# Титульный лист

# Задание (ТЗ)

# Реферат

# Содержание

# Обозначения и сокращения

# Введение

Еще недавно самым популярным средством создания интерактивных графических веб-приложений, насыщенных анимацией, трехмерной графикой и т.д., была технология Flash от фирмы Adobe[10], что требовало покупки среды разработки Flash Professional и изучения разработчиком языка программирования Acton Script, а так же требовало от пользователя установки в браузер средства для воспроизведения(Flash Player) Flash-роликов. Долгое время Flash был по сути был единственной технологией для создания игр, мультфильмов, интерактивны веб-приложений. Для всех основных платформ существовал и поддерживался Flash-player, но с развитием открытой технологии HTML5, постепенно, Adobe отказалось от поддержки Flash-player для операционных систем семейства Linux[11], а затем и от поддержки для мобильных платформ [12], так же ряд крупных фирм отказались от поддержки Flash:

* Apple в пользу HTML5 для iPhone, iPod touch и iPad;
* Microsoft в пользу Silverlight для Windows Phone 7 и Windows 8;
* Oracle в пользу JavaFX.

С появлением поддержки нового, предоставляющего схожие с Flash графические возможности, но еще не готового стандарта HTML(HTML5), ведущими производителями браузерных ядер (WebKit (Chrome), Gecko(FireFox), Presto(Opera)), перед разработчиком интерактивных веб-приложений встает необходимость использования графических возможностей HTML5.

В данной работе решается проблема сложности разработки графических веб-приложений с помощью 2d-context HTML5.

# HTML5

HTML5 (*англ. HyperText Markup Language, version 5*) – это язык разметки гипертекстовых документов являющийся одной из самой популярной технологией в интернете. Это пятая версия HTML-стандарта пришедшая на смену HTML4. Стандарт HTML5 находится в стадии тестирования – W3C объявил о планах, согласно которым окончательная версия стандарта HTML5 будет утверждена лишь к 2014 году[6].

HTML5 по сравнению с HTLM4 вводит новые интерфейсы:

* Медиа теги video[2] и audio[3], позволяющие воспроизводить видео и музыку на странице;
* Интерфейс позволяющий создавать веб-приложения, для работы, которых интернет соединение необходимо только на этапе загрузки[4];
* Drag&Drop интерфейс;
* Интерфейс 2d-context[5] для создания растровой графики;
* Интерфейс webgl-context[6] для создания трехмерной графики;
* Интерфейс к истории посещения[7];
* Интерфейс к хранилищу данных типа ключ–значение на стороне клиента[8].

Создание графических приложений в HTML возможно с помощью следующих технологий:

1. 2d-context – контекст тега canvas;
2. Webgl-context – контекст тега canvas;
3. SVG – язык описания векторных изображений;
4. Манипуляции DOM-моделью HTML документа.

# SVG, 2d-context, WebGL-context, DOM – как возможности HTML5 для создания графических приложений\игр

Выбрав HTML5 для создания графического приложения, разработчику доступны:

1. 2d-context – Это контекст тега <canvas> предоставляющего интерфейс для создания двумерной растровой графики. 2d-context позволяет манипулировать изображением на плоской двумерной системе координат с центром в левом верхнем углу экрана. Интерфейс представлен набором методов и свойств, определяющих графические примитивы их внешний вид и аффинные преобразования над экранной плоскостью.

Данный контекст можно считать простой альтернативой технологии Flash и вероятно будет подходить для создания простых игр, анимированных интерфейсов и т.д.

1. Webgl-context – Это контекст тега <canvas> предоставляющего интерфейс для создания трехмерной графики. Интерфейс является производным от OpenGL ® ES 2.0 и имеет схожие возможности, включая работу с вершинными и пиксельными шейдерами. Webgl-context позволяет задействовать вычислительные мощности видеокарты, что может подвергать пользователя риску, через открытие доступа к привилегированному режиму видеокарты и оборудования[13];
2. SVG – язык описания векторных изображений;

/\* недописанное \*/

1. Манипуляции DOM-моделью HTML документа.

/\* недописанное \*/

# Описание 2d-context

# Сложности и проблемы разработки при помощи 2d-context

# Варианты решения проблем и сложностей разработки при помощи 2d-context

# Требования к программному каркасу

# Обзор существующих решений

# Архитектура каркаса

# Проектирование архитектуры

# Парадигма программирования

# UML – диаграммы

# Используемые шаблоны проектирования

# Описание алгоритмов предлагаемого решения

# Средства разработки

# Интегрированная среда разработки

# Браузер

# Дебагер

# Профайлер

# Система контроля версий

# Принятые стандарты кодирования

# Ход работы

# Выбор сторонних вспомогательных каркасов\библиотек

# Интеграция классов каркаса

# Результат

# Список использованных источников

# Приложения

# ТЗ

# Руководство программиста (ГОСТ 19.504)