

# Three JS와 React를 이용한 증강현실 웹사이트 구현

김선화\* · 문상호 · 이성진

부산외국어대학교

## Implementation of Augmented Reality Website Using Three.js and React

Seon-hwa Kim\* · Sang-Ho Moon · Sung-jin Lee

Busan University of Foreign Studies

E-mail : seonhwakei@gmail.com

### 요 약

연구개발특구진흥재단의 2021년 모바일 증강현실 시장 보고서에 따르면 스마트폰의 발전과 더불어 모바일 증강현실 시장의 성장으로 인해 최근 증강 현실 시장이 성장하고 있다. 증강현실 서비스를 모바일 사용자에게 제공하기 위해서는 기기별 네이티브 앱을 제작해야 한다. 하지만 이에는 멀티 플랫폼 대응에 소비되는 유지보수 비용과 앱 다운로드로 인한 낮은 접근성 등의 문제점이 존재한다. 최근에는 WebXR Device API를 활용하여 웹 환경에서 증강현실 시스템을 구축이 진행되는 추세이나, WebXR Device API를 사용한 시스템은 아직 연구단계에 있다고 판단된다.

본 논문에서는 WebXR Device API, Three JS, React를 이용하여 반응형 멀티 플랫폼 환경을 구축하여, 모바일 및 웹 사용자에게 증강현실 서비스를 제공하는 기능을 구현하였다. 실험 결과 반응형 웹 환경에서 증강현실의 연동을 성공적으로 구현되었음을 확인하였으며, 향후 웹 기반에서 동작되는 증강현실 서비스를 기대해본다.

### ABSTRACT

According to the 2021 Mobile Augmented Reality Market Report by Korea Innovation Foundation, the augmented reality market has been growing recently due to the growth of the mobile augmented reality market along with the development of smartphones. In order to provide augmented reality services to mobile users, it is necessary to create native apps for each device. However, there are problems such as maintenance costs for multi-platform and low accessibility due to app download. Recently, the construction of augmented reality system in a web-based environment by using the WebXR Device API is in progress, but it is judged that the system using the WebXR Device API is still in the research stage.

In this paper, a responsive multi-platform environment was built using the WebXR Device API, Three JS, and React, and a function to provide augmented reality service to mobile and web users was implemented. As a result of the experiment, it was confirmed that the interlocking of augmented reality was successfully implemented in a responsive web environment, and look forward to the augmented reality service operating on the web in the future.

### 키워드

Three JS, React, WebXR Device API, Augmented Reality, Multi Platform

### 1. 서론

최근 AR 기술은 다양한 쇼핑 기술과 결합되는 추세이다. 모바일 쇼핑 AR 기술은 소비자의 구매 의도에 긍정적 영향을 미치는 것으로 드러났다[1].

그러나 현재 AR 애플리케이션을 서비스하기 위해서는 ARCore, ARKit 등을 이용해 네이티브 앱을 제작해야 한다는 한계를 가진다. 이러한 배경을 바탕으로 본 논문에서는 갤러리 웹사이트에서 소비자들이 현실 공간의 벽에 그림을 걸어볼 수 있는 기능을 WebXR Device API를 사용해 구현한다.

---

\* corresponding author

## II. WebXR Device API

WebXR Device API(API: Application Programming Interface)는 웹 환경에서 확장현실을 서비스하기 위해 개발된 API로 출력 장치의 선택을 관리하며, 적절한 프레임 속도로 선택된 장치에 3D 장면을 렌더링(Rendering)하며, 입력제어기 사용을 통해 생성된 움직임 벡터(Vector)를 관리한다[2]. 현재도 개발 중인 API로 최신 버전은 2022년 8월에 발표되었다[3]. 이는 HTTPS 또는 localhost 환경에서만 실행할 수 있고 Chrome, Edge, Opera, Chrome Android, Samsung Internet, Opera Android 브라우저를 지원한다. 그중 약 65%를 차지하는 Chrome과 약 5%를 차지하는 Samsung Internet을 서비스 지원 환경으로 채택하였다[4]. 현재 iOS의 경우 WebXR Device API를 지원하지 않지만 추후 지원될 것을 고려하여 iOS 환경에서도 polyfill을 이용할 수 있는 Mozilla XRViewer를 기반으로 개발하였다.

Table 1-1. Experiment environments

	Android	iOS
Browser	Google Chrome Android 106.0.5249.65, Samsung Internet 18.0.4.14	Mozilla XRViewer
Polyfill / without Polyfill	Both	Polyfill
Debugging Tool	Chrome Inspect	
Device	Samsung galaxy s20+ 5G(SM-G986N)	Iphone SE(2nd generation)(MX9 R2KH/A)
Development environment	Localhost	

본 논문에서는 아래의 그림 1과 같이 WebXR 시스템을 디자인하였다. ArtPage는 그림 4에 출력된 페이지를 의미하고 XRButton과 XRCanvas는 React-xr 라이브러리에서 제공하는 WebXR 구현을 위해 제공하는 컴포넌트이다. 이외에도 React-xr 라이브러리에서 제공하는 useXR과 useHitTest 함수를 사용하였다.

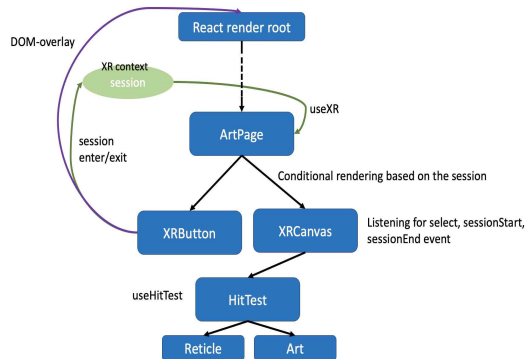


Fig. 1. Web-xr system design

## III. 증강현실과 React

클라이언트는 React 라이브러리를 사용해 모바일 접근성을 고려하여 그림 2와 같이 반응형 UI로 디자인하였으며 UI 및 WebXR 관련 요소들의 재사용성을 높이기 위해 컴포넌트화 하였다. WebGL 렌더링 라이브러리로 three.js를 사용하였고 이를 리액트 상에서 간편하게 구현하기 위해 react-three-fiber 라이브러리를 이용하였다. WebXR Device API는 React-xr 라이브러리를 통해 조작하였다.

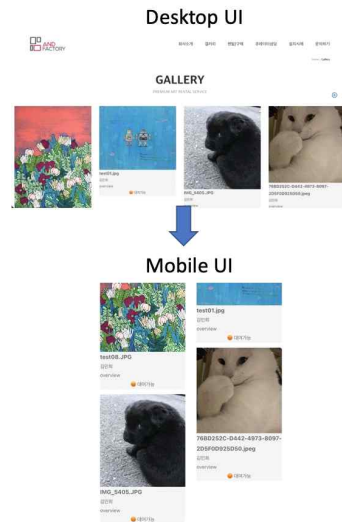


Fig. 2. Website UI

클라이언트와 서버는 RESTful API를 활용하여 통신한다. API를 통해 데이터베이스에서 작품 생성, 읽기, 삭제 기능을 수행하도록 구현하였다. 작품 생성 api는 호출시 함께 전달된 이미지의 url은 데이터베이스에 저장하고 저장된 이미지의 url은 데이터베이스에 입력한다. 이미지를 가져오기 위해선 작품 불러오기 api를 호출해 데이터베이스에서 가져온 이미지 url을 이용해 서버 파일시스템에 접근한다. 등록된 이미지를 삭제하기 위해선 삭제 이미지 url과 함께 삭제 api를 호출하면 된다.

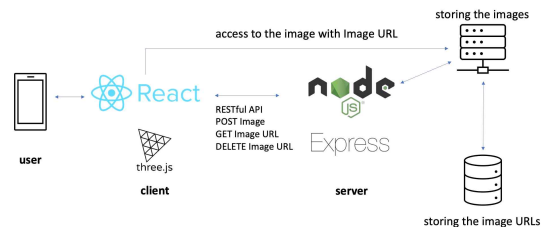


Fig. 3. System architecture

#### IV. 실험 결과 및 구현

그림 4와 같이 각 작품의 페이지(그림 1의 ArtPage) 내에서 React-xr 라이브러리에서 제공하는 XR Button 컴포넌트를 출력해 클릭을 통해 XR 세션에 진입할 수 있게 한다. React-xr 내부적으로 React-contextAPI를 사용해 구현되었기 때문에 별도로 XR-session을 관리하지 않아도 되고 추후 useXR custom-hook을 통해 XR-session 값에 접근할 수 있다.

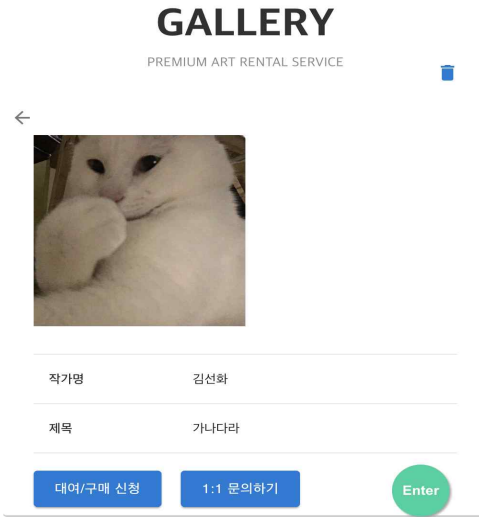


Fig. 4. ArtPage component rendering result

XR 세션이 존재할 경우 React-xr 라이브러리에서 제공하는 XRCanvas를 출력한다. 해당 세션의 종료 역시 XRButton 컴포넌트에서 관리된다. 세션을 종료하기 위해서는 그림 5와 같이 XRCanvas 위에도 XRButton 컴포넌트가 출력되어야 하는데 이는 DOM-overlay 기능을 이용해 구현한다. XRButton 초기화시 DOM-overlay 기능 설정과 함께 XR 세션 위에 중첩할 React root DOM 노드(이하 root 노드)를 등록한다. root 노드는 React 엘리먼트들이 렌더링되는 최상위 노드이기도 하므로 해당 노드를 DOM-overlay로 등록할 경우 하위의 모든 컴포넌트들이 화면에 노출된다. 그러나 XRButton만 출력되어야 하므로 sessionStart 이벤트가 발생하면 root 노드의 visibility를 visible로, XRButton 컴포넌트의 visibility를 invisible로 설정하며 sessionEnd가 발생하면 이와 반대로 설정하여 노출되어야 하는 컴포넌트를 조절한다.

React-xr 라이브러리에서 제공하는 useHitTest 함수를 사용해 Hit-test를 구현한다. Hit-test가 성공했을 경우 레티클의 matrix에 Hit-test 결과를 등록함으로써 그림 5의 a)와 같이 출력한다. 사용자의 화면 클릭에 의해 XRSessionEvent 중 하나인 select 이벤트가 발생되면 그림 5의 b)와 같이 레티클 위치에 작품을 출력한다. 레티클과 작품 모두 벽과 평행하게 출력되어야 하므로 X축을 기준으로 90도

회전시킨다. 작품 geometry의 texture를 현재 선택된 작품에 따라 동적으로 변경하기 위해서는 서버와의 RESTful API 통신을 통해 받아온 이미지 URL을 출력한다.

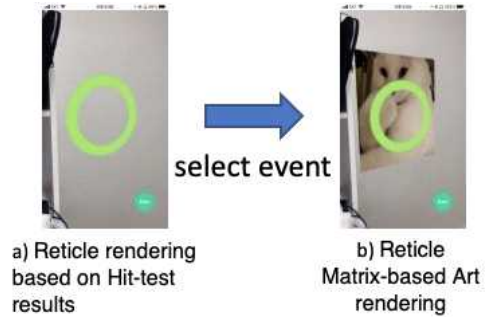


Fig. 5. Reticle and Art rendering

#### V. 결론

본 논문은 WebXR Device API, Three JS, React를 이용하여 반응형 멀티 플랫폼 환경을 구축하여, 모바일 및 웹 사용자에게 증강현실 서비스를 제공하는 기능을 구현하였다. 실험 결과 반응형 웹 환경에서 증강현실의 연동을 성공적으로 구현되었음을 확인할 수 있다. 본 연구의 성과는 비록 안드로이드 기반 환경에 국한되지만 WebXR Device API를 실제 서비스에 활용될 수 있는 기능을 구현했다는 의미를 가진다. 추후 iOS의 WebXR 지원이 확장됨에 따라 WebXR Device API를 활용한 멀티 플랫폼 서비스의 발전을 기대해본다. 향후 과제로는 AR과 VR를 결합한 XR 구현을 통해 메타버스 시스템 구현에 도전할 계획이다.

#### References

- [1] 이사정, 경성림, “모바일 쇼핑 AR기술의 편리성이 구매의도에 미치는 영향에 관한 연구 : 구매만족도의 조절효과를 중심으로”, 한국디지털정책학회 19, no.5 (2021): 45.
- [2] 이대현, “모바일 웹의 증강현실 환경에서 WebXR Device API 성능평가에 관한 연구”, (석사학위, 광운대학교 스마트융합대학원, 2021), 14.
- [3] W3C, “Latest published version”, [Internet]. Available : <https://www.w3.org/TR/webxr/#intro>
- [4] StateCounter, “Mobile Browser Market Share Worldwide”, [Internet]. Available : <https://gs.statcounter.com/browser-market-share/mobile/worldwide>