



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.44

ПРОЕКТИРОВАНИЕ ДВИЖКА БРАУЗЕРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ: РАЗРАБОТКА МОДУЛЯ ПАРСИНГА ЭЛЕМЕНТОВ СТРУКТУРЫ HSML ДОКУМЕНТА ДЛЯ ПОСЛЕДУЮЩЕГО ОТОБРАЖЕНИЯ ЭЛЕМЕНТОВ В 3D СЦЕНЕ

¹ Кудинов Н.Г., ² Ралко К.И., ³ Ралко О.М.

ФГБОУ ВО «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ», Ростов-на-Дону, Россия, (344003, Ростовская область, город Ростов-на-Дону, пл Гагарина, зд. 1), e-mail: ¹ nkudinov8@gmail.com, ² kirillralco.education@gmail.com, ³ olga32-13-33@yandex.ru

Данная статья посвящена разработке движка браузера, поддерживающего технологию дополненной реальности (AR), с акцентом на создание модуля для парсинга элементов структуры документа HSML (Hyper Scene Markup Language) и их последующего отображения в 3D-сцене. В работе рассматриваются современные подходы к проектированию AR-браузеров, анализируются ключевые функциональные компоненты, необходимые для их эффективной работы, и предлагаются методы стандартизации структуры HSML для оптимизации процесса парсинга и валидации документов. Представлен демонстрационный модуль, реализованный на языке JavaScript, который интерпретирует HSML-документы и отображает AR-контент, используя three.js. Описаны основные технологические вызовы и предложены практические решения для улучшения производительности и совместимости с различными устройствами.

Ключевые слова: Дополненная реальность, браузер дополненной реальности, браузер AR, проектирование ПО.

DESIGNING AN AUGMENTED REALITY BROWSER ENGINE: HSML DOCUMENT STRUCTURE ELEMENTS PARSING FOR SUBSEQUENT ELEMENTS DISPLAY IN A 3D SCENE MODULE DEVELOPMENT

¹ Kudinov N. G., ² Ralko K.I., ³ Ralko O.M.

DON STATE TECHNICAL UNIVERSITY, Rostov-on-Don, Russia, (344003, Rostov region, Rostov-on-Don, Gagarin square, building. 1), e-mail: ¹ nkudinov8@gmail.com, ² kirillralco.education@gmail.com, ³ olga32-13-33@yandex.ru

This article is dedicated to the development of a browser engine supporting augmented reality (AR) technology, focusing on creating a module for parsing elements of the HSML (Hyper Scene Markup Language) document structure and their subsequent display in a 3D scene. The paper reviews modern approaches to AR browser design, analyzes the key functional components necessary for effective operation, and proposes methods for standardizing the HSML structure to optimize the parsing and validation process. A demonstration module, implemented in JavaScript, is presented, which interprets HSML documents and displays AR content using three.js. The main technological challenges are described, and practical solutions are proposed to improve performance and compatibility across various devices.

Keywords: Augmented reality, augmented reality browser, AR browser, software design.

Введение

С развитием технологий дополненной реальности (*перевод от англ. яз.: augmented reality, AR – аббр.*) всё большее значение приобретает создание специализированных браузеров, способных эффективно обрабатывать и отображать AR-контент. Одним из ключевых элементов таких браузеров является движок, ответственный за интерпретацию и

рендеринг документов, написанных на языках разметки, разработанных специально для AR. Одним из таких языков является разрабатываемый стандарт HSML (Hyper Scene Markup Language), который позволяет описывать трёхмерные сцены и объекты, а также их взаимодействие с окружающим миром [4].

Проектируемый браузер дополненной реальности состоит из множества модулей, каждый из которых выполняет специфические функции. Одним из наиболее критичных компонентов является модуль парсинга и отображения структуры [5] HSML документа. Данный модуль отвечает за интерпретацию HSML-кода, построение внутренней представительной структуры данных и взаимодействие с модулем рендеринга AR контента на экране пользователя.

В рамках настоящей работы рассматривается вариант предполагаемой структуры концепта HSML, методы ее стандартизации и описания, приводится демонстрационный модуль парсинга такой структуры с отображением результатов парсинга в виде 3D сцены [1].

Методы стандартизации структуры HSML

Стандартизация структуры HSML (Hyper Scene Markup Language) является критической задачей для обеспечения функционирования модуля парсинга документа, а также для валидации такого документа, в составе дополненной реальности. Одним из ключевых инструментов для стандартизации XML-документов, к которым относится и HSML, является DTD (Document Type Definition).

DTD (Document Type Definition) — это стандарт, используемый для определения структуры XML-документа. Он описывает допустимые элементы и атрибуты, их взаимосвязи и порядок следования. DTD позволяет определить, какие элементы могут присутствовать в документе, как они могут быть вложены друг в друга и какие атрибуты могут использоваться с каждым элементом [5].

Ниже приведен пример формализованной структуры DTD для HSML-подобного языка.

```
<!ELEMENT trigger (content)>
<!ATTLIST trigger type (space | image | body | geo) #REQUIRED
                src CDATA #IMPLIED
                latitude CDATA #IMPLIED
                longitude CDATA #IMPLIED
                altitude CDATA #IMPLIED>

<!ELEMENT content (model | image | video | video-alpha)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT image (#PCDATA)>
<!ELEMENT video (#PCDATA)>
```

В таблице 1 представлена обосновательная характеристика предлагаемых структурных элементов такого языка.

Таблица 1 – Обосновательная характеристика предлагаемых структурных элементов HSML

Наименование тега	Наименование атрибута	Характеристика атрибута	Характеристика тега
<trigger>	type	Тип триггера. Принимает такие значения как: space – для отображения AR в режиме “в пространстве”; image – для отображения AR в режиме “по изображению”; body – для отображения AR в режиме “по частям тела”; geo – для отображения AR в режиме “по координатам”	Применяется для обозначения триггера, при срабатывании которого отображается контент в AR. Контент является дочерним компонентом такого тега.
	src	Ссылка на триггер	
	latitude	Широта, применяется для отображения AR в режиме “по координатам”.	
	longitude	Долгота, применяется для отображения AR в режиме “по координатам”.	
	altitude	Высота, применяется для отображения AR в режиме “по координатам”.	
<content>	-	-	Служебный тег. Применяется для обозначения блока, содержащего контент.
<model>	src	Ссылка на модель	Отображение контента типа 3D-модель при срабатывании триггера. Является дочерним компонентом тега <content>.
	scale	Масштаб отображаемой модели	
	marginX	Смещение по оси X	
	marginY	Смещение по оси Y	
	marginZ	Смещение по оси Z	
<image>	src	Ссылка на изображение	

	scale	Масштаб отображаемого изображения	Отображение контента типа изображение при срабатывании триггера. Является дочерним компонентом тега <content>.
	marginX	Смещение по оси X	
	marginY	Смещение по оси Y	
	marginZ	Смещение по оси Z	
<video>	src	Ссылка на видео	Отображение контента типа видео при срабатывании триггера. Является дочерним компонентом тега <content>.
	scale	Масштаб отображаемого видео	
	marginX	Смещение по оси X	
	marginY	Смещение по оси Y	
	marginZ	Смещение по оси Z	
	isAlpha	Флаг, есть ли в видео альфа-канал	

Механизм работы модуля парсера в составе движка браузера AR приведен на рисунке 1.

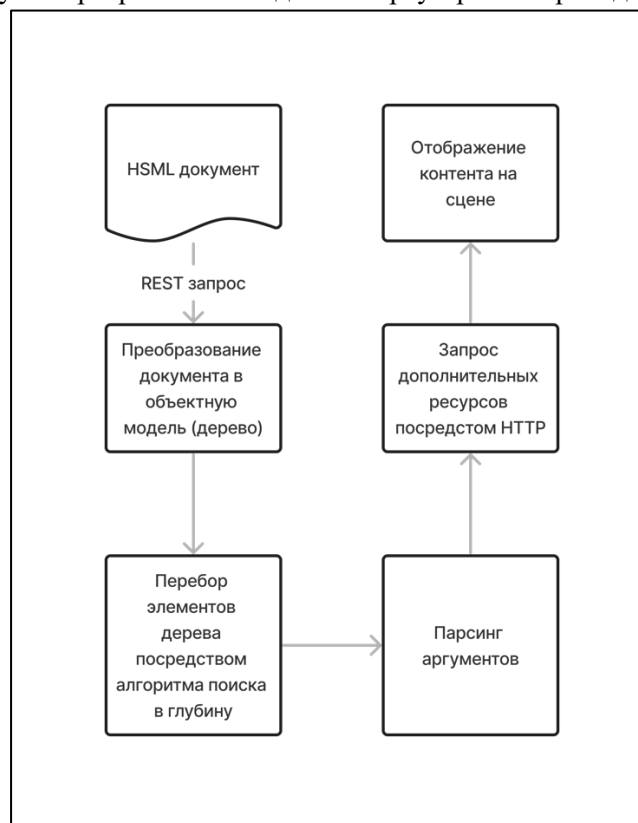


Рисунок 1 – Механизм работы модуля парсинга

Разработка модуля парсинга структуры HSML документа

Исходя из выведенного принципа работы модуля парсинга AR браузера, был разработан демонстрационный модуль, разбирающий структуру HSML документа и отображающий контент в 3D сцену three.js, совместимой с фреймворком отображения AR WebXR [3]. Ниже приведена версия такого модуля на языке JavaScript.

```

async function initWebXR() {

```

```
if (navigator.xr) {
  const supported = await navigator.xr.isSessionSupported('immersive-ar');
  if (supported) {
    const enterArButton = document.createElement('button');
    enterArButton.textContent = 'Enter AR';
    enterArButton.onclick = () => startArSession();
    document.body.appendChild(enterArButton);
  } else {
    console.log('AR not supported');
  }
} else {
  console.log('WebXR not available');
}

async function startArSession() {
  const session = await navigator.xr.requestSession('immersive-ar');
  session.addEventListener('end', onSessionEnd);

  const gl = document.createElement('canvas').getContext('webgl',
{xrCompatible: true});
  const xrLayer = new XRWebGLLayer(session, gl);

  session.updateRenderState({baseLayer: xrLayer});

  const refSpace = await session.requestReferenceSpace('local');

  const scene = new THREE.Scene();
  const camera = new THREE.PerspectiveCamera(70, window.innerWidth /
window.innerHeight, 0.01, 10);
  const renderer = new THREE.WebGLRenderer({ alpha: true,
preserveDrawingBuffer: true, canvas: gl.canvas });

  loadGLBModel(scene);

  session.requestAnimationFrame((time, frame) => onXRFrame(time, frame,
session, refSpace, renderer, scene, camera));
}

function onSessionEnd(event) {
  console.log('AR session ended');
}

function onXRFrame(time, frame, session, refSpace, renderer, scene, camera)
{
  session.requestAnimationFrame((time, frame) => onXRFrame(time, frame,
session, refSpace, renderer, scene, camera));
}
```

```
const pose = frame.getViewerPose(refSpace);
if (pose) {
  const glLayer = session.renderState.baseLayer;
  renderer.setSize(glLayer.framebufferWidth, glLayer.framebufferHeight,
false);
  renderer.setFramebuffer(glLayer.framebuffer);

  const view = pose.views[0];
  const viewport = glLayer.getViewport(view);
  renderer.setViewport(viewport.x, viewport.y, viewport.width,
viewport.height);

  camera.matrix.fromArray(view.transform.matrix);
  camera.projectionMatrix.fromArray(view.projectionMatrix);
  camera.updateMatrixWorld(true);

  renderer.render(scene, camera);
}
}

function loadGLBModel(scene) {
  const hsml = `
  <trigger type="image" path="path/to/model.glb">
    <content>
      <model>path/to/model.glb</model>
    </content>
  </trigger>
</hsml>`;

  const parser = new DOMParser();
  const xmlDoc = parser.parseFromString(hsml, 'text/xml');
  const modelPath = xmlDoc.querySelector('trigger[type="image"]').getAttribute('path');

  const loader = new THREE.GLTFLoader();
  loader.load(modelPath, function (gltf) {
    scene.add(gltf.scene);
  }, undefined, function (error) {
    console.error('An error occurred loading the model', error);
  });
}
window.addEventListener('DOMContentLoaded', initWebXR);
```

Вывод

Разработка браузера дополненной реальности, способного интерпретировать и отображать контент HSML-документов, демонстрирует значительный потенциал для расширения возможностей

AR-технологий. Основываясь на предложенной структуре и методах стандартизации HSML, можно обеспечить эффективный парсинг и отображение AR-контента.

Применение DTD для стандартизации структуры HSML позволяет четко определить допустимые элементы, атрибуты и их взаимосвязи, что значительно упрощает валидацию и парсинг документов. Это критически важно для обеспечения корректной работы парсера и последующего рендеринга AR-контента.

Предложенный демонстрационный модуль на языке JavaScript показывает, как можно реализовать парсинг и отображение AR-контента на основе HSML.

Настоящая работа является составной работой в рамках исследования и проектирования кроссплатформенного браузера AR.

Список литературы

1. Тахара Т. и др. Ретаргетируемая дополненная реальность: контекстно-зависимая дополненная реальность в помещениях на основе 3D-графика сцен //Международный симпозиум IEEE 2020 по смешанной и дополненной реальности (ISMAR-Adjunct). – IEEE, 2020. – С. 249-255.
2. Как работает веб-дополненная реальность? // Ссылка на Rock Paper Reality: <https://rockpaperreality.com/insights/web-ar/how-does-web-based-augmented-reality-work/> (дата публикации: 01.05.2024).
3. Чельтекин А. и др. Расширенная реальность в науках о пространстве: обзор проблем исследований и будущих направлений //Международный геоинформационный журнал ISPRS. – 2020. – Т. 9. – №. 7. – С. 439.
4. Исмаил А. А., Дарвиш С. М., Мохаллель А. А. Усовершенствованный алгоритм отслеживания объектов на основе языка разметки дополненной реальности (ARML) для медицинской инженерии // Передовые технологии и приложения машинного обучения: Материалы AMLTA 2021. – Издательство "Спрингер Интернэшнл Пабблишинг", 2021. – С. 249-259.
5. Как работает браузер // Web.dev URL: https://web.dev/articles/howbrowserswork#The_browser_high_level_structure/ (дата обращения: 01.12.2023).

References

1. Tahara T. et al. Retargetable AR: Context-aware augmented reality in indoor scenes based on 3D scene graph //2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). – IEEE, 2020. – С. 249-255.
 2. How Does Web-based Augmented Reality Work? // Rock Paper Reality URL: <https://rockpaperreality.com/insights/web-ar/how-does-web-based-augmented-reality-work/> (дата обращения: 01.05.2024).
 3. Çöltekin A. et al. Extended reality in spatial sciences: A review of research challenges and future directions //ISPRS International Journal of Geo-Information. – 2020. – Т. 9. – №. 7. – С. 439.
 4. Ismail A. A., Darwish S. M., Mohallel A. A. An Enhanced Object Tracking Algorithm Based on Augmented Reality Markup Language (ARML) for Medical Engineering //Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2021. – Springer International Publishing, 2021. – С. 249-259.
 5. How browser work // Web.dev URL: https://web.dev/articles/howbrowserswork#The_browser_high_level_structure/ (дата обращения: 01.12.2023).
-