщено на потрібних позиціях.

Корпусу і колесам було задано динамічний тип, що означає їх майбутній рух по землі, якій відповідно даємо статичний тип.

Колеса прикріплювалися до корпусу цвяховими кріпленнями фізичного рушія. Після цього були створені екземпляри візуалізації елементів на базі прототипів із створеної раніше swc бібліотеки та додано їх на сцену.

Наступним кроком було підключення слухача переключення кадрів, що викликало функцію оновлення 60 разів на секунду (fps). Таким чином, для людського ока це виглядатиме, як звичайний рух об'єкта, чого ми і хотіли досягнути. Разом з ним, відбулося підключення слухачів на події натискання/віджимання кнопок на клавіатурі.

У функції оновлення було викликано оновлення фізичного світу, синхронізацію візуальних представників елементів з фізичними по координатам та куту, а також виконано керування фізичними елементами через публічні поля, які редагуються у делегатах підписників на натискання/віджимання клавіш. Після усіх пророблених вище дій, було викликано компіляцію та проводилося спостереження за машинкою.

Машинка мала можливість їздити вліво/вправо, за часовою та проти часової стрілки. Також ми переконатися у роботі фізичного рушія, при його потраплянні на перешкоди по нижніх боках екрану нашого скомпільованого додатку (рис. 2).

**Узагальнюючи результати** проведеного дослідження та виконаної практичної роботи слід зазначити, що їх наслідком є створення двох рушіїв, які моделюють поведінку машинки та човна у двохвимірному просторі. Їх можна вдосконалити до рівня повноцінних інтерактивних додатків. У наведеному в статті прикладі було розглянуто простий додаток, який в реальній грі формує багато складних завдань.



Рис. 2. Скомпільований додаток у Adobe Flash

1. Rex Spuy. AdvancED Game Design with Flash / Spuy Rex // Apress, 2010. – 500 p.
2. Milbourne P. The Essential Guide to Flash CS4 with ActionScript / P. Milbourne, M. Oliver, C. Kaplan // Apress, 2009. – 568 p.
3. Вогелир Д. Macromedia Flash MX Professional / Д. Вогелир, М. Пицци // Вильямс Unleashed, 2004.
4. Альберт Д. ActionScript 2.0. Наиболее полное руководство / Д. Альберт, Е. Альберт // БХВ-Петербург, 2005. – 1120 с.
5. Климяк В. Є. Інженерна і комп'ютерна графіка : навчальний посібник / В. Є. Климяк. – Х. : Вид. ХНЕУ, 2013. – 92 с.