BRINGING EVERYWHERE
UNDERSTANDING THE PROJECT
WHO'S MARCELO MANCINI (HIPREME)

- GAME DEVELOPER;
- ENGINE PROGRAMMER;
- OPEN SOURCE;
- HOBBY MUSICIAN.

- D
- TYPESCRIPT
- LUA
- JAVA
- HAXE
- C++
- C#
WHAT IS HIPREME ENGINE

- D GAME ENGINE;
- 5% JAVA, C/C++, OBJ-C, JS;
- 2D;
- AVOIDS C;
- OOP BASED;
- HOT RELOADING;
- MUCH MORE...

- XBOX SERIES;
- ANDROID;
- BROWSER (WASM);
- PS VITA;
- MACOS;
- WINDOWS;
- LINUX.

Available at
https://github.com/MrcSnm/HipremeEngine
THE START
INITIAL IDEA

- USE SDL 2;
- DO ABSTRACTIONS;
- DO IT THE "D" WAY;
- LEARN OPENGL WITH D.
MAIN REFERENCES
PORTING PHASE 1 - ANDROID

- TESTING D;
- THE BUILD SYSTEM;
- THE LACK OF DOCS.
PORTING PHASE 1 - XBOX SERIES S

- TESTING D;
- THE BUILD SYSTEM;
- C++/CX AND UWP;
- DIRECTX.
PORTING PHASE A - LESSONS

- USE MORE THAN 1 LANGUAGE;
- SHARED LIBRARIES ROCKS;
- AVOID C DEPENDENCIES;
- USE D LIBRARIES.
THE PLAN
PORTING EVERYTHING FIRST

- LESS REFACCTORS;
- MORE STABILITY;
- MORE THAN A SANDBOX;
- PROOF OF D CAPABILITIES.
TARGET WEBASSEMBLY - THE WALL
LOOKING INTO PRIOR WORK

**UNDERRUN**

Available in
https://skoppe.github.io/spasm/examples/underrun/

**THE ART OF MACHINERY**

EMSCRIPTEN-D

Available in
https://theartofmachinery.com/2018/12/20/emscripten_d.html
MAIN PROBLEMS

- NO D-RUNTIME;
- HAZY DOCUMENTATIONS;
- LACK OF REFERENCES;
- MOST PROJECTS WERE BIG.
SOME HOPE

Accessible in
https://github.com/hmmdyl/LWDR

Accessible in
https://github.com/adamdruppe/webassembly
EXTENSIVE TESTING

https://github.com/MrcSnm/webassembly/blob/062b2555fbdb14b278f3e3d6a508a5749b6e4b5a/test_runtime.d
PORTING PHASE 2 - WEBASSEMBLY

- D - JS COMMUNICATION;
- BYE-BYE, C STDLIB;
- THE WEB PHILOSOFY;
- ASSETS LOAD REFACTOR;
- ASYNC INTERFACE ONLY;
- NO-COLLECT GC.
WASM COMMUNICATION

Sending User Defined Callbacks

```wasm
module hip.wasm;
version(WebAssembly):

//WebAssembly.Table replacement for HipremeEngine
private _shared ubyte* function(ubyte* args)[1]_anonymousFunctionTable;
//JSFunctions are represented opaquely right now.
alias JSFunction(I) = ubyte*;

//Gets a unique function index for usage in the table
extern(C) size_t _getFuncAddress(ubyte* fn);

//Javascript function to call a D callback
export extern(C) ubyte* __callDFunction(size_t addr, ubyte* args)
{
    return _anonymousFunctionTable[addr](args);
}

//Checks if function has been called with required arguments.
private ubyte* validateArguments(alias fn)(ubyte* args)
{
    import std::traits;
    //Only checking the count of
    assert(Parameters!(fn).length <= *cast(size_t*)args,
        fn.stringof="Expected "+Parameters!(fn).length.stringof=" parameters");
    return args + size_t.sizeof; //Only uses 1 size_t to determine arguments validity
}
```
WASM COMMUNICATION

Javascript Implementation

Receiving D Callback

D Calling JS Function and sending its arguments
WASM FILESYSTEM

JSON Representation of the available FS

Compile Time JSON Import
**WASM ASSET LOADING SOLUTION**

How the assets are found in the game.
WASM ASSET LOADING SOLUTION

```
Mixin template HipEngineMain(alias StartScene, HipAssetLoadStrategy strategy = HipAssetLoadStrategy.loadAll)
{
    immutable string ScriptModules = import("scriptmodules.txt");
    pragma(msg, ScriptModules);
    version(UseExternalScene)
    {
        _gshared AScene _exportedScene;
    }
    version(Windows)...}
    export extern(System) AScene HipremeEngineGameInit();
    export extern(System) void HipremeEngineGameDestroy();
}

export extern(System) void HipremeEngineMainScene()
{
    mixin LoadAllAssets!(ScriptModules);
    loadReferenced();
    return new StartScene();
}
```

Loads every asset before instantiating first scene
WEBASSEMBLY RESULT

Match 3. Playable at https://hipreme.itch.io/hipmatch3
PSVITA RESULT (SAME CUSTOM RUNTIME)

PSVita Version - Same code.
PORTABILITY DESIGN
COMPILE TIME REFLECTION

Using Reflection for Shader Vertex Buffer Descriptor

```c
@HipShaderVertexUniform("Cbuf1")
struct HipSpriteVertexUniform
{
    Matrix4 uModel = Matrix4.identity;
    Matrix4 uView = Matrix4.identity;
    Matrix4 uProj = Matrix4.identity;
}

@HipShaderFragmentUniform("Cbuf")
struct HipSpriteFragmentUniform
{
    float[4] uBatchColor = [1,1,1,1];

    @(ShaderHint.Blackbox | ShaderHint.MaxTextures)
    IHipTexture[] uTex;
}
METAPROGRAMMING

Using Reflection + Metaprogramming for binding to Java.
API POLYMORPHISM

Interface describing what is needed to implement a new renderer.
IS OOP INHERENTLY SLOW?

Intel VTune Profiler Statistics for Hipreme Engine.
Running for ~1 minute, 10K Dynamic Sprites.
BUILD AUTOMATION TOOL
MAIN REFERENCE

✅ POSITIVES:
- PORTABLE
- EASY CONFIGURATION;

❌ NEGATIVES:
- NEEDS JAVA RUNTIME;
- NO ANDROID SDK INSTALL;
- NEEDS ENVIRONMENT VAR.

libGDX Project Generator Tool

libGDX Project Generator

[Image of libGDX Project Generator Tool]
ANDROID SDK INSTALLATION

✅ POSITIVES:
- WORKS WELL;
- NEEDS JAVA RUNTIME;
- REQUIRES ANDROID STUDIO;
- HARD TO BEGINNERS.

❌ NEGATIVES:
- HARD TO BEGINNERS.
HIPREME ENGINE BUILD SELECTOR

- Integrated project generator
- Android SDK Tools Auto Install
- Target DFLAGS Auto Configuration
- Java Runtime Auto Download
- Automatic D Compiler Install (LDC)
- Prebuilt Binary Distributed
- Easy project selector
- MSVC Runtime Auto Install
- MSBuild Auto Install
- Compatible with OSX, Windows and Linux
- Uses no Environment Var
- External Configuration File
MSBuild Installation, required for specific DFLAGS.
SHORT TERM

- GAMES DEVELOPMENT;
- INCREASE VISIBILITY IN D;
- IOS PORT: EXTERN OBJECTIVE-C;
- DOCS;
- GAME FRAMEWORK.
MID-LONG TERM

- ENGINE UI DEVELOPMENT;
- PS4/5 PORT;
- VISIBILITY OUTSIDE D;
- 3D
THANKS FOR WATCHING!

Hipreme Engine