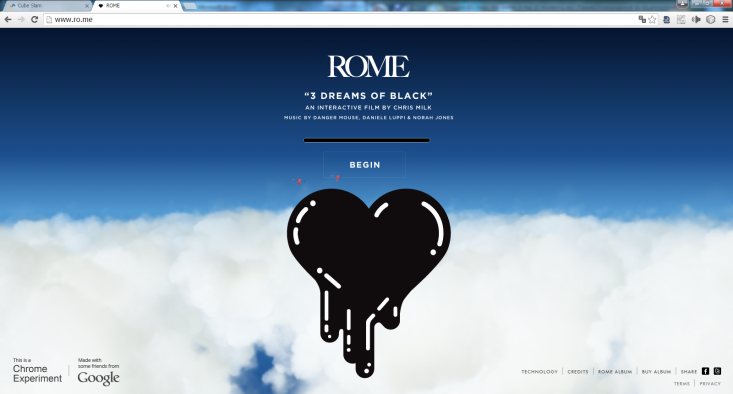
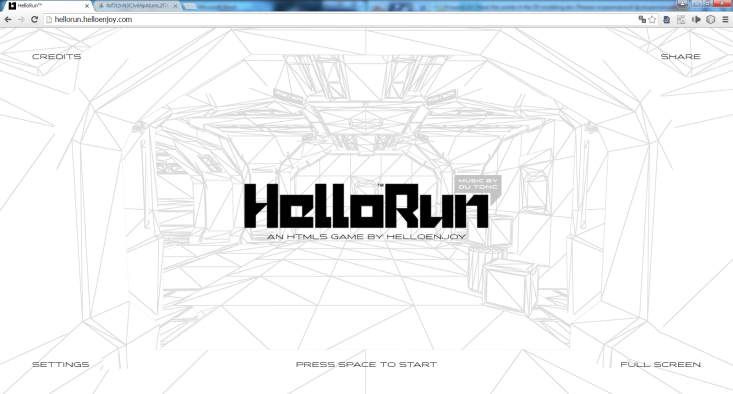
Днепропетровский национальный университет имени Олеся Гончара  
**Основные технологии визуализации сцен в веб-приложениях, достоинства и недостатки**

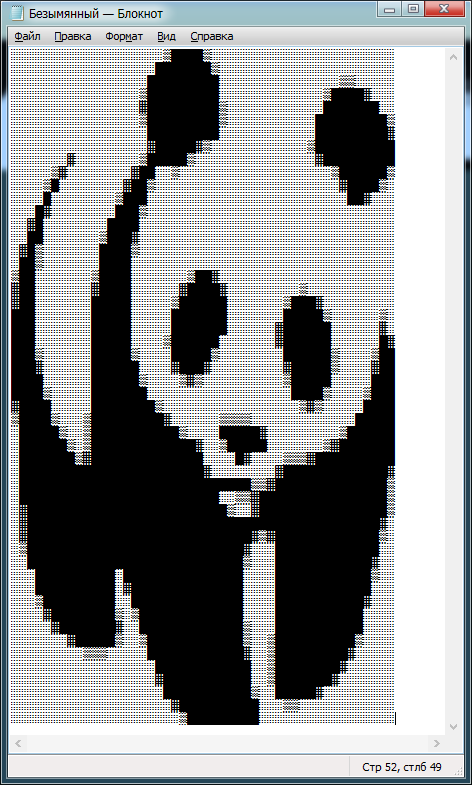
Современный мир веб-технологий развивается стремительными темпами, если раньше веб-страницы содержали в себе, лишь текстовую информацию, то сейчас, сайт может играть роль текстового редактора, базы знаний и набор вычислительных алгоритмов[1], магазина, проигрывателя видео или музыки, 2D и 3D интерактивного приложения (HelloRun[2], CUBE SLAM[3], 3 DREAMS OF BLACK[4]) .



****

**Что же такое сцена?** В нашем случае мы будем говорить об 2D и 3D сценах, фактически сцена – это часть двухмерного или трехмерного мира, подлежащая расчёту и выводу на экран в соответствии с текущей точкой наблюдения. Здесь можно заметить следующее, что фактически экран, сам по себе плоский. То есть, это некоторая поверхность на которую все объекты сцены проецируются при помощи определенной матрицы проекции. Так как, веб-приложения не имеют полного доступа ко всем API операционной системы, вся визуализация так же должна производится на некоторую поверхность, именно веб-приложения. Этой поверхностью может быть: текстовое поле – далее будет показано, как текстовое поле может быть использовано для вывода псевдографики, элемент Image, SVG, Canvas и другие.

**Основные технологии визуализации изображений в веб-приложениях.** Каждая из этих технологий имеет свои достоинства и недостатки, каждая из них является достаточно хорошей для своего времени и целей.

**ASCII.** В начале 90-х годов, когда веб-технологии только начинали развиваться, а скорость интернета не была столь высока, на веб-страницах, в основном размещали, только текстовую информацию. Для отрисовки изображений использовалась технология, подобная ANSI-графике [5], которая стала популярной в 1986 году [6] - это технология создания изображений при помощи символов ASCII [7], так как поддержка ANSI – графики не была предусмотрена. Первоначально эти изображения создавались в ручную, но со временем появлялись программы позволяющие конвертировать любые изображения в ASCII текст.

Достоинства: высокая скорость загрузки; отображение и поддержка во всех браузерах.

Недостатки: малое расширение изображений; не реалистичное отображение изображений; достаточно сложная реализация анимации.

**Image.** С развитием стандарта [HTML](https://en.wikipedia.org/wiki/HTML_element) элементов, появился тег <img />, который позволял внедрять изображения на веб-страницу. Это позволило весьма просто добавлять изображения, но для создания не линейной анимации необходимо постоянно изменять атрибут src, через JavaScript. Для цикличных(линейных) анимации применяется формат GIF.

Достоинства: простота добавления изображений на страницу.

Недостатки: создание анимации приводит к значительной нагрузке CPU;

**CSS.** С появлением CSS в 1996 году, стало возможным задавать определенные стили для отображения тех или иных объектов на веб-страницах. Это позволило, задавать изображения внутри элементов DOM [8], при помощи свойства background. Таким образом, манипулируя элементами DOM при помощи JavaScript, создаётся анимация [10].

Достоинства: высокая скорость загрузки; отображение и поддержка во всех браузерах.

Недостатки: малое расширение изображений; не реалистичное отображение изображений; достаточно сложная реализация анимации.

**SVG.** С 1999 года появляется технология масштабируемой векторной графики SVG. Анимация реализована в SVG с помощью языка SMIL (Synchronized Multimedia Integration Language). SVG-элементами можно управлять с помощью JavaScript. Применение скриптов и анимации в SVG позволяет создавать динамичную и интерактивную графику [9].

Достоинства: Масштабируемость — SVG является векторным форматом, по этому при увеличении изображения, оно не теряет качества; SVG — открытый стандарт;

Недостатки: Чем больше в изображении мелких деталей, тем быстрее растёт размер SVG-данных; Не поддерживает описание трёхмерных объектов (не путать с имитацией трёхмерности путём светотени)

**Canvas.** Элемент Canvas был разработан для отображения растрового изображения. Данный элемент может создаваться и меняться динамически при помощи языка JavaScript.

Достоинства: В отличие от SVG гораздо удобнее иметь дело с большим числом элементов; Имеет аппаратное ускорение; Есть возможность перерисовки определенного пикселе или области.

Недостатки: Плохая производительность при большом разрешении.

**Визуализации 3D сцен.** Из всех методов рассмотренных ранее для визуализации трехмерных сцен с настоящими трёхмерными объектами, высоким качеством и разрешением изображения, подходит элемент HTML Canvas. Именно по этому, современные фреймворки базирующиеся на библиотеке WebGL используют элемент Canvas, как поверхность для визуализации результатов. Сама библиотека WebGL ускоряет процесс визуализации результатов тем, что часть расчетов переноситься на графический процессор GPU, а не на центральный CPU. После чего, уже рассчитанное изображение выводиться на элемент Canvas [11]. Фреймворки для создания 3D сцен: Blend4Web, Three.js [11], Unreal Engine 4.7

Литература:

1. <https://www.wolframalpha.com/>
2. <http://hellorun.helloenjoy.com/>
3. <https://www.chromeexperiments.com/experiment/cube-slam>
4. <http://www.ro.me/>
5. <https://ru.wikipedia.org/wiki/ANSI-графика>
6. <https://en.wikipedia.org/wiki/TheDraw>
7. <https://ru.wikipedia.org/wiki/ASCII-графика>
8. Бахирев А. Сюрреализм на JavaScript. - Санкт-Петербург СИНЭЛ, 2014. - 116 c.
9. <http://buildnewgames.com/dom-sprites/>
10. Кит Джереми HTML5 для веб-дизайнеров. - A Book Apart, 2012. -80 c.
11. Isaac Sukin Game Development with Three.js. - Packt Publishing, 2013. -118 с.