Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)

Кафедра *Системотехніки*
(повна назва)

УДК 004.928

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(освітньо-кваліфікаційний рівень)

ГЮИК. 501600.012 ПЗ
(позначення документа)

"Дослідження методів управління анімацією та їх вдосконалення"
(тема)

Виконав:
студент II курсу, групи *СПРм-18-2*
Яровой І.Д.
(прізвище, ініціали)

Спеціальність
122 *Комп'ютерні науки*
(шифр і назва напряму, спеціальності)
Тип програми Освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма
Системне проектування
(повна назва освітньої програми)
Керівник *проф. Калита Н.І.*
(прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ _____ *проф. Гребеннік І.В.*
(підпис) (прізвище, ініціали)

2020 р

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
Кафедра *Системотехніки*
Рівень вищої освіти *другий (магістерський)*
Спеціальність *122 Комп'ютерні науки*
Тип програми *освітньо-наукова.*
Освітня програма *Системне проектування*

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ (ПРОЕКТ)

студентові Яровому Івану Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) *Дослідження методів управління анімацією та їх вдосконалення*

затверджена наказом по університету від "*30*"*03*20*20* р. № 477СТ
2. Термін подання студентом роботи (проекту) _____
3. Вихідні дані до роботи (проекту) *Функція: Дослідження методів управління анімацією та їх вдосконалення. Організація даних: файлова з прямим доступом. Форма діалогу: плагін до середовища Adobe After Effects. Перелік використовуваних програмних засобів: ОС MacOS High Sierra, інтегроване середовище розробки Adobe ExtendScript, програмне середовище Adobe After Effects CC 2020, текстових редактор Sublime Text 3. Технічне забезпечення: ноутбук зі встановленим програмним середовищем Adobe After Effects CC 2020 та вище, середовищем розробки Adobe ExtendScript 4.0 та вище, не менш ніж 4ГБ оперативної пам'яті.*
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити) *4.1 Вступ.* 4.1.1 Покадрова анімація. 4.1.2 Анімація за допомогою "Puppet Tool". 4.1.3 Анімація задопомогою настройки зв'язків шарів через вирази. 4.1.4 Побудова ріга за допомогою DUIK. 4.1.5 Постановка задачі. *4.2 Дослідження існуючих способів анімації персонажів. 4.2.1 Аналіз можлівостей скриптінгу в After Effects. 4.2.2 Ключові кадри. 4.2.3 Відмінності скриптів та виразів. 4.2.4 Специфіка роботи з виразами. 4.2.5 Ієрархія об'єктів в After Effects. 4.2.6 Інтерфейс скриптів в After Effects. 4.2.7 Використанняанімації за межами After Effects. 4.3 Аналіз засобів розробки та проектування додатку. 4.3.1 Особливостіі реалізації проекту для анімації. 4.3.2 Розробка проекту. 4.3.3 Структура проекту та програмна реалізація. 4.3.4 Демонстрація створення рігінгу. 4.3.5 Демонстрація редагування ключових кадрів 4.4 Розробка програмного засобу. 4.5 Висновки*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів) ***5.1*** *Способи анімації персонажів (1 аркуш формату А4).* ***5.2*** *Інтерфейс ExtendScript Toolkit (1 аркуш формату А4).* ***5.3*** *Ієрархія об'єктів в Adobe After Effects (1 аркуш формату А4).* ***5.4*** *Приклад графічного інтерфейсу скрипту в АЕ (1 аркуш формату А4).* ***5.5*** *Приклад графіка швидкості в Adobe After Effects (1 аркуш формату А4).* ***5.6*** *Графічний інтерфейс розробленого скрипту «Character Animator» (1 аркуш формату А4).* ***5.7*** *Процес створеня інверсної кінематики (1 аркуш формату А4).* ***5.8*** *Приклад ключових кадрів (1 аркуш формату А4).* ***5.9*** *Приклад персонажів анімованих за допомогою рігу (1 аркуш формату А4).*

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Калита Н.І.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів проекту	Примітка
1	*Розробка постановки задачі на дослідження*	*30.03.2020*	
2	*Аналіз завдання, пошук літератури та аналогів з теми атестаційної роботи*	*02.04.2020*	
3	*Опрацювання літератури та аналіз об'єкту дослідження*	*10.04.2020*	
4	*Вибір оптимального підходу до розробки програмних засобів*	*15.04.2020*	
5	*Проектування програмного засобу*	*25.04.2020*	
6	*Вибір програмних та технічних засобів для розробки*	*01.05.2020*	
7	*Розробка програмного засобу*	*07.05.2020*	
8	*Аналіз результатів дослідження*	*10.05.2020*	
9	*Оформлення пояснювальної записки та програмної документації*	*15.05.2020*	
10	*Оформлення графічної частини та презентаційних матеріалів*	*17.05.2020*	
11	*Представлення на рецензування*	*19.05.2020*	
12	*Подання атест роботи у ЕК*	*23.05.2020*	

Дата видачі завдання 30 березня 2020 р

Студент _____
(підпис)

Керівник роботи _____ *проф. Калита Н.І.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи: 90 сторінок, 65 рисунків , 2 додатки, 24 джерел. Графічна частина атестаційної роботи містить 9 плакатів.

ADOBE AFTER EFFECTS CC, АВТОМАТИЧНИЙ РІГІНГ, ПЕРСОНАЖНА АНІМАЦІЯ, КЛЮЧОВІ КАДРИ, КРИВІ ШВИДКОСТІ, EXTENDSCRIPT

Об’єктом досліджень є проектування та розробка ігрових додатків.

Предметом досліджень є програмні засоби управління анімацією, що при використанні середовища розробки анімацій дозволяє створювати ріг кінцівок графічного персонажа, оптимізувати роботу з кривими швидкості та додавати допоміжні нуль-контролери.

Мета досліджень: встановити найєфективніший метод управління персонажною анімацією.

Методи досліджень – методи створення персонажної анімації, методи графічного дизайну елементів, методи об’єктно-орієнтованої розробки, алгоритми рігінгу, системний аналіз.

У роботі був розроблений плагін для програмного середовища Adobe After Effects, що надає користувачу можливість створювати рігінг кінцівок графічного персонажа, адаптувати ілюстрації під необхідні задачі, редагувати тип анімації ключових кадрів та їх графіки швидкостей. Програмний засіб є плагіном-додатком з декількома параметрами для запуску. Для його використання необхідно мати навички початкового рівня роботи з графічним середовища Adobe After Effects.

Галузь застосування – програмний засіб використовується для автоматизації роботи з персонажною анімацією.

ABSTACT

Master's thesis contains: 90 pages, 65 images, 2 applications, 24 sources. Graphic part of the thesis contains 9 posters.

ADOBE AFTER EFFECTS CC, AUTOMATIC RIGGING, CHARACTER ANIMATION, KEYFRAMES, SPEED CURVES, EXTENDSCRIPT

The object of development is design and development of game applications.

The subject of research is animation control software, which when using the animation development environment allows you to create a horn of the graphic character's limbs, optimize the work with speed curves and add auxiliary zero-controllers.

The purpose of research is to establish the most effective method of managing character animation.

Methods of development are methods of designing character animation, methods of graphic design, methods of object-oriented development, rigging algorithms, system analysis.

The results of the thesis is a plug-in for Adobe After Effects environment, which gives to user the opportunity to create character rigging, adapt illustrations to the necessary needs, edit type of the keyframes and their speed graphs. This software is a plug-in with several parameters for launch. To use it, you need to have the advanced level skills of the Adobe After Effects graphical environment.

Scope – A software tool used to optimize process of character animation.

ЗМІСТ

ВСТУП

Сьогодні моушен-дизайн зайняв тверду позицію в сфері відео. Його використовують скрізь, де це доречно: в рекламі, промо-матеріалах, заставках. Суть моушен-дизайну в одночасному використанні графіки, анімації, спец ефектів, динаміки, фотографій, типографіки, звукових ефектів і т.д.

Навчальні анімації роблять навчання більш інтерактивною і цікавою. А ще вони допомагають краще пояснити те чи інше явище, так як показують його наочно. Плюс до всього, сьогодні все більша кількість людей легше сприймає візуальний тип інформації, тому логічно, якщо освіту підлаштовується під учнів і створює анімовані пояснення. Також варто не забувати про те, що сьогодні, наприклад, пілоти навчаються на симуляторах. І це значно зменшує витрати на тренування, але жодним чином не погіршує підготовку пілотів.

Анімація в інтернеті сьогодні відіграє вирішальну роль. У багатьох є набір гифок "на випадок важливих переговорів". Деякі намагаються створювати власні анімаційні ролики, перетворюючи їх в повноцінний контент для ютуба, наприклад. Ну і, звичайно ж, анімація відмінно працює, як частина веб-сайту. Анімація на веб-сайті допомагає привернути увагу користувача в потрібну точку сторінки і підштовхнути його до дії. Постійно рухається об'єкт буквально примушує будь-якої людини подивитися на нього і зрозуміти, що цей об'єкт хоче йому показати. Також анімація приносить веб-сайту ту ж користь, що і реклама - можливість докладніше продемонструвати те, заради чого користувач на сайт і зайшов. Тому для інтернет-магазинів анімація може стати справжнім помічником.

Моушен-графіка – візуальне оформлення для відео, телебачення і кіно, створене переважно за допомогою комп'ютерних технологій. Моушен-дизайн – це найбільш привабливий напрямок на ринку графічного дизайну так як створення телевізійної графіки вимагає відразу кілька умінь. Крім цього Моушн дизайн працює в досить високому ціновому сегменті.

Яскраво виражені тренди і інструменти в цьому напрямку стали з'являтися лише кілька років тому. Одним з головних інструментів оповідання є персонажна анімація. Однак на даний момент немає єдиного і універсального способу швидко і якісно розробити анімацію персонажа.

Актуальність роботи обумовлена відсутністю стандартних універсальних інструментів і методів розробки анімації персонажів, або їх частин, зокрема розробка стандартизованих ілюстрацій персонажів та їх базових анімацій.

Одним з найефективніших способів прискорення роботи с персонажною анімацією може бути програмний засіб для управління анімацією який дозволяє створювати ріг кінцівок персонажа, управляти кривими швидкості та швидко створювати нуль-об’єкти.

Об’єктом досліджень є проектування та розробка ігрових додатків.

Предметом досліджень є програмні засоби управління анімацією, що при використанні середовища розробки анімацій дозволяє створювати ріг кінцівок графічного персонажа, оптимізувати роботу з кривими швидкості та додавати допоміжні нуль-контролери.

Мета роботи полягає в дослідженні існуючих способів розробки анімації, виявленні їх особливостей і переваг. Як результат роботи, на основі отриманих відомостей, розробити скрипт за допомогою якого можна дуже швидко створити анімацію практично будь-якого персонажа, а так само гнучко управляти його характеристиками під час анімації.

1 ДОСЛІДЖЕННЯ ІСНУЮЧІХ СПОСОБІВ АНІМАЦІЇ ПЕРСОНАЖІВ

1.1 Покадрова анімація

Покадрова анімація (Frame-by-frame) – це анімація, яка являє собою набір зображень різних фаз руху – кадрів, які змінюють один одного з великою швидкістю, при цьому всі зміни в кожнім кадрі задаються вручну за допомогою ключових кадрів (рис.1.1).

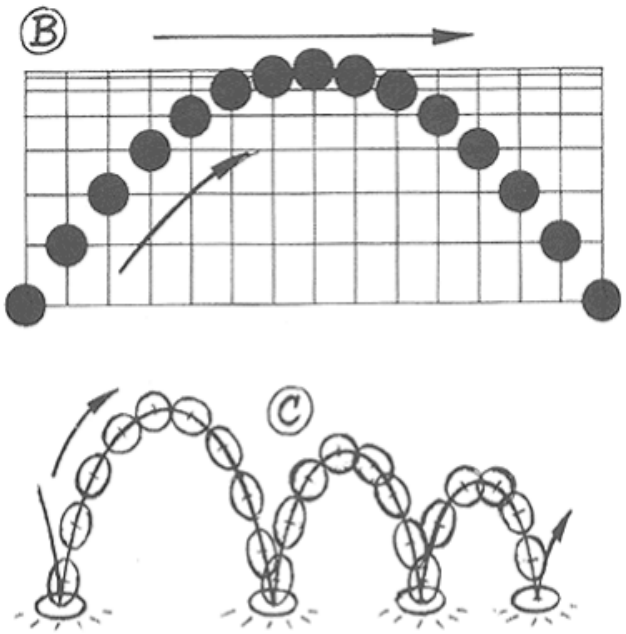

Рисунок 1.1 – Приклад розробки покадрової анімації м’яча

Покадрова анімація змінює зміст ролика в кожнім кадрі і найкраще підходить для створення складної анімації, у якій об'єкти замість простого переміщення змінюються в кожнім кадрі. Покадрова анімація збільшує розмір файлу більше ніж автоматична (tweened) [1].

Після того, як ви створюєте новий кадр, чи ключовий кадр, ви можете перемістити його в інше місце поточного шару, чи в інший шар, видалити його і зробити з ним інші зміни. Тільки ключові кадри доступні для редагування. Важливим моментом при створенні анімації є вибір частоти

кадрів (fps).

Алгоритм роботи зазвичай буде таким:

1. Малювання зображення для нового кадра.
2. Перехід до наступного кадра (просто натисніть на кадр лівою клавішою миші в шкалі з кадрами).
3. Якщо не кінець фільму, то перехід до першого пункту.

Ґрунтуючись на цьому можна виділити основні переваги покадрової анімації в програмі After Effects:

- точний контроль картинки в кадрі;
- можливість створення оригінальних і нестандартних ефектів, які неможливо повторити автоматично.

Основні недоліки наведеного способу:

- необхідні розвинені художні знання і навички, тонке відчуття тайминга;
- трудомістке внесення правок у фільм;
- часткова робота в сторонніх програмах для покадрової анімації.

1.2 Анімація за допомогою «Puppet Tool»

За допомогою інструментів «Puppet Tool» можна швидко додати природний рух в растрові зображення і в векторну графіку, включаючи нерухомі зображення, фігури і символи.

Хоча інструменти «Puppet Tool» працюють в межах ефекту (ефект «Puppet Tool»), цей ефект не застосовується за допомогою меню «Ефект» або панелі «Ефекти і шаблони». Інструменти «Puppet Tool» на панелі «Інструменти» застосовуються безпосередньо до ефекту на панелі «Шар» або «Композиція».

Ефект «Puppet Tool» працює шляхом деформації частини зображення (рис. 1.2). по встановлюються або переміщуються шпильок. Ці шпильки визначають, які частини зображення повинні бути переміщені, які частини залишаться незмінними, а які частини повинні перемістится на передній план при накладенні частин. Кожен інструмент «Puppet Tool»

використовується для розміщення та зміни шпильок певного типу.

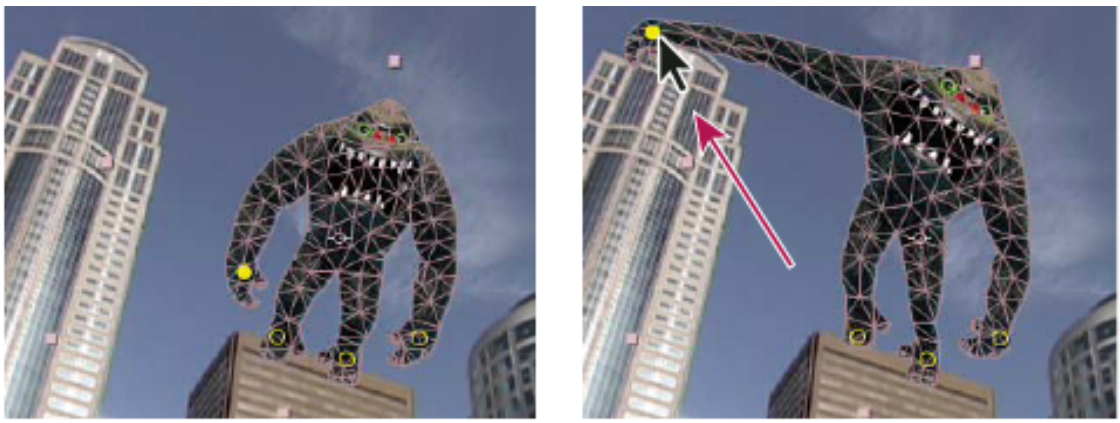

Рисунок 1.2 – Сітка, створена шляхом установки шпильок деформації (зліва), і результат перетягування шпильки деформації

Після установки першої шпильки область всередині контуру автоматично розбивається на сітку трикутників. Контур буде видно, тільки якщо був застосований ефект «Puppet Tool», а покажчик інструменту «Puppet Tool» знаходиться над областю, яку визначає контур. Кожна частина сітки також пов'язана з пікселями зображення, тому пікселі рухаються разом з сіткою.

Якщо перемістити одну або кілька шпильок деформації, сітка змінить фігуру згідно цього руху, максимально зберігаючи нерухомою всю сітку [2]. В результаті переміщення в одній частині зображення призведе до природного і правдоподібного переміщенню в інших частинах зображення.

При спотворенні однієї частини зображення може знадобитися задати, які частини зображення будуть відображатися поверх інших частин. Наприклад, може знадобитися зберегти руку перед особою, коли вона буде рухатися. Інструмент «Накладення маріонетки» використовується для застосування шпильки накладення до частин об'єкта, для якого потрібно управляти видимої глибиною. Булавки «Puppet Tool» застосовуються до вихідного контуру, а не до деформованому зображенню (рис. 1.3).

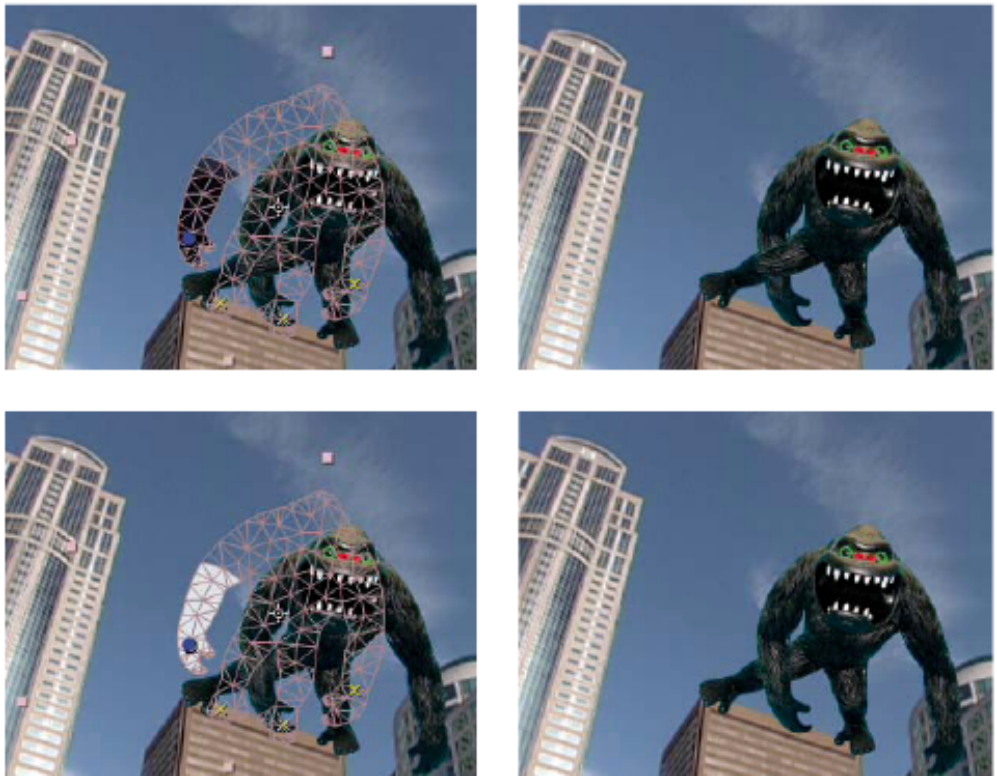

Рисунок 1.3 – Булавки накладення з від'ємним значенням «На передньому плані» (вгорі) і позитивним значенням «На передньому плані» (внизу)

Грунтуючись на цьому можна виділити основні переваги даного способу:

- можливість контролювати частини або окремі зони об'єкта;
- налаштування зв'язків між вузлами;
- програмування зв'язків між вузлами.

Основні недоліки наведеного способу:

- жорстка прив'язка до шару, що анімується;
- при зміні контуру об'єкта виникають артефакти деформації, як і з покадровою анімацією – багато ручної праці.

1.3 Анімація за допомогою настройки зв'язків шарів через вирази

При необхідності створення і підключення складних анімацій без необхідності створення десятків або сотень ключових кадрів вручну можна використовувати вирази. Вираз – це невелика частина програмного

забезпечення, багато в чому нагадує сценарій і використовується для оцінки одиночної властивості шару в конкретний момент часу. Сценарії повідомляють додатком про необхідні дії, вираження повідомляють про характеристику властивості [3].

За допомогою виразів можна створювати зв'язку між властивостями шару і використовувати ключові кадри конкретного властивості для створення динамічної анімації інших верств. Наприклад, можна використовувати інструмент «Ласо» для зв'язку властивостей шляху таким чином, щоб маска брала свій початок в штриху кисті або об'єкті шару-фігури.

Мова виразів заснований на стандартній мові JavaScript, але для використання виразів знання JavaScript не потрібно. Вирази можна створювати за допомогою інструменту «Ласо» або шляхом копіювання простих прикладів і зміни їх відповідно до вимог.

Шаблони налаштувань анімації можуть містити вирази або навіть повністю складатися з одного виразу. Шаблони налаштувань анімації, що використовують вирази замість ключових кадрів, іноді називають поведінками.

Після додавання виразу до властивості можна продовжити додавати або змінювати ключові кадри властивості. Вираз може приймати значення властивості, як це визначено відповідним ключовим кадром, і використовувати ці вхідні дані для створення нових, змінених значень.

Наприклад, такий вираз властивості «Поворот» конкретного шару додає 90 градусів до значення властивості «Поворот» на додаток до руху по ключовим кадрам:

```
value + 90
```

Деякі методи, такі як погойдування, працюють безпосередньо зі значеннями кадрированої властивості. При редагуванні вираження всі типи попереднього тимчасово недоступні; в нижній частині активних панелей, з режиму редагування тексту, яких потрібно вийти, відображається червона смуга .

Значення властивості, яке містить вираз, відображаються червоним або рожевим кольором. Найкраще почати роботу з виразами зі створення простого вираження за допомогою інструменту «Ласо», а потім налаштувати поведінку вираження, використовуючи прості математичні операції [4].

Нижче наведено ріг персонажа, шари якого поєднані за допомогою виразів (рис. 1.4).

Рисунок 1.4 – Ріг персонажа

Ґрунтуючись на цьому можна виділити основні переваги даного способу:

- гнучке управління графікою;
- ланцюговий зв'язок між усіма елементами.

Основні недоліки наведеного способу:

- обмеження на рух в сцені;
- необхідні навички програмування.

1.4 Побудова ріга за допомогою DUIK

DUIK – (duduf inverse kinematic) розробка французького анімаційного дизайнера, це плагін для AE (рис.1.5), який дозволяє створювати контролери на передбачуваних кінцівках і за допомогою зав’язків між шарами налаштовувати умовний скелет персонажа. Коли це зроблено – плагін

автоматично підв'язує до верствам складні тригонометричні вирази, які фіксують довжини кісток і дозволяють, наприклад, рухаючи тільки контролер – управляти відразу всією рукою, включаючи три суглоба: кисть, лікоть, плече (рис. 1.6).

Рисунок 1.5 – Інтерфейс плагіна

Кістьми в цьому плагіні називаються шари, які замінюють паппети в AE, він можуть використовувати правила успадкування. За допомогою налаштування кісток можна створювати «Авторіг». На думку розробників більшість персонажів щодо однакові і часом досить просто називати шари відповідними іменами, потім виділити все, натиснути кнопку «Авторіг» і плагін автоматично створить контролери і налаштує зв'язку.

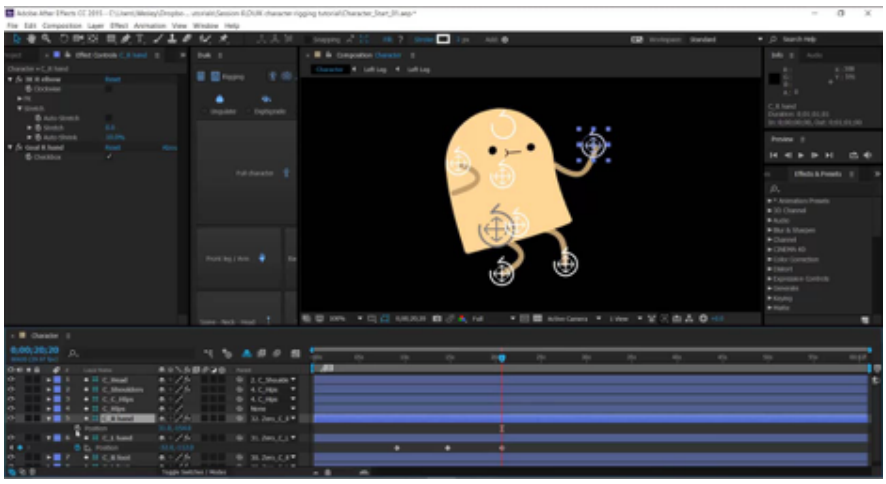

Рисунок 1.6 – Приклад ріга персонажа за допомогою DUIK

При уявній простоті використання плагін досить сильно збільшує обсяг проекту, додає значну кількість шарів, які потрібні для реалізації програмної складової плагіна, але не беруть участі в розробці анімації. Це в значній мірі засмічує виробничі приміщення та відповідно збільшує час рендеру анімації. Так само існує певна проблема з особливо випирають суглобами типу "коліно" і "лікоть", а точніше неможливості їх гнучко трансформувати при поворотах, або імітації 3D об'єкта. Навіть якщо елементарно персонажу потрібно підняти руку вгору доводиться йти на масу хитрощів, тому що найочевидніше рішення яке пропонує плагін - підняти її описавши півколо з радіусом в довжину руки [5].

Ґрунтуючись на цьому можна виділити основні переваги даного способу:

- можливість створення інверсної кінематики;
- частковий авторіг персонажа;
- додаткові можливості, гнучке налаштування контролерів.

Основні недоліки наведеного способу:

- немає можливості дублювати анімацію;
- немає можливості гнучкого редагування графіки через прив’язки кісток до паппетам;
- значно збільшує кількість шарів, змушує використовувати прекомпози. Значно збільшує час рендеру.

1.5 Постановка задачі

Ґрунтуючись на отриманих даних про специфіку і способи побудови рігу слід виділити наступні особливості, якими повинен володіти файл-проект, для того щоб мати право на життя в реальних робочих проектах:

1. Файл повинен включати більшість переваг попередніх способів анімації, настройка зв'язків шарів, гнучкість редагування анімації суглобів, можливість контролювати окремі зони об'єкта не задіяючи інші шари.
2. Нескінченний зум векторної графіки, а саме - відсутність необхідності робити прекомпозиції.

3. Можливість клонування персонажів, збереження пресетів графіків і рухів, можливість зберігання та імпорту напрацювань.

4. Можливість використання векторних шарів як плейсхолдер для растрової графіки.

AE підтримує імпорт проектів в проект, таким чином можна клонувати/ переносити напрацювання з інших проектів в поточний без втрати зв'язків/ налаштувань (чого не трапляється в разі використання DUIK). А тому, з можливих способів реалізації даного завдання, найбільш оптимальним буде розробка скрипту, який буде оптимізувати процес створення персонажної анімації.

2 АНАЛІЗ ЗАСОБІВ РОЗРОБКИ ТА ПРОЕКТУВАННЯ ДОДАТКУ

2.1 Аналіз можливостей скриптінгу в After Effects

Сценарій – це послідовність команд, розпорядчих додатком виконати послідовність операцій. Сценарії можна використовувати в більшості додатків Adobe для автоматичного виконання завдань, що повторюються, виконання складних обчислень і навіть виконання певних функцій, безпосередньо не представлених в графічному інтерфейсі. Наприклад, можна прямо вказати After Effects, щоб змінити порядок шарів в композиції, знайти і замінити вихідний текст в текстових шарах або відправити повідомлення по електронній пошті після завершення рендеринга.

Коли ми працюємо в After Effects, то в процесі роботи ми створюємо проекти, композиції, шари, маски, додаємо ефекти, і т.д. У скриптинга кожен такий елемент називається об'єктом. Кожен об'єкт має свої власні атрибутами і своїми значеннями параметрів, які відрізняють його від інших об'єктів.

Скрипт буде дотримуватися всіх інструкцій, маніпулюють цими об'єктами: створення, скасування, копіювання, анімація, рендеринг і т.д .. Іншими словами, те, що ми змушені зазвичай робити за допомогою миші або клавіатури, може бути представлено у вигляді інструкцій всередині скрипта (однак деякі дії все-таки бувають недоступні).

Скриптинг добре справляється з повторюваними процедурами, які утомливо здійснювати вручну. Те, що могло б зайняти довгі хвилини або години, може іноді бути виконаним за частки секунди за допомогою кількох рядків скрипта.

Він автоматизує всі завдання, крім того, дозволяє проводити деякі маніпуляції, які, не вдаючись до скриптингу, при створенні проекту неможливо здійснити (наприклад, на рівні масок).

Скрипт пишеться в редакторі тексту. Переконайтеся в тому, що текстовий Редактор не додає заголовки протягом записи і зберігає файл з кодуванням Unicode UTF-8. Інтегровані програми (такі як "блокнот" на

Windows або "Textedit" на Макінтош) за замовчуванням дозволяють написати скрипт (відзначимо, що з версії 7.0 в After Effects є вбудований редактор скриптів - ExtendScript Toolkit, доступ до якого можна отримати через меню «Файл»)(рис. 2.1).

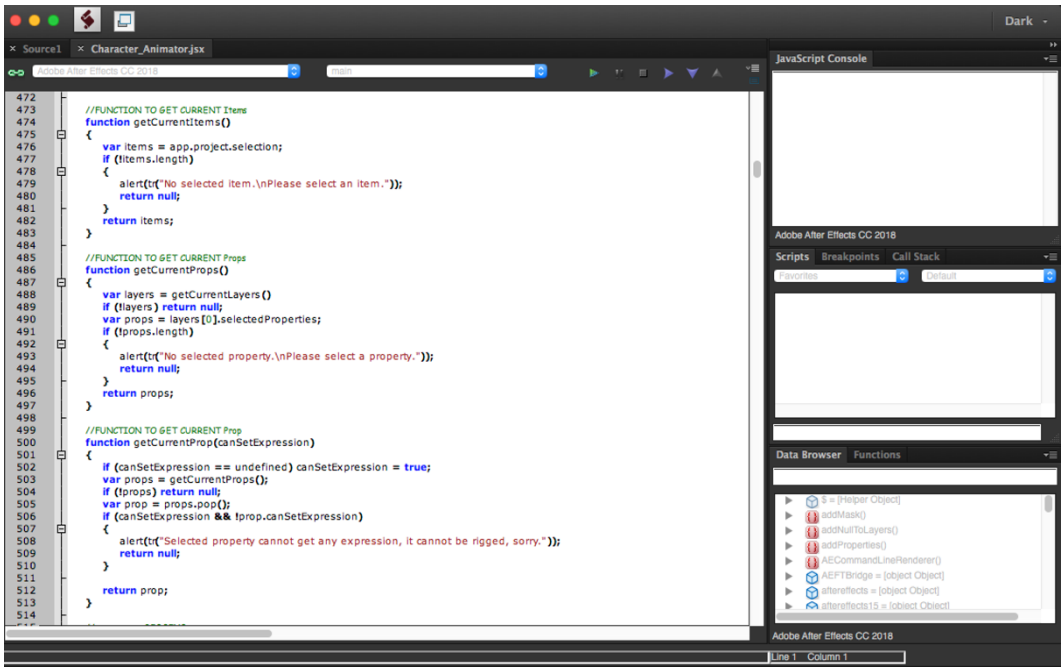

Рисунок 2.1 – Інтерфейс ExtendScript Toolkit

Скрипт After Effects використовує мову Javascript. У файлів цього типу є розширення ".js", в разі скриптів в After Effects, розширення файлів - ".jsx".

Як тільки скрипт був написаний, ми можемо його зберегти в папці "\ Scripts" (який розташовується в директорії установки програми), і тоді він з'явиться в інтерфейсі програми. Якщо його помістити в папку "\ Scripts \ Startup", він виконається при запуску програми (або в "\ Scripts \ Shutdown" тоді він виконався при закритті) [6].

2.2 Ключові кадри

Ключові кадри використовуються для завдання параметрів руху, ефектів, аудіо та інших властивостей, які зазвичай змінюються з плином часу. Ключовий кадр позначає певний момент часу, де задається значення

для властивості шару, таке як положення в просторі, ступінь прозорості або гучність звуку. Значення між Ключовими кадрами інтерполюються. При створенні змін з плином часу на основі ключових кадрів, як правило, використовується не менше двох ключових кадрів: один містить стан на початку зміни, а другий - новий стан в кінці конфігурації.

Якщо для певного властивості активний секундомір, додаток After Effects автоматично задає або змінює ключовий кадр для властивості в точці поточного часу при кожній зміні значення властивості. Якщо секундомір для властивості неактивний, воно не має ключових кадрів. Якщо значення для властивості шару змінити при неактивному секундомірі, це значення залишиться незмінним на панелі тривалості шару.

Якщо секундомір відключити, всі ключові кадри для цієї властивості шару будуть видалені, а значення постійної для властивості буде замінено значенням на поточний момент часу. НЕ відключайте секундомір, якщо тільки не впевнені в необхідності безповоротно видалити всі ключові кадри для цієї властивості.

Значки ключових кадрів в режимі панелі шарів можна замінити на числа, Вибравши в меню панелі «Таймлайн» команду «Використовувати індекси ключового кадру»(рис. 2.2).

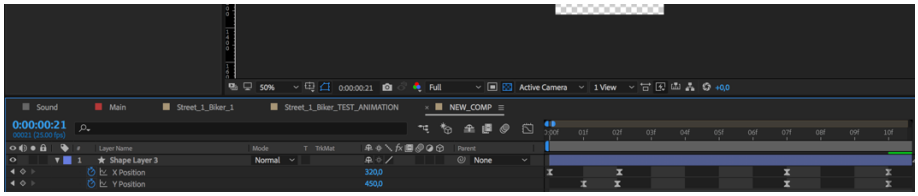

Рисунок 2.2 – Приклад ключових кадрів

2.3 Відмінності скриптів та виразів

Вирази – це інструкції, написані для властивості шару з метою привласнити певне значення цій властивості. Вираз досить часто здійснює динамічну зв'язок між різними властивостями шару. Вираз розраховується на кожному кроці рендеринга зображення, і те, що прорахувалось у кадрі,

негайно забуте в наступному кроці прорахунку кадру (у виразів немає списку даних). Вираз не здатна створити об'єкт, бо нічого не може, наприклад, створити шар (рис. 2.3).

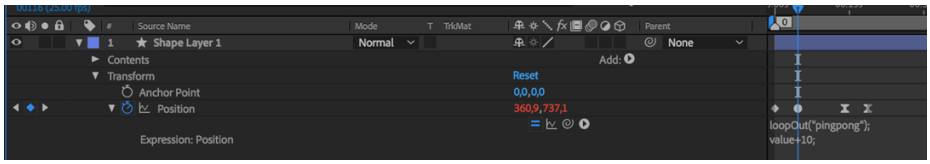

Рисунок 2.3 – Приклад виразу в AE

Скрипт в After Effects зовнішній. Він написаний в іншому текстовому редакторі, він незалежний і різноманітний в наданих можливостях. При запуску скрипт виконує свою роботу, звільняючи нас при цьому від рутини. Він дозволяє контролювати виконання, створювати об'єкти (композиції, шари, рендеринг, і т.д.), анімувати параметри об'єктів, вставляти ключі анімації або виразу, автоматизувати процедури, відправляти файл на рендер, і т.д. Змінні в скриптах глобальні, скрипт може зберігати дані, які йому потрібні. Але скрипт в деякому сенсі статичний: він діє тільки тоді, коли він запущений.

Ми можемо користуватися виразами, не знаючи особливо про мову Javascript, наприклад, просто пов'язуючи параметр з селектором виразів або просто написати математичну операцію, щоб обчислити параметр. Скриптинг ж вимагає більше зусиль і кращого розуміння мови. У той час, як вираз містить часто менше п'яти рядків, скрипт набагато довше, і написання скрипта можна порівняти з написанням програми, інформаційної повною мірою.

Однак є деякі завдання, які можуть бути здійснені одночасно із застосуванням виразів і скриптом. Загалом, вибір між виразами або скриптами треба робити відповідно до поставленим завданням.

Мова виразів After Effects заснований на JavaScript 1.2 з розширеним набором вбудованих об'єктів. After Effects використовує тільки стандартний мову JavaScript 1.2, а не браузерні розширення. After Effects містить власний набір об'єктів розширення, таких як «Шар», «Композиція», «Відеоряд» і

«Камера», які можна використовувати для отримання максимальної вигоди з роботи зі значеннями в проекті After Effects [7].

Незважаючи на те, що мова виразів заснований на мові сценаріїв, між сценарієм і виразом є незначне, але важлива відмінність: сценарії повідомляють додатком про необхідний дії, вираження повідомляють про характеристику властивості.

- Додаткові відомості про JavaScript, див. Довідку по JavaScript.
- При створенні виразів слід пам'ятати наступне.
- Значення виразу є значенням останнього обчисленого оператора.
- JavaScript - це чутливі до регістру мову.

Для поділу операторів або рядків необхідно використовувати крапку з комою. Прогалини між словами ігноруються, за винятком пробілів в рядках.

В JavaScript значення, збережене в об'єкті, називається властивістю. Однак в After Effects термін властивість використовується в відношенні компонентів шару, як це визначено на панелі «Таймлайн». З цієї причини в After Effects властивості JavaScript розглядаються в якості методів або атрибутів. Як правило, відмінність між методом і атрибутом полягає в тому, що метод зазвичай виконує будь-яку операцію для створення свого вихідного (поворотного) значення, тоді як атрибут просто посилається на існуюче значення для визначення свого вихідного (поворотного) значення. Відрізнити метод від атрибута найпростіше за наявністю дужок після імені методу, в які укладені будь-які вхідні аргументи методу.

Об'єкт – це елемент, який може містити інші об'єкти, атрибути і методи. Композиції, шари і відеоряд – це приклади об'єктів. Зокрема, композиції, шари і відеоряд є глобальними об'єктами, т. Е. До них можна встановлювати посилання в будь-якому контексті без посилання на який-небудь об'єкт більш високого рівня.

2.4 Специфіка роботи з виразами

Цілі вирази можна вводити самостійно вручну або за допомогою меню мови вираження. Також можна створити вираз за допомогою інструменту

«Ласо» або вставити його з зразка або іншої властивості.

Всі дії з виразами можна виконати на панелі «Таймлайн», проте іноді зручніше перетягнути інструмент «Ласо» до властивості на панелі «Елементи управління ефектами». Введення і зміна виразів виконуються в поле виразів, текстовому полі змінюваного розміру на діаграмі часу. Поле виразів знаходиться поряд з потрібними в режимі панелі «Шар», також воно відображається в нижній частині редактора діаграм в режимі «Редактор діаграм». Вираз можна ввести в редакторі тексту, а потім скопіювати його в поле виразів. При додаванні вираження до властивості шару, вираз за замовчуванням відображається в полі виразів. Вираз за замовчуванням по суті не виконує жодної функції, але задає значення властивості самому собі, що спрощує самостійне корегування вираження (рис. 2.4.)

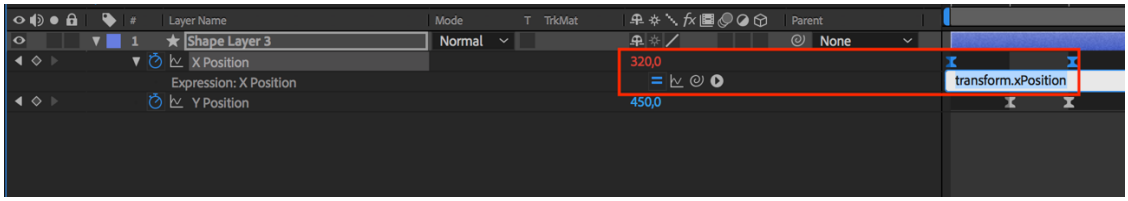

Рисунок 2.4 – Інтерфейс вираження на панелі «Таймлайн» в режимі панелі «Шар»

При редагуванні вираження всі типи препереглядy тимчасово недоступні; в нижній частині активних панелей, з режиму редагування тексту яких потрібно вийти, відображається червона смуга.

Значення властивості, яке містить вираз, відображаються червоним або рожевим кольором [8].

Найкраще почати роботу з виразами зі створення простого вираження за допомогою інструменту «Ласо», а потім налаштувати поведінку вираження, використовуючи прості математичні операції.

Наприклад, можна подвоїти результат, додавши * 2 в кінець виразу; або можна розділити результат навпіл, додавши / 2 в кінець виразу.

В процесі розвитку навичок роботи з виразами можна комбінувати ці прості дії і виконувати навіть складніші операції. Наприклад, можна додати /

360 * 100 в кінець виразу, щоб змінити його діапазон з 0-360 на 0-100. Ця зміна буде корисно, якщо Ви бажаєте перевести значення 360-градусної кругової шкали в повзункового шкалу вимірювання у відсотках.

Меню мови вираження на панелі «Таймлайн» містить мовні елементи, що відносяться до After Effects, які можна використовувати в вираженні. Це меню корисно для визначення відповідних елементів і їх правильного синтаксису. Використовуйте його в якості довідки за доступними елементам. При виборі будь-якого об'єкта, атрибута або методу в меню After Effects автоматично додає його в поле виразів в точці вставки. Якщо в поле вираження виділений текст, він буде замінений новим текстом вираження. Якщо точка вставки знаходиться не в поле вираження, то новий текст вираження замінить весь текст в поле.

В меню мови вираження містяться аргументи і значення за замовчуванням. Це правило дозволяє запам'ятати, якими елементами можна управляти при написанні вираження. Наприклад, в меню мови метод похитування в категорії «Властивість» має форму wiggle (freq, amp, octaves = 1, amp_mult = .5, t = time). П'ять аргументів перераховані в дужках за виразом wiggle. Знак = в останніх трьох аргументах вказує на те, що використання цих аргументів є необов'язковим. Якщо їм не задати значення, значеннями за замовчуванням для них будуть 1, 5 і поточний час відповідно.

Коли не вдається обчислити вираз, помилки вираження відображаються в банері попередження внизу на панелях «Композиція» та «Шар». After Effects не відключати вираз, а продовжує його обчислення. Банер з попередженням відображається, поки зберігається помилка вираження, тобто поки вираження не буде виправлено або відключено вручну [9].

2.5 Ієрархія об’єктів в After Effects

Одним з найнеобхідніших теоретичних знать під час створення скриптів є розуміння ієрархії об’єктів програмного середовища Adobe After Effects. Завдяки цьому розробник має можливість гнучко проектувати

допоміжні засоби які можуть полегшати користування програмою для вирішення специфічних задач.

В програмному середовищі Adobe After Effects виділяють наступні логічні об’єкти, які застосовуються в програмуванні:

- Application – об’єкт, що відображає логіку роботи усієї програми. Доступ до усіх інших об’єктів (окрім System, File, Folder, Socket) здійснюється через наслідування від даного об’єкту;
- Settings – об’єкт в якому відображуються усі налаштування програми. Унаслідується від об’єкту Application;
- Project – об’єкт проекту програми. Один з найважливіших та часто використуємих елементів програми. Унаслідується від Application
- RenderQueue – об’єкт черги рендеру. Через нього ми бачемо елементи які стоять в черзі на рендер та їх порядок. Унаслідується від об’єкту project;
- RenderQueueItem(s) – об’єкт елементу черги рендера. Відображає конкретний елемент який стоїть в черзі на рендер. Унаслідується від Render Queue;
- OutputModule(s) – об’єкт налаштування рендеру для окремого елементу. Відображає такі параметри, як шлях куди буде зберігатися результат після рендеру, формат вихідного файлу, специфічні налаштування відео та аудіо кодеку, назва вихідного файлу та його інші параметри. Унаслідується від RenderQueueItem(s);
- System – об’єкт, що відображає характерні властивості певної ОС. Його особливість в тому що він не унаслідується та не має наслідників;
- File – об’єкт відповідальний за роботу з файлами. Використовується коли необхідно записати/прочитати файл за межами програмного середовища Adobe After Effects. Його особливість в тому що він не унаслідується та не має наслідників;
- Folder – об’єкт який відповідає за пошук правильних шляхів до файлів, які використовуються програмою. Його особливість в тому що він не унаслідується та не має наслідників;
- Socket – об’єкт, що відповідає за підключення зовнішніх

елементів до програмного середовища Adobe After Effects через окремі сокети. Його особливість в тому що він не унаслідується та не має наслідників;

- CompItem – об’єкт композиції. Тобто композиція та її специфічні властивості, з якими можна працювати. Унаслідується від об’єкта Project;
- Layer(s) – об’єкт слою композиції. Унаслідується, в свою чергу, від об’єкта CompItem;
- Properties – об’єкт властивостей окремного слою. Може відрізнятися для різних типів слоїв, в залежності від їх специфіки. Унаслідується, в свою чергу, від об’єкта Layer(s);
- FootageItem – об’єкт, який відповідає за футажі кожного об’єкту композиції. Тобто він посилається на елементи у меню проекту. Унаслідується від об’єкта Project;
- FolderItem – об’єкт, який відповідає за папки створені в меню проекту для більш зручного зберігання футажей та елементів проекту. Унаслідується від Project(s);
- SolidSource – об’єкт солідів, тобто об’єктів які не мають футажей та створені безпосередньо в середовищі Adobe After Effects. Унаслідується, в свою чергу, від об’єкта CompItem;
- Color – об’єкт, що зберігає інформацію про колір соліду. Особливою рисою є те що кожний колір RGB змінюється в інтервалі від 0 до 1. Унаслідується від об’єкта SolidSource;
- FileSource – об’єкт, який відповідає за шлях до файлу елементу, який використовується в проекті. Унаслідується від об’єкта FootageItem.

Усі ці об’єкти використовуються під час роботи кожної операції в програмному середовищі Adobe After Effects. Тому під час проектування допоміжного програмного засобу необхідно розуміти особистості кожного елементу програми, та як вони впливають один на одного (рис. 2.5).

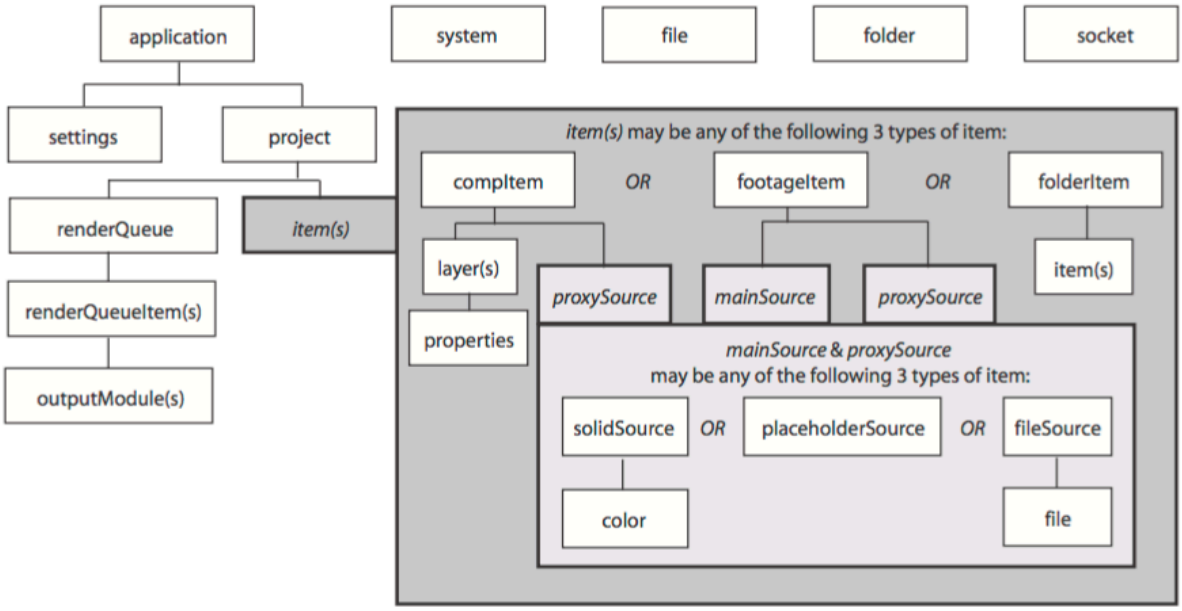

Рисунок 2.5 – Ієрархія об’єктів в AE

Кожен з об’єктів має свій набір функцій та властивостей, детальний опис яких можливо знайти в повному керівництві по скриптінгу в Adobe After Effects [10].

2.6 Інтерфейс скриптів в After Effects

Створення призначеного для користувача інтерфейсу трохи змінилося з версії 6.5 (версія скрипта, на якій ці рядки ґрунтувалися). Однак інформація, яка тут представлена, зберігає свою працездатність. Навіть якщо деякі методи є застарілими, вони все одно завжди підтримуються новими версіями AE (які пропонують більше елементів і спрощену їх побудову).

After Effects пропонує структуру мови Javascript, що дозволяє створити елементи призначеного для користувача інтерфейсу. Існують різні типи елементів (кнопки, текстові поля, елементи управління, слайдери, і т.д.), які використовуються для взаємодії з користувачем, що дозволяє йому в свою чергу безпосередньо контролювати дію скрипта.

Елементи призначеного для користувача інтерфейсу включені в контейнер «Window». Цей об'єкт має декілька атрибутів (bounds).

Існує два типи вікна: типове вікно palette і типове вікно dialog. Всі

елементи, які ми можемо помістити у вікно, розташовані в цьому контейнері з урахуванням їх координат (задаються координати верхнього лівого кута і правого нижнього кута).

Коли додаються елементи (такі, як типові елементи button наприклад), то можна асоціювати з ними будь-яку дію. До кожного з цих дій відповідає функція, яка виконує певні дії.

Кожен елементарний тип має свої власні методами: метод onClick (), наприклад, для типового елемента button і метод onChange () – для текстового поля (рис. 2.6).

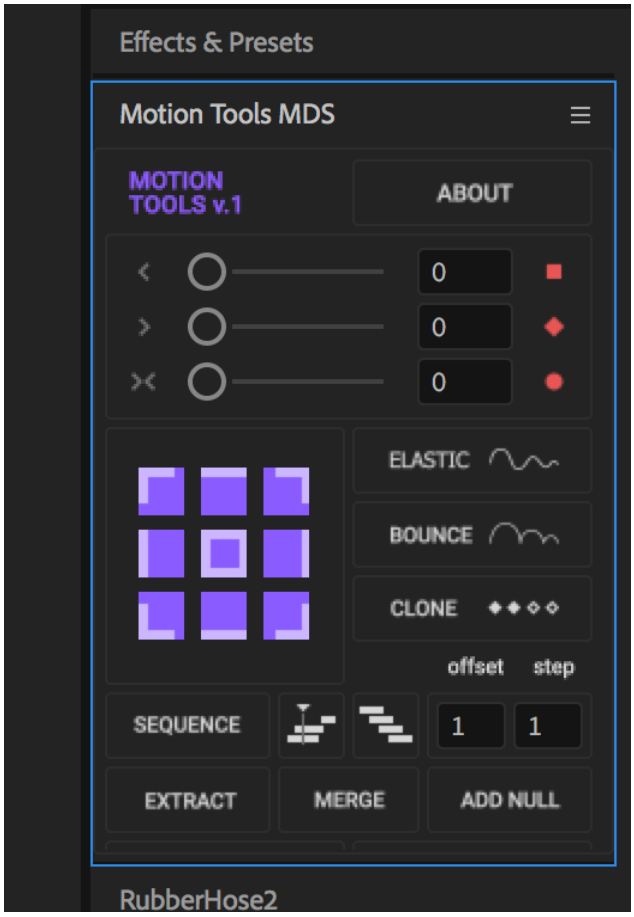

Рисунок 2.6– Приклад графічного інтерфейсу скрипту в AE

2.7 Використання анімації за межами After Effects

Отже, за рахунок чого ж досягається ефект анімації? Чому елемент сторінки здається що рухаються по ній?

Справа в тому, що координати елемента періодично змінюються. І змінюються дуже швидко, а не рідше 12 разів на секунду, по рахунок чого ми і спостерігаємо безперервний рух. Така частота обраний тому, що саме на ній людське око втрачає здатність розрізняти окремі приріст, "скачки" переривчастому руху. Коротше кажучи, за цією межею переривчасте рух стає для людини візуально безперервним. (На цьому ж принципі, до речі, працюють кінематограф і телебачення.) Комп'ютери у нас зараз досить потужні, так що забезпечити таку частоту зміни координат цілком реально [12].

Але мало всього лише змінити координати елемента. Потрібно змінити їх за особливим законом, що описує траєкторію руху елемента. Якщо елемент рухається по прямолінійній траєкторії, цей закон дуже простий. У разі криволінійного руху він сильно ускладнюється. Тому не варто створювати на своїх сторінках занадто складні анімації: малопотужний комп'ютер просто не "потягне" надто мудрі траєкторії.

Закон, за яким змінюються координати анімованого елементу, інакше кажучи, його траєкторія, описується особливою функцією (назвемо її функцією траєкторії). Така функція приймає деякі параметри і повертає координати елемента. Вона має вигляд (1.1):

$$\{x, y, z\} = f(Q, q, dq) \qquad (1.1)$$

З повертаються цією функцією результатами все просто, x, y і z - координати анімованого елементу, відповідно, горизонтальна, вертикальна і

порядок перекриття (вже знайомий вам z-індекс). Звичайно, функція, яка повертає відразу три координати, - це загальний випадок. Зазвичай змінюється одна або дві координати елемента - x і y.

А ось з параметрами, прийнятими функцією, все дещо складніше. Всього, як ви бачите, їх три, і призначення їх неочевидно. Розглянемо всі ці параметри по порядку.

Найперший параметр - Q. Це довжина траєкторії руху анімованого елементу. Вона може вимірюватися в пікселях, міліметрах, градусах або будь-яких відносних одиницях, наприклад відсотках. Взагалі, одиниця виміру залежить від самої траєкторії: для прямолінійного більше підійдуть пікселі або міліметри, а для кругової - градуси або радіани. Важливо одне: параметр Q повинен якимось чином позначати повну довжину траєкторії, по якій буде рухатися наш елемент.

Другий параметр (q) позначає позицію на траєкторії, яку займає в даний момент часу нашим елементом. Іншими словами, це відстань в одиницях виміру траєкторії Q, яке він вже "пробіг". Отримавши його, наша функція повинна сказати: "елемент тут" і, що називається, "ткнути пальцем" в точку на його траєкторії.

Останній, третій параметр - dq. Він задає приріст, на яке буде змінюватися дистанція q при кожному "стрибку" анімованого елементу. Величіна цього параметра задає швидкість руху анімованого елементу.

У розмові про простої анімації малося на увазі, що існує одна-єдина функція, що реалізує її. Реально для створення анімації може використовуватися кілька програм-сценаріїв. Автор об'єднав їх в одну функцію, щоб вам було зрозуміліше.

Передбачалося, що коли анімований елемент досягає кінця траєкторії (q стає рівною або більше Q), анімація зупиняється. Насправді, функція траєкторії може продовжити роботу. Вона може, наприклад, просто знову

виконати початкові установки, помістивши анімований елемент в початок траєкторії, і запустити анімацію знову. Але найкраще інвертувати значення dq (змінити позитивне значення на негативне і навпаки) і пустити анімацію "задом наперед". Така анімація, програє кілька або нескінченну кількість разів, називається зацикленої.

Звичайно, зациклена анімація виглядає ефектніше. Але зловживати їй не варто, особливо нескінченною. Розглянуто спосіб створення на Web-сторінці простої анімації. Але справа в тому, що в реальному житті така анімація застосовується вкрай рідко. Справа в тому, що вищеописаний спосіб створення анімації має тільки одне більш ніж сумнівне гідність і два найбільших нестачі.

Гідність - простота, більш того, очевидність реалізації. Будь-який програміст, навіть малокваліфіковану, може сісти і в п'ять хвилин накропать сценарій, який реалізує рух елемента по прямолінійній траєкторії. Так, він це зробить, навіть якщо до цього жодного разу не займався Web-програмуванням.

Уявімо двох інтернетчиків, які вирішили відвідати його сторінку з анімацією. У одного комп'ютер десятирічної давності, старенький, заслужений, ледь-ледь обробляє сучасні складні Web-сторінки, навіть без мультимедійних "наворотів". У іншого - найпотужніша найсучасніша машина, тільки що зійшла з складальних столів якогось відомого "бренду". Ось вони заходять-таки на сторінку, завантажують її і приймаються спостерігати за анімацією.

Комп'ютер першого відвідувача з натугою завантажить її і, потріскуючи жорстким диском, почне як мокре горить виконувати сценарії, що реалізують анімацію. Анімований елемент худо-бідно рухається по сторінці, відвідувач задоволений - працює. Тепер залишимо його і поглянемо на відвідувача номер два. Його комп'ютер миттєво завантажить сторінку і

миттєво ж виконає всі сценарії. Анімований елемент пролетить по своїй траєкторії так швидко, що людське око його руху навіть не помітить.

Але це півбіди. Біда настане, якщо наш невдалий програміст вирішить зробити анімацію зацикленої, нескінченною. Тепер дивіться, що вийде. Сторінка завантажується в Web-браузері, інтерпретатор починає виконувати сценарії ... і виконує ... виконує ... виконує ... А поки він їх виконує, користувач нічого не зможе зробити зі сторінкою: ні клацнути по посиланню, ні навіть прокрутити її у вікні. Єдиний спосіб перервати тривале "кіно" - закрити сам Web-оглядач [13].

Але що робити? Адже Web-програмісти якось реалізують анімацію на своїх сторінках, і вона працює нормально, в сенсі, не заважає користувачеві прокручувати сторінку і клацати по посиланнях. Вони використовують трохи іншу функцію траєкторії (1.2):

$$\{x, y, z\} = f(Q, q, dq, t) \qquad (1.2)$$

Від вже знайомої нам функції вона відрізняється тим, що приймає ще один параметр - t. Цей параметр - час. Інакше кажучи, при розрахунку координат функція траєкторії враховує поточний час. І анімація виявляється жорстко прив'язаною до часу, яке всюди тече однаково, на відміну від тактової частоти процесорів, які розрізняються у різних комп'ютерів.

Як це реалізується? Справа в тому, що нова функція виконується не весь час, поки працює анімація, а викликається час від часу, тоді, коли потрібно провести черговий "стрибок" анімації, і після цього припиняє свою роботу, чекаючи чергового виклику. Ця функція реалізується у вигляді сценарію-обробника внутрішнього події - "тика" системного таймера.

Таким чином, анімація виявляється чітко прив'язаною до часу. І на старенькому комп'ютері відвідувача номер один, і на суперсучасної машині

номера два анімований елемент буде рухатися з однієї і тієї ж швидкістю. (Звичайно, на більш потужному комп'ютері анімація, можливо, буде виконуватися плавніше, але з тією ж швидкістю, що і на більш потужному.)

Так як функція траєкторії викликається тільки час від часу, а не працює постійно, користувач може нормально взаємодіяти з Web-сторінкою. Web-оглядачеві між "тиками" таймера залишається достатньо часу, щоб обробити запити користувачів.

Така анімація, прив'язана до системного таймеру, а не до процесора, називається анімацією реального часу. Саме вона і застосовується для створення рухомих елементів на Web-сторінках. Найпростіша анімація, описана нами раніше, практично ніде не використовується.

Описаний вище спосіб створення анімації за допомогою сценаріїв, які реалізують функцію траєкторії, має безліч переваг. Такі сценарії мають дуже малий розмір і дуже швидко виконуються, так як програміст має можливість написати функцію траєкторії найбільш оптимальним способом. Однак у них є один величезний недолік -негібкость. Фактично для зміни траєкторії руху анімований-ного елемента потрібно писати новий сценарій, який реалізує нову функцію траєкторії. Но негибкость — не единственный недостаток этого способа создания анимации. Если анимированный элемент должен совершать достаточно сложное движение, реализующий эту траекторию сценарий получается очень большим и медленным. И чем сложнее траектория, тем больше и медленнее получается сценарий [14].

Третя вада - складність реалізації складних траєкторій. Малодосвідчені програмісти, особливо не мають серйозної математичної підготовки, зазвичай обмежуються простенькими траєкторіями, як правило, прямолінійними. Максимум, на що вони наважуються, - це простий еліпс. Але ж для деяких завдань, які будуть розглянуті нижче, потрібні якраз досить

складні траєкторії руху. І обійтися простими траєкторіями вельми проблематично.

Однак і з цього становища є вихід. На допомогу недосвідченим програмістам, які не володіють вищою математикою, прийде новий вид функції траєкторії, що приймає лише два параметри: масив ключових точок траєкторії і, природно, поточний час (1.3):

$$\{x, y, z\} = f([p1, t1, p2, t2..., t]) \qquad (1.3)$$

Щодо часу все зрозуміло, але що таке масив ключових точок? Нічого складного: це набір точок, за допомогою якого задається траєкторія руху нашого анімованого елементу. Пояснимо це більш докладно.

Припустимо, що нам потрібно створити дуже складну анімацію, коли елемент сторінки рухається по вельми химерної траєкторії. При цьому вищою математикою ми не володіємо, тому вивести формулу цієї траєкторії не зможемо. Однак ми придумали ось що.

Можна зробити трохи по-іншому. Перш за все, позначимо і пронумеруємо ключові точки прямо на намальованою траєкторії. Далі окремо по лінійці пряму і проставимо на ній координатну шкалу часу, проградуювати її, скажімо, в секундах. Відзначимо уздовж цієї шкали навпроти відповідних ділень точки початку і кінця анімації, проведемо між ними лінію товстіший і назвемо її доріжкою анімації. Після цього залишиться тільки пронумерувати ключові точки і проставити їх на доріжці напроти відповідних відміток часу. В результаті ми отримаємо набір точок, координати і час проходження яких можна легко обчислити (відповідно, на намальованою схемою і часовій шкалі). Накреслена траєкторія руху елемента на міліметрівці, після чого виділяємо на даній траєкторії потрібну кількість ключових точок, які її, власне, і створюють (точки перегину, початок і кінець

траєкторії тощо), і виписали їх координати на окремий папірець. Далі, знаючи час, за яке анімований елемент пройде всю траєкторію, можна з'ясувати, в який момент часу він з'явиться в тій чи іншій ключовій точці. Залишилося тільки виписати ці значення часу на ту ж папірець, проставивши їх навпроти координат відповідних точок. Ось ми і отримали масив ключових точок, який можна передати новій функції траєкторії. Цей підхід і використовується Dreamweaver.

Для прикладу наведена анімація, що включає один єдиний анімований елемент. Але Dreamweaver дозволяє поміщати на тимчасову шкалу відразу потрібну кількість елементів, створюючи кілька анімаційних доріжок, що може бути використано для створення дуже складних анімацій, в яких всі елементи рухаються узгоджено. Більш того, Dreamweaver дозволяє створювати на одній сторінці кілька працюють незалежно один від одного анімацій, кожна з яких може включати в себе будь-яку кількість анімованих елементів. Так що навіть самий вибагливий аніматор буде задоволений.

Функція траєкторії, що приймає як параметр масив ключових точок, дуже складна, але нам і не потрібно її реалізовувати. Багато потужні Web-редактори надають користувачам можливість створення анімації і використовують для цього якраз такий підхід. Набір Web-сценаріїв, які реалізують функцію траєкторії, в цьому випадку вставляється в HTML-код сторінки автоматично, користувач навіть не здогадується про це.

Звичайно, Dreamweaver не виняток. Він теж надає користувачам таку можливість. І також робить всю чорнову роботу сам.

Недоліком такого способу створення анімації є великий розмір і не дуже високу швидкодію отриманого коду. Тому для найпростіших анімацій, напевно, буде виправданий все-таки перший підхід: спеціально написані сценарії, безпосередньо реалізують функцію траєкторії.

Розваги в Інтернеті - досить молода галузь Web-будівництва. Спочатку Інтернет був створений як мережа для вчених, яким потрібно було обмінюватися текстовими документами (спочатку навіть без графіки) і пов'язувати їх в якусь подобу структури. Потім в мережу прийшов обиватель, і Web-дизайнери кинулися догоджати його смакам. (Не будемо сперечатися, наскільки вони піднесені або низовинні. Особиста думка автора: обиватель надто багатоманітним, щоб звести його до одного-єдиного ярлику.) В Інтернеті з'явилися аудіо і відео, на Web-сторінки прийшли складна графіка і анімація. Зараз всі ці "навороти" використовуються так часто, що ними вже мало кого можна здивувати.

Вміле і помірне використання анімації значно пожвавить сторінки. Важливо тільки зрозуміти, якого ефекту треба досягти, і зробити все для того, щоб він був досягнутий. Не потрібно перестаратися, а застосовувати анімацію тільки там, де вона дійсно потрібна. І вже, не дай бог, не буде працювати рябіти, як екран телевізора з відключеною антеною, - в цьому немає нічого хорошого. Не треба дозволяти прикрасам затулити вміст.

Тепер реклама. Вона з'явилася в Мережі разом з розвагами, а значить, разом з обивателем. Вона вже здорово набридла, ця інтернет-реклама, чи не більше, ніж реклама телевізійна. Але віддамо належне рекламі (і інтернетівський, і телевізійної) - завдяки їй отримали можливість існувати дуже багато популярних некомерційні проекти. Якщо реклама раптом зникне, ці проекти пропадуть відразу ж слідом за нею.

Традиційно для рекламних цілей в мережі використовуються так звані банери - невеликі графічні зображення жорстко стандартизованих форматів. Майже всі банери створюються в форматі "анімований GIF", тобто вже використовують можливості анімації по залученню уваги потенційного клієнта. Рекламну анімацію, засновану на Web-сценаріях, схоже, ніхто ще не

застосовував, а якщо і використовував, то дуже мало. Так що перед вами велике неоране поле діяльності, яке цілком може принести непогані плоди.

Анімація, заснована на вільно позиціонуються елементах і Web-сценаріях, значно компактніше будь-яких відеофайлів, будь то анімований GIF, відеофайли форматів AVI або Apple QuickTime. Однак анімовані GIF-файли підтримуються абсолютно всіма Web-оглядачами, навіть найстарішими, тому і застосовуються так широко. Web-сценарії ж будуть працювати тільки на достатньо нових Web-браузерах (якими, треба сказати, зараз користується переважна більшість користувачів Інтернету).

Від реклами плавно переходимо до утворення. Програми - навчальні посібники, часто використовувані в освіті, як правило, пишуться на компільованих мовах програмування і являють собою звичайні MS-DOS- або Windows-програми. Навчальні посібники, зроблені на основі "живих" Web-сторінок, зустрічаються поки ще досить рідко, хоча цей напрямок досить перспективно. Такі навчальні посібники можна дуже швидко створювати і модифікувати; а в порівнянні зі звичайними програмами вони виключно компактні (ну скільки місця можуть займати нескладна Web-сторінка і пара зображень?). До того ж такі посібники прямо-таки просяться в Інтернет, на Web-сайти, а значить, можна без особливих проблем організувати модне нині дистанційне навчання.

Правда, дані навчальні посібники мають один величезний недолік. Будь досить досвідчений интернетчик без праці зможе переглянути вихідний код самої сторінки і її Web-сценаріїв і, при бажанні, запозичити звідти парочку знахідок.

Для роботи нам відразу ж знадобиться панель Timelines, в якій відображаються всі анімації, створені на Web-сторінці. Щоб вивести цю панель на екран, включите пункт-вимикач Timelines в підміню Others меню Window або натисніть комбінацію клавіш <Alt> + <F9> (Рис 2.7).

Як видно, панель Timelines знаходиться в ще одному доці, що займає нижню частину головного вікна Dreamweaver. Це означає, що можна приховати панель Timelines, якщо вона нам не потрібна, натиснувши кнопку приховування дока, а потім повернути її на екран.

Більшу частину цієї панелі займає тимчасова шкала, але проградуіро-ванна не в секундах, а в кадрах анімації - так зручніше. Значення часу (в кадрах) написані на сірій часовій шкалі, розташованій вище. Під цією шкалою відображаються всі доступні канали анімації, і зайняті, і незайняті. У верхній же частині панелі знаходяться декілька елементів управління, призначених для завдання деяких параметрів доріжок і самої анімації.

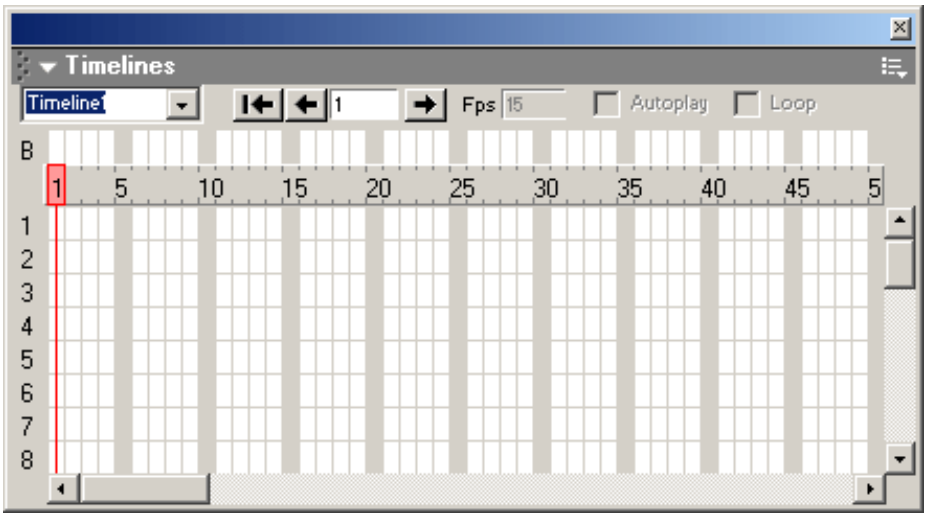

Рис. 2.7 – Панель Timeline

Але ж Dreamweaver дозволяє працювати одночасно з декількома тимчасовими шкалами. Де ж решта шкали?" Щоб побачити їх, потрібно вибрати відповідний пункт комбінованого списку анімацій (Рис 2.8).

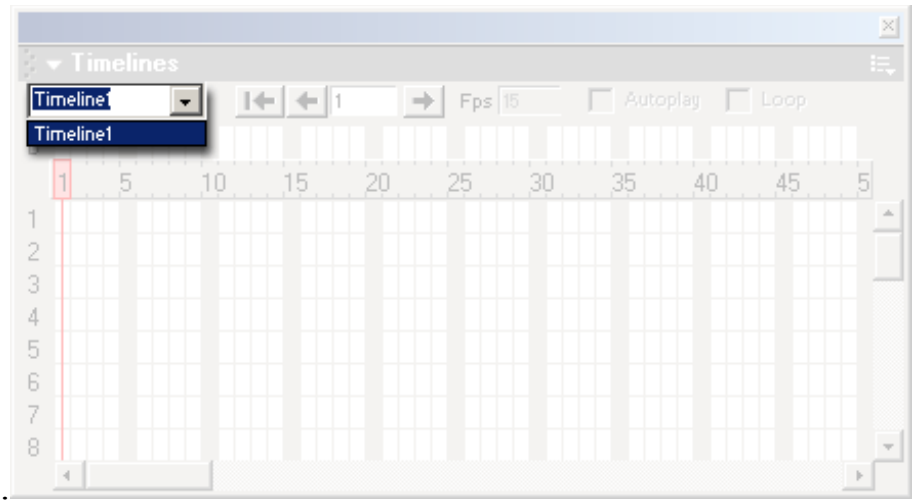

Рис. 2.8 – Комбінований перелік анімацій

Щоб анімувати будь-якої з доступних вільно позиціонуються, потрібно помістити його на шкалу часу, створивши доріжку анімації. Для цього треба виділити необхідний вільний елемент, клацнути по ньому правою кнопкою миші і вибрати в контекстному меню пункт Add to Timeline. Ви також можете вибрати пункт Add Object контекстного меню тимчасової лінії або пункт Add Object to Timeline підміню Timeline меню Modify. І, нарешті, ви можете просто натиснути комбінацію клавіш <Ctrl> + <Alt> + <Shift> + <T>.

Після додавання в анімацію вільного елементу Dreamweaver виводить попередження (Рис 2.9).

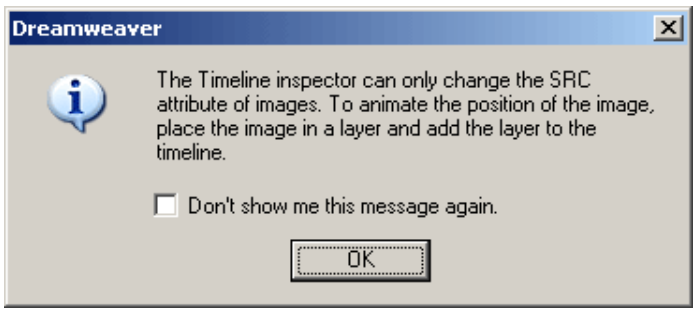

Рис. 2.8 – Попередження після додавання вільного елементу

Це означає, що можна управляти місцеположенням, розмірами і видимістю вільного елементу.

Тепер треба закрити дане попередження натисканням кнопки ОК. Якщо немає бажання більше його бачити, перед закриттям треба включити прапорець Do not show me this message again.

Нарешті, після всіх хвилювань, видно в списку панелі Timelines нову доріжку (Рис 2.10). Вона відображається у вигляді світло-синьої смуги, на якій написано ім'я вільного елементу, якому вона належить. По обидва боки цієї лінії можна помітити світлі гуртки. Це ключові точки; їх поки всього дві: початок і кінець траєкторії. Перша ключова точка - початок - знаходиться на першому кадрі; це означає, що анімація для даного елемента почнеться з першого кадру. Друга ключова точка - кінець траєкторії - знаходиться на п'ятнадцятому кадрі; там наш анімований елемент перестане рухатися.

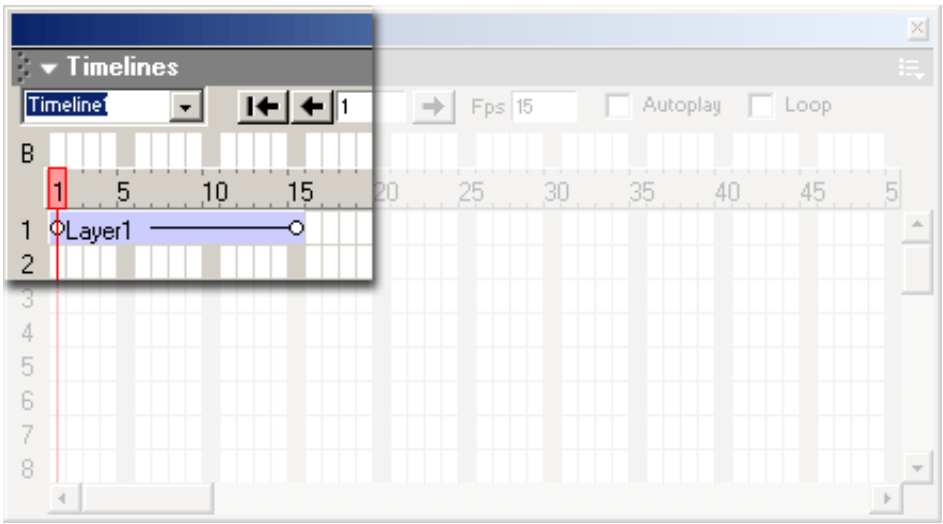

Рис. 2.10 – Нова дорожка анімації

Можна вибирати як ключові точки (в даному випадку вони виділяються темно-синім кольором), так і саму доріжку (в цьому випадку вона вся стає темно-синьою), просто клацаючи по ним мишею. Більш того, можна клацнути по будь-якого місця на доріжці анімації, виділивши таким чином будь-якої її кадр. У цьому випадку на часовій шкалі з'явиться маркер виділеного кадру (Рис 2.11).

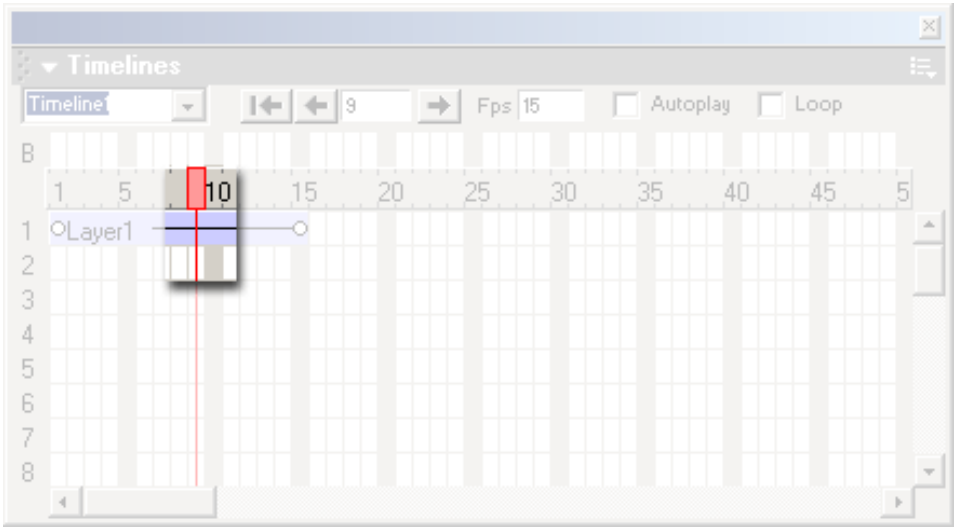

Рис. 2.11 – Маркер виділеного кадру

Можна легко змінити місце розташування доріжки анімації, якщо воно не влаштовує.

Для цього "схопити" мишею доріжку (не ключовий точки!) І перетягнути її уздовж шкали, поки вона не займе потрібну позицію. Наприклад, ви можна змусити анімацію починатися з десятого кадру, а закінчуватися - двадцять п'ятим. Також ви можете змінювати довжину доріжки, а значить, тривалість анімації, перетягуючи на необхідну позицію тепер вже ключові точки. І, зрозуміло, ви можете видалити непотрібну

доріжку, виділивши її і натиснувши клавішу <Del> або вибравши пункт Remove Object контекстного меню або підменю Timeline меню Modify [15].

Траєкторія в Dreamweaver задається наступним чином. Кожна ключова точка являє собою як би "знімок" того стану, в якому перебуватиме анімований елемент. Вибравши ключову точку, потрібно задати потрібні параметри, що вільно позиціонується (координати, розміри і видимість), а Dreamweaver їх запам'ятовує, створюючи тим самим даний "знімок". Виходячи з цих знімків, він сам обчислює всі проміжні стану анімованого елементу, в яких він буде знаходитися між ключовими точками. Тому, щоб створити анімацію будь-якої складності, досить буде задати необхідні параметри потрібних елементів сторінки тільки в ключових точках анімації. Dreamweaver сам зробить все інше.

Припустимо, заголовок буде "випливати" з нижнього правого кута сторінки і переміщатися прямо на своє законне місце. Оскільки траєкторія його руху дуже проста (пряма лінія), для її завдання виникає потреба всього в двох ключових точках. У першій ключовій точці заголовок знаходиться в нижньому правому кутку сторінки, а в другій - в її верхній частині, там, де він і повинен бути. У цих точках і повинні задати параметри нашого заголовка, точніше, всього два параметри - горизонтальну і вертикальну координати. (Наш заголовок під час руху не міняє ні розміри, ні видимість.)

Треба виділити першу ключову точку, що знаходиться на початку доріжки, клацнувши по ній мишею. Далі "захопимо" заголовок (вільне володіння елемент Header) мишею і перемістимо його в правий нижній кут сторінки, на початок його траєкторії. Тепер виділимо другу ключову точку. Заголовок вже стоїть на своєму місці. (Рис 2.12).

Добре видно тонка сіра лінія траєкторії, яка відображається у вікні документа, якщо у вікні документа виділений анімований елемент Header. Причому анімований елемент буде знаходитися в тому місці траєкторії, яке

треба виділити на доріжці анімації. Так, якщо виділити другу ключову точку, він перескочить на своє законне місце у верхній частині сторінки. А якщо виділити яку-небудь проміжну крапку між першою і другою ключовими точками, він займе відповідне цій точці місце.

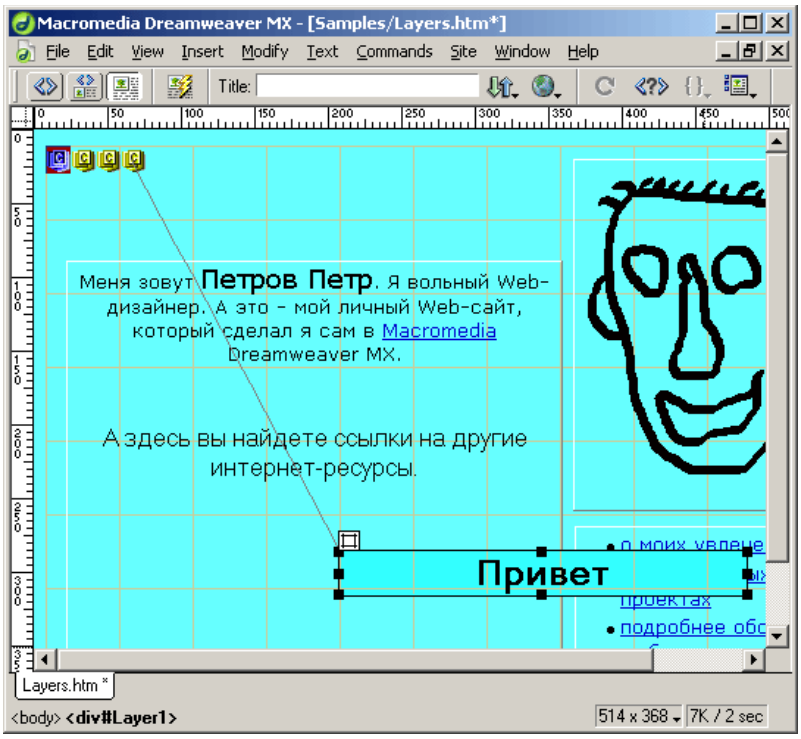

Рис. 2.12 – Траєкторія анімації

Щоб анімація запустилася відразу після завантаження сторінки Web-браузером, Dreamweaver повинен створити невеликий сценарій, який, власне, її і запускає. Але, за замовчуванням, він цього не робить. Він передбачає, що хто то хоче, щоб анімація програвалася у відповідь на дію користувача, наприклад клацання по зображенню. Але потрібно, щоб заголовок починав свій шлях відразу ж, як тільки сторінка буде завантажена. Для цього доведеться зробити відповідні установки.

Зробити їх дуже просто. Треба повернутися в панель Timelines і включимо прапорець Autoplay, розташований у верхній частині цієї панелі. Dreamweaver, за своїм звичаєм, видасть чергове попередження, що зараз в код сторінки буде доданий відповідний сценарій. Позбутися від цього попередження можна, натиснувши кнопку ОК; якщо немає бажання більше його бачити, можете включити перед цим прапорець Do not show me this message again. Ось тепер все насправді готове. Треба завантажити сторінку в Web-оглядач і переконатися в цьому.

Анімований заголовок бадьоро пробіжить задану траєкторію і замре прямо над своєю тінню. Звичайно, погано, що тінь існує незалежно від нього, але це незабаром можна виправити. Не зовсім добре також і те, що заголовок переміщається під рештою вільно позиціонуються елементами, але це можна виправити прямо зараз, просто змінивши його z-індекс. Зберігається отримана сторінка.

Dreamweaver дозволяє створювати кілька одночасно працюючих і незалежних один від одного анімацій. Вони можуть працювати як синхронно, так і асинхронно, створюючи на Web-сторінці справжні хороводи анімованих елементів. Звичайно, якщо анімованих елементів на сторінці трохи, цілком можна обійтися і однією анімацією. Але якщо їх кількість перевищує десяток, значно зручніше працювати з декількома незалежними анімаціями, кожна з яких охоплює невелику групу елементів, ніж намагатися "запхати" їх в одну анімацію.

В панелі Timelines в даний момент відображається тільки одна анімація. Щоб переключитися на іншу, треба вибрати відповідний пункт вже знайомого комбінованого списку анімацій. Щоб створити нову анімацію, треба вибрати пункт Add Timeline контекстного меню панелі Timelines. Також можна вибрати однойменний пункт підміню Timeline меню Modify.

Якщо тепер відкрити комбінований список анімацій, видно, що в ньому з'явився новий пункт.

За замовчуванням Dreamweaver привласнює при створенні нових анімацій імена виду Timeline <номер>. Якщо ж хочеться дати будь-якої анімації більш зрозуміле ім'я, вводимо його прямо в комбінований список анімацій. Інший спосіб - вибераем пункт Rename Timeline, що знаходиться в контекстному меню панелі Timelines і в підміню Timeline меню Modify. Після цього на екрані з'явиться діалогове вікно Rename Timeline (Рис 2.13).

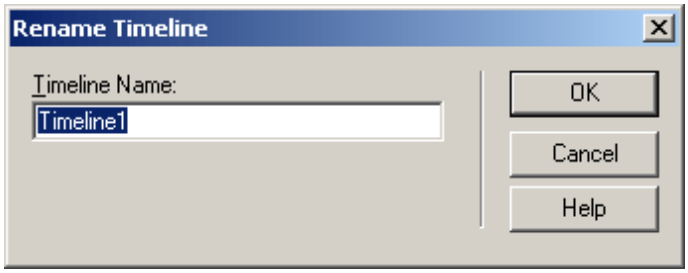

Рис. 2.13 – Діалогове вікно Rename Timeline

Щоб видалити непотрібну анімацію, перемикається на неї, використавши список анімацій, і вибираємо пункт Remove Timeline, що знаходиться в контекстному меню панелі Timelines і в підміню Timeline меню Modify. Майте, однак, на увазі, що найпершу анімацію, створювану Dreamweaver при першому відкритті панелі Timelines, видалити не можна - замість видалення Dreamweaver її просто очистить. У верхній частині панелі Timelines знаходяться кілька ще не знайомих елементів управління. Розглянемо деякі з них.

Поле введення Fps служить для завдання частоти кадрів анімації, що визначає швидкість її відтворення. Ця величина вимірюється в кадрах в

секунду (по-англійськи - frames per second, або fps). Значення за замовчуванням - 15.

Прапорець Autoplay вже знаком. Він сигналізує Dreamweaver, що в код сторінки потрібно додати сценарій, який запускає анімацію відразу ж після того, як сторінка буде завантажена в вікні Web-оглядача. За замовчуванням цей прапорець відключений, і включити його починаючі користувачі Dreamweaver часто забувають. Тому, якщо анімація чомусь не працює, перш за все, перевірте, чи включений цей прапорець.

Прапорець Loop зациклює анімацію. Якщо він включений, анімація буде програватися нескінченне число разів. Якщо його включити, Dreamweaver виведе ще одне з незліченної безлічі своїх повідомлень; поки що просто треба закрити його. За замовчуванням цей прапорець вимкнений.

І, нарешті, група з трьох кнопок і поля введення, розташована правіше комбінованого списку анімацій, служить для переміщення між окремими кадрами. Розглянемо його докладніше (Рис 2.14).

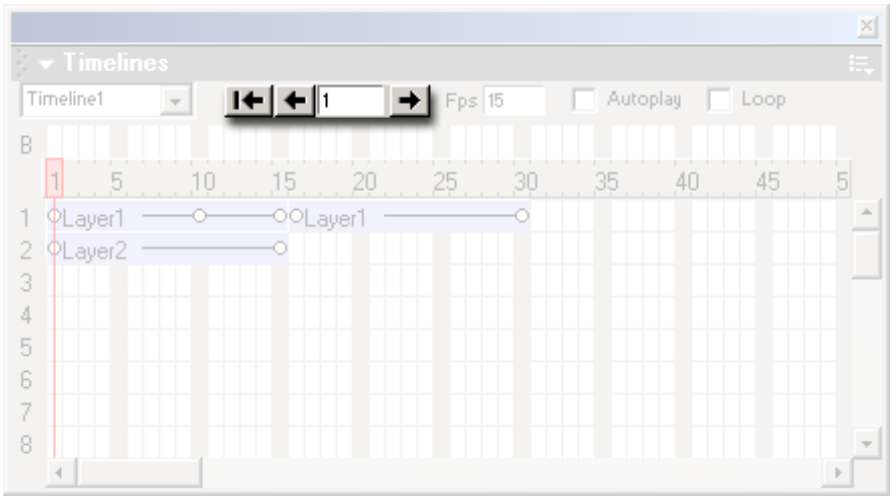

Рис. 2.14 – Набір елементів управління

Кнопки, на яких зображені спрямовані в різні боки стрілки, дозволять переміщатися по окремих кадрів анімації. Припустимо, виділили першу ключову точку анімації (початок траєкторії), що знаходиться на першому кадрі. Якщо натиснути кнопку зі стрілкою вправо, то переміститеся на другий кадр. При цьому на другому кадрі анімації в панелі Timelines буде встановлений маркер виділеного кадру, а анімований заголовок у вікні документів переміститься на один "стрибок". Клацаючи далі на кнопці зі стрілкою вправо, переміщення буде все далі до кінця траєкторії. Якщо ж клацнути на кнопці зі стрілкою вліво, то переміщення відбудеться на попередній кадр.

Кнопку зі стрілкою вправо можна використовувати для попереднього перегляду створеної анімації прямо в вікні, документа, не відкриваючи сторінку в Web-браузері. Для цього треба поставити на неї курсор миші, натиснути ліву кнопку і не відпускати. При цьому переміщення буде з кадру на кадр вперед по траєкторії; в панелі Timelines по доріжці буде переміщатися маркер, а у вікні документа - анімований елемент. Треба відпустити кнопку, коли "прокрутіца" анімацію до кінця. На жаль, більш зручного способу попереднього перегляду анімації Dreamweaver не пропонує [21].

Кнопка, на якій намальована стрілка вліво, що впирається в перешкоду, служить для швидкого переміщення на найперший кадр. Якщо потрібно переміститися прямо на якийсь кадр анімації, відомий за номером, можна просто ввести цей номер в поле введення, розташоване між кнопками-стрілками, і натиснути клавішу <Enter>. Потрібний кадр буде негайно знайдений і виділений.

Можна вирізати і копіювати музичні записи в буфер обміну Windows, а також вставляти їх в інші канали поточної анімації або взагалі в іншу анімацію на будь-який Web-сторінці. Щоб вирізати виділену доріжку, треба

вибрати пункт Cut контекстного меню або меню Edit або натиснути комбінацію клавіш <Ctrl> + <X>. Щоб скопіювати виділену доріжку, треба вибрати пункт Copy в цих же меню або натиснути комбінацію клавіш <Ctrl> + <C>. Ну, а щоб вставити знаходиться в буфері обміну доріжку в один з вже зайнятих каналів, додавши до вже існуючої доріжці, треба вибрати пункт Paste або натиснути комбінацію клавіш <Ctrl> + <V>. На жаль, вставити доріжку у вільний канал немає можливості вставити.

Dreamweaver поміщає в буфер обміну не тільки доріжку анімації, але й вільно позиціонується, для якого вона була створена, з усіма його параметрами і вмістом. І якщо вставити цю доріжку в іншу сторінку, яка вже містить вільно позиціонується з таким же ім'ям, Dreamweaver надає цю доріжку йому. Якщо ж такого елемента на сторінці немає, він створений за образом і подобою скопійованого.

Після того як вставили нову доріжку з буфера обміну, можна захотіти привласнити її іншому вільному елементу. Dreamweaver надає і таку можливість. Вибираємо пункт Change Object контекстного меню або підменю Timeline меню Modify. Після цього на екрані з'явиться діалогове вікно Change Object (Рис 2.15).

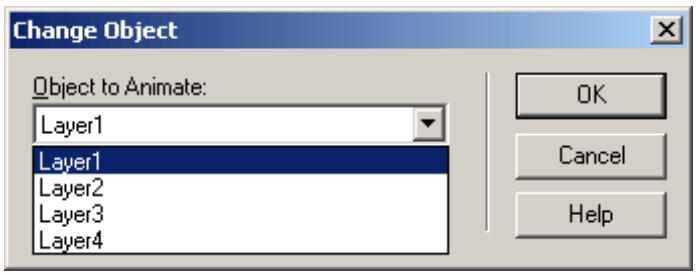

Рис. 2.14 – Набір елементів управління

3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

3.1 Особливості реалізації проекту для анімації

Основним засобом реалізації цієї атестаційної роботи є програмне середовище Adobe After Effcts CC. After Effects – програмне забезпечення компанії Adobe Systems для редагування відео і динамічних зображень, розробки композицій (композітінг), анімації і створення різних ефектів. Широко застосовується в обробці знятого відеоматеріалу (корекція кольору, пост-продакшн), при створенні рекламних роликів, музичних кліпів, у виробництві анімації (для телебачення і web), титрів для художніх і телевізійних фільмів, а також для цілого ряду інших завдань, в яких потрібно використання цифрових відеоефектів.

AE має велику кількість інструментів для створення та роботи з анімацією, але ми будемо використовувати лише їх частину для виконання поставленої задачі, а саме створення рігінгу персонажів. Для цього нам буде необхідно використовувати вирази, нуль об’єкти, графічні ілюстрації кінцівок та криві швидкості.

Вирази необхідні для того щоб зробити прив’язку декількох частин кінцівок, при цьому задавши математичну залежність що до положення між цими частинами.

Нуль об’єкти використовуються для того щоб контролювати поведінку суглобів та усієї кінцівки тільки за допомогою анімації одного контролеру.

Графічні ілюстрації в свою чергу використовуються як основні елементи які анімуються.

Криві швидкості змінюються за допомогою скрипту для того щоб точно задавати значення інтерполяції та зросту швидкості анімації в ключових кадрах, що дуже полегшує процес анімації (рис. 3.1).

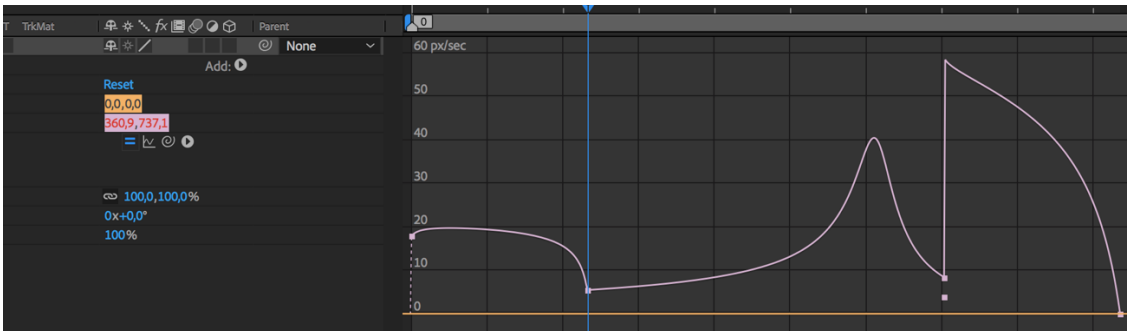

Рисунок 3.1 – Крива швидкості

3.2 Розробка проекту

З огляду на поставлені завдання необхідним критерієм для створення рігінгу є наявність кінцівок персонажа. Найкраще для цього підходять графічні елементи на основі простих кривих незмінної довжини. Такі функції як Offset path, Trim path, Stroke, Dash та ін., дозволять гнучко редагувати його зовнішній вигляд не впливаючи на самі криві, які в свою чергу будуть схильні до анімації. Для зміни параметрів цих кривих логічно використовувати базові трансформації контейнера та вбудовані контролери (рис. 3.2).

Варто зазначити, що проблема платформ також вже вирішена, оскільки скрипти представляють собою набір *jsx або *aep файлів, які інтерпретуються програмою After Effects абсолютно однаково, незалежно від операційної системи (Windows або Mac OS).

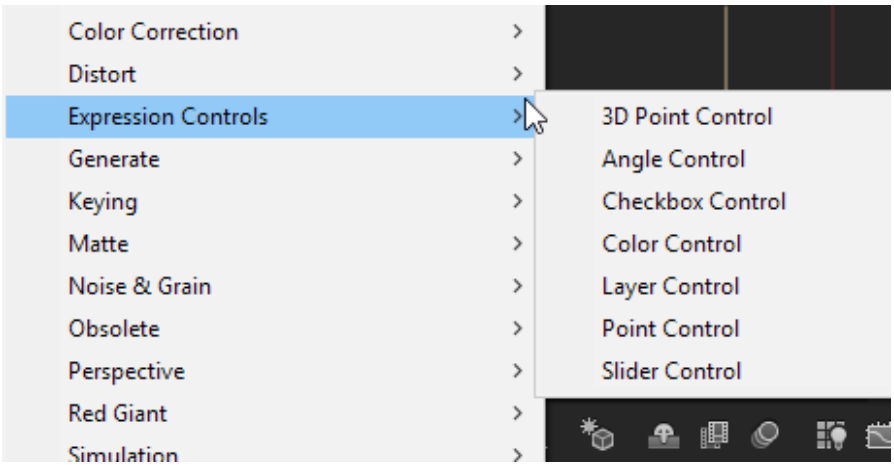

Рисунок 3.2 – Перелік вбудованих контролерів

За основу архітектури проекту можливо використовувати графічну панель з декількома вкладками інтерфейсу які виконують різні функції, в нашому випадку це будуть рігінг, редагування ключових кадрів та кривих швидкості, меню налаштування скрипта. Для більш детального налаштування рігінгу, усі основні параметри зможуть редагуватися через контролери, які будуть програмно додаватися до кожного ключового об’єкту композиції (рис. 3.3). Панель редагування являє собою набір слайдерів, які містять набори параметрів. Панель "попередньо встановлений стиль" містить 10 заготовлених налаштувань персонажу, перша з яких ручна, а решта 9 заблоковані для редагування і мають встановлені параметри [11]. Таким чином можна розробляти оновлення для контролера попередні установки, а користувачі скрипту зможуть поповнювати бібліотеку персонажів. Також треба визначити, що усі параметри редагуються, даючи гнучкі можливості для редагування параметрів кісток рігінгу.

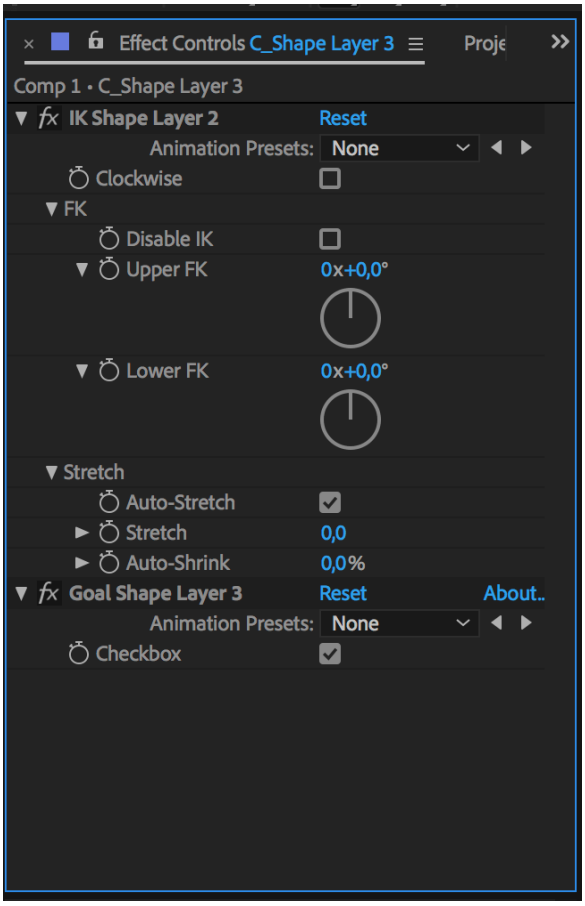

Рисунок 3.3 – Перелік панелей налаштувань

Також ми налаштовуємо зв'язку інверсної кінематики між шарами. Головною особливістю і відмінністю цих зв'язків є те, що ми застосовуємо їх не на зовнішній (рис. 3.4) контейнер трансформації об'єкта, а на внутрішній (рис. 3.5), таким чином значною мірою звільняючи простір для дій над шаром.

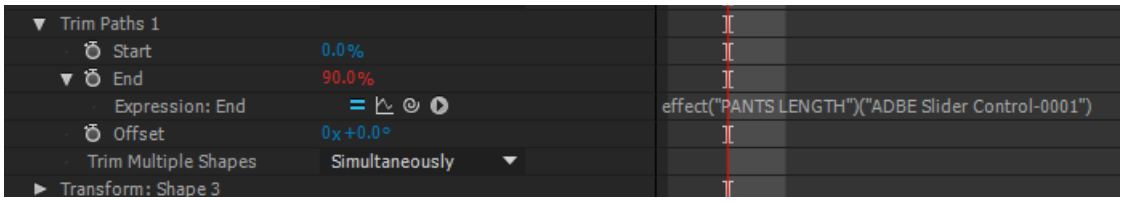

Рисунок 3.4 – Залежність довжини кінцівок від контролера

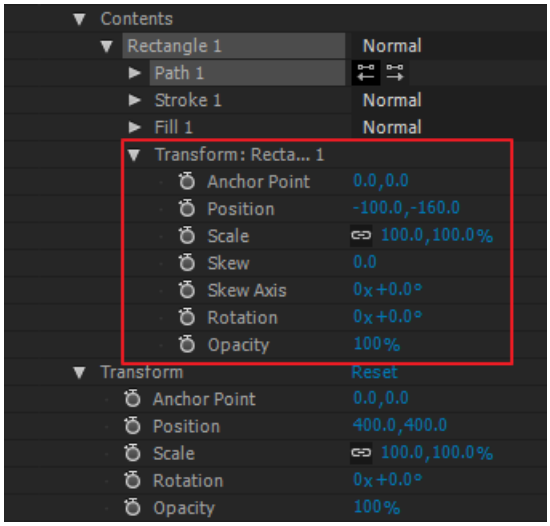

Рисунок 3.5 – Структура шару з вкладеними та глобальними трансформаціями

Цей спосіб вирішує ряд проблем притаманних плагіну DUIK, таких як, наприклад, ліктьові суглоби [22]. Тобто не впливаючи на математичну сторону зв'язку ми можемо анімувати весь шар в глобальній системі координат (рис. 3.5). Трансформації в локальній системі координат дозволяють спростити побудову залежностей між кінцівками, а саме - побудова інверсної кінематики суглобів, просто додавши до параметру позиції код, що описує залежність від кута обертання батьківської кінцівки (рис. 3.6).

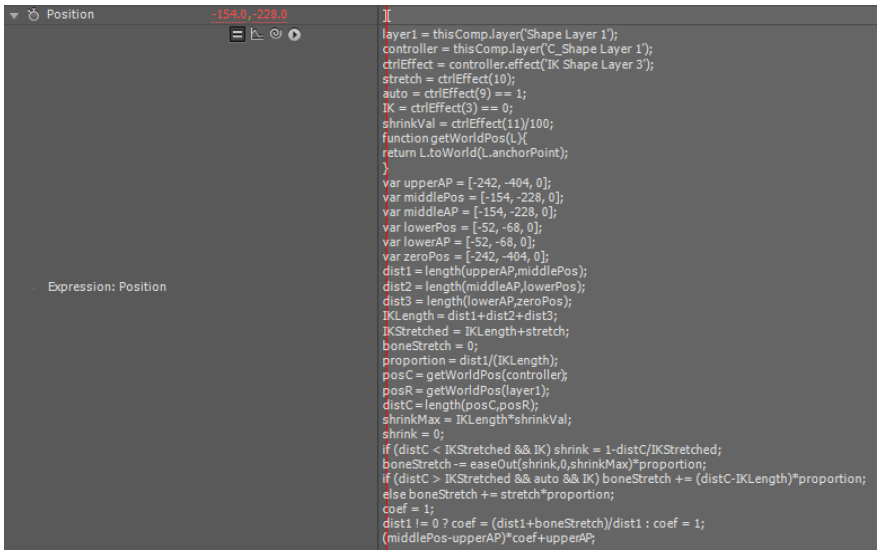

Рисунок 3.6 – Програмування інверсної кінематики відносно параметра Position

При розробці виразів для рігінгу ми використовували ряд контролерів і середовище розробки/налагодження вбудовану в програму After Effects. Так само в програмі запропонована бібліотека базових функцій і скриптів з метою прискорення розробки (рис. 3.7). При налаштуванні зв'язків між об'єктами використовували стандартну функція Rope control.

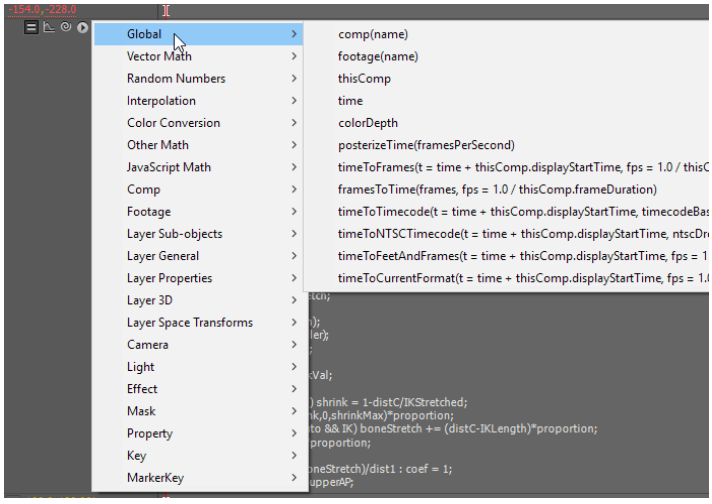

Рисунок 3.7 – Бібліотека базових функцій

Всі анімовані трансформації об'єктів пов'язані з панеллю-контролером, що дозволяє накласти певні обмеження на анімацію елементів і уникнути

аномальних деформацій. Нижче наведено приклад побудови залежності обертання двох суглобів через панель контролерів та обмеження на ступінь обертів.

```
r = effect ("ROTATION") ("ADBE Slider Control-0001");
k = linear (r, -90,90, + 10,0);
if(k>10) {s=10;}
if(k<0) {s=0;}
[Transform.position [0] + k, transform.position [1]]
```

Так само до батьківських елементів кінцівок застосовані вираження описують затухаючі коливання для додання природності рухам.

```
amp = .06;
freq = 3;
decay = 7;
n = 0;
k = amp*freq;
s = amp/decay;
if (numKeys > 0){
n = nearestKey(time).index;
if (key(n).time > time){
n--; } }
if (n == 0){
t = 0;
}else{
t = time - key(n).time; }
if (n > 0){
v = velocityAtTime(key(n).time - thisComp.frameDuration/10);
value + v*amp*Math.sin(freq*t*2*Math.PI)/Math.exp(decay*t);
}else{
value; }
```

3.3 Структура проекту та програмна реалізація

Оскільки JavaScript є об’єктно-орієнтованою мовою програмування, для реалізації алгоритмів роботи скрипту користуватиметься такими принципами, як спадкоємство, поліморфізм і інкапсуляція [23].

Також треба зазначити що основним логічним елементом скриптінгу є функція, бо кожна операція, що оптимізує роботу в After Effects, являє собою алгоритм декількох операцій, які йдуть одна за одною. Наприклад ця функція відповідає кнопці створення інверсної кінематики:

```
function ik()
    {
    var calques = getCurrentLayers();
    if (!calques) return null;
    if (calques.length < 2 || calques.length > 5){
    alert(tr("Select the bones and the controller before creating IK"));
    return;
    }
    panoik.hide();
    ikPanel.show();
    //prepIK
    app.beginUndoGroup(tr("Duik - prepare IK"));
    var ikRig = Duik.utils.prepIK(calques);
    app.endUndoGroup();
    if (ikRig.type == 0) return;
    if (ikRig.type == 1 && ikRig.goal == null){
    ikType1Group.show();
    ikType2Group.hide();
    ikType3Group.hide();
    ikType4Group.hide();
    ik3DGroup.hide();
    }else if (ikRig.type == 1 && ikRig.goal != null){
    ikType1Group.hide();
```

```
kType2Group.show();
ikType3Group.hide();
ikType4Group.hide();
ik1GoalButton.value = true;
ik2LayerButton.value = false;
ik3DGroup.enabled = false;
if (ikRig.threeD){
ik3DGroup.show();
ikFrontFacingButton.value = ikRig.frontFacing;
ikRightFacingButton.value = !ikRig.frontFacing;
}else{
ik3DGroup.hide();
}
}else if (ikRig.type == 2 && ikRig.goal == null){
ikType1Group.hide();
ikType2Group.show();
ikType3Group.hide();
ikType4Group.hide();
ik1GoalButton.value = false;
ik2LayerButton.value = true;
ik3DGroup.enabled = true;
if (ikRig.threeD){
ik3DGroup.show();
ikFrontFacingButton.value = ikRig.frontFacing;
ikRightFacingButton.value = !ikRig.frontFacing;
}else{
ik3DGroup.hide();
}
}else if (ikRig.type == 2 && ikRig.goal != null){
ikType1Group.hide();
ikType2Group.hide();
ikType3Group.show();
```

```
ikType4Group.hide();
ik3LayerButton.value = false;
ik2GoalButton.value = true;
ik3DGroup.enabled = true;
if (ikRig.threeD){
ik3DGroup.show();
ikFrontFacingButton.value = ikRig.frontFacing;
ikRightFacingButton.value = !ikRig.frontFacing;
}else{
ik3DGroup.hide();
}
}else if (ikRig.type == 3 && ikRig.goal == null){
ikType1Group.hide();
ikType2Group.hide();
ikType3Group.show();
ikType4Group.hide();
ik3LayerButton.value = true;
ik2GoalButton.value = false;
ik3DGroup.enabled = false;
if (ikRig.threeD){
ik3DGroup.show();
ikFrontFacingButton.value = ikRig.frontFacing;
ikRightFacingButton.value = !ikRig.frontFacing;
}else{
ik3DGroup.hide();
}
}else if (ikRig.type == 3 && ikRig.goal != null){
ikType1Group.hide();
ikType2Group.hide();
ikType3Group.hide();
ikType4Group.show();
ik3DGroup.hide();
```

```
}
ikCreateButton.onClick = function() {
ikRig.frontFacing = ikFrontFacingButton.value;
if (ikType2Group.visible){
if (ik2LayerButton.value && ikRig.type == 1){
ikRig.layer2 = ikRig.goal;
ikRig.goal = null;
ikRig.type = 2;
}else if (ik1GoalButton.value && ikRig.type == 2){
ikRig.goal = ikRig.layer2;
ikRig.layer2 = null;
ikRig.type = 1;
}
}else if (ikType3Group.visible){
if (ik3LayerButton.value && ikRig.type == 2){
ikRig.layer3 = ikRig.goal;
ikRig.goal = null;
ikRig.type = 3;
}else if (ik2GoalButton.value && ikRig.type ==3{
ikRig.goal = ikRig.layer3;
ikRig.layer3 = null;
ikRig.type = 2;
}
}
app.beginUndoGroup(tr("Duik - IK"));
ikRig.create(ikLockAndShyButton.value);
app.endUndoGroup();
ikPanel.hide();
panoik.show();
}
}
```

Даний код містить інструкції, які оптимізують кілька кроків механічних операцій в After Effects, та зводить їх до простого натискання на одну кнопку (рис. 3.8).

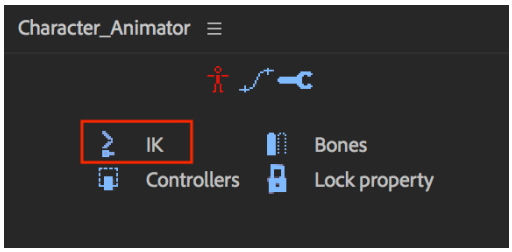

Рисунок 3.8 – Розроблений скрипт у панелі вікон

3.4 Демонстрація створення рігінгу

Для демонстрації роботи алгоритму рігінгу створюємо нову композицію у програмному середовищі Adobe After Effects та запускаємо розроблений скрипт під назвою «Character Animator» (рис. 3.9). Після цього у графічному інтерфейсі скрипту вибираємо необхідну функцію

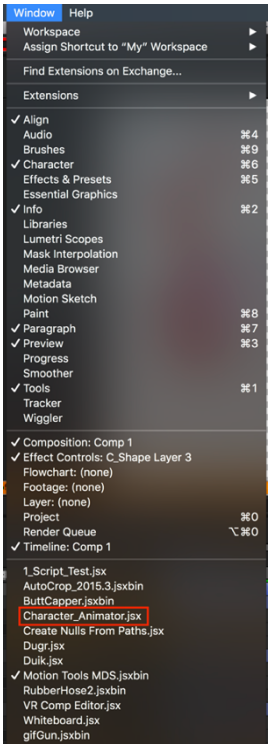

Рисунок 3.9 – Розроблений скрипт у панелі вікон

Після цього у графічному інтерфейсі скрипту вибираємо необхідну функцію (рис. 3.10). В нашому випадку необхідно спочатку створити нуль контролер до якого буде прив’язуватися ріг (рис. 3.11).

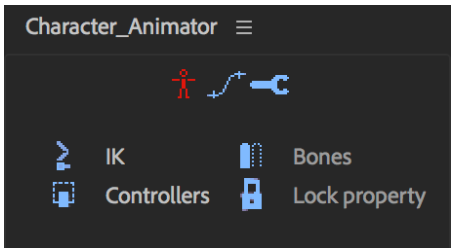

Рисунок 3.10 – Графічний інтерфейс скрипту

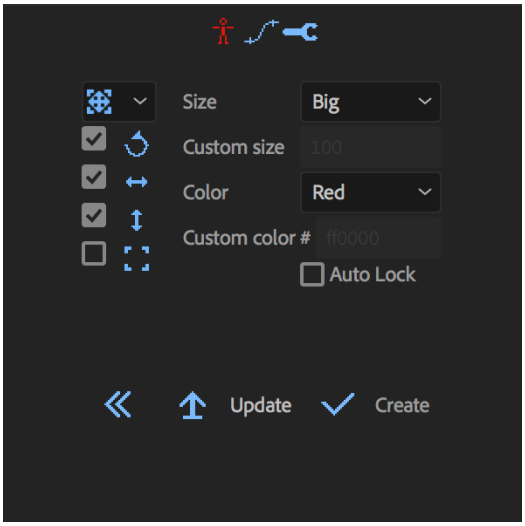

Рисунок 3.11 – Меню функції «Controller»

На попередньому скріншоті можливо побачити що настройка нуль контролер дуже гнучка, вона дозволяє управляти розміром, формою та кольором [24].

Далі ми створюємо ріг інверсної кінематики для вибраної кінцівки. Для цього необхідно виділити необхідні шари, нуль контролер та натиснути на кнопку «IK» в панелі скрипту (рис. 3.12).

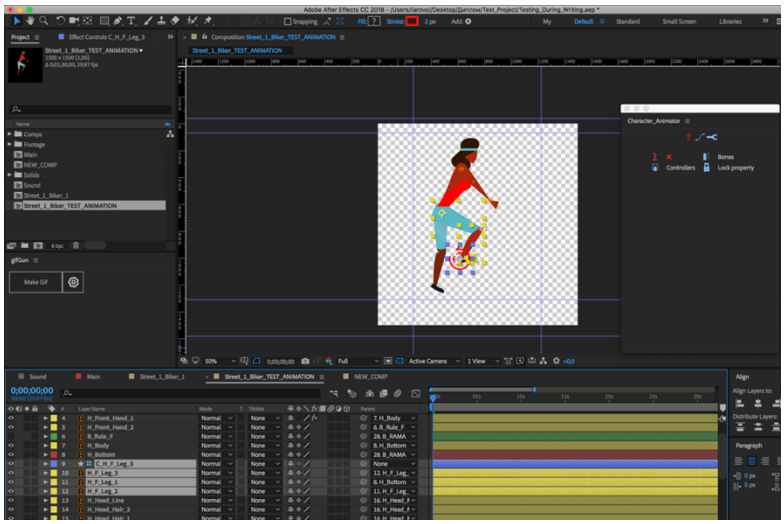

Рисунок 3.12 – Процес створення інверсної кінематики

Після цього необхідно у панелі скрипту зробити вибір який тип рігінгу ми обираємо. Різниця полягає у тому скількома суглобами буде управляти нуль контролер після створення рігінгу (рис. 3.13).

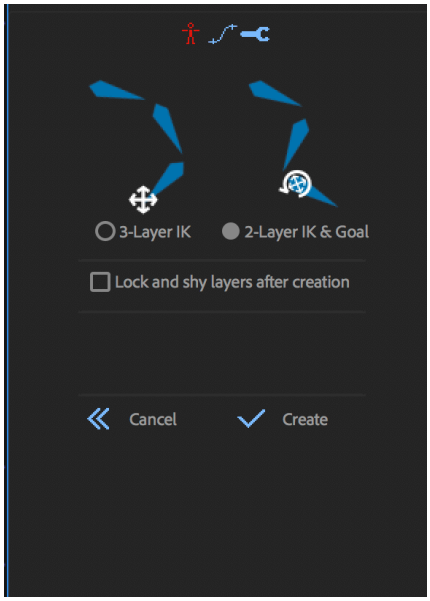

Рисунок 3.13 – Вибір рігу у панелі скрипту

3.5 Демонстрація редагування ключових кадрів

Також створений скрипт дозволяє редагувати ключові кадри. А саме їх тип та криву швидкості. Це дуже економить час, бо оптимізує рутинну задачу

редагування кадрів. Об’єднуючи декілька кроків в одне просте натискання кнопки. Інтерфейс також був спеціально розроблений для зручного використання (рис. 3.14).

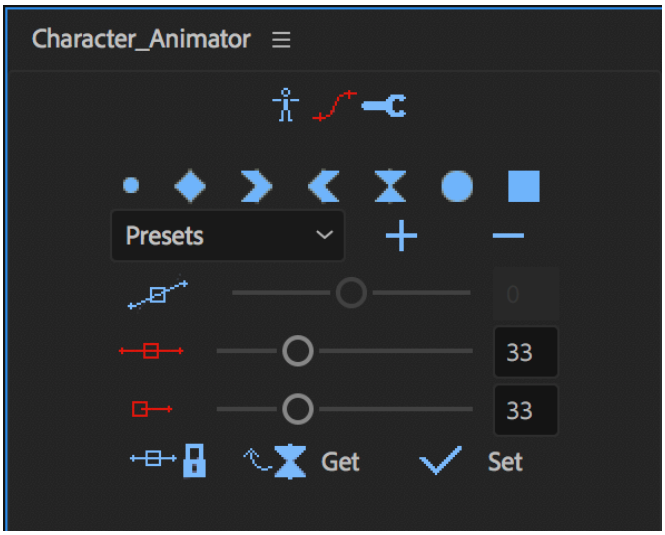

Рисунок 3.14 – Інтерфейс меню редагування ключових кадрів

Користувач має декілька можливостей під час користування цім меню. По-перше він може змінювати тип кадру. Для цього необхідно натиснути на одну з кнопок у першому ряду інтерфейсу [20]. Доступні такі варіанти: Roving, Linear Interpolation, Easy In, Easy Out, Easy Ease, Auto Bezier, Hold. Натиснувши на одну з кнопок, вибраний кадр автоматично змінюється на необхідний (рис. 3.15).

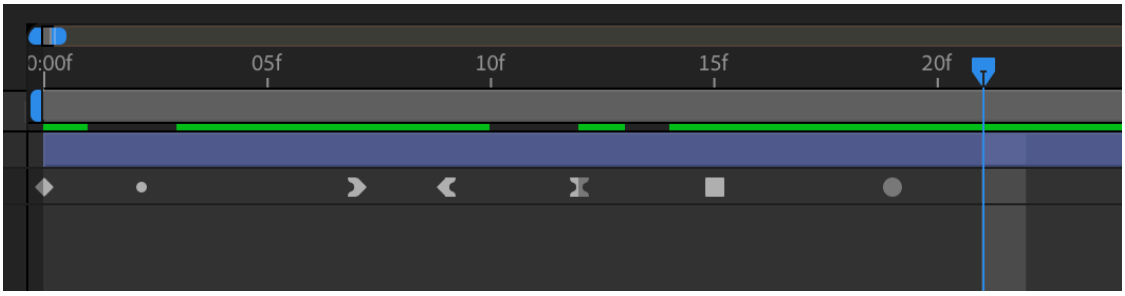

Рисунок 3.15 – Змінені кадри на таймлайні

Наступні повзунки служать для роботи с кривими швидкості. Ми можемо зберегти необхідний прісет з данними інтерполяції та потім

використовувати його багаторазово. Для того щоб виставити необхідне значення зростання та спадання швидкості просто необхідно змінити показник біля другого та третього повзунків. Перший же повзунок відповідає за початкову швидкість ключового кадру. Роботу усіх цих елементів може швидко та наглядно продемонструвати крива швидкості (рис. 3.16).

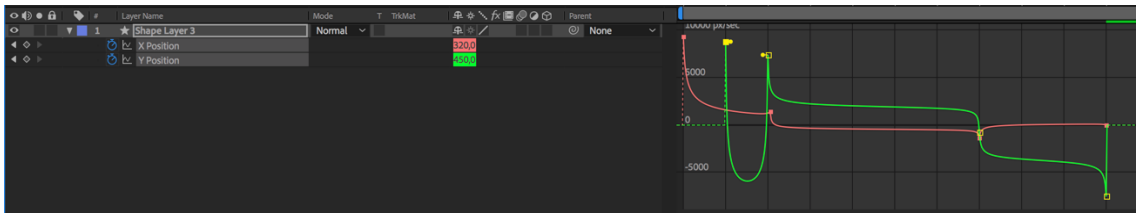

Рисунок 3.16 – Крива швидкості відредагована за допомогою скрипту

ВИСНОВКИ

В даній атестаційній роботі було показано як використання програмування може оптимізувати та покращити можливості користувача при створені персонажної анімації в Adobe After Effects.

Було проведено детальний аналіз такоії області діяльності як моушен дизайн, зокрема персонажної анімації. Через порівняння різних способів анімації були виявлені переваги та недоліки кожного із методів. Після чого ми змогли встановити що потрібно реалізувати в нашому скрипті для рішення цих недоліків, зберігши переваги кожного із методів.

Був розроблений скрипт, що складається з декількох незалежних компонентів – інструменти для створення рігу кінцівок персонажа, роботи із ключовими кадрами та кривими швидкості . За допомогою даної системи, користувачі можуть автоматично робити ріг кінцівок та зручно працювати із ключовими кадрами анімації.

Також був розроблений зручний графічний інтерфейс програми, який наглядно відображає усі дії зроблені користувачем. А панель анімації ключових кадрів дозволяє швидко отримувати значення інтерполяції кожного з кадрів.

Розроблене програмне забезпечення буде мати цінність в першу чергу для моушн дизайнерів працюючих з персонажною анімацією. Тепер вони зможуть не витрачати часу на створення контролерів та зав’язків між суглобами, та написання виразів інверсної кінематики. Розроблене ПО зручне в використанні та легке для розуміння.

ПЕРЕЛІК ПОСИЛАНЬ

1. Дж. Лі, Б. Уер. Тривимірна графіка та анімація – 2-е вид. [Текст] / М. :Вільямс, 2002. - 640 с.

2. Блок Брюс. Визуальное повествование. Создание визуальной структуры фильма, ТВ и цифровых медиа [Текст] / Пер. с англ. Юлии Чиликиной под ред. Виктора Монетова, Максима Казючица – М.:ГИТР, 2012 – 320 с.: илл.

3. Карвер Li, Секрети відеовиробництва в Adobe Premier и After Effects (+ DVD-ROM) [Текст] / Ii Карвер, Стен. // – М.: Діалектика. 2004. – 296 с.

4. Video-copilot [Електроний ресурс] – Режим доступу: www/URL: http://www.videocopilot.net

5. Red Giant [Електроний ресурс] – Режим доступу: www/URL: https://www.redgiant.com

6. Adobe Systems Incorporated. Adobe After Effects CS6 Scripting Guide – 2012 – 293 pages.

7. Карась І.В., Савченко Л.М., Матюшенко М.В., Воронцова Д.В., Створення пресетів на основі динаміки частинок. Материалы молодежной школы-семинара 2 Международной научно-технической конференция «Полиграфические, мультимедийные и web-технологии» [Текст] (PMW-2017) 16-22 мая 2017 года г. Харьков изд-во ООО "Типография Мадрид", 2017, Том 2, с.266-268

8. Dan Ebberts’ AE Expressions and Scripting Resource [Електроний ресурс] – Режим доступу: www/URL: http://www.motionsctipt.com

9. Онлайн-бібліотека Video tutorials [Електроний ресурс] – Режим доступу: www/URL: http://www.videotuts.ru

10. Bancroft Tom. Creating Characters with Personality: For Film, TV, Animation, Video Games, and Graphic Novels. — Watson-Guptill, 2006. — ISBN 978-0-8230-2349-3.

11. Lasseter, John (1987-07). «Principles of Traditional Animation applied to 3D Computer Animation». ACM Computer Graphics 21 (4): pp. 35–44. DOI:10.1145/37402.37407.

12. Mattesi Mike. Force: Dynamic Life Drawing for Animators, Second Edition. — Focal Press, 2002. — ISBN 978-0-240-80845-1.

13. JavaScript Tools Guide CS6. Adobe Systems Incorporated, 2012, 202 pages.

14. Gizma [Електроний ресурс] – Режим доступу: www/URL: http://gizma.com/easing/

15. W3 Schools – Java Script Tutorial [Електроний ресурс] – Режим доступу: www/URL: http://www.w3schools.com/js/

16. Adobe DreamWeaver [Електроний ресурс] – Режим доступу: www/URL: https://www.adobe.com/ua/products/dreamweaver.html

17. Adobe Illustrator [Електроний ресурс] – Режим доступу: www/URL: https://www.adobe.com/ua/products/illustrator.html

18. VideoSmile.ru [Електроний ресурс] – Режим доступу: www/URL: https://videosmile.ru/lessons/read/origami.html

19. Motionographer.com [Електроний ресурс] – Режим доступу: www/URL: https://motionographer.com/

20. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item–based collaborative filtering recommendation algorithms. In Proceedings of the Tenth International World Wide Web Conference pp. 285–295 (2001)

21. Lops P., De Gemmis M., Semeraro G. Content-based recommender sys-tems: state of the art and trends. In: Ricci F., Rokach L., Shapira B. (eds.) Recommender systems handbook, pp. 73-105. Springer, Hedelberg. 2011. pp.73-106.

22. Entagma [Електроний ресурс] – Режим доступу: www/URL: https://entagma.com/

23. Aescripts.com [Електроний ресурс] – Режим доступу: www/URL: https://aescripts.com/

24. Scripting Guide [Електроний ресурс] – Режим доступу: www/URL: http://docs.aenhancers.com/