2. Лабораторная работа №2. Создание динамических X3D-сцен.

2.1 Цель работы

Целью работы является приобретение навыков создания динамических X3D-сцен с использованием сенсоров и интерполяторов различных типов, а также с применением обработки DOM-событий на языке JavaScript в коде HTML-страницы. Установка источников освещения и задание способов навигации X3D-сцен,

2.2. Порядок выполнения лабораторной работы №2

- На основе варианта задания дополнить композицию X3D-сцены (подготовленной в ЛР№1) интерактивными элементами.
- Пользуясь геометрическими узлами, объединить их в именованные группы DEF/USE, применить к ним пространственные преобразования.
- Линейная анимация объектов. По заданному варианту (Табл.2.1) задать необходимое число таймеров и маршрутами привязать к анимируемым свойствам геометрических X3D-объектов, интерполяторы значений соответствующего типа.
- Для добавления интерактивности использовать сенсоры согласно заданному варианту и маршрутами привязать к геометрическиv X3D-объектам.
- Для добавления интерактивности с использованием обработки DOM-событий применить скрипты с JavaScript кодом, как минимум для одного элемента управления HTML (например, нажатие ЛКМ или др.) к геометрическим объектам X3D-сцены.
- Задать для сцены освещение согласно варианту задания.
- Задать не менее 3-х точек наблюдения в сцене.
- Задать параметры навигации пользователя по сцене.
- 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.

Таблица 2.1. Варианты для выполнения лабораторной работы №2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Интерполяторы (X3D)																			
PositionInterpolator	+		+		+		+		+		+		+		+		+		+	
ScalarInterpolator		+		+		+		+		+		+		+		+		+		+
ColorInterpolator	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
CoordinateInterpolator	+		+		+		+		+		+		+		+		+		+	
OrientationInterpolator		+		+		+		+		+		+		+		+		+		+
	Сенсоры (X3D)																			
TimeSensor	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
TouchSensor		+		+		+		+		+		+		+		+		+		+
PlaneSensor	+		+		+		+		+		+		+		+		+		+	
SphereSensor		+		+		+		+		+		+		+		+		+		+
CylinderSensor	+		+		+		+		+		+		+		+		+		+	
	Скрипты (JavaScript)																			
Script	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
	Источники освещения, Камеры, Навигация																			
PointLight	+			+			+			+			+			+			+	
DirectionalLight		+			+			+			+			+			+			

SpotLight	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Viewpoint	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
NavigationInfo	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

2.3 Краткие пояснения к выполнению лабораторной работы №2

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

1. А.В. Аксенов. «Интерактивная компьютерная графика». Учебное пособие. СПб.: ГУАП.2020г. 89с. Электронная версия пособия опубликована в разделе «Материалы» АИС ГУАП. (дата обращения 01.09.2023)
2. Аксенов А.В. Каталог примеров по X3D [Электронный ресурс] URL: https://aksenov.in/guap/x3dom/ (дата обращения 31.08.2023)
3. Официальная X3DOM документация. Fraunhover [Электронный ресурс] URL: https://www.x3dom.org/ (дата обращения 31.08.2023).
4. Web 3D Consortium (W3C) -[Электронный ресурс] URL: https://www.web3d.org/standards/version/V3.3 (дата обращения 03.009.2023).

2.3.1. Генераторы событий.

Узлы X3D помимо полей, задаваемых при проектировании сцены определяют события (входные и выходные), которые могут быть связаны между собой, вызывать друг друга с целью получения анимации и позволяют пользователю взаимодействовать с объектами X3D-сцен.

Анимация X3D-объектов непосредственно связана с изменениями графа сцены. События, генерируемые через определенные промежутки времени, дают возможность создавать динамически изменяющиеся объекты X3D-сцены. Синтаксис узлов включает в числе параметров(полей) каркас событий(рис.1).

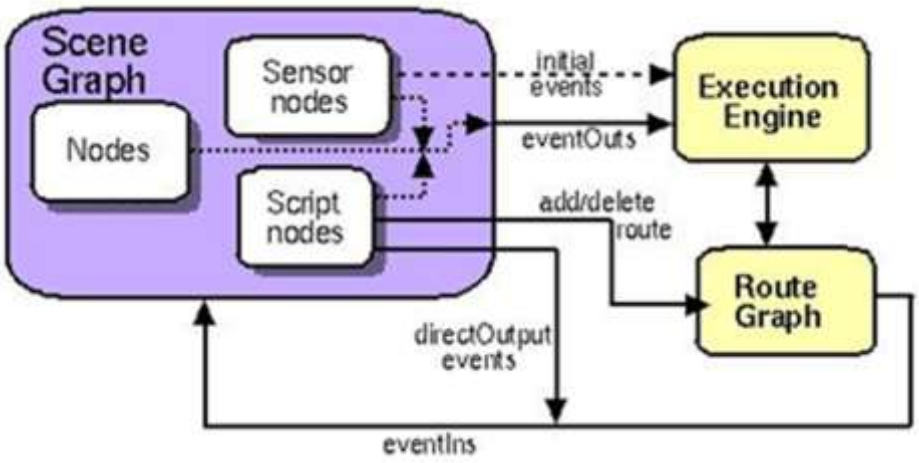

Рисунок 1. – Каркас событий

Для того чтобы одни события вызывали другие, необходимо связывать их между собой маршрутами (рис.2).

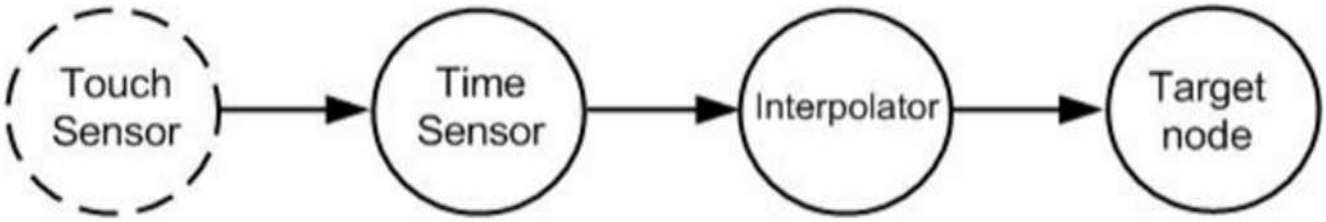

Рисунок 2. – Механизм маршрутов

Передача события от одного узла к другому, осуществляется при помощи маршрутов объявлением ключевых слов **ROUTE** и **TO.**

С их помощью происходит связывание действий, происходящих во времени и\или пространстве, т.е. их маршрутизация от одного узла к другому. Эти связи представлены графом маршрутов и дают возможность контролировать наступление того или иного события, сопровождающееся изменением свойств геометрических узлов.

Каждое событие имеет тип, связанный с ним, например, SFFloat, MFString,….

Соединены могут быть только события одного типа.

2.3.2. TimeSensor - счётчик времени

Основным генератором событий по времени служит узел TimeSensor – таймер.

Модель X3D предполагает, что любая анимация во времени может быть аппроксимирована кусочно-линейной функцией(рис.3).

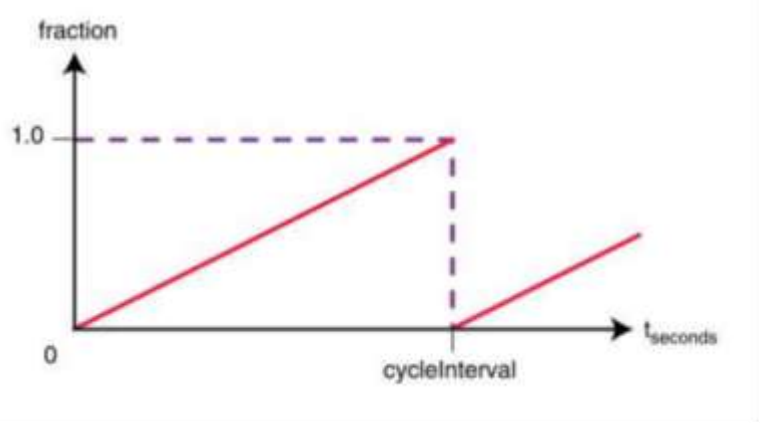

Рисунок 3. – Принцип изменения интервалов времени

Это позволяет разработчику X3D–сцены вручную задавать настолько детализированную функцию анимации, насколько это необходимо.

Синтаксис узла TimeSensor. Таймер генерирует события в заданном интервале времени.

TimeSensor : X3DTimeDependentNode, X3DSensorNode {

SFTime [in,out] cycleInterval 1 (0,∞) - интервал цикла (секунды).

SFBool [in,out] enabled - таймер включен (true) или отключен (false)

SFBool [in,out] loop – зациклен таймер или нет (true/false).

SFTime [in,out] pauseTime 0 (-∞,∞) – время, когда таймер будет поставлен на паузу.

SFTime [in,out] resumeTime 0 – время, когда таймер возобновит работу.

SFTime [in,out] startTime 0 (-∞,∞) – время включения таймера.

SFTime [in,out] stopTime 0 (-∞,∞) – время выключения таймера.

SFTime [out] cycleTime– это событие генерируется каждый раз, когда таймер достигает интервала цикла, и имеет значение текущего времени.

SFTime [out] elapsedTime – общее время работы таймера с момента включения (без учёта пауз)

SFFloat [out] fraction_changed– выходное событие, представляющее собой непрерывный поток сигналов, необходимый для интерполяторов. Событие генерируется постоянно, и так быстро, как возможно. Значение этого события

интерполируется во время интервала цикла от 0 до 1 (при достижении таймером значения, кратного интервалу цикла,
значение fraction_changed = 1).
SFBool [out] isActive– генерируется, когда таймер начинает работать или останавливается (true – таймер запущен, false – прекратил работу).
SFTime [out] time - генерируется так же, как и fraction_changed, и имеет значение текущего времени.
}
Подход к созданию линейной анимации таков: задаются ключевые точки анимации во времени, затем для значений между ними выполняется линейная интерполяция. Для реализации этой концепции используются узлы-интерполяторы, осуществяющие линейную интерполяцию анимируемой величины.

2.3.3. Узлы-интерполяторы (X3D)

Узлы-интерполяторы различаются по типу интерполируемого значения:

- PositionInterpolator – интерполятор позиции, выполняет интерполяцию между значениями одиночной координаты (тип SFVec3f),
- ScalarInterpolator – интерполятор скалярной величины, выполняет интерполяцию между значениями скалярной величины (одиночного вещественного числа, тип SFFloat).
- ColorInterpolator – интерполятор цвета, выполняет интерполяцию между значениями цвета (тройками вещественных чисел 0..1
- CoordinateInterpolator – интерполятор координат, выполняет интерполяцию между значениями набора координат в трехмерном пространстве (тип MFVec3f).
- OrientationInterpolator – интерполятор ориентации, выполняет интерполяцию между значениями вектора ориентации в пространстве (тип SFRotation).

Порядок действий при создании анимации:

1) Определите геометрический узел, который должен быть анимирован.

2) Укажите для него DEF-имя.

3) Определите поле, которое должно изменять значение и определите тип анимации.

4) Выберите интерполятор, генерирующий значения value_changed соответствующего типа. Например, PositionInterpolator генерирует события типа SFVec3f.

5) Добавьте TimeSensor, установите длительность периода анимации.

6) Создайте маршрут от поля fraction_changed таймера к полю set_fraction интерполятора.

7) Создайте маршрут от поля value_changed интерполятора к анимируемому полю нужного узла.

Пример. Перемещение геометрического объекта Cone(конус) в X3D-сцене с использованием PositionInterpolator.

```
<! Задание объекта конус в пространстве X3D-сцены>
        <transform DEF="Cone">
        <shape>
        <appearance>
           <material diffuseColor="0 1 0"></material>
        </appearance>
        <cone></cone>
      </shape>
      </transform>
<! Перемещение конуса с использованием механизма маршрутов>
<TimeSensor DEF="Time" cycleInterval="3" loop="true"></TimeSensor>
<PositionInterpolator DEF="Move" key="0 0.5 1" keyValue="0 0 0  0 3 0  0 2 0">
</PositionInterpolator>
      <Route fromNode="Time" fromField ="fraction_changed" toNode="Move"
toField="set_fraction"></Route>
```

<Route fromNode="Move" fromField ="value_changed" toNode="Cone" toField="translation"></Route>

2.3.4. Сенсоры (узлы-манипуляторы) X3D

Сенсоры, реализованные в стандарте X3D (версия 3.3) должны принимать ввод данных от пользователя, точнее от устройств ввода (клавиатура, мышь и т.п., которыми управляет пользователь), для изменения внешнего вида геометрических объектов либо изменения их расположения в X3D – сцене:

- ***PlaneSensor*** – датчик перемещения. Отслеживает и реализует действия пользователя по перемещению объектов геометрии. Перемещение происходит в одной плоскости. Объекты, к которым привязан этот датчик, можно двигать в плоскости, меняя локальные X и Y координаты. Локальная Z-координата считается равной 0.
- ***SphereSensor*** – датчик сферического вращения. Этот датчик позволяет вращать геометрические объекты вокруг любой оси. Датчик определяет нажатие кнопки мыши, когда указатель находится на привязанной к нему геометрии. Датчик можно прикрепить к геометрическим объектам, используя их в одной группе. Пользователь может перетаскивать датчик (при нажатой кнопке мыши передвигать указатель), тем самым вращая объекты.
- ***CylinderSensor*** – датчик цилиндрического (вокруг вертикальной оси) вращения. В отличии от SphereSensor, этот датчик позволяет вращать объекты только вокруг оси Y.
- ***TouchSensor*** – датчик соприкосновений. Узел TouchSensor отслеживает местоположение и состояние устройства ввода(манипулятор) и определяет, когда пользователь указывает на геометрию (нажимает на геометрические объекты), содержащуюся в родительской группе узла TouchSensor.
- ***StringSensor*** – динамическая строка, выводимая на экран. Действия пользователя, а так же происходящие в сцене события, могут влиять на содержимое этой надписи посредством использования маршрутов.

– ***KeySensor*** – датчик, реагирующий на нажатие клавиш клавиатуры, что позволяет использовать эту информацию для дальнейшей обработки.

2.3.5. Скрипты (Scripts)

Программы, написанные на языке JavaScript и выполняемые на стороне клиента, позволяют встраивать в web-страницы произвольную логику. В частности, X3DOM открывает доступ к объектам сцены как к элементам DOM, позволяя создавать, удалять элементы, изменять значения атрибутов и т.д. Действия срабатывают как реакция на события, которые могут исходить от пользователя. В JavaScript определено несколько полезных DOM-методов, которые можно применять для управления содержимым X3D-сцены:

- document.getElementById – поиск элемента по его идентификатору;
- document.createElement – создание элемента;
- setAttribute – установка значения атрибута;
- getAttribute – получение значения атрибута;
- appendChild – добавление элемента-потомка;
- removeChild – удаление элемента-потомка.

В браузере возникают различного рода события, в том числе инициированные пользователем. Для них можно назначать обработчики в виде кода на JavaScript. Рассмотрим, как можно обработать событие наведения курсора (mouseover) на элемент, и щелчка левой кнопкой мыши (click) по геометрическому объекту X3D-сцены.

Чтобы задать код, обрабатывающий событие щелчка по сфере, необходимо указать его в специальном атрибуте onclick.

Пример.

```
<script>
    function changeColor()
    {
      if(document.getElementById("color").getAttribute('diffuseColor')=="1 0 0")
         document.getElementById("color").setAttribute('diffuseColor', '0 0 1');
      else
```

```
        document.getElementById("color").setAttribute('diffuseColor', '1 0 0');
    }
        </script>
```

Обращение к функции changeColor() в X3D сцене

……

```
    <shape onclick="changeColor();">
      <appearance>
        <material id ="color" diffuseColor='1 0 0'></material>
      </appearance>
      <box></box>
    </shape>
```

Здесь в обработчике DOM-события click происходит вызов функции changeColor(), которая обеспечивает изменение цвета геометрического узла Box при щелчке по нему левой кнопкой мыши.

Аналогично задаются обработчики других событий.

Примечание 2. Полную документацию языка JavaScript можно найти, например, на сайте http://learn.javascript.ru.

2.3.6. Освещение

Для проектирования реалистичных трехмерных сцен недостаточно разработать Геометрическое представление объектов и свойства их внешнего вида.

Важную роль играет освещение сцены с использованием узлов-источников света, которые могут не только повысить реалистичность сцены, но и добиться интересных визуальных эффектов. Принцип действия освещения в X3D подобен физическим явлениям реального мира, но является их сильно упрощенной аппроксимацией, поскольку рендеринг осуществляется в реальном времени, и вычислительная нагрузка при вычислении освещения не может быть слишком интенсивной. Среди упрощений модели освещения следует отметить отсутствие отражений и преломления.

Узлы-источники освещения не создают какой-либо сопутствующей геометрии,

их наличие в сцене можно определить по освещенности других объектов. Если необходимо визуальное представление источника, необходимо разработать его вручную, как геометрический объект и сгруппировать с источником освещения. Описание общих полей:

- ambientIntensity – доля источника в рассеянном освещении сцены за счет отражения от объектов (от 0 до 1). Рассеянное освещение не имеет направленности и освещает все поверхности одинаково. В качестве примера можно привести освещенность интерьера комнаты в дневные часы (даже при отсутствии прямой видимости солнца).
- color – цвет освещения.
- intensity – интенсивность освещения (от 0 до 1).
- on – источник света включен (при значении true) или выключен (при значении false).
- global – является ли источник глобальным (true) или локальным (false).

Глобальный источник освещает все объекты сцены, локальный – только содержимое группирующего узла, в котором он размещен. Данный прием может использоваться для избегания ненужного освещения (например, когда источник освещения внутри комнаты освещает предметы снаружи), а также для уменьшения вычислительных затрат путем сокращения количества источников, которые нужно принимать во внимание при расчете освещенности поверхности. Типы источников освещения:

- DirectionalLight - создает источник направленного освещения.
- PointLight - задает точечный источник света, который излучает во всех направлениях.
- SpotLight - определяет источник освещения, который имеет свое местоположение и светит в определенном направлении коническим пучком лучей. Результатом освещения является световое пятно с размытием.
- Headlight не является узлом X3D, а представляет собой встроенный в браузер источник освещения типа DirectionalLight, который фиксирован в

положении и ориентации текущей точки наблюдения пользователя. Источник включен по умолчанию и может быть отключен булевым полем headlight узла NavigationInfo.

2.3.7. Навигация.

Навигация пользователя по X3D-сцене является важным компонентом интерактивности. Стандарт X3D предоставляет расширенные возможности управления навигацией: задание набора точек наблюдения, контроль параметров перемещения пользователя по сцене.

Узел **NavigationInfo** - позволяет задать параметры перемещения пользователя по сцене. Описание полей:

- avatarSize – определяет размеры аватара пользователя. Представляет собой тройку вещественных чисел:
  - a – размер по горизонтали – используется для проверки на столкновение с другими объектами.
  - b – размер по вертикали – определяет, насколько высоко над объектами находится позиция наблюдения.
  - c – максимальная высота объектов, которые можно “перешагнуть”. По этому признаку можно отличить к примеру, лестницу, на которую можно подняться, от стены, которая является непреодолимым препятствием (в режиме “WALK”).
- headlight – включена ли подсветка для сцены по умолчанию в виде направленного источника освещения белого цвета единичной интенсивности, совпадающего с направлением взгляда пользователя. При значении false освещение по умолчанию выключено, необходимо использовать создаваемые вручную источники освещения.
- speed – скорость перемещения по сцене (м/с).
- type – тип навигации, который будет установлен в браузере. Можно указывать
- несколько типов навигации, среди которых пользователь может выбрать наиболее удобный для него. Возможные значения параметра:
  - "ANY" – доступны все режимы навигации по выбору пользователя.

  - "WALK" – ходьба. Пользователь перемещается по геометрии земной поверхности, если она задана, проверка на столкновения не дает ему провалиться сквозь землю. Гравитация включена.
  - "EXAMINE" – изучение. Движением мыши вращается вся сцена вокруг центра вращения текущей точки наблюдения.
  - "FLY" – полет. Отличается от ходьбы отсутствием гравитации.
  - "LOOKAT" – рассматривание. Пользователь может указывать произвольные объекты для рассматривания, что влечет перемещение и переориентацию камеры и смену центра ее вращения.
  - "NONE" – навигация отключена, возможно только переключение между точками наблюдения.
- visibilityLimit – определяет, как далеко пользователь может видеть. Рендеринг за пределами этого значения браузер не проводит. Значение по умолчанию соответствует бесконечному пределу.
- transitionType – тип пути, по которому следует камера при переключении между точками наблюдения. Возможные значения параметра:
  - “ANIMATE” – сглаженное перемещение;
  - “LINEAR” – линейная интерполяция позиции и ориентации;
  - “TELEPORT” – мгновенное перемещение в точку назначения.

2.3.8. Настройка точек наблюдения (камеры).

Узел **Viewpoint** - позволяет задавать местоположение и ориентацию точек наблюдения. Точку наблюдения можно отождествить с камерой, через которую пользователь смотрит на мир. Точек наблюдения в сцене может быть произвольное количество, каждая из них может обладать своими свойствами и располагаться удобно с точки зрения целей, которые хочет достичь разработчик сцены. Некоторые из точек наблюдения могут быть анимированы. Переключение между точками осуществляется с помощью клавиш PgUp/PgDn.

Описание полей:

- fieldOfView – угол обзора камеры в радианах. Анимация этого поля может давать интересные эффекты.

- orientation – начальная ориентация точки наблюдения.
- position – начальная позиция точки наблюдения.
- description – описание точки наблюдения (выводится в браузере).
- centerOfRotation – координата точки в пространстве, вокруг которой происходит вращение камеры, если в узле NavigationInfo установлен режим EXAMINE. Если пользователь выбирает режим LOOKAT и выбирает другую точку значение этого поля изменяется.

2.4 Содержание отчета

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

2.5. Контрольные вопросы

1. Каковы принципы создания линейной анимации в X3DOM?
2. Как работает механизм узлов-интерполяторов?
3. Для чего нужны маршруты?
4. Какие виды событий определены для узлов DOM?
5. Какими способами можно задать обработчик DOM-события?
6. Каковы недостатки линейной анимации с использованием интерполяторов в X3DOM?
7. Какие механизмы существуют в JavaScript для работы с содержимым сцены?
8. Для каких целей служит группировка узлов в X3DOM?

9. По каким принципам осуществляется тиражирование объектов сцены?
10. Какие типы источников освещения доступны в X3DOM?
11. Какие свойства навигации можно задать для сцены?
12. Какими способами можно задать фон для сцены?