R. Ivanyuk, Master student
A. Morozov, PhD in Engr., Ass. Prof., research advisor
V. Shadura, Senior lecturer, language advisor
Zhytomyt State Technological University

3D GRAPHICS IN A BROWSER (THREE.JS)

When we have a look at the past it will be possible to see many plugins which run on “Flash”. Flash is the multimedia Adobe System platform used for web application design or multimedia presentations. But “Flash” has several disadvantages such as:

- the excessive CPU overloading caused by the inefficiency virtual machine;
- the lack error control, leading to frequent failures of both the applications, and in some cases the entire browser, and the ability to flash-application to disrupt the entire browser.

The evolution of technology and constant improvement of gadgets have caused the necessity to enlarge the spectrum of device opportunities for a modern person. r We want something new, something that will make our lives more comfortable, and enable us to follow all the events, to work efficiently, to communicate, to have fun. Many people satisfy these requests using the Internet. In this case we can see that mobile devices do not work in flash and it is a fact. So, the necessity to design the advanced mobile flash-players is becoming more and more urgent.

Flash is improving. Recently the comparison of Stage3D (a set of 3D API, which brought on 3D flash platform) with WebGL (Web-based Graphics Library - a software library for programming language JavaScript, which allows you to create JavaScript interactive 3D-graphics, that operates in a wide range of compatible web browsers it) using the library «three.js» has appeared. This comparison is unexpected because “Flash” has got the prior positions. So, having 30 times excessive number of objects Flash demonstrates a higher FPS (Frames per Second - the number of frames per second). But the conditions were far from being equal.

The advantage of WebGL is obvious. It is, first of all, 3D responsive design for mobile devices in some browsers, such as Android Firefox version 4. Today, and in general, we can confidently say that not only the HTML5 has become the standard, but the technologies such as WebGL, reached the sufficient maturity. But we can not help noticing the fact that most applications (games, applications) are developed in flash. So, it goes about the social media (for example: "VKontakte" - game "Vormyks." But it is not the only one example). As a result, cell phone games are designed separately and one has to download them separately. And it is not easy for the people who are away from their personal computer or game console. It is not only about the games. The internet is becoming faster and people are having more demands. Usually, people are not surprised with the shadow on text and shaded blocks of frames, but this technology is able to design video games, medical and scientific visualization and so on.

The advantages of WebGL application for more sophisticated users are:

- the cross-browser and cross-platform compatibility. Windows, MacOS, Linux matter nothing, as long as your browser supports WebGL.

- the use language JavaScript, which is quite common.

- the automatic memory management. In contrast to OpenGL in WebGL it is not necessary to perform the specific actions for the memory isolation and deleting .

- WebGL graphics possesses high performance that can be compared to the performance of native applications.

IT - specialists need new technologies and libraries which use the languages easy to use. That is why such libraries as js. and Babylon js use WebGL.

One of the most popular libraries is «three.js». Three.js - a small size cross -browser library JavaScript, which is used to create and display animated 3D computer graphics for web application development. Three.js scripts can be used in conjunction with elements of HTML5, CANVAS, SVG and WebGL.

A striking example of WebGL is a demo site «Assassin's Creed Pirates» and the site «Dino Hunt TV», and the libraries such as «Three.js» which are easy to create solutions based on WebGL.

WebGL mastering gives a lot of possibilities for IT-industry to create cross-browser and cross-platform masterpieces.

Preparation Phase

1. [Setup flattened list] For each scene graph object:
1. Init matrices.
2. Create geometry groups corresponding to face materials.
3. [Setup buffers] For each geometry group object:
1. Create WebGL buffers.
2. Create Non-WebGL buffers.
3. Add an item to flattened list.
2. Update buffers (Vertex buffer, Index buffer...).

Rendering Phase

1. Update matrices on all of the descendants of scene.
2. Set up camera matrices.
3. [Setup-loop] For each flattened list item:
1. Set up matrix.
2. Pick materials.
4. [Draw-loop] For each flattened list item:
1. Set up shader programs(and textures).
2. Prepare buffers.
3. Draw primitives