

# UNITY WebGL 入門

ユニティ・テクノロジーズ・ジャパン

池和田 有輔

皆さん、Unity使ってますか？

Unityは多数のプラットフォームに対応しています



# UNITYのブラウザ対応の歴史

Unity Web Player



2005.6 -

- 2016.5

1.0

5.3

NaCl (Google Native Client)



2011.12 -

-2015.4

3.4

4.6

Flash Player



2012.11 - 2013.4

4.0 4.1

WebGL



2016.5 -

5.3

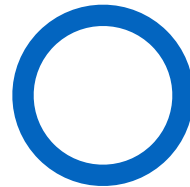
# 今日のポイント

UnityのWebGLって、どのくらい使い物になるの？

ゲーム以外のリッチコンテンツ制作用途にも使えるの？

# UnityのWebGLって、どのくらい使い物になるの？

- モバイルのネイティブアプリであれば近いパフォーマンスが得られる。
  - Standard Shaderのクオリティはモバイルよりもデスクトップに近い。
  - ソフトシャドウも使える。
- 反面、CPUとメモリはボトルネックになりやすい。
  - シングルスレッド、ヒープ領域に制限がある。

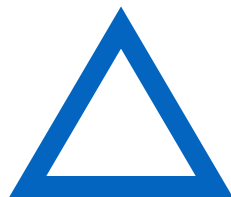


## ゲームのリリースも着実に増えてます

- すでに多くのゲームがリリースされています
  - Facebook上に多数
  - DMM Games 「一血卍傑」 など。
  - 他、Made With Unity (<https://madewith.unity.com/>) などを参考に

## ゲーム以外のリッチコンテンツ制作用途にも使えるの？

- ブラウザは64bit版がほぼ必須。Firefox = Edge > Chrome > Safari。
- モバイルでの動作は現状サポート外。一応動くこともある程度。
- ロード時間はまだ課題が残る。
- 他のサービスや技術との併用はそれほど考慮されていない。
  - 外部のJSからメソッドを叩いたり、叩かれたり是可以する。





DEMO

## Unityでの設定のポイント

- 使用するテクスチャは基本的にCrunched Textureを使おう。
  - Crunched Textureは、DXTCをディスクサイズを減らすために圧縮したもの。
- 通常のビルドはディベロップメントビルドで行おう。書き出しが速く、ユーザーコードの可読性も一応保たれ、プロファイリングが可能。
- Use Pre-Build Engineにチェックを入れるとでさらに書き出しが速くなる。
- リリースビルドを行うと難読化したものをgzipによって圧縮する。ブラウザ上で実行時に解凍して表示。上記を130MB→5MB程度まで圧縮する。

## Unityでの設定のポイント

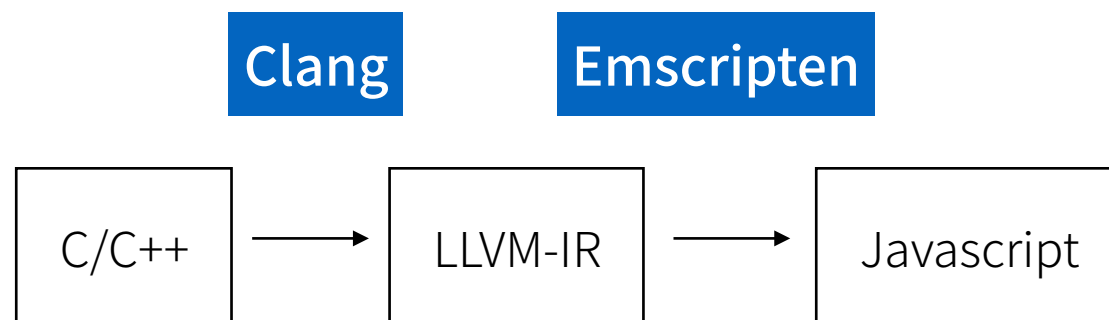
- WebGL2.0対応させるには、PlayerSettingのAuto Graphic APIのチェックを外し、Graphics APIsにOpenGL ES 3を追加、出てきたWebGL2.0を1.0の上にドラッグ。
- また、Strip Engine Codeにチェックを入れると使っていないエンジンコードを片っ端から削除してくれる。
- Publishing SettingsのWebGL Memory Sizeでヒープメモリの割り当て量を変えられる
  - 小さすぎるとメモリ不足で実行されず、大きすぎるとブラウザが落ちたりするので軽視しないように。デフォルトは256だが、2000程度が無難だと思われる
- WebAssemblyもここから設定できる（5.5以降）

Unity WebGLを支える技術

# Asm.js

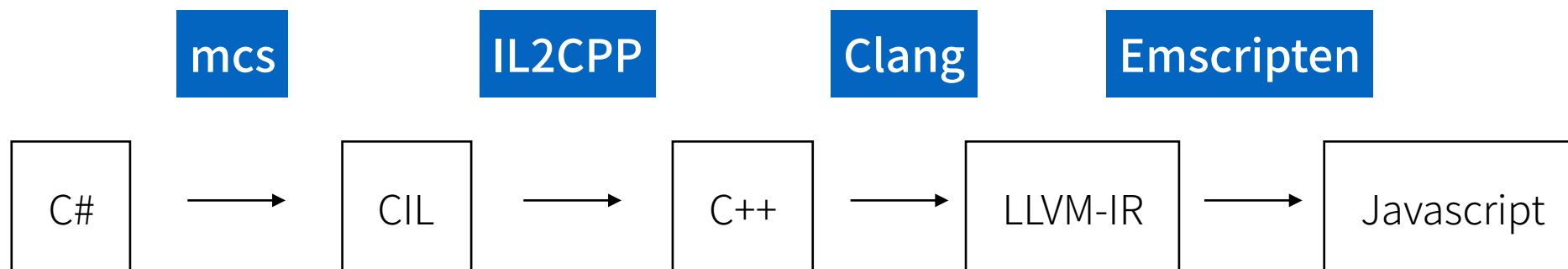
- 処理速度向上を目的としたJavaScript言語仕様のサブセット。
  - 動的型付けの問題を回避。
  - 可読性や保守の効率性は考慮されていない。
- 事前コンパイル（AOT）。
- 主要ブラウザはほぼ対応済み（IE以外）。
- サブセットなので対応していなくても実行される。

# Unityのランタイムの変換フロー



- エンジンコード部分に相当するコードはC++
- 一旦コンパイラ基盤LLVMバイトコード（中間コード）に変換
- そして前述したams.jsに最適化されたJavascriptに変換

## ユーザーが記述したコード変換フロー



- monoのRuntimeを使わず、Unityの独自技術であるIL2CPPによってC++に変換。
- その後は概ね同じ手順でJavascriptに（或いはWebAssemblyバイナリに）。
- IL2CPPの技術はiOSの64bit対応などにも使われました。

## 特筆すべき点など

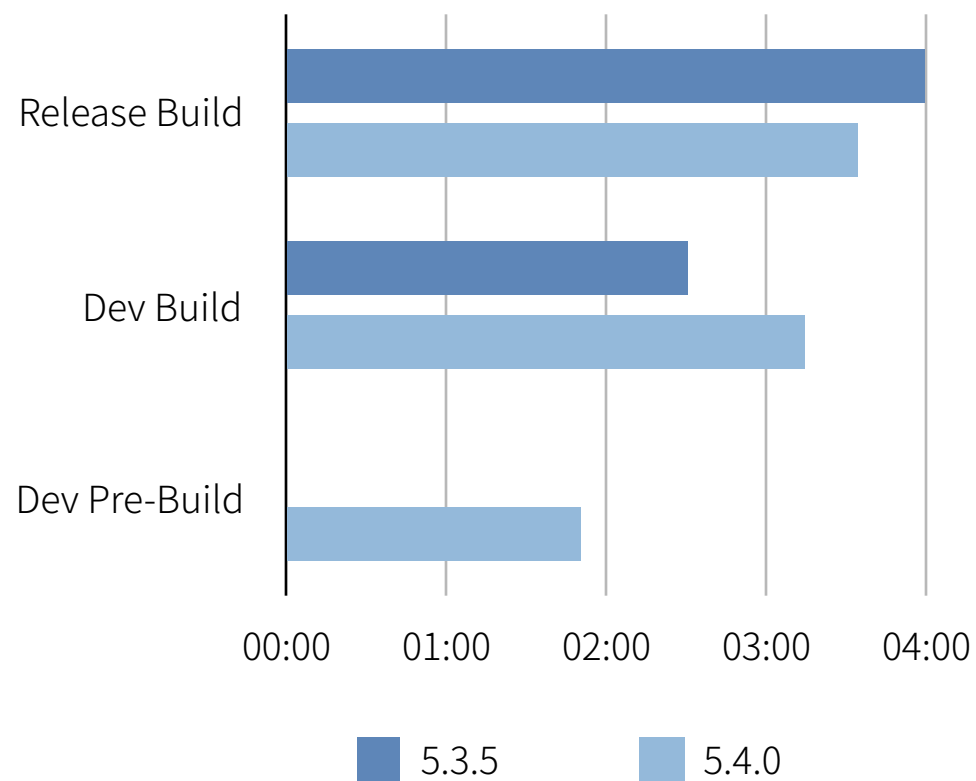
- AOTなのでiOSなどと同様に動的コード生成は使用できない。
  - System.Reflection.Emitとかダメ
- シングルスレッドの問題
  - System.Threading 名前空間とか使うとエラーになる
- オーディオの制限
  - 最低限の機能しか使えない。時にはあきらめも肝心。
- ゲームパッドが使えるのはFirefox & Chromeのみ。



## 最適化のコツ

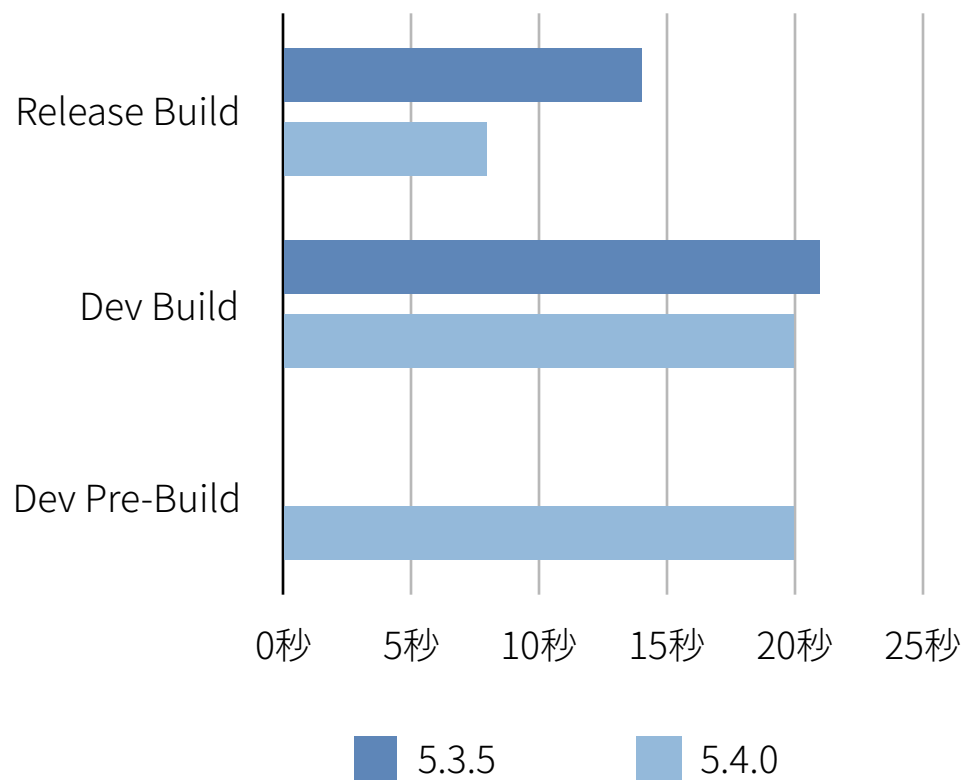
- エフェクトなどの処理は可能な限りGPUに投げよう。
- アセットバンドルを有効にしよう
  - 追加コンテンツなどを作るものだが、LZ4 + gzipが使えるため全体の容量が削減される。

# Unity5.4でのトピック



- ビルド時間の短縮
- Use Pre-Build Engineオプションでさらに早く

## Unity5.4でのトピック



- 起動時間の短縮
- リリースビルドで5.3の時の半分近い起動時間になりました

## その後のロードマップ

- Unity5.5でWebGL2.0に完全対応（現在は一部対応中という感じ）
- WebAssemblyを開発チームが検証・最適化中。
  - すでにUnity5.5のベータ版で実装済み
  - 正式リリースはブラウザベンダー待ち（年末か来年？）
  - ベンチマークスコア：x1.64
  - メモリ：x0.68
  - ビルドサイズ：x0.66
- スマートフォンなどのモバイル対応は・・・もう少し先になる見込み

## よくある質問、その回答

- **ビルド設定にWebGLが見当たらないんだけど**

- WebGLで書き出せない場合はアドオンが入っていないことを疑おう。
- その場合インストーラから追加インストールでインストールすべし

- **ローカルで確認しようとしても何も表示されないよ**

- ビルドをBuild & Runで実行して確認するのがベター
- Unityが中にテンポラルサーバーを立ち上げているため、ブラウザのURLを見るとlocalhostになってる。これは一部ブラウザはセキュリティの問題でローカル環境ではアセット等がロードできないことがある。それを回避するための手段。

# よくある質問、その回答

- **フォントは？**

- ユーザーのマシンのフォント、WEBフォントどちらも使えません。
- 通常はUnity内にアセットとしてFontデータを読み込んで使います。

- **日本語が表示されないよ**

- おそらく標準のArialから変えてないせい。
- WebGLは表示されないフォントがフォールバックされず、結果欧文フォントのArialは表示されません。
- 適切なフォントを指定することで回避しましょう。

# よくある質問、その回答

- 入力フォントとして日本語が使えない
  - IMが使用できないので難しい問題です。
  - 回避策として、こちらのアセットを使おう。
  - WebGL Native InputField
  - HTMLのダイアログを出し、そこに入力させることで解決してます



# よくある質問、その回答

- **ビルド時間、長いよね・・・**

- そこでCloud Buildですよ。
- 無料版であるUnity Personalでも使えるようになりました。

- **2Dゲームは作れる？**

- もちろん！
- 物理エンジンとしてはパフォーマンスを得られるのは2Dの方かも。
  - (box2d: シングルスレッド / PhysX: マルチスレッド)





Unity x WegGLで、素敵な開発体験を！

ご清聴ありがとうございました