3. Векторная графика

Формат SVG - Scalable Vector Graphics с английского - «пропорциональная векторная графика», является языком для описания 2D-изображений используя язык XML. Это стандарт организации W3C, проектирование которого началось в 1999 году. Позволяет определять изображения 3 методами: текст, векторная графика и bitmap (файлы в формате BMP).

Хотя существуют специализированные приложения для создания и редактирования SVG, для этого можно использовать любой текстовый редактор. Просмотр изображения SVG можно сделать с помощью любого современного браузера.

Технические характеристики

На данный момент SVG имеет несколько профилей для лучшей адаптации к различным ограничениям. Так, профили «SVG Tiny» и «SVG Basic» были созданы специально для мобильных устройств с ограниченными ресурсами. В то же время профиль «SVG Print» предназначен для печати документов.

Для анимации изображения SVG организация W3C рекомендует стандарт «SMIL». Помимо официальной рекомендации, существуют и другие решения, такие как «ECMAScript».

Компоненты

Основными элементами в структуре SVG-файла являются:

- *Paths* «Контуры» можно использовать для описания контура фигуры. Контур может оставаться пустым или быть заполненным. Контуры могут быть использованы и как области обрезки;
- *Основные формы.* SVG фаилы предлагает возможность использования следующих основных фигур: прямоугольник, круг, эллипс, линия и многоугольник. Формы также могут быть построены с помощью контуров;
- *Текст.* Для написания текста, который появляется на изображении, необходимо использовать элементы типа текста;
- *Painting.* Относится к возможности заполнения форм, указанных в SVG. Для этого можно использовать только один цвет, один цвет с прозрачностью, градиент или модель;
- *Цвет.* Свойство color используется для спецификации цвета;
- *Градиент и узор.* Используется для окрашивания заданных фигур;
- *Резка, маскировка.* В SVG можно использовать области резки или маскировки;
- *Фильтры.* Фильтры описывают различные эффекты, применяемые к изображениям;
- *Интерактивность*. Изображение SVG имеет возможность взаимодействовать с пользователем. Таким образом, при нажатии кнопки или использовании мыши могут запускаться различные скрипты;
- *Ссылки.* Файл может содержать ссылки на другие страницы или элементы веб-сайта;
- *Сценарии.* В SVG могут быть определены скрипты с различными функциями;
- *Анимации.* Для SVG могут быть использованы различные типы анимации;
- *Шрифты.* Конечному пользователю не обязательно, чтобы у него были установлены различные наборы символов («шрифты»). Шрифты могут быть включены в изображение, и они могут быть редактированы в любой ситуации;
- *Метаданные.* Для лучшей интеграции семантическая сеть также предлагает возможность указания метаданных (это данные, которые описывают сами данные).

HTML-элемент <svg>

HTML-элемент <svg> является контейнером для векторной графики SVG.

Browser Support

В таблице 3.1 указаны версии браузеров, которые полностью поддерживают элемент <svg>.

Таблица 3.1. Версии браузера, поддерживающие элемент <svg>.

Element					
<svg>	4.0	9.0	3.0	3.2	10.1

Тег <svg> имеет несколько методов рисования линий (отрезков), контуров, прямоугольников, кругов, текста и графических изображений.

Фигуры SVG

Формат SVG имеет несколько элементов для описания предопределенных фигур, которые могут быть использованы разработчиком:

- Прямоугольник <rect>;
- Круг <circle>;
- Эллипс <ellipse>
- Линия <line>;
- Полилиния <polyline>;
- Полигон <polygon>;
- Траектория <path>;

Свойство контура

Формат <svg> предоставляет широкий спектр свойств для контура:

- stroke
- stroke-width
- stroke-linecap
- stroke-dasharray

Все свойства контура могут быть применены к любому типу линии, контуру текста или контуру рисунков.

Свойство **stroke** определяет цвет линии, контура текста или рисунков.

Пример

```
<svg height="80" width="300">
<g fill="none">
        <line stroke="red" x1="0" y1="10" x2="200" y2="10"/>
        <line stroke="black" x1="0" y1="20" x2="200" y2="20"/>
        <line stroke="blue" x1="0" y1="30" x2="200" y2="30"/>
</g>
</svg>
```

Свойство stroke-width

Свойство **stroke-width** определяет ширину линии, контура текста или обводки элемента.

Пример

```
<svg height="80" width="300">
 <g fill="none">
  <line stroke="red" stroke-width="1", x1="0" y1="10" x2="200" y2="10"/>
        <line stroke="black" stroke-width="2", x1="0" y1="20" x2="200" y2="20"/>
        <line stroke="blue" stroke-width="3", x1="0" y1="30" x2="200" y2="30"/>
 </g>
</svg>
```

Свойство stroke-linecap

Свойство **stroke-linecap** определяет различные типы окончания отрезков:

Пример

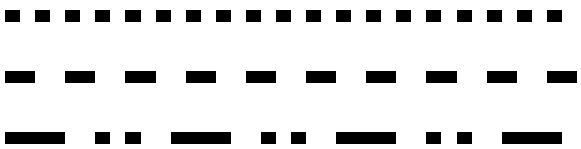
```
<svg height="80" width="300">
 <g fill="none" stroke="black" stroke-width="10">
  <line stroke-linecap="butt", x1="10" y1="20" x2="200" y2="20" />
        <line stroke-linecap="round", x1="10" y1="40" x2="200" y2="40" />
        <line stroke-linecap="square", x1="10" y1="60" x2="200" y2="60" />
</g>
</svg>
```

Свойство stroke-dasharray

Свойство **stroke-dasharray** используется для создания штрихпунктирных линий:

Пример

```
<svg height="80" width="300">
 <g fill="none" stroke="black" stroke-width="4">
  <line stroke-dasharray="5,5", x1="10" y1="20" x2="200" y2="20" />
        <line stroke-dasharray="10,10", x1="10" y1="40" x2="200" y2="40" />
        <line stroke-dasharray="20,10,5,5,5,10", x1="10" y1="60" x2="200" y2="60" />
 </g>
</svg>
```



Линия (отрезок) – <line>

Элемент <line> используется для рисования отрезка прямой:

<line> определяет отрезок;

x1 – абсцисса x начальной точки отрезка прямой;

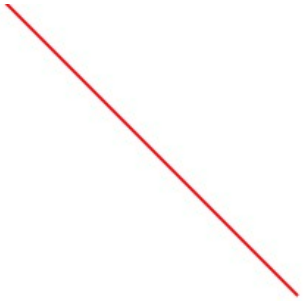
y1 – ординат y начальной точки отрезка прямой;

x2 – абсцисса x конечной точки отрезка прямой;

y2 – ордината y конечной точки отрезка прямой.

Пример

```
<svg height="210" width="500">
<line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0); stroke-width:2" />
</svg>
```



Полилиния – <polyline>

Элемент <**polyline**> используется для создания ломаных линий, построенных из прямых отрезков, соединенных несколькими точками.

Пример

```
<svg height="200" width="500">
<polyline points="20,20 40,25 60,40 80,120 120,140 200,180"
  style="fill:none;stroke:black;stroke-width:3" />
</svg>
```

Атрибут *points* содержит список точек (пар координат x и y), необходимых для рисования ломаной линии.

Круг

<circle> определяет круг
cx – координата по оси x центра окружности;
cy – координата по оси y центра окружности;
r – радиус окружности.

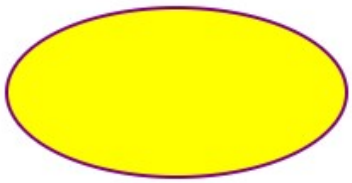
```
<svg width="100" height="100">
 <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
```

Эллипс – <ellipse>

Элемент <ellipse> используется для создания эллипса.
<ellipse> определяет эллипс
cx – координата по оси x центра эллипса;
cy – координата по оси y центра окружности;
rx – длина радиуса эллипса по оси x (горизонтальный радиус – ширина);
ry – длина радиуса эллипса по оси y (вертикальный радиус – высота).

Эллипс является более обобщенным случаем круга. Разница между этими цифрами заключается в том, что в эллипсе длины радиусов на осях x и y различаются, в то время как в случае окружности они равны.

```
<svg height="140" width="500">
 <ellipse cx="200" cy="80" rx="100" ry="50"
 style="fill:yellow;stroke:purple;stroke-width:2" />
</svg>
```

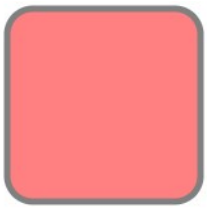


Прямоугольник - <rect>

Элемент <rect> используется для создания прямоугольников и его производных фигур:
<rect> определяет прямоугольник
x – координата по оси x левого угла верхней части прямоугольника;
y – координата по оси y левого угла верхней части прямоугольника;
rx – радиус по оси x (для округления вершин элемента);
ry – радиус по оси y (для округления вершин элемента);
width – ширина прямоугольника;
height – высота прямоугольника.

Пример

```
<svg width="400" height="180">
 <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
 style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>
```



Многоугольник – <polygon>

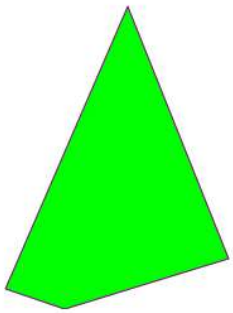
Элемент <polygon> используется для создания графики, которая должна содержать не менее 3 сторон. Многоугольники создаются из прямых линий, а рисунок представляет собой замкнутый контур.

<polygon> определяет многоугольник, содержащий по меньшей мере три стороны;

points – содержит координаты x и y каждой точки в множестве точек, составляющих вершины многоугольника. Общее количество точек должно быть четным.

fill-rule – атрибут представления, который определяет алгоритм, используемый для представления внутренней части фигуры (метод заполнения/раскраски) и может принимать *четные* (*evenodd*) или *ненулевые* (*nonzero*) значения.

```
<svg height="250" width="500">
<polygon points="220,10 300,210 170,250 123,234" style="fill:lime;stroke:purple;stroke-width:1"
/>
</svg>
```



Траектория – <path>

Элемент <path> используется для определения траектории.

<path> определяет траекторию

d – параметр данных path представляет собой набор команд, определяющих траекторию;

pathLength – при наличии контур будет масштабироваться таким образом, чтобы вычисляемая длина траектории, проходящего через точки, была равна этому значению;

transform – список преобразований.

Параметр данных path может иметь следующие значения:

M = moveto

L = lineto

H = horizontal lineto

V = vertical lineto

C = curveto

S = smooth curveto

Q = quadratic Bézier curve

T = smooth quadratic Bézier curveto

A = elliptical Arc

Z = closepath

Все вышеперечисленные команды также могут быть написаны строчными буквами. Использование заглавных букв означает абсолютное позиционированный, использование строчных букв означает относительное позиционирование.

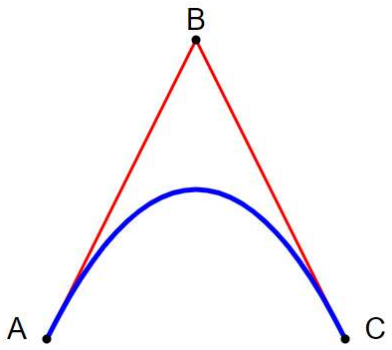
Кривые Безье

Кривые Безье используются для формирования плавных линий, которые можно масштабировать бесконечно без потери качества. Для представления кривой Безье пользователь устанавливает две точки, определяющие концы кривой и одну или две контрольные точки. Кривая Безье с контрольной точкой называется квадратичной кривой Безье, а кривая с двумя контрольными точками — кубической кривой Безье.

В приведенном ниже примере показано создание квадратичной кривой Безье, где точки A и C представляют концы, а точка B представляет контрольную точку:

Пример

```
<svg height="400" width="450">
 <path id="lineAB" d="M 100 350 l 150 -300" stroke="red"
 stroke-width="3" fill="none" />
 <path id="lineBC" d="M 250 50 l 150 300" stroke="red"
 stroke-width="3" fill="none" />
 <path d="M 100 350 q 150 -300 300 0" stroke="blue"
 stroke-width="5" fill="none" />
 <!-- Mark relevant points -->
 <g stroke="black" stroke-width="3" fill="black">
  <circle id="pointA" cx="100" cy="350" r="3" />
  <circle id="pointB" cx="250" cy="50" r="3" />
  <circle id="pointC" cx="400" cy="350" r="3" />
 </g>
 <!-- Label the points -->
 <g font-size="30" font-family="sans-serif" fill="black" stroke="none"
 text-anchor="middle">
  <text x="100" y="350" dx="-30">A</text>
  <text x="250" y="50" dy="-10">B</text>
  <text x="400" y="350" dx="30">C</text>
 </g>
</svg>
```



Текст– <text>

Элемент <text> используется для определения текста.

<text> определяет текст

x – список координат по оси x символов. Элемент *n* в списке координат по оси x соответствует символу с порядковым номером *n* в тексте. Если есть дополнительные символы, которые не соответствуют значениям координат оси x, они помещаются после последнего символа. Значение по умолчанию – 0.

y – список координат по оси Y символов.

dx – список значений по оси x, с которыми символы перемещаются относительно абсолютного положения последнего нарисованного символа;

dy – список значений по оси y, с которыми символы перемещаются относительно абсолютного положения последнего нарисованного символа;

rotate – список углов поворота. Значение *n* в списке поворота применяется к символу *n* в отображаемом тексте. Символы дополнительные не вращаются.

Пример

```
<svg height="200" width="200">
<text font-size="50" x="0" y="0" fill="red" transform="translate(50,50) rotate(40)"> SVG</text>
</svg>
```

Разница между SVG и Canvas

SVG — это язык, описывающий 2D-графику в XML.

Canvas (холст) позволяет динамически генерировать 2D-растровую графику с использованием языка JavaScript.

SVG основан на XML, который предполагает, что каждый элемент доступен с помощью методов SVG DOM. Это обеспечивает возможность присоединения событий JavaScript к каждому элементу.

В SVG каждый рисунок представляет объект, и, если атрибуты такого объекта SVG изменяются, автоматический браузер перерисовывает рисунок.

Холст рисуется пиксель за пикселем. На холсте, как только рисунок нарисован браузером, он теряет с ним связь. Если положение фигуры в сцене необходимо изменить, в этом случае необходимо перерисовать всю сцену.

Таблица 3.2. Сравнение Canvas и SVG

Canvas	SVG
– Зависим от разрешения – Не поддерживает события – Плохой рендеринг текста – Возможность экспорта графики в форматах .png или .jpg — Подходит для игр с продвинутой графикой	– Независим от разрешения – Поддержка событий — Подходит для приложений с большими игровыми зонами (Google Maps) — Медленное воспроизведение в сложных сценах (чрезмерное использование DOM распространяется на производительность) — Не подходит для игр.

Интеграция векторной графики в p5.js Web Editor

Для того чтобы интегрировать код *<svg>* в язык HTML с помощью веб-редактора p5.js необходимо авторизоваться с помощью аккаунта *Google* или *GitHub* на соответствующей платформе. Для просмотра составляющих файлов необходимо нажать кнопку со значком стрелки в левой части окна (рис. 3.1).

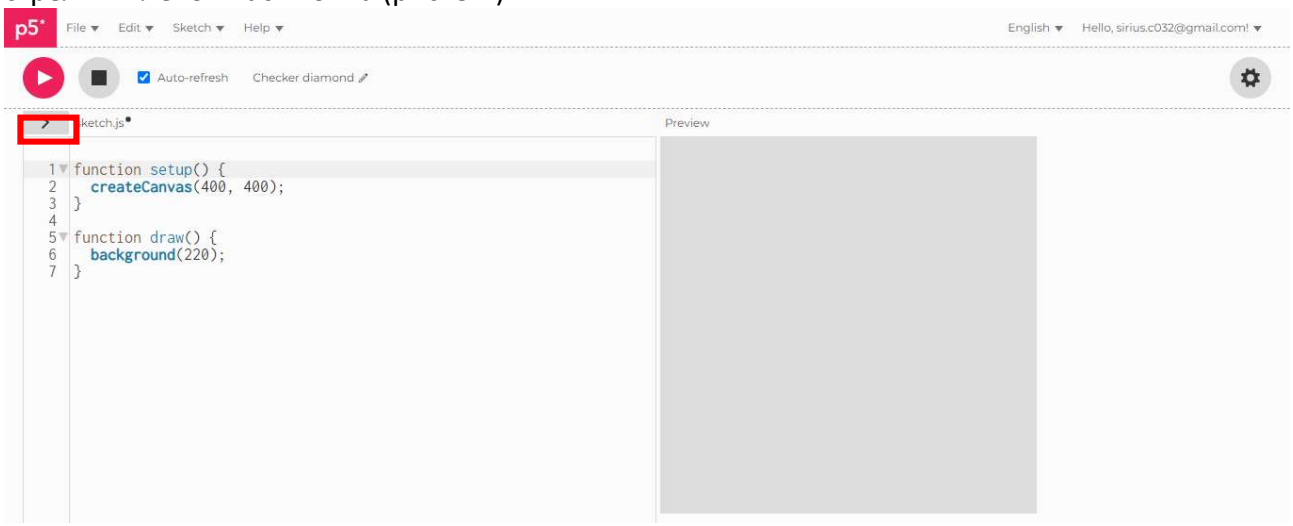


Рис. 3.1. Доступ к окну Scketch Files веб-редактора p5.js.

После заданного действия открывается окно Sketch Files, в котором можно увидеть файлы компонентов проекта: *index.html, sketch.js* и *style.css*. Файл *index.html*, содержит HTML-код для маркировки интерфейса компонентов веб-страницы проекта (рис. 3.2).

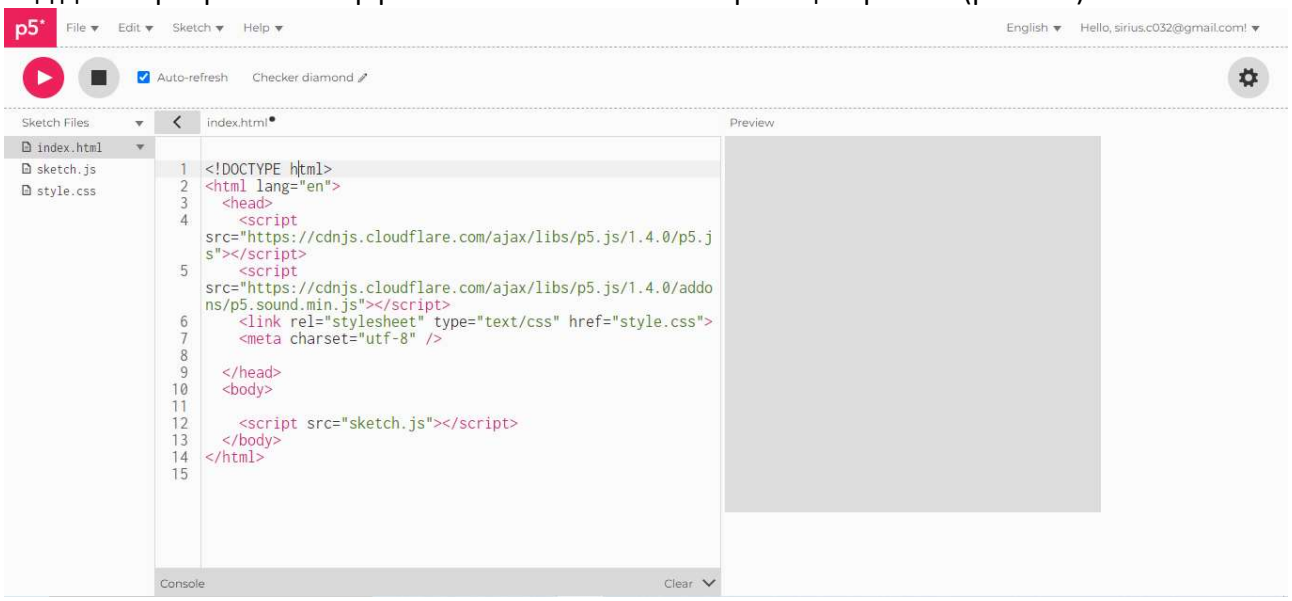


Рис. 3.2. Содержимое *index.html* файла проекта

Видно, что в *index.html* включена библиотека p5.js которая содержит все графические функции, облегчающие работу с 2D и 3D графикой в браузере. Файл *p5.sound.min.js* который представляет собой звуковую библиотеку, расширяющую основные функциональные возможности библиотеки p5 с функциональностью Web Audio, включая аудиовход, воспроизведение, анализ и синтез. Файл *style.css* в который могут быть добавлены стили для HTML-компонентов внутри проекта (рис. 3.3). Файл I, представляющий исходную программу сцены, отредактированную в библиотеке JavaScript, (рис. 3.1).

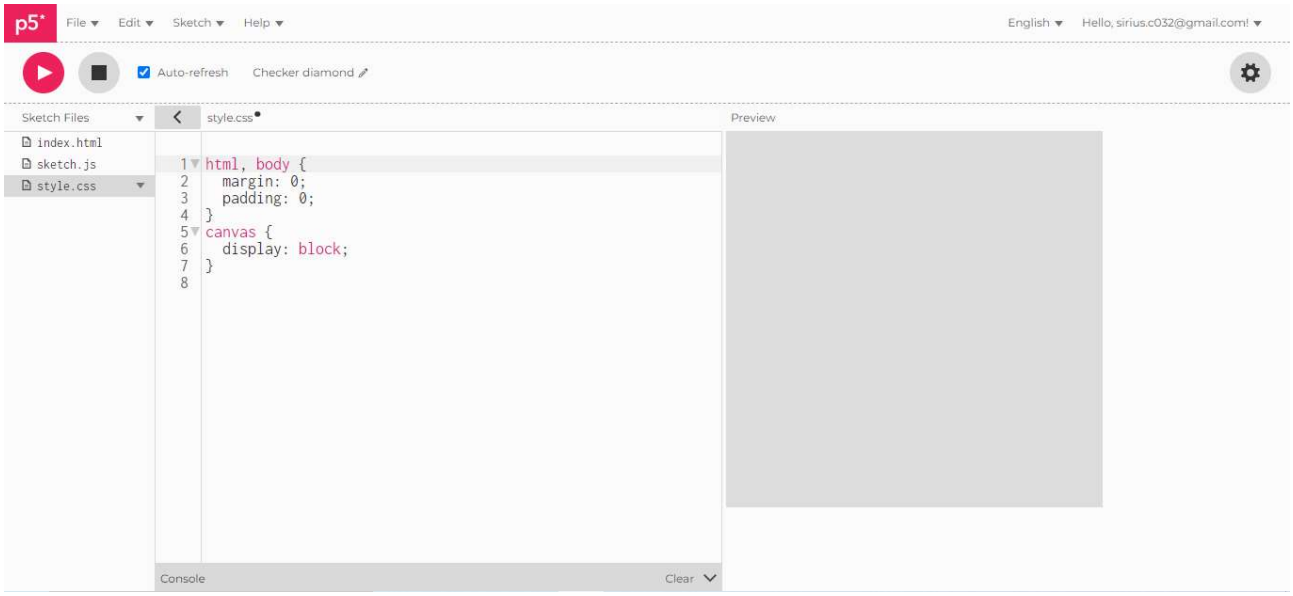


Рис. 3.3. Содержимое файла *style.css* проекта

Для того чтобы интегрировать код *<svg>* в проект, выполненный с помощью редактора p5.js Web Editor, необходимо отредактировать файл *index.html* а именно замену строки *<script src="sketch.js"></script>* внутри тега *<body>"></body>* требуемым svg кодом, (рис. 3.4).

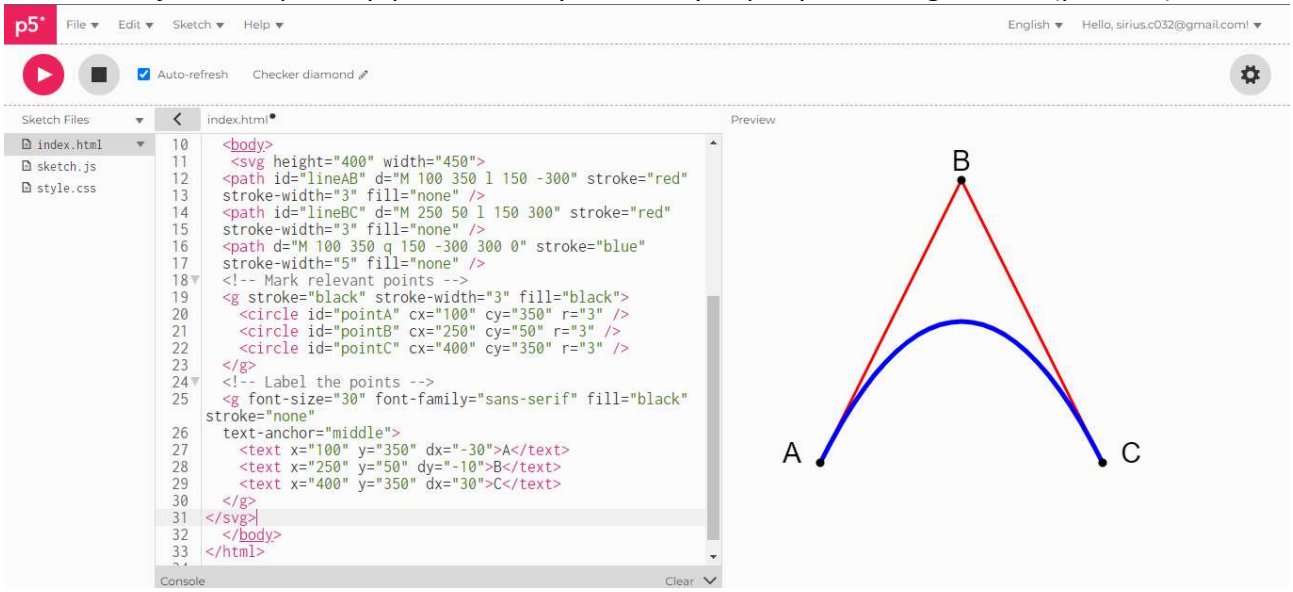


Рис. 3.4. Результат запуска программы

Результат выполнения программы можно увидеть в правой части редактора.

Программное обеспечение для векторной графики

Платное программное обеспечение для векторной графики

Adobe Illustrator (Mac, Win)

Adobe Illustrator — это профессиональное программное обеспечение, которое считается стандартом для редактирования графики в векторном формате. Он позволяет создавать логотипы, рисунки и иллюстрации для печати, интернета, видео и мобильных устройств.

Affinity Designer (Mac, Win)

Affinity Designer - программа для редактирования графики в векторном формате, быстрая, однородная и точная.

Graphic (Mac)

Graphic (Mac) — программное обеспечение для графического редактирования для дизайна и иллюстраций. Создавайте подробные технические иллюстрации или впечатляющие художественные рисунки.

Corel Draw (Win)

Corel Draw — это программное обеспечение для редактирования векторной графики, используемое в графическом дизайне. Возможность ограниченного взаимодействия с графическими форматами файлов Adobe Illustrator.

Sketch (Win, Mac)

Sketch — это программное обеспечение для редактирования графики с набором инструментов дизайна, предназначенных для создания графики в векторном формате, а именно для веб-приложений и мобильных приложений.

Бесплатное программное обеспечение для векторной графики

Inkscape (Mac, Win, Linux)

Inkscape — это графическое программное обеспечение для редактирования в Windows, Mac OS X и Linux. Является бесплатным с открытым исходным кодом.

Gravit Designer (Mac, Win, Linux, Chrome OS, браузер)

Gravit Designer — это бесплатное онлайн-приложение для векторного дизайна и редактирования. Можно редактировать векторные файлы онлайн, прямо в браузере: Gravit Designer Online

Vectr (Mac, Win, Linux, Chrome OS, браузер)

Vectr — это бесплатное программное обеспечение для редактирования графики, используемое для создания векторной графики. Это простой и мощный инструмент для веб-платформ и настольных платформ.

Программа векторной графики *Vectr*

Можно использовать этот бесплатный векторный редактор на компьютере или в браузере для создания векторов и другой графики. Это действительно мультиплатформенное и простое в использовании программное обеспечение: доступны версии приложения для Mac, Windows, Linux или Chromebook.

Vectr является удобным инструментом и широко используется для аннотирования, редактирования изображений и рисования макетов и диаграмм. Кроме того, его можно использовать для:

— логотипов;

– водяных знаков;

– макетов сайтов;

– баннеров в социальных сетях;

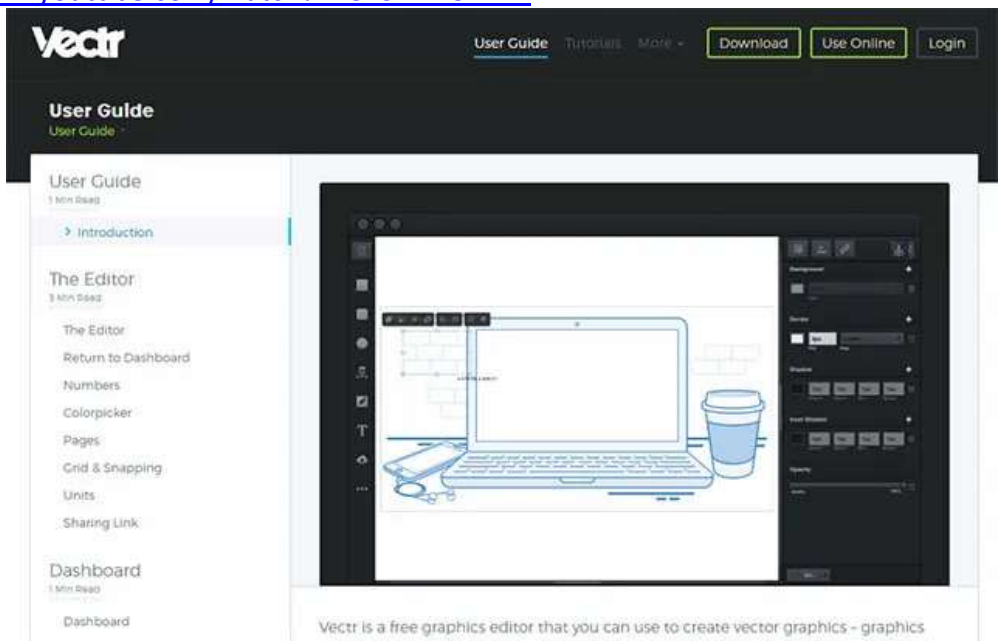
– иконок;

– и практически любой другой вид 2D-искусства.

Программа позволяет создавать масштабируемые векторные рисунки с помощью простых и понятных функций. Его не сложно освоить, он отлично подойдет тем, кто только начинает знакомиться с векторной графикой и кому достаточно базовых возможностей иллюстрации.

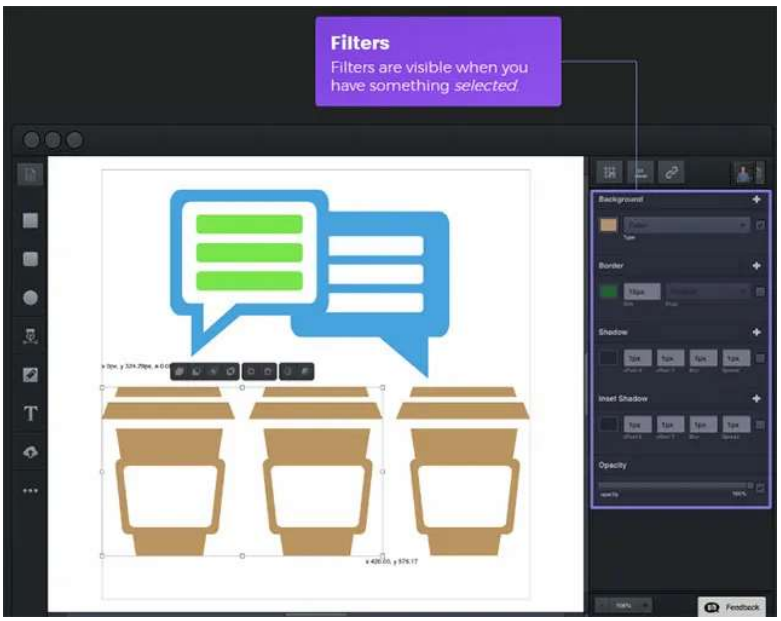
Как работает *Vectr*

Когда вы начнете углубляться в специфику этого программного обеспечения, первое, с чем вы столкнетесь, это целый набор интерактивных руководств. С Vectr вы узнаете, как создавать и редактировать контуры, поворачивать и масштабировать объекты, использовать инструменты в рабочей области и управлять слоями. На Youtube есть несколько видео-гидов. https://www.youtube.com/watch?v=CnSRzM91FYY



Чтобы начать, просто загрузите или перетащите изображение в Vectr. Допускается импорт файлов в формате EPS, AI, SVG, PNG или JPEG – это значительно облегчает процесс создания векторной графики.

Интерфейс тщательно продуман и удобно организован, поэтому у новичков не возникнет трудностей с ненужными функциями. Вкладки Страницы и Слои в левом верхнем углу позволяют контролировать процесс создания проектов. Панель фильтров расположена в правой части интерфейса. Вы можете сделать иллюстрации еще более интересными, например, изменив угол, добавив тени, края или эффект размытия.



По окончании проектирования программа позволяет экспортировать чертежи в PNG, JPEG или SVG. В качестве примечания, SVG является единственным масштабируемым векторным форматом, который можно открыть в других приложениях, в то время как растровые изображения PNG и JPEG больше подходят для Интернета.

Очень интересно, что много полезной информации и практических советов в разделе обучения посвящено новичкам. Команда разработчиков векторного редактора Vectr бесплатно предлагает пошаговые инструкции по решению самых популярных дизайнерских задач: создание иконок, логотипов, типографики, меню, коллажей, инфографики и многое другое.

Лабораторная работа 3

Тема: Изучение примитивов векторной графики

Цель работы: Получить практические знания в области синтеза 2D векторной графики сцен, используя простые SVG графические примитивы.

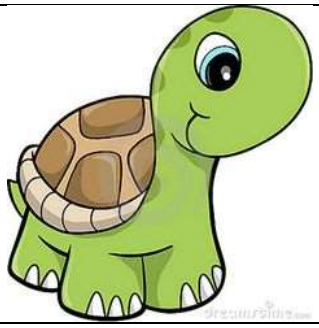
Задача работы:

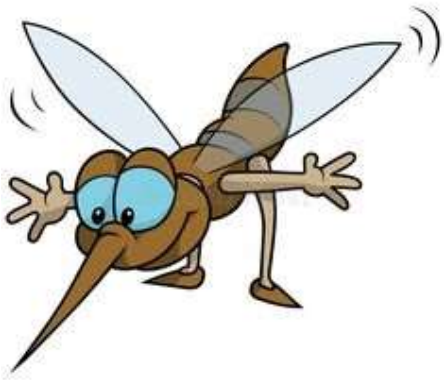
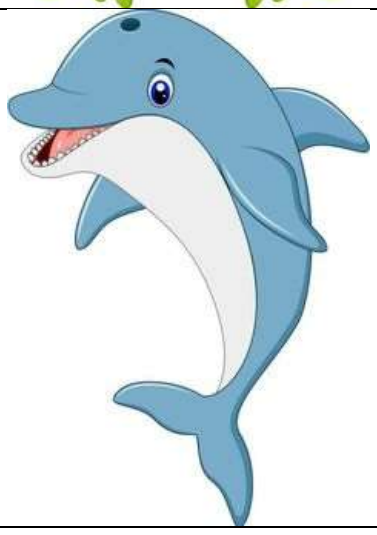
1. Разработать программу для синтеза статической 2D сцены с использованием необходимых графических элементов, таких как: *<rect>, <circle>, <ellipse>, <line>, <polyline>, <polygon>, <path>*, вместе с соответствующими атрибутами сцена должна содержать *<text>* размещенный в правом нижнем углу экрана, который бы указывал название, имя и группу студента.

2. Разработать векторную версию персонажа, нарисованного в таблице 3.3 согласно варианту. Разрешено создавать изображение в любом редакторе векторной графики.

Таблица 3.3 Варианты для лабораторной работы

Вариант	Персонаж	Вариант	Персонаж
1		17	
2		18	
3		19	
4		20	

Вариант	Персонаж	Вариант	Персонаж
5		21	
6		22	
7		23	
8		24	
9		25	

Вариант	Персонаж	Вариант	Персонаж
10		26	
11		27	
12		28	
13		29	

Вариант	Персонаж	Вариант	Персонаж
14		30	
15		31	
16		32