

# ウェブに統合されたXR技術 "Immersive Web" の現在と今後の発展・連携の展望

W3C/Keio

Atsushi Shimono

# Outline

- “Immersive-Web”技術の概要
- WebXR Device API (Core)の現状・勧告化への展望
- WebXR Core技術を拡張するモジュール群
- 開発者向けリソース
- Immersive-Webと他の標準技術との連携・発展
  - 新しい仕様に向けたincubation
  - WebGPU、3Dデータ形式、“ブラウザ”の拡張

# Immersive-Web (WebXR) とは

- もともとウェブ上でVRを実現するために考案されていたWebVRがベース
  - 2014年頃のMozillaのゲーム系の活動からの派生
  - 2016/3にWebVR 1.0、2017/12に[WebVR 1.1](#)リリース
- 初期段階の動き
  - 2016/3に[W3C WebVR CG](#)が設立
  - 2016/10に[ウェブとVRのW3C ワークショップ](#)開催
- 2017年頃にWebVR 2.0の議論の中で、VRのみでなくARなども視野に入れた名称として[WebXRが採用](#)することとなり、[navigator.xrを利用する](#)ように変更された
- 2018/9にImmersive-Web WG/CGが立ち上がり、本格的にWebXRの勧告仕様を目指した活動が始まった
  - この段階でWebVR CGはImmersive-Web CGとなった
  - CGで新提案の受付とまとめた仕様案とするためのインキュベーションの議論、WGで勧告仕様を目指しての仕様開発を行う体制、で運用している
  - CGで議論された仕様案がある程度固まったとWGにおいて判断したところでCGからWGに移行
  - 2020/5に将来的に標準化の作業に移行可能と目されたモジュールを追加して[WGの第2期](#)を開始
  - また、2019/6の[W3C Webとゲームのワークショップ](#)でも関連の話題が出た
- 現在ではWebXRの仕様は、WebXR Device APIというコア仕様と、コアAPIに対して各種の拡張を行うモジュール群に分かれている
  - モジュールは個別に提案されたまとまりごとに開発・検討されている
  - 強く関連するモジュールについてまとめる議論が必要？という議論も出はじめてはいる
  - また、WebXRの直接の拡張でない仕様の議論も：[XRマーカー](#)、[WebXRとWebGPUのバインディング](#)、[3Dのfavicon](#)など

# Immersive-Web (WebXR)の概要 I

- WebXR Device API本体の主要機能：
  - 入出力デバイスの検出と接続
  - デバイスの状態が変化した時にアプリケーションに通知するイベントの発火
  - アプリケーションから提供されたレイヤーデータを出力デバイスに送出する操作とそれに伴うタイミング・スケジューリングのハンドリング
- 各モジュールはWebXR Device APIに定義されているインターフェースやプロパティーに、モジュール独自の処理などを追加する形で拡張している
  - 仕様が短くて拡張の意味がわかりやすいのが[gamepads module](#)でしょうか
- レンダリングするデータ自体を扱うのはLayerのカテゴリに入る機構で、WebXR Device API自体ではレンダリングは行わず、現在のところXRWebGLLayer()が実用になる状態。
  - XRWebGLLayerはWebGLでサポートされている現状で唯一のフレームバッファとなり、WebGLのbindFramebuffer()でつないでWebXRの出力デバイスへの送信を行う。
  - WebXRのモジュールとして追加のレイヤーハンドリングのためのAPIは検討されている
- 入力デバイスはXRInputSourceで表現され、デバイスのプロパティーが得られる
  - Gamepadsを含めて入力装置のレジストリが[WebXR Input Profiles](#)として提供されている
  - 入力自体はXRSessionのXRInputSourceEventによりイベントの形で渡されてくるので、イベントリスナーを登録するなどしてアプリケーションから読み込む

# Immersive-Web (WebXR)の概要 II

WebXRのアプリケーションのベーシックなライフサイクルは次のような流れに：

- `navigator.xr`の存在を確認し、WebXRをサポートしていることを確認する
- `navigator.xr.isSessionSupported()`を呼び、WebXRが利用可能かと使えるモードを確認 (要ユーザによる認証)
  - 引数に入れられるモードは`inline`, `immersive-vr`, `immersive-ar`など、モジュールの追加により増える可能性も
  - 利用したいモードが使えるかを確認し、だめなら終了する(か別なモードへフォールバックする)
- `navigator.xr.requestSession()`を発行し、戻ってきたPromiseが完了してXRSessionオブジェクトが渡されるのを待つ
  - このオブジェクトに対してXRSession.requestAnimationFrame(callback)でコールバックを登録しその中でフレームを投げたり、オブジェクトにイベントハンドラをつないだりすることで操作を行う
  - 登録したコールバックにはタイムスタンプと描画すべきターゲット情報を含むXRFrameの2つが渡される
  - コールバックの処理の最後にもう一度requestAnimationFrameを呼びなおす
- 終了時にはXRSession.end()を呼ぶ

詳細や具体的なコードについては、あとで紹介する開発者向けリソースの中からサンプルをいくつか見ていただいて動かしながら確認していただくのがわかりやすいと思います。(のでここではコードには触れません)

非常に重いXRのレンダリングを行ったりするようなアプリケーションでは、各フレームを描画するコールバックの中で処理を行わずに、Workerを別途立ち上げておいて処理結果だけを渡すことが推奨されます。XR環境では特に高フレームレートでの描画が重要になってくるため、コールバックが呼ばれた後での描画を戻すタイミングが重要となってきます。

# WebXR Device API (Core)の現状・展望

- 2019/2に[FPWDをリリース](#) (その後の[履歴はこちら](#))
- 2019/6にXRについて関係するさまざまな機構を一つのモノリシック仕様として同時にまとめることの困難さ(大きくなりすぎる、仕様としてまとめる時期が異なる、など)から、[モジュール化提案](#)が提示され[議論](#)の結果WebXR Device APIから[gamepads](#)と[AR](#)のモジュールを[分離](#)した
- 2019/10頃にほぼfeature freeze状態になり、細かいつめやエラーハンドリングなどの最終的な勧告仕様のレベルに必要なアップデートを行う状態に
  - 一時期はWebXR Device API Level 1という表記ではあったが、いまのところLevel 2として次期バージョンを策定する予定はない
- 2020/7にCRに向けた最終のレビュー用WDを公開
  - いろいろなデバイスですでに実装されている(&エラーハンドリングなどもさまざまなインプットから対応策が導入されている)ので、このバージョンでほぼそのまま勧告仕様となる予定
  - `navigator.xr.isSessionSupported()`についてのfingerprinting対策の最後の詰めを行っているところ
    - XRデバイス内のウェブブラウザなどXR対応が明示的にわかっている環境ではユーザの認証なしにtrueで返すのが望ましいが、これまでのPCブラウザではなかった分類のためにどういう条件で行うかなどの詳細を仕様に記述しないといけないというコメントに対する議論、など
  - プライバシーレビュー対応の部分のみat-riskにしてCRに行き、解決後すぐに勧告化のプロセスを、という話もあったがまだWD段階

# WebXR技術を拡張するモジュール群 Ia

- WebXR Gamepads Module

- 2019/10にFPWD、それ以降は明確化などの微修正のみの状態で、ほぼ固まっている
- Web Applications WGによる Gamepad API を WebXR Device API につなぐための仕様で、もともと WebVR では InputSource の一部に入っていたが、モジュール化に伴って gamepad のモジュールとして分離された
- ボタンとタッチパッド(かスティックにも読み替え可能な構成)についての入力イベントへの統合についての定義

- WebXR AR Module

- 2019/10にFPWD、それ以降あまり更新はされていない(固まっていないというわけではなく、細かいつめの修正は入っている)
- WebXR Device API (core) から AR の部分のみが抜き出された拡張モジュールで、XRSessionMode に immersive-ar を追加する

- Dom Overlays

- ドラフト仕様 が作成されつつある段階
- HUD (Head Unit Display) のような現実世界や表示している AR コンテンツの手前に(通常は)透過する状態でオーバーレイする UI を DOM を用いて構成するための仕様
- オーバーレイに適用するための CSS の仮想クラス(と既定のブラウザ用スタイルシート)も定義される

# WebXR技術を拡張するモジュール群 1b

- [Hit Test](#)

- [ドラフト](#)をアップデートしている段階で、まだ安定的なAPIになっている段階ではなく型定義やインターフェース定義の多数が未確定状態
- ARで必須となる床検出やオブジェクト検出などの現実世界の状況の把握を行うための方法の一つとしてのhit-testのAPI

- [Layers](#)

- 2020/12初頭にFPWD予定
- 映画館のようなプロジェクション、複数領域、シリンダー状などさまざまなレイヤーに対してレイヤーとして表示するためのモジュール
- `xr.requestSession`の`optionalFeatures`に'layers'を入れて有効化し、レンダリングについてはWebGL、テクスチャ、ビデオなどのメディアをソースとして適用できるように拡張されている

- [WebXR Hands Input](#)

- [FPWD 2020/10/22](#)
- 手のポーズ(指・関節の状態・位置)をトラックしてXRInputSourceにXRHandとして追加するためのモジュール
- 各手に指の位置をトラックするための24ポイントが定義され、それぞれの位置が入力される



# WebXR技術を拡張するモジュール群 IIa

- [Anchors](#) (CG)
  - [ドラフトの仕様](#)を作成しつつある段階
  - XRの下位レイヤーに対して、いろいろな表示や処理を行う際のアンカーなる基準位置を定義し利用するための機構
- [Computer Vision](#) (CG)
  - さまざまなcomputer visionに関連する機能・モジュールを検討するための議論用に立ち上げられたレポジトリ
  - 幾つかのexplainer的な文書を作成中ではあるものの、ここ(だけ)でまとまった仕様を策定するという流れに進むわけではない状態
- [Geo Alignment](#) (CG)
  - [Explainer](#)が存在するだけの状態
  - 提供しているARのシーンに対して、赤経・赤緯・高度といった地理的情報を提供することを目標とするモジュール
- [Lighting Estimation](#) (CG)
  - [仕様案の初期ドラフト](#)が提案されている段階、WebXR AR moduleとWebXR Layers moduleを必要とするWebXR Device APIに対する拡張
  - ユーザ環境における光の当たり方の状態を推定し、光源位置や方向、また光の反射状況などの光に関する状況を取得し、またWebXR Layersに対しWebGLTextureに対して適用する

# WebXR技術を拡張するモジュール群 IIb

- [Navigation](#) (CG)
  - まだ影も形もない段階
  - 2Dのブラウザではボタンクリックなどの方法でVRの起動などの操作を行えるが、リンクが張られたVRコンテンツ内での遷移の場合に2Dページを読み込まずにシームレスに3Dコンテンツを呼び出すなどの操作をどうやって行うかなどの移動操作の最良慣行を形成していくための仕様
- [Real World Geometry](#) (CG)
  - [ドラフトの仕様](#)の作成が少し進みつつあるところ
  - 現実世界の状況を把握するためのAPI群になる予定で、表面検出APIに最初は注力する予定。床やメッシュの判定を行ったデータを取得する目的。
- [Depth Sensing](#) (CG)
  - まだ[explainer](#)を作成中の初期段階、WebXR AR moduleを必須とするWebXR Device APIに対する拡張
  - スマートフォンとかのデバイスでARを実行する際に周辺環境を調べるセンサーがついているデバイスが増えてきていて、実世界に配置されているものの距離が測定できることがある。その場合に、ARで描画する物体と現実との衝突判定やロボットなどの移動経路ナビゲーションなどに有用な画面上の特定の位置(x,y)に対する距離を取得するためのAPI群の提案
  - 点群としてのデータ取得や、テクスチャ的な取得などがターゲットに入っている。ARKitやARCoreのDepth APIをベースとする予定
- [Detached Elements](#) (CG)
  - [とりあえずレポジトリがある](#)、というだけの段階。WebXRのセッションのウィンドウからdetachedな状態になっているHTML/CSSの要素を描画するためのモジュール

# 開発者向けリソースの紹介

- WG/CGとして維持している開発者向け導入・解説サイト:  
<https://immersiveweb.dev/>
  - デモ、各種ライブラリ・開発用エンジン、ブラウザ・デバイスでのサポート状況についてのまとめ
- WebXR Samples: <https://immersive-web.github.io/webxr-samples/>
- WebXR Polyfill: <https://github.com/immersive-web/webxr-polyfill>
- MDN:  
[https://developer.mozilla.org/ja/docs/Web/API/WebXR\\_Device\\_API](https://developer.mozilla.org/ja/docs/Web/API/WebXR_Device_API)
  - ガイド・チュートリアル・API解説については日本語翻訳はほとんどまだない
- ImmersiveWebWeekly: <https://immersivewebweekly.com/>
  - メールニュース、ウェブ上で[見れなくもない](#)(がインデックスはいまのところない?)

# 他の標準技術との連携・発展 I

Immersive-Web WG/CGではモジュール案についての提案受付や初期議論をproposalsのレポジトリでissueの形で一括して扱っている。このレポジトリでの活動はCGとしての運営となっており、ある程度まとまった段階で専用レポジトリの作成を行い、CGからWGへの移行を目指す仕様案策定に向けての議論を開始するという流れで活動している。

また、この中には他の標準化仕様との連携によるアップデートを目指すような項目や、WebXR関連モジュールに幅広く関連するような項目も存在している。

提案段階の議論としては:

- Spatial Favicons

- PC版ブラウザでなく、XR版ブラウザ、特にXRデバイスに組み込まれたブラウザや、XRデバイスのホーム画面上で、サイトへのリンクをアイコンとして置くときに、現在の2Dのfaviconでは見えが悪い(!?)ために、3Dでのfaviconが欲しいけれども、という議論。
- いまのところexplainer位しかないレポジトリではあるものの、だいぶ前から検討やHTMLへの提案などを行う動きはあるが、どのような条件を課す伽耶、どのように表示されるべきなのかなどの詳細を詰めるまでには非常に時間がかかりそうな状態

- Native GLTF

- もともとWebゲームに関するW3Cワークショップで議論された3DエレメントをHTMLに追加することについての提案で、画像や動画(EME/MSEなどを含む著作権保護的なレイヤーも含み)と同様にネイティブでレンダリング・再生できるようにしたいという提案
- 関連して、HTMLなどのような宣言的なXRのための言語仕様についても昔から提案があるが、X3DやAframeなどに見るように既存の実装から見ると、幅広い取りうる柔軟性の度合いがあるために、よりいろいろな実装・提案が発展して、ある程度のスイートスポット的な収斂点が見つかるまで議論は継続するが保留という形になっている。

# 他の標準技術との連携・発展 II

(提案段階の議論続き)

- [WebXR WebGPU Binding](#)

- [explainer](#)のレポジトリが作成された。もうすぐFPWDが出る予定のWebGPUの仕様に合わせて、Layersモジュールなどでの利用を簡単にするためのXRWebGLBindingと同様のXRGPUBindingを策定する

- [Extending WebExtensions for XR](#)

- [既存のブラウザ拡張機能の機構](#)をどのようにXRに拡張するかについての議論

- [Performance Improvements](#)

- 何かのドキュメントを作成する、というのではなく、WebXRのパフォーマンス改善に関わる議論を行っていくためのレポジトリとして開設
- 行われている議論や予定されている議論には次のようなものがある
  - Layerモジュールにも関連するものの、テクスチャデータの形式に圧縮形式を追加する
    - これについては、WebGPU WGにおいても、どのフォーマットがパフォーマンスやbindingなど様々なメトリクスを考慮すると扱うのに望ましいかについての議論が進行中でもある
  - WebGPUなどのメディア系のAPI同士の間で処理をつなぐ間でメモリコピーによるオーバーヘッドを減らすための設計
  - フレームレートに間に合う計算処理が行えなかった場合にどうやってコマ落ち状態になっているかをアプリケーション側から把握するのかという問題 (WebVR時代からの持ち越し)

# 他の標準技術との連携・発展 III

(提案段階の議論続き)

- 360度ビデオ関係

- 360度のビデオは制御を掛けようと思うといまのところ専用のクライアントを書いて出す必要がある。それに対して、[プレイバック対応のレンダラー](#)、[字幕フォーマットの対応](#)(Timed-Text WG)、[WebXRでの字幕の表示](#)、などの議論が提示されており、また、Mozilla Mixed Realityでの[immersive向けcustom element](#)の開発なども存在する。

- アクセシビリティー担保

- XR上にレンダリングされているオブジェクトにどのように[アクセシビリティー向けの情報\(ものが何かの情報\)を出す](#)のか、の議論や、音声に対する字幕などについての議論がある
- また、この分野については、[W3C workshop on Inclusive Design for Immersive Web standards](#)が開催され、[Inclusive design for the Immersive Web CG](#)も設立されている