

1. Лабораторная работа №1. Разработка X3D-сцены, её представление в формате HTML-страницы и публикация в Web

1.1 Цель работы

Целью работы является ознакомление с принципами использования геометрических объектов для построения X3D-сцен. Ознакомление с узлами пространственных преобразований, группировки и тиражирования объектов X3D-сцены, создания гиперссылок и задания свойств материалов и текстур геометрических объектов, а также принципами создания HTML-страниц с внедренным X3D-кодом.

1.2 Порядок выполнения лабораторной работы №1

- На основе варианта задания (Таблица 1.1.) разработать композицию X3D-сцены. При необходимости включить в сцену дополнительные геометрические X3D-объекты в соответствии с сюжетом композиции.
- Пользуясь описанием геометрических узлов стандарта X3D и применяя пространственные преобразования реализовать сцену в формате HTML-страницы с внедренным X3D-кодом. Для реализации использовать любой текстовый редактор (Notepad++, Sublime Text, Visual Studio Code или т.п.).
- Пользуясь геометрическими узлами, объединяя их в именованные группы и применяя к ним пространственные преобразования, реализовать сцену в виде HTML-страницы с внедренным X3D-кодом.
- Для некоторых объектов (или групп объектов) применить тиражирование с использованием DEF/USE.
- Задать для объектов X3D-сцены свойства материалов и текстур.
- Для некоторых объектов (или групп объектов) применить тиражирование с использованием DEF/USE.
- Привязать к некоторым объектам гиперссылки на другие сцены или произвольные ресурсы.
- Задать в сцене фон (Background) и добавить ФИО автора работы.

- Сохранить сцену в формате HTML-страницы с внедренным X3D-кодом.
- Продемонстрировать результат преподавателю в web-браузере и в программе, используемой для написания X3D-кода.
- Оформить отчет и опубликовать его в личном кабинете АИС ГУАП.

Примечание. В приложении к ЛР №1. приведён пример листинга HTML-страницы с внедренным X3D-кодом

Таблица 1.1. Варианты для выполнения лабораторной работы №1

[illegible]

IndexedTriangleStripSet		+		+		+		+		+		+		+		+		+		+
ElevationGrid	+		+		+		+		+		+		+		+		+		+	
Extrusion		+		+		+		+		+		+		+		+		+		+
	Группирование и встраивание																			
DEF/USE	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Group	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Inline	+		+		+		+		+		+		+		+		+		+	
Anchor		+		+		+		+		+		+		+		+		+		+
	Текстурирование (Texture)																			
Appearance	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Material	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
ImageTexture	+		+		+		+		+		+		+		+		+		+	
PixelTexture		+		+		+		+		+		+		+		+		+		+

1.3 Краткие пояснения к выполнению лабораторной работы №1

Рекомендуемая литература и электронные ресурсы с описанием формата узлов стандарта X3D и примерами моделирования:

1. А.В. Аксенов. «Интерактивная компьютерная графика». Учебно-методическое пособие. СПб.: ГУАП.2020г. 89с.
2. Аксенов А.В. Каталог примеров по X3D [Электронный ресурс] URL: <https://aksenov.in/guap/x3dom/> (дата обращения 06.02.2023)
3. Официальная X3DOM документация. Fraunhofer [Электронный ресурс] URL: <https://www.x3dom.org/> (дата обращения 08.02.2023).
1. 4. Web 3D Consortium (W3C) -[Электронный ресурс] URL: <https://www.web3d.org/standards/version/V3.3> (дата обращения 08.02.2023).

1.3.1. Геометрические примитивы.

Узел Shape является основополагающим в любой сцене и призван определять одиночный видимый объект. В качестве дочернего он может содержать один из возможных геометрических узлов (для задания геометрических свойств определяемого объекта). Также, узел Shape может содержать узел Appearance, отвечающий за внешний вид формы, который в свою очередь содержит узел Material, определяющий цветовые свойства объекта и/или узлы, отвечающие за текстурирование. Таким образом, Shape содержит совокупность информации о геометрических свойствах и свойствах внешнего вида для одиночного объекта сцены.

Простые геометрические узлы в X3D описываются как один из заранее заданных геометрических примитивов, а детали отображения, в т. ч. триангуляция (разбиение на треугольники) осуществляется средствами WebGL. При этом для примитивов параметрически задаются их геометрические свойства: длины сторон для прямоугольного параллелепипеда, радиус для сферы и т.д.

В X3DOM определены следующие геометрические примитивы:

- Box – прямоугольный параллелепипед;
- Sphere – сфера;
- Cone – конус;
- Cylinder – цилиндр;
- Text (2D)– текст;
- Torus – тор;
- Dish – полусфера;
- Nozzle – сопло;
- Pyramid – усеченная пирамида;
- RectangularTorus – прямоугольный тор;
- Snout – косоусеченный конус;

1.3.2 Геометрические преобразования.

Узел Transform используется для выполнения над геометрическими объектами пространственных преобразований. Все вложенные в него узлы подвергаются описанному им преобразованию, которое может включать:

- масштабирование вдоль произвольных осей

- вращение вокруг произвольной оси
- смещение вдоль осей глобальной системы координат .

1.3.3. Внешний вид

Узел Appearance является дочерним узлом Shape и предназначен для вмещения в себе информации о внешнем виде объекта, который может быть описан заданием следующих параметров (в виде дочерних узлов):

- Material – материал;
- TwoSidedMaterial – двусторонний материал;
- ImageTexture – неподвижная текстура;
- MovieTexture – подвижная текстура;
- PixelTexture – пиксельная текстура;
- TextureTransform – преобразование текстуры.

1.3.4. Фон и туман

Узел Background позволяет задавать для сцены фон, состоящий из набора цветов или панорамных текстур. Фон в сценах X3D может быть реализован либо в форме набора изображений, формирующих панораму, либо в виде набора цветов, представляющих небо и землю. Возможно совмещение обоих подходов.

1.3.5. Сложные геометрические узлы.

В X3D существуют геометрические узлы, позволяющие создавать объекты произвольной формы, не ограниченной набором заданных примитивов. Так же, как и узлы геометрических примитивов, они являются дочерними узлами узла Shape и схожим образом могут быть использованы в паре с узлом внешнего вида (Appearance), определяющим материал или текстурирование. Тем не менее, для сложных объектов существует возможность более детального задания параметров внешнего вида для отдельных поверхностей/полигонов/линий/точек с использованием узлов Color, ColorRGBA и Normal. Сложные объекты описываются на низком уровне – путем задания массива координат вершин полигонов (вертексов) с последующим их индексированием и объединением в поверхности. Сложные объекты можно представить поверхностью, состоящей из плоских соприкасающихся друг с другом полигонов.

Все объекты в процессе рендеринга подвергаются триангуляции – разбиению поверхности на треугольники. Основное правило при задании сложных объектов путем перечисления координат вершин их полигонов – избегать неплоских полигонов. Рекомендуется, чтобы все точки, образующие полигон, были компланарны, т.е. лежали в одной плоскости. Триангуляция неплоских полигонов может быть неоднозначной, что может вызвать непредсказуемые эффекты при рендеринге. В то же время проверка на плоскость полигонов не производится из соображений эффективности, и ответственность ложится целиком на разработчика.

В X3DOM определены сложные геометрические узлы:

- **PointSet** - создает набор несвязанных между собой точек.
- **IndexedLineSet** - позволяет определить набор ломаных линий определённого цвета в пространстве.
- **IndexedFaceSet** - служит для создания сложных геометрических фигур, состоящих из набора полигонов.
- **ElevationGrid** - служит для создания поверхностей из набора четырехугольников, определяя сетку возвышенностей над горизонтальной плоскостью.
- **IndexedTriangleSet** - служит для создания сложных геометрических фигур, состоящих из треугольников.
- **IndexedTriangleStripSet**
 - - служит эффективным способом создания объектов, состоящих из полосы треугольников.
- **Extrusion** - геометрический узел, который позволяет перемещать 2D-поперечное сечение вдоль линии пути в локальной системе координат. Масштабирование и повороты поперечных сечений могут создавать различные сложные 3D-формы геометрических объектов.

1.3.6 Тиражирование.

С помощью атрибута **DEF** можно задавать уникальное имя для объекта сцены и впоследствии использовать этот объект повторно с помощью указания

заданного имени в качестве значения атрибута **USE** узла того же типа. Это сокращает код и вычислительные затраты, т.к. объект создается единожды. Помимо этого, осмысленные DEF имена позволяют легче ориентироваться в коде сцены.

Имя, задаваемое с помощью DEF, должно быть буквенно-цифровым и начинаться с буквы. Допускается только латиница. Имена регистрозависимы.

1.3.7. Гиперссылки

К объектам сцены или их группам можно привязать гиперссылки, которые активируются при нажатии на объект.

Узел **Anchor** группирует дочерние узлы и связывает с ними гиперссылку.

Описание полей:

url – URL, на который осуществляется переход

parameter – параметры перехода. Значение parameter="target='_self'" указывает, что переход осуществляется в текущей вкладке (по умолчанию открывается новая). Например, <Anchor url="http://guap.ru" parameter="target='_self'">.

1.3.8. Встраивание.

В сцену X3DOM можно встроить внешний X3D-файл. Объекты из встроенной сцены появятся в основной, как если бы они были описаны в текущем файле.

Для этого используется узел **Inline**.

Описание полей:

url – набор путей к файлу сцены. Путь может быть несколько, в этом случае достигается большая надежность загрузки файла, поскольку по некоторым из них он может быть недоступен.

load – загружать ли файл сцены автоматически (true/false). Атрибут может быть выставлен в true динамически по какому-то событию, тогда подгрузка произойдет в этот момент, а не при начальной загрузке сцены. Например

```
<Scene>
  <Shape>
    <Snout></Snout>
    <Appearance>
      <Material></Material>
    </Appearance>
```

```

</Shape>
  <Transform translation="0 3 0">
    <Inline url="cube.x3d","http://website.com/x3d/cude.x3d">
  </Transform>
</Scene>

```

Во встраиваемом файле должен присутствовать корневой элемент <Scene>, чтобы он мог быть корректно транслирован как файл X3D. Для приведенного выше примера файл cube.x3d может иметь вид:

```

<Scene>
  <Shape
    <Box size='2 1 3'></Box>
    <Appearance>
      <Material diffuseColor='0 1 0'></Material>
    </Appearance>
  </Shape>
</Scene>

```

1.3.9. Группирование.

К группирующим узлам относятся **Group, Anchor, Transform, Collision**.

Все они способны включать в себя в качестве дочерних наборы других узлов сцены. При этом все дочерние элементы объединяются в единую группу.

Группирование позволяет достичь следующих целей:

- структурирование отдельных элементов сцены логическим образом;
- объединение связанных элементов для упрощения работы с ними;
- поддержание общей для группы системы координат, в пределах которой объекты могут быть легко позиционированы и ориентированы друг относительно друга;
- упрощение тиражирования групп связанных объектов.

Группирующие узлы могут быть вложены друг в друга:

```

<Group DEF="Gr1">
  ...
  <Transform DEF="Tr">
    ...
    <Group DEF="Gr2">
      ...
    </Group>
  </Transform>
</Group>

```


Узел **Collision** - позволяет задать, проходит ли аватар (пользователь) сквозь геометрические объекты сцены. Является группирующим узлом и определяет соответствующие свойства для всех своих дочерних узлов.

1.4 Содержание отчета

- 1) Титульный лист;
- 2) Цель работы;
- 3) Номер варианта (таблица 1.1);
- 4) Словесное описание сцены.
- 5) Графическое представление графа сцены в терминах HTML-узлов
- 6) Листинг HTML-страницы с внедренным X3D-кодом.
- 7) Скриншоты HTML-страницы с X3D-сценой в Web-браузере.
- 8) Выводы.

1.5. Контрольные вопросы

1. Какие геометрические примитивы можно использовать при построении сцен?
2. Какие геометрические трансформации определены в X3DOM?
3. В каком порядке применяются трансформации, заданные узлом Transform, к объекту?
4. Какими способами можно задать внешний вид объекта?
5. Каковы составляющие материала объекта?
6. Какие виды текстур используются в X3DOM?
7. Что позволяет делать узел преобразования текстуры?
8. Каковы принципы задания сложных геометрических объектов в X3DOM?
9. Какие типы сложных геометрических объектов доступны в X3DOM?
10. Для каких целей служит группировка узлов в X3DOM?
11. По каким принципам осуществляется тиражирование объектов сцены?
12. Какими способами можно задать фон для сцены в X3DOM?

Приложение к ЛР №1. Листинг HTML-страницы с внедренным X3D-кодом.

```
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="utf-8" />
  <title> Lab1 X3DOM </title>
  <script src="https://www.x3dom.org/download/dev/x3dom-full.js"></script>
  <link rel='stylesheet' type='text/css'
href='https://www.x3dom.org/download/x3dom.css'>
  <style>
    x3d {
      border: 4px solid darkorange;
      background-image: url('GUAP_SUAI.png');
    }
    body {
      font-size: 100%;
      font-family: verdana, sans-serif;
      background: rgba(128, 128, 128, 0.4);
      color: darkred;
    }
    h1
    {color: darkred;
    }
  </style>
</head>
<body>
<h1> кафедра 44 </h1>
<p>
  Это html-страница с 3d-объектами.
</p>
```

```
<x3d width='800px' height='600px' >
  <scene>
    <shape>
      <appearance>
        <material diffuseColor='1 0 0'></material>
      </appearance>
      <box size='2 2 2'></box>
    </shape>
    <transform translation='0 4.5 0'>
      <shape>
        <appearance>
          <material diffuseColor='0 1 0'></material>
        </appearance>
        <cone></cone>
      </shape>
    </transform>
    <transform translation='-3 2.5 0'>
      <shape>
        <appearance>
          <material diffuseColor='0 0 1'></material>
        </appearance>
        <sphere radius="1.2" />
      </sphere>
    </shape>
  </transform>
</scene>
</x3d>
</body>
</html>
```

кафедра 44

Это html-страница с 3d-объектами.

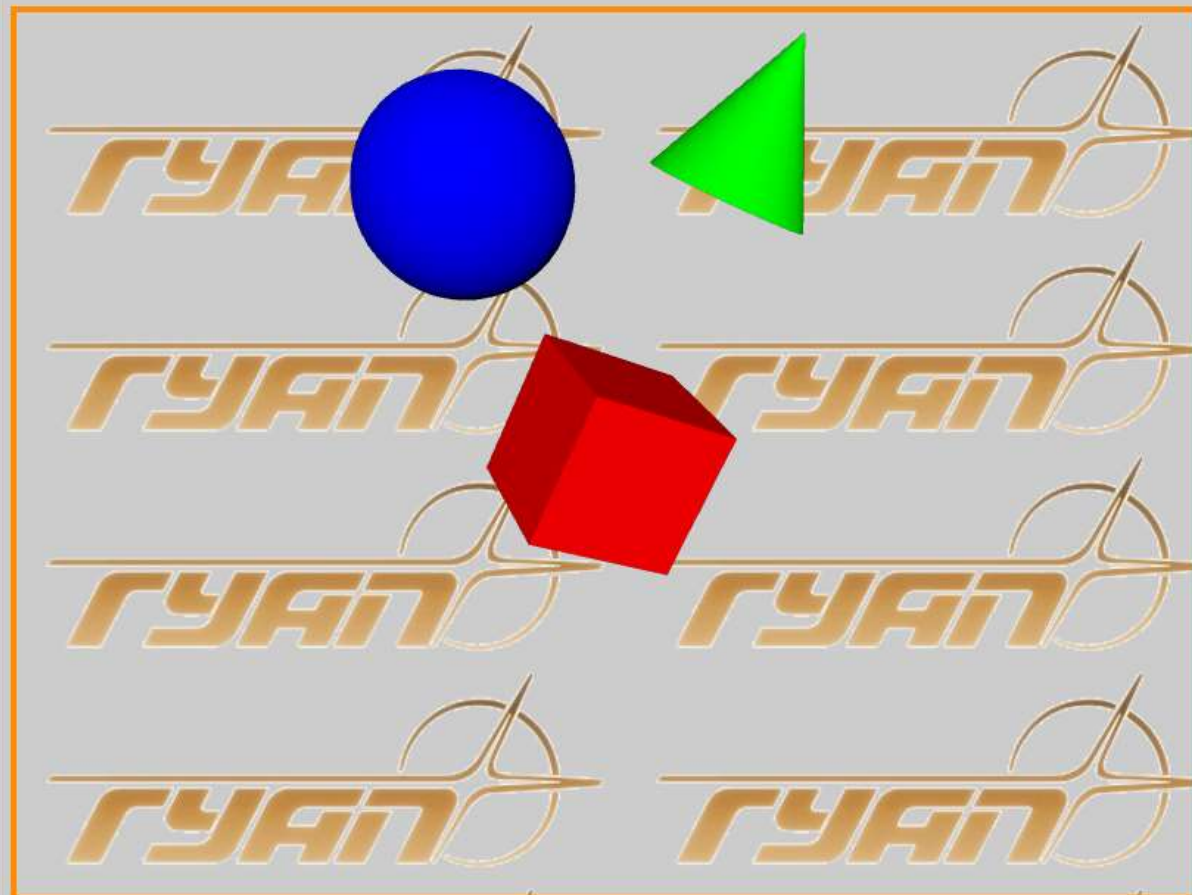


Рисунок 1 – визуализация HTML-страницы с X3D-кодом в Web-браузере