haxe NME

vs

The World

Presented by Philippe *"FlashDevelop guy"* Elsass

Part 1:

In the shadow of Flash

@elsassph

A bit of history

2004	2006	2008
AS2	AS3	haxe C++
MTASC*	haxe*	hxcpp/NME**

* created by ***Nicolas Cannasse***

** created by ***Hugh Sanderson***

Fast forward to 2011

Flash is the 1st indie game platform
but it ***failed*** to go mobile

HTML5 and ***tablets*** are the buzz

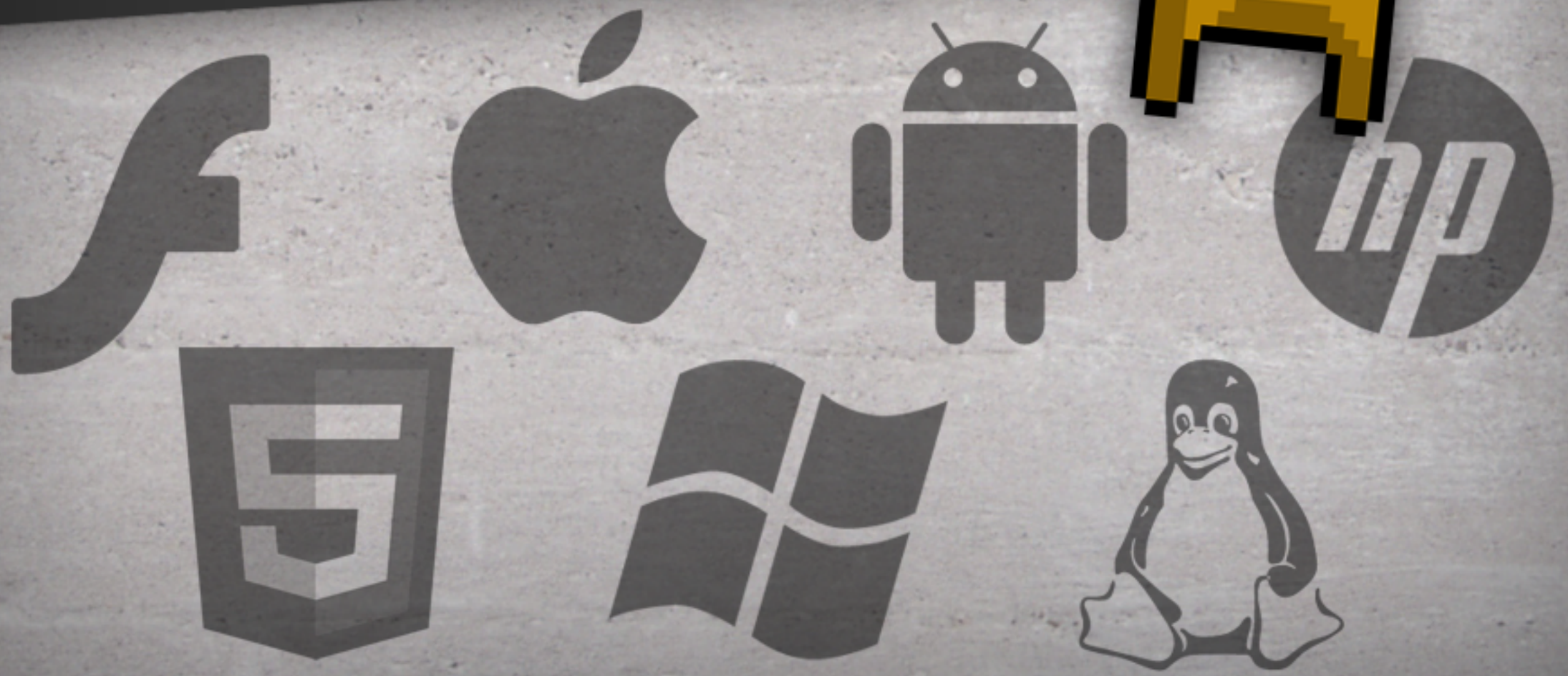
haxe NME has discreetly matured

In the shadow of Flash

Fast forward to 2011

Part 2:

Bunnies vs The World!



haxe NME under the spotlights,

thanks to...

haxe NME under the spotlights,

thanks to a professional evangelist,
Joshua Granick:

- *haxenme.org*
- *haxejs.org*
- *joshuagranick.com*
- *@haxelang*
- *@haxenme*
- *@haxejs*

haxe NME under the spotlights,

thanks to a kick-ass
BunnyMark:

- showcases native-level performance,
- with source:
  github.com/elsassph/nme-bunnymark

Bunnies vs The World

haxe NME under the spotlights,

haxenme.org/showcase

haxe NME vs The World

@elsassph

Part 3:

Options for

"Native" languages

Mono-platform *(nowadays?)*
Obj-C (Cocos2D...), Java (LibGdx...)

Cross-platform
C++ (Marmalade, Cocos2d-X...)

Native is not productive.

"Interpreted" languages

Cross-platform VM

LUA (Corona, Moai...),
Javascript (Phonegap, Spaceport.io...),
AS (ScaleForm),
etc.

Easy to start, hard to scale.

"Cross-compiled" languages

AOT/cross-compilation

C# (Unity3D, MonoGame...),
AS3 (Adobe AIR, Stencyl...)
haxe (NME).

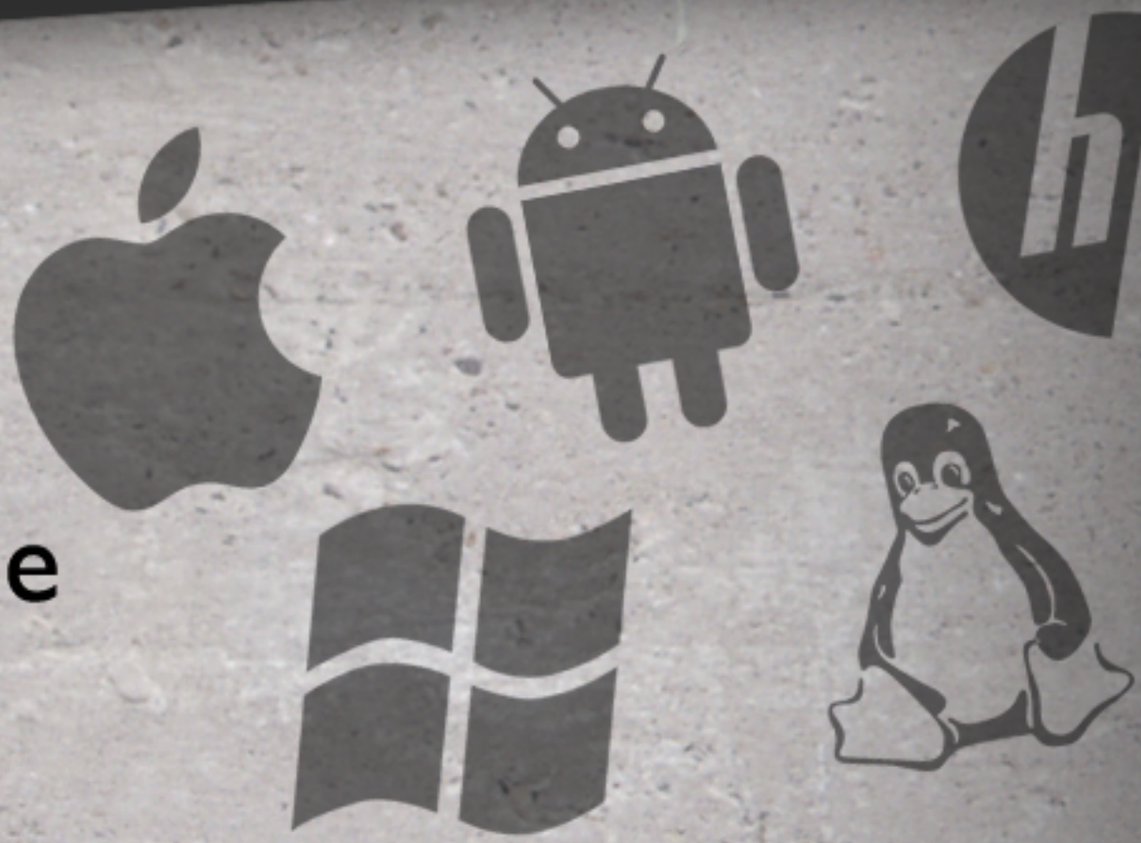
Sweet spot, if performance follows!

Choose your poison

Is your tech...

- performant on mobile?
- commercial? pay for beta?
- open source?
- extensible ?
- both desktop and mobile friendly?
- alive? *(beware of startups)*

Choose your poison



haxe NME is:

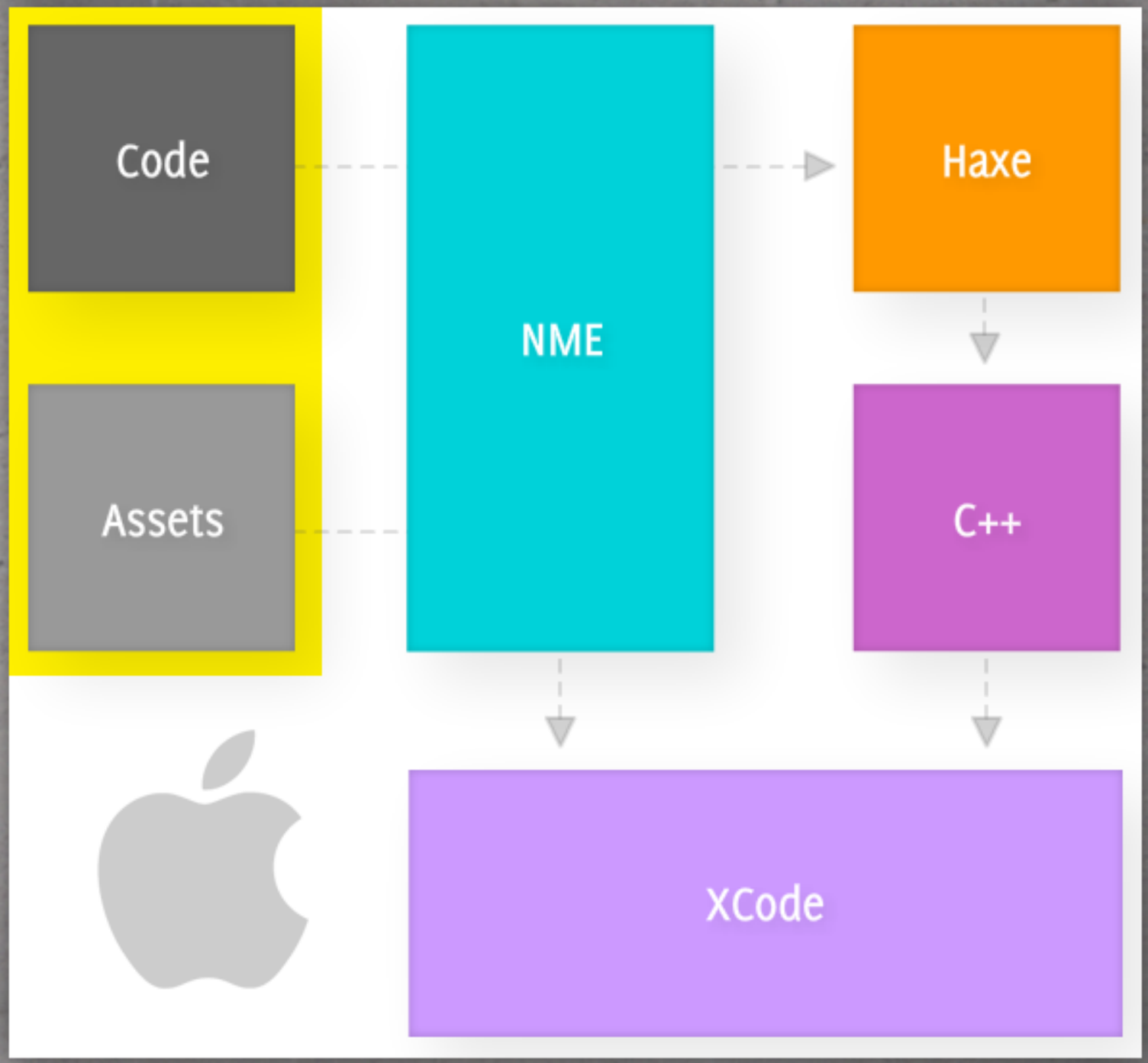
- performant on mobile
- completely free
- open source
- extensible
- both desktop and mobile friendly
- alive!

Part 4:

Inside the NME toolbox

"You are there"

The iOS workflow:



NME is... the framework

Cross-platform, ***Flash-like API***:

ex: nme.display.Sprite
nme.events.JoystickEvent

Cross-platform ***assets loading***:

ex: Assets.getBitmap("imgs/logo.png")

NME is... a command line tool

Configuration

$ nme setup android

Code generation

output templates, assets embedding,...

Compilation

NME is... entirely hackable

Framework, tools, templates:

0% black magic

100% open source

Part 5:

- Holy GPU, Batman!

said Rob-C++

Stick to GPU rendering

***Avoid software*-bound features**
(copyPixels, filters, masks, blendModes,...)

***Use images*, and batching**
(ie. bitmaps, bitmapFills, spritesheets)

Batching? Spritesheets?

Problem

GPU is fast, but CPU => GPU is slow

Solution

Give a lot of work at once to the GPU

Now do you homework

Holy GPU, Batman!

NME's secret ingredient

```
var spritesheet = Assets.getBitmapData("assets/spritesheet.png");
var tilesheet = new Tilesheet(spritesheet);
tilesheet.addTileRect(new Rectangle (0, 0, 32, 32));

var drawList = new Array<Float>();
drawList[0] = x;
drawList[1] = y;
drawList[2] = 0; // rect index
drawList[3] = scale;
drawList[4] = rotation;
drawList[5] = alpha;

graphics.clear();
tilesheet.drawTiles(graphics, drawList, smooth,
    Tilesheet.TILE_SCALE | Tilesheet.TILE_ROTATION | Tilesheet.TILE_ALPHA);
```

Tilesheet

the fastest path to render stuff en masse

Holy GPU, Batman!

NME's secret ingredient

```
var spritesheet = Assets.getBitmapData("assets/spritesheet.png");
var tilesheet = new Tilesheet(spritesheet);
tilesheet.addTileRect(new Rectangle (0, 0, 32, 32));

var drawList = new Array<Float>();
drawList[0] = x;
drawList[1] = y;
drawList[2] = 0; // rect index
drawList[3] = scale;
drawList[4] = rotation;
drawList[5] = alpha;

graphics.clear();
tilesheet.drawTiles(graphics, drawList, smooth,
    Tilesheet.TILE_SCALE | Tilesheet.TILE_ROTATION | Tilesheet.TILE_ALPHA);
```

MOAR OF THAT!

Note: it also supports RGB coloring and additive blendmode.

Holy GPU, Batman!

Dude, that's really low level!

Like Flash,
NME provides a ***low level API***.

Like Flash,
code or ***use a library***.
hint: NME inventor also makes 'gm2d'

Holy GPU, Batman!

I'm sold, anything I should know?

There are limitations...

- OpenGL ES 1.1, no 3D *(think Cocos2D)*
- No video or webviews *(for now)*
- Very limited HTML in TextFields

I'm sold, anything I should know?

...and stuff in-progress

- SWFs rendering
- OpenGl ES 2.0
- Blackberry target
- friendlier native extensions
  (ads, analytics, game centers,...)

Native Extensions <3

Some good resources

- a sample extension on haxenme.org
- nmex on google code
- nme-extensions on google code

Not that hard, really!
(you should learn C++ anyway)

Part 6:

What about the web?

Flash target

Great for interactive ***debugging***, ***quick iterations***, demos...

But...

- no Stage3D support
- limited Tilesheet support

HTML5 target (Jeash)

Exciting and improving quickly

- source-mapping*
- smaller footprint

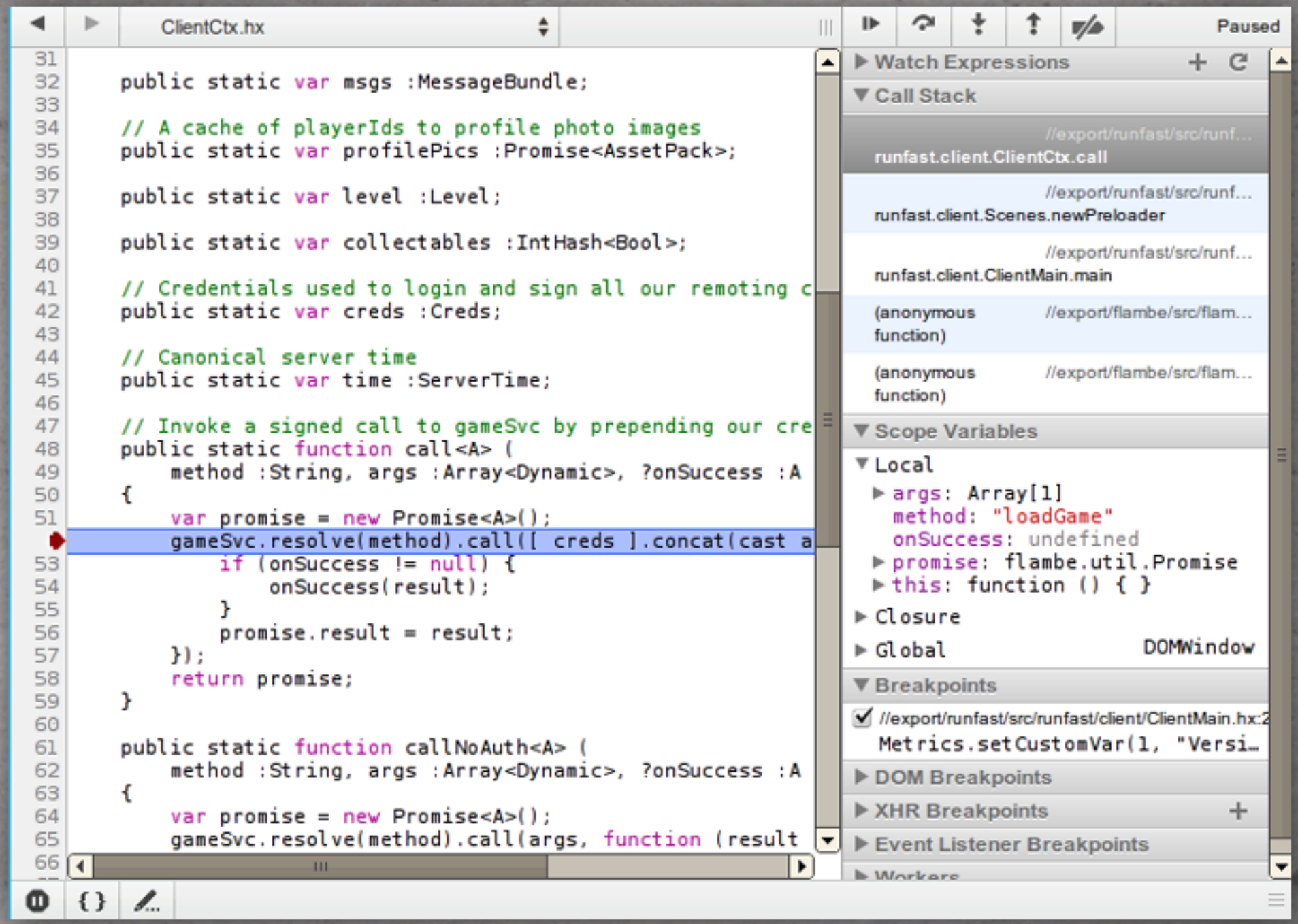
But...

- DOM+Canvas 2D only
- No Tilesheet, no drawTriangles

HTML/Flash

WTF is source-mapping?

*Look! haxe code in Google Chrome here:



@elsassph

HTML/Flash

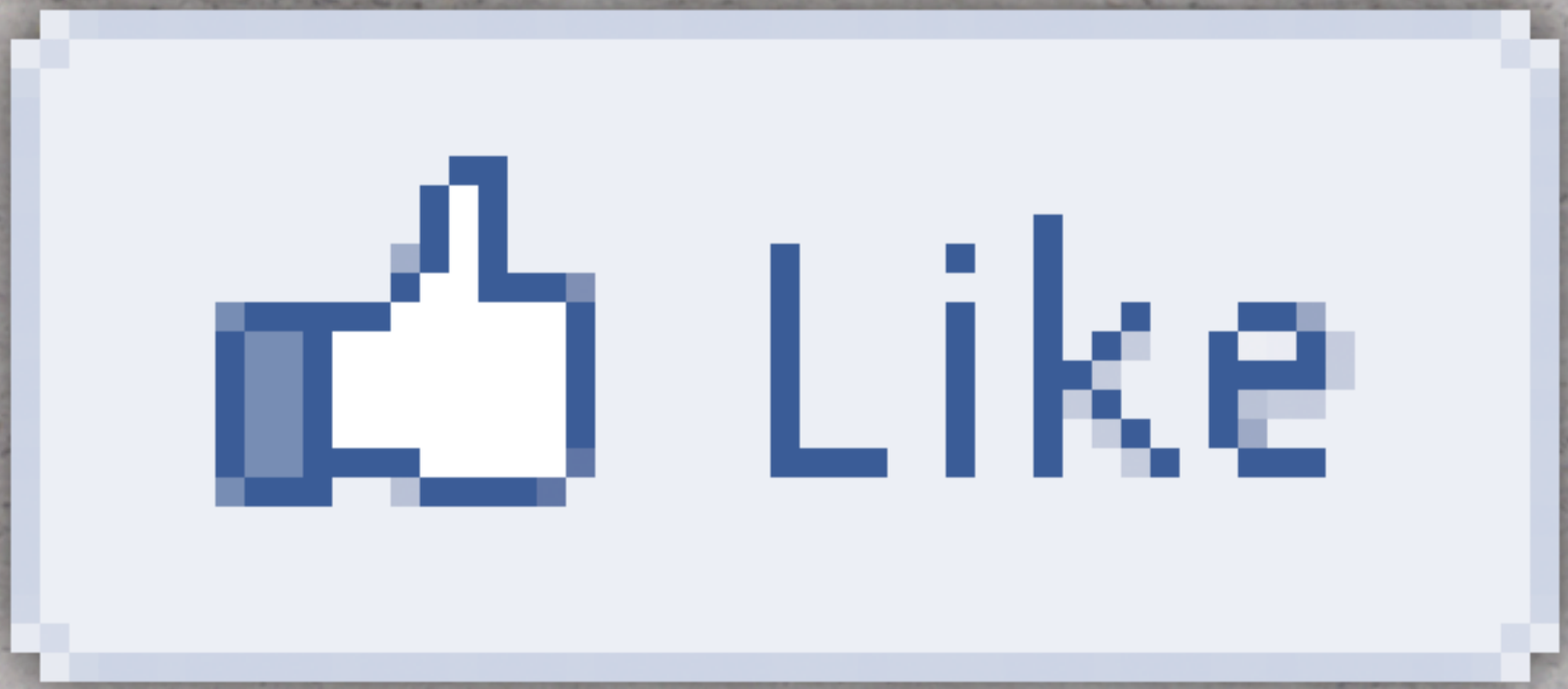
Future of the web targets (imho)

NME needs a web-friendly batching API

HTML5/Jeash has a bright future, contribute!

Is ***Flash*** worth our efforts?

Make it yours!



haxe NME now!