



Insomniac's Web Tools (a postmortem)

Andreas Fredriksson
Lead Engine Programmer
Insomniac Games



Hi, I'm Andreas





Hi, I'm Andreas

- I lead the tools & infrastructure team at Insomniac





Hi, I'm Andreas

- I lead the tools & infrastructure team at Insomniac
- “So you guys have web tools?”
 - #1 conversation opener





Our web tools exodus





Our web tools exodus

- Disclosure: We have abandoned web tools development





Our web tools exodus

- Disclosure: We have abandoned web tools development
- This talk is about what was good





Our web tools exodus

- Disclosure: We have abandoned web tools development
- This talk is about what was good
- ...And what wasn't so good





The 2010 Web Tools Vision





The 2010 Web Tools Vision

- Overhauling tools+engine for cross-platform tech reboot





The 2010 Web Tools Vision

- Overhauling tools+engine for cross-platform tech reboot
- Games having ever-bigger data sets
 - New focus on visualization and mining





The 2010 Web Tools Vision

- Overhauling tools+engine for cross-platform tech reboot
- Games having ever-bigger data sets
 - New focus on visualization and mining
- More UI innovation in the web space!
 - Compared to desktop space
 - Also true of mobile space

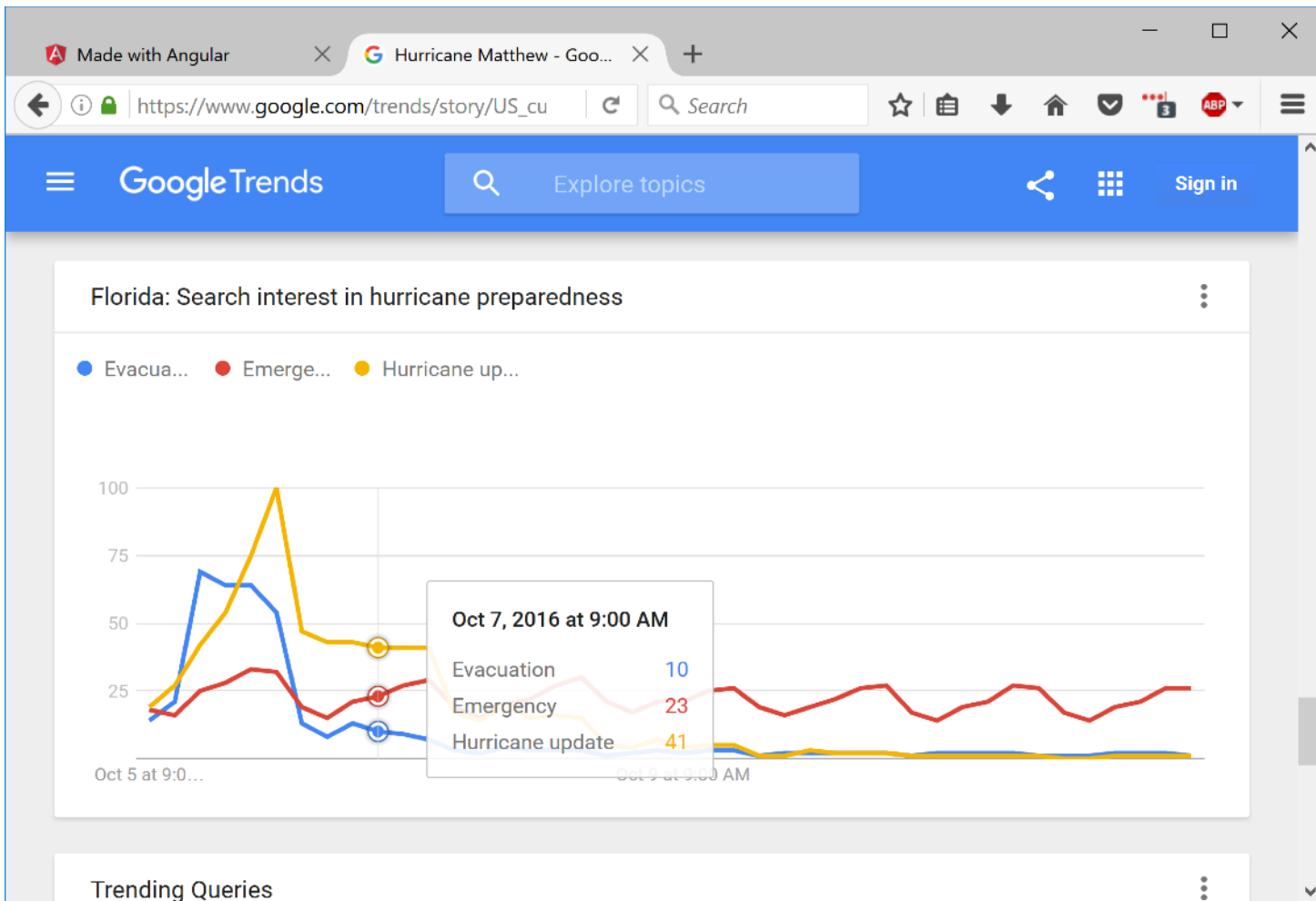




The 2010 Web Tools Vision

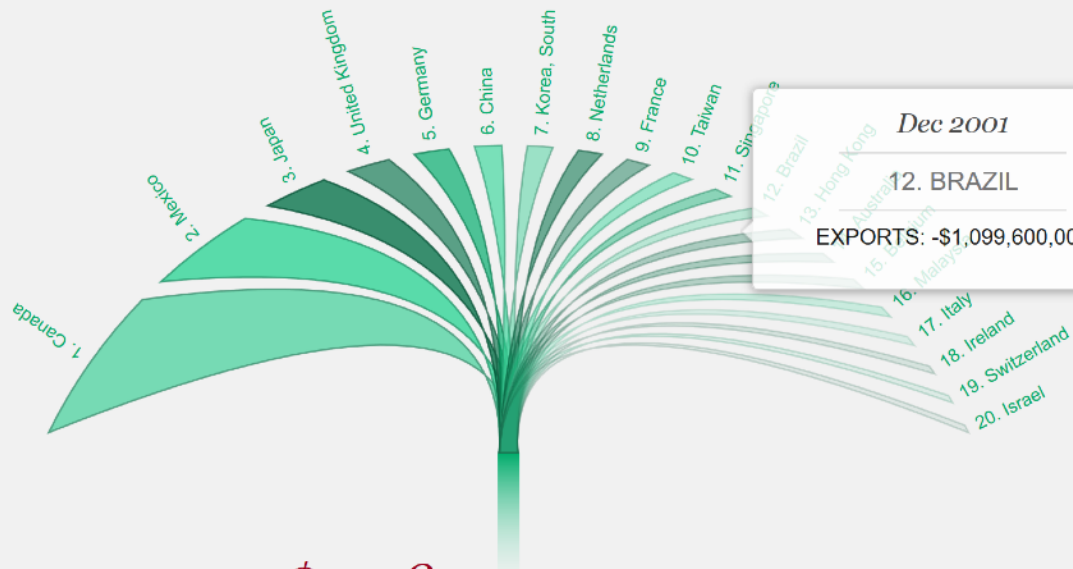
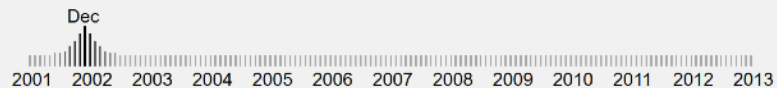
- Overhauling tools+engine for cross-platform tech reboot
- Games having ever-bigger data sets
 - New focus on visualization and mining
- More UI innovation in the web space!
 - Compared to desktop space
 - Also true of mobile space
- Strong team drive to try new things







click on a timeline year to skip to that point.



-\$4,118,975,000,000

* Cumulative trade imbalance since 2001

20. Sweden

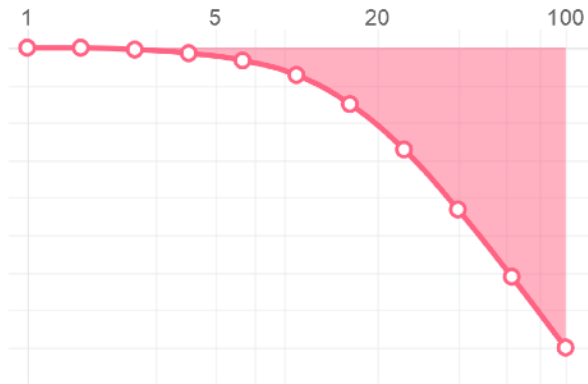
19. Belgium



www.chartjs.org



Search



New in 2.0

New chart axis types

Plot complex, sparse datasets on date time, logarithmic or even entirely custom scales with ease.

New in 2.0

Animate everything!

Out of the box stunning transitions when changing data, updating colours and adding datasets.



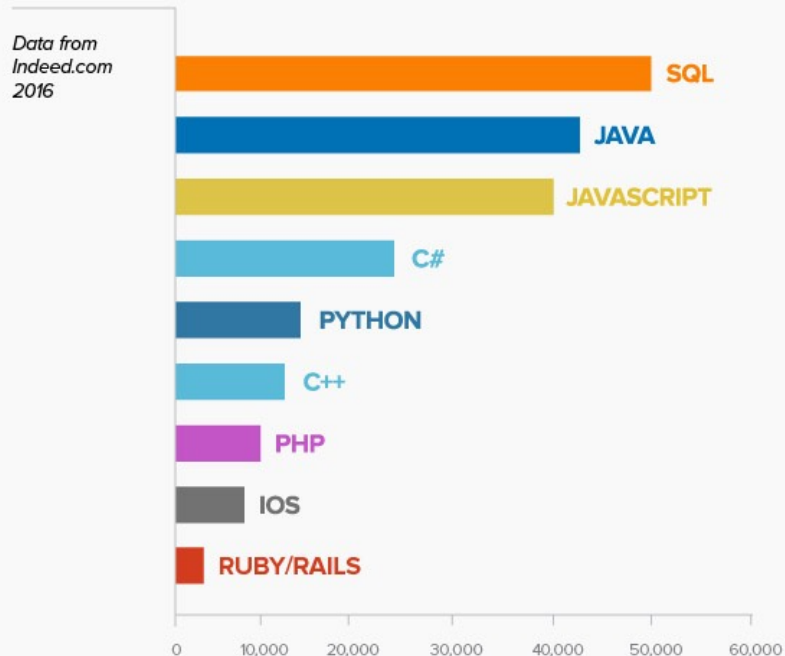


2010 vision, continued

- Hire awesome web people!

<http://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2016/>

Languages ranked by number of programming jobs





2010 vision: “Everyone can make a web page”

- Lower the barrier to entry!
- Empower the team to roll out simpler stuff whenever





2010 vision: “There’s tons of off the shelf web stuff”





2010 vision: “There’s tons of off the shelf web stuff”

- Lots, lots, lots out there





2010 vision: “There’s tons of off the shelf web stuff”

- Lots, lots, lots out there
- Open ecosystem





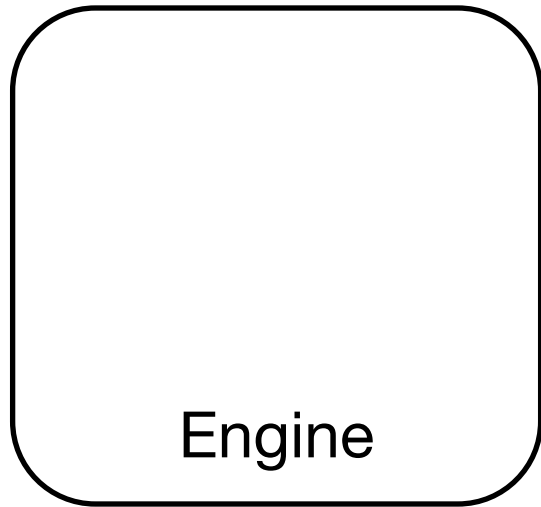
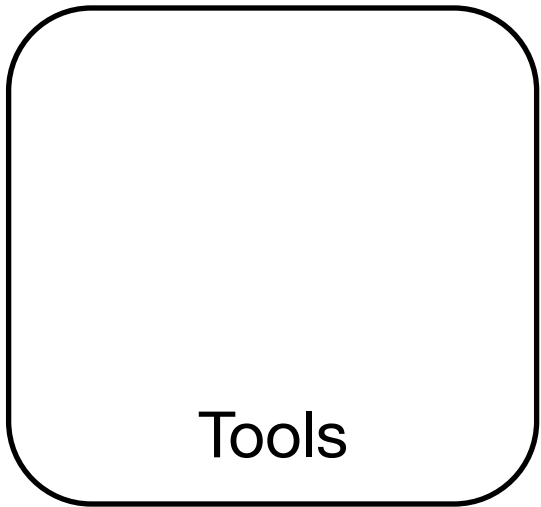
2010 vision: “There’s tons of off the shelf web stuff”

- Lots, lots, lots out there
- Open ecosystem
- Fewer wheels to reinvent



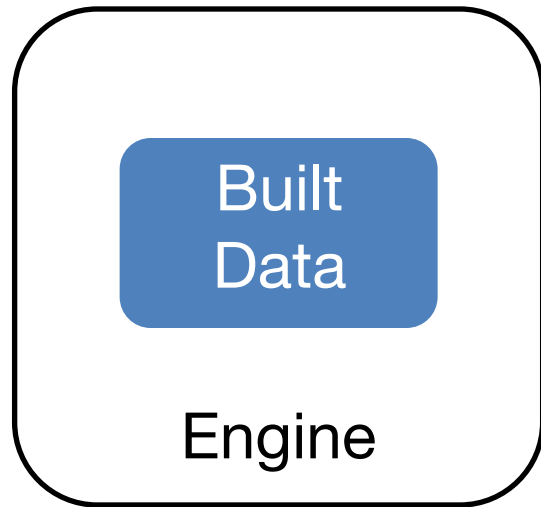
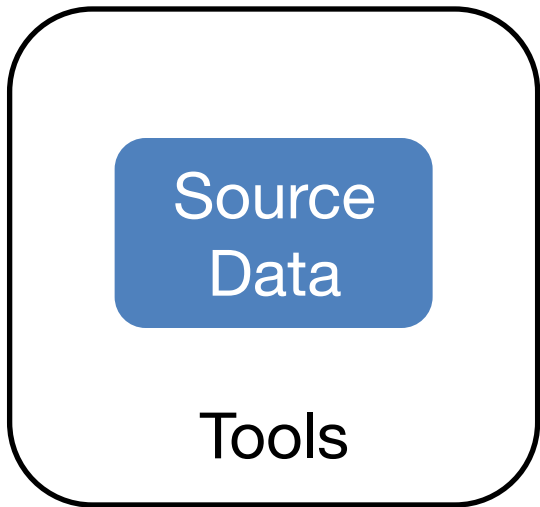


2010 vision: Enforce one-way data pipeline



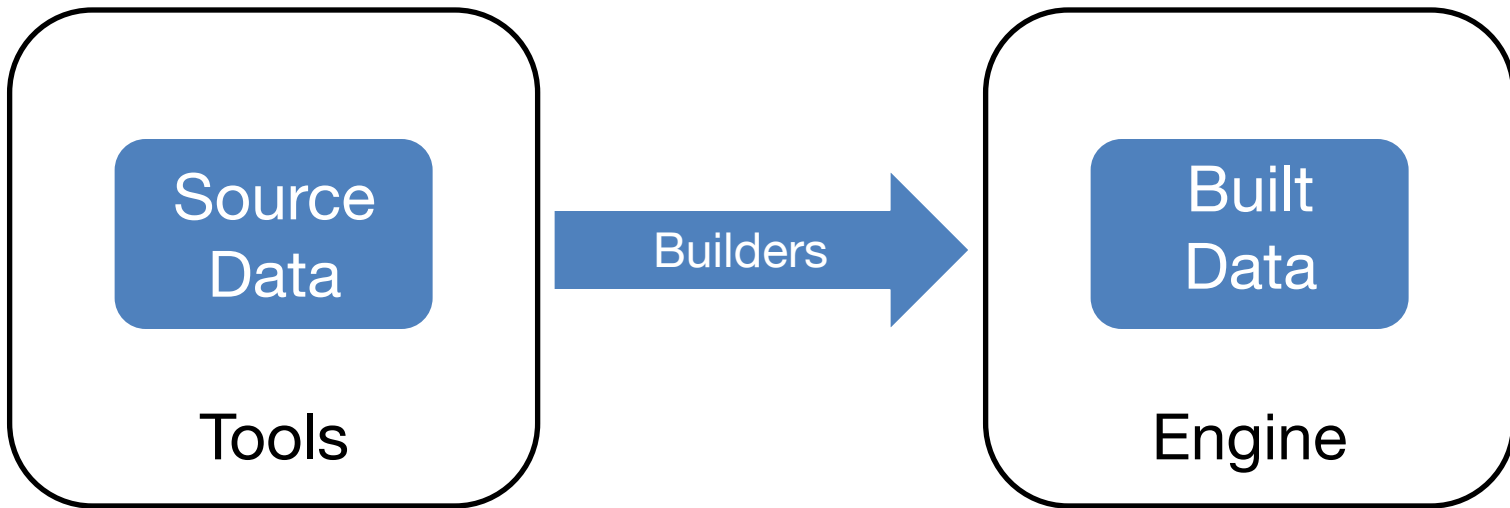


2010 vision: Enforce one-way data pipeline





2010 vision: Enforce one-way data pipeline





2010 vision: Web browser = Data Sandbox





2010 vision: Web browser = Data Sandbox

- Hard to call browser logic from engine





2010 vision: Web browser = Data Sandbox

- Hard to call browser logic from engine
- Hard to call engine logic from browser





2010 vision: Web browser = Data Sandbox

- Hard to call browser logic from engine
- Hard to call engine logic from browser
- Much easier to do the right thing instead (write a builder)





2010 grand vision

- Zero install tools
 - “Have a browser? You’re good to go.”





So that was the vision..





So that was the vision..

- ...and the team started building stuff





So that was the vision..

- ...and the team started building stuff
- Core Team = a bunch of C++ experts
 - With almost no web experience at this point





So that was the vision..

- ...and the team started building stuff
- Core Team = a bunch of C++ experts
 - With almost no web experience at this point
- We had 3-5 people working on web tools full time for 5 years





Early days, early decisions





Early days, early decisions

- Single browser: Chrome
 - Avoid the web compatibility/standards problem
 - Leverage Chrome team smarts - let them worry about browser!





Early days, early decisions

- Single browser: Chrome
 - Avoid the web compatibility/standards problem
 - Leverage Chrome team smarts - let them worry about browser!
- 3D native engine view?
 - Jam a native child window into Chrome!
 - Tiny amount of glue code in a plugin needed





Early days, early decisions

- Single browser: Chrome
 - Avoid the web compatibility/standards problem
 - Leverage Chrome team smarts - let them worry about browser!
- 3D native engine view?
 - Jam a native child window into Chrome!
 - Tiny amount of glue code in a plugin needed
- Where do web pages come from?
 - A server!



GDC

GAME DEVELOPERS CONFERENCE®

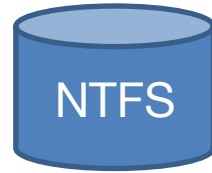
| FEB 27-MAR 3, 2017

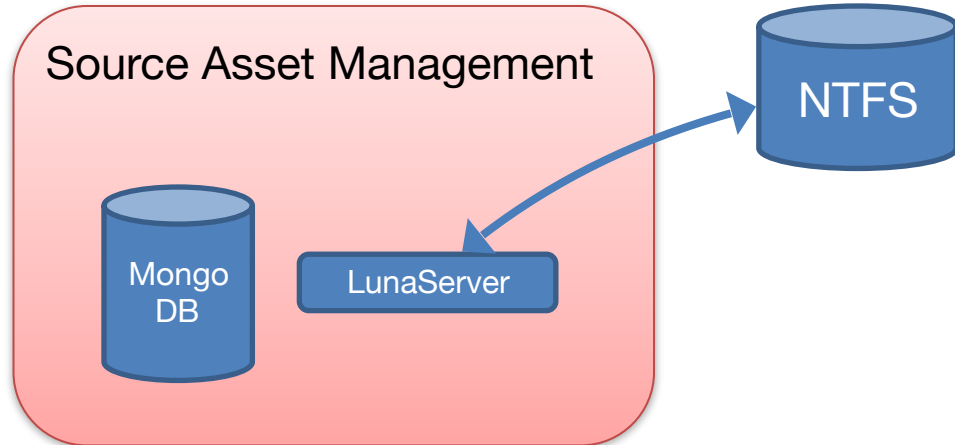
| EXPO: MAR 1-3, 2017

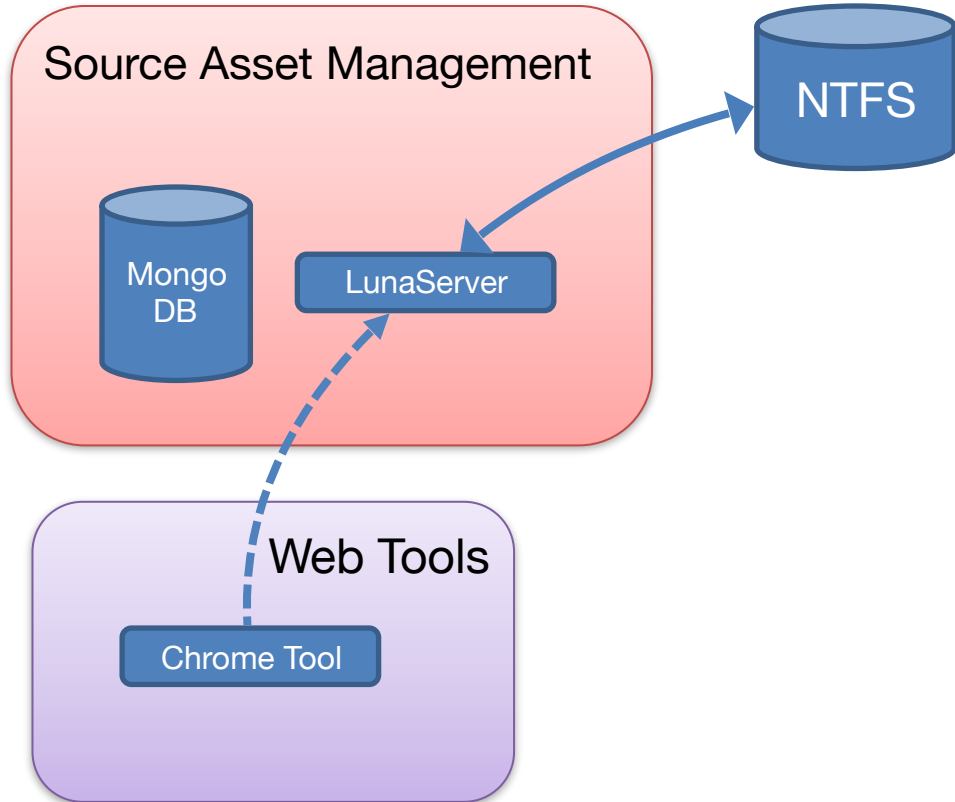
#GDC17

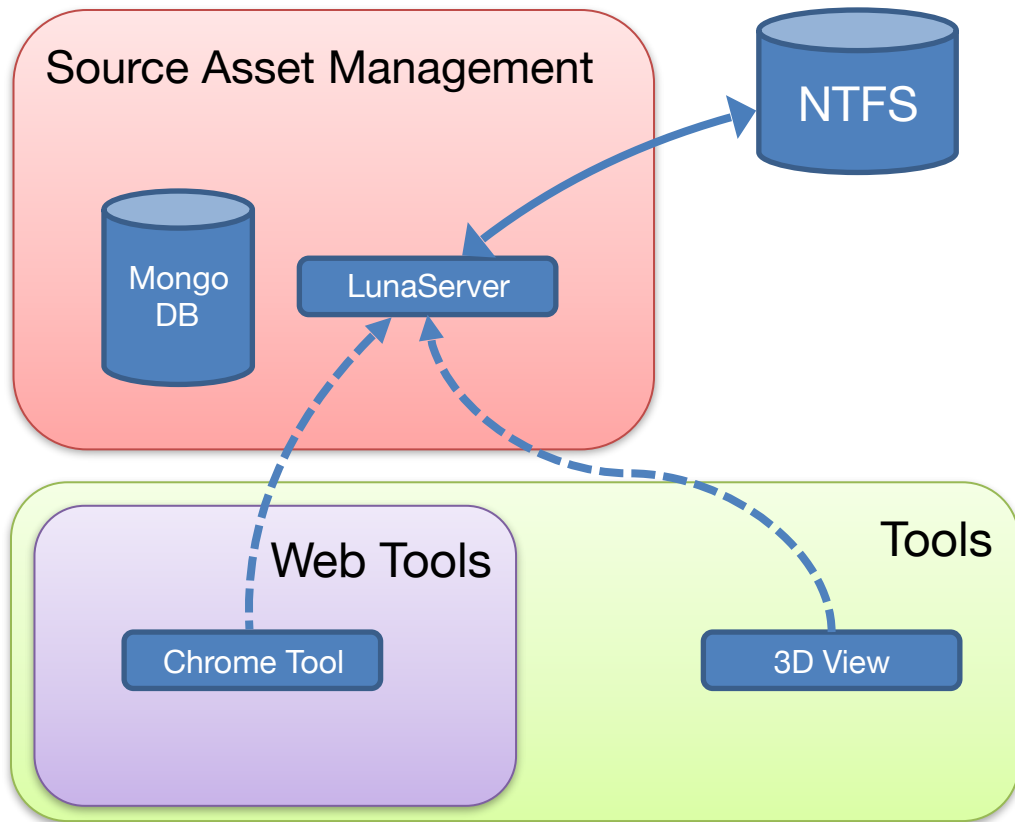


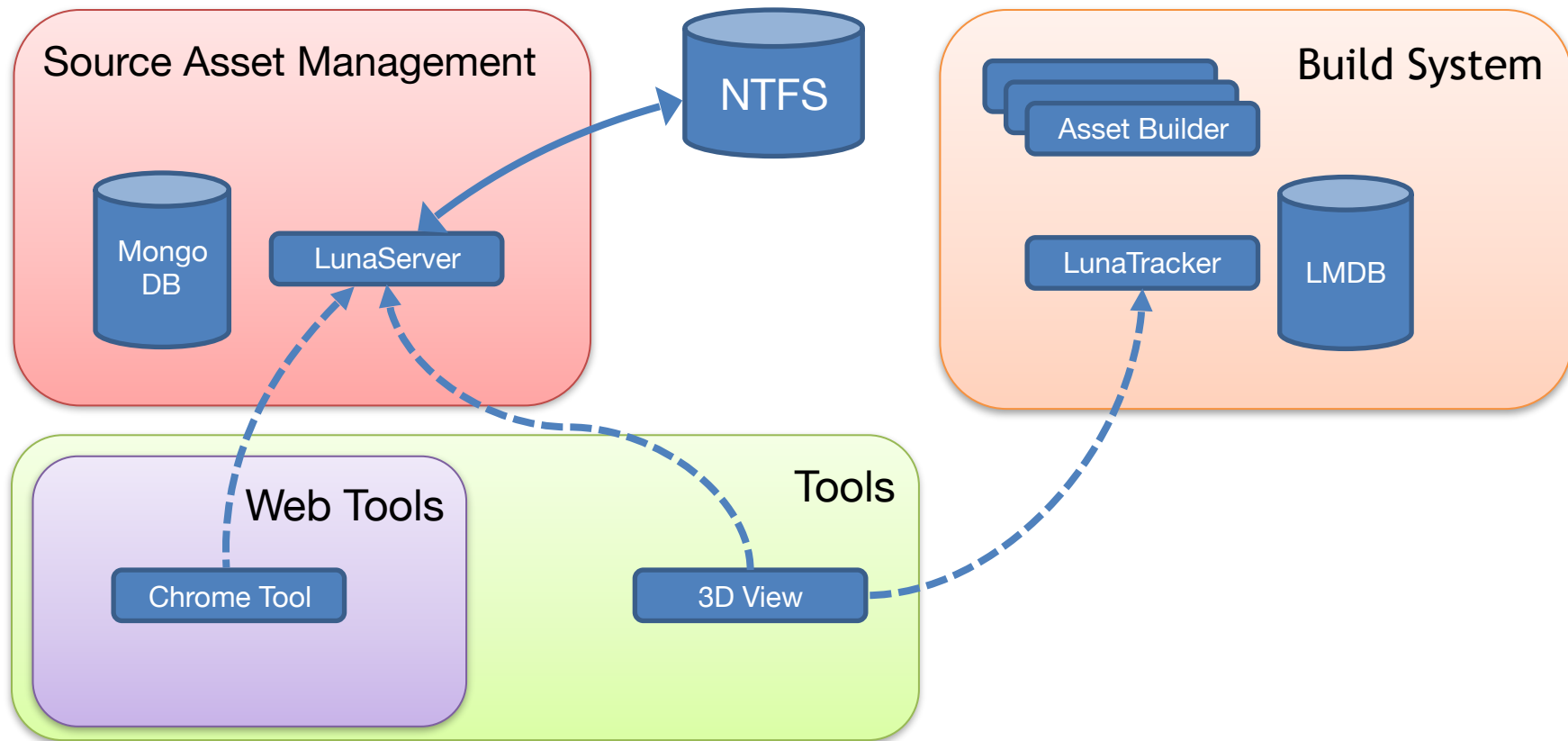
UBM

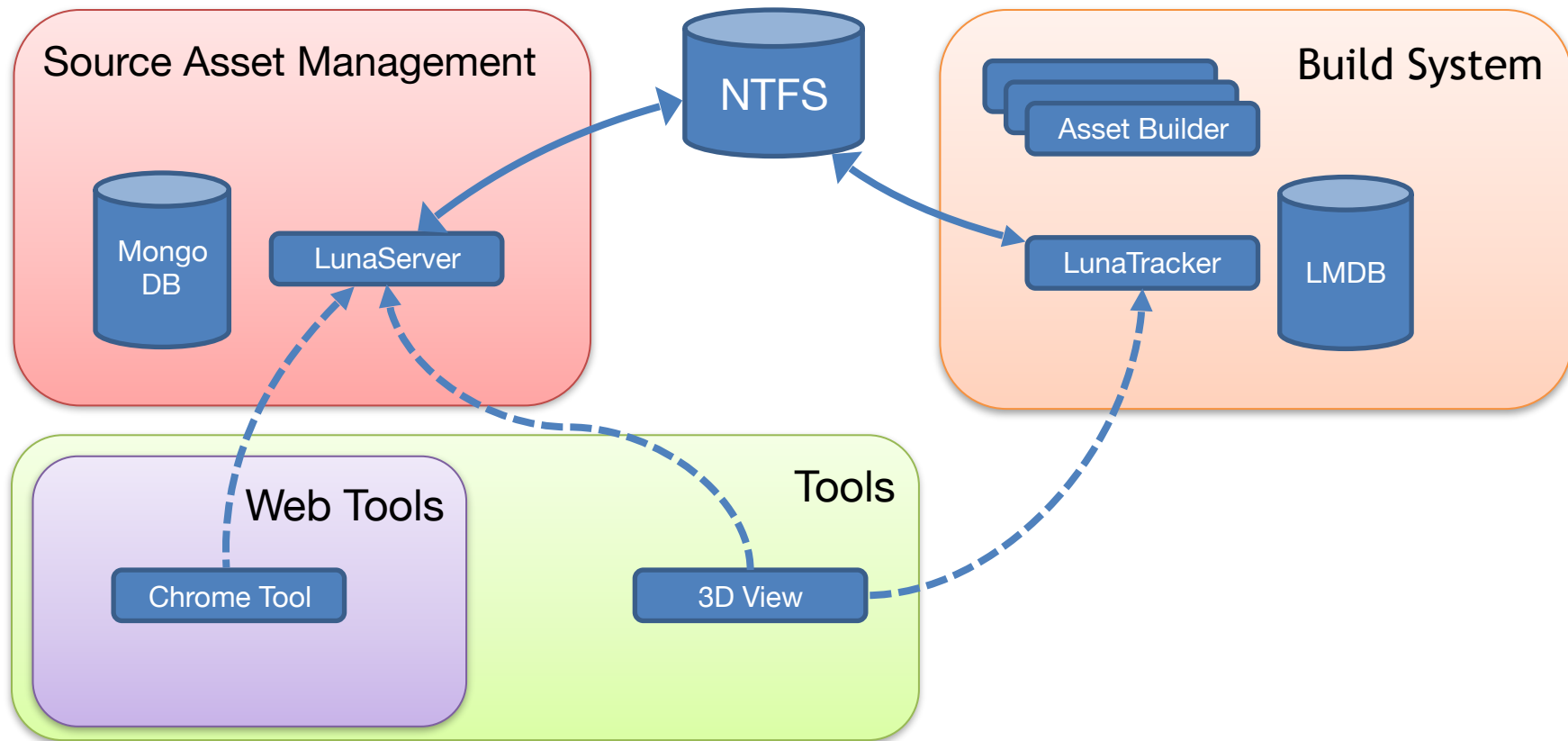


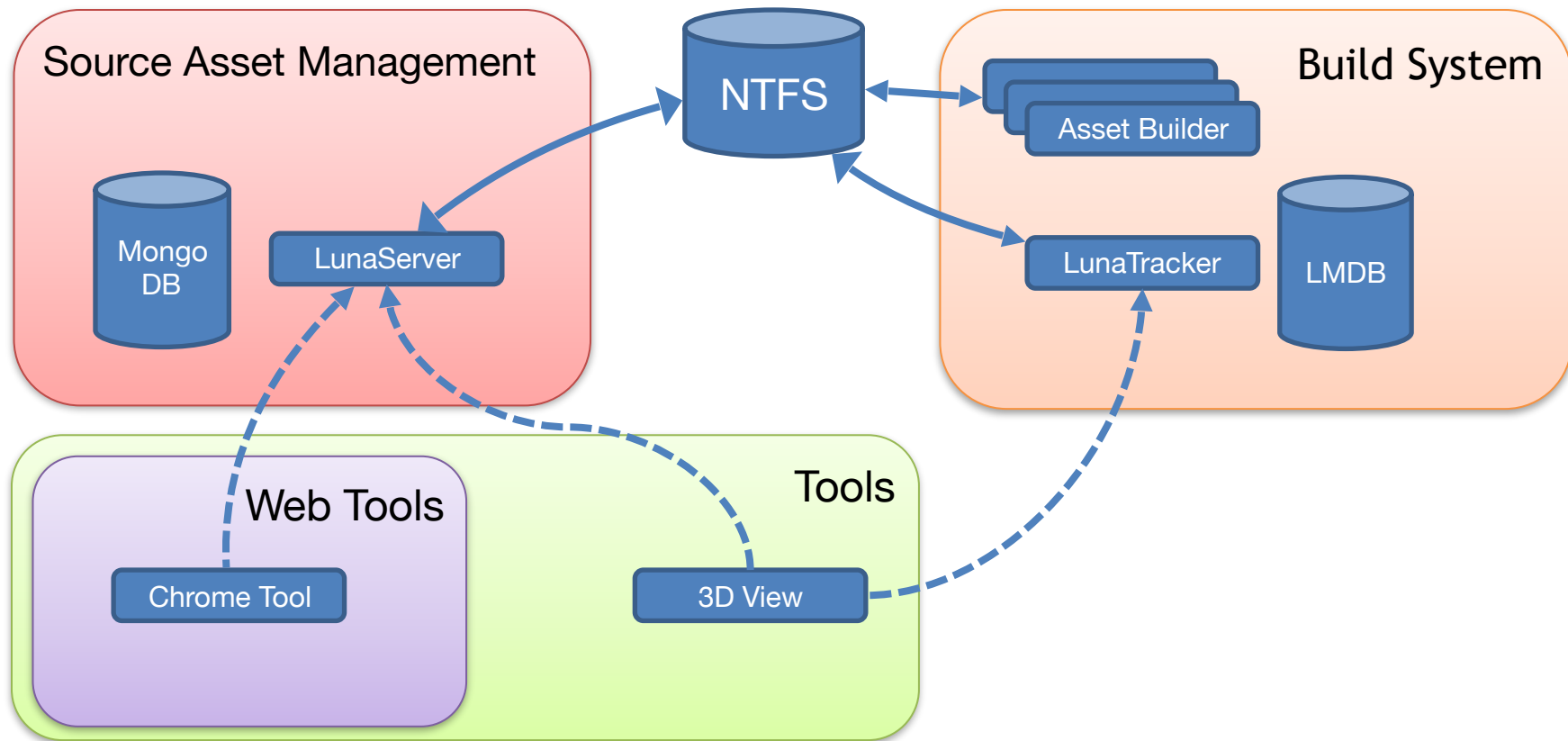


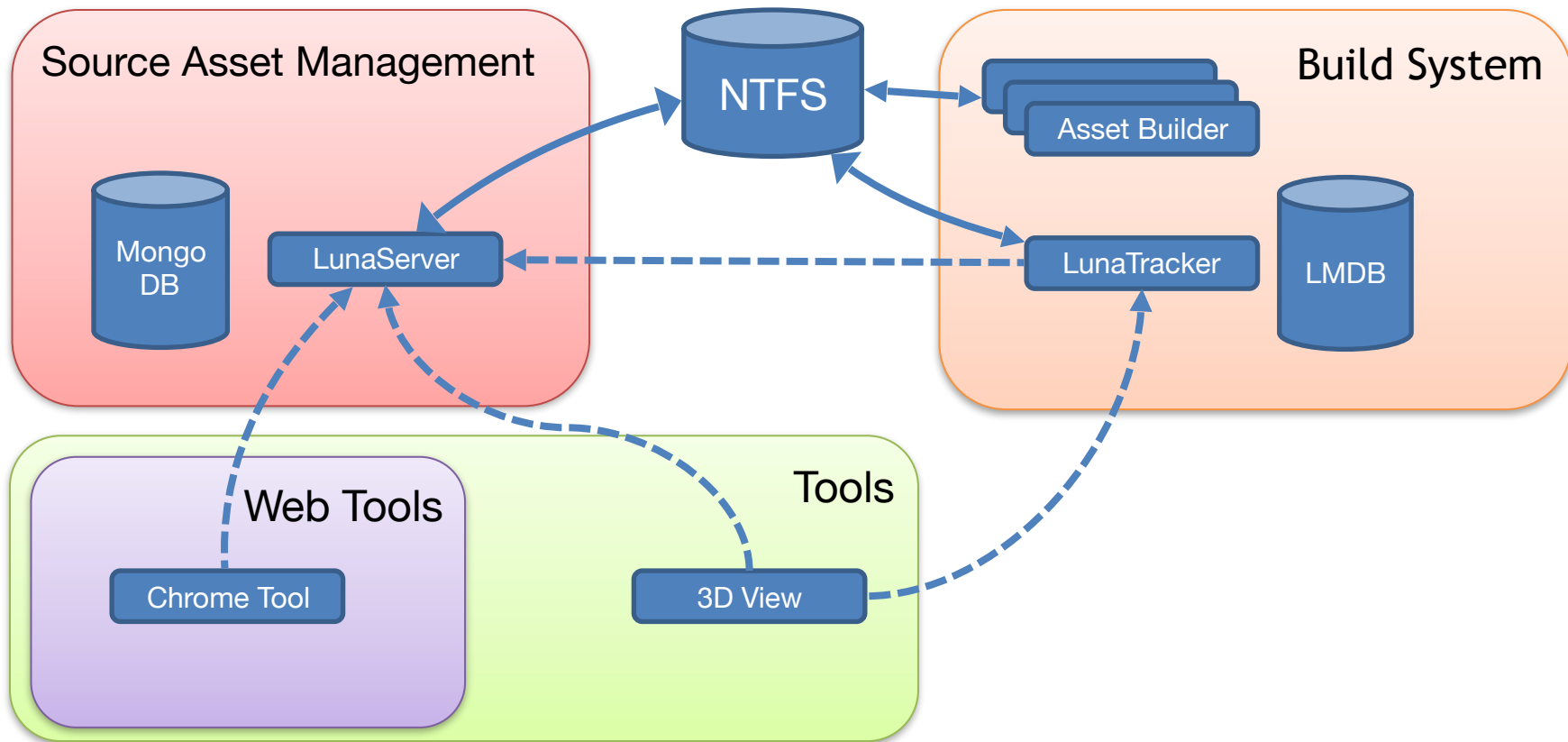


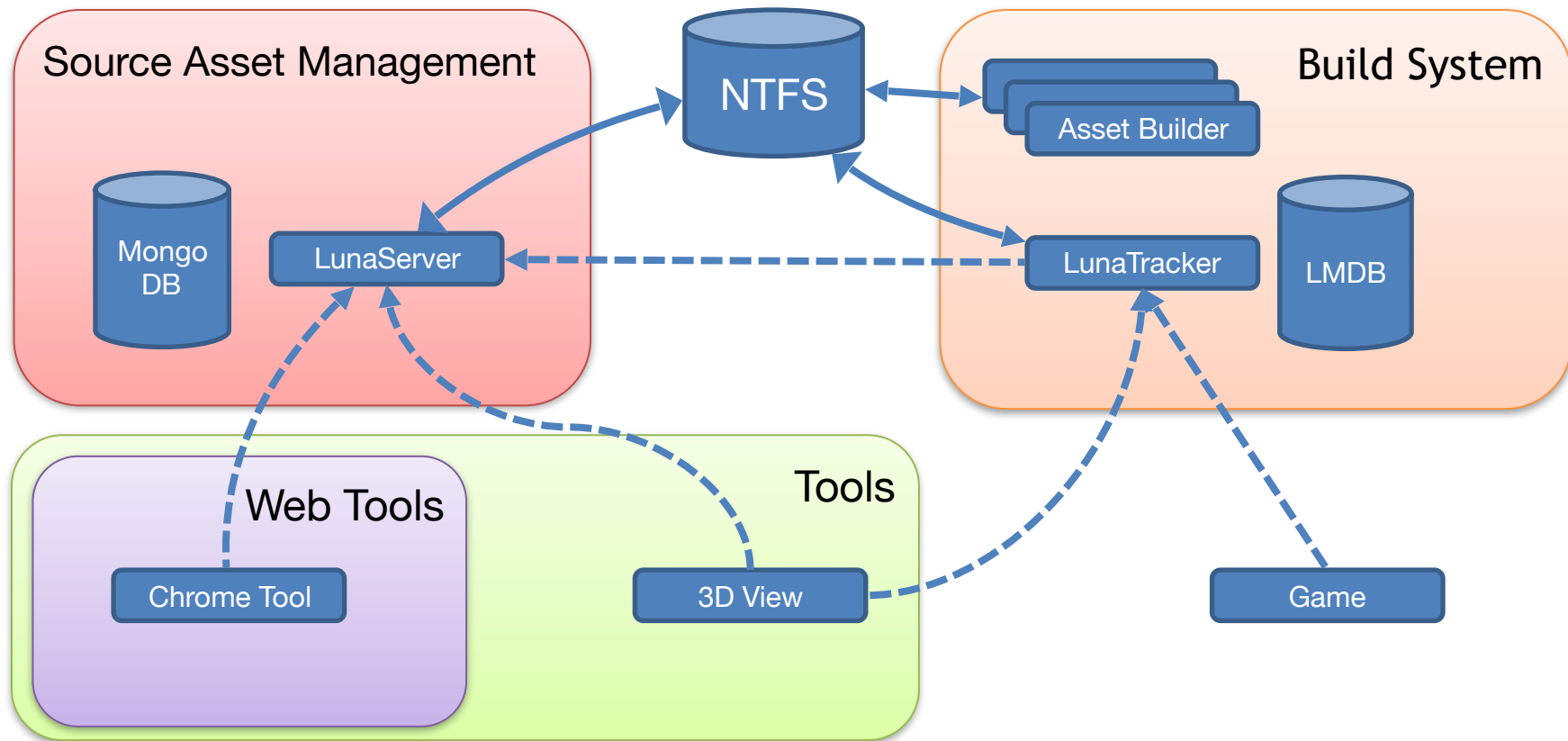














Vision meets reality: Early tradeoffs





Vision meets reality: Early tradeoffs

- Desktop look and feel
 - Early UX tests revealed dislike of “odd stuff”
 - Traditional concepts like menu bars strongly desired





Vision meets reality: Early tradeoffs

- Desktop look and feel
 - Early UX tests revealed dislike of “odd stuff”
 - Traditional concepts like menu bars strongly desired
- Has to work *together* with Maya, Photoshop, ...
 - It's not a web app on a deserted island, it exists side by side





Vision meets reality: Early tradeoffs

- Desktop look and feel
 - Early UX tests revealed dislike of “odd stuff”
 - Traditional concepts like menu bars strongly desired
- Has to work *together* with Maya, Photoshop, ...
 - It's not a web app on a deserted island, it exists side by side
- “We can't do that because it's a web app” not a valid excuse

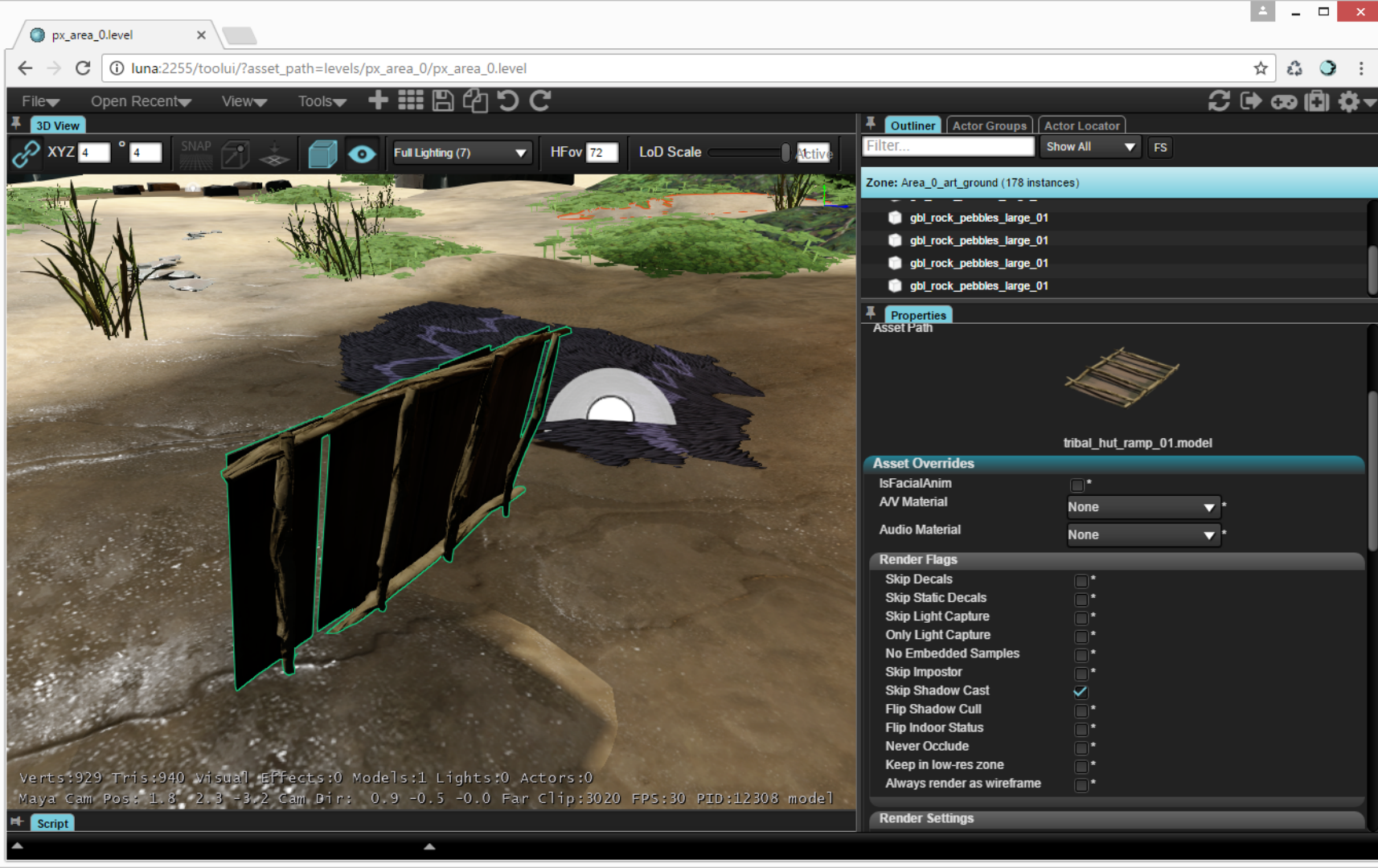


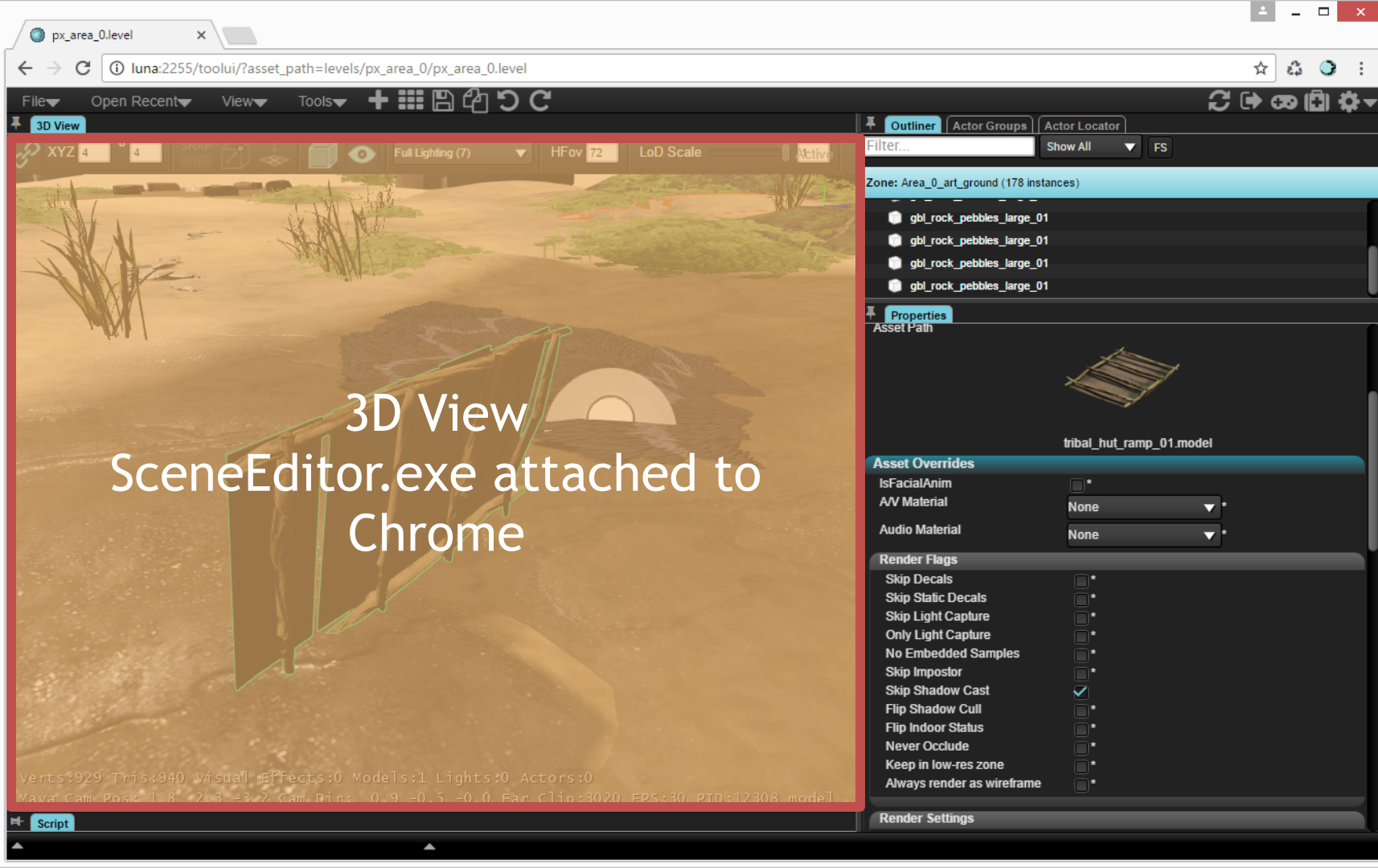


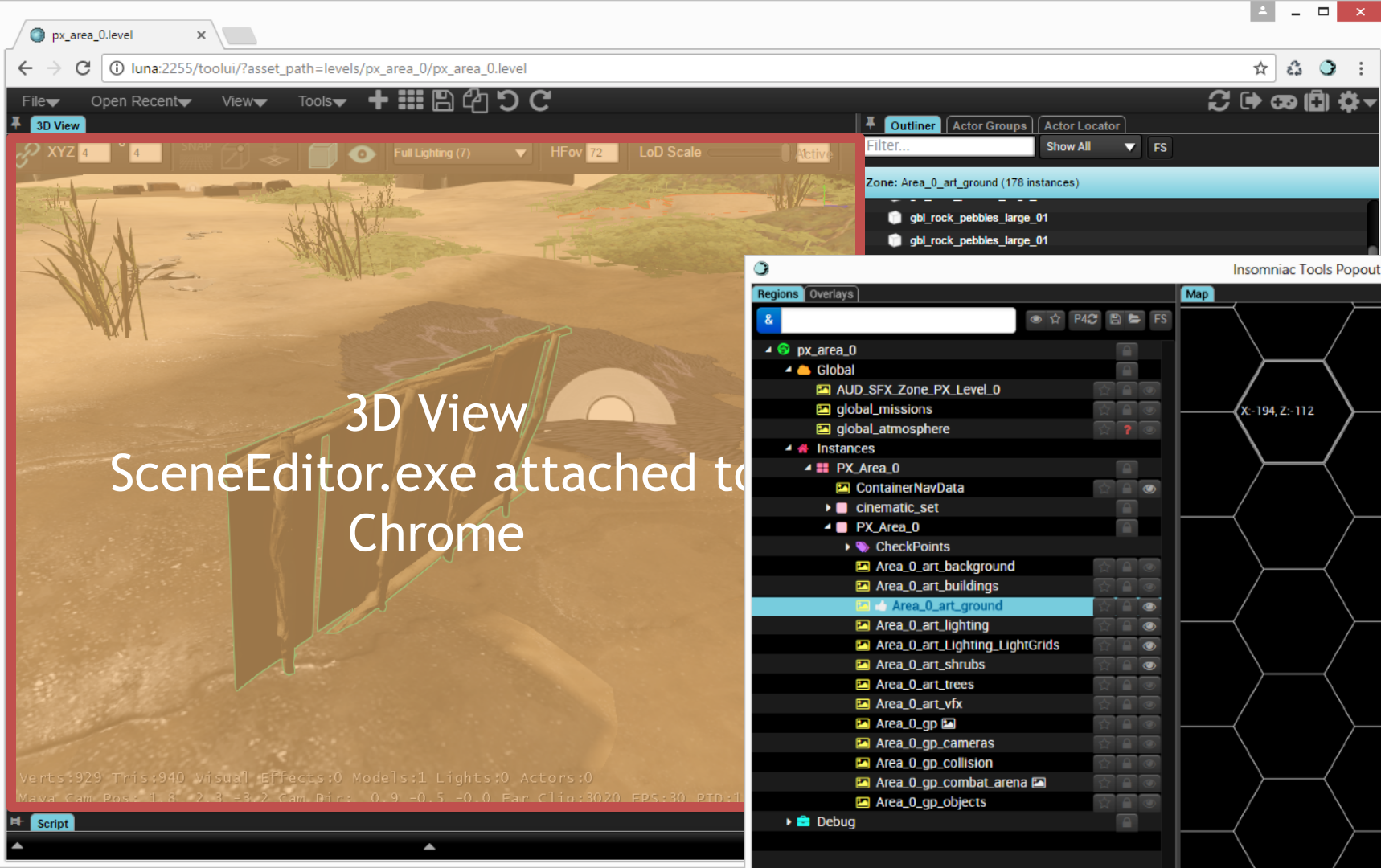
Our web tools: A gallery

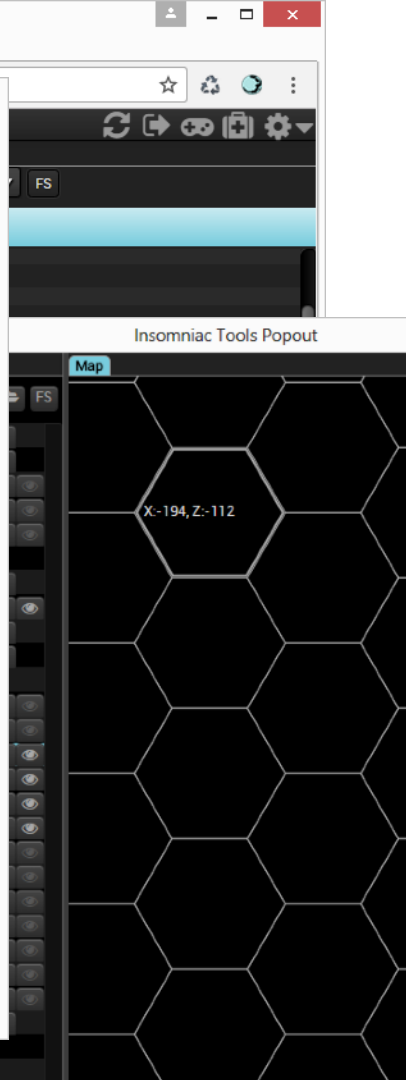
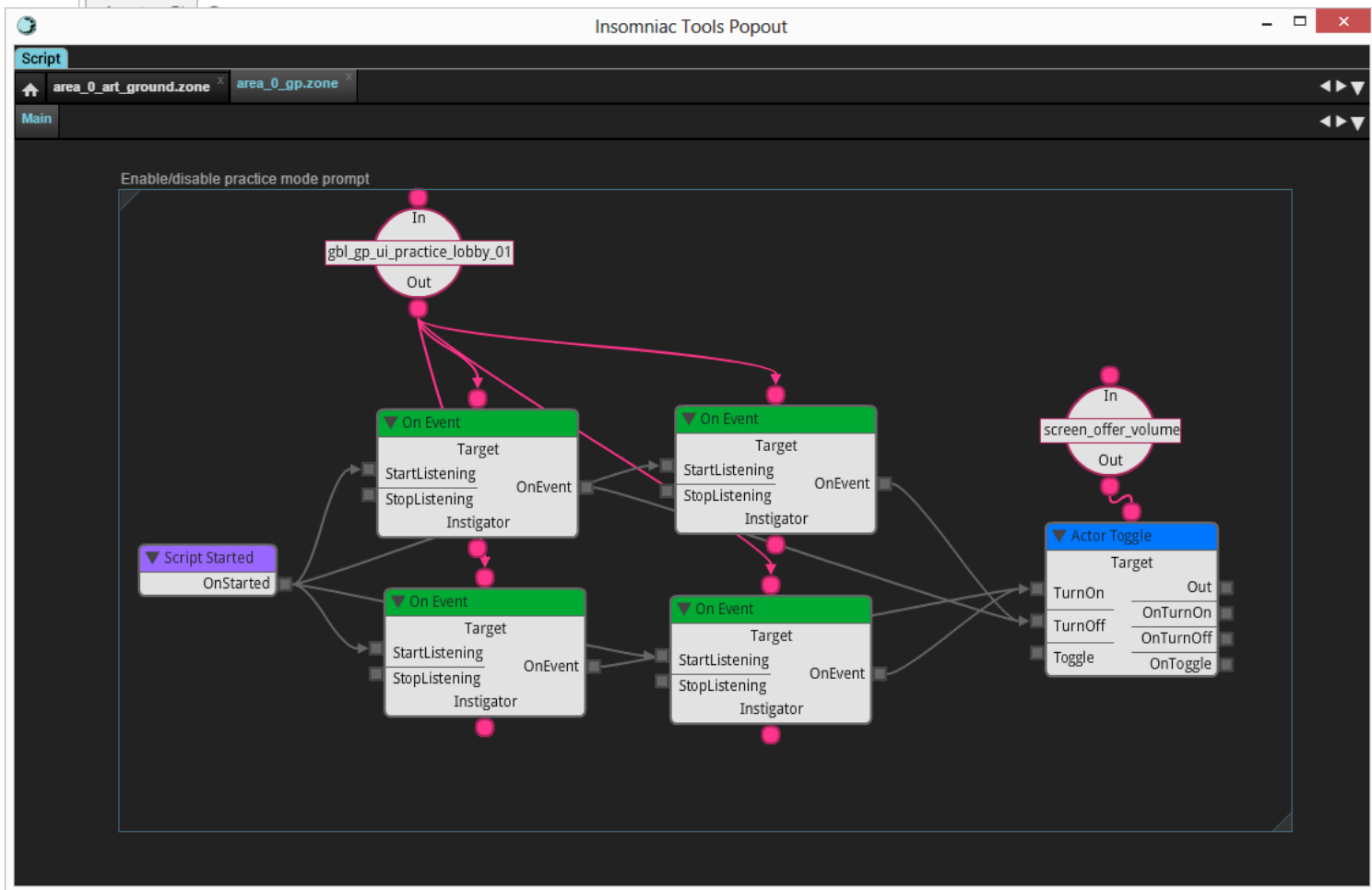
- *A complete AAA* tool suite in the browser
- Not trying to impress you
 - Need to understand the scope to get the rest of the talk

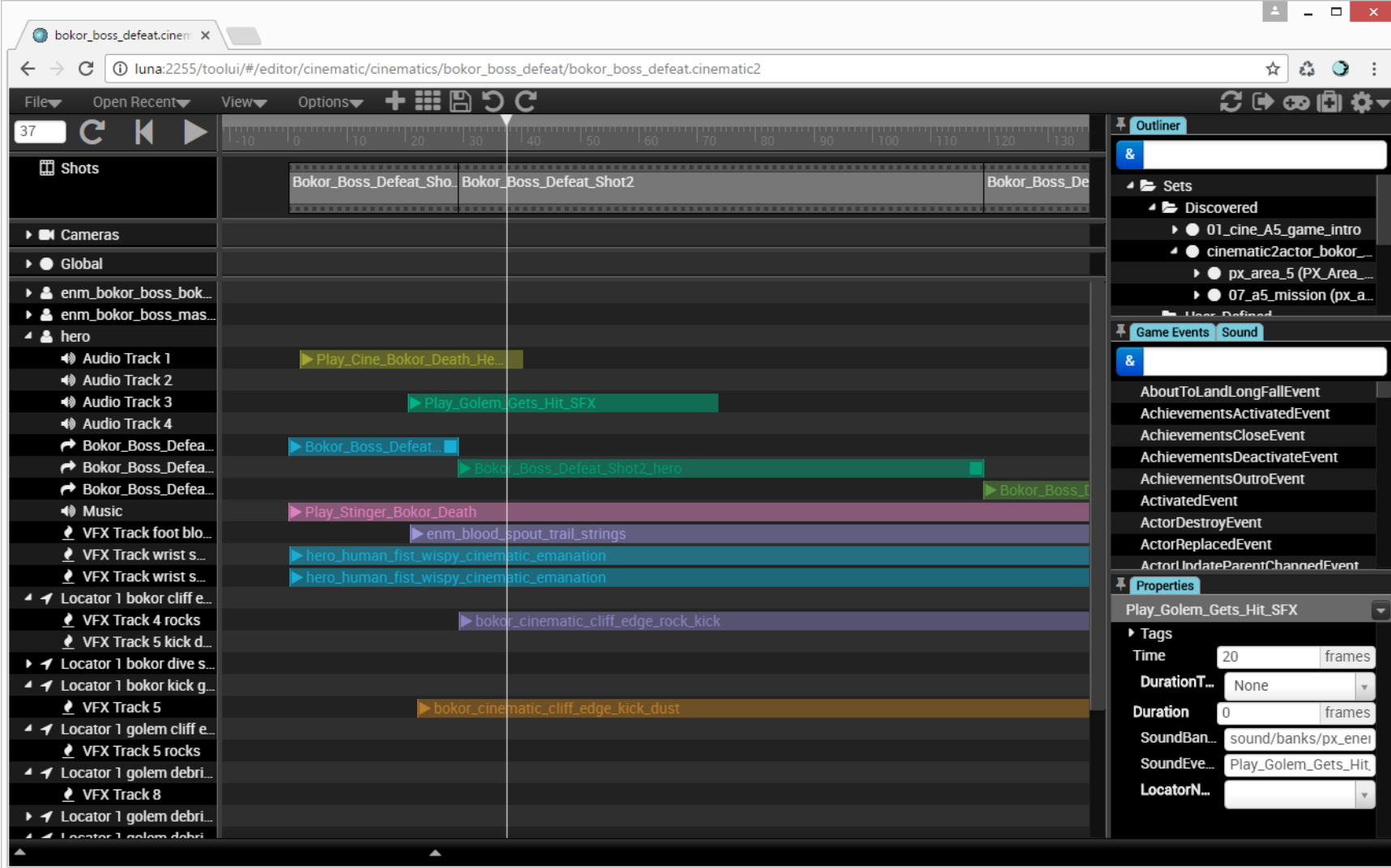


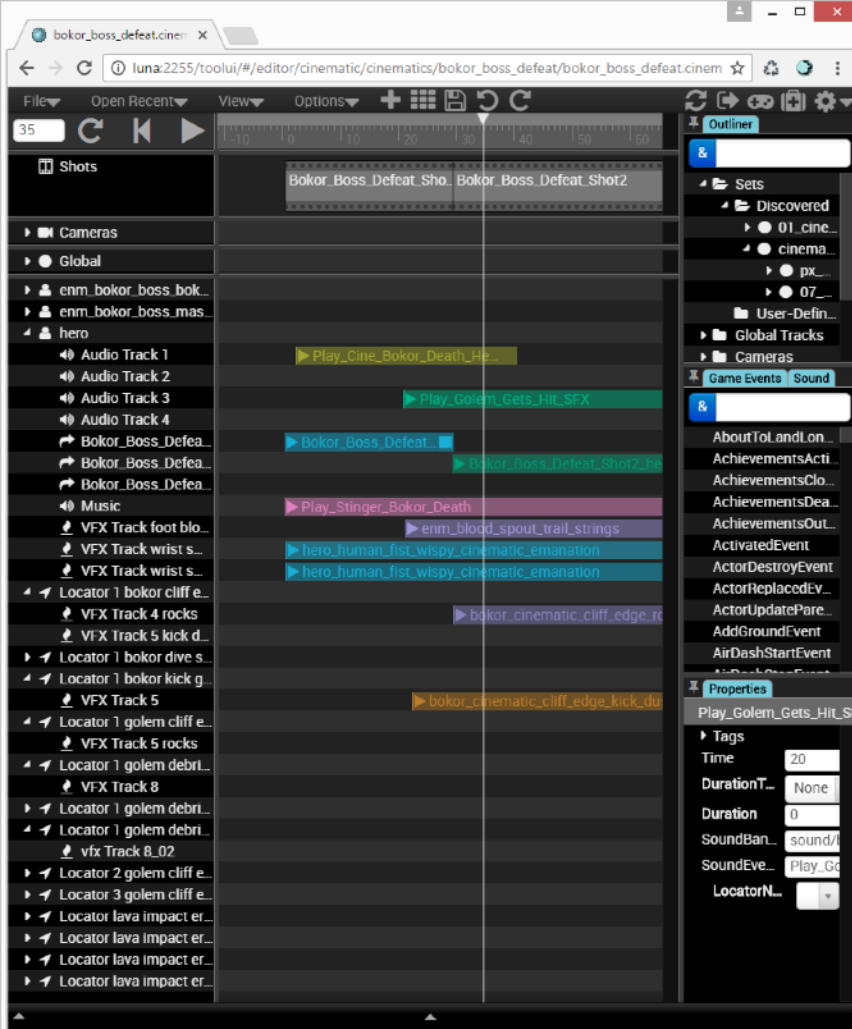


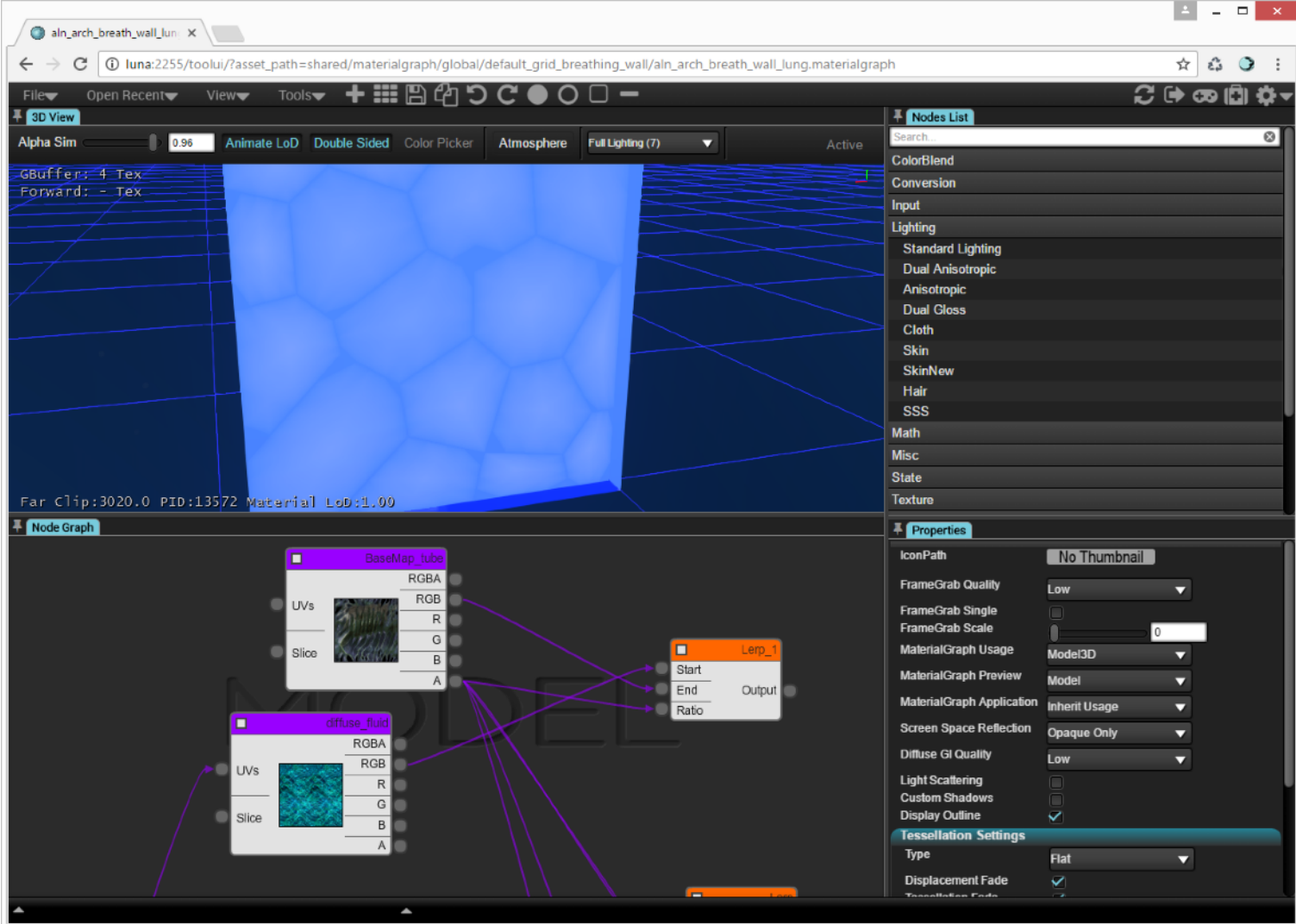


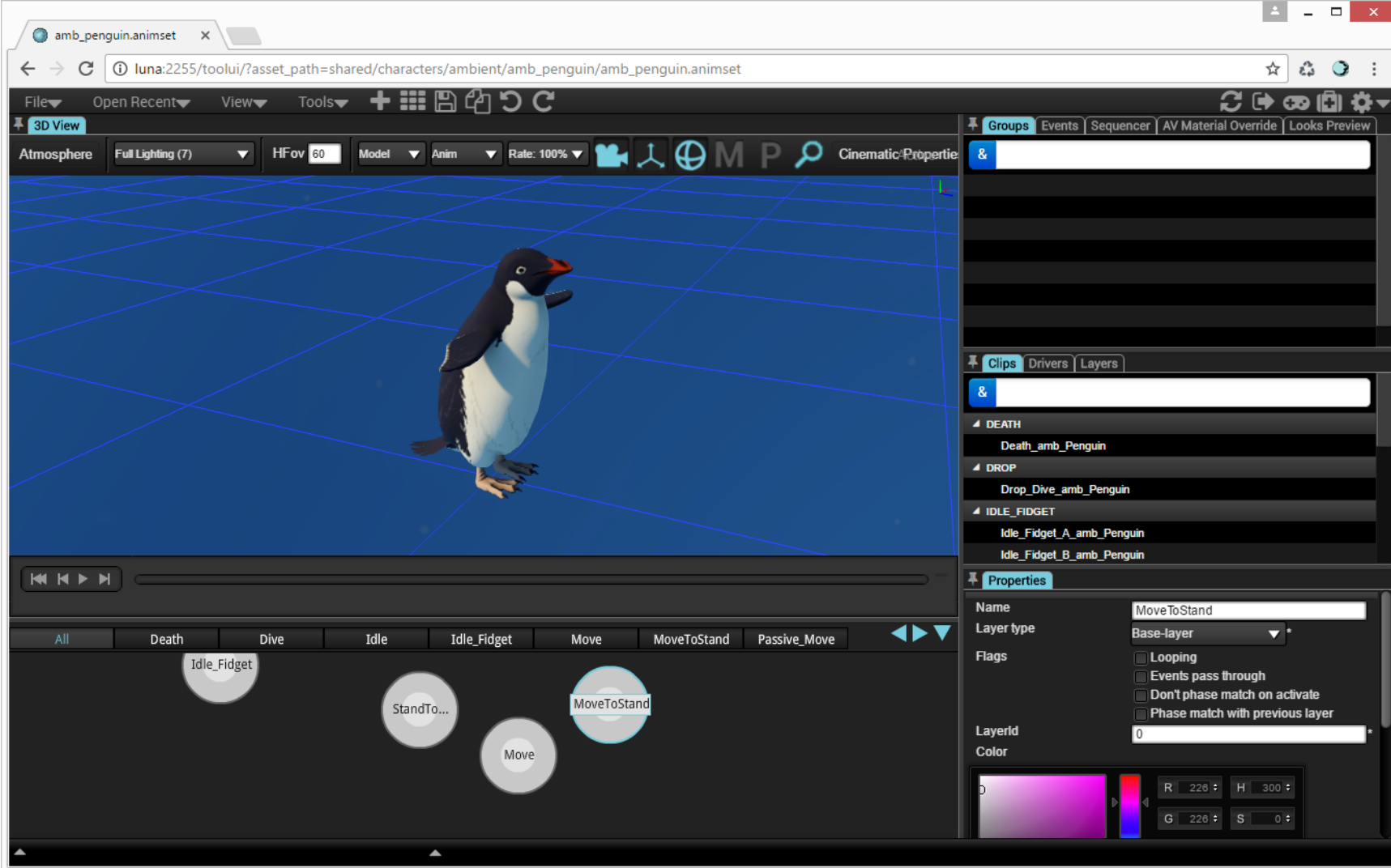


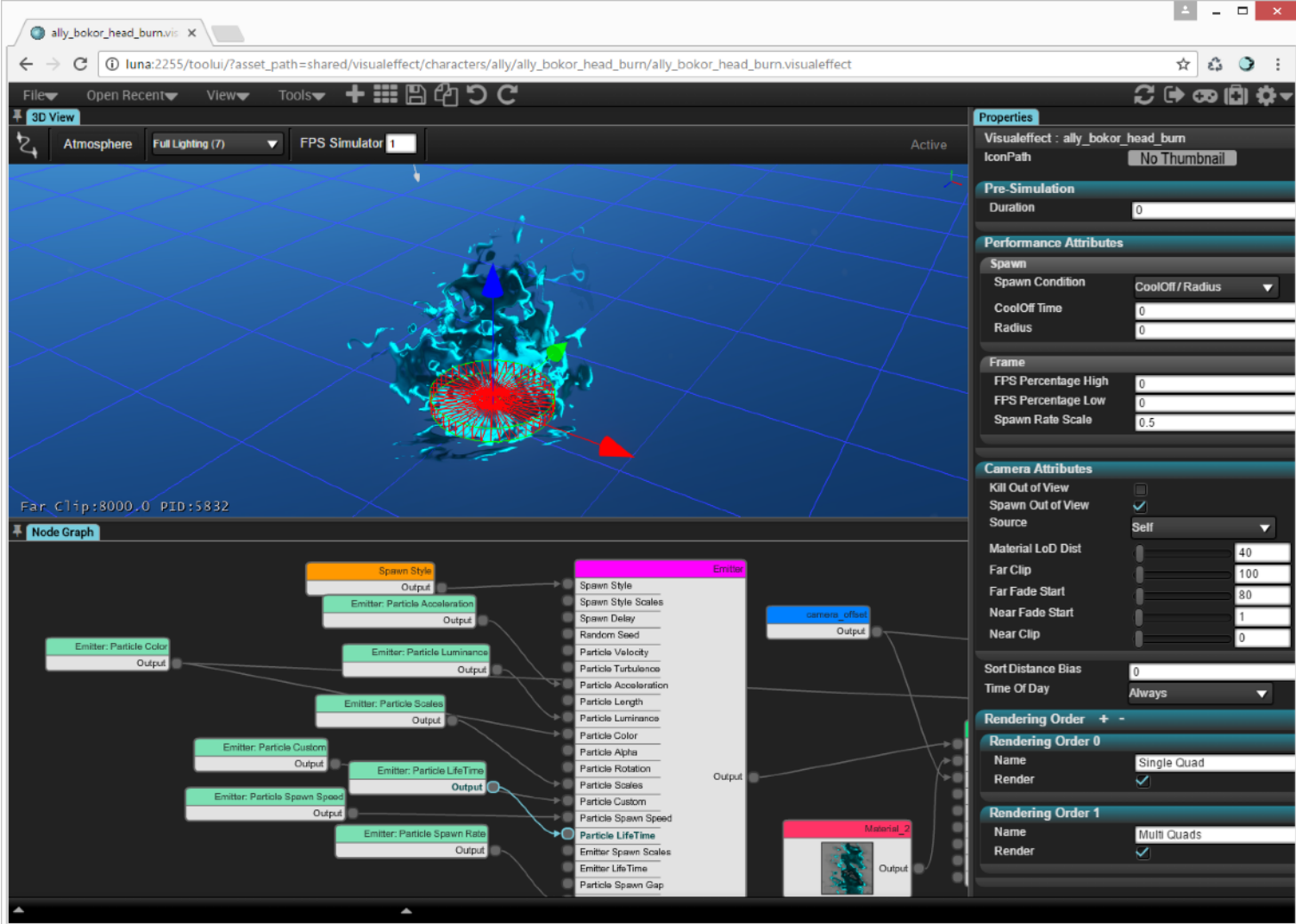












Sort By ▾

History ▾

View ▾

Results ▾

Asset Filters

- ☐ Actor ☆
- ☐ Animclip ☆
- ☐ Animset ☆
- ☐ Atmosphere ☆
- ☐ Cinematic2 ☆
- ☐ Collection ☆
- ☐ Conduit ☆
- ☐ Config ☆
- ☐ Level ☆
- ☐ Light ☆
- ☐ Localization ☆
- ☐ Material ☆
- ☐ Materialgraph ☆
- ☒ **Model** ☆
- ☐ Modelvariant ☆
- ☐ Movie ☆
- ☐ Performanceclip ☆
- ☐ Performanceset ☆
- ☐ Prefab ☆
- ☐ Region ☆
- ☐ Scriptgraph ☆
- ☐ Soundbank ☆
- ☐ Texture ☆
- ☐ Visualeffect ☆
- ☐ Volume ☆
- ☐ Zone ☆

Collections

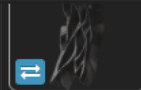
Drag Collection Here to Add

tools119

tools120

Search: Name ▾

cave



aln_arch_cavern_wall_lq_flat



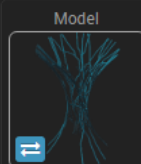
aln_rock_cave_arc_h_01



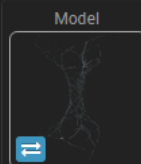
aln_rock_cave_col_umn_01



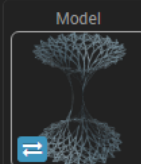
aln_rock_cave_col_umn_02



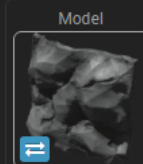
aln_rock_cave_col_umn_web_01



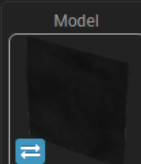
aln_rock_cave_col_umn_web_02



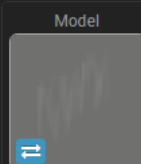
aln_rock_cave_col_umn_web_03



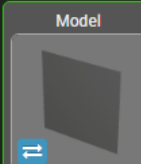
aln_rock_cave_wall_6x6



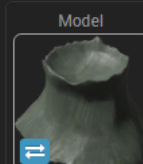
aln_rock_cave_wall_6x6_02



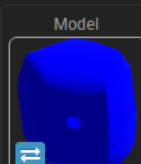
arc_amb_cave_lighting_01



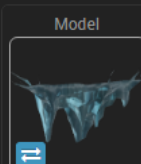
arc_amb_cave_tunnel_artificial_lighting



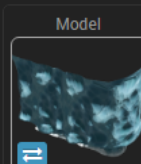
arc_gnd_allen_cave_mound_01



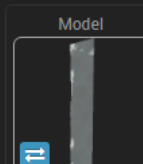
arc_gnd_allen_cave_shell_01



arc_rock_cave_ceiling_ice_01



arc_rock_cave_croch_ice_wall_01



arc_rock_cave Entrance_rib_01

General Info

File Name: arc_amb_cave_tunnel_artificial_lighting.model

Directory: shared\visualeffect\environment\arctic\ambient\arc_amb_cave_tunnel_artificial_lighting

Created By: yyoung

Last Modified By: yyoung

Modified Time: 5/11/2015, 6:06:59 AM

Tags

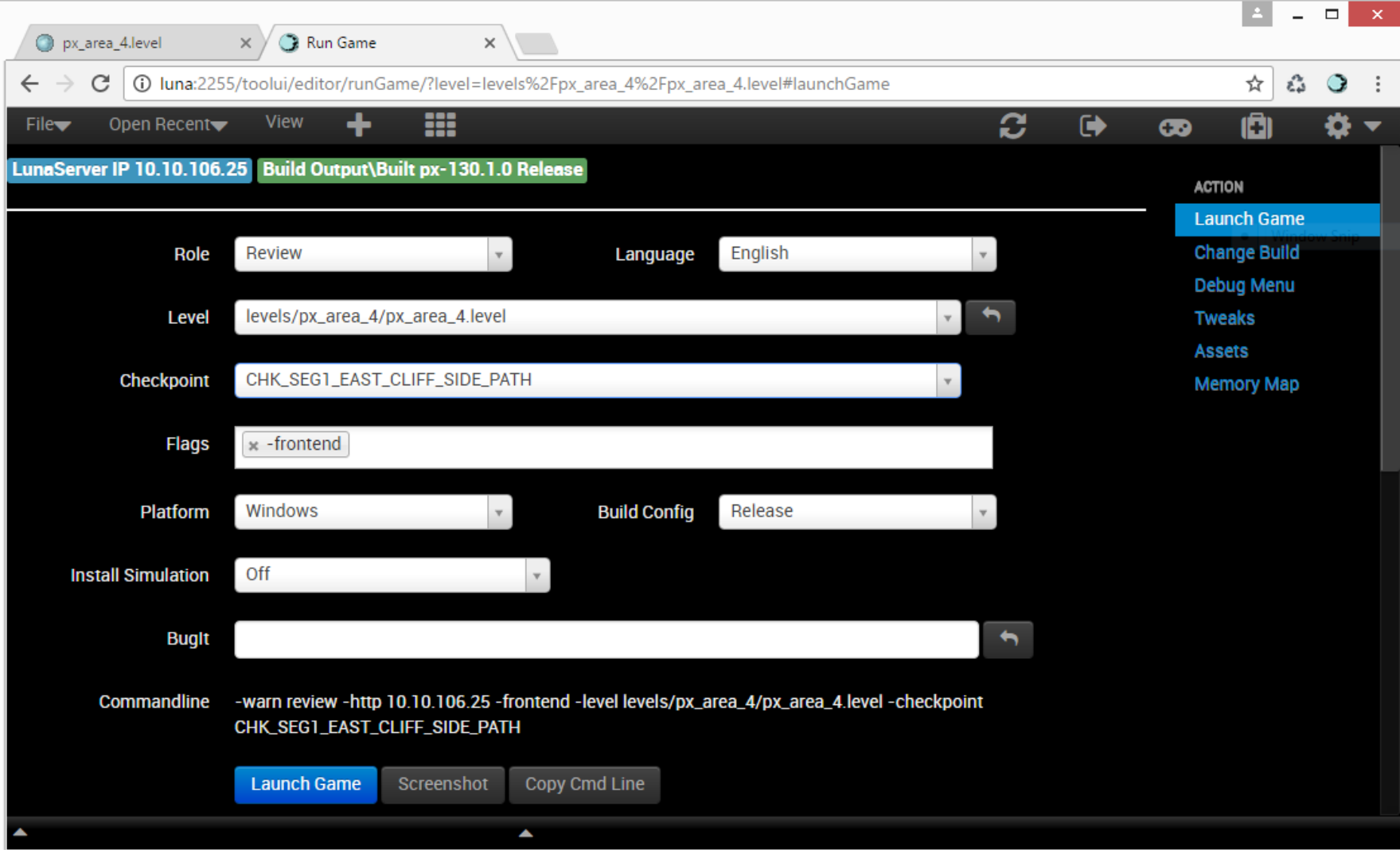
Type tag and hit Enter to Apply

Referenced By

arc_amb_cave_tunnel_artificial_lighting.actor

References

arc_amb_cave_tunnel_artificial_lighting.material



Insomniac Engine

Build Monitor

luna:2255/UI/BuildMonitor.html

Current Jobs

Errors: 8

Search

Settings

Ongoing Builds

File Changes: 0

Unrequested: 301

Requests: 0

Built Targets: 200719

> Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-1016_-40_328.lightgrid for Common - 0:00

> Accessing CCS shared/material/environment/global/prop/gbl_prop_specimen_ghoul_mummy_01/gbl_prop_specimen_ghoul_mummy_01.material for Common - 0:00

> Accessing CCS levels/px_area_3/lightgrids/grid_-664_24_312.lightgrid for Common - 0:00

> Accessing CCS levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-472_168_632.lightgrid for Common - 0:00

> Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-568_-8_184.lightgrid for Common - 0:00

> Building levels/px_area_4/instance/px_area_4/px_area_4/lightgrids/grid_312_152_680.lightgrid for Common - 0:00

> Accessing CCS levels/px_area_2/instance/px_area_2/px_area_2/lightgrids/grid_-120_24_472.lightgrid for Common - 0:00

Build Manager Log

Info

Clear Build Log

Pause

> 09:50:52 - Building levels/px_area_2/instance/px_area_2/px_area_2/lightgrids/grid_8_-8_-104.lightgrid for Common

> 09:50:52 - Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-520_72_632.lightgrid for Common

> 09:50:52 - Building levels/px_area_4/instance/px_area_4/px_area_4/lightgrids/grid_-360_248_1144.lightgrid for Common

> 09:50:52 - Building levels/px_area_0/instance/px_area_0/px_area_0/lightgrids/grid_-56_-24_-88.lightgrid for Common

> 09:50:52 - Building levels/px_area_3/lightgrids/grid_-440_-120_-136.lightgrid for Common

> 09:50:52 - Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-440_-8_-152.lightgrid for Common

> 09:50:52 - Building levels/px_area_2/instance/px_area_2/px_area_2/lightgrids/grid_-120_24_472.lightgrid for Common

> 09:50:52 - Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-472_168_632.lightgrid for Common

> 09:50:52 - Building shared/material/environment/global/prop/gbl_prop_specimen_ghoul_mummy_01/gbl_prop_specimen_ghoul_mummy_01.material for Common

> 09:50:52 - Building levels/px_area_3/lightgrids/grid_-664_24_312.lightgrid for Common

> 09:50:52 - Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-1016_-40_328.lightgrid for Common

> 09:50:52 - Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-568_-8_184.lightgrid for Common

> 09:50:52 - Building levels/px_area_4/instance/px_area_4/px_area_4/lightgrids/grid_312_152_680.lightgrid for Common

> 09:50:52 - Building levels/px_area_3/lightgrids/grid_-216_24_-72.lightgrid for Common

> 09:50:51 - Building levels/px_area_1/instance/px_area_1/px_area_1/lightgrids/grid_-88_40_1336.lightgrid for Common

> 09:50:51 - Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_40_-8_312.lightgrid for Common

> 09:50:51 - Building levels/px_area_3/instance/px_area_3/px_area_3/lightgrids/grid_-504_8_936.lightgrid for Common

> 09:50:51 - Building levels/px_area_5/instance/px_area_5/px_area_5/lightgrids/grid_376_104_232.lightgrid for Common

> 09:50:51 - Building levels/px_area_5/instance/px_area_5/px_area_5/lightgrids/grid_472_56_456.lightgrid for Common

> 09:50:51 - Building levels/px_area_5/instance/px_area_5/px_area_5/lightgrids/grid_280_-24_-216.lightgrid for Common

> 09:50:51 - Building levels/px_area_2/instance/px_area_2/px_area_2/lightgrids/grid_104_40_392.lightgrid for Common

> 09:50:51 - Building levels/px_area_2/instance/px_area_2/px_area_2/lightgrids/grid_-404_8_120.lightgrid for Common



That's a lot of stuff!





That's a lot of stuff!

- We definitely answered the “is it even possible” question





That's a lot of stuff!

- We definitely answered the “is it even possible” question
- 340,000 Javascript LOC, ~1,000 files
 - Excluding generated and 3rd party code



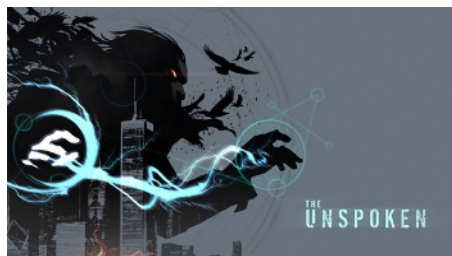


That's a lot of stuff!

- We definitely answered the “is it even possible” question
- 340,000 Javascript LOC, ~1,000 files
 - Excluding generated and 3rd party code
- 500,000 C++ LOC of server infrastructure & builders



What went right





What went right





What went right

- Debugger and profiler on every machine
 - Chrome dev tools are decent





What went right

- Debugger and profiler on every machine
 - Chrome dev tools are decent
- Achieved clean data separation





What went right

- Debugger and profiler on every machine
 - Chrome dev tools are decent
- Achieved clean data separation
- Centralized undo/redo





What went right

- Debugger and profiler on every machine
 - Chrome dev tools are decent
- Achieved clean data separation
- Centralized undo/redo
- Symbolic data manipulation





Right: Central undo/redo

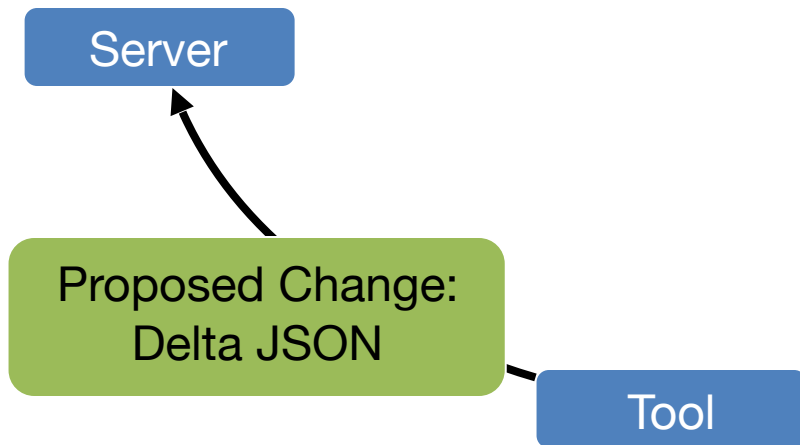
Server

Tool



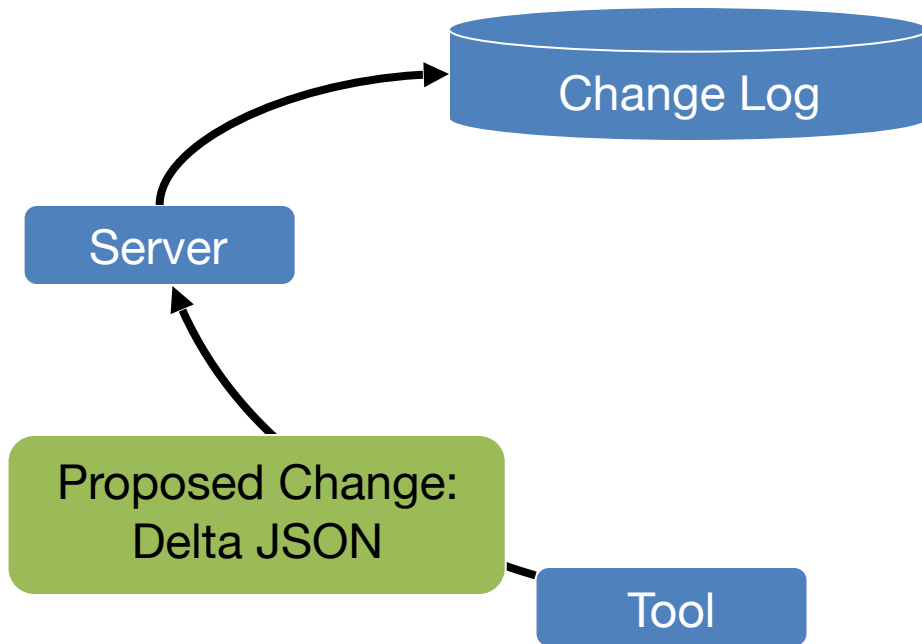


Right: Central undo/redo



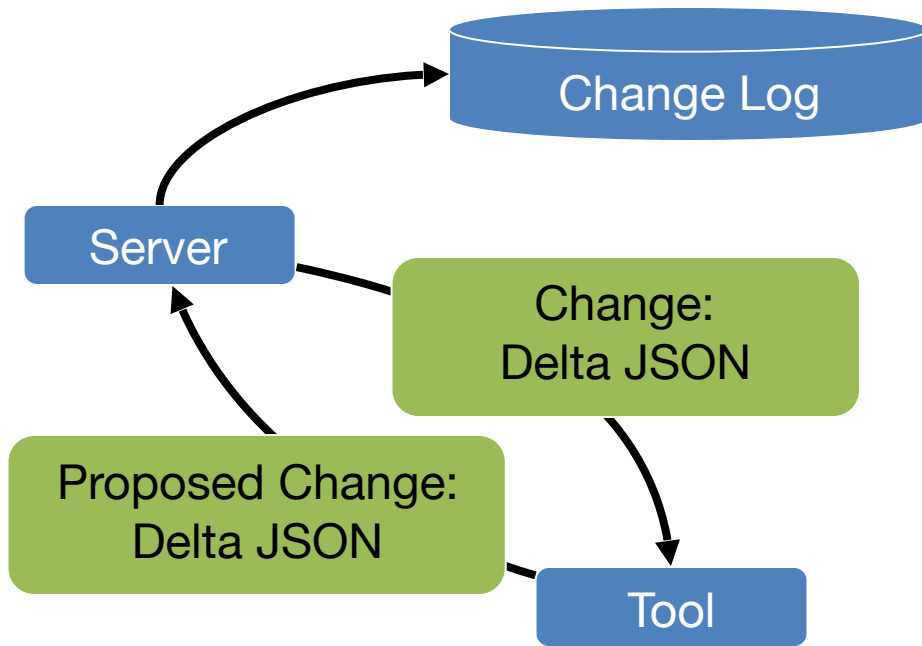


Right: Central undo/redo



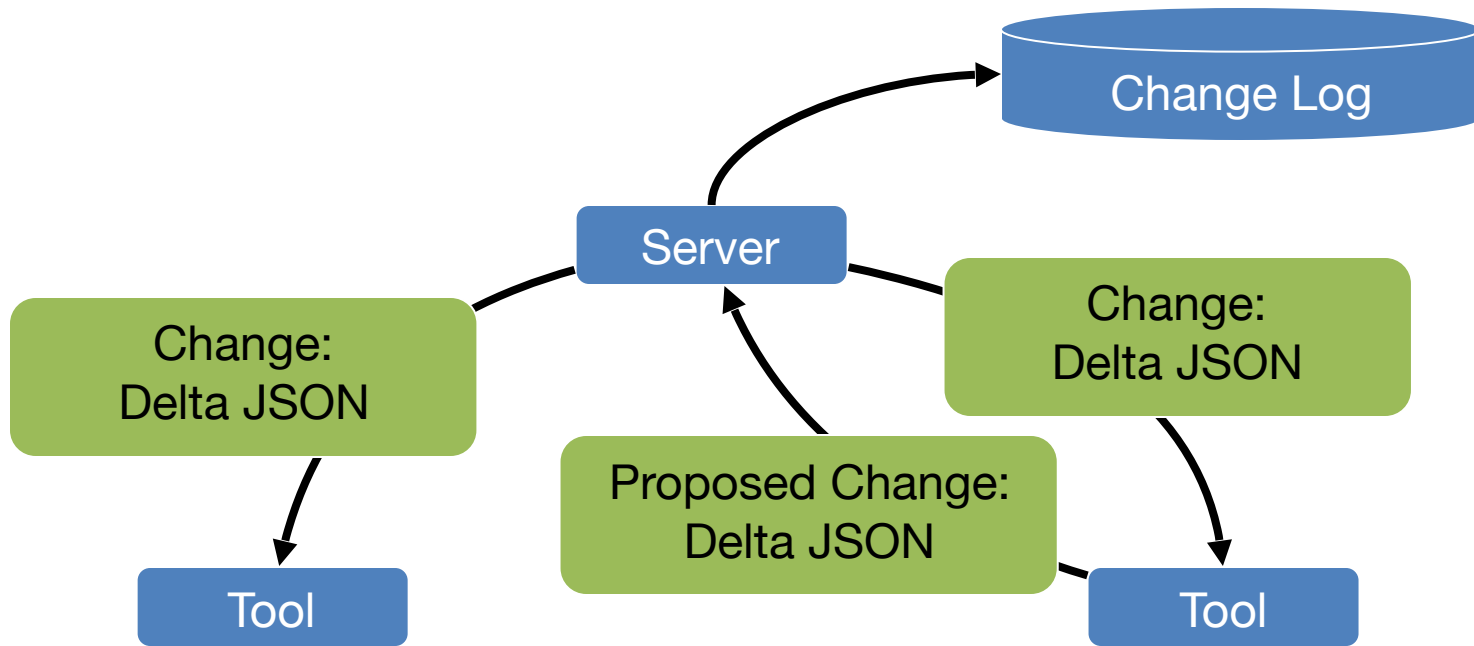


Right: Central undo/redo





Right: Central undo/redo



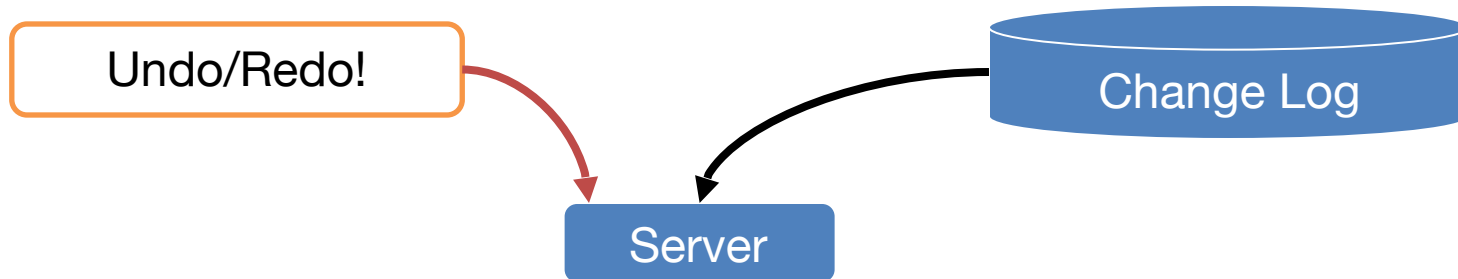


Right: Central undo/redo



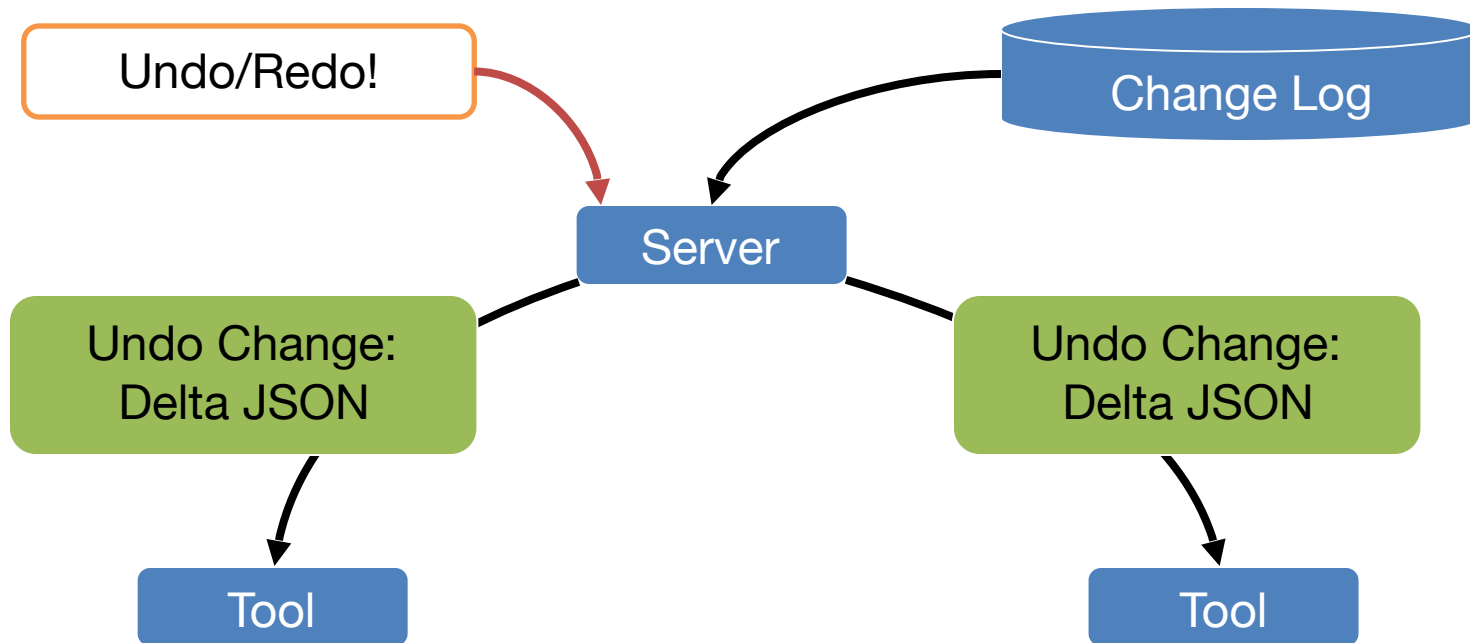


Right: Central undo/redo





Right: Central undo/redo





Right: Symbolic Data Manipulation





Right: Symbolic Data Manipulation

- Need to be able to edit any piece of game data in web tools





Right: Symbolic Data Manipulation

- Need to be able to edit any piece of game data in web tools
- Can't link to game (C++) code from web browser





Right: Symbolic Data Manipulation

- Need to be able to edit any piece of game data in web tools
- Can't link to game (C++) code from web browser

```
struct MySpecialObject {  
    int32 Hitpoints; // Default = 100  
    float Shininess; // Default = 3.5  
    // ...  
}
```





Right: Symbolic Data Manipulation

- Solution: Edit all data formats using meta-data
 - DDL compiler outputs both C++ code and metadata for tools
 - Use symbolic meta-data in tools to edit assets
 - Useful in native tools as well, reduces coupling

MySpecialObject metadata:

```
Name="Hitpoints" Type=i32 Default=100
```

```
Name="Shininess" Type=f32 Default=3.5
```

```
// ...
```





So, what went wrong?





So, what went wrong?

`}` yourselves





Wrong: Early web mindset





Wrong: Early web mindset

- “It’s just a web page”





Wrong: Early web mindset

- “It’s just a web page”
- In rush to try to get something out the door





Wrong: Early web mindset

- “It’s just a web page”
- In rush to try to get something out the door
- Poor engineering style early on
 - No tests
 - Global variables aplenty
 - Learning as we were going





Wrong: Javascript





Wrong: Javascript

- So much pain – a toy language that is still growing up





Wrong: Javascript

- So much pain – a toy language that is still growing up
- `undefined` is not a function





Wrong: Javascript

- So much pain – a toy language that is still growing up
- `undefined` is not a function
- Silently breaking other parts of the codebase
 - Renaming a function = super high risk
 - Adding a parameter? Good luck!
 - Relying on `grep` or IDE searching to fix up references often fails





Wrong: Javascript

- So much pain – a toy language that is still growing up
- `undefined` is not a function
- Silently breaking other parts of the codebase
 - Renaming a function = super high risk
 - Adding a parameter? Good luck!
 - Relying on `grep` or IDE searching to fix up references often fails
- Add quick fixes on top and you've got a nightmare





Wrong: No heavy lifting





Wrong: No heavy lifting

- JS single-threaded (web workers are a joke)





Wrong: No heavy lifting

- JS single-threaded (web workers are a joke)
- Garbage collection





Wrong: No heavy lifting

- JS single-threaded (web workers are a joke)
- Garbage collection
- Impractical to work with full game dataset
 - Sunset Overdrive's streaming optimization, for example





Wrong: LunaServer became a dumping ground





Wrong: LunaServer became a dumping ground

- Anything you can't do in the browser goes in the server..





Wrong: LunaServer became a dumping ground

- Anything you can't do in the browser goes in the server..
- LunaServer quickly grew bigger and bigger
 - Random program launching
 - File system interfaces
 - Desktop interop services
 - Perforce interfaces





Wrong: LunaServer became a dumping ground

- Anything you can't do in the browser goes in the server..
- LunaServer quickly grew bigger and bigger
 - Random program launching
 - File system interfaces
 - Desktop interop services
 - Perforce interfaces
- Lots of technical debt unless planned for carefully!





Wrong: LunaServer became a dumping ground

- Anything you can't do in the browser goes in the server..
- LunaServer quickly grew bigger and bigger
 - Random program launching
 - File system interfaces
 - Desktop interop services
 - Perforce interfaces
- Lots of technical debt unless planned for carefully!
- Is this still used? Grep all the code and hope for the best!





Mixed: Javascript on V8





Mixed: Javascript on V8

- Optimizing for performance can be non-intuitive
 - Programming several layers away from actual compiler





Mixed: Javascript on V8

- Optimizing for performance can be non-intuitive
 - Programming several layers away from actual compiler
- V8 optimization best practices change over time
 - Keeping up on this for 300k LOC is a hard problem





Mixed: Javascript on V8

- Optimizing for performance can be non-intuitive
 - Programming several layers away from actual compiler
- V8 optimization best practices change over time
 - Keeping up on this for 300k LOC is a hard problem
- Often conflicts between idiomatic vs fast





Mixed: Javascript on V8

- Optimizing for performance can be non-intuitive
 - Programming several layers away from actual compiler
- V8 optimization best practices change over time
 - Keeping up on this for 300k LOC is a hard problem
- Often conflicts between idiomatic vs fast
- Lack of information and “what good looks like”
 - Many web frameworks have poor performance





Wrong: V8 Stability





Wrong: V8 Stability

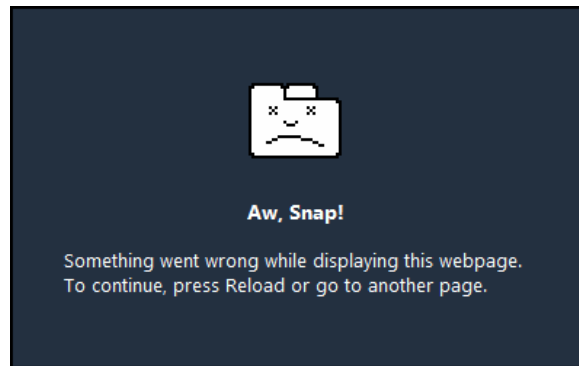
- We wanted to use ES6/Harmony features
 - In particular, 'let' fixes block scope terribleness





Wrong: V8 Stability

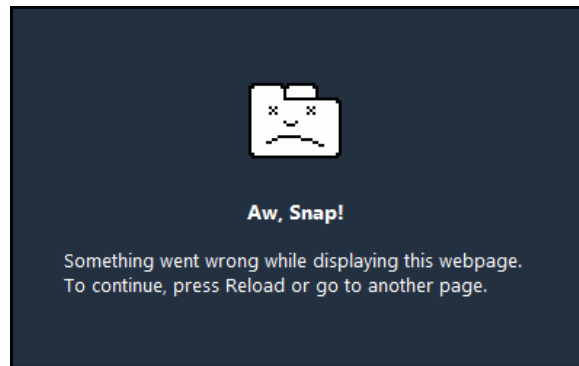
- We wanted to use ES6/Harmony features
 - In particular, 'let' fixes block scope terribleness
- Great until
 - Browser crashed in production, randomly
 - Debugger crashed trying to debug it





Wrong: V8 Stability

- We wanted to use ES6/Harmony features
 - In particular, 'let' fixes block scope terribleness
- Great until
 - Browser crashed in production, randomly
 - Debugger crashed trying to debug it
- Resorted to taking it all out again





Wrong: Promises

- Lots of things you wait for in JS
 - Because you're constantly talking to some server





Wrong: Promises

- Lots of things you wait for in JS
 - Because you're constantly talking to some server
- Event handling this is messy





Wrong: Promises

- Lots of things you wait for in JS
 - Because you're constantly talking to some server
- Event handling this is messy
- Promises hope to solve this problem





```
animSetMojo.Source.prototype.loadAnimSetActor = function() {  
    if ( !this.isConnected() ) {  
        return Q.resolve();  
    }  
    return this.onStage1FetchActor()  
        .then( this._private.callbacks.onStage2PreLoadAssets )  
        .then( this._private.callbacks.onWaitForAssetLoad )  
        .then( this._private.callbacks.onStage3CreateActor )  
        .then( this._private.callbacks.onWaitForAssetLoad )  
        .then( this._private.callbacks.onStage4WatchActor )  
        .then( this._private.callbacks.onStage5Focus );  
};
```





```
animSetMojo.Source.prototype.loadAnimSetActor = function() {  
    if ( !this.isConnected() ) {  
        return Q.resolve();  
    }  
    return this.onStage1FetchActor()  
        .then( this._private.callbacks.onStage2PreLoadAssets )  
        .then( this._private.callbacks.onWaitForAssetLoad )  
        .then( this._private.callbacks.onStage3CreateActor )  
        .then( this._private.callbacks.onWaitForAssetLoad )  
        .then( this._private.callbacks.onStage4WatchActor )  
        .then( this._private.callbacks.onStage5Focus );  
};
```

Promise Chain





Promises = Now all your crashes look like this

```
TypeError: Cannot read property 'Path' of null  
at vault.js:1614  
at _fulfilled (q-0.9.6.js:714)  
at self.promiseDispatch.done (q-0.9.6.js:743)  
at Promise.promise.promiseDispatch (q-0.9.6.js:  
680)  
at q-0.9.6.js:554  
at MessagePort.flush (q-0.9.6.js:108)  
msgLog.error @ msgLog.js:168  
(anonymous) @ msgLog.js:195  
defaultQPromiseErrorHandler @ env.js:928  
(anonymous) @ q-0.9.6.js:1553  
flush @ q-0.9.6.js:108
```





Promises = Now all your crashes look like this

```
TypeError: Cannot read property 'Path' of null  
at vault.js:1614  
at _fulfilled (q-0.9.6.js:714)  
at self.promiseDispatch.done (q-0.9.6.js:743)  
at Promise.promise.promiseDispatch (q-0.9.6.js:  
680)  
at q-0.9.6.js:554  
at MessagePort.flush (q-0.9.6.js:108)  
msgLog.error @ msgLog.js:168  
(anonymous) @ msgLog.js:195  
defaultQPromiseErrorHandler @ env.js:928  
(anonymous) @ q-0.9.6.js:1553  
flush @ q-0.9.6.js:108
```

← Your code





Promises = Now all your crashes look like this

```
TypeError: Cannot read property 'Path' of null  
at vault.js:1614  
at _fulfilled (q-0.9.6.js:714)  
at self.promiseDispatch.done (q-0.9.6.js:743)  
at Promise.promise.promiseDispatch (q-0.9.6.js:  
680)  
at q-0.9.6.js:554  
at MessagePort.flush (q-0.9.6.js:108)  
msgLog.error @ msgLog.js:168  
(anonymous) @ msgLog.js:195  
defaultQPromiseErrorHandler @ env.js:928  
(anonymous) @ q-0.9.6.js:1553  
flush @ q-0.9.6.js:108
```

← Your code

Unrelated,
generic & useless





Typescript





Typescript

- Adds a type system on top of JS





Typescript

- Adds a type system on top of JS
- We started migrating in late 2014





Typescript

- Adds a type system on top of JS
- We started migrating in late 2014
- Adding types means adding documentation/semantics





Typescript

- Adds a type system on top of JS
- We started migrating in late 2014
- Adding types means adding documentation/semantics
- It also means adding build steps
 - JS just lost 98% of the “quick fix” appeal





Right/Wrong: Typescript





Right/Wrong: Typescript

- Definitely helps, but still not perfect





Right/Wrong: Typescript

- Definitely helps, but still not perfect
- Still very possible to write buggy code in it
 - Use **all** the strictness options if you have a choice





Right/Wrong: Typescript

- Definitely helps, but still not perfect
- Still very possible to write buggy code in it
 - Use **all** the strictness options if you have a choice
- AMD/CommonJS module system interop was hairy





Right/Wrong: Typescript

- Definitely helps, but still not perfect
- Still very possible to write buggy code in it
 - Use **all** the strictness options if you have a choice
- AMD/CommonJS module system interop was hairy
- Wrapping JS modules with declarations was hairy





Wrong: Too many frameworks





Wrong: Too many frameworks

- Web is all about rapid evolution





Wrong: Too many frameworks

- Web is all about rapid evolution
- Most web apps don't have 300k LOC
 - Ours did, can't keep up with web evolution at this codebase size





Wrong: Too many frameworks

- Web is all about rapid evolution
- Most web apps don't have 300k LOC
 - Ours did, can't keep up with web evolution at this codebase size
- Ended up with way to many 3rd party frameworks
 - All at once





Wrong: Too many frameworks

- Web is all about rapid evolution
- Most web apps don't have 300k LOC
 - Ours did, can't keep up with web evolution at this codebase size
- Ended up with way to many 3rd party frameworks
 - All at once
- Prevents sharing solutions between tools/pages
 - Too many one-offs that can't be reused





Wrong: Off the shelf is mostly useless





Wrong: Off the shelf is mostly useless

- Most stuff simply doesn't scale to AAA data sizes
 - Most web sites have dozens of things, we have tens of thousands
 - Different constraints





Wrong: Off the shelf is mostly useless

- Most stuff simply doesn't scale to AAA data sizes
 - Most web sites have dozens of things, we have tens of thousands
 - Different constraints
- Often designed to be “easy to use” or “friendly”





Case in point: Tree views





Case in point: Tree views

- jstree is a popular web tree control we evaluated





Case in point: Tree views

- jstree is a popular web tree control we evaluated



*I would think that performance with a mere **1,000 nodes** should be unnoticeable. However, on my i7 machine with 12 GB ram, it takes almost **a minute and a half** in Chrome to drag and drop those 1,000 nodes somewhere else in the tree*





Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements





Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls





Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

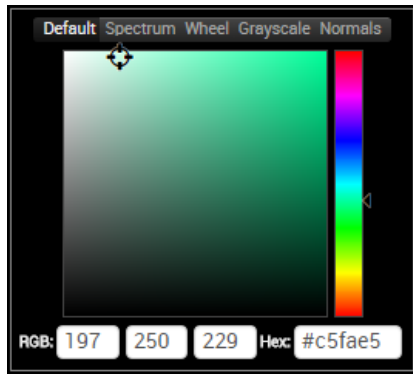




Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

Pro space

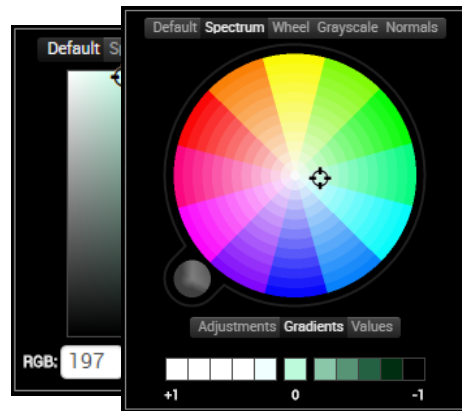




Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

Pro space

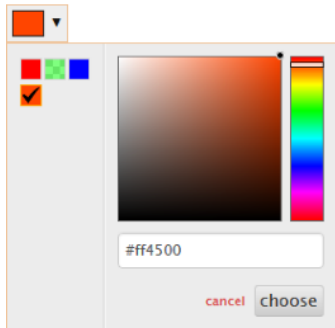




Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

Pro space

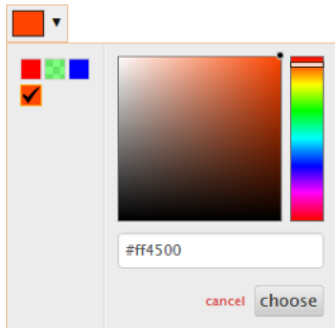




Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

Pro space

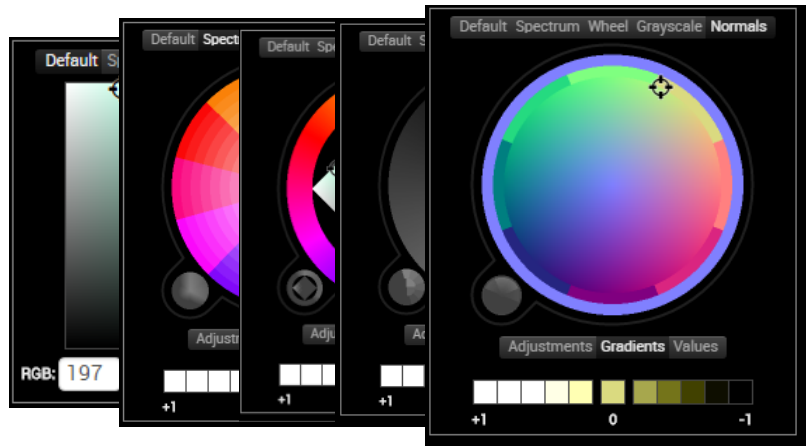




Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

Pro space

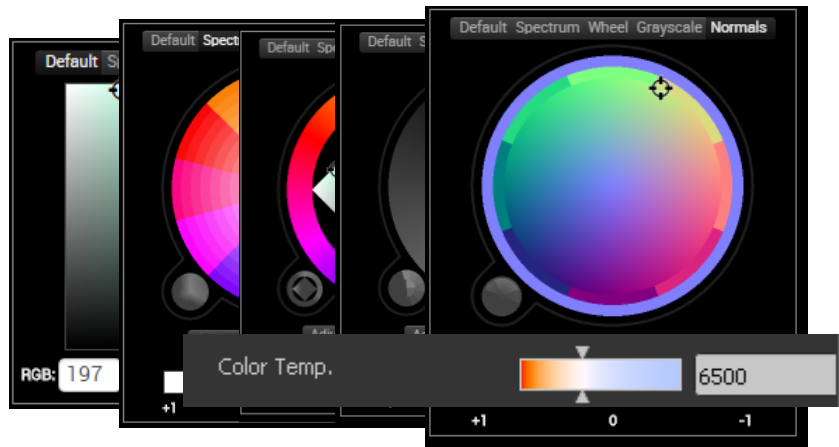




Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

Pro space





Case in point: Color picker

- “pick a color you like” vs hard core pro user requirements

Color: AB2567



Most web controls

Advanced web

Pro space





Right/Wrong: node.js





Right/Wrong: node.js

- Using the web stack from scripts has value
 - Reuse well tested code in batch scripts





Right/Wrong: node.js

- Using the web stack from scripts has value
 - Reuse well tested code in batch scripts
- node.js is a terrible ecosystem
 - Dependency hell (remember left-pad?)





Right/Wrong: node.js

- Using the web stack from scripts has value
 - Reuse well tested code in batch scripts
- node.js is a terrible ecosystem
 - Dependency hell (remember left-pad?)
- After *light* use we lug around 90 MB of deps
 - 8,500 files





Right/Wrong: node.js

- Using the web stack from scripts has value
 - Reuse well tested code in batch scripts
- node.js is a terrible ecosystem
 - Dependency hell (remember left-pad?)
- After *light* use we lug around 90 MB of deps
 - 8,500 files
- Found debugging & profiling super flaky





Wrong: JSON data in code as objects





Wrong: JSON data in code as objects

- Mistake: Used JSON asset data as JS objects directly
 - 'undefined' everywhere when data formats change
 - Fix litters code with `if (typeof(thing) !== "undefined")`





Wrong: JSON data in code as objects

- Mistake: Used JSON asset data as JS objects directly
 - 'undefined' everywhere when data formats change
 - Fix litters code with `if (typeof(thing) !== "undefined")`
- Obviously bad in hindsight, but was an easy trap to fall into
 - Maintenance cost is astronomical, because it's just strings
 - No compile time help, grep and test all code branches





Wrong: JSON data in code as objects

- Mistake: Used JSON asset data as JS objects directly
 - 'undefined' everywhere when data formats change
 - Fix litters code with `if (typeof(thing) !== "undefined")`
- Obviously bad in hindsight, but was an easy trap to fall into
 - Maintenance cost is astronomical, because it's just strings
 - No compile time help, grep and test all code branches
- Partly fixed with symbolic access library (too late)





Wrong: “It’s OK, it’s just in the Javascript”





Wrong: “It’s OK, it’s just in the Javascript”

- Belief that it was OK to hack stuff in because “it’s just script”
 - Very easy to deploy hacks and local changes and get away with it
 - Tempting to work around hot issues this way
 - Probably our C++ bias led to this mistake





Wrong: “It’s OK, it’s just in the Javascript”

- Belief that it was OK to hack stuff in because “it’s just script”
 - Very easy to deploy hacks and local changes and get away with it
 - Tempting to work around hot issues this way
 - Probably our C++ bias led to this mistake
- Manifested as a legacy of unmaintainable hacks
 - Performance regressions, copy-pasta
 - Especially bad in first generation JS code which is still in production





The Document Object Model (DOM)





The Document Object Model (DOM)

- Hardest part of performant web tools
 - Black art





The Document Object Model (DOM)

- Hardest part of performant web tools
 - Black art
- Reflows kill your performance
 - Sandwich in a complete rebuild of your UI at “random” places





The Document Object Model (DOM)

- Hardest part of performant web tools
 - Black art
- Reflows kill your performance
 - Sandwich in a complete rebuild of your UI at “random” places
- Using animation frames is critical to performance
 - Not well understood





Lesson: JS needs *more* rigor than C++, not less





Lesson: JS needs *more* rigor than C++, not less

- Suspect this is true in any dynamically typed ecosystem





Lesson: JS needs *more* rigor than C++, not less

- Suspect this is true in any dynamically typed ecosystem
- Especially hard for people jumping in only occasionally
 - Typically “non JS” programmers left huge messes
 - Did not want to invest in learning the “web tools way”





Wrong: Have you cleared cache?





Wrong: Have you cleared cache?

- Chrome caches *very* aggressively
 - Clear cache, or risk running with random JS code out of date





Wrong: Have you cleared cache?

- Chrome caches *very* aggressively
 - Clear cache, or risk running with random JS code out of date
- Constant drain on support and user base
 - Handful of support issues every week due to this





Wrong: Have you cleared cache?

- Chrome caches *very* aggressively
 - Clear cache, or risk running with random JS code out of date
- Constant drain on support and user base
 - Handful of support issues every week due to this
- Laughably bad compared to native tool deployment





From: -REDACTED-

Sent: Tuesday, January 31, 2017 6:30 PM

To: Syndicate

Subject: FW: BUG - cannot create new collections

I swung by -REDACTED-,

No amount of clearing cache/reloading extensions/restarting chrome got this fixed. We cannot repro either. Anyone know what might be going on here?



Wrong: “Try reinstalling the plugin”





Wrong: “Try reinstalling the plugin”

- Plugin part of the desktop interop
 - 3D view, window management





Wrong: “Try reinstalling the plugin”

- Plugin part of the desktop interop
 - 3D view, window management
- NPAPI drop mandated rewrite





Wrong: “Try reinstalling the plugin”

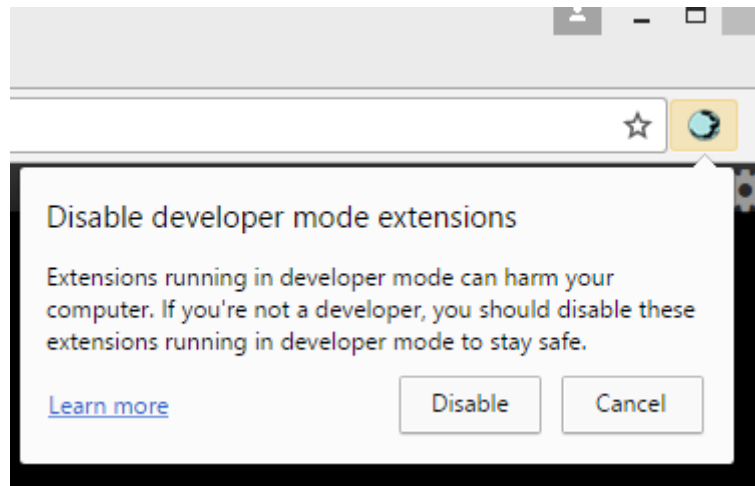
- Plugin part of the desktop interop
 - 3D view, window management
- NPAPI drop mandated rewrite
- Forced Chrome Store requirement
 - Now have security warnings on every launch of the tools





Wrong: “Try reinstalling the plugin”

- Plugin part of the desktop interop
 - 3D view, window management
- NPAPI drop mandated rewrite
- Forced Chrome Store requirement
 - Now have security warnings on every launch of the tools





Wrong: You're not in charge





Wrong: You're not in charge

- Chrome team doesn't care about your tools





Wrong: You're not in charge

- Chrome team doesn't care about your tools
- Auto-updates *will* break everything
 - Constant source of downtime & panic
 - Random API changes, flash 0-day bugfixes, V8 regressions





Wrong: You're not in charge

- Chrome team doesn't care about your tools
- Auto-updates *will* break everything
 - Constant source of downtime & panic
 - Random API changes, flash 0-day bugfixes, V8 regressions
- Can run Canary builds to catch some of it early
 - But you're still not in charge
 - Always a scramble to fix things last moment





Policies = The Nuclear Option

- You *can* control some policies for Chrome





Policies = The Nuclear Option

- You *can* control some policies for Chrome
- Used auto-update disable hammer to ship Sunset
 - Frozen branch meant we couldn't get Chrome breakage fixes into that game's codebase





Policies = The Nuclear Option

- You *can* control some policies for Chrome
- Used auto-update disable hammer to ship Sunset
 - Frozen branch meant we couldn't get Chrome breakage fixes into that game's codebase
- Plenty of user pushback about restricting Chrome
 - Favored daily browser





Policies = The Nuclear Option

- You *can* control some policies for Chrome
- Used auto-update disable hammer to ship Sunset
 - Frozen branch meant we couldn't get Chrome breakage fixes into that game's codebase
- Plenty of user pushback about restricting Chrome
 - Favored daily browser
- Should have done this from day 1
 - And would have been hard sell then too





From: -REDACTED-

Sent: Monday, October 14, 2013 10:51 AM

To: Lunarest

Subject: ads in Luna

This awesome ad popped up in my workspace last week. Not sure if this issue has anything to do with whatever's causing the white bar at the bottom of the screen.



fx_proximity_mine_glow.v

luna2255/Main.html

Apps The Team | Insomni... fizzie tower cine Microsoft Team Fou...

Other bookmarks

No build errors

Current Asset - fx_proximity_mine_glow.visualeffect

Atmosphere Full Lighting (7) HFOv 60

(P) aused

Properties

Pre-Simulation

Duration 0

Camera Attributes

Near Clip 0

Near Fade Start 1

Far Fade Start 80

Far Clip 120

Material LoD Dist 40

Spawn Out of View ☒

Kill Out of View ☐

Rendering Order + -

Rendering Order 0

Name light

Render ☒

Rendering Order 1

Name glowOuter

Render ☒

1 Weird Tip For Gaining Muscle Fast

See how this 1 weird tip makes you gain muscle fast. [See How](#)

HIDE AD





fx_proximity_mine_glow.v

luna2255/Main.html

Apps The Team | Insomni... fizzie tower cine Microsoft Team Fou...

Other bookmarks

No build errors

Current Asset - fx_proximity_mine_glow.visualeffect

Atmosphere Full Lighting (7) HFOv 60

(P) aused

Properties

Pre-Simulation

Duration 0

Camera Attributes

Near Clip 0

Near Fade Start 1

Far Fade Start 80

Far Clip 120

Material LoD Dist 40

Spawn Out of View ☒

Kill Out of View ☐

Rendering Order + -

Rendering Order 0

Name light

Render ☒

Rendering Order 1

Name glowOuter

Render ☒

1 Weird Tip For Gaining Muscle Fast

See how this 1 weird tip makes you gain muscle fast. [See How](#)

HIDE AD





Wrong: A hostile environment for your tools





Wrong: A hostile environment for your tools

- Malware





Wrong: A hostile environment for your tools

- Malware
- Rampant extensions, addons and random junk
 - Adblock added 30 ms of latency for each outgoing AJAX request
 - When you have 1000s of them, it starts to really suck





Wrong: A hostile environment for your tools

- Malware
- Rampant extensions, addons and random junk
 - Adblock added 30 ms of latency for each outgoing AJAX request
 - When you have 1000s of them, it starts to really suck
- Themes





Wrong: Culture Shock





Wrong: Culture Shock

- JS and web stack is very different from engine dev
 - Deep specialization inevitable and necessary to do the job





Wrong: Culture Shock

- JS and web stack is very different from engine dev
 - Deep specialization inevitable and necessary to do the job
- Started to see cracks in the team culture





At road's end (2015)





At road's end (2015)

- Dev time for features snowballed
 - Frustrated at lack of progress
 - Lots of bugs were outside our control
 - Maintenance costs killing productivity





At road's end (2015)

- Dev time for features snowballed
 - Frustrated at lack of progress
 - Lots of bugs were outside our control
 - Maintenance costs killing productivity
- Had lost 2 of our JS specialists





At road's end (2015)

- Dev time for features snowballed
 - Frustrated at lack of progress
 - Lots of bugs were outside our control
 - Maintenance costs killing productivity
- Had lost 2 of our JS specialists
- Team tension was mounting
 - We needed a new direction





Our current tools direction (2016-)





Our current tools direction (2016-)

- Client layer in C++, with Qt for UI
 - Right cultural choice for us
 - Go back to mature dev tools, pipelines





Our current tools direction (2016-)

- Client layer in C++, with Qt for UI
 - Right cultural choice for us
 - Go back to mature dev tools, pipelines
- Keep what works!
 - Edit loop, server infrastructure – enables gradual transition





Our current tools direction (2016-)

- Client layer in C++, with Qt for UI
 - Right cultural choice for us
 - Go back to mature dev tools, pipelines
- Keep what works!
 - Edit loop, server infrastructure – enables gradual transition
- 9 months to get complete level editor beta up
 - Included rewriting all JS infrastructure in C++





2016 Surprises





2016 Surprises

- Maintenance cost went way down as we stopped JS dev
 - Breakage in production proportional to # of JS changes





2016 Surprises

- Maintenance cost went way down as we stopped JS dev
 - Breakage in production proportional to # of JS changes
- Still firefighting things outside our control
 - Auto-updates: gift that keeps on giving
 - Flash update broke all node graphs over night
 - About a week of engineering effort to drop everything and fix





How could you make it work?





How could you make it work?

- Embedded browser engine
 - No auto-updates, add-ins and other craziness
 - No more clear cache





How could you make it work?

- Embedded browser engine
 - No auto-updates, add-ins and other craziness
 - No more clear cache
- Use TypeScript exclusively





How could you make it work?

- Embedded browser engine
 - No auto-updates, add-ins and other craziness
 - No more clear cache
- Use TypeScript exclusively
- Settle on small number of technologies
 - Use them consistently even if they're not the latest web craze





How could you make it work?

- Embedded browser engine
 - No auto-updates, add-ins and other craziness
 - No more clear cache
- Use TypeScript exclusively
- Settle on small number of technologies
 - Use them consistently even if they're not the latest web craze
- Be conscious of specialization and team culture





Vision post-mortem: Hiring





Vision post-mortem: Hiring

- We put the word out, and people know about our web tools..





Vision post-mortem: Hiring

- We put the word out, and people know about our web tools..
- ..but no web specialists applied





Vision post-mortem: Hiring

- We put the word out, and people know about our web tools..
- ..but no web specialists applied
- Even if they had applied, would we have hired them?
 - We need generalists on the Core team





Vision post-mortem: Lower the UI barrier





Vision post-mortem: Lower the UI barrier

- We found that web apps of this magnitude *raise* the barrier
 - Extra context switching: JavaScript, client-server, DOM, HTML
 - Takes months to learn all the intricacies of the tech stack





Vision post-mortem: Lower the UI barrier

- We found that web apps of this magnitude *raise* the barrier
 - Extra context switching: JavaScript, client-server, DOM, HTML
 - Takes months to learn all the intricacies of the tech stack
- It's no better than a traditional desktop GUI dev
 - At least not for apps this size
 - You can totally make a throwaway log viewer in 10 minutes
 - But no one wants that thing in production





Conclusion





Conclusion

- No regrets – 6 games shipped speak for themselves





Conclusion

- No regrets – 6 games shipped speak for themselves
- Our engine team is very versatile – but why stretch it?





Conclusion

- No regrets – 6 games shipped speak for themselves
- Our engine team is very versatile – but why stretch it?
- Probably wouldn't have done this if we knew the *real* costs
 - First explorer gets to discover all the traps 😊





Conclusion

- No regrets – 6 games shipped speak for themselves
- Our engine team is very versatile – but why stretch it?
- Probably wouldn't have done this if we knew the *real* costs
 - First explorer gets to discover all the traps 😊
- It could have been a lot smoother if we prepared better





Conclusion

- No regrets – 6 games shipped speak for themselves
- Our engine team is very versatile – but why stretch it?
- Probably wouldn't have done this if we knew the *real* costs
 - First explorer gets to discover all the traps 😊
- It could have been a lot smoother if we prepared better
- “Zero install” remains an attractive goal
 - Can probably get close with native tools too though





Thanks

- Q & A
- afredriksson@insomniacgames.com
- @deplinenoise on twitter

