

Технічні науки

УДК 004.4

**Дячук Роман Миколайович**

студент

Національного технічного університету України  
«Київський політехнічний інститут імені Ігоря Сікорського»

**Дячук Роман Николаевич**

студент

Национального технического университета Украины  
«Киевский политехнический институт имени Игоря Сикорского»

**Diachuk Roman Nikolaevich**

Student of the

National Technical University of Ukraine  
"Igor Sikorsky Kiev Polytechnic Institute"

**ДОДАТОК ДЛЯ МОДЕЛЮВАННЯ РУХУ ЛІТАЮЧИХ ОБ'ЄКТІВ З  
ІМІТАЦІЄЮ НА ЕЛЕКТРОННІЙ КАРТІ**

**ПРИЛОЖЕНИЕ ДЛЯ МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ  
ЛЕТАЮЩИХ ОБЪЕКТОВ С ИМИТАЦИЕЙ НА ЭЛЕКТРОННОЙ  
КАРТЕ**

**APPLICATION FOR MODELING THE MOVEMENT OF RELATED  
OBJECTS WITH SIMULATION ON THE ELECTRONIC MAP**

**Анотація:** Розроблено Інтернет додаток для імітації літаючого об'єкту на електронній карті.

**Ключові слова:** імітація, моделювання, інтернет додаток.

**Аннотация:** Разработан интернет приложение для имитации летающего объекта на электронной карте.

**Ключевые слова:** имитация, моделирование, интернет приложение.

**Annotation:** Internet application is developed for simulating a flying object on an electronic map.

**Key words:** simulation, simulation, internet application.

У сучасному суспільстві існує клас великих і складних автоматизованих систем управління положенням або рухом множини об'єктів, які переміщуються в тому чи іншому просторі – наземному, надводному, повітряному, космічному. До класу таких систем відносяться системи управління: вуличним рухом в масштабах міста, розміщенням морських або річкових суден в гаванях, переміщенням сільськогосподарської техніки на полях великого агрохолдингу, рухом літаків у повітряному просторі поблизу аеропортів [1, с. 15], повітряними боями і зустрічними боями механізованих наземних частин і т.п. Окрім проблем переробки великих обсягів інформації за обмежений час, цей клас систем має ще таку властивість, яка полягає в тому, що максимально успішне управління в окремих частинах простору може виявитися невдалим для всієї системи в цілому. Такого роду ситуації дуже схожі на гру в шахи, коли стратегія збереження фігур виявляється менш ефективною, ніж стратегія, пов'язана з жертвами фігур з метою отримання вирішальної переваги. Крім того, при швидких переміщеннях об'єктів у великих просторах, істотним може бути запізнювання обміну інформацією між окремими частинами системи – в такому разі програмний комплекс управління повинен прогнозувати поточний стан тих чи інших частин системи на основі їх попередньої поведінки і стану інших частин.

Системи імітаційного моделювання руху аерокосмічних та наземних об'єктів, по-перше, дозволяють прогнозувати обстановку шляхом моделювання фізичних процесів, що призводять до зміни стану і положення об'єктів, а по-друге, дають змогу моделювати конструкторські рішення щодо запобігання аварійним ситуаціям шляхом як маневрування

керуваними об'єктами, так і активним впливом на небезпечні об'єкти. Імітаційне моделювання, при якому логіко-математична форма досліджуваного предмета уособлює собою метод працездатності предмета, реалізований у варіанті програмного ансамблю для РС, дає можливість передбачити можливі результати тих чи інших керуючих впливів, в свою чергу надає можливість отримувати найбільш чіткі, міцні і нешкідливі укладення. Попри це даний макропрограмний комплекс дозволяє з підтримкою ланцюжка розрахунків і графічного поняття їх підсумків відтворювати (підробляти) рух функціонування предмета або організації предметів при обставинах впливу на предмет різних, як правило, хаотичних (імовірнісних) умов. З метою дослідження вищезгаданих нюансів розроблено клієнт-серверний web-додаток, що дозволяє візуалізувати розрахунки імітаційних моделей руху аерокосмічних та наземних об'єктів у вигляді динамічних сцен, які представляють агрегацію елементів багатопланового картографічного фону і виведених поверх нього одного або більше складних символів рухомих об'єктів з прив'язкою до екранної системи координат. Стратегії організації клієнт-серверної взаємодії у середовищі Web детально описано в [2, с. 22]. Запропонована система складається з двох частин, рис. 1: клієнтської, яка відповідає за взаємодію з користувачем, а також за відображення процесу моделювання у web-браузері, і серверної, яка відповідає за завантаження і обробку потрібної для моделювання інформації.

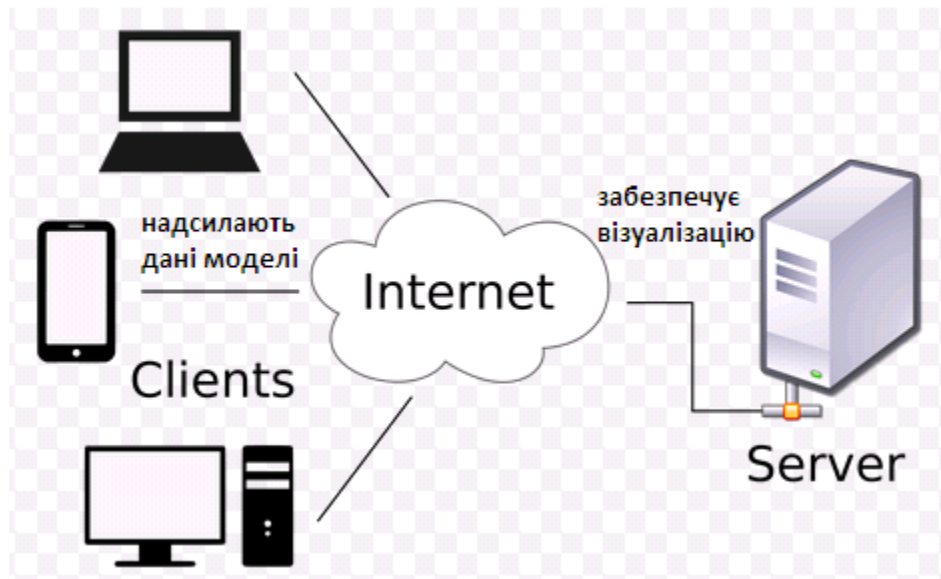


Рис. 1. Архітектура web-додатку імітаційного моделювання

Серверна частина, що реалізована на об'єктно-орієнтованій мові програмування PHP, управляє процесом збереження необхідної інформації та її зчитуванням, надаючи клієнтській частині структуру моделі в текстовому обмінному форматі JSON (JavaScript Object Notation). Вибір цього формату зумовлений тим, що за рахунок своєї лаконічності в порівнянні з XML, формат JSON, який представляє собою набір пар «ключ: значення», є більш підходящим для серіалізації складних структур, яка використовується для збереження програмних об'єктів у файли і для передачі їх по мережі. Результат серіалізації передається приймаючій стороні по транспортному протоколу HTTP. Додаток-одержувач створює об'єкт того ж типу і викликає код десеріалізації – зворотної до серіалізації операції, яка забезпечує відновлення початкового стану переданої структури даних із серіалізованого потоку даних – в результаті отримуючи об'єкт з тими ж даними, що були в об'єкті програми-відправника.

Клієнтська частина представляє собою JavaScript-додаток, який за допомогою низки JS-бібліотек відображає структуру моделі у вікні

браузера з подальшою візуалізацією процесу обробки моделі у часі. В перелік використаних бібліотек входять:

- jQuery, що фокусується на взаємодії JavaScript і HTML, причому на кшталт того, як CSS відокремлює візуалізацію від структури HTML [3, с. 18], jQuery відокремлює поведінку від структури HTML;
- Arbor.js для створення інтерактивних графів, у вигляді яких подаються визначені елементи динамічної сцени (наприклад, дорожня мережа як шар картографічного фону, або ж векторний символ рухомого об’єкта);
- Tween.js (зі складу CreateJS) для забезпечення плавної анімації динамічних елементів сцени.

Підкреслимо, що CreateJS - це комплект з деяких javascript-бібліотек, що можуть працювати так само як самостійно так і разом допомагаючи створювати діалогові інтерфейси в HTML5. Технологічні процеси HTML5 навмисно сформовані для сфери «one web world», значними особливостями якого є міжплатформенні здатності, збіжність комунікації, стандартизована мова, повсюдна комп’ютеризація і відкриті системи. Tween.js представляється додатком до Easel.js – одного з найбільш відомих javascript-фреймворків для комфортної і простої роботи з Canvas. Основним завданням цієї бібліотеки є анімація в Canvas, а також автоматизація і спрощення написання програмного коду. Альтернативою Tween.js є Move.js – javascript-бібліотека, що дозволяє робити все те ж, що і tween.js, тільки з використанням CSS3 і HTML5 натомість Canvas.

У розробленій системі технологія HTML5 Canvas використовується для відображення моделі на боці клієнта. Зважаючи на те, що HTML5-тег Canvas є корисним інструментом для малювання та анімації, цей елемент підтримується всіма популярними браузерами останніх версій. Для малювання графічних об’єктів безпосередньо на web-сторінці, тег Canvas використовує технологію JavaScript, при цьому Canvas являє собою

прямокутну область, що задається і контролюється розробником. Ця область дозволяє здійснювати візуалізацію двовимірних 2D-форм і растрових зображень, яка допускає використання сценаріїв.

Візуалізацією графа динамічної сцени, а саме його компоновкою в браузері займається бібліотека Arbor.js. Динамічна візуалізація елементів сцени реалізується за допомогою бібліотеки Tween.js. Під час завантаження сторінки, яка візуалізує модель, викликається функція `start ()`. Функція ініціалізує систему Arbor, задаючи фізичні параметри системи, та встановлює засіб рендерингу для елемента Canvas, на якому буде відбуватися промальовування кадрів динамічної сцени.

Інтерфейс клієнтського додатку влаштований так, що після статичної візуалізації сцени користувач може запустити її динамічну обробку, натиснувши на кнопку «Reload». При створенні події натискання на кнопку ініціалізується функція `redraw ()`, яка перетворює дані, отримані з сервера, в об'єкт і передає в функцію `init ()`, потім викликається функція `animate ()`.

У функції `init ()` задається алгоритм перемальовування елементів динамічної сцени. Дані для обробки представляються у вигляді масиву, який передається як параметр конструктору класу Tween. Екземпляр Tween задає послідовність кадрів від джерела до приймача, встановлюючи ряд властивостей, таких як швидкість генерації, час очікування перед наступним кроком, траєкторія руху. Функція `animate ()` викликає вбудовану функцію `requestAnimationFrame ()`, яка забезпечує вбудований API для запуску в браузері будь-яких ти-пів анімації, включаючи поворот. Також в `init ()` оновлюється екземпляр Tween.

Інтерактивна робота з системою зводиться до виконання декількох дій з боку користувача. На стартовій сторінці в браузері йому пропонується завантажити необхідні дані для візуалізації моделі шляхом множинного вибору. Після завантаження файлів (на сервер/із серверу)

відбувається візуалізація моделі у вигляді динамічної сцени з графовими елементами в двовимірній площині. Далі користувач в інтерактивному режимі може при необхідності змінювати конфігурацію сцени. Після обробки моделі він може або перейти на стартову сторінку для завантаження нової моделі (натиснувши кнопку «Home»), або закрити сторінку. За допомогою програми були звізуалізовані приклади різних імітаційних моделей руху аерокосмічних та наземних об'єктів. Для забезпечення геоприв'язки символів рухомих об'єктів до інтерактивної електронної карти, на якій кожен регіон підсвічується різними кольорами, залежно від значень, одержуваних з бази даних, використано Geolocation API.

Візуальне в динаміці представлення результатів моделювання дозволяє в заданих умовах і з потрібним ступенем деталізації продемонструвати перебіг процесів, пов'язаних з управлінням рухомими об'єктами, встановити причини виникнення тих чи інших ситуацій, а також оцінити адекватність кожної відпрацьованої моделі.

### **Література**

1. Васюхін, М. Імітаційна геоінформаційна модель представлення наземної та повітряної обстановки району аеропорту [Текст] / М. Васюхін [та ін.] // Геодезія, картографія і аерофотознімання: міжвідомчий наук.-техн. зб. / М-во освіти і науки України, Національний університет "Львівська політехніка" – Львів, 2011. – Вип. 75. – С.100-109.
2. Касім, А.М. Стратегії організації клієнт-серверної взаємодії у web-орієнтованих геоінформаційних системах [Текст] / А.М. Касім, М.М. Касім // Глушковські читання: зб. матеріалів IV наук.-практ. конф., Київ, 2 груд. 2015р. – К.: Вид-во «Політехніка», 2015. – С.73-75.

3. Дронов, В.А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. / В.А. Дронов. – СПб.: БХВ-Петербург, 2011. – 416 с.