

УДК 004.92

АВТОМАТИЗИРОВАННАЯ ОБРАБОТКА ФОТОГРАФИЙ В ADOBE PHOTOSHOP

Миниярова Л.В., Габдрахманов А.Н.

*Башкирский государственный педагогический университет им. М. Акмуллы, Уфа,
e-mail: ainur1@bk.ru*

Результат данной научно-исследовательской работы – скрипт, выполняющий автоматизированную обработку изображения. Скрипт написан на языке JavaScript, разработка выполнялась в программе AdobeExtendScriptToolkitCS5.5. Программным обеспечением для обработки изображения взят AdobePhotoshopCC. Данная работа направлена на обработку большого массива фотографий в один клик. Продукт был разработан по просьбе нашего знакомого-дизайнера, сталкивающимся с подобной задачей довольно часто. Проанализировав литературу и программные утилиты по данной теме, нам и заказчику не удалось найти продукт удовлетворяющий этим требованиям. Имеются утилиты по обработке фотографий, установки рамки, задания определенного цвета, но нет ни одной которая при этом выравнивала картинку по размеру и как бы прорисовывала картинку в перспективе, с заданными параметрами искажения, создавая иллюзию просмотра картинки под определенным углом обзора. По этой причине было решено спроектировать и воплотить в жизнь свой собственный скрипт.

Ключевые слова: автоматизация, обработка изображений, скрипт

AUTOMATED PROCESSING OF PHOTOGRAPHS IN ADOBE PHOTOSHOP

Miniyarova L.V., Gabdrahmanov A.N.

Bashkir state pedagogical university n.a. M. Akmulla, Ufa, e-mail: ainur1@bk.ru

The result of the research work is a script which makes automated processing of an image. Script is written in Javascript. The development has been made in Adobe ExtendScript Toolkit CS5. 5 software. For processing of images is used Adobe Photoshop CC. The research dedicated to process a huge number of photos in a fraction of time. The script is developed by request of our friend who is a designer. He faces this task pretty frequently. After analyzing specialized literature and programming utilities which already exist, the client found out that there is no proper solution that could satisfy our needs. There are a lot of utilities for photo processing, setting frames, establishing color hue but none of these could align the image to size with certain parameters of distortion which creates an view-illusion in a special angle. That's why we decided to create and to implement this very script.

Keywords: automation, image processing, script

В современном мире совершается огромное количество фотографии, почти для каждой применяются какие-либо изменения. Следовательно, необходимо программное обеспечение, которое позволяет без лишних усилий преобразовывать фотографию. Но как же быть, если нужно обработать несколько фотографий с одинаковым набором редактирования, потратив минимум времени? Перспективу для решения данной проблемы открывает автоматизация процесса с помощью скриптов, удобный функционал для пользователей графических редакторов. Это, заметно, сократит время необходимую на обработки схожих фотографии, поможет решать проблемы, которые уже попадались ранее в процессе. Соответственно, целью нашей работы является – разработка утилиты для программы Adobe Photoshop позволяющую преобразовать одновременно несколько фотографий, с помощью скриптов.

Скрипты – высокоуровневый язык сценариев (англ. script) – кратких описаний действий, выполняемых системой. Разница между программами и сценариями довольно размыта. Сценарий – это программа, име-

ющая дело с готовыми программными компонентами [2]. Скрипт занимает маленький объем памяти, с готовым скриптом может справиться даже неопытный пользователь.

По нижеперечисленному ряду причин, для реализации идеи автоматизированной обработки фотографий было взято программное обеспечение Adobe Photoshop:

1. Photoshop занимает лидирующее место среди графических редакторов.

2. Приоритет отдан растровой обработке фотографии, нежели векторным изображениям.

3. Совместим с языком JavaScript используемый для сценария.

JavaScript (аббревиатура JS) – язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования [1].

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее

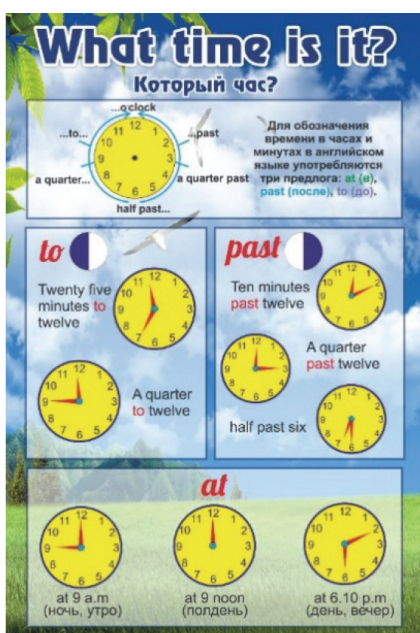
широкое применение находит как язык сценариев для придания интерактивности [3].

Определившись с языком и ПО, необходимо составить блок-схему процесса обработки изображения, включая туда: добавление изображения, использование всех необходимых инструментов и функции программы Photoshop, закончить блок схему сохранением и закрытием обработки. Работоспособность и полезность скрипта будет зависеть от валидности блок-схемы.

У нас есть исходные изображения (рис. 1). Их необходимо собрать воедино: общая картинка должна казаться, будто изображение «stend.jpg» висит как стенд на «fon.jpg».

Изображение «stend.jpg» было взято из просторов всемирной паутины, «figura.png» – векторная картинка сделанная в Adobe Photoshop, «fon.jpg» – фотография стены института.

Используя ранее составленную блок-схему пишем скрипт, сохраняем файлом формата «.Jsx». Чтобы воспроизвести сценарий, необходимо перетащить файл в рабочее поле программы Adobe Photoshop. Все записанные действия начнут происходить с очень быстрой скоростью, но не в фоновом режиме, можно видеть как происходят изменения в рабочем поле редактора. Скрипт откроет указанное изображение, наложит градиент и тени, добавит новые слои в определен-



а)

б)



в)

Рис. 1:

а – «stend.jpg»; б – «figura.png»; в – «fon.jpg»

ную зону рабочего пространства по прямоугольной системе координат, применит слой-заливку «Яркость/Контрастность» поверх всех остальных слоев. Следующими шагами сохранит всё обработанное в указанную папку в двух форматах «.Jpg» и «.PSD», а после закроет окно обработки. Всё происходит автоматический, пользователю остается лишь проверить готовое изображение. При необходимости пользователь может запустить файл «.PSD» и внести дополнительные корректировки, потому что скрипт все изменения создает в новых слоях, каждый слой независим от общей картины.

Пример кода для сохранения файлов (рис. 2).

Если вам приходится часто работать с графическими редакторами, необходимо поискать скрипты в просторах интернета, которые удовлетворяли вашим потребностям.

Запустим наш скрипт и посмотрим готовый результат (рис. 4). Теперь мы можем посмотреть, как выглядел бы стенд в данном коридоре. Скрипт выполнил все стадии обработки и сохранил изображение. Стоит отметить что у данного скрипта есть свои недостатки, например: не может определить в каком положении изображение `stand.jpg` горизонтальном или вертикальном, из-за этого возможен некорректный результат, но все шире-

```
var idsave = charIDToTypeID( "save" );
desc153.putObject( idAs, idPhththree, desc154 );
var idIn = charIDToTypeID( "In " );
desc153.putPath( idIn, new File( "C:\\kursov\\gor_ugol\\konez\\konez1.psd" ) );
desc157.putPath( idIn, new File( "C:\\kursov\\gor_ugol\\konez\\konez1.jpg" ) );
var idDocI = charIDToTypeID( "DocI" );
desc153.putEnumerated( idsaveStage, idsaveStageType, idsaveBegin );
desc157.putEnumerated( idsaveStage, idsaveStageType, idsaveBegin );
executeAction( idsave, desc153, DialogModes.NO );
```

Рис. 2

Пример кода для закрытия окна обработки (рис. 3).

ватости можно исправить углубившись в код скрипта.

```
// =====
var idCls = charIDToTypeID( "Cls " );
executeAction( idCls, undefined, DialogModes.NO );
```

Рис. 3



Рис. 4

Подводя итог, можно сказать, что использование скриптов в редактировании изображений ощутимо ускоряет процесс, за счет минимальной траты времени на обработку, так же сберегает энтузиазм и энергию пользователя для более

утонченных и уникальных задач. Стоит заметить, что возможности скриптов не ограничиваются лишь редакторами изображений.

Для более подробного изучения процесса предоставим несколько диаграмм.

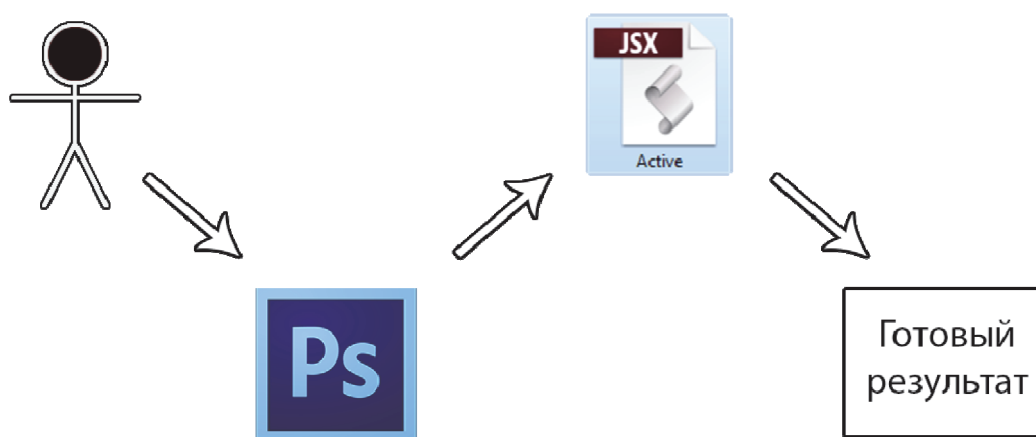


Рис. 5. Диаграмма работы пользователя

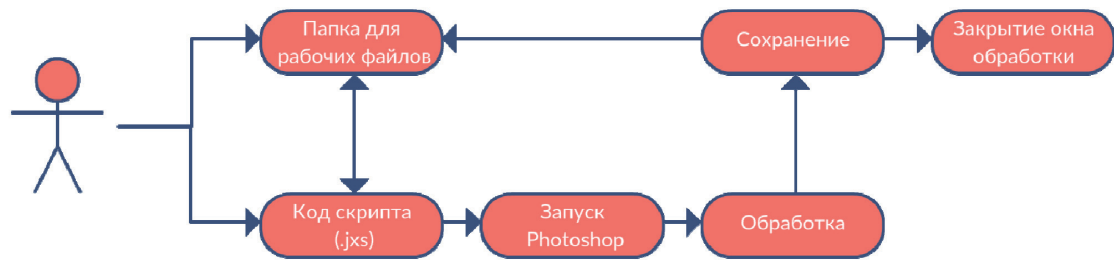


Рис. 6. Диаграмма прецедентов

Список литературы

1. Рейсиг Д. Инструменты отладки и тестирования // JavaScript 4. Профессиональные приёмы программирования = Pro JavaScript™ Techniques / Перевод Н. Вильчинский. – СПб.: Питер, 2008. – С. 76. – (Библиотека программиста). – 2500 экз. – ISBN 978-5-91180-904-1.
2. Сузи Р. Сценарные языки: Python // Мир ПК. – 2001. – №9.
3. Флэнаган Д. JavaScript. Карманный справочник. Сделайте веб-страницы интерактивными! / Пер. А.Г. Сысонок. – М.: Издательский дом «Вильямс», 2015. – С. 3200.