

**А. С. Чеботаревский, Е. А. Левчук**  
(ГГУ им. Ф. Скорины, Гомель)

### **ОТОБРАЖЕНИЕ МЕДИАДАНЫХ НА МОБИЛЬНЫХ УСТРОЙСТВАХ С ПОМОЩЬЮ БИБЛИОТЕКИ D3.JS**

При разработке мобильного приложения с помощью технологии cordova возник ряд проблем при отображении медиаданных. Решение должно быть написано на JavaScript, но при этом показывать максимальную скорость рендера на мобильных устройствах от компании Apple (в частности на планшетах iPad). С другой стороны, оно должно просто встраиваться в существующую архитектуру приложения, которое написано с помощью фреймворка AngularJS.

На сегодняшний день существует множество решений для графического представления медиаданных в виде графиков на мобильных устройствах, которые не уступают нативным решениям, например: d3.js, google charts, chartjs и так далее. Наиболее полным и популярным решением на данный момент является opensource библиотека d3.js, которая предоставляет не только высокую скорость отрисовки и широкие возможности по настройке вида отображения полученных медиаданных, но и удобный синтаксис для генерации изображений, графиков, таблиц.

Основные способы вывода изображения в d3 – это вывод с помощью html5 canvas и с помощью генерации svg изображений. Несмотря на то, что отображение графиков на canvas происходит быстрее и тратит меньше памяти, чем при выборе svg в качестве итогового формата вывода, для генерации графиков на мобильных устройствах следует выбрать именно генерацию svg изображений, так как вследствие большого количества мобильных браузеров и неполной поддержки canvas на мобильных устройствах, возможны проблемы с отображением медиаданных.

Большим плюсом при работе с d3, является философия данной библиотеки, которая привела к максимальному упрощению настройки отображения медиаданных, что привело к реализации fluence interface подхода. Суть которого заключается в использовании так называемых pipe для написания конфигураций.

Благодаря подобному подходу решается не только проблема с уменьшением сложности написания конфигураций отображения данных, но и так же основная проблема языка JavaScript, а именно его асинхронность.