

Killer Engine for Remixing Games

Jesse Himmelstein  
CRI-Paris

# tl;dr

- New game engine, with different goals
- Made for remixing games and mashups
- Electronics-inspired visual programming
- Excellent debugging and live coding tools
- All done in the browser

- Interdisciplinary science and engineering



Higher education

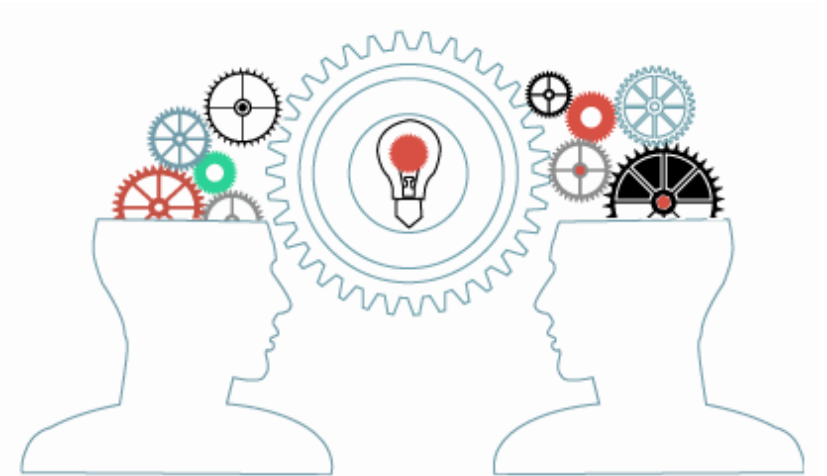
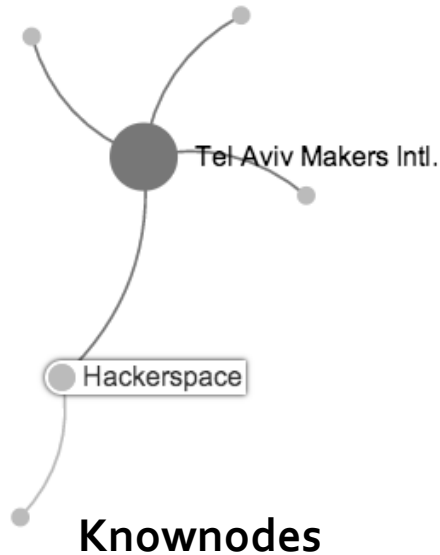


School and after-school programs

# CTO @ [cri-paris.org](http://cri-paris.org)



- Education technology



Coldea

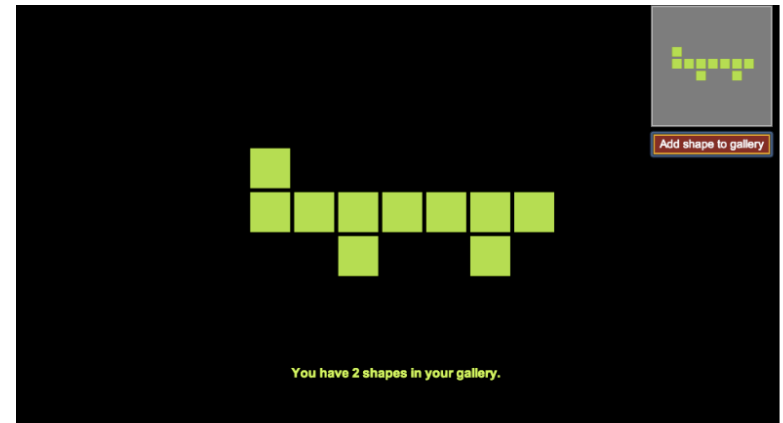


**SYNBIO4ALL**  
  
[www.synbio4all.org](http://www.synbio4all.org)

- Scientific and Educational Games



Archimedes' Escape



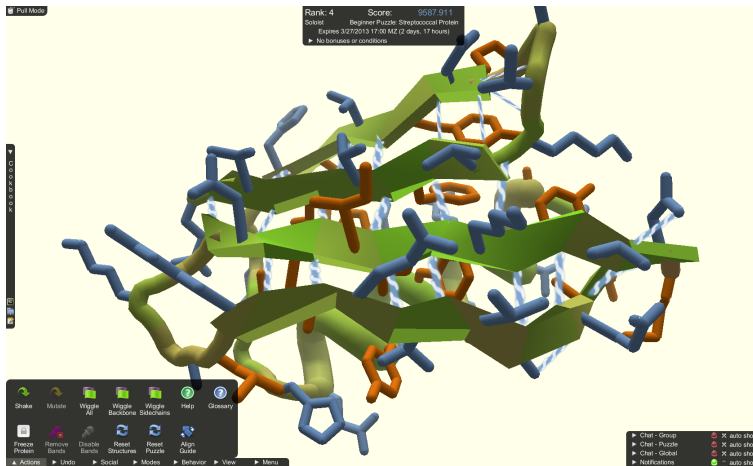
Leap  
Weizmann Institute



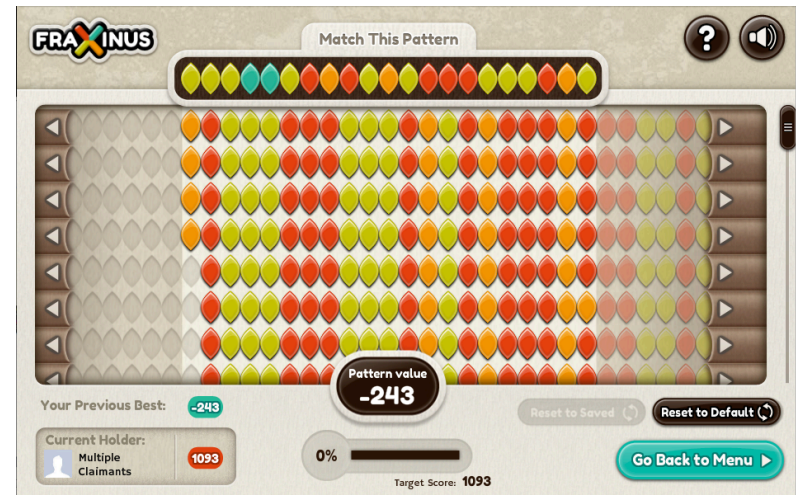
# What I Do



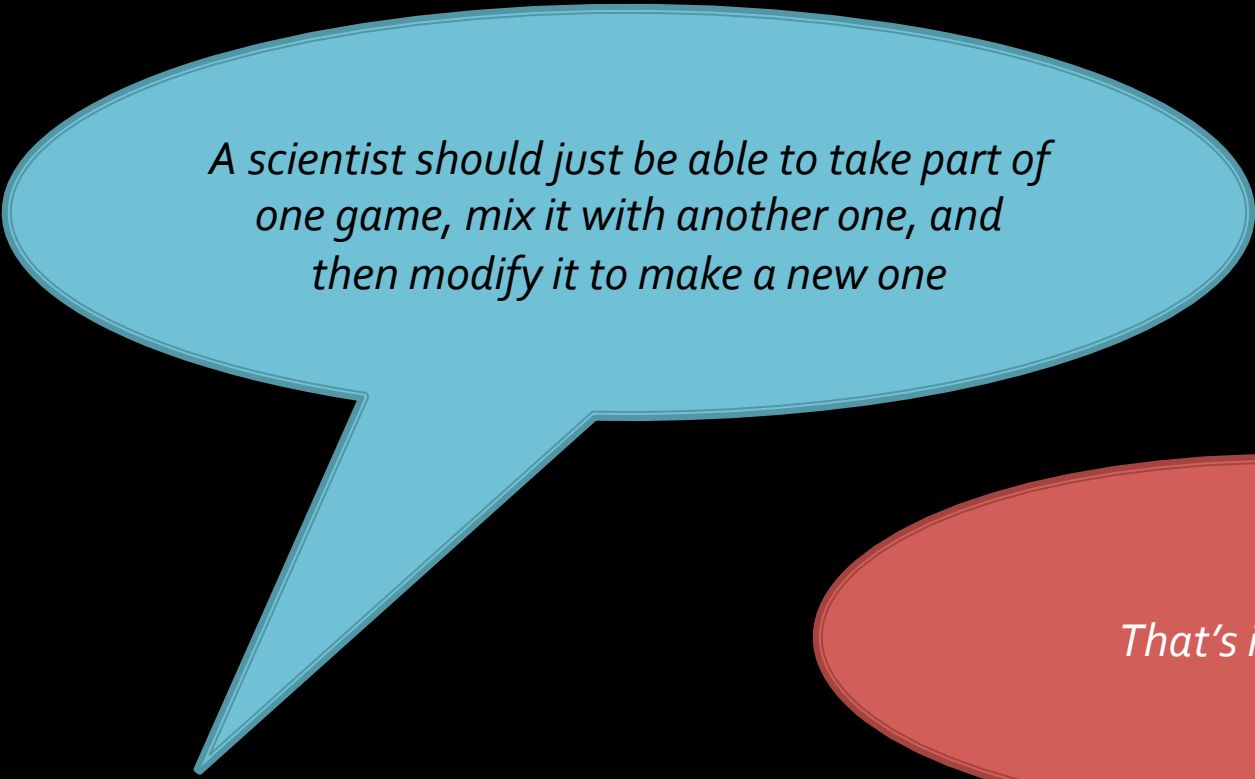
## ■ Citizen Science Games



FoldIt  
Univ of Washington



Fraxinus  
The Sainsbury Laboratory



*A scientist should just be able to take part of  
one game, mix it with another one, and  
then modify it to make a new one*

MY BOSS



*That's impossible*

ME



*Why?*

# Why?

- We have:

Classes

Functions

Metaclasses

Blocks

Entity Component Systems

Unit tests

End to end tests

Distributed Version Control

Virtual machines

A million programming languages

A billion game engines

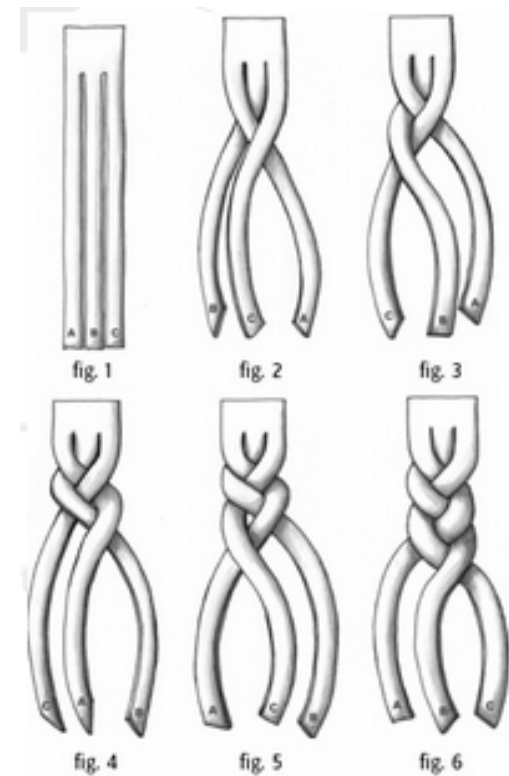
A kabillion libraries





# Simplicity

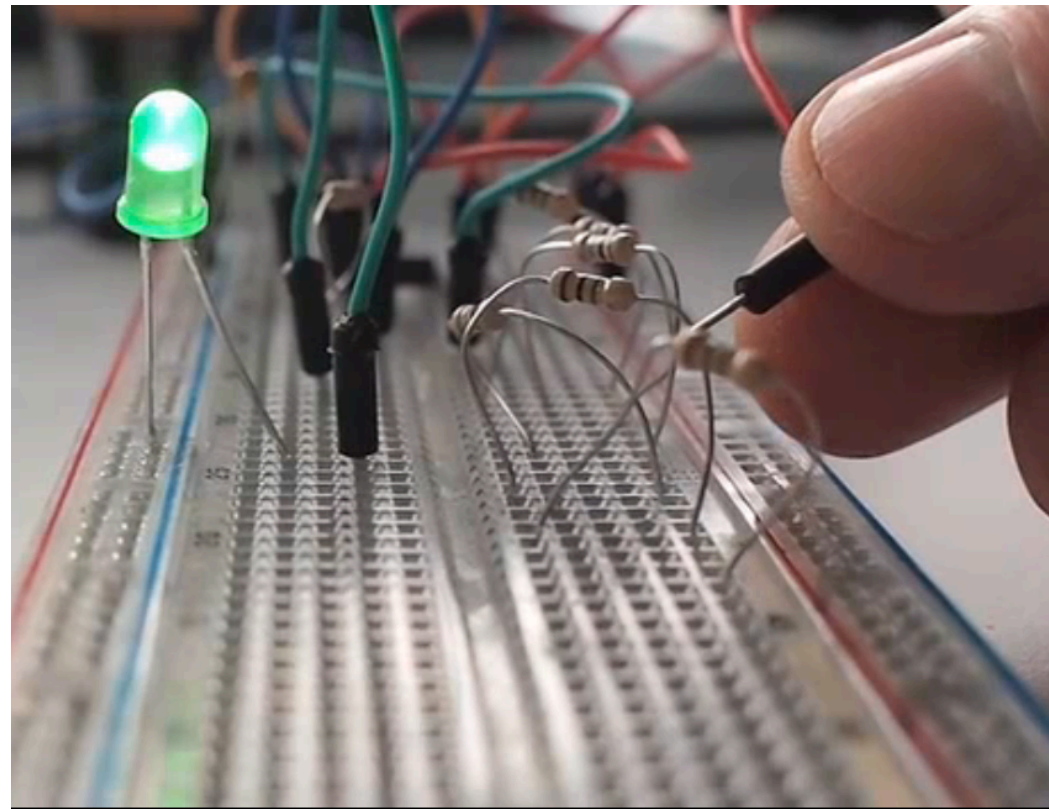
- Interleaving -> incomprehension
- Lessons
  - Avoid tangling
  - Leave data alone
  - Sane abstractions for state and time



Rich Hickey  
"Simple made Easy"

# Inspiration

- Breadboard
  - Connect anything
  - Measure anywhere
  - Replace everything



# Visual Programming

- Been done before...

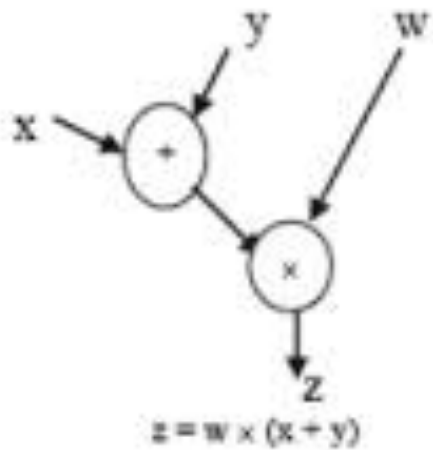
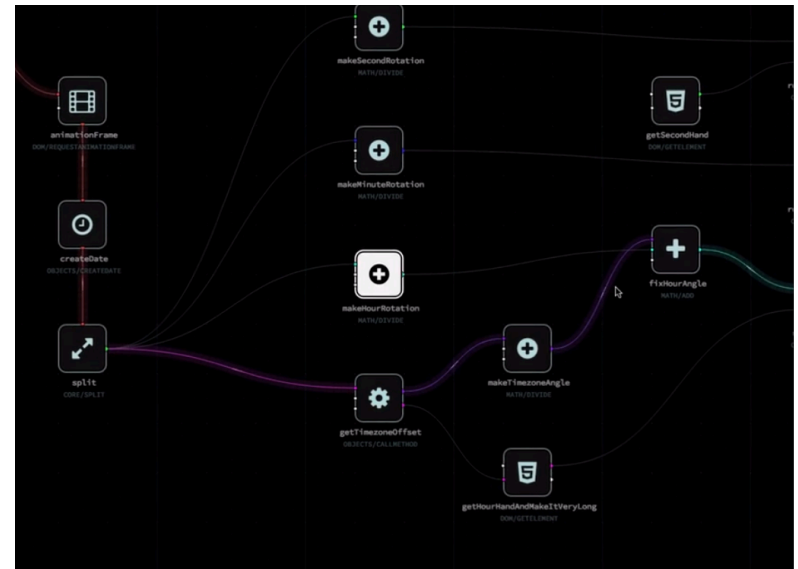
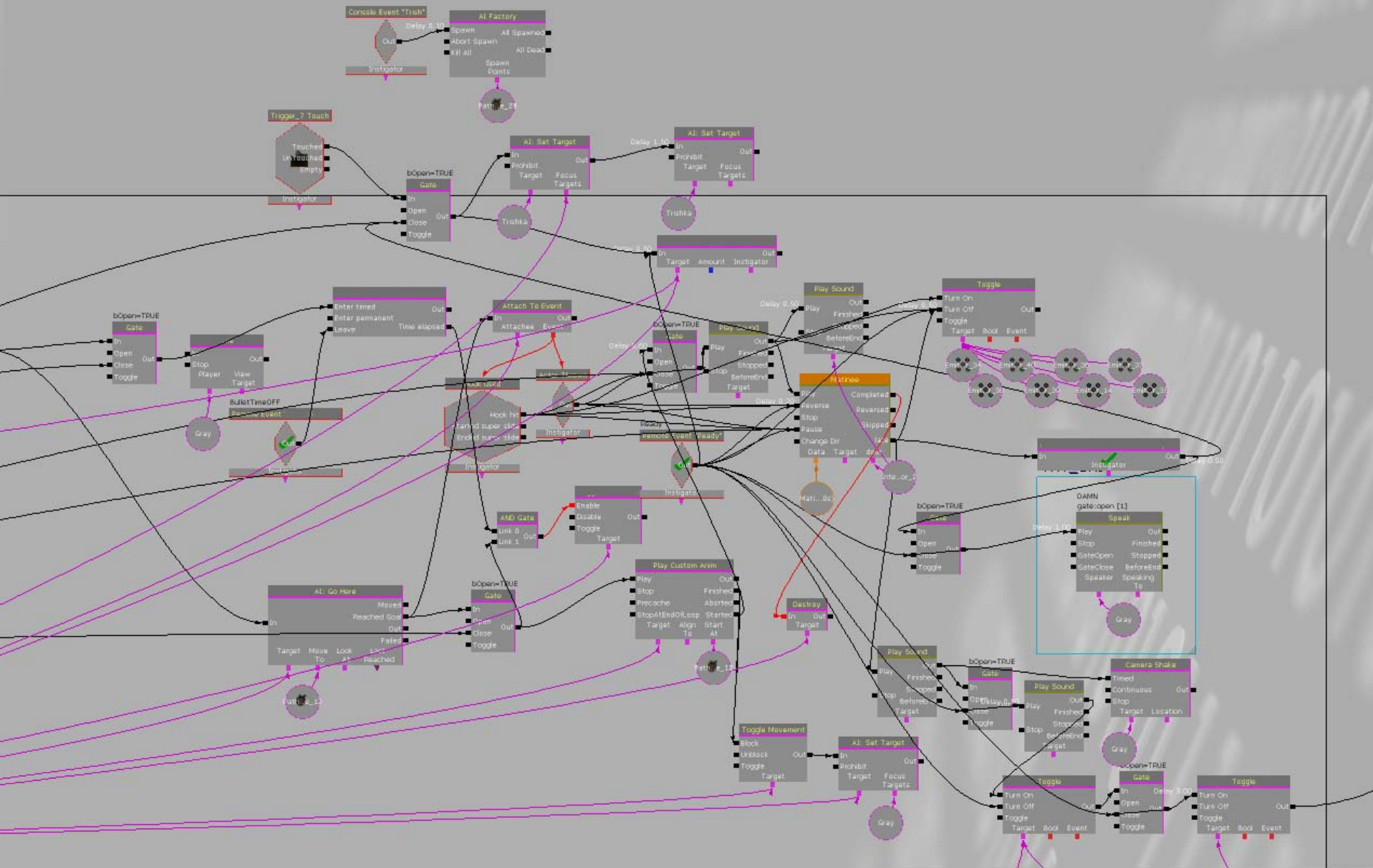


Figure 4: DFG for  $z = w \times (x + y)$

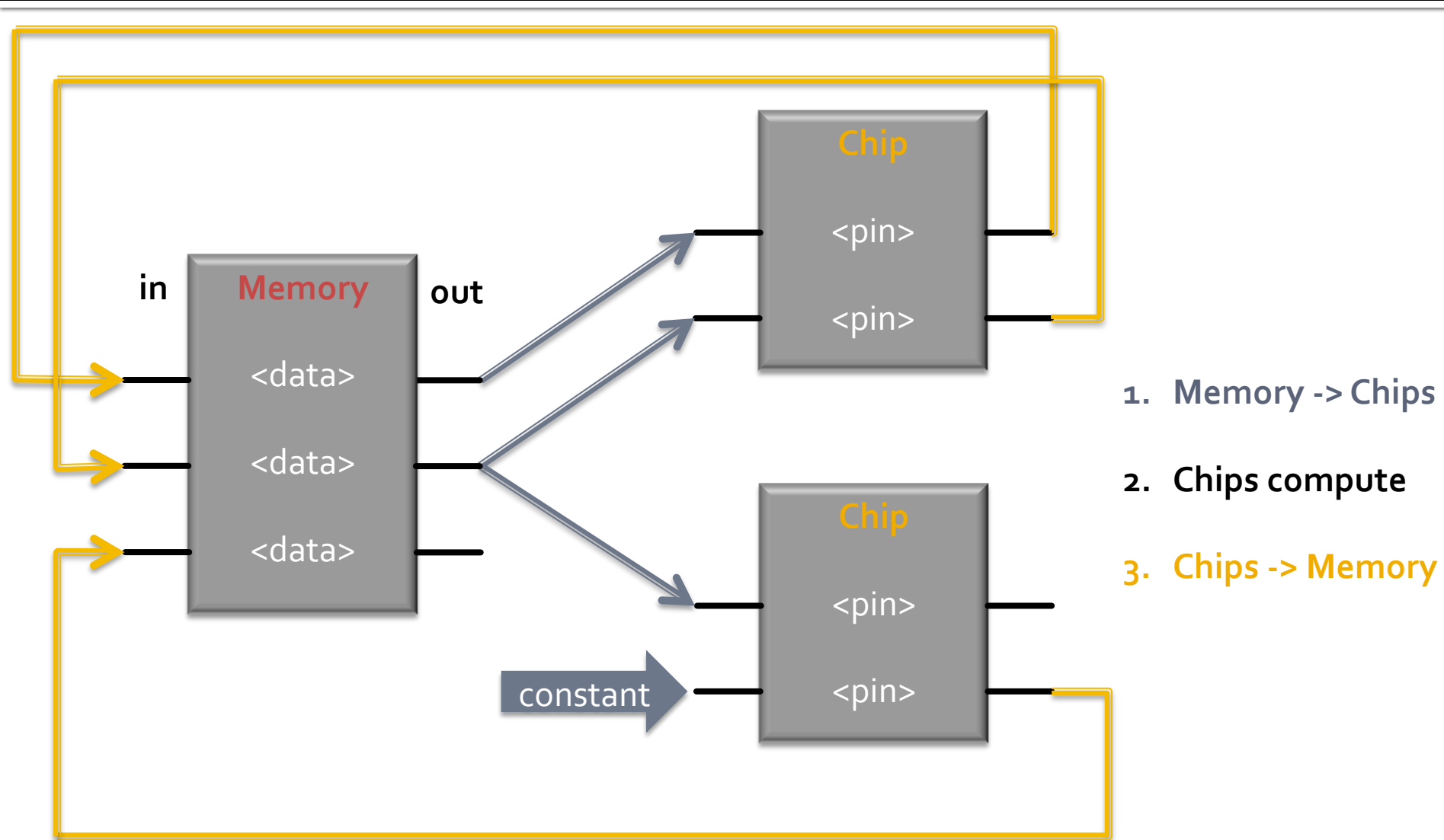
Data flow



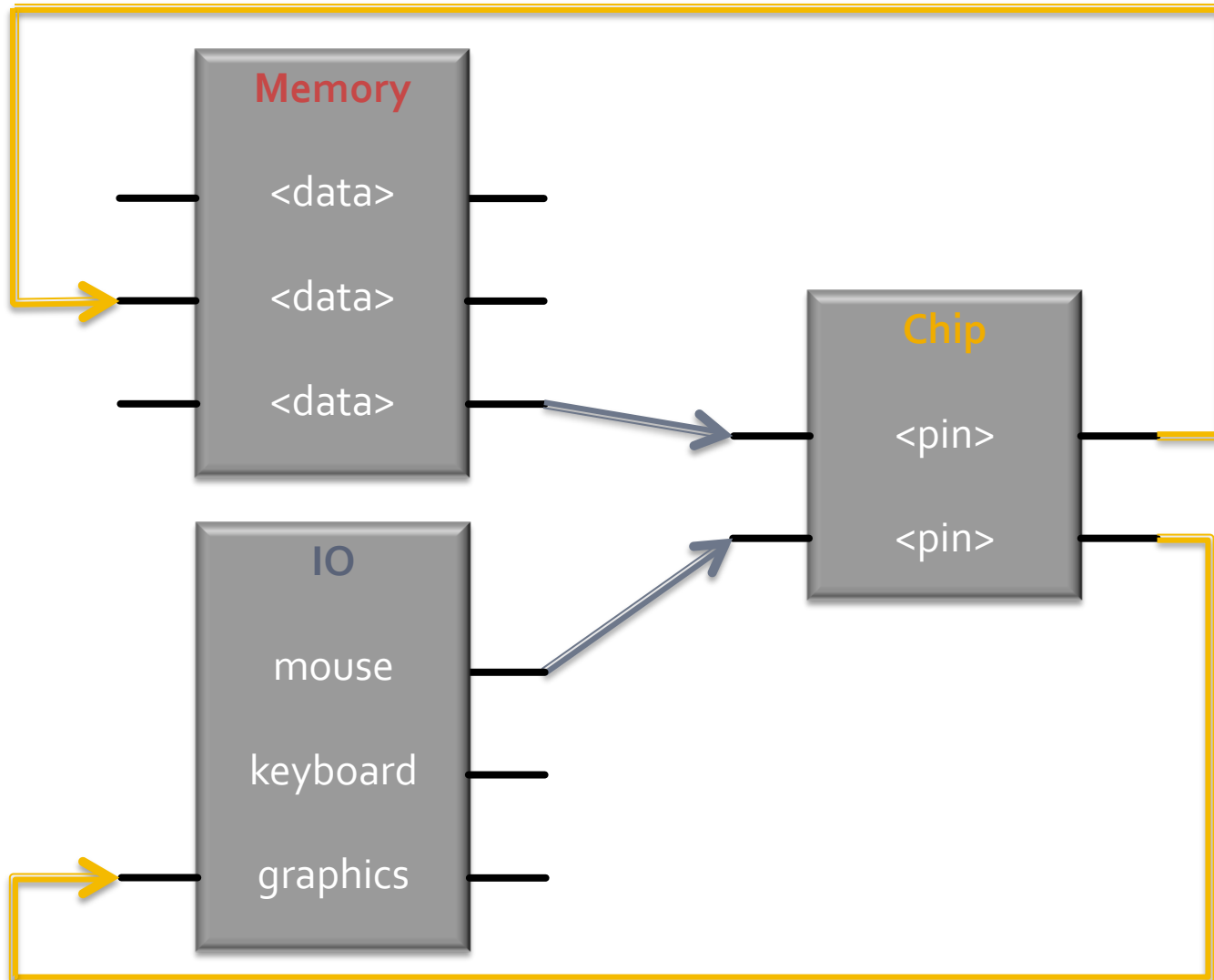
flowhub



# Putting order



# IO

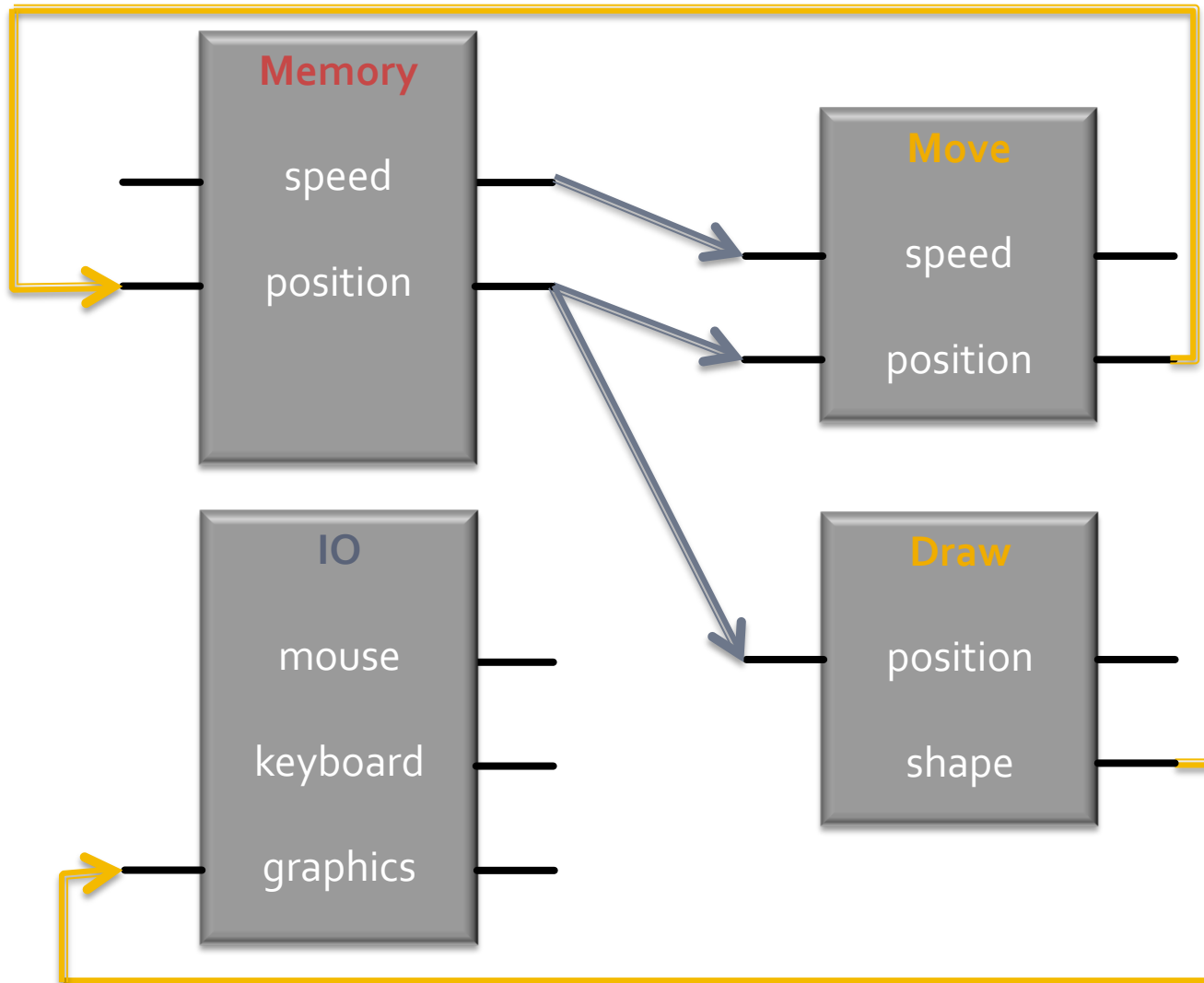


1. IO & Memory -> Chips

2. Chips compute

3. Chips -> IO & Memory

# Collaboration





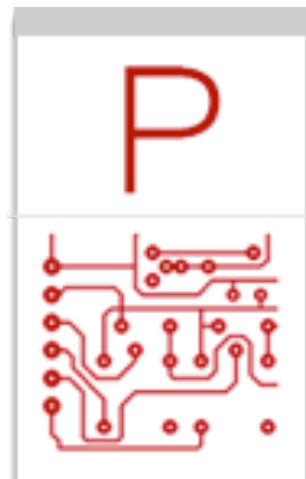
# Chips



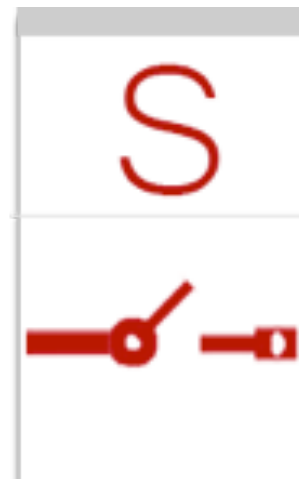
Emitter



Splitter

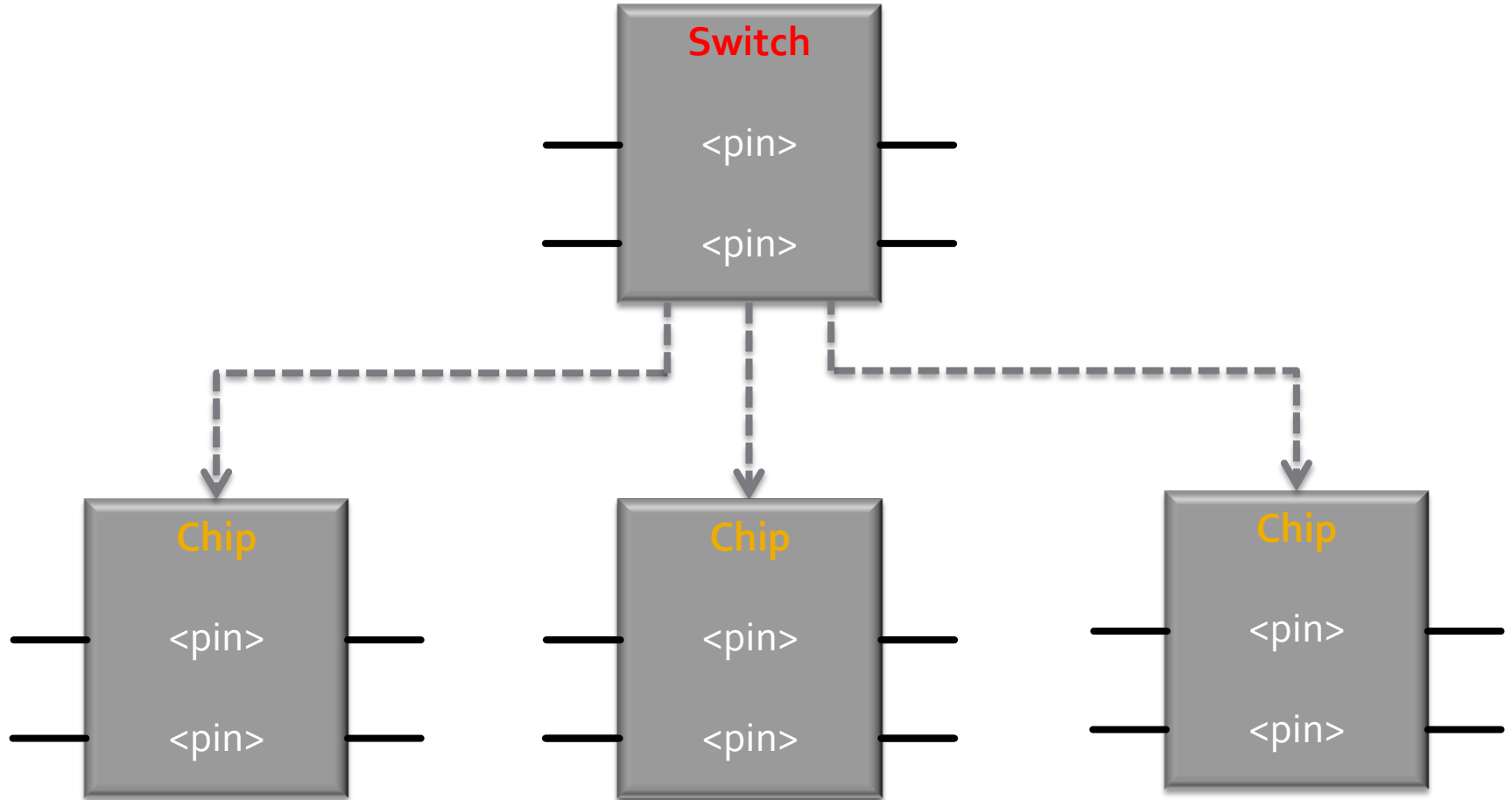


Processor



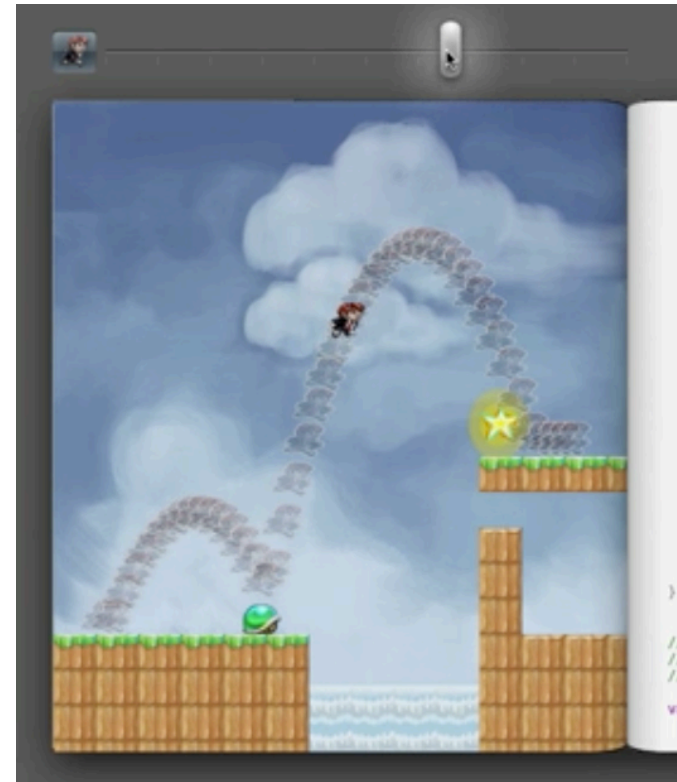
Switch

# Switches



# Cool debugging tools

- “What would have happened”
- “Mute” chips
- Find chips from memory or IO
- Test each chip in isolation



Brett Victor  
“Inventing on Principle”

# Technical Challenges

---

# Sandboxing



- User code is dangerous
- Contain within iframe with sandbox attribute
- Messages replace function calls

Parent page - Editor

Sandbox - Game execution

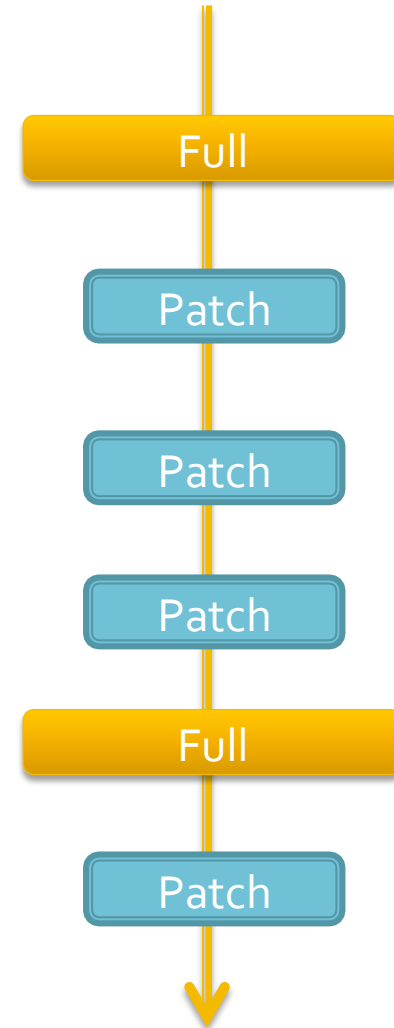
```
iframe  
sandbox="allow-scripts"
```

# Patching

- Two chips modifying the same array/object?
- Like version control
  1. Create patches from a common ancestor
  2. Merge patches
- Some merges are hard
  - LCS (Longest Common Subsequence)
  - Operational transforms

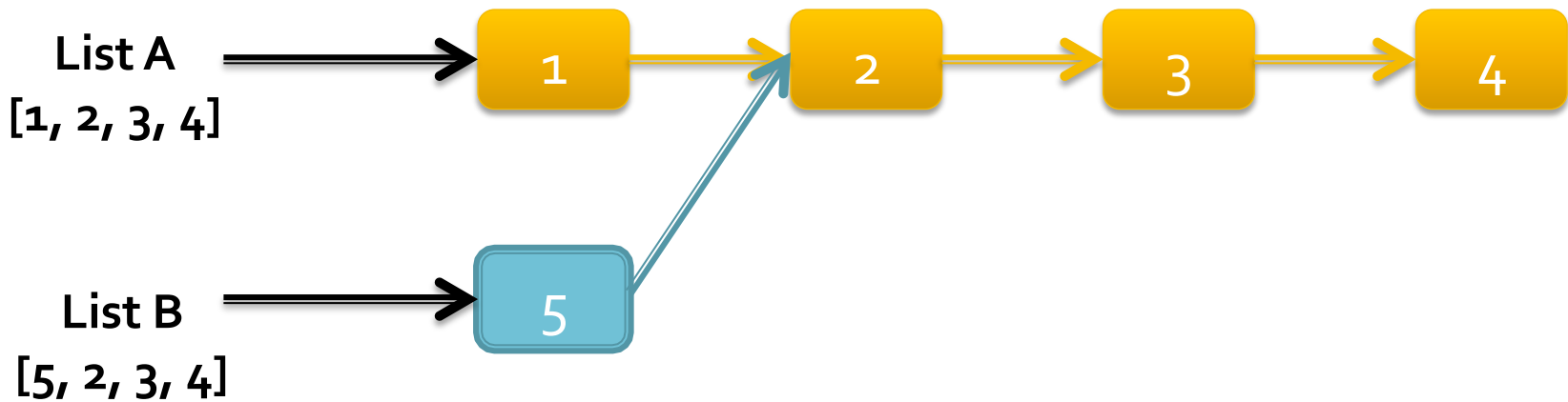
# History

- Track *all* past state
- Cloning data takes space!
- Better solutions
  - Store mostly patches



# History

- Persistent data structures
  - Applies to lists, trees, hash maps, etc.

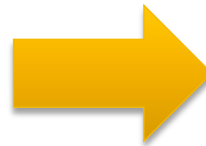




# Buffer IO

- Treat IO as a buffer instead of a set of methods
- Input
  - Mouse (`getMousePos()` -> `mouse.position`)
  - Keyboard (`isKeyDown()` -> `keyboard.keysDown`)
- Output
  - Canvas

```
ctx.fillStyle = "black";  
ctx.fillRect(0, 0, 100, 100);
```



```
canvas.shapes = [{  
  fillStyle: "black",  
  position: [0, 0],  
  size: [100, 100]  
}];
```

# Buffer IO

- Input/Output
  - HTML using Rivets.js
    - Template + values
    - 2-way data binding
  - HTTP
    - Data replaces events and callbacks!

RIVETS.JS

**Almost done...**

---

# We need you!

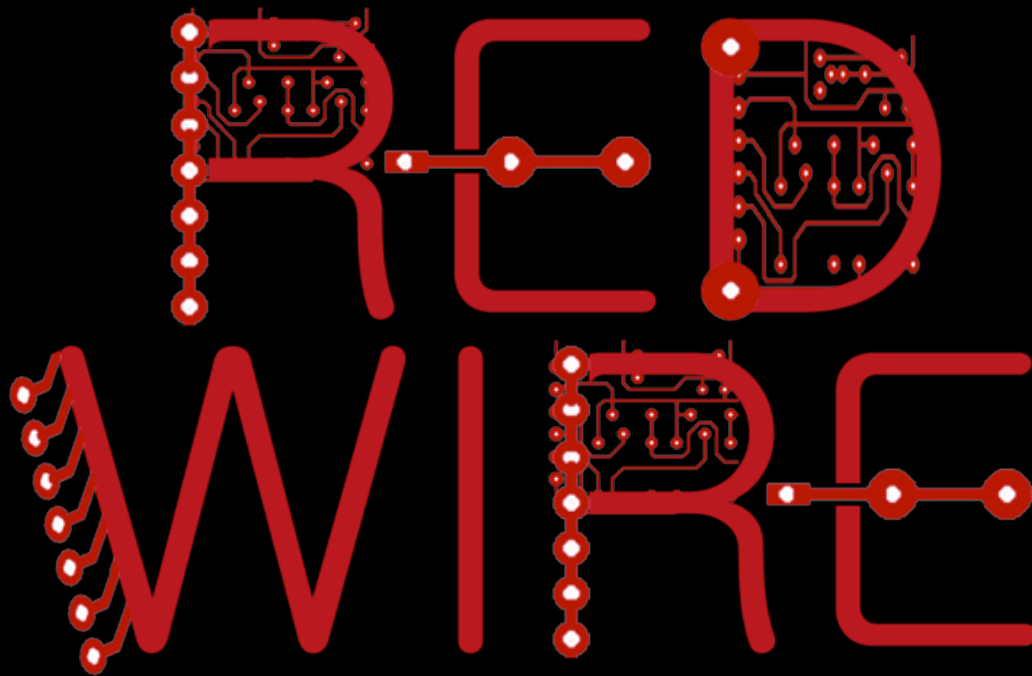
- New version landing Feb 14th
- Looking for new and different games
- Developers and designers always welcome!
  - Lots of new features to work on



## Paris Game Club - Making, playing, learning

- Invited speakers
- Micro-game jams
- Videos + agenda at

[gamelier.org](http://gamelier.org)



[github.com/CyberCRI/RedWire](https://github.com/CyberCRI/RedWire)

@RedWireIO

[jesse.himmelstein@parisdescartes.fr](mailto:jesse.himmelstein@parisdescartes.fr)

+JesseHimmelstein

@himmelattack