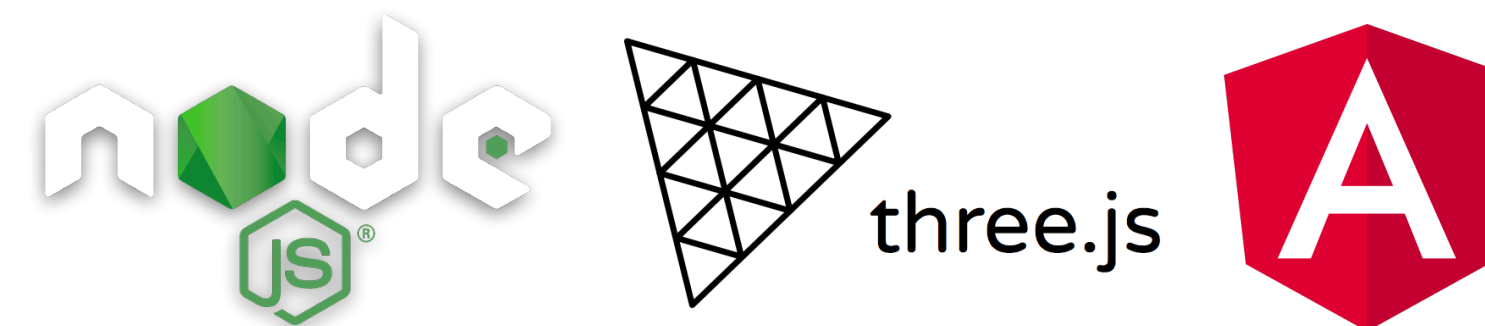




EDWARD MOYSE

PHOENIX

- ▶ Phoenix is an *experiment agnostic* event display, and owned by the HSF visualisation group:
 - ▶ Github
 - ▶ <https://github.com/HSF/phoenix>
 - ▶ Demo:
 - ▶ <http://hepsoftwarefoundation.org/phoenix/>
 - ▶ **Caveat:** *Feel free to try it, but (on macos) I find Firefox gives a better performance than Safari, so recommend this*
- ▶ Runs ~entirely in the browser, so no powerful servers needed
- ▶ Uses [three.js](#) and [angular](#), nodeJS, NPM (+ other libraries)
- ▶ Currently has support for LHCb, ATLAS and TrackML **geometry** and **event data**

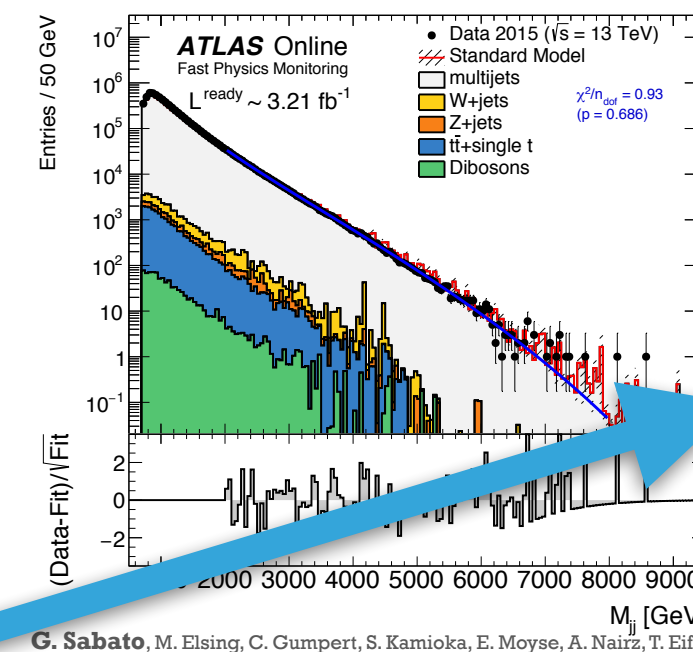


- ▶ It wasn't always called Phoenix!
- ▶ **2015** : first import to [repository](#) of 'jseventdisplay'
- ▶ **2016** : simple visualisation for TADA
- ▶ **2016** : new version ('webeventdisplay) to show 'procedurally generated geometries'
- ▶ i.e. can use very simple JS to show a test detector

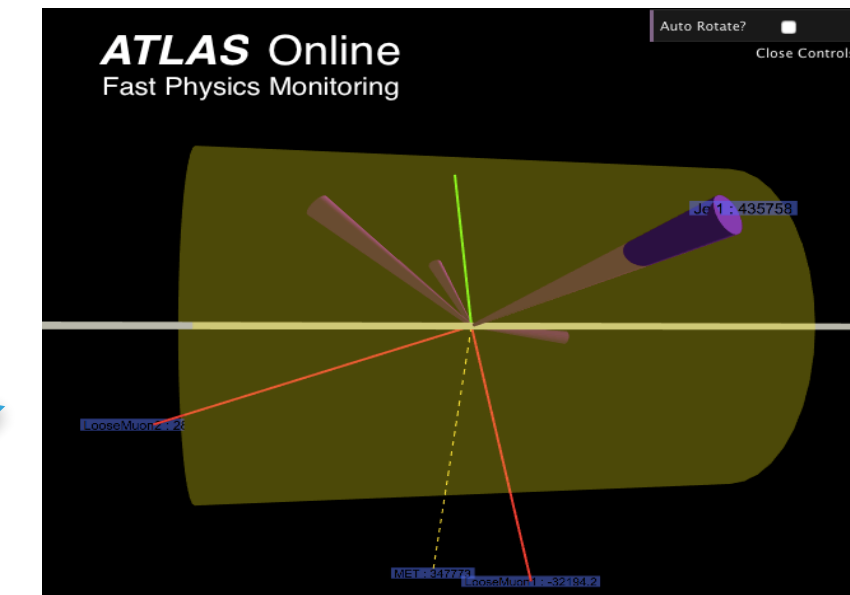
+

TADA: Fast search for new physics

- ▶ Many channels sensitive to broad spectrum of new physics are implemented in TADA, grouped by:
 - SM, top, Higgs, exotics and SUSY searches
- ▶ Distributions, as invariant mass of ee and di-jet events, are displayed on the TADA webpage
- ▶ TADA provides as well a built-in event viewer to inspect interesting events



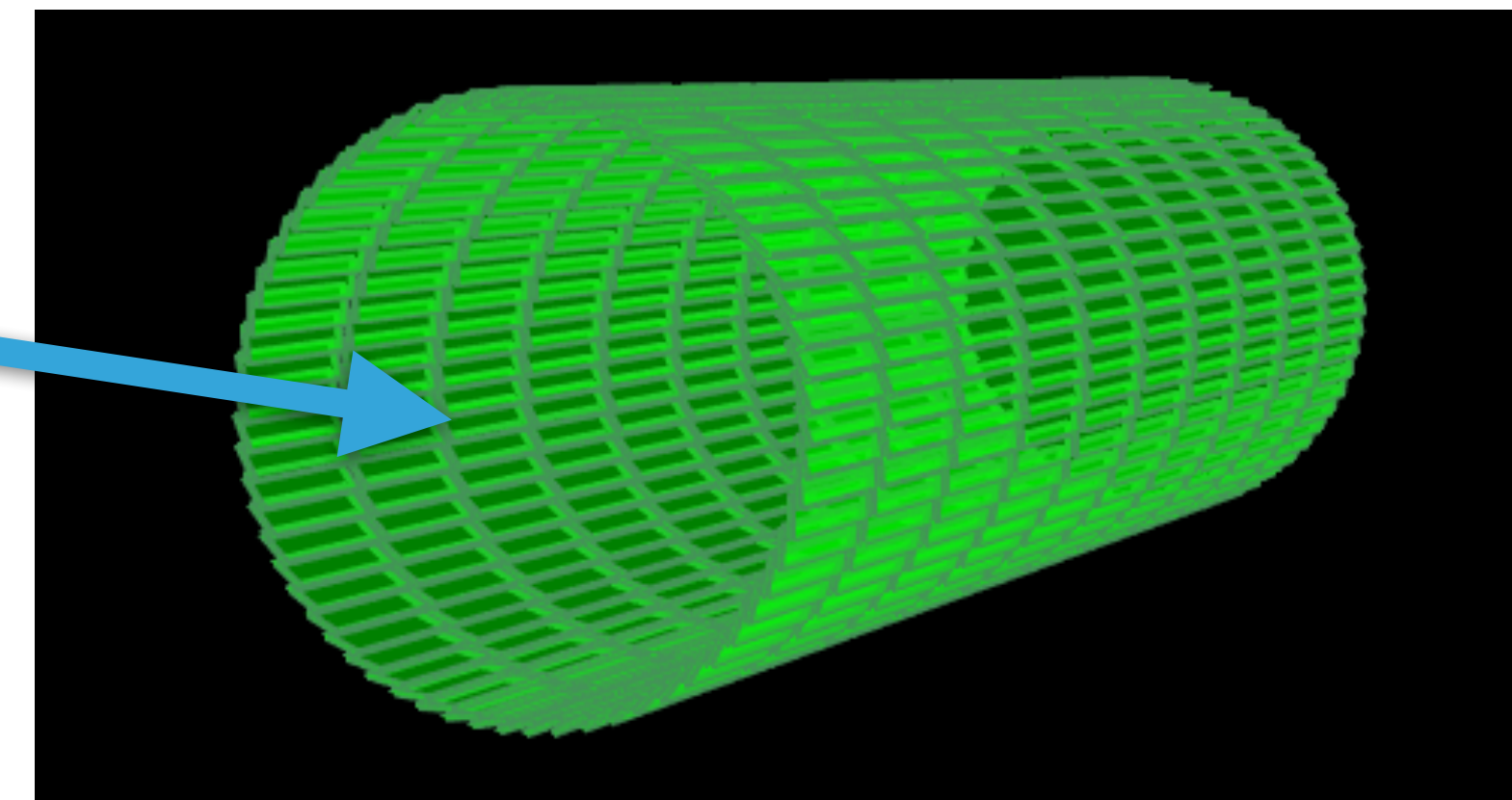
G. Sabato, M. Elsing, C. Gumpert, S. Kamioka, E. Moyse, A. Nairz, T. Eifert



ATLAS Fast Physics Monitoring: TADA

CHEP 2016

CHEP presentation 2016



```
var parameters = { ModuleName: "Module 2", Xdim:10.,  
Ydim:1., Zdim:45, NumPhiEl:64, NumZEl:10, Radius:75,  
MinZ:-250, MaxZ:250, TiltAngle:0.3, PhiOffset:0.0,  
Colour:0x00ff00, EdgeColour:0x449458 };  
  
window.EventDisplay.buildGeometryFromParameters(parameters)  
;
```

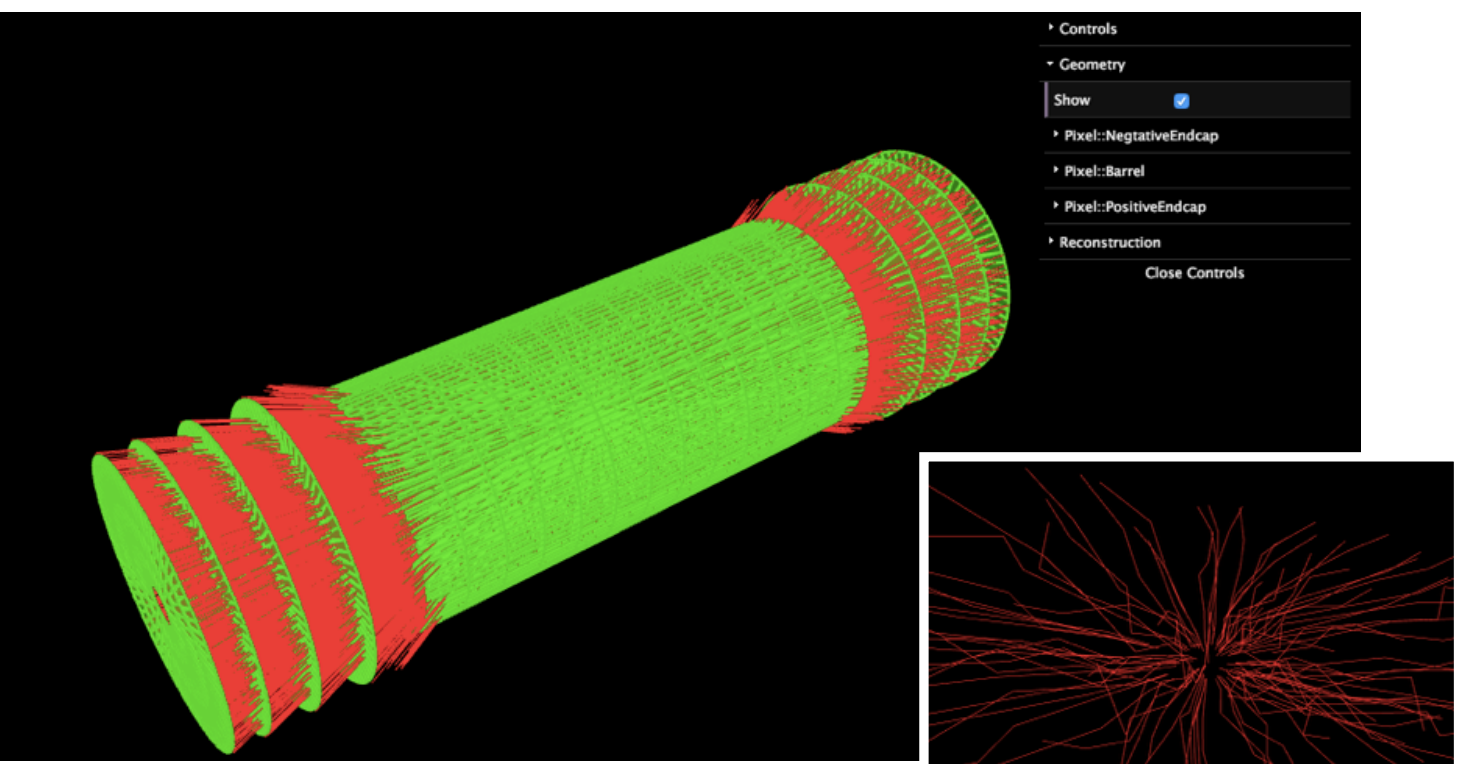

- ▶ **2016** Shown in two talks at [Connecting The Dots](#)

ATS release planning - alpha

0.1.0
alpha release: first repository build Athena/Gaudi
Release date: 19/Feb/16
14 Issues · [Release Notes](#)

14 of 14 issues have been resolved

WebEventDisplay by Edward Moyses



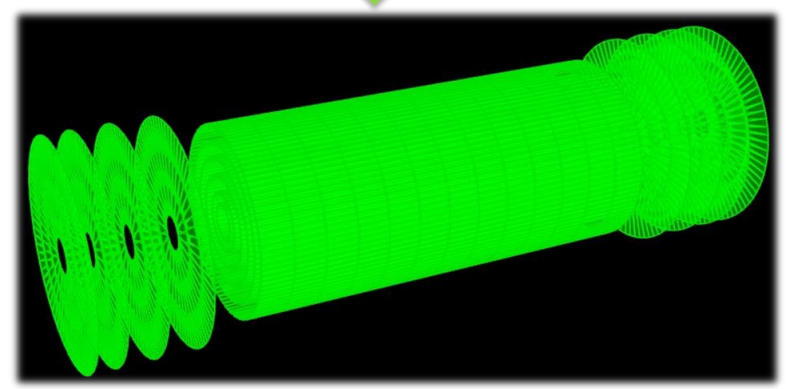
the frank reason why we haven't released

18

<https://indico.hephy.oeaw.ac.at/event/86/session/5/contribution/37>

Example: GenericDetectorPlugin

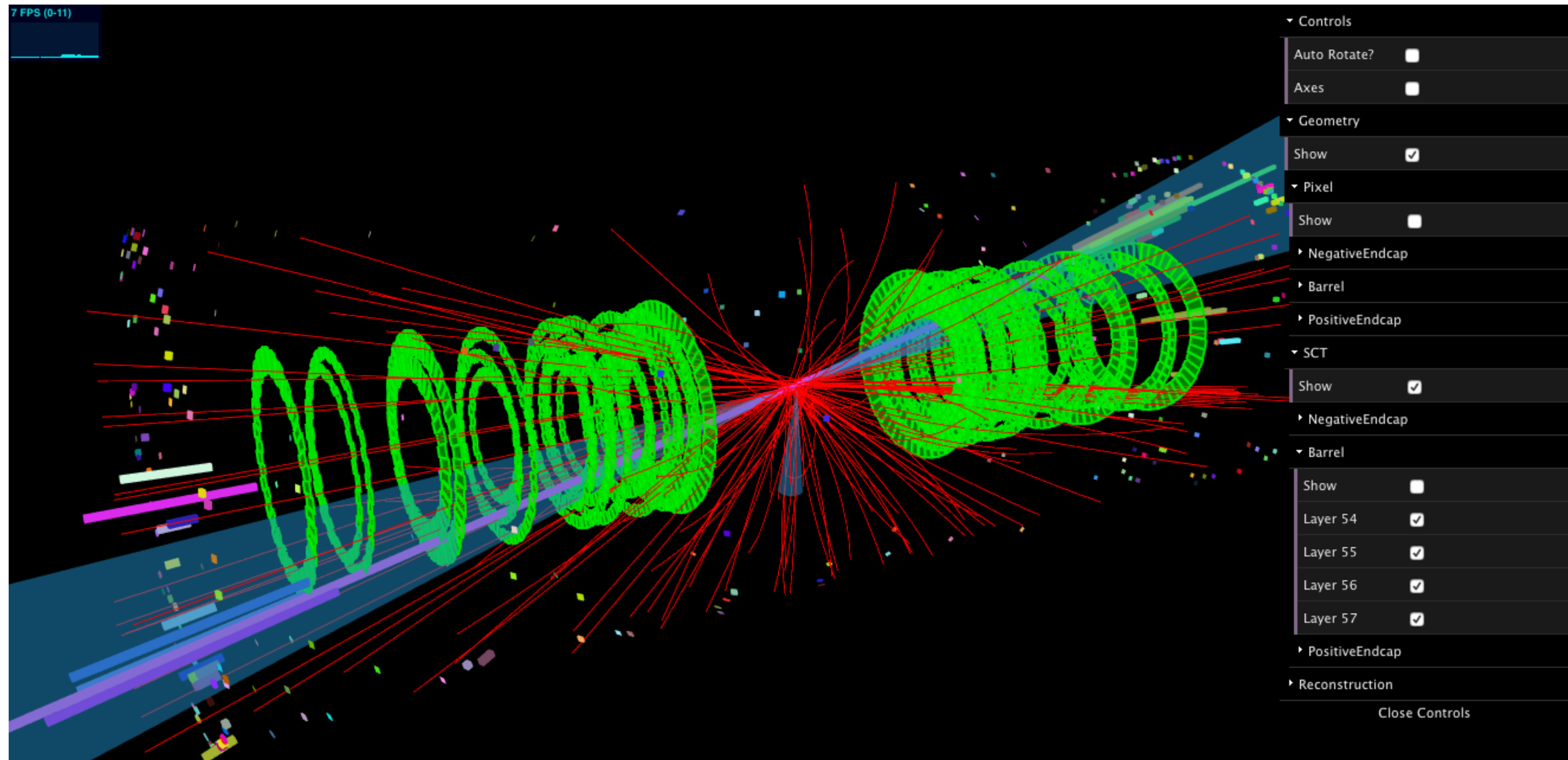
```
21 # the pixel module
22 PixelModule = DetectorModule(None,8.4,32.0,0.15)
23 # the first layer
24 PixelLayer0 = CylinderLayer(PixelModule, 33., 24, 13, 0.2, 2., 0.5, 5.)
25 PixelLayer1 = CylinderLayer(PixelModule, 55., 40, 13, 0.2, 2., 0.5, 5.)
26 PixelLayer2 = CylinderLayer(PixelModule, 88., 60, 13, 0.2, 2., 0.5, 5.)
27 PixelLayer3 = CylinderLayer(PixelModule, 120., 72, 13, 0.2, 2., 0.5, 5.)
28 PixelLayer4 = CylinderLayer(PixelModule, 150., 84, 13, 0.2, 2., 0.5, 5.)
29 # define the pixel barrel volume
30 PixelBarrel = BarrelVolume( [ PixelLayer0, PixelLayer1, PixelLayer2, PixelLayer3, PixelLayer4 ] )
```



24/02/2016 J.Hrdinka, A.Salzbürger, C.Gumpert - Overview of the ATS project and its use for FCC - Connecting the Dots 2016 15

<https://indico.hephy.oeaw.ac.at/event/86/session/5/contribution/36>

- ▶ **2017** - Added JSON dumpers to VP1, and visualised more complex data



SHORT HISTORY

- ▶ **2017** HSF [visualisation white paper](#) identified the desirability of having a common event format, and a common tool : renamed to **Phoenix** and imported to GitHub

- ▶ <https://github.com/HSF/phoenix>

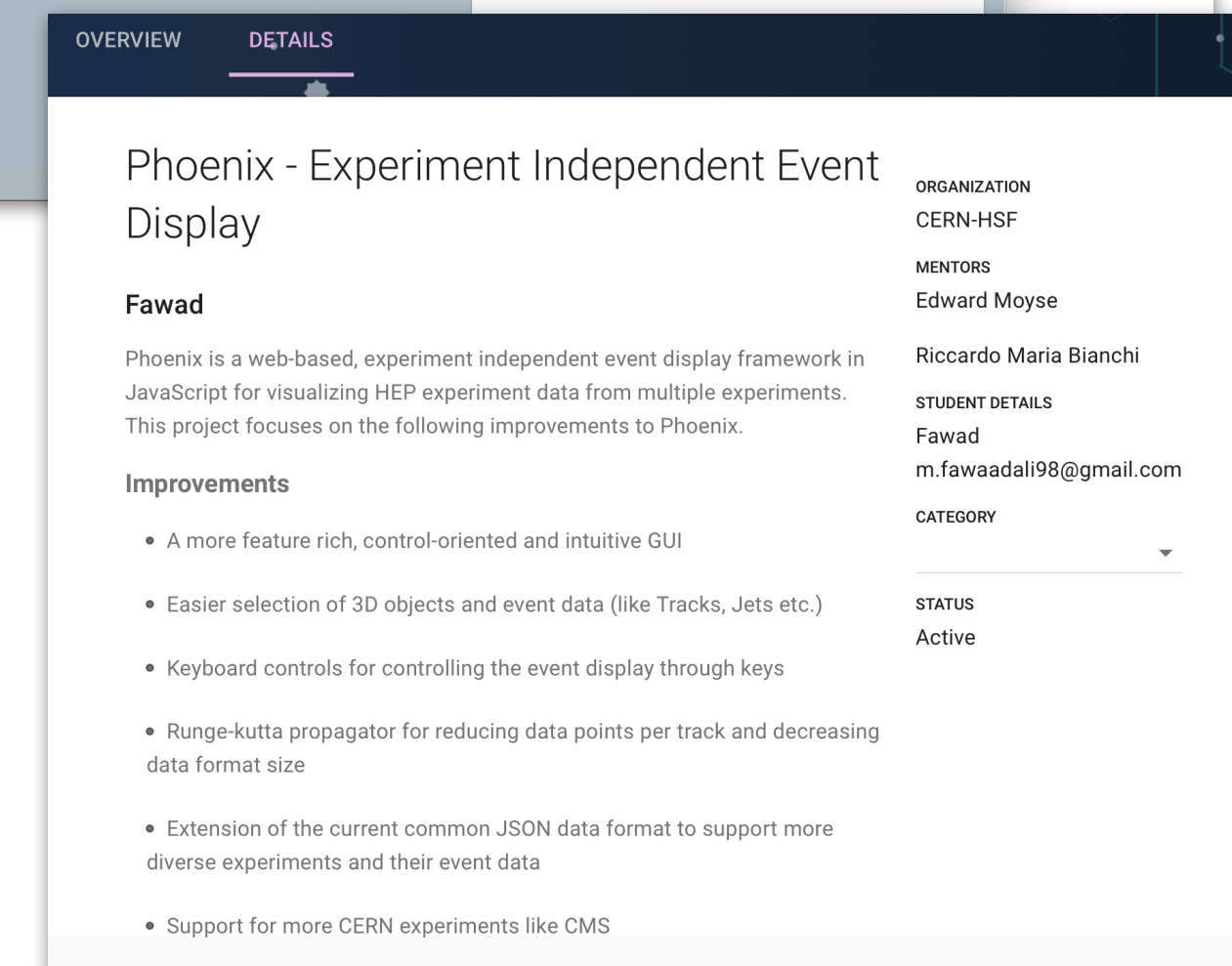
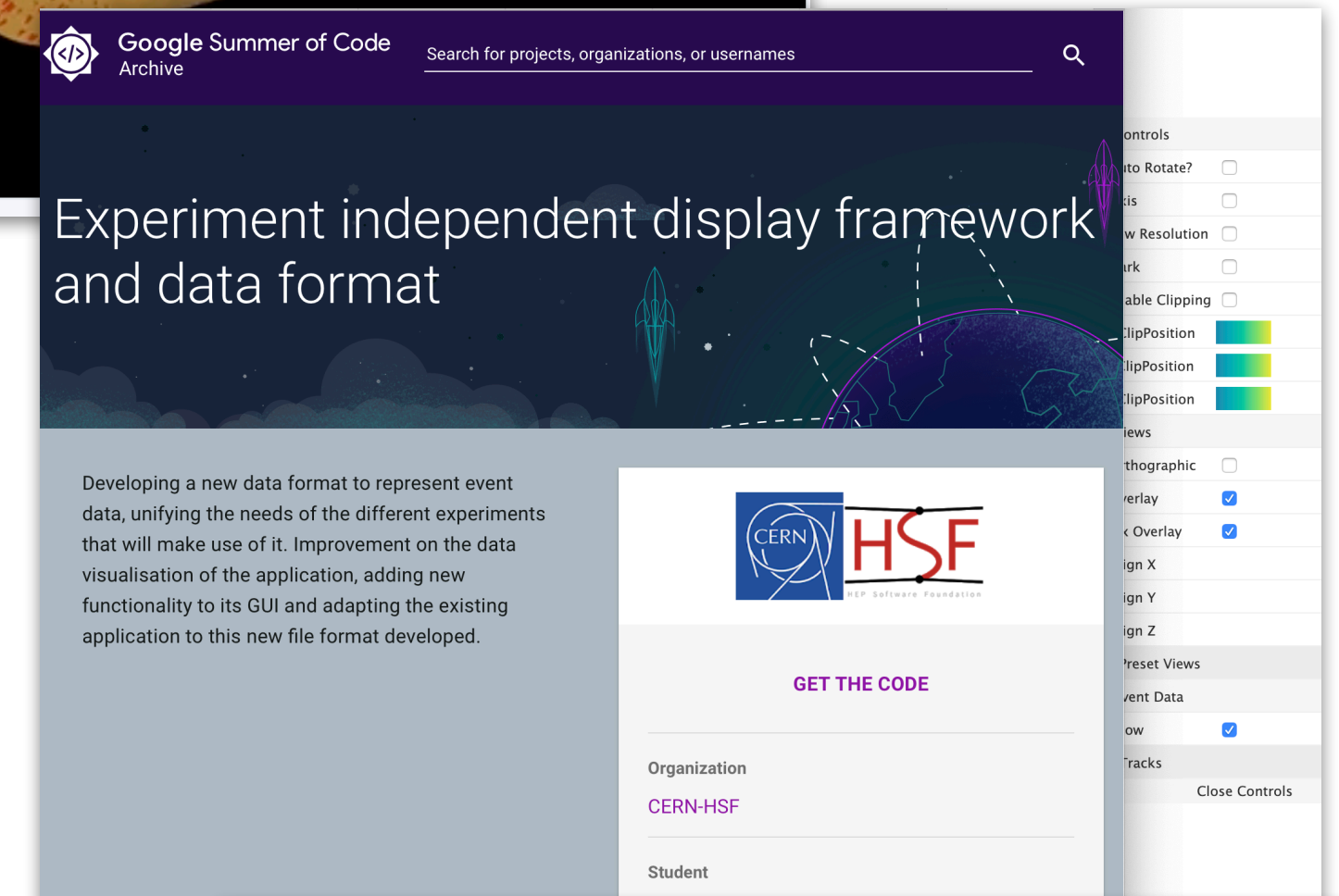
- ▶ Apache Licence

- ▶ **2018** Used for **TrackML** challenge

- ▶ **2019** Accepted for [Google Summer of Code](#)

- ▶ Student (Emilio) presented Phoenix at "[Learning to Discover - Advanced pattern Recognition](#)"

- ▶ **2020 Accepted again for GSOC! (Fawad Ali)**



- ▶ In order to be accessible to many experiments, some key concepts:
 - ▶ **Permissive licence and open source** (Apache Licence)
 - ▶ **Use industry standards** where possible
 - ▶ e.g. angular, nodeJS, common formats for geometry
 - ▶ **Simple standard format for Event Data**
 - ▶ **Good documentation,**
 - ▶ **Don't make experiment specific assumptions,**
 - ▶ Make Phoenix **configurable, extendable and modular**

▶ Angular

- ▶ Probably the largest, most popular framework for dynamic web pages with many powerful features
 - ▶ Dynamic loading
 - ▶ TypeScript (**vast** improvement over plain JavaScript)

▶ NodeJS

- ▶ Run-time environment for applications written in JavaScript language

▶ NPM

- ▶ Package manager for Node packages, takes care of dependencies etc

▶ Threejs

- ▶ Hugely impressive open source library for 3d in browser.

▶ 3D formats

- ▶ One problem with a common event display is specialised geometry formats (my impression is ATLAS is particularly bad)
- ▶ **Phoenix** currently supports OBJ and glTF (*threejs* supports [LOADS](#) so would be easy to add more if necessary)

USE INDUSTRY STANDARDS

▶ We make use of many standard features provided by Github + co

▶ Travis CI on each PR

▶ Coverage tests

▶ Vulnerability scans

[\[link\]](#)

[\[link\]](#)

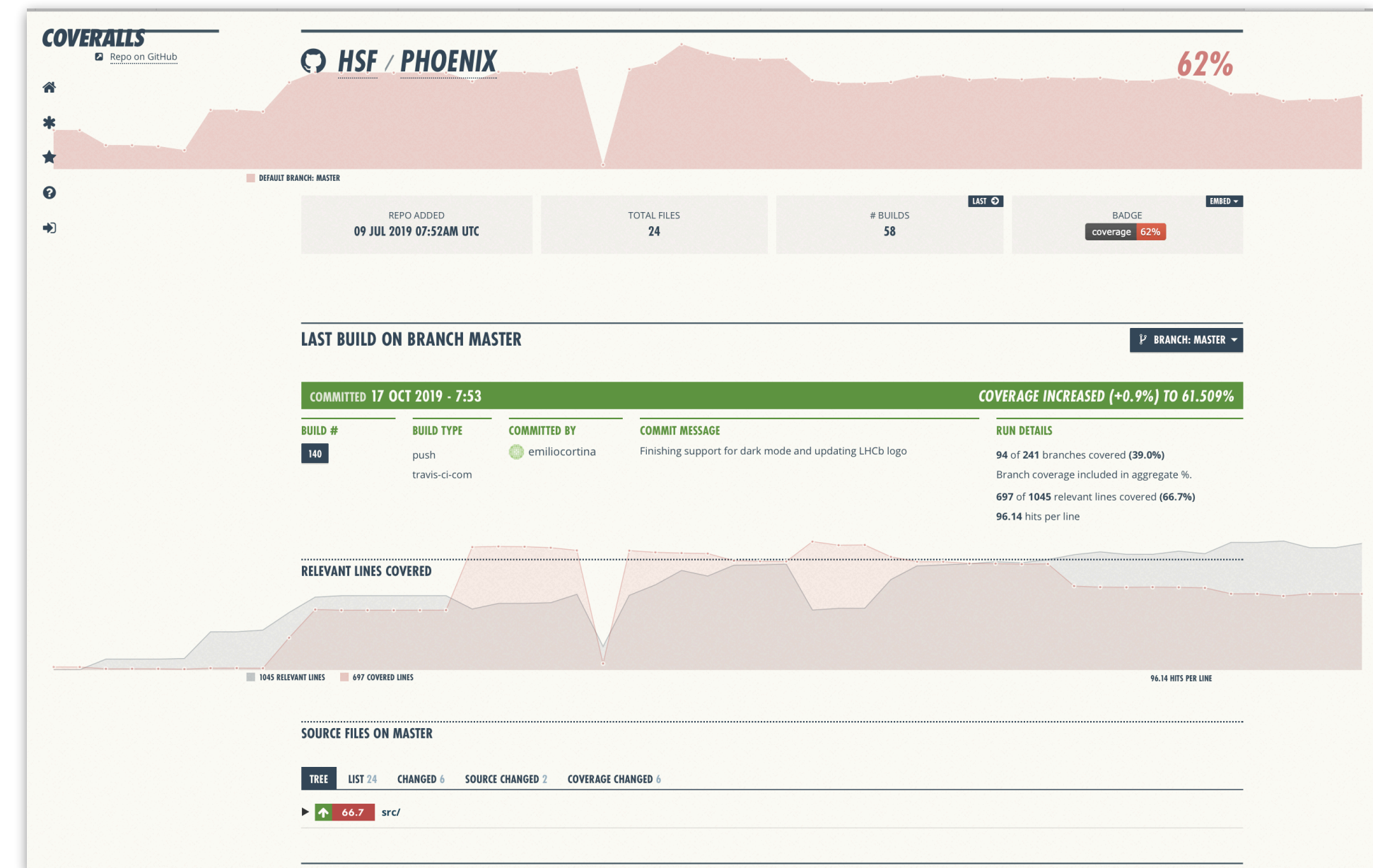
[\[link\]](#)

emiliocortina merged commit 922891e into HSF:master on Aug 18, 2019

Hide details Revert

1 of 2 checks passed

- ✗ coverage/coveralls Coverage decreased (-0.2%) to 65.484% [Details](#)
- ✓ Travis CI - Pull Request Build Passed [Details](#)



Vulnerabilities 0 via 0 paths | Dependencies 74 | Source GitHub

Commit 1e89ded1

Find, fix and prevent vulnerabilities in your code.

Test and protect my applications alternatively follow this public repository to track its vulnerability status

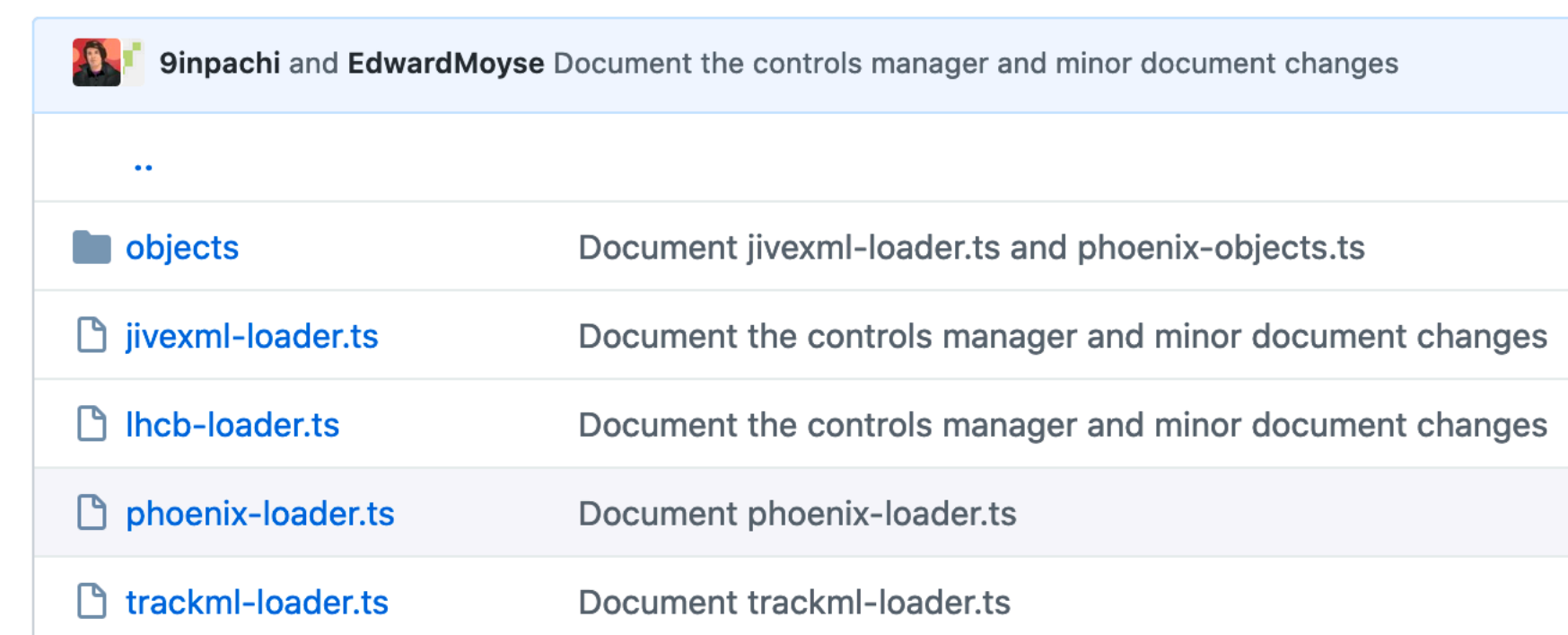
Issues Dependencies

No known vulnerabilities found

- ▶ One initial decision was to, by default, use JSON for event data
 - ▶ Very easy to manipulate (especially in typescript) and extremely well supported format
 - ▶ Human readable, but more compact than XML e.g.

```
{ "event number":123, "run number":234, "Tracks" : {"Inner Detector Tracks": [ {"chi2":52.1087, "dof":34, "params": [-0.0150713, 0.725162, 2.11179, 2.86823, -3.23906e-05], "pos": [] } ] }
```

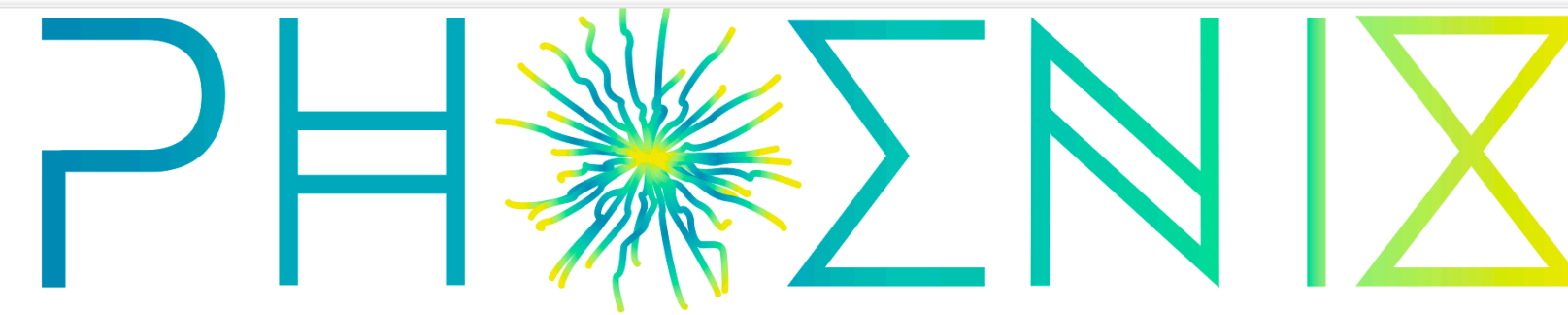
- ▶ Documented [here](#)
- ▶ However, possible to add “loaders”:
 - ▶ Can support arbitrary formats



A screenshot of a GitHub commit by 9inpachi and EdwardMoyle. The commit message is "Document the controls manager and minor document changes". Below the message is a list of files that were added or modified in the commit:

File	Description
objects	Document jivexml-loader.ts and phoenix-objects.ts
jivexml-loader.ts	Document the controls manager and minor document changes
lhcb-loader.ts	Document the controls manager and minor document changes
phoenix-loader.ts	Document phoenix-loader.ts
trackml-loader.ts	Document trackml-loader.ts

- ▶ Good documentation is very important.
- ▶ Main [README.md](#) explains how to
 - ▶ install & run phoenix,
 - ▶ contribute,
 - ▶ developers guide



Phoenix Project

The idea of this project is to have a simple way to visualise event and geometry data using nothing more than a web browser. The data should be as detector-agnostic as possible. Technically the 3D is done with [three.js](#), the menu (at the moment) uses [DAT.GUI](#) and the data format is just plain JSON.

This project was generated with [Angular CLI](#) version 7.3.4.

You can see an online version at <https://hepsoftwarefoundation.org/phoenix/>

Build and install

This is an [Angular](#) application, so you will need to have [Node js](#) and [Angular CLI](#) installed locally.

You can follow [this guide](#) to set up your local environment.

Once everything is set up, run `ng serve` from the command line in the project directory for a dev server. Then navigate to `http://localhost:4200/` from your browser. The app will automatically reload if you change any of the source files.

Documentation

- [User manual.](#)
- [How to contribute.](#)
- [Developer guide.](#)

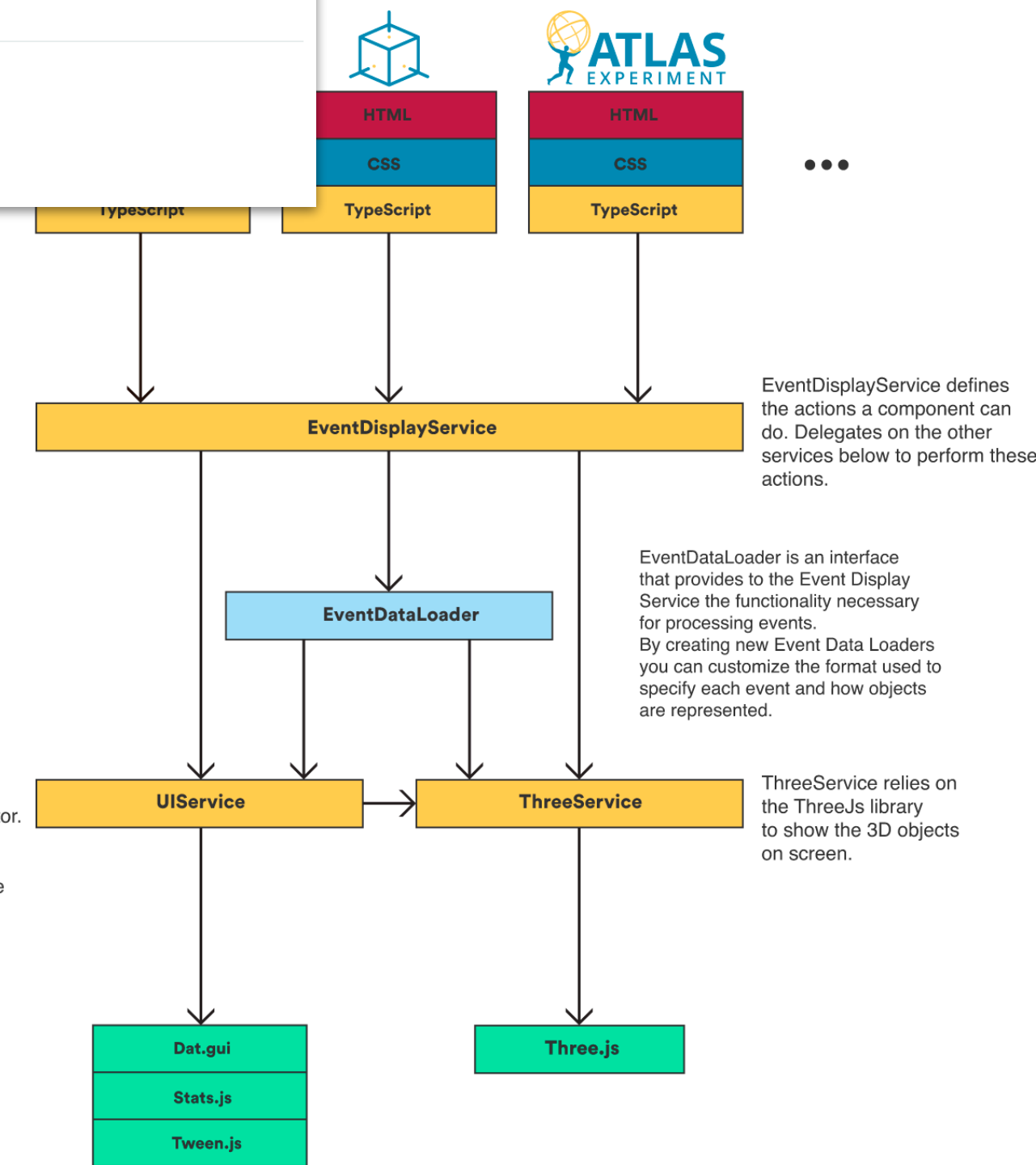
Services

Provide functionality that is common to every component.

UIService is in charge of placing UI elements such as the controls and performance monitor. It has a reference to ThreeService to allow UI elements manipulate the 3D objects.

Node Modules

External libraries that the services use to provide the functionality. They are imported with Node Package Manager (npm).



- ▶ We now also, thanks to Fawad!, have very nice API documentation, generated using [compodoc](#)
- ▶ (Temporary) link:
 - ▶ <https://9inpachi.github.io/phoenix/docs/api-docs/classes/JiveXMLLoader.html>

The screenshot shows the API documentation for the `JiveXMLLoader` class. On the left is a navigation sidebar with a list of classes including `Configuration`, `ControlsManager`, `Cut`, `ExportManager`, `ImportManager`, `JiveXMLLoader` (highlighted), `LHCbLoader`, `PhoenixLoader`, `PhoenixObjects`, `PresetView`, `RenderManager`, `SceneManager`, `SelectionManager`, `TrackmlLoader`, `Injectables`, `Interfaces`, `Routes`, and `Documentation coverage`. The main content area is titled "Classes / JiveXMLLoader" and contains tabs for "Info" and "Source". Under the "File" section, the path `src/app/services/loaders/jivexml-loader.ts` is shown. The "Description" section states: "PhoenixLoader for processing and loading an event from the JiveXML data format." The "Extends" section lists `PhoenixLoader`. The "Index" section is divided into "Properties" and "Methods".

Properties

Private data	Private graphicsLibrary
Private eventData	Private ui

Methods

Public getCaloClusters	Public getTRT_DriftCircles	Public getEventData
Public getEventData	Public process	Public getJets
Public getJets	Private addCollection	Public getPixelClusters
Public getPixelClusters	Private addObjectType	Public getSCTClusters
Public getSCTClusters	Public buildEventData	Public getTracks
Public getTracks	Public getCollection	

Documentation generated using

MODULAR, EXTENDIBLE, NO EXPERIMENT-SPECIFIC INSTRUCTIONS

- ▶ Cannot assume that we have symmetric geometry layout like ATLAS or CMS
- ▶ Must make re-usable core modules, with easy-to-configure GUI and functionality
 - ▶ ... and then experiment/detector specific implementations
- ▶ Make it easy to “roll-your-own” version
- ▶ Must make it possible to read data other than ‘standard’ format

AN EXAMPLE... LHCb

(A good example of a non-z-axially-symmetric detector)

- ▶ Work done by Ben Couturier (from LHCb)
- ▶ [PR](#) has 3 commits - *relatively* little work
- ▶ Took him ~1 week IIRC

Conversation 1 | Commits 3 | Checks 1 | Files changed 8

bcouturi commented on Oct 2, 2019 Contributor + 😊 ...

This is prototype of what I could do with the LHCb run1 detector, and an example event.

bcouturi added 3 commits on Sep 18, 2019

- Initial version with an LHCb event and a slider for detector opacity 909a83b
- Fix typescript errors 13d3780
- Merge branch 'master' of <https://github.com/HSF/phoenix> into lhcb_event ❌ 6183905

EdwardMoyse approved these changes on Oct 3, 2019 View changes

> 15	■■■■■	src/app/sections/lhcb/lhcb.component.ts	Viewed ...
> 2	■ ■ ■ ■	src/app/services/eventdisplay.service.ts	Viewed ...
> 2	■ ■ ■ ■	src/app/services/loaders/configuration.model.ts	Viewed ...
> 7	■■■■■	src/app/services/loaders/lhcb-loader.spec.ts	Viewed ...
> 45	■■■■■	src/app/services/loaders/lhcb-loader.ts	Viewed ...
> 17	■■■■■	src/app/services/three.service.ts	Viewed ...
> 4	■ ■ ■ ■	src/app/services/ui.service.ts	Viewed ...
> 1	■ ■ ■ ■	src/assets/files/lhcb/00191749_0005296728.json	Viewed ...

- ▶ Have LHCb section:
 - ▶ phoenix/src/app/sections/lhcb
- ▶ In there you need 4 files, main one that matters is the XXX.components.ts
- ▶ Here we define:
 - ▶ The loader
 - ▶ The configuration
 - ▶ How to get event data
- ▶ Everything else is Core Phoenix

emiliocortina Refactoring and commenting code Latest commit 4da7ea3 on Oct 15, 2019

..		
lhcb.component.html	Initial quick test	5 months ago
lhcb.component.scss	Initial quick test	5 months ago
lhcb.component.spec.ts	Initial quick test	5 months ago
lhcb.component.ts	Refactoring and commenting code	3 months ago

45 lines (39 sloc) | 1.58 KB

```
1 import {Component, OnInit} from '@angular/core';
2 import {EventdisplayService} from '../services/eventdisplay.service';
3 import {Configuration} from '../services/extras/configuration.model';
4 import {PresetView} from '../services/extras/preset-view.model';
5 import {HttpClient} from '@angular/common/http';
6 import {LHCbLoader} from '../services/loaders/lhcb-loader';
7
8
9 @Component({
10   selector: 'app-lhcb',
11   templateUrl: './lhcb.component.html',
12   styleUrls: ['./lhcb.component.scss']
13 })
14 export class LHCbComponent implements OnInit {
15   events: any;
16   loader: LHCbLoader;
17
18   constructor(private eventDisplay: EventdisplayService, private http: HttpClient) {
19   }
20
21   ngOnInit() {
22     const configuration = new Configuration();
23     configuration.presetViews = [
24       new PresetView('Right View', [0, 0, 6000], 'right'),
25       new PresetView('Center View', [-500, 1000, 0], 'circle'),
26       new PresetView('Left View', [0, 0, -6000], 'left')
27     ];
28     this.eventDisplay.init(configuration);
29     this.eventDisplay.loadGLTFDetector('assets/geometry/LHCb/lhcb.gltf');
30     this.loader = new LHCbLoader();
31     configuration.eventDataLoader = this.loader;
32     this.loadEventData(configuration);
33
34   }
35
36   private loadEventData(config: Configuration) {
37     this.http.get('assets/files/lhcb/00191749_0005296728.json').subscribe((data: any) => {
38       this.loader.process(data);
39       const eventData = this.loader.getEventData();
40       this.eventDisplay.buildEventDataFromJSON(eventData);
41       // TODO Current implementation throws error
42       // this.eventDisplay.setDetectorOpacity(config.detectorOpacity);
43     });
44   }
45 }
```


Phoenix

Playground Geometry More

Controls

- Auto Rotate?
- Axis
- Low Resolution
- Dark
- Enable Clipping
- xClipPosition
- yClipPosition
- zClipPosition

Views

- Orthographic
- Overlay
- Fix Overlay
- Align X
- Align Y
- Align Z

Preset Views

Event Data

- Show

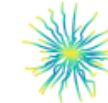
Tracks

Close Controls

25 FPS (7-54)

Even with very complex geometry, still get 25 FPS

Preset views





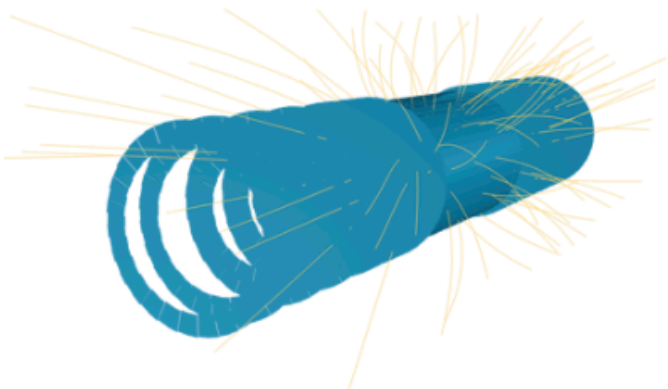



WALKTHROUGH



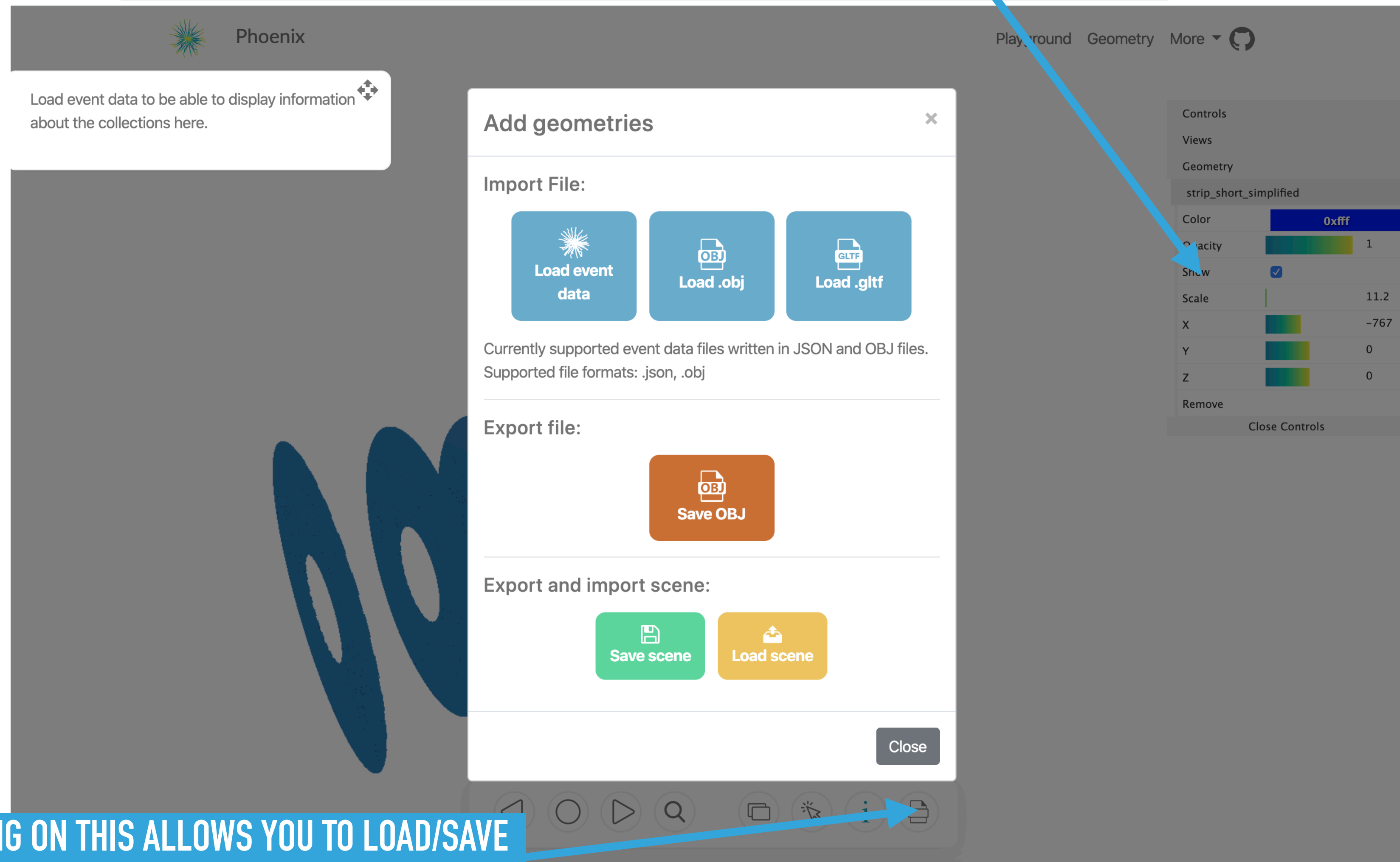
Application for visualizing High Energy Physics data.

- ▶ This is the homepage!
- ▶ Let's do a very quick walkthrough some of this
 - ▶ Best is for you to try this yourself of course, remembering that performance is not optimal with Safari
- ▶ Will start with **Playground**, which is a bit special
 - ▶ Intended primarily for testing new detectors
 - ▶ Should have CMS here too
 - ▶ (Work is ~done but needs a cleanup)

 <p>Playground Get started with the different Phoenix features.</p> <p>Show</p>	 <p>Geometry display This test should show some simple geometry.</p> <p>Show</p>	 <p>ATLAS Show the ATLAS detector. One simple event.</p> <p>Show</p>
 <p>LHCb Show the LHCb detector. One simple event.</p> <p>Show</p>	 <p>TrackML Visualisation for TrackML. Shows how to write a custom event loader.</p> <p>Not Available</p>	 <p>Playground VR Get started with the different Phoenix features in VR.</p> <p>Show</p>

PLAYGROUND

2. ONCE LOADED, YOU CAN SCALE AND MOVE GEOMETRY, PLUS CHANGE COLOUR ETC



1. CLICKING ON THIS ALLOWS YOU TO LOAD/SAVE GEOMETRY, EVENT DATA, ENTIRE SCENE

Phoenix Playground Geometry More

Jet

coneR : 0.4
phi : 1.56555
eta : -0.121234
energy : 5943.56

Choose a collection: Choose

No.	chi2	dof	dparams	pos
#0	13.1858	19	0.244629,-29.2197,-1.06571,0.205962,0.000213952	12.3897,
#1	4.34172	18	-0.115155,4.52161,0.115751,2.93235,-0.000288252	
#2	74.3116	57	-0.0168833,3.54235,-1.84479,0.469622,0.000441226	-8.0534
#3	3.10637	16	-0.0599799,2.41625,3.10217,2.90213,-0.000223167	-27.2185
#4	16.7874	27	0.355117,38.1047,0.898299,0.254831,-0.000182632	
#5	5.66898	15	0.118117,-6.40268,-1.03025,0.219572,-0.000205163	13.502,-
#6	12.5637	36	0.12397,55.2766,-2.92716,0.309612,-0.000251071	-26.426
#7	8.53843	15	0.280532,56.24,1.01413,2.97544,-0.000242302	
#8	4.63204	15	-0.340253,-62.3096,1.49247,2.93067,0.000370728	
#9	19.361	27	-0.381201,3.08244,-0.828424,2.88526,-0.00043514	17.7086,
#10	40.34	46	-0.0388354,-7.18896,-3.00131,2.66459,-0.000190977	-26.867
#11	3.22256	14	-0.264261,9.92097,-1.35809,0.20283,0.0000896858	4.60277,

Controls

Views

Geometry

strip_short_simplified

Color 0xff

Opacity 0.4

Show

Z 0

Remove

Event Data

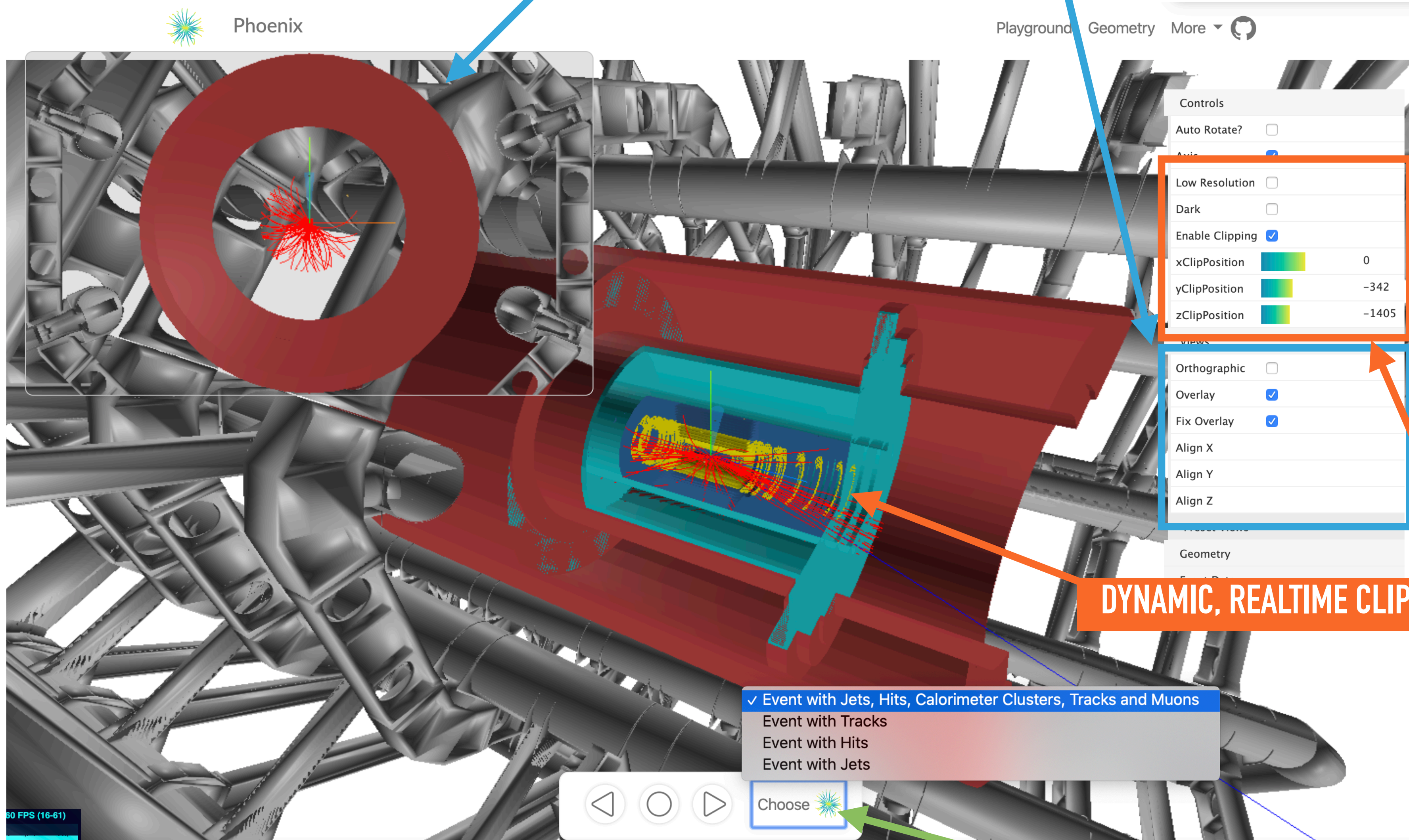
12 FPS (7-60)

Event w

Annotations:

- 2. ... AND CLICK ON INDIVIDUAL OBJECTS (orange box with arrow pointing to Jet details)
- 1. YOU CAN SHOW INFORMATION ABOUT COLLECTIONS (blue box with arrow pointing to table)
- MUONS ARE 'COMPOUND OBJECTS': PHOENIX UNDERSTANDS THAT ID TRACKS, CLUSTERS, MUON TRACKS CAN BE LINKED (purple box with arrow pointing to red lines)

OVERLAY – CAN ROTATE WITH MAIN VIEW, OR BE FIXED AND SET TO ORTHOGRAPHIC. CAN BE DRAGGED AROUND.



DYNAMIC, REALTIME CLIPPING

SELECT EVENT DATA

i Collections Info ^

Choose a collection: Tracks

No.	chi2	dof	pos
#0	7.76711	13	0.781852,-3.38859,-13.5421,1.22476,-5.0
#1	16.9393	13	2.65908,-1.86285,-15.3481,4.24569,-2.7
#2	25.2923	12	0.162369,-3.30089,7.18604,0.217287,-4.9
#3	50.3791	29	1.62379,-2.87191,-13.4919,2.7659,-4.526
#4	69.3624	43	0.260677,-3.31365,-0.592984,1.05083,-8
#5	45.0789	44	0.85049,3.07486,-7.2364,1.262,4.647,-11.
#6	23.3833	15	2.52306,-4.08805,-20.759,2.83771,-4.57
#7	22.7328	13	2.61446,-2.53436,-20.4593,7.14388,-6.8
#8	5.21887	11	-2.88021,1.56283,16.5136,-8.8156,5.2422

Playground Geometry More

Geometries

Show

Wireframe

Toroids

TRT

SCT

Pixel

LAr Barrel

LAr EC1

LAr EC2

Tile Cal

Event Data

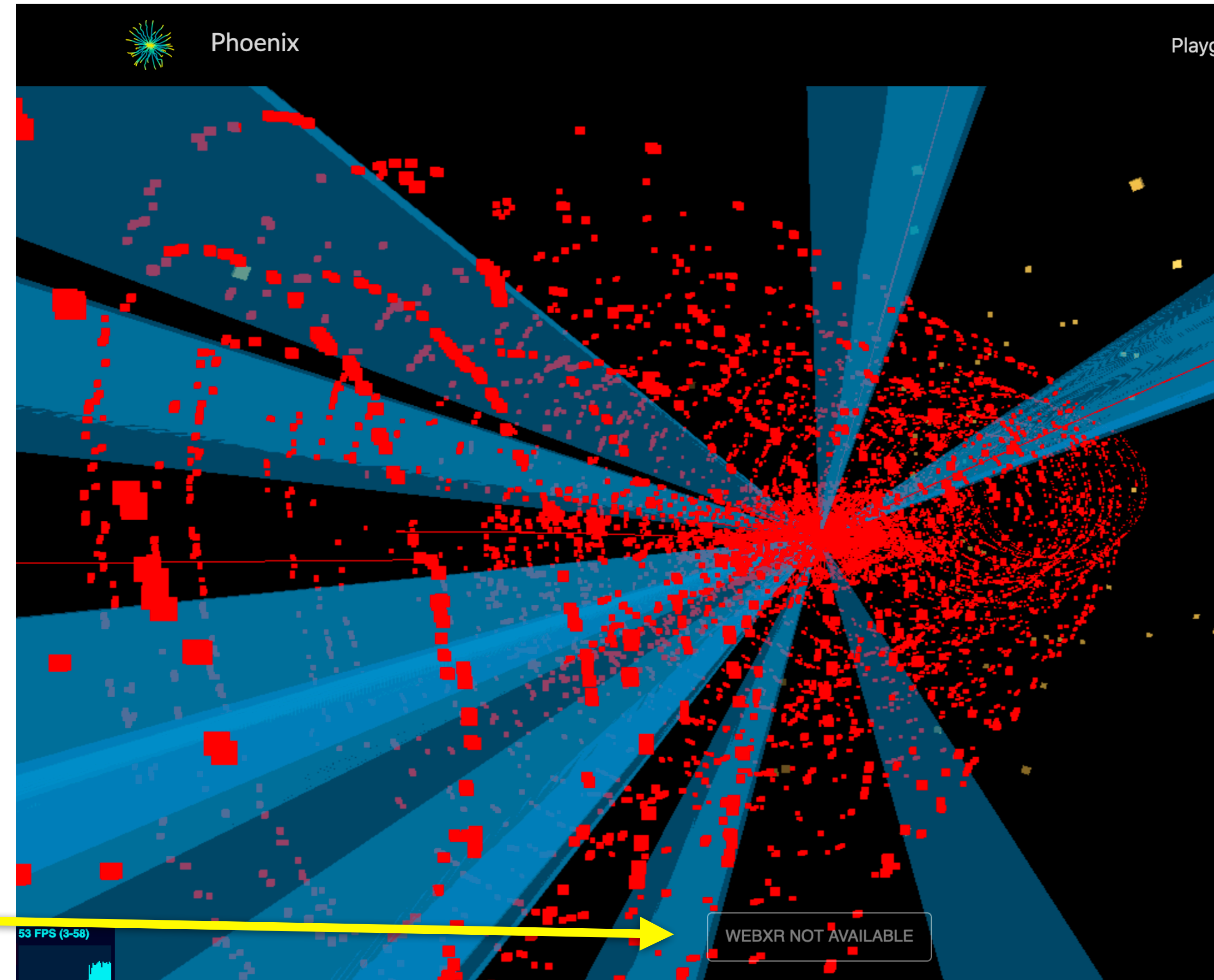
Close Controls

5 FPS (1-29)

Choose Event



- ▶ WebVR/XR is very new, and not widely supported yet
 - ▶ More info here: <https://immersiveweb.dev/>
- ▶ VR support within threejs is improving rapidly
 - ▶ Emilio a while back found a fix for a problem with angular/node and threejs
- ▶ Work underway to clean up menus etc for this - basically the 3D side is easy, trickier is interface
- ▶ How to view
 - ▶ WebVR doesn't work in Safari on iOS - but can install application e.g. VRBrowser
 - ▶ On PC, best is probably Firefox
 - ▶ **Currently not working (possibly https related)**



▶ We have track issues on the GitHub repository

▶ <https://github.com/HSF/phoenix/issues>

▶ Plus a gitter group

▶ <https://gitter.im/phoenix-developers/community>

☐ ⓘ 13 Open ✓ 30 Closed Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

- ☐ ⓘ **Line width issue for tracks** #73 opened on Mar 29 by 9inpachi 9
- ☐ ⓘ **Make UI configurable** enhancement #61 opened on Mar 11 by EdwardMoyses 1
- ☐ ⓘ **Add info panel** enhancement #60 opened on Mar 11 by EdwardMoyses 1
- ☐ ⓘ **Make overlay more functional** enhancement #57 opened on Mar 11 by EdwardMoyses
- ☐ ⓘ **Investigate opening ATLAS AOD files directly via jsroot** enhancement #44 opened on Jan 22 by EdwardMoyses
- ☐ ⓘ **Implement JiveXML loader for ATLAS** enhancement #43 opened on Jan 22 by EdwardMoyses
- ☐ ⓘ **Close sections created by clipping** enhancement #39 opened on Jan 21 by EdwardMoyses 1 9
- ☐ ⓘ **Improve track/objection selection by adding highlighting** enhancement #37 opened on Nov 7, 2019 by EdwardMoyses
- ☐ ⓘ **question: track parameters** #36 opened on Nov 6, 2019 by kratsg 3
- ☐ ⓘ **Add more unit tests to improve coverage** enhancement #27 opened on Jul 13, 2019 by emiliocortina
- ☐ ⓘ **Make it possible to switch perspective view off/on** enhancement #22 opened on Jun 24, 2019 by EdwardMoyses 7
- ☐ ⓘ **Implement analytic extrapolation** enhancement #11 opened on May 14, 2019 by EdwardMoyses
- ☐ ⓘ **Add animation** enhancement #4 opened on Mar 22, 2019 by EdwardMoyses 1

github.com/HSF/phoenix/issues?is=issues&open-label=enhancement

- ▶ Would like to do a few things for ATLAS
 - ▶ Make it possible to use phoenix as a library ... i.e. don't have homepage, just specially tweaked
 - ▶ Implement TypeScript RungeKutta propagator (and simple magnetic field)
- ▶ General list of tasks are in GitHub issues.
 - ▶ Comments and suggestions are welcomed!
- ▶ Lots of work is ongoing