

JavaScriptベースのゲームエンジンを用いたプロジェクト型VR用360度画像表示機能の開発

| | |
|-------|---|
| メタデータ | 言語: Japanese 出版者: 武蔵野大学アジアAI研究所 公開日: 2024-03-29 キーワード (Ja): キーワード (En): 作成者: 圓崎, 祐貴 メールアドレス: 所属: |
| URL | https://mu.repo.nii.ac.jp/records/2000270 |

[論文]

JavaScript ベースのゲームエンジンを用いたプロジェクション型

VR 用 360 度画像表示機能の開発

Development of 360 Degree Image Viewer Function for the Projection VR System with the Game Engine Based on JavaScript

圓崎 祐貴

概要

近年 VR 技術普及の進展に伴って 360 度カメラを使って比較的簡単に没入型コンテンツを 360 度画像コンテンツが広がりを見せている。それらを体験するための HMD ベースのハードウェアとそれらで 360 度画像コンテンツを再生するためのソフトウェアは充実していてウェブアプリで利用可能なものがある一方でプロジェクション型 VR とそれらで 360 度画像コンテンツを再生するためのソフトウェアはハードウェアの特殊性やそれらを利用するための開発キットに依存するためそれらの知識がないと開発するのは困難である。そこで本稿では JavaScript ベースのゲームエンジンである PLAYCANVAS に着目し、これを用いた 360 度画像コンテンツ表示機能を開発して利用可能とすることを目標として開発を行なった。

1. はじめに

近年 MetaQuest[1]や HTC VIVE[2]などの HMD に代表されるように VR 技術普及の進展に伴って様々な VR コンテンツが増えてきており様々な分野で VR の活用が増えつつある。HMD を使った VR の開発環境も Unity[3]や UnrealEngine[4]など複数のゲームエンジンでも統一的に利用できるように OpenXR[5]などのようにある程度共通化が進んでいるため、VR コンテンツを開発しての活用が行ないやすくなっている。なかでも THETA[6]や GoProMax[7], insta360[8]などに代表される 360 度カメラの普及により手軽に 360 度画像を撮影可能になったことで VR コンテンツの専門的な知識が無くても手軽に活用できるようになってきた。

HMD による 360 度画像は体験者が一度に見える視野角が狭く、1 人ずつ装着して体験するため複数人で共有しながら体験するのに向いていないという問題がある。しかし体験者の周囲をスクリーンに映った映像で覆うという LargeSpace[9]などのようなプロジェクション型 VR による 360 度画像の体験の場合は一度に見える視野角が広く、プロジェクション型 VR の内部の大きさによるものの複数人で共有して体験することも出来るため HMD では難しい用途への応用、例えば現場の 360 度没入映像を複数人で共有しながらの会議を行なうなどの応用が期待されている。

しかしながらプロジェクション型 VR は体験車の周囲を映像で覆うという概念は共通なもの

のその方法は多種多様であるためHMDのように有る程度の構造の共通化などが行なえないため、CAVE とその派生システムに関する分野が UniCAVE[10]と呼ばれる開発環境で一部共通化されている程度であり、あまり開発環境の整備はあまり進んでいない。特に開発にあたってOSに依存するような機能を利用しなければならないことも多いため、あるプロジェクション型VRのソフトウェアを別のプロジェクション型VRのソフトウェアに流用することはシステム構成が似通っているもの以外は困難を極める。

そこでウェブブラウザ上で3Dゲームを開発することが可能なJavaScriptベースのゲームエンジンであるPLAYCANVAS(図1)[11]を用いてプロジェクション型VRのソフトウェアを開発することが出来ればウェブブラウザ上で動作するウェブアプリにはOSに依存性がないため移植性が高く、また将来的にJavaScriptのビデオチャット用の開発ライブラリであるWebRTCの機能を使った柔軟な映像伝送機能を利用できる可能性がある。そのため本稿ではプロジェクション型VRのソフトウェアの試作として360度画像コンテンツ表示機能に焦点を当てて開発を行なうことを目的とする。

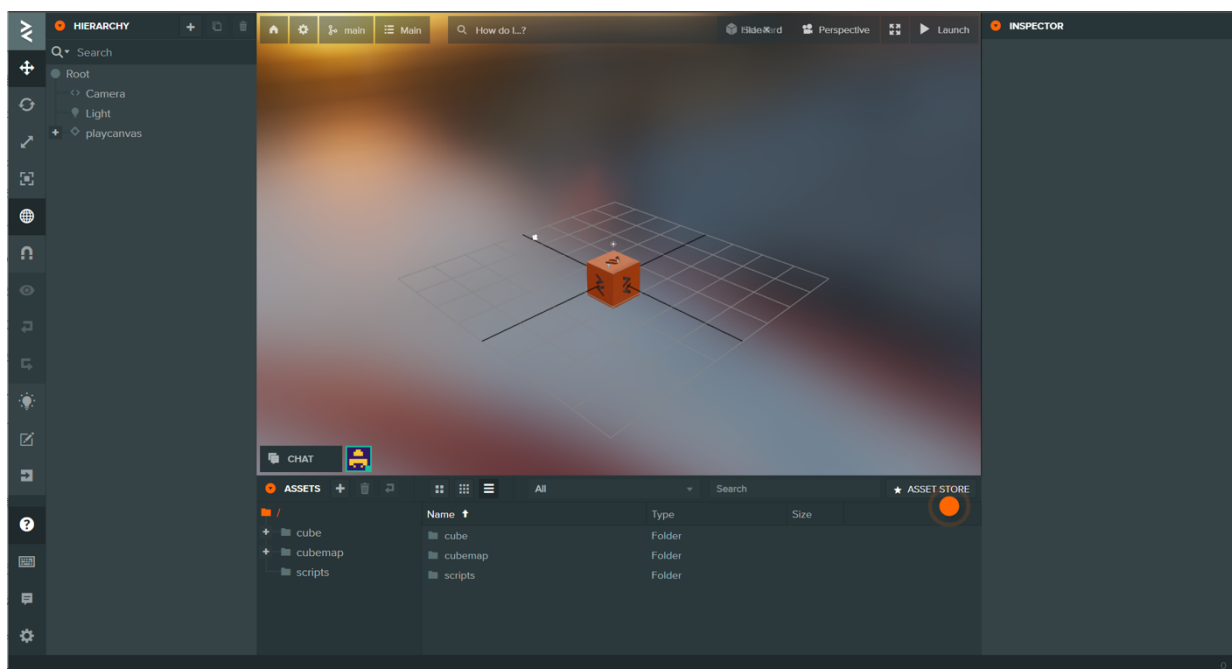


図2 PLAYCANVAS

2. 開発環境

JavaScriptベースのゲームエンジンであるPLAYCANVASを用いる。近年のウェブブラウザ上で動作するウェブアプリはHTMLとJavaScriptを用いたものが主流となっていて、PLAYCANVASはそれらに加えてウェブブラウザ上での3DCG描画機能であるWebGLを活用してUnityのようなゲーム開発ができるプラットフォームである。ウェブブラウザ上で開発環境も動作するため開発にあたって特別なソフトウェアの導入が不要である。UnityのようなGUIを持つためGUIの操作で3D空間上に3Dデータを配置する機能などを持ち、プログラムのコーディングができな

い開発者であっても 3D データのビューアーなど簡単なアプリ開発が行なえるというメリットがあるため PLAYCANVAS を採用した。

ソフトウェアの動作を検証するプロジェクション型 VR にはサイバーフィジカル美術館(図 2) [12] で用いたシステムを利用する。このシステムは平面を組み合わせた構造のため映像補正がシンプルな方法で済む。そのため試作に適している。



図 2 サイバーフィジカル美術館

3. システム構成

360 度映像コンテンツ表示機能を開発するためには、360 度映像をソフトウェア上で正しく展開する機能と体験者からスクリーンに映った画像が正しく見えるようにするために映像を補正する機能が必要である。

まず 360 度映像をソフトウェア上で正しく展開する機能について説明する。この機能を実現するにはまず 360 度映像を読み込むことが必要である。現在普及している 360 度カメラの画像形式について画像のフォーマットは通常のデジタルカメラと同じく jpg などが主流であるが座標系が異なる。Equirectangular 形式(図 3) [13] と呼ばれる座標系となっており画像の横方向が方位角、縦方向が仰俯角となっている。そのためそのまま展開すると仰俯角が 0 度の付近の

映像は実際の見え方と変わらないものの仰俯角が大きくなるにつれて映像の見え方が実際の見え方と異なってくる。

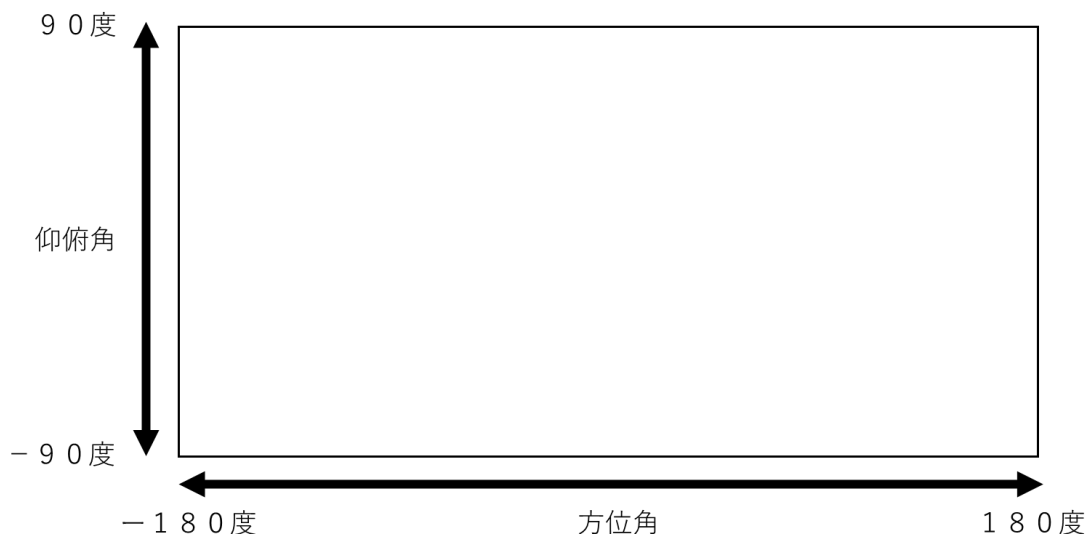


図3 Equirectangular 形式

そのため 3D 空間上に球状のモデルデータを配置してそのテクスチャとして方位角，仰俯角を基に適切に貼り付けることが必要である。Equirectangular 形式はちょうど地球全体を表した世界地図のうちのメルカトル図法と呼ばれるものに近く，テクスチャを貼る原理は世界地図を地球儀に貼る操作とほぼ同じ原理である。こうして 360 度画像のテクスチャを貼られた球状のモデルデータを中心から見ることでどの方向の画像も正しい見え方となる。

次に体験者からスクリーンに映った映像が正しく見えるようにするために映像を補正する機能について説明する。通常のディスプレイでの表示などの一般的な 3DCG の表示は透視投影変換を行なって 3D データの空間をディスプレイ平面上に射影変換して表示されている。しかしサイバーフィジカル美術館は複数のプロジェクターを用いた複数の平面スクリーンで構成されているため，構成している平面の数の分だけ違うパラメータの透視投影変換が必要である。厳密には体験者の位置に応じて適切に透視投影変換を行なう必要があるが本稿では簡略化のためシステムの中央に体験者の位置に視点が固定されているものとして実装する。またプロジェクターとスクリーンの配置の関係が物理的な制約により一対一で対応せず，1 つのスクリーンの四隅を結んだ四角形がコンピュータからは長方形とならず台形のような形となるため透視投影変換のあとさらに図4のようにホモグラフィ変換によりスクリーン上で正しい見え方になるように補正を行なう必要がある。そのためプロジェクターの数の分だけそれぞれのスクリーンの配置に対応した透視投影変換を施して一旦テクスチャをレンダリング先として描画を行ない，レンダリングされたテクスチャを基にホモグラフィ変換を行なってプロジェクターに出力することで体験者が画像を正しい見え方で見る事が可能となる。

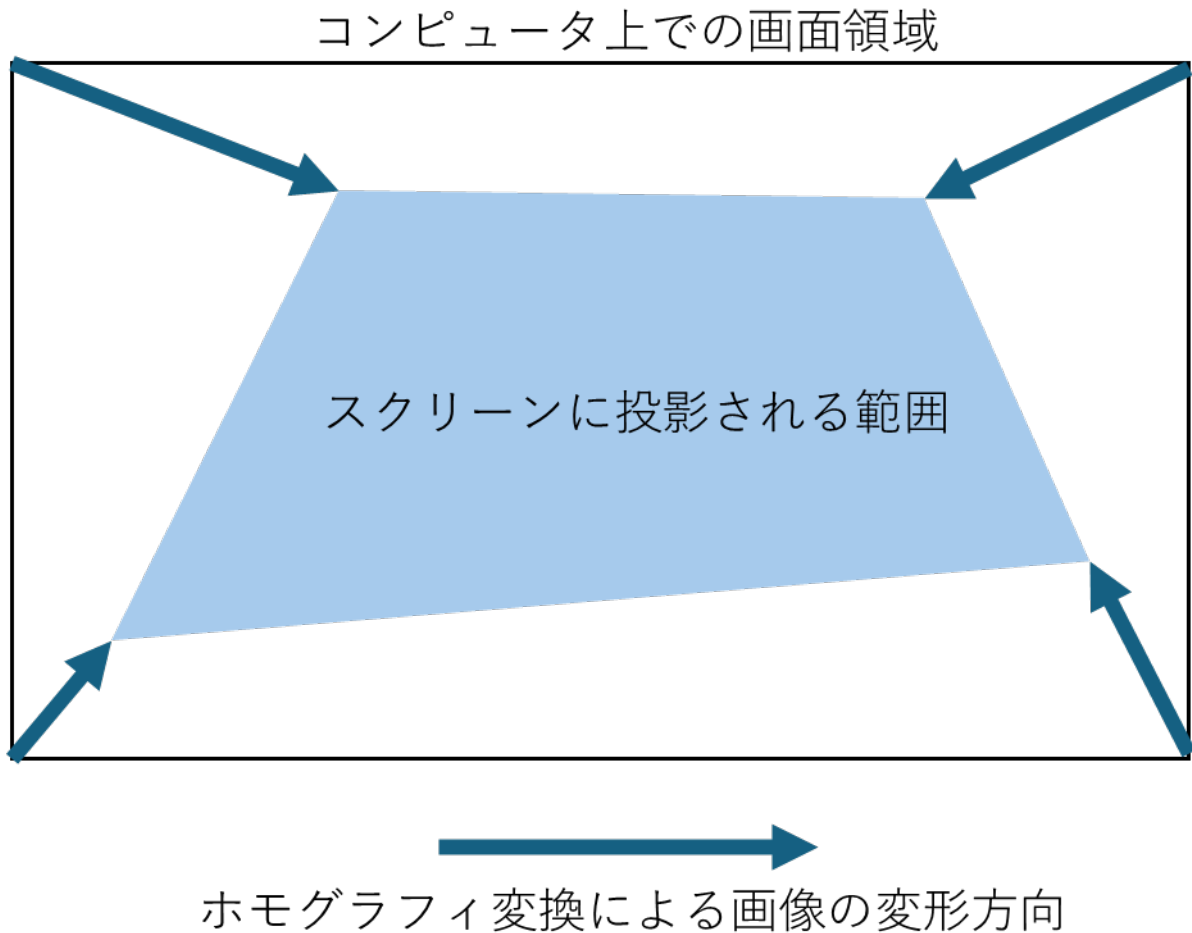


図4 ホモグラフィ変換によるスクリーン上での見え方の補正

360度映像をソフトウェア上で正しく展開する機能を用いて60度画像のテクスチャを貼られた球状のモデルデータの中心を視点として体験者からスクリーンに映った画像が正しく見えるようにするために映像を補正する機能を使ってプロジェクターに出力することで360度映像コンテンツ表示が可能となる。

4. システム実装

まず360度映像をソフトウェア上で正しく展開する機能に関して、事前にBLENDERという3Dモデリングソフトを用いて球状のモデルデータにテクスチャを適切に貼るためのUV座標をEquirectangular形式に合わせた形埋め込んで作成した。この作成したモデルデータのPLAYCANVASへの読み込みと3D空間上への配置を行なった結果が図5である。

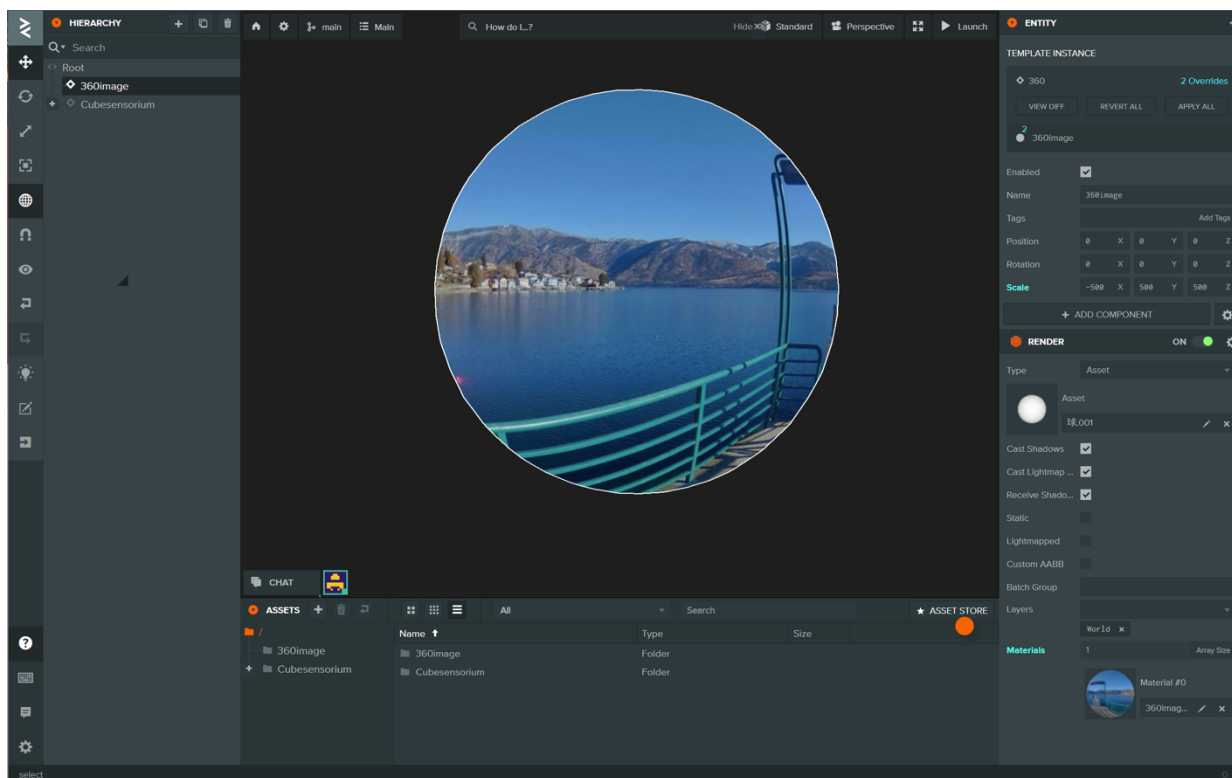


図5 PLAYCANVAS 上で配置した360度画像を貼り付けたモデルデータ

次に体験者からスクリーンに映った映像が正しく見えるようにするために映像を補正する機能を実装する。まずコンピュータからは4つのプロジェクター毎に1つの巨大なディスプレイとして認識されていてスクリーン全体は2つの巨大なディスプレイとして認識されている。ウェブブラウザは全画面表示に対応しているが複数のディスプレイをまたぐ形での全画面表示には対応していないため1つ目の巨大なディスプレイ用のモードと2つ目の巨大なディスプレイ用のモードの2つのモードを用意してウェブブラウザを2つ起動してそれぞれ全画面にして適切なモードへ操作して表示することとした。次に巨大なディスプレイの中に4つのプロジェクター用の透視投影変換を行なう。PLAYCANVAS 上での透視投影変換はカメラのコンポーネントを通して行なうため、仕様を確認したところビューポートの指定で1つの画面を区切って複数の視点カメラを表示することが可能であると確認できたためそれを用いてカメラを設定した。またテクスチャへのレンダリングについてはスクリプトに記述することで可能であったためそれを用いて実装をおこなった。ホモグラフィ変換については任意の形にできる平面を用意してそこにレンダリングしたテクスチャを貼り、その平面を適切なビューポートを指定したカメラのコンポーネントで撮影してプロジェクターにて出力した。

図6は4台のプロジェクターに対してそれぞれ右目用と左目用の映像を左右に並べて出力するためにビューポートを8つに分割して出力した画面である。今回は立体視を行わないために左右の視差がないため同じ映像が2つずつ出力されている。実際の動作の様子が図7である

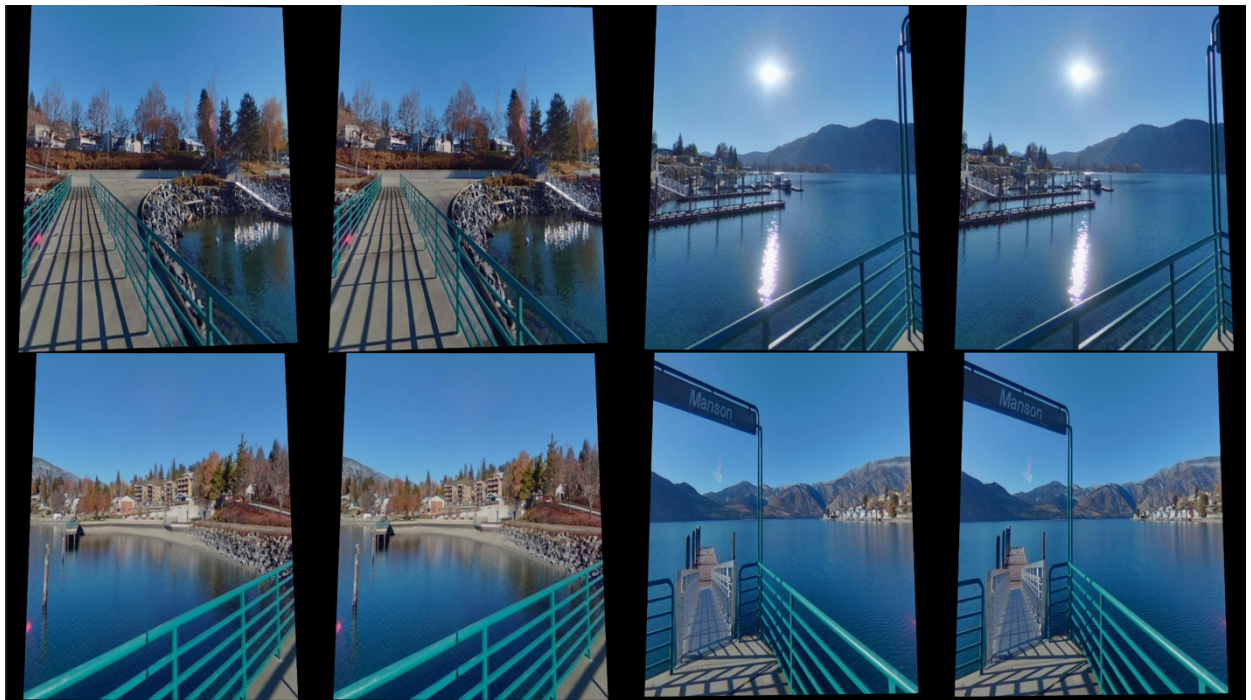


図6 プロジェクター4台分の出力画面

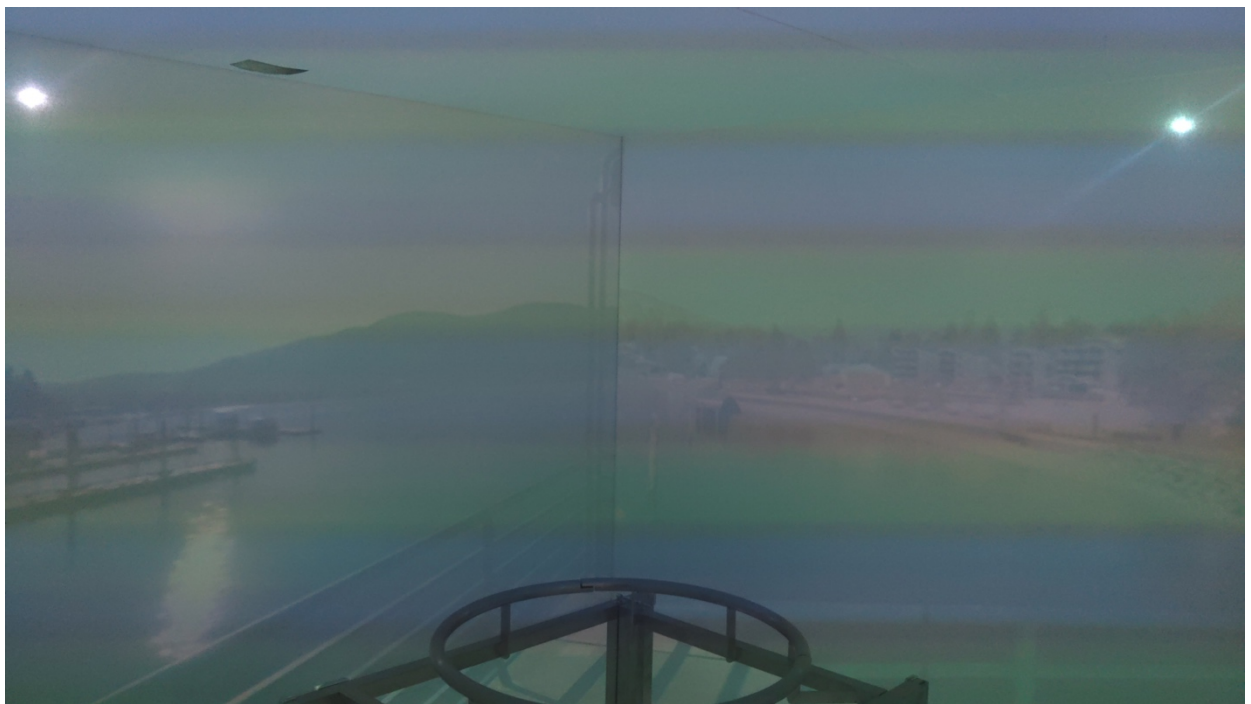


図7 実際の動作の様子

5. まとめと今後の展望

JavaScript ベースのゲームエンジンである PLAYCANVAS を利用してプロジェクション型 VR 用 360 度画像表示機能を開発することができた。今回はウェブブラウザを2つ開いて表示をおこなったものの相互に通信を行なっている訳では無いいため同期しておらず、動的コンテンツ表示

をすることが現状では出来ないので通信機能の実装あるいはハードウェア側で全てのプロジェクターを1つの巨大なディスプレイとして認識させるなどして動的コンテンツ表示に対応させることや、WebRTCを活用して360度映像によるビデオチャット機能の開発、またウェブアプリである利点を活かして複数のデバイスで連携して1つのプロジェクション VR を構成するような機能の開発などを行なってよりプロジェクション型 VR での VR コンテンツ活用の促進を図って行きたい。

参考文献

- [1] Meta, Meta Quest, <https://www.meta.com/jp/quest/>, 最終アクセス 2024/03/24
- [2] HTC, VIVE, <https://htcvive.jp/>, 最終アクセス 2024/03/24
- [3] Unity Technologies, Unity, <https://unity.com/ja>, 最終アクセス 2024/03/24
- [4] Epic games, UnrealEngine, <https://www.unrealengine.com/ja>, 最終アクセス 2024/03/24
- [5] The Khronos Group, OpenXR, <https://www.khronos.org/openxr/>, 最終アクセス 2024/03/24
- [6] Richo, THETA, <https://www.ricoh360.com/ja/theta/>, 最終アクセス 2024/03/24
- [7] GoPro, MAX, <https://gopro.com/ja/jp/shop/cameras/max/CHDHZ-202-master.html>, 最終アクセス 2024/03/24
- [8] Insta360, insta360, <https://store.insta360.com/product/x3?c=2118&from=homepage>, 最終アクセス 2024/03/24
- [9] 高鳥 光, 圓崎 祐貴, 矢野 博明, 岩田 洋夫, 大規模没入ディスプレイ LargeSpace の開発, 日本バーチャルリアリティ学会論文誌, 2016 年 21 巻 3 号 p. 493-502
- [10] Wisconsin Institute for Discovery, UniCAVE, <https://widve.github.io/UniCAVE/>, 最終アクセス 2024/03/24
- [11] Playcanvas, <https://playcanvas.jp/>, 最終アクセス 2024/03/24
- [12] 筑波大学、武蔵野大学、株式会社丹青社, 西新宿サイバーフィジカル美術館, shino-u.ac.jp/event/20230227-00000999.html, 最終アクセス 2024/03/24
- [13] Richo, 360 度カメラの仕組み、エクイレクタングラー形式について RICOH THETA を題材にご紹介!, <https://blog.ricoh360.com/ja/8212>, 最終アクセス 2024/03/24