ap^2

allpix squared

cern.ch/allpix-squared

Allpix Squared 2.0

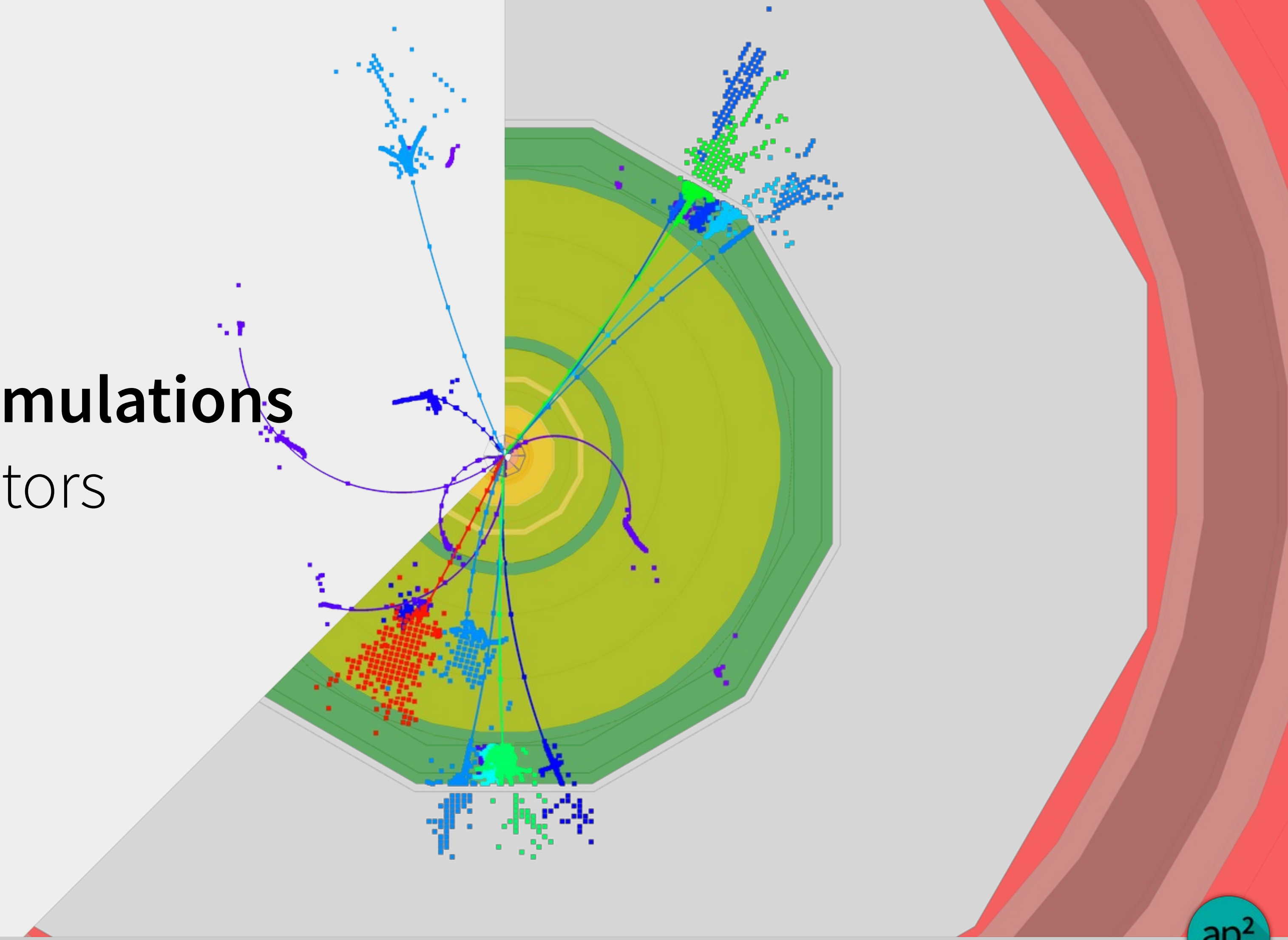
An Overview

Simon Spannagel, DESY

2nd Allpix Squared User Workshop

17 August 2021

Monte Carlo Simulations
of Silicon Detectors



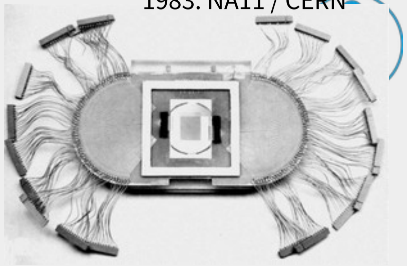
Silicon Detectors in Particle Physics

- Silicon detectors vital for many measurements
  - Fine segmentation, fast readout: high track multiplicities
  - Precise position measurement: momentum determination
    collision point (vertexing)
    particle identification (flavor tagging)

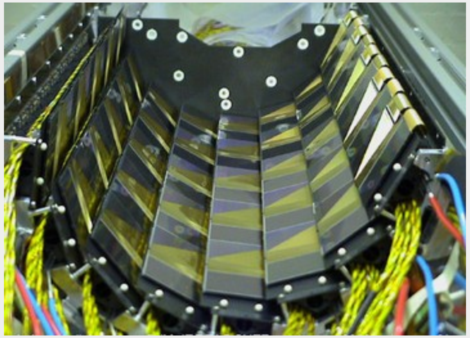
- Instrumental in discovery of Higgs boson at LHC
  - Tracking detectors: strips, 200 m^2 silicon, 70M channels
  - Vertex detectors: pixels, 1 m^2 silicon, 140M channels

- Detector R&D underway for
  - Upgrade of HL-LHC: more radiation damage resilience
  - Future colliders: *faster, higher, better*

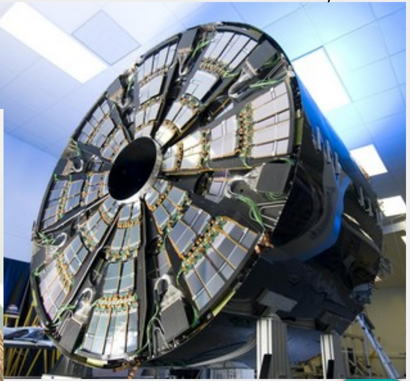
1983: NA11 / CERN



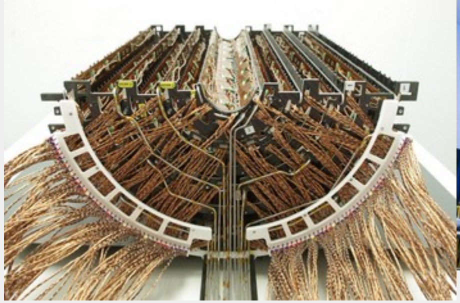
2000: ZEUS MVD / DESY



2007: CMS Tracker / CERN



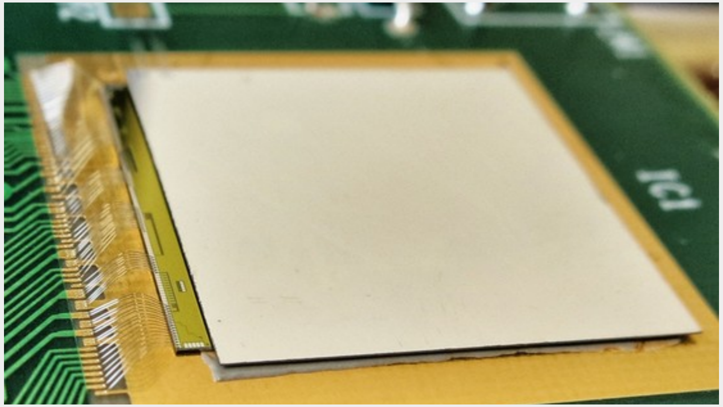
2017: CMS Phase 1 Pixel / CERN



Silicon Detectors in Particle Physics

Demands on detectors are high:

- Very high particle flux, tens of MHz / cm^2
- Maximum resolution, minimum (scattering-) mass
- Very high granularity for high particle rates,
  fast readout, minimal dead time (few ns)
- “Smart” detectors
  (zero suppression, clustering, on-chip processing, fast data links)

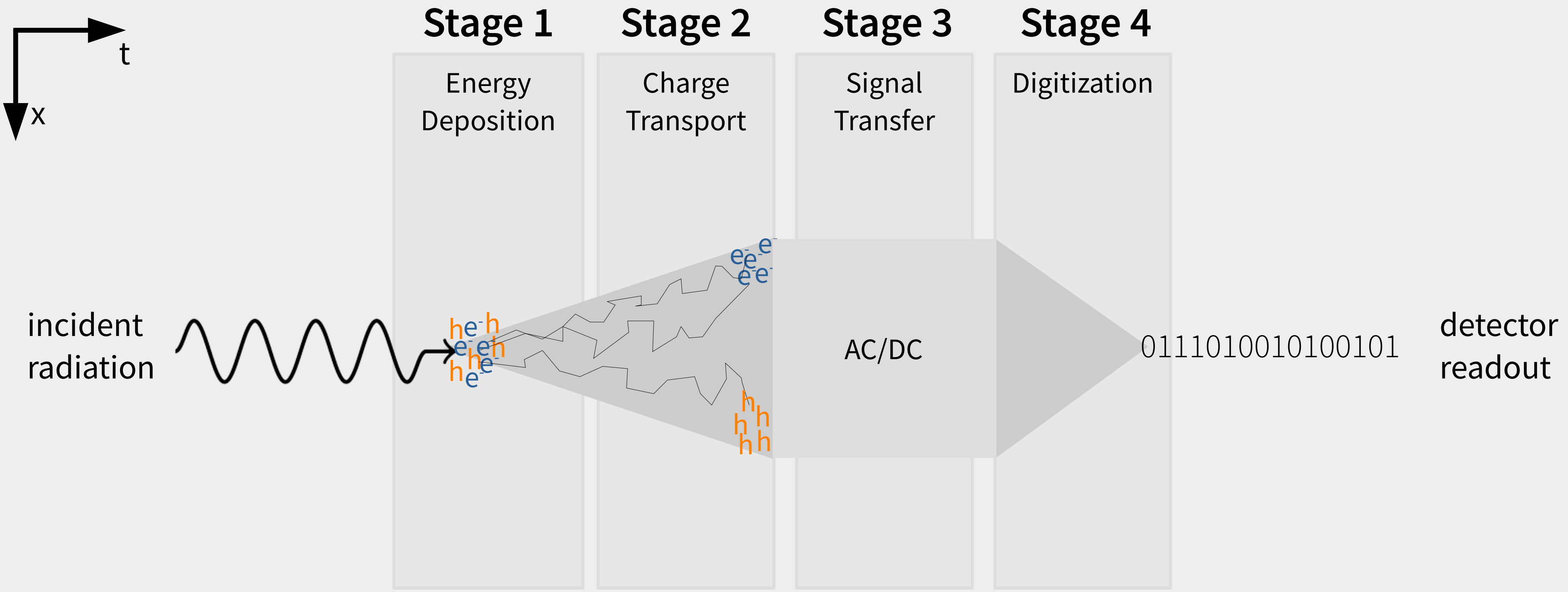


100 µm Timepix with 100 µm Sensor

Many different technologies used for different purposes:
hybrid – dedicated sensor + mixed-mode CMOS, monolithic CMOS imaging, LGADs, 3D sensors, …

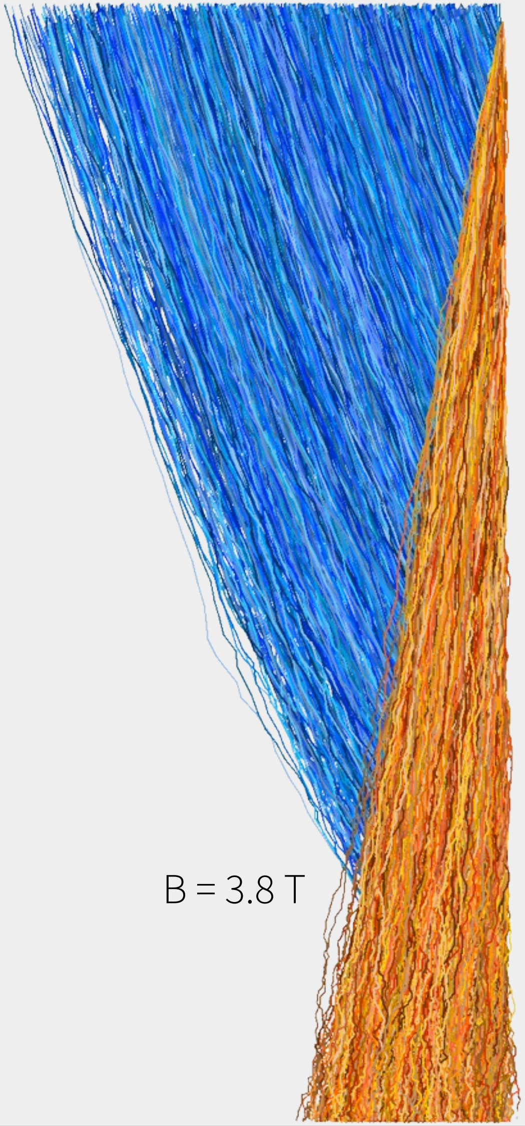
- Simulations required for thoroughly understanding detector performance in realistic conditions
- Tools needed to cover wide range of detector technologies

Minimum Ionizing Particle Detector – Broken Down



The Allpix Squared Framework

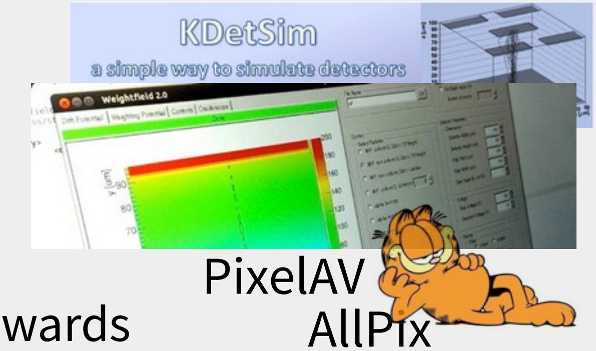
Monte Carlo Simulation



Yet Another Monte Carlo Simulation Framework?

Proliferation of many different codes for detector simulation:

- Some are experiment-specific
- Some are specialized on specific detector types
- Some are written as part of a PhD thesis and abandoned afterwards



PixelAV

AllPix

A new framework with…

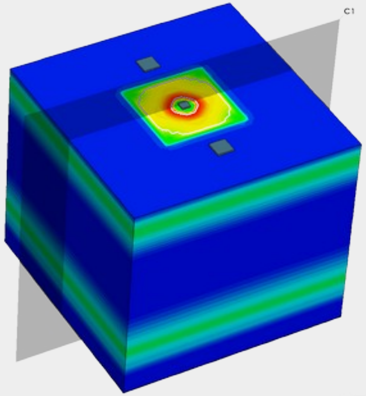
- Integration of Existing Toolkits
- Well-Tested & Validated Algorithms
- Low Entry Barrier for New Users
- Clean & Maintainable Code

Integration of Existing Toolkits

- Many very powerful tools developed and employed over decades of detector R&D
  Leverage their capabilities by providing interfaces for their integration

- **Geant4** – simulating energy deposition of particles passing through matter
  - Extensive toolkit, detailed simulation of many interactions & processes (e.g. decays)
  - Cumbersome to use for beginners, complexity often overwhelming at first
  - Provide abstraction layer that auto-generates models and calls Geant4 kernel

- **TCAD** – solving Poisson's equation using doping information
  - Detailed understanding of field configuration, sensor behavior
  - Tools & knowledge widely spread in community → see e.g. talk by J. Schwandt
  - Provide possibility to import results to complement MC simulations



Well-Tested & Validated Algorithms

- Simulations provide insights into physical processes – but only if they model them correctly! Validation of algorithms crucial and time-consuming process

- With Allpix Squared, we strive for
  - Validating as much as possible against known data
  - Publishing reference studies including full simulation configuration used
  - Providing automated test for every new feature

- More on automated testing: talk by me

Nuclear Inst. and Methods in Physics Research, A

Allpix²: A modular simulation framework for silicon detectors

S. Spannagel, K. Wolters, D. Hynds, N. Alipour Tehrani, M. Benoit, D. Dannheim, N. Gauvin, A. Nürnberg, P. Schütze, M. Vicente

NIMA 901 (2018) 164 – 172
doi:10.1016/j.nima.2018.06.020

Nuclear Inst. and Methods in Physics Research, A

Combining TCAD and Monte Carlo methods to simulate CMOS pixel sensors with a small collection electrode using the Allpix² framework

D. Dannheim, K. Dort, D. Hynds, M. Munker, A. Nürnberg, W. Snoeys, S. Spannagel

NIMA 964 (2020) 163784
doi:10.1016/j.nima.2020.163784

Combining Electrostatic Finite-Element and Transient Monte Carlo Simulations for the Optimisation and Characterisation of Silicon Sensors

In preparation…

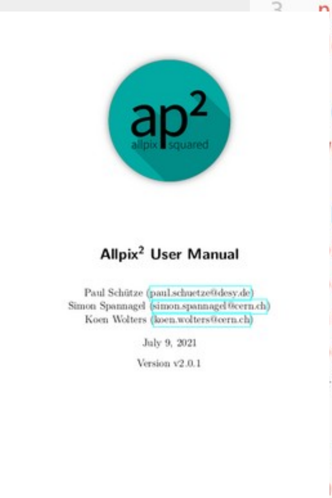
Low Entry Barrier for New Users

- Simulation frameworks often very complex: code complexity, lack of documentation, physics

- Allpix Squared attempts to facilitate quick starts:
  - Extensive documentation / user manual / help forum
  - Human-readable configuration files
  - Support for physical units
  - No coding or code-reading required

- Successfully used e.g. in university education, summer schools, …

  → See talk by P. Schütze

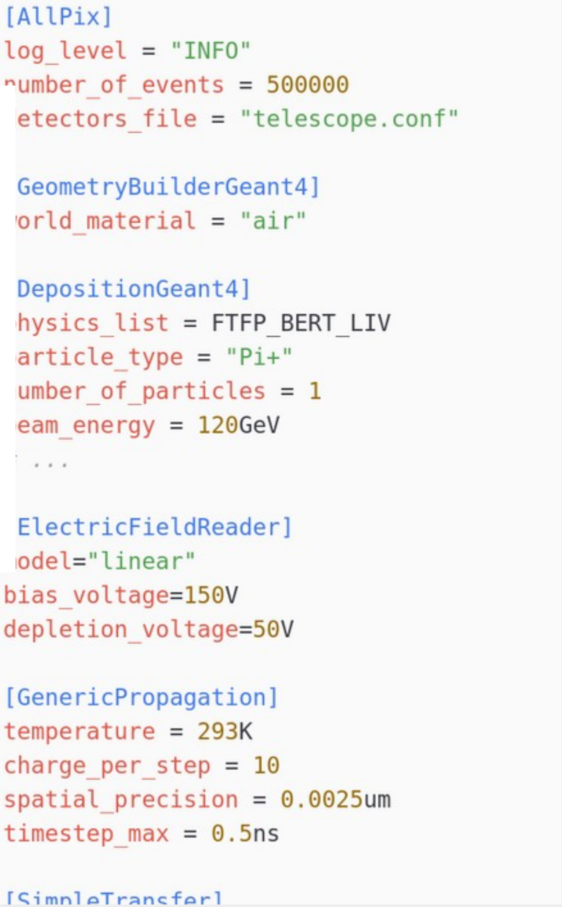


Allpix^2 User Manual

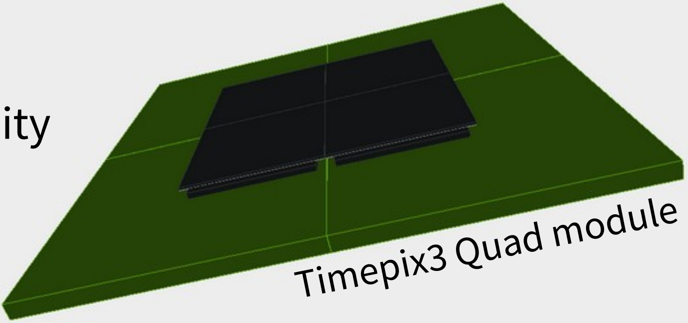
Paul Schütze (paul.schuetze@desy.de)
Simon Spannagel (simon.spannagel@cern.ch)
Koen Wolters (koen.wolters@cern.ch)

July 9, 2021

Version v2.0.1



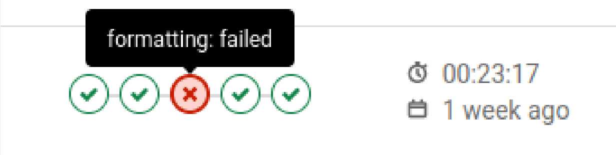
```
[AllPix]
log_level = "INFO"
number_of_events = 500000
etectors_file = "telescope.conf"

GeometryBuilderGeant4]
orld_material = "air"

DepositionGeant4]
hysics_list = FTFP_BERT_LIV
article_type = "Pi+"
umber_of_particles = 1
eam_energy = 120GeV
 ...

ElectricFieldReader]
odel="linear"
bias_voltage=150V
depletion_voltage=50V

[GenericPropagation]
temperature = 293K
charge_per_step = 10
spatial_precision = 0.0025um
timestep_max = 0.5ns

[SimpleTransfer]
```



Timepix3 Quad module

Clean & Maintainable Code

- Collaborative software development requires well-defined procedures – Otherwise quickly becomes unmaintainable

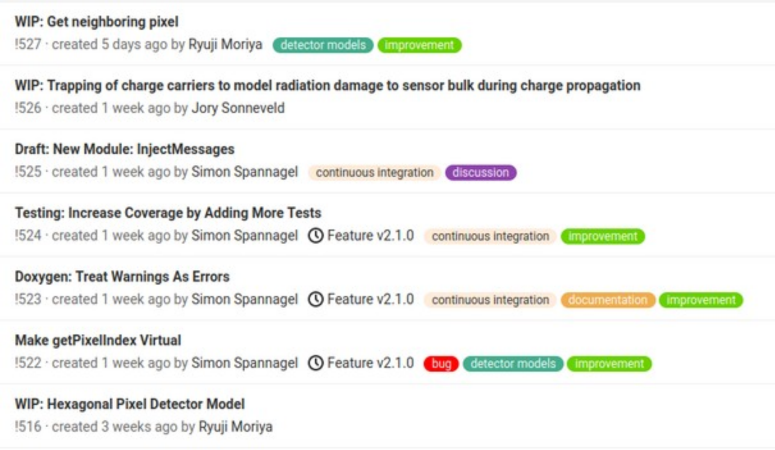


formatting: failed

00:23:17

1 week ago

- Allpix Squared implements *best practices* for software development
  - Permissive open-source license: MIT
  - Extensive code reviews via merge requests
  - Strict enforcement of coding conventions & form
  - Regular static code analysis



WIP: Get neighboring pixel
!527 · created 5 days ago by Ryuji Moriya detector models improvement

WIP: Trapping of charge carriers to model radiation damage to sensor bulk during charge propagation
!526 · created 1 week ago by Jory Sonneveld

Draft: New Module: InjectMessages
!525 · created 1 week ago by Simon Spannagel continuous integration discussion

Testing: Increase Coverage by Adding More Tests
!524 · created 1 week ago by Simon Spannagel Feature v2.1.0 continuous integration improvement

Doxygen: Treat Warnings As Errors
!523 · created 1 week ago by Simon Spannagel Feature v2.1.0 continuous integration documentation improvement

Make getPixelIndex Virtual
!522 · created 1 week ago by Simon Spannagel Feature v2.1.0 bug detector models improvement

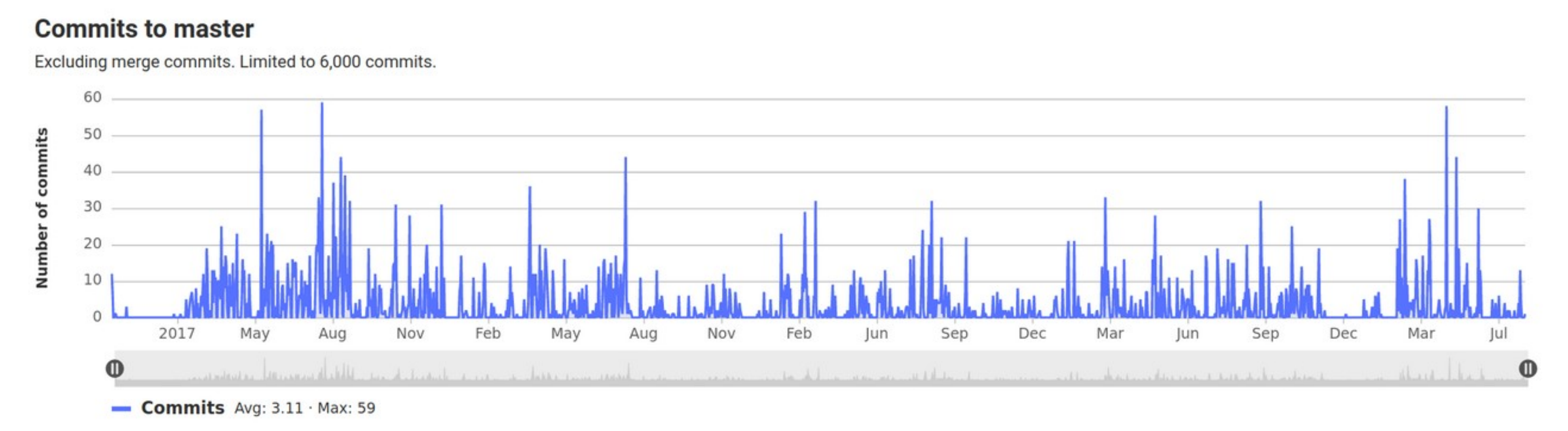
WIP: Hexagonal Pixel Detector Model
!516 · created 3 weeks ago by Ryuji Moriya

- More on collaborative coding: talk by me

A Brief History of Allpix Squared

- Started end of 2016 at CERN in EP-LCG group, now main development at DESY
- Development driven by technical student during 2017 (K. Wolters)
- First release: August 2017
  Ever since: continuous support / development / releases / improvements
- Many applications in different fields,
  By now 38 contributors, more soon to come (pending merge requests)

v2.0.1	2021-07-09
v2.0	2021-06-10
v1.6.2	2021-04-01
v1.6.1	2021-01-28
v1.6	2020-10-29
v1.5.2	2020-09-14
v1.5.1	2020-07-26
v1.5	2020-04-14
v1.4.4	2020-03-10
v1.4.3	2020-01-10
v1.4.2	2019-11-26
v1.4.1	2019-09-13
v1.4	2019-07-09
v1.3.4	2019-06-07
v1.3.3	2019-04-13
v1.3.2	2019-02-21
v1.3.1	2018-12-17
v1.3	2018-11-21
v1.2.3	2018-11-13
v1.2.2	2018-09-07
v1.2.1	2018-08-02
v1.2	2018-06-13
v1.1.2	2018-04-25
v1.1.1	2018-03-08
v1.1	2018-01-11
v1.0	2017-08-[illegible]



Release of Allpix Squared 2.0

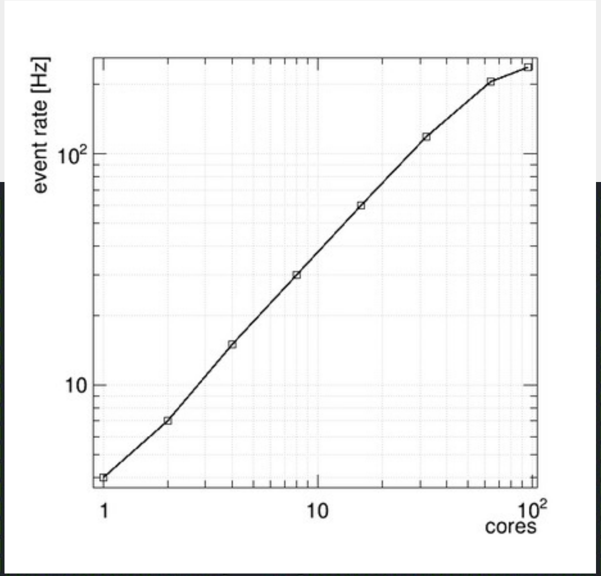
- First major release introducing structural changes to framework since 1.0 (08/2017)
  - More than 1500 commits over previous feature release 1.6
  - Introduced fully parallel event processing (Started as Google Summer of Code project)

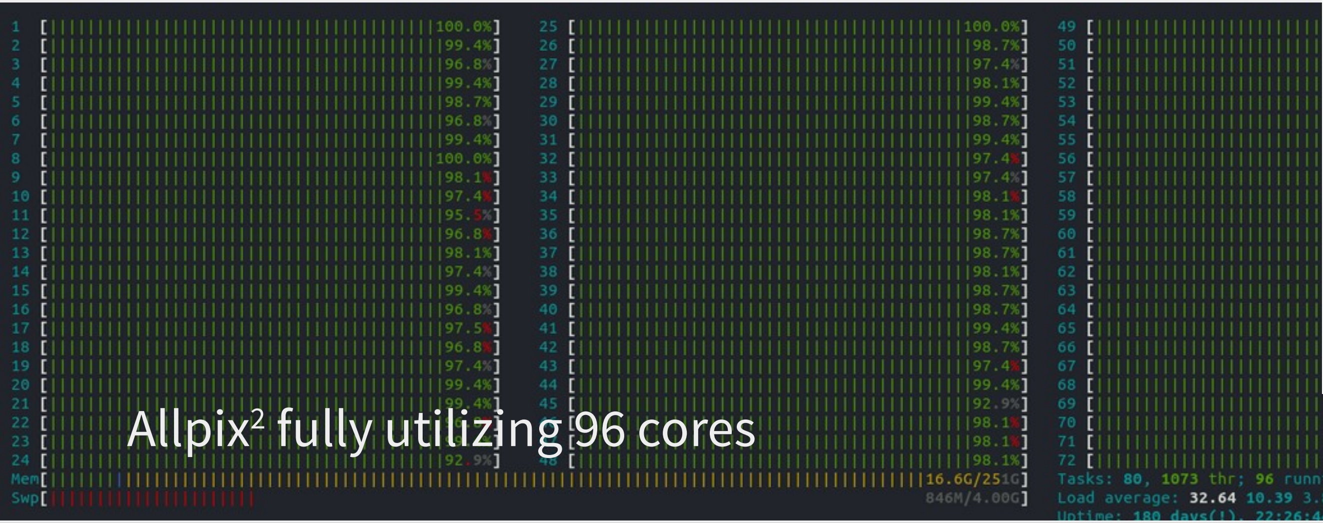
- Further separation between physics models & algorithms
- A few selected features presented in the following, separate talks on
  - Charge-Sensitive amplifier front-end → by A. Vauth
  - Multithreading → by K. Wolters

- Tons of small improvements, cleanup, documentation improvements: https://cern.ch/allpix-squared/post/2021-06-15-version-2.0.0/

Event-Based Seeding & Multithreading

- Efficient use of system resources / multiple cores
- Retaining strong reproducibility: exact same result, independent of # workers, fully transparent to user / simulation
- See talk by K. Wolters



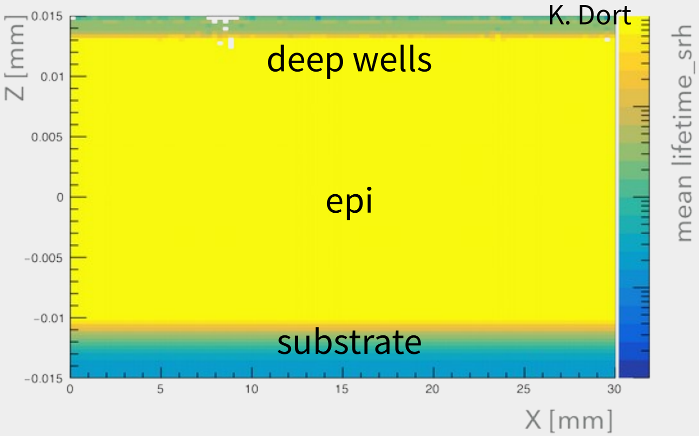


```
1  [||||||||||||||||100.0%]   25 [||||||||||||||||100.0%]   49 [||||||   ||||||97.4%]
2  [||||||||||||||||99.4%]    26 [||||||||||||||||98.7%]    50 [||||||   ||||||98.7%]
3  [||||||||||||||||96.8%]    27 [||||||||||||||||97.4%]    51 [||||||   ||||||99.4%]
4  [||||||||||||||||99.4%]    28 [||||||||||||||||98.1%]    52 [||||||   ||||||98.7%]
5  [||||||||||||||||98.7%]    29 [||||||||||||||||99.4%]    53 [||||||   ||||||97.5%]
6  [||||||||||||||||96.8%]    30 [||||||||||||||||98.7%]    54 [||||||   ||||||98.7%]
7  [||||||||||||||||99.4%]    31 [||||||||||||||||99.4%]    55 [||||||   ||||||97.4%]
8  [||||||||||||||||100.0%]   32 [||||||||||||||||97.4%]    56 [||||||   ||||||98.7%]
9  [||||||||||||||||98.1%]    33 [||||||||||||||||97.4%]    57 [||||||   ||||||98.1%]
10 [||||||||||||||||97.4%]    34 [||||||||||||||||98.1%]    58 [||||||   ||||||97.4%]
11 [||||||||||||||||95.5%]    35 [||||||||||||||||98.1%]    59 [||||||   ||||||100.0%]
12 [||||||||||||||||96.8%]    36 [||||||||||||||||98.7%]    60 [||||||   ||||||99.4%]
13 [||||||||||||||||98.1%]    37 [||||||||||||||||98.7%]    61 [||||||   ||||||97.4%]
14 [||||||||||||||||97.4%]    38 [||||||||||||||||98.1%]    62 [||||||   ||||||97.4%]
15 [||||||||||||||||99.4%]    39 [||||||||||||||||98.7%]    63 [||||||   ||||||98.1%]
16 [||||||||||||||||96.8%]    40 [||||||||||||||||98.7%]    64 [||||||   ||||||98.7%]
17 [||||||||||||||||97.5%]    41 [||||||||||||||||99.4%]    65 [||||||   ||||||98.1%]
18 [||||||||||||||||96.8%]    42 [||||||||||||||||98.7%]    66 [||||||   ||||||98.7%]
19 [||||||||||||||||97.4%]    43 [||||||||||||||||97.4%]    67 [||||||   ||||||98.1%]
20 [||||||||||||||||99.4%]    44 [||||||||||||||||99.4%]    68 [||||||   ||||||99.4%]
21 [||||||||||||||||99.4%]    45 [||||||||||||||||92.9%]    69 [||||||||||||||||96.8%]   93 [||||||||||||||||98.7%]
22 [||||||                    [||||||||||||||||98.1%]    70 [||||||||||||||||98.7%]   94 [||||||||||||||||98.7%]
23 [||||||                    [||||||||||||||||98.1%]    71 [||||||||||||||||94.2%]   95 [||||||||||||||||97.4%]
24 [||||||||||||||||92.9%]    48 [||||||||||||||||98.1%]    72 [||||||||||||||||98.7%]   96 [||||||||||||||||99.4%]
Mem[||||||||||||||||||||||||||||||||||||||||||16.6G/251G]   Tasks: 80, 1073 thr; 96 running
Swp[||||||||||||                               846M/4.00G]   Load average: 32.64 10.39 3.81
                                                             Uptime: 180 days(!), 22:26:44
```

Allpix² fully utilizing 96 cores

Recombination of Charge Carriers

- In many applications: fast signal formation
  no need for recombination – all e/h pairs reach electrodes
- Sometimes, finite charge carrier lifetime becomes interesting:
  - High-dopant regions
  - Low electric fields, signal formation via diffusion



- Allpix Squared supports position-dependent doping maps & lifetime calculation
  - Shockley-Read-Hall recombination: medium doping concentrations
  - Auger recombination: high doping concentrations
  - Combination

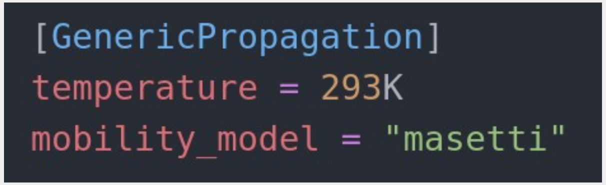
$$\tau^{-1}(N_d) = \begin{cases} \tau_{srh}^{-1}(N_d) + \tau_a^{-1}(N_d) & (minority) \\ \tau_{srh}^{-1}(N_d) & (majority) \end{cases}$$

- Application example: monolithic active pixel sensors, see talk by K. Dort

Different Carrier Mobility Models

- Providing different charge carrier mobility models
  - Field dependent
  - Doping concentration dependent
  - Optimized for high-field situations
  - …

- Description & reference provided in user manual
- Selected via configuration file:



```
[GenericPropagation]
temperature = 293K
mobility_model = "masetti"
```

$$\mu(E) = \frac{v_m}{E_c} \frac{1}{\left(1 + (E/E_c)^{\beta}\right)^{1/\beta}},$$

Jacoboni/Canali

$$\begin{aligned} \mu_e^{-1}(E) &= 1/\mu_{0,e} + E/v_{sat} && \\ \mu_h^{-1}(E) &= 1/\mu_{0,h} && E < E_0 \\ &= 1/\mu_{0,h} + b \cdot (E - E_0) + c \cdot (E - E_0)^2 && E \geq E_0 \end{aligned}$$

Hamburg

$$\mu_e(N) = \mu_{0,e} + \frac{\mu_{max,e} - \mu_{0,e}}{1 + (N/C_{r,e})^{\alpha_e}} - \frac{\mu_{1,e}}{(1 + (C_{s,e}/N)^{\beta_e})}$$

$$\mu_h(N) = \mu_{0,h} + \frac{\mu_{max,h}}{1 + (N/C_{r,h})^{\alpha_h}} - \frac{\mu_{1,h}}{(1 + (C_{s,h}/N)^{\beta_h})} + e^{P_c/N}$$

Masetti

$$\mu_e(N) = \mu_{min,e} + \mu_{0,e}/\left(1 + (N/N_{ref,e})^{\alpha}\right)$$

$$\mu_h(N) = \mu_{min,h} + \mu_{0,h}/\left(1 + (N/N_{ref,h})^{\alpha}\right)$$

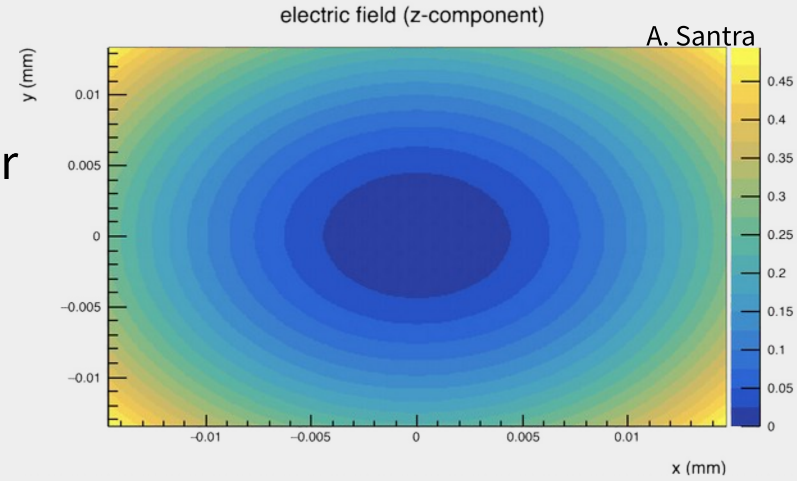
Arora

$$\mu(E, N) = \frac{\mu_m(N)}{\left(1 + (\mu_m(N) \cdot E/v_m)^{\beta}\right)^{1/\beta}}$$

Extended Canali/Masetti

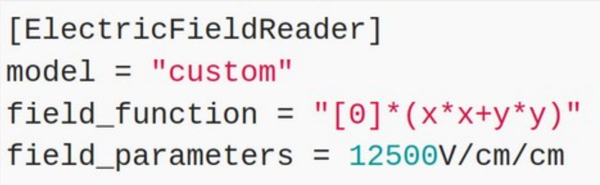
Custom Analytical Electric Field Functions

- From beginning on, implementations for
  - Linear electric fields → "standard" planar sensor
  - TCAD field maps → complex "known" sensors



A. Santra

- Often other models are required, now added:
  - Parabolic shape → double-peaked electric field after irradiation
  - Possibility for custom analytical field functions → approximations of complex sensors

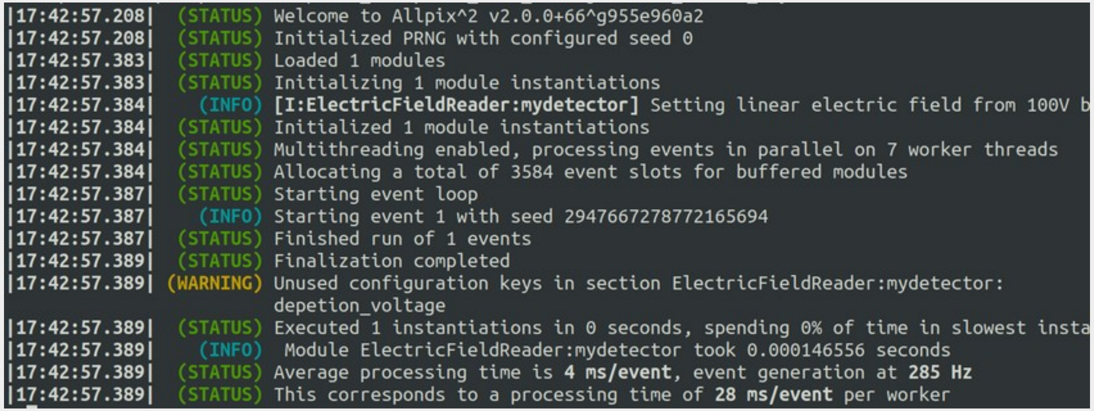


```
[ElectricFieldReader]
model = "custom"
field_function = "[0]*(x*x+y*y)"
field_parameters = 12500V/cm/cm
```

- Application example: approximating MAPS field, see talk by A. Santra

Unused Configuration Keys

- Small but handy feature…!
- Usage of configuration is tracked



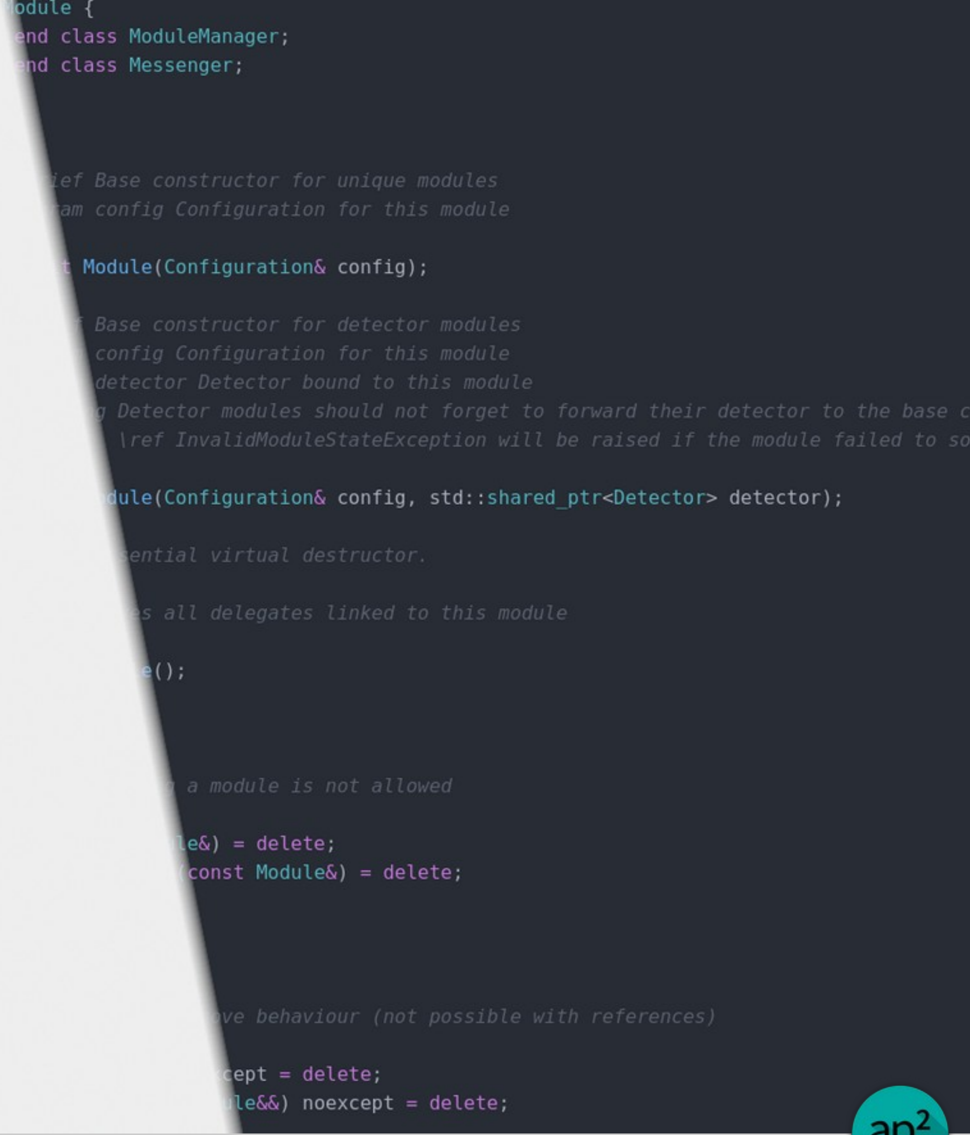
```
|17:42:57.208|   (STATUS) Welcome to Allpix^2 v2.0.0+66^g955e960a2
|17:42:57.208|   (STATUS) Initialized PRNG with configured seed 0
|17:42:57.383|   (STATUS) Loaded 1 modules
|17:42:57.383|   (STATUS) Initializing 1 module instantiations
|17:42:57.384|     (INFO) [I:ElectricFieldReader:mydetector] Setting linear electric field from 100V b
|17:42:57.384|   (STATUS) Initialized 1 module instantiations
|17:42:57.384|   (STATUS) Multithreading enabled, processing events in parallel on 7 worker threads
|17:42:57.384|   (STATUS) Allocating a total of 3584 event slots for buffered modules
|17:42:57.387|   (STATUS) Starting event loop
|17:42:57.387|     (INFO) Starting event 1 with seed 2947667278772165694
|17:42:57.387|   (STATUS) Finished run of 1 events
|17:42:57.389|   (STATUS) Finalization completed
|17:42:57.389|  (WARNING) Unused configuration keys in section ElectricFieldReader:mydetector:
                          depetion_voltage
|17:42:57.389|   (STATUS) Executed 1 instantiations in 0 seconds, spending 0% of time in slowest insta
|17:42:57.389|     (INFO)  Module ElectricFieldReader:mydetector took 0.000146556 seconds
|17:42:57.389|   (STATUS) Average processing time is 4 ms/event, event generation at 285 Hz
|17:42:57.389|   (STATUS) This corresponds to a processing time of 28 ms/event per worker
```

- User receives a **WARNING** for unused configuration keys at the end of the run – providing the possibility to
  - Clean up configuration files from unused parameters
  - Spot typos leading to wrong simulation results (`depetion_voltage != depletion_voltage`)

- Might be promoted to an **ERROR** in the future – let's see…

Ongoing Projects

and Developments



```
odule {
end class ModuleManager;
end class Messenger;



  ief Base constructor for unique modules
  am config Configuration for this module

  t Module(Configuration& config);

   f Base constructor for detector modules
    config Configuration for this module
    detector Detector bound to this module
    g Detector modules should not forget to forward their detector to the base c
     \ref InvalidModuleStateException will be raised if the module failed to so

    dule(Configuration& config, std::shared_ptr<Detector> detector);

     sential virtual destructor.

      es all delegates linked to this module

      e();



       g a module is not allowed

       le&) = delete;
       (const Module&) = delete;




        ove behaviour (not possible with references)

         cept = delete;
         le&&) noexcept = delete;
```

Ongoing Projects / Future Developments

- Alternative energy deposition modules

- More detector geometries
  - Hexagonal pixel layouts → see talk by R. Moriya
  - Radial strip geometries → see talk by R. Privara

- Other sensor materials, e.g. Germanium → see talk by T. Saleem
- 3D sensors
- Charge multiplication / LGAD detectors
- Radiation damage effects → see talk by J. Sonneveld

- Dedicated front-end modules, e.g. Timepix3 → see talk by P. Christodoulou

- Renovation of the user documentation

Summary

- Monte Carlo simulations:
  vital component of understanding & interpreting detector performance
- Allpix Squared:
  comprehensive MC simulation framework for silicon detectors
- Continuous development and support
  - New major release 2.0: multithreading capabilities & many new features
  - Regular patch releases with bug fixes
- Many new features already underway

New contributors always welcome!
Have a nice workshop!

Allpix Squared Resources

Website

https://cern.ch/allpix-squared

Repository

https://gitlab.cern.ch/allpix-squared/allpix-squared

Docker Images

https://gitlab.cern.ch/allpix-squared/allpix-squared/container_registry

User Forum:

https://cern.ch/allpix-squared-forum/

Mailing Lists:

allpix-squared-users https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10262858

allpix-squared-developers https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10273730

User Manual:

https://cern.ch/allpix-squared/usermanual/allpix-manual.pdf

DESY.