

유니티 기반 건물 환경 시각화 시스템 구축을 위한 연출 전략

서현경, 권동우, 지영민

한국전자기술연구원

{hkyung99, dwkwon, ym.ji}@keti.re.kr

Visual Strategies for Developing a Unity-Based Building Environment Visualization System

Hyeonkyeong Seo, Dongwoo Kwon and Youngmin Ji

Korea Electronics Technology Institute (KETI)

요약

본 논문은 유니티 환경에서 건물 환경 시각화 시스템의 에너지 정보를 효과적으로 전달하기 위한 연출 전략을 제시한다. 제안된 전략은 대시보드, 카메라 연출, 웹 배포 환경의 3가지 요소로 구성되며, 각 요소는 다양한 프로젝트의 시각적 연출 측면에서 범용적으로 활용될 수 있다. 각각 다양한 데이터 시나리오 및 구조에 유연하게 대응하는 정보 전달 방식으로 가독성, 퍼포먼스, 접근성을 향상시키는 데에 기여할 수 있다. 이 3가지 요소에 대한 실효성을 예시를 통해 검토한다.

I. 서론

최근 스마트 팩토리 및 빌딩 기술이 빠르게 발전하면서 효율적인 에너지 사용을 위한 실시간 데이터 모니터링에 대한 수요가 증가하고 있다. 이에 따라 복잡한 데이터 측정값을 직관적으로 관제할 수 있는 다양한 시각화 시스템 적용 사례가 생기고 있다.[1] 기존의 에너지 시각화 시스템은 대부분 정적인 수치로 구성되어 있으나, 효과적인 시각화를 위해 쉽고 빠르게 이해할 수 있는 시각적 전략이 필요하다. 이러한 흐름에 따라 Unity는 실시간 렌더링 성능과 사용자와 상호작용이 가능한 시스템을 바탕으로 실시간 시뮬레이션, 공간 정보 연동 등 다양한 적용 사례를 보여주고 있다.

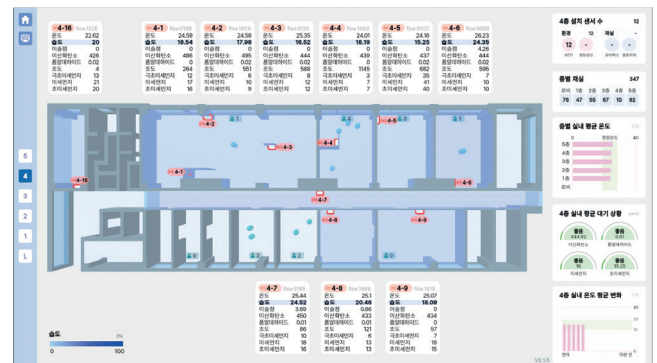
본 논문에서는 Unity 기반의 시각화 시스템 구축에 있어 직관적인 정보 전달, 사용자 몰입도, 다양한 플랫폼 접근성을 갖춘 연출 전략 세 가지를 제안한다. 우선 대시보드 구성 전략을 통해 에너지 데이터를 구조화하여 사용자가 핵심 정보를 빠르게 파악할 수 있도록 한다. 그리고 카메라 연출 기법을 활용해 사용자의 시선에서 공간 내 맥락을 자연스럽게 탐색할 수 있도록 한다. 마지막으로, WebGL 환경에서 배포함으로써 웹에서의 호환성과 접근성을 확보하고 활용 범위를 확대할 수 있다. 이 세 가지 전략은 각각 시각화 시스템의 전달력, 몰입감, 확장성을 발전시키는 핵심 요소로 작용할 것으로 기대된다.

II. 본론

Unity 기반 건물 에너지 시각화 시스템 구축을 위한 핵심 전략 3가지는 시스템의 가독성, 퍼포먼스, 접근성을 향상시키기 위한 요소로, 실제 적용 예시를 통해 그 유효성을 검토한다.

첫 번째는 대시보드 구성 전략이다. 모든 데이터를 동일한 위계로 나열하는 방식은 정보의 중요도를 반영하지 못해 사용자가 필요한 데이터를 빠르게 파악하기 어려워 탐색의 효율성이 저해된다. 따라서 관제하고자 하는 범위를 기준으로 구성해야 한다는 것이다. 에너지 사용 및 흐름 모니터링 예시의 (그림 1)은 4층을 관제하기 때문에 4층 전체를 알 수 있는 정

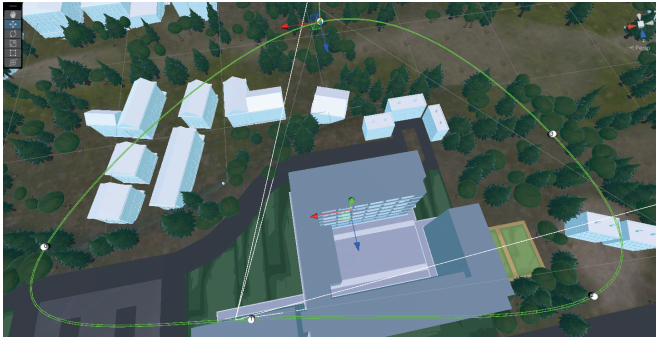
보를 제공한다. 건물 전체를 관제하는 화면이라면, 전체를 통합적으로 관제할 수 있는 정보로 확장되어야 한다.



(그림 1) 대시보드 예시

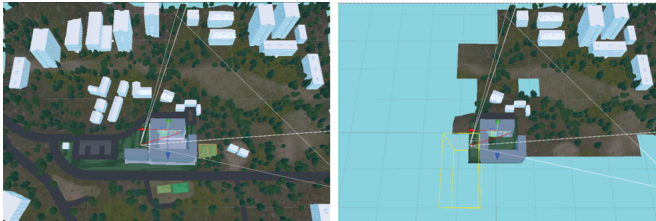
해당 (그림 1) 대시보드의 세부적인 내용은 다음과 같다. 우선 기기는 특성에 맞게 그룹화하고, 전체 기기 수, 그룹별 총 수량, 세부 항목별 수치를 함께 기재했다. 그리고 사용자가 각 그룹 내 관계를 쉽게 인식할 수 있도록 집합 형태로 구성했다. 층별 채실 인원과 같이 구조가 단순하고 수치 비교가 명확한 항목은 일반적인 표의 형태로 제시했다. 이때에도 전체 채실 인원을 기재하여 전반적인 규모를 파악할 수 있도록 했다. 평균 온도와 같이 수치의 높낮이가 시각적으로 드러나야 이해하기 쉬운 항목에는 그래프를 사용했다. 층별 실내 평균 온도는 각 층의 비교가 핵심이므로 층 이름이 왼쪽에 위치하는 가로 그래프를, 시간에 따른 전체 평균 온도 변화는 수치의 흐름을 강조하기 위해 세로 그래프를 채택했다. 실내 평균 대기 상황은 미세먼지 지표처럼 사용자가 익숙한 게이지 차트를 사용했다.

두 번째는 카메라 연출 기법이다. Unity는 Scene에서 구성된 화면을 Camera로 렌더링 하며 Cinemachine을 사용하면 영화 같은 카메라 연출을 코드나 키 프레임 없이 쉽고 부드럽게 제어할 수 있다. Virtual Camera를 설정하면 Main Camera가 따라가는 원리로, 성능 면에서도 Main Camera를 여러 개 사용하는 것보다 효율적이다. (그림 2)에서의 예시는 Cinemachine의 여러 기능 중 Dolly Camera를 사용했다.



(그림 2) Dolly Track 예시

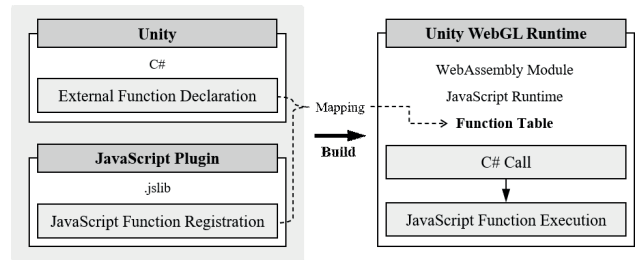
Dolly Track이란 일종의 카메라 이동 경로로, (그림 2)의 초록색 트랙처럼 Scene의 Waypoint 배열로 Virtual Camera를 배치하거나 고정 코스를 지정할 수 있는 트랙을 말한다.[2] Waypoint는 Dolly Track 위에 위치하는 개별 경유 지점을 의미하며, Camera가 해당 지점을 이동하면서 자동으로 방향을 회전하거나 속도를 조절할 수 있다. 일반적으로 이러한 방식은 반복적인 관제 경로 이동을 설정하게 되는데, 장시간 실행 시 실시간 렌더링의 부하를 높이고 GPU 사용량 증가로 이어질 수 있다. 이를 보완하고자 아래 (그림3)은 Occlusion Culling을 적용한 예시를 제시한다.



(그림 3) Occlusion Culling 적용 예시

Occlusion Culling은 카메라에 보이지 않는 오브젝트는 렌더링 하지 않도록 자동으로 제외하는 기능이다. (그림 3)에서 보이는 것처럼 카메라 시점에서 다른 오브젝트에 가려지는 부분은 렌더링 하지 않는다. 이를 설정하기 위해, 적용하려는 범위를 Occlusion Area로 설정하고, 다른 오브젝트를 가릴 가능성이 높은 오브젝트를 Occluder로, 가려지기 쉬운 오브젝트는 Occludee로 설정한다. 카메라에 보이지 않는 오브젝트를 미리 계산하는 Bake 과정을 거치면 (그림 3)과 같이 적용된다. 본 연구처럼 대부분의 오브젝트가 정적으로 구성된 상황에서는 GPU 렌더링 부하가 감소함으로써 실시간 렌더링 퍼포먼스를 안정적으로 유지하는 데에 기여할 수 있다. 그러나 동적 오브젝트가 포함된 경우 Occlusion Culling의 효과는 제한적이며, 구조가 복잡하거나 실시간 변형이 많은 경우 효과가 감소할 수 있음에 유의해야 한다. 이와 같은 연출 전략은 불필요한 렌더링을 줄이는 동시에 사용자의 시선을 유도하고 맥락을 부여함으로써 정보에 대한 몰입도를 높이는 데에 효과적이다.

세 번째는 WebGL 배포 전략이다. 웹 기반의 3D 서비스를 사용하게 되면 상대적으로 가용 메모리가 적어 최적화가 필수적이지만, 설치 없이 웹 브라우저로 실행이 가능하기 때문에 접근이 용이해져 웹 기반의 3D 서비스 시장은 더욱 확대될 것으로 보인다.[3] 따라서 웹 브라우저에서 고성능 그래픽을 실시간 렌더링 가능한 WebGL 빌드 전략을 제안한다. 이 과정에서 HTML5 및 JavaScript 기반의 파일을 생성하며 웹페이지의 틀을 구성하는 시스템인 WebGL 템플릿을 직접 설정할 수 있다. 즉 Unity 콘텐츠가 실행되는 웹페이지의 구조를 사용자가 직접 정의할 수 있다. 이는 JavaScript Plugin Library와의 통합 및 기능 확장에도 유리한 구조를 제공한다. JavaScript Plugin Library란 Unity의 WebAssembly 기반 C# 코드와 브라우저의 JavaScript를 연결하는 라이브러리이다. Unity WebGL 환경에서 외부 JavaScript를 호출하기 위해서 다음과 같은 과정을 거친다.



(그림 4) Unity WebGL 환경에서의 외부 JavaScript 호출 흐름

Unity C#에서 extern 키워드로 외부 함수를 선언하고, 해당 함수가 Unity WebGL 런타임 내부에 존재함을 명시한다. JavaScript에서는 jslib 확장자를 가진 플러그인을 작성하고 대응되는 JavaScript 함수를 등록한다. 이 선언 및 등록은 Unity WebGL 빌드 과정에서 C# 선언과 JavaScript 정의를 연결하는 Function Table에 매핑된다. 따라서 런타임 중 C# 코드에서 외부 함수 호출이 발생하면, Function Table을 참조하여 대응되는 JavaScript 함수가 실행된다. 즉, C#은 호출 준비를 하고, jslib는 실행할 함수를 정의하며, WebGL 런타임은 둘을 매핑해 실행하는 구조를 가진다. 이와 같은 JavaScript Plugin 구조를 통해 프로젝트의 브라우저 기반 개발에 있어 유연성과 확장성을 향상시킬 수 있다.

III. 결론

본 논문에서는 Unity의 시각적 연출 기능을 기반으로 건물 에너지 정보를 효과적으로 전달하기 위한 3가지 전략을 제시하였다. 대시보드의 내용은 강조하고 싶은 특성이 잘 드러나도록 계층적 배치 방식으로 구조화했다. 그리고 카메라 연출 기법을 통해 관제 흐름을 유도해 몰입감을 주었다. 이때 Occlusion Culling으로 필요한 부분만 렌더링 할 수 있도록 최적화했다. 마지막으로 시스템 활용 범위를 극대화하도록 WebGL 기반의 배포 환경을 제안했다. 이는 플랫폼 간 호환성을 확보하는 것과 동시에 JavaScript Plugin Library를 활용한 기능 확장성도 강화할 수 있다.

향후 관련 연구에서는 Unity6 WebGL 개선 사항을 반영하여 보다 높은 웹 브라우저 퍼포먼스와 호환성을 갖추는 시각화 시스템을 구축할 수 있을 것이다. 또한 실시간 인터랙션 기능을 함께 적용해 사용자 몰입도와 정보 이해도를 더욱 향상될 것으로 기대한다.

ACKNOWLEDGMENT

본 연구는 산업통상자원부(MOTIE)와 한국에너지기술연구원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (RS-2021-KP002461)

참 고 문 헌

- [1] Minji Kim and Jihyun Lee, "A Study on the Development of Data Visualization Design for Smart Factory Monitoring System," Proceeding of HCI Korea 2023, pp. 671-677, Gangwon, Feb 2023.
- [2] Sung Suk Park, "Scene Production using Unity Cinemachine," Journal of Information Technology Applications & Management, vol. 28, no. 6, pp. 133-143, 2021.
- [3] Jeong Eun Young, Yoo Han Kyu, Kim Byung Chul, and Suh Jung Min, "Comparing Productivity and Performance between Naive-WebGL Development Tools and Unity-WebGL," Proceedings of Symposium of the Korean Institute of communications and Information Sciences, pp. 928-929, Gangwon, Feb 2023.