

仮想空間での発表資料展示会の提案

斎藤 翼[†] 広瀬 雄二[†]

東北公益文科大学[†]

1. はじめに

COVID-19の影響により、オンラインでできる会議や娯楽の需要が増加した。そのため、仮想空間上で発表資料の展示会を行うことで、インターネットを接続しさえすれば発表資料が見られることに需要があるのではないかと考えた。本研究では、JavaScriptのライブラリである Three.js を用いて発表資料の展示会を行い、そのシステムが様々な端末から操作ができる環境を目指し、システムを構築していく。

2. 先行研究

澤崎敏文は、メタバース（三次元仮想空間）を授業や学生活動の一部に活用することで、その可能性について検討を行った。それぞれの活用の終了後にアンケート調査・インタビュー調査を行った結果、概ね好意的な回答が多く、メタバース活用の可能性を感じたと示している。メタバース内では、仮想的な距離や方向によって音声の聞こえ方なども変化し、自由に空間を歩き来してコミュニケーションを取ることができるという点を考慮すると、ポスター発表のような場面での活用にも親和性が高いと考える [1]。

3. システムの提案

先行研究の中の通信環境に関する意見として、「発表を含めるとPCのスペックが足りない」、「パソコンのスペックが足りないため、発表などが非常にやりにくかった」、「資料が見にくい、ネット環境やデバイスによって差が出る」があった。

この研究を踏まえて、本システムではより多くの端末で操作できるように軽量のシステム的设计する。またその場で発表するのではなく発表資料を見ることに重点を置いたシステムを提案する。

4. 実装

4.1 開発環境

- 開発言語
 - HTML
 - CSS
 - JavaScript
 - * Three.js - 0.126.1
- ソフトウェア
 - Blender - 3.0.1

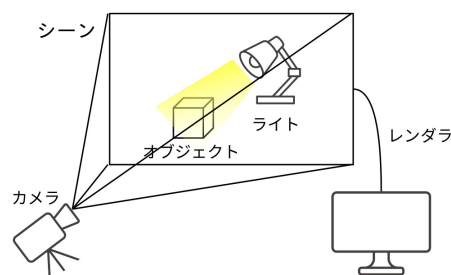


図 1 Three.js のシステム概念図

- ブラウザ
 - Mozilla Firefox - 119.0.1

4.2 要件定義

関連研究における課題を踏まえた上で、本システムにおける前提条件と機能要件を以下に示す。

- 前提条件
 - 単一接続
- 機能要件
 - 仮想空間上での PDF 形式資料表示
 - 仮想空間上での MP4 形式資料表示
 - MP4 形式資料の再生と停止
 - カメラの移動と方向転換

前提条件としたのは、複数接続だとサーバに負荷がかかりシステムがうまく動かなくなることが示唆されるためである。

機能要件として、本システムでは発表資料を閲覧するにあたり、PDF 形式・MP4 形式の資料を表示する。しかし、複数の MP4 形式の動画を再生し続けてはサーバに負荷がかかることになるため、MP4 形式資料の再生と停止を機能要件とする。

4.3 Three.js

Three.js は JavaScript ライブラリである。Three.js の仕組みとして、シーンにカメラ・ライト・3D オブジェクトを追加する。そのシーンをレンダラ通してディスプレイに出力する (図.1)。

4.3.1 ビデオの表示と再生・停止

ビデオを再生する部分だけ設計し、音を再生してしまうと Firefox のポリシーに接触してしまい、うまく動作しなくなる。そのため、ビデオからの音声は

Proposal for exhibition of presentation materials in a virtual space

[†] Tohoku University of Community Service and Science

表 1 Object.csv

dataPath	loc.X	loc.Y	loc.Z
model/cube.glb	15	0	0
model/ball.glb	25	0	0

無音にし、ビデオ再生と音の再生を別々に設計する必要がある。

ビデオの再生は HTML の video 要素で設計しており、音声の無音は muted 属性によって実現している。

本システムでは、ビデオが再生されている特定のオブジェクトをクリックすることで再生・停止の切り替えができるように設計している。

特定オブジェクトの判定は「Vector2」・「Raycaster」クラスインスタンスを用いて設計している。「Vector2」クラスインスタンスで、x 座標と y 座標を 0 に設定する。つまり、ディスプレイ中央に設定する。「Raycaster」クラスインスタンスで画面中央から光線を飛ばし、1 番最初に光線に触れたオブジェクトを判定している。これはカメラの視点を変更する度に光線も追従する。

動画の再生・停止の切り替えは flag を用いて行う。ビデオとオーディオが設定されたオブジェクトが一度クリックすると再生を実行し、もう一度クリックすると停止を実行する。

4.3.2 CSV 読み込みによる 3D オブジェクトの Web への反映

「GLTFLoader」クラスインスタンスで 3D オブジェクトの Web への反映はできるが、その都度無駄にコードを書かなくてはならなくなってしまい冗長である。そのため、CSV から値を読み込む手法を取る。CSV 読み込みは、Fetch API を用いて行う。

4.3.3 資料をオブジェクトに貼付

本システムでは PDF 資料を表示することを機能要件としている。Three.js ではオブジェクトの形を決める「BoxGeometry」クラスインスタンスとオブジェクトをどのように描画するかを決める「Mesh-BasicMaterial」クラスインスタンスでオブジェクトが設計される。「BoxGeometry」クラスインスタンスで縦横比 $1:\sqrt{2}$ のオブジェクトを作成する。そこに「TextureLoader」クラスインスタンスで PNG 形式のテクスチャデータを読み込んだものを「Mesh-BasicMaterial」に取り込む。これら 2 つで PDF 資料の展示を実現した (図.2)。

4.4 Blender での 3D モデル作成

3D モデルは Blender を用いて作成し、glTF(Graphics Language Transmission Format) でエクスポートしたものを使用する。glTF の利点と



図 2 PDF 資料の貼付

して、

- (1) web ブラウザ、モバイルデバイスなどのプラットフォームで 3D シーンとモデルを効率的に実行して送信及びロードができる。
- (2) 3D アセットを解凍・処理するための実行時処理を最小限に抑え、高度な広帯域グラフィックス機能を実現する。

の 2 点が挙げられる。これにより、web ブラウザ上での軽量かつ高度なグラフィックスが実現される。

4.4.1 3D モデルの光源

Three.js ではライトを設定することで、オブジェクトを照らしている。しかし、光の当たらないところは影になる。本システムでは四角柱のオブジェクトを設置し、そのオブジェクトの四面から資料を見られるように設計する。そのため、四面が影にならないようにする必要がある。

どの方向から見ても影ができないように設計するのは、Three.js でライトを多数設定することで解決するが手間がかかる。そのため Blender でエクスポートする際に「Punctual Lights」をオンにしてエクスポートすることで Blender で設定したライトをオブジェクトのデータに組み込まれるようになる。

5. まとめ

通常、発表資料を見る度にファイルを指定して開かなければならない。本システムでは一つの WEB サイトで完結して PDF 発表資料が展示されているため、何度もファイルを開き直す必要がない。また、プレゼンテーション発表を行った MP4 形式の資料も展示されているため、いつでも発表を閲覧することができるという利点がある。

今後の展望として、利用者が発表資料のアップロードを行うことで展示会場に自動追加されるように設計する。

参考文献

- [1] 澤崎敏文. メタバースを活用した多様な学習環境の構築と実践. 日本教育工学会研究報告書. 2023, vol.2023, no.2, p.83-87.