

Лабораторная работа №1. Знакомство с Google SketchUp

1. Изучить возможности программы Google SketchUp
2. Создать 3D-модель дома с несколькими комнатами и двумя этажами
3. Заполнить комнаты моделями интерьера, мебели и т.д.
4. Оформить отчёт со скриншотами и кратким описанием используемых функций.

Лабораторная работа №2. Создание 3D-модели автомобиля.

5. Создать трёхмерную модель автомобиля с использованием примитивов (линии, полигоны, окружности и т.д.)
6. Задать интерьер (сидения, руль, приборная панель, педали и т.д.)
7. Задать экстерьера автомобиля (наложить текстуры или сплошной цвет, добавить фары, двери, окна, зеркала, ручки и т.д.)

Лабораторная работа 3. Создание стрелочных часов с использованием HTML5 Canvas.

- 1) Создайте HTML-страницу с элементом Canvas.
- 2) Задайте размеры рабочей области.
- 3) Укажите фоновое изображение для часов.
- 4) Создайте циферблат аналоговых часов с помощью элементов line, text.
- 5) Нарисуйте секундную, минутную и часовую стрелки с обновлением их позиции с учетом текущего времени.
- 6) Добавьте отображение текущего числа и названия месяца.

Инструкция:

1. Создайте папку своего проекта.
2. В этой папке создайте файлы:
 - index.html
 - main.css
 - main.js
3. В файле index.html добавьте элемент CANVAS и с помощью CSS-стилей задайте его оформление. Чтобы создать Canvas-контекст, достаточно просто добавить элемент `<canvas>` в HTML-документ:

```
<canvas id="myCanvas" width="300" height="150">
```

Альтернативное содержимое, которое будет показано, если браузер не поддерживает Canvas.

```
</canvas>
```

4. Чтобы нарисовать окружность, нужно выполнить такой код:

```
context.beginPath();
```

```
context.arc(75, 75, 10, 0, Math.PI*2, true);
```

```
context.closePath();
```

```
context.fill(); // Если нужен круг, можно залить окружность
```

5. Контур Canvas позволяют рисовать фигуры любой формы. Сначала нужно нарисовать "каркас", а потом можно использовать стили линий или заливки, если это необходимо. Чтобы начать рисование контура, используется метод `beginPath()`, потом рисуется контур, который можно составить из линий, кривых и других примитивов. Как только рисование фигуры окончено, можно вызвать методы назначения стиля линий и заливки, и только потом вызвать функцию `closePath()` для завершения рисования фигуры.
6. Метод `drawImage` позволяет вставлять другие изображения (`img` и `canvas`) на канву. В браузере Opera также существует возможность рисования SVG-изображений внутри элемента `canvas`. `drawImage` довольно сложный метод, который может принимать три, пять или девять аргументов:
 - Три аргумента: Базовое использование метода `drawImage` включает один аргумент для указания изображения, которое необходимо вывести на канву, и два аргумента для задания координат.
 - Пять аргументов: Используются предыдущие три аргумента и еще два, задающие ширину и высоту вставляемого изображения (в случае если вы хотите изменить размеры изображения при вставке).
 - Девять аргументов: Используются предыдущие пять аргументов и еще четыре: два для координат области внутри исходного изображения и два для ширины и высоты области внутри исходного изображения для обрезки изображения перед вставкой в `Canvas`.
7. Используя метод `fillRect`, вы можете нарисовать прямоугольник с заливкой. С помощью метода `strokeRect` вы можете нарисовать прямоугольник только с границами, без заливки. Если нужно очистить некоторую часть канвы, вы можете использовать метод `clearRect`. Три этих метода используют одинаковый набор аргументов: `x`, `y`, `width`, `height`. Первые два аргумента задают координаты (`x,y`), а следующие два — ширину и высоту прямоугольника.
8. Контур Canvas позволяют рисовать фигуры любой формы. Сначала нужно нарисовать "каркас", а потом можно использовать стили линий или заливки, если это необходимо. Чтобы начать рисование контура, используется метод `beginPath()`, потом рисуется контур, который можно составить из линий, кривых и других примитивов. Как только рисование фигуры окончено, можно вызвать методы назначения стиля линий и заливки, и только потом вызвать функцию `closePath()` для завершения рисования фигуры.

Лабораторная работа 4. Создание изображения с помощью SVG-графики.

Цель работы: нарисовать дом, солнце, деревья и железную дорогу.

Научиться рисовать основные фигуры: прямоугольник, окружность, радугу, солнце.

Руководство:

Scalable Vector Graphics (масштабируемая векторная графика) - формат изображений на основе текста. Каждое SVG-изображение определено с использованием разметки кода, похожей на HTML. SVG-код может быть включен напрямую в HTML-документ. Каждый веб-браузер поддерживает SVG, исключением является только Internet Explorer версии 8 и старше. SVG основан на XML, поэтому вы можете заметить, что элементы, не имеющие закрывающего тега, должны быть самозакрывающимися. Например:

```
1. <element></element>    <!-- Uses closing tag -->
2. <element/>             <!-- Self-closing tag -->
```

Перед тем как вы сможете что-нибудь рисовать, вам надо создать SVG-элемент. Думайте об SVG-элементе, как о холсте, на котором отрисовываются все ваши визуальные образы (В такой трактовке, SVG концептуально схож с элементом HTML - canvas). Как минимум, хорошо задать ширину и высоту с помощью атрибутов width и height, соответственно. Если вы их не зададите, SVG растянется на всю площадь блока.

```
1. <svg width="500" height="50">
2. </svg>
```

Простые фигуры

Существует некоторый набор фигур, которые вы можете поместить внутрь элемента SVG. Этот набор включает: rect, circle, ellipse, line, text и path.

Если вы знакомы с программированием компьютерной графики, вы вспомните, что обычно координатная система на основе пикселей начинается с левого верхнего угла и имеет координаты точки (0,0). Увеличение x происходит слева направо, увеличение y - сверху вниз.

0,0100,20200,40

rect рисует квадрат. Квадрат задается четырьмя значениями: x, y - указывают точку верхнего левого угла; width, height - указывают ширину и высоту квадрата. Этот квадрат занимает все пространство нашего SVG:

```
1. <rect x="0" y="0" width="500" height="50"/>
```

circle рисует круг. Круг задается тремя величинами: cx, cy указывают точку, расположенную в центре описываемой окружности, r задает радиус круга. Этот круг расположен в центре нашего SVG, потому что атрибут cx("center-x") равен 250. Пример:

```
1. <circle cx="250" cy="25" r="25"/>
```

ellipse задается схоже с circle, но предполагается, что радиус задается по двум осям: по x и по y. Вместо x используйте rx, вместо y - ry.

```
1. <ellipse cx="250" cy="25" rx="100" ry="25"/>
```

line рисует линию. Используйте x1 и y1 для задания координат начала линии, и x2 и y2 для задания координат конца. Атрибут stroke должен задавать цвет линии, иначе она будет невидимая.

```
1. <line x1="0" y1="0" x2="500" y2="50" stroke="black"/>
```

Стилизация SVG-элементов

По умолчанию любой элемент SVG имеет черную заливку и не имеет рамку. Если вы хотите что-то сверх этого, вам придется применить стили к вашему элементу. Общие SVG свойства перечислены ниже:

- fill - заливка. Цветовое значение. Также как и в CSS цвет может быть указан несколькими способами:
 - по имени: orange;
 - значение в шестнадцатиричной системе счисления: #3388aa, #38a;
 - значение в формате RGB: rgb(10, 150, 20);
 - значение в формате RGBA: rgba(10, 150, 20, 0.5).
- stroke - рамка. Цветовое значение.
- stroke-width - ширина рамки(обычно в пикселях).
- opacity - прозрачность. Числовое значение в промежутке от 0.0(полностью прозрачно) до 1.0(полностью видимо).

С text также можно использовать следующие свойства:

- font-family
- font-size