1. Добрый день уважаемая комиссия и все, присутствующие сегодня здесь.
2. Цель моей работы – разработка и реализация программного каркаса для создания спрайтовой анимации на HTML5. Данный каркас призван упростить процесс создания графических веб-приложений при использовании технологии 2d-context HTML5.

Для достижения данной цели было необходимо решить следующие задачи.

1. На сегодняшний день создание программного обеспечения в виде веб-приложения позволяет создавать приложения с функциональностью сравнимой с традиционными приложениями, а по таким параметрам как сохранность личных данных и системные требования даже выгодно превосходить традиционные приложения. Так же из-за того, что пользователь каждый раз загружает приложение с сервера – у всех пользователей почти одновременно одна версия приложения
2. **Что же такое 2d-context HTML5?**

2d-context является веб-технологией позволяющей создавать интерактивные графические веб-приложения рамках HTML5.

С программной точки зрения 2d-context является контекстом тега <canvas> и представляет интерфейс для создания двумерной растровой графики.

**Рассмотрим спецификацию на данную технологию:**

В случае с 2d-context, спецификация в первую очередь определяет программный интерфейс.

Можно видеть, что это самые основные методы для работы с двумерной графикой.

**Определенные особенности использования интерфейса и некоторые трудности**, связанные с технологией приводят к актуальности моей дипломной работы.

Я имею ввиду такие трудности как:

1. Нет возможности работать с частями изображения как с самостоятельными объектами
2. Для создания анимации требуются перерисовка экрана. Анимация это последовательная смена кадров, но так как 2d-context направлен на создание статичного изображения, он не имеет соответствующих методов
3. Описание несложных сцен требуют достаточно большое количество кода. Это вызвано тем фактом, что изображение создается с помощью минимального набора графических примитивов.
4. Неготовый стандарт. Стандарт в стадии тестирования – Консорциум всемирной паутины (World Wide Web Consortium) объявил о планах, согласно которым окончательная версия стандарта HTML5 будет утверждена лишь к 2014 году
5. Отсутствие визуальных сред, вроде Flash Professional CS6. Проблема в отсутствии возможности не программировать интерфейс графического приложения, а описывать в некотором визуальном редакторе
6. Слабое развитие специализированных каркасов, вызванное, скорее всего, незавершенностью стандарта.

**Требования к каркасу**

В ходе анализа: интерфейса технологии, существующих решений и процесса создания графических приложений с использованием 2d-context, были выявлены следующие требования. Вы можете видеть их на данном слайде.

**Объектное представление примитивов**.

В графическом приложении, интерактивность достигается за счет взаимодействия пользователя с отдельными элементами на экране представленными набором графических примитивов. То есть для изменения внешнего вида элемента необходимо изменять свойства каждого примитива входящего в состав элемента;

**Call-back функции для событий объектов**

В случае представления некоторой сущности на экране в виде некоторого объекта, нужно понимать, что объект может иметь не только свойства и методы, но и должен иметь возможность реагировать на некоторые события, например: реагировать на щелчок мыши;

**Загрузка мультимедиа-зависимостей объектов**

Такие примитивы как изображение или видео перед выводом на экран требуют загрузки данных. Примитивы должны иметь возможность самостоятельно загрузить необходимые данные и в соответствующем обработчике среагировать на некоторые события связанные с загрузкой;

**Приоритеты очереди прорисовки**

Объекты могут пересекаться на экране и определенные объекты должны находится выше или ниже других по этому нужен способ указывать приоритет объекта на прорисовку;

**Графический цикл**

Если есть потребность в изменяющемся динамичном изображении программист должен сам думать о очистке экрана организации смены кадров и т.д., то есть есть потребность в графическом цикле.

**Слои**

Под слоями подразумевается набор тегов <canvas>, расположенных друг над другом. Одни тег – один слой. Визуальные объекты распределяются по слоям, что позволяет разделить объекты по некоторой логике и оптимизировать время прорисовки кадра;

**Модульность**

Необходимо учитывать, что размер каркаса влияет на общее время загрузки и старта приложения. В разных приложениях могут быть задействованы далеко не все возможности, то есть необходимо реализовать модульности каркаса.

**Работа с мышью и клавиатурой**

Необходима для взаимодействия с пользователем;

**Вызовы методов каркаса цепочкой**

Данная функциональность используется и в крупных библиотеках и каркасах не связанных с графикой, например в jQuery и позволяет сделать код более читаемым;

**Спроектировать и реализовать программный каркас**

Структуру разработанного каркаса можно представить следующей диаграммой:

Классом высшего уровня является класс ArmLib, данный класс содержит основные методы управления каркасом. Класс ArmLib содержит в себе объекты класса Layer.

Класс Layer описывает слой и содержит в себе объекты классов наследующих от класса VisualObj. VisualObj является общим описанием визуального объекта. Класс Primitive является общим описанием графических примитивов.

Класс EventQueue данный класс описывает очеред сообщений объекта класса ArmLib объектам класса Layer. Примером сообщения может служить событие щелчка мыши или нажатие кнопок клавиатуры.

Каждый графический примитив представлен классом наследником класса Primitive. Класс Skeleton описывает полигон точек описывающих очертания определенного визуального объекта.

Класс Object описывает объект-контейнер содержащий в себе объекты классов производных от класса VisualObj. Данный класс позволяет ассоциировать набор визуальных объектов с некторым одним объектом.

**На следующей диаграмме приведен процесс работы главного цикла каркаса:**

**Для разработки каркаса были использованы следующие инструменты:**

* Netbeans в качестве IDE;
* Браузеры Chrome, Firefox, Opera для тестирования каркаса;
* Статический анализатор кода JSLint;
* Библиотека Gizmo
* Отладчик и профайлер браузера Chrome;
* Система контроля версий Git;
* Язык программирования JavaScript.

**В ходе работы были достигнуты следующие результаты:**

* Разработан и реализован программный каркас для упрощения разработки графического веб-приложения с помощью технологии 2d-context HTML5.
* Написана статья на тему: «Программный каркас для создания спрайтовой анимации на HTML5». Данная статья получила третье место в секции «Технологии разработки и проектирования информационных систем» на конференции «Технологии Microsoft в теории и практике программирования».
* Получен реальный опыт использования системы контроля версий
* Улучшено знание языка JavaScript

В дальнейшем работа по данной теме будет продолжена. В частности планируется существенно пересмотреть архитектуру каркаса с целью повышения гибкости разработки с использованием каркаса.

В завершении доклада я хочу поблагодарить свою кафедру и своего научного руководителя за помощь в написании диплома.

Студент Лизин, доклад окончил.