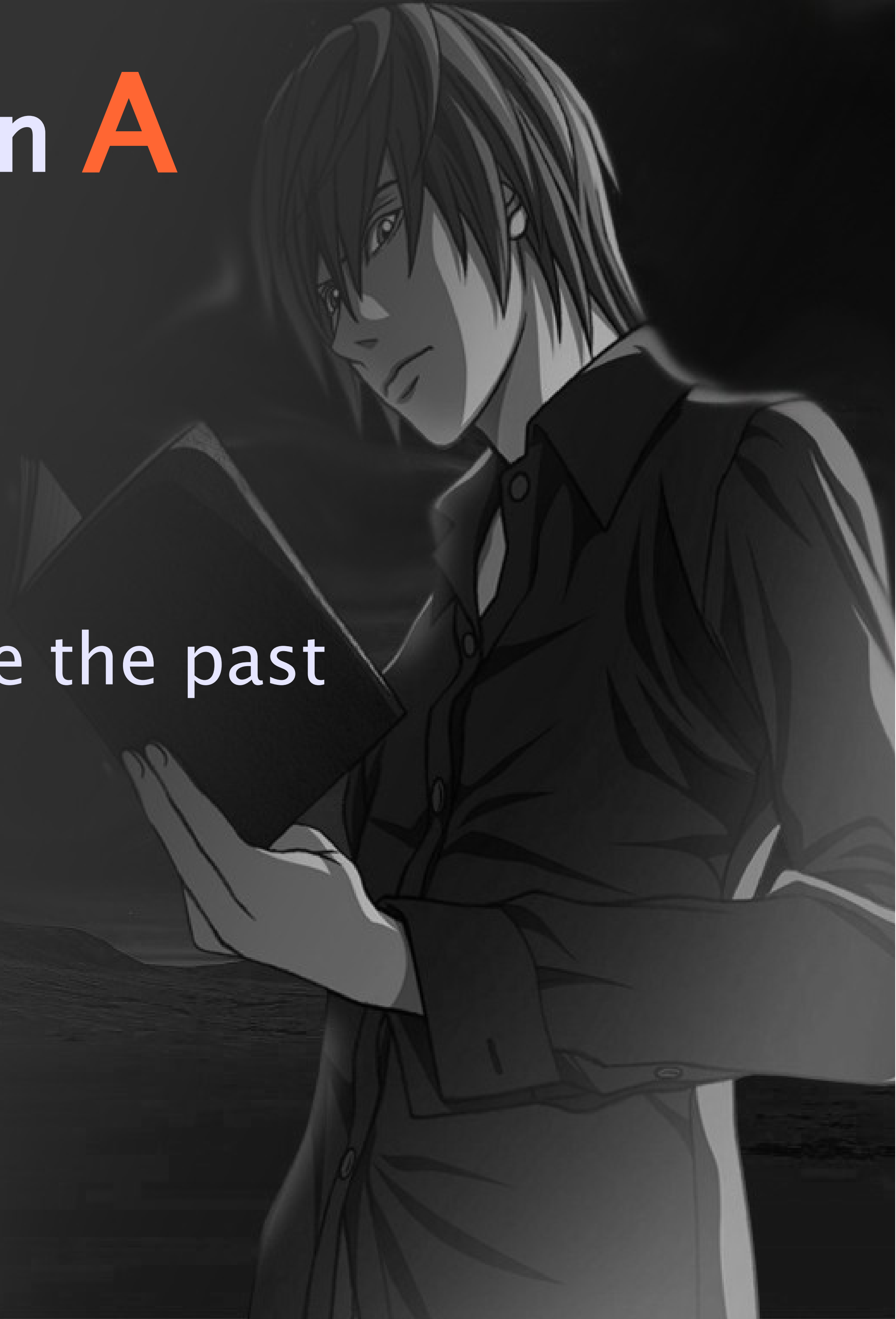
Haxe 3 & 4

Plans for World Domination

Plan A

✓ Analyze the past



"One Year of Haxe"

- ✓ We got momentum !
- ✓ We got NME !
- ✓ We got a new haxe.org website !
- ✓ We got many improvements !
- ✓ ... we still have a long way to go

Plan B

- ✓ The Future is NOW

Time for 2.09 !

- ✔ Now available on haxe.org !
- ✔ Includes "Compilation Server"

  ```
  haxe -v --wait 880
  haxe --connect 880 myproject.hxml
  ```

  - ✔ compilation time x2-x3
  - ✔ completion time x10 !!!
  - ✔ ready for future usages !

New in 2.09 !

✓ Optional Structure Fields

```
typedef Field = {
  name : String,
  ?type : String,
}


function foo( f : Field ) { .... }
foo({ name : "bar" });
```

New in 2.09 !

- ✓ JSON Fields Notation

```
var x = {
  "hello" : "World",
  "$cmd" : "PUSH",
};
```

New in 2.09 !

- ✓ New D-C-E :

```
haxe --dead-code-elimination
```

- ✓ Reduce a lot code size

No compiler cache

New in 2.09 !

✓ Smart Array Inference :

```
var a : Array<BaseClass> = [
    new MySubClass1(),
    new MySubClass2(),
];
```

New in 2.09 !

- More standard libs !
- haxe.Utf8
  - manipulate all strings as UTF8
- haxe.Json
  - use native in JS(auto-detec) / Flash11+
  - can force Haxe std impl. with -D haxeJSON

New in 2.09 !

✓ Reflect.getProperty / setProperty

```
class MyClass {
  public var x(getX,setX) : Int;
  ...
}
var inst = new MyClass();
Reflect.getField(inst,"x");
Reflect.getProperty(inst,"x");
```

New in 2.09 !

- new `sys.io` and `sys.net` packages
  - File, Process, Host, Socket...
- new `Sys` and `sys.FileSystem` classes
  - print, args, env, cwd, command, stdin/out/err...
- Ready for Haxe 3 transition
  - using `-D haxe3`

New in 2.09 !

- Javascript Improves !
- Source mapping with -debug
- Modern mode with --js-modern
    - `"use strict";`
    - `(function() {})()` wrapper
    - Global exposure with `@:expose`

New in 2.09 !

✓ Flash 11.2 and 11.3 beta api

✓ Support for binary files :

```
@:file("myfile.dat")
class MyFile extends ByteArray {
}
```

  ✓ And sounds :

```
@:sound("mysound.wav")
class MySound extends Sound {
}
```

✓ 'flash9' dir becomes 'flash'

New in 2.09 !

- ✓ And much more....
  - ✓ many fixes for cpp,php,macros
  - ✓ SPOD 'in' operator, by-relation searchs, SQLite
  - ✓ Type.allEnums
  - ✓ more JQuery
  - ✓ haxe.web.Request for neko/php/js
  - ✓ opt. args in function type `(?Int -> Void)`
  - ✓ \uXXXX in regexp
  - ✓ ....

Plan C

✓ Plan for Battle

Haxe 3.0

- ✓ Major release == allow breaks

  (unless it's smart to avoid)

- ✓ Guidelines :

  - ✓ simplify, avoid feature-bloating
  - ✓ make it easy(ier) for crossplaform

    (both users AND compiler contributors)

Haxe 3.0

- ✓ What to consider when adding a feature ?
  - ✓ Well-defined behavior
  - ✓ Strictly typed
  - ✓ Can be efficiently emulated on all platforms
  - ✓ Good Usefulness / Complexity ratio

    (especially when it creates syntax)

Haxe 3.0

- String interpolation :

```
var p = new User();
"$p is ${p.getAge()} old"
```

  - now available with Std.format
  - will become the default for all Strings

Haxe 3.0

✓ Simplify properties :

```
class MyClass {
   var prop(get,set) : Int;
}
```

✓ Standardized get_prop/set_prop functions

(greatly simplify reflection)

✓ _prop/get_prop/set_prop automagically created

✓ no _prop when getter is **default**

Haxe 3.0

- ✓ Import improvements :

```
import my.pack.*;   // package
import js.JQuery.*; // statics
import haxe.Log.trace; // single
```

- ✓ Allow a `DefaultImports.hx`
  - ✓ can also exists in libs : `NmeImports.hx`
- ✓ `using` will imply `import`

Haxe 3.0

✓ Replacement for **`callback`** :

```
onClick = callback(clickElement,e);
```

become :

```
onClick = clickElement(e,_);
```

can also be :

```
onClick = clickElement(_,e);
```

Haxe 3.0

✓ More switch :

```
switch( getEnumValue() ) {
case A(x) if( x > 0 ):
case A(-1):
case B(C("hello"),false):
default(e):
}
```

Haxe 3.0

- ✓ Evolving the language is nice,

  but that's not all...

- ✓ New targets !

  @cwaneck Java/C#

- ✓ More libs / tools

  IntelliJ, NME 3+, new haxelib ?

- ✓ CONTRIBUTE !

Haxe 3.0

haXe becomes Haxe

Plan D

✓ And after 3.0 ?

Haxe Next

- ✓ Do we need MORE features ?
  - ✓ maybe, but...
  - ✓ with time : more libs, less language changes
  - ✓ let's think of Haxe globally

Haxe Next

- ✓ Long-term vision :
  - ✓ One single language
  - ✓ ALL mainstream platforms
  - ✓ ALL major IDE
  - ✓ Complete set of Tools
  - ✓ Complete Manual / Docs

Haxe Next

- ✓ Keep it open !
  - ✓ Open Source
  - ✓ Free
  - ✓ Community-based
- ✓ Still allow for companies involvement
  - ✓ Marketing IS necessary here
  - ✓ Requires more centralized process

Haxe Next

- The Haxe Foundation !
  - NonProfit
  - Open to individuals
  - Open to companies
  - Haxe Design still a Dictatorship

Thank You

@ncannasse