AN APPROACH OF MOVIE CLIP OBJECT DEVELOPMENT VIA JAVASCRIPT INTENDED FOR LEARNING APPLICATIONS

VALENTIN T. ATANASOV

ABSTRACT: *Highlyinteractive generation, as it was formed today, demands certain ways of knowledge perception in the lecture rooms and labs. One of the key factors in the digital based learning process, alongside interactivity and Atychiphobia, is dynamic visual data presentation forming phenomenon called animation. This paper expresses an approach of visual data animation via movie clip object development with JavaScript language. The developed movie clip object poses common object features, which allowing its programming integration without user-agent's plugins or external virtual machines. The presented approach could be used in WEB based learning apps, presentations and various HTML 5 compatible tools.*

KEYWORDS: *Movie clip, JavaScript animated object, HTML 5 animation, Highlyinteractive generation.*

DOI: https://doi.org/10.46687/MT22VAL

ЕДИН ПОДХОД ЗА РАЗРАБОТКА НА АНИМИРАЩ ОБЕКТ С ИЗПОЛЗВАНЕТО НА JAVASCRIPT ЗА ЦЕЛИТЕ НА ОБУЧАВАЩИ ПРИЛОЖЕНИЯ

ВАЛЕНТИН Т. АТАНАСОВ

АБСТРАКТ: *Високоинтерактивното поколение, формирано днес, предявява определени начини за възприемане на знанията в лекционните зали и лабораториите. Един от ключовите фактори в дигитално базирания учебен процес, наред с интерактивността и страхът от провал, е динамичното изобразяване на визуален тип данни, представящо феномен, наречен анимация. Тази статия представя един подход за анимация на визуален тип данни чрез разработване на анимиращ обект с използването на езика JavaScript. Разработеният анимиращ обект притежава общи характеристики на обекта, което позволява неговата програмна интеграция без добавки на клиентския агент или външни виртуални машини. Представеният подход може да се използва в УЕБ базирани обучаващи приложения, презентации и различни HTML 5 съвместими инструменти.*

Приети определения:

Анимация – Промяна параметрите на визуален обект във времето, която създава зрително възприятие, пораждащо „*психологически*" ефект на непрекъснатост в изменението на тези параметри.

JavaScript виртуална машина - Автономен програмен компонент за модифициране елементите на документния обектен модел чрез използване на скриптов език, базиран на стандарта ECMAScript.

1 Въведение

Базирани на съвременната характеристика на обучаваните [1,2,3] и влиянието на комплексни фактори като високата степен на високотехнологичното обкръжение в ежедневието и в лекционните зали, както и влиянието на страха от провал в учебния

процес[4,5], очертават определени рационални решения в обхвата на които е използването на мотивираща и завладяваща нестресова среда за обучаваните, изграждана чрез компютърно базирани обучаващи игри или приложения с елементи на игровизиране.

Разглежданото технологично решение се опира на тезата за интеграция на дидактически принципи в игровия подход като ефективен метод за преподаване. Основни характеристики като мотивация и целеустременост у обучаваните в процеса на приложение на игровия подход, определят степента на ефективност на учебния процес за постигането на педагогическите цели. Но, при разглеждането на този подход в обучението на съвременната образователна среда, следва да бъдат отчитани фактори, влияещи върху самото технологично реализиране на тези цифрови игрови похвати. Предмет на настоящата статия е представянето на един подход за формирането на ефикасен процес при изграждането на цифрово представяни знания, чрез разглеждането на определен сегмент от мултимодалния подход – изграждане на модалност с визуален тип данни посредством анимирането им чрез език за програмиране JavaScript.

2 Области на приложение

В Шуменски университет „*Епископ Константин Преславски*“ има внедрена цифрово базирана среда за обучение чрез университетска платформа за обучение в електронна среда. Изградеият цифрово базиран учебен процес притежава следните характеристики:

- *Базиран на УЕБ приложен програмен интерфейс;*
- *Базиран на JavaScript виртуална машина;*
- *Ориентирани към съдържание/ресурси;*
- *Широка честотна лента;*
- *Поддържани формати на съдържание;*
- *Некомерсиалност;*
- *Прилагане на принципа – на всяко устройство, от всяко място, по всяко време.*

Следва да бъде отчетено, че някои виртуални машини, чиято функционалност се реализира от клиентски агенти(браузери), еволюираха, но други бяха спрени от поддръжка и развитие[8], въпреки водещите им технологични решения, и то не по технологични причини или аргументи.

В настоящата статия се предлага едно възможно решение за анимиране на визуален тип данни, базирано на технология, внедрена и поддържана в широко използваните клиентски агенти. Съществуват известни решения за анимиране на визуални типове данни чрез скриптови езици, но с известно изключение[11], използващо „*чист*“ JavaScript, останалата част от тях използват езикови производни на JavaScript в специализирани среди за разработка или изискването за включване на специализирани библиотеки[12].

Предлаганото технологично решение се реализира от JavaScript виртуална машина, интегрирана в даден клиентски агент. Това обстоятелство, заедно с водещия дял при използването на този език[6,7] от разработчиците на приложения, определя ефективността на предлагания подход. Възможни области на приложение са инженерните дисциплини, в чиито лекционни или семинарни занятия би могло да бъде използвано това решение за изграждане на ефективен мултимедиен продукт за постигане по-висока степен на разбиране в процеса на възприятие у обучаваните.

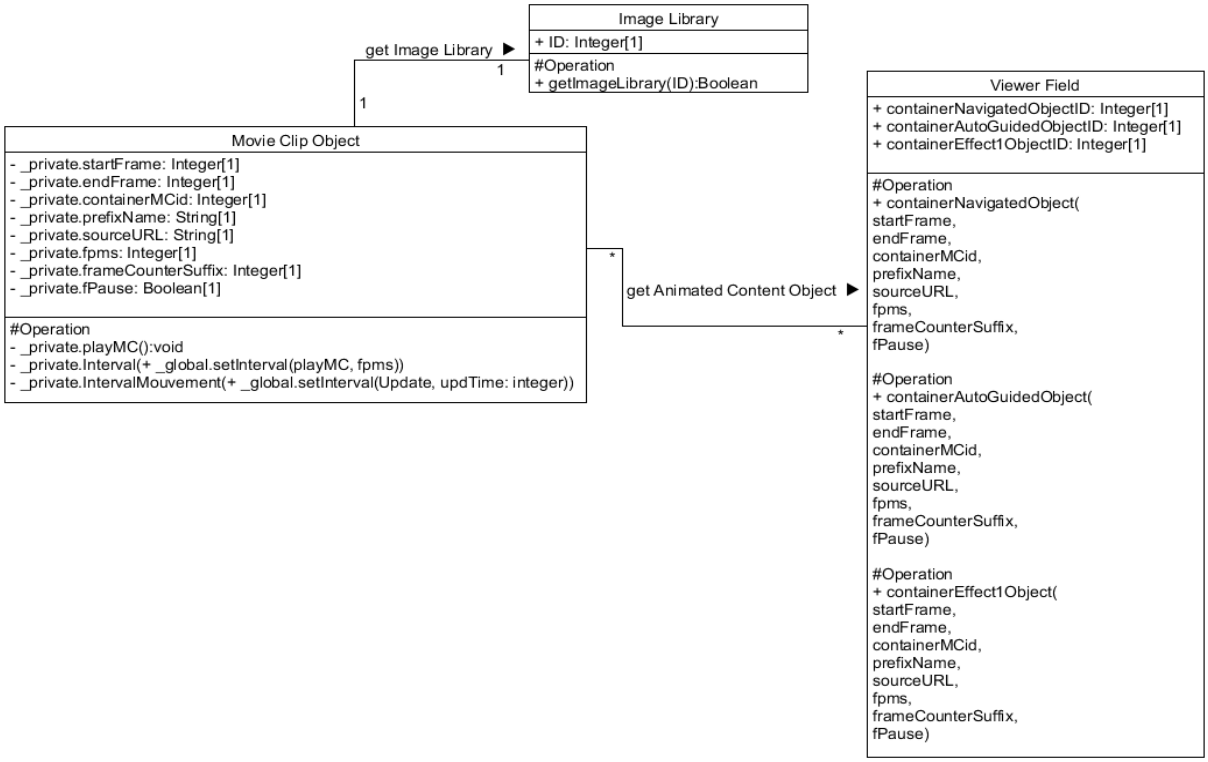
3 Дефиниране на технологично програмно решение

Това решение предполага подход, почиващ на общите принципи при разработването на системи, без оглед на мащабите и сложността на задачите.

Като първи етап от технологичното програмно решение се разглежда изграждането на библиотека от изображения, които ще бъдат анимирани. Този тип визуални данни, спадащ към основния тип данни, следва да бъде разработен с използването на някои от програмните инструменти за създаване и обработка на изображения. В обхвата на настоящата статия не се разглеждат тези програмни инструменти, а се отбелязват няколко основни изисквания към тези изображения:

- *Разрешаващата способност(резолюция) да осигурява детайлност на изображението, предоставяща достатъчна информация за неговото предназначение;*
- *Разрешаващата способност да осигурява плавен процес на визуализация при анимирането(зареждане, покадрово изобразяване, позициониране);*
- *Файловият формат на изображението да бъде съвместим с поддържаните формати на клиентския агент и от стандартите ECMAScript[10] и HTML5xx;*
- *Препоръчително е в изграденото изображение използването на канал за прозрачност (alpha), с цел постигане на възприятие близко до възприятията на високоинтерактивното поколение.*

Като втори етап се разглежда разработката на програмния код. Тази разработка се основава на добрата практика „*обектно ориентирано програмиране*“. Дефинират се класове, чрез които се реализира механизмът за анимиране на визуалния тип данни (Фиг.1.1.).



Фигура 1.1. Диаграми на класове „*Анимиращ обект*“ и „*Визуализиращо поле*“

В представените на фиг.1.1 диаграми на класове, диаграмата на класа „*Библиотека изображения*“ (Image library) не е разгърната цялостно, не е описан типът обект детайлно,

а само е изведена концептуално статичната връзка/взаимоотношение с класа „*Анимиращ обект*“.

В основата на предлаганата концепция за програмно разработван механизъм за анимиране на визуален тип данни лежат следните принципи:

- *Крайно множество от изображения;*
- *Унифицирана именна конвенция на файловете;*
- *Константен път за достъп до съдържащите изображения файлове;*
- *Използване на предефинирани времеви измерители (Timers) на стандарта HTML 5.xx, извикващи методите setTimeout() и setInterval(), позволяващи на разработчиците на организират във времето дадени процеси[9].*

Основано на горното се предлага следните програмни алгоритми (1.1. и 1.2) и код (1.3.) при разработката на анимиращия обект:

(1.1.)
```
function MovieClipGameObj(startFrame, endFrame, containerMCid, prefixName,
sourceURL, fpms, frameCounterSuffix, fPause){

  this.playMC = function(){
       if(!flagPause){
          frameCounterSuffix++;
          if(frameCounterSuffix > endFrame){
              frameCounterSuffix = startFrame;
       };
       this.imageURL = sourceURL;
       this.imageFileName = prefixName + frameCounterSuffix + fileExtention;
       document.getElementById(containerMCid).style.backgroundImage =
       this.imageURL + this.imageFileName + ")";
       }
  }

  this.intervalMain = setInterval(this.playMC, fpms);

}
```

(1.2.)
```
const MainAnimatedObject = new MovieClipGameObj(param1:integer,
param2:integer, param3:string, param4:string, param5:string, param6:integer,
param7:integer, param8:boolean)

const AuxilaryAnimatedObject = new MovieClipGameObj(param1:integer,
param2:integer, param3:string, param4:string, param5:string, param6:integer,
param7:integer, param8:boolean)
```

(1.3.)
```
<div id="NavigatedObject"> </div>
<div id="AutoGuidedObject"> </div>
```

С предложените алгоритми и код биха могли да се реализират множество анимиращи обекти в среда на ДОМ документ. Представеният в (1.2.) алгоритъм описва възможността за изграждане на множество анимиращи обекти. В дадения пример (1.2.) са представени два анимиращи обекта.

В настоящата статия не се представят всички компоненти на ДОМ документа, интегриращ разглеждания механизъм, а само конкретната технологична кодова част, оперативно изпълняваща процеса на анимирането на визуални данни. Но, следва да се отчита наличието на възможно ограничение, възникващо при значителен брой анимиращи обекти в клиентския агент, които крайните изчислителни ресурси на компютърната система биха обслужвали.

4 Заключение

С цел формиране на критична емпирична база за анализ, се подхожда към всяка конкретна реализация на разглеждания метод, в случая, започвайки с използването на набор от индивидуални изображения. В разработвания вариант на УЕБ базирано игрово обучаващо приложение, с използване на анимиращ обект, са реализирани и функционалности за управление на позиционирането на анимиращите обекти.

Предвижда се бъдещо изследване с използването на листове с многосъставни изображения(sprite sheet) и на атласи, изградено от листове с многосъставни изображения (sprite atlas). Предвижда се и сравнителен анализ на предлагания подход със съществуващи програмни решения.

ЛИТЕРАТУРА:

[1] Атанасов, В., „Измерване и оценка на образователна интерактивност на уеб базирани обучаващи приложения“, Международна научна конференция „Форум отбранителни технологии 2020“, стр. 245-252, 2020, ISSN 2367-7902

[2] Atanasov, V., Ivanova, A., Student modelling in a web-based platform for learning games composing, The 13th International Scientific Conference eLearning and Software for Education, Bucharest, vol.2, 2017, pp. 272-279, https://doi.org/10.12753/2066-026x-17-040a

[3] Atanasov, V.,T., An algorithm and functional model of game based knowledge level test, Collection of papers 2019, Annual collection of papers 2019, National Military University Vasil Levski Press, Volume I, pp.29-36, 2019, ISSN 1312 6148

[4] Learning academic resilience at Stanford, 2018, https://parents.stanford.edu/2018/04/09/learning-academic-resilience-at-stanford/(Посетен на 01.03.2022

[5] 7 ways to deal with your fear of failure, 2018, (https://www.ucl.ac.uk/students/news/2018/may/7-ways-deal-your-fear-failure) (Посетен на 01.02.2022)

[6] Usage statistics of JavaScript for websites, https://w3techs.com/technologies/details/pl-js (Посетен на 01.02.2022)

[7] Most used programming languages among developers worldwide, as of 2021, https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/ (Посетен на 01.02.2022)

[8] Adobe Flash Player EOL General Information Page, https://www.adobe.com/products/flashplayer/end-of-life.html (Посетен на 01.03.2022)

[9] HTML Living Standard, 2022, https://html.spec.whatwg.org/multipage/timers-and-user-prompts.html#dom-setinterval-dev (Посетен на 01.03.2022)

[10] ECMAScript® 2021 Language Specification, 2021, https://www.ecma-international.org/wp-content/uploads/ECMA-262_12th_edition_june_2021.pdf (Посетен на 01.03.2022)

[11] How to convert a Flash/After Effects animation into a JSMovieclip animation, 2020, https://gist.github.com/jeremypetrequin/14a77d4652c096a1fac7 (Посетен на 23.05.2021)

[12] MovieClip Class, https://createjs.com/docs/easeljs/classes/MovieClip.html (Посетен 12.12.2021)