УДК 004.92
DOI 10.51691/2541-8327_2023_7_2

РАЗРАБОТКА КЛАССА EVENTCONTROLS ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ ТРЕХМЕРНЫХ ПРИЛОЖЕНИЙ В WEB С ПОМОЩЬЮ THREE.JS

Вильданов А.Н.
к.ф.-м.н., доцент
ФГБОУ ВО «Уфимский университет науки и технологий», Нефтекамский филиал
Нефтекамск, Россия

Аннотация: важными элементами интерактивного взаимодействия пользователя и приложения трехмерной графики являются обработка событий наведения или нажатия мышки, а также перетаскивание объектов. Например, в визуализациях различных инженерных, медицинских и пр. данных пользователи могут кликать на объекты, чтобы получить дополнительную информацию, или перемещать объекты, чтобы изучать их с разных ракурсов и взаимодействовать с ними. В статье рассматриваются примеры создания интерактивных трехмерных приложений в web. Предлагаемый в работе класс EventControls является пользовательским классом, разработанным для управления событиями взаимодействия с мышью в Three.js. Он предоставляет разработчику достаточно несложный и внятный интерфейс для создания интерактивных трехмерных приложений с помощью Three.js. Результаты, изложенные в статье, могут быть полезны как преподавателям ИТ-дисциплин, так и учащимся ИТ-направлений, при разработке курсовых и пр. работ.

Ключевые слова: JavaScript, WebGL, Three.js, 3D-моделирование, компьютерная графика, интерактивность, drag-and-drop, Eventcontrols.

DEVELOPMENT OF THE EVENTCONTROLS CLASS FOR CREATING INTERACTIVE THREE-DIMENSIONAL APPLICATIONS ON THE WEB USING THREE.JS

Vildanov A.N.
candidate of Physical and Mathematical Sciences,
Ufa University of Science and Technology, Neftekamsk branch
Neftekamsk, Russia

Abstract: important elements of interactive interaction between the user and the application of three-dimensional graphics are the handling of mouseover or click events, as well as the dragging of objects. For example, in visualizations of various engineering, medical, and other data, users can click on objects to get additional information, or move objects to view them from different angles and interact with them. The article discusses examples of creating interactive three-dimensional applications in the web. The proposed EventControls class is a custom class designed to handle mouse interaction events in Three.js. It provides the developer with a fairly simple and clear interface for creating interactive 3D applications using Three.js. The results presented in the article can be useful both for teachers of IT disciplines and students of IT areas in the development of term papers and other papers.

Keywords: JavaScript, WebGL, Three.js, 3D modeling, computer graphics, interactivity, events, drag-and-drop, Eventcontrols.

Сегодня `WebGL` позволяет создавать сложные и производительные 3D-приложения, такие, как визуализация данных, игры, симуляции и виртуальная реальность, непосредственно в браузере [1]. `Three.js` – это библиотека JavaScript, которая упрощает процесс создания `WebGL`-приложений,

предоставляя разработчикам высокоуровневые инструменты для создания и отображения объектов 3D-графики в веб-браузере [2].

Важными элементами интерактивного взаимодействия пользователя и приложения трехмерной графики являются, например, определение объектов трехмерной сцены (пространства), на которые пользователь кликнул мышкой, а также перетаскивание объектов. Например, в различных инженерных, медицинских и пр. визуализациях данных пользователи могут кликать на объекты, чтобы получить дополнительную информацию, или перемещать объекты, чтобы изучать их с разных ракурсов.

Различные виртуальные тренажеры, образовательные приложения или интерактивные презентации используют элементы интерактивности для того, чтобы предоставить пользователям возможность взаимодействовать с трехмерным контентом и активно участвовать в обучении или познании информации.

Наконец, в играх интерактивность сцен важна для реализации различных игровых механик, таких, как выбор и перемещение объектов, взаимодействие с ними, сбор предметов и многое другое.

В `Three.js` для этих целей используется класс `Raycaster`. Именно он помогает осуществлять т.н. «рейкастинг» (raycasting) – определение объекта, на который (первым) указывает луч (ray) в трехмерном пространстве. Данный луч строится по направлению от нас (камеры) до точки, на которую пользователь щелкнул мышью [3].

Несмотря на то, что `Three.js` предоставляет высокоуровневый программный интерфейс (API) для упрощения создания и взаимодействия с трехмерной графикой в веб-приложениях, программная реализация интерактивности может вызвать у разработчиков определенные трудности.

Поэтому возникла идея разработать класс `Eventcontrols`, который как раз служит для упрощения создания интерактивных приложений.

Он работает с `Three.js` и предоставляет следующие возможности:

– добавление событий наведения мышки на объекты;

– добавление событий клика мышкой по объекту;

– перетаскивание объектов с помощью мышки;

– добавленные события работают как с классом `THREE.Mesh`, так и с классом-контейнером `THREE.Group`. Последний позволяет группировать несколько трехмерных объектов `Mesh` вместе, чтобы ими можно было легко манипулировать как единым целым.

Для иллюстрации возможностей класса рассмотрим пример создания виртуального пианино (рисунок 1):

Рисунок 1 – Виртуальное пианино

Наш музыкальный инструмент будет состоять из восьми клавиш и обладает следующими возможностями и свойствами:

– при наведении мышки на клавишу она становится светлее;

– при потере фокуса к клавише возвращается исходный цвет;

– при клике клавиша опускается вниз на короткое время, инструмент проигрывает звук ноты, и клавиша возвращается на исходную позицию.

Объявим массив `dragObjects`, который будет использоваться для хранения объектов, которые пользователь хочет сделать интерактивными:

```
var dragObjects = [];
```

Создадим клавиши в виде трехмерных кубоидов, и разметим их в массиве:

```
var geometry = new THREE.BoxGeometry( 20, 20, 80 );

for ( var i = 0; i < 8; i ++ ) {
     var key = new THREE.Mesh( geometry, autoMaterial );
     …
     scene.add( key );
     dragObjects.push( key );
}
```

Создадим экземпляр класса `EventControls`:

```
eventControl = new EventControls(
       [ ... dragObjects ], camera, renderer.domElement );
```

Теперь устанавливаем обработчик события `mouseOver`, которое происходит, когда указатель мыши наводится на объект на сцене. При наведении на клавишу ее материал меняется на более светлый, при этом также изменится указатель мышки, а в консоли появится номер выделенной клавиши:

```
eventControl.attachEvent( 'mouseOver', function () {
     container.style.cursor = 'pointer';
     this.event.object.material = selMaterial;
     console.log( 'the key at number ' +
                        this.event.item + ' is select' );
});
```

Далее нужно описать событие, когда курсор мыши покидает наведенную клавишу. Курсор меняет вид на обычный, и цвет клавиши возвращается к прежнему:

```
eventControl.attachEvent( 'mouseOut', function () {
     container.style.cursor = 'auto';
     this.event.object.material = autoMaterial;
});
```

Наконец, при нажатии клавиша опускается ниже, проигрывается звук, в консоли выводится номер нажатой клавиши. Через определенное время клавиша возвращается на место:

```
eventControl.attachEvent( 'click', function () {
     this.event.object.position.y = -20;
     audios[ this.event.item ].play();
    setTimeout( () =>
     dragObjects[ this.event.item ].position.y = 0, 100 );
     console.log( 'the key at number ' +
                          this.event.item + ' is pressed' );
});
```

Как видим, код был небольшой, и, что более важно, достаточно простой для понимания. Ознакомиться с результатом можно в GitHub по адресу https://alexan0308.github.io/threejs/examples/controls_events_piano.html.

Следующая важная возможность, которую предоставляет класс `EventControls` – это перемещение трехмерных фигур (рисунок 2):

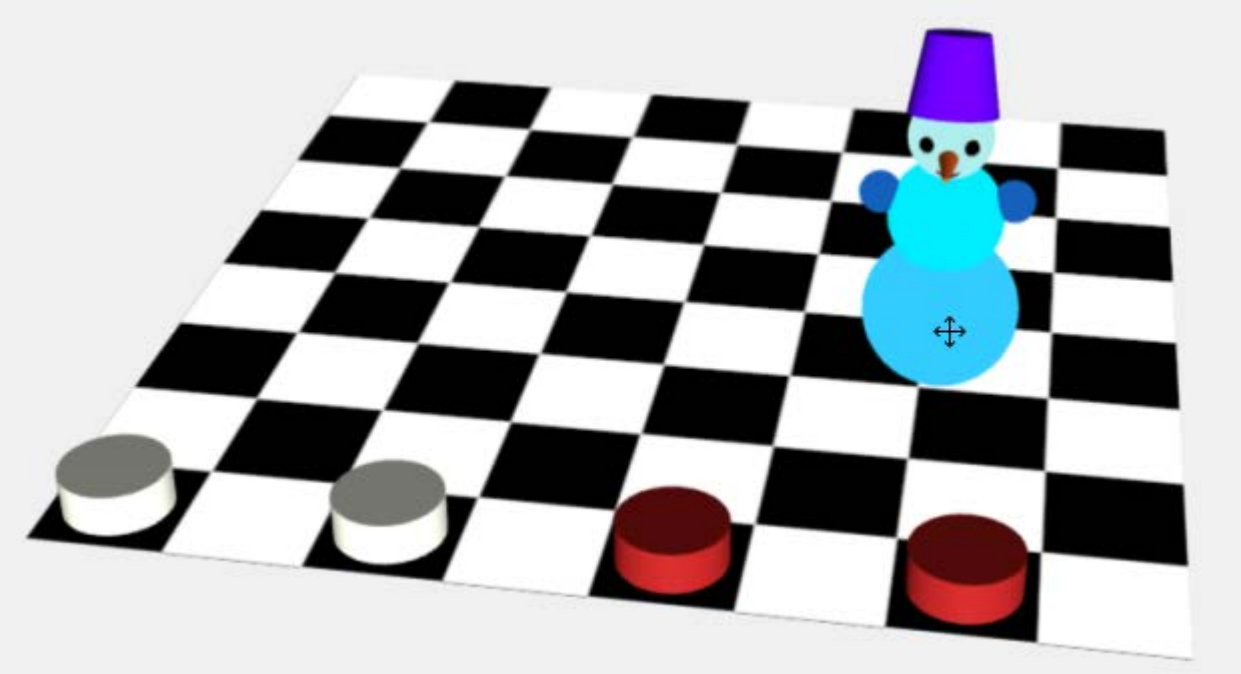

Рисунок 2 – Перемещение фигур

Мы рассмотрим различные классы фигур с разным поведением. Создадим белые шашки в виде цилиндров и фигуру снеговика, как группу из сфер и конусов [1]. Аналогично разместим эти фигуры в массив `dragObjects`, и укажем его при создании нового экземпляра класса `EventControls`:

```
eventControl = new EventControls(
        [ ... dragObjects ], camera, renderer.domElement );
```

Далее мы определим следующие свойства и параметры:

1. Устанавливается возможность перетаскивания объектов или элементов в трехмерной сцене:

```
eventControl.setDraggable( render, true );
```

Параметр `render` является объектом, отвечающим за отрисовку сцены, и передается в качестве аргумента. Второй параметр `true` указывает, что перетаскивание разрешено.

2. Указывается поверхность (в данном случае – шахматная доска) для нашего контроллера событий. Предполагается, что все перемещения трехмерных объектов осуществляются вдоль данной поверхности:

```
eventControl.setMap ( checkerboard );
```

3. Устанавливается контроль над объектом `orbitcontrol`. Этот объект позволяет пользователю вращать или перемещаться вокруг объектов или сцены при помощи мыши или других устройств ввода. Чтобы он не мешал перемещать объекты, мы его автоматически отключаем, когда мышка визуально находится в пределах поверхности `checkerboard`:

```
eventControl.setOrbitControl( orbitcontrol );
```

Этих команд достаточно, чтобы заставить трехмерные фигуры перемещаться в пределах шахматной доски с помощью мышки.

Если хочется задать некоторое поведение при перемещении, например, заставить фигуры перемещаться только вдоль черных клеток, создается слушатель события `dragAndDrop`:

```
eventControl.attachEvent( 'dragAndDrop', function () {
     this.event.object.position.x = 25 + 50 * Math.round(
          ( this.event.object.position.x - 25 ) / 50 );
     this.event.object.position.z = 25 + 50 * Math.round(
          ( this.event.object.position.z - 25 ) / 50 );
});
```

В данном коде округляются позиции объекта к ближайшему кратному 50 и добавляется смещение 25. Это позволяет «прилипнуть» шашкам к черным клеткам.

Ознакомиться с кодом и попробовать перемещать объекты можно по адресу https://alexan0308.github.io/threejs/examples/controls_events_example.html.

Проверим, как класс `EventControls` работает с готовыми трехмерными объектами, разработанными в 3D-редакторе (Autodesk 3ds Max, Autodesk Maya, Blender и т.д.).

Сначала с помощью `LoadingManager` из библиотеки `Three.js` определим выполнения действий, аналогичных ранее произведенным, после завершения загрузки модели:

```
loadingManager = new THREE.LoadingManager( function () {
     scene.add(dae);
     dragObjects.push( dae );
     eventControl = new EventControls(
       [ ... dragObjects ], camera, renderer.domElement );
     eventControl.setDraggable( render, true );
     eventControl.setMap ( checkerboard );
     eventControl.setOrbitControl( orbitcontrol );
     render();
} );
```

Загрузим на сцену саму трехмерную модель эльфа в формате `dae` (рисунок 3):

```
const loader = new ColladaLoader( loadingManager );
loader.load( './models/elf/elf.dae', function (collada) {
     dae = collada.scene;
     dae.position.set(0,0,0);
     dae.scale.set(25,25,25);
```

```
        dae.updateMatrix();
    } );
```

Рисунок 3 – Перетаскивание трехмерной модели эльфа в формате `dae`

Теперь модель можно перемещать по шахматной доске мышкой (https://alexan0308.github.io/threejs/examples/controls_events_loaders.html).

Итак, предлагаемый в работе класс `EventControls` является пользовательским классом, разработанным для управления событиями взаимодействия с мышью в `Three.js`. Он предоставляет разработчику достаточно несложный и внятный интерфейс для создания интерактивных трехмерных приложений с помощью `Three.js`. Класс `EventControls` позволяет назначить следующие события для взаимодействия с трехмерными фигурами:

– `mouseOver`;

– `mouseOut`;

– `onclick`;

– `dragAndDrop`.

Результаты, изложенные в статье, могут быть полезны как преподавателям ИТ-дисциплин, так и учащимся ИТ-направлений, при разработке курсовых и пр. работ. Класс `EventControls` можно использовать при разработке небольших трехмерных игр, в которых нужно кликать по предметам и перетаскивать их. Можно создавать наглядные обучающие виртуальные лабораторные работы по физике, химии, и т.д.

Ознакомиться с примерами, представленными в статье, можно по адресу: https://alexan0308.github.io/threejs/examples/.

Библиографический список:

1. Get started with WebGL [Электронный ресурс] / Microsoft. – Электрон. текстовые дан. – Режим доступа: http://msdn.microsoft.com/ru-ru/Library/dn385807(v=vs.85).aspx, свободный. – Загл. с экрана.

2. Three.js – JavaScript 3D library [Электронный ресурс] / Mr.doob. – Электрон. текстовые дан. – Режим доступа: http://threejs.org, свободный. – Загл. с экрана.

3. Вильданов А.Н. 3D-моделирование на WebGL с помощью библиотеки Three.js: учебное пособие. - Уфа: РИЦ БашГУ, 2014. – 114 с. – ISBN: 987-5-7477-3560-6.

Оригинальность 78%